

**UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE  
CAMPUS AVANÇADO DE NATAL  
DEPARTAMENTO DE COMPUTAÇÃO  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**LUYSLA DYANA DA SILVA TAVARES**

**TUDOLIST: UM APLICATIVO PARA ORGANIZAÇÃO DE TAREFAS**

**NATAL  
2019**

**LUYSLA DYANA DA SILVA TAVARES**

**TUDOLIST: UM APLICATIVO PARA ORGANIZAÇÃO DE TAREFAS**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte – UERN – como requisito obrigatório para a obtenção do título de bacharela em ciência da computação.

ORIENTADOR: Dr. Alberto Signoretti

NATAL

2019

T231t Tavares, Luysla Dyana da Silva  
TudoList: Um aplicativo para organiação de tarefas. /  
Luysla Dyana da Silva Tavares. - Natal/RN, 2019.  
89p.

Orientador(a): Prof. Dr. Alberto Signoretti.  
Monografia (Graduação em Ciência de Computação).  
Universidade do Estado do Rio Grande do Norte.

1. Aplicativo. 2. Mobile. 3. Web. 4. Ionic. 5.  
Organização de tarefas. I. Signoretti, Alberto. II.  
Universidade do Estado do Rio Grande do Norte. III.  
Título.

**LUYSLA DYANA DA SILVA TAVARES**

**TUDOLIST: UM APLICATIVO PARA ORGANIZAÇÃO DE TAREFAS**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte – UERN – como requisito obrigatório para a obtenção do título de bacharela em ciência da computação.

Aprovado em: \_\_\_/\_\_\_/\_\_\_

Banca examinadora

---

Dr. Alberto Signoretti

UERN

---

Me. André Gustavo Pereira da Silva

UERN

---

Me. Raul Benites Paradedá

UERN

## RESUMO

A organização das tarefas pode se tornar algo complexo sem o auxílio de ferramentas. Por isso, algumas empresas optam pelo uso de ferramentas que as auxiliem. Uma pesquisa realizada em 2019 revelou que 71,1% das empresas brasileiras utilizam alguma ferramenta para o gerenciamento de projetos (CUSTÓDIO, 2019). Portanto, esse trabalho tem como foco principal auxiliar na organização de tarefas em projetos, além de oferecer um ambiente para facilitar o trabalho em equipe. O TudoList é um aplicativo híbrido feito com o framework Ionic, ele funciona nos principais sistemas operacionais *mobile*, Android e iOS. Além disso, ele pode ser utilizado na Web. A utilização do banco de dados Cloud Firestore do Firebase, possibilita o uso do aplicativo em modo *offline* e modificação dos dados em tempo real. Com isso, o aplicativo oferece um ambiente para que o usuário possa organizar suas tarefas de forma simples e prática, contribuindo para o aumento da produtividade.

**Palavras-chave:** Aplicativo. *Mobile*. *Web*. Ionic. Organização de tarefas.

## **ABSTRACT**

The organization of the tasks can become something complex without the help of tools. Therefore, some companies choose by use of tools that help them. A survey conducted in 2019 showed that 71.1% of Brazilian companies use some tool for project management (CUSTÓDIO, 2019). Therefore, this work has the main focus to help in the organization of tasks in projects, besides offering an environment to facilitate the work in team. O TudoList is a hybrid app made with the Ionic framework, it works on the main mobile systems, Android and iOS. In addition, it can be used on the Web. Using the Cloud Firestore database of Firebase, it's possible enables offline mode use and real-time data modification. With this, the application provides an environment to be able to perform its tasks in a simple and practical, contributing to the increase of productivity.

**Keywords:** App. *Mobile*. *Web*. Ionic. Task organization.

## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
MWA	<i>Mobile Web Application</i>
NoSQL	<i>No Structured Query Language</i>
SQL	<i>Structured Query Language</i>
TI	Tecnologia da Informação
TCC	<i>Task Creator &amp; Controller</i>

## LISTA DE TABELAS

1 - Comparando as funcionalidades dos aplicativos.....	15
2 - Requisito funcional (criar projeto).....	24
3 - Requisito funcional (criar tarefa).....	25
4 - Requisito não-funcional (modo <i>offline</i> ).....	26
5 - Requisito funcional (cadastrar usuário).....	56
6 - Requisito funcional ( <i>login</i> ).....	56
7 - Requisito funcional (criar lista).....	57
8 - Requisito funcional (criar grupo).....	57
9 - Requisito funcional (prioridade na tarefa).....	58
10 - Requisito funcional (lembrete na tarefa).....	59
11 - Requisito funcional (adicionar subtarefa).....	60
12 - Requisito funcional (prioridade na subtarefa).....	60
13 - Requisito funcional (lembrete na subtarefa).....	61
14 - Requisito funcional (colaborador na tarefa).....	62
15 - Requisito funcional (editar conta).....	62
16 - Requisito funcional (excluir conta).....	63
17 - Requisito não-funcional (acesso).....	63
18 - Requisito não-funcional (usuário administrador).....	64
19 - Requisito não-funcional (software).....	64



## LISTA DE FIGURAS

1 - Tela web do sistema TCC.....	11
2 - Tela inicial do sistema SGT.....	12
3 - Tela do aplicativo OTA no sistema iOS.....	13
4 - Processo da metodologia Scrum.....	15
5 - Quadro kanban.....	16
6 - Estrutura do framework Angular.....	21
7 - Diagrama de casos de uso geral.....	26
8 - Diagrama de casos de uso para a criação de tarefas.....	27
9 - Diagrama de sequência de cadastro.....	28
10 - Diagrama de sequência de login.....	29
11 - Representação da estrutura de subcoleções utilizada inicialmente.....	30
12 - Representação da estrutura de coleções em nível raiz.....	31
13 - Diagrama de bloco simples do aplicativo.....	32
14 - Tela de login.....	34
15 - Tela de cadastro.....	34
16 - Página principal do aplicativo.....	35
17 - Botão de novo projeto.....	36
18 - Tela de novo projeto.....	37
19 - Acessar projeto.....	38
20 - Botão de criar nova lista.....	39
21 - Criar nova lista.....	39
22 - Tela do grupo.....	40
23 - Novo grupo.....	41
24 - Acessar tela de tarefas.....	42
25 - Criar novas tarefas.....	43
26 - Acessar detalhes da tarefa.....	44
27 - Tela de detalhes da tarefa.....	44
28 - Adicionar prioridade na tarefa.....	45
29 - Tarefa com prioridade.....	46
30 - Adicionar lembrete na tarefa.....	47

31 - Tarefa com lembrete.....	47
32 - Tela de subtarefas.....	48
33 - Tela de subtarefas com prioridades e lembrete.....	49
34 - Adicionar colaborador no grupo.....	50
35 - Adicionar colaborador pesquisando.....	51
36 - Concluir tarefa.....	52
37 - Tela de aviso que contém subtarefas.....	52
38 - Acessar tarefas concluídas.....	53
39 - Tela de tarefas concluídas.....	54
40 - Diagrama de casos de uso do projeto.....	69
41 - Diagrama de casos de uso de lista.....	69
42 - Diagrama de casos de uso de grupo.....	70
43 - Diagrama de casos de uso para a criação de subtarefas.....	70
44 - Diagrama de sequência de projetos.....	71
45 - Erro e-mail ou senha incorretos.....	72
46 - Aviso de verificação do e-mail.....	73
47 - Janela de redefinição de senha.....	73
48 - Menu principal do aplicativo TudoList.....	74
49 - Tarefas em que o usuário é colaborador.....	75
50 - detalhes das tarefas do colaborador.....	75
51 - Tela de meus projetos.....	76
52 - Botão de editar projetos.....	77
53 - Tela de editar e excluir projetos.....	77
54 - Botão de editar perfil.....	78
55 - Editar perfil.....	78
56 - Botão de pesquisar projetos.....	79
57 - Tela de pesquisa de projetos.....	79
58 - Botão para editar listas.....	80
59 - Editar listas.....	80
60 - Botão de editar grupo.....	81
61 - Editar grupo.....	81
62 - Botão de editar grupo.....	82
63 - Tela de excluir grupo.....	82

64 - Detalhes da tarefa com colaborador.....	83
--	----

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>8</b>
1.1 OBJETIVO.....	9
<b>1.1.1 Objetivos específicos.....</b>	<b>9</b>
1.2 ESTRUTURA DO TRABALHO.....	9
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>10</b>
2.1 A IMPORTÂNCIA DA ORGANIZAÇÃO EM PROJETOS.....	10
2.2 TRABALHOS RELACIONADOS.....	11
<b>2.2.1 Task Creator &amp; Controller (TCC).....</b>	<b>11</b>
<b>2.2.2 SGT.....</b>	<b>12</b>
<b>2.2.3 OTA APP.....</b>	<b>12</b>
2.3 COMPARANDO AS FUNCIONALIDADES.....	13
<b>3 METODOLOGIA.....</b>	<b>15</b>
3.1 SCRUM.....	15
3.2 KANBAN.....	16
3.3 SCRUMBAN.....	16
<b>4 FERRAMENTAS E TECNOLOGIAS UTILIZADAS.....</b>	<b>18</b>
4.1 VISUAL STUDIO CODE.....	18
4.2 GIT.....	18
4.3 <i>NO STRUCTURED QUERY LANGUAGE</i> (NoSQL).....	18
4.4 FIREBASE.....	19
<b>4.4.1 Cloud Firestore.....</b>	<b>19</b>
<b>4.4.2 Cloud Storage.....</b>	<b>20</b>
4.5 ANGULAR.....	20
4.6 IONIC.....	21
<b>5 DESENVOLVIMENTO.....</b>	<b>23</b>
5.1 DOCUMENTO DE REQUISITOS.....	23
<b>5.1.1 Requisitos funcionais.....</b>	<b>23</b>
<b>5.1.2 Requisitos não-funcionais.....</b>	<b>25</b>
5.2 DIAGRAMAS UML.....	25
<b>5.2.1 Diagrama de casos de uso.....</b>	<b>26</b>
5.2.1.1 Criar tarefa.....	27

<b>5.2.2 Diagrama de seqüência.....</b>	<b>27</b>
5.2.2.1 Cadastro.....	27
5.2.2.2 Login.....	28
5.3 MODELAGEM DO BANCO DE DADOS.....	29
<b>6 O APLICATIVO TUDOLIST.....</b>	<b>32</b>
6.1 DIAGRAMA DE BLOCO.....	32
6.2 PRINCIPAIS TELAS DO APLICATIVO.....	33
<b>6.2.1 Tela de cadastro e <i>login</i>.....</b>	<b>33</b>
<b>6.2.2 Home.....</b>	<b>35</b>
6.2.2.2 Acessar projetos.....	36
<b>6.2.3 Listas.....</b>	<b>37</b>
<b>6.2.4 Grupo.....</b>	<b>38</b>
<b>6.2.5 Tarefas.....</b>	<b>39</b>
6.2.5.1 Prioridade.....	41
6.2.5.2 Lembrete.....	42
6.2.5.3 Subtarefas.....	43
6.2.5.4 Colaborador.....	45
6.2.5.5 Concluir tarefa.....	46
<b>7 TRABALHOS FUTUROS.....</b>	<b>49</b>
<b>8 CONSIDERAÇÕES FINAIS.....</b>	<b>51</b>
<b>REFERÊNCIAS.....</b>	<b>52</b>
<b>APÊNDICE.....</b>	<b>54</b>

## 1 INTRODUÇÃO

Com o avanço tecnológico e a chegada dos *smartphones*, o uso de aplicativos se tornou bastante comum, principalmente entre os brasileiros. Uma pesquisa realizada pelo App Annie revelou que “o brasileiro usa em média 12 aplicativos a cada 24 horas em seu *smartphone*” (RIBEIRO, 2018). Mediante isso, o mercado tem investido na criação de aplicativos com diversos propósitos. Em ambientes empresariais, o uso de ferramentas, bem como aplicativos também se tornou comum. Segundo Custódio (2019), 77,1% das empresas brasileiras fazem uso de alguma ferramenta de gestão de projetos, com o intuito de aumentar a produtividade e melhorar a organização.

A gestão de projetos, bem como a organização das tarefas que o compõem pode se tornar algo complicado para as empresas. Sem um ambiente organizado, alguns fatores podem dificultar a produtividade e o andamento do projeto, como a má organização do projeto e a não definição de prazos. É possível que essas dificuldades nos projetos contribuam para o aumento da improdutividade.

Fatores como, não definir as tarefas a serem executadas no projeto, não definir prazos ou não definir prioridades nas tarefas, pode acarretar na má organização do projeto sendo possível que o projeto tenha dificuldades para se concluir ou seja adiado. Por isso, é importante que se tenha organização para que haja sucesso na conclusão do projeto.

O surgimento dos aplicativos de organização de tarefas tem o intuito de ajudar na organização e como consequência, diminuir a improdutividade. Esses aplicativos oferecem diversas funcionalidades, porém acabam não se tornando simples em sua utilização. Com isso, esse trabalho propõe a criação de um aplicativo simples para ser utilizado para a organização de tarefas com o intuito de auxiliar uma melhor organização, como consequência uma melhor gestão no tempo e produtividade no projeto.

Basicamente, o TudoList irá disponibilizar alguns recursos que podem ser considerados importantes para se ter uma melhor organização das tarefas. Funcionalidades como, ambiente para a criação de projetos, criação de listas para inserir as tarefas e subtarefas. A definição de prazos das tarefas e subtarefas com prioridades e a delegação de tarefas. Além disso, disponibilizar um ambiente que auxilie na comunicação e no compartilhamento de informações pela equipe do projeto.

## 1.1 OBJETIVO

O foco deste trabalho é auxiliar na organização de tarefas em projetos, com o objetivo de diminuir a improdutividade, que pode estar ligada à desorganização. Além disso, oferecer um ambiente que facilite o trabalho em equipe.

### 1.1.1 Objetivos específicos

Os objetivos específicos do trabalho são:

- Possibilidade de criar projetos, listas, grupos, tarefas e subtarefas;
- Oferecer uma aplicação móvel que possa ser utilizada em qualquer *smartphone*, com Android<sup>1</sup> ou iOS<sup>2</sup> e também na Web.

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em 8 capítulos. A partir daqui o próximo Capítulo é o 2, que corresponde à fundamentação teórica, explicando alguns conceitos que envolvem a problemática e a apresentação e comparação dos trabalhos relacionados.

O Capítulo 3 explica as metodologias que foram adotadas para o desenvolvimento do projeto.

O Capítulo 4 mostra as ferramentas e tecnologias que foram utilizadas e serviram para o desenvolvimento do aplicativo TudoList.

O Capítulo 5 mostra o desenvolvimento exibindo a documentação de requisitos, a modelagem do banco de dados e alguns diagramas UML do aplicativo.

O Capítulo 6 exhibe algumas telas das funcionalidades do aplicativo TudoList.

O capítulo 7 fala sobre as propostas de trabalhos futuros no aplicativo, a adição de novas funcionalidades e a modificação de algumas partes do aplicativo.

Por último, o Capítulo 8 são as considerações finais sobre o trabalho desenvolvido e a relevância do presente estudo na sociedade.

---

1 Disponível em: <<https://www.android.com/>>

2 Disponível em: <<https://developer.apple.com/ios/>>

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados alguns conceitos que envolvem o objetivo do trabalho e a análise e comparação de trabalhos semelhantes.

### 2.1 A IMPORTÂNCIA DA ORGANIZAÇÃO EM PROJETOS

A organização de tarefas é uma parte importante no desenvolvimento de projetos. Sem ela, é possível que alguns fatores dificultem o andamento do projeto, bem como seu desenvolvimento. Organizar as tarefas auxilia na produtividade e no bom andamento do projeto.

Um dos fatores que podem dificultar o andamento do projeto é a má organização e não definição das tarefas que precisam ser executadas. Em um projeto, é importante haver a definição das atividades que serão realizadas para sua conclusão. Sem isso, é possível que haja a execução de atividades que não eram necessárias.

A definição de prazos para a execução das tarefas é outro fator importante e que se não for feito, pode acabar acarretando em problemas como o aumento do tempo de projeto. A definição das datas de início e término da tarefa, irá auxiliar na execução das tarefas e no tempo que será gasto para a conclusão do projeto.

A definição de prioridade nas tarefas também é outro fator importante e que não pode ser esquecido. Essa propriedade irá auxiliar no andamento do projeto de acordo com as principais funcionalidades. Assim, o projeto será sempre feito seguindo níveis de tarefas que tem maior relevância e são mais importantes para a conclusão do projeto.

A comunicação com a equipe que executa o projeto também é um fator importante. Sem uma boa comunicação, é possível que algumas informações não sejam compartilhadas de forma correta. Além disso, a delegação de tarefas para os membros da equipe, auxilia no andamento do projeto, assim os membros da equipe saberão quais atividades pertencem a ele e mantém o foco em executá-las.



## 2.2 TRABALHOS RELACIONADOS

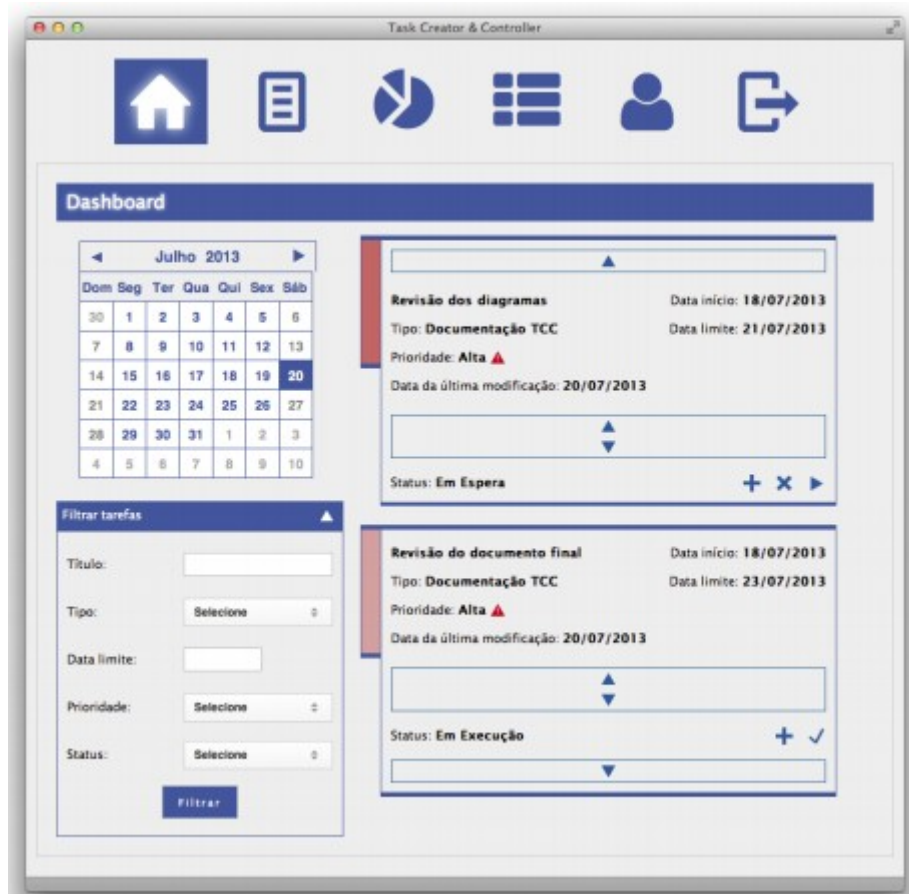
Nesta seção são apresentados alguns trabalhos relacionados que possuem o mesmo objetivo ou são parecidos com o aplicativo TudoList.

### 2.2.1 Task Creator & Controller (TCC)

O Task Creator & Controller é um sistema desenvolvido para o controle de tarefas utilizando uma hierarquia piramidal (ALMEIDA, 2013). Pode ser usado na Web e também *mobile* nos sistemas Android e iOS.

A aplicação possibilita ao usuário principal (gerente) criar tarefas e subtarefas e delegá-las a outros usuários (funcionários). As tarefas e as subtarefas possuem prioridade e lembrete. É possível visualizar o status da tarefa/subtarefa e a data a ser entregue. Além disso, o sistema fornece relatórios de alguns dados, como tarefas e funcionários.

**Figura 1:** Tela web do sistema TCC



**Fonte:** <https://acervodigital.ufpr.br/bitstream/handle/1884/40685/Task%20Creator%20%26%20Controller.pdf?sequence=1&isAllowed=y> (2013)

## 2.2.2 SGT

O SGT é um sistema web para o gerenciamento de tarefas em setores de Tecnologia da Informação (TI). Desenvolvido em C#, ele oferece recursos de criação e delegação de tarefas, bem como a criação de datas para indicar o término da tarefa (MELO, 2017).

