

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Bruno César Sales Alves

**Republic: Aplicação mobile para divulgar e
procurar vagas em repúblicas**

Uberlândia, Brasil

2017

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Bruno César Sales Alves

**Republic: Aplicação mobile para divulgar e procurar vagas
em repúblicas**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. André Ricardo Backes

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2017

Bruno César Sales Alves

Republic: Aplicação mobile para divulgar e procurar vagas em repúblicas

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 03 de agosto de 2017:

Prof. Dr. André Ricardo Backes
Orientador

Professor

Professor

Uberlândia, Brasil
2017

Dedico este trabalho aos meus pais, Edmar César e Sandra Maria e à minha irmã Mariana Rocha, pois esta vitória não é apenas minha, mas deles também. Que este trabalho seja objeto de estudo para futuros acadêmicos e interessados pela área de desenvolvimento de aplicativos. Em especial, dedico este trabalho ao meu grande amigo e professor Mauricio Cunha Escarpinati, por me aconselhar, me apoiar e me guiar em toda a minha trajetória acadêmica.

Agradecimentos

Quero agradecer primeiramente à Deus, pois sem fé, eu não iria a lugar algum. Agradeço profundamente aos meus pais Edmar César e Sandra Maria e à minha irmã Mariana Rocha, pois sem o apoio e incentivo deles, nada deste trabalho estaria feito, obrigado por estarem comigo em todos os momentos da vida. Agradeço também ao meu professor orientador André Ricardo Backes por acreditar em mim, pela responsabilidade e por sempre me nortear para os caminhos corretos da evolução do trabalho. Agradeço à minha namorada Luciana Pirete, pelo carinho e paciência comigo e por sempre me incentivar a concluir este trabalho. Ao meu grande amigo Vandrê Leal, por me ajudar em momentos difíceis, compartilhando de seu conhecimento técnico. Obrigado a todos, por me ajudar a realizar este sonho!

“O que fazemos na vida, ecoa na eternidade” - Russel Crowe (filme Gladiador)

Resumo

O presente trabalho apresenta um aplicativo *mobile* para ajudar universitários a encontrar ou disponibilizar vagas em repúblicas universitárias. A ideia da aplicação é ser uma camada intermediária, facilitando o processo entre a comunicação de quem procura e de quem fornece a vaga. Assim, o usuário evitará desperdiçar tempo procurando informações dispersas redes sociais, sem a necessidade de se locomover pelos bairros em busca de vagas próximas à Universidade. O aplicativo foi desenvolvido em um *framework* multiplataforma para que pudesse atingir usuários dos sistemas operacionais Android, iOS e Windows Phone. O Ionic foi o *framework* híbrido escolhido para o desenvolvimento. Como banco de dados, o Firebase foi escolhido, pois, fornece serviços fáceis de utilizar, como por exemplo a autenticação de usuários, armazenamento de dados em tempo real e interface intuitiva para o desenvolvedor. O design foi pensado especialmente seguindo conceitos de *User Experience* (UX), para que o usuário tenha uma melhor experiência navegando entre todas as telas, com maior fluidez e facilidade para encontrar o que deseja.

Palavras-chave: Aplicação Mobile. Multiplataforma. Usabilidade. Moradia. República.

Lista de ilustrações

Figura 1 – Screenshot da tela de encontrar repúblicas	14
Figura 2 – Screenshots do aplicativo Minha República	14
Figura 3 – Screenshot da tela inicial do website	16
Figura 4 – Planilha feita pela ARU para gerenciar as vagas em repúblicas	16
Figura 5 – Filtros de pesquisa no sistema Domuz	17
Figura 6 – Tela inicial do sistema Morar Unicamp	18
Figura 7 – Estrutura do aplicativo utilizando Ionic 2	21
Figura 8 – Estrutura de nós do Firebase	25
Figura 9 – Diagrama de Caso de Uso da aplicação	26
Figura 10 – Estrutura JSON dos dados que compõem a aplicação	27
Figura 11 – Estrutura JSON representada graficamente em forma de árvore	27
Figura 12 – <i>Wireframe</i> do projeto, mostrando o fluxo da aplicação	29
Figura 13 – Continuação do <i>wireframe</i>	30
Figura 14 – (a) Tela de splashscreen, (b) tela de <i>login</i> e (c) menu principal	31
Figura 15 – (a) Tela de cadastro da república, (b) tela de acomodações e (c) tela de despesas	32
Figura 16 – (a) Tela de moradias disponíveis, (b) tela de informações adicionais e (c) tela de chat	33
Figura 17 – (a) Tela de adicionar fotos, (b) tela de visualizar fotos e (c) a tela de edição da conta	34
Figura 18 – Faixa etária dos participantes da enquete	39
Figura 19 – Gênero dos participantes da enquete	39
Figura 20 – Participantes que saíram da cidade por conta da faculdade	40
Figura 21 – Nível de dificuldade para encontrar repúblicas para morar	41
Figura 22 – Nível de aceitação dos participantes caso houvesse um aplicativo que busca e divulga moradias	41
Figura 23 – Nível de utilidade do aplicativo para os estudantes	42
Figura 24 – Detalhes importantes que são avaliados quando os participantes vão alugar um quarto	42
Figura 25 – Como o participante encontrou vaga na moradia que reside atualmente	43
Figura 26 – Meio de comunicação em que o participante divulga vagas disponíveis	44

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
UX	<i>User Experience</i>
IDE	<i>Integrated Development Environment</i>

Sumário

1	INTRODUÇÃO	10
1.1	Objetivos	10
1.2	Método	11
2	TRABALHOS CORRELATOS	13
2.1	Reepub	13
2.2	Minha República	14
2.3	Associação de Repúblicas da Unicamp - ARU	15
2.4	Domuz	16
2.5	Morar Unicamp	17
3	DESENVOLVIMENTO	19
3.1	Tecnologias Utilizadas	19
3.1.1	Aplicativos Híbridos	19
3.1.2	Ionic Framework	20
3.1.3	TypeScript	23
3.1.4	HTML5/CSS3	23
3.1.5	Firebase	24
3.2	Atividades Desenvolvidas	25
3.2.1	Análise de dados e diagramas da aplicação	25
3.2.2	User Experience (UX) e explicação das telas	28
3.2.3	Componentes utilizados no desenvolvimento	34
4	RESULTADOS OBTIDOS	38
4.1	Estado Atual do Desenvolvimento	38
4.2	Testes Realizados	38
4.3	Trabalhos Futuros	44
5	CONCLUSÃO	45
	REFERÊNCIAS	46

1 Introdução

A urbanização com a posterior evolução técnico-científica geraram a crescente migração de pessoas para regiões em busca de melhores oportunidades de trabalho e de aperfeiçoamento da mão-de-obra. Além disso, o interesse de qualificação profissional por meio do ingresso das universidades foi cada vez mais crescente.

O processo de migração no contexto universitário é bastante conflitante na vida pessoal, uma vez que os universitários precisam se adaptar na dinâmica da nova região. Neste contexto, muitos estudos abordam este processo como desafio da inclusão social (ASSIS *et al.*, 2013).

Dentre estes, destacam-se os conflitos com adaptações a diferentes culturas, convívio com outras pessoas, distância do apoio familiar pessoal, exploração do ambiente e dificuldades com mobilidade entre as instituições com as moradias estudantis.

Alguns processos seletivos são tão dinâmicos e/ou burocráticos que o aluno não consegue ter tanto tempo para andar pela vizinhança em busca de um lugar para ficar.

Outro problema citado na reportagem do portal de notícias brasileiras G1 (G1, 2016) é a burocracia inicial de um processo de alugar um imóvel, pois, é necessário um fiador, e algumas das vezes, um depósito adiantado, sem contar da resistência de muitos proprietários que evitam alugar seu imóvel para estudantes, geralmente preferem famílias.

Diante deste cenário e da oportunidade de facilitar a procura por repúblicas, a proposta será desenvolver um aplicativo a fim de reduzir todas essas barreiras encontradas pelos estudantes, fornecendo uma aplicação simples, fácil e acessível para o público estudantil.

Além disso, o aplicativo denominado Republic, fornecerá melhor otimização de tempo e de esforço do estudante que busca a moradia, e este disponibilizará melhor divulgação de vagas de aluguel de inquilinos.

Assim, a partir da hipótese do bom funcionamento do aplicativo para buscas e cadastros de moradias, o trabalho proposto apresentará aos usuários uma forma prática e rápida de aplicação que agrupe todas as opções de moradias disponíveis em determinada região e que facilite para o fornecedor da vaga divulgar sua república.

1.1 Objetivos

O objetivo geral do trabalho proposto é desenvolver um aplicativo mobile híbrido para buscar e divulgar as vagas em repúblicas universitárias. Os objetivos específicos do

trabalho proposto são:

1. Agrupar informações das repúblicas em um só lugar, para que o usuário não precise procurar vagas em grupos de redes sociais;
2. Permitir uma forma de comunicação simples e direta entre os interessados pela vaga da república.

1.2 Método

Nesta seção, serão abordadas algumas funcionalidades do aplicativo, ou melhor, seus requisitos funcionais de uma forma geral. Além disso, serão apresentados, brevemente, os motivos pela escolha das ferramentas utilizadas para o desenvolvimento do projeto.

Abaixo estão listados todos os requisitos funcionais assumidos na aplicação:

1. O usuário deve realizar um cadastro para utilizar a aplicação;
2. O usuário deve ser capaz de visualizar toda a lista de vagas disponíveis;
3. O usuário deve ser capaz de realizar o cadastro de sua república, além de poder editá-la ou removê-la, assim que a vaga tiver sido ocupada;
4. Os usuários poderão se comunicar, através de um chat, para saber maiores informações e até mesmo negociar uma possível visita à república;
5. Cada endereço de república deverá ser associado a um ponto no mapa que será exibido ao usuário;
6. Será possível fazer *upload* de imagens da república;
7. Será possível realizar *login* através da conta do Facebook;
8. O usuário poderá a qualquer momento deletar a sua conta no aplicativo.

Como não foi utilizado nenhuma linguagem compilada, então não houve necessidade de utilizar nenhuma IDE (Ambiente de Desenvolvimento Integrado), porém, para a escrita do código, foi utilizado o editor de texto Atom, um software gratuito contendo milhares de pacotes que auxiliam o desenvolvedor na hora de codificar sua aplicação. Para o desenvolvimento deste aplicativo, foi utilizado os seguintes pacotes, disponíveis na própria loja do Atom:

1. *atom-beautify*, um pacote que auto-formata o código, melhorando a leitura e compreensão;

2. `atom-typescript`, responsável por interpretar e diferenciar as palavras reservadas da linguagem Typescript;
3. `auto-indent`, capaz de indentar todo o código sem a necessidade de se preocupar em organizar o espaçamento do código;
4. `autoclose-html`, pacote responsável por fechar as *tags* da linguagem de marcação HTML;
5. `busy-signal`, é ele quem mostra, visualmente, se uma tarefa está processando, o que facilita ao desenvolvedor saber o que está sendo executado no momento;
6. `linter`, é um dos pacotes mais importantes, ele é responsável por detectar alguns erros, principalmente de sintaxe, ou até mesmo acusar *warnings* que passam despercebidos pelo programador.

Por fim, foi escolhido o banco de dados Firebase, sustentado pela Google, que é um banco de dados não relacional poderoso, foi a melhor escolha para este projeto pelo fato de ser um banco capaz de comportar tudo o que foi preciso, pela facilidade de gerenciar e trafegar os dados entre a aplicação. O Firebase será melhor discutido também no capítulo Desenvolvimento.

2 Trabalhos Correlatos

Neste capítulo, serão abordados alguns trabalhos correlatos que serviram de suporte para o desenvolvimento deste Trabalho de Conclusão de Curso.

2.1 Reepub

O Reepub é uma plataforma que funciona de forma colaborativa, onde usuários utilizam o site para buscar e fornecer vagas em repúblicas (REEPUB, 2017). A ideia do site é basicamente realizar o intermédio entre o fornecedor e o interessado pela vaga.

