

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
DEPARTAMENTO DE COMPUTAÇÃO

RICARDO MANUEL DAVID RODRIGUES  
Orientador: Saul Emanuel Delabrida Silva

**TAREFAÇA: FERRAMENTA GAMIFICADA PARA GESTÃO DE  
MORADIAS COMPARTILHADAS**

Ouro Preto, MG  
2021

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
DEPARTAMENTO DE COMPUTAÇÃO

RICARDO MANUEL DAVID RODRIGUES

**TAREFAÇA: FERRAMENTA GAMIFICADA PARA GESTÃO DE MORADIAS  
COMPARTILHADAS**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

**Orientador:** Saul Emanuel Delabrida Silva

Ouro Preto, MG  
2021



## FOLHA DE APROVAÇÃO

**Ricardo Manuel David Rodrigues**

**TAREFA: FERRAMENTA GAMIFICADA PARA GESTÃO DE MORADIAS COMPARTILHADAS**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 20 de Abril de 2021.

### Membros da banca

Saul Emanuel Delabrida Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto  
Marcelo Luiz Silva (Examinador) - Mestre - Universidade Federal de Ouro Preto  
Elton José da Silva (Examinador) - Mestre - Universidade Federal de Ouro Preto

Saul Emanuel Delabrida Silva, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 20/04/2021.



Documento assinado eletronicamente por **Saul Emanuel Delabrida Silva, PROFESSOR DE MAGISTERIO SUPERIOR**, em 22/04/2021, às 14:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0160362** e o código CRC **B57DEFFA**.

*Dedico este trabalho à minha mãe Maria Tereza David, ao meu pai Manuel Delgado Rodrigues  
e à minha irmã Andrea Raquel David Rodrigues.*

# Agradecimentos

Agradeço primeiramente à minha família pelo apoio, pois sem ela nada disso seria possível. Agradeço a todos os professores do Departamento de Computação da Universidade Federal de Ouro Preto que contribuíram para minha formação. Agradeço ao meu orientador, Saul Delabrida pela oportunidade de realizar este trabalho e pelas orientações e direcionamentos. Agradeço a todos os meus amigos em especial Manoel, João, Patrick, Daniela, Eduardo, Pablo, D. Maria e Dayanne pelos momentos de descontração e pelo apoio nos momentos mais complicados.

*“No que diz respeito ao empenho, ao compromisso, ao esforço, à dedicação, não existe meio termo. Ou você faz uma coisa bem feita ou não faz.” - Ayrton Senna -*

# Resumo

Compartilhar moradias desde sempre tem sido uma necessidade básica para o ser humano que vê neste ato a possibilidade de gerir despesas básicas como luz, água e internet sem as quais a vida torna-se um bocado difícil de "fuir". Desde a antiguidade que as moradias compartilhadas são bastante populares e este conceito vem tornando-se cada vez mais popular nos dias atuais, principalmente nas grandes cidades. O preço dos aluguéis permite aos grupos que partilham essas moradias estar menos sobrecarregados devido aos valores altíssimos que os mesmos retratam. Tarefas complicadas como os fazeres do dia a dia e a gestão das contas transformam-se em situações menos difíceis quando é possível, haver alguém que consiga controlar a gestão das contas, com o objetivo de manter a casa sempre organizada e as contas em dia. Perante este contexto, é objetivo e intenção deste trabalho, desenvolver um aplicativo para a gestão de contas e tarefas de moradias compartilhadas com o uso de gamificação para auxiliar na organização de tarefas do cotidiano.

**Palavras-chave:** Gestão de tarefas, Gamificação, Moradias compartilhadas.

# Abstract

Sharing homes has always been a basic need for the human being who sees in this act the possibility of managing basic expenses such as electricity, water and internet without which life becomes a little difficult to "run away". Since ancient times, shared housing has been quite popular and this concept has become increasingly popular today, especially in large cities. The rental price allows the groups that share these houses to be less burdened due to the very high values that they portray. Complicated tasks such as day-to-day tasks and account management become less difficult situations when it is possible to have someone who can control account management, with the aim of keeping the house always organized and the accounts up to date. Given this context, it is the objective and intention of this work, to develop an application for the management of accounts and tasks of shared housing with the use of gamification to assist in the organization of daily tasks.

**Keywords:** Task management, Gamification, Shared housing.

# Lista de Ilustrações

Figura 4.1 – Metodologia do Trabalho . . . . .	14
Figura 4.2 – Menu de Navegação . . . . .	19
Figura 4.3 – Telas de Navegação . . . . .	20
Figura 4.4 – Adicionar Tarefas - Adicionar Contas . . . . .	21
Figura 4.5 – Tela de aviso . . . . .	22
Figura 4.6 – Tela de Contas . . . . .	22
Figura 4.7 – Tela de Navegação . . . . .	23
Figura 4.8 – Tela de Autenticação . . . . .	24
Figura 4.9 – Tela de adicionar conta . . . . .	24
Figura 4.10–Tela de trocar de pontos acumulados . . . . .	25
Figura 4.11–Tela de ranking . . . . .	25
Figura 5.1 – Resultados . . . . .	29
Figura 5.2 – Comparação entre a versão 1 e 1.1 do layout . . . . .	30
Figura 5.3 – Desempenho . . . . .	31
Figura 5.4 – Lista de celulares compatíveis . . . . .	31
Figura 5.5 – Tamanho da área de toque . . . . .	32
Figura 5.6 – Baixo Contraste . . . . .	33
Figura A.1 – Questão 01 da avaliação . . . . .	38
Figura A.2 – Questão 02 da avaliação . . . . .	38
Figura A.3 – Questão 03 da avaliação . . . . .	38
Figura A.4 – Questão 04 da avaliação . . . . .	39
Figura A.5 – Questão 05 da avaliação . . . . .	39
Figura A.6 – Questão 06 da avaliação . . . . .	39
Figura A.7 – Questão 07 da avaliação . . . . .	39
Figura A.8 – Questão 08 da avaliação . . . . .	40
Figura A.9 – Questão 09 da avaliação . . . . .	40
Figura A.10–Questão 10 da avaliação . . . . .	40

# Lista de Tabelas

Tabela 2.1 – Tabela comparativa das funções dos aplicativos analisados . . . . .	6
Tabela 3.1 – Fórmula da normalização dos resultados . . . . .	9
Tabela 4.1 – Apresentação visual do método . . . . .	15
Tabela 4.2 – Apresentação visual do método - Atualização . . . . .	15
Tabela 4.3 – Requisitos Funcionais . . . . .	17
Tabela 4.4 – Requisitos Não Funcionais . . . . .	18
Tabela 4.5 – Roteiro das atividades . . . . .	27
Tabela 5.1 – Resultado obtido através do questionário SUS . . . . .	29

# Lista de Abreviaturas e Siglas

ABNT	Associação Brasileira de Normas Técnicas
BaaS	<i>Backend As A Service</i>
DECOM	Departamento de Computação
DOM	<i>Document Object Model</i>
HTML	<i>Hypertext Markup Language</i>
SUS	<i>System Usability Scale</i>
UFOP	Universidade Federal de Ouro Preto
XML	<i>Extensible Markup Language</i>

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problema e Justificativa	2
1.2	Objetivos Geral e Específicos	2
1.3	Organização do Trabalho	2
1.3.1	Estrutura da Monografia	3
<b>2</b>	<b>Revisão de Literatura</b>	<b>4</b>
<b>3</b>	<b>Fundamentação Teórica</b>	<b>7</b>
3.1	Teste de usabilidade	7
3.2	Gamificação	9
3.2.1	Pontuação	10
3.2.2	Ranking	10
3.2.3	Atualização de status em tempo real	10
3.2.4	Histórico de objetivos	10
3.2.5	Premiações	10
3.2.6	Medalhas	11
3.2.7	Recompensas	11
3.3	Ferramentas de desenvolvimento	11
3.3.1	React-native	11
3.3.2	Firebase	12
3.3.3	Realtime Database	12
3.3.4	Authentication	13
3.3.5	Cloud Messaging	13
3.3.6	Storage	13
3.3.7	AdMob	13
3.3.8	Analytics	13
<b>4</b>	<b>Desenvolvimento</b>	<b>14</b>
4.1	Metodologia	14
4.2	Ferramenta de gerenciamento de atividades proposto	15
4.3	Levantamento de Requisitos	16
4.4	Prototipagem	18
4.5	Aplicação da gamificação no aplicativo proposto	24
4.6	Primeiro Teste	26
4.6.1	Objetivo do Teste	26
4.6.2	Roteiro	26
<b>5</b>	<b>Resultados</b>	<b>28</b>
5.1	Caracterização da amostra e condução dos testes	28

5.2	Discussões e Correções . . . . .	29
5.3	Disponibilização e Avaliação na PlayStore . . . . .	30
<b>6</b>	<b>Conclusão . . . . .</b>	<b>34</b>
	<b>Referências . . . . .</b>	<b>35</b>
	<b>Anexos</b>	<b>37</b>
	<b>ANEXO A Formulário de Avaliação . . . . .</b>	<b>38</b>

# 1 Introdução

Com o custo de vida cada vez mais elevado nas pequenas e grandes cidades, uma alternativa é o compartilhamento de moradias uma vez que esse estilo de vida tem se mostrado cada vez mais econômico e prático. Dessa forma, para evitar aluguéis caros as pessoas têm optado por essa estratégia. Mas afinal, o que é a moradia compartilhada? Moradia compartilhada é um conceito que vem ganhando cada vez mais força no Brasil e no mundo, com destaque principalmente nas grandes cidades. Diversos fatores estão impulsionando a escolha para esse tipo de moradia, tais como, dividir valores de aluguéis, condomínio e despesas básicas como contas de luz, água e internet são vantagens óbvias de dividir moradia, mas não são as únicas.

O conceito de moradia compartilhada muitas vezes é confundido com o conceito de repúblicas estudantis, que são bastante tradicionais e conhecidas. Repúblicas estudantis são destinadas apenas aos universitários, tendo como objetivo oferecer moradias para estudantes por um preço justo para eles. Por outro lado, o termo moradia compartilhada diz respeito a divisão das despesas entre os moradores que alugam quartos individuais ou compartilhados em casas e/ou apartamentos, sendo assim, uma ideia que não se restringe apenas a universitários. Porém, em ambos os cenários há a necessidade de ter um controle das contas e das tarefas do dia a dia.

