

FUNDAÇÃO EDUCACIONAL DO MUNICÍPIO DE ASSIS  
INSTITUTO MUNICIPAL DE ENSINO SUPERIOR DE ASSIS

**Igor Armando de Oliveira Novaes**

**APLICATIVO DE ASSISTENCIA PESSOAL COM A  
TECNOLOGIA REACT NATIVE**

ASSIS  
2021

FUNDAÇÃO EDUCACIONAL DO MUNICÍPIO DE ASSIS  
INSTITUTO MUNICIPAL DE ENSINO SUPERIOR DE ASSIS

**APLICATIVO DE ASSISTÊNCIA PESSOAL COM A  
TECNOLOGIA REACT NATIVE**

ALUNO: Igor Armando de Oliveira Novaes

ORIENTADOR: Dr. Osmar Aparecido Machado

<b>Nota do Orientador:</b>	<b>Nota do Avaliador:</b>
----------------------------	---------------------------

ASSIS  
2021

# 1 INTRODUÇÃO

Com a evolução da tecnologia, tarefas complexas foram se tornando cada vez mais simples e fáceis, e com a introdução dos smartphones, surgiram os aplicativos com funcionalidades que otimizam ainda mais as tarefas simples e são soluções para as tarefas complexas, que demandam tempo e esforço.

De acordo com a Think Tank Pew Research Center, uma empresa que estuda sobre as tendências que estão atualmente moldando o mundo, o desenvolvimento de aplicativos deve movimentar 6,3 trilhões até esse ano 2021. Por conta disso e da alta demanda de aplicações mobile observadas na atualidade influenciadas pelas necessidades que a pandemia trouxe, para a realização deste trabalho foi feito um levantamento de dados por meio de entrevistas, a fim de orientar, na qual foi constatado que as pessoas que estão trabalhando como home-office têm dificuldades para se organizar. Como resultado dessa rápida pesquisa, é analisado que, aplicações que auxiliam na organização pessoal, têm sido procuradas com frequência, porém, algumas pessoas se sentem desconfortáveis com algumas delas por não atenderem completamente suas necessidades ou serem difíceis de se utilizar, visto que a aplicação será muito usada durante o dia a dia, isso torna necessário que o usuário se sinta confortável, e que não exija muito tempo ou habilidade para o utilizar.

## 1.1. OBJETIVOS

O objetivo deste trabalho é o desenvolvimento de uma aplicação mobile para controle de atividades cotidianas do usuário, sendo um assistente pessoal que auxilia na organização de tarefas do usuário, ajudando ele a não se perder diante de varias datas e horarios dos seus compromissos, de facil compreensão e utilização, visando que é preciso o maximo de fluides para que o usuário não se perca dentro do proprio aplicativo, ou não consiga utiliza-lo por ser muito complexo O software será desenvolvido com o framework React Native, utilizando algumas

bibliotecas para tornar a aplicação fluida e auxiliar no desenvolvimento. A persistência dos dados será realizada utilizando MongoDB, visto que, um banco de dados não-relacional, atende melhor os requisitos de desenvolvimento. A aplicação será desenvolvida até o final de Outubro de 2021, com custo inicial sendo zero, mas com a possibilidade de aumentar de acordo com a demanda da aplicação, pois serão utilizadas, ferramentas exentas de pagamentos.

## **1.2. JUSTIFICATIVAS**

Segundo Annie App uma empresa norte-americana que realiza pesquisas sobre o mercado das aplicações mobile, na quarentena houve um aumento de 40% na procura de aplicações comparado ao começo do ano de 2020, quando foi o início da pandemia. Muitas necessidades foram desenvolvidas com a quarentena, é dever da tecnologia sanar partes destas dificuldades. Mesmo no ano de 2018 e 2019 o aumento da procura foi considerável, e a estimativa é crescer ainda mais esses números nos próximos anos. Com isso o desenvolvimento de aplicações mobile, que auxiliem os usuários em diversos aspectos, se torna muito necessário.

O aumento da popularização da tecnologia React Native vem sendo observada positivamente por muitos, e devido a isso, o número de fóruns e artigos sobre o mesmo é muito comum, o que influencia diretamente na eficiência do desenvolvimento da aplicação. Esse framework vem sendo muito buscado por diversas empresas, por conta dessa popularidade, aumenta ainda mais o prestígio da tecnologia, facilitando muito no desenvolvimento de aplicativos mobile.

Com a pandemia a Covid-19, muitas coisas mudaram na vida e no cotidiano de muitas pessoas. O fato de termos que fazer tudo o que for possível sem sair de casa, trouxeram algumas dificuldades para pessoas mais atarefadas, pois fazer tudo no conforto de suas casas acaba trazendo uma falsa sensação de conforto e controle sobre todo tipo de ocasião, principalmente para aqueles que atualmente trabalham como home-office.

Devido às necessidades que emergiram, uma aplicação para auxiliar o controle e definição das tarefas diárias, conforme o tempo passa, se torna necessário

### **1.3. FOCO DO TRABALHO**

Aplicar o conhecimento adquirido no curso de Análise e Desenvolvimento de Sistemas e os estudos para o desenvolvimento de aplicações mobile com React Native, que tornam os códigos mais rápidos, pois aproveita todos os códigos para as plataformas Android e IOS, o que torna a programação do aplicativo ainda mais rápida e torna a experiência do usuário mais fluida.

O grande foco deste trabalho é auxiliar usuários em tarefas do cotidiano, para que ele possa se organizar melhor, com uma aplicação que atenda aos requisitos necessários para uma melhor experiência do mesmo.

Com este trabalho, o foco é a assistência que será entregue para os usuários, que devido a pandemia, principalmente para aqueles que estão trabalhando em home-office, e também para os estudantes que estão fazendo suas aulas remotamente. O auxílio para controle de tarefas pode ser utilizado por qualquer usuário, que tenham dificuldades para se organizarem ou até mesmo que tenham muitas tarefas durante o dia.

A assistência pessoal tem como principal foco, manter as tarefas controladas, e organizadas, para que a pessoa que esteja a utilizando tenha o conforto de que tudo que ele fará durante o dia, semana ou mês estará documentado de forma simples, e que ele será notificado sobre seus compromissos.

### **1.4. METODOLOGIA DE PESQUISA**

Este projeto foi definido depois de um levantamento de informações realizado com pessoas que compartilham do dia-a-dia do autor do estudo. Constatou-se, que algumas pessoas apresentam dificuldades em controlar o tempo gasto em suas atividades diárias, como ir ao banco, padaria, atividades do trabalho, dentre outras. Desta forma, as necessidades destas pessoas influenciaram na a definição deste trabalho.