Segundo o cofundador, Mailson Dutra Soares, sempre foi um grande problema encontrar vagas em repúblicas desde sua adolescência, foi então que ele percebeu a oportunidade de criar o Reepub, visto que procurar vagas em grupos do Facebook não era tão eficiente (GLOBO, 2015). Sua plataforma foi a pioneira neste tipo de fornecimento de serviço.

É possível notar algumas vantagens ao se utilizar o site. Uma delas é a facilidade de realizar o cadastro da república. Em uma única página o usuário consegue carregar as fotos, inserir informações básicas sobre a moradia, colocar observações do tipo: número de vagas e tipo de república (público-alvo). Outro ponto positivo é a possibilidade de trocar mensagens com os fornecedores das vagas, pois assim, é possível tirar dúvidas e obter mais informações sobre a república.

Em uma breve pesquisa através do campo de busca do site fora constatado que a ferramenta tem sido mais utilizada em algumas cidades da região sudeste do Brasil, (por exemplo, a Figura 1, mostrando as repúblicas da cidade de São Paulo). Ainda não se popularizou, mas tem potencial de se destacar no mercado, visto que está funcionando desde 2015. A proposta inicial era que o site se tornasse um aplicativo mobile, porém, está disponível apenas na versão web.

Alguns pontos vistos como negativos seria o cuidado com a responsividade do site e a usabilidade em relação aos componentes da plataforma. Um outro cuidado com informações antigas também deveria ser tomado, pois, ainda é possível encontrar vários anúncios antigos (por exemplo do ano de 2015), ou seja, as vagas já podem ter sido ocupadas, porém, ainda se mantém visível para quem está procurando alguma moradia.

Uma vantagem do aplicativo apresentado neste trabalho em relação ao Reepub é a mobilidade. Por ser uma plataforma híbrida, é possível que o usuário utilize Android, iOS ou Windows Phone, evitando a necessidade de acessar uma página web através de um computador.

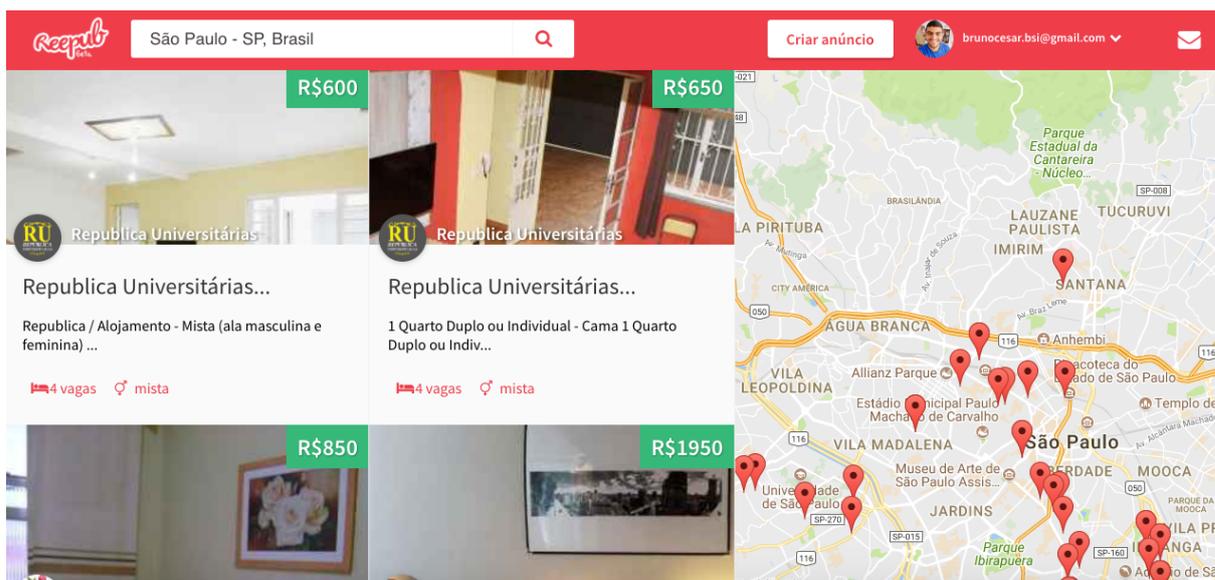


Figura 1 – Screenshot da tela de encontrar repúblicas

2.2 Minha República

Minha República é um aplicativo criado por estudantes universitários que perceberam a oportunidade e a grande dificuldade que os alunos de Sobral, município do Ceará, aprovados no vestibular, tinham em encontrar moradia.



Figura 2 – Screenshots do aplicativo Minha República

A startup “Minha República” foi uma iniciativa fomentada pelo programa Cor-

redores Digitais, sustentado pelo governo do Ceará, o qual incentiva e fornece todos os recursos necessários para os jovens abrirem seus próprios negócios na área de Tecnologia da Informação e Comunicação (CEARÁ, 2016b).

O aplicativo teve bastante repercussão no ano de 2016, pois, foi escolhido para participar da Campus Party, o maior evento tecnológico do mundo, que expõe novidades científicas, inovadores, empreendedoras e criativas (CEARÁ, 2016a).

Pesquisas foram realizadas nas lojas de aplicativos da App Store e Play Store, o “Minha República” não encontra-se disponível no momento para uma melhor avaliação, porém, algumas imagens do aplicativo foram encontradas em (APKPURE, 2015), como podemos ver na Figura 2.

2.3 Associação de Repúblicas da Unicamp - ARU

Trata-se de um website que reúne informações sobre todas as repúblicas da Unicamp, como podemos ver na Figura 3. O usuário pode consultar vagas disponíveis, eventos próximos, cadastrar sua república e ficar por dentro de todas as novidades (UNICAMP, 2016). Resumindo, é um portal voltado especificamente para repúblicas.

A forma que são divulgadas as vagas é através de uma planilha do Google Sheets, como exemplo, a Figura 4. É uma forma simples e rápida de consultar a situação atual de cada moradia. A planilha conta com a divisão por tipos de repúblicas, sendo, mista, masculina ou feminina. Além disso, há uma coluna na planilha para preencher observações, algumas vagas são exclusivamente para alunos de determinados cursos. Caso o interessado tenha vontade de saber maiores informações sobre a moradia, ele deve enviar mensagem ou ligar para o representante, essa é a forma de comunicação que o site fornece.

Um fato interessante sobre a ARU é a preocupação que tem para com os alunos. A cada ano, a organização (formada pelos próprios alunos) desenvolve um livreto com informativos sobre os eventos, por exemplo, campeonatos esportivos, festas, reuniões e estatísticas sobre as repúblicas (dados sobre valores, quantidades de moradores, ano de fundação, entre outros).

A vantagem de utilizar o *website* da ARU é que o estudante não se perde com muitas informações, pois todas as vagas do site são direcionadas à Unicamp, ou seja, o aluno não perderá tempo filtrando moradias de localidades diferentes. A desvantagem é que utilizando uma planilha para controlar as vagas, sempre haverá necessidade de pessoas por trás do sistema, para organizar e atualizar a planilha.



Figura 3 – Screenshot da tela inicial do website

Relação de Vagas - Repúblicas

REPÚBLICAS	TIPO DE REPÚBLICA	VAGAS DISPONÍVEIS	CONTATO	OBSERVAÇÕES
República do Patrão	Masculina		(19) 99699-1444 (Vitor) (35) 8849-1605 (Thales) https://www.facebook.com/RepDoPatrão/?ref=ts	
República Paidabarti	Masculina	1	(19) 991702096 (Babidi); (11) 981082482 (Jacaré)	Engenharias, Economia e Estat
República Panela	Masculina	2	(11) 974080784 - Gustavo; (11) 982421327 - Rodrigo;	1 vaga p/ 017, 1 vaga intercambist
República Zer071	Masculina	1	(13) 99607-0420 - Rivo (11) 99472-8706 - Pepeka	
Sapecasa	Masculina			
Tangamandápio	Masculina			
Traz o Martelo	Masculina		(41)992426024	
Viracopos	Masculina	3	(19) 99731-9786 - Shao e (19) 97122-7687 - Didi	
Vaca Mucha	Masculina	1		
Bjos Me Liga	Masculina		(11)972718733, (19)971041998	
Rep Our	Masculina	4	(13) 99773-6177 - Amin) e (11) 95349-2282 - Kardec	
SantoMê	Masculina		(19)9981261627 - Cantia, https://www.facebook.com/repasantome/?ref=ts	
Rep X	Masculina	2	(19) 98120-4800 - Testa // (19) 981584244 - Wallace // https://www.facebook.com/reppublicax/	
Casa Amarela	Masculina	2 em duplo	https://www.facebook.com/martinsf89?ref=ts	
3 Pinheiros	Feminina	1	(11) 98275-7897	Somente dança
Casa das Primas	Feminina	1		

Figura 4 – Planilha feita pela ARU para gerenciar as vagas em repúblicas

2.4 Domuz

O Domuz foi fundado em 2013, com objetivo de integrar usuários anunciantes de vagas e estudantes procurando por repúblicas, no estado de São Paulo. A ideia é que usuários pudessem utilizar não só em computadores, mas também em dispositivos móveis (DOMUZ, 2016).

O Domuz é um buscador muito parecido com o Reepub. A diferença é a preocupação com a experiência do usuário, onde é possível perceber ao usar um dispositivo móvel,

o sistema é responsivo, facilitando a navegação em aparelhos eletrônicos menores. Outra diferença é visível na tela de pesquisa, como mostra a Figura 5, pois, é possível filtrar os resultados por preço, o usuário consegue definir um intervalo de valores que ele esteja disposto a pagar, de forma intuitiva. Além disso, o estudante pode filtrar por nome de Universidades. Essas funcionalidades são de grande ajuda para o usuário.

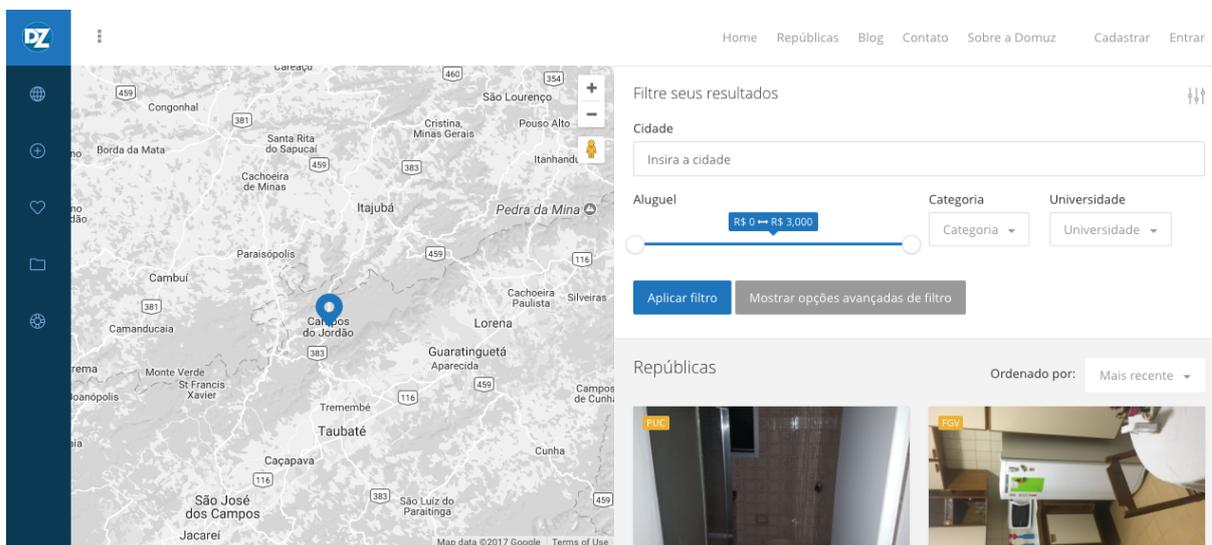


Figura 5 – Filtros de pesquisa no sistema Domuz

2.5 Morar Unicamp

Os alunos criadores da plataforma Morar Unicamp contam que sempre sentiram uma certa deficiência na relação imobiliária e universitário e que a busca por moradias era sempre um empecilho para os estudantes (UNICAMP, 2017). Foi então que surgiu o interesse em criar um sistema de busca por repúblicas, tanto para quem fornece e para quem procura vagas.

