

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Bacharelado em Sistemas de Informação

Luiz Otávio Teixeira Caldonazo

**UM ESTUDO DA ABORDAGEM LEAN UX COM SUA APLICAÇÃO NO
DESENVOLVIMENTO DE UM APLICATIVO PARA ANDROID**

Monografia de Projeto Orientado em Sistemas de Informação

Volume I

Minas Gerais
2018

Luiz Otávio Teixeira Caldonazo

**UM ESTUDO DA ABORDAGEM LEAN UX COM SUA APLICAÇÃO NO
DESENVOLVIMENTO DE UM APLICATIVO PARA ANDROID**

Monografia de Projeto Orientado em Sistemas de Informação apresentado como requisito da disciplina de Projeto Orientado em Sistemas de Informação do Curso de Bacharelado em Sistemas de Informação da UFMG.

Orientador: Prof. Dr. Clarindo Isaías Pereira da Silva e Pádua

Volume I

**Minas Gerais
2018**

Resumo

Vivemos em um contexto onde a pressão por inovação e agilidade são cada vez maiores, para que o mercado conseguisse se adaptar a este ritmo foram desenvolvidas as metodologias ágeis, Lean UX é uma nova metodologia de solução de problemas de negócio que faz uso dos princípios ágeis, este trabalho demonstra como colocá-la em prática e obter resultados de qualidade rapidamente utilizando como base a opinião das partes interessadas no produto/serviço sendo desenvolvido.

Abstract

We live in a context where the pressure for innovation and agility are increasing, so for the market to be able to adapt to this rhythm agile methodologies were developed, Lean UX is a new methodology that focus on solving business problems and makes use of agile principles, this work demonstrates how to put it into practice and obtain quality results quickly based on the opinion of the stakeholders in the product/service being developed.

Lista de ilustrações

Figura 1 – Modelo Cascata não modificado	11
Figura 2 – Comparação metodologia Cascata (a) x Espiral (b) x XP (c)	14
Figura 3 – O processo Design Thinking	17
Figura 4 – O processo Lean Startup	18
Figura 5 – O foco de um MVP	21
Figura 6 – Metodologia	24
Figura 7 – Proto-Persona 1: Olivia	29
Figura 8 – Proto-Persona 2: Beto	29
Figura 9 – Proto-Persona 3: Luiza	30
Figura 10 – Proto-Persona 4: Alfredo	30
Figura 11 – Protótipo 1: Funcionalidades Gerais	32
Figura 12 – Protótipo 1: Usuários - Funcionalidades	33
Figura 13 – Protótipo 1: Administradores - Funcionalidades	34
Figura 14 – Protótipo 2: Funcionalidades Gerais	37
Figura 15 – Protótipo 2: Usuários - Funcionalidades	38
Figura 16 – Protótipo 2: Administradores - Funcionalidades	39
Figura 17 – Protótipo 3: Funcionalidades Gerais	42
Figura 18 – Protótipo 3: Usuários - Funcionalidades	43
Figura 19 – Protótipo 3: Administradores - Funcionalidades	44
Figura 20 – Protótipo 3: Feedbacks	45
Figura 21 – Visualização Geral do Aplicativo	47
Figura 22 – Visualização do Administrador	48
Figura 23 – Visualização do Usuário	49

Lista de símbolos

CD	Compact Disc
HD	Hard Drive
MVP	Minimum Viable Product
RUP	Rational Unified Process
SDLC	Software Development Life Cycle
TI	Tecnologia da Informação
UX	User Experience
XP	Extreme Programming

Sumário

1	INTRODUÇÃO	8
1.1	Objetivo Geral	9
1.2	Objetivos Específicos	9
2	REFERENCIAL TEÓRICO	10
2.1	Metodologias tradicionais	10
2.1.1	Modelo em Cascata	10
2.1.2	Vantagens e desvantagens do modelo tradicional	12
2.2	Metodologias Ágeis	12
2.2.1	Programação Extrema (XP)	14
2.2.2	Vantagens e desvantagens do modelo ágil	15
2.3	Design Thinking	16
2.4	Lean Startup	17
2.5	Lean UX	19
2.5.1	Vision, framing and outcomes	19
2.5.2	MVP's & Experiments	20
2.5.3	Colaborative Design	21
2.5.4	Feedback and research	22
2.5.5	Mudanças Organizacionais	23
3	METODOLOGIA	24
3.1	Declaração do Problema	24
3.2	Declaração de Premissas	25
3.3	Declaração de Hipótese	25
3.4	Criação e teste de MVPs	25
3.5	Desenvolvimento do Aplicativo	26
4	PROCESSO LEAN UX	27
4.1	Contexto e declaração do problema	27
4.2	Premissas	27
4.3	Hipótese	28
4.4	Protótipo 1	32
4.4.1	Resultado da Avaliação	35
4.5	Prototipo 2	37
4.5.1	Resultado da Avaliação	40
4.6	Prototipo 3	42
4.6.1	Resultado da Avaliação	45
4.7	Produto Final	46

5	CONCLUSÃO	50
5.1	Limitações do trabalho	50
5.2	Trabalhos Futuros	51
	Referências	52
	ANEXOS	53

1 INTRODUÇÃO

Os métodos tradicionais de desenvolvimento de software surgiram entre as décadas de 80 e 90, em um contexto onde havia várias limitações: A internet e a infraestrutura eram bem menos desenvolvidas; Os softwares eram distribuídos em mídias físicas, como CDs, disquetes e HDs; Não havia tantos “frameworks” e as ferramentas que suportavam o desenvolvimento de software como há hoje em dia; O custo de se realizar alterações era muito alto e a distribuição era limitada.

Logo, fazia sentido naquela época planejar e documentar todo o software antes de começar seu desenvolvimento. Porém, esta abordagem apresenta diversos problemas que elevam os custos e a dificuldade de se desenvolver software, como: Falta de clareza nos requisitos; Requisitos mudam durante o processo de desenvolvimento; Falta de envolvimento dos clientes durante o desenvolvimento; Dificuldade de precisão ao estimar prazos; Pouco tempo para testar o software e alta carga de documentação.

Como agravante, no final de todo o processo geralmente o produto não atende os reais requisitos dos consumidores.

Buscando superar essas limitações dos métodos tradicionais de desenvolvimento de software em um contexto onde a pressão do mercado por inovação, produtividade e qualidade cresce a cada dia, foram desenvolvidas as metodologias ágeis, que propõem mudanças de prioridade no processo de desenvolvimento de software, adotando princípios (BECK et al., 2001):

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Lean UX é uma nova abordagem ágil, não apenas de desenvolvimento de software, mas também uma abordagem de como se resolver problemas de negócio, que visa entregar soluções de valor para o cliente final e maximizar a eficiência das equipes envolvidas em seu desenvolvimento.

1.1 Objetivo Geral

Demonstrar, na prática a utilização da abordagem “*Lean UX*” no contexto de desenvolvimento de software e como ela consegue contornar os problemas da abordagem tradicional usando como exemplo um aplicativo para o compartilhamento de informações entre moradores de condomínios.

1.2 Objetivos Específicos

- Apresentar as fundações, princípios e conceitos da abordagem “*Lean UX*”;
 - “*Lean UX*” é baseado em três fundações: “*Design Thinking*”; Desenvolvimento Ágil de Softwares e “*Lean Startup*”, também propõe uma gama de princípios e conceitos que derivam de suas fundações. No decorrer do trabalho serão apresentados detalhadamente cada uma das fundações e princípios que compõe o método.
- Apresentar os processos, ferramentas e artefatos prescritas pelo método;
 - Os processos do Lean UX acontecem de forma iterativa, e podem ser divididos em quatro atividades:
 - * Vision, framing and outcomes: Declaração de premissas, hipóteses, personas e funcionalidades. . .
 - * Design Colaborativo: Envolver todos que podem contribuir com o processo de design, Design Studio, Style Guides. . .
 - * “*MVPs*” e Experimentos: Determinar o foco do produto, criar versão mínima do produto, Prototipagem e Experimentação. . .
 - * Feedback e Pesquisa: Como coletar a resposta do mercado em relação ao produto. . .
- Produzir um protótipo de software funcional a partir da utilização da abordagem “*Lean UX*”.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta a fundamentação teórica necessária para que se possa melhor compreender o contexto e a proposta do trabalho.

É apresentado os principais paradigmas da engenharia de software, um exemplo de modelo de cada um deles e também a metodologia “Lean UX” e os conceitos utilizados por esta metodologia.

2.1 Metodologias tradicionais

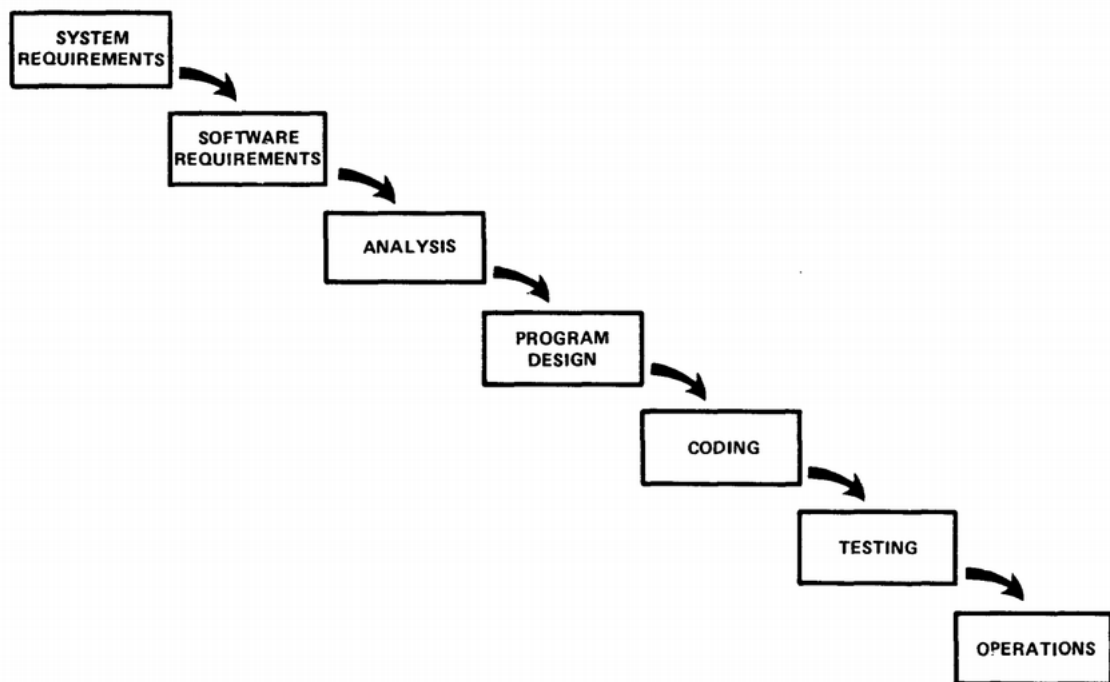
Ciclos de vida do desenvolvimento de software (SDLCs) são processos utilizados para se desenvolver software e manter softwares (BENDER RPT INC., 2003). Existem duas metodologias de SDLC que são amplamente utilizadas no mercado: As metodologias tradicionais e as metodologias ágeis.

As metodologias tradicionais, são frequentemente considerados como métodos rígidos e lineares, orientadas à um planejamento prévio, requerendo que os requisitos iniciais, documentação e planos detalhados sejam definidos antes do desenvolvimento do software. Três exemplos de metodologias tradicionais são: O modelo em cascata (ROYCE, 1970), o modelo em espiral (BOEHM; GROUP, 1988) e o RUP (KRUCHTEN, 2004).

2.1.1 Modelo em Cascata

O modelo de desenvolvimento em cascata foi primeiramente introduzido em (ROYCE, 1970) existem diversas modificações deste modelo, inclusive propostas por Royce no mesmo artigo, o modelo original sem modificações é dado como mostra a Figura 1 abaixo.

