



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

Simple Finanças - Um aplicativo móvel para organização de finanças pessoais

Gustavo Estevam Sena

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
George Henrique Godim da Fonseca

**Setembro, 2023
João Monlevade–MG**

Gustavo Estevam Sena

**Simple Finanças - Um aplicativo móvel para
organização de finanças pessoais**

Orientador: George Henrique Godim da Fonseca

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Setembro de 2023



FOLHA DE APROVAÇÃO

Gustavo Estevam Sena

Simple Finanças - Um aplicativo móvel para organização de finanças pessoais

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 01 de setembro de 2023

Membros da banca

Prof. Dr. George Henrique Godim da Fonseca - Orientador(a) (Universidade Federal de Ouro Preto)
Prof. Dr. Fernando Bernardes de Oliveira (Universidade Federal de Ouro Preto)
Profa. Dra. Tatiana Alves Costa (Universidade Federal de Ouro Preto)

George Henrique Godim da Fonseca, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 20/09/2023



Documento assinado eletronicamente por **George Henrique Godim da Fonseca, PROFESSOR DE MAGISTERIO SUPERIOR**, em 20/09/2023, às 20:07, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0592629** e o código CRC **10B58D4A**.

Dedico esse trabalho aos meus pais, aqueles cuja compreensão e apoio ao longo de todo o trajeto tornaram possível a construção e realização desse momento.

Agradecimentos

Agradeço a Deus, pai soberano de todos, que me deu forças, saúde e determinação para não se abalar diante das adversidades no caminho, e me manteve motivado até a chegada desse momento.

Agradeço aos meus pais, que me auxiliaram da maneira que melhor encontraram na minha trajetória até a reta final do curso, obrigado por me apoiarem e torcerem pelo meu sucesso Lucilene da Silva Sena e Rodolfo de Sena.

Ao meu orientador, por sua paciência, compreensão e disposição em ajudar durante todo o processo de desenvolvimento desse trabalho.

Aos meus colegas de curso e amigos próximos, pelo apoio nos momentos de maior dificuldade e pelos momentos de socialização, que também são importantes num ambiente acadêmico.

Por fim agradeço aos professores que de fato se comprometem com a missão de promover um ensino de qualidade e acessível dentro das Instituições de ensino públicas do Brasil.

“O homem que não se decide a cultivar o hábito de pensar, perde o maior prazer da vida.”

— Thomas Edison

Resumo

Este trabalho apresenta as etapas para desenvolvimento de um aplicativo móvel para organização financeira pessoal. Seu objetivo principal é auxiliar cidadãos brasileiros interessados em organizar suas finanças a alcançar esse objetivo usando apenas o smartphone Android, conexão com a internet e o aplicativo. O Simple Finanças possibilita para essa finalidade, o cadastro de transações, visualização de saldo total e resultado mensal, visualização de relatórios de receita ou despesa e ajuste mensal de limite para despesas. A comparação com aplicativos similares para a mesma função demonstra que sua interface simples e gratuidade contribuem para um bom posicionamento no mercado, sendo uma opção prática e funcional. Foram utilizadas tecnologias modernas em sua construção, Figma para prototipação, Expo e React Native para o front-end e o Real Time Database do Google Firebase para o back-end, tornando seu desenvolvimento, manutenção e monitoramento mais ágil e prático, assim como a persistência de dados e processo de autenticação de seus usuários. Os resultados obtidos através de testes individuais e com usuários beta foram satisfatórios e evidenciam que a aplicação é estável, rápida e leve, dessa forma é uma solução acessível e útil, que auxilia significativamente aqueles interessados em ter um maior controle e organização sobre suas finanças.

Palavras-chaves: Aplicativo Móvel, Finanças Pessoais, Organização, Android.