**Figura 2:** Tela inicial do sistema SGT

Status	Nome	Descrição	Recorrência	Usuário	Uploads	Atualizar	Excluir
Concluído	Banco de Dados	Realizar backup das tabelas	Semanal	Usuario	Uploads	Atualizar	Excluir
Aberto	Impressoras	Verificar funcionamento das impressoras	Quinzenal	Teste	Concluir	Atualizar	Excluir
Concluído	Notebooks	Realizar manutenção	Mensal	Usuario	Uploads	Atualizar	Excluir
Concluído	Rede	Verificar integridade da rede	Diário	Usuario	Uploads	Atualizar	Excluir
Aberto	Softwares	Manter softwares atualizados	Mensal	Default	Concluir	Atualizar	Excluir
Aberto	Telefonia	Verificar integridade	Semanal	Default	Concluir	Atualizar	Excluir
Atrasado	Testes de Software	Executar bateria de testes	Quinzenal	Default	Concluir	Atualizar	Excluir

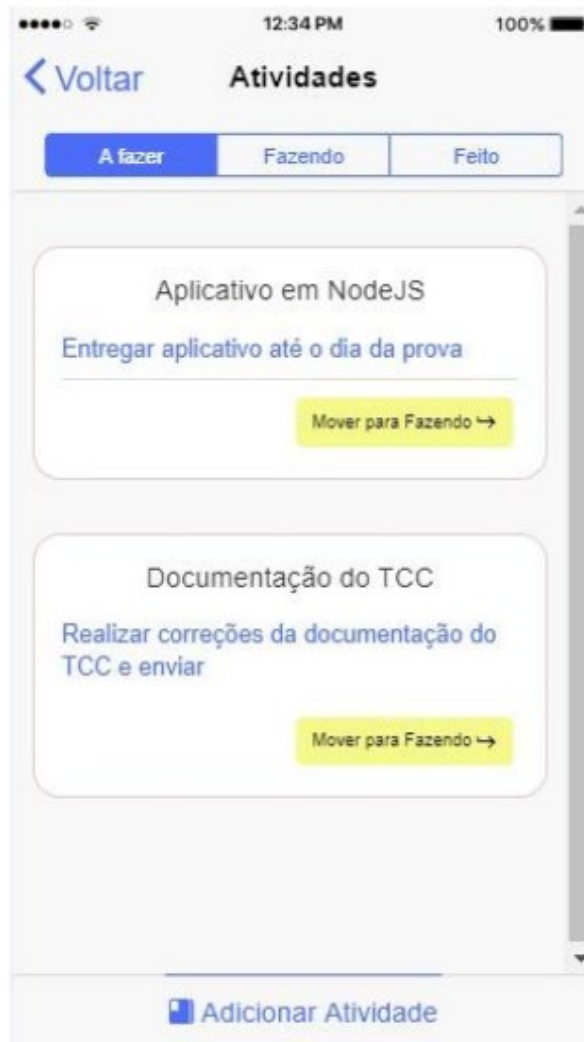
**Fonte:** <http://revista.faqi.edu.br/index.php/ADS/article/view/286/301> (2017)

## 2.2.3 OTA APP

O OTA APP é um aplicativo desenvolvido para a organização de tarefas acadêmicas utilizando a metodologia Kanban<sup>1</sup> (RAMOS;BRANCALHÃO;PIRES,2018). Desenvolvido utilizando o framework Ionic (IONIC, 2019), ele possibilita ser utilizado nos sistemas Android e iOS.

Ao utilizar o aplicativo o usuário tem a possibilidade de criar atividades e adicioná-las a um quadro Kanban, uma metodologia utilizada para o controle de tarefas. Basicamente, o quadro possui três listas: tarefas a fazer, tarefas em andamento e tarefas concluídas. O OTA APP possibilita ao usuário mover as atividades entre as listas do quadro dependendo do seu estado atual (Figura 3).

1 Veja mais em: <<https://www.atlassian.com/agile/kanban/boards>>

**Figura 3:** Tela do aplicativo OTA no sistema iOS

**Fonte:** <http://periodicos.unifacel.com.br/index.php/resiget/article/view/1624> (2018)

### 2.3 COMPARANDO AS FUNCIONALIDADES

Os trabalhos relacionados apresentados possuem objetivos parecidos ou semelhantes aos do aplicativo TudoList. O aplicativo Task Creator & Controller (TCC) possui o mesmo objetivo proposto neste trabalho, se distinguindo apenas na arquitetura utilizada. No caso do TCC utiliza-se uma arquitetura piramidal, já o TudoList não possui nenhuma arquitetura associada, somente baseia-se na criação de projetos com listas contendo tarefas e subtarefas. Outro trabalho relacionado é o SGT, um sistema web para gerenciar tarefas em setores de TI. Possui algumas características parecidas com o TudoList, como a criação de tarefas e a delegação das mesmas, porém não é um aplicativo, além disso o objetivo de sua utilização não é o mesmo, mas sim, parecido. Já o OTA APP

é um aplicativo híbrido desenvolvido com Ionic utilizando o Firebase, usa a estrutura Kanban para a criação de atividades acadêmicas. Em relação ao objetivo proposto neste trabalho, o OTA APP possui pouca semelhança, porém utiliza-se o conceito de criação de atividades e a conclusão das mesmas.

A Tabela 1 compara as principais funcionalidades entre o TudoList e as outras aplicações. Pode-se perceber que em relação aos outros, o TudoList se destaca por diferenciais como o uso do aplicativo em modo offline e a criação de lembretes nas tarefas e subtarefas.

**Tabela 1:** Comparando as funcionalidades dos aplicativos

<b>Funcionalidades</b>	Tarefas	Tarefas e subtarefas com prioridade	Lembretes nas tarefas e subtarefas	Delegação de tarefas	Favoritar projetos	Relatórios	Andro id/iOS	Web	Modo offline
<b>TudoList</b>	x	x	x	x	x		x	x	x
<b>Task Creator &amp; Controller (TCC)</b>	x	x		x		x	x	x	
<b>SGT</b>	x			x		x		x	
<b>OTA APP</b>	x						x		

**Fonte:** Tabela criada pelo próprio autor

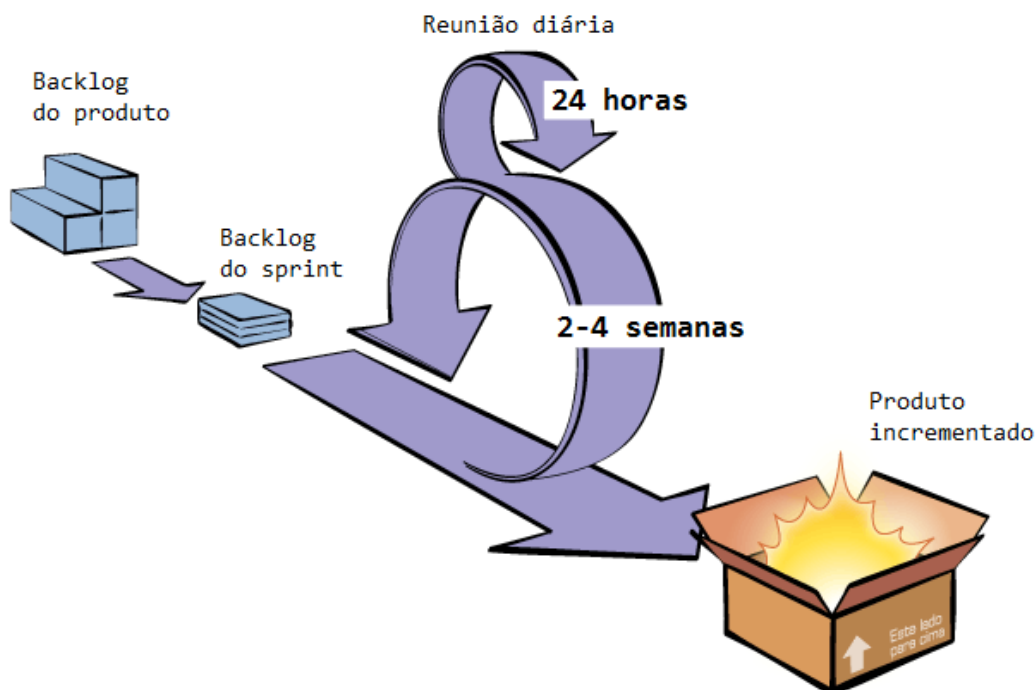
### 3 METODOLOGIA

Neste capítulo são apresentados alguns métodos que foram utilizados no desenvolvimento deste projeto.

#### 3.1 SCRUM

O Scrum é um *framework* utilizado para gerenciar o desenvolvimento de projetos, sejam eles simples ou complexos (SCRUM, 2019). Ele utiliza ciclos iterativos e incrementais para o desenvolvimento de diversas partes do projeto. A Figura 4 ilustra no geral o funcionamento do Scrum. Os ciclos servem para desenvolver as funcionalidades que foram previamente listadas chamadas de backlog do produto. A cada ciclo (sprint), é necessário listar as funcionalidades que serão desenvolvidas (backlog do sprint), sendo possível atribuir novas prioridades as funcionalidades. Além disso, define-se um tempo para o desenvolvimento das funcionalidades escolhidas para cada ciclo, que pode ser de até 4 semanas.

**Figura 4:** Processo da metodologia Scrum



**Fonte:** <https://www.desenvolvimentoagil.com.br/scrum/> (2014)<sup>1</sup>

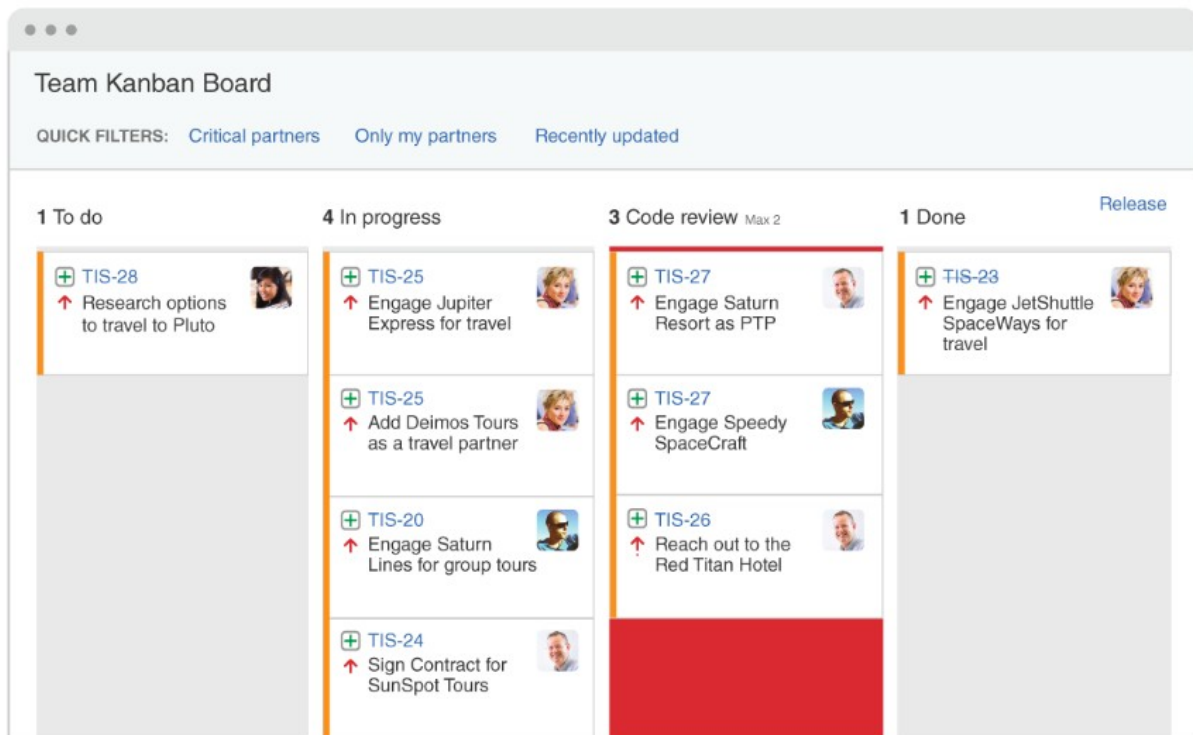
<sup>1</sup> Imagem modificada pelo autor

As reuniões diárias são feitas com o objetivo de avaliar o andamento do projeto. Ao final de cada ciclo pode-se avaliar o que foi feito e recomençar outro ciclo até a conclusão do projeto.

### 3.2 KANBAN

O kanban é uma metodologia que busca organizar o fluxo de atividades em projetos que utilizam metodologias ágeis (RADIGAN, 2019). Ele utiliza quadros para organizar as atividades que serão feitas, que estão em andamento e as que já foram feitas. Neste trabalho, o kanban serviu para complementar o uso do scrum. A Figura 5 ilustra como é a estrutura do kanban.

**Figura 5:** Quadro kanban



**Fonte:** <https://www.atlassian.com/agile/kanban> (2019)

### 3.3 SCRUMBAN

O Scrumban é a junção das metodologias Scrum e Kanban. Inicialmente, criou-se um quadro kanban no trello<sup>1</sup>, um sistema para a criação de projetos e lista de tarefas utilizando quadros. As atividades que ainda seriam feitas representavam a lista de funcionalidades criadas no scrum para o projeto, a cada ciclo do Scrum, as atividades ficavam na lista das tarefas em andamento e por

<sup>1</sup> Veja mais em: <<https://trello.com/en>>

fim, assim que as atividades eram concluídas, elas iam para a lista das tarefas que já foram feitas. A cada ciclo do Scrum, esse processo foi repetido, utilizando o quadro kanban para manter a organização.

## 4 FERRAMENTAS E TECNOLOGIAS UTILIZADAS

Neste capítulo são apresentadas as ferramentas e tecnologias que foram utilizadas para o desenvolvimento do aplicativo, bem como os motivos que pelas quais foram selecionadas.

### 4.1 VISUAL STUDIO CODE

O Visual Studio Code é uma *Integrated Development Environment* (IDE) criada pela empresa Microsoft (MICROSOFT, 2019). Uma de suas principais funcionalidades é oferecer extensões para facilitar no desenvolvimento do código, auxiliando no trabalho do desenvolvedor. Neste projeto, além de outras extensões básicas para o desenvolvimento web, foi utilizado um pacote com extensões para o desenvolvimento com o *framework* Ionic.

### 4.2 GIT

O Git é um sistema de código aberto utilizado para o controle de versionamento de projetos (GIT, 2019). Neste trabalho, o Git foi utilizado para armazenar as atualizações feitas no código e fazer o controle das versões.

### 4.3 NO STRUCTURED QUERY LANGUAGE (NoSQL)

O NoSQL é um termo utilizado para caracterizar bancos e modelos de dados que surgiram com estruturas diferentes dos bancos relacionais. Os bancos relacionais, que utilizam o *Structured Query Language* (SQL) possuem uma estrutura de esquemas com tabelas. Já os bancos não relacionais possuem estruturas variadas, como por exemplo: chave-valor, documento, etc. (AMAZON WEB SERVICES, 2019).

Os bancos de dados NoSQL possuem algumas vantagens, como: flexibilidade, é capaz de se adaptar aos dados que estão sendo armazenados; escalabilidade, é capaz de distribuir melhor os dados e conseqüentemente ter uma capacidade maior de armazenar os dados; independentes, foram projetados para exigir menos gerenciamento por parte do desenvolvedor; economia, são bancos que



exigem menos gastos já que utilizam *clusters* e servidores mais baratos que os bancos relacionais (MUNIZ, 2018).

Os bancos não-relacionais costumam utilizar o conceito de replicação de valores e a não utilização de chave estrangeira. No geral, os dados armazenados utilizam a replicação de dados, isso evita o uso de chave estrangeira que necessita da função JOIN ao pegar os dados no banco, em alguns casos essa função afeta muito a performance da consulta, o que pode acarretar em lentidão, além disso a replicação de dados pode ser bastante útil caso a informação fique indisponível em alguma parte do banco (ALVES, 2015).

É importante ressaltar que assim como qualquer tecnologia, o NoSQL também possui suas desvantagens em alguns casos, como: ausência de ferramentas para testes de desempenho, poucas pessoas com conhecimentos em NoSQL, não possui linguagem de consulta padrão, como o SQL. No geral, o NoSQL é uma ótima opção para aplicações que não possuem modelos de dados definidos, aplicações que posteriormente podem gerar uma grande quantidade de dados e aplicações *mobile* (FERREIRA, 2018).

#### 4.4 FIREBASE

O Firebase é uma plataforma móvel do Google que fornece diversas *Application Programming Interfaces* (APIs) com o objetivo de auxiliar o desenvolvedor na construção do *back-end* de seus aplicativos. De modo geral, os recursos disponíveis pelo Firebase incluem o armazenamento de dados e gerenciamento do servidor (FIREBASE, 2019a). Para o armazenamento de dados do aplicativo utilizou-se um dos banco de dados que o Firebase disponibiliza, o Cloud Firestore (FIREBASE, 2018a).

##### 4.4.1 Cloud Firestore

O Firebase (FIREBASE, 2019a), atualmente, oferece dois tipos de bancos de dados NoSQL: o Realtime Database e o Cloud Firestore. A estrutura dos dois bancos são distintas. A estrutura do Cloud Firestore é em coleções, subcoleções e documentos para o armazenamento de dados, já o Realtime Database utiliza a estrutura em formato JSON<sup>1</sup>. A escolha do banco se deu por meio da

---

1 Veja mais em: <<https://www.json.org/>>

estrutura que melhor se encaixou com a modelagem do banco desenvolvido para o aplicativo. Nesse caso: o Cloud Firestore.

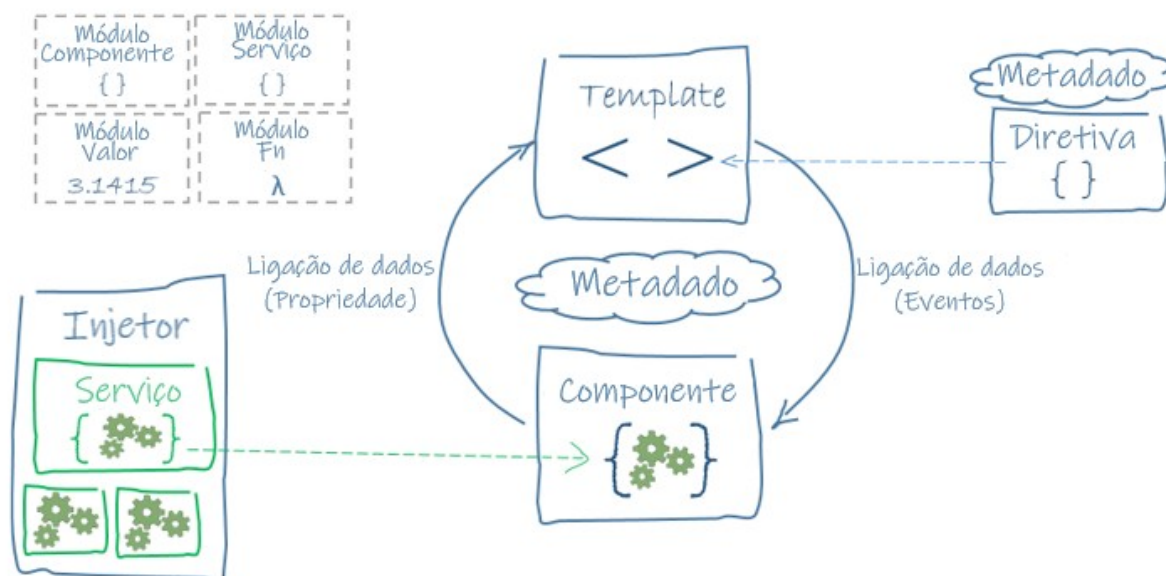
O Cloud Firestore é um banco de dados NoSQL hospedado na nuvem. Esse banco possui uma estrutura baseada em documentos, onde cada documento é armazenado em coleções (contêineres), que servem para organizar os dados (FIREBASE, 2018a). Os documentos podem armazenar vários tipos de dados, desde dados simples até dados aninhados ou subcoleções. O Cloud Firestore possui algumas formas de estruturação de dados, cada estrutura tem uma característica específica podendo ser utilizada de acordo com o objetivo do projeto (FIREBASE, 2018b).

