

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

AGENDA INTERATIVA PARA AUXÍLIO EM TAREFAS
DOMÉSTICAS BÁSICAS

YURI TRIERVEILER

BLUMENAU
2022

YURI TRIERVEILER

**AGENDA INTERATIVA PARA AUXÍLIO EM TAREFAS
DOMÉSTICAS BÁSICAS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof^a. Luciana Pereira de Araújo Kohler, Mestra – Orientadora

**BLUMENAU
2022**

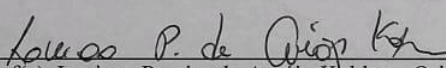
**AGENDA INTERATIVA PARA AUXÍLIO EM TAREFAS
DOMÉSTICAS BÁSICAS**

Por

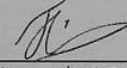
YURI TRIERVEILER

Trabalho de Conclusão de Curso aprovado
para obtenção dos créditos na disciplina de
Trabalho de Conclusão de Curso II pela banca
examinadora formada por:

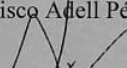
Presidente:


Prof(a). Luciana Pereira de Araújo Kohler – Orientador(a), FURB

Membro:


Prof(a). Francisco Adell Péricás – FURB

Membro:


Prof(a). Simone Erbes da Costa – FURB

Blumenau, 07 de julho de 2022

Dedico este trabalho a minha família, minha namorada, colegas do curso, amigos e professores que contribuíram para a realização deste trabalho e sua conclusão.

AGRADECIMENTOS

Primeiramente gostaria de agradecer à minha família, começando pelos meus pais, pelas possibilidades e portas que me abriram e me permitiram adquirir todo o conhecimento e a base necessária para o desenvolvimento deste trabalho.

À minha namorada, que sempre se mostrou otimista e compreensiva durante o tempo que tive que investir para o desenvolvimento, sendo algumas vezes o pouco tempo que podemos passar juntos.

Aos meus amigos e colegas, que me auxiliaram com conhecimentos quando tive dificuldades em progredir por conta e me proporcionaram momentos de descontração, para aliviar o estresse envolvido no desenvolvimento do trabalho.

E por fim à minha orientadora, Luciana Pereira de Araújo Kohler, pelo trabalho impecável durante o acompanhamento deste projeto, estando sempre disponível para tirar dúvidas e auxiliar no desenvolvimento do mesmo.

Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que há muito a fazer.

Alan Turing

RESUMO

Este trabalho apresenta o desenvolvimento de um aplicativo para registro de atividades do cotidiano dos usuários, apresentando-os em uma agenda simples e que seja de fácil utilização. Diante da necessidade identificada de se ter uma ferramenta que auxilie a lembrar de atividades rotineiras, este trabalho focou em oferecer um aplicativo móvel que permite aos usuários realizarem o cadastro de atividades que desejam, de maneira fácil e rápida, para conseguirem um melhor controle sobre suas atividades do cotidiano. O aplicativo foi desenvolvido utilizando a linguagem de programação Javascript, com o *framework* React Native, permitindo a utilização tanto no sistema operacional Android quanto iOS. Como banco de dados, a ferramenta Firestore Database, da Firebase e o editor de código-fonte o Visual Studio Code. O aplicativo desenvolvido foi testado por 10 avaliadores voluntários que se prontificaram a baixar o aplicativo e realizar alguns testes propostos. Os avaliadores responderam questionários de perfil e de escala de usabilidade que no final alcançou resultados positivos na questão de boa usabilidade dentro das questões apresentadas no questionário.

Palavras-chave: Agenda. Aplicativo móvel. Atividades do cotidiano. React Native.

ABSTRACT

This work presents the development of an application to record users' daily activities, presenting them in a simple and easy-to-use agenda. Faced with the identified need for a tool that helps us remember routine activities, this work focused on offering a mobile application that allows users to easily and quickly register the activities that they want to achieve better control over their activities. The application was developed using the Javascript programming language, with the React Native framework, allowing its use on both Android and iOS operating systems. As a database, Firebase's Firestore Database tool and source code editor is Visual Studio Code. The developed application was tested by 10 volunteer evaluators who volunteered to download the application and carry out some proposed tests. The evaluators answered profile and usability scale questionnaires, which in the end reached positive results in the question of good usability within the questions presented in the questionnaire.

Key-words: Agenda. Everyday activities. Mobile app. React Native.

LISTA DE FIGURAS

Figura 1 - Quantidade total de celulares no município de Blumenau	14
Figura 2 - Diagrama de casos de uso	21
Figura 3 - Diagrama de atividades.....	22
Figura 4 - Mapa mental do banco não relacional	23
Figura 5 - Ferramentas de desenvolvimento	24
Figura 6 - Interface do Expo Go.....	25
Figura 7 - Arquitetura do aplicativo	25
Figura 8 - Modo de autenticação de usuário.....	26
Figura 9 - Tela de cadastro com e-mail incorreto.....	28
Figura 10 - Tela Geral do aplicativo.....	29
Figura 11 - Tela de Informações e listagem de eventos	33
Figura 12 - Informações salvas no banco de dados	35
Figura 13 - Faixa etária, gênero e utilização de outros aplicativos	38
Figura 14 - Tarefas cotidianas, frequência com aplicativos e sistema operacional.....	39
Figura 15 - Seção 1 do formulário.....	44
Figura 16 - Seção 2 do formulário - Termo de Consentimento.....	45
Figura 17 - Primeira parte da Seção 3 do formulário	46
Figura 18 - Segunda parte da seção 3 do formulário	47
Figura 19 - Primeira parte da seção 4 do formulário	48
Figura 20 - Segunda parte da seção 4 do formulário	49
Figura 21 - Primeira parte da seção 5 do formulário	50
Figura 22 - Segunda parte da seção 5 do formulário	50
Figura 23 - Terceira parte da seção 5 do formulário	51
Figura 24 - Quarta parte da seção 5 do formulário.....	51

LISTA DE QUADROS

Quadro 1 - Funcionalidades do protótipo	16
Quadro 2 - Exemplo de relatório	17
Quadro 3 - Matriz de Rastreabilidade dos RFs.....	20
Quadro 4 - Requisitos não funcionais do sistema	20
Quadro 5 - Diretório TelaCadastro.js	27
Quadro 6 - Fonte do ModalItemAgenda - Salvar informações	30
Quadro 7 - Diretório ModalItemAgenda - função de exclusão	31
Quadro 8 - Diretório TelaGeral.tsx - definição de abas	32
Quadro 9 - Diretório ModalItemAgenda - identificação de repetidos	33
Quadro 10 - Diretório firebaseconfig.js.....	34
Quadro 11 - Comparativo entre trabalhos correlatos e aplicativo	36
Quadro 12 - Relação das perguntas com as porcentagens de cada resposta.....	40
Quadro 13 - Descrição detalhada do caso de uso UC02 - Manter Login	52
Quadro 14 - Descrição detalhada do caso de uso UC03 - Manter eventos.....	52

LISTA DE ABREVIATURAS E SIGLAS

AIML – Artificial Intelligence Mark-up Language

API – Application Programming Interface

App – Aplicativo

IHC – Interação Humano Computador

RF – Requisito Funcional

RNF – Requisito Não Funcional

SUMI – Software Usability Measurement Inventory

UC – Uses Case

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS.....	13
1.2 ESTRUTURA.....	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 AVANÇO TECNOLÓGICO E SEUS IMPACTOS NA SOCIEDADE	14
2.2 INTERAÇÃO HUMANO COMPUTADOR.....	15
2.3 TRABALHOS CORRELATOS.....	15
2.3.1 PROTÓTIPO DE UM ASSISTENTE PESSOAL DIGITAL INTELIGENTE.....	16
2.3.2 EATING AND DRINKING RECOGNITION FOR TRIGGERING SMART REMINDERS.....	17
2.3.3 TIMEFUL	18
3 DESENVOLVIMENTO	19
3.1 LEVANTAMENTO DE INFORMAÇÕES.....	19
3.2 ESPECIFICAÇÃO	20
3.2.1 Diagrama de caso de uso do aplicativo	20
3.2.2 Diagrama de atividades	22
3.2.3 Estrutura de banco de dados não relacional	23
3.3 IMPLEMENTAÇÃO	23
3.3.1 Técnicas e ferramentas utilizadas.....	23
3.3.2 Operacionalidade da implementação	26
3.4 RESULTADOS E DISCUSSÕES.....	35
3.4.1 Comparação com os trabalhos correlatos.....	35
3.4.2 Análise de usabilidade.....	36
3.4.3 Resultados	37
4 CONCLUSÕES.....	41
4.1 EXTENSÕES	42
REFERÊNCIAS	43
APÊNDICE A – FORMULÁRIO ENCAMINHADO PARA VERIFICAÇÃO DA USABILIDADE DO APLICATIVO	44
APÊNDICE B – DESCRIÇÃO DE CASOS DE USO	52

1 INTRODUÇÃO

Há anos tem-se experimentado um avanço tecnológico cada vez maior na sociedade. Vive-se uma revolução tecnológica constante por décadas que ainda está apenas no começo (FORBES, 2019). Estes avanços são motivados pelos mais variados motivos e percebe-se a facilidade que isso traz no dia a dia. Hoje em dia é possível realizar tarefas com apenas um celular na mão, coisa que antigamente poderia demandar horas investidas em apenas uma atividade.

Tem-se percebido um ganho no tempo que as pessoas possuem em seu cotidiano, diminuindo o tempo que gastam com atividades corriqueiras, permitindo-os fazer mais coisas em um dia. Conforme afirma Toscan (2019, p. 1), “[...] a tecnologia, em constante desenvolvimento, possui ferramentas e/ou programas computacionais que podem auxiliar na economia e gestão do tempo.”.

Em contrapartida, o avanço tecnológico desenfreado é um empecilho para aqueles que não conseguem acompanhá-lo. Vê-se isso frequentemente em pessoas de terceira idade, por exemplo. Essa dificuldade pode ser atribuída às diferentes gerações e meios que viveram em sua infância, pois, é na infância que se tem a maior capacidade de aprendizagem (BRANCO, 2018). Essa dificuldade é ainda mais acentuada quando a tecnologia está presente em atividades que deveriam ser simples, porém com o excesso de novas funcionalidades embutidas, tem-se apenas um programa complicado e não intuitivo.

Seguindo o raciocínio, vê-se uma crescente dificuldade de pessoas em manter suas atividades do dia a dia organizadas, sendo ainda perceptível em pessoas de mais idade, seja por esquecimento ou negligência. Esse grupo é considerado frágil e a tecnologia deveria prestar mais assistência, porém este mesmo grupo tem grandes dificuldades de conseguir utilizar as tecnologias atuais (Gomes *et al.*, 2019).

Por conta deste cenário, este trabalho tem o intuito de desenvolver um aplicativo móvel que serve como uma agenda interativa. Sua aplicabilidade é focada em atividades do cotidiano, podendo oferecer lembretes em atividades frequentes, como por exemplo, a troca do gás. O aplicativo utiliza as atividades como meio para moldar um perfil e prever futuras necessidades, podendo advertir o usuário caso identifique alguma rotina que poderá vir se repetir em breve. Sua apresentação se dá por um calendário no qual o usuário pode consultar e/ou cadastrar suas atividades e lembretes, facilitando a consulta e permitindo a visualização do resultado obtido pelo aplicativo ao estudar a rotina do usuário.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um aplicativo em formato de agenda para auxiliar no controle de atividades do dia a dia.

Os objetivos específicos são:

- a) analisar e avaliar as interfaces desenvolvidas baseadas no questionário Software Usability Measurement Inventory (SUMI) fornecendo um leiaute amigável.
- b) facilitar o agendamento de tarefas rotineiras com base nos agendamentos já realizados.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos, sendo que cada capítulo possui seções e subseções. O primeiro capítulo é a introdução do trabalho, apresentando a motivação por trás do desenvolvimento do projeto, os objetivos definidos e a apresentação da estrutura geral do trabalho.

O segundo capítulo é composto pela fundamentação teórica, sendo dividida em quatro seções. Na primeira seção é descrito sobre o avanço tecnológico dos últimos anos e todo o impacto que esse avanço causou na sociedade em que vivemos. A segunda seção aborda sobre as dificuldades encontradas na utilização das tecnologias por pessoas da terceira idade. A terceira relata sobre a interação humano-computador e na quarta seção são apresentados os trabalhos correlatos com a descrição de cada um.