Quanto ao mérito metodológico, o levantamento de dados e materiais

bibliográficos enquadram o estudo como uma pesquisa exploratória e descritiva. De outro lado, continuarão sendo executadas para que a aplicação seja mais completa possível para os usuários, além da produção do aplicativo e testes pelos usuários. Isso coloca o estudo na categoria experimentação, que segundo (Fonseca, 2002) é o tipo de pesquisa em que se é definido o foco de estudo, e logo após, explicitar as problemáticas que influenciam no objetivo e observar a forma com que essas problemáticas atuam.

Quanto ao mérito tecnológico, o estudo será desenvolvido segundo os preceitos da orientação a objetos. As tecnologias utilizadas serão, React Native, para desenvolvimento front-end da aplicação que irá implementar toda a parte visual e ser a ligação entre o usuário e as funcionalidades da aplicação. Node.js para desenvolvimento do back-end onde será feita os controladores, modelos e rotas da API, que será consumida pelo front-end da aplicação. MongoDB para persistência dos dados em formas de objetos (JSON) que será armazenado em um cluster no Mongoddb Atlas.

Em síntese, as pesquisas serão executadas com entrevistas e também com questionários para que tenha cada vez mais informações para execução do projeto. Diagramas de casos de uso, diagrama de classes e diagrama de atividades serão utilizados para diagramação e auxílio do projeto.

## 1.5. RECURSOS NECESSÁRIOS

<b>Recursos</b>	<b>Justificativas</b>
Scholar Google	Para referencias e pesquisas
Computador	Acesso à internet para pesquisa, e desenvolvimento da aplicação
Smartphone	Utilizado para testes e definições da aplicação e também como forma de obter feedback de usuário.

Visual Code	Ambiente de desenvolvimento da aplicação
AWS	Onde será feito o deploy da aplicação
Mongodb Compass	Cluster de banco de dados
Google Forms	Pesquisa a partir de formulários
Trello	Organizar o projeto por partes e definir quais os dias elas serão feitas.
Microsoft Word	Para realizar a escrita do trabalho.
Astah	Ferramenta UML para criação de diagramas

## 1.6. ESTRUTURA DO TRABALHO

O trabalho será estruturado em 5 capítulos, que serão respectivamente:

Capítulo 1: Trata-se desta introdução e da apresentação do trabalho

Capítulo 2: Neste capítulo será realizada a revisão da literatura, onde serão apresentados os principais conceitos e teorias envolvidos no desenvolvimento do trabalho.

Capítulo 3: Será introduzido a diagramação e a documentação deste projeto

Capítulo 4: Onde será descrita a conclusão e os planos futuros deste projeto.

Capítulo 5: É onde estará todas as referências utilizadas para escrita deste projeto

A parte de pesquisa sobre o aumento da necessidade de aplicativos mobile durante a pandemia, outra pesquisa sobre as necessidades dos usuários, para definir as melhores funcionalidades para a aplicação, definição e explicação das escolhas das tecnologias que serão utilizadas no aplicativo, modelagem dos casos de uso, e por fim, amostra e elucidação do aplicativo e das funcionalidades nele existentes.

**1.7. CRONOGRAMA (pode excluir o Cronograma, ele não é necessário nesta fase)**

ATIVIDADE	MÊS	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
Entrevistas/Coleta de informações de usuários											
Levantamento de requisitos											
Diagrama de Caso de uso											
Definição geral das tecnologias necessárias para desenvolvimento											
Diagrama de Classes											
Diagrama de Atividades											
Criação de protótipos											
Desenvolvimento Back-end											
Desenvolvimento Front-end											
Testes											
Liberação de teste para Usuários											



## **2. REVISÃO DA LITERATURA**

Neste capítulo será realizada uma revisão das principais tecnologias utilizadas no desenvolvimento do estudo. Dentre tantas opções disponíveis no mercado para essa finalidade, a revisão da literatura tornou-se uma etapa fundamental para se definir as ferramentas tecnológicas mais adequadas para se alcançar o objetivo proposto.

Serão relatadas, no entanto, somente as ferramentas selecionadas. Foram desconsideradas as ferramentas similares que mesmo sendo fundamentais para o processo de escolha, optou-se por não listá-las, a fim de tornar o material mais prático para o leitor interessado especificamente nestas tecnologias.

### **2.1. REACT NATIVE**

React Native é um framework de JavaScript, feito para o desenvolvimento de aplicações mobile híbridas, ou seja, aplicações para os dois sistemas operacionais que existem atualmente no mercado, Android e IOS. React Native, foi baseado no framework React.js criado pelo Facebook para aplicações web, ambos são bem parecidos tendo como principal semelhança (além do nome) a possibilidade de trabalhar com componentes e suas propriedades, podendo reaproveitá-los dentro do código. As principais vantagens do React Native são, a experiência que o usuário terá na utilização do aplicativo, maior performance no geral, carregamento rápido dos componentes, integração com todas as principais ferramentas de um smartphone entre outras.

Algumas ferramentas do framework são extremamente úteis, como a de Hot Reloading que faz as modificações que são feitas na aplicação, serem injetadas nela mesmo com ela rodando, além da funcionalidade de depurar a aplicação no navegador podendo assim inspecionar elementos, utilizar ferramentas de

desenvolvedor, o que facilita para quem trabalha com o desenvolvimento web. Os elementos não são escritos em HTML, sendo assim, as sintaxes são diferentes do convencional para os frameworks de JavaScript. Uma tecnologia muito utilizada atualmente, com diversas bibliotecas e plugins que diminuem o tempo de desenvolvimento, auxiliam na metodologia CleanCode, deixam a aplicação ainda mais fluida, etc..

Essa tecnologia vem crescendo muito cada dia mais no desenvolvimento de aplicações mobile, existindo diversos fóruns e artigos sobre o framework facilitando no desenvolvimento.

## **2.2. MONGODB**

MongoDB é um banco de dados NoSQL orientado a documento, ele é diferente dos bancos de dados convencionais, nele não é necessário a criação de tabelas e colunas. Os dados podem ser moldados a partir da necessidade deles, podendo ter campos diferentes entre um dado e outro. Os dados são armazenados em formato JSON ou BSON e são bastante intuitivos sendo fácil o entendimento sobre o que se trata os dados apresentados no documento. MongoDB oferece alto desempenho se comparado com os bancos SQL e escalabilidade com serviços de sharding e replicaset que permite que você replique suas collections para outros servidores. As tabelas são chamadas que collections e um conjunto delas formam a database. MongoDB atualmente vem crescendo muito, aparecendo em 5º lugar no ranking mundial dos bancos de dados populares do ano de 2020, empresas gigantes como Ebay, Google e Adobe utilizam esse banco de dados não relacional

## **2.3. API**

A APIs (*Application Programming Interface*) é um conjunto de rotinas que dão acesso a um determinado software. APIs normalmente são desenvolvidas quando existe intenção de integrar os dados desta API com outras aplicações, sendo consumidas através de requisições HTTP. Com isso pode se definir as APIs como uma ponte entre os dados de um determinado software, e a aplicação que

consumirá esses dados. Atualmente as APIs estão presente em grande parte dos softwares e principalmente em aplicações web, visto que a comunicação de uma API com a tecnologia front-end usada na aplicação é muito simples, isso da flexibilidade, controle e segurança para os websites.