Abstract

This document presents the steps for developing a mobile application for personal financial organization. Its main objective is to help Brazilian citizens interested in organizing their finances to achieve this goal using only their Android smartphone, internet connection and the application. For this purpose, Simple Finance makes it possible to register transactions, view the total balance and monthly result, view income or expense reports, and adjust the monthly limit for expenses. Its comparison with similar applications for the same function demonstrates that its simple interface and gratuity contribute to a good positioning in the market, being a practical and functional option. Modern technologies were used in its construction, Figma for prototyping, Expo and React Native for the front-end and the Real Time Database of Google Firebase for the back-end, making its development, maintenance and monitoring more agile and practical, as well as the data persistence and user authentication process. The results obtained through individual tests and with beta users were satisfactory and show that the application is stable, fast and light, thus it is an accessible and useful solution, which significantly helps those interested in having greater control and organization over their finances.

Key-words: Mobile App, Personal Finance, Organization, Android.

Lista de ilustrações

Figura 1 – Telas do Organizze	16
Figura 2 – Telas do Mobills	17
Figura 3 – Visão geral do sistema	20
Figura 4 – Pesquisa sobre os tipos de relatórios para o aplicativo de finanças	22
Figura 5 – Diagrama de casos de uso	22
Figura 6 – Telas do Simple Finanças no Figma - I	32
Figura 7 – Telas do Simple Finanças no Figma - II	33
Figura 8 – Telas do Simple Finanças no Figma - III	33
Figura 9 – Configuração de conexão com o banco de dados	34
Figura 10 – Estrutura <i>users</i> do banco de dados	35
Figura 11 – Estrutura <i>trxhistoric</i> do banco de dados	35
Figura 12 – Estrutura básica de um arquivo <i>app.json</i>	37
Figura 13 – Estrutura básica de um arquivo <i>eas.json</i>	38
Figura 14 – Simple Finanças na Google Play Store	39
Figura 15 – Tela de login	40
Figura 16 – Tela de cadastro de usuário	41
Figura 17 – Tela home - Visualização de transações	42
Figura 18 – Tela home - Pesquisa de transações	42
Figura 19 – Tela home - Excluindo transação	43
Figura 20 – Tela de Lançamento de transação - Despesa	44
Figura 21 – Tela de Lançamento de transação - Receita	44
Figura 22 – Tela de gráficos - Despesas	45
Figura 23 – Tela de gráficos - Receitas	46
Figura 24 – Tela de ajuste de limite para despesas - I	47
Figura 25 – Tela de ajuste de limite para despesas - II	47
Figura 26 – Tela de perfil do usuário - I	48
Figura 27 – Tela de perfil do usuário - II	49

Lista de tabelas

Tabela 1 – Comparativo Mobills, Organizze, Finanças e Simple Finanças	18
---	----

Lista de abreviaturas e siglas

AAB Android App Bundle

APK Android Package Kit

CNC Confederação Nacional do Comércio de Bens, Serviços e Turismo

CNN Cable News Network

EAS CLI Interface de Linha de Comando do Expo Application Services

FGV Fundação Getúlio Vargas

IBGE Instituto Brasileiro de Geografia e Estatística

JSON JavaScript Object Notation

NoSQL Não somente Structured Query Language

SHA Secure Hash Alogrithm

UFRGS Universidade Federal do Rio Grande do Sul

Sumário

1	INTRODUÇÃO	13
1.1	Problema	13
1.2	Objetivos	14
1.3	Justificativa	14
1.4	Estrutura do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Tecnologias Utilizadas	18
2.1.1	React Native	18
2.1.2	Expo	19
2.1.3	Firebase	19
3	METODOLOGIA	20
3.1	Visão Geral do Sistema	20
3.2	Requisitos	21
3.3	Diagrama de Casos de Uso	22
3.3.1	Casos de Uso	23
3.4	Prototipação usando Figma	32
3.5	Banco de Dados	33
3.6	Geração de Build e Versão de Lançamento	36
4	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	40
4.1	Login e Cadastro	40
4.2	Home, Pesquisa e Exclusão de Transação	41
4.3	Lançar Transação	43
4.4	Visualizar Gráfico de Despesa ou Receita	45
4.5	Ajustar Limite de Despesas	46
4.6	Editar dados de Perfil e Fazer Logout	48
4.7	Testes realizados	49
5	CONCLUSÃO E TRABALHOS FUTUROS	51
	REFERÊNCIAS	53

1 Introdução

Segundo a [CNN Brasil](#), ([JANONE; ALPACADA, 2022](#)): um estudo da Confederação Nacional de Comércio de Bens, Serviços e Turismo, [CNC](#), divulgado em Junho de 2022, revelou que 28,7 por cento do total de lares do Brasil era a proporção de famílias endividadas no país em maio daquele mesmo ano.