No terceiro capítulo é apresentado todo o desenvolvimento do projeto, em quatro seções. Começando pelo levantamento de requisitos na primeira seção. Na segunda seção tem-se a especificação e o Diagrama de Casos de Uso. Na terceira seção foi detalhado o desenvolvimento do aplicativo, além de apresentar as tecnologias e conhecimentos utilizados. Na quarta seção, são discutidos os resultados obtidos.

Por fim, no quarto capítulo é apresentada a conclusão deste trabalho. Nele estão descritas as limitações e contribuições alcançadas, como também as sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas as fundamentações que o trabalho se estruturou para ser desenvolvido. Na seção 2.1 é aprofundado sobre o tema da evolução tecnológica e os impactos disso na sociedade. Na seção 2.2 é apresentado sobre Interação Humano Computador (IHC). Por fim, na seção 2.3 são apresentados e discutidos os trabalhos relacionados a este.

2.1 AVANÇO TECNOLÓGICO E SEUS IMPACTOS NA SOCIEDADE

Nas últimas décadas, a sociedade está passando por uma constante revolução tecnológica, em que se vê dia após dia o acesso à informação tornar-se mais fácil e rápido (FORBES, 2019). O acesso da informação fácil influenciou e muito a necessidade de melhorar as tecnologias que são porta de entrada para esse mundo, como por exemplo o celular.

Com a informação sobre o aumento da quantidade de dados que trafegam na internet, se percebe um crescimento alarmante utilizando apenas os últimos anos como base, vendo um número assustador de mais de 12 bilhões de dispositivos habilitados para conexão móvel. Este número é estimado entre 2019 e 2022 e somente este período de 3 anos, é superior a todo o período de 32 anos anteriores juntos (FORBES, 2019).

Esse aumento tornou cada vez mais comum as pessoas terem um *smartphone* nas atividades do cotidiano, auxiliando a realizar praticamente qualquer tarefa e permitindo otimizar o tempo, organizar a rotina, entre muitas outras aplicabilidades (VITORIA, 2020). Como pode-se ver na Figura 1, baseado nos dados fornecidos pela Anatel, só no município de Blumenau tem-se uma densidade de aparelhos telefônicos superiores a quantidade de residentes (366.418 mil habitantes em 2021, conforme o IBGE, 2021) na cidade.

Figura 1 - Quantidade total de celulares no município de Blumenau

Celulares em BLUMENAU - SC									
Celulares: 406.078									
Densidade: 112,22 cel/100 hab.									
Market Share:			Modalidade:			Tecnologia:			
VIVO	169.477	41.74%	Pós-pago	250.055	61.58%	4G	357.330	88.00%	
TIM	137.529	33.87%	Pré-pago	156.023	38.42%	3G	24.703	6.08%	
CLARO	60.203	14.83%	Tipo de Pessoa:			2G	24.045	5.92%	
OI	31.637	7.79%				Pessoa Física	334.903	82.47%	M2M:
SURF	7.228	1.78%	Pessoa Jurídica	71.175	17.53%	M2M	14.401	3.55%	
							PONTO_DE_SERVICO	8.810	2.17%

Fonte: Teleco (2021).

Por mais que se tenha uma quantidade grande de benefícios, há impactos negativos nesta mudança, que podem ser vistos acontecendo diariamente na vida das pessoas. A necessidade

de se adequar a esta nova sociedade está cada vez mais evidente, conforme ela segue evoluindo. Contudo, há aqueles que possuem receio ou dificuldade de seguir este processo de evolução (BUCHEER, 1998).

2.2 INTERAÇÃO HUMANO COMPUTADOR

A Interação Humano-Computador é uma área que estuda o usuário dentro dos sistemas de computação.

Esse conceito vai além da estética da tela (disposição de menus, cores, etc.). Ele se estende a questões como corretude, tempo de resposta, grau de dificuldade de uso, rapidez no desempenho de tarefas-chave, nível de erros cometidos por usuários durante o uso, facilidade de aprendizado, fadiga produzida pelo uso prolongado, acomodação de usuários com necessidades especiais, entre outros. (OLIVEIRA; OLIVEIRA, 2015, p. 11).

Para conseguir desenvolver sistemas que satisfaçam a maior quantidade de pessoas possíveis, é preciso se importar com pontos que possam gerar qualquer problema de compreensão por parte dos usuários, tentando diminuir ao máximo a quantidade de problemas que o usuário pode vir a ter (OLIVEIRA; OLIVEIRA, 2015).

Não é errado assumir que um programa ou tecnologia que descarte esses conceitos, possui grandes chances de não ser um sucesso. Ao analisar algumas tecnologias existentes no dia a dia é possível perceber a IHC aplicada em diversos detalhes, como por exemplo no gesto que realizado para ampliar uma fotografia, ao afastar dois dedos, criando um espaço entre eles (OLIVEIRA; OLIVEIRA, 2015).

A aplicabilidade da IHC não se limita apenas à utilização de usuários típicos de sistemas. Quando analisado no âmbito empresarial, se percebe um crescente investimento no desenvolvimento de sistemas simples e intuitivos, que não sejam um desafio para os usuários entenderem e que consigam realizar as tarefas propostas pelo sistema. Quanto menos tempo o usuário levar para aprender, menor será o investimento que uma empresa terá para treiná-lo (OLIVEIRA; OLIVEIRA, 2015).

2.3 TRABALHOS CORRELATOS

São apresentados três trabalhos com característica semelhantes a este trabalho. O trabalho apresentado na subseção 2.3.1 se trata de um protótipo de assistente pessoal digital inteligente (TOSCAN, 2019). Na subseção 2.3.2, o desenvolvimento de uma ferramenta que identifica por sensores quando pessoas de terceira idade realizam o ato de comer e beber e agendam lembretes para notificá-las caso não o façam novamente em certo período (Gomes et al., 2019). E na subseção 2.3.3, uma ferramenta adquirida pelo Google, que auxilia os usuários

a organizarem melhor o seu dia a dia através de uma análise que o aplicativo faz para auxiliar o usuário (GANNES, 2014). Esses trabalhos são descritos nas subseções 2.4.1, 2.4.2 e 2.4.3 respectivamente.

2.3.1 PROTÓTIPO DE UM ASSISTENTE PESSOAL DIGITAL INTELIGENTE

O protótipo do assistente pessoal digital inteligente foi desenvolvido com o intuito de auxiliar e trazer mais comodidade para o dia a dia das pessoas, prestando assistência pessoal básica no âmbito residencial. A integração do assistente com os indivíduos ocorre por voz. O protótipo recebe uma gravação de no máximo oito segundos para então interpretá-la e gerar um texto coerente com o áudio de entrada. Esta interpretação é feita pelo interpretador Artificial Intelligence Mark-up Language (AIML) (TOSCAN, 2019).

Para repassar alguma informação para o protótipo, é preciso pressionar um botão específico no aparelho e então segurá-lo pelo período informado anteriormente e passar assim a instrução desejada. Com isso, o protótipo possui algumas funcionalidades que podem ser vistas no Quadro 1.

Quadro 1 - Funcionalidades do protótipo

Funcionalidade	Descrição
Agendar compromissos	Através da integração com um calendário, o usuário deve ser capaz de agendar compromissos para que seja lembrada posteriormente pela interface.
Escrever listas	Escrever e salvar listas ditadas pelo usuário, como listas de compras para o supermercado, papelaria etc.
Escrever notas	Escrever e salvar pequenas notas que sejam ditadas pelo usuário.
Fornecer informações gerais	Fornecer assistência sobre informações gerais que podem ser obtidas de forma imediata com uma simples pesquisa em um site buscador, bem como informações sobre a previsão do tempo etc.
Lembretes ou alarmes	Ser capaz de lembrar o usuário de tarefas previamente programadas pelo próprio usuário, além de pequenos lembretes gerais, como por exemplo, a data de aniversário de alguém importante para o usuário, reuniões.

Fonte: Toscan (2019).

Considerando suas funcionalidades, o dispositivo estará sempre funcionando e aguardando uma entrada do usuário. Diariamente o usuário receberá um e-mail com as informações do dia em questão. O e-mail terá um formato de relatório pré-definido, conforme ilustra a Quadro 2.

Quadro 2 - Exemplo de relatório

Olá Luiz, estas são as atividades que você possui programado:

Para hoje (10 de junho de 2019):

- Dentista às 14h30
- Mercado

Para a semana (08 de junho de 2019 à 14 de junho de 2019):

- Reunião às 15:30 na Quinta-feira dia 11 de junho de 2019

Pela previsão parece que hoje teremos um dia nublado e com probabilidade de chuva. É bom levar o guarda-chuva.

Fonte: Toscan (2019)

Além de seu relatório diário, na primeira interação com o usuário, o assistente informará todos os compromissos agendados no dia, após receber uma saudação do usuário. Estas informações serão apresentadas de maneira aleatória, com algumas frases definidas, para não ser um processo repetitivo.

De acordo com a proposta do trabalho, pôde ser tratado como um sucesso utilizando a base padrão do chatterbot A.L.I.C.E e a linguagem AIML, porém entendendo as limitações do dispositivo e das ferramentas. Como a linguagem natural é muito variada e é necessário definir alguns comandos como padrão para iniciar certos processos, quando não se consegue alcançar esses padrões, o dispositivo irá se perder e não conseguirá se comunicar com o usuário. Toscan (2019) informou que essa limitação poderia ser superada implementando ao dispositivo algoritmos de aprendizagem de máquina, fornecendo a ele um banco de conhecimento expansível e com o tempo ele se adaptaria a mais expressões (TOSCAN, 2019).

2.3.2 EATING AND DRINKING RECOGNITION FOR TRIGGERING SMART REMINDERS

O trabalho de Gomes et al. (2019) teve o objetivo de desenvolver uma ferramenta que consiga identificar o ato de comer e beber no dia a dia de maneira prática e que não dependa do usuário. Seu objetivo foi baseado na negligência de necessidades básicas de pessoas de mais idade. Com essa ferramenta deseja-se auxiliar esse grupo etário a manter-se saudável e ter uma melhor qualidade de vida.

Apoiando-se na popularização de smartphones e *wearable devices*, como o smartwatch, Gomes et al. (2019) sugere a utilização dos sensores já existentes nestes dispositivos para captar as atividades de comer e beber separadamente, utilizando estas informações para gerar alertas e notificações para os idosos ou seus cuidadores. Gomes et al. (2019) informa que é possível

encontrar bastante conteúdo sobre algoritmos que captam estas atividades, porém sem isolá-las, tratando o ato de comer e de beber como se fosse o mesmo (Gomes et al. 2019).

A primeira dificuldade encontrada foi conseguir captar e diferenciar estas atividades, utilizando os gestos das pessoas como informações de entrada para tentar diferenciar uma atividade da outra. Por padrão, alguns gestos precisam ser feitos para realizar a atividade de ingestão de alimento, como por exemplo, levar a comida para a boca ou manter um copo próximo da boca, como no ato de beber.

Contudo, estes gestos podem ser interrompidos por outros gestos esporádicos, por exemplo, quando alguém está sentado na mesa comendo, pode estar realizando outras atividades paralelamente, impactando na captação desta ação. Por conta disso, Gomes et al. (2019) testou algumas faixas de tempo para a captação dos gestos e percebeu que utilizando janelas de 10 segundos entre ações, é possível ter uma porcentagem satisfatória no resultado. Para aperfeiçoar ainda mais, foi incluso nos sensores uma validação de tempo mínimo de movimentos repetitivos de cinco minutos, considerando que uma refeição não pode ser feita em tempo menor que este, além de captar aumentos de batimentos cardíacos, provocados pelo processo de alimentação. Essa validação foi suficiente para conseguir isolar o ato de comer, porém quando foi validado com o ato de beber, tornou-se ambíguo.

Deste modo, um parâmetro de 5 segundos foi incluso no movimento de levar algo para próximo da boca e voltar o gesto para o ponto de origem, conseguindo isolá-lo de maneira mais eficiente (Gomes et al, 2019).

2.3.3 TIMEFUL

O Timeful é um aplicativo para aparelhos móveis desenvolvido originalmente para iOS, com o intuito de auxiliar os seus usuários a conseguirem administrar a sua vida de maneira mais eficiente (GANNES, 2014). Inspirados principalmente em administrar melhor o tempo no dia a dia, Dan Ariely em conjunto com Jacob Bank e Yoav Shoham, desenvolveram o Timeful, que se assemelhava a um calendário para iPhone (GANNES, 2014).