## **2.4. NODE.JS**

Node.js é um ambiente para execução de códigos JavaScript. Criado a partir do motor v8 da Google, Node.js foi um dos principais responsáveis pela popularização do JavaScript, tornando ele popular também na parte de back-end das aplicações que usam uma arquitetura single-thread que trata as requisições como um evento, que é recriado após ficar aguardando a entrada de uma nova requisição. O Nodejs é indispensável em uma stack JavaScript, além de ser responsável por controlar o back-end de aplicações de grandes empresas como, Netflix, LinkedIn, Uber Walmart e muitas outras empresas de diversos seguimentos. Além disso, o Nodejs conta com o NPM e o Yarn, gerenciadores de pacote, que contam com um repositório de milhões de softwares, além de que, o Yarn é mantido pela comunidade, dando uma grande flexibilidade para o gerenciador.

## **2.5. DIAGRAMAS**

O UML sigla para linguagem de modelagem unificada em português, será a linguaguagem utilizada para modelar esse projeto. Ela foi criada com o intuito de auxiliar a engenharia de software, com a criação de diagramas de modelagem, como forma de prever os limites e testar a estrutura do projeto e, ainda, realizar levantamentos que descrevem toda a base do projeto antes que ele seja iniciado de forma pratica.

### **2.5.1. Diagrama de Classe**

O diagrama de classe é um dos tipos de modelagens mais utilizados do UML, ele serve tanto para definir algumas funções dos seus projetos, e também definir relacionamentos entre entidades, e com isso algumas modelagens do seu banco de dados também. A modelagem desse diagrama definirá algumas funções importantes e descreverá sobre o que pode estar presente no seu sistema.

### **2.5.2. Diagrama de Caso de Uso**

O diagrama de UC ou de caso de uso, definirá sobre como seu sistema irá se comunicar com o usuário. Essa modelagem, serve para definir os passos que o usuário vai realizar para determinadas funções que existirão, e definir quais são as melhores para se colocar, e qual será a melhor hora para apresentar ela ao usuário. O Caso de Uso é feito da seguinte forma, os atores (pessoas que utilizaram o sistema) são o início de tudo, e com eles, podem ser definidos os processos deste diagrama. O sistema é são os passos que fazem os atores e os sistemas interagirem entre si, também pode se chamar de cenário, e as Metas de acordo com o site Lucidchart(2018), são os resultados dos casos de uso, quando um diagrama é criado corretamente, eles devem poder descrever todas as atividades desenvolvidas para alcançar a meta.

### **2.5.3. Diagrama de Atividade**

Esse diagrama é utilizado em conjunto com o diagrama de caso de uso, ele é mais referente a parte das regras de negócio do projeto, e atua em facilitar o entendimento de casos de uso descrevendo sobre todos os passos que serão tomados de forma mais completa, e quais os passos poderão ser tomados de acordo com o que o usuário pretende fazer....

### **3. DIAGRAMAÇÃO E DOCUMENTAÇÃO DO APLICATIVO**

Neste capítulo, será mostrado e explicado, a parte de diagramas feitos com a linguagem UML, que fará o auxílio na construção do sistema, definindo assim, parte da etapa de engenharia de software do projeto. Depois de pronto, os diagramas servirão como base no desenvolvimento e como ferramenta de entendimento para leitores e usuários do próprio aplicativo, para com as formulações explícitas neles.

### 3.1 CASO DE USO

Existem diferentes formas de documentar um caso de uso. Neste projeto optou-se pelo caso de uso unificado com todos os casos de uso juntos, pois desta forma é

A Figura 1 apresenta o caso de uso do aplicativo, nele se observa um unico Ator, que será o usuario da aplicação. Em azul escuro, estão assinalados os casos de uso que se relacionam diretamente com usuario.