#### 4.4.2 Cloud Storage

O *Cloud Storage* é um serviço oferecido pelo Firebase para o armazenamento de mídia dos projetos criados pelos usuários (FIREBASE, 2019c). Com esse serviço, é possível armazenar ou realizar o download a mídia da aplicação. Nesse trabalho, o *Cloud Storage* para o armazenamento das mídias geradas pelos usuários do aplicativo.

#### 4.5 ANGULAR

Angular é um framework para o desenvolvimento de aplicações web (ANGULAR, 2019). Ele utiliza o conceito de componentes e serviços. Basicamente, os componentes são as *views* da aplicação e os serviços são as funções utilizadas pelos componentes. A Figura 6 ilustra o funcionamento do Angular. A aplicação pode utilizar módulos para o funcionamento, existem os módulos nativos porém pode-se utilizar outros módulos, como componentes, páginas e módulos necessários para a aplicação. A organização do código por meio de módulos faz diferença no carregamento da aplicação que utiliza somente o que é necessário para carregar a página (*lazy loading*), isso evita que outros dados desnecessários sejam carregados, como consequência a aplicação terá maior rapidez. A parte visual dos componentes é chamada de *template* que utiliza o *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS). Existem duas formas para a exibição dos dados no *template*: a ligação de dados por evento, que utiliza os dados de entrada do HTML para o componente e a ligação de dados por propriedade que utiliza os dados a partir do componente e os exibe no *template*. As diretivas servem para modificar as informações do *template* de acordo com a lógica do programa e é por meio do injetor que é possível fornecer os serviços para o componente (ANGULAR, 2019).

**Figura 6:** Estrutura do framework Angular

**Fonte:** <https://angular.io/guide/architecture> (2019)<sup>1</sup>

#### 4.6 IONIC

O Ionic é um *framework* para o desenvolvimento de aplicações híbridas, isso significa que o aplicativo desenvolvido com Ionic será multiplataforma, podendo ser utilizado em diversos sistemas operacionais de dispositivos móveis existentes no mercado. Aplicativos híbridos possuem a característica de serem parte nativos e parte *Mobile Web Application* (MWA), pois podem ser utilizados das duas formas. Na forma nativa, o aplicativo é específico para cada sistema operacional, onde o usuário poderá baixar a aplicação por meio de uma loja e mantê-lo no celular e na forma web, o aplicativo pode ser utilizado por meio de um navegador embutido no aplicativo, sendo todo o conteúdo, ou apenas parte dele carregado na web.

O Ionic utiliza o Angular para o desenvolvimento dos projetos, porém no primeiro semestre de 2019, o Ionic lançou sua versão 4, dando a possibilidade do usuário integrar o projeto com outros frameworks web além do Angular, como o Vue.js e o React (LYNCH, 2019). No momento em que o aplicativo estava em desenvolvimento, a versão 4 do Ionic estava em fase beta. Para evitar erros que ainda podiam ocorrer na versão mais recente, foi utilizada uma versão anterior do Ionic, a 3.20.0, que estava estável.

Um dos recursos oferecidos pelo Ionic é a função de *lazy loading*, que tem como objetivo deixar a aplicação mais leve na utilização. Basicamente, essa função é utilizada nas páginas para

<sup>1</sup> Imagem modificada pelo autor

que somente o conteúdo selecionado, seja carregado. No geral, as aplicações tendem a realizar um pré-carregamento de dados, porém com o *lazy loading* é possível que a aplicação carregue somente os dados necessários para executar determinada página. Com isso, a aplicação ficará mais leve na hora de carregar os dados.

## 5 DESENVOLVIMENTO

Neste capítulo são apresentadas todas as partes do desenvolvimento de acordo com a documentação de engenharia de *software*.

### 5.1 DOCUMENTO DE REQUISITOS

A seguir, são apresentados os principais requisitos funcionais e não-funcionais do sistema.

#### 5.1.1 Requisitos funcionais

Os requisitos funcionais são as funções principais do sistema. A seguir são apresentados os que possuem maior relevância no sistema. Os demais requisitos estão listados no Apêndice A.

**Tabela 2:** Requisito funcional (criar projeto)

ID	RF003
<b>Função</b>	Criar projeto
<b>Descrição</b>	Essa função tem como objetivo a criação de projetos
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> no aplicativo [RF002]
<b>Pós-condições</b>	Novo projeto criado
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário insere o nome e a descrição do projeto;</li> <li>2. O sistema envia as informações para o banco;</li> <li>3. Um novo projeto é criado;</li> </ol>
<b>Fluxo alternativo</b>	<p><b>FA001 – Favoritar projeto</b></p> <ol style="list-style-type: none"> <li>1. O usuário clica no botão de favoritar projeto;</li> <li>2. O projeto é favoritado;</li> <li>3. O sistema envia a informação para o banco;</li> </ol> <p><b>FA002 – Desfavoritar projeto</b></p> <ol style="list-style-type: none"> <li>1. O usuário clica no botão de favoritar, caso ele esteja favoritado;</li> <li>2. O projeto é desfavoritado;</li> <li>3. O sistema envia a informação para o banco;</li> </ol> <p><b>FA003 – Editar projeto</b></p> <ol style="list-style-type: none"> <li>1. O usuário clica no botão de editar;</li> <li>2. Seleciona o projeto;</li> <li>3. Edita o projeto;</li> <li>4. O sistema envia as novas informações para o banco;</li> </ol> <p><b>FA004 – Excluir projeto</b></p> <ol style="list-style-type: none"> <li>1. O usuário clica no botão de excluir projeto;</li> <li>2. O sistema verifica se o usuário tem permissão de administrador para excluir o projeto;</li> <li>3. Caso o usuário seja administrador, o projeto é excluído;</li> <li>4. O sistema envia as informações para o banco;</li> </ol>

**Fonte:** Tabela criada pelo próprio autor

Tabela 3: Requisito funcional (criar tarefa)

ID	RF006
Função	Criar tarefa
Descrição	Essa função tem como objetivo a criação de tarefas na lista
Ator	Usuário
Pré-condições	Ter efetuado o <i>login</i> [RF002] e ter criado uma lista [RF004]
Pós-condições	Nova tarefa criada
Fluxo principal	<ol style="list-style-type: none"> <li>1. Usuário seleciona a lista para inserir a tarefa;</li> <li>2. Insere o nome da tarefa;</li> <li>3. O sistema envia a informação para o banco;</li> <li>4. Nova tarefa é inserida na lista.</li> </ol>
Fluxo alternativo	<p><b>FA001 – Concluir tarefa sem subtarefas</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona a lista para concluir a tarefa;</li> <li>2. Seleciona a tarefa;</li> <li>3. A tarefa é concluída;</li> <li>5. O banco é atualizado;</li> </ol> <p><b>FA002 – Concluir tarefa com subtarefas</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona a lista para concluir a tarefa;</li> <li>2. Seleciona a tarefa;</li> <li>3. O sistema verifica no banco se existem subtarefas vinculadas;</li> <li>4. Caso exista, o usuário cancela a ação ou seleciona que deseja excluir mesmo assim e a tarefa é concluída;</li> <li>5. O banco é atualizado;</li> </ol> <p><b>FA003 – Restaurar tarefas</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona a lista para restaurar tarefa;</li> <li>2. Seleciona a tarefa na lista de tarefas concluídas;</li> <li>3. A tarefa é restaurada.</li> <li>4. O banco é atualizado;</li> </ol>

Fonte: Tabela criada pelo próprio autor

### 5.1.2 Requisitos não-funcionais

Os requisitos não-funcionais são funções do sistema que possuem menor relevância. A seguir é apresentado o principal requisito não-funcional do sistema. Os demais requisitos não-funcionais estão listados no Apêndice B.

**Tabela 4:** Requisito não-funcional (modo *offline*)

ID	RNF001
<b>Função</b>	Modo <i>offline</i>
<b>Descrição</b>	Essa função tem como objetivo permitir o uso do aplicativo em modo <i>offline</i> , onde o usuário poderá ver informações já existentes e inserir novos dados
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002]
<b>Pós-condições</b>	Sincroniza os dados inseridos ou excluídos assim que o usuário estiver <i>online</i>
<b>Fluxo principal</b>	1. Usuário utiliza o aplicativo em modo <i>offline</i> ; 2. Insere algum dado ou exclui; 3. O dado é armazenado ou excluído temporariamente; 4. Assim que o usuário estiver <i>online</i> os dados são sincronizados e atualizados.
<b>Fluxo alternativo</b>	Nenhum

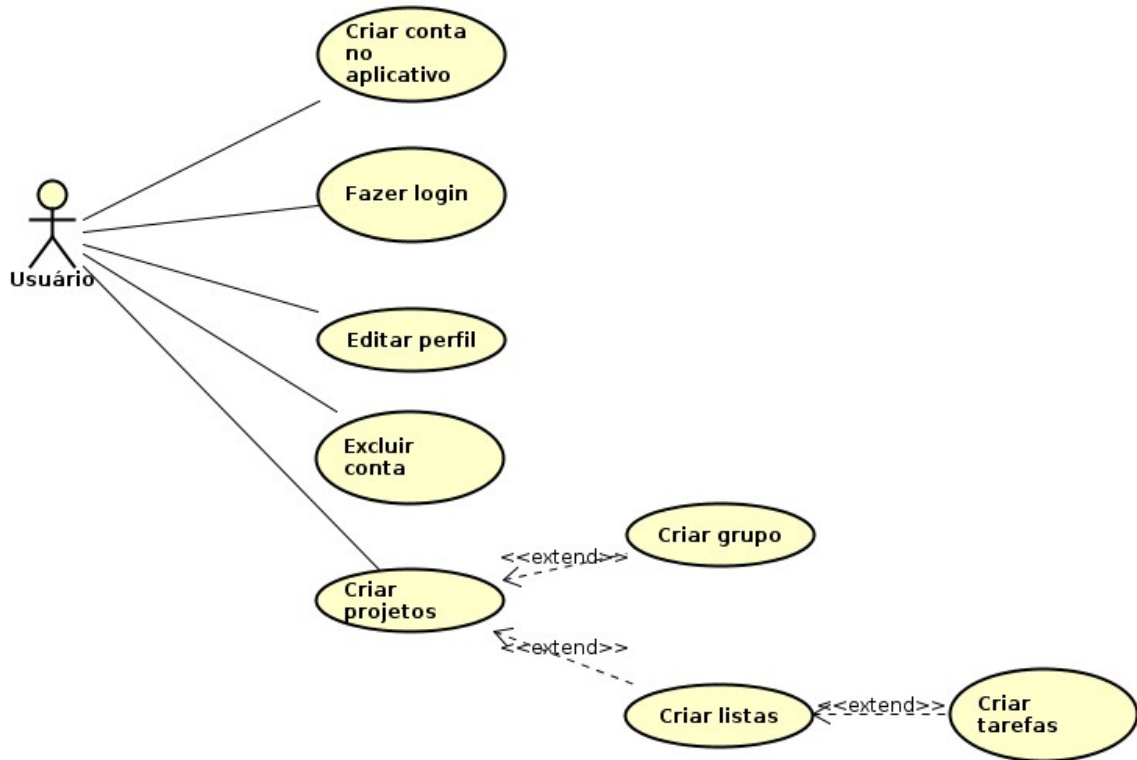
**Fonte:** Tabela criada pelo próprio autor

## 5.2 DIAGRAMAS UML

Nesta seção serão apresentados alguns diagramas UML que foram utilizados para exibir as funcionalidades correspondentes ao aplicativo TudoList.

### 5.2.1 Diagrama de casos de uso

O diagrama de casos de uso exibe a relação do usuário com o sistema e quais ações ele pode executar no aplicativo. A Figura 7 mostra os casos de uso para as principais funcionalidades do TudoList. Basicamente, o diagrama geral exibe quais ações o usuário poderá realizar no aplicativo. Inicialmente, para acessar as funcionalidades do aplicativo é necessário que o usuário possua conta, para isso existe a função de cadastrar usuário. Após o usuário ter se cadastrado no aplicativo, é necessário realizar o login para utilizá-lo. As principais funcionalidades que o usuário poderá utilizar é a criação de projetos, grupos, listas, tarefas e subtarefas, além de ter a possibilidade de alterar suas informações da conta ou excluí-la. Os demais diagramas de casos do uso do sistema estão no Apêndice C.

**Figura 7:** Diagrama de casos de uso geral

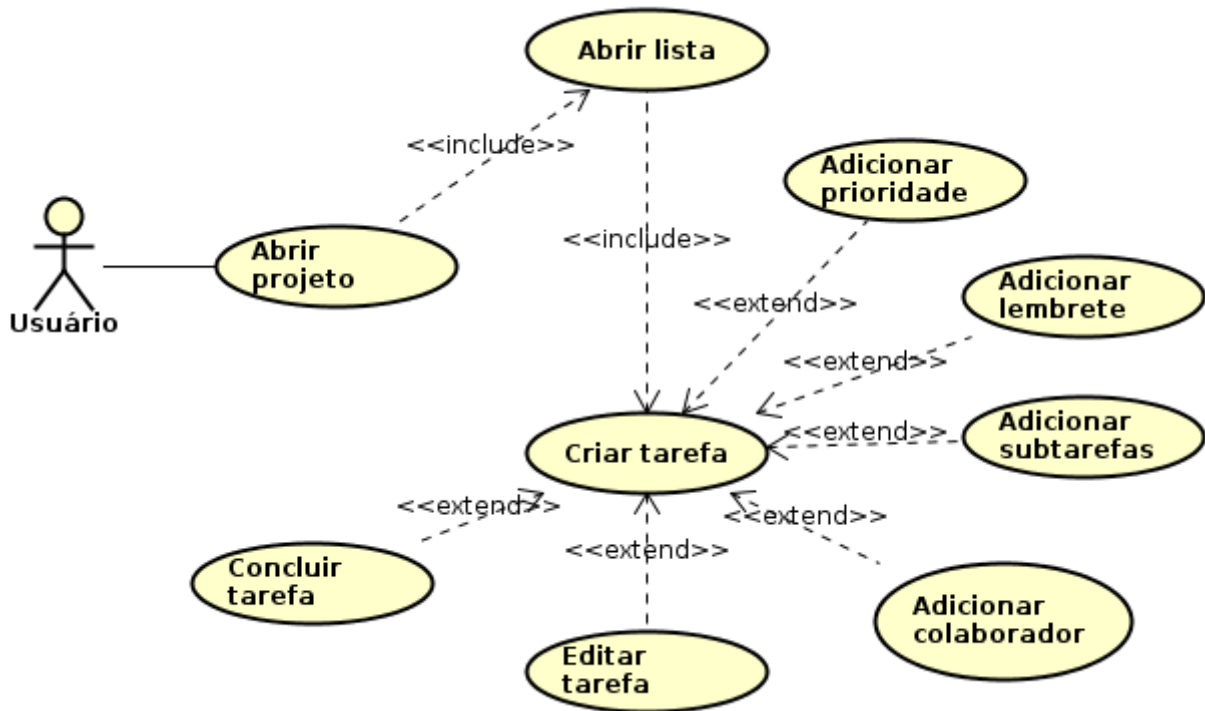
**Fonte:** Imagem criada pelo autor

#### 5.2.1.1 Criar tarefa

A Figura 8 mostra o diagrama de casos de uso para a criação de tarefas no aplicativo. Inicialmente, para criar tarefas é necessário que o usuário selecione um projeto, selecione a lista que ele deseja incluir as tarefas e criar a tarefa. Assim que o usuário cria a tarefa é possível acrescentar algumas propriedades, como: prioridade, lembrete, subtarefas e colaborador.



**Figura 8:** Diagrama de casos de uso para a criação de tarefas



Fonte: Imagem criada pelo autor

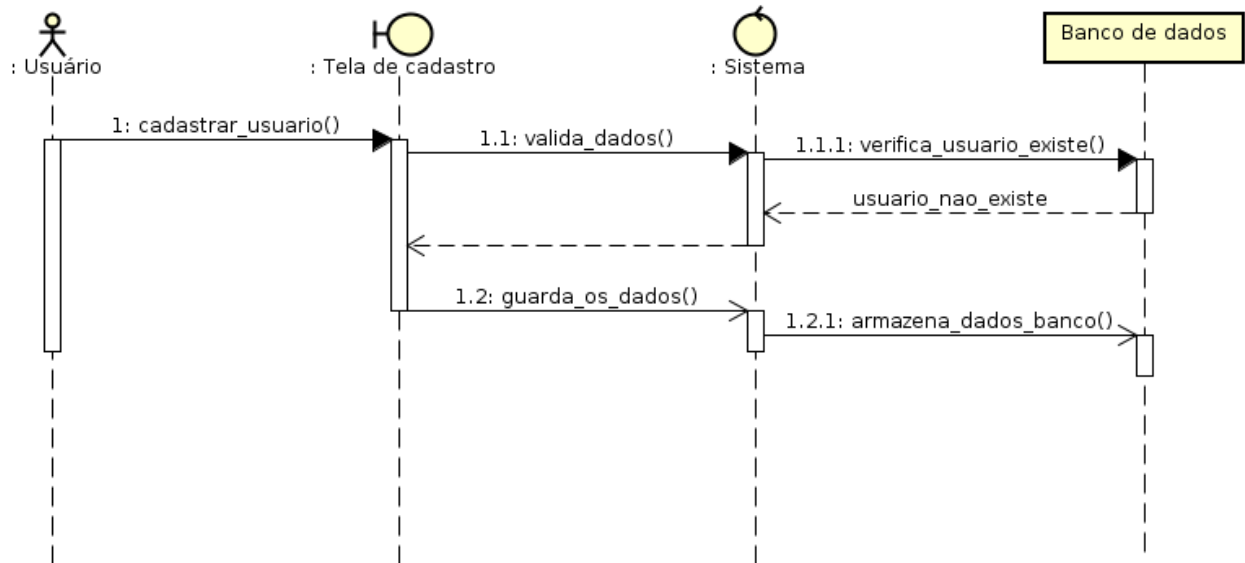
## 5.2.2 Diagrama de sequência

O diagrama de sequência exibe o comportamento do aplicativo de acordo com as ações que o usuário executa.

### 5.2.2.1 Cadastro

A Figura 9 mostra o diagrama de sequência para o cadastro do usuário no aplicativo. Na tela de cadastro, o usuário irá inserir os dados necessários para a criação da conta, assim que o usuário clicar no botão de cadastrar o sistema irá verificar se as informações estão corretas, caso estejam, a próxima ação será verificar no banco de dados se já existe um usuário com as mesmas informações. Caso não exista, o sistema guarda os dados e os repassa para o banco onde serão armazenados, concluindo assim o cadastro do usuário no sistema.

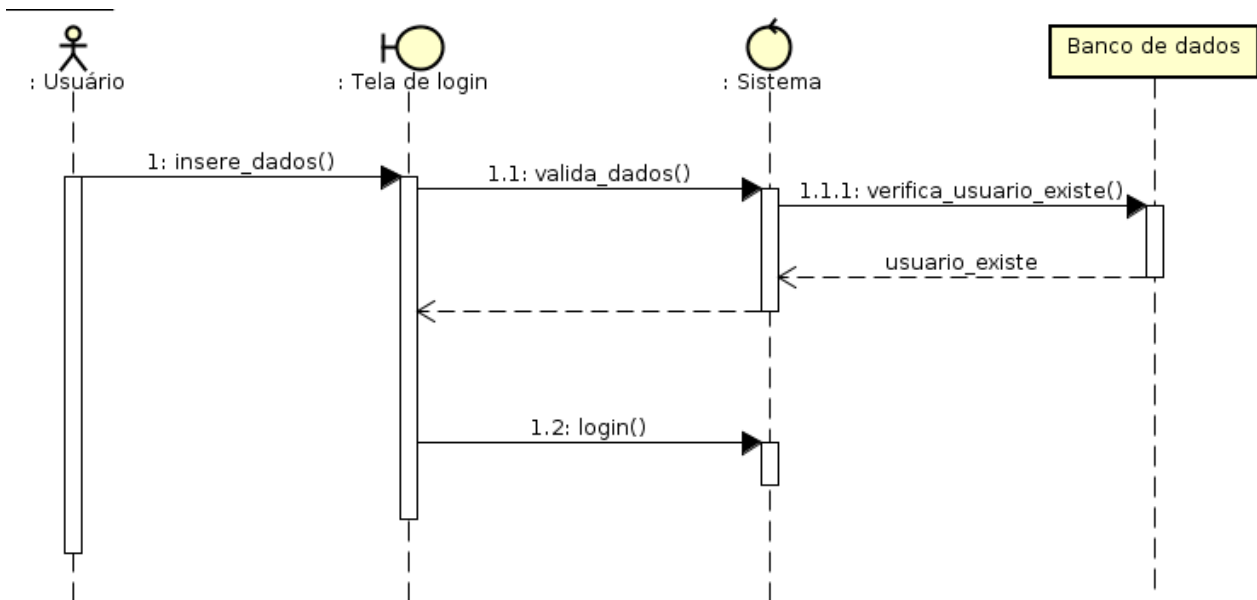
**Figura 9:** Diagrama de seqüência de cadastro



**Fonte:** Imagem criada pelo autor

#### 5.2.2.2 Login

O diagrama de seqüência de login é exibido na Figura 10. Inicialmente, na tela de login o usuário insere os dados de cadastro, o sistema valida os dados, caso os dados inseridos nos campos estejam de acordo com o que se pede, o banco de dados verifica se existe algum usuário no banco com aquelas informações, caso ele encontre no aplicativo, o *login* será realizado com sucesso.