A falta de organização e controle de finanças pessoais pode levar o indivíduo a gastar mais do que ganha, com isso torna-se mais difícil gerir assuntos financeiros e alcançar realizações pessoais que envolvam injeção de capital. Uma maneira de auxiliar nessa problemática é através do constante registro e monitoramento de transações financeiras, o que permite ao usuário estar sempre por dentro da sua situação. Normalmente isso é feito com o principal intuito de controlar as despesas, criando assim um equilíbrio entre suas contas. Esse trabalho tem como propósito a modelagem, prototipação, desenvolvimento e lançamento de um aplicativo que contribua com a organização e controle de finanças pessoais do usuário, com ênfase mensal, de forma simples, direta e funcional.

Uma pesquisa realizada pela Fundação Getúlio Vargas, [FGV](#), estimou que em 2021 o Brasil possuía aproximadamente 242 milhões de smartphones em uso em Junho daquele ano ([VARGAS, 2021](#)), o que é aproximadamente 30 milhões a mais que a população estimada pelo [IBGE](#), de 212,7 milhões de habitantes, ([IBGE, 2021](#)). Tão grande é a importância dessa tecnologia, que a interação social passou a apresentar mudanças após seu uso, assim como novas formas de compartilhamento e entretenimento foram surgindo.

Apesar do aplicativo ter sido escrito com o auxílio de um *framework* multiplataforma, o React Native, ao se considerar a crescente popularização do uso de smartphones e do sistema operacional Android, que teve uso de 81.57 por cento contra 18.05 por cento de uso do IOS entre Julho de 2022 e Julho de 2023 por pessoas de diversas faixas etárias do mercado brasileiro ([STATCOUNTER, 2022](#)), pensou-se na solução portátil para controle de finanças pessoais, o Simple Finanças.

1.1 Problema

Atualmente as soluções para organização de finanças pessoais apresentam falta de simplicidade em seu uso, o que pode confundir o usuário mais direto, que deseja por exemplo substituir a planilha do excel ou o papel e caneta que hoje usa para organizar suas finanças, há também soluções que funcionam gratuitamente por tempo limitado. No entanto, apesar da alta presença de smartphones no Brasil e da disponibilidade de diversos aplicativos de finanças, muitos indivíduos ainda enfrentam desafios para manter

suas finanças em ordem.

Dentre essas questões, deve-se atentar ao fato de um aplicativo com um propósito específico poder ser simplificado e ajudar o usuário a atingir seus objetivos sem dar voltas.

Nesse contexto, surge a necessidade de uma solução que facilite a organização das finanças pessoais de maneira acessível, use ferramentas atuais de interação com usuário e melhore o engajamento. Considerando esta demanda, foi desenvolvido o Simple Finanças.

1.2 Objetivos

O principal objetivo desse trabalho consiste em prototipar e desenvolver um aplicativo para smartphones com sistema operacional Android, que possibilite organização e controle de finanças pessoais, com ênfase mensal e na população brasileira economicamente ativa.

Como objetivos específicos desse trabalho podem ser citados:

- Cadastrar transações de receita, despesa ou investimento.
- Possibilitar a visualização do resultado mensal, receitas, despesas e saldo mensal e total na tela principal do aplicativo.
- Possibilitar a visualização de relatórios de receitas ou despesas, por categoria.
- Possibilitar a pesquisa e exclusão de transações.
- Possibilitar o agendamento de notificação sobre o limite de suas despesas.
- Facilitar a relação entre o usuário e suas finanças.

1.3 Justificativa

A elaboração desse trabalho tem justificativa no fato de fornecer aos usuários uma nova opção totalmente gratuita, que auxilie na organização e facilite o controle de suas finanças pessoais de forma simples, eficaz e direta.