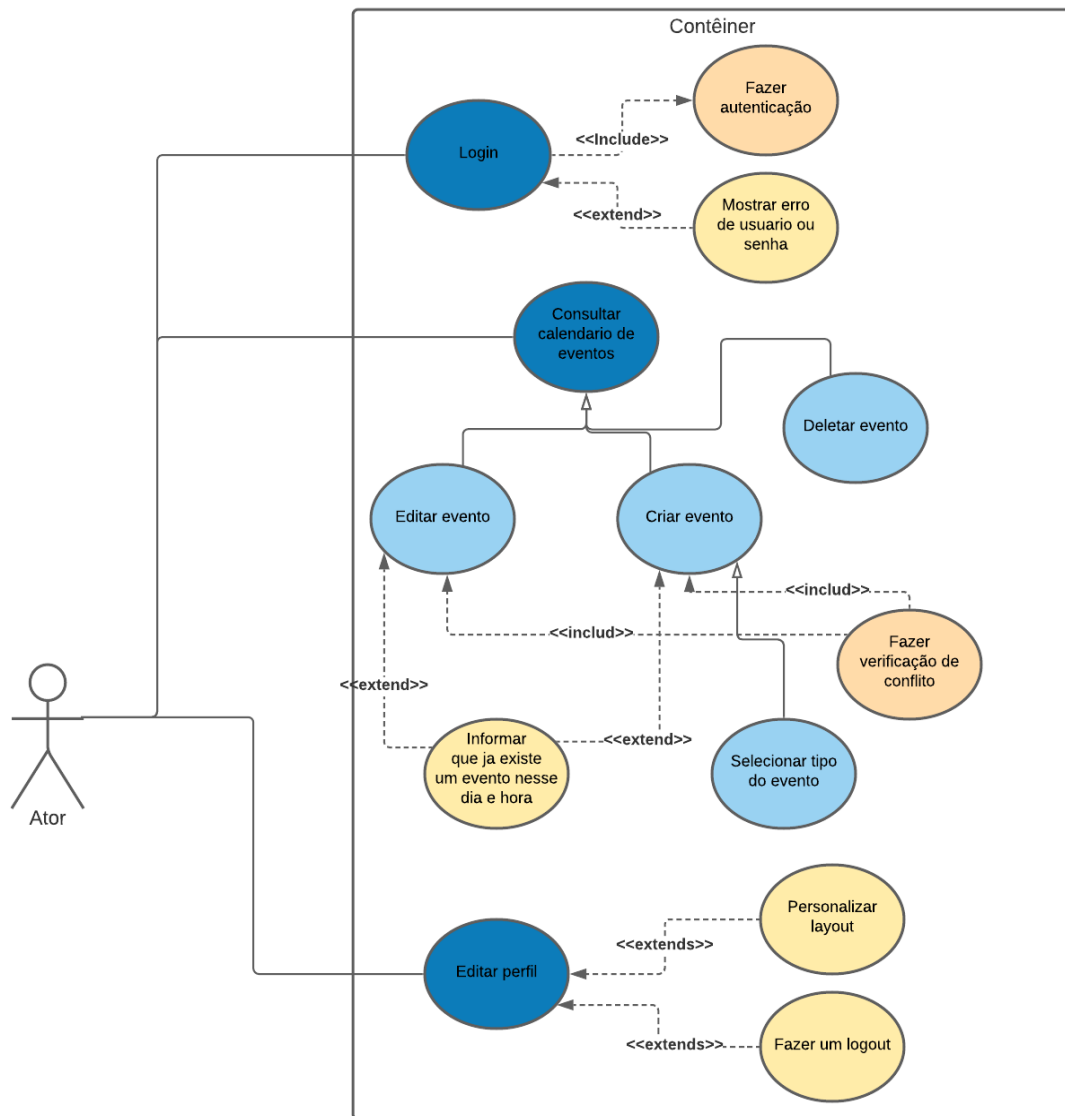


Figura 1 – Diagrama de Caso de Uso

Como se observa na Figura 1, o caso de uso principal está estruturado em tres casos que são utilizados para ilustrar alguns esquemas que serão utilizados no aplicativo.

- Login: representa o usuario inserindo seus dados e terá um relacionamento de inclusão que é chamado de “Fazer Autenticação”. Ainda relacionado com o Login, se observa um caso de uso extendido chamado “Mostrar erro de usuario ou senha”, no qual é exibido somente quando ocorre um erro de validação das credenciais.
- Consultar calendario de eventos: refere-se a visualização do usuario, em seus eventos, e junto a ele existem três classes secundarias, “Editar evento”, “Criar evento” e “Deletar evento”, que são as operações que o usuario fará para ter controle do todos seus compromissos. Ainda neste caso de uso, observa-se uma inclusão que é feita em dois casos de uso secundarios:
  - ✓ Fazer verificação de conflito: relacionada com “Editar evento” e;
  - ✓ Criar evento: onde ocorrerá a verificação dos requisitos para realização da operação.

Existe também uma extensão para esses dois casos de uso secundarios denominada “Informar que já existem eventos nessa data e hora” e isto servirá como um controle, para que o usuário não marque compromissos para o mesmo dia e hora caso ele não consiga executar os dois simultaneamente se necessário.

O caso de uso secundário “Criar evento”, terá um outro caso de uso secundário chamado “Selecionar tipo do evento” no qual será responsável de armazenar e informar respectivamente ao usuário qual será a classe do compromisso em que ele terá

Por fim, o caso de uso principal “Editar perfil”, demonstra que será feito a edição visual e aonde o cliente fará algumas alterações. Existem duas extensões:

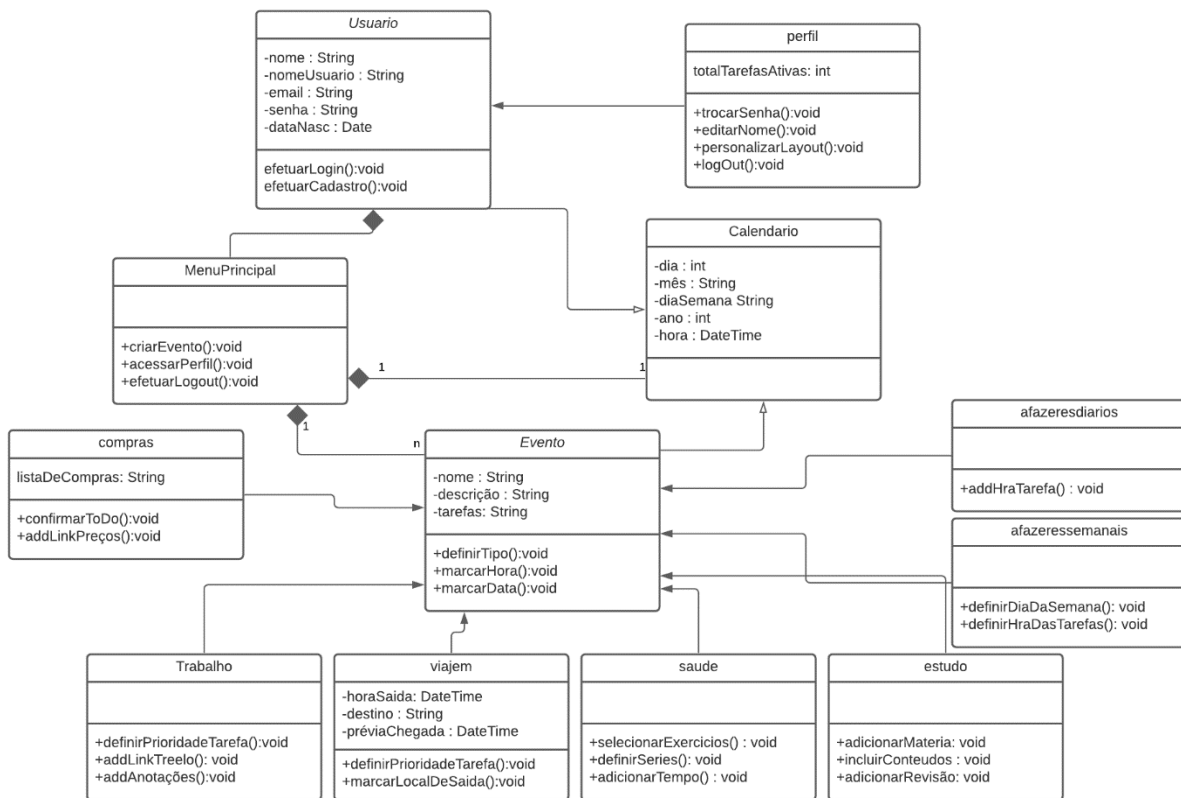
- Personalizar layout: para edição dos componentes visuais, e essas informações ficarão registradas na conta dele;

- Fazer logout: que servirá para sair da conta que está atualmente logada.

Estes componentes integram a lógica da aplicação, assim dando um entendimento maior para os usuários sobre como está acontecendo as coisas por trás dessa aplicação

### 3.2 DIAGRAMA DE CLASSE

Para melhor compreensão deste diagrama, serão destacadas as funcionalidades de cada classe e seus relacionamentos:



**Figura 2 – Diagrama de Classe**

Usuário: classe referente ao utilizador do aplicativo, os atributos nela referenciados serão utilizados para captação de dados na aplicação, e os métodos são para



realização do login e para autenticar o usuário, e efetuar o cadastro para poder realizar o login na próxima etapa.