Seguindo a premissa que o ser humano possui por volta de duas horas de pico cognitivo, Gannes (2014) se questionava sobre o que as pessoas costumam fazer com esse tempo. Considerando que esse pico ocorria por volta das duas horas após acordar completamente pela manhã, percebeu-se que era comum que as pessoas gastassem esse tempo precioso em coisas corriqueiras, como responder e-mail, checar o Facebook, beber café etc.

Não sendo apenas um calendário, o aplicativo servia como auxílio para os usuários administrarem melhor os tempos do cotidiano. O aplicativo inicialmente importa um calendário

já existente do usuário para então, com base no que o usuário já faz, começar a realizar sugestões que seriam o melhor para ele naquele determinado momento.

O usuário por sua vez, possui algumas ações para que o aplicativo funcione corretamente. A primeira é aceitar ou rejeitar a proposta do aplicativo, gerando assim um feedback que auxilia a inteligência do aplicativo a entender o que é ou não possível para o usuário. A segunda é o ato de fazer o proposto pelo aplicativo, tornando deste modo sua vida melhor. Por fim, o usuário pode realizar a inclusão de suas atividades em seu calendário gerenciado pelo Timeful, informando ao aplicativo qual é sua prioridade e o próprio programa se encarrega do resto, ajustando a ordem das atividades com base no tempo que cada uma delas demanda e enquadrando isso com atividades já existentes para o usuário. Atualmente, o aplicativo é propriedade do Google, que o comprou em 2015, para implementar funcionalidades inteligentes no aplicativo Google Agenda (GANNES, 2014).

3 DESENVOLVIMENTO

Neste capítulo são apresentados os detalhes do desenvolvimento do aplicativo. Na seção 3.1 é apresentado o levantamento de informações do aplicativo. Na seção 3.2 é apresentada a especificação do aplicativo desenvolvido, mostrando os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF). Na mesma seção também é apresentado os diagramas de Casos de Uso (UC) e de arquitetura. A seção 3.3 mostra as técnicas e ferramentas utilizadas para o desenvolvimento do aplicativo, mostrando os códigos fontes e exemplos de utilização do aplicativo. E a seção 3.4 apresenta os resultados obtidos com o desenvolvimento do trabalho e a análise deles.

3.1 LEVANTAMENTO DE INFORMAÇÕES

Este trabalho apresenta o desenvolvimento de um aplicativo para Android e iOS, que permite ao usuário realizar o cadastro de atividades de seu dia a dia e utilizá-lo como uma agenda, gerenciando seus compromissos e lembretes que desejar. Além disso, o aplicativo alerta e sugere ao usuário compromissos semelhantes que se repetem em determinado período, auxiliando-o a controlar melhor o seu dia a dia.

O aplicativo permite a realização de cadastro de usuários, utilizando e-mail e senha como padrão e após realizar o *login*, possui acesso completo às funcionalidades do aplicativo, como acesso a agenda, verificar os compromissos já cadastrados e realizar o cadastro de novos eventos ou exclusão de eventos que já não deseja mais. Ainda, o aplicativo apresenta uma listagem completa por ordem de todos os compromissos que o usuário já cadastrou na agenda.

O aplicativo apresenta sugestões de inclusão de eventos ao usuário quando identifica eventos que possuem descrições idênticas e se repetem em determinado período. Ele pergunta se o usuário deseja que o aplicativo preencha de maneira automática futuros eventos respeitando as descrições utilizadas e o intervalo de tempo que o usuário repetiu o evento em questão.

3.2 ESPECIFICAÇÃO

Nesta seção é apresentada a especificação do aplicativo, mostrando os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) do aplicativo, como também o diagrama de Uses Cases (UC), encontrado na subseção 3.2.1. Para o desenvolvimento do diagrama, foi utilizada a ferramenta web Draw.io.

O Quadro 3 apresenta os RFs do aplicativo junto dos casos de uso relacionados a cada requisito. No Quadro 4 estão apresentados os RNFs do aplicativo desenvolvido neste trabalho.

Quadro 3 - Matriz de Rastreabilidade dos RFs

Requisitos Funcionais	Casos de Uso
RF01: O aplicativo deve permitir a criação de usuários	UC01
RF02: O aplicativo deve permitir realizar login com o usuário cadastrado	UC01
RF03: O aplicativo deve permitir ao usuário manter atividades no calendário	UC02
RF04: O aplicativo deve permitir ao usuário o aceite de sugestões feitos pelo aplicativo.	UC03
RF05: O aplicativo deve apresentar todos os eventos cadastrados em uma listagem	UC04
RF06: O aplicativo deve permitir realizar o logout	UC01
RF07: O aplicativo deve apresentar sugestões para o usuário	UC05

Fonte: elaborado pelo autor.

Quadro 4 - Requisitos não funcionais do aplicativo

Requisitos Não Funcionais
RNF01: O aplicativo deve ser desenvolvido em React Native
RNF02: O aplicativo deve ser desenvolvido utilizando o Visual Studio 2019
RNF03: O aplicativo deve permitir o aceite de sugestões do aplicativo
RNF04: O aplicativo deve utilizar o Firebase como banco de dados
RNF05: O aplicativo apenas apresentará os compromissos do usuário autenticado
RNF06: O aplicativo funcionará apenas em plataformas móveis

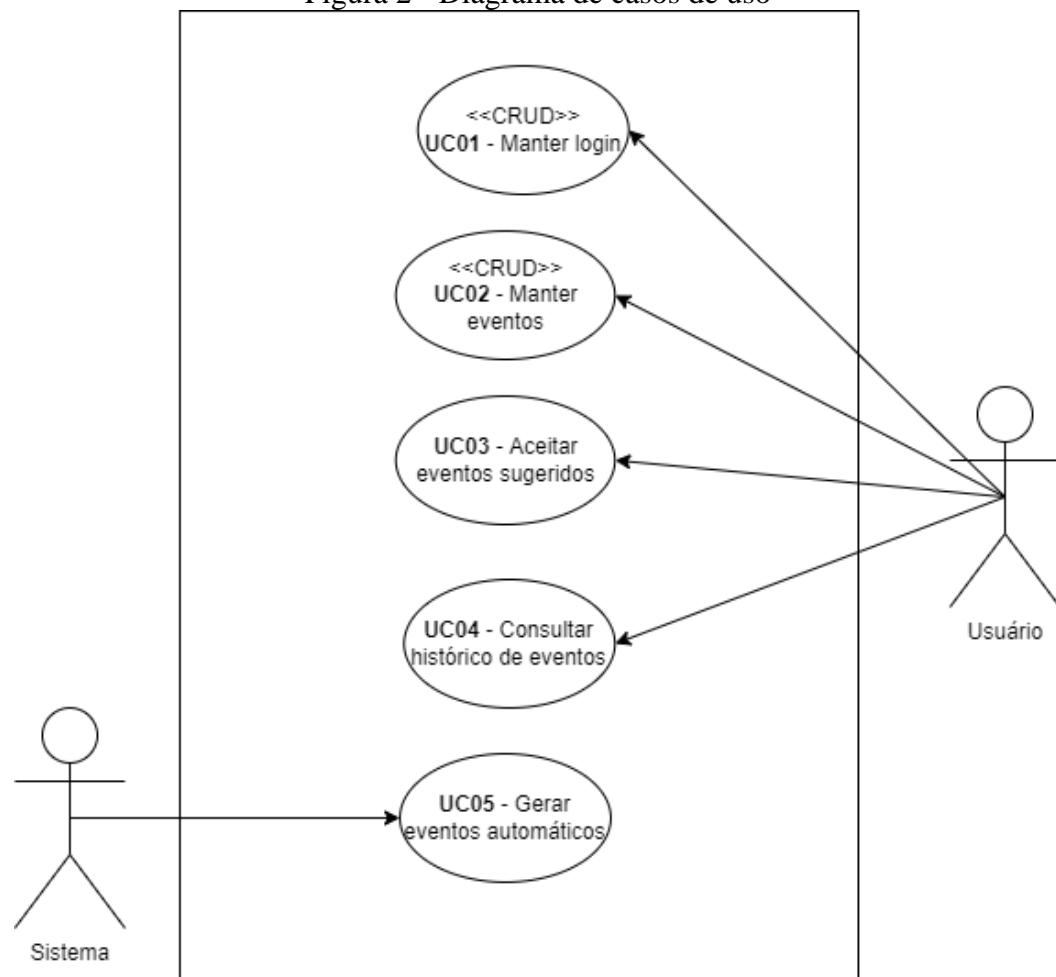
Fonte: elaborado pelo autor.

3.2.1 Diagrama de caso de uso do aplicativo

A Figura 2 representa o diagrama de casos de uso do aplicativo desenvolvido neste trabalho. No diagrama é apresentado o ator único *Usuário*, este que é responsável por efetuar todas as ações possíveis dentro do aplicativo. O *Usuário* fará primeiramente a criação de usuário conforme UC01 - Manter login, para então conseguir realizar a autenticação do

usuário e realiza o login na aplicação. Caso o Usuário não realize o *logout* no momento que fechar o aplicativo, o aplicativo irá acessar o último usuário autenticado. Na tela inicial, o Usuário poderá cadastrar eventos conforme descrito no UC02 - Manter eventos, além de poder excluí-los e alterá-los. Após o Usuário possuir uma série de eventos cadastrados e estes se repetirem após 2 vezes em um período constante, o ator Sistema irá sugerir a inclusão de eventos iguais no mesmo intervalo que o evento já se repetiu anteriormente conforme UC05 - Gerar eventos automáticos, no momento que o Usuário realizar a inclusão do terceiro evento. Esta inclusão só ocorrerá se o Usuário aceitar a sugestão do aplicativo, demonstrado no UC03 - Aceitar evento sugeridos. Além disso, o Usuário poderá consultar todos os eventos que já cadastrou no aplicativo em uma aba separada, recebendo-os por ordem crescente, começando em datas anteriores, de acordo com o UC04 - Consultar histórico de eventos. Por fim, o Usuário poderá realizar o logout caso deseje acessar o aplicativo com um novo usuário, conforme UC01 - Manter login ou apenas encerrar a aplicação para manter o usuário logado.

Figura 2 - Diagrama de casos de uso

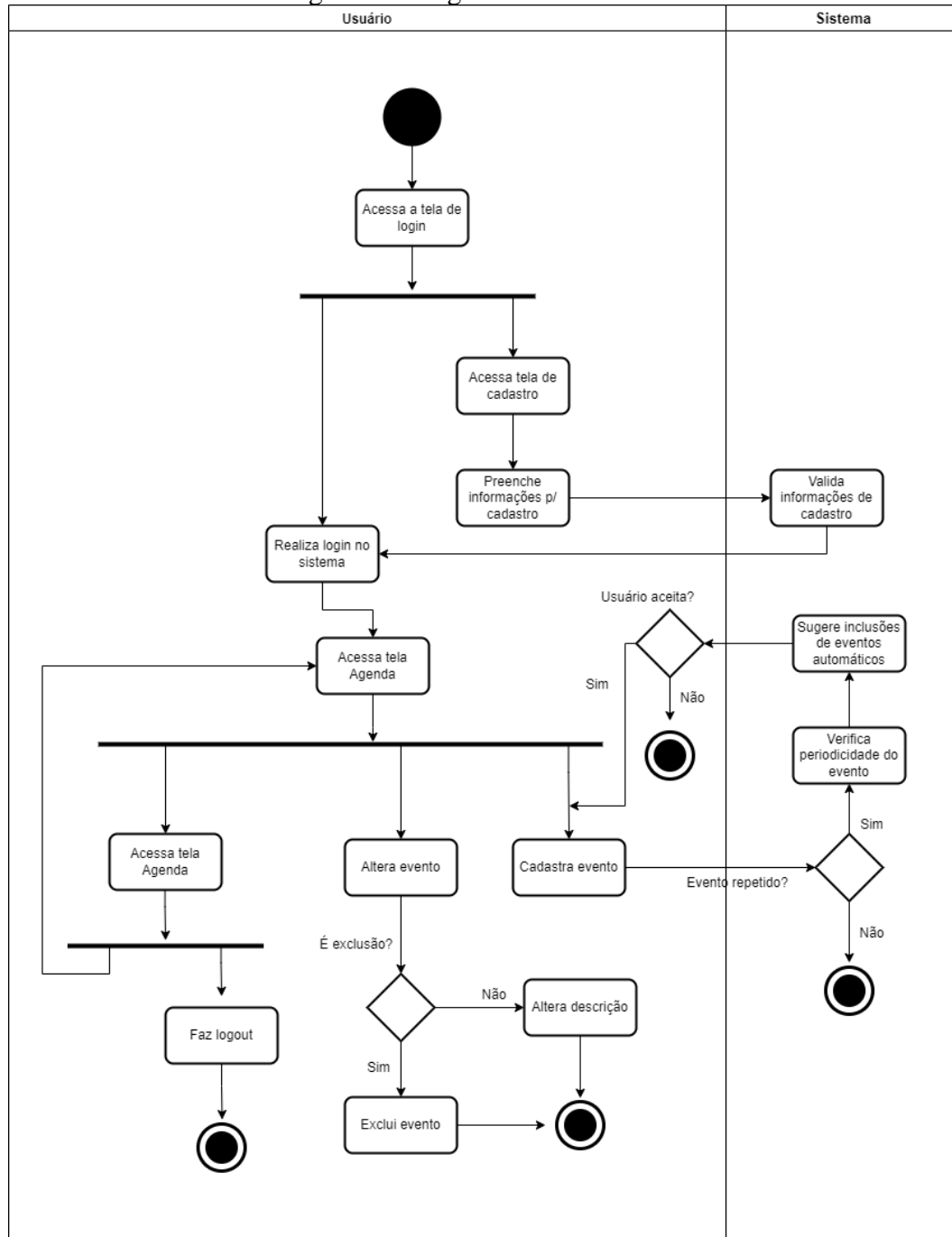


Fonte: elaborado pelo autor.