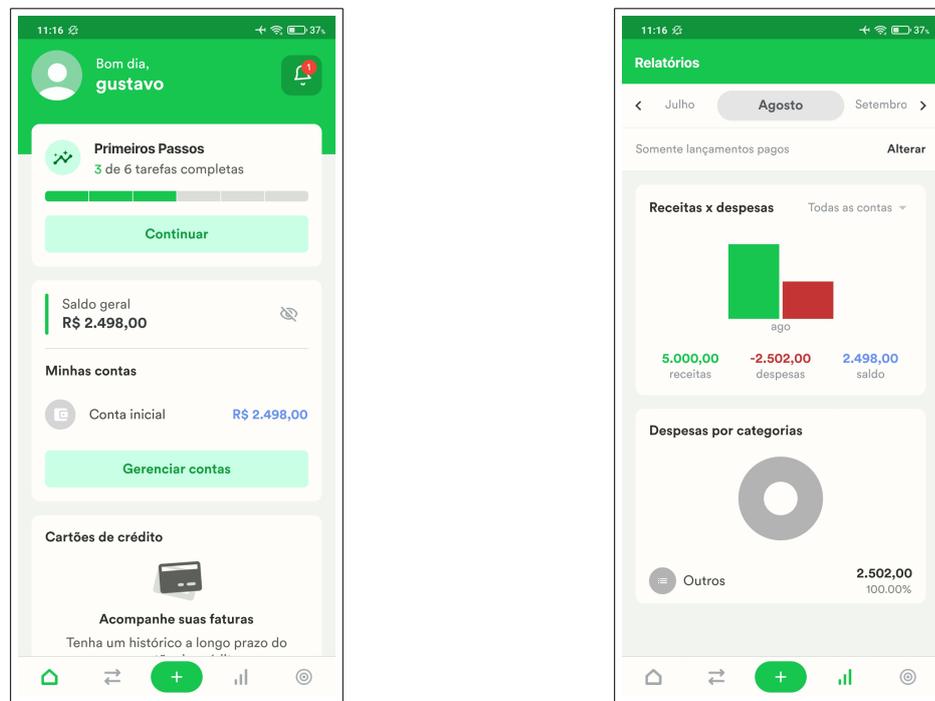
1.4 Estrutura do Trabalho

No [Capítulo 1](#) se faz presente a contextualização do assunto abordado, o problema que leva a sua justificativa, também são citados seus objetivos, principal e específicos e é definido o escopo de abrangência. No [Capítulo 2](#) são abordados aplicativos existentes com funcionalidades semelhantes as do Simple Finanças, ocorre um comparativo com a solução executada e um detalhamento das principais tecnologias para seu desenvolvimento. No

Capítulo 3 ocorre a descrição da metodologia utilizada no desenvolvimento do trabalho, uma visão geral da arquitetura do sistema, seus requisitos, seus casos de uso, prototipação de alta fidelidade, detalhamento do banco de dados e *deploy*. No Capítulo 4 são apresentadas as funcionalidades do aplicativo desenvolvido e os testes realizados para garantia correta do seu funcionamento. Finalmente, Capítulo 5 são feitas as considerações finais e apresentados os trabalhos futuros associados.

2 Fundamentação Teórica

Através de uma busca na Google Play Store, a loja oficial para dispositivos Android, os usuários podem encontrar aplicativos para organização de finanças pessoais, porém muitos deles são de alguma forma pagos e/ou possuem tantas funcionalidades que confundem o usuário mais direto e casual, como exemplo de aplicação paga podemos citar o (ORGANIZZE, 2023), desenvolvido pelos ex-estudantes da UFRGS, Felipe e Tiago, em 2009. O Organizze é um aplicativo, que apesar de layout simplificado e objetivo, tem um sistema de cobranças após o período de testes de 34 dias. A Figura 1 ilustra a tela inicial do aplicativo, assim como um de seus gráficos.



(a) Tela Principal

(b) Tela de Relatórios

Figura 1 – Telas do Organizze

Fonte: (Organizze, 2023)

Esse contexto pode não favorecer uma parcela do público que deseja uma aplicação simples e gratuita, que permita o lançamento e controle de finanças de uma forma prática, substituindo o lançamento manual.