Perfil: esta classe servirá para fazer algumas alterações na conta do usuário e fazer algumas verificações, os métodos são para alteração de senha para efetuar o login, editar nome que será exibido no layout da aplicação, personalização do layout para edição visual de alguns componentes, e efetuar a saída do usuário de sua própria conta.

Menu Principal: referência todo o dashboard da aplicação onde o usuário encontrará todos os seus eventos, onde estará o menu lateral para visualização das opções e os métodos são respectivamente para criação de um evento e visualização, acessar o perfil, e também é possível realização do logout

Calendário: essa é a classe que fará a dinâmica da aplicação, podendo ver quais dias e horas possuem eventos, e assim poderá selecionar um dia diretamente do calendário para criar um compromisso. Seus atributos são referentes a todas as informações necessárias para visualizar quando um evento ocorrerá e também saber definir uma futura tarefa.

Evento: referente à onde ficaram registrado os eventos e como eles serão identificados, tendo os atributos registro e identificação dos quais ele também herda de calendário, e também sete subclasses, que são referentes aos tipos de evento dos quais o usuário poderá definir para sua tarefa, assim tendo algumas opções diferentes para se organizar dependendo do seu tipo de evento.

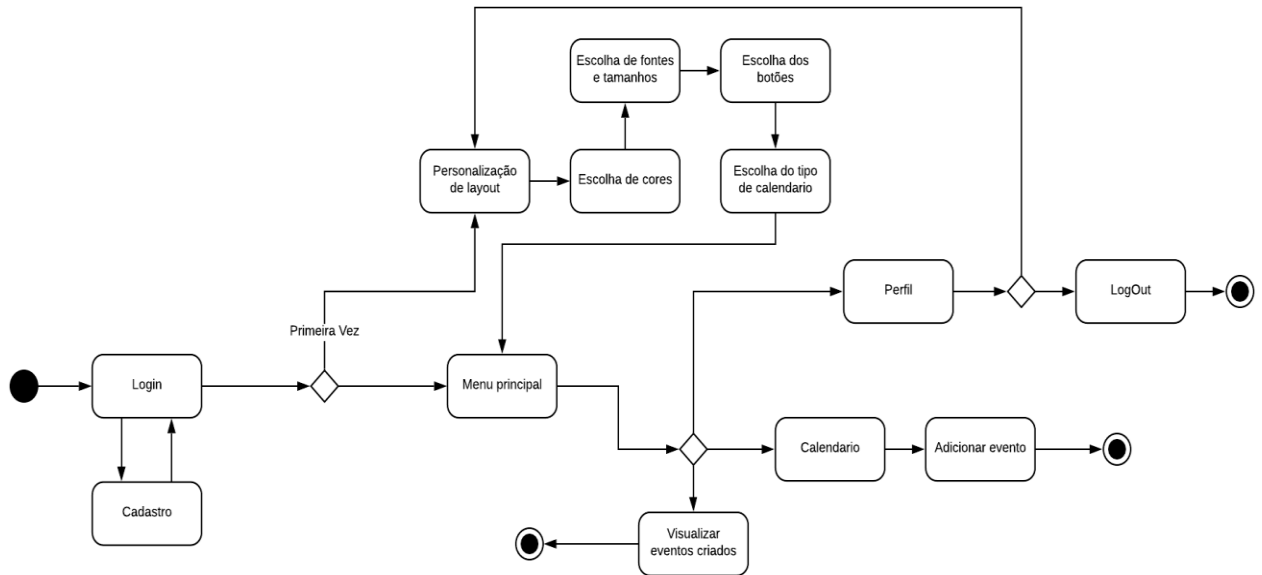
- Viagem: se refere a marcar viagens, e as tarefas que estarão diretamente ligadas com essa viagem
- Trabalho: é o tipo de evento onde o usuário pode definir os eventos e tarefas que serão realizadas em relação ao seu trabalho.
- Saúde: este tipo de evento envolve a saúde no geral, tendo métodos específicos para cuidados com o corpo, mas também se refere a eventos como uma ida ao médico.
- Estudo: tipo de evento com métodos voltados para o meio acadêmico, onde o

usuário poderá definir horário de provas, trabalhos e também adicionar alguns materiais, conteúdos e revisões.

- Afazeres diários e afazeres semanas: este tipo serve para coisas que você precisa fazer normalmente todos os dias ou em algum determinado dia da semana, e quer manter-se organizado, e seus métodos são simples apenas para registrar a hora e o dia ou a hora e o dia da semana.
- Compras: tipo que está relacionado com compras no geral, tendo a definição de uma lista de comprar que realizará em um determinado dia, além de métodos para definir os itens da lista como confirmado e adicionar links de pesquisas para não perder um produto ou ofertas.

### **3.3 DIAGRAMA DE ATIVIDADE**

O diagrama de atividades é utilizado para demonstrar todo o caminho que o usuário irá percorrer até determinado ponto para realizar uma ação específica e mostra quais as condicionais para que uma ação seja feita. Como se observa na Figura 3, é possível visualizar como o usuário fará o trajeto dentro da aplicação para criar um evento, deletar, atualizar e também, qual caminho seguir afim de mudar suas preferências de layout

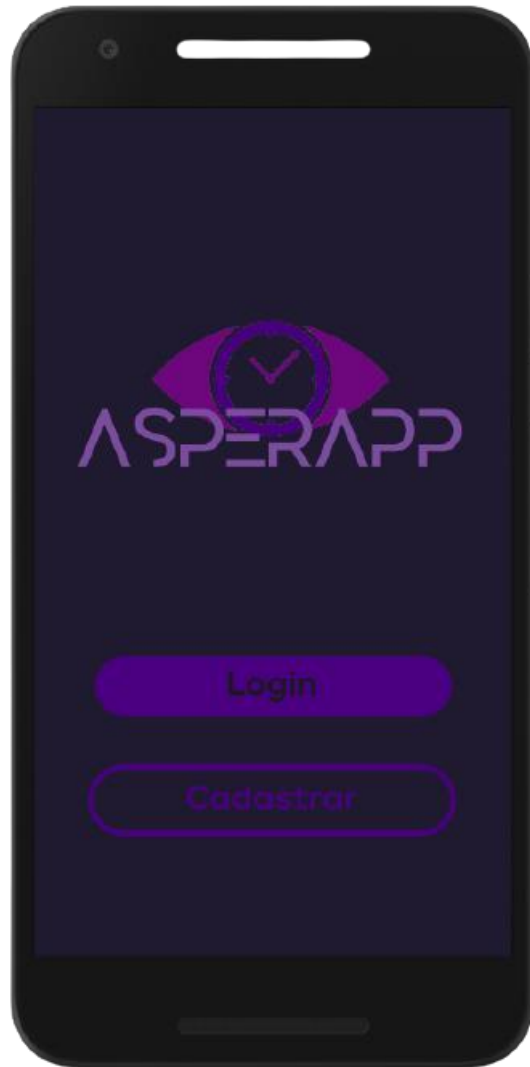


**Figura 3 – Diagrama de Atividade**

### 3.4. FRONT-END / UI

Nesta seção serão apresentadas todas as etapas que compõe o front-end da aplicação, ou seja, as interfaces que farão a interação com o usuário da aplicação, com o intuito de facilitar o entendimento sobre cada uma das telas deste aplicativo.