3.2.2 Diagrama de atividades

A Figura 3 representa o diagrama de atividades que descreve os possíveis processos a serem feitos na utilização do aplicativo, apresentando as atividades que o Usuário poderá realizar e as ações que o Sistema realiza com base nas atividades do Usuário.

Figura 3 - Diagrama de atividades

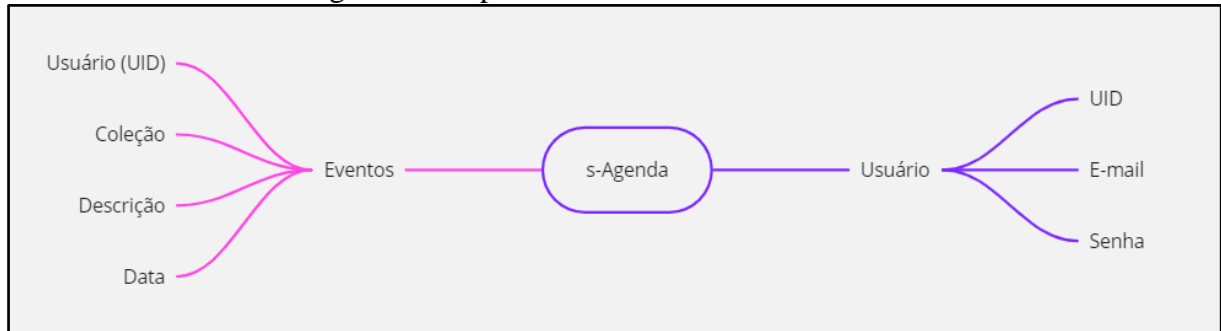


Fonte: elaborado pelo autor.

3.2.3 Estrutura de banco de dados não relacional

Nesta subseção é apresentada a estrutura do banco de dados não relacional utilizado para armazenar as informações coletadas pelo aplicativo, armazenando os diferentes usuários e as informações de eventos atrelados para cada um dos usuários. Para uma melhor representação, foi desenvolvido o mapa mental apresentado na Figura 4.

Figura 4 - Mapa mental do banco não relacional



Fonte: elaborado pelo autor.

A partir do nó **Usuário**, o banco gera automaticamente uma **UID** e o usuário fornece as informações de **e-mail** e **senha** que serão vinculadas ao **UID** gerado. Este **UID** é reutilizado no momento que um usuário faz a inclusão de um evento, em que cada usuário poderá registrar eventos apenas na sua **UID**, vinculando deste modo cada evento cadastrado à um usuário específico. A **UID** do usuário é utilizada como definição da **Coleção** para cada **Evento**, firmando deste modo o vínculo entre os **Eventos** e **Usuários**.

Cada **Evento** cadastrado pelo usuário receberá a **Descrição** que o usuário informar e receberá automaticamente a informação de **Data**, captada pelo **Firestore** no momento da inclusão do evento no aplicativo. A **Data** atrelada ao evento fica armazenada e é utilizada para posicionar cada evento corretamente no dia desejado.

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas e detalhadas as técnicas e ferramentas utilizadas para o desenvolvimento do aplicativo descrito neste trabalho. Como complemento, serão mostrados trechos de códigos do aplicativo e o modo que foi desenvolvido. A última seção mostrará a operacionalidade do aplicativo.

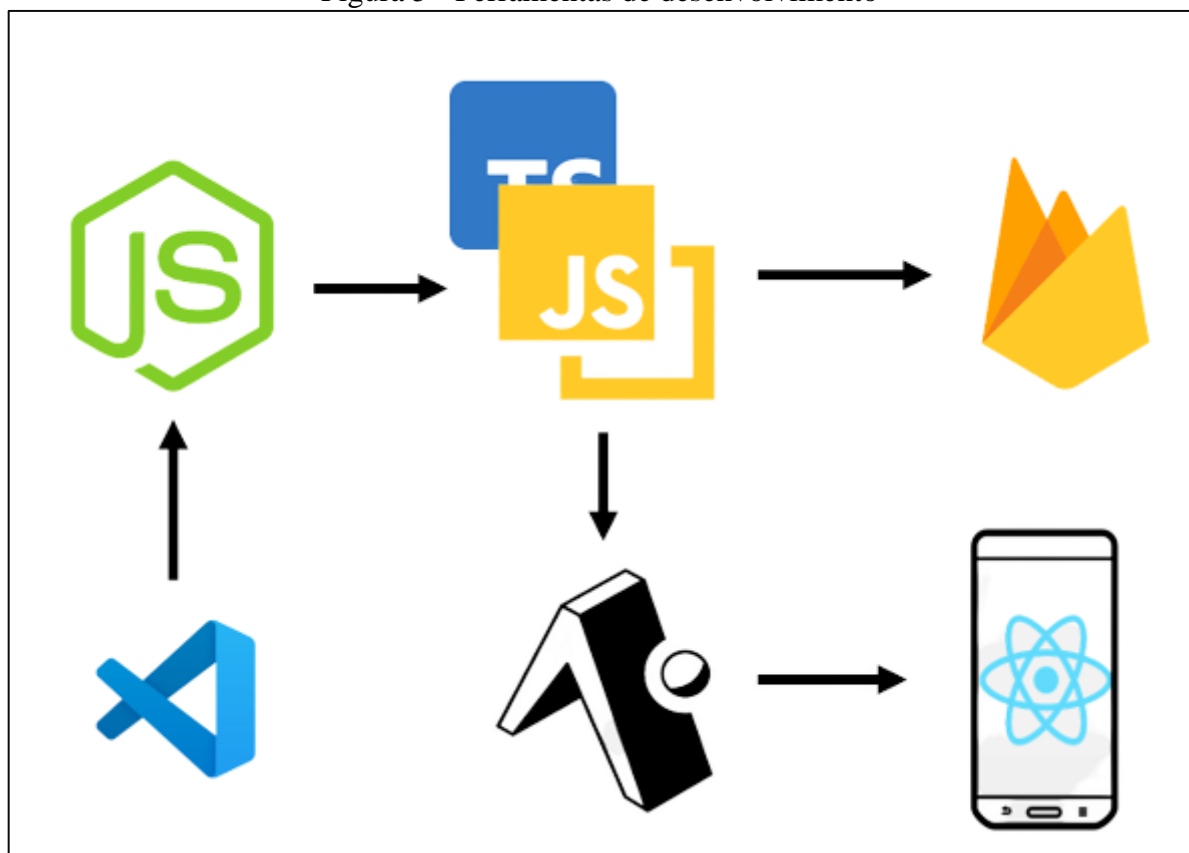
3.3.1 Técnicas e ferramentas utilizadas

Esta subseção apresenta as técnicas e tecnologias utilizadas para o desenvolvimento do aplicativo. São elas:

- a) Visual Studio Code como editor de código-fonte;

- b) Javascript como linguagem de programação;
- c) React Native como biblioteca principal para desenvolvimento Android e iOS;
- d) Firebase como banco de dados não relacional;
- e) Expo como ferramenta utilizada para emulação do aplicativo;
- f) Autenticação de usuários nativa do Firebase;
- g) Xiaomi Mi 9 Lite como dispositivo utilizado para simulação do aplicativo;
- h) AirDroid como software para espelhar o celular no computador.

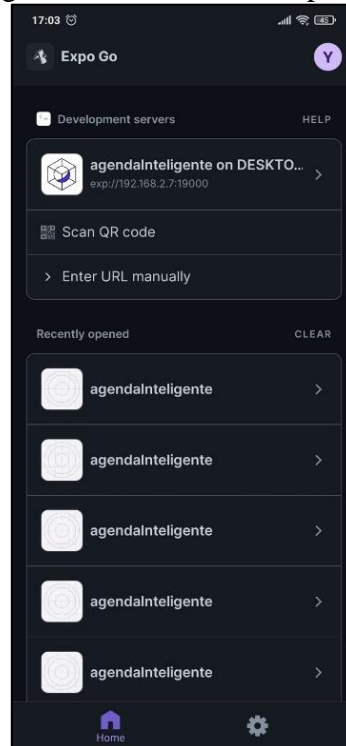
Figura 5 - Ferramentas de desenvolvimento



Fonte: elaborado pelo autor.

Inicialmente foi configurado o ambiente realizando a instalação do Node Package Manager (NPM) para realizar o controle de pacotes que foram utilizados no desenvolvimento do aplicativo. Além disso, foi usada a ferramenta Expo para rodar a aplicação e verificar o resultado em tempo real do desenvolvimento. Como pode-se ver na Figura 6, o Expo Go é o aplicativo que se utiliza para aproveitar as *Application Programming Interfaces* (APIs) nativas do dispositivo e assim conseguir rodar o aplicativo desenvolvido, sendo preciso que tanto o computador em que está o código fonte do aplicativo desenvolvido neste projeto, quanto o dispositivo móvel que está sendo utilizado para testes estejam na mesma rede Wi-Fi.

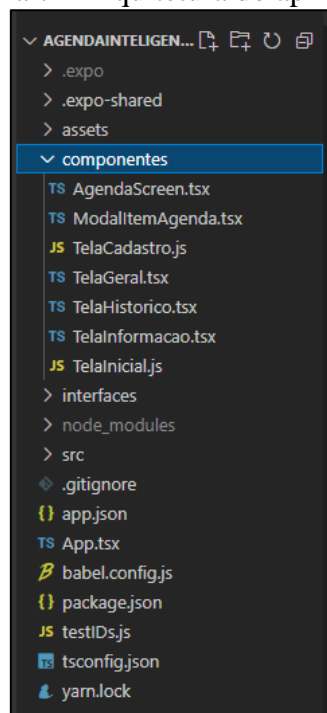
Figura 6 - Interface do Expo Go.



Fonte: desenvolvido pelo autor.

O aplicativo foi desenvolvido para funcionar apenas em plataformas móveis, utilizando a linguagem de programação Javascript e CSS para adicionar os estilos utilizados no desenvolvimento. Para organização do código fonte, foi realizada a criação da pasta `componentes` em que estão os principais componentes do aplicativo.

Figura 7 - Arquitetura do aplicativo



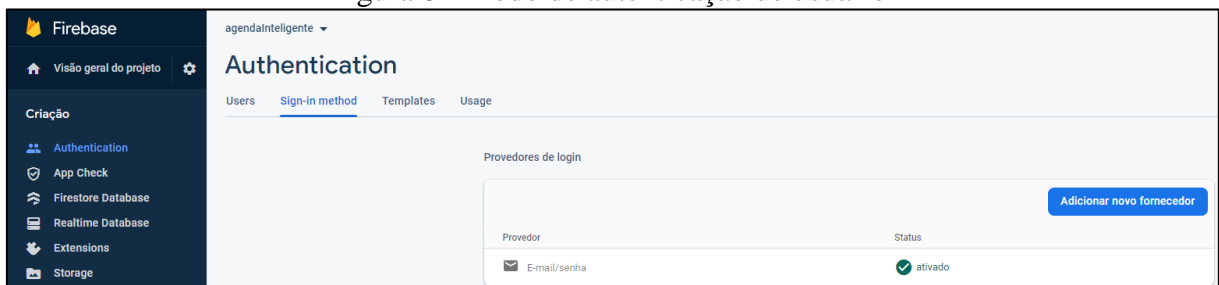
Fonte: elaborado pelo autor.