**Figura 10:** Diagrama de sequência de login

**Fonte:** Imagem criada pelo autor

Esse diagrama de sequência possui o mesmo comportamento, caso o usuário crie uma nova lista, grupo, tarefa ou subtarefa.

### 5.3 MODELAGEM DO BANCO DE DADOS

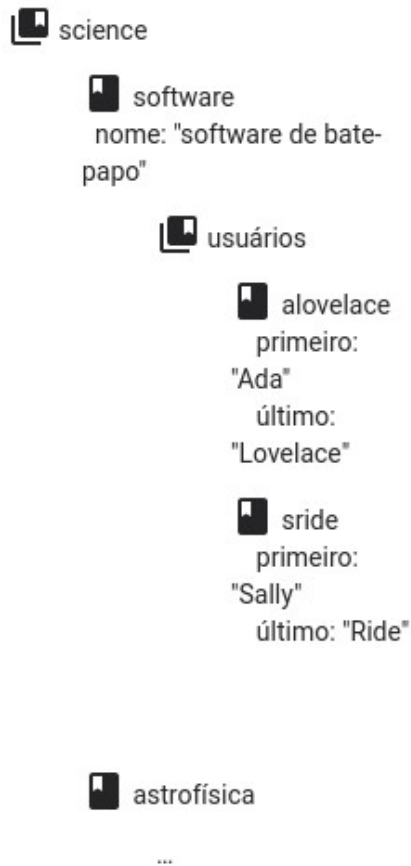
Por se tratar de um banco de dados NoSQL, a modelagem do banco foi feita de forma diferente de um banco relacional. Quando se trata de um banco relacional, em geral, a modelagem que se utiliza é a criação de diagramas que identifiquem as entidades do sistema e suas relações, onde em alguns casos essas relações poderão gerar chaves estrangeiras ou novas tabelas. Entretanto, o uso de um banco não-relacional diferencia a forma como essas entidades se relacionam no sistema.

A modelagem do Cloud Firestore (CLOUD FIRESTORE, 2018a) foi feita por meio de algumas estruturas recomendadas pelo Firebase (FIREBASE, 2018b), que oferece esse serviço. Na documentação disponível no site do Firebase é possível observar que são oferecidos três sugestões de estruturar os dados no banco Cloud Firestore, indicando quais as vantagens e desvantagens na utilização de cada uma.

Inicialmente, neste projeto utilizou-se a estrutura de subcoleções, onde há uma coleção (pai) que armazena várias subcoleções (filhos) aninhadas (FIREBASE, 2018b), como mostra a Figura 11.

Esta estrutura de subcoleções foi utilizada pois todos os dados relacionados a determinado usuário estariam vinculadas diretamente a ele.

**Figura 11:** Representação da estrutura de subcoleções utilizada inicialmente



**Fonte:** <https://firebase.google.com/docs/firestore/manage-data/structure-data> (2018)

Entretanto, essa estrutura se mostrou ineficiente em algumas funções que foram desenvolvidas para o aplicativo, como: a vinculação de outros usuários (colaboradores) em tarefas, bem como a exclusão desses dados nos quais colaboradores estavam vinculados, pois a exibição e a manipulação dessas tarefas para o usuário colaborador exigia dados do usuário administrador que criou aquela tarefa. Além disso, por ter muitas subcoleções aninhadas em uma mesma coleção, o acesso à essas informações por meio da consulta se tornava extensa.

Como consequência, o banco foi reestruturado e passou-se a utilizar a estrutura de coleções em nível raiz (FIREBASE, 2018b). Nesta estrutura, são criadas as principais coleções que compõem o sistema e em cada coleção poderá existir referência de documentos de coleções ou subcoleções na qual estas estão relacionadas, como mostra a Figura 12.

**Figura 12:** Representação da estrutura de coleções em nível raiz



**Fonte:** <https://firebase.google.com/docs/firestore/manage-data/structure-data> (2018)

Nessa estrutura, todas as coleções terão apenas um documento que corresponderá as informações do usuário. Com essa estrutura, foi possível gerar relações entre as coleções, facilitando a replicação e a manipulação desses dados.

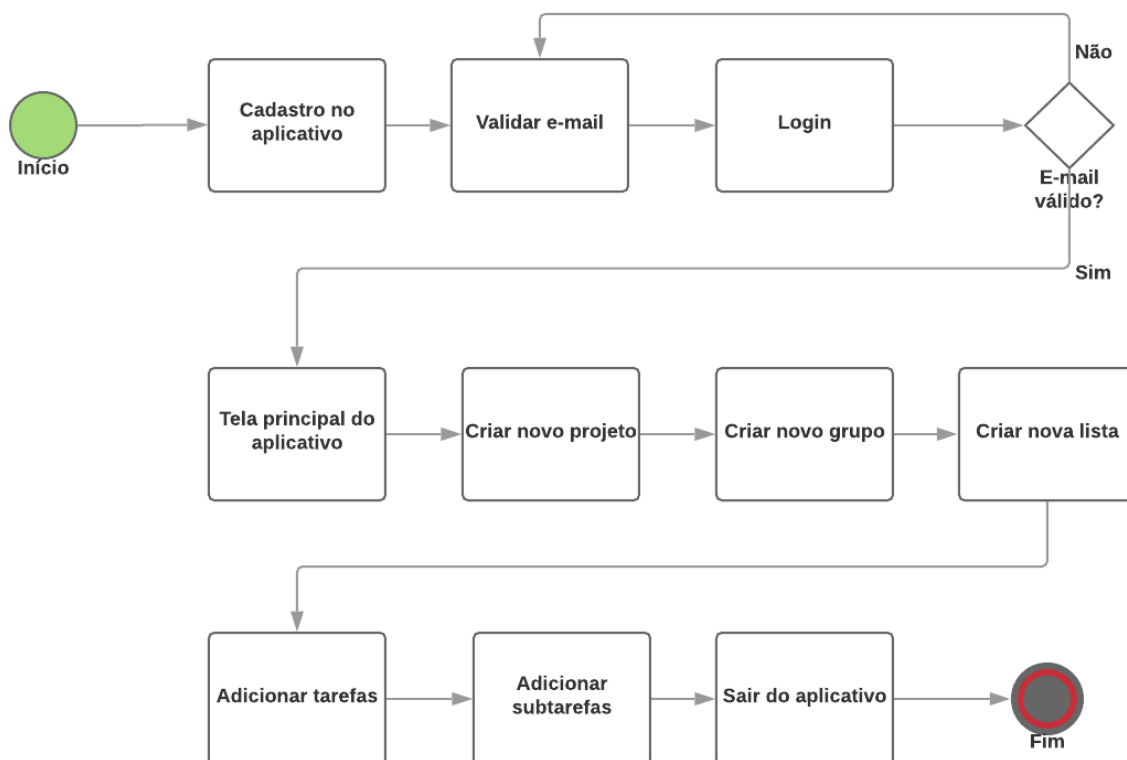
## 6 O APLICATIVO TUDOLIST

O TudoList é um aplicativo *mobile* para organização de tarefas em ambientes onde são desenvolvidos projetos de software. Os principais recursos que o aplicativo oferece ao usuário são a criação de tarefas e subtarefas com prioridade, lembrete utilizando o calendário do dispositivo e a adição de colaborador na tarefa.

### 6.1 DIAGRAMA DE BLOCO

O diagrama de bloco (Figura 13) representa o fluxo do aplicativo de uma maneira simples. Basicamente, após o login no aplicativo o usuário tem a possibilidade de criar projetos, grupo para o projeto, novas listas e a partir das listas criar novas tarefas, sendo possível a criação de subtarefas também.

**Figura 13:** Diagrama de bloco simples do aplicativo



**Fonte:** Imagem criada pelo autor

## 6.2 PRINCIPAIS TELAS DO APLICATIVO

A seguir, são apresentadas algumas das principais telas do aplicativo. As demais telas constam no Apêndice E.

### 6.2.1 Tela de cadastro e *login*

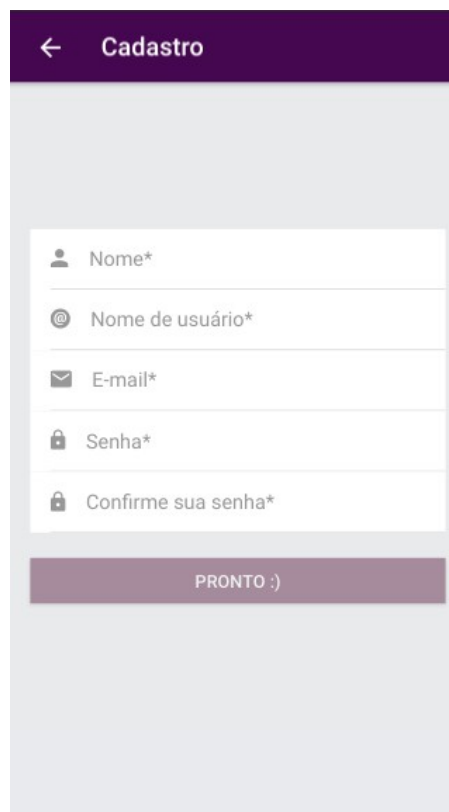
A tela inicial do aplicativo corresponde a tela de login, a partir dela o usuário poderá: Criar uma nova conta no aplicativo; fazer o login ou redefinir sua senha, no caso de perda. Acessando o aplicativo pela primeira vez, é necessário criar uma nova conta. O usuário poderá fazer isso clicando no link abaixo do botão de login, como mostra a Figura 14. Assim, o usuário será redirecionado para a página de cadastro. Onde será necessário preencher todas as informações: nome, nome de usuário, e-mail, senha e confirmação de senha, como mostra a Figura 15.

Por questões de segurança no fornecimento dos dados, o formulário de cadastro permanece com o botão de concluir desabilitado até que o usuário forneça corretamente os dados, nesse caso espera-se que o usuário forneça seu nome e um nome de usuário que possua mais de três caracteres, um e-mail válido e uma senha que possua pelo menos seis caracteres. Além disso, existem erros que são tratados após a conclusão do formulário, como: e-mail já possui cadastro no aplicativo; e-mail fornecido seja inválido; senha que o usuário forneceu seja fraca (possui caracteres fora do padrão ideal). Após o cadastro, o usuário irá receber um link para confirmação do e-mail.

**Figura 14:** Tela de login

A tela de login do aplicativo TudoList apresenta o logo 'TudoList' em uma fonte cursiva branca no topo. Abaixo, há dois campos de entrada: 'E-mail' com um ícone de envelope e 'Senha' com um ícone de cadeado. Um link azul 'Esqueceu sua senha?' está posicionado à direita do campo de senha. Um botão cinza 'ENTRAR' está centralizado abaixo dos campos. Na base da tela, há o texto 'Não possui conta? [Cadastre-se](#)'.

**Fonte:** Imagem criada pelo autor

**Figura 15:** Tela de cadastro

A tela de cadastro do aplicativo TudoList possui um cabeçalho escuro com um ícone de seta para trás e o título 'Cadastro'. O formulário contém cinco campos de entrada obrigatórios, cada um com um ícone e um asterisco: 'Nome\*' (ícone de pessoa), 'Nome de usuário\*' (ícone de @), 'E-mail\*' (ícone de envelope), 'Senha\*' (ícone de cadeado) e 'Confirme sua senha\*' (ícone de cadeado). Um botão cinza 'PRONTO :)' está localizado na base do formulário.

**Fonte:** Imagem criada pelo autor



Após o cadastro, o usuário será redirecionado para a página de login novamente com um aviso para a verificação do e-mail. A validação do e-mail é feita por meio de um link que é disponibilizado no e-mail cadastrado.

A validação do e-mail garante maior segurança no armazenamento dos dados, pois o usuário irá se autenticar no aplicativo com um e-mail válido, além de confirmar que o e-mail pertence a ele. Para a validação é necessário que o usuário acesse o link enviado e valide o e-mail.

## 6.2.2 Home

Ao entrar no aplicativo, o usuário irá ver a tela principal de acesso (Figura 16). Por meio desta tela o usuário poderá: acessar o menu (1), acessar páginas principais por meio das *tabs* (2), criar um novo projeto (3), pesquisar projetos (4) e entrar na página de um projeto (5).

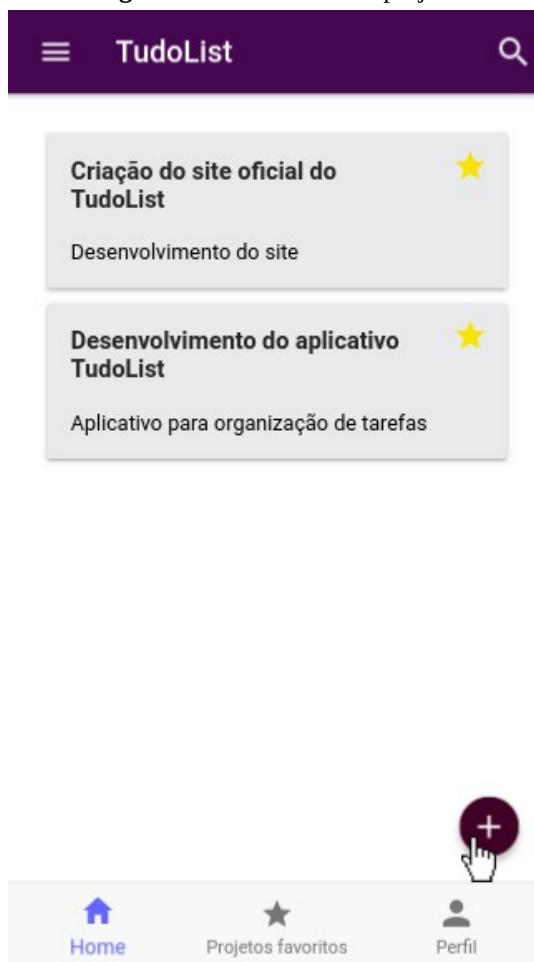


**Fonte:** Imagem criada pelo autor

### 6.2.2.1 Criando novos projetos

Para criar um novo projeto, é necessário que o usuário clique no botão de “+” que fica na parte inferior, como mostra a Figura 17. Após clicar no botão, o usuário será redirecionado para a tela com um formulário para criar um novo projeto (Figura 18). O formulário do projeto tem o campo nome e descrição, ambos devem ter mais de três caracteres para que o botão de criar projeto seja habilitado.

**Figura 17:** Botão de novo projeto



**Fonte:** Imagem criada pelo autor

**Figura 18:** Tela de novo projeto

← Novo projeto

Nome do projeto\*  
Site de vendas

Descrição\*  
Desenvolvimento de um site de vendas de

OK

Home    Projetos favoritos    Perfil

**Fonte:** Imagem criada pelo autor

#### 6.2.2.2 Acessar projetos

Para que o usuário acesse o conteúdo dos projetos, é necessário que ele clique no projeto desejado (Figura 19). Ao clicar, ele será redirecionado para a página do projeto (Figura 20). Nesta página, o usuário terá acesso as listas do projeto e ao grupo.

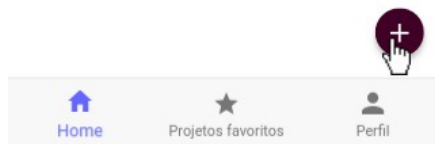
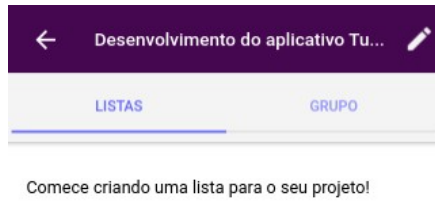
Figura 19: Acessar projeto



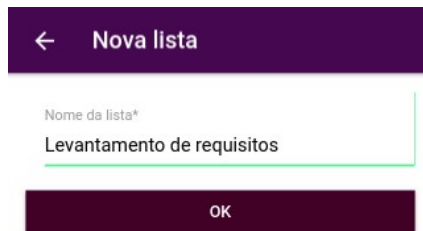
Fonte: Imagem criada pelo autor

### 6.2.3 Listas

Inicialmente, para criar as tarefas e subtarefas, é necessário anteriormente criar uma lista no projeto. O usuário poderá criar suas listas clicando no botão de “+” na parte inferior da tela (Figura 20), após clicar no botão ele será redirecionado para a página de nova lista (Figura 21).

**Figura 20:** Botão de criar nova lista

Fonte: Imagem criada pelo autor

**Figura 21:** Criar nova lista

Fonte: Imagem criada pelo autor

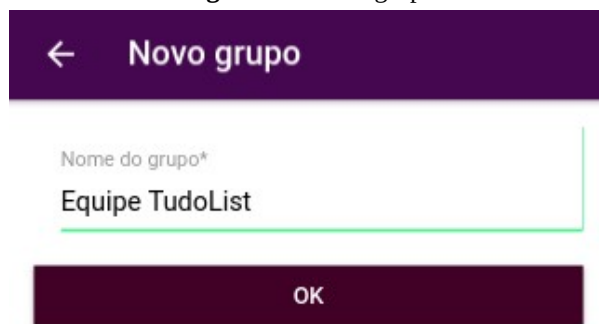
## 6.2.4 Grupo

Cada projeto possui um grupo associado, o usuário administrador poderá adicionar outros usuários a esse grupo para terem acesso ao projeto. Por questões de segurança, os membros do grupo só terão permissão de leitura no projeto. As figuras 22 e 23 mostram, respectivamente, a tela do grupo e a tela para criar um novo grupo.