O sistema é funcional, simples e eficiente. Possui uma interface que não segue padrões de usabilidade, como vemos na Figura 6, pode confundir alguns usuários, em contrapartida, possui recursos interessantes, que não foram encontrados em outros trabalhos correlatos, que é a separação por tipo de moradia (república, pensionato, casa de família, kitnet, flat, entre outros). Cada tipo de moradia possui uma legenda no mapa, para que o usuário possa distinguir melhor qual é o tipo de moradia que ele procura.

Outra facilidade é que não é necessário de realizar um cadastro prévio do usuário para utilizar o sistema, ou seja, é possível visualizar todas as vagas sem o compromisso de se inscrever no site.

Por fim, outra diferença notável no sistema são as políticas de uso. O sistema é gratuito apenas para estudantes, já às imobiliárias, donos de aluguéis (pensionatos, kitnets

e similares) que divulgam vagas, devem pagar uma taxa simbólica ao sistema, esta taxa ajuda na manutenção do site (UNICAMP, 2017).

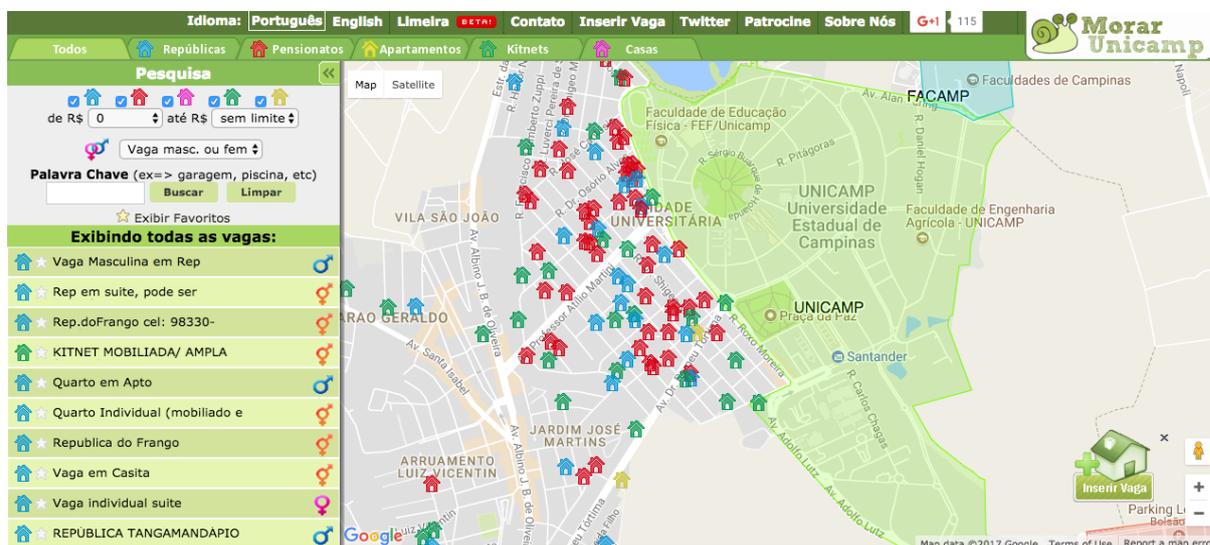


Figura 6 – Tela inicial do sistema Morar Unicamp

3 Desenvolvimento

Neste capítulo, serão abordados as tecnologias utilizadas para o desenvolvimento da aplicação, tais como, linguagem, framework e banco de dados. Alguns diagramas serão apresentados, para facilitar a visualização e entendimento do trabalho proposto.

3.1 Tecnologias Utilizadas

3.1.1 Aplicativos Híbridos

Segundo o trabalho de (SERRANO; HERNANTES; GALLARDO, 2013), aplicativos híbridos são aplicações *web mobile* encapsuladas em uma aplicação nativa. Os aplicativos híbridos funcionam e se comportam como um aplicativo nativo, possuem as mesmas capacidades e recursos de uma aplicação nativa. A diferença é que aplicações híbridas são construídas com as mesmas ferramentas e linguagens que uma aplicação web, ou seja, são baseadas em HTML, CSS e JavaScript.

Os aplicativos híbridos oferecem uma importante troca de recursos. O aplicativo é capaz de utilizar recursos nativos e ter acesso à APIs providas pelo *smartphone*. É ciente que há um custo nessa troca de recursos, visto que esses serviços serão chamado através das tecnologias web envolvidas (SERRANO; HERNANTES; GALLARDO, 2013). De acordo com (CORPORATION, 2012), a porção nativa da aplicação utiliza a API do sistema operacional para criar um mecanismo de renderização do HTML, ou seja, uma ponte entre o navegador e as APIs do dispositivo é criada, permitindo essa troca de recursos.

O motivo da escolha por desenvolvimento híbrido foi essa flexibilidade que é proporcionada na criação das interfaces utilizando tecnologias web. Desta forma, é possível que se tenha o mesmo aplicativo suportado em várias plataformas móveis, por exemplo Android, iOS e Windows Phone. (DALMASSO et al., 2013). Em (XANTHOPOULOS; XINO GALOS, 2013) são realizados alguns testes de compatibilidade e o resultado é confirmado, que o mesmo código escrito em JavaScript funcionou tanto para Android quando para iOS.

Quando se desenvolve para múltiplas plataformas ao mesmo tempo, o tempo e o custo de desenvolvimento é reduzido (DALMASSO et al., 2013). Este foi outro motivo pela escolha por aplicativos híbridos, o custo e o tempo de desenvolvimento é bem reduzido, comparado ao desenvolvimento nativo, visto que, cada plataforma tem uma linguagem específica para se escrever um aplicativo. Sem contar que existem outros custos mantendo várias plataformas. Em (SERRANO; HERNANTES; GALLARDO, 2013), nota-se a pre-

ocupação com os gastos por plataformas específicas, pois exige treinamento da equipe e novos recursos para o desenvolvimento, são problemas que consomem tempo e dinheiro.

Segundo, (MALAVOLTA et al., 2015), o código utilizado no Android, escrito em Java, não pode ser reutilizado no iOS, que é feito em Objective-C, tornando a manutenção dos aplicativos nativos um dos principais desafios técnicos para a comunidade de desenvolvimento.

Muitos analistas preveem que seja provável que o desenvolvimento híbrido se torne padrão para desenvolvimento de aplicativos *client-side*, visto que HTML5 tem crescido bastante. Ao escrever um código, se o HTML for utilizado na maior parte e o código nativo apenas quando necessário, as empresas podem garantir que o investimento utilizado não se tornem tão obsoletos, visto esse rápido crescimento das tecnologias web (CORPORATION, 2012).

3.1.2 Ionic Framework

O framework escolhido para o desenvolvimento deste trabalho foi o Ionic, que é o framework com muitos recursos e muito utilizado no desenvolvimento de aplicações móveis. O Ionic nada mais é que a fusão das tecnologias web, mais especificamente *front-end* (JavaScript, TypeScript, HTML e CSS) e recursos nativos do *smartphone* (VIEBRANTZ; CAMPOS,). A comunicação com os recursos nativos é dada através do Cordova, que é um intermediário entre as chamadas do código com as chamadas nativas do dispositivo móvel (LOPES, 2016). Além desses recursos, o Ionic segue guidelines de design e provê todo um conjunto de componentes (VIEBRANTZ; CAMPOS,) capazes de deixar o visual do aplicativo totalmente agradável e bonito.

Atualmente, o Ionic encontra-se na versão 2, deixando de lado alguns padrões que eram predominantes da versão 1. Na versão 1 do Ionic, são utilizados conceitos do Angular ¹ e sua linguagem predominante é JavaScript. Já o Ionic 2 utiliza paradigmas do Angular 2, que já utiliza como linguagem predominante o TypeScript (IONIC, 2017). Apesar dos paradigmas terem sido alterados, o Ionic 1 ainda continua sendo muito utilizado pelos desenvolvedores. Um dos motivos é a facilidade do JavaScript e por estar no mercado há tempo. Neste projeto, foi escolhido o Ionic 2 para o desenvolvimento.

As vantagens de se utilizar Ionic 2 são várias, mas podemos citar como principais:

1. A organização estrutural do código. Cada componente ou página no Ionic 2 é organizada em pastas, facilitando o entendimento e a localização fácil para o desenvolvedor, como podemos ver na Figura 7

¹ Angular é uma plataforma para desenvolvimento de aplicativos que utiliza tecnologias web. O Angular é um conjunto de boas práticas de programação que combina templates declarativos, injeções de dependências, de forma bem organizada

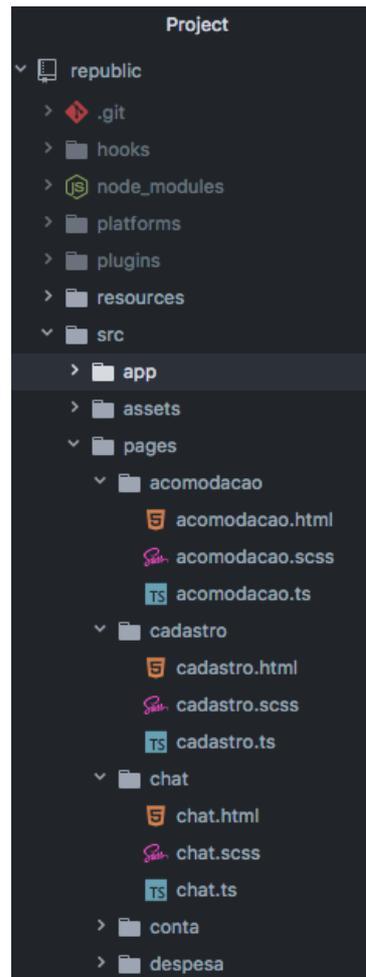


Figura 7 – Estrutura do aplicativo utilizando Ionic 2

que permite organizar a aplicação através de módulos.

Com a modularização do sistema, temos as vantagens de isolarmos melhor nosso código fonte, assim como deixar o código mais fácil para manutenção e implementação de novos recursos.

2. Organização das dependências da aplicação utilizando o `@NgModule`¹, no lugar de declará-las várias vezes dentro da aplicação (HARTINGTON, 2016):

```
@NgModule({
  declarations: [
    MyApp,
    AcomodacaoPage,
    CadastroPage,
    ChatPage,
```

¹ NgModule permite organizar a aplicação através de módulos e a vantagem é o isolamento do código, que facilita sua manutenção

```
        //etc...
    ]
})
```

3. Utilização do Ionic CLI, que é um gerador de componentes para o desenvolvedor, através de linhas de comando é possível gerar páginas, *providers*, *tabs*, *pipes*, componentes e diretivas. Além disso, irá organizar todos esses arquivos na estrutura do projeto. O código abaixo é um exemplo de utilização do Ionic CLI, em que o comando **generate** gera um componente chamado “modal”, na segunda linha de comando, gera uma página chamada “login”, que facilita para o desenvolvedor, pois todas as dependências e arquivos necessários são criadas automaticamente com esse comando.

```
$ ionic generate component modal
$ ionic generate page Login
```

4. Facilidade em fazer chamadas para outras telas. A navegação pelo Ionic 1 era feita através de rotas URLs, em um arquivo de configuração no qual era necessário especificar o caminho da página na estrutura, qual era o *template* da tela (arquivo HTML) e qual *controller*¹ gerencia a tela.

```
.config(function($stateProvider, $urlRouterProvider) {

    $stateProvider
        .state('intro', {
            url: '/',
            templateUrl: 'templates/intro.html',
            controller: 'IntroCtrl'
        })
        .state('main', {
            url: '/main',
            templateUrl: 'templates/main.html',
            controller: 'MainCtrl'
        });

    $urlRouterProvider.otherwise("/");
```

¹ componente responsável por fazer a ligação entre a camada de visualização com os dados da aplicação

```
});
```

Já no Ionic 2, é utilizado o método de *push* e *pop* (empilhar e desempilhar as telas), que por sua vez, é muito mais simples:

```
this.nav.push(MyPage);
```

Ambos os trechos de códigos foram retirados do blog ([MORONY, 2017](#)).