Segundo o site [ROOMGO \(2018\)](#) a moradia compartilhada era bastante popular nos anos de 1970, mas devido aos avanços da sociedade moderna as pessoas passaram a se isolar cada vez mais. Nesses últimos 50 anos, esse conceito de *co-living*<sup>1</sup> evoluiu bastante, tornando-se muito popular nos dias de hoje devido a alguns fatores mencionados anteriormente.

---

<sup>1</sup> Estilo de vida onde integrantes da mesma moradia compartilham despesas e tarefas do cotidiano.

## 1.1 Problema e Justificativa

Como já sabemos, ter as nossas obrigações organizadas no dia a dia é de suma importância para que possamos realizá-las no tempo certo. Isso não é diferente quando falamos de moradia compartilhada onde temos a necessidade de organizar e dividir várias atividades com um número maior de pessoas. Após a realização de algumas entrevistas com alguns desses membros, foi possível concluir que em muitas dessas moradias as tarefas acabavam não sendo realizadas no tempo que havia sido combinado e gerando alguns problemas dentro dessas moradias. Isso por vários fatores, tais como: esquecimento, utilização de vários aplicativos para organização onde não existe uma comunicação entre eles, isto é, cada aplicativo resolve um problema em específico que nesse contexto poderia ser resolvido com apenas uma aplicação. Por exemplo, ao invés de um aplicativo para controle de tarefas e outro para gerenciamento de contas da moradia, ambas as funcionalidades poderiam ser encontradas no aplicativo proposto. Tendo em vista essa necessidade, a principal contribuição deste trabalho é o desenvolvimento de um aplicativo *mobile*, gamificado, para engajar as pessoas com os afazeres domésticos em moradias compartilhadas.

## 1.2 Objetivos Geral e Específicos

Este trabalho tem como objetivo geral desenvolver um aplicativo *mobile* para auxiliar na organização de tarefas e contas para moradias compartilhadas.

E como objetivos específicos, os seguintes:

- Propor e desenvolver o protótipo do layout;
- Avaliar a usabilidade do layout proposto;
- Desenvolver uma ferramenta para o gerenciamento de atividades e responsabilidades domésticas.

## 1.3 Organização do Trabalho

Este trabalho está organizado em capítulos. O Capítulo 2 apresenta a revisão de literatura, Capítulo 3 apresenta a fundamentação teórica, o Capítulo 4 apresenta a metodologia do trabalho, os métodos utilizados para o desenvolvimento do trabalho, o Capítulo 5 apresenta os resultados e as discussões e por último o Capítulo 6 apresenta uma conclusão e trabalhos futuros.

### **1.3.1 Estrutura da Monografia**

**Capítulo 1:** Introdução.

**Capítulo 2:** Revisão de Literatura.

**Capítulo 3:** Fundamentação Teórica.

**Capítulo 4:** Metodologia.

**Capítulo 5:** Resultados e Discussões.

**Capítulo 6:** Conclusão (e trabalhos futuros).

## 2 Revisão de Literatura

Neste capítulo são apresentados os trabalhos relacionados, bem como, as principais características de cada um dos aplicativos relacionados. Também será apresentada uma tabela comparativa com as principais características de cada um desses aplicativos com o aplicativo proposto.

Com uma grande quantidade de aplicativos disponíveis na *Google Play* e *Apple Store* para o auxílio da organização de tarefas e contas em moradias compartilhadas é importante ressaltar que seria inviável conseguir analisar todos. Dado esse ponto de vista, os aplicativos foram selecionados seguindo alguns critérios, sendo eles: a quantidade de *downloads* do aplicativo sendo necessariamente superior a mil(1000) *downloads* e uma avaliação igual ou superior a quatro estrelas. Também vale ressaltar as palavras-chave utilizadas para procurar os aplicativos, sendo eles, Tarefas, Organização de moradias, Tarefas e Contas, Tarefas diárias. Alguns outros aplicativos atendiam os critérios definidos porém não foram selecionados pois fugiam do tema abordado ou eram muito simples. A seguir, será feito um pequeno resumo dos aplicativos utilizados como trabalhos relacionados:

*Todoist* (CONQUER, 2019) é um aplicativo de gerenciamento de tarefas e projetos. É um aplicativo gratuito, mas possui algumas funcionalidades que são pagas. Podendo ser utilizado no ambiente *web* e na versão *mobile*. Permite criar tarefas e subdividi-las, assim como os projetos. Com o *Todoist* é possível compartilhar as tarefas ou mesmo os projetos com outros usuários. Na criação de uma tarefa é possível definir sua prioridade, um período que será realizada, como por exemplo, configurar para que uma tarefa apareça para ser feita toda quarta-feira, no mesmo horário além da data de vencimento, notificando ao usuário por email quando as mesmas estiverem perto do prazo final. As tarefas podem ser delegadas para outros usuários. O aplicativo conta com a estratégia de gamificação, onde são atribuídos pontos aos usuários a fim de avaliar a produtividade dos mesmos. Com a versão *premium* o usuário conta com o acesso a gráficos de progressos e de aumento de produtividade, além de outros recursos.

O *Bills Organizer* (SMOBILEAPPS, 2013) é um aplicativo de gerenciamento de contas. Está disponível somente na versão *mobile*, conta com uma versão gratuita e uma *premium*. Com o *Bills Organizer* é possível criar contas e dividi-las em categorias, definir uma prioridade dessa conta, data de vencimento, o *status* da conta entre outras configurações. Uma das funcionalidades que se destaca nesse aplicativo é a possibilidade de gerar relatórios das despesas ou receitas por categorias. Porém, o aplicativo apresenta algumas desvantagens, como por exemplo apresentar uma interface não muito amigável ao usuário e também não é possível compartilhar as contas com outros integrantes, sendo apenas de uso pessoal. A versão *Premium* conta com as seguintes funcionalidades: sincronização, proteção por senha, sem anúncios, entre outros.

O (SPLITWISE, 2019) é um aplicativo de gerenciamento de despesas do dia a dia. Conta com uma versão gratuita e uma versão *Premium*. O *SplitWise* é um aplicativo com dois focos sendo eles, a divisão de despesas durante uma viagem e em moradias compartilhadas. Ele permite a criação de grupos onde é possível dividir igualmente as contas entre os membros. É possível criar despesas com uma certa periodicidade. A versão *Premium* conta com as seguintes funcionalidades, gráficos e imagens, conversão de moeda e não contém sem anúncios.

O *WunderList* (2019) é um aplicativo de gerenciamento de tarefas como os outros mencionados anteriormente. Está disponível em versão *web* e *mobile*, contando com uma versão gratuita e a versão *Premium*. Com o *WunderList* também é possível criar tarefas e dividi-las em subtarefas, definir prioridades sobre as mesmas, criar prazo de entrega assim como receber notificações com base nestes prazos e compartilhar uma lista de tarefas. Um dos pontos falhos desse aplicativo é que não é possível ter um noção da produtividade de um determinado usuário. A versão *premium* conta com subtarefas ilimitados, atribuições ilimitadas, arquivos ilimitadas.

Os seguintes aplicativos, Minhas Contas (BRASIL, 2013), Sm Plan (TACT, 2017), Simples Checklist (CHECKLIST, 2013), Any.do (CALENDAR, 2011), Tasks(LIMETED, 2016), Atividades Domésticas(23APPS, 2013) e Tarefas Domésticas (WHISPERARTS, 2018) são aplicativos que apresentam características semelhantes entre si, onde todos possuem uma versão gratuita, porém nenhum deles faz o uso da gamificação para o engajamento dos usuários para realização das suas tarefas, todos apresentam anúncios dentro do aplicativo com a exceção do Sm Plan, Any.do, Tasks, Atividades domésticas e Tarefas domésticas. Todos os aplicativos mencionados anteriormente com exceção do Any.do não permitem o compartilhamento de tarefas ou contas, sendo específicos para o uso individual.

Realçando assim as principais funcionalidades de cada um dos aplicativos estudados foi possível concluir que os aplicativos mencionados não resolvem o problema por completo, mas sim apresentam apenas soluções parciais. Dessa forma é apresentado na Tabela 2.1 um resumo dos trabalhos mencionados anteriormente, apontando as principais características de cada um deles, com o objetivo de compara-los com o aplicativo proposto. Vale realçar que a comparação foi realizada levando em consideração a capacidade de atender a demanda e não a qualidade das soluções obtidas.

Um dos principais diferenciais do **Tarefaça** é a possibilidade de encontrar várias funcionalidades referentes ao gerenciamento de contas e de tarefas no mesmo aplicativo com o auxílio da gamificação e fornecendo ao usuário final um produto completo.

Tabela 2.1 – Tabela comparativa das funções dos aplicativos analisados

<b>App</b>	<b>Custo</b>	<b>Possui Premium</b>	<b>Gamificado</b>	<b>Anúncio</b>	<b>Compartilhar Tarefas</b>	<b>Compartilhar Contas</b>
Todoist	Free	Sim	Sim	Não	Sim	Não
WunderList	Free	Sim	Não	Sim	Não	Não
SplitWise	Free	Sim	Não	Sim	Não	Sim
Bills Organizer	Free	Sim	Não	Sim	Não	Sim
Minhas Contas	Free	Sim	Não	Sim	Não	Sim
Sm Plan	Free	Não	Não	Não	Sim	Não
Simple Checklist	Free	Sim	Não	Sim	Sim	Não
Any.do	Free	Sim	Não	Não	Sim	Não
Tasks	Free	Não	Não	Não	Sim	Não
Atividades Domésticas	Free	Não	Não	Não	Sim	Não
Tarefas Domésticas	Free	Não	Não	Não	Sim	Não
<b>Tarefaça</b>	<b>Free</b>	<b>Não</b>	<b>Sim</b>	<b>Sim</b>	<b>Sim</b>	<b>Sim</b>

## 3 Fundamentação Teórica

Neste capítulo será apresentada de forma detalhada o teste de usabilidade que foi aplicado, o conceito da gamificação e os seus elementos e também as ferramentas utilizadas para o desenvolvimento do aplicativo.

### 3.1 Teste de usabilidade

De acordo com (NOGUEIRA, 2003) a maioria dos métodos de avaliação de interface baseiam-se em engenharia cognitiva, cujo objetivo é a criação de sistemas computacionais que exijam do usuário baixa carga cognitiva para serem utilizados, isto é, que sejam fáceis de apreender, de usar e que sejam agradáveis para as pessoas. Ao medirmos tais características e propriedades do sistema com a finalidade de verificar o nível de aceitação da interface pelo usuário, estaremos verificando a usabilidade desta interface.

