



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLOGIA DA INFORMAÇÃO
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Heron Carlos da Silva Santos

TIMESHARE: Uma Aplicação Mobile para Gerenciamento de Agendas

ANGICOS-RN

2025

Heron Carlos da Silva Santos

TIMESHARE: Uma Aplicação Mobile para Gerenciamento de Agendas

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Francisco de Assis Pereira Vasconcelos de Arruda.

ANGICOS-RN

2025

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tornar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

d237t da Silva Santos, Heron Carlos .
TIMESHARE: Uma Aplicação Mobile para
Gerenciamento de Agendas / Heron Carlos da Silva
Santos. - 2025.
105 f. : il.

Orientador: Francisco de Assis Pereira
Vasconcelos de Arruda.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Sistemas de
Informação, 2025.

1. Flutter Framework. 2. gestão de tempo. 3.
aplicativo mobile. 4. rede de usuários. I.
Pereira Vasconcelos de Arruda, Francisco de Assis
, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo) autor(a).
Biblioteca Campus Angicos
Bibliotecário: Helder Romero Maia Duarte
CRB: 15/673

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

Heron Carlos da Silva Santos

TIMESHARE: Uma Aplicação Mobile Para Gerenciamento de Agendas

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Defendida em: 27 / 06 / 2025.

BANCA EXAMINADORA

Prof. Dr. Francisco de Assis Pereira Vasconcelos de Arruda (UFERSA)
Presidente

Profa. Dra. Joêmia Leilane Gomes de Medeiros (UFERSA)
Membro Examinador

Profa. Dra. Thatiana Cunha Navarro Diniz (UFERSA)
Membro Examinador

DEDICATÓRIA

*Dedico este trabalho a minha mãe, Silvanete
Severino da Silva, a maior razão de poder
estar aqui hoje.*

AGRADECIMENTOS

Agradeço primeiramente a minha mãe, Silvanete Severino da Silva, por todos os ensinamentos e as oportunidades que proporcionou em minha vida. Me impulsionou, me guiou, e que se culminaram quando formalizei minha educação.

Agradeço a minha avó, Maria José Severino, e a minha tia, Albaniza Maria da Costa, que me deram apoio e foram mais duas mães para mim.

Agradeço ao meu orientador, Dr. Francisco de Assis Pereira Vasconcelos de Arruda, por me guiar e ser paciente em nossa colaboração, e por ampliar meu amor por desenvolvimento de software.

Agradeço também aos meus amigos, Douglas Ryan, Equiton Gomes, e Francisco Fernandes, por me acompanharem e fortalecerem minha experiência acadêmica.

EPÍGRAFE

“Eu errei mais de 9 mil arremessos em minha carreira. Eu perdi quase 300 jogos. 26 vezes eu fui confiado a arremessar para ganhar o jogo, e errei. Eu falhei de novo, de novo, e de novo em minha vida. E é por isso que eu tive sucesso.”

Michael Jordan, 1997. (Tradução própria)

RESUMO

Com a crescente necessidade de gerenciamento de tempo, empresas e instituições investem em soluções para resolver problemas genéricos em ações coletivas, como ambiguidade temporal, interesses e requerimentos conflitantes de tempo e escassez de recursos. Neste sentido, objetivou-se com esta monografia, criar um aplicativo que otimize atividades em grupo, promovendo uma comunicação eficiente, permitindo o acesso rápido e fácil para ações cotidianas. Para isto, foi utilizado a tecnologia de desenvolvimento Flutter Framework como principal ferramenta para criar uma aplicação de fácil usabilidade, obedecendo a estes objetivos. Pesquisando tentativas anteriores para objetivos semelhantes, e analisando a documentação da arquitetura de *software*, criou-se uma versão para testes de usabilidade com os usuários. Estes testes revelaram a satisfação e sugestões de melhorias para o Timeshare, concluindo o objetivo deste projeto. Assim, para próximas iterações, fica claro um caminho onde podem ser melhoradas as funcionalidades, visto que esta aplicação partiu de um ponto teórico, e é através de um estudo empírico onde sua versão melhorada será alcançada.

Palavras-chave: Flutter Framework; gestão de tempo; aplicativo *mobile*; rede de usuários.

ABSTRACT

With the growing necessity of time management, companies and institutions invest in solutions to solve common problems during collective activities, such as time ambiguity, time conflicting requirements and interests, and the overall resource scarcity. In that aspect, this work's objective is to create an application that facilitates group activities, and that keeps good communication, which allows for quick and easy access to routine events. For that objective, the Flutter framework technology was the main tool to create an application of easy use. Searching for previous attempts at similar projects, and analyzing the framework's documentation, it was possible to arrive at a test version with random users. These tests revealed the users' satisfaction and new suggestions to improve Timeshare, which validated this project's end goal. As such, for the next iterations it becomes a clear path where the improvement and new functionalities come from. Seeing as this application started as a theoretical concept, it is through an empirical study where its improved version will be achieved.

Keywords: Flutter Framework; time management; mobile application; .

LISTA DE FIGURAS

Figura 1 - Fluxograma de Atividades.....	38
Figura 2 - Contextualização de uma aplicação Flutter.....	40
Figura 3 - Diagrama de Classes do Aplicativo Timeshare.....	42
Figura 4 - Console de Cloud Firestore.....	45
Figura 5 - Regras de Segurança do Cloud Firestore.....	45
Figura 6 - Criação de uma Instância no Cloud Firestore.....	46
Figura 7 - Função para Criar Grupo.....	48
Figura 8 - Table Calendar.....	49
Figura 9 - Calendário Flutter.....	50
Figura 10 - Encapsulamento de Widgets.....	51
Figura 11 - Tela de Login do Aplicativo Timeshare.....	40
Figura 12 - Email de Recuperação de Senha.....	41
Figura 13 - Redefinição de Senha.....	42
Figura 15 - Tela Principal.....	44
Figura 16 - Tela de Detalhes/Edição de Atividades.....	45
Figura 17 - Concluir/Excluir Atividades.....	46
Figura 18 - Adicionar Atividades.....	47
Figura 19 - Tela de Amigos.....	48
Figura 20 - Tela de Perfil do Usuário.....	49
Figura 21 - Tela de Gerenciamento de Grupos.....	50
Figura 22 - Tela de Agenda de Grupos.....	51
Figura 23 - Gerenciamento de Membros de Grupos.....	52
Figura 24 - Tela de Configurações de Grupo.....	53
Figura 25 - Tela de Suporte.....	54
Figura 26 - Tela de Configuração e Atualização de Perfil.....	55
Figura 27 - Atualização das páginas.....	64
Figura 28 - Modificações Para Adaptar Largura da Tela.....	65
Figura 29 - Adição do Começo e Fim da Atividade.....	66

Figura 30 - Tutorial.....	67
---------------------------	----

LISTA DE QUADROS

Quadro 1 - Comparação entre ferramentas de gerenciamento de tempo.....	30
Quadro 2 - Comparativo Flutter, React Native, React JS.....	35
Quadro 3 - Tarefas do Teste de Usabilidade do Aplicativo Timeshare.....	56
Quadro 4 - Relatório de Erros Identificados na Versão Alfa.....	57
Quadro 5 - Vantagens e Desvantagens dos Testes Remotos (Rubin & Chisnell, 2008).....	58
Quadro 6 - Perguntas de Perfil Social para o Teste de Usabilidade.....	59
Quadro 7 - Análise de Tempo para Realização da Tarefas.....	61
Quadro 8 - Respostas de Melhoria do Timeshare.....	63
Quadro 9 - Mudanças Futuras Previstas Para o Timeshare.....	69

LISTA DE ABREVIATURAS E SIGLAS

APK	Kit de Pacotes Android
BMW	Bayerische Motoren Werke
CLE	Expressão de Linguagem Comum
CLI	Interface de Linha do Cliente
CRUD	Criar, Ler, Atualizar, Deletar
CSS	Folhas de Estilo em Cascada
DEI	Departamento de Engenharia e Informática
DOM	Modelo de Objeto do Documento
HTML	Linguagem de Marcação Hipertexto
IA	Inteligência Artificial
ID	Identificação
IEEE	Instituto de Engenheiros de Elétrica e Eletrônica
SQL	Linguagem de Consulta Estruturada
VS	Visual Scheduler
UI	Interface de Usuário
UX	Experiência de Usuário
XML	Linguagem de Marcação Extensível

SUMÁRIO

1 INTRODUÇÃO.....	16
1.1. Justificativa e Problemática.....	17
1.2. Objetivos.....	18
1.3. Organização do Documento.....	19
2 FUNDAMENTAÇÃO TEÓRICA.....	21
2.1 Gerenciamento de Tempo.....	21
2.2 Um Calendário Visual para Agendamento de Reuniões em Grupo.....	22
2.3 Concepção de Software para Gestão de Horários Escolares.....	25
2.4 Estado da Arte.....	27
2.5 Flutter Framework.....	31
2.6 Comparativo Flutter vs React Native vs React JS.....	34
2.7 Firebase: Ecossistema de Serviços da Google.....	37
3 METODOLOGIA.....	40
3.1 Desenvolvendo com Flutter Framework.....	41
3.2 Estrutura de Objetos e Bancos de Dados.....	43
3.3 Interface de Usuário.....	51
4 RESULTADOS.....	40
4.1 Tela de Login.....	40
4.2 Tela Principal.....	44
4.3 Tela de Amigos.....	47
4.4 Telas de Grupo.....	49
4.5 Tela de Suporte e Configuração da Conta.....	53
4.6 Teste Versão Alpha.....	55
4.7 Teste de Usabilidade.....	58
4.8 Resultados do Questionário.....	61
4.9 Ajustes Após Teste de Usabilidade.....	64
5 CONSIDERAÇÕES FINAIS.....	68
6 RECOMENDAÇÕES.....	69

7 REFERÊNCIAS.....	44
APÊNDICE A - Formulário de Teste de Usabilidade.....	48
APÊNDICE B - Gráfico de Respostas do Questionário de Usabilidade.....	59

1 INTRODUÇÃO

O gerenciamento de tempo é definido como uma forma de decisão usada por indivíduos para estruturar, proteger e adaptar o seu tempo às mudanças de condições (AEON; FABER; PANACCIO, 2021, APUD AEON; AGUINIS, 2017). Como alternativa para gerenciar o tempo podem ser usadas as agendas, que por sua vez, passaram a não ser apenas um pequeno caderno de anotações com compromissos, horários e informações diversas. No mundo cada vez mais tecnológico, observa-se que a maioria das pessoas, de um forma ou de outra, já tem no mínimo um bloco de notas e, provavelmente, tenha acesso à Agenda Virtual. Em 1990 já se mostravam incentivos para agendamento online, técnicas e algoritmos para agendamento automático (BEARD; PALANLAPPAN, 1990). As tecnologias só têm avançado e os desenvolvimentos alcançados propagam mais esta ideia, desde a renovação à criação de novos processos para melhoria do gerenciamento temporal.

Assim sendo, as agendas fazem parte do cotidiano do adulto funcional que precisa gerenciar seus compromissos nos mais diversos contextos: acadêmico, social, familiar e profissional. Esses indivíduos precisam organizar ações e resolver problemas de forma eficiente. Acima de tudo, a gestão do tempo contribui para o bem-estar do ser humano (AEON; FABER; PANACCIO, 2020).

Para McGrath (1990 apud MCGRATH & KELLY, 1986; MCGRATH & ROTCH-FORD, 1983), há três tipos de problemas genéricos em ações coletivas: ambiguidade temporal, interesses e conflitos de tempo. Todos estes tipos de empecilhos para colaboração também podem ser encontrados em gestão de projetos. Como exemplo disto, aponta Nascimento (2017 apud ALMEIDA, 2014, p. xi), no seu estudo sobre gestão de tempo e a importância de projetos com cumprimento de prazos. Este evidenciou que 64,5% das empresas relatam que as principais dificuldades estão relacionadas a gerenciamento de tempo e, conseqüentemente, no atraso da entrega de projetos.

Neste contexto, empresas e instituições têm investido tempo e recursos no gerenciamento do tempo, embora muitas ainda enfrentem dificuldades nessa área. Existem diversas ferramentas e técnicas disponíveis, como Kanban, Getting Things Done e listas de tarefas, porém sua adoção depende de fatores como preferências pessoais. Além disso, observa-se certa limitação na disponibilidade de soluções para o gerenciamento de agendas compartilhadas, visto que muitas das ferramentas disponíveis oferecem funcionalidades distintas entre si.

Como alternativa têm-se o Google Agenda e o Doodle, as quais são ferramentas que permitem agendar reuniões em conjunto, mas de maneira total não oferecem muita versatilidade para fornecer acesso compartilhado com vários usuários, não permitem que se preencha de forma compartilhada várias atividades importantes no contexto acadêmico e, por fim, não há possibilidade de relacionar atividades com diferentes tipos de prioridade.

Sendo assim, este trabalho propõe o desenvolvimento de uma aplicação capaz de auxiliar no agendamento de compromissos, permitindo ao usuário selecionar quem terá acesso ao seu planejamento semanal. A aplicação trará opções de compartilhamento mais flexíveis do que as oferecidas por outras soluções disponíveis, utilizando o Flutter Framework como base tecnológica.

Além disso, a aplicação permitirá a comunicação e a colaboração em torno de compromissos agendados, facilitando o envio de links úteis para a execução de tarefas, como documentos do Google Drive, links de reuniões, guias e resultados colaborativos. Também será possível visualizar, de forma simplificada, as tarefas e atividades compartilhadas previamente agendadas.

1.1. Justificativa e Problemática

No cotidiano do adulto funcional ou do aluno de graduação, por exemplo, é comum haver dificuldades na gestão de horários. Esses desafios podem envolver o ajuste a um novo período letivo, a adaptação a novas atividades, o cumprimento de prazos de trabalhos ou avaliações, a reorganização dos horários de estudo, ou até mesmo o controle do tempo dedicado ao lazer (SOARES et al., 2022; OLIVEIRA et al., 2016). Além disso, muitas vezes existe a necessidade de se manter informado sobre a agenda de amigos ou colegas, seja para organizar momentos de estudo conjunto, saídas em grupo ou mesmo para facilitar a comunicação.

Esse tipo de problema pode ser minimizado por meio do compartilhamento de agendas. A ferramenta, além de manter o indivíduo informado sobre suas próprias atividades, também pode facilitar a comunicação com colegas, indicando os momentos disponíveis para realizar tarefas em conjunto (MCGRATH, 1990; ROGELBERG et al., 2014). No contexto universitário, essa necessidade não se restringe apenas aos alunos, mas também envolve professores que desenvolvem projetos colaborativos, orientadores que precisam agendar

reuniões com seus orientandos, entre outras situações em que o alinhamento de horários é fundamental (VISSON, 2009 apud YOUNG, 2002).

No entanto, não basta que a agenda compartilhada esteja disponível para visualização por todos. É fundamental que haja mecanismos de controle de acesso, incluindo restrições sobre quem pode comentar, prazos de expiração para o compartilhamento e regras específicas de visibilidade.

Além de funcionar como uma ferramenta para o desenvolvimento pessoal, propõe-se o uso do Flutter no desenvolvimento da aplicação, inicialmente voltada para o sistema operacional Android. Posteriormente, com a reutilização da mesma base de código, o aplicativo poderá ser adaptado para outras plataformas, como Windows, Linux, iOS e Web.

1.2. Objetivos

As subseções a seguir contêm os objetivos geral e os específicos deste trabalho.

1.2.1. Objetivo Geral

Criar um aplicativo que melhore atividades em grupo, que tenha boa comunicação, permita o acesso rápido e fácil para ações cotidianas, utilizando o pacote de desenvolvimento Flutter Framework.

1.2.2. Objetivos Específicos

- Realizar uma revisão bibliográfica a respeito da importância do gerenciamento de tempo;
- Estudar as características do desenvolvimento de software *mobile*, com ênfase no Framework Flutter, na linguagem de programação Java para funcionalidades do *back-end* e na linguagem Dart para funcionalidades do *front-end*;
- Realizar testes preliminares com estudantes para avaliação do sistema; e
- Analisar e validar as funcionalidades e requisitos necessários para o desenvolvimento do sistema.

1.3. Organização do Documento

Este trabalho encontra-se dividido e estruturado por capítulos, detalhados a seguir. No Capítulo 1 foi apresentada a introdução e a justificativa, as quais apresentam a ideia inicial do

trabalho e que pode-se identificar a problemática que deseja-se resolver. Já no Capítulo 2 é apresentada a fundamentação teórica e/ou o estado da arte, a fim de relacionar a dinâmica do gerenciamento de tempo de forma fácil e acessível aos usuários para garantir o controle do acesso seguro, rápido e que permita o compartilhamento entre a rede de usuários. No Capítulo 3, apresenta-se a metodologia, a qual indica as etapas que foram utilizadas para obter os resultados estabelecidos que são utilizando a tecnologia framework Flutter, na linguagem de programação Java para funcionalidades do *back-end* e na linguagem Dart para funcionalidades do *front-end*. No Capítulo 4 será apresentado os resultados e discussões, a partir da criação da interface com o *Software* para dispositivos móveis, por fim, poder finalizar o Aplicativo, evidenciando vantagens e desvantagens. Por último, no Capítulo 5 apresentam-se as considerações finais e recomendações.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada a fundamentação teórica do trabalho proposto, o qual está dividido em seis seções: 2.1 Gerenciamento de Agendas; 2.2 Um Calendário Visual para Agendamento de Reuniões em Grupo; 2.3 Concepção de Software para Gestão de Horários Escolares; 2.4 Comparativo Flutter vs React Native vs React JS; 2.5 Flutter Framework; 2.6 Firebase: Ecossistema de serviços do Google..

2.1 Gerenciamento de Tempo

O gerenciamento de agendas está relacionado diretamente com o tempo, já que é através de uma agenda que pode-se alocar o tempo para cada atividade que se pretende realizar (Dicionário Priberam, 2024). Dessa forma, pode-se gerenciar o tempo não só para as atividades, mas o que estas atividades irão causar na vida pessoal de um indivíduo ou grupo.

Em um dos estudos mais relevantes para gerenciamento de tempo, a meta-análise de Brad Aeon, Aïda Faber, Alexandra Panaccio (2020) focou pesquisa levando em consideração mais de 150 estudos sobre a área. Analisando o impacto do gerenciamento de tempo na performance profissional, acadêmica e no bem estar pessoal. Ainda nesta pesquisa investigaram os picos de publicação sobre o assunto, quantificaram a parcialidade dos pesquisadores e tentaram ao máximo entender os aspectos que se contextualizam aos participantes dos estudos.

Ao todo foram 53.957 participantes dos estudos escolhidos, fazendo com que os resultados sejam quantitativos e portanto mais preciso que estudos individuais. Isto quando se trata de questionar o que é gerenciamento de tempo, e se funcionam nos três fatores de impacto elencados pelos autores (Aeon; Faber; Panaccio, 2020).

Nos resultados foi constatado que o gerenciamento de tempo impacta o bem estar das pessoas e, particularmente, a satisfação de vida. Por outro lado, não foi notado aumento significativo na performance, e sim que há diversos fatores que alteram o efeito do gerenciamento de tempo. Os autores ainda apontaram que os próximos estudos devem focar nestas variáveis internas e externas que afetam o indivíduo, a fim de descobrir as causas de maneira mais precisa.

Atualmente tem-se buscado mecanismos para melhorar o gerenciamento de tempo através de aplicações, de maneira a otimizar o tempo dos indivíduos ou grupos. Isto é o que se explicará no próximo tópico.

2.2 Um Calendário Visual para Agendamento de Reuniões em Grupo

Agendar reuniões sempre foi um desafio, uma vez que a confirmação prévia dos participantes é essencial. Um equipamento, uma pessoa, ou até mesmo o ambiente são participantes de uma reunião e acima de tudo, o horário desta reunião é o fator mais impactante. Todos os participantes em algum momento estarão disponíveis, mas é essencial para uma reunião que estejam disponíveis em um determinado horário aceito por todos.

O Visual Scheduler (VS), ou em português, Agendamento Visual, é uma ferramenta criada por David Beard e Murugappan Palanlappan (1990) utilizando a linguagem de programação Xerox Smalltalk. A ferramenta utiliza de uma rede ou um sistema de arquivos compartilhados, que através de uma agenda em arquivo no formato “.schedule” salvo no diretório do usuário de cada participante, é lida, salva e editada pelo programa Visual Scheduler (BEARD, 1990).

A aplicação possui uma interface simples, com a tecnologia da época o mouse possuía três botões e o VS só precisava de dois botões para criar um espaço de reunião na agenda. Arrastando o botão esquerdo sobre as janelas dos horários, o usuário pode selecionar mais de um horário na agenda, podendo escolher entre 8 horas da manhã até às 18 horas da noite. Escolhendo um determinado horário para a reunião o usuário também pode arrastar as bordas da reunião agendada para aumentar sua duração.

Para selecionar um evento já criado na agenda o usuário poderá clicar com o botão direito e editar a caixa de detalhes, este que aparece em um campo abaixo da visão geral do calendário. O campo informa o nome do evento, a data, um campo para informações públicas onde também informa o horário e o grau de dificuldade para remarcar a reunião. Também informa a frequência da reunião, por quem foi agendada, comentários gerais do evento, entre outros. Ainda em outro campo ao lado, é reservado informações privadas do evento que pertence a cada usuário, como anotações pessoais e somente o usuário que escreveu pode visualizar.

Ao clicar no botão central do mouse o usuário pode definir a prioridade de cada evento, onde 1 é a prioridade mais baixa, e 5 a mais alta. Esta prioridade é percebida visualmente com a diferenciação de sombra sobre cada evento, diminuindo a transmissão de luz em 20% a cada aumento de prioridade, ou seja, quanto maior a prioridade mais escura é a janela do evento.

Para agendar uma reunião, o usuário deve selecionar os participantes e assim

visualizar a agenda composta destes indivíduos. A visualização desta agenda composta permite mesclar as agendas de vários participantes para que o usuário possa escolher um horário que todos possam participar. Esta agenda composta também é importante no caso de um dos participantes não serem essenciais, então o criador da reunião pode considerar horários que mais usuários essenciais possam participar sem comprometer outros eventos importantes.

Outra função especial do VS é o algoritmo de prioridade da agenda composta, onde a prioridade é alterada de acordo com a seguinte equação:

$$x_c = 5 - 5(1 - x_i)$$

Nesta equação, o x_c é a prioridade composta do evento, e x_i é a prioridade individual de cada calendário que faz parte do grupo agendas analisadas. Após o cálculo, se x_c for um número fracionário ele é arredondado para o número inteiro mais próximo.

Para mostrar a função, supondo-se que haja três horários para relacionar uma atividade composta: $p_1=5$ e $p_2=3$. Para realizar o cálculo, considerando $x_i = p_1 + p_2$:

$$x_c = 5 - [5(1 - (85))]$$

$$x_c = 5 - 5(1 - 1,6)$$

$$x_c = 5 - 5(-0,6)$$

$$x_c = 5 - -3$$

$$x_c = 8$$

Diante do resultado acima, temos que x_c é igual a 8. Seguindo a regra de que o resultado maior possível é 5, temos que a prioridade composta é 5. Para outro caso, onde: $p_1=1$ e $p_2=1$ e considerando $x_i = p_1 + p_2$:

$$xc=5-[5(1-(25))]$$

$$xc=5-51-0,4$$

$$xc=5-5(0,6)$$

$$xc=5-3$$

$$xc=2$$

Neste caso, pode ser observado que o valor de xc é de 2, esta é a prioridade composta. Por isso, temos que a prioridade final desta atividade representa a importância do evento para ambos os donos da agenda. Isso é de bastante relevância por dois motivos principais: primeiro, quanto maior o grau, maior a dificuldade para dois ou mais usuários possam remarcar o evento; segundo, os espaços vazios demonstram horários que pode ser preenchido, ou que casos com determinada prioridade composta podem ser re-agendados, se necessário.