5. Por fim, outra vantagem de se utilizar Ionic 2 é o seu suporte ao ECMAScript 6 (ES6), que é uma sintaxe do JavaScript reformulada e com melhorias ([ECMAS-CRIPT, 2015](#)). Nem todos os *browsers* suportam o ES6 ainda, porém, um ponto positivo do Ionic 2 é que se o código for escrito em ES6 e o *browser* não suportar a sintaxe, uma tradução para o ES5 é feita assim que o código for compilado, portanto, essa tradução simultânea faz com que o código funcione mesmo assim. ([MORONY, 2017](#))

3.1.3 TypeScript

O TypeScript é uma extensão do JavaScript, criada para facilitar o desenvolvimento de aplicações. Todo programa em JavaScript é também TypeScript, pois a tradução é viável, porém, o TypeScript possui algumas peculiaridades que o JavaScript não possui. Por exemplo, TypeScript possui módulos, classes e interfaces ([BIERMAN; ABADI; TORGERSEN, 2014](#)). A ideia é que, futuramente, desenvolvedores de JavaScript consigam migrar para o TypeScript, visto que a linguagem tende a evoluir. Neste projeto, foi utilizado o TypeScript como linguagem de desenvolvimento, pois esta é a linguagem padrão para o Ionic 2.

3.1.4 HTML5/CSS3

O HTML5 está presente no projeto, pois, o Ionic é um framework que utiliza tecnologias web como *front-end* da aplicação. O Ionic fornece diversos componentes customizáveis em HTML5 para que o desenvolvedor possa tirar o proveito máximo da usabilidade e das interfaces. Para que o design tenha uma boa aparência, o CSS3 faz o papel de maquiador, trazendo vida e beleza aos componentes, tornando toda a interface gráfica da aplicação totalmente customizável.

3.1.5 Firebase

O banco de dados escolhido para o desenvolvimento deste trabalho pertence ao grupo dos bancos de dados não relacionais (NoSQL), são bancos projetados para trabalhar com escalabilidade e suportar aplicações em nuvem (GONÇALVES, 2016).

O Firebase é uma plataforma web mantida pela Google, na qual é possível construir aplicações móveis e web. A vantagem de utilizar a plataforma Firebase é que ela possui vários serviços que ajudam bastante os desenvolvedores. Por exemplo, Firebase Auth, que permite controlar todas as formas de autenticação do sistema, habilitando ou não para o usuário alguns serviços; Firebase Realtime Database, que é o serviço de banco de dados em tempo real, a sincronização dos dados acontece em questão de milissegundos; Firebase Storage é o responsável pelo armazenamento dos dados, que inclusive, está sendo utilizado nesse projeto para armazenar imagens. Além desses serviços, o Firebase possui vários outros (Cloud Message, ADWords, Google Analytics, entre outros) (FIREBASE, 2017).

O Firebase armazena os dados em forma de nós. É possível visualizar uma árvore contendo várias ramificações, que por sua vez possuem os nós, que também podemos representá-los em forma de JSON. É bastante interessante esse tipo de modelagem, pois os dados ficam bem organizados e fáceis de serem compreendidos através de sua hierarquia, como podemos visualizar na Figura 8.

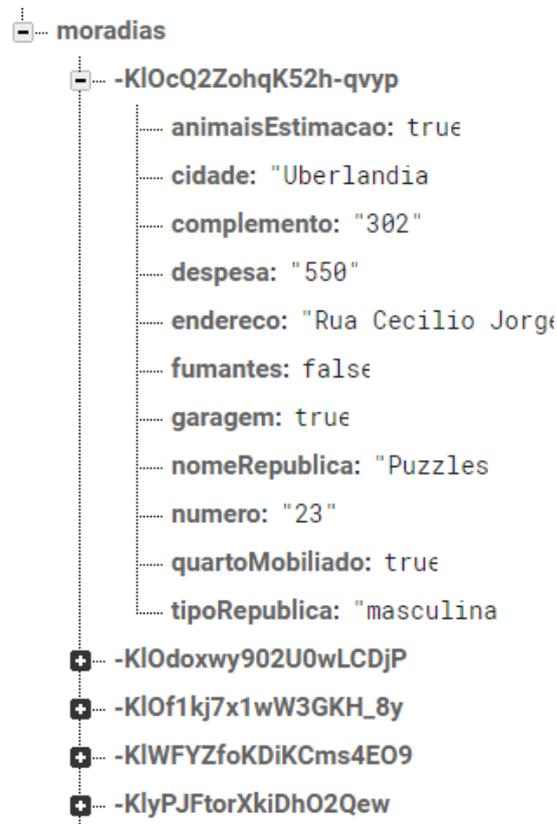


Figura 8 – Estrutura de nós do Firebase

3.2 Atividades Desenvolvidas

3.2.1 Análise de dados e diagramas da aplicação

Para que fosse dado início ao desenvolvimento, foi importante desenhar alguns diagramas para que os dados fossem analisados melhor, a fim de tornar a estrutura da aplicação sólida.

Primeiramente foi pensado no diagrama Caso de Uso, como visto na Figura 9, onde o ator da aplicação é o Estudante. Para que o estudante interaja com o sistema, ele deve realizar o login. O caso de uso “Realizar cadastro” estende (*extends*) o caso de uso “Realizar login”, isso significa que quando o “Realizar login” for executado o “Realizar cadastro” poderá ou não ser executado.

O ator Estudante poderá: Consultar vagas, cadastrar república, editar república, deletar república, visualizar detalhes da vaga e conversar via *chat*, mas para que todas essas ações sejam executadas, o ator deverá estar logado no sistema. Por isso foi utilizado a relação de *include* com o caso de uso “Realizar login”, ou seja, qualquer ação citada anteriormente só é executada se a o estudante estiver autenticado no sistema.

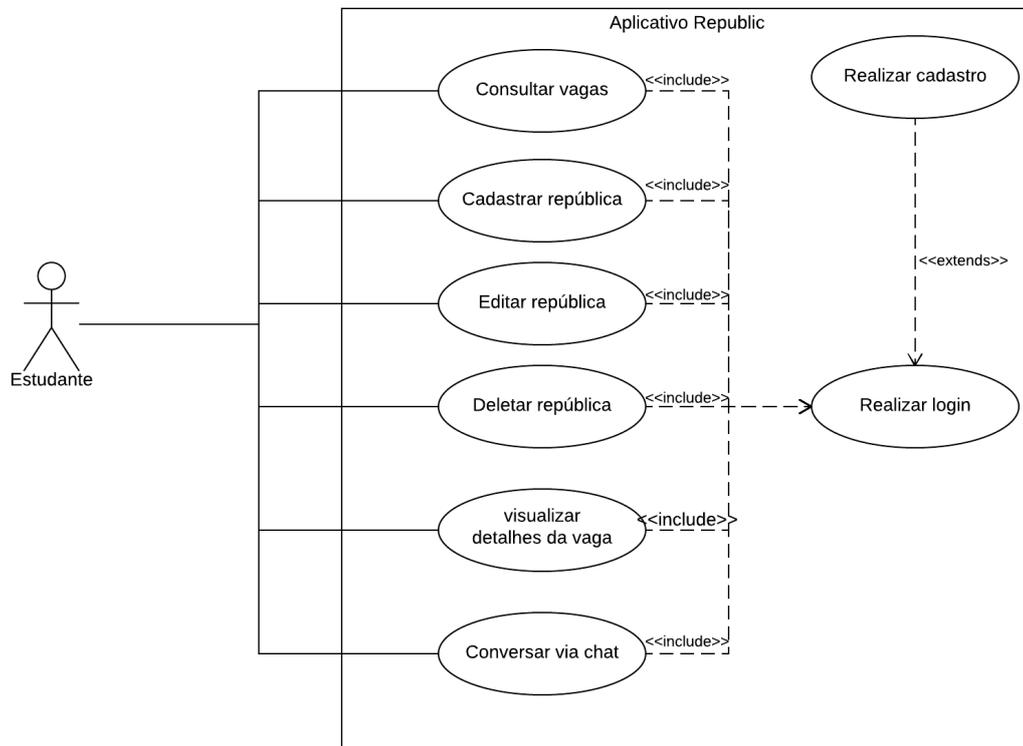


Figura 9 – Diagrama de Caso de Uso da aplicação

Como foi escolhido o Firebase como banco de dados e este não é relacional, sua estrutura é representada em forma de JSON, onde cada nó é um dado no banco. No caso desta aplicação, existem 3 ramificações na estrutura: **moradia**, **usuario** e **image**, como visto na Figura 10. Por ser uma cardinalidade de 1:1 entre moradia e usuário (um usuário possui apenas uma moradia e vice-versa), a estrutura moradia possui um id do usuário, que servirá como referência, neste caso chamamos de “uid”. Além dessa referência, possui informações da moradia em si.

A estrutura usuário possui informações capturadas através da API do Facebook e do Google+ como, por exemplo, nome, sobrenome e imagem do perfil do usuário na rede social que foi utilizada como *login*

A estrutura imagem possui o nome do arquivo e o uid do usuário. Neste caso, um usuário poderá ter mais de uma imagem associada a ele (1:N).

```
republic
{
  "moradia": {
    "animaisEstimacao": "",
    "cidade": "",
    "endereco": "",
    "numero": "",
    "complemento": "",
    "despesa": "",
    "fumantes": "",
    "garagem": "",
    "quartoMobiliado": "",
    "nomeRepublica": "",
    "tipoRepublica": "",
    "uid": ""
  },
  "usuario": {
    "uid": "",
    "displayName": "",
    "imgUrl": ""
  },
  "image": {
    "fileName": "",
    "uid": ""
  }
}
```

Figura 10 – Estrutura JSON dos dados que compõem a aplicação

Por se tratar de uma estrutura com ramificações, o JSON pode ser representada graficamente em forma de árvore, como visto na Figura 11. Neste caso, foi acrescentado na figura o tipo de cada variável da aplicação, a fim de representar melhor o que é cada dado.



Figura 11 – Estrutura JSON representada graficamente em forma de árvore

3.2.2 User Experience (UX) e explicação das telas

O trabalho em desenvolvimento foi em grande parte pensado nos conceitos de usabilidade para que o usuário tivesse uma melhor experiência ao usar o aplicativo. Infelizmente, não foi possível realizar nenhum teste de usabilidade nesta primeira etapa do desenvolvimento, porém, nas próximas versões da aplicação serão realizados testes A/B, testes de fluxo, enquetes e pesquisas com os usuários, para oferecer uma melhor experiência à todos.

Em contrapartida, foi possível realizar os primeiros passos de um desenvolvimento centrado no usuário e pensando nos aspectos de design foi criado o *wireframe* da aplicação, que é uma prototipagem de baixo nível, praticamente um rascunho do que está para ser desenvolvido. O *wireframe* da aplicação apresenta pouca informação, sem se preocupar muito com métricas, cores, aspectos gráficos, mas já é possível visualizar e ter uma ideia de como as telas serão e é possível também, entender melhor o fluxo da aplicativo, a navegabilidade de cada tela. Nas Figuras 12 e 13 podemos visualizar o *wireframe* do projeto, em que foi desenhado a mãos, para ser representado de uma forma simples e sem muitos detalhes, mostrando a navegabilidade entre as telas. Na Figura 12 podemos ver o fluxo do cadastro de uma nova república, percorrendo todos os passos, desde a tela de preencher informações sobre a localidade da moradia até a tela de fazer *upload* das imagens da república. Já a Figura 13, mostra o fluxo do menu até a página “Minha Conta”, onde o usuário pode editar a conta. O fluxo da tela de *splashscreen* até a tela de *login*, e por fim, o fluxo entre a tela que exibe informações da república (localização no mapa, valor da despesa) até a tela de chat ou a tela de visualizar imagens da república.