Os testes de usabilidade são importantes pois nos permitem saber a interação do usuário com um determinado produto, podendo ser, um *layout*, um site ou mesmo um produto físico para que seja possível ter uma experiência satisfatória e positiva. Desta forma os testes nos permitem conhecer o comportamento dos usuários dentro do sistema e até mesmo realizar mudanças que trarão benefícios para os mesmos.

*“A usabilidade refere-se à capacidade de um produto de ser compreendido, aprendido, utilizado e ser atrativo para o utilizador, em condições específicas de utilização” (BOFF; KAUFMAN; THOMAS, 1986)*

A avaliação da experiência do usuário não é subjetiva ou simplesmente uma questão de gosto. Essa avaliação segue a norma ISO 9241 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011) que descreve a prática de forma objetiva e clara. Segundo essa norma, para medir o nível de uma interface, devem-se avaliar os seguintes elementos: eficácia, eficiência e satisfação.

- Eficácia: precisão com que os usuários atingem os objetivos específicos, gerando os resultados esperados
- Eficiência: precisão que os usuários atingem seus objetivos, em relação à quantidade de recursos gastos.
- Satisfação: conforto e aceitabilidade que o usuário tem ao usar o produto.

Dado esse contexto, o *layout* será avaliado utilizando o método SUS (System Usability Scale) proposto por (BROOKE, 1996), utilizado para avaliar produtos, serviços, aplicações ou mesmo qualquer tipo de interface. Esse método pode ser utilizado independente da tecnologia, podendo ser utilizado para teste de *hardware*, *software* de consumidor. Por conta disso, a metodologia se tornou uma referência para indústria, tendo assim mais de 1200 publicações (BROOKE, 2013).

O SUS é um método confiável pois a confiabilidade dá-se pela frequência com que os usuários respondem as perguntas. Sendo possível perceber diferenças em amostras menores em comparação com os questionários produzidos em casa e outros disponíveis comercialmente (LEWIS; SAURO, 2009). Dessa forma, como o tamanho e a confiabilidade da amostra não estão relacionados, o SUS pode ser aplicado em um grupo de até dois participantes e ainda gerar resultados bem confiáveis.

Com os critérios do SUS é possível avaliar a efetividade, a eficiência e a satisfação do usuário ao realizar o teste. O questionário consiste em 10 perguntas, e para cada uma delas o usuário pode responder em uma escala *likert* de 1 a 5, onde 1 significa **discordo completamente** e 5 significa **concordo completamente**, sendo elas:

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

Para chegar a um resultado final é preciso subtrair 1 ponto das respostas ímpares e das respostas pares, subtrair a resposta de 5, como apresentado na Tabela 3.1. Ou seja, se o usuário respondeu 2, será contabilizado 3. De seguida deve-se fazer a soma dos valores das 10 questões e multiplicar por 2.5. A pontuação obtida estará entre 0 e 100. Obtendo um pontuação acima de 68 que é a média do SUS em todos os 500 estudos, o produto tem uma boa usabilidade e qualquer

valor abaixo de 68 está abaixo da média.

Tabela 3.1 – Fórmula da normalização dos resultados

Questão	Fórmula
Pares	5 - Resposta
Ímpares	Resposta - 1

Inicialmente o SUS foi proposto apenas para medir a facilidade de uso de um determinado produto, só que algumas pesquisas recentes de (LEWIS; SAURO, 2009) mostram que ele fornece uma medida global de satisfação dos sistemas e capacidade de aprendizado. Dessa forma as questões 4 e 10 fornecem qual foi a dimensão de aprendizado do participante e as questões restantes fornecem uma dimensão de usabilidade.

Outro fato interessante desse método é que é possível fazer uma relação de cada uma das perguntas com às heurísticas de (NIELSEN; MOLICH, 1990) como é apresentado no Capítulo 4, resultando assim em um avaliação mais estruturada. Desta forma as perguntas do método SUS ficam associados com algumas das 10 heurísticas de Nielsen como é apresentado a seguir:

- Facilidade de aprendizagem: 3, 4, 7 e 10
- Eficiência: 5, 6 e 8
- Facilidade de memorização: 2
- Minimização dos erros: 6
- Satisfação: 1, 4, 9

## 3.2 Gamificação

Segundo Deterding, Dixon e Nacke (2011) gamificação é o uso de elementos de *design* de jogos em contextos não relacionados a jogos, ou seja, trata-se de pensamentos de jogos, majoritariamente virtuais, com o objetivo de engajar as pessoas por meio de desafios constantes e bonificações. Gamificação é um conceito que pode ser adaptado e aplicado em diferentes realidades, tais como na educação, nos ambientes empresariais, na saúde entre outras aplicações, podendo ser acessível a qualquer público.

Dessa forma, esse conceito é aplicado no ambiente de moradias compartilhadas com o objetivo de engajar e melhorar o desempenho dos moradores nas suas atividades do dia a dia, por diversos fatores que acabam dificultando esse engajamento, tais como a maioria das atividades do dia a dia são cansativos chegando a ser enfadonhos, ou mesmo por ter integrantes com hábitos diferentes, entre outros. Desta forma, o uso da gamificação é essencial para suprir esse tipo de

problema, onde será possível atribuir pontos nas atividades e acumulá-los quando essas forem realizadas nos prazos estipulados, entre outros. Assim, a gamificação conta com alguns elementos para que esse processo seja possível, tais como: a pontuação, o *ranking*, a atualização do *status* em tempo real, histórico de objetivos, premiações, medalhas e recompensas, explicadas a seguir:

### **3.2.1 Pontuação**

Os benefícios adquiridos são baseados a partir da pontuação adquirida pelo usuário após a realização de uma tarefa. Essa pontuação é inicialmente definida pelos integrantes da moradia e adquirida pelo usuário referente ao esforço pela realização das tarefas e não apenas pelo resultado final que consiste em manter a moradia limpa e organizada.

### **3.2.2 Ranking**

Um painel com um *ranking* funciona como um *feedback* instantâneo aos integrantes gerando reconhecimento perante os moradores que fazem bem seu trabalho.

### **3.2.3 Atualização de status em tempo real**

Cada tarefa concluída representa uma tarefa realizada pelo integrante da moradia e o resultado pode ser processado imediatamente conduzindo a uma sensação de progresso e de dever cumprido.

### **3.2.4 Histórico de objetivos**

As realizações obtidas no passado, seja de forma individual ou em participações de grupos é estimulante para os colaboradores continuarem a sua jornada.

### **3.2.5 Premiações**

Os pontos acumulados no ambiente da gamificação podem ser utilizados como estratégias de premiações entre os integrantes da moradia. Dessa forma, cabe a cada moradia definir como será a premiação que melhor convém aos seus integrantes, levando sempre em consideração seus desejos.

### 3.2.6 Medalhas

Fazendo uma analogia com os atletas, este elemento pode motivar os integrantes na realização das suas tarefas da mesma forma que as medalhas incentivam os competidores a darem o seu melhor para conquistar itens durante suas carreiras, sentindo-se reconhecidos por terem alcançado seus objetivos.

### 3.2.7 Recompensas

Recompensas são prêmios ainda maiores podendo ser considerada uma das mais importantes, pois a partir de uma pontuação específica, o integrante conquista novas regalias. Com a realização das tarefas ou pagamento das contas no prazo estipulado o morador vai acumulando pontos de acordo com a tarefa realizada, podendo realizar a troca desses pontos por uma tarefa a ser realizada. Com a sensação de ser recompensado pelas suas ações, o usuário acaba ficando engajado na realização das suas tarefas. Essa estratégia permite que essas moradias alcancem um bom nível de motivação dos colaboradores, estimulando uma competição saudável entre os membros.

## 3.3 Ferramentas de desenvolvimento

### 3.3.1 React-native

*React native* é uma biblioteca para criar aplicativos móveis usando tecnologias da web familiares sem sacrificar o desempenho ou a aparência tipicamente associada a aplicativos totalmente nativos (MASIELLO; FRIEDMANN, 2017).

Desenvolvido pelo *Facebook* em 2015, cuja estrutura é baseado em componentes, onde esses componentes são pequenos ou grandes trechos de códigos podendo ser botões, modal, *card*, entre outros. Com isso permite que o desenvolvedor faça combinações para construir telas completas formadas por esses vários componentes. Um dos principais motivos por ter escolhido essa tecnologia é sua capacidade de desenvolver aplicativos nativos para diferentes plataformas utilizando apenas um único código-fonte, eliminando o processo extremamente complexo de aprendizagem das linguagens *Objective-C (iOS)* e *Java (Android)* para o desenvolvimento, onde praticamente nada do código desenvolvido numa plataforma é aproveitado em outra, tendo a necessidade de desenvolver para outra linguagem ou desenvolvendo aplicativos híbridos.

O *react native* resolve ainda um outro grande problema que era bastante comum dos outros inúmeros *frameworks*, como o *Cordova*, onde os códigos eram gerados em um navegador e encapsulados dentro de um aplicativo. Apresentando a principal desvantagem dessa abordagem onde os aplicativos desenvolvidos utilizando os *frameworks* como por exemplo o *Cordova* são

mais lentos que um código nativo e tendo um desempenho inferior aos aplicativos nativos. Com o objetivo de resolver esse problema o *react native* possui uma abordagem totalmente diferente. O código é desenvolvido utilizando *JavaScript* através do *React* e uma linguagem *XML(Extensible Markup Language)* equivalente ao *HTML(Hypertext Markup Language)* e quando o código é gerado ele é convertido para código nativo de cada uma das plataformas, ficando idênticos aos códigos gerados pelo *Swift* para o desenvolvimento nativo para *iOS* e *Java* para desenvolvimento para *Android*.

Outra grande vantagem do *react native*, é a forma que trabalha com o DOM (*Document Object Model*), atualizando os componentes de acordo com seus estados. Nas aplicações tradicionais, o desenvolvedor é o responsável por saber se houve modificações nos dados e alterar com o DOM. Entrando assim a vantagem de usar o *react native* onde o desenvolvedor cria o próprio DOM, e cada componente que for renderizado o *React* atualiza o DOM com as mudanças realizadas, de uma forma mais eficiente e mais rápida.

### 3.3.2 Firebase

*Firebase* é uma plataforma de desenvolvimento *mobile* e *web* adquirida pela *Google* em 2014. Com foco em ser um *back-end* completo e de fácil usabilidade, essa ferramenta disponibiliza diversos serviços diferentes que auxiliam no desenvolvimento e gerenciamento de aplicativos Gasperin (2016).

