**Especificação dos Requisitos do Software**

**Lucas Eduardo Spreng**

**Matheus Sardeli Malheiros**

**Rafael Alves Camillo**

**São João da Boa Vista, 2020**

1. **Introdução**

**1.1) Visão Geral deste Documento**

Este documento consiste em apresentar os requisitos funcionais, os requisitos não-funcionais, o diagrama de casos de uso e as regras de negócio do aplicativo mobile, que visa a implementação do método GTD.

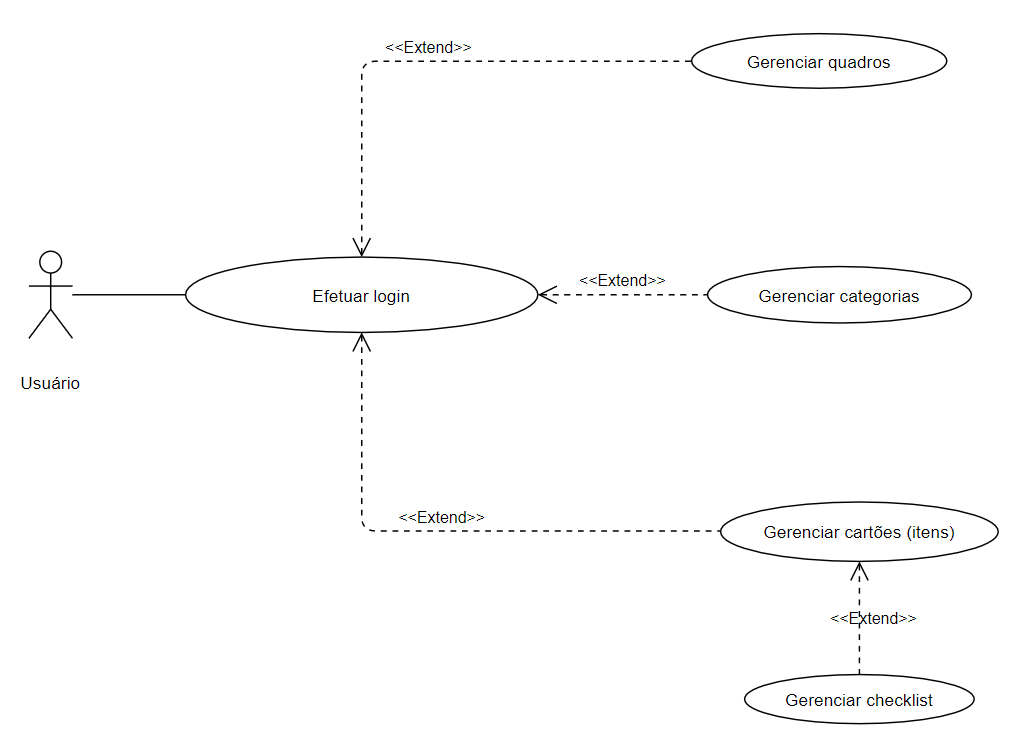
1. **Requisitos Funcionais**

* **RF#001**: O sistema deverá permitir o acesso ao sistema através da autenticação do Google.
* **RF#002**: O sistema deverá permitir o gerenciamento de quadros;
* **RF#003**: O sistema deverá permitir o gerenciamento de categorias;
* **RF#004**: O sistema deverá permitir o gerenciamento de cartões (itens);
* **RF#005**: O sistema deverá permitir a inserção e manipulação de uma lista de verificação (checklist) nos cartões, caso necessário;
* **RF#006**: O sistema deve permitir que o usuário logado informe o horário e o local para realização do evento.
* **RF#007**: O sistema deve permitir que o usuário logado informe o tempo estimado de duração para a realização do evento.

1. **Requisitos não Funcionais**

* **RNF#001**: O sistema deverá demorar menos de 3 segundos para efetivar uma ação.
* **RNF#002**: O sistema deverá ser acessado pelas plataformas mobile Android e iOS.
* **RNF#003**: O sistema deverá possuir uma interface agradável, simples e de fácil uso;
* **RNF#004**: O sistema deve ter disponibilidade de 99% do tempo;
* **RNF#005**: O sistema deverá ser desenvolvido na linguagem React Native;
* **RNF#006:** O sistema deverá possibilitar o armazenamento de dados na nuvem.

1. **Diagrama de Casos de Uso**

****

1. **Regras de Negócio**
2. O usuário pode fazer login ao abrir o aplicativo, caso não esteja autenticado. Este login é realizado através de uma conta Google com o mecanismo OAuth.
   1. Na primeira vez que um usuário fizer o login com sua conta Google, caso não haja um cadastro, este é criado automaticamente.
3. Caso o usuário não realize o login, não poderá utilizar nenhuma outra funcionalidade do aplicativo.
4. Quando o usuário já estiver autenticado, irá se deparar com a tela principal do sistema, a qual se refere ao gerenciamento de quadros.
5. O gerenciamento de quadros consiste em vários quadros organizados em uma ScrollView horizontal, os quais podem ser modificados e removidos, além de um botão como último elemento desta View para a criação de novos quadros.
6. Dentro de cada quadro, o usuário autenticado pode criar novos cartões.
7. Ao selecionar um cartão, o usuário pode modificar os dados deste cartão, e o quadro ao qual ele pertence. Além disso, pode remover o cartão.
8. Para os dados internos dos cartões, o usuário pode optar por adicionar novas categorias e selecioná-las nos cartões em que deseja.
9. Opcionalmente, ao criar/editar um cartão o usuário pode também inserir um checklist sobre as tarefas que precisam ser realizadas para que a tarefa principal do cartão seja realizada.
10. As checklists dentro dos cartões podem ser modificadas, adicionando novas tarefas e removendo-as, caso necessário.
11. **Priorização das Tarefas a serem feitas**
12. Criar interface de login/cadastro, possuindo informações de boas vindas e um botão de acesso com a conta Google.
13. Implementar uma função de acesso com conta Google ao clicar no botão do formulário de login.
14. Persistir estado de autenticação no aplicativo para que o usuário possa continuar autenticado após realizar o login.
15. Criar interface para a visualização dos quadros, tarefas e categorias.
16. Desenvolver mecanismo para permitir a adição de novos quadros na interface.
17. Permitir a atualização do nome dos quadros e sua remoção.
18. Desenvolver mecanismo para permitir a adição de novos cartões dentro dos quadros.
19. Criar interface para inserção/modificação dos dados dos cartões.
20. Implementar função de remoção dos cartões através de um botão na interface de modificação dos cartões.
21. Desenvolver mecanismo para a gerência de checklists na descrição do conteúdo dos cartões.
22. Desenvolver mecanismo para a gerência de categorias que possam ser utilizadas no conteúdo dos cartões.
23. Integrar todas as funcionalidades com o backend para garantir a persistência dos dados no servidor.
24. Desenvolver mecanismos para persistir os dados de maneira local para que o usuário possa utilizar o aplicativo de forma offline.
25. Compilar aplicativo.