Dentro das funcionalidades apresentadas há vários motivos para utilização do VS, dessas razões as principais são: rapidez e menor taxa de erros em comparação com agendamento manual de reuniões; a utilização do método de priorização composta onde permite a flexibilidade para agendar reuniões considerando múltiplas agendas; e a usabilidade como ferramenta de grupo.

Para comprovar a plausibilidade desses argumentos, foram realizados experimentos que compararam o agendamento de reuniões por meio do *Visual Schedule* com técnicas manuais, tanto visuais quanto textuais. Os resultados foram analisados estatisticamente com o uso do procedimento de Newman-Keuls (WINER apud BEARD, 1990). A análise revelou que, embora todos os métodos tenham se mostrado eficazes, o *Visual Schedule*, por ser uma técnica visual automatizada, apresentou um desempenho significativamente superior em termos de velocidade e menor propensão a erros. Além disso, o método visual manual também se mostrou mais rápido e eficiente do que o método textual.

Este aplicativo se mostrou bastante útil diante do contexto que o projeto descrito neste documento busca se desenvolver. Já que um aplicativo capaz de utilizar algoritmos de prioridade se demonstra único na busca por automação de uma agenda pessoal, ao criar horários para realizar eventos e cruzando referências sobre outras atividades já citadas.

Outro aspecto interessante sobre a aplicação do Visual Scheduler é que o desenvolvimento de tal conceito foi feito antes da revolução tecnológica causada pelo acesso à Internet massificado, e ainda assim se manteve bastante a frente com seu objetivo.

Considerando que nos dias atuais muitos aplicativos ajudam no agendamento de atividades, mas que nenhum pode-se dizer que conclui uma automação eficiente deste processo mesmo com maior quantidade de tecnologias disponíveis.

Para este projeto especificamente, o planejamento do Visual Scheduler mostrou alguns dos fatores que mais tarde serão importantes para o desenvolvimento do Timeshare, como a divisão de participantes, que mostra a possibilidade de não ser somente pessoas físicas, mas também equipamentos e locais. O VS mostra que é importante a sincronização desses horários com a prioridade composta, o próprio esquema de prioridades abre um leque de possibilidades para um usuário que deseja além compartilhar sua agenda, mas também mostra que a organização é um dos fatores mais importantes, especialmente em um ambiente de trabalho como estudado e apontado no artigo.

2.3 Concepção de Software para Gestão de Horários Escolares

Durante seu estágio na Universidade de Coimbra, o autor e engenheiro, Alexandre R. S. Correia foi responsável por escrever a dissertação, "Concepção de Software para Gestão de Horários Escolares" em 2009. Buscando realizar melhorias no sistema de comunicação e elaboração de horários no Departamento de Engenharia Informática (DEI), da mesma universidade, Correia analisou o problema com o intuito de levantar os requisitos necessários.

Com isto, o autor notou que havia algumas ferramentas que poderiam resolver o problema. Ele analisou estas ferramentas e as classificou em grupos que representavam as mesmas características, a partir disto as aplicações UniTime e Fet foram escolhidas para representar os seus grupos.

Como critério, foram avaliados os seguintes parâmetros:

- Tipos de licença: gratuita ou paga;
- Sistema Operacional: Windows, Linux, Mac, outro;
- Dificuldade de instalação configuração do software;
- Grupo de usuários da aplicação;
- Formas de elaboração do horário;
- Integração com outras aplicações;
- Suporte Técnico;
- Disponibilidade como Sistema Web;
- Eficiência no cadastro de parâmetros;

- Flexibilidade ao Departamento de Engenharia Informática.

Além das aplicações UniTime e Fet, o engenheiro previu certas dificuldades e também considerou a possibilidade de criar uma aplicação específica para o Departamento de Engenharia Informática.

Avaliando as ferramentas diante dos critérios, o autor concluiu que criar a aplicação que obedeceria aos requisitos do DEI seria a melhor opção, já que as ferramentas avaliadas possuíam muitas variáveis que deixam a desejar o desempenho do Departamento, assim foi necessário utilizar desta abordagem pensando no seu futuro.

A aplicação passou a ser estudada e levou-se em consideração realizar a implementação em outros departamentos da universidade. Durante o desenvolvimento do software o engenheiro demonstrou ótimas técnicas para o levantamento de requisitos, arquitetando o projeto com diagramas, protótipos, designs e métodos.

Aos participantes do projeto, o autor desenvolveu classes de Curso, Disciplina, desenvolveu tipos de aula, estudou os grupos de usuário, as características de uma sala, e levantou o conjunto de restrições a serem aplicadas no sistema. Isso é de extrema necessidade devido ao projeto necessitar de funcionalidades específicas, como regras de negócio mais restritas e outras mais flexíveis. Por exemplo:

□ Restrições Rígidas (*Hard Constraints*).

- o HC1 - Quando pertence a mais de um curso, a disciplina terá um horário comum a todos os cursos, no mesmo período.
- o HC2 - Aulas de uma disciplina para cursos diferentes não podem ocorrer no mesmo espaço de tempo.
- o HC3 - Duas aulas diferentes não podem ocupar a mesma sala no mesmo espaço de tempo, no mesmo período.

□ Restrições Flexíveis (*Soft Constraints*).

- o SC1 - Utilizador (professor, aluno, etc) não pode ser alocado para duas salas ao mesmo tempo.
- o SC2 - Cada aula deve ter um utilizador presente.
- o SC3 - O utilizador deverá ministrar a aula no seu período escolhido.

Dentre diversas restrições sobre o departamento o autor da dissertação demonstrou uma metodologia bastante interessante para documentar o trabalho e tornar o

desenvolvimento do software mais eficiente. Demonstrando qualidade necessária em qualquer gerenciamento de projeto e especial a um que atendesse a restrições tão específicas.

2.4 Estado da Arte

Para entender melhor como o Timeshare pode ajudar seus usuários é importante analisar as ferramentas disponíveis no mercado, assim entendendo o que deve ser esperado e como melhorar a aplicação desenvolvida com este trabalho. É isso que será discutido neste tópico.

2.4.1 Google Agenda

Dentre as ferramentas de gerenciamento de tempo disponíveis para Android, a mais proeminente é o aplicativo Google Agenda (Revista Tech, 2025). Por ser desenvolvido pela empresa Google que já possui bastante história na internet e está presente em serviços como Gmail, YouTube, Google Search, entre diversos outros. Isto torna o Google Agenda mais desejável para pessoas que precisam compartilhar sua agenda com outros indivíduos.

Entre suas forças, o Google Agenda permite a sincronia com o Google Meet, ferramenta de reuniões online usada majoritariamente no meio acadêmico e profissional. Especialmente no Brasil onde cresceu 275% em 2021 (Pancini; 2021), majoritariamente devido ao advento da pandemia que impossibilitou os encontros presenciais. Esta ferramenta permite que usuários possam marcar reuniões e agendar diretamente nos e-mails dos usuários, assim participando do dia-a-dia dos indivíduos.

O Google Agenda também é um aplicativo instalado diretamente em diversos dispositivos móveis já de fábrica, funcionando como calendário padrão do celular mesmo que não necessariamente para agendar reuniões. Também possui as opções de visualização mensal, semanal e diária do calendário. Além destas há opção de mostrar a cada 3 dias, e de Agenda que mostra somente os eventos sem os dias vagos.

Há opções quando o usuário deseja adicionar à agenda, sendo: Evento, Tarefa, Local de trabalho, Fora do escritório. Evento sendo a opção que permite uma customização maior, tem o formulário do título do evento, seguido de instâncias pré-definidas e em qual conta está sendo cadastrado. Após isto o usuário pode informar as restrições de horário, a duração do evento, em qual zona temporal ocorrerá e se este evento irá se repetir. Também é possível adicionar participantes, criar relação com reuniões do Google Meet, definir o local, criar alertas da atividade e a cor da atividade enquanto aparece no calendário geral. Ainda é

possível adicionar a descrição detalhada e anexar arquivos do Google Drive.

Há ainda funcionalidades para editar os eventos ao clicar na atividade em seu dia do calendário. Ao lado, a local de trabalho e fora do escritório permite descrever de forma rápida situações de trabalho em que o usuário possa estar ou não, disponível. Já as atividades cadastradas com participantes serão notificadas a partir do email do usuário e aparecerão em seu calendário pessoal. Em configurações também há possibilidade de compartilhar a agenda com usuários, indo em configurações e adicionando o email do indivíduo.

2.4.2 Doodle e Calendly

O aplicativo Doodle é uma ferramenta bastante interessante para designação de horários em grupo, possuindo 3 formas relevantes de auxiliar o usuário a organizar sua agenda (Doodle, 2025):

- Criação de enquetes de tempo, chamado de “Group Poll” que pode significar “Enquete em Grupo” (tradução própria). Qualquer usuário pode votar sobre qual alocação de tempo se encaixa melhor com sua agenda e disponibilizando um link para esta votação;
- Também é possível criar “One-on-one”, ou “Um-a-um” (tradução própria), que cria uma “enquete” com um usuário específico;
- E o “Booking Page”, que pode significar “Página de Agendamento” (tradução própria), e que permite o usuário organizador abrir dias em horários específicos onde outros indivíduos podem escolher seu horário de agendamento.

O que faz estas funcionalidades ainda melhores é capacidade de votar nos horários sem precisar ter uma conta Doodle e também é possível sincronizar o calendário do Google Agenda com os eventos cadastrados, tornando a integração com outros serviços mais fáceis (Doodle, 2025).

Apesar de suas forças, o Doodle possui algumas fragilidades marcantes já que o aplicativo *mobile* não possui o idioma Português em suas funcionalidades, encontrando somente a apresentação do site em português. A ligação com o Google Agenda só pode ser feita a partir da web sem redirecionamento a partir do aplicativo. Além disso, o Doodle se torna mais viável para um ambiente profissional visto que possui pouca utilidade para agendar tarefas pessoais. Portanto, o aplicativo serve mais como complemento de outras aplicações.

Por sua vez o Calendly é uma ferramenta bastante utilizada para criação de reuniões,

facilitando o agendamento para usuários e grupos. Utilizada por 86% das “500 empresas da Fortune” (Calendly; 2025. Tradução própria), está presente em um grande escopo e importância das ferramentas de agendamento em grupo.

Similar ao Doodle, os usuários podem escolher um tempo de disponibilidade para receber agendamentos de outros indivíduos, assim não precisa realizar discussões de tempo e ambos podem marcar para um tempo que estejam disponíveis.

2.4.3 Análise Comparativa Com Outros Aplicativos

Nesta seção serão comparadas as funcionalidades dos aplicativos mencionados e como estes se equiparam em relação ao Timeshare, a fim de validar algumas características inovativas e melhoradas.

Quadro 1 - Comparação entre ferramentas de gerenciamento de tempo.

Funcionalidade/ Aplicação	Google Agenda	Doodle	Calendly	Timeshare
Privacidade	Suporte intermediário.	Suporte básico.	Suporte básico.	Suporte aprimorado.
Gerenciamento de Atividades	Suporte intermediário.	Não possui.	Suporte básico.	Suporte aprimorado.
Gerenciamento de Grupos	Suporte básico.	Não possui.	Não possui.	Suporte aprimorado.
Idioma Português	Suporte aprimorado.	Suporte básico.	Não possui.	Suporte aprimorado.

Fonte: Autoria própria.

A privacidade dos eventos cadastrados no Google Agenda se baseia em proteger a conta do usuário e escolher quem possui acesso às suas atividades. Porém, não possuem discernimento em quem pode marcar um evento na agenda, apenas excluindo quando for indesejado. Já as atividades do Timeshare são armazenadas por agendas de um usuário ou grupo. No caso de grupos, o gerenciamento das atividades é compartilhado com os outros membros, e na agenda individual, apenas a visualização é compartilhada com outros usuários. Esta visualização depende da privacidade ou status de amizade de um indivíduo com outro, podendo ser visualizada por amigos, mas também alternando entre privado e público para qualquer outro usuário. Para o Doodle e Calendly, não há atividades em si, porém há eventos

no calendário, com Calendly focando em reuniões e o Doodle em enquetes e manuseio de horário.

Para o gerenciamento de atividades, o melhor entre os aplicativos terceiros é o Google Agenda que possui início, fim, descrição, título e uma gama de atributos que permite a customização da atividade. Porém, com o Timeshare não só a privacidade de quem pode influenciar sua agenda ou visualizá-la, a aplicação traz o sistemas de prioridades e completude da atividade de forma integrada, diferente do Google Agenda que depende do Google Tarefas, um outro aplicativo.

O gerenciamento de grupos do Google Agenda é feito por meio de organizações, mas de forma restrita pois necessita do e-mail organizacional para participar destas atividades. De outra forma é necessário adicionar cada usuário, sem filtrar o ambiente de eventos ao qual cada usuário pertence. Para o Calendly, por ser voltado para reuniões empresariais faz com que o sistema de agendamento de reuniões seja bastante útil, embora não atenda as necessidades individuais como Google Agenda e Timeshare. O mesmo para Doodle que foca em atividades conjuntas e necessitando de pelo menos um outro indivíduo para justificar suas utilização.

Por fim, somente o Google Agenda possui suporte completo para língua portuguesa brasileira, com o Doodle possuindo idioma parcial, e o Calendly somente na página de apresentação do site (até a revisão deste trabalho).

2.5 Flutter Framework

O Flutter Framework criado pela empresa Google em 2018 (Flutter, 2024), tem o objetivo de reproduzir aplicações multiplataforma, ou seja, para Android, IOS, Windows, Linux e Web. Além destes sistemas operacionais também é possível criar aplicações para sistemas embarcados como relógios digitais e sistemas automotivos.

Assim, como exemplo do uso do Flutter é possível citar a empresa automotiva, Bayerische Motoren Werke (do português, Fábrica de Motores da Baviera), ou simplesmente BMW, a qual em 2019 passou a utilizar esta tecnologia para desenvolver o seu sistema integrado aos seus veículos. Focando em desenvolver seu sistema para iOS, viram tal demanda para outros ambientes operacionais crescendo, fazendo com que desenvolvedores da BWM se reunissem e assumissem o Flutter como tecnologia para desenvolver os novos aplicativos. Desta vez garantindo que as mesmas funcionalidades do sistema iOS estivessem disponíveis também para usuários Android (Flutter, 2024).

Utilizando da linguagem de programação Dart ao qual pertence a arquitetura Flutter, é possível criar aplicativos multiplataformas com menor quantidade de código. Principalmente se a outra opção for, por exemplo, criar um sistema em linguagens preferidas para cada plataforma como Javascript para Web, Kotlin para Android, etc (W3Techs, 2024; Kotlin, 2024). Isso necessitaria aprender múltiplas linguagens de programação, ou dividir equipes para cada projeto e por isso o Flutter se torna uma das melhores opções.

Contando também com algumas vantagens, o Flutter foi criado a 7 anos, o qual é endossado por um conglomerado mundial como a Google. Devido a ser uma tecnologia recente, possui bastante avanços tecnológicos, assim não sendo tão tradicional, traz consigo uma comunidade buscando produzir da melhor forma possível. Sua empresa fundadora também criou um ecossistema de serviços, o Firebase, destinado a atender muitas necessidades de *software*. Para implementar aplicações utilizando o Flutter Framework são necessários alguns conhecimentos, em especial a linguagem de programação Dart. Ao longo desta análise pode-se obter três principais ideias que foram aprendidas com o estudo sobre o Flutter e também sobre a linguagem Dart. Por isso, na próxima seção será apresentado algumas das características desta linguagem.

2.5.1 Linguagem de Programação Dart

Dart é uma linguagem de programação orientada a objetos, criada por Lars Bak e Kasper Lund, e desenvolvida pela Google (Medium, 2022). Em conjunto com Flutter se torna *client-based*, que significa ser centralizada no dispositivo do usuário ao realizar transações de dados para um servidor ou base de dados externa. Em tempo real pode ser transpilada (traduzida e compilada) para Javascript em sistemas web e realizar interoperabilidade com C, Java e Kotlin com suas bibliotecas nativas (Dart, 2024).

A tipagem de Dart é bem comum, parecido com Java e Javascript, possuindo por volta de 12 tipos de dados. Estes são:

- **Números (int, double):** int para números inteiros, e double para reais.
- **Strings (String):** para texto e sequência de caracteres.
- **Binário, ou Booleano (bool):** assumindo 0 para falso e 1 para verdadeiro, ou também *true* ou *false*.
- **Registros ((value1, value2)):** Classes imutáveis que possuem a vantagem de diminuir o código repetitivo (*boilerplate*).
- **Funções (Function):** Funções são blocos de códigos reutilizáveis para executar tarefas.

- **Listas (List):** Listas são conjuntos de objetos mutáveis, encadeados, de tamanho fixo ou dinâmico.
- **Sets (Set):** Set é uma lista não ordenada, não indexada, de elementos únicos.
- **Mapas (Map):** Mapas são listas de relação chave-valor, onde estes podem ser mutáveis e duplicados.
- **Runas (Runes; muitas vezes substituído pela API characters):** Runas são valores representados em Unicode, uma sequência de números inteiros escalados.
- **Símbolos (Symbol):** são valores de identificação únicos para entidades, tais como Funções, Classes, Getter (busca informação de classe), Setter (define informações dentro da classe), e nome de operadores.
- **O valor nulo (Null):** valor inexistente, sem alocação de espaço de memória.

Além destes tipos ainda é possível trazer outros conceitos que podem não estar presentes em todas as linguagens ou possuem papel diferente na Linguagem Dart. Alguns destes tipos especiais, são:

- **Dinâmico (Dynamic):** O tipo dinâmico é capaz de inferir através da variável atribuída qual o seu tipo natural. Por exemplo, uma sequência de caracteres cercada de aspas pode ser considerada uma String, um valor numérico sem ponto pode ser considerado Int.
- **Objetos (Object):** Todas as variáveis são um objeto.
- **Enumeração (Enum):** Enumerado é um tipo onde o usuário define um conjunto de valores constantes.
- **Future e Stream:** Tipos bastante utilizados na programação assíncrona, onde a linha principal de compilação não é sobrecarregada, assim são executados paralelamente. O tipo Future é carregado em paralelo, e a Stream é uma corrente de dados presente na compilação e em comunicação com o processador.
- **Iterável (Iterable):** O tipo usado para iterar o laço “for”.
- **Never:** Usado para funções não terminadas, onde sempre resultam em uma exceção, um erro.
- **void:** Usado para indicar o tipo de retorno de uma função, onde não há um tipo de dado como resultado, apenas uma cadeia de execução.

Há ainda muitas informações que somente são reconhecidas ao estudar a linguagem Dart a fundo. Em geral, Dart é a linguagem orientada a objetos escolhida pela empresa Google para hospedar a arquitetura Flutter e possui bastantes vantagens em sua implementação.

Durante o pré-projeto deste trabalho foi proposto o uso de outras linguagens de programação como Java, Javascript e casos como HTML e CSS. Porém, durante o aprendizado notou-se que não seria necessário dividir os aspectos do projeto em linguagens diferentes. Dart em conjunto com Flutter faz possível o desenvolvimento das interfaces de usuário e dos métodos operacionais da aplicação.

Na arquitetura Flutter, cada widget permite customizações onde os atributos do objeto são o que determina a aparência do aplicativo. Isso se torna semelhante a um formulário de propriedades, onde cada página pode ser preenchida de uma forma, utilizando opções pré-definidas que tornam o uso do Flutter mais rápido e fluido. Algumas destes componentes pré-definidos podem ser inserido a partir da injeção de dependências, visto no tópico a seguir.

2.5.4 Injeção de Dependências

Dependências são componentes reutilizáveis, desenvolvidos por terceiros, e que permitem um trabalho mais rápido e fluido. Elas contém bibliotecas e métodos, se tornando ferramentas importantes, desde implementação para tecnologias como SQL, e tendem a atender as necessidades de caso de uso.

Dentre as dependências utilizadas na construção do Timeshare, pode-se citar as dependências de integração do Firebase. Ao adicioná-la, incorporou-se às funcionalidades da base de dados do Firestore e as regras de autenticação do Firebase Auth, reduzindo bastante o trabalho de configuração.

Para realizar a adição de dependências, o desenvolvedor pode ir até o site do pub.dev, que é mantido pela Google e gerenciado pela equipe do Flutter/Dart. Composto de extensões da arquitetura, tanto oficiais quanto criadas pela comunidade de desenvolvedores, o site disponibiliza uma gama de funcionalidades prontas para serem utilizadas.

2.6 Comparativo Flutter vs React Native vs React JS

Em outro aspecto do desenvolvimento, em especial sistemas *web*, há algumas tecnologias utilizadas comumente. Por exemplo, usa-se Javascript, HTML e CSS para que cada uma desenvolva seu papel no funcionamento das páginas. O HTML é uma linguagem de marcação usada para estruturar o conteúdo da página e integrar outros componentes; o

JavaScript é uma linguagem de programação utilizada para adicionar interatividade através de programas executados diretamente no navegador, assim atuando no *frontend* e no *backend*; e o CSS é responsável pela estilização da página (HAVERBEKE, 2024).

Indo ainda mais a fundo pode-se incluir a utilização de arquiteturas do CSS, traduzido para “Folha de Estilo em Cascada”, que serve para atribuir estilos às propriedades do HTML. Outra arquitetura como Bootstrap que é capaz de diminuir a mão de obra do CSS, permitindo a utilização de componentes predefinidos para implementar responsividade e criação de websites primeiramente para dispositivos móveis (W3Schools, 2024).

Em 2024, segundo a pesquisa anual de desenvolvedores do StackOverflow (2024), a linguagem de programação mais utilizada no mundo é o JavaScript. Das arquiteturas mais utilizadas o React está em primeiro lugar entre os desenvolvedores profissionais (STATISTA, 2025), sendo a que procuram mais para aprenderem. React como um núcleo, serve para duas funções, cada um possuindo suas vantagens. O React JS para desenvolvimento interfaces web (BANKS; PORCELLO, 2023, p. 42), e React Native para interfaces de aplicações Android/iOS (DABIT, 2019, p. 78). Pela maior quantidade de sistemas serem baseados em *websites*, a programação partindo do JavaScript se torna mais fácil uma vez que React JS e React Native possuem poucas diferenças tornando mais viável para certos clientes que desejam uma aplicação *mobile*.

Com o Flutter a programação em base de código única se torna ainda melhor, servindo não só para *web* e *mobile*, mas em computadores também é compatível com Linux e Windows, mostrando diversidade muito maior para desenvolvimento. Até para sistemas embarcados, como por exemplo interfaces automotivas e relógios digitais.

A seguir, no Quadro 2 foi feito um comparativo entre estas tecnologias, podendo notar vantagens e desvantagens, e identificando o que se aplica para cada uma.

Quadro 2 - Comparativo Flutter, React Native, React JS.

Especificações	Flutter	React Native	React JS
Compilação.	Compilação nativa em aplicações <i>mobile</i> .	Comunicação em ponte através da API*.	Renderização depende do DOM**.
Interface.	<i>Engine</i> gráfica	Componentes	Utiliza HTML e

	própria, Impeller.	nativos, iOS ou Android	CSS.
Plataforma.	Multiplataforma.	<i>Mobile.</i>	<i>Web.</i>
Linguagem.	Dart.	JavaScript.	JavaScript.
Desempenho.	Alto Desempenho para <i>mobile.</i>	Bom Desempenho para <i>mobile.</i>	Ótimo desempenho para <i>web.</i>

Fonte: Autoria Própria

* - *Application Programming Interface*, traduz para Interface de Programa de Aplicação. Interage com o usuário criando uma interface para realizar as requisições de cada operação em um determinado sistema.

** - DOM, ou *Document Object Model*, traduz-se como Modelo de Objeto de Documento. É uma interface de programação que permite que scripts interajam com a estrutura do HTML/XML, possibilitando a modificação dinâmica dos elementos e de suas propriedades em uma página *web*.