Esse tipo de serviço é chamado de *BaaS (Backend As A Service)*, que nada mais é do que um tipo de serviço que oferece uma infraestrutura e o *backend* de uma aplicação podendo ser ela *mobile* ou *web*, de forma simplificada. Com esse tipo de serviço podemos realçar as principais vantagens, sendo elas, a segurança na comunicação e transferência de dados, a disposição de funcionalidades mais básicas e comuns presentes em várias aplicações, como por exemplo o serviço de autenticação, serviço de notificação, armazenando de mídias, entre outros, de forma totalmente gratuita no plano básico. Segue abaixo uma lista e uma breve explicação dos principais serviços oferecidos pelo *Firebase* utilizados na aplicação:

### 3.3.3 Realtime Database

O Banco de Dados em Tempo Real armazena e sincroniza os dados dos usuários com todos os dispositivos em tempo real em questão de segundos. Dessa forma, permitindo que os dados permanecem disponíveis mesmo quando o aplicativo fique sem conexão com a internet, proporcionando assim uma ótima experiência para o usuário.

### 3.3.4 Authentication

O *Firebase* já disponibiliza um serviço de autenticação que permite que o usuário faça a autenticação através de contas do *Google*, *Facebook*, *Twitter*, *Github*, login anônimo, e-mail e senha que é básico, entre vários outros, deixando isso de uma forma simples e segura.

### 3.3.5 Cloud Messaging

O serviço *cloud messaging* é um outro serviço oferecido que permite o envio de mensagens e notificações para os usuários de várias plataformas, podendo ser *Android*, *iOs* ou mesmo *web*, de forma totalmente gratuita. Essas mensagens pode, ser enviadas de forma totalmente personalizada, como por exemplo: para um grupo específico de usuários, um grupo específico de dispositivos entre entre outros.

### 3.3.6 Storage

O *Firebase Storage* oferece de uma forma simples e prática de guardar e compartilhar arquivos, podendo ser imagens, vídeos ou mesmo alguns outros conteúdos gerados pelos usuários. Contando com o próprio sistema de regras de segurança para proteger seus dados.

### 3.3.7 AdMob

A *Admob* facilita a monetização do aplicativo, de uma forma bem simples, onde permite que o desenvolvedor encaixe anúncios, sem afetar a experiência do usuário dentro do aplicativo. O próprio serviço já priorizar de forma automática quais os anúncios que retornam maior lucro.

### 3.3.8 Analytics

O *Analytics* é um outro dos serviços oferecido pelo *Firebase* que produz *insights* em tempo real sobre o usuário e o aplicativo, possibilitando assim na melhor tomada de decisões por parte do desenvolvedor. Entendendo como os usuários estão utilizando o seu aplicativo.

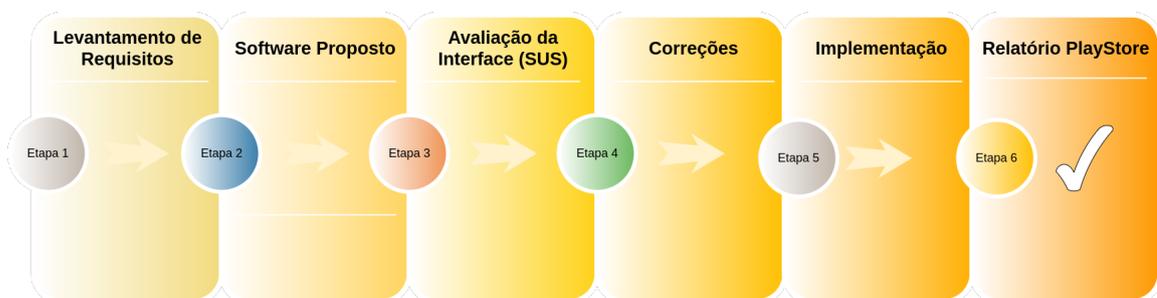
## 4 Desenvolvimento

Neste capítulo será apresentada a metodologia abordada no trabalho, os requisitos funcionais, não funcionais, e a prototipagem do *layout* do *software* proposto utilizando as heurísticas propostas por (NIELSEN; MOLICH, 1990). Ademais, os *layouts* que cerceiam o funcionamento do aplicativo e representam o mesmo de forma padronizada, conjuntamente com a representação gráfica das informações deste. Assim, o Tarefaça, será desenvolvido em *react native*, utilizando o *firebase* como banco de dados.

### 4.1 Metodologia

Neste seção são apresentados os métodos que foram utilizados para realizar a pesquisa, as técnicas para fazer o levantamento dos requisitos, as heurísticas utilizadas para realizar a prototipagem do *layout*, a heurística para realizar a avaliação do aplicativo proposto e quais os métodos para coletar os dados do usuários. Nessa pesquisa foi utilizada a abordagem quantitativa. Na Figura 4.1 é apresentado um resumo da metodologia utilizada para a realização do trabalho.

Figura 4.1 – Metodologia do Trabalho



Para levantar o requisitos do aplicativo foi utilizada a técnica da entrevista como sugerido por (BEZERRA, 2007). Em seguida é apresentado o software proposto fazendo uma correlação com algumas das dez heurísticas propostas (NIELSEN; MOLICH, 1990). Na terceira etapa do trabalho como apresentado na Figura 4.1 é utilizado como instrumento de avaliação da interface o método SUS proposto por (BROOKE, 1996) que consiste em 10 perguntas. Foram escolhidos esses métodos pois possibilitam uma avaliação e uma prototipagem rápida. Na quarta etapa seguinte foram feitas algumas correções do *layout* após a avaliação da interface. Na quinta etapa foi desenvolvido o aplicativo em si e por último analisada o relatório gerado pela *Playstore*.

O questionário foi aplicado a 11 participantes sendo alguns alunos de outros cursos com o intuito de diversificar a pesquisa, realizada no ano de 2019. Esse questionário foi respondido nas dependências da universidade. Após a coleta das respostas obtidas no questionário, esses

foram normalizados utilizando o método SUS proposto por (BROOKE, 1996) para analisar as respostas coletadas como apresentado no Capítulo 4.

## 4.2 Ferramenta de gerenciamento de atividades proposto

Inicialmente para que haja uma boa organização dos nossos afazeres no dia a dia é muito importante que tenhamos uma estratégia. Dessa forma depois de passar por algumas moradias compartilhadas, algumas estratégias foram adotadas e foram sendo melhoradas ao longo desses anos. A estratégia dessa ferramenta de organização e distribuição das atividades proposto consiste em distribuir todas as atividades de forma que cada integrante fique responsável no mínimo por uma atividade. Como por exemplo, supondo que a moradia tenha cinco cômodos e cinco moradores. Dessa forma cada morador ficará responsável por um cômodo, como se pode observar na Tabela 4.1.

Tabela 4.1 – Apresentação visual do método

	Tarefa_1	Tarefa_2	Tarefa_3	Tarefa_4	Tarefa_5
Morador_1	x				
Morador_2		x			
Morador_3			x		
Morador_4				x	
Morador_5					x

Cada tarefa tem uma periodicidade, dependendo da moradia, podendo ser diário, semanalmente, entre outros. Dessa forma cada morador deve realizar sua tarefa antes do prazo estipulado. Vamos considerar que o morador\_1 tinha até o próximo final de semana para completar sua tarefa e após concluir sua tarefa a mesma deve ser passada para o próximo, no caso para o morador\_2. Outra situação que pode ocorrer é que um morador pode realizar sua tarefa antes da sua vez, o morador\_3 realizou a tarefa\_5 antes da sua vez, dessa forma ele ganha um crédito de forma que quando chegar a vez dele, ele passa a tarefa para o próximo consumindo esse crédito disponível, como pode se observar na Tabela 4.2.

Tabela 4.2 – Apresentação visual do método - Atualização

	Tarefa_1	Tarefa_2	Tarefa_3	Tarefa_4	Tarefa_5
Morador_1					
Morador_2	x				
Morador_3		x	x		o
Morador_4				x	
Morador_5					x

### 4.3 Levantamento de Requisitos

Requisitos de um sistema são as descrições do que o sistema deve fazer, quais os serviços que oferece e as suas restrições e funcionamento, refletindo assim as necessidades dos clientes para um sistema que serve para uma determinada finalidade (SOMMERVILLE, 2011). O levantamento de requisitos é uma etapa muito importante na construção de um *software*, aplicativo ou mesmo um site, onde devem ser analisadas todas as informações com o objetivo de identificar quais as funções que o *software* final irá disponibilizar.

Segundo (BEZERRA, 2007) o levantamento de requisitos pode ser feito utilizando várias técnicas, tais como, a entrevista, os questionários, os cenários, *brainstorming*, etnografia e caso de uso. Dessa forma foi escolhido a entrevista por ser uma das técnicas mais simples e tradicionais, essa técnica consiste em serem aplicadas questões formuladas pela equipe de desenvolvimento sobre os *stakeholders*<sup>1</sup> que usará o sistema futuramente a ser desenvolvido. Com o objetivo de obter bons resultados, a entrevista deve ser planejada para que não fique longa e exaustiva para o entrevistado. Dessa forma, na Tabela 4.3 são apresentados os requisitos funcionais e na Tabela 4.4 os requisitos não funcionais do sistema.

---

<sup>1</sup> Stakeholder significa público estratégico e descreve uma pessoa ou grupo que tem interesse em uma empresa, negócio ou indústria.

Tabela 4.3 – Requisitos Funcionais

<b>Código</b>	<b>Requisito Funcional</b>
<b>USUÁRIOS/CADASTROS</b>	
RF 01	O sistema permitirá que usuários sejam cadastrados classificados como: membros.
RF 02	O sistema permitirá que os usuários façam alterações em seus cadastros.
RF 03	O cadastro do usuário deverá conter: nome, senha, foto de perfil e e-mail.
<b>USUÁRIOS/LOGIN</b>	
RF 04	O sistema permitirá o acesso dos usuários através do login e senha.
<b>USUÁRIOS/CONTA</b>	
RF 05	O sistema permitirá que os usuários façam o cadastro das contas contendo: nome, uma descrição da conta, data limite que deve ser realizada, para qual usuário deve ser assinalada.
RF 06	O sistema permitirá que os usuários tenham acesso a um feed onde será apresentado um resumo dos acontecimentos
RF 07	O sistema permitirá que os usuários façam o cadastro das tarefas contendo: nome, uma descrição da tarefa, data limite que deve ser realizada, o período e para qual usuário deve ser assinalada.
RF 08	O sistema permitirá que os usuários marquem as tarefas como feitas ou notificar um membro para realizar determinada tarefa.
RF 09	O sistema permitirá que os usuários marquem as contas como pagas ou notificar um membro para realizar o pagamento.
RN 10	O sistema permitirá que os usuários adicionem itens a lista de compra.
RN 12	O sistema permitirá que os usuários adicionem avisos para relatar algum problema ou mesmo melhorias dentro do aplicativo.