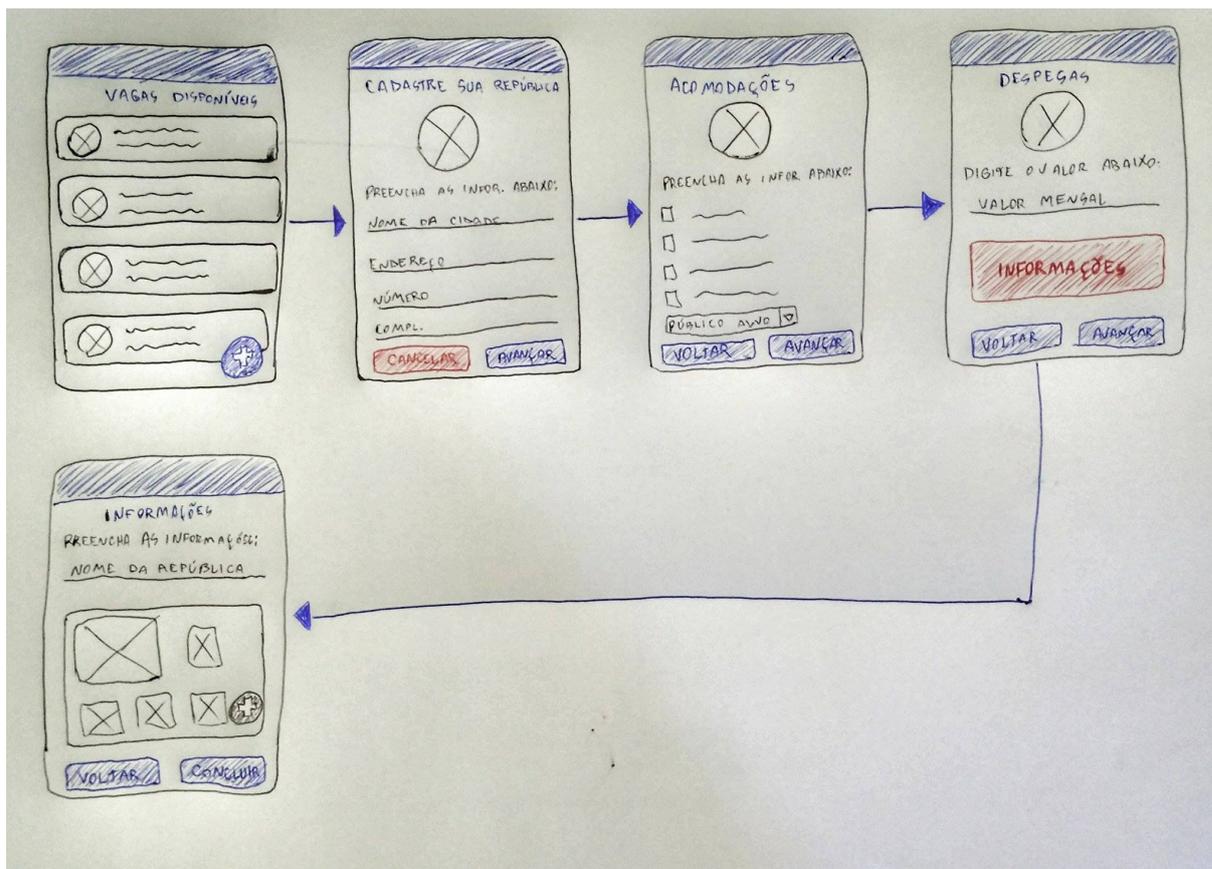


Figura 12 – Wireframe do projeto, mostrando o fluxo da aplicação

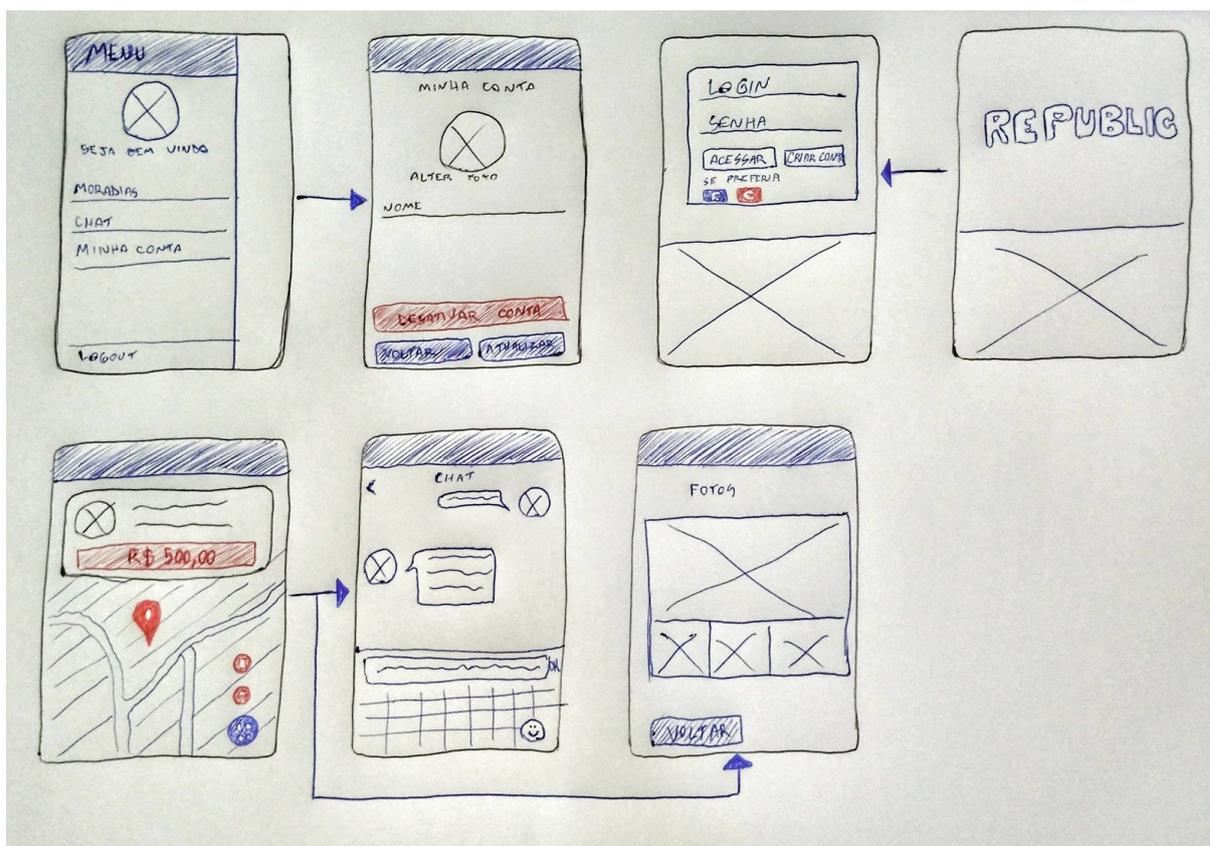


Figura 13 – Continuação do *wireframe*

Com o *wireframe* em mãos, é possível pensar em uma prototipagem melhor, de maior alto nível e apresentando mais informações ou quase todas. No caso da prototipagem deste aplicativo, foi pensado em um design muito fiel ao que se espera das telas da própria aplicação, portanto, o que será apresentado de protótipo nas Figuras seguintes será muito parecido com a User Interface (UI) do aplicativo. O design dos protótipos de alta fidelidade foram desenvolvidos utilizando a ferramenta online chamada Figma (FIGMA, 2017), que é uma plataforma para geração de *mockups* (protótipos).

A ideia era que o aplicativo tivesse um design limpo, sem muita dificuldade para o usuário, de modo que evitasse confusão em encontrar determinada funcionalidade e que houvesse uma fluidez no fluxo em realizar algo na aplicação. Como podemos ver, na Figura 14 foi utilizado uma imagem gratuita para a tela de *splashscreen*. O azul claro foi escolhido pois é uma cor que não agride e não polui o layout (dependendo da forma que é utilizado). A parte de *login* foi pensada em uma estrutura que pudesse utilizar da sua conta em redes sociais. Caso o usuário preferir criar um cadastro com *email* e senha, também é possível. O menu é bem limpo, com poucas informações e exibe a foto do usuário que está logado.

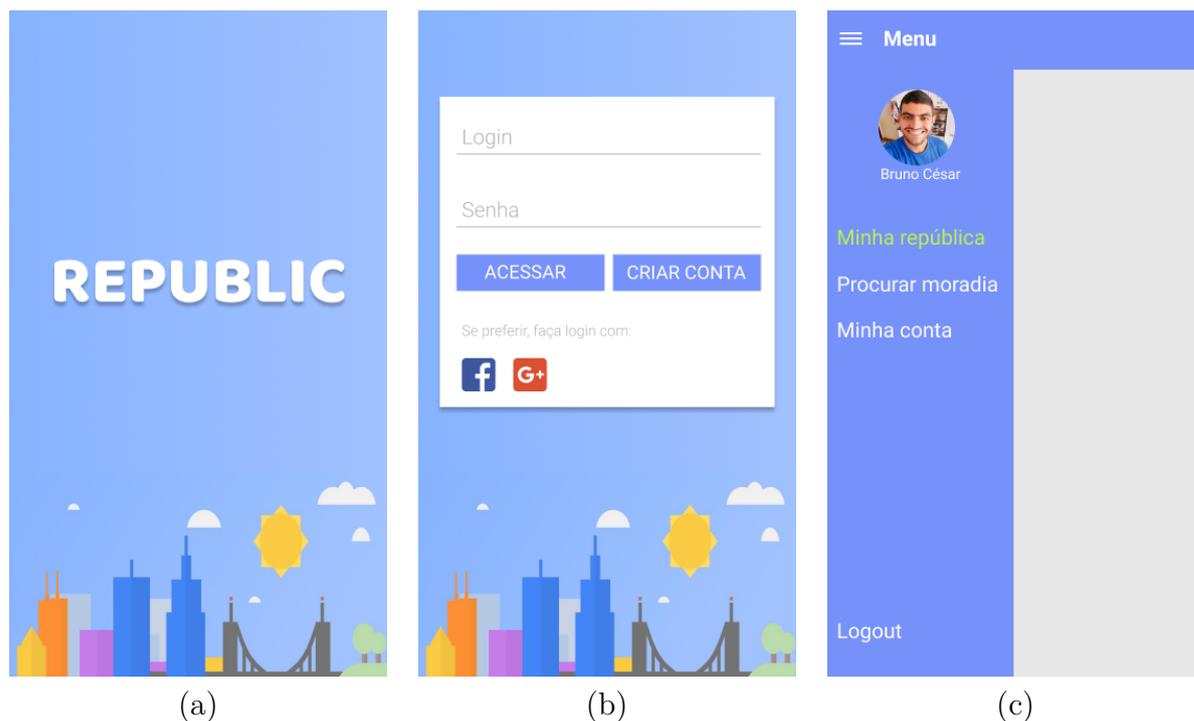


Figura 14 – (a) Tela de splashscreen, (b) tela de *login* e (c) menu principal

As telas de cadastro, na Figura 15, foram desenhadas pensando nos recursos que o próprio Ionic disponibiliza: *float inputs* para os campos, botões simples da própria biblioteca de componentes do Ionic. Para dar um pouco de vida ao design, foi acrescentado imagens em forma de avatar¹. As imagens seguem padrões do material design, que é uma linguagem visual que tem como princípio o emprego de um bom design, seguindo métricas, boas práticas de iluminação, sombreado, e ideia de superfícies bem definidas (GOOGLE, 2014).

Ainda na Figura 15, o cadastro de uma república se baseia em dados básicos, tais como, cidade, endereço, número e complemento, após isso, o usuário irá preencher alguns *checkboxes* informando se a república possui garagem disponível, se o quarto é mobiliado, se aceita fumantes e animais de estimação, além disso, informa o público alvo (masculino, feminino ou misto), em seguida, o valor das despesas mensais, que é um valor total mensal aproximado, o cadastro termina na Figura 17 (a), onde o usuário insere um nome (optativo) para a república e carrega e/ou tira fotos de alguns cômodos da moradia através do próprio aplicativo. Foi pensado em algo simples, em forma de *grid*, possibilitando o usuário adicionar ou não mais fotos.