**Figura 22:** Tela do grupo



**Fonte:** Imagem criada pelo autor

**Figura 23:** Novo grupo

← Novo grupo

Nome do grupo\*

Equipe TudoList

OK

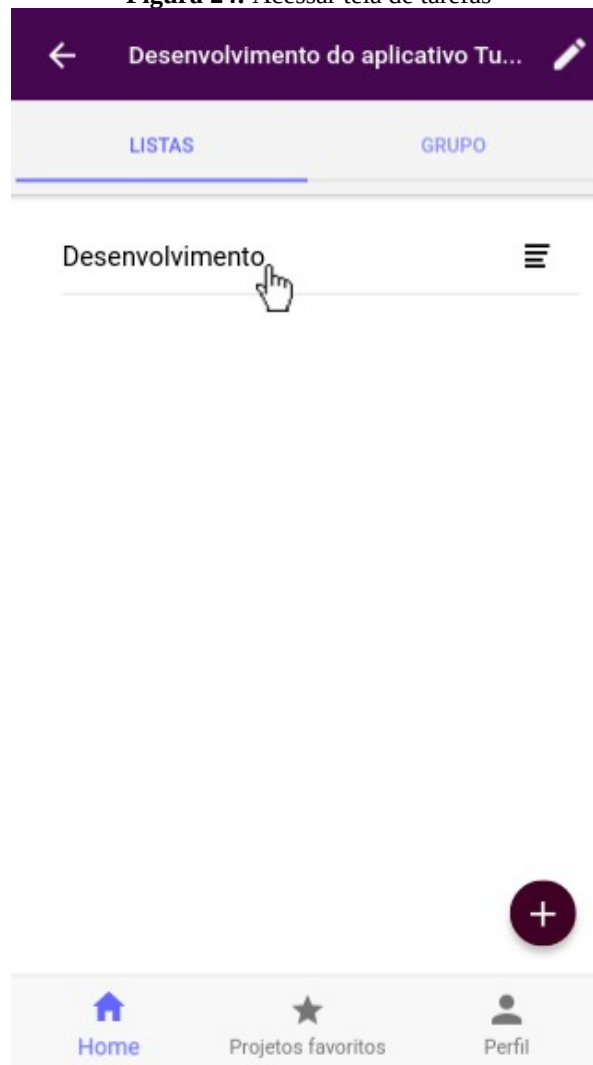


**Fonte:** Imagem criada pelo autor

### 6.2.5 Tarefas

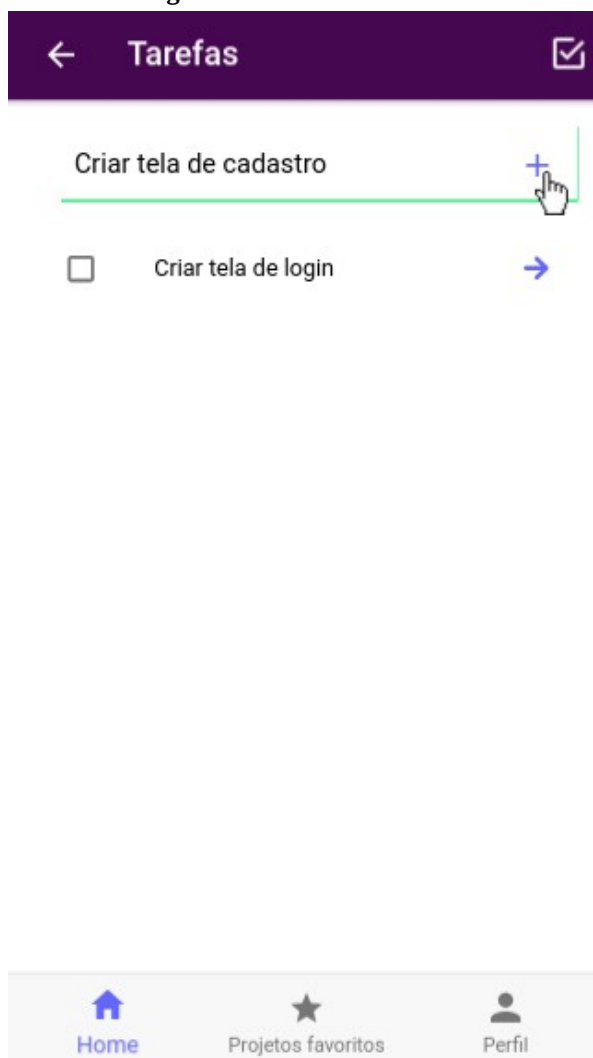
As tarefas são criadas dentro das listas do projeto. Para que o usuário adicione tarefas a uma determinada lista, é necessário clicar na lista desejada (Figura 24) e ele será redirecionado para a página de suas respectivas tarefas (Figura 25).

**Figura 24:** Acessar tela de tarefas



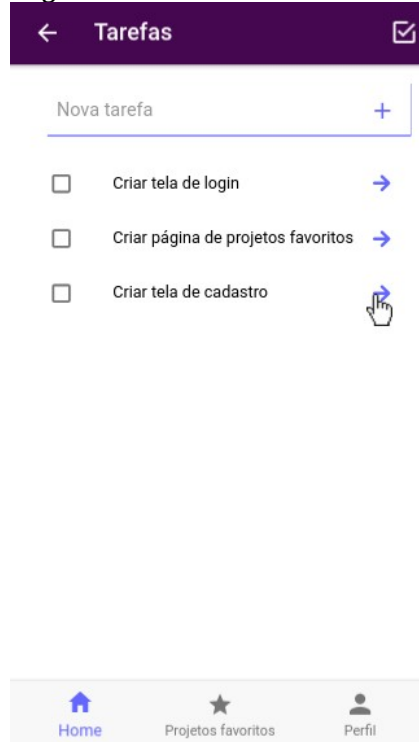
**Fonte:** Imagem criada pelo autor



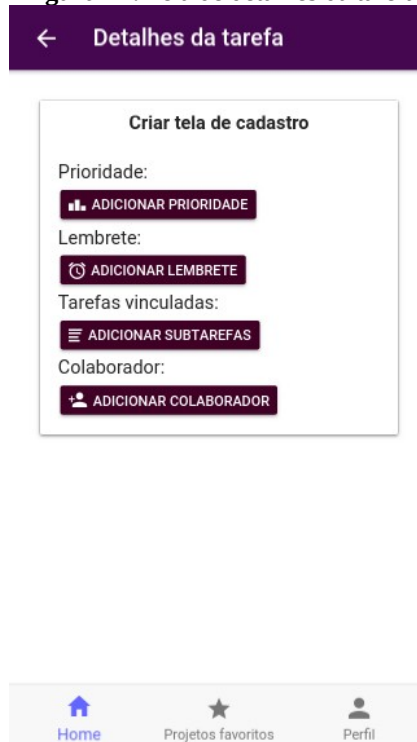
**Figura 25:** Criar novas tarefas

**Fonte:** Imagem criada pelo autor

Toda tarefa possui uma página de detalhes, o usuário pode acessar ela clicando na seta ao lado do nome da tarefa (Figura 26). A página de detalhes (Figura 27), contém todas as propriedades da tarefa. O usuário poderá: dar prioridade a tarefa, criar um lembrete, criar subtarefas e adicionar um colaborador.

**Figura 26:** Acessar detalhes da tarefa

**Fonte:** Imagem criada pelo autor

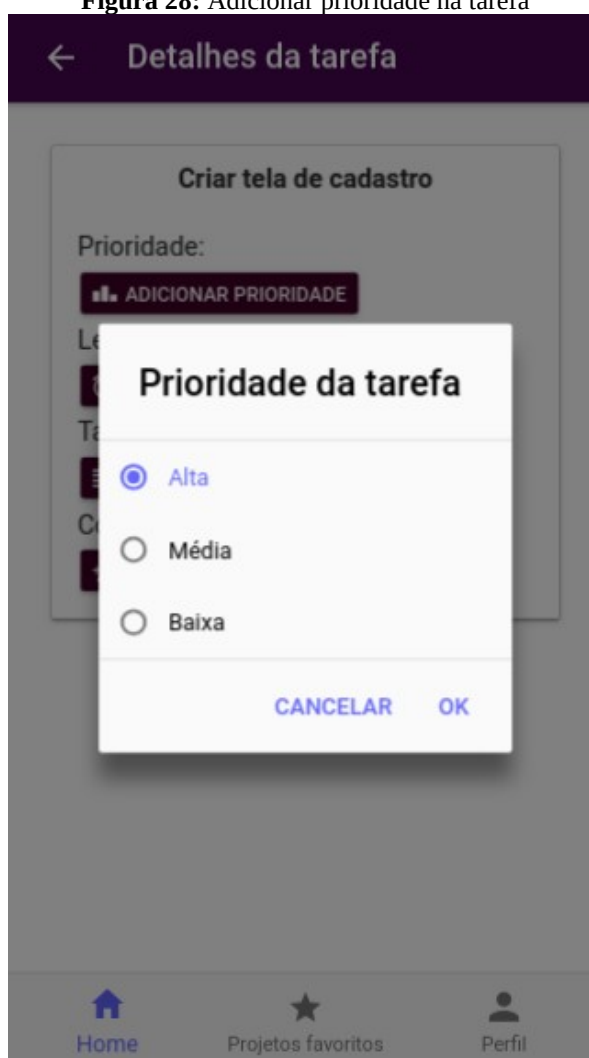
**Figura 27:** Tela de detalhes da tarefa

**Fonte:** Imagem criada pelo autor

### 6.2.5.1 Prioridade

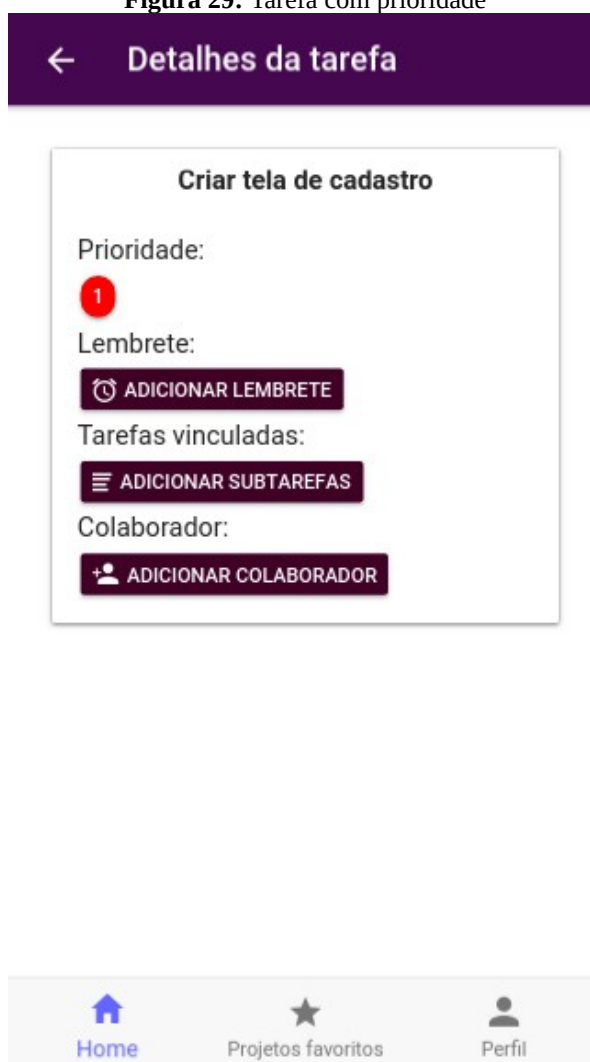
Uma das propriedades que podem ser definidas na tarefa é a prioridade. Ao clicar no botão de “adicionar prioridade” será aberta uma janela onde o usuário pode selecionar qual o nível da prioridade de acordo com o grau de importância da tarefa. O aplicativo oferece as opções de prioridade em três níveis: Alto, representa o maior nível (1) e utiliza a cor vermelha; médio, representa um nível razoável (2) de importância e utiliza a cor laranja, já o nível baixo possui o menor grau (3) e utiliza a cor amarela (Figura 28). Caso o usuário clique em “ok”, a prioridade escolhida aparecerá na página de detalhes com a cor e numeração definidas para o nível (Figura 29). Se o usuário desejar trocar a prioridade da tarefa, ele pode clicar na prioridade já definida que a janela para adicionar prioridade aparecerá novamente.

**Figura 28:** Adicionar prioridade na tarefa



**Fonte:** Imagem criada pelo autor

Figura 29: Tarefa com prioridade

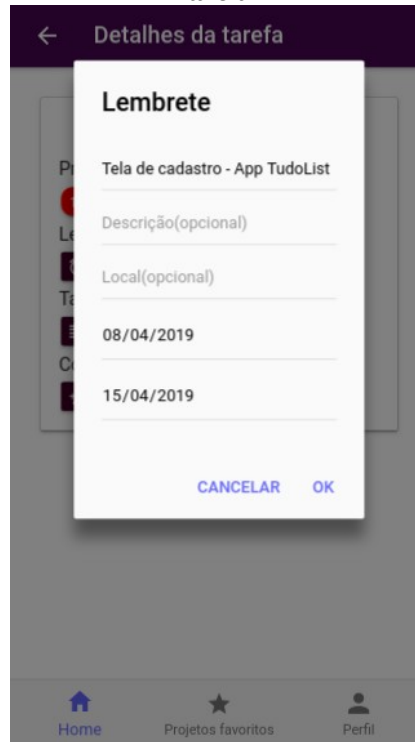


Fonte: Imagem criada pelo autor

#### 6.2.5.2 Lembrete

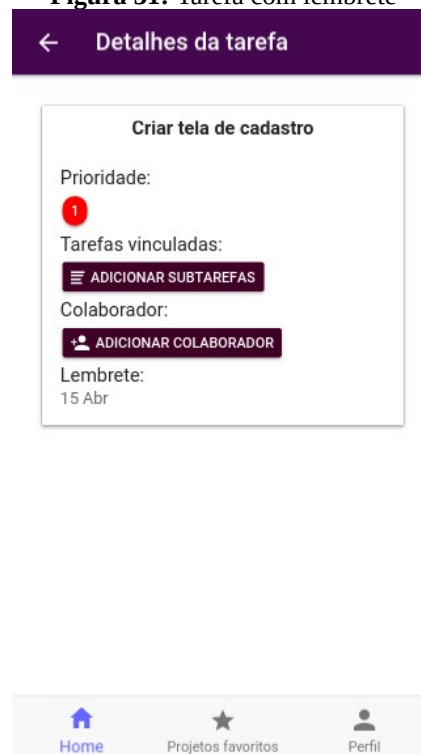
O lembrete tem como objetivo adicionar uma data de início e fim para a tarefa. Ao clicar no botão de “adicionar lembrete” será aberto uma janela com alguns campos correspondentes ao lembrete (Figura 30). Esses campos são: nome, o usuário precisa digitar o título do lembrete; local e descrição, esses dois campos são opcionais; data de início e fim da tarefa. Ao clicar em “Ok” o usuário será redirecionado para o calendário do dispositivo que estiver sendo utilizado, com isso será possível criar lembretes para que o dispositivo alarme quando a data de término da tarefa estiver próxima ou quando for necessário para o usuário. Após criar o lembrete, a data final ficará na página de detalhes da tarefa (Figura 31). Caso o usuário deseje excluir o lembrete, ele poderá fazer isso clicando em cima da data final, assim aparecerá uma janela para a exclusão do lembrete.

**Figura 30:** Adicionar lembrete na tarefa



**Fonte:** Imagem criada pelo autor

**Figura 31:** Tarefa com lembrete

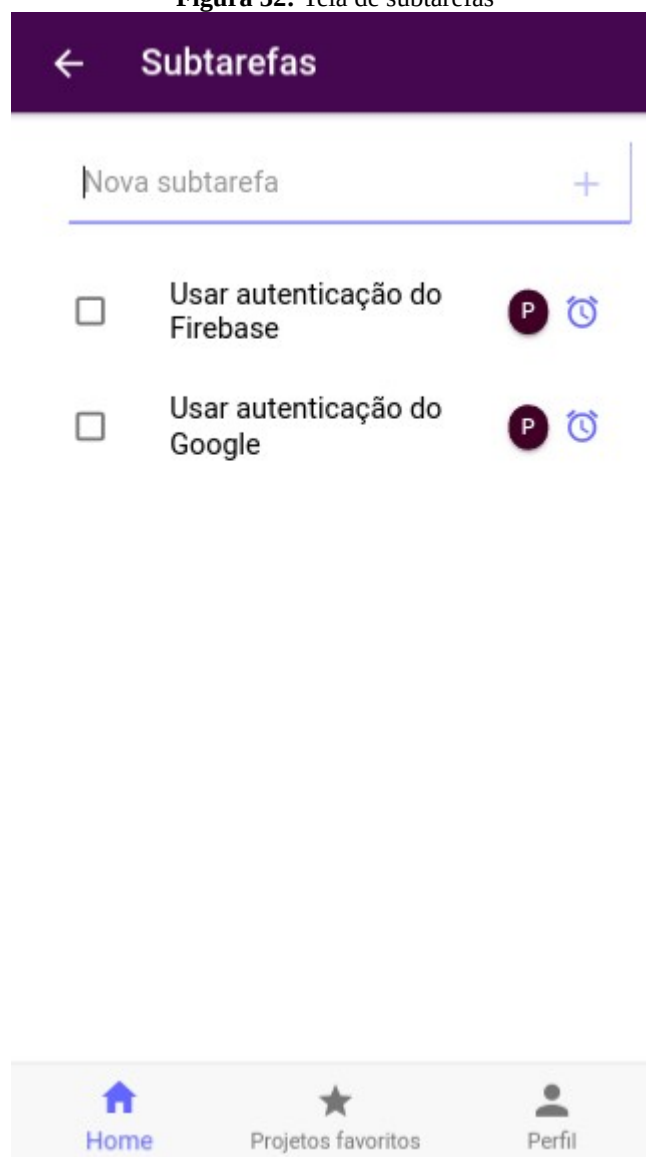


**Fonte:** Imagem criada pelo autor

### 6.2.5.3 Subtarefas

As subtarefas são pequenas tarefas relacionadas a tarefa principal. O usuário poderá criá-las clicando no botão de “adicionar subtarefas”. Ao clicar nesse botão, o usuário será redirecionado para a página de subtarefas (Figura 32). O usuário poderá criar novas subtarefas e assim como na tarefa principal, poderá criar lembrete e adicionar prioridade.

**Figura 32:** Tela de subtarefas

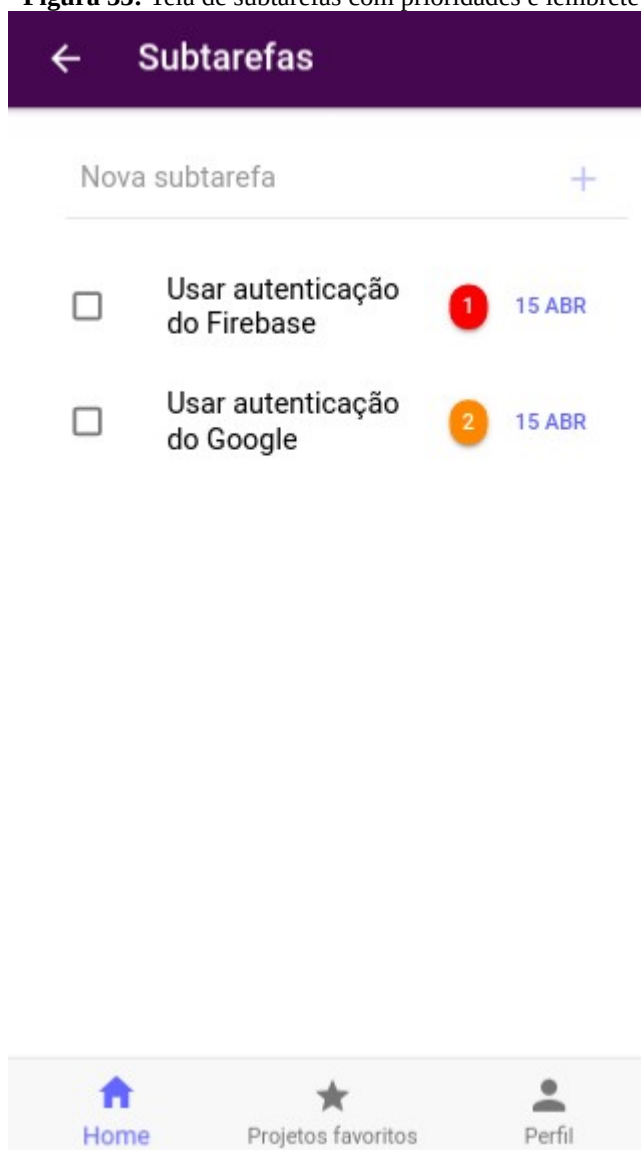


**Fonte:** Imagem criada pelo autor

Para adicionar prioridade o usuário precisa clicar no botão “P” que está localizado ao lado do nome, caso o usuário deseje adicionar um lembrete para a subtarefa ele pode clicar no botão ao

lado da prioridade, indicado por um ícone de relógio. As janelas de prioridade e lembrete são iguais as da tarefa principal, então para adicionar é só seguir o mesmo procedimento. Assim que o usuário adicionar, pode-se ver na tela as propriedades modificadas, como mostra a Figura 33. Caso o usuário deseje modificar a prioridade, basta clicar na prioridade já definida e alterar. Caso o usuário deseje excluir o lembrete, assim como na tarefa, é preciso clicar na data final e aparecerá uma janela para a exclusão.

**Figura 33:** Tela de subtarefas com prioridades e lembrete



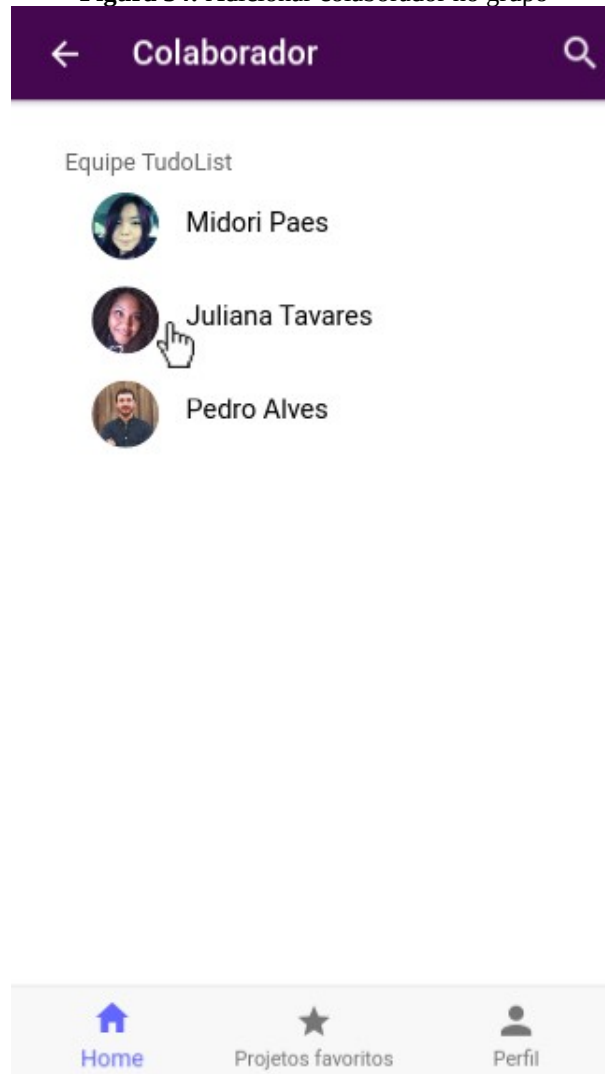
**Fonte:** Imagem criada pelo autor

#### 6.2.5.4 Colaborador

Toda tarefa poderá ter um colaborador. Quando o usuário se torna colaborador, automaticamente essa tarefa será vinculada a sua conta e ele saberá que possui uma tarefa pendente.