Tabela 4.4 – Requisitos Não Funcionais

Código	Requisito Não Funcional	Justificativa
RNF 01	O sistema deverá executar em qualquer plataforma mobile.	Para que o sistema possa ser executado em qualquer plataforma mobile, o software será desenvolvido em react native, podendo ser instalado em Android, iOS, entre outros.
RNF 02	O sistema deverá se comunicar com o banco firebase.	O sistema será desenvolvido utilizando o firebase como banco, pois permite uma fácil portabilidade em caso de desenvolver uma versão web, e também por atender todos os requisitos necessários para o seu funcionamento no seu pacote free.
RNF 03	A interface deve ser agradável e de fácil utilização.	
RNF 04	O aplicativo não deverá armazenar dados localmente, sendo estes redirecionados a um servidor externo.	

## 4.4 Prototipagem

Com base nas heurísticas de (NIELSEN; MOLICH, 1990), procura-se identificar no aplicativo proposto problemas de usabilidade que possam comprometer ou atrapalhar a realização das tarefas por meio de uma correlação entre as telas apresentadas e as heurísticas.

Tendo uma proposta inicial das telas do *layout* do aplicativo proposto foram utilizadas algumas das 10 heurísticas, desenvolvido por (NIELSEN; MOLICH, 1990). Essas heurísticas realçam pontos importantes para o desenvolvimento de interfaces e devem ser considerados no momento das criação dos *layouts* ou mesmo após seu desenvolvimento.

*“Até os melhores designers produzem produtos de sucesso apenas se seus projetos resolverem os problemas certos. Uma interface maravilhosa para os recursos errados falhará- Jakob Nielsen”*

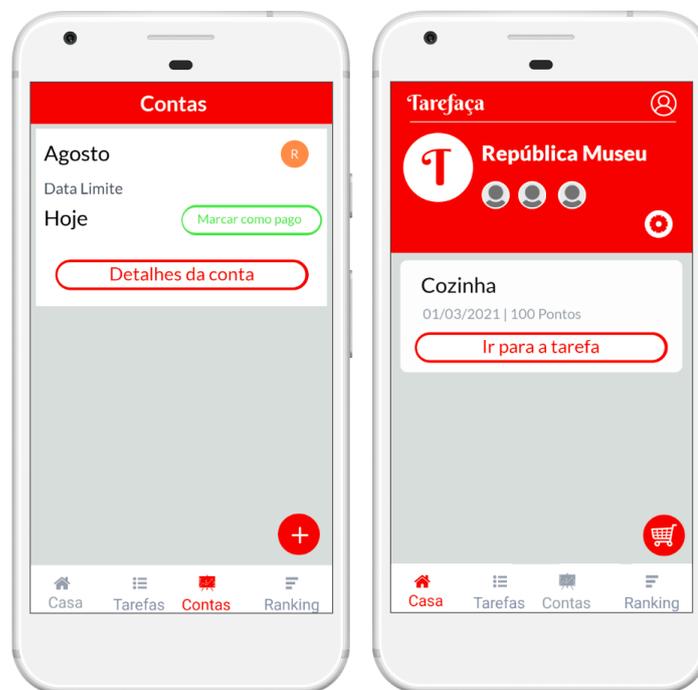
### 1. Visibilidade do estado do sistema

Segundo (NIELSEN; MOLICH, 1990) a visibilidade do estado do sistema trata-se da comunicação e transparência entre o usuário e o sistema, fornecendo assim uma maior quantidade de informações traduzindo-se assim em melhores tomadas de decisão. Assim Jakob Nielsen define a visibilidade do estado do sistema como o quão bem o estado

do sistema é transmitido aos seus usuários, informando aos usuários sobre o que está acontecendo, por meio de *feedback* apropriado dentro de um prazo razoável.

Dessa forma, essa primeira heurística foi aplicada nas telas conforme é apresentada na Figura 4.2. Como podemos observar durante o tempo todo o sistema informa ao usuário qual a aba de navegação em que ele se encontra no momento, quais as abas disponíveis dentro dos ambientes que ele poderá navegar a partir da sua localização atual. Outra tela aonde essa heurística foi aplicada foi na tela de pagamentos de contas, onde o botão muda de cor. Esses indicadores comunicam que o sistema está funcionando e reduzem a incerteza impedindo que os usuários cliquem no mesmo botão várias vezes pela incerteza de que já tenha funcionado.

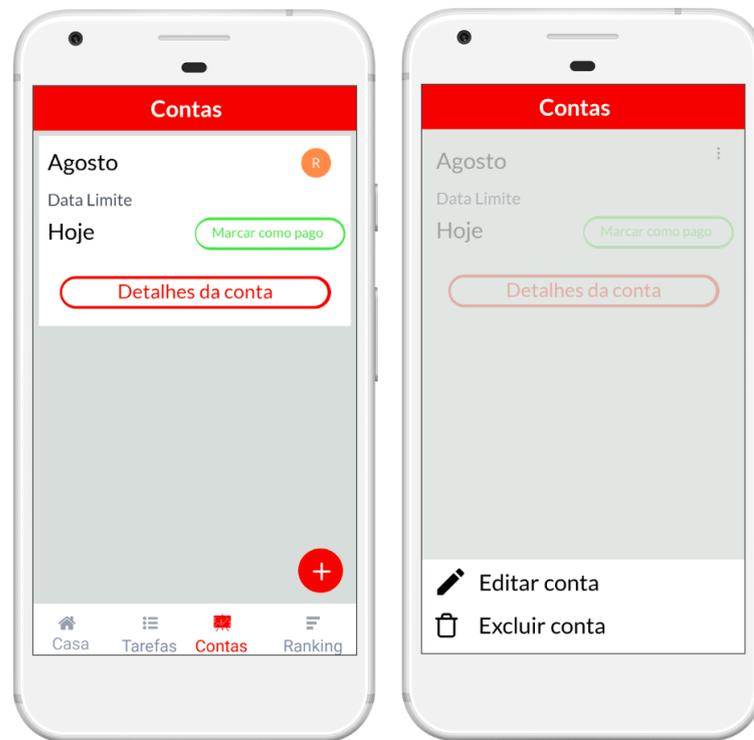
Figura 4.2 – Menu de Navegação



## 2. Equivalência entre o sistema e o mundo real

Nessa segunda heurística o princípio é que o sistema deve falar a língua dos usuários com termos, conceitos familiares, símbolos havendo uma compatibilidade entre os símbolos utilizados na interface do sistema com o mundo real. Essa segunda heurística foi aplicada nas telas de menu de tarefas e menu de contas, onde na tela de menu de tarefas o usuário tem a opção de excluir uma tarefa ou mesmo uma conta na tela de contas utilizando o ícone de uma lixeira enquanto que na tela de menu de contas o usuário tem a opção de adicionar uma tarefa ou uma conta pelo ícone de “+”. Essa heurística consiste em incluir elementos na interface para ajudar os usuários a passarem pela experiência com facilidade, conforme é apresentada na Figura 4.3.

Figura 4.3 – Telas de Navegação



### 3. Liberdade e controle do usuário

Nesta heurística, o principal objetivo é passar para o usuário a liberdade de ele fazer o que quiser dentro do sistema desde que as regras do negócio sejam respeitadas ou não interfira em outra funcionalidade e fornecer uma “saída de emergência” quando alguma ação é realizada por engano, algo que é frequente. O sistema deve permitir que o usuário tenha a liberdade de avançar, voltar, entre outros. Essa heurística é aplicada em algumas telas como apresentado na Figura 4.4, onde é possível observar que o usuário tem a opção de voltar na tela anterior. Respeitando a segunda parte dessa heurística é fornecido para o usuário uma saída de emergência quando alguma ação dentro do aplicativo foi realizado por engano, pois essa possibilidade de reverter essa ação remove a insegurança do usuário de utilizar o aplicativo, como é apresentado na Figura 4.4. Após o usuário tentar voltar na página anterior sem terminar de completar os dados para adicionar uma conta é apresentado um mensagem de alerta.

Figura 4.4 – Adicionar Tarefas - Adicionar Contas



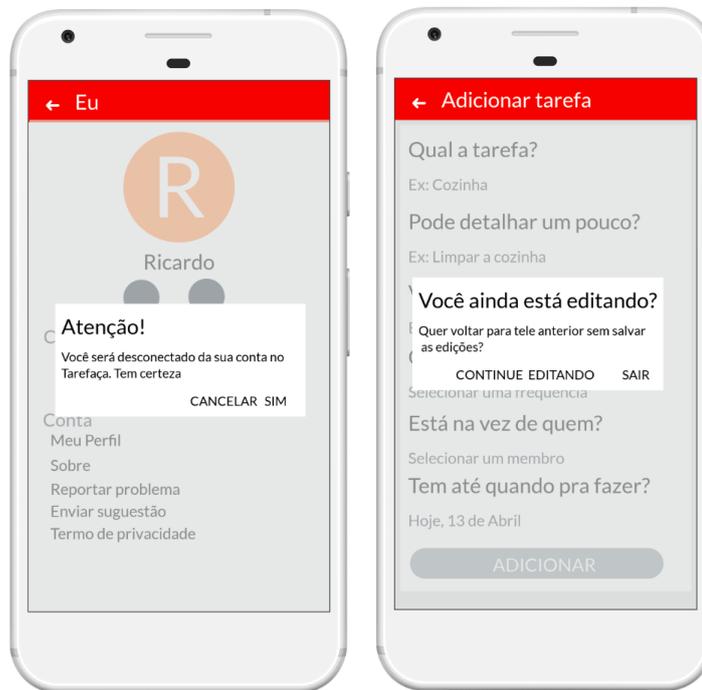
#### 4. Consistência e Padronização

Nessa heurística é importante manter a consistência e padrão visual, como preservar os padrões já conhecidos pelo usuário, por como exemplo botões à direita para confirmar uma ação. Essa heurística foi aplicada na tela de adicionar uma tarefa como pode ser observado na Figura 4.4.

#### 5. Prevenção de erros

A outra heurística tem como principal objetivo evitar que os usuários cometam erros dentro do *software*. Esses erros podem ser classificados como erros por deslize e erros por engano. O erro por deslize é quando o usuário pretende realizar uma ação mas acaba realizando outra, por não estar focado em sua ação dentro do aplicativo, enquanto que erro por engano é quando uma informação é equivocada ou tenha dupla interpretação. Dessa forma, essa heurística foi aplicada nas telas, onde o usuário está para sair da tela de adicionar uma tarefa sem salvar, será apresentado uma mensagem do erro que ele está prestes a cometer e prevenir que ele cometa esse erro ou quando o mesmo quer desconectar do aplicativo como é apresentado na Figura 4.5. Ações drásticas como deletar e sair devem ser confirmadas antes de serem executadas, evitando, assim, erros.