¹ Avatar é um componente que cria uma forma arredondada para as imagens. No caso do ionic, este componente chama-se **ionic-avatar**



Figura 15 – (a) Tela de cadastro da república, (b) tela de acomodações e (c) tela de despesas

Na tela que exibe informações sobre as vagas disponíveis, na Figura 16 (a), foi pensando em uma forma de exibir a lista de repúblicas por meio de *cards*, que também é um componente do Ionic. Além de dar um ar de organização, é muito utilizado em aplicações móveis. Os botões flutuantes também foram utilizados na composição do layout, economiza espaço e fica disponível a todo instante, já que ele sobrepõe todos os elementos da tela, no caso desta tela, o botão flutuante leva para a tela de cadastro de uma nova república. Na tela mostrada na Figura 16 (b), a ideia foi criar um *card* sobrepondo o mapa fornecido pela API do Google Maps, que será abordada nas seções seguintes, nessa tela, o usuário visualiza o endereço da moradia no mapa. Além disso, um botão flutuante leva o usuário às telas seguintes, de Chat e de Visualizar imagens, que serão melhor visualizados nos *screenshots* da própria aplicação, nas seções seguintes também. Para o chat, mostrado na Figura 16 (c), foi pensado em um layout simples, sem muita informação, apenas será um intermediário entre os usuários, o design ainda pode ser alterado em futuras versões do aplicativo, pois ainda está em desenvolvimento.

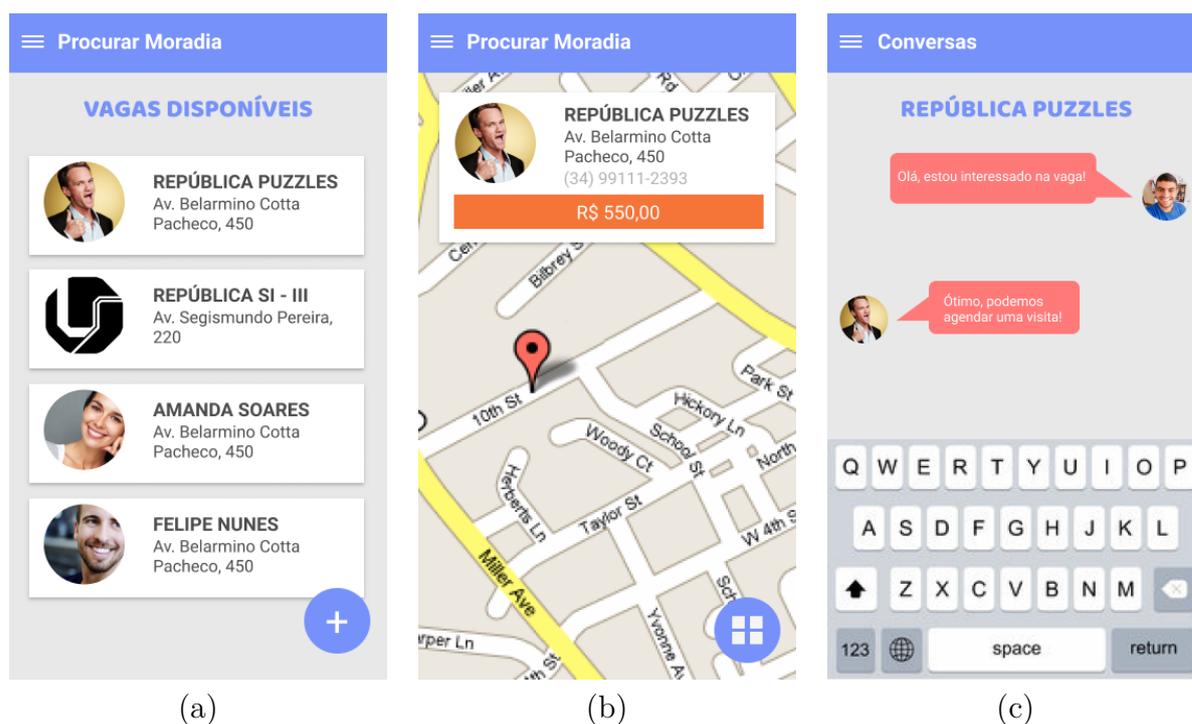


Figura 16 – (a) Tela de moradias disponíveis, (b) tela de informações adicionais e (c) tela de chat

A Figura 17 (b) é como o usuário interessado pela vaga irá visualizar as fotos da república. O design também seguirá o mesmo padrão do *upload*, em forma de *grid*. Já a tela de edição da conta, mostrada na Figura 17 (c), é apenas para alterar o avatar do usuário e seu nome, e por fim, desativar sua conta. A cor vermelha do botão de Desativar Conta foi escolhida para dar ênfase de que é uma ação importante a ser tomada ou não.

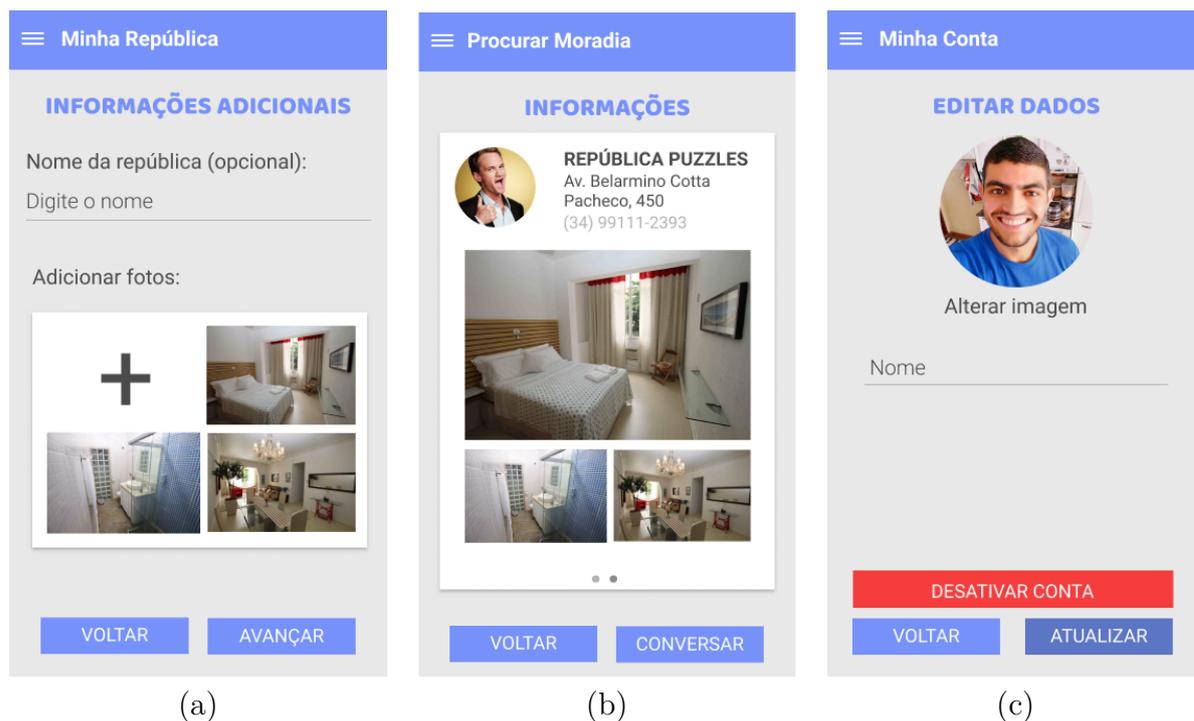


Figura 17 – (a) Tela de adicionar fotos, (b) tela de visualizar fotos e (c) a tela de edição da conta

3.2.3 Componentes utilizados no desenvolvimento

Para utilizarmos o Firebase no projeto Ionic é necessário realizar uma configuração na aplicação, como mostra no trecho de código abaixo:

```
export const firebaseConfig = {
  apiKey: "XXXXXX",
  authDomain: "XXXXXXXXX.firebaseio.com",
  databaseURL: "https://XXXXXXXXX.firebaseio.com",
  projectId: "XXXXXXXXXX",
  storageBucket: "XXXXXXXXX.appspot.com",
  messagingSenderId: "XXXXXXXXXX"
};
```

As informações reais foram substituídas por motivos de segurança. Porém é possível entender que configuração exige uma `apiKey`, um domínio, o id do projeto, entre outros. Essas informações são obtidas na plataforma do Firebase e são responsáveis por fazer a comunicação entre a aplicação e os serviços do Firebase.

O desenvolvimento da parte de login da aplicação foi baseada em uma sequência de tutoriais em (CÓDIGO, 2017), onde o autor ensina e disponibiliza o código fonte para

utilização. Atualmente, a aplicação pode ser acessada realizando login via Facebook, Google+ ou por uma conta criada na própria aplicação. Para que se utilize o Facebook como autenticação, é necessário importar no projeto, as dependências da API do Facebook, como visto no código abaixo:

```
import { Facebook } from '@ionic-native/facebook';
```

Além disso, é necessário realizar configurações na plataforma de desenvolvimento do Facebook (DEVELOPER, 2017), para que seja gerado uma chave que habilita que o Facebook informações de contas de usuários para o aplicativo em desenvolvimento. Além disso, também é necessário habilitar a autenticação via Facebook na plataforma do Firebase. O processo com o Google+ e outras redes, por exemplo o Twitter, é parecido.

Para entendermos melhor, segue o trecho do código que faz a autenticação com o Facebook:

```
signInWithFacebook() {  
  return this.facebook.login(['public_profile', 'email'])  
    .then((res: FacebookLoginResponse) => {  
      return this.angularFireAuth.auth.signInWithCredential(  
        firebase.auth.FacebookAuthProvider  
          .credential(res.authResponse.accessToken));  
    });  
}
```

O Facebook disponibiliza um perfil público para que o usuário se conecte a uma aplicação. O Firebase permite a autenticação caso a chave de acesso (`accessToken`) seja válida.

Para utilizarmos a API do Google Maps foi necessário importar a dependência de *Geolocation*, como vemos abaixo:

```
import { Geolocation } from '@ionic-native/geolocation';
```

Até o momento, é possível carregar o mapa na tela de informações adicionais, o qual exibido na posição atual em que o usuário se encontra. Além disso, algumas configurações já podem ser feitas dentro de `mapOptions`, por exemplo ajustar o nível do zoom, desabilitar os ícones padrões do Google Maps (*zoom in*, *zoom out*, *street view*), através do `disableDefaultUI`, como visto no trecho abaixo:

```
loadMap(){
```

```
this.geolocation.getCurrentPosition().then((position) => {

  let latLng = new google.maps.LatLng(
    position.coords.latitude, position.coords.longitude);

  let mapOptions = {
    center: latLng,
    zoom: 15,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    disableDefaultUI: true
  }

  this.map = new google.maps.Map(this.mapElement.nativeElement,
    mapOptions);

}, (err) => {
  console.log(err);
});

}
```

O que está sendo desenvolvido é a conversão do endereço que o usuário cadastra para a república em latitude e longitude, ou *geocode*, que são as coordenadas geoespaciais padronizadas. Esse recurso facilitará a visualização de onde está localizada a moradia no mapa.

Para a utilização dos recursos da câmera do smartphone foi utilizado o componente abaixo, que é nativo do Ionic:

```
import { Camera, CameraOptions } from '@ionic-native/camera';
```

Com esse componente, é possível controlar as funcionalidades da câmera do dispositivo, visto que o Cordova é o responsável por fazer essa comunicação entre os recursos do celular com o framework Ionic. Para captura de imagem, foi realizada a configuração da câmera, ajustando a qualidade da imagem em *quality*, o destino da imagem, o formato em que é salva é configurada em *encodingType* e o tipo de mídia, então a foto é capturada, como vemos no trecho de código abaixo, em *getPicture()*:

```
capture() {
```

```
const cameraOptions: CameraOptions = {
  quality: 50,
  destinationType: this.camera.DestinationType.DATA_URL,
  encodingType: this.camera.EncodingType.JPEG,
  mediaType: this.camera.MediaType.PICTURE,
};

this.camera.getPicture(cameraOptions).then((imageData) => {
  this.captureDataUrl = 'data:image/jpeg;base64,' + imageData;
}, (err) => {
  console.log(err);
});
}
```

Os templates (*front-end* da aplicação) foram todos desenvolvidos, até o momento, utilizando os próprios componentes de interface do framework Ionic. Não foi necessário utilizar nenhum outro framework gráfico para construir algum determinado componente visual.