Figura 1 – Modelo Cascata não modificado



Royce, 1970

As etapas deste modelo são descritas em (PETERSEN; WOHLIN; BACA, 2009) conforme abaixo:

- **System and software requirements:** As necessidades dos clientes são identificadas e documentadas em um alto nível de abstração. Depois, os requisitos são refinados para que possam ser usados como entrada para a fase de design e implementação.
- **Analysis:** É verificado se todos os requisitos são compreendidos, acordados e documentados. Além disso, verifica-se se as partes interessadas relevantes são identificadas e se a solução apoiaria a estratégia de negócios.
- **Design:** O processo de design estabelece a arquitetura do software
- **Coding:** Os desenvolvedores realizam testes unitários básicos antes de passar o código desenvolvido para a fase de teste.
- **Testing:** Após o passo 4 todos as unidades são integradas e o software integrado é testado para se descobrir defeitos no sistema, nesta fase, a integração do sistema é testada em relação à qualidade e aspectos funcionais.
- **Operations:** O último passo é a instalação e migração do software para a máquina do cliente e a realização de suporte e manutenção quando necessário.

Como pode ser visto este modelo possui uma característica linear, cada passo é executado sequencialmente, a saída produzida por uma fase é utilizada como entrada para a próxima fase, existem modificações deste modelo que permitem realizar iterações entre qualquer fase do processo, mas ainda sim é um processo linear é pouco flexível.

2.1.2 Vantagens e desvantagens do modelo tradicional

- Alguns autores apontam desvantagens no modelo tradicional:
 - Fred Brooks que diz em seu artigo “*No Silver Bullet: Essence and Accidents of Software Engineering*”, que as estruturas conceituais de hoje são muito complicadas para serem precisamente especificadas previamente ao desenvolvimento e encoraja o desenvolvimento incremental de software (BROOKS; KUGLER, 1987).
 - Tom Gilb que em “*Principles of software engineering management*”, diz que nunca se deve tentar entregar um software grande e complexo de uma vez, e sim realizar várias entregas incrementais, para possibilitar a identificação de problemas e sua correção mais cedo no processo (GILB; FINZI, 1988).
- De acordo com (J. FERNANDEZ; D. FERNANDEZ, 2008) algumas vantagens apontadas pelo modelo tradicional são:
 - Os projetos tradicionais podem suportar mais facilmente equipes de trabalho distribuídas de especialistas e membros juniores devido aos requisitos bem definidos e outras documentações .
 - Os projetos tradicionais são claramente definidos com recursos, funções e requisitos bem documentados e compreendidos.
 - Poucas solicitações de mudança de escopo acontecem nos projetos tradicionais, é possível utilizar modelos pré-estabelecidos.

2.2 Metodologias Ágeis

Como alternativa às metodologias tradicionais foi criado o manifesto ágil, publicado no artigo “*Manifesto for agile software development*” em 2001, onde 17 es-

pecialistas em engenharia de software representando métodos consagrados como "Scrum" (SCHWABER; BEEDLE, 2002) e "Extreme Programming" (BECK, 1999), se juntaram para compartilhar princípios comuns. As metodologias que compartilham estes princípios são chamadas de metodologias ágeis.

Os princípios compartilhados pelos autores do Manifesto Ágil são:

- Indivíduos e interações ao invés de processos e ferramentas:
 - É importante que a equipe mantenha uma comunicação constante quando se trata de desenvolvimento de softwares. Assim o negócio se torna mais responsivo às mudanças, já que a priorização de processos e ferramentas trata a comunicação como um processo que possui hora marcada para acontecer, diminuindo a flexibilidade e a interação da equipe.
- Software executável ao invés de documentação:
 - Os métodos ágeis não descartam a documentação; ela é uma parte importante do desenvolvimento de software, porém, documentação em excesso é custosa e pode ser uma perda de tempo. É mais interessante desenvolver protótipos de baixo custo que promovem o aprendizado da equipe e a identificação de erros nas fases iniciais do projeto.
- Colaboração do cliente ao invés de negociação de contratos:
 - Apenas o cliente pode dizer o que ele quer, porém, geralmente ele não consegue especificar exatamente o que deseja logo na primeira tentativa, e pode mudar de ideia e criar novos requisitos durante o desenvolvimento do projeto; portanto, sua colaboração é essencial para que suas necessidades sejam realmente atendidas.
- Respostas rápidas as mudanças ao invés de seguir planos:
 - Durante o desenvolvimento do projeto o cliente começa a entender mais sobre o domínio do problema e por várias razões suas prioridades podem

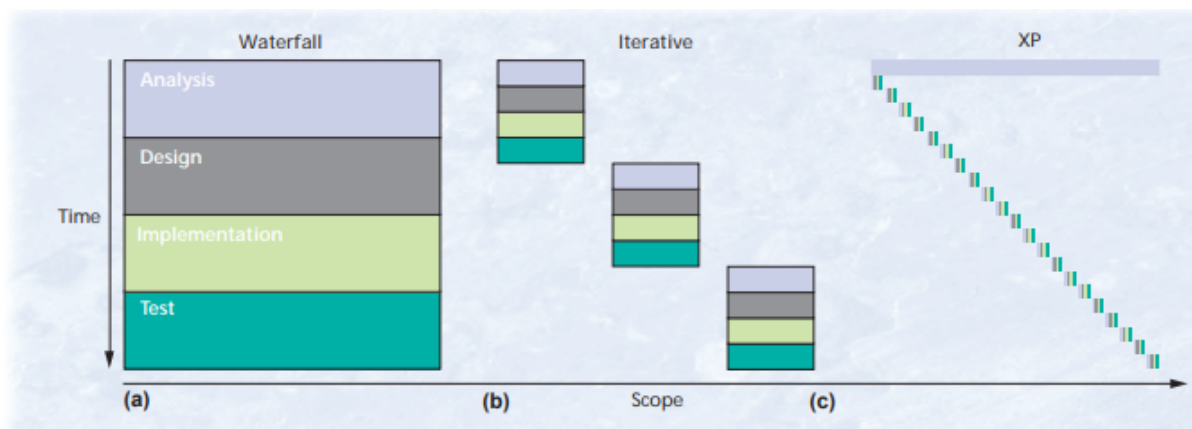
mudar. Pessoas mudam, tecnologias mudam o tempo todo e é importante que o processo de desenvolvimento de software acompanhe essas mudanças e seja flexível, não é possível se prever todas as variáveis para se fazer um bom planejamento.

2.2.1 Programação Extrema (XP)

A metodologia “*Extreme Programming*” (XP) foi publicada em (BECK, 1999) segundo o autor esta metodologia pode ser vista como a evolução do modelo cascata, modelo este que devido ao seu longo ciclo único de desenvolvimento (análise, design, implementação e teste) muitas vezes falhava em identificar e solucionar as necessidades dos clientes, logo este grande ciclo foi dividido em ciclos menores, incrementais, lembrando o modelo espiral.

XP leva este conceito ao extremo, ele foca em entregar pequenos incrementos do projeto durante todo o processo de desenvolvimento, tornando extremamente mais flexível e adaptável do que as metodologias tradicionais, a Figura 3 abaixo apresenta este conceito.

Figura 2 – Comparação metodologia Cascata (a) x Espiral (b) x XP (c)



Beck, 1999

As práticas da metodologia XP (BECK, 1999) são:

- **Planejamento:** Consumidores decidem o escopo e o intervalo dos lançamentos baseado nas estimativas dos programadores.
- **Pequenos Lançamentos:** O sistema é colocado em produção dentro de poucos meses, antes de todo o problema ser resolvido, novos lançamentos são feitos frequentemente.

- **Metáfora:** O formato do sistema é definido por uma metáfora ou conjuntos de metáforas compartilhadas entre os consumidores e os programadores.
- **Design Simple:** A cada momento o design roda todos os testes, comunica tudo que os programadores querem comunicar, não contém código duplicado, e possui a menor quantidade de classes e métodos possíveis.
- **Testes:** Programadores escrever testes unitários minuto a minuto. Estes testes são coletados e eles devem executar corretamente.
- **Refatoração:** O design do sistema é evoluído através de transformações de designs existentes que mantêm todos os testes executando.
- **Programação em pares:** Toda produção de código é escrita por duas pessoas em uma tela.
- **Integração contínua:** Novo código é integrado com o sistema atual em não mais que duas horas. Quando integrado o sistema é construído do começo e todos os testes devem passar ou as mudanças são descartadas.
- **Responsabilidade Coletiva:** Todo programador melhora todo código em qualquer lugar do sistema a qualquer momento se houver a oportunidade.
- **Consumidor participativo:** O consumidor se senta com a equipe o tempo todo.
- **40 horas semanais:** Ninguém pode trabalhar uma segunda semana consecutiva além do horário delimitado. Mesmo trabalhos extras isolados usados muito frequentemente e um sinal de problemas profundos que devem ser endereçados.
- **Espaço de trabalho aberto:** A equipe trabalha em uma grande sala com pequenos cubículos. Programadores em pares trabalham em computadores no centro.
- **Apenas regras:** Sendo parte de uma equipe que utiliza a metodologia de Programação Extrema você se inscreve para seguir as regras. A equipe pode mudar as regras a qualquer momento.

2.2.2 Vantagens e desvantagens do modelo ágil

- Segundo (J. FERNANDEZ; D. FERNANDEZ, 2008) entre os benefícios de se utilizar os métodos ágeis podemos citar:

- Os requisitos são fluidos e as mudanças são encorajadas e fazem parte do processo, possui maior flexibilidade.
 - Aproxima o cliente do processo de desenvolvimento.
 - Todas as partes interessadas recebem “feedback” rapidamente.
 - Cria documentação baseada no valor gerado para o cliente.
 - O processo é realizado em pequenos incrementos que permitem identificar problemas em fases iniciais do desenvolvimento.
- Algumas desvantagens:
 - Projetos ágeis descobrem os requisitos do projeto em iterações, portanto, reduzindo e eliminando a incerteza. Por causa disso, os métodos ágeis tendem a ter um risco maior em comparação aos projetos tradicionais.
 - O nível de comprometimento dos membros do projeto ágil deve ser maior que o dos membros tradicionais da equipe, pois, eles são chamados para assumir um papel maior em seus projetos.

2.3 Design Thinking

Em (BROWN; WYATT, 2010) Design Thinking é apresentado como uma abordagem que foca não em desenhar produtos, mas em desenhar experiências de usuário, é uma forma de design alternativa às tradicionais, então para distinguir esta forma de design foi dado a ele o nome de Design Thinking.

Esta abordagem explora as capacidades que todo ser humano possui, mas que são negligenciadas por práticas convencionais de solução de problemas. Não se concentra apenas na criação de produtos e serviços centrados no ser humano, mas o processo também é profundamente humano.

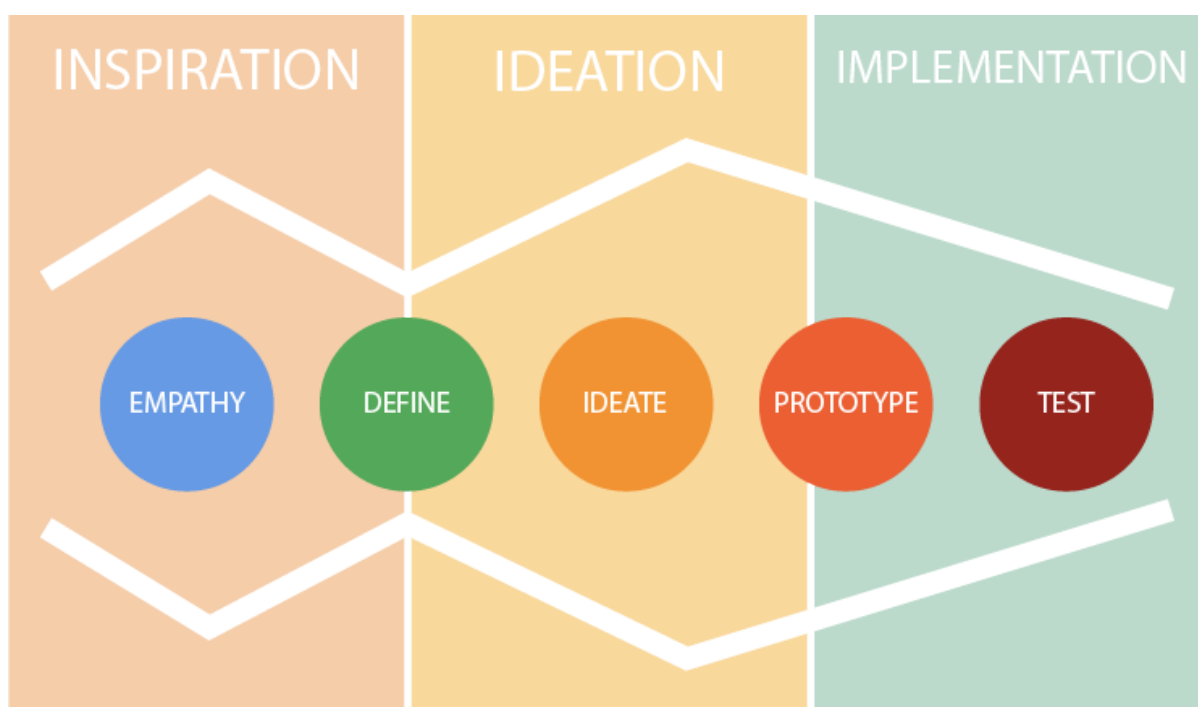
Design Thinking depende da capacidade de ser intuitivo, contruir e ideias que contenham significado emocional e serem funcionais, reconhecer padrões e de se expressar de outras formas que não sejam apenas palavras ou símbolos, é uma forma de combinar o raciocínio lógico e emocional no desenvolvimento da experiência de usuário.