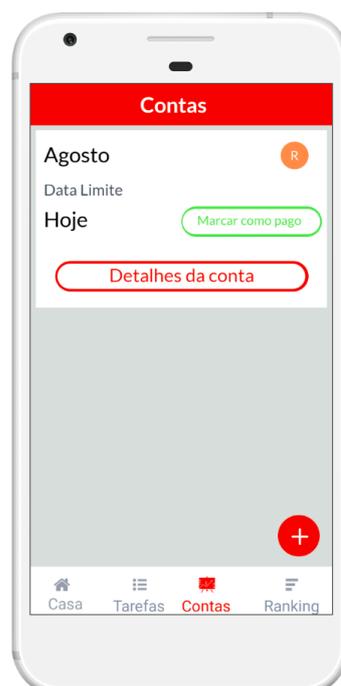
Figura 4.5 – Tela de aviso



## 6. Reconhecimento em vez de memorização

Nessa heurística é importante facilitar o reconhecimento em vez de memorização, e desta forma essa heurística foi aplicada nas telas de adicionar tarefa e contas. Onde o usuário associa o símbolo “+” para adicionar alguma coisa no sistema. Essa heurística é apresentado na Figura 4.6

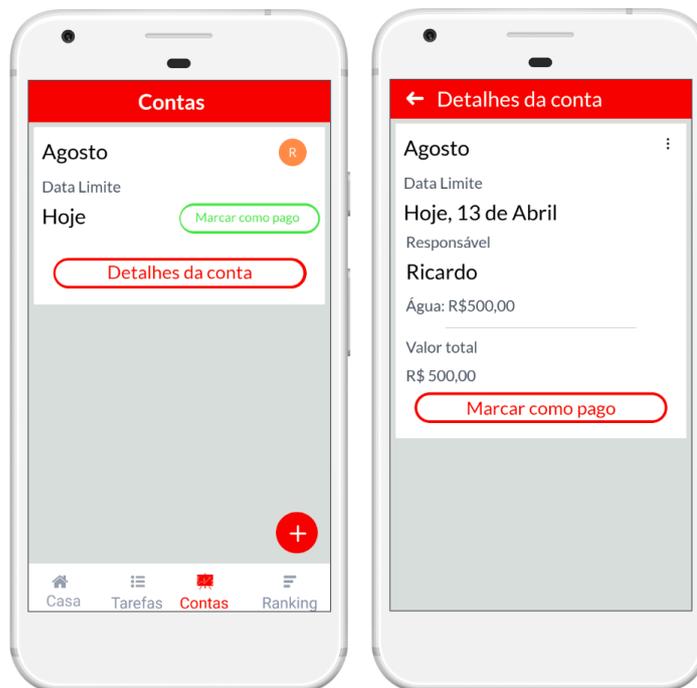
Figura 4.6 – Tela de Contas



## 7. Estética e Minimalismo

Esta heurística sugere que a interface deve ser *clean*, ou seja, evitar colocar muitas informações juntas. Apresentando para o usuário apenas as informações que sejam relevantes naquele momento e permitir que ele visualize a informação completa apenas quando quiser. A aplicação dessa heurística pode ser observada na tela de visualização de informações das tarefas, onde no primeiro momento o usuário tem acesso apenas a informações básicas e no segundo momento tem informações detalhadas como é apresentado na Figura 4.7.

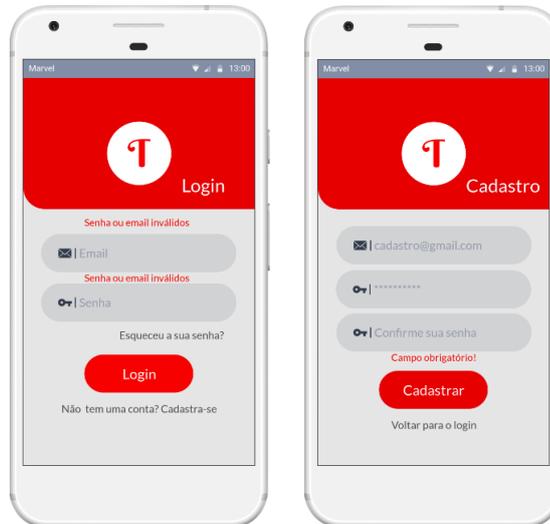
Figura 4.7 – Tela de Navegação



## 8. Ajude os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros

Esta heurística sugere que as mensagens de erro devem ser expressas em linguagem direta e simples, indicando o problema de forma clara e sugerindo uma solução, ou seja, ajudar o usuário a recuperar um erro depois de ele ser cometido. Como por exemplo o usuário erra a senha do login ou o e-mail e é passada uma mensagem para o usuário como é apresentado na Figura 4.8, outro exemplo, no momento que o usuário está fazendo o cadastro e acidentalmente esquece de completar um dos campos obrigatórios, e é passado uma mensagem de erro, como é apresentado na Figura 4.8.

Figura 4.8 – Tela de Autenticação



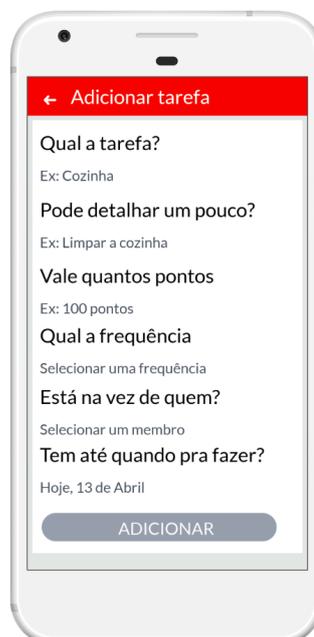
## 4.5 Aplicação da gamificação no aplicativo proposto

Esta seção apresenta algumas das telas onde é aplicada alguns dos elementos da gamificação apresentados no Capítulo 3. Desta forma temos:

- **Atribuição de pontos em uma determinada atividade:**

Ao definir uma atividade para ser realizada na moradia, o usuário tem a possibilidade de adicionar os pontos equivalentes a essa atividade. Em caso de ser uma atividade um pouco mais complexa acaba ganhando um pontuação mais alta, como apresentado na Figura 4.9.

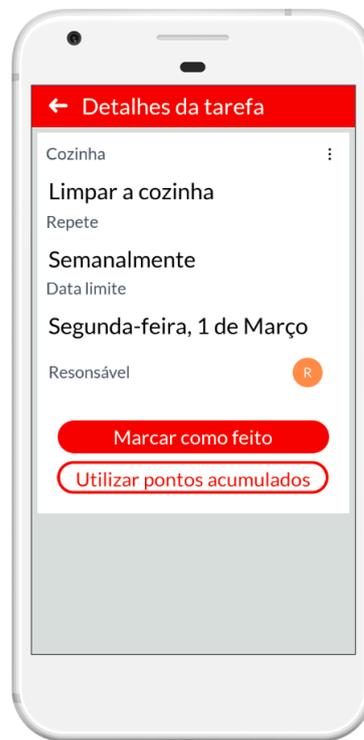
Figura 4.9 – Tela de adicionar conta



- **Troca dos pontos acumulados:**

Com a realização das tarefas no tempo definido, o usuário vai acumulando pontos como forma de o engajar, tendo a opção de trocar seus pontos por uma atividade que ele não quer realizar na semana, como apresenta a Figura 4.11.

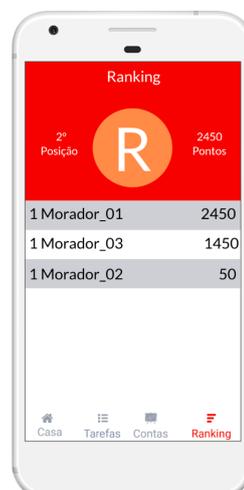
Figura 4.10 – Tela de trocar de pontos acumulados



- **Ranking**

Ao realizar as tarefas dentro do prazo estipulado o usuário ganha pontos, dessa forma, ele tem a possibilidade de observar o ranking geral da casa.

Figura 4.11 – Tela de ranking



## 4.6 Primeiro Teste

Esta seção apresenta o objetivo do teste, o roteiro na qual os usuários foram submetidos, a análise dos resultados e por fim uma discussão dos resultados obtidos levando em consideração algumas observações feitas pelos participantes durante a realização das tarefas.

### 4.6.1 Objetivo do Teste

O teste tem como objetivo avaliar a usabilidade da versão 1.0 do *layout* do *software* proposto, com a finalidade de:

- Detectar problemas no fluxo entre as telas;
- Integração das funcionalidades;
- Grau de aprendizado do usuário;
- Avaliar a satisfação dos usuários sobre a proposta;

### 4.6.2 Roteiro

Os testes foram realizados nas dependências da universidade. Os participantes foram convidados previamente pessoalmente, onde foram explicadas as atividades. Foram elaborados 5 tarefas, onde algumas delas são divididas em etapas, conforme é apresentada na Tabela 4.5.

Tabela 4.5 – Roteiro das atividades

Atividades	Cenários de iteração
1	Deparando com a falta de alguns produtos, como por exemplo, detergente, água sanitária que são produtos compartilhados entre os membros da moradia, adiciona esses itens na lista de compra.
2	Você é o responsável por limpar um dos cômodos da casa, por exemplo a cozinha, após realizar a tarefa você deve marcar como feito no aplicativo, em seguida adiciona uma nova tarefa definindo o nome, a descrição, entre os outros atributos.
3	Você é responsável pelas contas no mês, você deve criar as contas referentes as conta de luz, água, internet do mês de Agosto. Dentro do referente mês você deve adicionar o valor de cada umas das contas.
4	Nesta tarefa você tem como objetivo adicionar uma nova dívida entre você e um outro membro da mesma moradia. Em seguida você deve verificar se tem algum membro da moradia com dívida pendente com você.
5	Nessa ultima tarefa você tem que verificar a sua posição no ranking pessoal e geral nas tarefas realizadas durante a semana e a quantidade de pontos acumulados para trocar esses pontos acumulados por alguma tarefa da semana. E por último ver os troféus conquistados.

## 5 Resultados

Neste capítulo são apresentados os resultados obtidos durante a realização dos testes, é também uma análise acima do relatório fornecido pela *PlayStore* após a publicação do aplicativo.