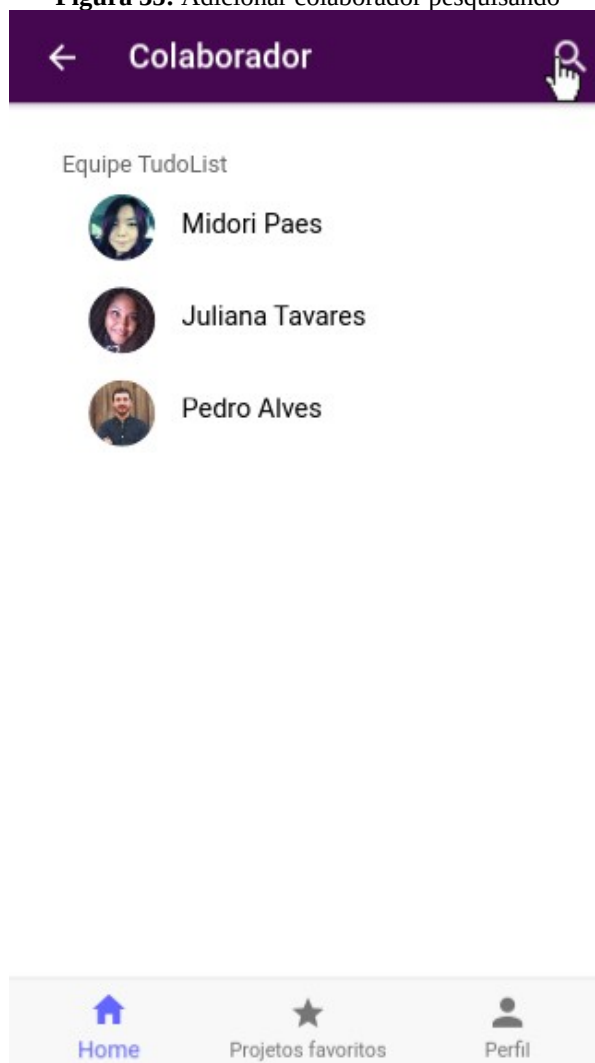
Inicialmente, para adicionar um colaborador a tarefa é necessário clicar no botão de “adicionar colaborador”. Após isso, o usuário é redirecionado para a página de adicionar colaborador. Para adicionar o usuário tem duas opções: selecionar algum usuário do grupo (Figura 34) ou clicar no botão de pesquisa e procurar o usuário colaborador (Figura 35).

**Figura 34:** Adicionar colaborador no grupo



**Fonte:** Imagem criada pelo autor

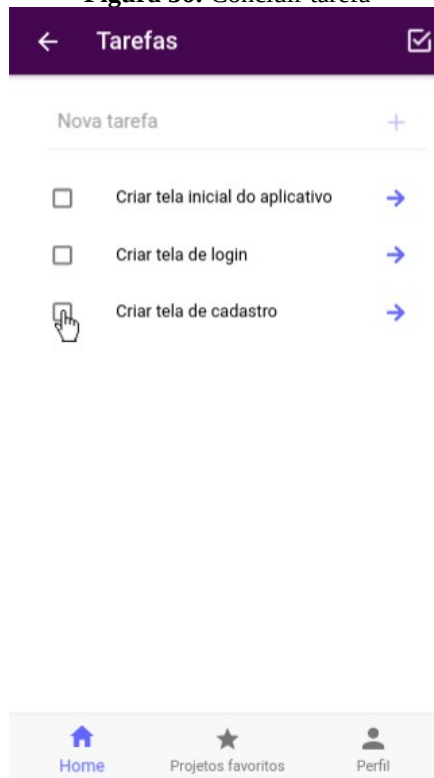


**Figura 35:** Adicionar colaborador pesquisando

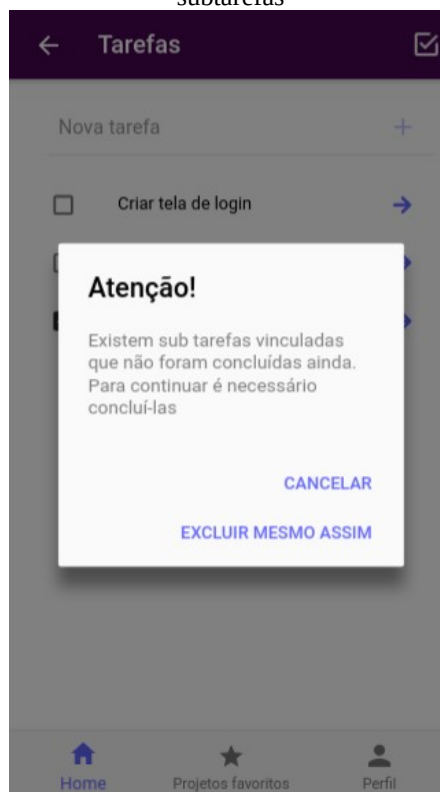
**Fonte:** Imagem criada pelo autor

#### 6.2.5.5 Concluir tarefa

Para concluir a tarefa, é só clicar no marcador da tarefa (Figura 36), caso a tarefa no momento da conclusão possua subtarefas vinculadas ainda não finalizadas, aparecerá um aviso antes (Figura 37). O usuário poderá optar em excluir mesmo com as subtarefas ainda existentes ou excluir as subtarefas antes de deletar a tarefa principal.

**Figura 36:** Concluir tarefa

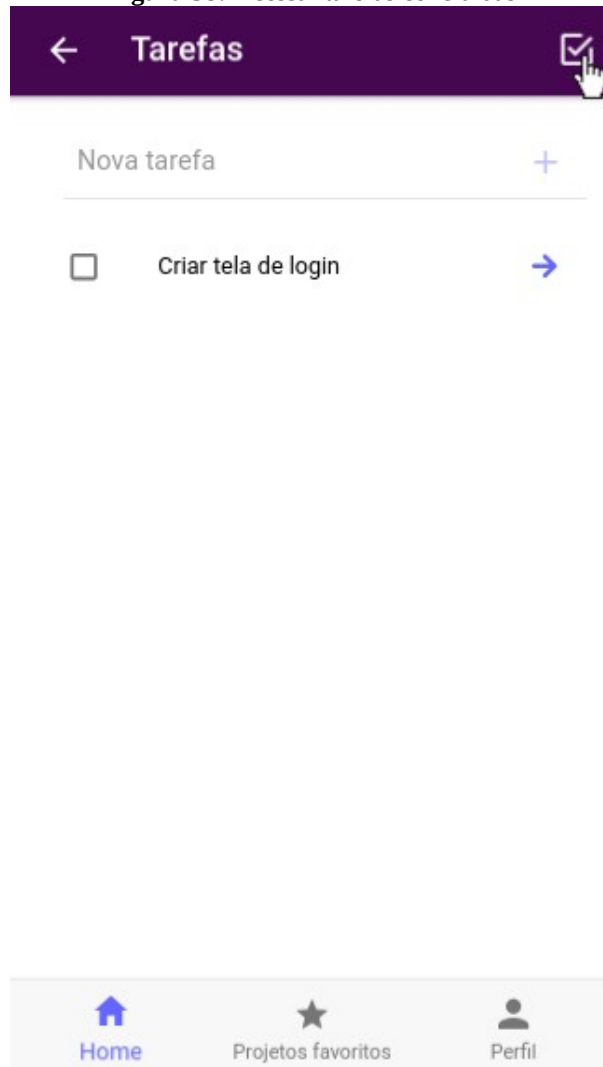
**Fonte:** Imagem criada pelo autor

**Figura 37:** Tela de aviso que contém subtarefas

**Fonte:** Imagem criada pelo autor

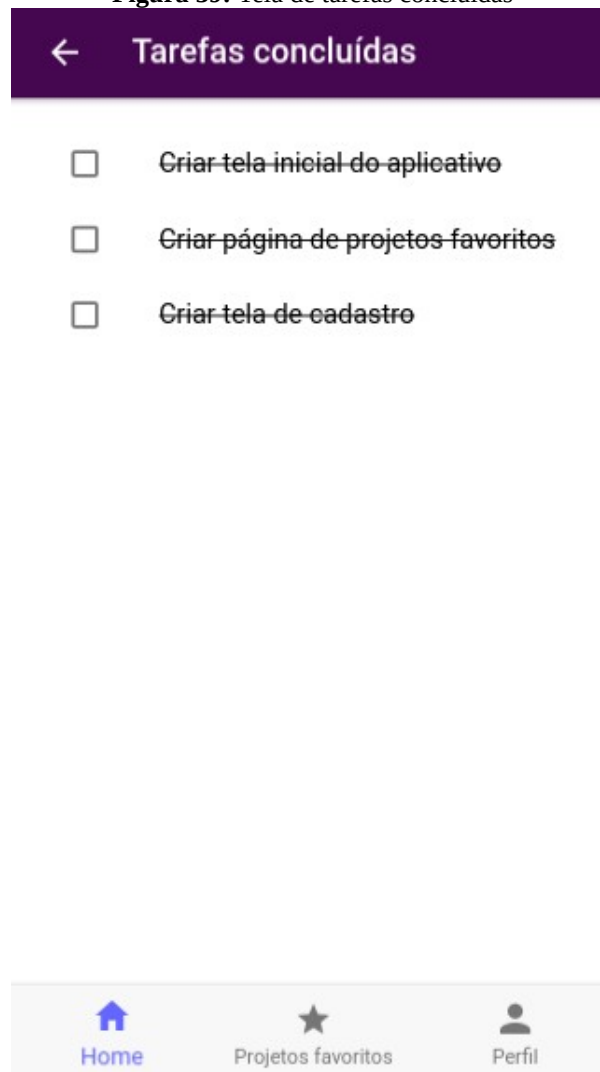
A conclusão das tarefas não exclui elas do banco, apenas modificam o estado, por isso o usuário poderá posteriormente visualizar as tarefas concluídas, tendo a possibilidade também de restaurá-las. Para acessar a página de tarefas concluídas, basta clicar no botão ao lado do título da página (Figura 38), assim a página de tarefas concluídas aparecerá (Figura 39). Caso o usuário deseje restaurar a tarefa, é só clicar em no marcador da tarefa desejada.

**Figura 38:** Acessar tarefas concluídas



**Fonte:** Imagem criada pelo autor

**Figura 39:** Tela de tarefas concluídas



**Fonte:** Imagem criada pelo autor

## 7 TRABALHOS FUTUROS

Apesar do aplicativo TudoList ter cumprido seu principal objetivo, que era fornecer uma ferramenta que facilitasse a organização de tarefas em projetos, existem alguns ajustes que pretende-se incluir futuramente.

Testes são importantes para validar o software, apesar do planejamento criado inicialmente, não foi possível realizar os testes neste trabalho. Espera-se que, futuramente sejam realizados testes para a validação do aplicativo.

Uma das funções do TudoList é possibilitar ao usuário a criação de lembretes nas tarefas e subtarefas por meio do calendário do dispositivo que está sendo utilizado. Futuramente, para facilitar o acesso à informação dos lembretes, propõe-se a utilização de um calendário dentro do aplicativo, exibindo o mês e as respectivas datas que possuem lembretes das tarefas e subtarefas.

Outra funcionalidade que o aplicativo disponibiliza é a criação de grupos. Cada projeto pode-se ter um grupo, sendo definido pelo administrador do projeto. Para futuramente facilitar a comunicação entre os usuários, propõe-se a criação de um ambiente de comunicação onde os usuários possam realizar troca de mensagens relacionadas ao projeto, a fim de facilitar a comunicação.

Para melhor notificar os usuários, propõe-se também a inclusão das Cloud Functions (FIREBASE 2019b). Que são funções criadas para a aplicação, assim a cada função executada dentro do banco, os usuários podem ser notificados automaticamente, facilitando assim o compartilhamento das informações.

O acesso às tarefas se dá por meio da página de listas, já as subtarefas pela página de detalhes da tarefa. Para que o acesso a essas informações seja mais rápido, propõe-se a criação de uma função para favoritar tarefas e subtarefas, assim o usuário terá acesso a elas de forma mais rápida.

Outra questão a ser ajustada é a criação de uma barra de progresso para as subtarefas. Assim, o usuário poderá visualizar a partir da tarefa o progresso das subtarefas até se chegar a conclusão.

Propõe-se a criação também de gráficos para exibirem os resultados e estatísticas dos projetos, a quantidade de tarefas feitas pelo usuário, quantas tarefas foram finalizadas no prazo e quais tarefas não foram finalizadas.

Outro fator importante é a adaptação do aplicativo para atender usuários que possuem alguma deficiência visual. Propõe-se o uso de ferramentas como o comando de voz utilizando o microfone do aparelho para o uso do aplicativo. Além disso, criar uma interface alternativa para pessoas que possuem baixa visão com cores que facilitem o uso do aplicativo e função que aumente a fonte.

Algumas outras funcionalidades que podem ser esperadas futuramente são o compartilhamento de listas de um projeto nas redes sociais e a possibilidade do usuário armazenar arquivos nas tarefas para o compartilhamento de informações.

## 8 CONSIDERAÇÕES FINAIS

O uso de ferramentas para a organização de tarefas é algo importante pois auxilia na produtividade, que está diretamente ligado ao processo de desenvolvimento dos projetos. O gerenciamento e a organização de projetos pode-se tornar complexo com o tempo, ações como o surgimento de novas tarefas, subtarefas e a delegação das mesmas pode acarretar em diversos problemas como a deficiência na comunicação entre a equipe e um mal compartilhamento de novas informações. Sem o uso de alguma ferramenta que auxilie a organização, esses erros podem se tornar frequentes, podendo atrasar o desenvolvimento do projeto.

Um fator importante a ser destacado é que o banco de dados NoSQL possui uma estrutura e conceito diferente do banco relacional. Por esse motivo, é possível que haja dificuldades no momento de se desenvolver um modelo de dados para o sistema. Além disso, como toda tecnologia, o NoSQL também possui desvantagens em alguns casos. Por isso, é importante avaliar se a tecnologia realmente serve para o projeto que está sendo criado.

Por fim, conclui-se o uso de algum tipo de ferramenta é importante para que os erros sejam diminuídos e como consequência, haja um aumento na produtividade do projeto. Ações como a criação de um grupo para o projeto e a delegação de tarefas facilita a organização e o compartilhamento de novas informações entre a equipe.

## REFERÊNCIAS

ALMEIDA, A. G. D. *et al.* **SOFTWARE TCC: TASK CREATOR & CONTROLLER**. 2013. 141 f. TCC (Graduação) - Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas, Universidade Federal do Paraná, Curitiba, 2013. Disponível em: <<https://acervodigital.ufpr.br/bitstream/handle/1884/40685/Task%20Creator%20%26%20Controller.pdf?sequence=1&isAllowed=y>>. Acesso em: 03 abr. 2019.

ALVES, G. F. O. (Ed.). **6 motivos para usar bancos de dados NoSQL**. 2015. Disponível em: <<https://dicasdeprogramacao.com.br/6-motivos-para-usar-bancos-de-dados-nosql/>>. Acesso em: 28 abr. 2019.

ANGULAR. **Architecture overview**. 2019. Disponível em: <<https://angular.io/guide/architecture>>. Acesso em: 29 mar. 2019.

AMAZON WEB SERVICES. **O que é NoSQL?** 2019. Disponível em: <<https://aws.amazon.com/pt/nosql/>>. Acesso em: 06 abr. 2019.

ASANA. Página oficial do aplicativo ASANA. 2019. Disponível em: <<https://asana.com/>> Acesso em: 26 abr. 2019

CUSTÓDIO, M. (Ed.). **9 ferramentas de gestão de projetos para usar na sua agência**. 2019. Disponível em: <<https://resultadosdigitais.com.br/agencias/ferramentas-de-gestao-de-projetos/>>. Acesso em: 14 abr. 2019.

FERREIRA, Lauren. **As diferenças entre SQL e NoSQL: MySQL x MongoDB**. 2018. Disponível em: <<https://medium.com/devtranslate/diferencas-entre-sql-e-nosql-51311f9069bd>>. Acesso em: 29 abr. 2019.

FIREBASE. **Cloud Firestore**. 2018a. Disponível em: <<https://firebase.google.com/docs/firestore/>>. Acesso em: 12 abr. 2019

FIREBASE. **Escolher uma estrutura de dados**. 2018b. Disponível em: <<https://firebase.google.com/docs/firestore/manage-data/structure-data>>. Acesso em: 02 abr. 2019.

FIREBASE. **Página oficial do Firebase**. 2019a. Disponível em: <<https://firebase.google.com/>>. Acesso em: 02 abr. 2019.

FIREBASE. **Cloud functions para Firebase**. 2019b. Disponível em: <<https://firebase.google.com/docs/functions>>. Acesso em: 10 maio 2019.

FIREBASE. **Cloud Storage**. 2019c. Disponível em: <<https://firebase.google.com/docs/storage>>. Acesso em: 12 maio 2019.



GIT. **Página oficial do Git.** 2019. Disponível em: <<https://git-scm.com/>>. Acesso em: 30 maio 2019.

IONIC. **Página oficial do Ionic.** 2019. Disponível em: <<https://ionicframework.com/>>. Acesso em: 29 mar. 2019.

LYNCH, M. (Ed.). **Introducing Ionic 4: Ionic for everyone.** 2019. Disponível em: <<https://blog.ionicframework.com/introducing-ionic-4-ionic-for-everyone/>>. Acesso em: 29 mar. 2019.

MELO, G. C. **Desenvolvimento de sistema web para gestão de tarefas em setores de TI – SGT.** 2017. 62 f. TCC (Graduação) - Curso de Análise e Desenvolvimento de Sistemas, Faculdade de Tecnologia de Gravataí, Porto Alegre - Rs, 2017. Disponível em: <<http://revista.faqi.edu.br/index.php/ADS/article/view/286/301>>. Acesso em: 08 maio 2019.

MICROSOFT. **Página oficial do Visual Studio Code.** 2019. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 30 maio 2019.

MUNIZ, F. (Ed.). **Vantagens de um Banco de Dados NoSQL, MongoDB.** 2018. Disponível em: <<https://blog.4linux.com.br/vantagens-de-um-banco-de-dados-nosql-mongodb/>>. Acesso em: 27 abr. 2019.

RADIGAN, D. **What is kanban?** Disponível em: <<https://www.atlassian.com/agile/kanban>>. Acesso em: 10 maio 2019.

RAMOS, M. V.; BRANCALHÃO, B. N.; PIRES, D. F.. OTA APP: UMA SOLUÇÃO PARA ORGANIZAR TAREFAS ACADÊMICAS. **Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica**, São José, SP, v. 9, n. 2, p.47-74, 2018. Disponível em: <<http://periodicos.unifacef.com.br/index.php/resiget/article/view/1624>>. Acesso em: 08 maio 2019.

RIBEIRO, G. F. (Ed.). **Brasil já é campeão mundial em uma coisa: uso de aplicativo no celular.** 2018. Disponível em: <<https://noticias.uol.com.br/tecnologia/noticias/redacao/2018/06/15/brasil-ja-e-campeao-mundial-em-uma-coisa-uso-de-aplicativo-em-celular.htm>>. Acesso em: 14 abr. 2019.

SCRUM. **What is Scrum?** 2019. Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>. Acesso em: 02 abr. 2019.