3.3.2 Operacionalidade da implementação

Nesta subseção é apresentada a parte operacional do aplicativo, demonstrando em imagens as principais telas e em ordem de processos, o passo a passo que o usuário seguirá para utilizar as funcionalidades desenvolvidas. A partir do momento que o usuário faz o acesso ao aplicativo, ele visualizará a tela inicial em que ele poderá preencher as informações de login ou o botão `Cadastro` para ir à tela responsável por realizar um novo cadastro no aplicativo.

Na tela de cadastro, o usuário precisará informar um e-mail que respeite a máscara padrão. Caso seja informado um padrão incorreto, o aplicativo irá solicitar um preenchimento correto como pode ser visto na **Erro! Fonte de referência não encontrada.** Esta validação ocorre de maneira nativa pela plataforma Firebase, visto que no próprio console do Firebase o modo de login é definido quando as configurações de Autenticação são selecionadas, como pode ser visto na Figura 8. No código fonte do aplicativo, foram utilizados os métodos específicos para realizar a criação e autenticação do usuário, conforme descrito na documentação do Firebase. Pode-se ver na linha 14 do Quadro 5, no componente `TelaCadastro.js`, o método utilizado para realizar o cadastro de um novo usuário. O método `createUserWithEmailAndPassword` já demonstra o modo de login definido por e-mail e senha.

Figura 8 - Modo de autenticação de usuário



Fonte: elaborado pelo autor.

Quadro 5 - Diretório TelaCadastro.js

```

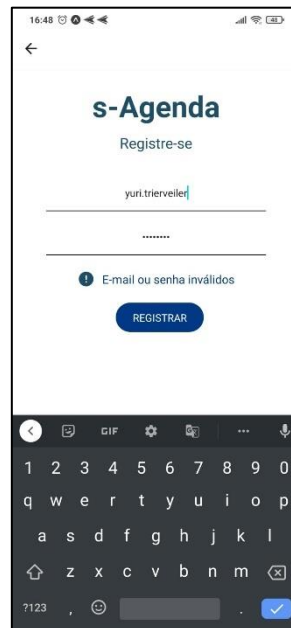
componentes > JS TelaCadastro.js > TelaCadastro
1  import React, {useState} from "react";
2  import { KeyboardAvoidingView, StyleSheet, View, Text, TextInput, TouchableOpacity } from "react-native";
3
4  import firebase from "../src/config/firebaseconfig";
5  import {MaterialCommunityIcons} from "@expo/vector-icons";
6
7  export default function TelaCadastro({navigation}) {
8
9      const [login, setLogin] = useState("");
10     const [senha, setSenha] = useState("");
11     const [errorRegister, setErrorRegister] = useState("");
12
13     const registrar = () => {
14         firebase.auth().createUserWithEmailAndPassword(login, senha)
15         .then((userCredential) => {
16             // Signed in
17             let user = userCredential.user;
18             navigation.navigate("Tela Inicial", { idUser: user.uid})
19             // ...
20         })
21         .catch((error) => {
22             setErrorRegister(true);
23             let errorCode = error.code;
24             let errorMessage = error.message;
25             // ..
26         });
27     }
28
29     return (
30         <KeyboardAvoidingView style = {styles.areaView}>
31             <Text style = {styles.titulo} >
32                 s-Agenda
33             </Text>
34             <Text style = {styles.subtitulo}>
35                 Registre-se
36             </Text>

```

Fonte: elaborado pelo autor.

Como pode ser visto nas linhas 9 e 10 do Quadro 5, foram criados `States` que recebem as informações que o Usuário preenche nos campos de edição do aplicativo, que podem ser vistos na Figura 9, recebendo e-mail e senha. Estes `States` armazenam a informação recebida e são utilizados para retornar à informação para o método de autenticação do Firebase. Nas linhas 17 e 18 é armazenado o usuário que foi efetuado o `login` e passado por meio do `Navigation` para as outras abas do aplicativo, para que o aplicativo apresente apenas as informações coerentes ao usuário autenticado.

Figura 9 - Tela de cadastro com e-mail incorreto



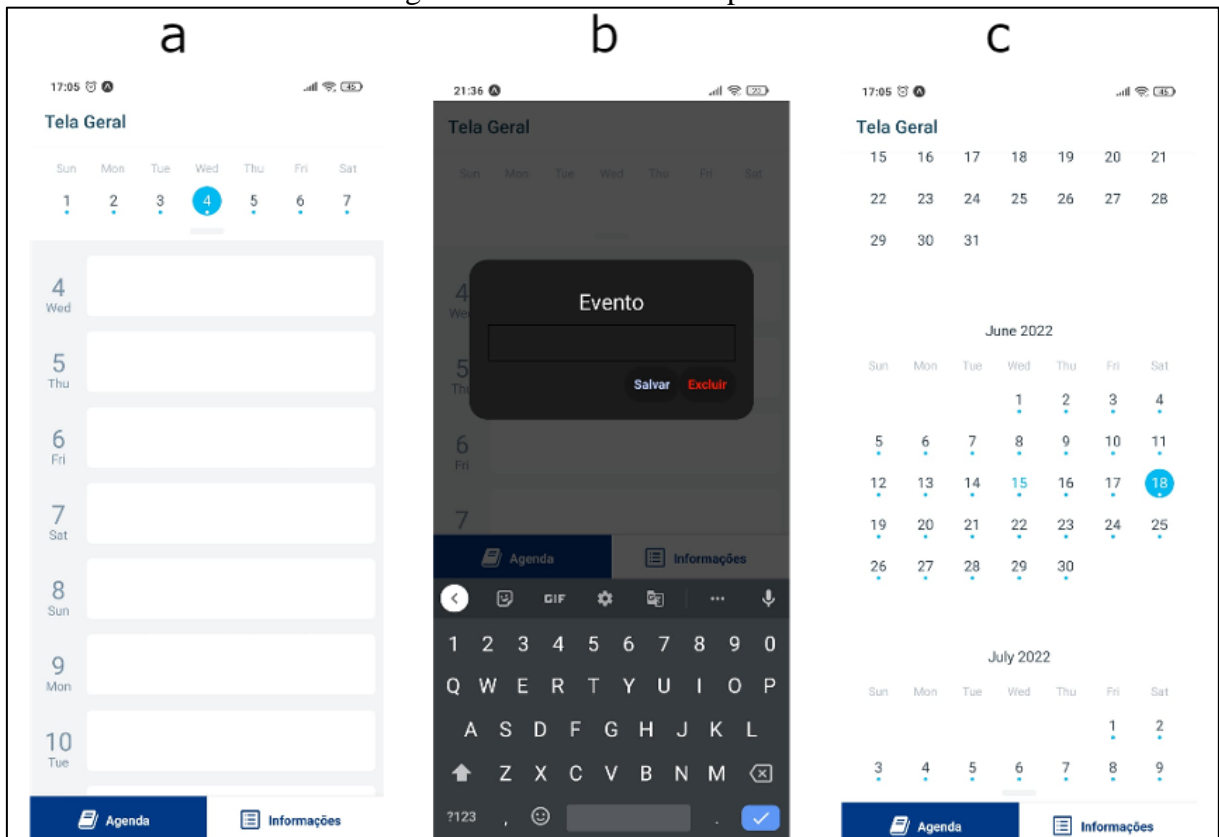
Fonte: elaborado pelo autor.

Com o *login* efetuado, o Usuário é movido para a Tela Geral em que a agenda é apresentada para ele. A aba *Agenda* possui espaços em branco que são campos editáveis que, quando pressionados abrem um novo campo onde a descrição do evento pode ser passada, como podemos ver na Figura 10 (a) e o campo que é aberto na Figura 10 (b). A agenda foi desenvolvida utilizando como base os pacotes da React Native Calendars, que possuem diferentes modelos de calendários customizáveis como o exemplo da agenda utilizada.

Pode-se ver na Figura 10 (a) como é o design da agenda e como ela é apresentada ao usuário. Para expandir o calendário completo e visualizar os dias do mês, o usuário pode arrastar os dias listados na parte superior do aplicativo para baixo ou pressionar o botão levemente destacado logo embaixo dos números dos dias, como pode ser visto na Figura 10 (c). Na agenda, o usuário precisa pressionar o campo destacado para então abrir um novo modal que possui o campo editável para realizar a inclusão de um evento, conforme demonstrado na Figura 10 (b). Com o campo recebendo alguma informação, o usuário pressiona o botão salvar para gravar a informação no banco de dados e voltar para a apresentação inicial da agenda.

De modo geral, o aplicativo é responsivo, pois quando é pressionado algum dos campos editáveis, em que o teclado padrão do celular é expandido, o aplicativo se redimensiona para que as informações não sejam sobrepostas e a usabilidade do aplicativo não seja prejudicada, como pode ser visto na Figura 10 (b) em que as abas *Agenda* e *Informação* se posicionam logo acima do teclado.

Figura 10 - Tela Geral do aplicativo



Fonte: elaborado pelo autor.

No momento que alguma informação é incluída nos dias, após pressionar o botão *Salvar*, a informação é salva para o usuário autenticado no momento, diretamente no Firebase. Como pode ser visto no Quadro 6, no código fonte `ModalItemAgenda.tsx`, da linha 16 até a linha 34 tem-se a configuração necessária para que as descrições que são incluídas nas entradas de texto editáveis sejam levadas para o banco de dados. Do mesmo modo que foi feito com a autenticação, o código fonte foi desenvolvido com base na documentação disponível no Firebase, sendo que foi necessário realizar o ajuste de validação do usuário autenticado para que as informações encaminhadas para o banco de dados fiquem vinculadas ao `UID` do usuário que está utilizando o aplicativo no momento.

Como pode ser visto pelo Quadro 6, especificamente na linha 23, foi armazenado na `const user` a informação do usuário autenticado no aplicativo naquele presente momento, utilizando método padrão do Firebase. Deste usuário é salvo o `UID` e este identificador é utilizado para definirmos as `collections` de cada evento do usuário logado. Com o vínculo entre `collection` e `UID` do usuário feita, os eventos são adicionados apenas para o usuário logado, como pode ser visto nas linhas 40 até 48.

Quadro 6 - Fonte do ModalItemAgenda - Salvar informações

```

20  const salvarTask = () => {
21    setLoading(true);
22    const database = firebase.firestore();
23    const user = firebase.auth().currentUser;
24
25    const finalizacao = () => {
26      obterEventosIguais().then((eventosIguais) => {
27        if (eventosIguais.length >= 3)
28          gerarEventoAutomatico(eventosIguais);
29        else {
30          setModalEventoAutomaticoVisible(false);
31          setModalVisible(false);
32          setTriggerRefresh(!triggerRefresh)
33        }
34
35        setLoading(false);
36      });
37    }
38
39    if (eventoSelecionado.uid === '')
40      database.collection(user.uid).add({
41        descricao,
42        data: new Date(eventoSelecionado.data)
43      }).then(finalizacao);
44    else
45      database.collection(user.uid).doc(eventoSelecionado.uid).set({
46        descricao,
47        data: new Date(eventoSelecionado.data)
48      }).then(finalizacao);
49  }
50

```

Fonte: elaborado pelo autor.

Foi desenvolvida a validação para verificar se o campo que é editado está vazio ou já possui informação, para diferenciar quando é tratado um evento novo ou um já existente no banco de dados. Quando é recebido um campo diferente de vazio, é buscado no banco de dados o campo correspondente e salvo na constante `eventoSelecionado.uid` o ID daquele evento específico, para que ele seja alterado diretamente no banco de dados e não seja criado um evento novo sem necessidade.

A exclusão identifica também o UID do usuário autenticado e realiza a exclusão com base na UID atrelada para o evento que se deseja excluir. Como pode ser visto no Quadro 7, na linha 43, o modo de exclusão é parecido com o de inclusão, alterando apenas o método `add` por `delete`.

Ainda no Quadro 6, especificamente nas linhas 25 até 37, há a `const finalizacao`, que é responsável por realizar a sugestão de inclusão de eventos quando o pré-requisito de 3 repetições é alcançado. Caso a validação de repetições não seja positiva, o aplicativo simplesmente irá salvar o evento e retornar para a tela padrão da agenda, como pode ser visto na Figura 10 (A). A `const finalizacao` é chamada sempre que é realizada a inclusão de um novo evento.