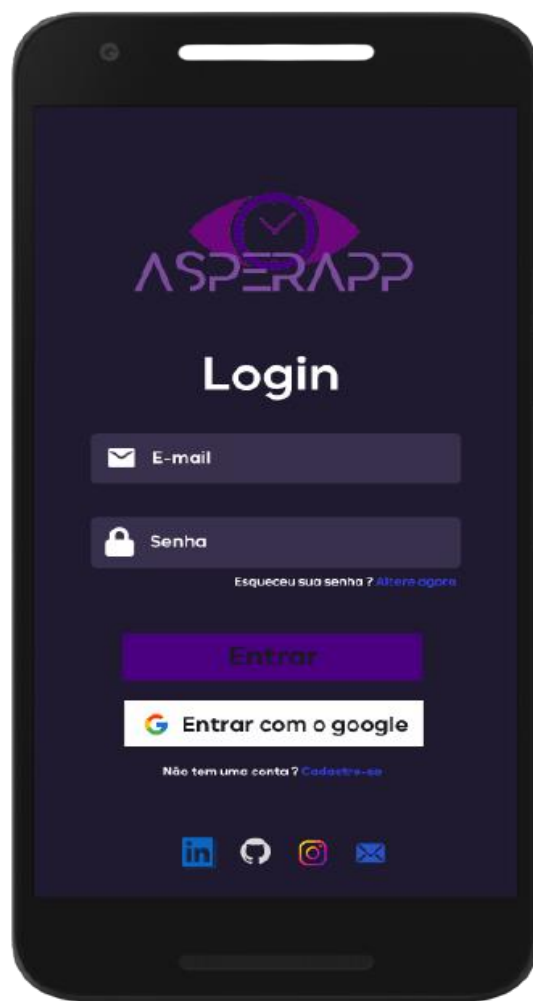
### 3.4.1. TELA INICIAL



**Figura 4 : Interface (tela inicial)**

Esta será a primeira tela que o usuário visualizará depois de fazer o download do aplicativo, uma tela simples onde ele poderá escolher entre duas opções. Clicando no botão de cadastro ele será enviado para a tela de cadastro, onde irá preencher um formulário contendo seus dados, sendo esta a primeira etapa de um usuário que está baixando o aplicativo pela primeira vez. Clicando no botão de login, ele será encaminhado para uma tela onde será inserido as credenciais para que seja validado a entrada do mesmo na aplicação.

### 3.4.2. TELA DE LOGIN

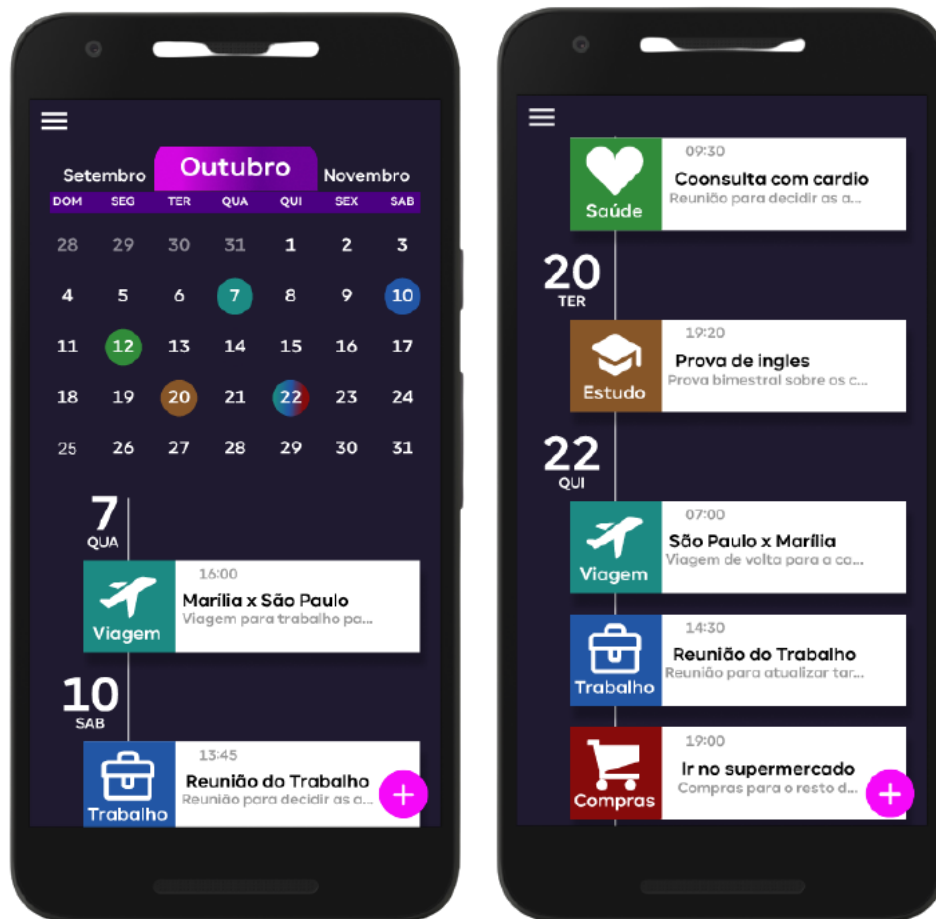


**Figura 5 : Interface (tela login)**

Na figura 5 mostra a tela de login, nesta tela o usuário colocará suas credenciais cadastradas nas bases de dados da aplicação, para gerar um token que será utilizado como validação para acessar todas as rotas da API. Existe uma opção em usuário pode efetuar um login com uma conta google clicando no botão branco escrito “Entrar com google”, o que torna o login no aplicativo um pouco mais prático, nesta tela o usuário pode redefinir sua senha clicando em um link logo abaixo do input de senha, e também, logo abaixo do botão de entrar com o google, existe um link para caso você ainda não esteja registrado, ir diretamente para a tela de

cadastro. No rodapé da tela, é mostrado alguns ícones, que ao clicar o usuário será redirecionado até os aplicativos (caso estejam instalados) ou ao browser nas redes sociais do desenvolvedor.