O processo de design thinking é melhor pensado como um sistema de sobreposição de espaços, em vez de uma sequência de etapas ordenadas. Existem três espaços para manter em mente: inspiração, ideação e implementação. Pense na inspiração como o problema ou a oportunidade que motiva a busca de soluções; ideação como o

processo de gerar, desenvolver e testar idéias; e implementação como o caminho que leva da fase do projeto para a vida das pessoas.

A razão para chamar esses espaços, em vez de passos, é que eles nem sempre são realizados sequencialmente. Os projetos podem passar por inspiração, ideação e implementação mais de uma vez, à medida que a equipe refina suas ideias e explora novas direções. Não é de surpreender que o design thinking possa parecer caótico para quem o faz pela primeira vez. Mas, ao longo da vida de um projeto, os participantes percebem que o processo faz sentido e alcança resultados, embora sua forma se diferencie dos processos lineares e baseados em marcos que as organizações normalmente empreendem.

Figura 3 – O processo Design Thinking



Stormz

2.4 Lean Startup

Em (SUDDRETH, 2011) apresenta o conceito de Lean Startup criado por Eric Ries, um empreendedor que aprendeu através da experiência e de ciclos de fracassos e sucessos como gerir uma “startup”. Hoje Ries é um empreendedor de sucesso que possui sua própria empresa multi-milionária e sua metodologia possui como fundação cinco princípios:

- Empreendedores estão em todo lugar.

- Toda e qualquer pessoa autônoma pode ser considerada empreendedora, desde os micro-negócios às grandes corporações.
- Empreendedorismo é administração.
 - O objetivo do empreendedorismo é criar um negócio sustentável, para isto, um método novo e previsível se torna necessário na nova era da tecnologia que todos estamos adentrando.
- O objetivo principal das “startups” não é servir os consumidores ou obter lucro, o objetivo principal é aprender a como criar um negócio sustentável.
 - A maioria dos empreendedores cria “startups” acreditando que possuem uma ideia revolucionária e apostam tudo no que pode ser um grande sucesso, ou um grande fracasso.
- Aprendizado validado: Crie, mensure e aprenda.
 - Ries diz que se a organização aprender o mais rápido possível o que o mercado valoriza, ela pode adaptar seu negócio e crescer com sustentabilidade, e esta metodologia é o que ele chama de aprendizado validado.

Figura 4 – O processo Lean Startup



Ignite Innovation

- A atividade fundamental de uma “startup” é transformar ideias em produtos, medir como os clientes respondem e, então, aprender se deve dinamizar ou perseverar. Todos os processos de “startups” bem-sucedidos devem ser orientados para acelerar esse ciclo de feedback.

- Contabilidade de inovação.
 - Para melhorar os resultados empresariais, precisamos nos concentrar em: como medir o progresso, como definir marcos, como priorizar o trabalho. Isso requer um tipo específico de contabilidade para “startups”.

Esta metodologia possui o objetivo de eliminar as incertezas, focar em trabalhar de forma inteligente ao invés de focar em aumentar a quantidade de trabalho, desenvolver produtos mínimos viáveis e utiliza-los para testar e validar idéias diretamente no mercado.

2.5 Lean UX

A abordagem que será utilizada como base da metodologia deste trabalho, em (GOTHELF, 2013) cita como os 3 principais fundamentos de sua metodologia os conceitos de *Lean Startup*, *Manifesto Agil* e *Design Thinking*, razão pelo qual estes tópicos foram abordados anteriormente nesta seção.

O Lean UX é focado na experiência em design e menos focado em entregas do que o UX tradicional. Requer um maior nível de colaboração com toda a equipe. O objetivo principal é focar na obtenção de feedback o mais cedo possível para que possa ser usado para tomar decisões rápidas. A natureza do desenvolvimento do *Agile* é trabalhar em ciclos iterativos rápidos e o Lean UX imita esses ciclos para garantir que os dados gerados possam ser usados em cada iteração.

Dado um problema de negócio o processo Lean UX para solucioná-lo é realizado pelas seguintes atividades:

2.5.1 Vision, framing and outcomes

- Comece com premissas em vez de requisitos.
 - Premissas são declarações em alto nível do que acreditamos ser verdade, a razão por começarmos com a declaração de premissas ao invés de levantamentos requisitos é que não podemos saber quais são os reais requisitos de um projeto de ante-mão, o que verdadeiramente podemos fazer é ter levantar suposições que precisam ser validadas pelo mercado para que o esforço feito no desenvolvimento do projeto não tenha sido em vão.
 - O levantamento de premissas pode se dar de diversas formas, em equipes colaborativas, por exemplo pode ser realizada através de “brainstorming”.

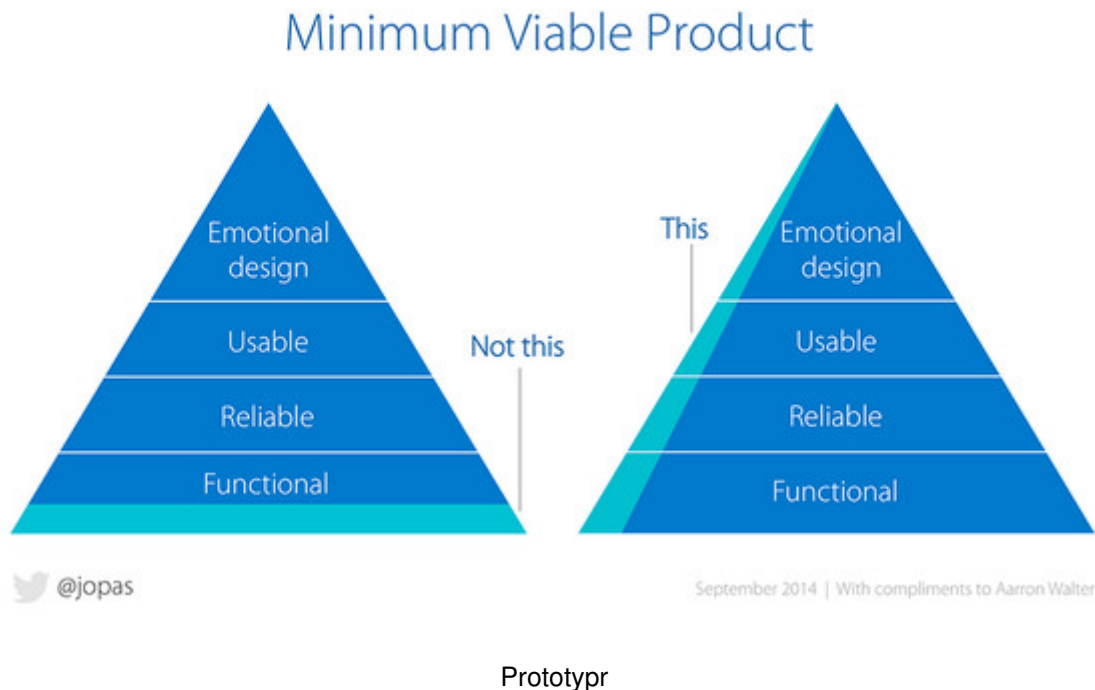
- Criar e testar hipóteses.
 - Após levantarmos nossas premissas precisamos de uma forma de as validar, hipóteses são meios de testar as premissas, a hipótese deve conter a crença que será testada, as personas que serão levadas em consideração, os resultados esperados do teste e como estes resultados serão medidos (qualitativamente ou quantitativamente).
 - Um exemplo para ilustrar o conceito: *“Acreditamos que os adicionando o botão ‘comprar em um click’ no nosso site poderá aumentar a taxa de conversão em até 30%, o que aumentaria os lucros da empresa, saberemos se estamos corretos verificando a variação da taxa de conversão em nosso site após aplicada a mudança.”*
 - **Crença:** *“Acreditamos que os adicionando o botão ‘comprar em um click’ no nosso site poderá aumentar a taxa de conversão em até 30%”*.
 - **Resultados Esperados:** *“aumentar a taxa de conversão em até 30%” e “o que aumentaria os lucros da empresa”*.
 - **Medição:** *“saberemos se estamos corretos verificando a variação da taxa de conversão em nosso site após aplicada a mudança”*.

2.5.2 MVP's & Experiments

- O que é o Produto Mínimo Viável (MVP)?
 - Após termos definido na fase do levantamento das hipóteses como vamos testar nossas premissas muitas vezes será necessário criar um protótipo que terá a função de implementar a hipótese, coloca-la em prática, o Produto Mínimo Viável (MVP) é a ferramenta que os usuários irão interagir para realizar o teste de nossas hipóteses.
 - O MVP é um produto com apenas as funcionalidades necessárias para satisfazer os consumidores iniciais e prover feedbacks para mudanças futuras.
- Determinar o foco do MVP.
 - Primeiro devemos definir o foco do MVP, seu foco pode ser definido de duas formas: Criar de valor ou aumentar o aprendizado.
 - Muitas vezes as empresas não estão preocupadas em entregar valor com o protótipo, ao invés disto, elas querem apenas descobrir o que o mercado valoriza, quais são as necessidades deste mercado, outras, o

protótipo pode ser utilizado como uma forma de começar a gerar valor, como o custo de um MVP é baixo é uma ótima forma para começar a gerar valor com um risco baixo.

Figura 5 – O foco de um MVP



2.5.3 Colaborative Design

Lean UX é um processo colaborativo que reúne Designers e Não-Designers para participarem do processo de criação, isto permite que surjam ideias melhores e mais completas do que advindas de contribuições individuais.

Lean UX aumenta a participação e responsabilidade da equipe pelo seu trabalho, dando oportunidade para que todas as opiniões sejam ouvidas de forma mais rápida, logo nas fases iniciais do processo.

As principais técnicas utilizadas nesta fase são:

- **Design Studio:** Um exercício colaborativo onde, geralmente onde uma pequena equipe em torno de 5 a 10 pessoas se reúne e desenha junto protótipos de soluções relacionadas a um problema e realizam iterações para refinar seus resultados, pode ser realizado de forma que reúna tanto equipes presenciais quanto equipes remotas geograficamente distribuídas, o processo envolve as seguintes atividades:

- Levantamento do problema e as restrições;
 - Geração de ideias (divergência);
 - Apresentação e crítica;
 - Seleção e refinamento;
 - Solução coletiva.
- **Style Guide:** Uma ferramenta criada e utilizada por toda equipe que possui padrões, está constantemente sendo atualizada com informações importantes, elementos que são frequentemente utilizados na contrição de um produto que podem ser utilizadas por todos os membros, basicamente, se contém pixels deve ir pro “style guide”, muitas grandes empresas utilizam “wikis” como forma de implementar seus “style guides”.

Os princípios que regem o Design Studio são:

- **Conversa como poderosa ferramenta:** A conversa contínua entre os colegas de projeto estimula a transparência e a confiança, identificação de requisitos e oportunidades de melhorar o trabalho sendo realizado.
- **Criação colaborativa:** A utilização de Design Studio para surgir com soluções criativas e colaborativas e Style Guides que dá autonomia para o time de desenvolvedores entregarem novas funcionalidades para o projeto e evita que eles percam tempo tentando “reinventar a roda”, ou seja, tentando criar algo que já foi criado e está disponível para todos poderem utilizar.