Quando se trata de uma escolha de sistema, podem ser notadas qualidades entre as diferentes plataformas. Para um sistema *web*, basta o website ter responsividade e ele servirá para qualquer aparelho, ainda com o usuário desfrutando dos mesmos recursos. Porém, algumas funcionalidades ficam limitadas e criam a necessidade de um aplicativo móvel, onde o processamento pode ocorrer mais rápido e manter dados diretamente no aparelho. No que diz respeito a arquitetura do React Native, é escolhida principalmente por sua facilidade de implementação com código em JavaScript. Já para o Flutter, sua principal vantagem para os desenvolvedores é sua replicação multiplataforma, que a partir de uma única base de dados pode ser renderizado uma multitude de aplicações: para Windows, Linux, macOS, Android, iOS, e Web.

Em particular o Flutter possui uma vantagem maior com aplicações *mobile*, possuindo alto desempenho devido a sua compilação por produzir o código na linguagem nativa do sistema operacional e permitindo maior eficiência de processamento. O React Native necessita de sua API para realizar seu processamento de interface, e o Flutter possui sua própria engine gráfica. No Android o motor gráfico Impeller consegue renderizar a Interface de Usuário utilizando de técnicas de predição para fazer cache dos frames, e distribuir threads

na geração de frames únicos. Trazendo maior eficiência e suavidade nas imagens, animações e utilização de unidades de processamento gráfico.

As plataformas orientadas ao cliente necessitam de uma base de dados para armazenar as informações geradas. Portanto, com um plano de autenticação para validar a identidade dos usuários e armazenar dados será apresentado na próxima seção a solução para estes requisitos.

2.7 Firebase: Ecossistema de Serviços da Google

O Firebase é um ecossistema que possui grande compatibilidade com o Flutter. Há vários serviços prestados por este sistema e que são para este projeto, estes sendo os serviços de autenticação e de armazenamento de dados. Para apresentar estes dois serviços, este tópico será dividido em duas (2) subseções: 2.3.1 Firebase Authentication; 2.3.2 Cloud Firestore.

2.7.1 Firebase Authentication

Os serviços de autenticação podem ser implementados por vários tipos de bibliotecas das linguagens de programação, ou até pelo framework, porém o Firebase Authentication possui um diferencial indispensável. Tal versatilidade, se comprova por sua autenticação de *emails* e senhas cadastrados dentro da aplicação ou até por permitir importar as credenciais de um outro provedor, como a própria Google e seu Gmail, ou até mesmo da Apple para plataformas iOS.

O Firebase Authentication contém métodos de redefinição de senha independentes, autenticados por *links* enviados automaticamente para o email cadastrado e também autenticação em dois fatores que é uma tendência cada vez maior na segurança de aplicações. O Firebase Authentication também gera *tokens* de identificação únicos, que permite sua reutilização na base de dados do Cloud Firestore.

2.7.2 Cloud Firestore

Para as bases de dados disponíveis no Firebase, há também um serviço similar, em tempo real, que se chama Firebase Realtime Database, porém se torna mais adequado para outros tipos de aplicação. Como conversas em tempo real (*chats*) ou aplicações de Internet das Coisas (IoT). No caso do Timeshare, o Cloud Firestore atende as necessidades de armazenamento na maioria dos casos, visto que o aplicativo não necessita de serviço em tempo real e sim de transações específicas de dados.

Ao acessar o site <https://firebase.google.com/docs/database/rtdb-vs-firestore>, é disponibilizado um questionário de vantagens e comparações para que o cliente entenda qual base de dados melhor se encaixa com suas necessidades. O Cloud Firestore também conta com uma consolidação maior para o seu uso, e uma comunidade de desenvolvedores mais aberta para compartilhar as experiências de uso.

Das aplicações do Cloud Firestore, temos uma base de dados diferente das tecnologias mais comuns como o SQL (Structured Query Language). O Firestore mantém seus dados em “documentos”, e estes documentos que possuem uma identificação (ID), são separados por coleções (Firebase, 2024). Estas identificações são registradas como Strings (tipo de dado para texto), ao invés de tipos numéricos (int, long, bigint, serial), que é o caso encontrado em muitas documentações do SQL por sua característica natural de incrementar os IDs seriais (Microsoft, 2024; W3Schools, 2024). Em sua vez, o Cloud Firestore também cria seus IDs de forma automática, mas não limitando a numeração e sim de forma mais versátil com Strings (Firebase, 2024).

Além disso, outras características dos tipos de dados são diferentes, como os tipos aceitáveis pelo Cloud Firestore. Os tipos são:

- Int: Integer, ou número inteiro, também assume uma forma hexadecimal;
- Arrays: lista de objetos, ou outros tipos de dados aceitado;
- Boolean: ou Booleano, um tipo de dado binário que geralmente assume valor verdadeiro ou falso;
- Byte: uma informação representada em linguagem computacional, um valor binário de oito dígitos;
- String: sequência de caracteres, estes podem ser símbolos, letras, números.
- DateTime: usado para referenciar hora e data.
- Ponto Flutuante: seguindo o padrão técnico IEEE 754, o armazenamento de números reais em ponto flutuante (IEEE, 1985).
- Ponto Geográfico: por latitude, e então longitude, para indicar uma localização.
- Map: ou Mapa, um dicionário definido por um valor único e seu valor armazenado.
- Vetor: uma matriz acessada pela dimensão, e então por valor individual do elemento.
- Referência: identificação de um documento, coleção, ou caminho dentro do Cloud Firestore.
- NaN: valor ausente.
- Null: valor nulo.

Vale notar que matriz e vetor podem ser diferentes em linguagens de programação, mas na documentação do Cloud Firestore está definido como vetor, que seria unidimensional, e com características de uma matriz, que seria multidimensional (Cendron, s.d.). Dos tipos numéricos, é especificado na documentação que sempre serão armazenados como double, um valor numérico real (Firebase, 2024).

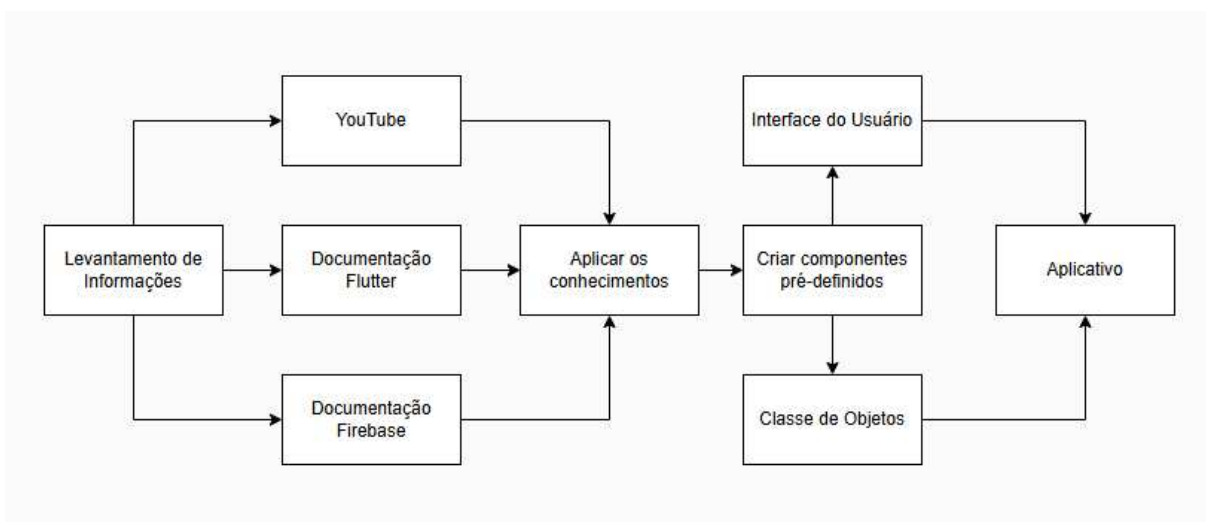
Dando continuação, no próximo capítulo será iniciado a discussão da metodologia deste projeto, buscando esclarecer quais técnicas e conhecimentos foram utilizados para criar a aplicação Timeshare.

3 METODOLOGIA

Neste estudo foi adotado um procedimento metodológico de caráter exploratório e descritivo, com abordagem qualitativa e quantitativa as quais se fundamentam através da coleta e análise de informações para auxiliar no desenvolvimento de *software*. Utilizando livros introdutórios, vídeo aulas explicativas, e comunidades de discussão acerca do Flutter Framework. Quanto à natureza da pesquisa, está fundamentada como aplicada e descritiva. Ela é aplicada, pois busca dispor de conhecimentos práticos para solucionar problemas específicos (PRODANOV, FREITAS, 2013). Também corresponde a pesquisa descritiva, pois foi construído um aplicativo para apoiar usuários no planejamento de ideias, atividades e/ou compartilhar compromissos.

Na Figura 1 é apresentado o fluxograma das atividades desenvolvidas até a execução do aplicativo. Inicialmente, realizou-se o levantamento de informações para criar uma base de conhecimento, o que seria necessário para criar um aplicativo de gerenciamento de agendas, sendo concluído durante o pré-projeto desta monografia. A partir do conhecimento reunido sobre a estrutura do Flutter, o plano de trabalho foi continuamente ajustado com base na assimilação dos componentes necessários ao projeto.

Figura 1 - Fluxograma de Atividades



Fonte: Autoria Própria

Ao pesquisar soluções para problemas específicos, como erros, ou formas melhores de aplicar a ideia, o website Stack Overflow foi indispensável. Assim como o YouTube, o site conta com bastante desenvolvedores que, ao contrário da plataforma de vídeos, os usuários podem obter respostas para dúvidas de maneira rápida e acessível sem uma produção maior

como a de um vídeo. A comunidade proporcionou redirecionamento eficiente para pontos específicos da documentação que contém conhecimento confiável e verídico, não apenas por depoimento. Estudando entre essas duas bases, YouTube e Stack Overflow, é possível obter conhecimento de várias linguagens de programação e arquiteturas, como Dart, Flutter, Javascript, Java, etc. Além disso, facilmente pode-se aprender sobre outras tecnologias de software relacionadas como SQL, Firestore, Docker, Github e muitas outras.

Após a consolidação dos conhecimentos as atividades foram direcionadas à fase de execução, e de uma maneira intercalada com mais estudo finalmente foram criados os componentes pré-definidos. Estes são a Interface de Usuário, estrutura de banco de dados e relação das classes de objeto com o sistema. A seguir, será discutido como foi utilizado o Flutter Framework, componente principal na criação do Timeshare.

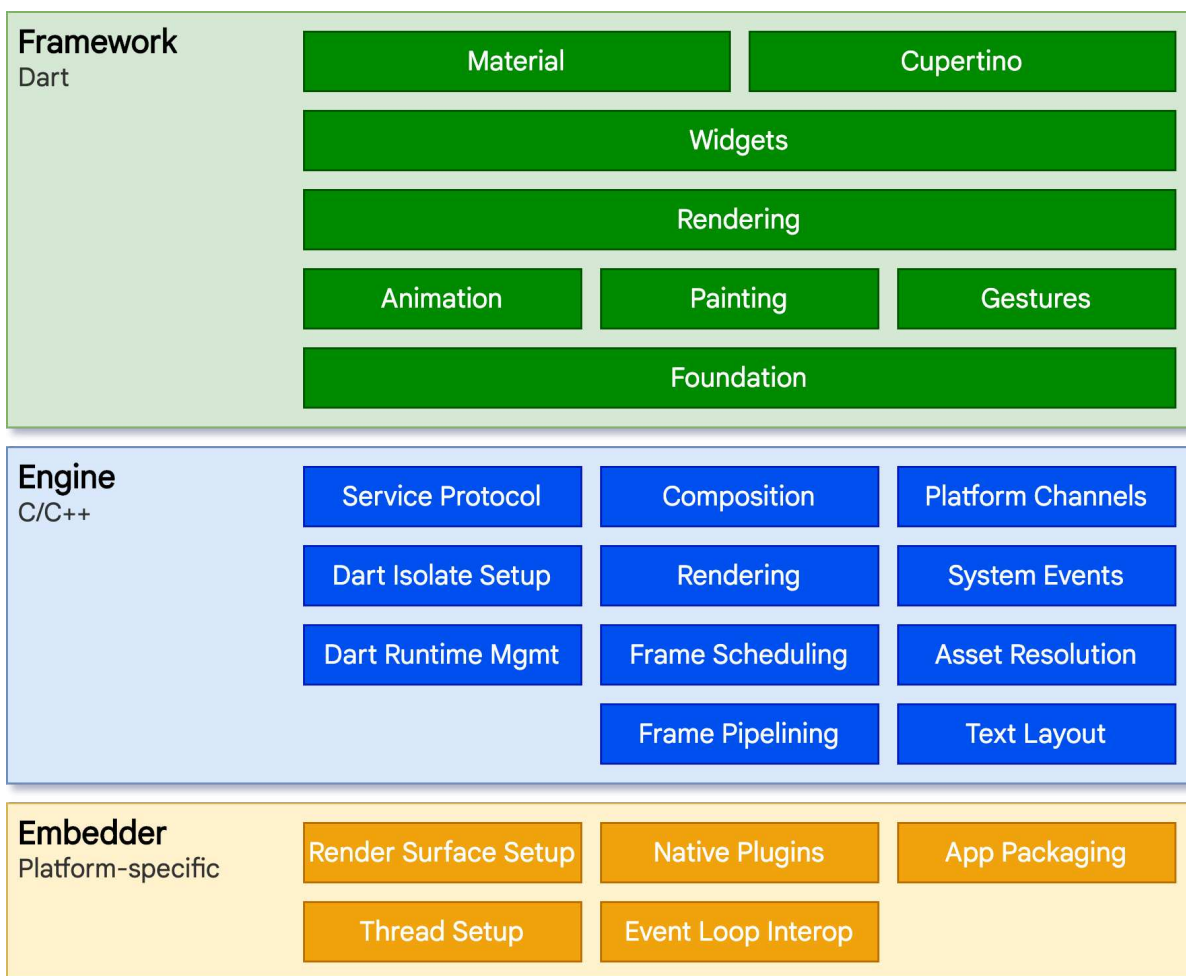
3.1 Desenvolvendo com Flutter Framework

Devido suas principais vantagens, isto é, compilação nativa, aplicações multiplataforma e uma única base de código, o Flutter apresentou-se como uma das melhores opções disponíveis no mercado. E com a pesquisa de material didático sobre seu uso, foi utilizado o livro *“Beginning Flutter: A hands on guide to app development”* (Do inglês, “Iniciante em Flutter: Um guia mão na massa sobre desenvolvimento de aplicativos”, tradução própria), e assim serviu a base dos conhecimentos para iniciar o projeto.

Buscando em plataformas digitais identificou-se uma comunidade ativa de desenvolvedores atualizados com as mudanças da tecnologia. Utilizando do YouTube e seus canais, tais como Flutterando TV (2024), e HeyFlutter.com (2025), foi realizada a reunião de práticas modernas de rotas de navegação, construção de interfaces e uso de ferramentas do ecossistema da arquitetura Flutter. Visto que esta não é uma maneira formal de obter conhecimento para uma pesquisa acadêmica, mesmo assim mostra valor prático em uma área que passa por mudanças constantes como a tecnologia.

Ao desenvolver a interface com o Flutter, cada elemento da tela pode ser entendido como um grande encapsulamento sem ser no sentido geral de programação, onde as classes se tornam protegidas em relação aos seus métodos e atributos. E sim que todos elementos são cercados por outros elementos. Para explicar isso, primeiro é necessário entender como inicializar uma aplicação em Flutter. É comum utilizar o widget “MaterialApp”, que faz parte do pacote “material.dart”. Ele é responsável por definir a estrutura básica da aplicação como o tema, rotas e navegação, seguindo os padrões do design Material na Figura 2 a seguir.

Figura 2 - Contextualização de uma aplicação Flutter



Fonte: Retirado do site de documentação do Flutter (docs.flutter.dev)

Construir uma tela em Flutter é simples, porém requer uma visualização mais atenta dos componentes da tela. Na camada do framework, mostrado na Figura 2 pode ser notado onde a estilização dos componentes gráficos estão localizados, assim Material sendo uma biblioteca destinada para aplicações Android, e o Cupertino para plataformas iOS. Os componentes do Material ou Cupertino são widgets, mas nem todo Widget pertence a essas bibliotecas. Estes componentes são renderizados pelos motores gráficos, como Impeller (Android, iOS, Desktop), Skia (Web, e outras plataformas). Os widgets são os parâmetros, e as imagens são geradas nos motores gráficos, cuidando dos protocolos de serviço, composição, agendamento de quadros (*frame*) e a resolução. Ainda para tarefas assíncronas são usadas técnicas como “*frame pipelining*”, que distribui a construção do quadro em outras *threads*. Na camada do *embedder*, são utilizados os quadros gerados pelo motor gráfico e então são apresentados através de APIs. O SurfaceView no Android, e CALayer no iOS.

Na construção é possível utilizar funções para componentizar e dividir cada item da tela, assim como também replicá-lo ao criar um *widget* customizado. Utilizando componentes pré-definidos como Container é possível moldar a estrutura do Material design, mantendo uniformidade na aparência da aplicação e recriando perfeitamente a visão do projeto.

Os *widgets* com filhos em arrays são importantes para criar um fluxo na aparência, e os *widgets* de filho únicos criam cada componente único da interface. Trazer listas de dados e iterar sobre cada instância para criar uma lista de objetos compreensíveis para usuários é uma das principais ferramentas de visualização presente. Descobrir as restrições de tamanho dos itens ao usar *widgets* expandidos para ocupar espaço a mais na tela e colocando os widgets de rolagem da tela para abrir mais espaço além do que o dispositivo oferece.

Quando estas nuances são compreendidas e testadas de modo que o desenvolvedor entenda os mecanismos do Flutter, assim começa o verdadeiro desenvolvimento do sistema. Conectando cada tela no sistema de navegação, inserindo em cada botão necessário e modificando o feedback necessário para a ação do usuário.

Ainda mais importante é o gerenciamento do contexto da aplicação, já que este se torna responsável por renderizar os componentes, gerir as rotas de navegação e salvar os atributos de cada tela. Atributos e componentes que abrem espaço à memória, quando fechá-los ou reiniciá-los. O Flutter é extremamente fluido neste aspecto devido aos erros serem apontados em compilação, mostrando um rastro (*stacktrace*) dos componentes afetados, assim sabendo imediatamente quando há restrições na estrutura criada.

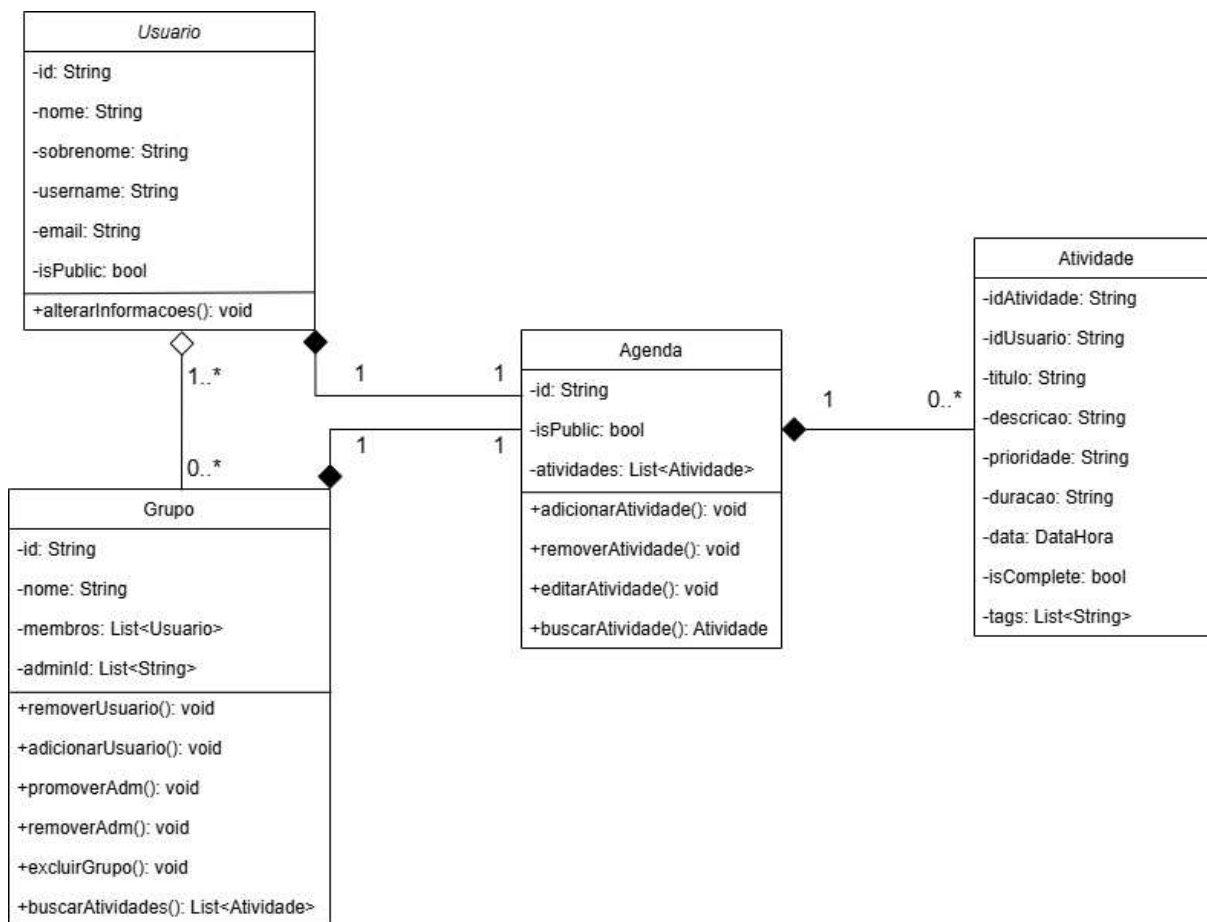
Para ser debatido a construção das interfaces utilizando Flutter, é necessário definir algumas classes de objetos na aplicação do Timeshare.

3.2 Objetos e Bancos de Dados

Neste tópico será abordado a estrutura de objetos da aplicação e também a estrutura do banco de dados. Utilizando da prática comum do Diagrama de Classes, é possível demonstrar a estrutura da aplicação e como os dados serão armazenados.

Objeto, em programação, é como referem-se a entidades que realizam um papel no sistema a ser criado. Um usuário pode ser um objeto, uma atividade ou até mesmo um objeto poder um objeto. Assim, as características destes objetos são definidas em atributos de uma classe e os métodos são as ações desse objeto dentro do sistema. Definido estes atributos, como é visto na Figura 3 a seguir, é possível reutilizar conceitos conhecidos para diminuir a manutenção de dados ao mantê-los em estruturas organizadas.

Figura 3 - Diagrama de Classes do Aplicativo Timeshare



Fonte: Autoria Própria

Na Figura 3 nota-se as características do Timeshare como um sistema, tratando do usuário que pode participar ou criar grupos. Além destas entidades possuírem uma Agenda, que então contém as atividades, separando cada aspecto do individual e grupal.

A primeira tela desenvolvida visando alcançar o núcleo da aplicação, foi a “Tela Principal” que permite realizar o “CRUD” das atividades. CRUD é um acrônimo em inglês, que significa “*Create, Read, Update, Delete*”. Uma maneira que desenvolvedores geralmente aprendem a gerenciar cada objeto na programação, também sendo muito utilizado em bancos de dados relacionais (Elmasri; Navathe. 2011). O objetivo do CRUD é que o usuário possa criar (*create*), visualizar ou ler (*read*), atualizar (*update*), e removê-lo ou excluí-lo (*delete*) durante suas atividades dentro de um sistema. Seguindo este raciocínio, o usuário do Timeshare deve criar, ler, atualizar e remover as atividades que ele deseja organizar e/ou compartilhar. Isto representado na Figura 3 são os métodos da Agenda, que por abstração se torna também os efeitos de um grupo. Ainda ressaltando esse aspecto, o usuário pode

completar sua atividade, não necessariamente a excluindo, mas trocando entre ocultar e visualizar, para rastrear seu progresso no dia-a-dia e saber quais ainda precisam ser feitas.