### 5.1 Caracterização da amostra e condução dos testes

Os teste foram realizados com um total de 11 participantes. Dos 11 participantes, 36% moram em moradias compartilhadas e 11% dos participantes já possuem uma formação no ensino superior. Como já foi comentado no Capítulo 3 com o método SUS é possível medir o grau de aprendizado dos participantes, obtendo assim uma média de 3,22 do total das respostas e uma média de 3,3 nas questão relacionadas a usabilidade. Após a normalização dos dados obtidos foi possível obter uma pontuação média de 88,27 na escala do método SUS como pode se observar na Tabela 5.1. Com base nesse método uma pontuação acima dos 68 é um *layout* com um ótima usabilidade. Dessa forma é possível afirmar que a primeira versão do *layout* do *software* proposto teve uma boa aceitação por parte dos participantes. Vale a pena também analisar os resultados obtidos fazendo uma relação entre o método SUS com as 10 heurísticas de Nielsen. Desta forma foi possível obter uma média de 3,20 referente ao critério de facilidade de aprendizagem como é apresentado na Figura 5.1, ficando bem próximo do resultado obtido pelo método SUS, obtendo uma média de 3,48 em relação a eficiência apresentado na Figura 5.1, uma média de 3,27 em relação a facilidade de memorização e por último uma satisfação média de 3,42 também apresentado na Figura 5.1.

Figura 5.1 – Resultados

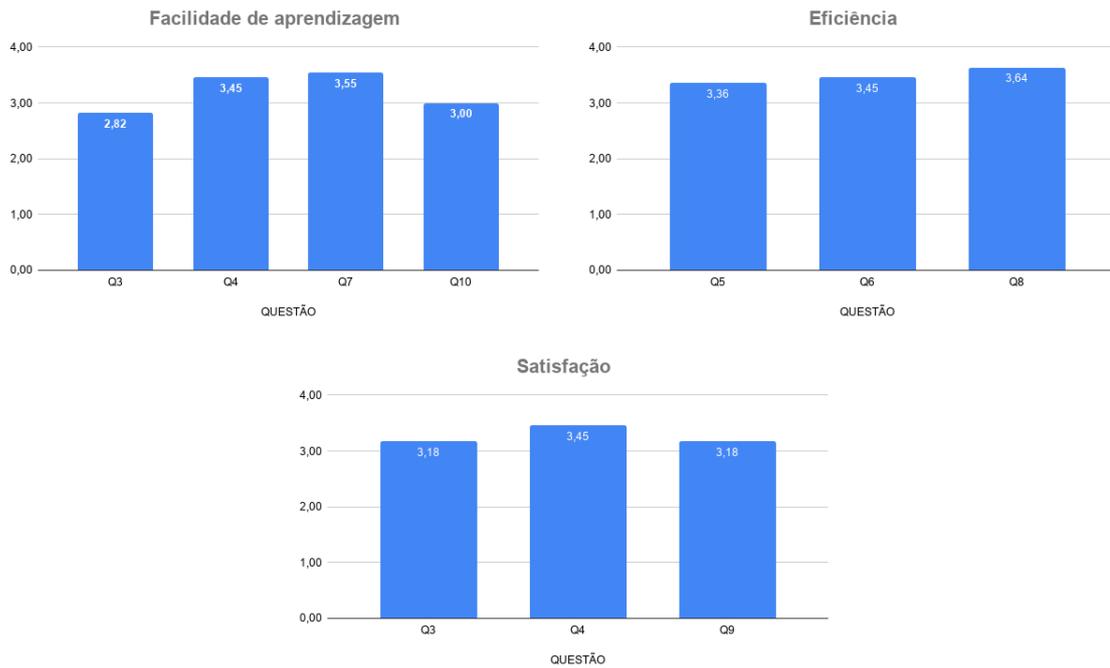


Tabela 5.1 – Resultado obtido através do questionário SUS

Questões	User 01	User 02	User 03	User 04	User 05	User 06	User 07	User 08	User 10	User 11
1	3	4	3	4	3	4	3	3	2	2
2	2	3	3	4	4	4	4	3	3	3
3	2	3	3	2	3	4	2	4	3	2
4	4	4	2	4	4	4	3	4	2	3
5	3	4	4	4	4	4	3	4	1	2
6	3	4	3	3	3	4	4	4	3	3
7	4	4	3	4	3	4	4	3	2	4
8	3	4	4	3	4	4	3	4	4	3
9	4	4	2	4	3	3	3	4	2	2
10	1	3	2	4	4	4	3	4	3	2
<b>Pontuação média: 82,27</b>										

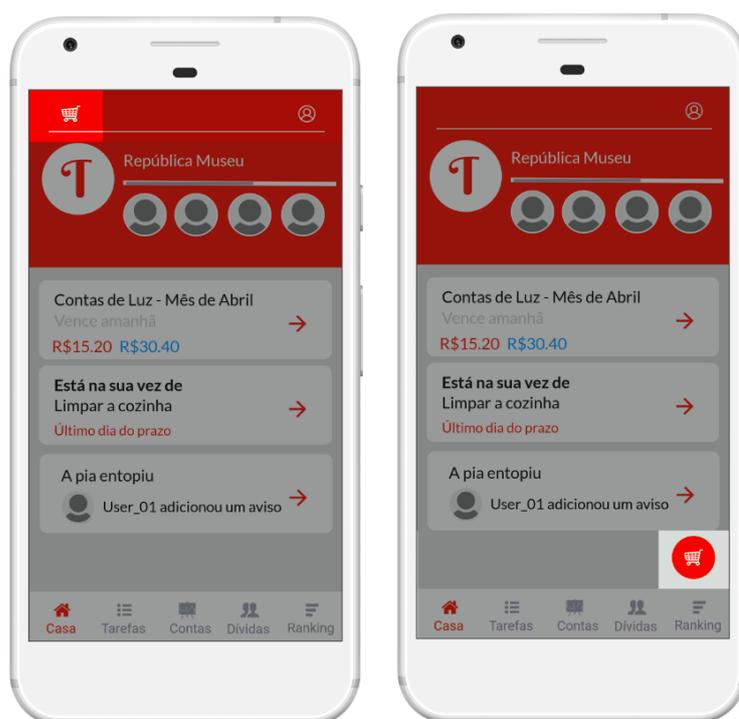
## 5.2 Discussões e Correções

Como foi possível observar na seção anterior o software proposto obteve uma boa pontuação de 82,27 dentro da escala do método SUS. Essa pontuação deve-se ao uso das 10 heurísticas apresentadas por Jacob Nielsen utilizadas no desenvolvimento do trabalho como apresentado no Capítulo 4.

Assim como nada é perfeito, serão apresentadas algumas observações relatadas por parte dos participantes no final das tarefas inicialmente definidas com o objetivo de melhorar a interface na versão 1.1 do *layout* do *software*.

Durante a realização dos testes foi possível observar que 90% dos participantes demonstravam dificuldades em completar a *Task 01* que consistia em adicionar alguns produtos na lista de compra como apresentado na Figura 4.5. Deparando com essa dificuldade apresentada pelos participantes foi apresentado novo layout utilizando as próprias sugestões de alguns participantes a fim de suprir essa dificuldade. Pode-se observar que o ícone de adicionar produtos na lista de compra inicialmente situado no canto superior esquerdo apresentava uma certa dificuldade para ser encontrado, assim na versão 1.1 do *layout* ficando situado no lado inferior direito como destaque, como é apresentado na Figura 5.6.

Figura 5.2 – Comparação entre a versão 1 e 1.1 do layout



### 5.3 Disponibilização e Avaliação na PlayStore

A primeira versão do aplicativo foi publicado na *Play Store* no dia 29 de setembro de 2020. Quando uma versão é disponibilizada na plataforma são executados alguns testes, com a finalidade de verificar a integridade do aplicativo, levando alguns critérios em consideração, como por exemplo, a estabilidade, o desempenho, a acessibilidade e por último a segurança e fidedignidade. Durante esses testes, foi possível concluir que o aplicativo é compatível com mais de 12.745 dispositivos como podemos observar na Tabela 5.4, o que é um resultado muito satisfatório levando em consideração que no dia a dia podemos encontrar várias derivações do sistema *Android*. Em relação ao desempenho o aplicativo apresentou uma ótima taxa de desempenho como podemos observar na Figura 5.4. Já em relação a segurança e fidedignidade não foram encontrado nenhum problema no aplicativo.

Figura 5.3 – Desempenho

Dispositivo	Média da CPU	Média da rede enviada	Média da rede recebida	Média da memória
 Teste dispositivos sem problemas				
Samsung Galaxy S9	4,69%	25.8 B	36.8 B	126 MB
Nokia Nokia 1	16,95%	11.6 B	126 B	69.4 MB
Google Pixel 3	4,77%	58.7 B	162 B	129 MB

Figura 5.4 – Lista de celulares compatíveis

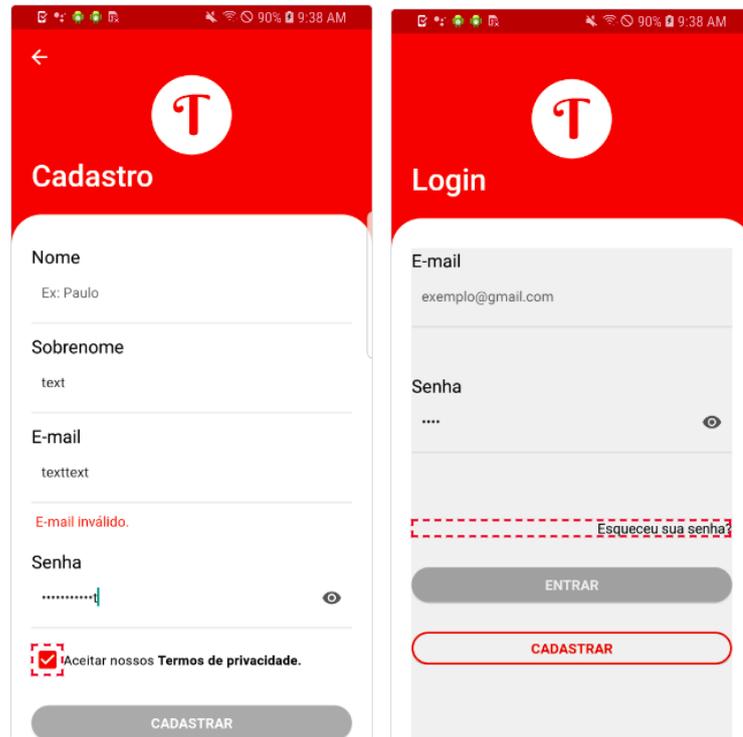
Dispositivo	RAM (memória total)	Sistema no chip	Estado
 ACE France AS0618	3 072 MB (2 826 MB)	Mediatek MT6752	 Suportado →
 ACE France BUZZ 2	1 024 MB (850 MB)	Spreadtrum SC9863A	 Suportado →
 ACE France BUZZ 3 Lite	1 024 MB (918 MB)	Spreadtrum SC7731E	 Suportado →
 ACE France BUZZ_1_L...	1 024 MB (904 MB)	Spreadtrum SC7731	 Suportado →
 ACE France BUZZ_1_...	1 024 MB (885 MB)	Mediatek MT6739WA	 Suportado →
 ACE France URBAN 1	2 048 MB (1 839 MB)	Mediatek MT6739WA	 Suportado →

Durante os testes de acessibilidade foram encontrados três problemas classificados como problemas menores, ou seja, são recomendações que podem ser seguidas para melhorar o experiência do usuário, como apresentado abaixo:

### Considerar aumentar o item clicável

Como podemos observar na Figura 5.5, a *PlayStore* recomenda aumentar o item clicável, que se encontra no tamanho de 32dp x 32dp para uma área de toque de 48dp x 48dp permitindo assim uma área de toque maior, para ter um controle adequado por parte do usuário.