2.5.4 Feedback and research

- Esta é a fase é o ultimo passo do ciclo do processo Lean UX, onde o MVP é colocado em prática para cumprir seu propósito: Ser testado e gerar feedback dos usuários com intuito de validar ou invalidar as premissas.
- Antes de tudo, é preciso saber que existem diversas formas de coletar as respostas dos usuários, para cada caso uma metodologia se aplica melhor do que a outra, a equipe responsável deve decidir sobre qual metodologia utilizar, (GOTHELF, 2013) divide as técnicas de coleta de feedback em dois grupos:
 - Pesquisas Colaborativas;
 - * Técnicas de pesquisas colaborativas permitem que você construa um entendimento compartilhado com a sua equipe sobre um determinado assunto.

– Pesquisas Contínuas.

- * Técnicas de pesquisas contínuas permitem que construir pequenos estudos qualitativos informais em cada vez que você realiza uma iteração da pesquisa.

2.5.5 Mudanças Organizacionais

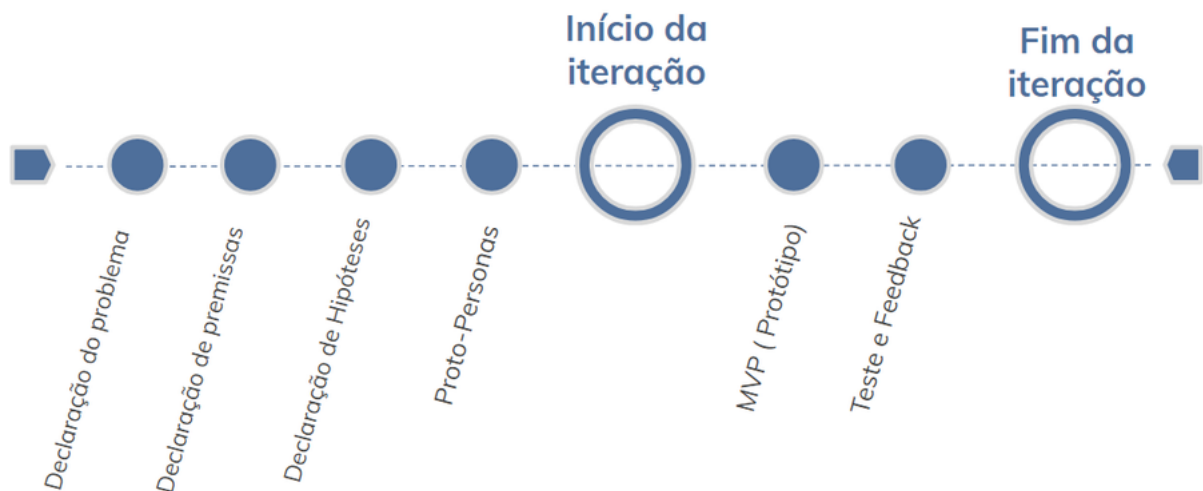
Lean UX propõe que as organizações interessadas em trabalhar com esta metodologia realize mudanças organizacionais que reforcem alguns valores como:

- Mudança de foco em saídas para foco em resultados;
- De pessoas que exercem funções limitadas para exercer capacidades colaborativas;
- Abraçar novas habilidades;
- Criar equipes multifuncionais;
- Crie equipes pequenas;
- Criar espaços de trabalho abertos e colaborativos;
- Não confiar em heróis, aqueles que acreditam que conseguem realizar tudo sozinho sem compartilhar nada;
- Priorize velocidade sobre estética;
- Resolução de problemas com base em valor;
- Mudança na cultura da empresa;
- Realizar mudanças sendo realista, considerando o contexto em que a empresa se encontra.

3 METODOLOGIA

A metodologia para o desenvolvimento deste trabalho foi dividida como mostrada na figura abaixo:

Figura 6 – Metodologia



3.1 Declaração do Problema

O primeiro passo da aplicação da metodologia Lean UX é a declaração de problema do negócio, não é possível solucionar um problema sem antes defini-lo primeiro, este é o ponto de início para se criar uma solução, a declaração de problema cria um foco claro para o trabalho a ser realizado.

Para realização deste passo tomamos como base uma empresa fictícia que esta interessada em contratar um serviço terceirizado para melhorar a qualidade de seu produto.

A declaração de problema é composta de 3 elementos:

- Os objetivos do produto ou sistema;
- O problema que as partes interessadas do negócio querem que seja resolvido (onde os objetivos não estão sendo alcançados);
- Um pedido explicito de melhoria que não dite uma solução especifica.

Um template para declaração de problemas poderia ser da seguinte forma:

*Nosso **[produto/serviço]** foi projetado para atingir **[esses objetivos]**. Observamos que o **[produto/serviço]** não está atingindo **[essas metas]**, o que está causando **[esse efeito]** adverso em nossos negócios. Como podemos melhorar o **[pro-***

duto/serviço] para que nossos clientes tenham mais sucesso [com base nesses critérios mensuráveis]?

3.2 Declaração de Premissas

Tendo a declaração do problema a ser resolvido é possível levantar as premissas necessárias para criar um ponto inicial no desenvolvimento do produto, neste trabalho serão levantadas premissas em relação a quem são nossos usuários, para que seja possível criar um produto personalizado que atenda as necessidades do publico alvo.

As premissas serão levantadas a partir das seguintes perguntas:

- Quem são os usuários?
- Onde nosso produto se encaixa em sua rotina?
- Quais funcionalidades são essenciais?
- Como nosso produto deve parecer e se comportar?

São poucas perguntas que mas que podem ser muito exploradas e revelar diversas crenças que são essenciais de serem validadas.

3.3 Declaração de Hipótese

Após o levantamento das premissas uma sentença representando a hipótese que permita validar ou invalidar as premissas será formulada, como parte desta etapa também serão levantadas hipóteses de quem são as personas do nosso puiblico alvo, denominadas de proto-personas.

3.4 Criação e teste de MVPs

Para testar as premissas e a hipóteses são utilizados protótipos que representam os MVPs do produto final, eles são produzidos em um software chamado *Balsamiq*, que permite criar telas de aplicações interativas, o que torna muito mais fácil e prático do que desenvolver utilizando programação.

Para testar os protótipos uma pequena quantidade de voluntários (5 pessoas) realizam um teste para cada protótipo, que são nada mais nada menos que passos que os usuários deve realizar dentro do aplicativo (o teste está disponível para vizualização na seção Anexos) e ao fim do teste eles respondem perguntas sobre a qualidade do produto de acordo com a visão deles, identificando problemas e dando sugestões do que deveria ser melhorado.

Ao fim de cada teste uma análise sobre as respostas coletadas é realizada para identificar problemas e soluções baseadas na resposta dos usuários que testaram o protótipo, as soluções são implementadas em um novo protótipo e enviadas novamente para teste pelas mesmas pessoas, este passo acontece o numero de vezes necessário para a criação de um protótipo satisfatório, que será utilizado para o desenvolvimento do aplicativo final na fase seguinte.

3.5 Desenvolvimento do Aplicativo

A ferramenta *Android Studio* com a linguagem *Java* será utilizada para o desenvolvimento do aplicativo final após a fase anterior, para desenvolve-lo é tomado como base o último protótipo desenvolvido, que deve representar uma versão satisfatória com base no feedback dos usuários, haverá diferenças entre o protótipo final e o aplicativo final, conforme o desenvolvedor achar apropriado.

O produto final desenvolvido será apenas o *front-end* do aplicativo, visto que o desenvolvimento completo do software com todas as funcionalidades implementadas junto a um banco de dados, que é necessário para aplicação comercial deste, seria muito complexo para o tempo disponível para realização deste trabalho.

4 PROCESSO LEAN UX

4.1 Contexto e declaração do problema

A FastSolutions é uma empresa que desenvolve soluções na área de TI que iniciou a seus serviços há dois anos atrás, sua equipe não é grande o suficiente para atender toda a demanda que está em crescimento, então a FastSolutions decidiu terceirizar o desenvolvimento de um de seus projetos.

Para isso ela decidiu nos contratar, a empresa está interessada em desenvolver uma solução informatizada para facilitar o compartilhamento de informações entre moradores e administradores de condomínios.

Observamos que a solução atual da FastSolutions possui um baixo nível de usabilidade, o que está fazendo com que os clientes abandonem o uso do aplicativo. Como podemos melhorar este produto para que a FastSolutions possa ter sucesso em obter uma usabilidade de qualidade para sua solução?

4.2 Premissas

- **Quem são os usuários?**

- Nossos usuários são em sua grande maioria pessoas maiores de idade;
- Possuem rotinas diversificadas umas das outras;
- Alguns sabem bem como interagir com a tecnologia enquanto outros possuem dificuldade;
- São moradores ou prestadores de serviços de condomínio;
- Alguns moram sozinhos enquanto outros moram com a família ou outras pessoas;
- Alguns são extrovertidos e outros introvertidos.

- **Onde nosso produto se encaixa em sua rotina?**

- Nosso produto pode ser usado de qualquer lugar e a qualquer momento desde que a pessoa possua um smartphone.

- **Quais problemas nosso produto resolve?**

- Nosso produto facilita o compartilhamento de informações entre os usuários do condomínio;
- Possui uma área de compartilhamento de informações sobre serviços prestados por outras pessoas que podem facilitar a vida dos moradores;

- Torna o processo de registrar reclamações e sugestões mais fácil e eficiente;
- Facilita a identificação dos membros que moram no condomínio;
- Restringe acesso aos apenas aos usuários que possuem autorização para utilizar os serviços disponibilizados por nosso aplicativo.

- **Quais funcionalidades são essenciais?**

- É essencial que os usuários possam registrar reclamações e sugestões sobre os serviços prestados pelo condomínio;
- Todos os usuários devem poder colaborar com o compartilhamento de informações que podem beneficiar todos os usuários que possuem acesso ao aplicativo;
- Os usuários devem possuir credenciais que os identifiquem tanto para o servidor do aplicativo quanto para os administradores e usuários do condomínio.
- Deve possuir um sistema de feedback para que os usuários sugiram mudanças no aplicativo de seus interesses.

- **Como nosso produto deve parecer e se comportar?**

- Nosso produto deve possuir um design moderno;
- Deve ser de fácil utilização mesmo para aqueles que não entendem de tecnologia;

4.3 Hipótese

Nós acreditamos que possibilitando o acesso à um protótipo simulando as funcionalidades e o design do aplicativo ao nosso publico alvo poderemos obter insights sobre os pontos positivos e negativos e realizar mudanças necessárias que atenderão melhor nossos usuários, mantendo o baixo custo de desenvolvimento e diminuindo as insatisfações dos usuários com realação a usabilidade, funcionalidade e design do aplicativo.