Quadro 7 - Diretório ModalItemAgenda - função de exclusão

```
51  const removerTask = () => {
52      setLoading(true);
53
54      const database = firebase.firestore();
55      const user = firebase.auth().currentUser;
56
57      if (eventoSelecionado.uid === '')
58          return
59
60      database.collection(user.uid).doc(eventoSelecionado.uid).delete().then(() => {
61          setLoading(false);
62          setModalVisible(false);
63      });
64  }
```

Fonte: elaborado pelo autor.

Na parte inferior da tela do aplicativo, pode ser visto a disposição de duas abas que diferenciam as telas que o usuário possui. A tela que é definida de maneira padrão, *Agenda*, e a tela de *Informações* em que o usuário pode verificar e consultar a listagem completa dos eventos que ele já cadastrou no aplicativo. Para o desenvolvimento da separação de telas foi usada a ferramenta nativa do React, o *Tab Navigator*.

Analisando o código apresentado no Quadro 8, pode-se verificar o modo que foi desenvolvido para apresentar de maneira separada as informações de uma aba para outra. Das linhas 13 a 22, define-se o estilo das abas que ficam sempre visíveis na parte inferior da aplicação. Das linhas 24 a 37 é feita a configuração dos caminhos que cada aba deve seguir quando pressionada pelo usuário. Os diretórios chamados em cada uma das abas foram importados previamente nas linhas 6 e 7, para então serem usados nas linhas 25 e 32.

Quadro 8 - Diretório TelaGeral.tsx - definição de abas

```

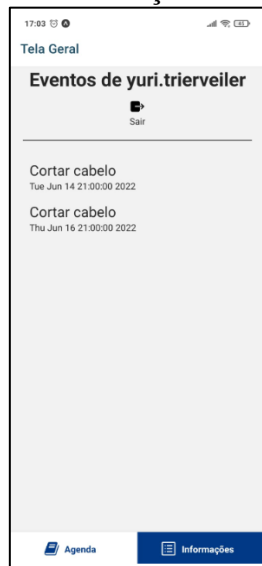
6 import AgendaScreen from './AgendaScreen';
7 import TelaInformacao from './TelaInformacao';
8
9 const Tabs = createBottomTabNavigator();
10
11 export default function TelaGeral() {
12   return (
13     <Tabs.Navigator
14       screenOptions={{
15         tabBarStyle: { backgroundColor: "#fff" },
16         tabBarActiveBackgroundColor: "#013987",
17         tabBarLabelStyle: { fontSize: 14, fontWeight: "bold"},
18         tabBarActiveTintColor: '#fff',
19         tabBarInactiveTintColor: "#013987",
20         tabBarLabelPosition: 'beside-icon',
21         headerShown: false
22       }}
23     >
24       <Tabs.Screen name = "Agenda"
25         component={ AgendaScreen }
26         options={{
27           tabBarIcon: ( { color } ) => (
28             <FontAwesome name="book" color= { color } size={ 24 } />
29           )
30         }}/>
31       <Tabs.Screen name = "Informações"
32         component={ TelaInformacao }
33         options={{
34           tabBarIcon: ( { color } ) => (
35             <AntDesign name="profile" color= { color } size={ 24 } />
36           ),
37         }}/>
38     </Tabs.Navigator>
39   );
40 }

```

Fonte: elaborado pelo autor.

Na aba de *Informações* é apresentado para o usuário a listagem conforme demonstrado na Figura 11, em que pode ser verificado todos os eventos já cadastrados anteriormente. No caso de alteração ou exclusão de algum evento, a listagem é atualizada apresentando apenas os eventos atualizados. Na parte superior dessa tela, há a informação do usuário autenticado no momento e logo em seguida o botão responsável pelo *logout*. Caso o usuário encerre o aplicativo sem realizar o *logout* da aplicação, no próximo momento de *login* o aplicativo irá conectar automaticamente no último usuário utilizado.

Figura 11 - Tela de Informações e listagem de eventos



Fonte: elaborado pelo autor.

A inteligência de sugestão foi desenvolvida por parte pelo código apresentado no Quadro 9. O comando roda de maneira assíncrona para garantirmos a utilização do comando `await`, presente na linha 106, garantindo que teremos um retorno completo do `get` realizado no banco e não haver problemas durante a execução do programa e identificação eventos repetidos. No final da execução, é retornado dentro de uma listagem os eventos que se repetiram, sendo que esta informação é utilizada posteriormente para realizar a sugestão de inclusão de eventos automáticos.

Quadro 9 - Diretório ModalItemAgenda - identificação de repetidos

```

101  const obterEventosIguais = async () => {
102      const database = firebase.firestore();
103      const user = firebase.auth().currentUser;
104
105      let eventosIguais = []
106      const snap = await database.collection(user.uid).get();
107      if (!snap.empty) {
108          snap.docs.map((doc) => {
109              const item = doc.data();
110              const dataSelecionada = diaSelecionado ? diaSelecionado.data : new Date(eventoSelecionado.data);
111
112              for (let i = 0; i < item.itens.length; i++) {
113                  const evento = item.itens[i];
114                  if (evento.descricao === descricao && item.data !== dataSelecionada) {
115                      eventosIguais.push(evento);
116                      break;
117                  }
118              }
119          });
120      }
121
122      return eventosIguais;
123  }

```

Fonte: elaborado pelo autor.

A parte do banco de dados foi incluso no código fonte com base na documentação disponível pela própria Firebase as linhas de código necessárias para realizar o vínculo entre o

banco de dados e o aplicativo. Como pode ser visto no Quadro 10, primeiramente foi realizado o `import` do Firebase em si para a aplicação, para depois ser definido quais funcionalidades seriam usadas. Foi utilizada a autenticação de usuário `auth` e o banco de dados como `storage`.

Quadro 10 - Diretório `firebaseconfig.js`

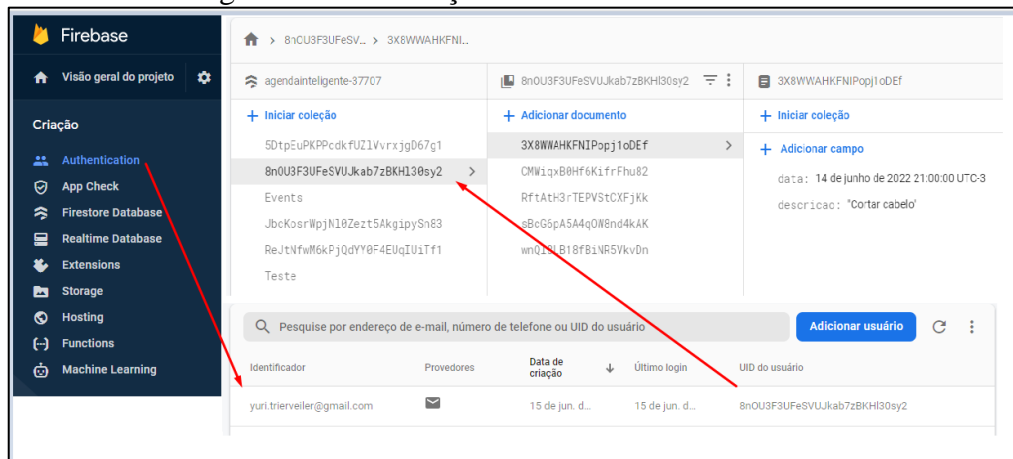
```
src > config > JS firebaseconfig.js > ...
1 // Import the functions you need from the SDKs you need
2 import firebase from "firebase"
3 import '@firebase/auth';
4 import '@firebase/storage'
5 // TODO: Add SDKs for Firebase products that you want to use
6 // https://firebase.google.com/docs/web/setup#available-libraries
7
8 // Your web app's Firebase configuration
9 const firebaseConfig = {
10   apiKey: "AIzaSyBabfoJbh_y3O_1KoqqYyuHXZaDo0niqQ",
11   authDomain: "agendainteligente-37707.firebaseio.com",
12   projectId: "agendainteligente-37707",
13   storageBucket: "agendainteligente-37707.appspot.com",
14   messagingSenderId: "800831321502",
15   appId: "1:800831321502:web:2e17abbd56e49f2697128f"
16 };
17
18 // Initialize Firebase
19 if(!firebase.apps.length){
20   firebase.initializeApp(firebaseConfig)
21 }
22 export default firebase
```

Fonte: elaborado pelo autor

O trecho de código fonte presente das linhas 9 até 21 foram copiados diretamente da documentação da Firebase, seguindo o padrão que a documentação impôs. Com essas informações inclusas, a palavra `firebase` foi definida para ser exportada do diretório e utilizada nos outros pontos do código fonte em que alguma funcionalidade do Firebase é chamada, como por exemplo a autenticação de usuário, `import` da linha 3, criação de usuário, `import` da linha 4, de eventos etc.

No console do banco de dados, pode-se consultar as informações que foram salvas na aplicação, como por exemplo os eventos ou os próprios usuários que foram cadastrados. Como pode ser visto na Figura , a informação do usuário fica salva e possui um `UID` específico para cada usuário que for registrado. Essa informação é utilizada para realizar a criação das coleções, como pode ser visto pelas setas vermelhas em destaque na Figura 12.

Figura 12 - Informações salvas no banco de dados



Fonte: elaborado pelo autor.

3.4 RESULTADOS E DISCUSSÕES

Esta seção apresenta os resultados obtidos, além das análises deles. A seção é separada em três subseções sendo a 3.4.1 a apresentação e relacionamento com os trabalhos correlatos. A subseção 3.4.2 apresenta como foi coletada a opinião dos usuários que testaram o aplicativo e realizaram a avaliação de usabilidade. Por fim, a subseção 3.4.3 relata os resultados obtidos das informações coletadas.

3.4.1 Comparação com os trabalhos correlatos

O Quadro 11 mostra a correlação dos trabalhos correlatos com o aplicativo desenvolvido neste projeto. Analisando o quadro pode-se perceber uma proximidade maior com o trabalho de Gannes (2014), sendo dentre os trabalhos correlatos o único que também foi desenvolvido para plataformas móveis. Os trabalhos de Toscan (2019) e Gomes et al. (2019) foram ambos desenvolvidos para serem incorporados em dispositivos diferentes de um aparelho telefônico, sendo que Gomes et al. (2019) apenas realiza a leitura das informações coletadas com o usuário em um dispositivo móvel.

Quadro 11 - Comparativo entre trabalhos correlatos e aplicativo

Características	Toscan (2019)	Gomes et al. (2019)	Gannes (2014)	Trierveiler (2022)
Coleta de informações dos usuários	Sim	Sim	Sim	Sim
Adequação ao perfil do usuário	Sim	Sim	Sim	Não
Colaboração dos usuários	Sim	Sim	Sim	Não
Mapeamento de preferências	Sim	Sim	Sim	Sim
Necessidade de internet	Não	Não	Não	Sim
Android	Não	Não	Não	Sim
iOS	Não	Não	Sim	Sim
Integração com outros dispositivos	Sim	Sim	Não	Não
Notificação para o usuário	Não	Não	Sim	Não

Fonte: elaborado pelo autor.

Diferentemente do trabalho de Gannes (2014) que possui a limitação pela linguagem de programação utilizada, sendo um aplicativo que funciona exclusivamente em iOS, o aplicativo desenvolvido como resultado deste trabalho pode ser utilizado tanto em Android quanto em iOS, sendo este um diferencial prático para atingir uma maior quantidade de usuários. Este diferencial se origina exclusivamente do *framework* utilizando, o React Native, que de maneira nativa permite que o aplicativo desenvolvido funcione em ambas as plataformas.

Os três trabalhos correlatos apresentados possuem a finalidade de auxiliar o cotidiano das pessoas, como o aplicativo desenvolvido neste trabalho. O programa de Toscan (2019) serve como assistente para o cotidiano do usuário. Gomes et al. (2019) desenvolveu um programa para acompanhar as tarefas básicas do dia a dia como se alimentar, ingerir líquidos, captando essas ações pelo movimentar do relógio presente no punho do usuário. Gannes (2014) também desenvolveu uma espécie de agenda que após analisar a rotina do usuário, sugere mudanças para que ele possua um melhor aproveitamento do tempo que possui em um dia. O aplicativo desenvolvido neste trabalho, auxilia de maneira mais simples ao usuário a ter um melhor controle do cotidiano, apresentando em um calendário as tarefas que o próprio usuário informou e oferecendo sugestões de inclusões de eventos para o usuário.