Figura 5.5 – Tamanho da área de toque



### Contraste do texto

Como podemos observar na Figura 5.5, a *PlayStore* recomenda aumentar a relação de contraste do texto do primeiro plano para o texto do segundo plano deste item. A relação de contraste do texto deste item é de 4,13. Esta relação é baseada numa estimativa com a cor do primeiro plano em #FFFFFF e a cor do segundo plano em #EB3715. Considere utilizar cores que resultem numa relação de contraste superior a 4,50 para texto pequeno ou 3,00 para texto grande.

Figura 5.6 – Baixo Contraste



## 6 Conclusão

As moradias compartilhadas já têm sua existência desde a antiguidade. Esse conceito está se tornando cada vez mais popular atualmente, principalmente em cidades grandes, devido a fatores como valores de aluguéis são caros e vantagens obtidas ao morar nessas moradias (despesas básicas como luz, água e internet, entre outros, têm o valor mais acessível, já que há a sua divisão entre os moradores). Quando se trata de afazeres do cotidiano neste tipo de moradia, percebe-se que a tarefa se torna complicada devido ao número de pessoas existente no local. Pensando nisso, foi desenvolvida a prototipagem da versão 1.0 do *layout* do Tarefaça. Tratando-se de um aplicativo gamificado para gestão de moradias compartilhadas, objetivando engajar os seus usuários no controle das suas tarefas do cotidiano e gestão das contas referentes a moradia. Também realizou-se uma avaliação com o objetivo de validar a usabilidade do *layout* proposto, conseguindo uma pontuação de 82,27 que dentro da escala do método SUS e segundo John Brooke, um layout com uma pontuação superior a 68 é considerado um layout com uma ótima usabilidade. Desta forma, o aplicativo foi implementado seguindo o protótipo proposto e publicado uma primeira versão em *PlayStore* em 29 de Setembro de 2020. O objetivo desse trabalho foi concluído e o resultado foi um aplicativo capaz de auxiliar e engajar os usuários nas tarefas do dia a dia com o uso da gamificação tendo em um único aplicativo boas parte das funcionalidades necessárias para gerir uma moradia compartilhada. Visando melhorar o trabalho, listou-se algumas atividades , tais como:

- Compilar e divulgar uma versão *iOS* do aplicativo.
- Implementar a funcionalidade que permite aos usuários usarem o aplicativo em modo *offline*.
- Apresentar uma estimativa do valor das contas tendo como base as contas anteriores.
- Desenvolver uma versão *premium*.

# Referências

- 23APPS. *Atividades Domésticas*. 2013. Disponível em: <<https://play.google.com/store/apps/details?id=com.apps23.household>>. Acesso em 28-Setembro-2019.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *Requisitos ergonômicos para o trabalho com dispositivos de interação visual Parte 11: Orientações sobre usabilidade*. 2011. Disponível em: <<https://www.abntcatalogo.com.br/norma.aspx?ID=86090>>.
- BEZERRA, E. *Princípios de Análise e Projeto de Sistemas com UML*. [S.l.]: Editora Campus/Elsevier, 2007.
- BOFF, K. R.; KAUFMAN, L.; THOMAS, J. P. *Handbook of perception and human performance*. edição. Volume 1 de *serie*, 1986. (serie, Volume 1). ISBN ISBN.
- BRASIL, M. M. *Minhas Contas*. 2013. Disponível em: <<https://play.google.com/store/apps/details?id=com.msk.minhascontas>>. Acesso em 28-Setembro-2019.
- BROOKE, J. Sus - a quick and dirty usability scale. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, v. 57, p. 189–194, 09 1996.
- BROOKE, J. Sus: A retrospective. *Journal of Usability Studies*, 2013.
- CALENDAR, A. T. do list . *Any.do: Lista de Tarefas e Calendário, Lembretes*. 2011. Disponível em: <<https://play.google.com/store/apps/details?id=com.anydo>>. Acesso em 28-Setembro-2019.
- CHECKLIST. *CHECKLIST*. 2013. Disponível em: <<https://play.google.com/store/apps/details?id=com.checklist.android>>. Acesso em 28-Setembro-2019.
- CONQUER, E. *TODOIST: A ferramenta que vai te ajudar a organizar a sua vida*. 2019. Disponível em: <<https://escolaconquer.com.br/blog/todoist/>>. Acesso em 30-Outubro-2019.
- DETERDING, S.; DIXON, R. K. D.; NACKE, L. E. Gamification: Toward a definition. may 2011. Presented at the Computer Human Interaction, Vancouver, British Columbia, Canada: ACM.
- GASPERIN, C. A. *Firebase: O Que é e Como Funciona*. 2016. Disponível em: <<https://micreiros.com/firebase-o-que-e-e-como-funciona/>>. Acesso em 02-Fevereiro-2021.
- LEWIS, J.; SAURO, J. The factor structure of the system usability scale. In: . [S.l.: s.n.], 2009. v. 5619, p. 94–103.
- LIMETED, P. B. *Tarefas: Lista de afazeres e lembretes*. 2016. Disponível em: <<https://play.google.com/store/apps/details?id=com.tasks.android>>. Acesso em 28-Setembro-2019.
- MASIELLO, E.; FRIEDMANN, J. *Mastering React Native*. [S.l.]: Packt, 2017.
- NIELSEN, J.; MOLICH, R. *10 Usability Heuristics for User Interface Design*. 1990.
- NOGUEIRA, J. L. T. *Reflexões sobre métodos de avaliação de interface*. Dissertação (Mestrado) — Universidade Federal Fluminense, Niterói, 2003.

ROOMGO. *O que é moradia compartilhada e quantos tipos existem?* 2018. Disponível em: <<https://www.roomgo.com.br/blog/o-que-e-flatsharing-moradia-compartilhada/>>. Acesso em 28-Outubro-2019.

SMOBILEAPPS. *Bills Organizer Free*. 2013. Disponível em: <<https://play.google.com/store/apps/details?id=com.smapps.android.bills.trail>>. Acesso em 30-Outubro-2019.

SOMMERVILLE, I. *Engenharia de Software*. ninth. [S.l.]: PEARSON BRASIL, 2011.

SPLITWISE. *Divida gastos com os amigos*. 2019. Disponível em: <<https://www.splitwise.com/index>>. Acesso em 03-Setembro-2019.

TACT. *Lista de tarefas diárias com alarme*. 2017. Disponível em: <<https://play.google.com/store/apps/details?id=com.mg.smplan>>. Acesso em 28-Setembro-2019.

WHISPERARTS. *Tarefas domésticas: planejador e lembrete*. 2018. Disponível em: <<https://play.google.com/store/apps/details?id=com.whisperarts.diaries.housekeeping>>. Acesso em 28-Setembro-2019.

WUNDERLIST. *Wunder|Lista de tarefas pendentes, Lembretes, Compromissos*. 2019. Disponível em: <<https://www.wunderlist.com/home>>. Acesso em 28-Setembro-2019.

# **Anexos**

# ANEXO A – Formulário de Avaliação

Figura A.1 – Questão 01 da avaliação

## Avaliação da Usabilidade - Protótipo 1.0

O questionário consiste em 10 perguntas, e para cada uma delas você pode responder em uma escala de 1 a 5, onde 1 significa Discordo Completamente e 5 significa Concordo Completamente.

**\*Obrigatório**

Q1 - Acho que eu gostaria de usar este sistema com frequência. \*

1   2   3   4   5

Discordo Completamente                  Concordo Completamente

Figura A.2 – Questão 02 da avaliação

Q2 - Achei o sistema desnecessariamente complexo. \*

1   2   3   4   5

Discordo Completamente                  Concordo Completamente

Figura A.3 – Questão 03 da avaliação

Q3 - Acho o sistema de fácil usar. \*

1   2   3   4   5

Discordo Completamente                  Concordo Completamente

Figura A.4 – Questão 04 da avaliação

Q4 - Acho que iria precisar de apoio técnico para ser capaz de usar este sistema. \*

1 2 3 4 5

Discordo Completamente      Concordo Completamente

Figura A.5 – Questão 05 da avaliação

Q5 - Achei as diversas funções deste sistema bem integradas. \*

1 2 3 4 5

Discordo Completamente      Concordo Completamente

Figura A.6 – Questão 06 da avaliação

Q6 - Achei o sistema muito inconsistente. \*

1 2 3 4 5

Discordo Completamente      Concordo Completamente

Figura A.7 – Questão 07 da avaliação

Q7 - Acredito que a maioria das pessoas aprenderia a usar o sistema rapidamente. \*

1 2 3 4 5

Discordo Completamente      Concordo Completamente

Figura A.8 – Questão 08 da avaliação

Q8 - Achei o sistema muito complicado de usar. \*

1 2 3 4 5

Discordo Completamente      Concordo Completamente

Figura A.9 – Questão 09 da avaliação

Q9 - Senti-me muito confiante usando o sistema. \*

1 2 3 4 5

Discordo Completamente      Concordo Completamente

Figura A.10 – Questão 10 da avaliação

Q10 - Precisei aprender muitas coisas antes de começar a usar o sistema completamente. \*

1 2 3 4 5

Discordo Completamente      Concordo Completamente