Estas são algumas proto-personas que representam nosso público alvo:

Figura 7 – Proto-Persona 1: Olivia

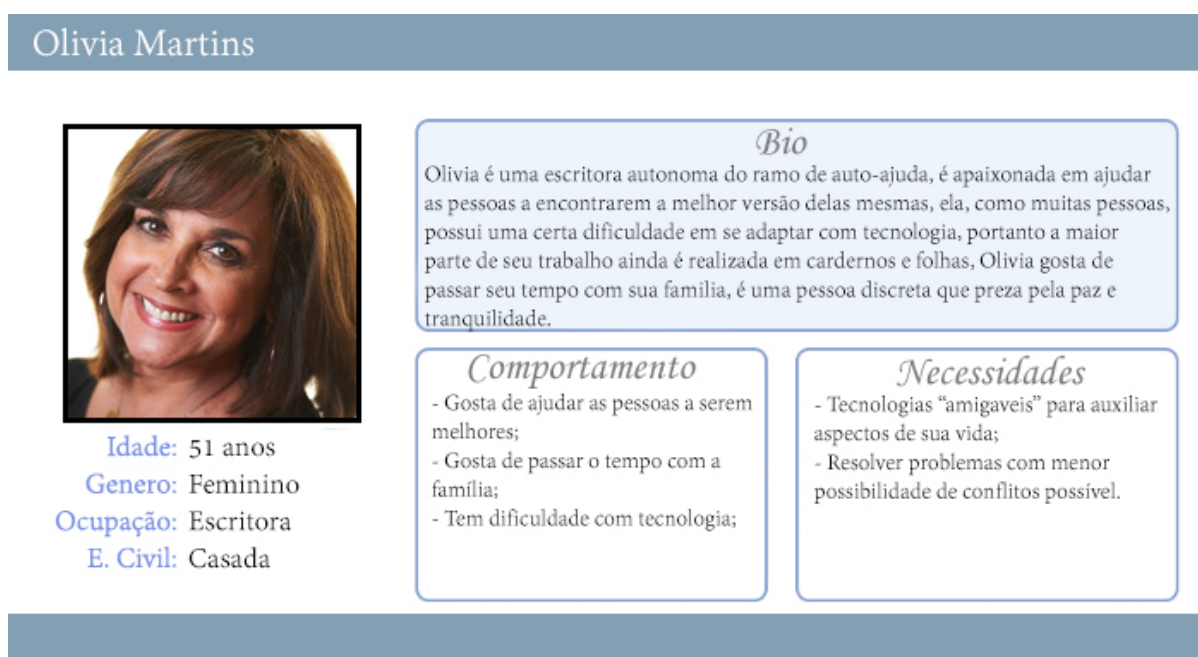


Figura 8 – Proto-Persona 2: Beto



Figura 9 – Proto-Persona 3: Luiza

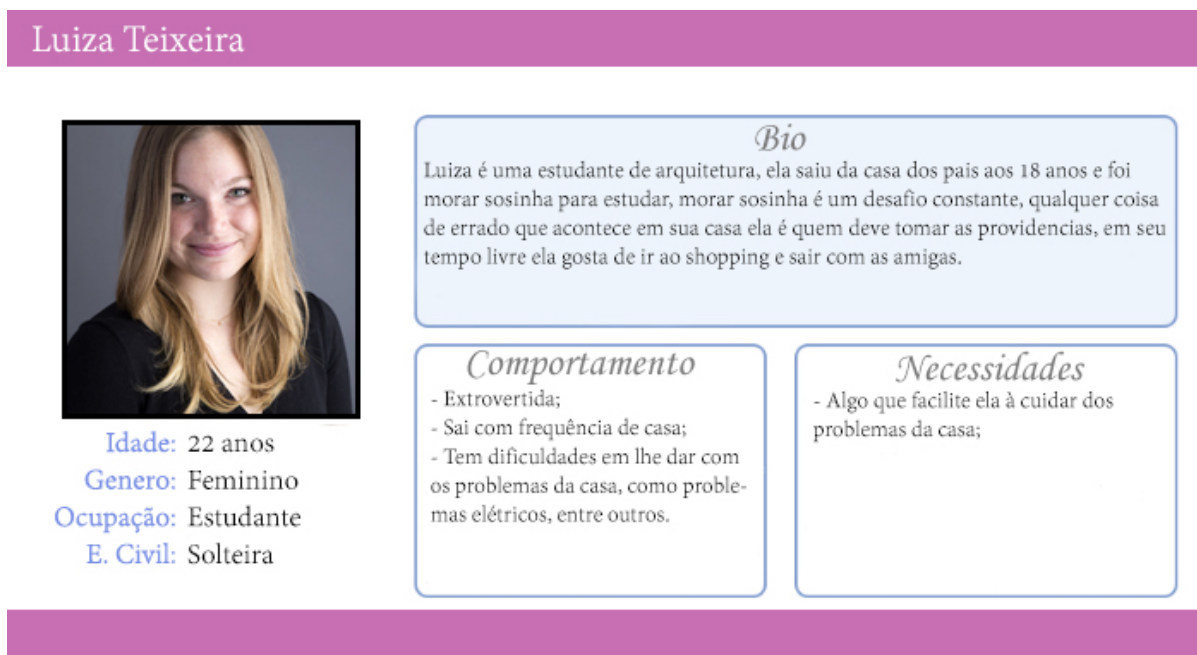


Figura 10 – Proto-Persona 4: Alfredo



Para testar esta hipótese foi realizado o seguinte experimento para cada protótipo desenvolvido:

- Um arquivo contendo o protótipo do aplicativo foi enviado para todos os participantes;
- Eles criaram uma conta;

- Se identificaram no sistema;
- Procuraram um condomínio;
- Participaram do condomínio;
- Registraram uma reclamação ou sugestão;
- Compartilharam uma informação de serviço na pagina de compartilhamento de serviços;
- Visualizaram serviços prestados pelo condomínio;
- Visualizaram informações gerais do condomínio;
- Entre outras ações especificadas no anexo presente no fim deste documento.

Cinco pessoas participaram de todos os testes, suas idades variam entre 22 e 55 anos, o anonimato delas foi preservado, o teste foi realizado por cada uma em suas casas e o resultado foi enviado remotamente.

Após realizado o experimento eles responderam um questionário que busca entender quais são os pontos positivos e negativos que foram encontrados durante o experimento, a hipótese foi validada pelo fato de que os testes trouxeram feedbacks valiosos que serviram para melhorar o aplicativo em cada iteração.

A seguir serão apresentados os protótipos desenvolvidos e um sumário dos resultados dos testes realizados pelos participantes.

4.4 Protótipo 1

Figura 11 – Protótipo 1: Funcionalidades Gerais

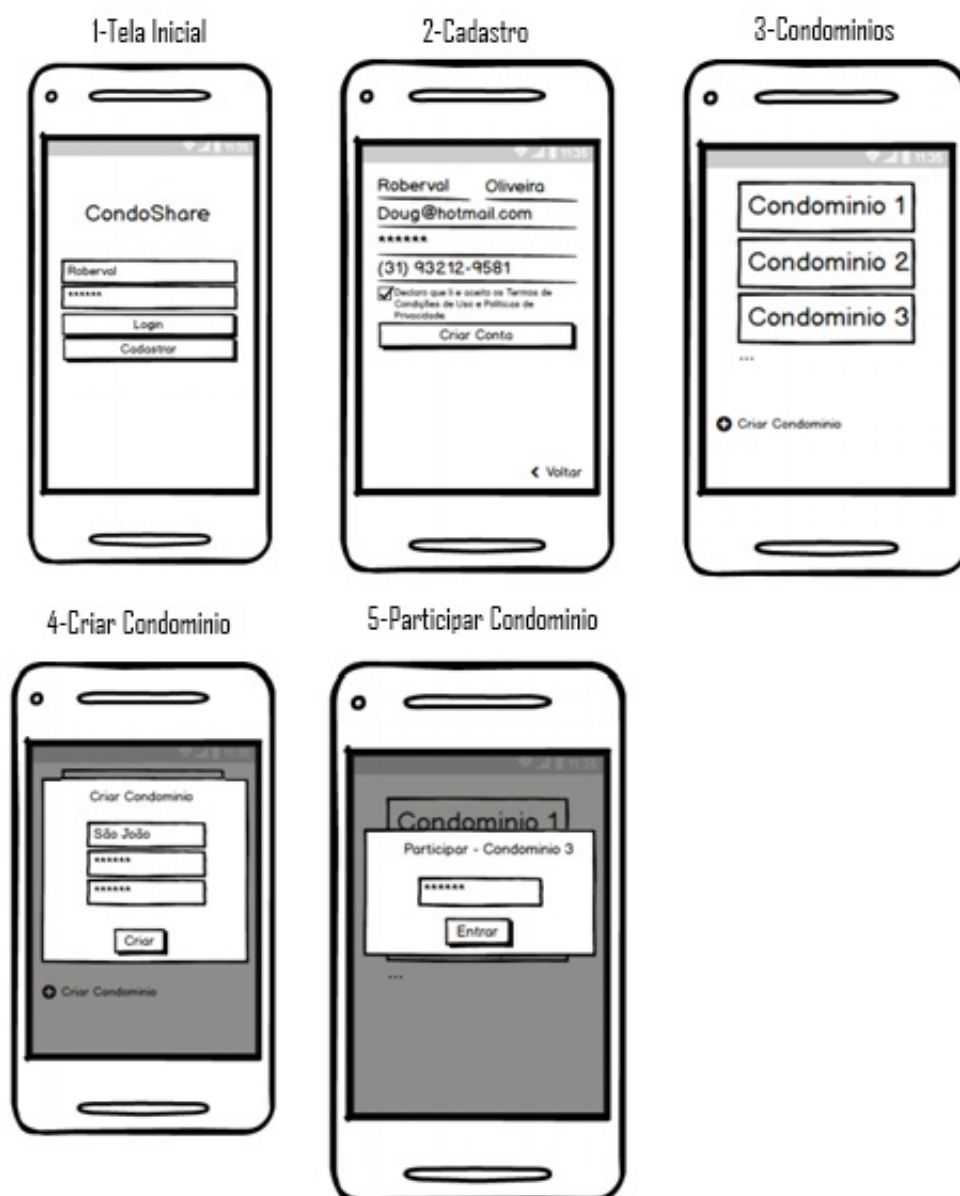


Figura 12 – Protótipo 1: Usuários - Funcionalidades

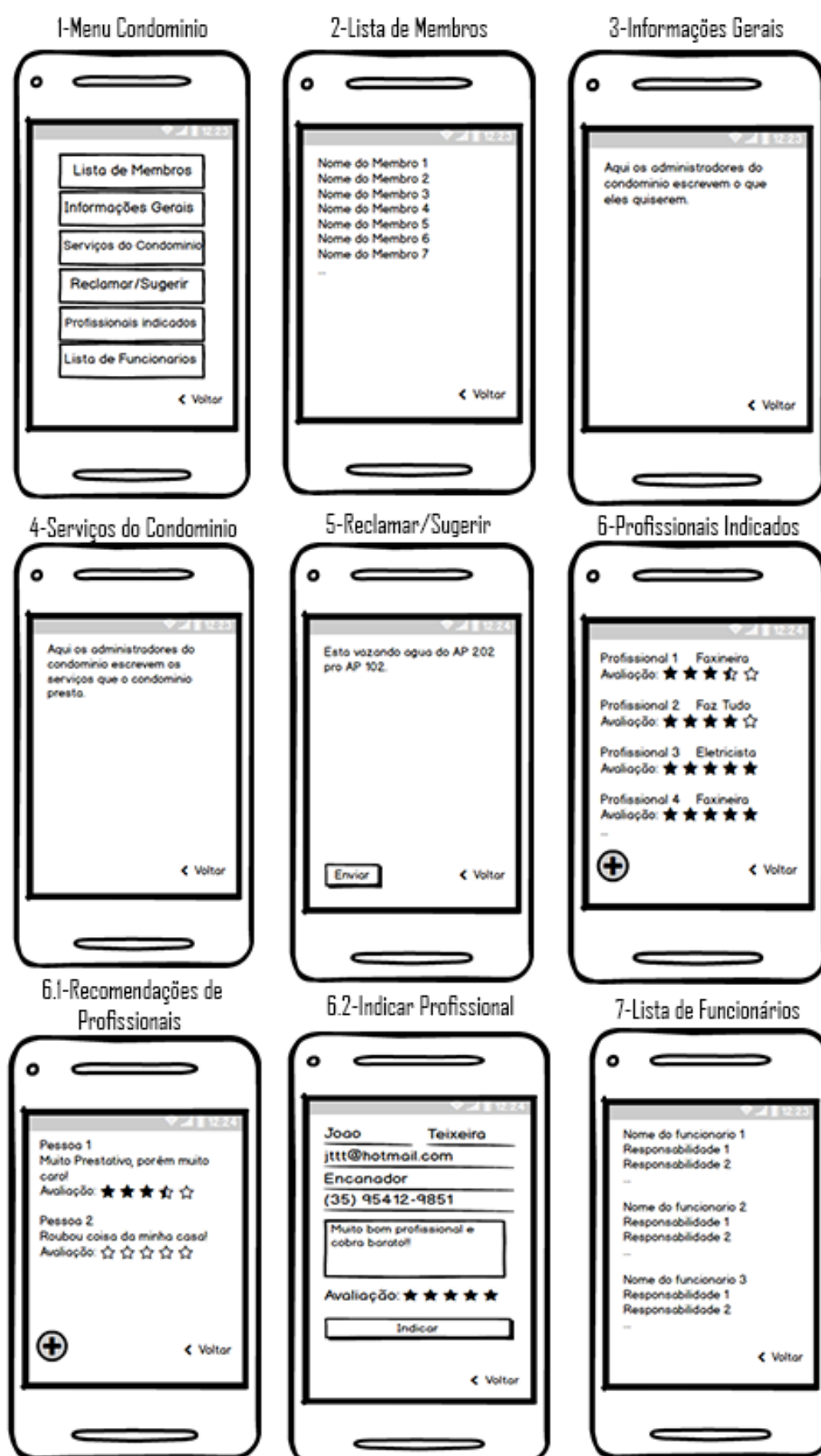


Figura 13 – Protótipo 1: Administradores - Funcionalidades



4.4.1 Resultado da Avaliação

O primeiro protótipo aos olhos dos usuários que o testaram apresentou alguns problemas de usabilidade, foi possível identificar os seguintes problemas e suas possíveis soluções:

- “Não é possível acessar o menu a partir de qualquer localização, a cada tarefa realizada é preciso ficar clicando em voltar até retornar ao menu”;
 - Solução: Acrescentar um menu presente em todas as páginas.
- “Não há uma barra de pesquisa para procurar o condomínio desejado”;
 - Solução: Acrescentar barra de pesquisa na área de condomínios.
- “Não há barra de pesquisa para procurar profissionais indicados”;
 - Solução: Acrescentar barra de pesquisa na área de profissionais indicados.
- “Como sei qual condomínio participo e qual não participo no menu de condomínios?”.
 - Solução: Acrescentar filtro para mostrar apenas condomínios de acordo com a relação do usuário com o condomínio.