Do ponto de vista humano, o planejamento financeiro emerge como uma ferramenta fundamental para o alcance de uma estabilidade financeira sustentável. Segundo (SOUSA, 2018), “o planejamento financeiro é o processo de formulação de estratégias para auxiliar os indivíduos a gerenciarem seus assuntos financeiros para atingirem seus objetivos de

vida”. Essa afirmação ressalta a importância do planejamento financeiro pessoal como uma abordagem estratégica e intencional para a gestão das finanças. Esse planejamento permite que os indivíduos estabeleçam e alcancem objetivos financeiros a curto, médio e longo prazo, sendo considerado uma premissa fundamental para a realização pessoal.

Através da análise das aplicações de organização financeira pessoal existentes vemos uma carência por uma solução simples, intuitiva e minimalista que ajude a controlar o fluxo financeiro diário. Alguns trabalhos acadêmicos nesse sentido já foram constatados, como é o caso do *finançapp*: aplicação para gestão de finanças pessoais citado no trabalho de (DUECK; MUCHINSKI; OLIVEIRA, 2021), mas a aplicação nunca chegou a ser lançada. Fato este que pode estar relacionado a sua natureza, pois apesar de citado como aplicativo, foi idealizada uma versão web, o que a torna mais difícil e menos flexível para usuários de aparelhos móveis.

Outra opção popular e no contexto de mercado é o (MOBILLS, 2023), desenvolvido pelos então estudantes da Estácio, Mosiah e Terceiro em 2013, e atualmente adquirido e mantido pela Toro Investimentos (EXAME, 2021), que é um aplicativo de controle financeiro com soluções inteligentes pra cartão de crédito e planejamento, porém, pesquisas recentes através da métrica de avaliação do usuário da própria Google Play Store indicam que as soluções citadas estão apresentando dificuldades em seu uso. A Figura 2 ilustra a tela inicial do aplicativo, assim como a sua função de planejamento de gastos.

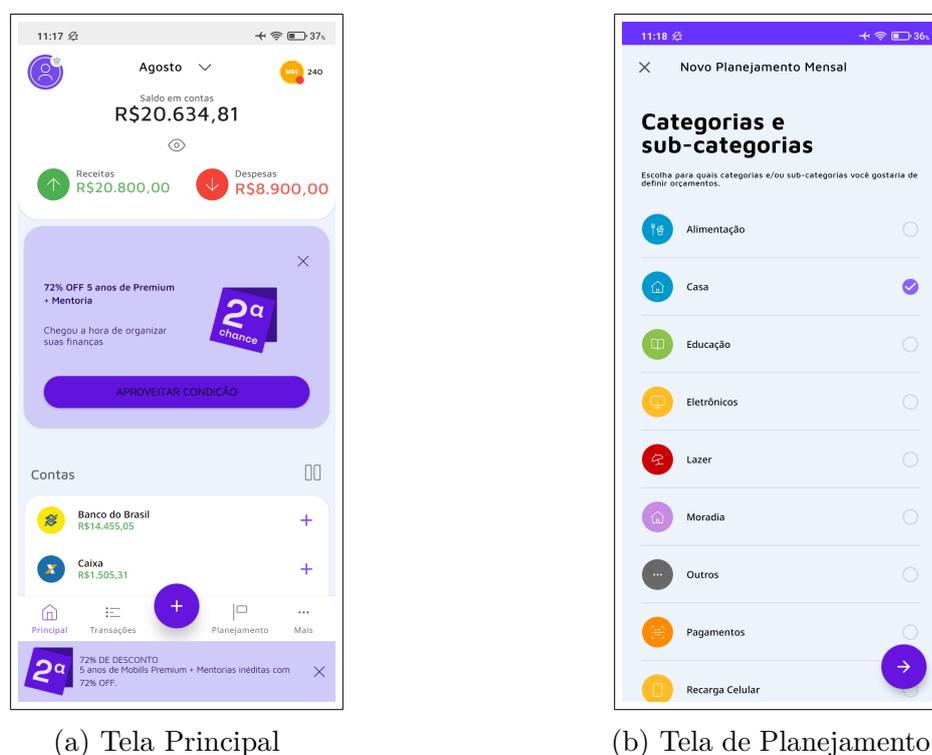


Figura 2 – Telas do Mobills

Fonte: (Mobills, 2023)

A Tabela 1 mostra uma análise comparativa entre as aplicações descritas acima e a solução proposta, Simple Finanças. Nota-se que sua gratuidade, uso simplificado, em especial da definição de limite para despesas, e característica de ser direto e minimalista são os pontos chave da aplicação.