As atividades devem possuir um título onde descreve brevemente sua importância ou o que será realizado durante o evento; uma data indicando o dia a ser realizada; pode apresentar uma descrição detalhando o que irá acontecer; e a duração desta atividade, se ela irá tomar o dia inteiro ou somente algumas horas. Uma das funcionalidades diferenciais do Timeshare é a pesquisa por *tags*, termo utilizado comumente e que significa etiqueta (Morville, 2005). Cada usuário não controla uma única atividade, para isto é necessário criar uma classe que possa reunir todas estas atividades e outros atributos relacionais. Este é o papel da Agenda onde serão escritos os atributos de ID, sendo o mesmo do usuário, assim podendo cruzar referências sobre a identificação; a Agenda também é um atributo para indicar sua visibilidade, caso o usuário deseje manter suas informações privadas ou em aberto para quem quiser pesquisá-lo.

E por fim, serão estruturada as informações do usuário: o ID como forma de identificação; o nome do usuário; o sobrenome, pois algumas vezes é mais sensato referir-se a este de uma maneira mais curta e então separando o nome; o username para que usuários possam pesquisa-los de uma maneira mais familiar, visto que muitos usuários possuem este tipo de identificação determinado entre os amigos; e-mail para que as informações de autenticação estejam atreladas a este atributo; e uma revalidação da visibilidade da agenda para que usuários possam pesquisar por agendas públicas e privadas, a primeira instância é a do usuário, então para evitar mais pesquisas de agenda que retornarão inválidas, o usuário já possui este atributo para dar uma eficiência maior na pesquisa de banco de dados.

Com estes objetos o usuário já consegue criar, visualizar, atualizar e excluir suas atividades, porém, para o Timeshare é interessante o compartilhamento destas informações. Assim, agora serão estruturados os conceitos de grupo e de pedido de amizade, notando que não há uma classe que representa amizade, pois o único tipo de informação relevante sobre o amigo será adquirido a partir da representação do usuário.

Primeiramente será imaginado o conceito de pedido de amizade: o pedido é feito pelo solicitante e enviado até o solicitado. Como este será armazenado de forma que o solicitado precisa avaliar as informações do solicitante para saber se pode confiar ou se conhece o solicitante, então suas informações serão atribuídas no pedido de amizade, colocando as informações do objeto de usuário previamente mencionadas. Para o solicitado, no entanto, ele só precisa do seu próprio ID para confirmar que o usuário solicitado é ele mesmo, já

possuindo suas próprias informações. Assim o objeto de pedido de amizade terá: seu próprio ID; o ID do solicitado; e o objeto de usuário representando o solicitante.

Já a estrutura do grupo necessita de quatro atributos: o ID para identificação, criado automaticamente pelo banco de dados; o nome do grupo criado pelo usuário criador do grupo e alterável pelos administradores; uma lista de objeto de usuário, contendo os membros do grupo; e uma lista de ID dos administradores, caso necessitando das informações, estarão presentes na instância deste usuário na lista de membros.

Em relação aos métodos de cada objeto, estes serão descritos no tópico a seguir enquanto é discutido como as classes persistem no Banco de Dados.

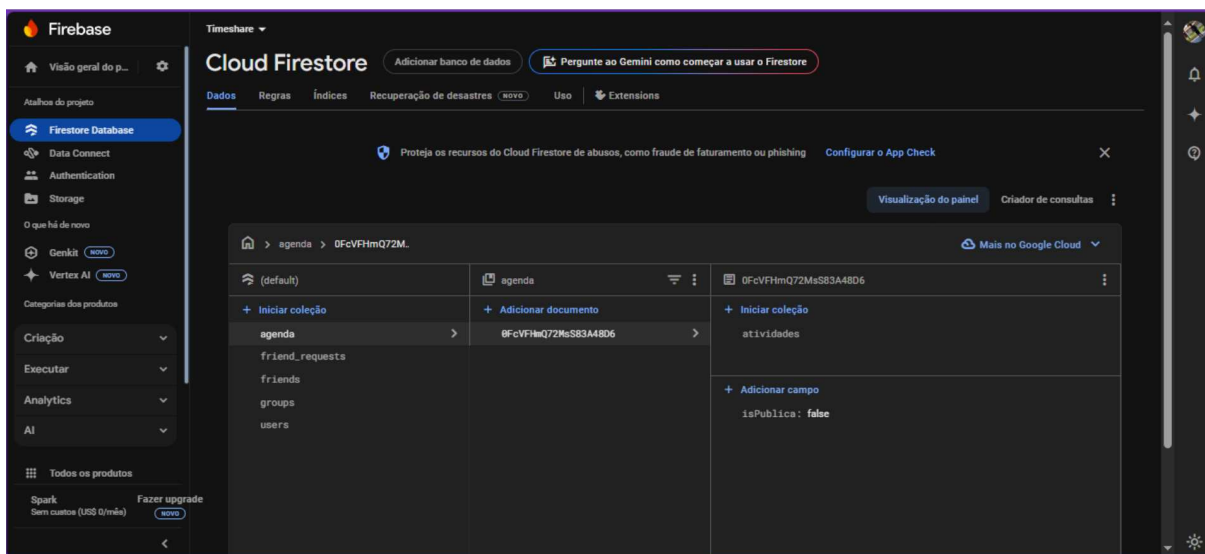
3.2.2 Banco de Dados

O banco de dados escolhido para realizar o armazenamento do Timeshare foi o Firebase Firestore. Criando um projeto no site deste serviço a integração ocorre de maneira simples, instalando o Firebase CLI, interface de comandos do console, como mostrado na documentação (Firebase, 2024). Ao instalar, o programa pode ser aberto a partir da pasta do projeto em que deseja implementar o Firestore, neste caso no arquivo raiz do Timeshare. Feito isto, o Flutter irá instalar algumas dependências do Firebase, criando a biblioteca necessária para ligar a base de dados, e fazendo login através da conta do Google que criou o projeto.

O banco de dados do Firestore funciona como lista de documentos em que cada um deste é escrito uma instância das informações, criando um ID aleatório ou definido manualmente no momento da inserção. Os documentos ficam armazenados em Coleções e cada documento pode ter uma subcoleção, com IDs não relacionados aos da coleção principal.

Neste *console* da Figura 4 nota-se as coleções criadas para o Timeshare separados em *agenda*, *usuários*, *friends*, *“friend_requests”*, e *groups*. Também é possível ver as abas de regras se referindo a segurança do banco de dados; índices que são criados para garantir performance da base de dados criando um mapa de cada atributo dos documentos para melhorar as pesquisas que possuem critérios; recuperação de desastres onde pode ser realizado backups; e o uso onde o Firestore monitora o banco de dados e mostra a cobrança do uso.

Figura 4 - Console de Cloud Firestore



Fonte: Firebase, 2025.

Outro aspecto, porém em certos casos seja o mais importante do banco de dados é o sistema de segurança chamado de regras de segurança do Firestore. Estas regras podem ser acessadas ou modificadas a partir do console na aba “Regras”. Estas instruções de segurança possuem sua própria linguagem, Common Language Expression (CLE, em português, Expressão de Linguagem Comum, tradução própria), mostrando como a base de dados pode permitir o acesso dos usuários que tentarem acessar cada coleção. Assim, será apresentado uma demonstração destas regras na Figura 5, a seguir.

Figura 5 - Regras de Segurança do Cloud Firestore

```

1 rules_version = '2';
2
3 service cloud.firestore {
4   match /databases/{database}/documents {
5
6     match /agenda/{agenda}{
7       allow create: if getAuth() != null && agenda == getAuth() || checkUserInGroupAdmins(agenda);
8
9       allow read: if getAuth() == agenda || resource.data.isPublica == true || checkUserInGroupMembers(agenda);
10
11      allow update: if agenda == getAuth() || checkUserInGroupAdmins(agenda);
12
13      allow delete: if getAuth() == agenda || checkUserInGroupAdmins(agenda);
14
15      match /atividades/{atividade}{
16
17        allow write: if agenda == getAuth() || checkUserInGroupMembers(agenda);
18
19        allow read: if agenda == getAuth() || isPublica(agenda) || checkUserInGroupMembers(agenda) || isFriend(agenda);
20
21        allow update: if agenda == getAuth() || checkUserInGroupMembers(agenda);
22
23        allow delete: if agenda == getAuth() || checkUserInGroupMembers(agenda);
24
25      }
26    }
27
28    function checkUserInGroupMembers(grupo){
29      return get(/databases/{database}/documents/groups/{grupo}).data.membros.keys().hasAny([getAuth()]);
30    }
31  }

```

Fonte: Autoria própria.

As regras ditam o protocolo de acesso às informações, seguindo a sintaxe *allow* (permitir), em seguida a operação que irá se aplicar e após os dois pontos serão especificadas as condições do acesso, começando com a expressão “*IF*”, que significa “se”. Além disso, há certas variáveis que podem ser utilizadas, como por exemplo “*request*”, que se refere ao pedido enviado pelo solicitador da informação. Dentro deste *request*, pode ser verificado o ID de autenticação do usuário, “*request.auth*” e a informação que ele está tentando acessar, com o atributo *resource*. Também pode ser verificado o recurso e todos os atributos, contanto que ele esteja no documento. Então se houver um campo no documento chamado de “*isPublica*”, o banco de dados pode acessá-lo para comparar na condição permitindo o acesso do usuário caso a condição seja validada. Isto é demonstrado na linha 9, Figura 5, onde a condição verifica se a agenda do usuário é pública e permitindo o acesso caso seja.

Também podem ser criadas funções que tornam o acesso a variáveis ou condições mais rápidas e melhorando a legibilidade das regras. A função é criada como visto na Figura 5, linha 28, com a sintaxe “*function*”, nome da função, a variável entre parênteses, e o código escrito entre as chaves, como visto a partir da linha 28. Já a inserção de informações no banco de dados é feita a partir da implementação no Flutter como pode ser notada na Figura 6, a seguir.

Figura 6 - Criação de uma Instância no Cloud Firestore



```

1 final _fire = FirebaseFirestore.instance;
2
3 createUser(Usuario user, bool agendaPub) async {
4   try {
5
6     final coll = _fire.collection('users').doc(user.id);
7
8     await coll.set(user.toMap());
9
10    await _fire.collection('agenda').doc(user.id).set({"isPublica": agendaPub});
11
12   } catch (e) {
13
14     throw FirebaseException(code: "database-failed", message: e.toString());
15
16   }
17 }

```

Fonte: Autoria própria

Primeiramente é inicializado uma instância do Firestore, visto na linha 1 da Figura 6, em seguida é feito a chamada desta instância no método de criação do usuário. Este método

na linha 3, recebe a instância do usuário a ser cadastrada e o valor da privacidade da sua agenda representado em booleano.

Recebendo a referência da coleção de usuário através da instância do banco de dados o método irá acessar o documento através do ID do usuário que é criado através do serviço Firebase Authentication. Com esta referência será utilizado um método para transformar os atributos da classe de usuário em um *Map* (dicionário). Este possui uma chave, neste caso o nome do atributo, e o valor da chave que será o valor do atributo na instância do usuário. Após isto o método da coleção, “set()”, irá modificar a instância, ou caso não exista, irá criar uma instância daquele documento dentro da base de dados.

Cada documento possui um ID não podendo haver um documento com o mesmo ID na mesma coleção ou subcoleção. Por isto, na coleção de “agenda” pode ser repetido o mesmo ID que na coleção de usuários, assim facilitando a busca das informações do usuário.

Para cada classe de objeto do Timeshare foi realizado um processo semelhante ao da Figura 7, na inserção no banco de dados. Em outros casos foram-se alterando os métodos de acordo com a operação feita. Por exemplo: operações de *update*, que apenas atualizam a instância do documento mas não criam se este não existir; a operação de *delete* que irá apagar o documento; e as operações de *get* que busca por instâncias.

Figura 7 - Função para Criar Grupo

```

1 createGroup(String nomeDoGrupo, Map<String, Map<String, dynamic>> addToGroup) async {
2   try {
3     Usuario user = await getUserInfo(id);
4
5     addToGroup.putIfAbsent(globalUser.id!, () => user.toMap());
6
7     final group = await _fire.collection('groups').add({
8       'adminId': FieldValue.arrayUnion([globalUser.id]),
9       'nome': nomeDoGrupo,
10      'membros': addToGroup
11    });
12
13    await _fire.collection('agenda').doc(group.id).set({'isPublica': false});
14
15    await _fire
16      .collection('users')
17      .doc(globalUser.id)
18      .collection('mygroups')
19      .doc(group.id)
20      .set({
21        'adminId': FieldValue.arrayUnion([globalUser.id]),
22        'nome': nomeDoGrupo,
23        'membros': addToGroup
24      });
25  } catch (e) {
26
27    authError(FirebaseAuthException(code: 'database-failed-create-group', message: e.toString()));
28
29  }
30 }
31 }

```

Fonte: Autoria própria.

Além disso há situações específicas de *delete* e *update*, mostrados na linha 8 da Figura 7, para quando é necessário modificar *arrays*. Estes casos precisam de uma classe chamada *FieldValue* que irá fazer a modificação do array sem excluí-lo inteiramente para substituir ou sem necessitar o envio do *array* inteiro já modificado. A classe *FieldValue* junto com o método “*arrayUnion()*” pode unir uma lista de objetos ao que está presente no banco de dados. Já o método “*arrayRemove()*” irá apagar a instância informada. Vale notar que no caso do “*arrayUnion()*”, se o campo do documento não for deste tipo ele irá transformá-lo em lista, assim perdendo os dados anteriores daquele campo.

3.3 Interface de Usuário

A maneira que a visualização das atividades pode ocorrer é através de um calendário, assim será necessário importar o pacote “*Flutter Calendar*”. Após adicionar este pacote o calendário é implementado como exibido na Figura 8, com a classe “*Table Calendar*” que significa Calendário em Tabela.

Figura 8 - Table Calendar

```

1 ...
2
3 TableCalendar(
4     eventLoader: (day) {
5         return atividades.where((element) => element.dataHora == day,).toList();
6     },
7     daysOfWeekHeight: 30,
8     calendarStyle: CalendarStyle(),
9     selectedDayPredicate: (day) => isSameDay(day, _selectedDay),
10    onDaySelected: onDaySelected,
11    headerStyle: HeaderStyle(
12        formatButtonVisible: false,
13        titleCentered: true,
14        titleTextFormatter: (date, locale) {
15            return DateFormat.yMMMM(locale)
16                .format(date)
17                .toString()
18                .capitalize();
19        },
20    ),
21    onPageChanged: (focusedDay) {
22        _focusedDay = focusedDay;
23    },
24    calendarFormat: CalendarFormat.month,
25    startingDayOfWeek: StartingDayOfWeek.monday,
26    locale: 'pt_BR',
27    availableGestures: AvailableGestures.all,
28    focusedDay: _focusedDay,
29    firstDay: DateTime(2024),
30    lastDay: DateTime(2100)),
31 ),
32 ...
33 ...
34 ...

```

Fonte: Autoria própria

Com a Figura 8 pode ser notada a configuração do calendário, permitindo uma grande quantidade de customização da aparência. Por exemplo, na linha 7 tem-se altura do dia da semana, seguido da classe de customização de estilo; na linha 11 há a classe de customização do cabeçalho, que pode alterar a visibilidade do botão de formato do calendário, centralizar o texto da data e formatar o texto de acordo com a localidade do usuário. Na linha 25 pode ser notado o formato pré-definido do calendário que está apontado para o formato mensal, onde a visualização ocorre por mês mostrando cada dia.

Também há configurações da interface onde é definido através de funções a data atual e a data em foco. Isto pode ser feito através do notificador de eventos do Flutter, que informa a mudança de data e fornece o valor para que possa haver outras funções em cima da data alterada.

Na Figura 9 a seguir, as mudanças de data ocorrem na seleção e na apresentação do calendário. É através desta funcionalidade já inserida na biblioteca do “FlutterCalendar” que permite a implementação de forma rápida.

Figura 9 - Calendário Flutter

Fonte: Autoria própria.

. Na Figura 9 é possível visualizar a aba de atividades e aba adicionar, assim na primeira serão listadas as atividades do dia selecionado e na outra será onde o usuário irá cadastrar estas atividades.

Ainda sobre a estrutura do Flutter é possível visualizar o encapsulamento previamente mencionado, visto que cada widget está dentro de outro. Foi notado a inicialização do Material App na Figura 2, porém ainda na página da agenda demonstrada na Figura 10 é possível visualizar isto em primeira mão.

Figura 10 - Encapsulamento de Widgets



Fonte: Autoria própria

Na Figura 10 o retângulo preto é a representação da coluna exterior que abriga os *widgets* em uma visualização vertical. Assim há dois *widgets* que compõem esta coluna, o calendário e o gerenciador das abas de atividade. A aba de atividades por sua vez possui uma representação com um quadrado vermelho, e neste *widget* há outra coluna com os *widgets* da barra de pesquisa e da lista de atividades. Na aba para adicionar há também uma coluna, porém com campos de entrada para digitar os atributos da atividade.

Então a composição ocorre cerca de quatro vezes, mostrando como os *widgets* suportam uns aos outros. Isto também mostra a relação da árvore de *widgets*, já que se imaginar galhos de uma árvore, o exemplo seria que a aba de atividades é um galho e a aba para adicionar atividades é outra. Quando o usuário acessa uma tela ele está percorrendo um caminho diferente da árvore, assim permitindo o Flutter mudar os *widgets* apresentados sem reconstruir a tela inteira.

4 RESULTADOS

Neste capítulo observam-se os resultados alcançados, dentre os quais serão apresentadas imagens do aplicativo em funcionamento e descritas suas funcionalidades. Para isto serão divididos em seis (6) subtópicos, a saber: 4.1 Tela de Login; 4.2 Tela Principal; 4.3 Tela de Amigos; 4.4 Tela de Grupos; 4.5 Tela de Suporte e Configurações de Conta; 4.6 Teste de Usabilidade; 4.7 Teste de Usabilidade; 4.8 Resultados do Questionário; e 4.9 Ajustes Após Teste de Usabilidade.

4.1 Tela de Login

Para tela de login foi pensado numa tela simples que executasse as funções de forma intuitiva e que pode ser comum entre os aplicativos. De forma que facilite a experiência do usuário, como mostrado na Figura 11.

Figura 11 - Tela de Login do Aplicativo Timeshare



Fonte: Timeshare em execução.

Observa-se como primeiro item da tela a logomarca, o símbolo seguido do nome, de forma que a identificação passe a boa experiência para o usuário. Para os campos de entrada

tem-se o email, o qual é a chave principal de identificação do usuário e em seguida a sua senha. Pode ser notado a label do que se trata o campo e ao clicá-lo estas labels sobem para ocupar um espaço na borda ao redor do campo de entrada. Pode ser observado o campo de senha em que há um ícone de olho sobreposto por uma barra, o que indica não poder ver a senha enquanto digitada, mas ao clicá-lo é possível revelar o conteúdo para que o usuário saiba se houve algum erro de digitação. Ambos os ícones de email, senha, e texto oculto foram fornecidos pelo Flutter.

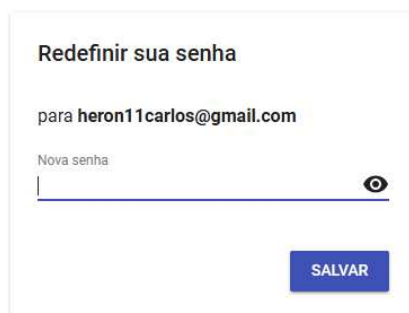
Após os campos de entrada tem-se um texto menos chamativo para indicar a funcionalidade de recuperar a senha. O texto representa uma pergunta intuitiva, ao pensar em uma resposta positiva, o próprio usuário entenderá que ao clicar conseguirá recuperar a sua senha, conforme a Figura 11.

Após clicar no texto, será acionado um *popup*, isto é, uma caixa de texto que diz, “Verifique seu e-mail para mudar a senha”, em seguida um botão para sair do *popup*. No e-mail do usuário receberá um texto instrutivo de como mudar a senha. O e-mail é enviado pelo próprio serviço Firebase Authentication do Google, mostrado na Figura 12, o qual possui vários protocolos de segurança para evitar marcação de *spam*.

Figura 12 - Email de Recuperação de Senha

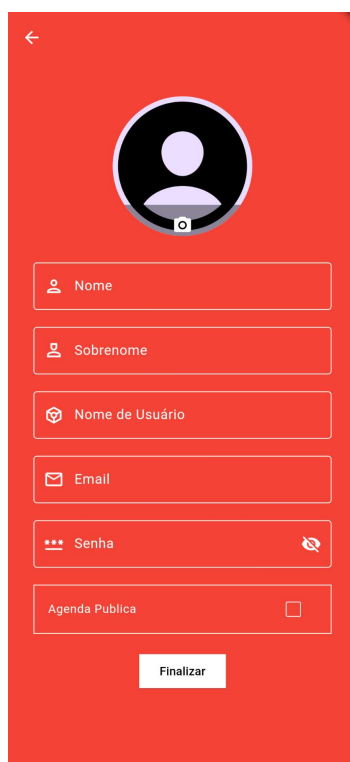


Fonte: Email enviado pelo Fibase Authentication, para conta Gmail.

Figura 13 - Redefinição de SenhaA imagem mostra um formulário web para redefinir a senha. No topo, o título "Redefinir sua senha" está em negrito. Abaixo dele, o texto "para heron11carlos@gmail.com" indica o e-mail associado. Há um campo de entrada rotulado "Nova senha" com uma barra de texto e um ícone de olho para alternar a visibilidade. No canto inferior direito, há um botão azul com o texto "SALVAR" em branco.

Fonte: Link único para troca de senha do usuário, enviado pelo Firebase Authentication.

Ao clicar no link o usuário poderá ir até a página para redefinir sua senha, mostrado na Figura 13. Após a funcionalidade de recuperar a senha, voltando para Figura 11, há o botão de entrar, o qual encaminha para a tela principal do aplicativo; o botão de cadastrar-se, que redireciona o usuário para uma página específica ao qual requer o preenchimento de um formulário com suas informações (Figura 14); e o botão que irá permitir acesso do usuário com as credenciais do provedor de conta Google. Podendo ser cadastrado e acessado rapidamente com um simples clique.

Figura 14 - Tela de CadastroA imagem mostra a tela de cadastro de um aplicativo, com um fundo vermelho sólido. No topo, há uma seta branca para voltar. Abaixo, há um ícone de perfil de usuário. Seguem-se campos de entrada brancos com bordas arredondadas: "Nome", "Sobrenome", "Nome de Usuário", "Email", "Senha" (com ícones de pontos e olho) e "Agenda Publica" (com uma caixa de seleção). No final, há um botão branco com o texto "Finalizar" em preto.

Fonte: Timeshare em execução.

Nesta tela, o usuário pode optar por adicionar uma foto ao seu perfil e, consequentemente, quando clicar no ícone de câmera ele pode tirar a foto no momento ou opcionalmente, escolher a partir da sua galeria. Caso opte por adicionar a foto, a mesma já ficará visível nesta página e prosseguindo para os próximos itens. Nos campos de entrada Nome e Sobrenome, fica claro o que deve ser digitado, e ao clicar em cada campo, o teclado já inicia com a primeira letra capitular seguindo a cada palavra informada, uma vez que, trata-se de nome próprio, a primeira letra deve ser maiúscula.