4 Resultados Obtidos

4.1 Estado Atual do Desenvolvimento

Como estado atual do desenvolvimento, já é possível realizar o login através de redes sociais, se cadastrar utilizando *e-mail*, cadastrar repúblicas e consultar as vagas disponíveis. Será desenvolvido o chat para que os usuários possam trocar mensagens utilizando o próprio aplicativo. Atualmente, para sanar a falta do *chat*, é fornecido o número do telefone do usuário, para que eles possam se comunicar utilizando outro mensageiro eletrônico, ou até mesmo por meio de ligações, mas não é o objetivo do aplicativo, a ideia sempre foi utilizar um mensageiro exclusivo para esse intermédio. Outro ponto que está sendo desenvolvido é melhoria do *upload* da imagem no banco de dados. A ideia é transformá-las totalmente em uma cadeia de caracteres em *base64* e depois salvá-las. Em seguida, quando o usuário necessitar visualizar a imagem, será feita a conversão de *base64* para o formato JPEG novamente, isso otimizará o banco e reduzirá o tamanho dos arquivos salvos. Além disso, o *front-end* do *grid* de visualização e *upload* das imagens está em desenvolvimento.

4.2 Testes Realizados

Para entender melhor a atual situação dos estudantes que precisam buscar vagas em repúblicas, foi realizada uma pesquisa online, disponível e aberta ao público no geral. A pesquisa foi divulgada em grupos do Facebook onde o foco maior era de estudantes, principalmente alunos da Universidade Federal de Uberlândia de diversos cursos. Além disso, também foi divulgado em grupos de repúblicas, onde é comum encontrar diariamente pessoas dialogando sobre vagas em moradias, pois é o nicho em que o aplicativo realmente vai atuar. A pesquisa contou com 116 participantes.

A primeira pergunta é sobre a faixa etária dos participantes. Como visto na Figura 18 a maior parte deles estão na faixa dos 24 a 30 anos, totalizando 52 de 116 pessoas.

Qual a sua faixa etária?

116 responses

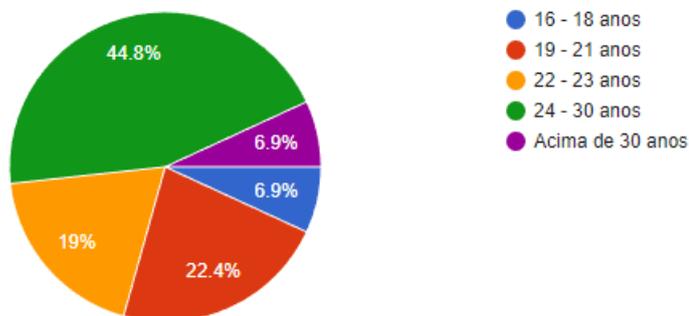


Figura 18 – Faixa etária dos participantes da enquete

O segundo gráfico, visto na Figura 19, demonstra o gênero dos participantes, em sua maior parte do sexo masculino (62 pessoas) e 54 do sexo feminino. Podemos perceber que o resultado foi próximo de 50% para cada gênero.

Qual seu gênero?

116 responses

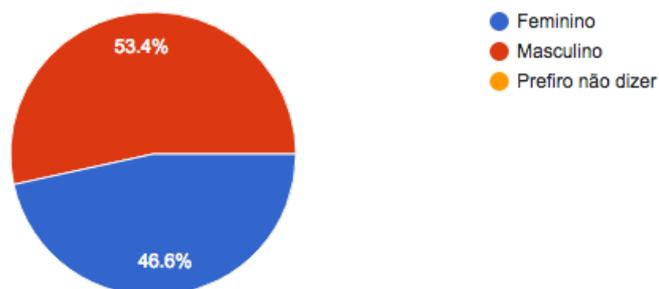


Figura 19 – Gênero dos participantes da enquete

A terceira pergunta analisa o percentual de estudantes participantes que precisaram mudar de cidade para cursar alguma faculdade, como visto na Figura 20. O resultado conclui que maior parte desses estudantes precisaram se deslocar para estudar, totalizando 71,6%, (83 de 116 participantes). É um resultado significativo para o propósito do trabalho.

Você já precisou mudar de cidade para cursar alguma faculdade?

116 responses

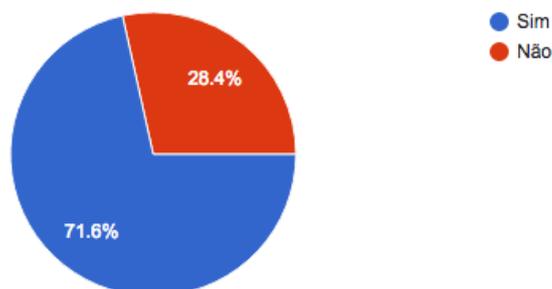


Figura 20 – Participantes que saíram da cidade por conta da faculdade

Caso o participante se enquadrasse ao grupo que se deslocou para outra cidade para cursar a faculdade, ele respondia o quão difícil foi encontrar uma república para morar, como visto na Figura 21. Em uma escala de 1 a 5 (sendo 1 muito fácil e 5 muito difícil), a maior parte (44,2%) escolheu a opção 3, ou seja, foi medianamente difícil encontrar uma moradia. Em seguida, o segundo maior resultado foi 4, ou seja, 19 pessoas consideraram que foi uma tarefa um pouco difícil. Já 9,3% dos participantes tiveram muita dificuldade em encontrar uma república. O restante dos participantes não tiveram muita dificuldade, 5,8% considerou fácil e 18,6% achou medianamente fácil.

Se sim, em uma escala de 1 a 5, o quão difícil foi encontrar repúblicas/moradias para morar?

86 respostas

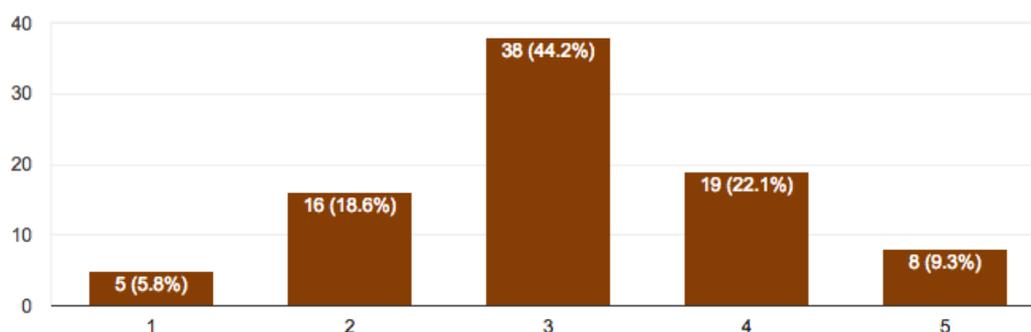


Figura 21 – Nível de dificuldade para encontrar repúblicas para morar

Também foi perguntado aos participantes sobre a aceitação de um aplicativo que auxiliasse a encontrar repúblicas. 94% das respostas foram sim e apenas 6% (7 de 116 pessoas) não utilizariam esse tipo de aplicativo como podemos ver na Figura 22.

Se existisse um aplicativo de smartphone que te ajudasse a encontrar e/ou divulgar vagas em repúblicas, você utilizaria?

116 respostas

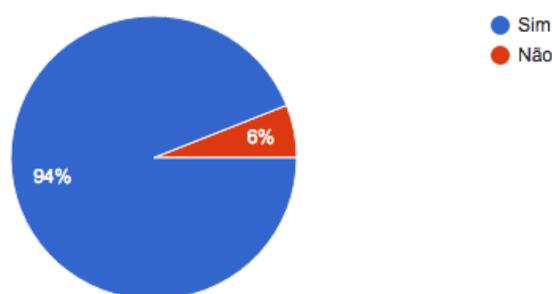


Figura 22 – Nível de aceitação dos participantes caso houvesse um aplicativo que busca e divulga moradias

Na pergunta sobre quão útil este aplicativo seria o resultado foi satisfatório, 69,8%

optou por “muito útil”, como vemos na Figura 23.

Em uma escala de 1 a 5, quão útil para os estudantes, este aplicativo seria?

116 responses

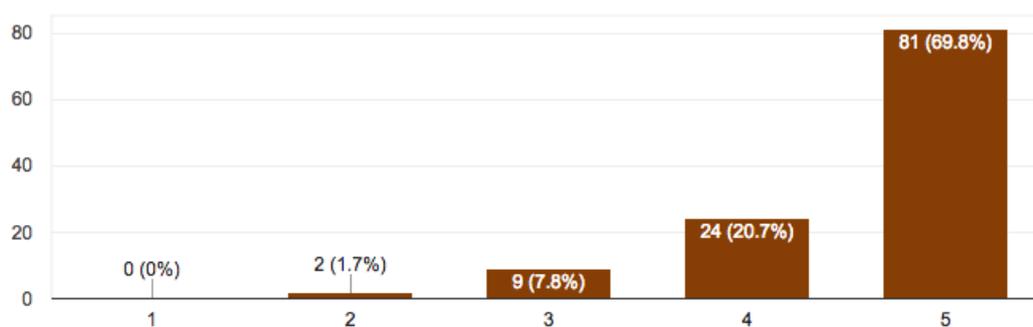


Figura 23 – Nível de utilidade do aplicativo para os estudantes

Uma pergunta sobre quais critérios os estudantes julgam importantes avaliar antes de alugar um quarto também foi feita, como vemos na Figura 24. Os pontos mais importantes para os estudantes são o preço, boa localização, colegas que dividem o aluguel, segurança e lugares silenciosos.

Quais detalhes você avalia e julga importante, antes de alugar uma república?

116 responses

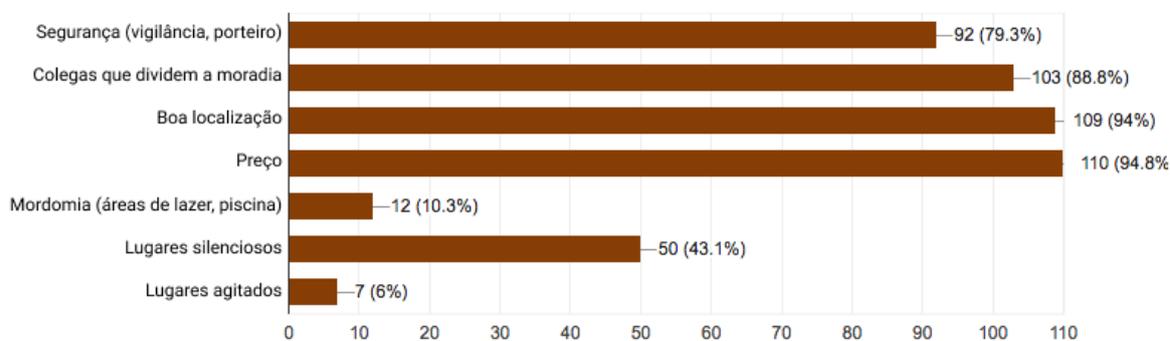


Figura 24 – Detalhes importantes que são avaliados quando os participantes vão alugar um quarto

Também foi perguntado aos participantes qual foi a forma que encontraram a moradia atual que residem, como mostra na Figura 25. A maior parte das respostas (36,4%) disseram que encontraram vagas através de seus amigos, uma grande parte respondeu “outros” (30,3%), e a terceira maior parcela das respostas disseram “facebook”, com (22,2%).

Como você encontrou vaga na sua moradia atual?