Uma tentativa de corrigir os problemas apresentados pelos feedbacks foi implementada no segundo protótipo, apresentado a seguir.

4.5 Protótipo 2

Figura 14 – Protótipo 2: Funcionalidades Gerais



Figura 15 – Protótipo 2: Usuários - Funcionalidades

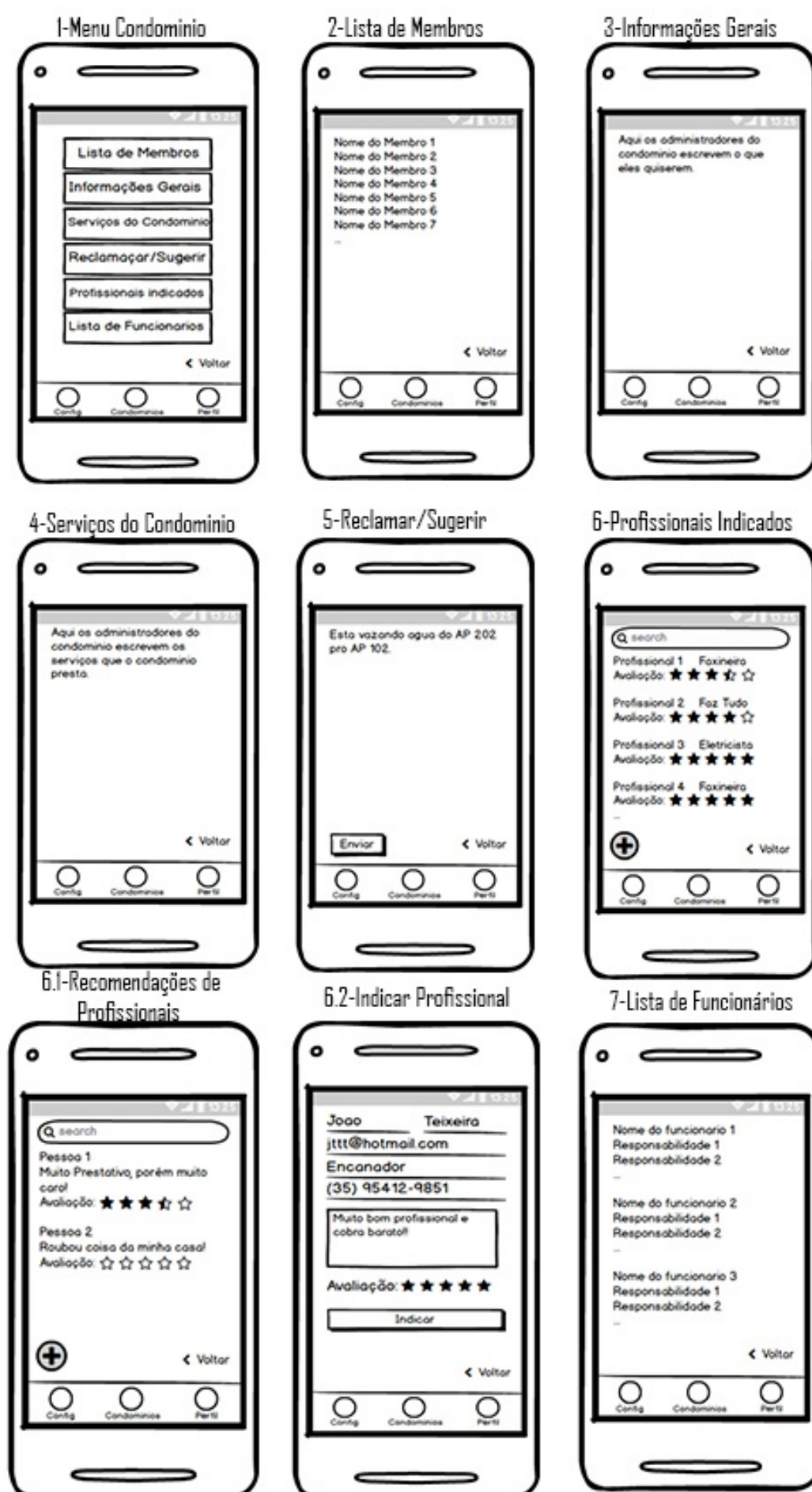


Figura 16 – Protótipo 2: Administradores - Funcionalidades



4.5.1 Resultado da Avaliação

Também foram identificados alguns problemas junto com as possíveis soluções no segundo protótipo:

- “Seria interessante se minha relação com o condomínio estivesse escrito logo de cara e eu não precisasse ir nos filtros para encontra-los”;
 - Solução: Escrever a relação do usuário com o condomínio logo no botão do condomínio e posicionar os condomínios que o usuário participa ou administra acima dos outros.
- “Percebi que a cada pagina que entro não há um título para dizer onde estou, isto acaba me deixando meio perdido no aplicativo”;
 - Solução: Adicionar uma barra superior de títulos em cada página.
- “Não é possível saber se minhas ações deram certo ou errado, o aplicativo não retorna nada após executar alguma ação.”.
 - Solução: Criar uma janela de resposta para cada ação que o usuário realizar retornando o resultado desta ação.

Mais uma vez as soluções foram implementadas em um novo protótipo que também foi testado pelos menos usuários que testaram os anteriores.