Tabela 1 – Comparativo Mobills, Organizze, Finançapp e Simple Finanças

Aplicativo	Mobills	Organizze	Finançapp	Simple Finanças
Organizador financeiro	V	V	V	V
Resultado mensal na home	V	X	V	V
Ajuste de limite de despesas simplificado	X	X	X	V
Relatórios de despesas por categoria	V	V	V	V
Relatórios de receitas por categoria	V	V	X	V
Interface móvel e funcional	V	V	X	V
Versão gratuita com principais funcionalidades	V	X	X	V
Possui aplicativo lançado em loja oficial	V	V	X	V

V - Possui X - Não Possui

2.1 Tecnologias Utilizadas

Nessa seção serão apresentadas as principais tecnologias utilizadas para a construção, manutenção e lançamento do aplicativo.

2.1.1 React Native

React Native é um *framework* de código aberto usado em diversos aplicativos do mundo, alguns que podem ser citados são o Facebook, Shopify, Discord, Coinbase e Tesla ([REACTNATIVE, 2023](#)).

Através do React Native é possível codificar uma aplicação móvel utilizando a linguagem de programação interpretada JavaScript, o que representa uma enorme vantagem em relação à curva de aprendizagem para desenvolvedores web, por exemplo. Seu uso é similar ao do React, utilizando *views*, que são como as *divs*, *containers* e outras facilidades que flexibilizam a edição de layout. Sua biblioteca de componentes pré-instalados como `TextInput`, `Text` e `Button` é otimizada para dispositivos móveis, conferindo praticidade ao desenvolvimento.

Entre as vantagens do uso do *framework* incluem-se o desenvolvimento multiplataforma, eficiência e rapidez no desenvolvimento, e comunidade ativa com várias bibliotecas de customização, nesse caso utilizamos por exemplo o *datetimpicker community* ([NPM, 2023](#)) como biblioteca de terceiros na nossa aplicação.

Suas desvantagens incluem uma diferença de performance entre um sistema desse tipo e um desenvolvido com código puramente nativo, várias atualizações ao longo do tempo, culminando na necessidade constante de manutenção e tamanho do aplicativo, que tende a ser maior que um aplicativo nativo.

2.1.2 Expo

O Expo é uma ferramenta crucial no cenário de desenvolvimento ágil de software móvel com React Native. Utilizado por uma gama de desenvolvedores, ele facilita o acesso a dependências nativas ([ROCKETSEAT, 2018](#)). Como exemplo podemos citar o sistema de notificação do projeto, que utilizou a API do Expo notifications ([EXPO, 2023c](#)), e a funcionalidade de uso da câmera, que usou a API do Expo camera ([EXPO, 2023a](#)).

Quando se dá início a criação de um projeto com Expo através do comando `npx create-expo-app my-app`, é possível escolher o modelo do template Javascript inicial, sendo em branco, com TypeScript ou com exemplos de telas de navegação.

A facilidade do Expo também impacta o processo de geração de build, que será melhor explicado na [seção geração de build e versão de lançamento](#) desse trabalho, o processo pode ser multiplataforma, isto é, é possível gerar tanto uma *build* para Android quanto para IOS através do mesmo comando, também é possível verificar um histórico de versões das últimas *builds* geradas em ordem de tempo. Sua desvantagem é ter que aguardar um longo período de tempo em fila de baixa prioridade para usuários grátis.