### 3.4.3. TELA PRINCIPAL



**Figura 6 e 7: Interface (tela principal)**

A tela principal é como um painel de controle, onde o usuário poderá visualizar seus eventos e alterar os mesmos, além disso, é nesta tela que será possível a adição de uma nova tarefa. O componente de calendário no topo da tela é aonde o usuário fará o controle da visualização das tarefas passadas, atuais e futuras através dos meses, arrastando o componente para o lado. Ainda no calendário é possível ver círculos de cores distintas nos dias de suas tarefas, que se referem a um tipo diferente de delas, deixando visualmente simples de diferenciar qual tipo de evento o usuário cadastrou naquele dia, caso o usuário tenha diversas tarefas de

diferentes tipos no mesmo dia, essa cor se tornará um gradiente com as cores dos eventos que estão cadastrados para aquele mesmo dia. Abaixo do calendário, será exibido uma serie de cards, com os dias marcados para a tarefa logo acima. Os cards contém algumas informações resumidas para uma visualização inicial como título da tarefa, hora, uma breve descrição, e uma caixa contendo a cor, o icone, e o tipo de tarefa para aquele dia.

#### 3.4.4. CARD DA TAREFA



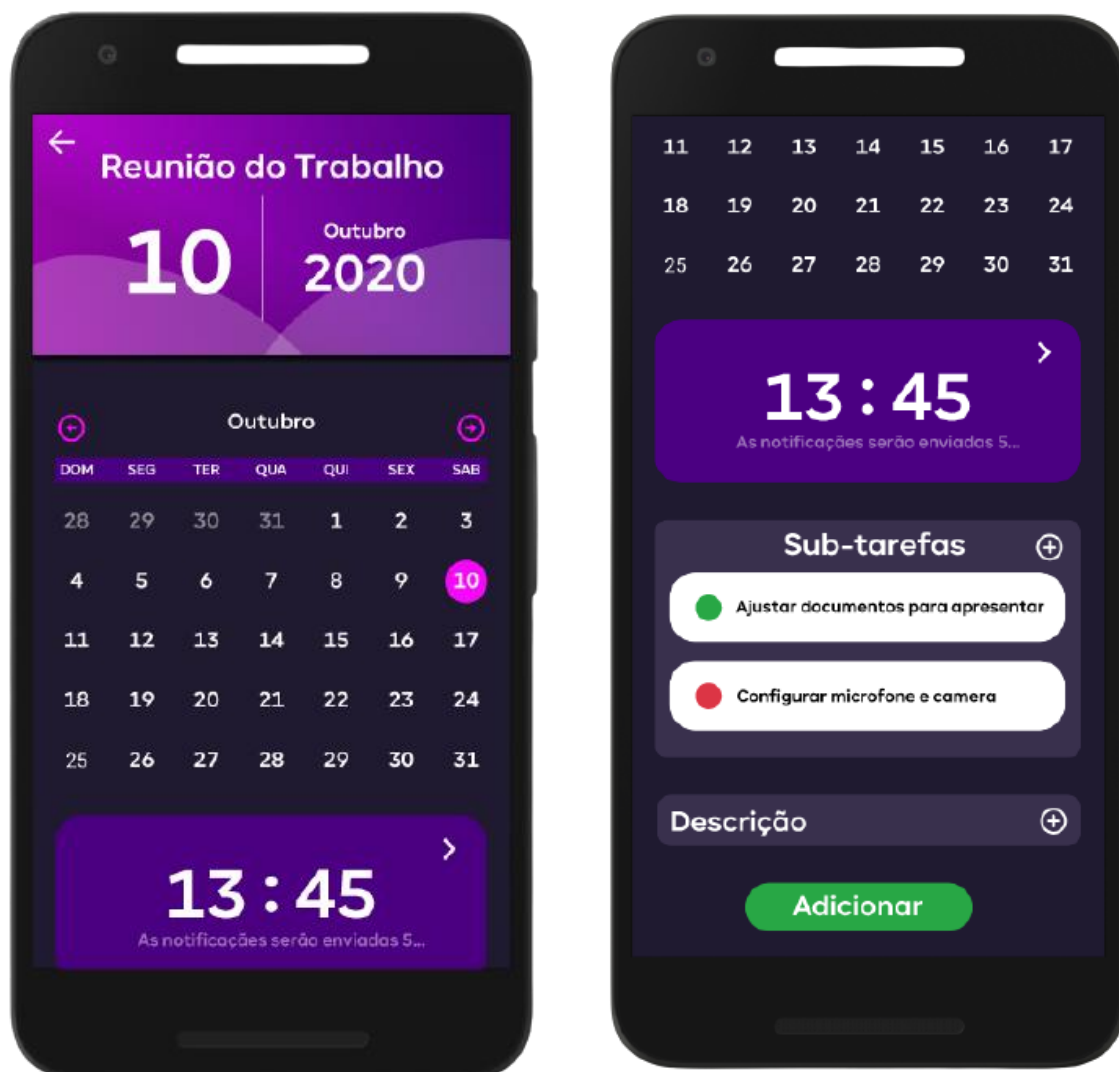
**Figura 8: interface (card da tarefa)**

. Na figura 8 se apresenta o painel principal da aplicação. Clicando sobre um dos

cards, no painel principal, é possível visualizar a tarefa com mais detalhes, em um modal, onde é mostrado com detalhe toda a descrição que foi dado para aquela tarefa.

Logo abaixo é visualizado as sub-tarefas, tarefas que estão diretamente ligadas a esta tarefa principal, ou seja, tarefas que precisam ser feitas antes ou até o dia marcado para a tarefa principal. Neste div de sub-tarefas, existem alguns círculos com cores, que definem a prioridade destas tarefas, onde vermelho representa alta prioridade, amarelo media prioridade e verde representando baixa prioridade

### 3.4.5. CADASTRO DE TAREFA





## **Figura 9 e 10: Interface (cadastro das tarefas)**

Assim que clicado no botão rosa com um ícone de mais na tela do painel principal, será redirecionado para esta tela, aqui é aonde todas as tarefas serão cadastradas. No cabeçalho desta tela, mostra o dia que você selecionou o mês e o ano, e acima destas informações, o título que pode ser alterado clicando em cima do mesmo. Para selecionar um mês, basta clicar nos ícones rosa acima do calendário, o da esquerda volta para os meses anteriores, e o da direita avança para os próximos meses. Para escolher um dia, basta clicar em cima dele calendário, e ficara com um círculo rosa para mostrar qual dia você clicou. Logo abaixo do calendário tem um card da cor roxa para escolher uma hora clicando no ícone de flecha na parte superior direita do card, em seguida, existe a opção para adicionar as sub-tarefas e suas prioridades, e por último a opção para colocar uma descrição, as duas últimas opções não são obrigatórias para a criação das tarefas

## **4 CONCLUSÃO**

Existiram alguns problemas durante o desenvolvimento prático deste projeto, a proposta de desenvolver com React Native foi devido a sua grande popularidade, e suas ferramentas potentes para desenvolvimento, sem contar suas milhares de bibliotecas que auxiliam em toda codificação, porém, ainda se trata de uma tecnologia muito utilizada mas recente, e a procura de alguns exemplos para o nicho escolhido (agenda de tarefas) acabaram se tornando um pouco mais complicadas, e isso demandou um pouco mais de tempo do que o planejado.