4.6 Protótipo 3

Figura 17 – Protótipo 3: Funcionalidades Gerais



Figura 18 – Protótipo 3: Usuários - Funcionalidades

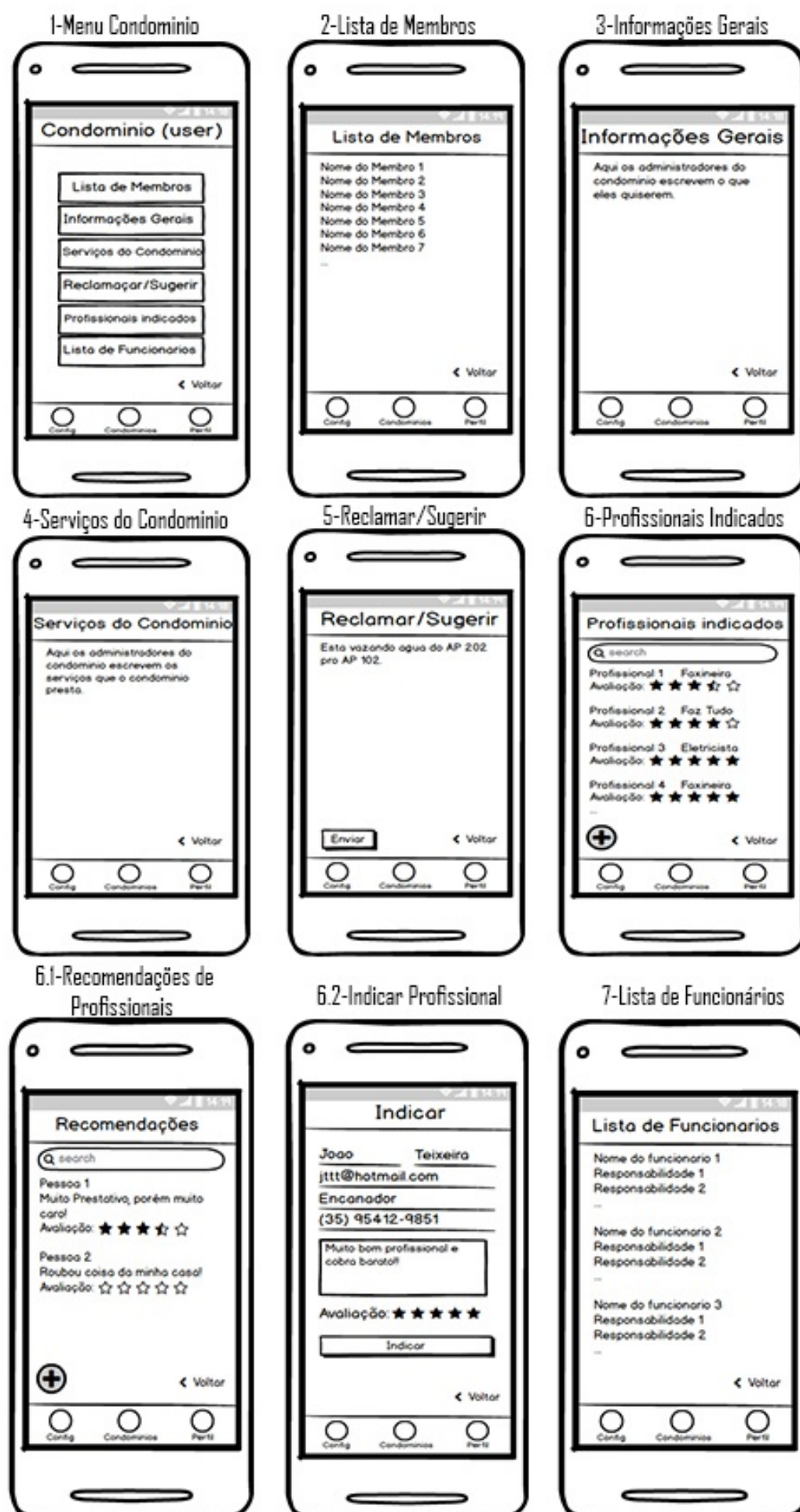


Figura 19 – Protótipo 3: Administradores - Funcionalidades

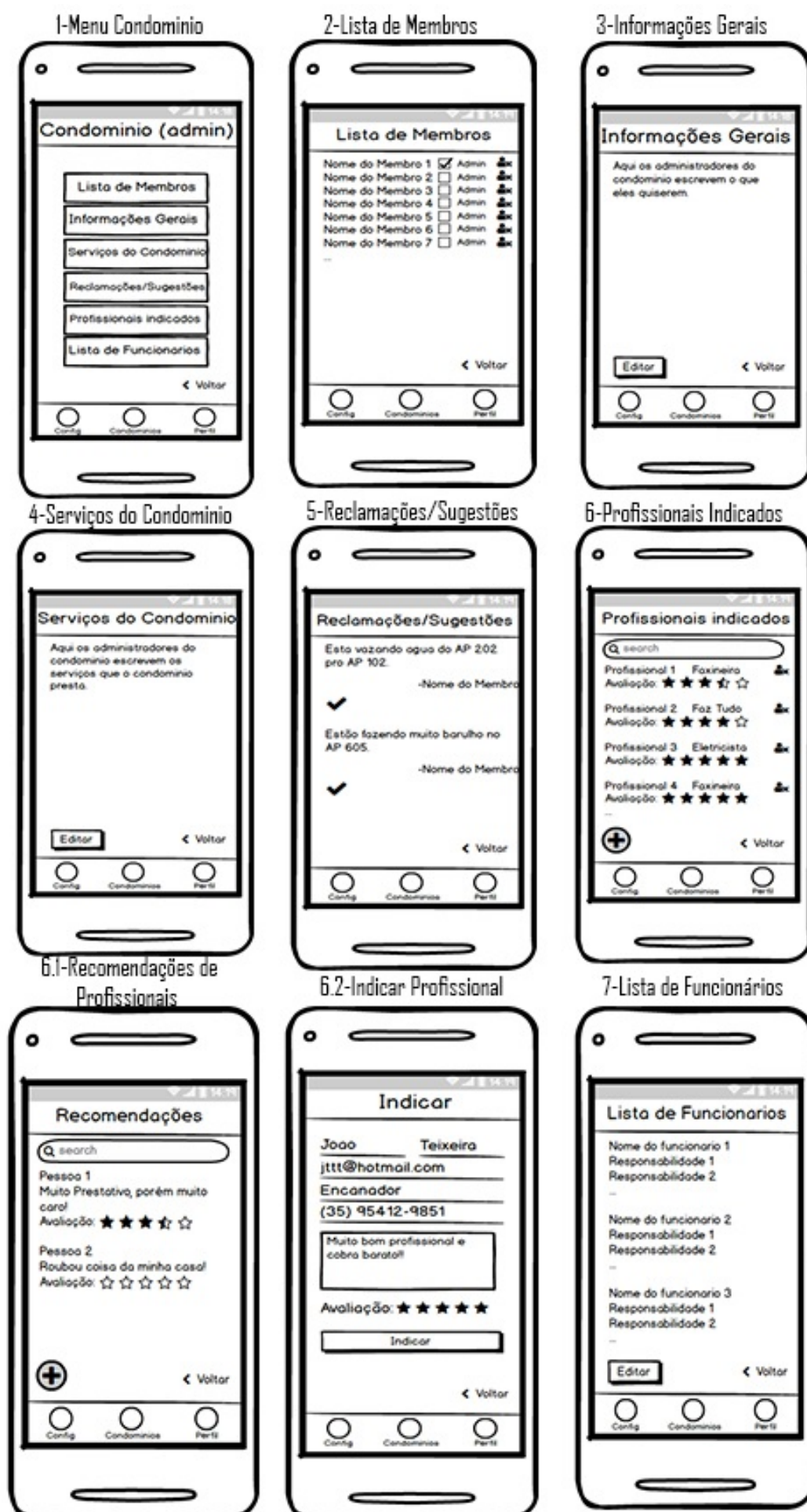
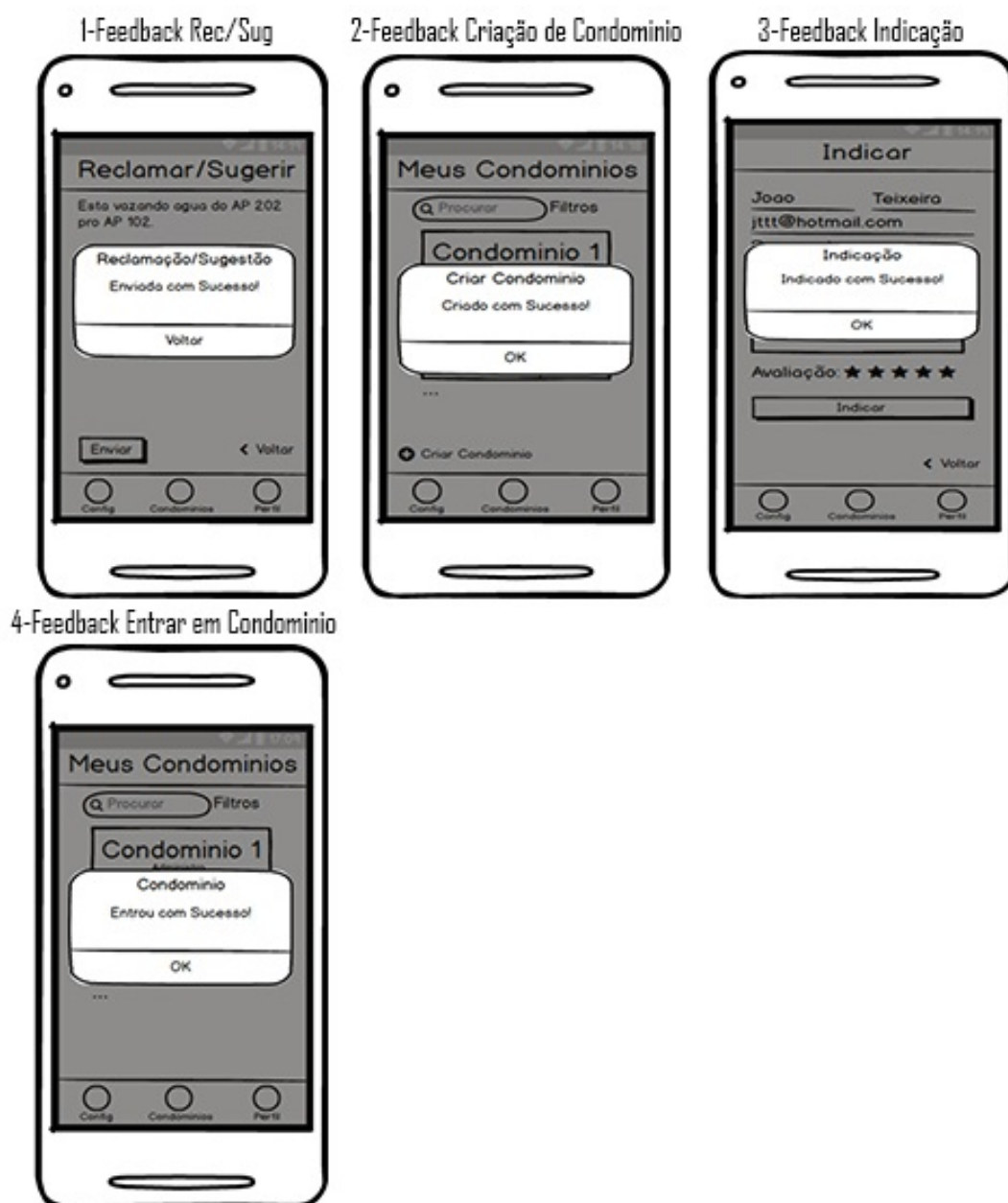


Figura 20 – Protótipo 3: Feedbacks



4.6.1 Resultado da Avaliação

Ao fim do teste do terceiro protótipo não foi detectado nenhum problema, o tempo médio de realização dos testes foi de 3 minutos, o que demonstra que o aplicativo é de fácil entendimento e utilização, isto não significa que o produto está completamente livre de erros e que não possa ser melhorado, quanto mais pessoas dispostas a avaliar o produto mais feedbacks são recebidos e mais insights são revelados, mas o resultado deste teste mostra que este protótipo possui uma qualidade maior que os anteriores e pode ser a base para o desenvolvimento do aplicativo real, que será o resultado deste trabalho.

4.7 Produto Final

Durante o desenvolvimento do aplicativo final surgiram desafios em relação à quantidade de trabalho necessária para implementar todas as funcionalidades e novas ideias que não foram previstas durante os ciclos de prototipagem, portanto, algumas funcionalidades foram removidas e uma outra adicionada devido sua importância.

As funcionalidades que tiveram de ser descartadas foram:

- Acrescentar notas e comentários para os profissionais indicados;
 - Esta funcionalidade possui uma complexidade maior do que as outras e adicioná-la demandaria muito tempo e trabalho, infelizmente teve de ser descartada.
- Seção de Reclamações/Sugestões;
 - Esta funcionalidade também foi descartada tanto pelo custo de sua implementação quanto pelo fato de que ao repensá-la ela foi considerada como imprópria para se estar no aplicativo.

A funcionalidade adicionada foi:

- Aprovações Pendentes, mesmo se o usuário souber a senha do condomínio ele não poderá entrar sem a aprovação do administrador.

Abaixo estão as telas do aplicativo separadas em 3 visões:

- Visualização Geral do Aplicativo;
 - As telas que todos usuários irão ver ao interagir com o aplicativo.
- Visualização do Administrador;
 - Mostra como as telas aparecem para o Administrador do condomínio, possibilita visualizar as funcionalidades exclusivas desta classe.
- Visualização do Usuário.
 - Mostra como as telas aparecem para os Usuários (Participantes) de um condomínio, este não possui acesso às funcionalidades do Administrador.