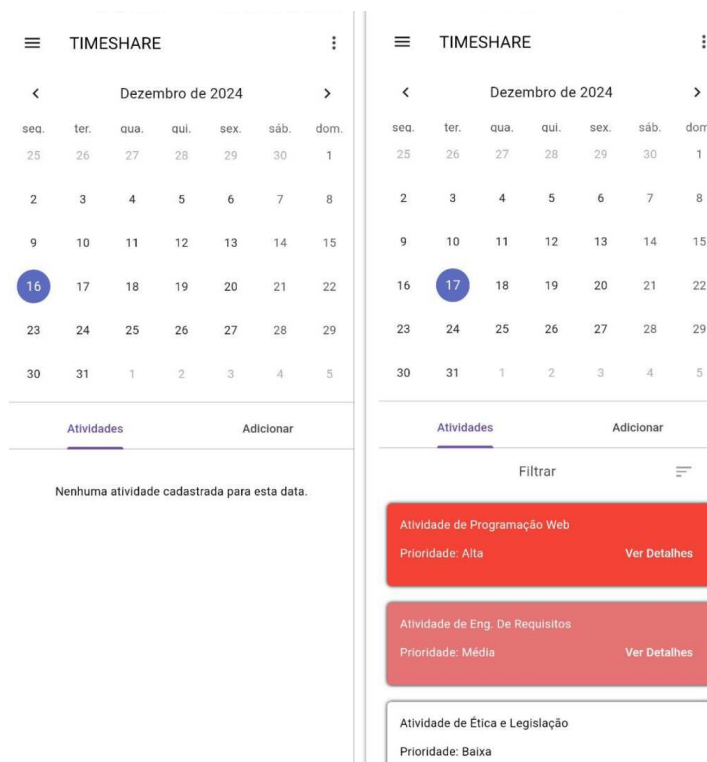
No campo Nome de Usuário, pode informar um apelido que será usado durante a lista de amigos, e outras formas de identificação. O nome de usuário é único, e será informado caso já tenha sido cadastrado.

O campo de email é intuitivo, este será informado uma conta como forma de identificação virtual e de contato para a aplicação. Após isto tem-se o campo de senha o qual foi estabelecido com a norma de segurança de tal modo que, o usuário deverá informar uma senha que contenha no mínimo uma letra maiúscula e minúscula, um número e um caractere especial (por exemplo: !, @, #, \$, %, &, *). Dentro deste campo, ainda há o mesmo ícone da tela de login, para visualizar a senha digitada.

Por último, a caixa de seleção para informar se sua agenda será pública ou privada de tal modo que previna que outros usuários possam visualizá-la. Esta mesma opção pode ser alterada nas configurações da conta do usuário. Ao terminar de preencher o formulário o botão de finalizar estará presente para poder realizar o login diretamente e visualizar a tela principal do aplicativo.

4.2 Tela Principal

Conforme a Figura 15, observa-se a tela principal, a qual será apresentada algumas informações como a visualização do calendário com atividades caso tenham sido submetidas.

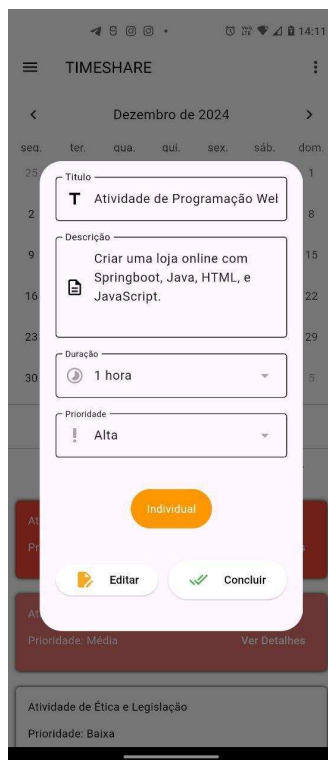
Figura 15 - Tela Principal

Fonte: Timeshare em execução.

Ao iniciar a tela principal o usuário se depara com atividades do dia atual. No caso da Figura 15, em sua primeira instância não houveram atividades cadastradas no dia 16 de dezembro de 2024, portanto, a lista apresenta uma mensagem informando o estado nesta data. Na segunda instância, caso haja atividades cadastradas, haverá um bloco informando o título e sua prioridade, seguido pelo botão de “Ver Detalhes”. Este botão abrirá um popup com as informações da atividade.

Das atividades mostradas na Figura 15 pode-se notar a diferença de prioridades. No caso da Atividade de Programação Web, a prioridade é alta, portanto, tem-se a cor vermelha em tom forte para dar ênfase na necessidade de concluí-la. Do mesmo modo a atividade com prioridade média tem sua cor vermelha, porém com tom mais suave. E por último, a atividade com prioridade baixa que deixa sua cor branca, não destacada como as outras.

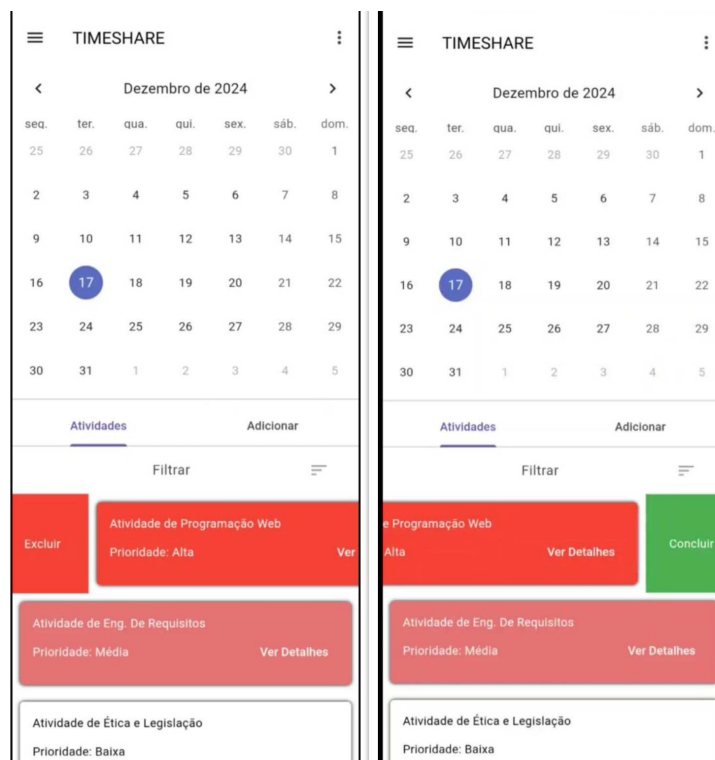
Por definição as atividades são organizadas de acordo com sua prioridade, o que significa as prioridades altas vem primeiro, em segundo as prioridades médias, e por último as prioridades baixas. Isto faz com que o usuário tenha controle do que é urgente para ele e podendo alterar a prioridade ao clicar nos detalhes visto no *popup* da Figura 16.

Figura 16 - Tela de Detalhes/Edição de Atividades

Fonte: Timeshare em execução.

Para manter a tela simples há o título, a descrição, duração e prioridade da atividade seguido pelas tags. Na última linha estão os botões onde pode-se editar ou concluir a atividade. Ao clicar em editar o usuário pode alterar os atributos mencionados, adicionar ou excluir as tags que é feito pelos campos que se tornam ativos para edição, e o ícone de lixeira para tags. Ao terminar a edição clicando no botão “finalizar edição”, o usuário retorna para primeira instância dos detalhes.

O usuário ainda pode concluir a atividade, marcando o cartão na lista de verde, e criando uma distinção das atividades. Ainda é possível completar a atividade como visto na Figura 17.

Figura 17 - Concluir/Excluir Atividades

Fonte: Timeshare em execução.

Para simplicidade de excluir ou concluir as atividades basta deslizar o cartão da atividade e então irá completar a ação. Os cartões completados serão ocultados da lista de atividades, mas sua visibilidade pode ser alterada ao alterar as configurações da conta. Já as atividades excluídas serão removidas da base de dados.

Nas opções marcadas pela elipse vertical no canto superior direito é possível abrir o menu de opções, ocultando esta visibilidade de forma rápida. A configuração seguirá para as páginas de grupos e de amigos, podendo ser alterada a qualquer momento. Na Figura 18 pode ser acompanhado o processo de adicionar novas atividades, marcado pela aba que pode ser arrastada ou clicada para trocar a visualização.

Figura 18 - Adicionar Atividades

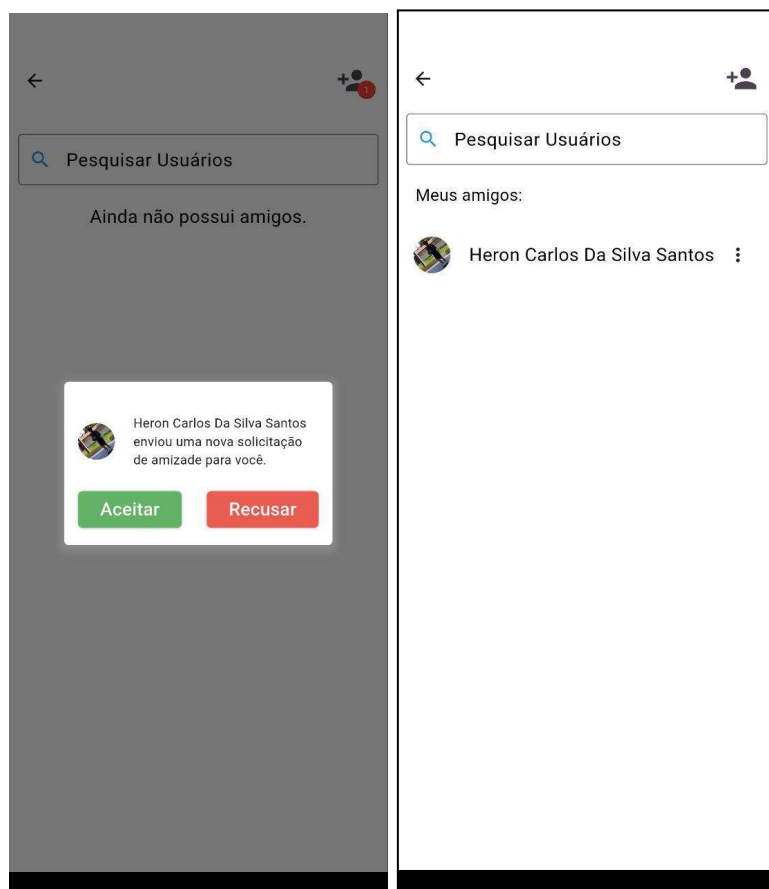
The image displays two side-by-side screenshots of the 'Adicionar Atividades' (Add Activities) form in the Timeshare system. Both screenshots show a calendar for December 2024 with the 16th selected. The left screenshot shows the form with fields for Date (16 de dezembro de 2024), Title (Título), Description (Descrição), and Duration (Duração). The right screenshot shows the form with a Priority (Prioridade) dropdown, a Tags field with an 'Adicionar' button, and a 'Nenhuma tag adicionada.' message. Both screenshots have 'Limpar' (Clear) and 'Salvar' (Save) buttons at the bottom.

Fonte: Timeshare em execução.

Adicionar as atividades requer preencher os atributos de título, descrição (pode ficar em branco), duração, prioridade e tags. As tags são preenchidas e ao clicar em adicionar o usuário pode inserir múltiplas tags que servem como marcadores de filtro de atividades. As atividades quando cadastradas recebem um aviso na tela em sua parte inferior, também acontece o mesmo quando editar, concluir, ou excluir atividades.

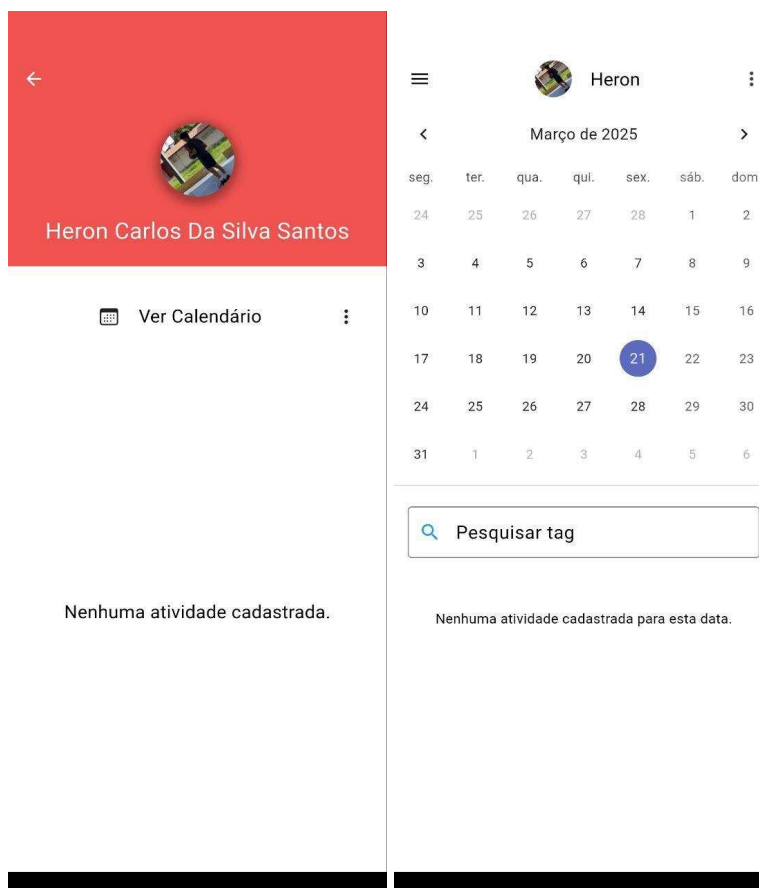
4.3 Tela de Amigos

Esta seção é destinada para as ações de gerenciamento de amigos, como pesquisar, adicionar ou remover usuários. Amigos podem ser pesquisados ou visualizados a partir do menu, onde há um botão com nome “Amigos”, e que direcionam para a tela da Figura 19 a seguir.

Figura 19 - Tela de Amigos

Fonte: Timeshare em execução.

Como é notado na Figura 19, ao entrar na tela será aberto um *popup* mostrando as solicitações de amizade, assim o usuário poderá aceitar ou recusar a solicitação. Clicando na notificação, poderá ser encaminhado para a tela do perfil do usuário, o que mostrará as últimas dez atividades do usuário, ou clicando no botão de “Ver Calendário”, irá para a agenda completa do amigo. A aparência da agenda é a mesma que a do próprio usuário e pode ser vista na Figura 20 a seguir.

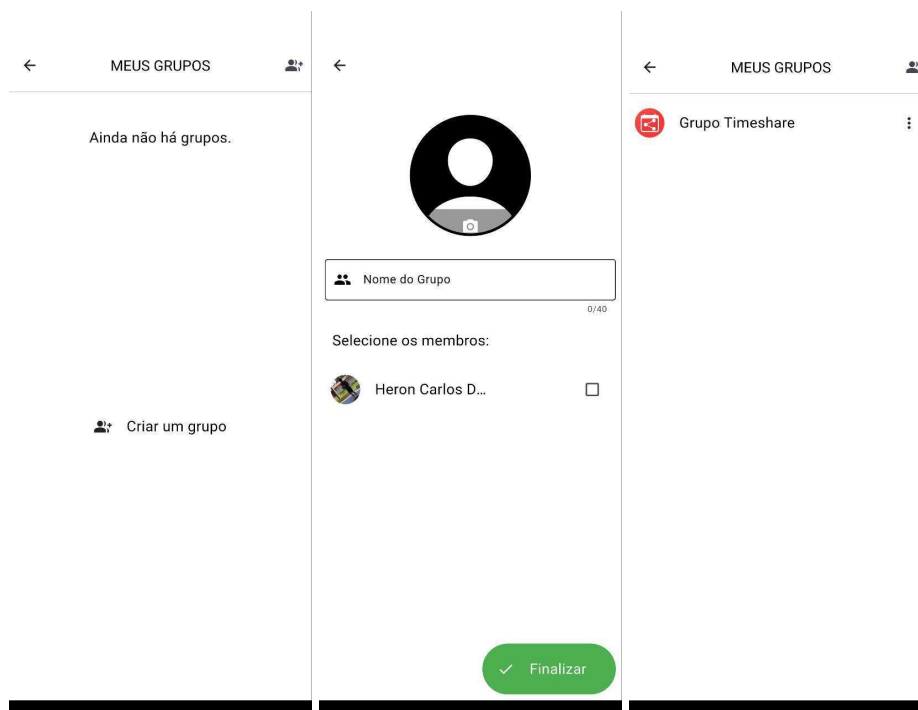
Figura 20 - Tela de Perfil do Usuário

Fonte: Timeshare em execução.

Por ser parecido, o usuário terá familiaridade com as funções, sendo o mesmo para pesquisa por tag ou data, e podendo visualizar as atividades completadas. Porém, como agenda de outro usuário, o visitante não poderá alterá-la.

4.4 Telas de Grupo

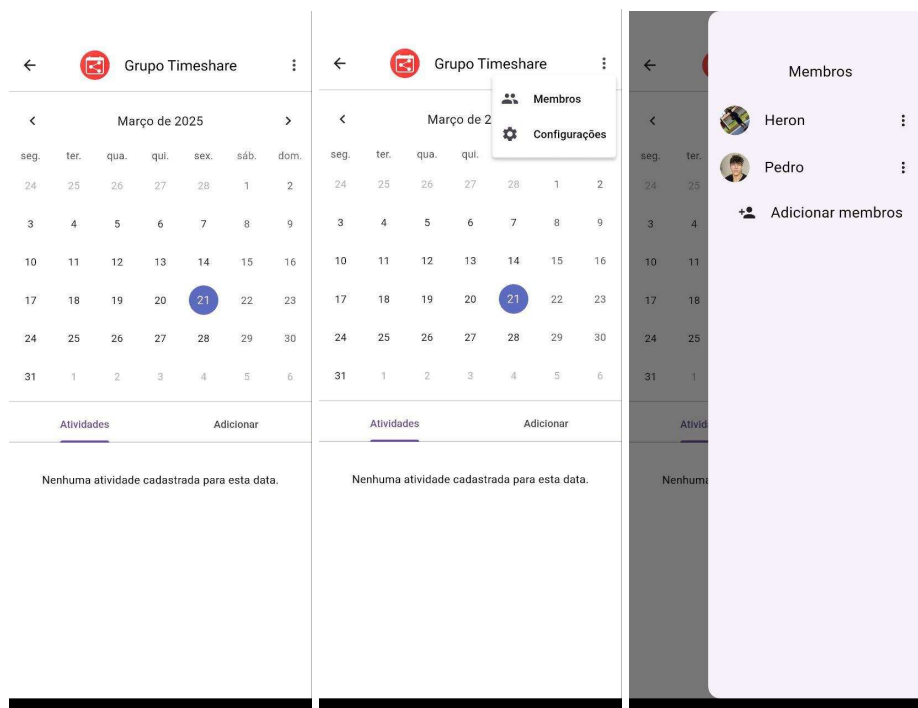
O gerenciamento de grupos ocorre a partir da página de visualização de Grupos, acessada pelo menu lateral da tela principal. Na Figura 21, é demonstrado como o usuário poderá cadastrar um grupo, assim partindo da listagem, até a criação, e a visualização da agenda compartilhada.

Figura 21 - Tela de Gerenciamento de Grupos

Fonte: Timeshare em execução.

No caso da Figura 21, é mostrada a mudança de fases, onde não há grupos cadastrados, prosseguindo para tela de criação em que define os membros, nome, e imagem do grupo. Finalizando o cadastro, a navegação irá levar de volta até a listagem de grupos, assim permitindo acesso do usuário para as telas da próxima figura. Semelhante a tela de amigos, o usuário pode utilizar das opções de cada grupo para sair, e se caso for administrador, poderá excluir o grupo inteiramente. O usuário terá papel de administrador para grupo que ele criar, ou para grupos que outros administradores lhe dar esta permissão.

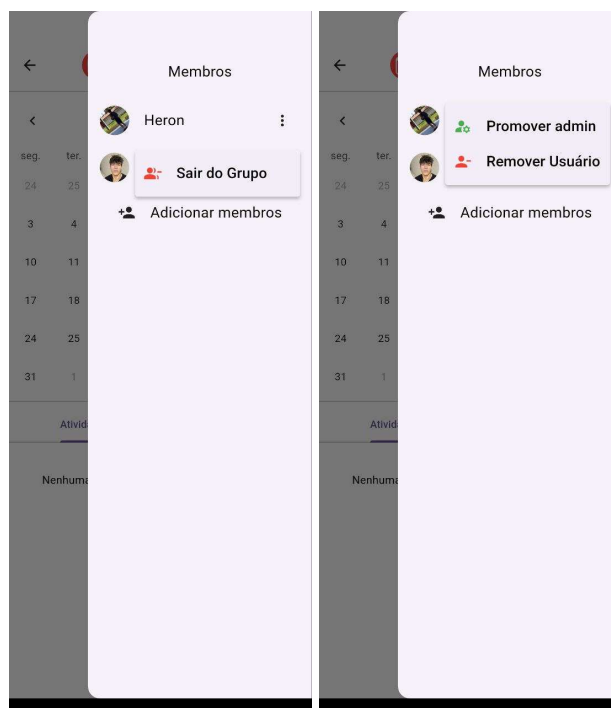
Na figura 22 é notado que a aparência da agenda, semelhante ao do usuário, mas que permite visualizar os membros, e a configuração do grupo. Ainda podendo alterar as permissões de administrador, e remover outros usuários.

Figura 22 - Tela de Agenda de Grupos

Fonte: Timeshare em execução.

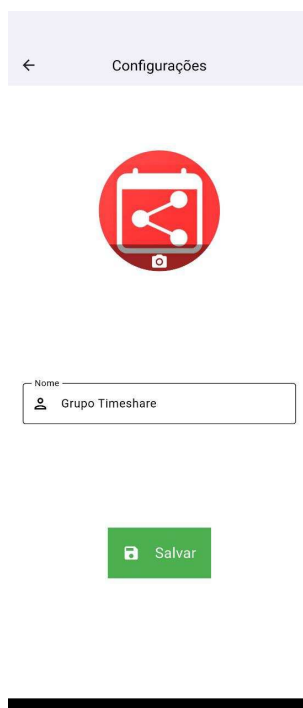
Também diferente da agenda pessoal, na agenda em grupo cada membro pode adicionar ou compartilhar as atividades. Nos cartões ainda é adicionado o criador da atividade, e qualquer usuário poderá manusear umas às outras atividades. Como na agenda pessoal, os usuários podem editar a atividade, completá-la ou excluí-la.

Ainda diferente de uma agenda individual, esta possui outras nuances, ao invés de informar que está na agenda do usuário, a barra superior irá mostrar a imagem do grupo seguido de seu nome. Esse pequeno cabeçalho dará opções de modificar o grupo, clicando na elipse vertical, o usuário pode gerenciar membros ou dar acesso a tela de configurações. Para a administração do usuário, segue a Figura 23 para destacar as opções disponíveis para cada usuário.

Figura 23 - Gerenciamento de Membros de Grupos

Fonte: Timeshare em execução.

Clicando nas opções do membro, pode ser notado que irá abrir uma barra lateral com os membros do grupo. Assim, cada membro possui uma elipse vertical que permite abrir opções adicionais sobre aquele usuário. Em caso de uma tela para criador do grupo, o usuário pode promover ou remover a administrador, que por sua vez compartilha das permissões de adicionar ou remover outros membros. Ainda clicando na instância do membro, o usuário pode seguir para tela de perfil do usuário, da mesma maneira que ele pode enquanto pesquisa um amigo para adicionar. Já na página de configurações, como apresenta a Figura 24, o administrador pode também realizar mudanças nas informações do grupo.

Figura 24 - Tela de Configurações de Grupo

Fonte: Timeshare em execução.