## APÊNDICE

### APÊNDICE A – Requisitos funcionais

**Tabela 5:** Requisito funcional (cadastrar usuário)

ID	RF001
<b>Função</b>	Cadastrar usuário
<b>Descrição</b>	Essa função tem como objetivo cadastrar os usuários no aplicativo
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ainda não ter nenhum cadastro no aplicativo
<b>Pós-condições</b>	Novo usuário cadastrado
<b>Fluxo principal</b>	<ol style="list-style-type: none"><li>1. Usuário insere os dados cadastrais;</li><li>2. O sistema faz uma consulta no banco para verificar se já existe usuário cadastrado com esses dados;</li><li>3. Se não existir usuário, os dados informados são armazenados no banco;</li><li>4. Usuário é cadastrado no sistema.</li></ol>
<b>Fluxo alternativo</b>	Nenhum

**Fonte:** Tabela criada pelo autor

**Tabela 6:** Requisito funcional (*login*)

ID	RF002
<b>Função</b>	Fazer <i>login</i>
<b>Descrição</b>	Essa função tem como objetivo realizar o <i>login</i> do usuário no aplicativo
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Possuir cadastro no aplicativo <b>[RF001]</b>
<b>Pós-condições</b>	Acesso ao aplicativo
<b>Fluxo principal</b>	<ol style="list-style-type: none"><li>1. Usuário insere suas credenciais para se logar;</li><li>2. O sistema faz uma consulta no banco para verificar se existe usuário cadastrado com esses dados;</li><li>3. Se usuário existir, o usuário entra no sistema;</li></ol>
<b>Fluxo alternativo</b>	Nenhum

**Fonte:** Tabela criada pelo autor

**Tabela 7:** Requisito funcional (criar lista)

<b>ID</b>	<b>RF004</b>
<b>Função</b>	Criar lista
<b>Descrição</b>	Essa função tem como objetivo a criação de lista dentro de um projeto
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o login <b>[RF002]</b> e ter criado um projeto <b>[RF003]</b>
<b>Pós-condições</b>	Nova lista criada
<b>Fluxo principal</b>	<ol style="list-style-type: none"><li>1. Usuário seleciona o projeto;</li><li>2. Insere o nome da lista;</li><li>3. O sistema envia a informação para o banco;</li><li>4. Nova lista é criada dentro do projeto.</li></ol>
<b>Fluxo alternativo</b>	<p><b>FA001 – Editar lista</b></p> <ol style="list-style-type: none"><li>1. O usuário clica no botão de editar;</li><li>2. Seleciona a lista;</li><li>3. Edita a lista;</li><li>4. O sistema envia as novas informações para o banco;</li></ol> <p><b>FA002 – Excluir lista</b></p> <ol style="list-style-type: none"><li>1. O usuário clica no botão de excluir lista;</li><li>2. O sistema verifica se o usuário tem permissão de administrador para excluir a lista;</li><li>3. Caso o usuário seja administrador, a lista é excluída;</li><li>4. O sistema envia as informações para o banco;</li></ol>

**Fonte:** Tabela criada pelo autor

**Tabela 8:** Requisito funcional (criar grupo)

ID	RF005
<b>Função</b>	Criar grupo
<b>Descrição</b>	Essa função tem como objetivo criar grupo para o projeto
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002] e ter criado um projeto [RF003]
<b>Pós-condições</b>	Novo grupo no projeto
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona o projeto;</li> <li>2. Clica no botão de novo grupo;</li> <li>3. Insere o nome do grupo;</li> <li>4. Pesquisa o usuário e o adiciona no grupo;</li> <li>5. O sistema envia as informações para o banco;</li> </ol>
<b>Fluxo alternativo</b>	<p><b>FA001 – Adicionar novo participante</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona o projeto;</li> <li>2. Abre página do grupo;</li> <li>3. Clica no botão de adicionar novo participante;</li> <li>4. Pesquisa o usuário e o adiciona no grupo;</li> <li>5. O sistema envia as novas informações para o banco;</li> </ol> <p><b>FA002 – Excluir participante</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona o projeto;</li> <li>2. Abre a página do grupo;</li> <li>3. Clica no botão de excluir participante;</li> <li>4. O sistema envia as novas informações para o banco.</li> </ol> <p><b>FA003 – Excluir grupo</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona o projeto;</li> <li>2. Abre página do grupo;</li> <li>3. Clica no botão de excluir grupo;</li> <li>4. O sistema envia as novas informações para o banco;</li> </ol>

**Fonte:** Tabela criada pelo autor

**Tabela 9:** Requisito funcional (prioridade na tarefa)

<b>ID</b>	<b>RF007</b>
<b>Função</b>	Adicionar prioridade na tarefa
<b>Descrição</b>	Essa função tem como objetivo adicionar prioridade na tarefa
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002] e ter criado uma tarefa [RF005]
<b>Pós-condições</b>	Tarefa com prioridade definida
<b>Fluxo principal</b>	<ol style="list-style-type: none"><li>1. Usuário seleciona a tarefa;</li><li>2. Escolhe o nível de prioridade para a tarefa;</li><li>3. O sistema envia a informação para o banco;</li><li>4. Nova prioridade é adicionada à tarefa.</li></ol>
<b>Fluxo alternativo</b>	<b>FA001 – Editar prioridade</b> <ol style="list-style-type: none"><li>1. Usuário seleciona a tarefa;</li><li>2. Clica na prioridade já existente para editar;</li><li>3. Seleciona nova prioridade para a tarefa;</li><li>4. O sistema envia a informação para o banco;</li><li>5. Nova prioridade é adicionada à tarefa.</li></ol>

**Fonte:** Tabela criada pelo autor

**Tabela 10:** Requisito funcional (lembrete na tarefa)

<b>ID</b>	<b>RF008</b>
<b>Função</b>	Adicionar lembrete
<b>Descrição</b>	Essa função tem como objetivo adicionar lembrete na tarefa
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002] e ter criado uma tarefa [RF005]
<b>Pós-condições</b>	Tarefa com lembrete definido
<b>Fluxo principal</b>	<ol style="list-style-type: none"><li>1. Usuário seleciona a tarefa;</li><li>2. Insere as informações para criar o lembrete no calendário;</li><li>3. O sistema redireciona o usuário para o calendário do dispositivo;</li><li>4. O usuário adiciona os lembretes;</li><li>5. O sistema envia a informação para o banco;</li><li>4. Novo lembrete é adicionado à tarefa.</li></ol>

<b>Fluxo alternativo</b>	<b>FA001 – Excluir lembrete</b>  1. Usuário seleciona a tarefa; 2. Seleciona o lembrete; 3. Exclui o lembrete; 4. Banco é atualizado.
--------------------------	--

**Fonte:** Tabela criada pelo autor

**Tabela 11:** Requisito funcional (adicionar subtarefa)

<b>ID</b>	<b>RF009</b>
<b>Função</b>	Adicionar subtarefa
<b>Descrição</b>	Essa função tem como objetivo adicionar subtarefa em uma tarefa
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002] e ter criado uma tarefa [RF005]
<b>Pós-condições</b>	Tarefa com nova subtarefa
<b>Fluxo principal</b>	1. Usuário seleciona a tarefa; 2. Cria nova subtarefa; 3. O sistema envia a informação para o banco; 4. Nova subtarefa é vinculada à tarefa selecionada.
<b>Fluxo alternativo</b>	<b>FA001 – Concluir subtarefa</b>  1. Usuário seleciona a tarefa; 2. Abre a lista de subtarefas; 3. Seleciona a subtarefa a ser concluída; 4. O sistema envia a informação para o banco;

**Fonte:** Tabela criada pelo autor

**Tabela 12:** Requisito funcional (prioridade na subtarefa)

<b>ID</b>	<b>RF010</b>
<b>Função</b>	Adicionar prioridade na subtarefa
<b>Descrição</b>	Essa função tem como objetivo adicionar prioridade na subtarefa
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002] e ter criado uma subtarefa [RF009]
<b>Pós-condições</b>	Subtarefa com prioridade definida

<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a tarefa;</li> <li>2. Abre a lista de subtarefas;</li> <li>3. Escolhe a subtarefa e define o nível de prioridade para a subtarefa;</li> <li>4. O sistema envia a informação para o banco;</li> </ol>
<b>Fluxo alternativo</b>	<p><b>FA001 – Editar prioridade</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona a tarefa;</li> <li>2. Abre a lista de subtarefas;</li> <li>3. Clica na prioridade já existente na subtarefa desejada;</li> <li>3. Seleciona nova prioridade para a subtarefa;</li> <li>4. O sistema envia a informação atualizada para o banco.</li> </ol>

**Fonte:** Tabela criada pelo autor

**Tabela 13:** Requisito funcional (lembrete na subtarefa)

<b>ID</b>	<b>RF011</b>
<b>Função</b>	Adicionar lembrete na subtarefa
<b>Descrição</b>	Essa função tem como objetivo adicionar lembrete na subtarefa
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002] e ter criado uma subtarefa [RF009]
<b>Pós-condições</b>	Subtarefa com lembrete definido
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a tarefa;</li> <li>2. Abre a lista de subtarefas;</li> <li>3. Usuário seleciona a subtarefa e insere as informações para criar o lembrete no calendário;</li> <li>4. O sistema redireciona o usuário para o calendário do dispositivo;</li> <li>5. O usuário adiciona os lembretes;</li> <li>6. O sistema envia a informação para o banco.</li> </ol>
<b>Fluxo alternativo</b>	<p><b>FA001 – Excluir lembrete</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona a tarefa;</li> <li>2. Abre a lista de subtarefas;</li> <li>2. Clica no lembrete já existente da subtarefa desejada para excluir;</li> <li>3. Exclui o lembrete;</li> <li>4. O lembrete é excluído do calendário do</li> </ol>

	dispositivo; 5. O sistema envia a informação atualizada para o banco.
--	--

**Fonte:** Tabela criada pelo autor

**Tabela 14:** Requisito funcional (colaborador na tarefa)

ID	RF012
<b>Função</b>	Adicionar colaborador
<b>Descrição</b>	Essa função tem como objetivo adicionar um colaborador na tarefa
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002] e ter criado uma tarefa [RF005]
<b>Pós-condições</b>	Tarefa com colaborador definido
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a tarefa;</li> <li>2. Escolhe o usuário que deseja inserir como colaborador;</li> <li>3. O sistema envia a informação para o banco;</li> <li>4. Novo colaborador é adicionado à tarefa.</li> </ol>
<b>Fluxo alternativo</b>	<p><b>FA001 – Excluir colaborador</b></p> <ol style="list-style-type: none"> <li>1. Usuário seleciona tarefa;</li> <li>2. Clica no colaborador;</li> <li>3. Exclui colaborador da tarefa;</li> <li>4. O sistema atualiza a informação no banco.</li> </ol>

**Fonte:** Tabela criada pelo autor

**Tabela 15:** Requisito funcional (editar conta)

ID	RF013
<b>Função</b>	Editar conta
<b>Descrição</b>	Essa função tem como objetivo editar as informações da conta do usuário
<b>Ator</b>	Usuário administrador
<b>Pré-condições</b>	Ter efetuado o <i>login</i> [RF002]
<b>Pós-condições</b>	Informações da conta editadas
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário entra na página de configurações do perfil;</li> <li>2. Clica em editar perfil;</li> <li>3. Insere as novas informações da conta;</li> </ol>



	4. O sistema atualiza a informação no banco.
<b>Fluxo alternativo</b>	Nenhum

**Fonte:** Tabela criada pelo autor

**Tabela 16:** Requisito funcional (excluir conta)

<b>ID</b>	<b>RF014</b>
<b>Função</b>	Excluir conta
<b>Descrição</b>	Essa função tem como objetivo excluir a conta do usuário no aplicativo
<b>Ator</b>	Usuário administrador
<b>Pré-condições</b>	Ter efetuado o <i>login</i> <b>[RF002]</b>
<b>Pós-condições</b>	Exclusão da conta no aplicativo
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário entra na página de configurações do perfil;</li> <li>2. Clica em excluir perfil;</li> <li>3. O sistema verifica se o usuário tem certeza da exclusão;</li> <li>4. Se sim, o sistema exclui o perfil;</li> <li>5. O sistema atualiza a informação no banco</li> </ol>
<b>Fluxo alternativo</b>	Nenhum

**Fonte:** Tabela criada pelo autor

## APÊNDICE B – Requisitos não-funcionais

**Tabela 17:** Requisito não-funcional (acesso)

<b>ID</b>	<b>RNF002</b>
<b>Função</b>	Acesso autorizado
<b>Descrição</b>	Essa função tem como objetivo garantir que o acesso do usuário ao aplicativo é autorizado, ou seja, que ele possui conta para acessar as funcionalidades.
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Possuir cadastro no sistema <b>[RF001]</b>
<b>Pós-condições</b>	Acesso ao aplicativo
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário tenta se logar no aplicativo;</li> <li>2. Sistema verifica se o usuário tem conta no aplicativo;</li> <li>3. Em caso afirmativo, o usuário tem o acesso ao aplicativo.</li> </ol>
<b>Fluxo alternativo</b>	Nenhum

**Fonte:** Tabela criada pelo autor

**Tabela 18:** Requisito não-funcional (usuário administrador)

<b>ID</b>	<b>RNF003</b>
<b>Função</b>	Usuário administrador
<b>Descrição</b>	Essa função tem como objetivo garantir que a exclusão de projetos, listas e grupos, sejam feitas pelo usuário administrador.
<b>Ator</b>	Usuário administrador
<b>Pré-condições</b>	Possuir cadastro no sistema <b>[RF001]</b>
<b>Pós-condições</b>	Exclusão dos dados
<b>Fluxo principal</b>	1. Usuário tenta excluir algum dado; 2. Sistema verifica se o usuário tem a permissão para exclusão; 3. Se sim, os dados são removidos.
<b>Fluxo alternativo</b>	Nenhum

**Fonte:** Tabela criada pelo autor

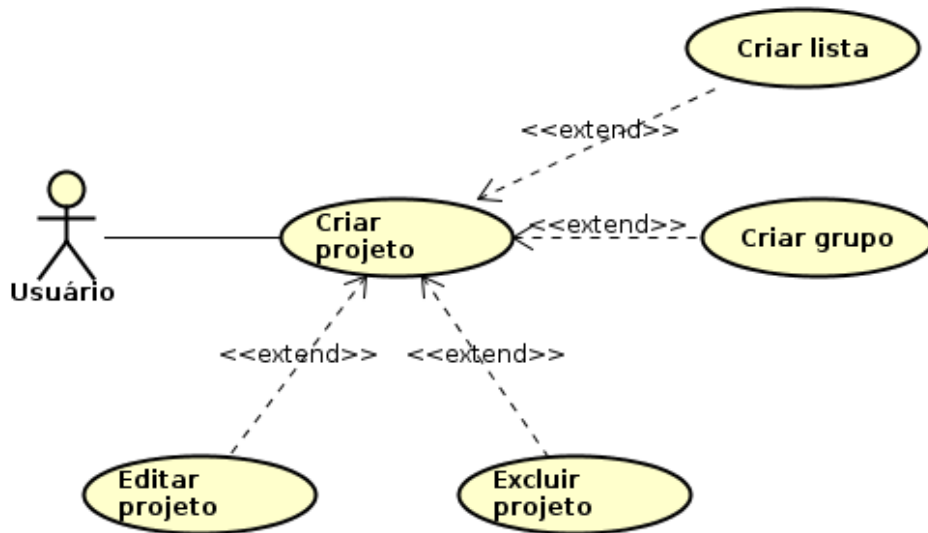
**Tabela 19:** Requisito não-funcional (software)

<b>ID</b>	<b>RNF004</b>
<b>Função</b>	Software
<b>Descrição</b>	O aplicativo deverá ser instalado em dispositivos com sistemas operacionais Android e iOS
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Possuir um dispositivo com algum dos sistemas operacionais citados
<b>Pós-condições</b>	Aplicativo instalado
<b>Fluxo principal</b>	Nenhum
<b>Fluxo alternativo</b>	Nenhum