2.1.3 Firebase

O Firebase é uma plataforma de gerenciamento de aplicativos que auxilia no processo de desenvolvimento, especialmente no que tange ao *back-end* da aplicação, este possui uma infraestrutura gerenciada, provendo diversas ferramentas e serviços para ajudar desenvolvedores a criar, melhorar e manter aplicações. ([FIREBASE, 2023b](#))

Entre seus principais serviços incluem-se o módulo de autenticação e a definição de um banco de dados. Os bancos de dados mais importantes são o Realtime Database e o Cloud Firestore, enquanto o primeiro tem como principais vantagens a atualização em tempo real e o suporte offline, armazenando os dados em documentos [JSON](#). O segundo possui estrutura de dados coleção-documento, o que pode ser mais intuitivo de se trabalhar do que [JSON](#), e também apresenta maior facilidade na realização de busca mais complexa. Ambos são escaláveis e possuem custo sob demanda.

O Firebase conta ainda com métricas de liberação e monitoramento, importantes para lançar uma aplicação com confiança, monitorar seu uso, desempenho e estabilidade. E métricas de engajamento, o que pode auxiliar no engajamento do usuário com a aplicação através de análises avançadas, testes experimentais de marketing e campanha de mensagens.

3 Metodologia

Neste capítulo serão apresentados os métodos e procedimentos adotados para o desenvolvimento do trabalho, a visão geral de alto nível do sistema, os requisitos, o diagrama de casos de uso, a prototipação de alta fidelidade com o Figma, uma visão sistemática do banco de dados e questões relacionadas a geração de *build* de produção e lançamento.

3.1 Visão Geral do Sistema

O objetivo principal do sistema é possibilitar ao usuário a organização e controle de suas finanças diárias pessoais, a aplicação pode ser dividida em quatro partes principais: A funcionalidade de lançamento de transações, onde é possível definir características revelantes, como valor da transação, seu tipo, se é receita, despesa ou investimento, sua descrição, categoria e também uma data. A função de visualização de relatórios, onde é possível realizar a consulta de dois tipos de relatórios para receita ou despesa, por categoria. A função de ajustes, onde é possível definir um valor de limite de despesa mensal para ser lembrado na data estipulada. E a função usuário, que permite trocar seu nome e foto por exemplo.

Para armazenamento de dados foi utilizada a plataforma Firebase da Google, através do Real Time Database, uma solução em nuvem que possibilita a sincronização de dados em tempo real, essa ferramenta será abordada de maneira detalhada na [seção banco de dados](#). A Figura 3 a seguir ilustra a visão geral da arquitetura do sistema.

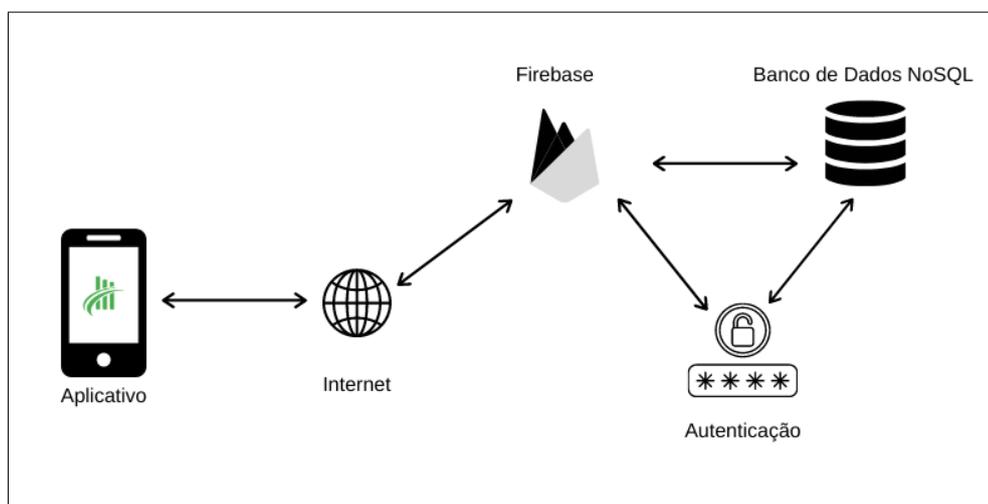


Figura 3 – Visão geral do sistema

3.2 Requisitos

Tomando como base teste realizados nos principais softwares do mercado para organização financeira pessoal, como o Organizze e o Mobillis, além de pesquisa de opinião de possíveis usuários futuros evidenciada através da Figura 4, foram levantados os seguintes requisitos.