3.4.2 Análise de usabilidade

Para a análise da usabilidade do aplicativo desenvolvido, como também para a confirmação se os objetivos levantados no trabalho foram alcançados, foi realizada uma pesquisa com um questionário desenvolvido com a ferramenta Google Forms e encaminhada a URL diretamente para os avaliadores. Os avaliadores escolhidos foram pessoas próximas do círculo familiar, além de amigos e conhecidos do trabalho do autor deste. O aplicativo foi

disponibilizado para download em um link gerado no website `File.fm` e o link foi disponibilizado na primeira seção do formulário, como pode ser visto no Apêndice A, Figura 15 e Figura 16.

A avaliação incluiu um termo de consentimento livre e esclarecido (Apêndice A, Figura 16), além de algumas perguntas básicas para o levantamento de informações sobre o avaliado (Apêndice A, Figura 17). O formulário solicita aos avaliadores que realizem algumas atividades durante o uso do aplicativo e retornem o resultado obtido informando 1, se acharam a tarefa muito difícil e 5 se acharam a tarefa muito fácil. O formulário conta com perguntas que se basearam no questionário Software Usability Measurement Inventory (SUMI), apresentando uma versão adaptada para o cenário desenvolvido pelo autor, selecionando algumas perguntas em vez de utilizar todas, visto que algumas não fariam sentido no cenário da pesquisa.

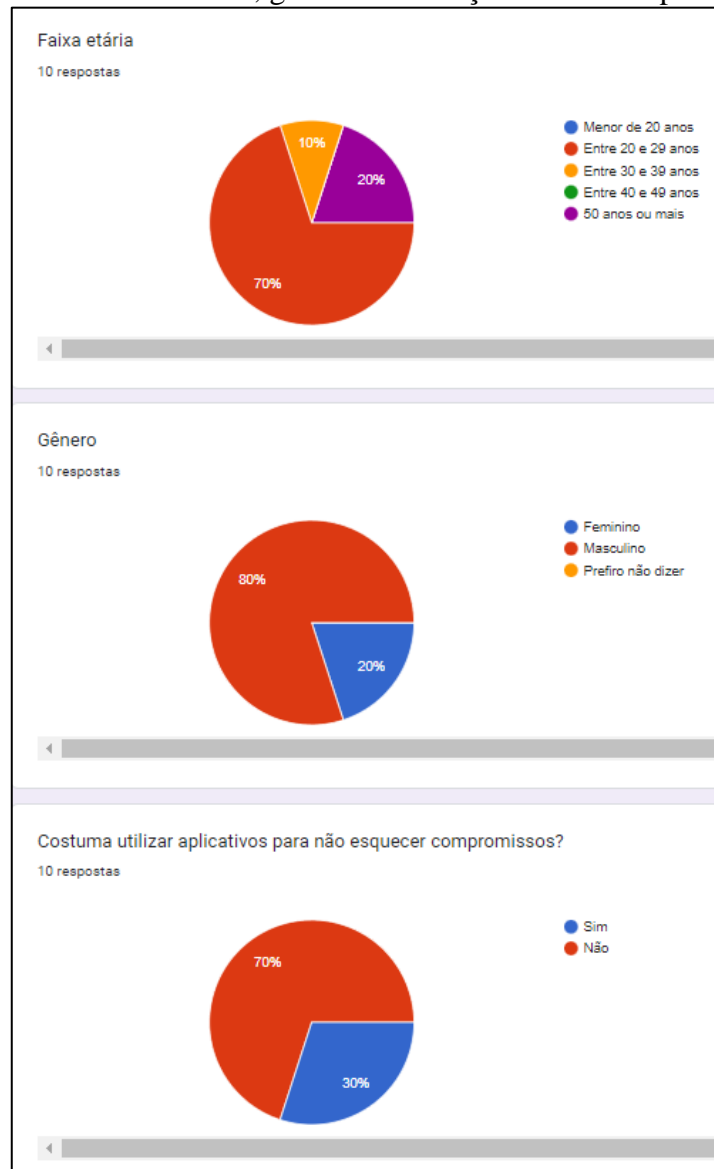
Por fim, o questionário possui três caixas de textos abertas para a inclusão dos pontos positivos, pontos negativos e comentários gerais que os avaliadores possuem.

3.4.3 Resultados

A pesquisa foi realizada por 10 avaliadores voluntários que se disponibilizaram para testar o aplicativo e responder o formulário. Os resultados do perfil dos avaliadores (Figura 13 e Figura 14) retornaram dados diversos, apresentando algumas de suas características. 70% dos participantes estão na faixa etária de 20 a 29 anos, 20% na faixa de 50 anos ou mais e 10% na faixa de 30 a 39 anos. A pesquisa foi majoritariamente respondida por homens, sendo eles 80% dos avaliadores e 20% mulheres. Analisando a utilização de aplicativos com finalidades semelhantes ao aplicativo desenvolvido, 70% dos avaliadores informaram que não utilizam aplicativos para não se esquecerem de compromissos enquanto 30% já utilizam outros aplicativos.

Seguindo o questionário, foi retornado que 70% dos avaliadores possuem dificuldade em lembrar de atividades que não realizam com tanta frequência, enquanto 30% demonstraram não ter problemas com isso. 100% dos avaliadores informaram utilizar bastante aplicativos em seu cotidiano e 100% dos avaliadores utilizam o sistema operacional Android, como pode-se confirmar na Figura 14, no terceiro gráfico apresentado.

Figura 13 - Faixa etária, gênero e utilização de outros aplicativos



Fonte: elaborado pelo autor.

Figura 14 - Tarefas cotidianas, frequência com aplicativos e sistema operacional



Fonte: elaborado pelo autor.

Conforme dito na subseção 3.4.2, o questionário apresentado na seção 5 (Apêndice A, Figuras 20 a 23) do formulário encaminhado para os avaliadores foi desenvolvido com base no questionário SUMI, utilizando algumas das perguntas para compreender de maneira mais direta a opinião dos avaliadores. No Quadro 12, tem-se o resultado obtido após coletar as respostas e organizá-las em uma tabela.

Quadro 12 - Relação das perguntas com as porcentagens de cada resposta

Pergunta	Percentual de respostas por escala				
	1	2	3	4	5
1. O aplicativo é fácil de usar			20%	10%	70%
2. É fácil de navegar por ele				20%	80%
3. A informação do aplicativo é crível		10%	70%	10%	10%
4. A informação do aplicativo é confiável		10%	70%	20%	
5. Eu recomendaria este aplicativo a um amigo ou colega				70%	30%
6. Eu provavelmente irei visitar este aplicativo no futuro	10%	20%	50%	10%	10%
7. Eu achei o aplicativo atraente		10%	20%	20%	50%
8. O aplicativo tem uma apresentação limpa e simples				50%	50%
9. Este tipo de aplicativo é muito importante para realizar as minhas tarefas	20%	40%	20%	10%	10%

Fonte: elaborado pelo autor.

Considerando o valor 1 como “Discordo totalmente” e o valor 5 como “Concordo totalmente”, os avaliadores apresentaram suas opiniões para cada uma das perguntas listadas no formulário (Apêndice A, Figura 21 a Figura 23). Analisando as informações coletadas, pode-se verificar pelo resultado obtido nas duas primeiras perguntas que a maior parte dos avaliadores concordou totalmente com a afirmação, o que nos permite entender que o aplicativo realmente se mostrou simples e fácil de utilizar. As perguntas 3 e 4 contaram com uma porcentagem maior no três, mostrando a neutralidade dos avaliadores em relação às perguntas, sendo que apenas 20% responderam de maneira positiva para ambas as perguntas, o que nos permite entender que o resultado não foi tão bom quanto o esperado. As perguntas 5, 7 e 8 receberam em sua maioria, respostas onde os avaliadores concordaram com a afirmação das frases, sendo possível entender que o aplicativo desenvolvido realmente possui uma aparência que facilita a utilização por parte do usuário e que os usuários que testaram ainda recomendariam o aplicativo para amigos e conhecidos. Por fim, as perguntas 6 e 9 mostraram-se preocupantes pela avaliação elaborada, onde na pergunta 6, apenas 20% dos avaliadores concordou em certo grau com a afirmação enquanto 30% discordaram e 50% se mostraram neutros. E na 9, a maior parte dos avaliadores discordou da afirmação, o que se pode atribuir às poucas funcionalidades que o aplicativo possui em comparação com outros aplicativos já existentes no mercado, oferecendo pouco suporte para os usuários em suas atividades do dia a dia.

4 CONCLUSÕES

Neste trabalho é apresentado o desenvolvimento de um aplicativo para plataformas móveis que tem como objetivo principal auxiliar no controle de atividades do dia a dia das pessoas. Com base nos questionários realizados e nos resultados obtidos, foi confirmado que o aplicativo desenvolvido é de fácil utilização e possui um leiaute amigável conforme o proposto nos objetivos do trabalho, porém não atingiu resultados satisfatórios o suficiente para considerarmos como completo mediante a opinião dos avaliadores.

O aplicativo foi desenvolvido na linguagem Javascript, utilizando a biblioteca React Native para todo o desenvolvimento. O React Native permite a utilização do aplicativo tanto em Android quanto iOS, porém vale ressaltar que todos os testes realizados na utilização do App foram feitos em aparelhos Android. Para realizar a emulação do aplicativo, a ferramenta Expo foi utilizada, permitindo acompanhar as alterações que eram feitas no código em tempo real no aparelho celular, facilitando para concluir a realização dos testes necessários.

As principais dificuldades encontradas durante o desenvolvimento do trabalho, ao pouco contato com a linguagem de programação Javascript. A princípio, a dificuldade de iniciar o desenvolvimento atrapalhou na progressão inicial, tendo perdido bastante tempo até que fosse possível visualizar as primeiras interfaces do App. A existência do módulo React Native Calendars facilitou o início do desenvolvimento, porém por ser um módulo que não recebia atualização há certo tempo, algumas das funcionalidades que ele oferecia não funcionavam da maneira pretendida, sendo necessário alguns reajustes que tomavam bastante tempo até que o resultado desejado fosse alcançado. Do meio para o fim do projeto, houve uma grande dificuldade para a implementação do controle de sugestões desenvolvido, tomando bastante tempo de análise para alcançar um modo que fosse possível retornar sugestões para o usuário com base nas atividades que ele cadastra no aplicativo. A proposta inicial seria utilizar as técnicas de Machine Learning, porém pela pequena quantidade de informações que o aplicativo possui, foi percebido que era inviável este nível de tecnologia para o desenvolvimento do trabalho.

No presente momento, o aplicativo possui algumas limitações bem básicas que necessitam de uma correta utilização do App por parte do usuário. Por exemplo, as sugestões que o aplicativo faz irão apenas respeitar descrições de eventos que forem inclusas de maneira idêntica, não havendo ainda uma inteligência desenvolvida para que o usuário possa incluir descrições com palavras chaves. As sugestões ocorrerão no momento de inclusão de novas atividades na agenda, sendo preciso que o usuário se recorde de realizar uma nova inclusão de

evento para que a regra desenvolvida seja chamada. O aplicativo desenvolvido não permite a alteração do cadastro realizado, nem a recuperação da senha cadastrada, sendo preciso que o usuário utilize sempre o mesmo *login* e senha que for definido.

Como contribuição social, este trabalho contribui com a entrega de uma ferramenta de organização do cotidiano para a população. O aplicativo possui o formato de agenda que permite aos usuários a inclusão de suas atividades especificamente nos dias desejados. Como contribuição tecnológica, este trabalho oferece um aplicativo desenvolvido em React Native, compatível com os sistemas operacionais Android e iOS, sendo utilizado além disso recursos do Firebase, como por exemplo a autenticação e armazenamento de dados. Como contribuição científica este trabalho traz a aplicação do questionário de avaliação de usabilidade *Software Usability Measurement Inventory* (SUMI), que já é consolidado, para validar o aplicativo desenvolvido.

4.1 EXTENSÕES

Nesta seção são apresentadas as possíveis extensões para o aplicativo desenvolvido neste trabalho. Os principais pontos que puderam ser observados durante o desenvolvimento do aplicativo e principalmente agora na etapa final são:

- a) aprimorar o controle de usuários para que um *login* possa ter mais de um perfil cadastrado;
- b) incluir a possibilidade de alteração do cadastro de *login*, permitindo a alteração de senha caso ocorra o esquecimento;
- c) aprimorar a regra de sugestões para que ocorra automaticamente em determinado período do dia por notificação, sem ser preciso abrir o aplicativo;
- d) aprimorar a listagem dos eventos cadastrados na tela informação para que o usuário possa ter opções de filtros com base em atividades comuns;
- e) aprimorar o desenvolvimento do módulo de agenda utilizado para melhorar a performance do aplicativo;
- f) disponibilizar uma nova tela para receber sugestões dos usuários.