Reservado apenas para administradores, estes podem alterar a imagem ou mudar o nome do grupo. Administradores são primeiramente definidos pelo criador, e então este poderá promover outros usuários. Em casos de só haver um administrador no grupo, e este desejar sair, outro usuário será promovido aleatoriamente, assim garantindo a integridade do grupo. Caso somente um usuário restante esteja no grupo, este grupo será excluído da base de dados.

4.5 Tela de Suporte e Configuração da Conta

Nesta seção será abordado sobre as telas de suporte e de configuração da conta do usuário, permitindo que reportem erros ou sugestões, e que haja mudanças no comportamento da aplicação.

A tela de suporte foi criada visto que, ao longo do ciclo de vida do aplicativo, haverá erros e dúvidas, sobretudo na sua fase inicial de desenvolvimento. Assim foi criada esta tela, Figura 25, para resolver parte deste problema, permitindo comunicação direta entre desenvolvedor e usuário, possibilitando o conserto de erros de forma mais rápida.

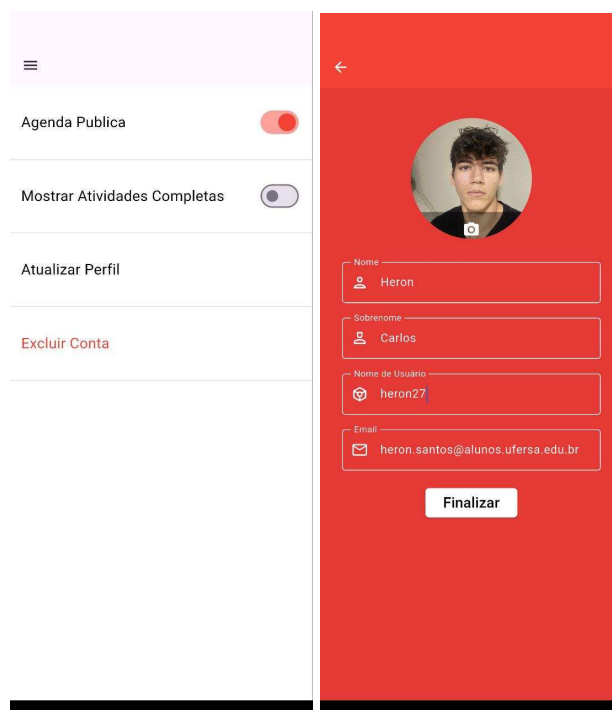
Figura 25 - Tela de Suporte

The mockup shows a mobile application screen for a support page. At the top, there is a light purple header bar with a white back arrow icon on the left. Below the header, the main content area is white and contains two paragraphs of text. The first paragraph explains that this is the support page and that any problem or question should be described here to be sent to the developer for evaluation. The second paragraph provides instructions for reporting system errors, asking the user to describe the steps to reproduce the issue and the undesired behavior. Below the text, there are two input fields: a smaller one labeled 'Assunto' (Subject) and a larger one labeled 'Descrição' (Description). The bottom of the screen features a solid black navigation bar.

Fonte: Timeshare em execução.

Se houver um problema ou erro durante o uso do sistema, a partir da barra lateral de menu, na tela principal, o usuário poderá abrir a opção de suporte. Esta página disponibiliza uma breve descrição de como o erro deve ser reportado, e guia o usuário sobre suas ações. Também podem sugerir mudanças para o aplicativo, e todas as informações enviadas serão encaminhadas para o email do desenvolvedor, a fim de realizar reparos ou melhorias.

Já em relação às configurações da conta, podendo ser acessado a partir da mesma tela que o suporte, o usuário terá as opções de alterar a privacidade da sua agenda pessoal, alternar a visibilidade das atividades concluídas, atualizar seu perfil, ou excluir a conta, visto na Figura 26.

Figura 26 - Tela de Configuração e Atualização de Perfil

Fonte: Timeshare em execução.

As quatro funcionalidades apresentadas na Figura 26 são extremamente importantes para a conta do usuário. Visto que em agendas privadas somente amigos podem visualizar suas atividades, não seria interessante que usuários de pessoas ou entidades públicas tenham que adicionar cada usuário que possa visualizar sua agenda. Já na visibilidade das atividades concluídas, permite ao usuário organizar melhor sua agenda, escolhendo entre manter a contabilidade do que já fez, ou concentrar nas atividades restantes.

A terceira opção, atualização do perfil, é essencial em qualquer descrição social, visto que possam haver mudanças, ou o usuário somente deseja mudar sua foto. E por último, o usuário pode excluir sua conta na quarta opção, garantindo que suas atividades, informações e presença não estejam atreladas ao Timeshare. Ao excluir a conta, todas as informações são apagadas dos sistemas em que o Timeshare depende, isto é, autenticação, base de dados, ou da visibilidade de outros usuários.

4.6 Teste Versão Alpha

Durante o desenvolvimento do aplicativo foram realizados diversos testes com a aplicação, porém é inevitável que alguma forma de interação com usuários reais não seja prevista. Enquanto desenvolvedor, planeja-se criar um aplicativo que execute certas

funcionalidades, e para isto é necessário entender como estas funcionam, e certamente serão desenvolvidas como devem funcionar.

Em outros casos, quando há um cliente que possui outras experiências, ou que ainda não entende a interface, este pode ter dificuldades para realizar certas tarefas. Assim, foram feitos testes preliminares onde os usuários poderiam usar a aplicação como desejarem, mas com o objetivo de concluir cinco tarefas, descritas no Quadro 3 a seguir.

Quadro 3 - Tarefas do Teste de Alfa do Aplicativo Timeshare

Tarefa	Descrição das Ações
Tarefa 1 – Autenticação de Usuário	- Cadastrar sua conta de usuário - Realizar logout - Realizar login novamente - Trocar a senha de usuário
Tarefa 2 – Gerenciamento de Atividades	- Cadastrar uma atividade - Editar uma atividade - Excluir uma atividade - Marcar uma atividade como completa
Tarefa 3 – Gerenciamento de Amigos	- Pesquisar por um amigo - Enviar um pedido de amizade - Excluir um amigo - Visualizar a agenda de um amigo
Tarefa 4 – Gerenciamento de Grupos	- Criar um grupo - Adicionar um ou mais amigos - Remover um amigo do grupo - Alterar nome ou foto do grupo - Adicionar um administrador ao grupo
Tarefa 5 – Configurações da Conta	- Alterar visibilidade da agenda - Alterar visibilidade de atividades concluídas - Excluir sua conta

Fonte: Autoria Própria

As atividades foram divididas em grupos por funcionalidade e tela, entendendo que, por exemplo, as atividades de login, recuperação de senha e logout do sistema fazem parte da autenticação do usuário. O mesmo método foi utilizado para designar as tarefas de gerenciamento de atividades, de amigos, de grupos, e por último, as configurações da conta do usuário.

Após as atividades realizadas, o aplicativo foi distribuído em formato APK, um

arquivo executável que permite instalar aplicações no sistema operacional Android. Esta distribuição ocorreu por email, e outras formas de contato pessoal. Os usuários de teste alfa então realizaram as atividades e encontraram discrepâncias, como *bugs*, e outros comportamentos inesperados. A seguir será apresentado um quadro das queixas e de como a funcionalidade foi melhorada e/ou consertada.

Quadro 4 - Relatório de Erros Identificados na Versão Alfa

Erro identificado	Solução
A funcionalidade para marcar atividade como incompleta retornando a mensagem “Atividade Completa”.	O método para marcar a atividade não possuía verificação para o valor booleano de completude da atividade, assim foi adicionado, e passou a notificar a mensagem correta.
Algumas funcionalidades presentes na agenda do usuário não estavam disponíveis na agenda do grupo.	Foi refatorado o código para utilizar as mesmas telas para usuário, grupo, e amigo. Assim permitindo a distribuição mais eficiente das funcionalidades, e condicionando para cada tipo de permissão.
As atividades não possuem conexão com a mensagem de concluir ou excluir ao arrastar, criando um efeito visual negativo.	Tirar o acolchoamento (<i>padding</i>) da margem da tela até os cartões de atividades.
As tags adicionadas mudam de cor após cada ação do usuário, criando um bug visual.	Mudar o gerenciamento de estado da página, criando mais embrulhadores (<i>wrapper</i>) para classes sem estado (<i>stateless</i>).

Fonte: Autoria própria.

Estas funcionalidades melhoradas e implementadas durante o teste alfa do aplicativo permitiram uma preparação maior para um teste de satisfação concreto. Assim, os testes desta versão não interferem com a opinião do usuário em relação ao sistema.

4.7 Teste de Usabilidade

Após o desenvolvimento do aplicativo Timeshare por parte do desenvolvedor, é necessário validar os resultados obtidos. Isto é, de acordo com o público alvo, levantar opiniões que mostram a satisfação e sugestões do usuário.

Embora alguns autores sugerem que os testes ocorram durante e ao fim do desenvolvimento (Rubin; Chisnell; 2008), o Timeshare tem uma característica especial que é a curva de aprendizado do desenvolvedor. Começando como alguém iniciante em programação, e que ganhou seu conhecimento sobre Flutter totalmente a partir deste projeto, houve partes em que o projeto não estaria pronto para testes. Assim, seguiu até o ponto em que o aplicativo estivesse pronto para testes públicos, e então foi disponibilizado. De outra maneira, os testes em desenvolvimento ocorreram para realizar métricas de desempenho do aplicativo, categorizando a efetividade e eficiência da atividade realizada.

Para a elaboração do roteiro de testes, foi decidido utilizar o Google Forms, que é uma ferramenta bastante utilizada na área de testes e de entrevistas de satisfação, permitindo realizar pesquisas remotas, e obter análise de dados automaticamente. Embora haja uma preocupação em testes remotos devido a falta de supervisão dos usuários, é sugerido pelos autores Rubin e Chisnell (2008), que os testes remotos ainda podem ser bastante eficazes. Ele ainda apresentam vantagens e desvantagens sobre sua realização, tais como:

Quadro 5 - Vantagens e Desvantagens dos Testes Remotos (Rubin & Chisnell, 2008)

Vantagens	Desvantagens
Não há deslocamento, para usuário ou entrevistador.	Difícil de notar questões individuais, como expressão e linguagem corporal.
Usuários podem utilizar o aplicativo no dia-a-dia, assim podendo simular situações reais.	Comunicação pode ser limitada em testes não moderados, em casos de má interpretação ou falta de contexto visual.
O tempo de resposta pode ser menor, já que pode ser realizado simultaneamente.	Problemas técnicos onde o usuário pode não saber instalar ou utilizar o aplicativo, ou até mesmo a falta de internet.
Disponibilidade maior para os usuários, já que em qualquer parte do mundo podem responder o questionário.	Não há clareza nas condições dos dados resultantes.

De outra maneira, os mesmo autores também propõe sugestões para remover essas desvantagens, como chamadas de vídeo e áudio, com espelhamento de tela do dispositivo. Sabendo que o observador apenas deve tomar dados para coleta do teste, e também para ajudar o usuário somente se ele não conseguir realizar a atividade, do mesmo jeito de uma entrevista presencial.

Para o caso do Timeshare, foi disponibilizado apenas o pacote de instalação do aplicativo, neste caso o APK, e ao final da realização das tarefas propostas, foi realizado o questionário. Cada usuário obteve apoio, e em caso de dúvidas, poderia contatar o desenvolvedor, até mesmo pelo próprio aplicativo, como mostrado no item 4.5 Tela de Suporte e Configurações.

O formulário (apresentado no Apêndice A), foi dividido em duas seções, a primeira buscou informações sociais. Estes dados são importantes para gerar alguns conceitos sobre o público alvo do Timeshare, assim, apresenta-se no quadro a seguir como foi pensado em cada aspecto.

Quadro 6 - Perguntas de Perfil Social para o Teste de Usabilidade.

Pergunta	Importância
Grau de escolaridade.	Pessoas com pouca habilidade de leitura podem achar que a interface seja difícil de entender.
Sexo.	Para comparar o grupo de usuários, e se há alguma discrepância em seu uso do aplicativo.
Faixa etária.	Pessoas mais jovens tendem a estar mais conectadas com tecnologia, podendo discernir se a usabilidade é a mesma para pessoas mais velhas.
Ocupação.	Usuários que possuem mais de uma obrigação (trabalho e estudo), podem necessitar mais do gerenciamento de agendas. Assim é importante discernir as

	opiniões de usuários com muitas atribuições.
Nível de experiência com aplicativos.	Se há algum contraste em relação à experiência do usuário, de forma que, por exemplo, se o usuário que possui baixa experiência com aplicativos de dispositivos móveis não consegue realizar certa tarefa, mas um usuário com alta experiência consiga.
Média diária de uso do celular.	Estando inserido em uma rotina digital de, por exemplo, 6 horas por dia, pode-se tornar mais fácil conciliar o uso de um aplicativo para dispositivo móvel.
Utiliza algum aplicativo de gerenciamento de agenda (Google Agenda, Doodle, Calendly).	Caso o usuário já possua algum tipo de ferramenta para gerenciar sua agenda, poderá dar opiniões construtivas sobre como o aplicativo pode ser melhorado, de acordo com experiência prévia, ou se ele desejaria algo mais parecido com o Timeshare. Caso não tenha uso prévio, pode ser notado o que faria este usuário utilizar o Timeshare, de acordo com sua opinião.

Há ainda muitas informações relevantes que podem ser analisadas a partir do perfil social, porém já pode ser notado sua importância durante o teste. Para a segunda seção, o teste de usabilidade foi realizado de acordo com as tarefas disponibilizadas. O usuário respondeu uma série de perguntas sobre seu desempenho e experiência, durante o uso do Timeshare.

Seguindo a premissa que para responder o questionário é necessário realizar o uso do aplicativo Timeshare, os usuários se disponibilizam a executar cinco tarefas. Cada uma das tarefas possuem um papel importante no desenvolvimento desta aplicação, assim demonstrando em execução o que foi proposto para o Timeshare. Desta forma, servem para

que o usuário se familiarize, e assim ele pode ter uma opinião de como o aplicativo se comporta, antes de responder o questionário.

Na segunda parte do questionário, as perguntas foram baseadas nas tarefas realizadas; já que estas tarefas capturam o que é a essência do aplicativo, as perguntas devem avaliar os mesmo aspectos. Dividido para cada tarefa, foi questionado as opiniões do usuário acerca da dificuldade/facilidade, satisfação, e sugestões que este veio a experienciar enquanto utilizava o aplicativo.

4.8 Resultados do Questionário

Para realizar o teste de usabilidade, foi dividido em duas etapas, primeiramente uma reunião online, e em seguida foi respondido o questionário disponibilizado no Google Forms (Apêndice A). Desta forma, a reunião serviu para cronometrar o uso do aplicativo durante a realização das atividades previstas, e também para documentar comportamentos anômalos do usuário e/ou do aplicativo, além de garantir que o usuário utilize o Timeshare antes de fornecer as perguntas.

Foram entrevistados 7 (sete) participantes, as reuniões foram feitas através do WhatsApp, Discord, ou Google Meet, ferramentas escolhidas pelos participantes a fim de atender suas preferências. Cada usuário aceitou transmitir a tela do celular para documentar as ocorrências durante o uso, e com microfone para fornecer comunicação durante as atividades.

Com base nos dados do Quadro 7, observou-se que o tempo médio para conclusão de todas as tarefas foi de 12 minutos e 50 segundos, com variação entre 6 minutos e 23 segundos e 15 minutos e 49 segundos. Os tempos mais longos foram atribuídos a fatores como conexão instável e exploração mais minuciosa da interface por parte dos usuários.

Quadro 7 - Análise de Tempo para Realização da Tarefas

Usuário	Tempo (min:seg)
Usuário 1	13m 34s
Usuário 2	11m 55s
Usuário 3	14m 12s
Usuário 4	15m 49s

Usuário 5	13m 23s
Usuário 6	6m 23s
Usuário 7	12m 25s

Fonte: Autoria Própria

Assim, para analisar as respostas das questões no Google Forms, foram feitas observações em relação ao perfil dos participantes, disponíveis no Apêndice B. O gráfico do sexo dos participantes demonstra que 83,9% eram do sexo masculino; sendo 57,1% com ensino superior incompleto, e o restante (42,9%) com ensino médio completo. A faixa etária predominante situa-se entre 20 e 30 anos (57,1%), seguida por 28,6% com menos de 20 anos, e 14,3% entre 30 e 40 anos. Quanto à ocupação, apenas um participante estava desempregado; os demais trabalhavam (14,3%), estudavam (28,6%), ou conciliavam ambas as atividades (42,9%).

No que se refere à experiência tecnológica, 57,1% relataram uso diário do smartphone entre 2 e 6 horas, enquanto os demais utilizam entre 6 e 10 horas por dia. Metade dos participantes utilizava o Google Agenda como ferramenta de gerenciamento, sendo esta a única citada.

As tarefas foram avaliadas com base em uma escala de Likert de 5 pontos, conforme descrito no Quadro 3. Cada tarefa correspondia a um requisito funcional da aplicação.

- Tarefa 1 (Autenticação): Avaliada positivamente por todos os usuários, com exceção de um participante que manifestou opinião neutra quanto à entrada/saída do sistema.
- Tarefa 2 (Navegação e Interação com Atividades): Obteve avaliações exclusivamente entre “satisfeito” e “completamente satisfeito”.
- Tarefa 3 (Gerenciamento de Amigos): Um usuário demonstrou insatisfação e foi o único que não conseguiu concluir a tarefa, devido a falhas na visualização e exclusão da agenda de um amigo. Os demais participantes relataram sucesso e satisfação na realização dessa tarefa.
- Tarefa 4 (Grupos): Registrou satisfação majoritária, com apenas um relato de neutralidade em relação à organização visual.
- Tarefa 5 (Configurações de Conta): Avaliações igualmente positivas, com predominância de satisfação parcial e completa.

Em relação à satisfação geral com o aplicativo Timeshare, 71,4% dos usuários marcaram “satisfeito”, enquanto 28,6% indicaram “completamente satisfeito”. O único caso de não conclusão de tarefa foi investigado e atribuído a uma interrupção na comunicação com a base de dados, durante o envio de um pedido de amizade.

Por fim, a questão aberta sobre sugestões de melhoria gerou seis respostas, abrangendo aspectos de design, funcionalidades adicionais e preferências individuais dos participantes.

Quadro 8 - Respostas de Melhoria do Timeshare

O que você melhoraria no aplicativo Timeshare?
“A atualização das páginas não serem imediatas, fazendo com que você tenha que atualizar manualmente.”
“O layout, tem que adaptar as dimensões para outros aparelhos”
“Falta colocar o espaço de hora (quando começa e acaba) quando vai cadastrar uma atividade.”
“Um pouco a aparência do aplicativo e adicionaria um mapa guiado para as pessoas que fossem realizar o uso pela primeira vez”
“Apenas a funcionalidade da mudança de foto de perfil para carregar/atualizar na mesma página em que o usuário esteja visualizando (ver a mudança da foto de perfil na mesma página, sem precisar sair dela). De resto o aplicativo está de parabéns, gostei bastante <3”
“Melhorias de desing, interface e mais opções de organizações de tarefas dentro do aplicativo.”

Fonte: Retirado da planilha gerada pelo Google Forms.

Estas respostas servirão como base para algumas melhorias finais do aplicativo antes que este trabalho chegue ao fim. Tendo em vista que o objetivo deste teste de usabilidade é

avaliar a satisfação do usuário e indicar a direção de evolução do Timeshare, assim pode se declarar foi bem sucedido. Embora ainda haja questões que precisam de melhorias além desta monografia, o que foi conquistado já mostra efetividade quando considerado as respostas obtidas.

4.9 Ajustes Após Teste de Usabilidade

Para não deixar o aplicativo com mudanças sugeridas e ainda não implementadas, neste capítulo será tratado como foram feitas essas alterações para atender as respostas do Quadro 8. Estas alterações são feitas na lógica, aparência, e em relação à melhoria de funcionalidades do Timeshare. Como nota-se no exemplo da Figura 26 a seguir, anteriormente a função de navegação levaria o usuário até a página de criação de grupos.

Figura 27 - Atualização das páginas

```

1
2 ...
3
4 Navigator
5   .push(context,
6         MaterialPageRoute(builder: (context) =>
7           GroupCreatePage()));
8
9 ...

1
2 ...
3   await Navigator
4     .push(context, MaterialPageRoute(
5       builder: (context) => AgendaPage(entity: entity,)))
6     .then((value) async {
7       await getGroups();
8     },);
9
10 ...
11
12

```

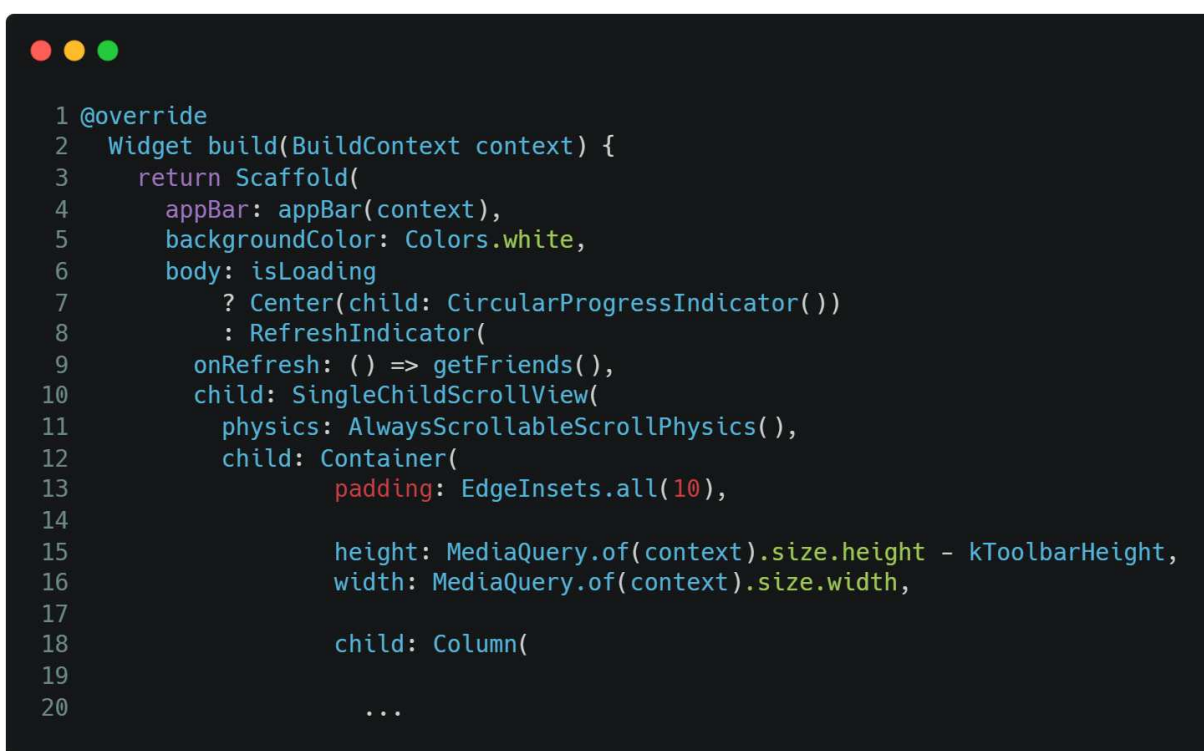
Fonte: Autoria própria.

Com isto, a atualização das páginas foi corrigida ao modificar os métodos de navegação de página, e atualizando as páginas de grupo, amigos, e perfil. Ao sair da tela de criação do grupo, a página irá atualizar automaticamente para mostrar o novo grupo

cadastrado. Ao adicionar o amigo ou removê-lo, irá adicionar visualmente na lista dos amigos do usuário sem recarregar a página, e o perfil irá carregar as novas alterações feitas para o amigo. Isto é possível graças a funções assíncronas, que são executadas, e através da função “then()”, pode ser executado funções após o término, garantindo que as informações sejam recarregadas.