Figura 21 – Visualização Geral do Aplicativo

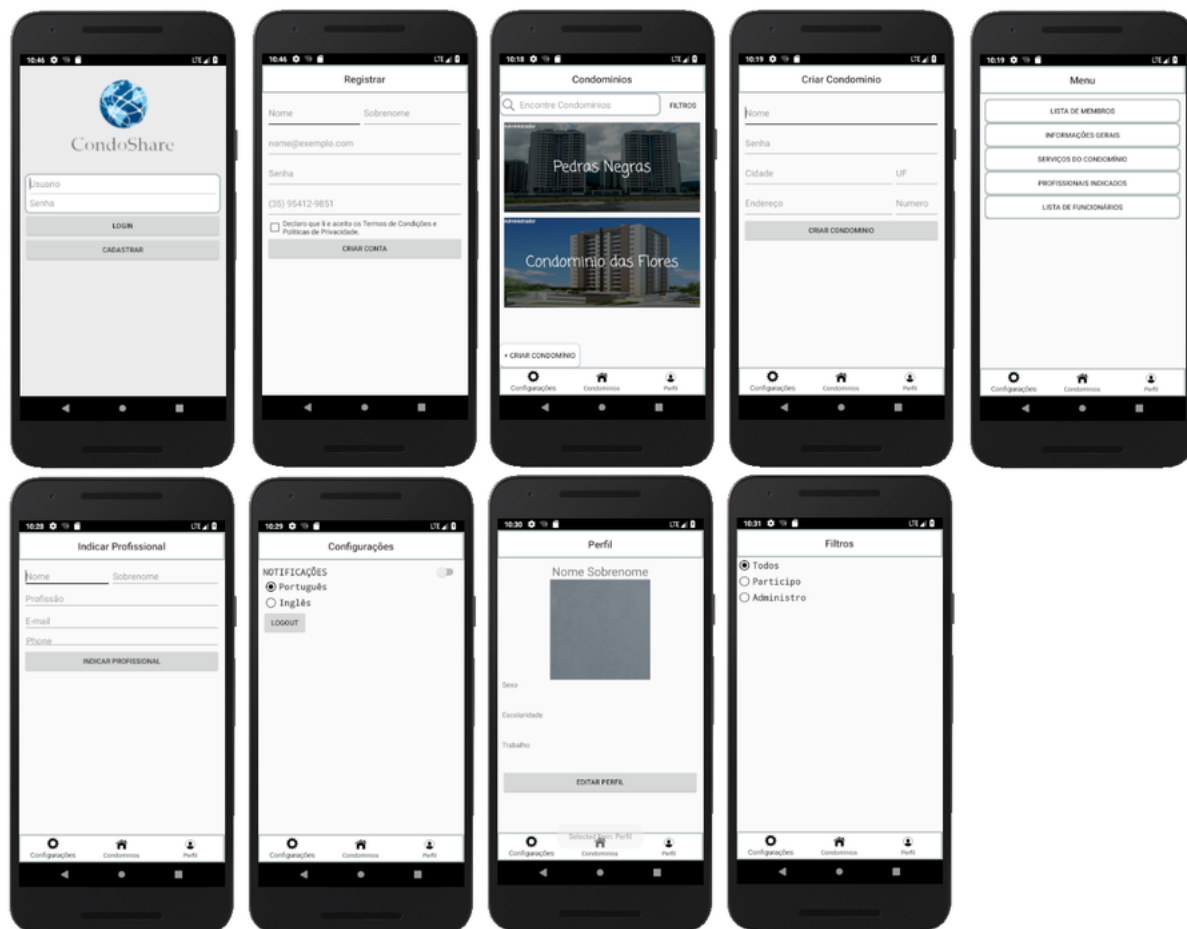


Figura 22 – Vizualização do Administrador

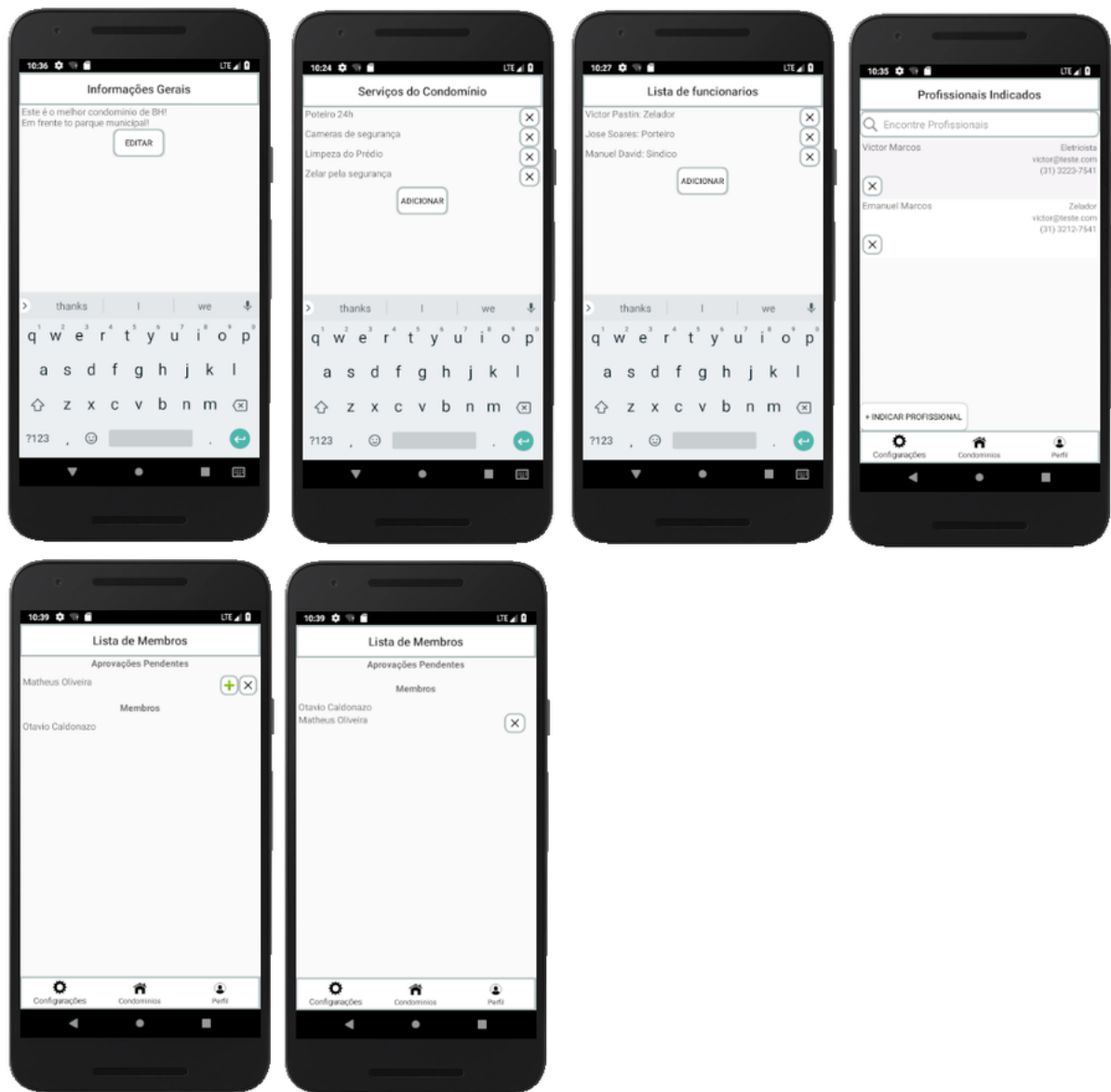


Figura 23 – Visualização do Usuário



5 CONCLUSÃO

- Neste trabalho foi demonstrado na prática como utilizar a abordagem “*Lean UX*” para desenvolver um produto;
- Foram apresentadas as fundações, princípios e conceitos que compõe esta metodologia;
- Foram apresentados os processos e os artefatos gerados em cada etapa dos processos;
- Foi produzido um protótipo de software funcional (porém não completo);

Este trabalho cumpre o objetivo geral e os objetivos específicos listados propostos.

5.1 Limitações do trabalho

No decorrer deste trabalho ao estudar Lean UX e implementá-lo no desenvolvimento deste aplicativo surgiram limitações que não permitiram a exploração completa da metodologia e do desenvolvimento do aplicativo completamente funcional, porém, as limitações sobre o desenvolvimento do aplicativo apesar de terem acontecido, não afetaram os resultados em relação ao que se pretendia demonstrar com este trabalho, as limitações encontradas foram:

- Tempo muito curto para desenvolver o servidor(*back-end*) e banco de dados do aplicativo;
- Alguns conceitos relacionados ao Design Colaborativo como a utilização de Style Guides e Design Studio não puderam ser aplicados pois são conceitos que se aplicam a equipes que estão trabalhando juntos com um mesmo propósito, e este trabalho foi realizado individualmente, a única possibilidade de trazer pessoas para colaborar foi na realização de testes dos protótipos.
- Outro ponto que este trabalho não pode abordar são as Mudanças Organizacionais, elas envolvem outro aspecto de Lean UX que necessitaria um trabalho específico voltado para esta área.

Apesar de que algumas limitações em relação a este trabalho que não puderam ser abordadas, na prática, seus conceitos foram abordados no Referencial Teórico.

5.2 Trabalhos Futuros

De modo a tornar este aplicativo completamente funcional o *back-end* pode ser desenvolvido junto às melhorias no *front-end*, seria interessante que o trabalho futuro fosse realizado em equipe(s) para demonstrar conceitos do processo Lean UX que não foram completamente abordados neste trabalho devido as suas limitações, como Design Colaborativo.

Outra oportunidade de trabalho futuro seria realizar trabalhos focados nas Mudanças Organizacionais, como realizar mudanças em organizações já existentes a fim de deixá-las mais eficientes e competitivas no cenário atual utilizando os princípios aplicados pelo Lean UX.

Referências

- BECK, K. Embracing change with extreme programming. *Computer* 32.10, p. 70 – 77, 1999.
- BECK, K. et al. Manifesto for agile software development. 2001.
- BENDER RPT INC. Systems Development Lifecycle: Objectives and Requirements. 2003.
- BOEHM, B. W.; GROUP, T. D. S. A spiral model of software development and enhancement. *Computer* 21.5, p. 61 – 72, May 1988.
- BROOKS, F.; KUGLER, H. J. No silver bullet. April 1987.
- BROWN, T.; WYATT, J. Design thinking for social innovation. *Development Outreach* 12.1, p. 29 – 43, 2010.
- GILB, T.; FINZI, S. Principles of software engineering management. *Reading, MA: Addison-Wesley*, v. 11, 1988.
- GOTHELF, J. Lean UX: Applying lean principles to improve user experience. *O'Reilly Media, Inc.*, 2013.
- J. FERNANDEZ, D.; D. FERNANDEZ, J. Agile project management—agilism versus traditional approaches *J. urnal of Computer Information Systems* 49.2 (2008): 10-17., p. 10 – 17, 2008.
- KRUCHTEN, P. The rational unified process: an introduction. *Addison-Wesley Professional*, 2004.
- PETERSEN, K.; WOHLIN, C.; BACA, D. The Waterfall Model in Large-Scale Development. *International Conference on Product-Focused Software Process Improvement.*, 2009.
- ROYCE, W. W. Managing the development of large software systems: concepts and techniques. *Technical Papers of Western Electronic Show and Convention (WesCon)*, August 1970.
- SCHWABER, K.; BEEDLE, M. Agile software development with Scrum. *Upper Saddle River: Prentice Hall*, v. 1, 2002.
- SUDDRETH, B. *The Lean Startup - Visual Summary*. 2011. Disponível em: <<https://www.slideshare.net/KUalum/the-lean-startup-visual-summary>>. Acesso em: 28/05/2018.

Anexos

CondoShare - Teste do protótipo

Dados do Participante

Idade:

Formação acadêmica:

Trabalho:

Relacionamento:

Gênero:

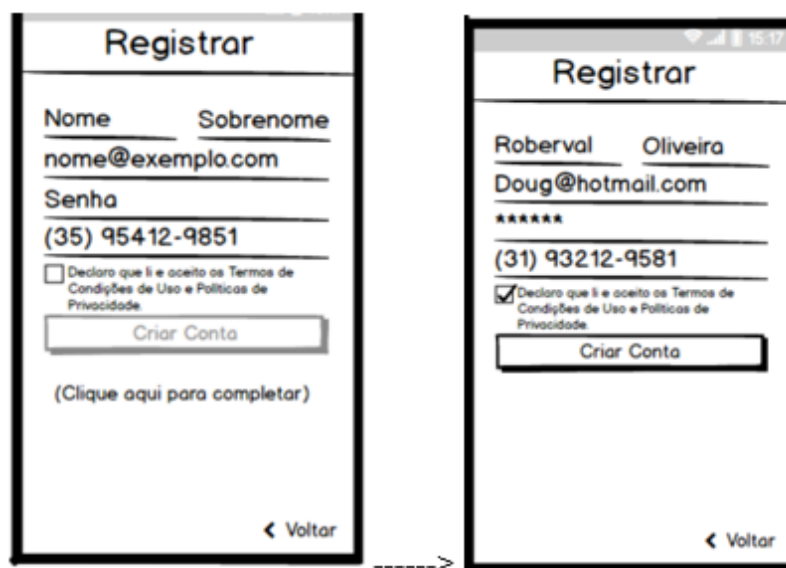
Horário de início do teste:

Horário de término do teste:

Pré-Instruções

- 1) Marque o tempo em que o teste foi iniciado;
- 2) Leia o questionário sobre o teste antes de realizá-lo;
- 3) Abra o pdf, o pdf simula o aplicativo, é um arquivo clicável, você pode clicar nas telas como se fossem o próprio aplicativo;
- 4) Realize o teste;
- 5) Marque o tempo em que o teste foi encerrado;
- 6) Preencha o questionário;

Observação: Quando for necessário inserir dados para prosseguir basta clicar em “(Clique aqui para completar)” como no exemplo abaixo:



Teste

Este teste possui o objetivo de entender quais as dificuldades que os usuários tem ao interagir com o aplicativo CondoShare, abaixo há uma bateria de tarefas que devem ser realizadas no aplicativo, realize as tarefas e depois responda o questionário de acordo com a experiência que teve com o aplicativo.

Realiza as seguintes etapas na ordem que aparecem:

- 1) Cadastre uma conta;
- 2) Realize Login;
- 3) Filtre condomínios por:
 - a) Todos;
 - b) Participo;
 - c) Sou Administrador.
- 4) Entre em um condomínio classificado como Participo e:
 - a) Visualizar os Serviços do Condomínio;
 - b) Visualizar lista de Membros do Condomínio;
 - c) Criar uma Reclamação;
 - d) Visualizar as informações Gerais do Condomínio;

- e) Visualizar a Lista de Funcionários;
 - f) Visualizar os Profissionais Indicados;
 - g) Visualizar as recomendações das pessoas sobre um profissional indicado;
 - h) Indicar um Profissional;
- 5) Visualize o seu Perfil;
- 6) Entre na área de condomínios;
- 7) Entre em um condomínio classificado como Administrador e:
- a) Dar privilégios de administrador ao Membro 1;
 - b) Remover o Membro 2;
 - c) Visualizar as reclamações que os usuários enviaram;
 - d) Visualizar as informações Gerais do Condomínio;
 - e) Visualizar a Lista de Profissionais Indicados.

Questionário

Deixe sua resposta embaixo de cada pergunta.

- 1) Você se deparou com alguma(s) funcionalidade(s) do aplicativo que não entendeu?
- a) Se sim, o que você não entendeu?
- 2) Encontrou dificuldade(s) em realizar alguma(s) tarefa(s)?
- a) Se sim, quais foram as dificuldades?
- 3) Na sua opinião, o design do aplicativo é amigável (de fácil interação)?
- a) Por quê?
- 4) Quais são suas críticas/sugestões relativas ao design do aplicativo?
- 5) Quais são suas críticas/sugestões relativas às funcionalidades do aplicativo?