REFERÊNCIAS

- BRANCO, Isabel. Saiba porque a infância é fase de maior desenvolvimento intelectual. **Revista EDUCAÇÃO**, São Paulo, v. 252, set. 2018. Disponível em: <https://revistaeducacao.com.br/2018/10/01/infancia-desenvolvimento>. Acesso em: 15 mai. 2022.
- FORBES. Revolução tecnológica deve transformar a vida no mundo. **Forbes**, 2019. Disponível em: <https://forbes.com.br/brand-voice/2019/08/revolucao-tecnologica-deve-transformar-a-vida-no-mundo>. Acesso em: 16 de mai. 2022.
- GANNES, Liz. **Dan Ariely's Timefull App Helps You Better Apply Your Time**. 2014 Disponível em: <https://www.vox.com/2014/7/31/11629372/dan-arielys-timeful-app-helps-you-better-apply-your-time>. Acesso em: 04 oct. 2021.
- GOMES, Diana. MOREIRA, João. SOUSA, Ines. SILVA, Joana. Eating and Drinking Recognition in Free-Living Conditions for Triggering Smart Reminders. **Sensors**, v19, n. 12, p. 2803, 2019. Disponível em: <https://www.mdpi.com/1424-8220/19/12/2803>. Acesso em: 25 nov. 2021.
- IBGE. **População de Blumenau**. 2021. Disponível em: <https://cidades.ibge.gov.br/brasil/sc/blumenau/panorama>. Acesso em: 14 jul. 2022.
- OLIVEIRA, Francisco. OLIVEIRA, Fernando. **Interação Humano Computador**. 2ª Edição. Fortaleza: Editora UAB/UECE, 2015.
- TELECO. **Inteligência em Telecomunicações**. 2021. Disponível em: https://www.teleco.com.br/ncel_cidades2.asp. Acesso em: 26 sep. 2021.
- TOSCAN, Luiz Fernando. **Protótipo de um assistente pessoal digital inteligente**. 2019. f.47. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Universidade Tecnológica Federal do Paraná, Pato Branco. 2019.
- VITORIA, Luana. **De que maneira as tecnologias nos ajudam no dia a dia?**, UpperSoft, 2019. Disponível em: <https://uppersoft.com.br/tecnologia-dia-a-dia>. Acesso em: 5 Oct. 2021.

APÊNDICE A – Formulário encaminhado para verificação da Usabilidade do aplicativo

Este apêndice apresenta o formulário desenvolvido para avaliação da usabilidade do aplicativo. Na Figura 15 pode-se ver a primeira seção visível para o avaliador e o link para realização do download do aplicativo.

Figura 15 - Seção 1 do formulário



The image shows a screenshot of a questionnaire form. At the top, there is a purple header bar. Below it, the title 'Usabilidade do aplicativo Agenda Inteligente' is displayed in a large, bold, black font. Underneath the title, there is a paragraph of text: 'O aplicativo avaliado é um projeto de conclusão do curso de Bacharelado em Sistemas de Informação, na instituição de ensino Universidade Regional de Blumenau (FURB), no 1º semestre de 2022.' Below this paragraph, there is another paragraph: 'Este aplicativo está disponível publicamente, no endereço <https://files.fm/f/5ws6yqkw3>, sendo acessada por meio de dispositivo Android. Para instalá-lo, basta abrir o arquivo disponível em seu aparelho telefônico.'

Fonte: elaborado pelo autor.

Figura 16 - Seção 2 do formulário - Termo de Consentimento

Termo de Consentimento Livre e Esclarecido

Eu, usuário que está avaliando este projeto, estou sendo convidado a participar de um estudo denominado Avaliação de usabilidade e experiência de usuário, cujos objetivos e justificativas são: avaliar a aplicação mencionada a partir da sua utilização e, posteriormente, da realização da avaliação de usabilidade e experiência da aplicação. Esta avaliação servirá como base das futuras melhorias e mudanças que a aplicação avaliada possa sofrer, além de levantar a viabilidade da continuação do projeto.

A minha participação no referido estudo será no sentido de executar a aplicação, utiliza-la e executar a avaliação por meio de um formulário de perguntas definidas.

Fui alertado de que, da pesquisa a se realizar, posso esperar alguns benefícios, tais como o direito de receber as informações sobre os dados da pesquisa a respeito da aplicação.

Recebi, por outro lado, os esclarecimentos necessários sobre os possíveis desconfortos e riscos decorrentes do estudo, levando-se em conta que é uma pesquisa, e os resultados positivos ou negativos somente serão obtidos após a sua realização. Assim, estou sujeito a utilização da aplicação por um tempo a ser definido. Ainda, a minha avaliação poderá ou não ser considerada no resultado final da aplicação, dependendo de como eu irei responder a avaliação.

Estou ciente de que minha privacidade será respeitada, ou seja, meu nome ou qualquer outro dado ou elemento que possa, de qualquer forma, me identificar, será mantido em sigilo.

Também fui informado de que posso me recusar a participar do estudo, ou retirar meu consentimento a qualquer momento, sem precisar justificar, e que, por desejar sair da pesquisa, não sofrerei qualquer prejuízo.

Os pesquisadores envolvidos com o referido projeto são: Yuri Trierweiler, da Universidade Regional de Blumenau (FURB), onde posso entrar em contato pelo e-mail yuri.trier@gmail.com e Professora Luciana Pereira de Araújo Kohler, da Universidade Regional de Blumenau (FURB), onde posso entrar em contato pelo e-mail lp@furb.br.

Enfim, tendo sido orientado quanto ao teor de todo o aqui mencionado e compreendido a natureza e o objetivo do já referido estudo, manifesto meu livre consentimento em participar, estando totalmente ciente de que não há nenhum valor econômico, a receber ou a pagar, por minha participação.

Em caso de reclamação ou qualquer tipo de denúncia sobre este estudo devo entrar em contato com Professora Luciana Pereira de Araújo Kohler, da Universidade Regional de Blumenau (FURB), onde posso entrar em contato pelo e-mail lp@furb.br.

Blumenau, 22 de junho de 2022.

Yuri Trierweiler, Acadêmico - Universidade Regional de Blumenau (FURB)
Luciana Pereira de Araújo Kohler, Professora - Universidade Regional de Blumenau (FURB)

AO PROSEGUIR PARA A PRÓXIMA SEÇÃO DESTE FORMULÁRIO DE AVALIAÇÃO, DECLARO QUE ESTOU DE ACORDO COM OS TERMOS EXPLÍCITOS ACIMA.

Fonte: elaborado pelo autor.

Figura 17 - Primeira parte da Seção 3 do formulário

Usabilidade do aplicativo Agenda Inteligente

[Faça login no Google](#) para salvar o que você já preencheu. [Saiba mais](#)

***Obrigatório**

Informações do entrevistado

E-mail para contato

Sua resposta _____

Seu conhecimento com relação a aplicativos móveis *

1 2 3 4 5

Nenhum Muito bom

Faixa etária *

Menor de 20 anos

Entre 20 e 29 anos

Entre 30 e 39 anos

Entre 40 e 49 anos

50 anos ou mais

Gênero *

Feminino

Masculino

Prefiro não dizer

Outro: _____

Fonte: elaborado pelo autor.

Figura 18 - Segunda parte da seção 3 do formulário

Costuma utilizar aplicativos para não esquecer compromissos? *

Sim

Não

Você tem o hábito de esquecer tarefas cotidianas que são feitas com menor frequência? Como por exemplo limpar o ar-condicionado, trocar o gás (antes que acabe), calibrar pneus do carro? *

Sim

Não

Você utiliza aplicativos com frequência? *

Sim

Não

Qual sistema operacional você utiliza? *

Android

iOS

Fonte: elaborado pelo autor.

Figura 19 - Primeira parte da seção 4 do formulário

Avaliação de Usabilidade do aplicativo Agenda Inteligente

O aplicativo Agenda Inteligente é um aplicativo que tem como finalidade auxiliar os usuários em seu cotidiano, fornecendo um meio para que os usuários possam registrar suas atividades e tenham um controle melhor de rotinas do dia a dia.

Com o aplicativo instalado em seu celular, realize as tarefas descritas nos tópicos seguintes e informe o seu grau de dificuldade em realizar a tarefa proposta, sendo 1 muito difícil e 5 muito fácil.

Realize o cadastro de um usuário. *

1 2 3 4 5

Muito difícil Muito fácil

Faça login no aplicativo. *

1 2 3 4 5

Muito difícil Muito fácil

Realize a inclusão de um evento. *

1 2 3 4 5

Muito difícil Muito fácil

Fonte: elaborado pelo autor.

Figura 20 - Segunda parte da seção 4 do formulário

Realize a inclusão de um evento diferente do anterior. *						
	1	2	3	4	5	
Muito difícil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito fácil
Realize a inclusão repetida de três eventos. *						
	1	2	3	4	5	
Muito difícil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito fácil
Realize a exclusão de um evento cadastrado. *						
	1	2	3	4	5	
Muito difícil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito fácil
Altere um evento cadastrado. *						
	1	2	3	4	5	
Muito difícil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito fácil

Fonte: elaborado pelo autor.

Figura 21 - Primeira parte da seção 5 do formulário

O questionário é baseado no questionário SUMI.

O aplicativo é fácil de usar. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

É fácil de navegar por ele. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

A informação do aplicativo é crível. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Fonte: elaborado pelo autor.

Figura 22 - Segunda parte da seção 5 do formulário

A informação do aplicativo é confiável. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Eu recomendaria este aplicativo a um amigo ou colega. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Eu provavelmente irei visitar este aplicativo no futuro. *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Fonte: elaborado pelo autor.

Figura 23 - Terceira parte da seção 5 do formulário

Eu achei o aplicativo atraente. *						
	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

O aplicativo tem uma apresentação limpa e simples. *						
	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

Este tipo de aplicativo é muito importante para realizar minhas tarefas *						
	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

Fonte: elaborado pelo autor.

Figura 24 - Quarta parte da seção 5 do formulário

Para qual atividade, em geral, você utiliza este tipo de aplicativo? *
Sua resposta

Pontos positivos do aplicativo: *
Sua resposta

Pontos negativos do aplicativo: *
Sua resposta

Comentários gerais: *
Sua resposta

Fonte: elaborado pelo autor.

APÊNDICE B – Descrição de Casos de Uso

Este Apêndice apresenta a descrição de dois dos principais casos de uso presentes na Figura 2.

O Quadro 13 detalha o caso de uso UC02 – Manter login.

Quadro 13 - Descrição detalhada do caso de uso UC02 – Manter Login

Caso de Uso	Manter login
Descrição	O aplicativo permite ao usuário que mantenha o usuário logado, sem precisar preenche-lo toda vez.
Ator	Usuário
Pré-condição	O usuário precisa ter um cadastro previamente feito e conexão com a internet.
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário realiza login no aplicativo. 2. O usuário utiliza o aplicativo normalmente. 3. O usuário fecha o aplicativo sem realizar o logout.
Fluxo alternativo	
Pós-condição	O aplicativo abre novamente com o último usuário utilizado.

Fonte: elaborado pelo autor.

O Quadro 14 detalha o caso de uso UC03 – Manter eventos.

Quadro 14 - Descrição detalhada do caso de uso UC03 – Manter eventos

Caso de Uso	Manter eventos
Descrição	O aplicativo permite ao usuário que realize a inclusão, alteração e exclusão dos eventos desejados.
Ator	Usuário
Pré-condição	O usuário precisa estar logado no aplicativo e possuir conexão com a internet.
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário realiza a inclusão de um evento na agenda ao pressionar o espaço demarcado na agenda e pressionando o botão salvar.
Fluxo alternativo	<ol style="list-style-type: none"> 1. O usuário realiza a alteração de um evento incluso pressionando o espaço designado com a descrição. 2. O usuário realiza a exclusão de um evento incluso pressionando o espaço designado e pressionando o botão Excluir. 3. O usuário realiza a inclusão de eventos repetidos.
Pós-condição	O aplicativo abre novamente com o último usuário utilizado.

Fonte: elaborado pelo autor.