A segunda resposta menciona adaptação do layout para celulares de diferentes tamanhos, isto ocorreu devido a falta de consulta no tamanho das telas ao definir a largura do *Container*, mostrado na figura a seguir.

Figura 28 - Modificações Para Adaptar Largura da Tela



```

1 @override
2 Widget build(BuildContext context) {
3   return Scaffold(
4     appBar: appBar(context),
5     backgroundColor: Colors.white,
6     body: isLoading
7       ? Center(child: CircularProgressIndicator())
8       : RefreshIndicator(
9         onRefresh: () => getFriends(),
10        child: SingleChildScrollView(
11          physics: AlwaysScrollableScrollPhysics(),
12          child: Container(
13            padding: EdgeInsets.all(10),
14
15            height: MediaQuery.of(context).size.height - kToolbarHeight,
16            width: MediaQuery.of(context).size.width,
17
18            child: Column(
19
20            ...

```

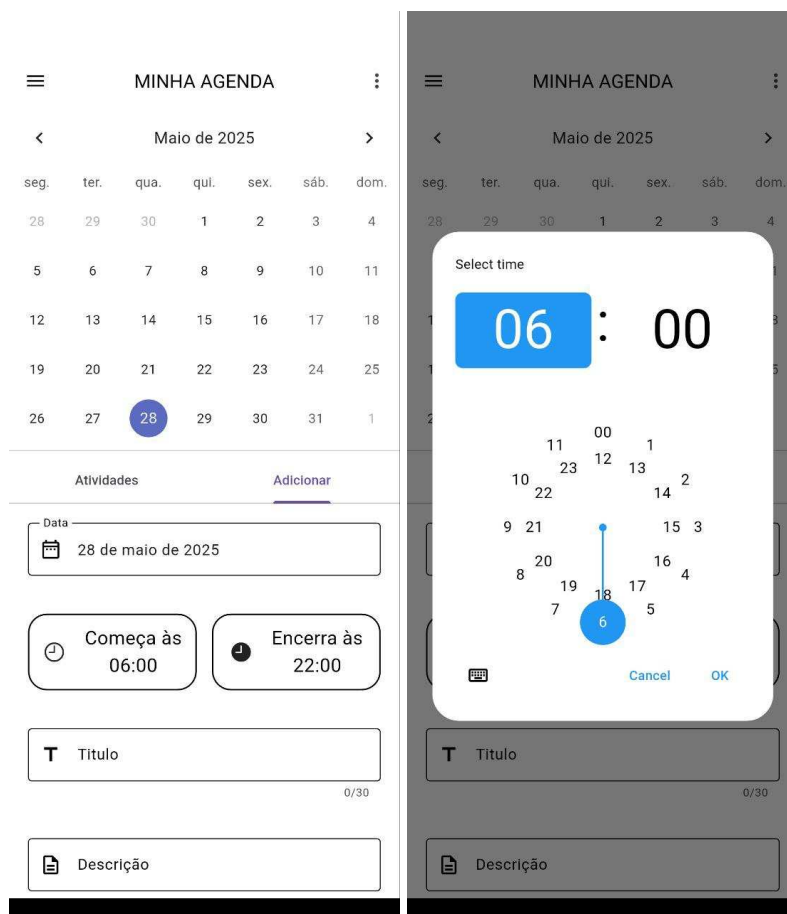
Fonte: Autoria Própria.

Como pode ser observado, os atributos da linha 15 e 16 buscam as medidas de altura e largura, respectivamente, de acordo com as especificações do sistema. Durante a reunião com o usuário em questão que reportou esta melhoria, notou-se que as telas estavam ocultando as bordas da tela, assim indicando que a largura estava maior que a tela do smartphone do usuário. A modificação mostrada na Figura 27 foi aplicada a todas as telas do aplicativo, assim garantindo que a tela esteja conforme as especificações individuais.

Seguindo a terceira sugestão, foi inserido um horário de começo e fim das atividades, assim indicando melhor do que marcação anterior, onde o usuário selecionava uma quantidade específica de horas em que a atividade levaria. Como visto na Figura 28 a seguir,

dois botões indicando começo e fim da atividade, que abre um *popup* para escolher a hora específica de cada atributo.

Figura 29 - Adição do Começo e Fim da Atividade



Fonte: Timeshare em execução.

Para melhoria que atende a quarta resposta do Quadro 8, foi inserido um tutorial das páginas iniciais, permitindo uma familiarização imediata da página da agenda, e dos menus das atividades. Importante ressaltar que as instruções utilizadas para agenda pessoal também servem para os grupos. O tutorial aparece para contas que estão logando pela primeira vez, mostrando um vídeo e uma breve descrição de como funciona o que é mostrado. Consiste de três telas, alternadas pelo botão de “próximo”, e pode ser pulado, caso o usuário deseje.

Figura 30 - Tutorial

Fonte: Timeshare em execução.

Por último, foi considerado sexta resposta do Quadro 8, porém, mudanças de layout seria uma mudança drástica e inconsistente com o restante das opiniões de satisfação do questionário. Somente melhorias que condizem com as funcionalidades e mais possibilidades poderão ser consideradas no futuro do aplicativo Timeshare. Assim chegando ao final desta etapa de desenvolvimento, tendo concluído as atividades iniciais propostas para um aplicativo de gerenciamento de agenda. A seguir seguem as considerações finais deste trabalho.

5 CONSIDERAÇÕES FINAIS

Neste Trabalho de Conclusão de Curso, utilizou-se o Flutter Framework com aplicabilidade de uma agenda para o gerenciamento de tempo, podendo estar disponível para qualquer usuário que tenha interesse em: auxiliar as suas atividades diárias, como eventos em grupo, criando uma rede social para divulgar sua disponibilidade temporal. Assim, não há ambiguidade de tempo, onde a participação de múltiplos usuários requer quantidade maior de comentários sobre quem está disponível, em qual horário e em qual dia, deixando tudo para uma representação visual.

Entre as atividades disponibilizadas no aplicativo estão: criar múltiplas atividades, gerenciar seu status de completude, editá-las se necessário, pesquisar por data e por etiquetas, e ordená-las por prioridade. Além disso, pode-se adicionar amigos, removê-los, e visualizar suas atividades diariamente. Também há a criação de grupos, adicionando qualquer usuário que já pertença aos amigos, podendo gerir seu papel ao grupo, assim promovendo o administrador. As atividades cadastradas ainda obedecem a um nível de privacidade, cada usuário podendo optar por fazer suas agendas públicas, ou privadas para qualquer outro indivíduo que pesquise seu perfil. Mudanças devem ocorrer para garantir a funcionalidade da agenda conforme a necessidade e/ou interesse do usuário, uma vez que, tanto as as tecnologias quanto às atividades laborais passam por frequentes adaptações.

Algumas dificuldades surgiram ao longo do processo de desenvolvimento e devem ser destacadas. A primeira foi a obtenção de dados confiáveis, completos e atualizados em língua portuguesa, sobre o Flutter Framework, pois este é ainda pouco utilizado no Brasil, tornando seu acesso e aprendizado menos intuitivos para os desenvolvedores que não possuem proficiência na língua inglesa. Somado a isso, as informações técnicas e práticas devem ser interpretadas a partir da documentação de cada tecnologia utilizada, e que muitas vezes ainda não estão disponíveis em outros meios.

Neste sentido, cada aplicação possui um contexto, cabendo ao desenvolvedor realizar soluções adaptadas ou alternativas ao seu problema. Ao longo do desenvolvimento, funcionalidades inicialmente previstas foram modificadas em função do aprofundamento do conhecimento sobre o projeto.

Os testes de usabilidade reafirmaram a direção de desenvolvimento do aplicativo, podendo então iterar com outras funcionalidades adicionais para deixar o aplicativo mais completo. A satisfação do usuário é o objetivo de qualquer aplicação de software, visto que

estas servem de ferramentas para o indivíduo realizar uma determinada atividade, assim sendo corroborado a motivação deste trabalho.

6 TRABALHOS FUTUROS

Como sugestão para pesquisas futuras, recomenda-se que os pesquisadores façam uma análise comparativa entre os aplicativos de gerenciamento de agendas disponibilizadas nas lojas de aplicativos, visto que outras tecnologias já poderão ser desenvolvidas e melhoradas. O Timeshare como projeto estará disponível no github do autor, acessado pelo link <https://github.com/HeronCSS/timeshare>, a fim de obter contribuições e ou sugestões para a aplicação.

Com os resultados apresentados anteriormente, é possível notar que ainda há melhorias a serem feitas, mas que deixa claro um caminho melhor para as próximas versões do aplicativo Timeshare. O intuito desta monografia é de criar um aplicativo que sirva para auxiliar no gerenciamento de tempo, tendo isto concluído, cabe a um trabalho futuro realizar outros testes e com respostas após um uso prolongado dos usuários. Considerando que o projeto teve origem em uma proposta teórica, a evolução completa de sua implementação dependerá de estudos empíricos subsequentes.

Além disso, sabendo que houve restrições financeiras que justificam a falta de desenvolvimento para dispositivos móveis iOS, ou restrições de tempo para reprodução Windows, Web, e Linux, é necessário adaptar o código futuramente para abranger mais plataformas. Esta sendo uma das principais forças do Flutter, não será um grande problema, e possibilitará um maior aprofundamento neste *framework*. Dentre as mudanças previstas, podem ser notadas os seguintes pontos:

Quadro 9 - Mudanças Futuras Previstas Para o Timeshare.

Tópico	Descrição
Sistema de Notificações	Enviar notificações em alterações de atividades, solicitações de amizades, novos grupos, etc.
Chat de Mensagens	Integrar os grupos com chat, para que as

	discussões sobre atividades possam ocorrer de forma mais rápida.
Indexação de Links	Em casos de atividades online, como reuniões, arquivos, etc. poder realizar cliques simples para redirecionar até os links.
Mudança de Visualização do Calendário	Mais formas de visualizar a atividade podem abrir outras maneiras de enxergar a agenda. Seja por dia, por mês, ou por hora.
Integração com IA	Ao combinar com Inteligência Artificial, se torna possível criar atividades de forma mais rápida e a partir de outras fontes, como pelo Whatsapp, e outras aplicações, já que muitas ferramentas possuem esta integração.
Multiplataforma	Como mencionado, tornar a aplicação multiplataforma, seja para Windows, Linux ou Web.
Publicação na Play Store	Tornar público e rentável a utilização do aplicativo.

Fonte: Autoria própria.

O sistema de notificações requer um processamento em nuvem para envio de dados que disparam o alerta de notificações no celular dos usuários, assim necessita do estudo de uma documentação inteira para integração completa. Sendo recomendado o Cloud Messaging para compatibilidade com as outras ferramentas do Firebase. Este mesmo serviço permite o serviço de chat de mensagens, assim criando duas oportunidades em um único trabalho para adição das funcionalidades.

Do mesmo modo que as tags permitem a melhor organização das atividades, um sistema de indexação de links, seja de reuniões e/ou documentos, permite que os usuários sincronizem suas atividades com outros mecanismos de tarefas. Para abrir mais possibilidades de visualização e criatividade na realização das atividades, disponibilizar mais visualizações

do calendário, principalmente considerando reprodução em outras plataformas, é instrumental para servir as necessidades dos usuários.

Em um mundo cada vez mais expansivo para as tecnologias de Inteligência Artificial, a integração da agenda com esta ferramenta pode ser de incrivelmente magnificente. Não só chatbots, mas também em conversação por voz, é possível utilizar IA para agendar as atividades da agenda ainda mais rápido, incorporando outra ferramenta do dia a dia ao Timeshare.

E por fim, sendo finalizado a concepção do Timeshare, é necessário escolher uma plataforma de redistribuição para disponibilizar o aplicativo para os usuários. Assim, a Play Store sendo a biblioteca oficial de aplicativos Android, faz bastante interessante sua distribuição.

7 REFERÊNCIAS

AEON, B.; FABER, A.; PANACCIO, A. **Does time management work? A meta-analysis.** *PLOS ONE*, v. 16, n. 1, e0245066, 2021. Disponível em: <https://doi.org/10.1371/journal.pone.0245066>.

American National Standards Institute; Institute of Electrical and Electronics Engineers. **IEEE Standard for Binary Floating-Point Arithmetic.** ANSI/IEEE, New York, 1985.

BANKS, Alex; PORCELLO, Eve. **Learning React: Modern Patterns for Developing React Apps.** 3. ed. Sebastopol: O'Reilly Media, 2023.

Barros, Anthony de. **Practical SQL: A Beginner's Guide to Storytelling with Data.** 1. ed. San Francisco: No Starch Press, 2018.

CEDRO TECHNOLOGIES. **Framework Flutter: 8 benefícios para utilizá-lo nos aplicativos.** 2024. Disponível em: <https://www.cedrotech.com/blog/framework-flutter>. Acesso em: 28 maio 2025.

Cendron, Marcelo. **Instituto Federal Catarinense.** s.d. Disponível em: <https://professor.luzerna.ifc.edu.br/marcelo-cendron/wp-content/uploads/sites/40/2018/05/Teoria-04-Arrays.pdf>. Acesso em: 18/11/2024.

DABIT, Nader. **React Native in Action.** Shelter Island: Manning Publications, 2019.

Dart. **Dart Overview.** 2024. Disponível em: <https://dart.dev/overview>. Acesso em: 18/11/2024.

DOODLE. **Products.** Disponível em: https://doodle.com/pt_BR/product-overview/. Acesso em: 7 maio 2025.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados.** 6. ed. São Paulo: Pearson, 2011.

Firebase. **Cloud Firestore.** 2024. Disponível em: <https://firebase.google.com/docs/firestore>. Acesso em: 18/11/2024.

Firebase. **Console Firebase**. 2025. Disponível em: <https://console.firebase.google.com>. Acesso em: 18/02/2025.

GIL, Antonio Carlos. **Métodos e Técnicas de Pesquisa Social**. 6. ed. São Paulo: Atlas, 2019.

HAVERBEKE, Marijn. **Eloquent JavaScript: A Modern Introduction to Programming**. 4. ed. No Starch Press, 2024.

Hexsel, Roberto A. **Universidade do Paraná**. s.d. Disponível em: https://www.inf.ufpr.br/roberto/ci067/04_vet.html. Acesso em: 18/11/2024.

LEFFA, Vilson J. **Não tem mais sesta: gestão do tempo em cursos a distância**. *Letras & Letras*, v. 25, 2009. Disponível em: https://leffa.pro.br/textos/trabalhos/Sem_sesta.pdf. Acesso em: 18/11/2024.

Microsoft. **Microsoft SQL Documentation**. Disponível em: <https://learn.microsoft.com/en-us/sql/>. Acesso em: 18/11/2024.

MORVILLE, P. *Ambient Findability: What We Find Changes Who We Become*. 1. ed. Sebastopol: O'Reilly Media, 2005.

Nyonggodwill. **A Bit About Dart**. *Medium*, 28 fev. 2022. Disponível em: <https://nyonggodwill11.medium.com/a-bit-about-dart-28b612168d0b>. Acesso em: 18/11/2024.

Oliveira, C. T.; Carlotto, R. C.; Teixeira, M. A. P.; Dias, A. C. G. **Oficinas de gestão do tempo com estudantes universitários**. *Psicologia: Ciência e Profissão*, v. 36, n. 1, p. 224–233, 2016. doi:10.1590/1982-3703001482014.

Pancini, Laura. **Exame: Google Meet comemora 1 ano no Brasil; uso no país aumentou 275% só em 2021**. 29 abr. 2021. Disponível em: <https://exame.com/tecnologia/google-meet-comemora-1-ano-no-brasil-uso-no-pais-aumentou-275-so-em-2021/>. Acesso em: 20/02/2025.

Priberam. **Agenda**. 2024. Disponível em: <https://dicionario.priberam.org/agenda>. Acesso em: 18/11/2024.

REVISTA TECH. **Organização e eficiência: aplicativos para gerenciar tarefas e agenda.**

Disponível

em:

<https://revista.tec.br/portaldoandroid/organizacao-e-eficiencia-aplicativos-para-gerenciar-tarefas-e-agenda/>. Acesso em: 7 maio 2025.

Rogelberg, S. G. et al. **Lateness to meetings: Examination of an unexplored temporal phenomenon.** European Journal of Work & Organizational Psychology, v. 23, p. 323–341, 2014. doi:10.1080/1359432X.2012.745988.

Soares, A. B. et al. **Revista Brasileira de Orientação Profissional**, jul.-dez. 2022, v. 23, n. 2, p. 151–161. doi:10.26707/1984-7270/2022v23n0204.

Statista. **Most used web frameworks among developers 2024** | Statista. Disponível em: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>. Acesso em: 08/03/2025.

W3 Schools. **SQL Tutorial.** Disponível em: <https://www.w3schools.com/sql/>. Acesso em: 18/11/2024.

W3 Tech. **Usage statistics of JavaScript as client-side programming language on websites.** Disponível em: <https://w3techs.com/technologies/details/cp-javascript>. Acesso em: 18/11/2024.

YOUNG, J. R. **The 24-hour professor. Chronicle of Higher Education**, p. A31–A33, maio 2002. Disponível em: <https://www.chronicle.com/article/the-24-hour-professor/>.

APÊNDICE A - Formulário de Teste de Usabilidade

Teste de Usabilidade - Timeshare

Neste formulário irão ser requisitados informações de usuário acerca do entrevistado. Estes dados servirão para identificar os perfis dos avaliadores em relação ao público alvo do aplicativo Timeshare. Nenhuma das informações pessoais serão identificadas, servindo apenas para dar contexto ao Trabalho de Conclusão de Curso do aluno Heron Carlos da Silva Santos, no curso de Bacharelado de Sistemas de Informação, Ufersa.

Após isto, também serão feitas perguntas avaliativas em relação ao desempenho e satisfação em realizar atividades pré-estabelecidas, que serão informadas na próxima seção.

Caso não tenha realizado as tarefas, por favor REALIZE ANTES DE RESPONDER este questionário.

Para duvidas, entrar em contato com heron.santos@alunos.ufersa.edu.br .

* Indica uma pergunta obrigatória

1. Por favor informe seu email. *

2. Informe seu grau de escolaridade *

Marcar apenas uma oval.

- ☐ Sem escolaridade.
☐ Ensino Fundamental.
☐ Ensino Médio Incompleto.
☐ Ensino Médio.
☐ Ensino Superior Incompleto.
☐ Ensino Superior.

3. Informe seu sexo. *

Marcar apenas uma oval.

- ☐ Masculino.
☐ Feminino.
☐ Prefiro não dizer.

4. Informe sua faixa etária. *

Marcar apenas uma oval.

- ☐ 10-20 anos
- ☐ 20-30 anos
- ☐ 30-40 anos
- ☐ 40-50 anos
- ☐ Mais de 50 anos.

5. Informe sua ocupação. *

Marcar apenas uma oval.

- ☐ Aposentado.
- ☐ Desempregado.
- ☐ Estudante.
- ☐ Trabalho.
- ☐ Trabalho e Estudo.
- ☐ Não trabalha, porém possui muitas obrigações.

6. Informe seu nível de experiência com aplicativos de celular em geral. *

Marcar apenas uma oval.

- ☐ 1 - Experiência nenhuma.
- ☐ 2 - Pouca experiência.
- ☐ 3 - Experiência razoável.
- ☐ 4 - Muita experiência.
- ☐ 5 - Bastante experiência.

7. Informe sua média de tempo no celular, diariamente. *

Marcar apenas uma oval.

- ☐ Menos de 2 horas por dia.
- ☐ Entre 2 e 6 horas por dia.
- ☐ Entre 6 e 10 horas por dia.
- ☐ Mais de 10 horas por dia.

8. Você utiliza algum aplicativo ou ferramenta para agendar/organizar seu tempo de forma específica? (Google Agenda, Calendly, Doodle, etc). *

Marcar apenas uma oval.

- ☐ Não utilizo ferramentas de gerenciamento de tempo.
- ☐ Google Agenda.
- ☐ Calendly.
- ☐ Doodle.
- ☐ Outro.

Nesta seção, será avaliada as tarefas concluídas durante o uso do aplicativo. Por favor LEIA ATENTAMENTE as questões, pois estão embaralhadas para evitar desatenção.

TAREFA 1 - AUTENTICAÇÃO DE USUÁRIO

- Cadastrar sua conta de usuário.
- Realizar operação de logout na sua conta cadastrada.
- Trocar a senha de usuário.
- Realizar uma nova operação de login na conta cadastrada.

TAREFA 2 - GERENCIAMENTO DE ATIVIDADES

- Cadastrar uma atividade na sua agenda.
- Editar uma atividade na sua agenda.
- Marcar atividade como completa na sua agenda.
- Excluir uma atividade na sua agenda.

TAREFA 3 - GERENCIAMENTO DE AMIGOS

- Pesquise por um amigo. (heron11carlos, se não houver alguém conhecido)
- Envie um pedido de amizade.
- Visualize a agenda de um amigo.
- Exclua um amigo.

TAREFA 4 - GERENCIAMENTO DE GRUPOS

- Crie um grupo.
- Adicione um ou mais amigos.
- Adicione um administrador ao grupo.
- Remova um amigo do grupo.
- Troque as informações do grupo (nome ou foto).

TAREFA 5 - CONFIGURAÇÕES DA CONTA

- Altere a visibilidade da sua agenda.
- Altere a visibilidade de atividades concluídas.
- Exclua sua conta.

9. AVALIAÇÃO - TAREFA 1 - Qual o seu grau de satisfação ao realizar o Cadastro de usuário? *

TAREFA 1 - AUTENTICAÇÃO DE USUÁRIO

- Cadastrar sua conta de usuário.
- Realizar operação de logout na sua conta cadastrada.
- Trocar a senha de usuário.
- Realizar uma nova operação de login na conta cadastrada.

Marcar apenas uma oval.

- ☐ 1 - Completamente Insatisfeito.
- ☐ 2 - Insatisfeito.
- ☐ 3 - Neutro.
- ☐ 4 - Satisfeito.
- ☐ 5 - Completamente Satisfeito.

10. AVALIAÇÃO - TAREFA 1 - Qual o seu grau de satisfação em relação as instruções de cadastro? *

TAREFA 1 - AUTENTICAÇÃO DE USUÁRIO

- Cadastrar sua conta de usuário.
- Realizar operação de logout na sua conta cadastrada.
- Trocar a senha de usuário.
- Realizar uma nova operação de login na conta cadastrada.

Marcar apenas uma oval.

- ☐ 5 - Completamente Satisfeito.
- ☐ 4 - Satisfeito.
- ☐ 3 - Neutro.
- ☐ 2 - Insatisfeito.
- ☐ 1 - Completamente Insatisfeito.

11. AVALIAÇÃO - TAREFA 1 - Qual o seu grau de satisfação em relação a entrada e saída do sistema (Login e Logout)? *

TAREFA 1 - AUTENTICAÇÃO DE USUÁRIO

- Cadastrar sua conta de usuário.
- Realizar operação de logout na sua conta cadastrada.
- Trocar a senha de usuário.
- Realizar uma nova operação de login na conta cadastrada.

Marcar apenas uma oval.

- ☐ 1 - Completamente Insatisfeito.
- ☐ 2 - Insatisfeito.
- ☐ 3 - Neutro.
- ☐ 4 - Satisfeito.
- ☐ 5 - Completamente Satisfeito.

12. AVALIAÇÃO - TAREFA 2 - Qual o seu grau de satisfação ao preencher o cadastro das atividades? *

TAREFA 2 - GERENCIAMENTO DE ATIVIDADES

- Cadastrar uma atividade na sua agenda.
- Editar uma atividade na sua agenda.
- Marcar atividade como completa na sua agenda.
- Excluir uma atividade na sua agenda.

Marcar apenas uma oval.