Uma outra dificuldade foi devido ao Android Studio, uma plataforma que simula o dispositivo android em uma desktop. A dificuldade veio por conta da instalação e configuração do software no desktop, pois as versões do Java, JRE e JDK que já haviam sido instaladas na máquina, acabaram dando conflito na hora da configuração do Android Studio.

Houve um problema na construção de códigos nativos para o sistema IOS, devido ao uso nativo do calendário data e hora, foi modificado alguns scripts

para a realização do mesmo, mas sem sucesso. Devido a isso, o uso do aplicativo se tornou restrito apenas para dispositivos com sistema operacionais android.

Algumas adversidades aconteceram, mas o desenvolvimento ainda sim, foi muito rápido e pratico devido ao grande numero de bibliotécas de animação, utilização de componentes, que tornam boa parte do código reutilizável, passando apenas propriedades entre eles para alteração visual ou mudança de dados e as bibliotécas de css e componentes já prontos que facilitam muito na hora de construir botão, inputs, labels etc..

Este projeto foi versionado utilizando o GitHub, e estará disponivel e publico para alterações da comunidade.

Após a finalização do desenvolvimento, foi concluído que a aplicação terá muito impacto na vida de seus usuários, por conta do grande feedback positivo que foi passado após o distruibuimento para testes.

#### **4.1. PROJETOS FUTUROS**

Os planos são de continuar com essa aplicação e modifica-la para que ela seja algo ainda maior que o proposto inicialmente com este trabalho, alguns planos para inclusão de tarefas com cards de DragAndDrop(arrastar e soltar) utilizando a api do Trello ou Jira, para sincronizar as tarefas no aplicativo com o site referente a api.

Alterações publicas serão bem vindas, sendo análisadas e se agregar de forma positiva para a aplicação serão incluídas, construindo assim, uma aplicação que atende a demanda de seus usuários.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

ALDABRA. **O crescimento do mercado de aplicativos na pandemia.** Disponível em <<https://aldabra.com.br/artigo/o-crescimento-do-mercado-de-aplicativos-na-pandemia>> Acesso em: 29 mar. 2021.

ALPHACODE. **As principais diferenças entre React Native e IONIC.** Disponível em <<https://site.alphacode.com.br/react-native/>> Acesso em: 29 mar. 2021.

BECKER, Lauro. **O que é React Native?** Disponível em <<https://www.organicadigital.com/blog/o-que-e-react-native/>> Acesso em: 29 mar. 2021.

CALOMENO, Adil. **React Native: O que é, quais as funcionalidades e vantagens desse framework.** Disponível em <<https://aldabra.com.br/artigo/o-crescimento-do-mercado-de-aplicativos-na-pandemia>> Acesso em: 29 mar. 2021.

DAKAR, Romulo. **MongoDB – O que é e para que serve?** Disponível em <<http://desenvolvedor.ninja/mongodb-o-que-e-e-para-que-serve/>> Acesso em: 02 abr. 2021. DEMARTINI, Felipe. **Mercado de apps deve movimentar US\$6,3 trilhões até 2021.** Disponível em <<https://canaltech.com.br/apps/mercado-de-apps-deve-movimentar-us-63-trilhoes-ate-2021-133229/>> Acesso em: 28 mar. 2021.

DEVMEDIA. **Introdução ao MongoDB.** Disponível em <<https://www.devmedia.com.br/introducao-ao-mongodb/30792>> Acesso em: 03 abr. 2021.

DEVMEDIA. **O que é um UML e Diagramas de Caso de Uso: Introdução Prática à UML.** Disponível em <<https://www.devmedia.com.br/introducao-ao-mongodb/30792>> Acesso em: 03 abr. 2021.

DIGITALHOUSE. **O crescimento do mercado de aplicativo do Brasil.** Disponível em <<https://www.digitalhouse.com/br/blog/crescimento-do-mercado-de-aplicativos-brasil>> Acesso em: 28 mar. 2021.

LENON. **Node.js – o que é, como funciona e quais as vantagens.** Disponível em <<https://www.opus-software.com.br/node-js/>> Acesso em: 04 mar. 2021.

LUCIDCHART. **O que é um diagrama de sequencia UML.** Disponível em <

<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml> >  
Acesso em: 20 mai. 2021.

LUCIDCHART. **O que é um diagrama de classe UML.** Disponível em <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>> Acesso em: 20 mai. 2021.

LUCIDCHART. **O que é um diagrama de caso de uso: O que é, como fazer e exemplos.** Disponível em <<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>> Acesso em: 20 mai. 2021.

MONGODB. **MongoDB Atlas Overview.** Disponível em <<https://docs.mongodb.com/realm/mongodb/>> Acesso em: 03 abr. 2021.

MUNDOJS. **Diferença entre React Native e React js.** Disponível em <<https://www.mundojs.com.br/2019/08/01/diferencas-entre-react-native-e-react-js/>> Acesso em: 02 abr. 2021.

MUNIZ, Flavio. **Vantagens de um banco de dados NoSQL, MongoDB.** Disponível em <<https://blog.4linux.com.br/vantagens-de-um-banco-de-dados-nosql-mongodb/>> Acesso em: 03 abr. 2021.

NODE. **Sobre Node.js.** Disponível em <<https://nodejs.org/pt-br/about/>> Acesso em: 04 mar. 2021.

PAIVA, Fernando. **Corona vírus aumenta o uso de aplicativos moveis.** Disponível em <<https://www.mobiletime.com.br/noticias/17/03/2020/coronavirus-provoca-aumento-no-uso-de-apps-movei>> Acesso em: 28 mar. 2021.

REDHAT. **O que é API?** Disponível em <<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>> Acesso em: 04 abr. 2021.





STACKOVERFLOW. **2020 Developer Survey.** Disponível em <<https://insights.stackoverflow.com/survey/2020#overview>> Acesso em: 03 abr. 2021.

TAKE. **API: conceito, exemplo de uso, e importância da integração para desenvolvedores.** Disponível em < <https://www.take.net/blog/tecnologia/api-conceito-e-exemplos>> Acesso em: 03 abr. 2021.

VENTURA, Plínio. **Entendendo Diagrama de Classe da UML.** Disponível em < <https://www.ateomomento.com.br/uml-diagrama-de-classes/> > Acesso em: 20 mai. 2021.