**Fonte:** Tabela criada pelo autor

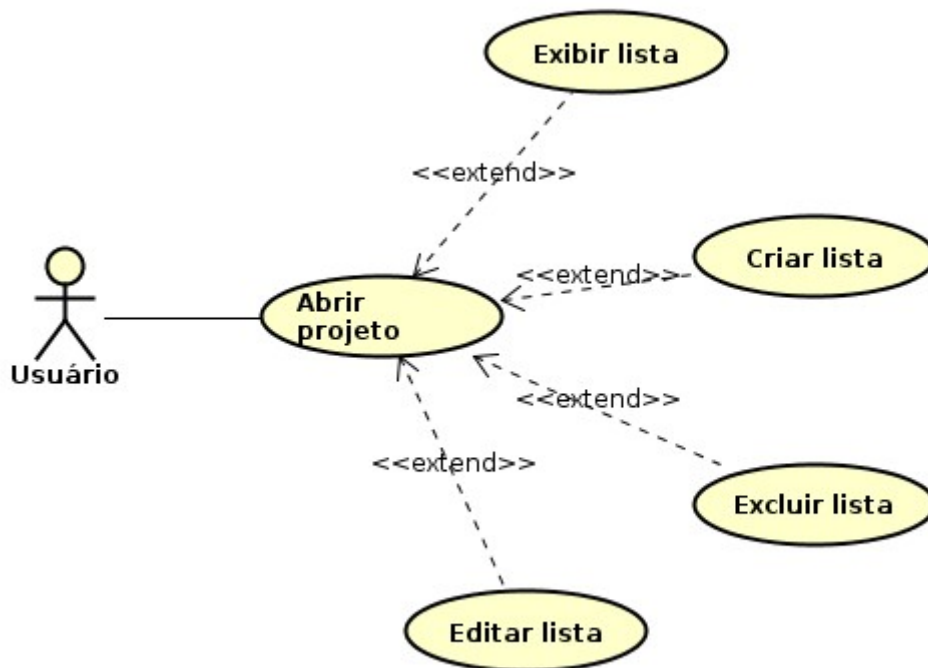
APÊNDICE C – Diagramas de casos de uso

Figura 40: Diagrama de casos de uso do projeto



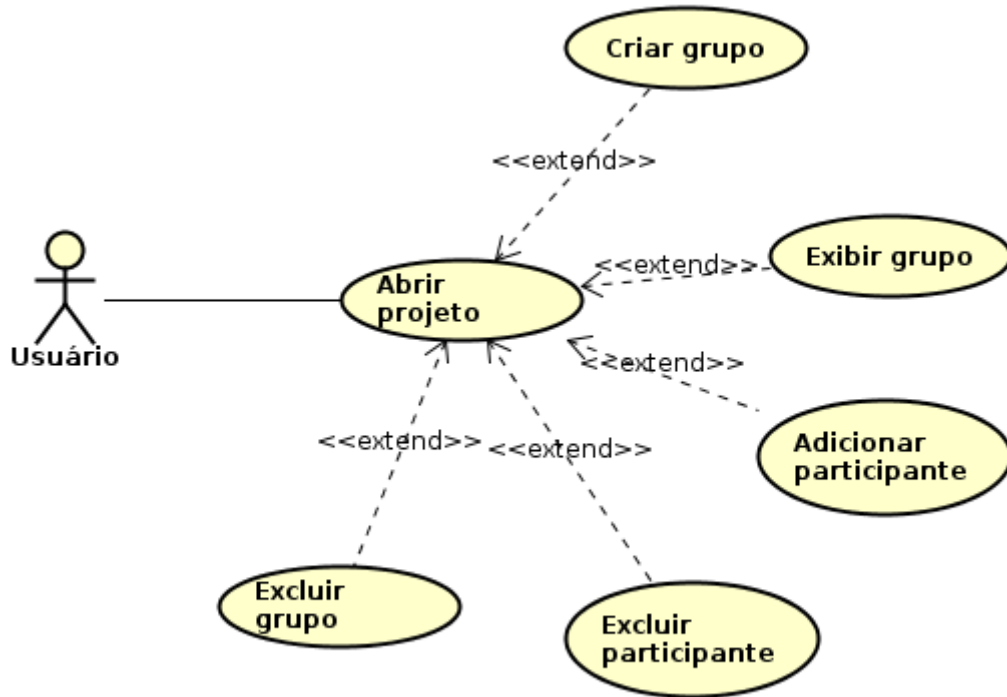
Fonte: Imagem criada pelo autor

Figura 41: Diagrama de casos de uso de lista



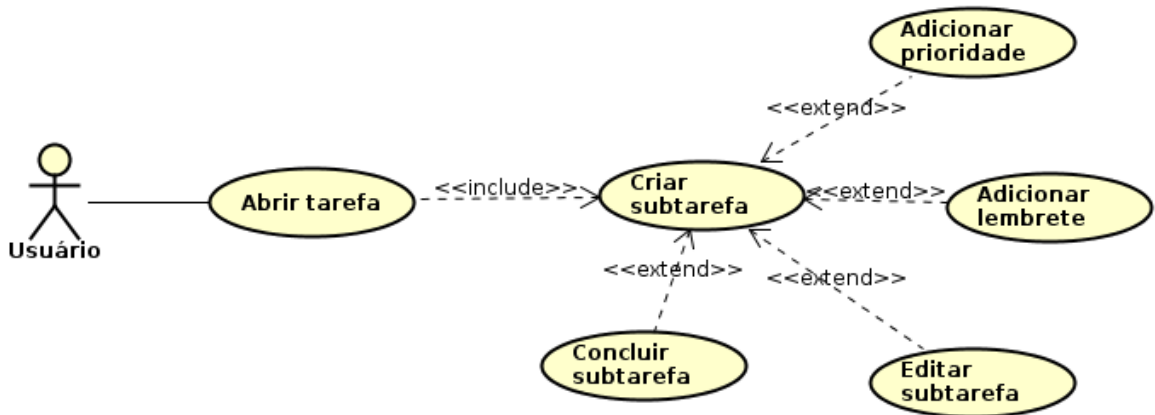
Fonte: Imagem criada pelo autor

Figura 42: Diagrama de casos de uso de grupo



Fonte: Imagem criada pelo autor

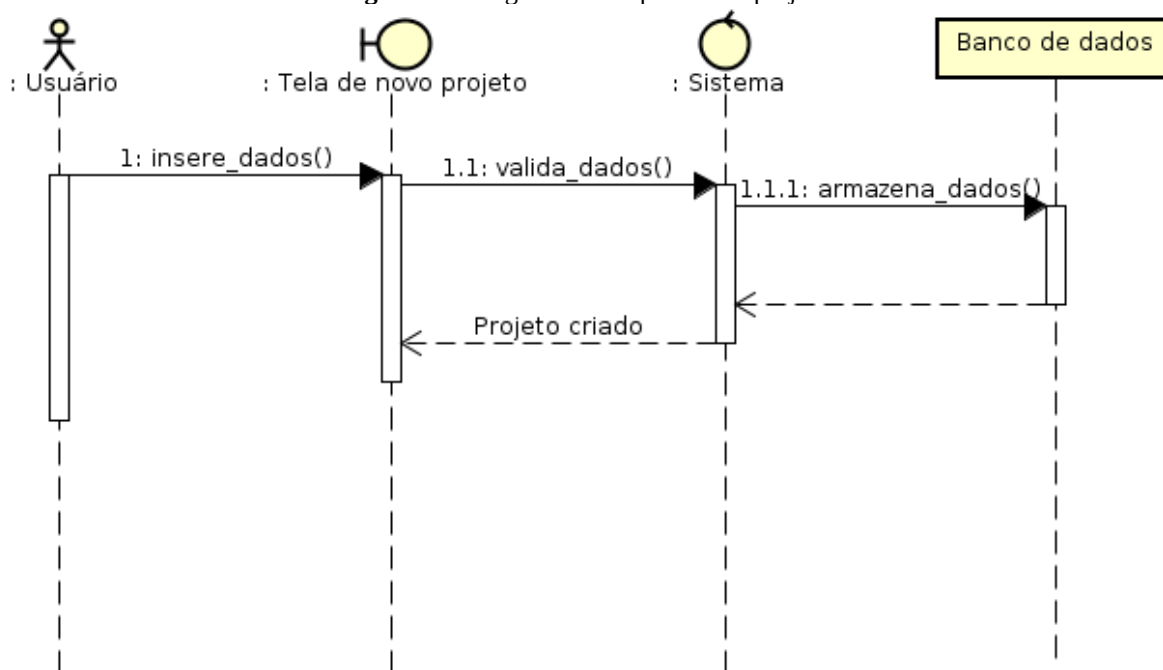
Figura 43: Diagrama de casos de uso para a criação de subtarefas



Fonte: Imagem criada pelo autor

APÊNDICE D – Diagramas de sequência

Figura 44: Diagrama de sequência de projetos



Fonte: Imagem criada pelo autor

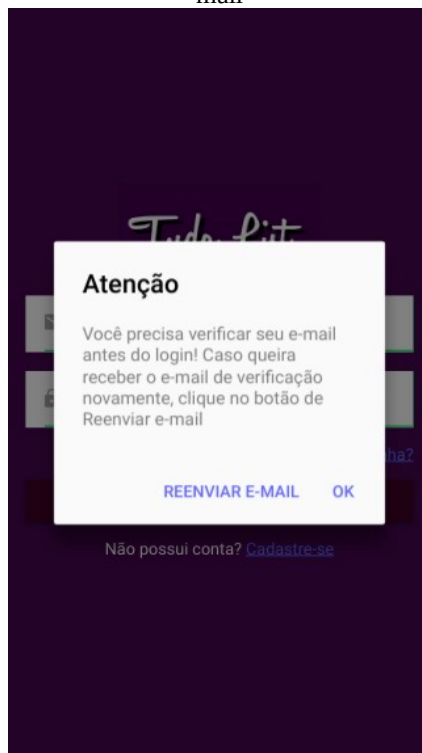
## APÊNDICE E – Principais telas do aplicativo

**Figura 45:** Erro e-mail ou senha incorretos



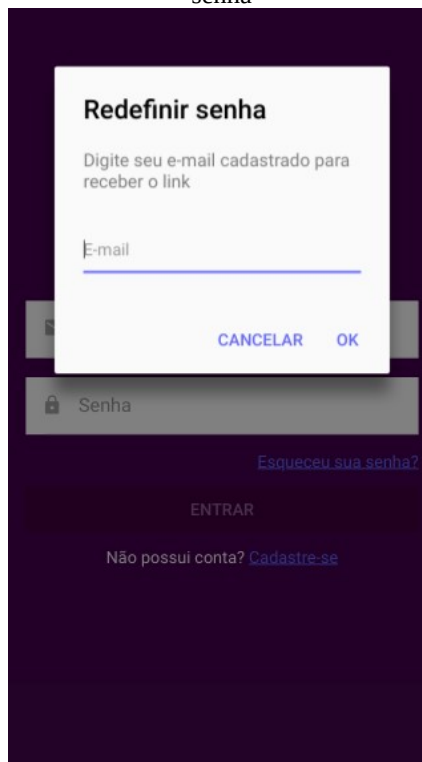
**Fonte:** Imagem criada pelo autor

**Figura 46:** Aviso de verificação do e-mail



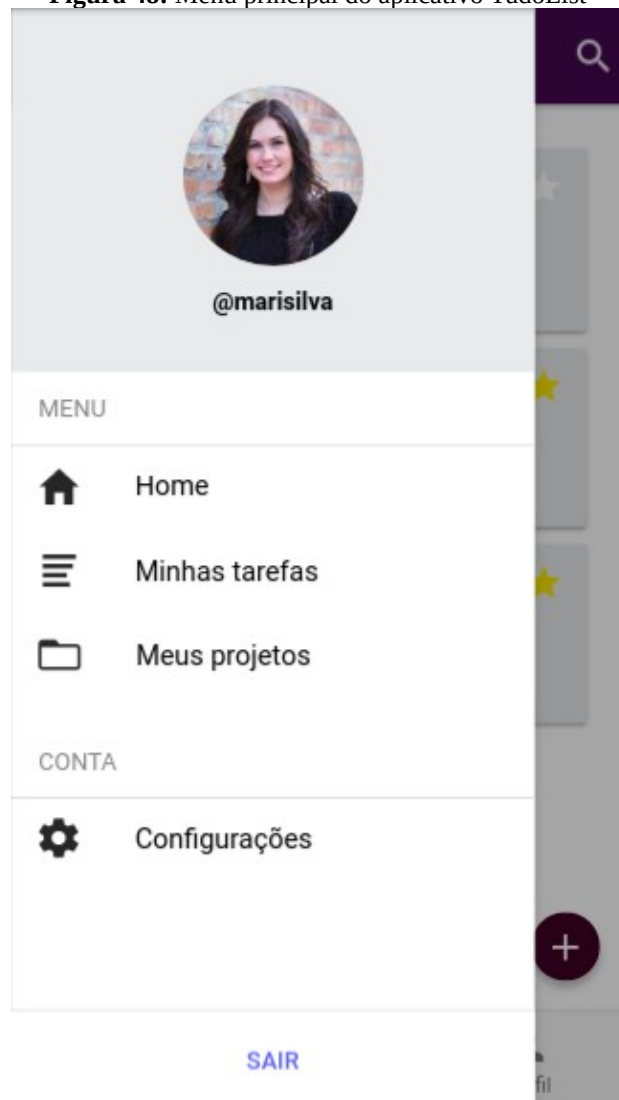
**Fonte:** Imagem criada pelo autor

**Figura 47:** Janela de redefinição de senha



**Fonte:** Imagem criada pelo autor

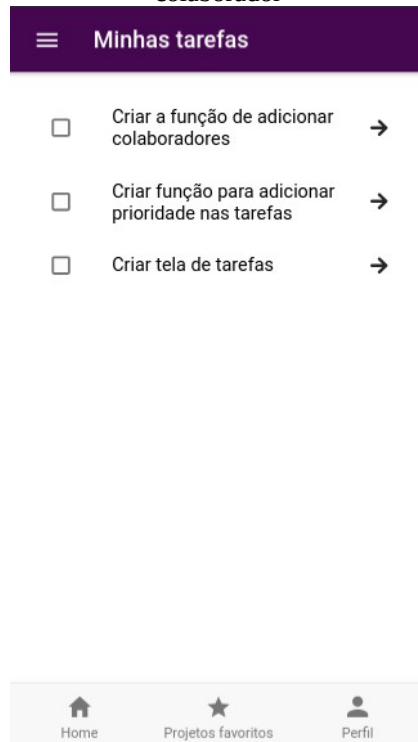
**Figura 48:** Menu principal do aplicativo TudoList



**Fonte:** Imagem criada pelo autor



**Figura 49:** Tarefas em que o usuário é colaborador



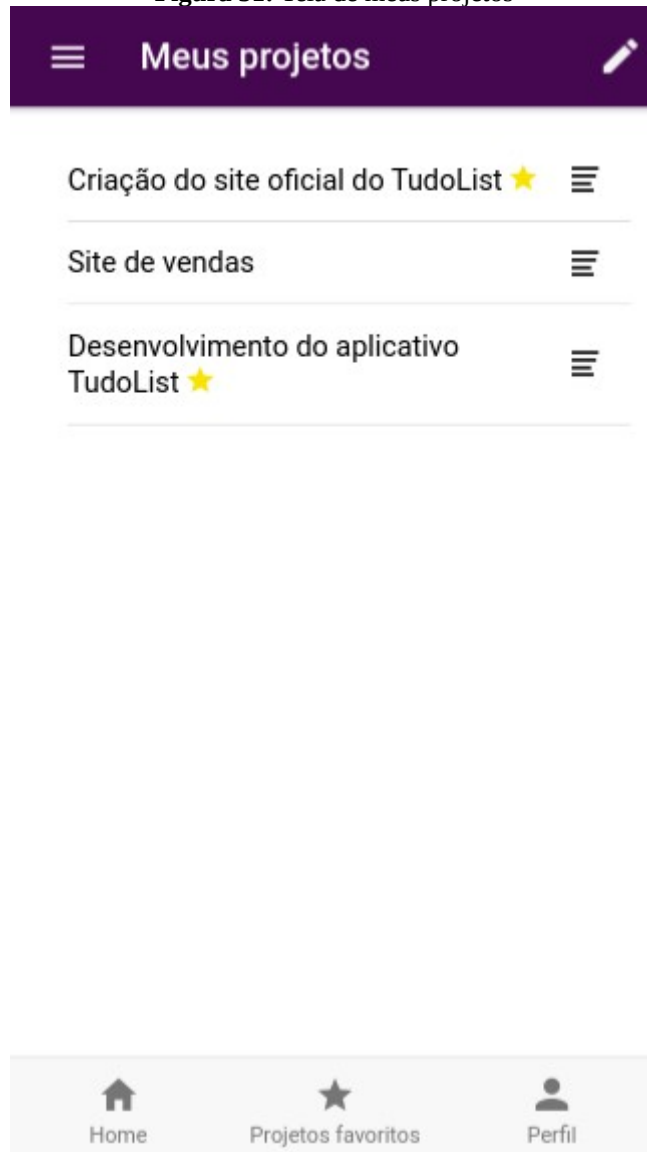
**Fonte:** Imagem criada pelo autor

**Figura 50:** detalhes das tarefas do colaborador



**Fonte:** Imagem criada pelo autor

**Figura 51:** Tela de meus projetos



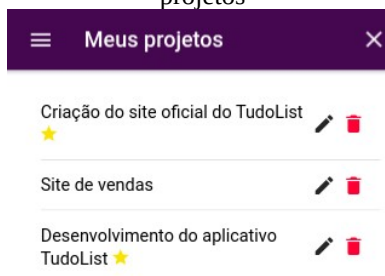
**Fonte:** Imagem criada

**Figura 52:** Botão de editar projetos



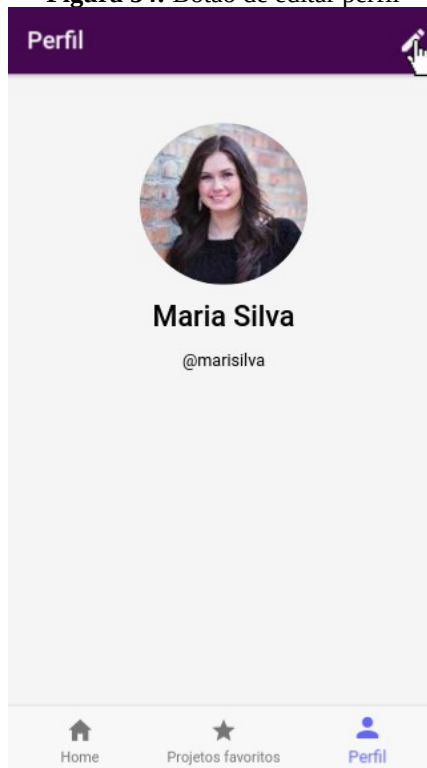
**Fonte:** Imagem criada pelo autor

**Figura 53:** Tela de editar e excluir projetos



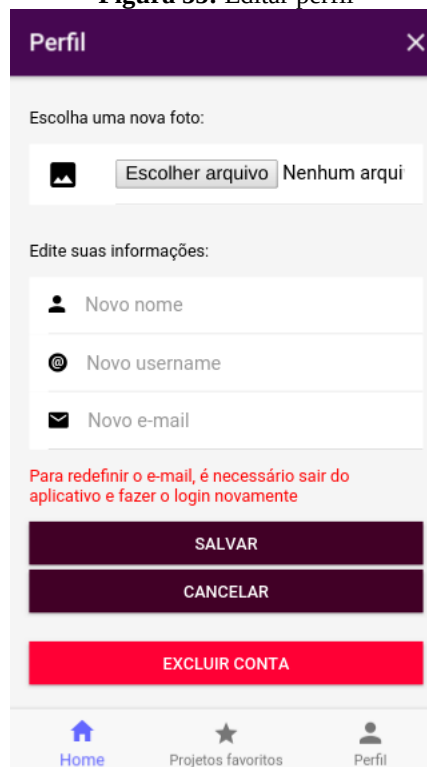
**Fonte:** Imagem criada pelo autor

**Figura 54:** Botão de editar perfil



**Fonte:** Imagem criada pelo autor

**Figura 55:** Editar perfil



**Fonte:** Imagem criada pelo autor

**Figura 56:** Botão de pesquisar projetos



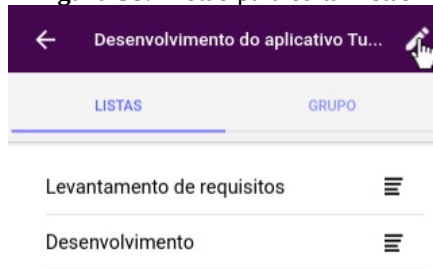
**Fonte:** Imagem criada pelo autor

**Figura 57:** Tela de pesquisa de projetos



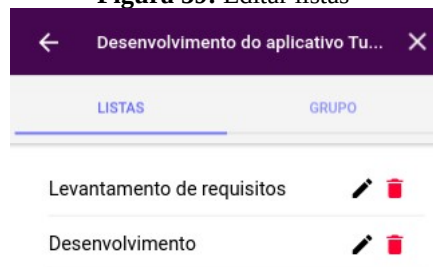
**Fonte:** Imagem criada pelo autor

**Figura 58:** Botão para editar listas



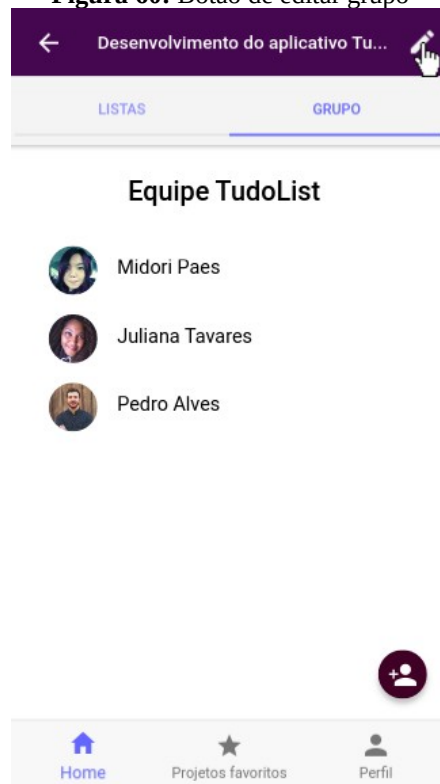
**Fonte:** Imagem criada pelo autor

**Figura 59:** Editar listas



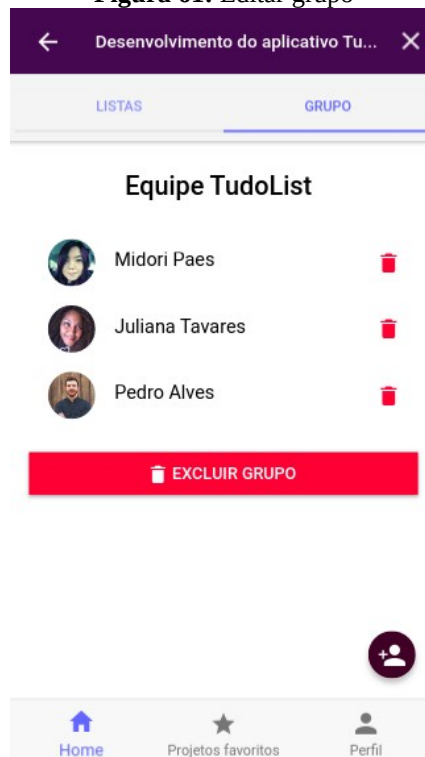
**Fonte:** Imagem criada pelo autor

**Figura 60:** Botão de editar grupo



**Fonte:** Imagem criada pelo autor

**Figura 61:** Editar grupo



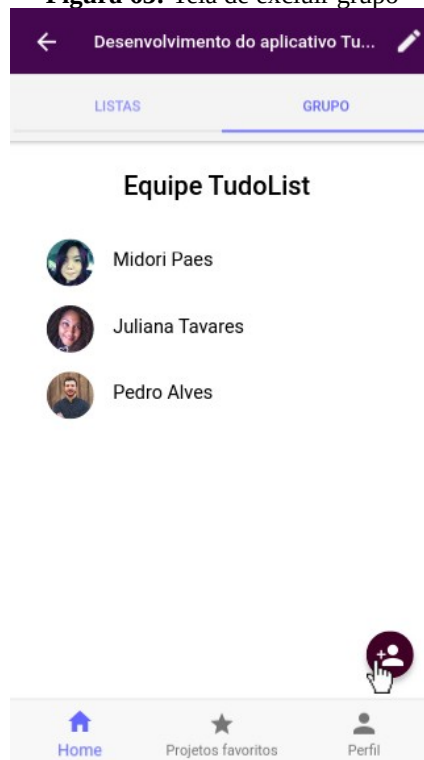
**Fonte:** Imagem criada pelo autor

**Figura 62:** Botão de editar grupo



**Fonte:** Imagem criada pelo autor

**Figura 63:** Tela de excluir grupo



**Fonte:** Imagem criada pelo autor



**Figura 64:** Detalhes da tarefa com colaborador



**Fonte:** Imagem criada pelo autor