- ☐ 5 - Completamente Satisfeito.
- ☐ 4 - Satisfeito.
- ☐ 3 - Neutro.
- ☐ 2 - Insatisfeito.
- ☐ 1 - Completamente Insatisfeito.

13. AVALIAÇÃO - TAREFA 2 - Qual o seu grau de satisfação ao encontrar as funcionalidades das atividades (Cadastrar, Editar, Completar, Excluir)? *

TAREFA 2 - GERENCIAMENTO DE ATIVIDADES

- Cadastrar uma atividade na sua agenda.
- Editar uma atividade na sua agenda.
- Marcar atividade como completa na sua agenda.
- Excluir uma atividade na sua agenda.

Marcar apenas uma oval.

- ☐ 1 - Completamente Insatisfeito.
- ☐ 2 - Insatisfeito.
- ☐ 3 - Neutro.
- ☐ 4 - Satisfeito.
- ☐ 5 - Completamente Satisfeito.

14. AVALIAÇÃO - TAREFA 2 - Qual o seu grau de dificuldade em realizar o gerenciamento de atividades (Cadastrar, Editar, Completar, Excluir)? *

TAREFA 2 - GERENCIAMENTO DE ATIVIDADES

- Cadastrar uma atividade na sua agenda.
- Editar uma atividade na sua agenda.
- Marcar atividade como completa na sua agenda.
- Excluir uma atividade na sua agenda.

Marcar apenas uma oval.

- ☐ 1 - Muito Difícil.
- ☐ 2 - Difícil.
- ☐ 3 - Razoável.
- ☐ 4 - Fácil.
- ☐ 5 - Muito Fácil.

15. AVALIAÇÃO - TAREFA 2 - Qual o seu grau de satisfação em relação a aparência da agenda? *

TAREFA 2 - GERENCIAMENTO DE ATIVIDADES

- Cadastrar uma atividade na sua agenda.
- Editar uma atividade na sua agenda.
- Marcar atividade como completa na sua agenda.
- Excluir uma atividade na sua agenda.

Marcar apenas uma oval.

- ☐ 5 - Completamente Satisfeito.
- ☐ 4 - Satisfeito.
- ☐ 3 - Neutro.
- ☐ 2 - Insatisfeito.
- ☐ 1 - Completamente Insatisfeito.

16. AVALIAÇÃO - TAREFA 3 - Qual o seu grau de satisfação em relação a facilidade de adicionar/remover os seus amigos? *

TAREFA 3 - GERENCIAMENTO DE AMIGOS

- Pesquise por um amigo. (heron11carlos, se não houver alguém conhecido)
- Envie um pedido de amizade.
- Visualize a agenda de um amigo.
- Exclua um amigo.

Marcar apenas uma oval.

- ☐ 1 - Completamente Insatisfeito.
- ☐ 2 - Insatisfeito.
- ☐ 3 - Neutro.
- ☐ 4 - Satisfeito.
- ☐ 5 - Completamente Satisfeito.

17. AVALIAÇÃO - TAREFA 4 - Qual o seu grau de satisfação em relação ao criar e gerenciar um grupo? *

Ex: Adicionar usuarios, Mudar a foto, Adicionar Administrador.

TAREFA 4 - GERENCIAMENTO DE GRUPOS

- Crie um grupo.
- Adicione um ou mais amigos.
- Remova um amigo do grupo.
- Troque as informações do grupo (nome ou foto).
- Adicione um administrador ao grupo.

Marcar apenas uma oval.

- ☐ 5 - Completamente Satisfeito.
- ☐ 4 - Satisfeito.
- ☐ 3 - Neutro.
- ☐ 2 - Insatisfeito.
- ☐ 1 - Completamente Insatisfeito.

18. AVALIAÇÃO - TAREFA 4 - Qual o seu grau de satisfação em relação a aparência, e execução das funcionalidades das paginas de gerenciamento de grupo? *

TAREFA 4 - GERENCIAMENTO DE GRUPOS

- Crie um grupo.
- Adicione um ou mais amigos.
- Adicione um administrador ao grupo.
- Remova um amigo do grupo.
- Troque as informações do grupo (nome ou foto).

Marcar apenas uma oval.

- ☐ 1 - Completamente Insatisfeito.
- ☐ 2 - Insatisfeito.
- ☐ 3 - Neutro.
- ☐ 4 - Satisfeito.
- ☐ 5 - Completamente Satisfeito.

19. AVALIAÇÃO - TAREFA 5 - Qual o seu grau de satisfação em relação a clareza das funcionalidades da tela de configuração? *

TAREFA 5 - CONFIGURAÇÕES DA CONTA

- Altere a visibilidade da sua agenda.
- Altere a visibilidade de atividades concluídas.
- Exclua sua conta.

Marcar apenas uma oval.

- ☐ 1 - Completamente Insatisfeito.
- ☐ 2 - Insatisfeito.
- ☐ 3 - Neutro.
- ☐ 4 - Satisfeito.
- ☐ 5 - Completamente Satisfeito.

20. No geral, qual o seu grau de satisfação em relação ao aplicativo Timeshare? *

Marcar apenas uma oval.

- ☐ 1 - Completamente Insatisfeito.
- ☐ 2 - Insatisfeito.
- ☐ 3 - Neutro.
- ☐ 4 - Satisfeito.
- ☐ 5 - Completamente satisfeito.

21. AVALIAÇÃO - Você conseguiu realizar todas as atividades? *

Marcar apenas uma oval.

- ☐ 1 - Não.
- ☐ 5 - Sim.

22. AVALIAÇÃO - Caso não tenha conseguido completar, ou encontrou alguma dificuldade nas tarefas, por favor informe o numero correspondente e por que não foi possível.

TAREFA 1 - Autenticação de usuários

TAREFA 2 - Gerenciamento de atividades

TAREFA 3 - Gerenciamento de Amigos

TAREFA 4 - Gerenciamento de Grupos

TAREFA 5 - Configurações da Conta

Ex. Formato da resposta: "TAREFA 1 - Não consegui realizar login porque o teclado não abriu."

23. O que você melhoraria no aplicativo Timeshare?

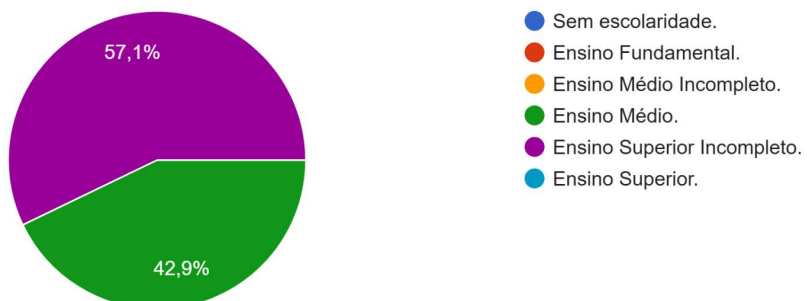
Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

APÊNDICE B - Gráfico de Respostas do Questionário de Usabilidade

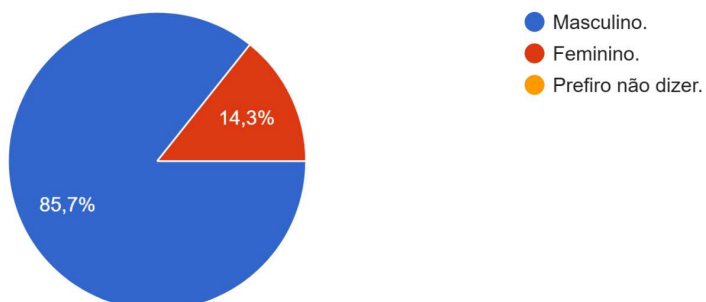
Informe seu grau de escolaridade

7 respostas



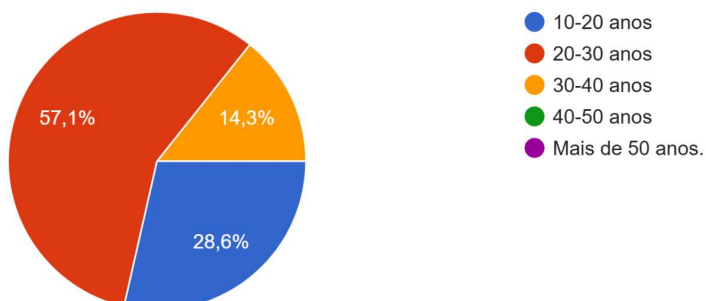
Informe seu sexo.

7 respostas



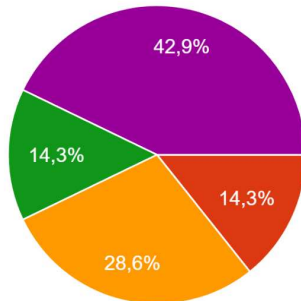
Informe sua faixa etária.

7 respostas



Informe sua ocupação.

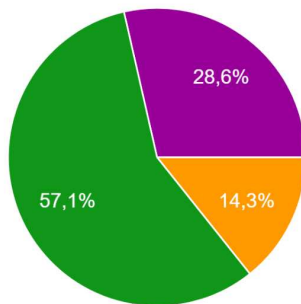
7 respostas



- Aposentado.
- Desempregado.
- Estudante.
- Trabalho.
- Trabalho e Estudo.
- Não trabalha, porém possui muitas obrigações.

Informe seu nível de experiência com aplicativos de celular em geral.

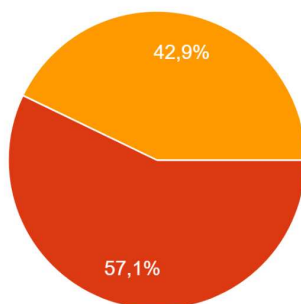
7 respostas



- 1 - Experiência nenhuma.
- 2 - Pouca experiência.
- 3 - Experiência razoável.
- 4 - Muita experiência.
- 5 - Bastante experiência.

Informe sua média de tempo no celular, diariamente.

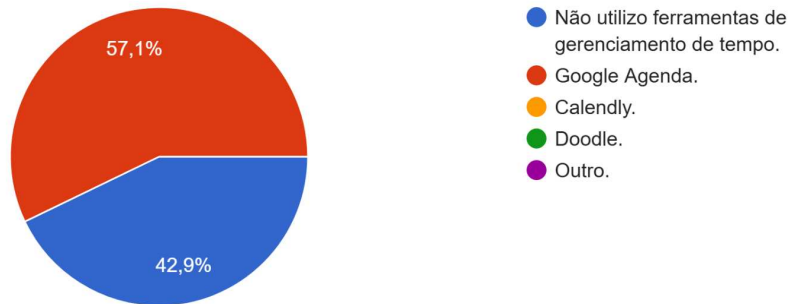
7 respostas



- Menos de 2 horas por dia.
- Entre 2 e 6 horas por dia.
- Entre 6 e 10 horas por dia.
- Mais de 10 horas por dia.

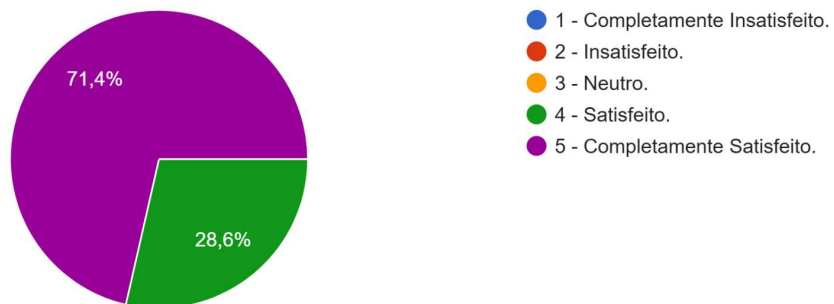
Você utiliza algum aplicativo ou ferramenta para agendar/organizar seu tempo de forma específica? (Google Agenda, Calendly, Doodle, etc).

7 respostas



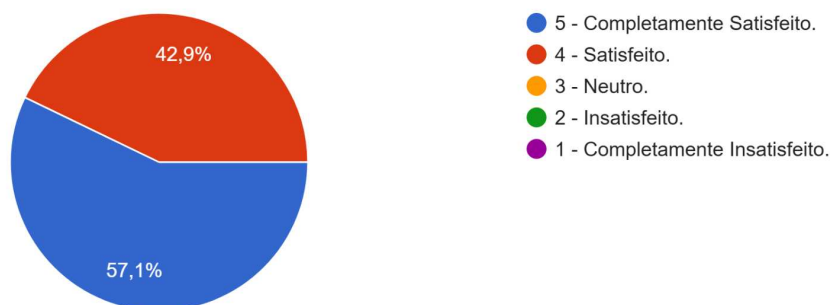
AVALIAÇÃO - TAREFA 1 - Qual o seu grau de satisfação ao realizar o Cadastro de usuário?

7 respostas



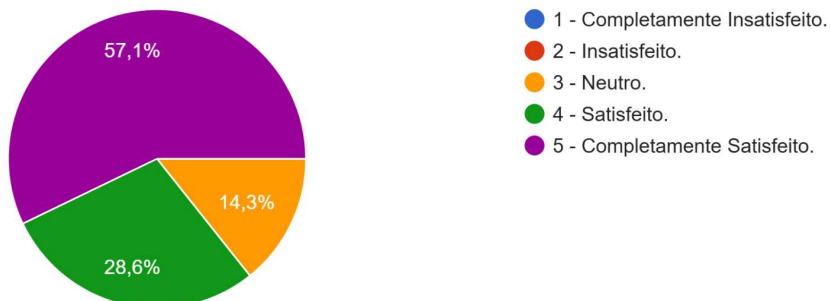
AVALIAÇÃO - TAREFA 1 - Qual o seu grau de satisfação em relação as instruções de cadastro?

7 respostas



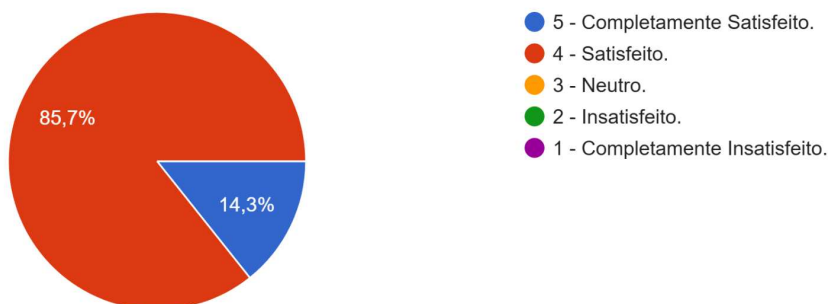
AVALIAÇÃO - TAREFA 1 - Qual o seu grau de satisfação em relação a entrada e saída do sistema (Login e Logout)?

7 respostas



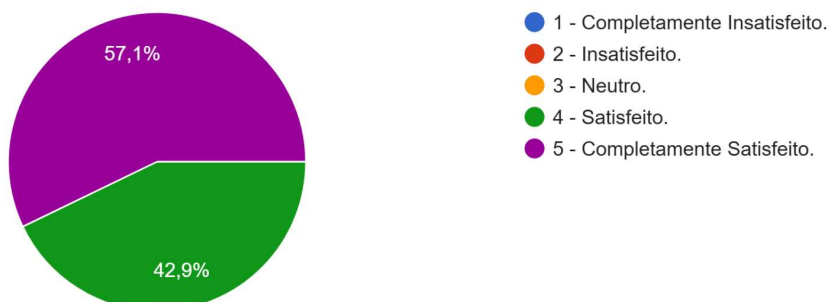
AVALIAÇÃO - TAREFA 2 - Qual o seu grau de satisfação ao preencher o cadastro das atividades?

7 respostas



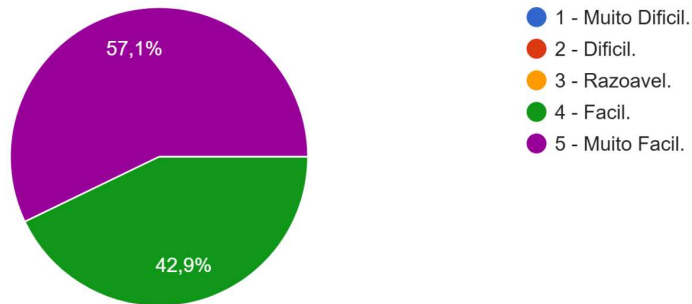
AVALIAÇÃO - TAREFA 2 - Qual o seu grau de satisfação ao encontrar as funcionalidades das atividades (Cadastrar, Editar, Completar, Excluir)?

7 respostas



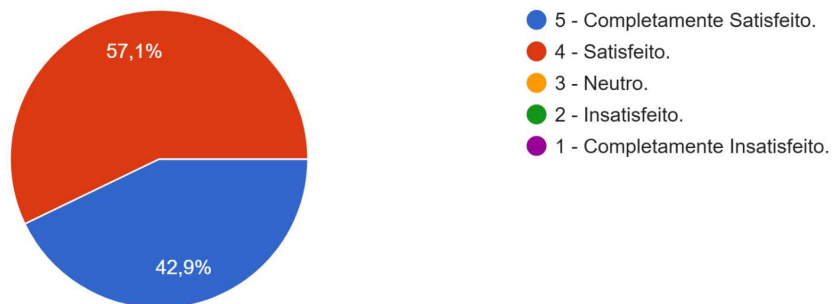
AVALIAÇÃO - TAREFA 2 - Qual o seu grau de dificuldade em realizar o gerenciamento de atividades (Cadastrar, Editar, Completar, Excluir)?

7 respostas



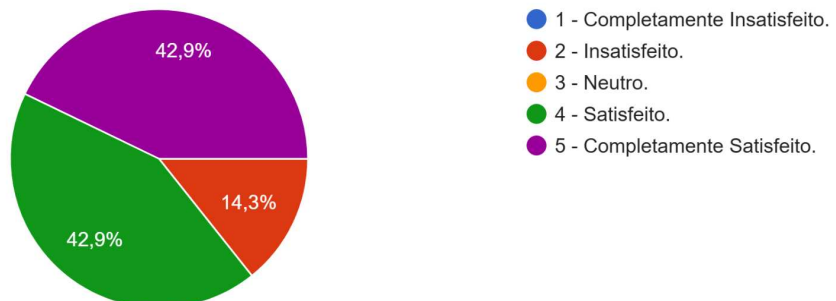
AVALIAÇÃO - TAREFA 2 - Qual o seu grau de satisfação em relação a aparência da agenda?

7 respostas



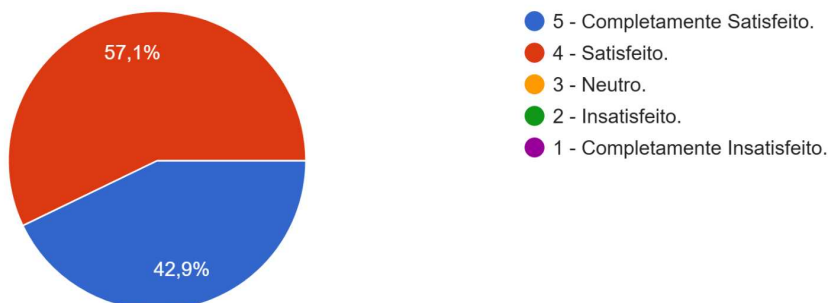
AVALIAÇÃO - TAREFA 3 - Qual o seu grau de satisfação em relação a facilidade de adicionar/remover os seus amigos?

7 respostas



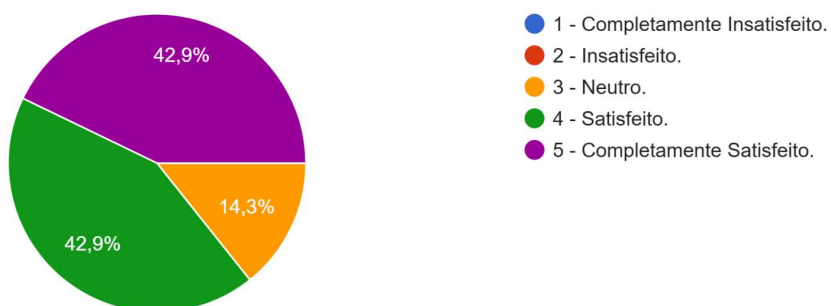
AVALIAÇÃO - TAREFA 4 - Qual o seu grau de satisfação em relação ao criar e gerenciar um grupo? Ex: Adicionar usuarios, Mudar a foto, Adicionar Administrador.

7 respostas



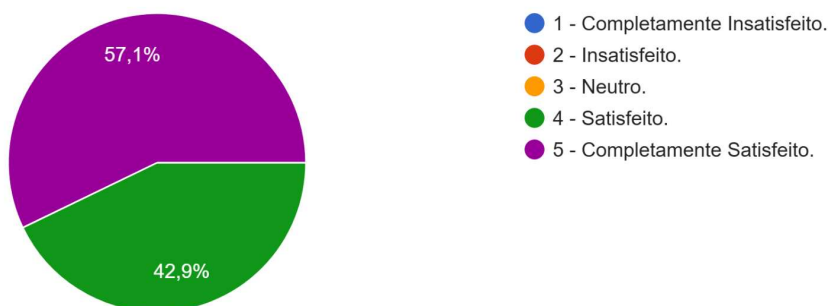
AVALIAÇÃO - TAREFA 4 - Qual o seu grau de satisfação em relação a aparência, e execução das funcionalidades das paginas de gerenciamento de grupo?

7 respostas



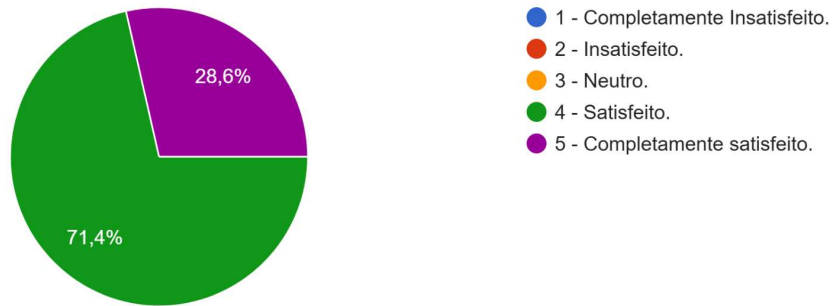
AVALIAÇÃO - TAREFA 5 - Qual o seu grau de satisfação em relação a clareza das funcionalidades da tela de configuração?

7 respostas



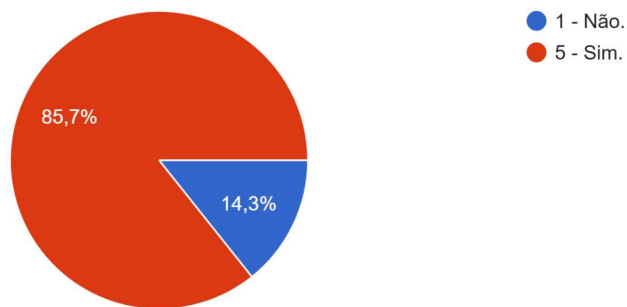
No geral, qual o seu grau de satisfação em relação ao aplicativo Timeshare?

7 respostas



AVALIAÇÃO - Você conseguiu realizar todas as atividades?

7 respostas



AVALIAÇÃO - Você conseguiu realizar todas as atividades?

5 - Sim.

1 - Não.

5 - Sim.

5 - Sim.

5 - Sim.

5 - Sim.

5 - Sim.

AVALIAÇÃO - Caso não tenha conseguido completar, ou encontrou alguma dificuldade nas tarefas, por favor informe o número correspondente e por que não foi possível.

TAREFA 1 - Autenticação de usuários

TAREFA 2 - Gerenciamento de atividades

TAREFA 3 - Gerenciamento de Amigos

TAREFA 4 - Gerenciamento de Grupos

TAREFA 5 - Configurações da Conta

Ex. Formato da resposta: "TAREFA 1 - Não consegui realizar login porque o teclado não abriu."

TAREFA 3 - Não foi possível entrar na agenda do meu amigo e nem excluí-lo.