99 responses

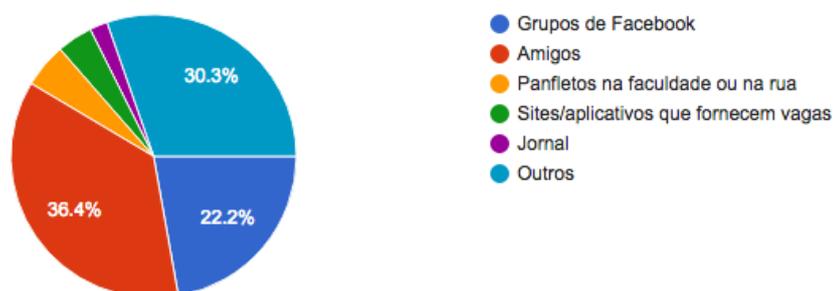


Figura 25 – Como o participante encontrou vaga na moradia que reside atualmente

A última resposta avalia os participantes que possuem vagas nas repúblicas e utilizam de meios de comunicação para divulgar essas vaga, como mostra na Figura 26. 78% das respostas foi grupos de Facebook, a segunda maior parcela é amigos (47,5%). As menores parcelas são panfletos na faculdade, sites/aplicativos, jornais e outros.

Caso seja o responsável pela república, qual é a o meio de comunicação que você divulga as vagas disponíveis?

59 respostas

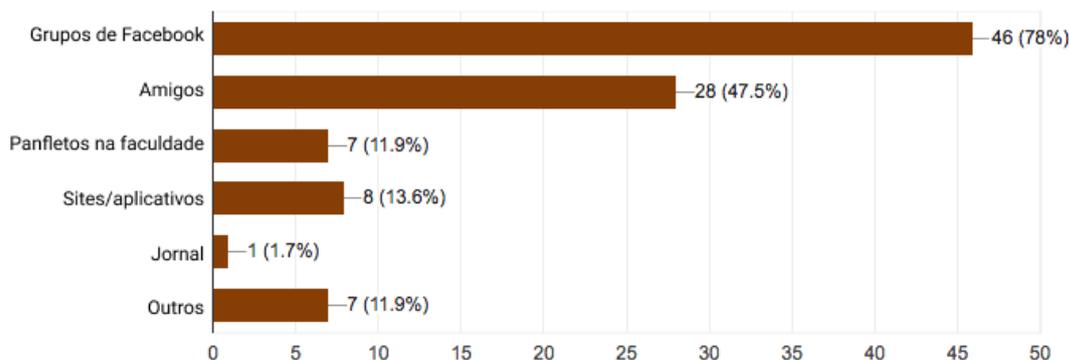


Figura 26 – Meio de comunicação em que o participante divulga vagas disponíveis

Considerando os resultados obtidos, podemos perceber uma grande oportunidade com o desenvolvimento deste trabalho, pois o aplicativo poderá ser bem recebido pelos usuários, que por sua vez, demonstram interesse em utilizar essa nova forma de divulgação e busca por repúblicas. Por ser um público jovem, que utiliza bastante smartphones e aplicativos móveis, a ferramenta poderá ser útil no intermédio entre os usuários, além disso, até agilizar o processo de negociação entre as partes. Nota-se um grande número de pessoas utilizando o Facebook como facilitador e intermediário, a ideia é que o aplicativo faça com que os usuários comecem a utilizar mais as ferramentas móveis, tornando-se mais popular.

4.3 Trabalhos Futuros

Para trabalhos futuros, uma das ideias é criar um mecanismo de interesses, no qual o aplicativo poderia sugerir aos usuários, pessoas que tivessem interesses parecidos, isso ajudaria os estudantes a encontrar pessoas com mais afinidades e evitaria possíveis problemas com personalidades muito diferentes. Outra ideia interessante é criar um sistema de ranqueamento, no qual, pessoas que já moraram ou moram na república pudessem avaliar de forma objetiva sobre a vaga em questão.

5 Conclusão

O trabalho apresentado foi de grande importância para o desenvolvimento acadêmico e profissional, pois foram utilizados conceitos teóricos e práticos para a realização do aplicativo.

Foi utilizado conceitos de modelagem de dados, para entender melhor como cada dados se relacionariam entre si. Através dos diagramas apresentados foi possível encontrar soluções e entender melhor graficamente o contexto.

Além da modelagem de dados, foi utilizado conceitos de Usabilidade (UX), para que fosse possível desenvolver algo acessível para o público alvo, que são os estudantes, de modo que facilite o entendimento de cada usuário em realizar determinada tarefa dentro do sistema. Tudo isso foi pensado adotando conceitos de interação humano computador. As cores, os botões, o design são fatores que influenciam bastante na hora da utilização do aplicativo por um usuário.

O recolhimento de informações através de enquetes online foi algo que contribuiu bastante para que fosse possível tomar decisões e caminhos corretos no desenvolvimento, pois, entender quem são os usuários e suas necessidades são tão importantes quanto proporcionar uma melhor experiência à todos.

Concluindo, o aplicativo está sendo desenvolvido para ajudar, facilitar e melhorar a relação entre quem quer alugar a moradia e quem está procurando por uma vaga. Pois, é algo que sempre preocupa os estudantes assim que se mudam de cidades para estudar.

O aplicativo Republic pode ser melhorado constantemente e essa é a missão, sempre progredir, inovar e desenvolver novas funcionalidades pensando no usuário.

Referências

- APKPURE. *Minha República*. 2015. Disponível em: <<https://apkpure.com/br/minha-rep%C3%BAblica/mr.minharepublica>>. Acesso em: 25 jul. 2017. Citado na página 15.
- ASSIS, M. D. et al. Adaptação À universidade no processo de migração e sofrimentos psíquicos. 2013. Disponível em: <<http://www.prac.ufpb.br/enex/trabalhos/6CCHLADPPROBEX2013110.pdf>>. Citado na página 10.
- BIERMAN, G.; ABADI, M.; TORGERSEN, M. Understanding typescript. In: _____. *ECOOP 2014 – Object-Oriented Programming: 28th European Conference, Uppsala, Sweden, July 28 – August 1, 2014. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 257–281. ISBN 978-3-662-44202-9. Disponível em: <http://dx.doi.org/10.1007/978-3-662-44202-9_11>. Citado na página 23.
- CEARÁ, G. do. *Corredores Digitais: Startup cearense é selecionada para participar da Campus Party, em São Paulo*. 2016. Governo do Ceará. Disponível em: <<http://ceara.gov.br/sala-de-imprensa/noticias/15321-corredores-digitais-startup-e-selecionada-para-participar-da-campus-party-em-sao-paulo>>. Acesso em: 23 jun. 2017. Citado na página 15.
- CEARÁ, G. do. *Terceira edição do 'Corredores Digitais' será lançada em março*. 2016. Governo do Ceará. Disponível em: <<http://ceara.gov.br/sala-de-imprensa/noticias/15669-terceira-edicao-do-corredores-digitais-sera-lancada-em-marco>>. Acesso em: 23 jun. 2017. Citado na página 15.
- CORPORATION, I. Native, web or hybrid mobile-app development. In: . IBM Corporation, 2012. Disponível em: <<ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf>>. Citado 2 vezes nas páginas 19 e 20.
- CÓDIGO, F. de. *Como criar um login com uma conta do facebook com ionic 2 e firebase*. 2017. Disponível em: <<http://www.fabricadecodigo.com/como-criar-um-login-com-uma-conta-do-facebook-com-ionic-2-e-firebase/>>. Acesso em: 25 jul. 2017. Citado na página 34.
- DALMASSO, I. et al. Survey, comparison and evaluation of cross platform mobile application development tools. In: *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. [S.l.: s.n.], 2013. p. 323–328. ISSN 2376-6492. Citado na página 19.
- DEVELOPER, F. *Adicione o Login no Facebook ao seu aplicativo ou site*. 2017. Disponível em: <<https://developers.facebook.com/docs/facebook-login>>. Acesso em: 25 jul. 2017. Citado na página 35.
- DOMUZ. *Domuz - Repúblicas Universitárias*. 2016. Repúblicas Universitárias. Disponível em: <<http://www.domuz.com.br/>>. Acesso em: 06 jul. 2017. Citado na página 16.
- ECMASCRIPT. EcmaScript® 2015 language specification. In: . [s.n.], 2015. Disponível em: <<http://www.ecma-international.org/ecma-262/6.0/>>. Citado na página 23.

- FIGMA. In: . [s.n.], 2017. Disponível em: <<https://www.figma.com>>. Citado na página 30.
- FIREBASE. In: . [s.n.], 2017. Disponível em: <<https://firebase.google.com/>>. Citado na página 24.
- G1, P. de notícias. *Moradia é um dos desafios para estudantes que passam no vestibular*. 2016. Moradia. Disponível em: <<http://g1.globo.com/jornal-hoje/noticia/2016/01/moradia-e-um-dos-desafios-para-estudantes-que-passam-no-vestibular.html>>. Acesso em: 21 jun. 2017. Citado na página 10.
- GLOBO, O. *Brasileiros criam site para universitários buscarem vagas em repúblicas*. 2015. ReepubGlobo. Disponível em: <<https://oglobo.globo.com/sociedade/tecnologia/brasileiros-criam-site-para-universitarios-buscarem-vagas-em-republicas-17341506>>. Acesso em: 22 jun. 2017. Citado na página 13.
- GONÇALVES, A. L. Desenvolvimento de um aplicativo android utilizando banco de dados não-relacional para organização e controle de presença de um time de futebol. 2016. Citado na página 24.
- GOOGLE. In: . [s.n.], 2014. Disponível em: <<https://material.io/guidelines/>>. Citado na página 31.
- HARTINGTON, M. Announcing the ionic 2 release candidate! In: . Ionic Blog, 2016. Disponível em: <<http://blog.ionic.io/announcing-the-ionic-2-release-candidate/>>. Citado na página 21.
- IONIC. In: . Ionic Official Website, 2017. Disponível em: <<https://ionicframework.com/>>. Citado na página 20.
- LOPES, S. *Aplicações mobile híbridas com Cordova e PhoneGap*. [S.l.]: Editora Casa do Código, 2016. Citado na página 20.
- MALAVOLTA, I. et al. Hybrid mobile apps in the google play store: An exploratory investigation. In: *Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems*. Piscataway, NJ, USA: IEEE Press, 2015. (MOBILESoft '15), p. 56–59. ISBN 978-1-4799-1934-5. Disponível em: <<http://dl.acm.org.ez1.periodicos.capes.gov.br/citation.cfm?id=2825041.2825051>>. Citado na página 20.
- MORONY, J. 7 reasons why ionic 2 is better than ionic 1. In: . [s.n.], 2017. Disponível em: <<https://www.joshmorony.com/7-reasons-why-ionic-2-is-better-than-ionic-1/>>. Citado na página 23.
- REEPUB. *Reepub - Encontre Repúblicas próximas a você*. 2017. Reepub. Disponível em: <<http://www.reepub.com/>>. Acesso em: 21 jun. 2017. Citado na página 13.
- SERRANO, N.; HERNANTES, J.; GALLARDO, G. Mobile web apps. *IEEE Software*, v. 30, n. 5, p. 22–27, Sept 2013. ISSN 0740-7459. Citado na página 19.
- UNICAMP, M. *Morar Unicamp*. 2017. Morar Unicamp. Disponível em: <<http://morarunicamp.com.br/>>. Acesso em: 06 jul. 2017. Citado 2 vezes nas páginas 17 e 18.

- UNICAMP, R. *Sistema de Repúblicas da ARU*. 2016. Repúblicas Unicamp. Disponível em: <<http://republicasunicamp.com.br/>>. Acesso em: 24 jun. 2017. Citado na página 15.
- VIEBRANTZ, A. F. P. M.; CAMPOS, G. F. da S. Construindo aplicativos híbridos com ionic framework. *Tendências e Técnicas em Sistemas Computacionais*, p. 60. Citado na página 20.
- XANTHOPOULOS, S.; XINOGALOS, S. A comparative analysis of cross-platform development approaches for mobile applications. In: *Proceedings of the 6th Balkan Conference in Informatics*. New York, NY, USA: ACM, 2013. (BCI '13), p. 213–220. ISBN 978-1-4503-1851-8. Disponível em: <<http://doi.acm.org.ez1.periodicos.capes.gov.br/10.1145/2490257.2490292>>. Citado na página 19.