Requisitos Funcionais

1. Permitir login social utilizando conta do Google.
2. Permitir cadastro e login através de conta criada no aplicativo.
3. Possibilitar o lançamento de transações do tipo despesa, receita ou investimento.
4. Possibilitar a visualização do saldo (mensal e total), receita e despesas mensais na tela inicial do aplicativo.
5. Possibilitar a visualização das transações relativas ao mês, sua descrição, data, valor e categoria na tela inicial do aplicativo.
6. Viabilizar a pesquisa individual por tipo de transação (despesa, receita ou investimento) na tela inicial do aplicativo.
7. Viabilizar a pesquisa por uma transação específica por descrição ou por valor.
8. Possibilitar a exclusão de uma transação específica.
9. Possibilitar a visualização de relatórios de despesas ou receitas por categoria.
10. Permitir a edição de dados cadastrais do usuário.
11. Viabilizar a definição de um limite mensal de despesas, com agendamento de notificação de lembrete.

Requisitos Não Funcionais

1. Funcionamento ideal em smartphones com Android 5 ou superior.
2. Interface minimalista, possibilitando fácil compreensão, interpretação e utilização.
3. Funcionamento com internet wi-fi ou dados móveis.

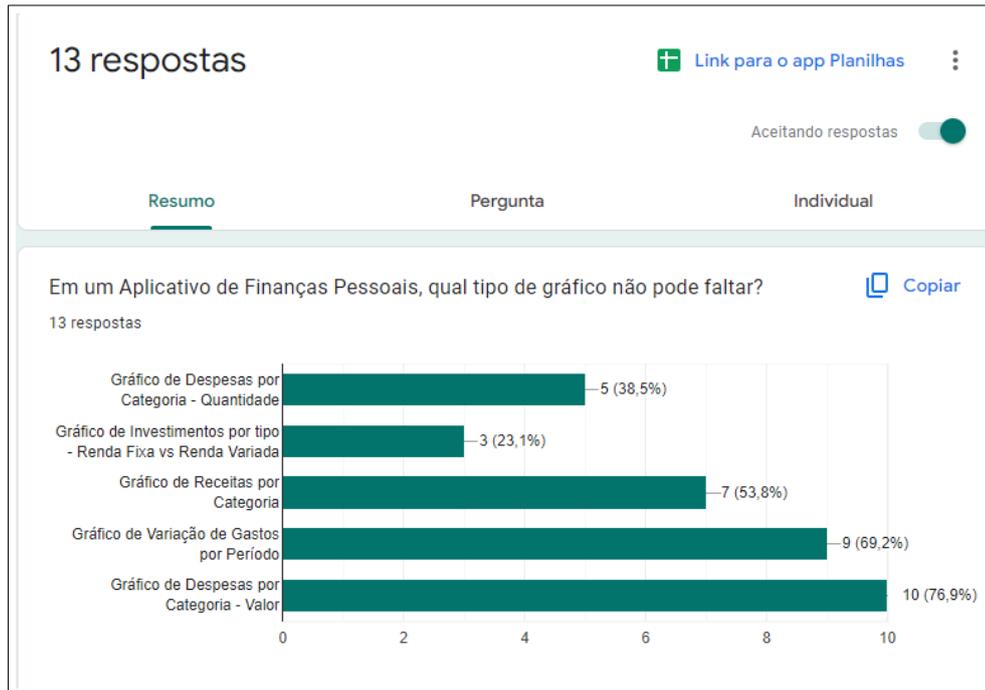


Figura 4 – Pesquisa sobre os tipos de relatórios para o aplicativo de finanças

3.3 Diagrama de Casos de Uso

O diagrama de casos de uso fornece uma ilustração do sistema do ponto de vista do usuário, descrevendo suas principais funcionalidades e a relação de inclusão e extensão entre elas, quando houver. Desse modo foi elaborado o diagrama através do software Astah, conforme mostrado na Figura 5.

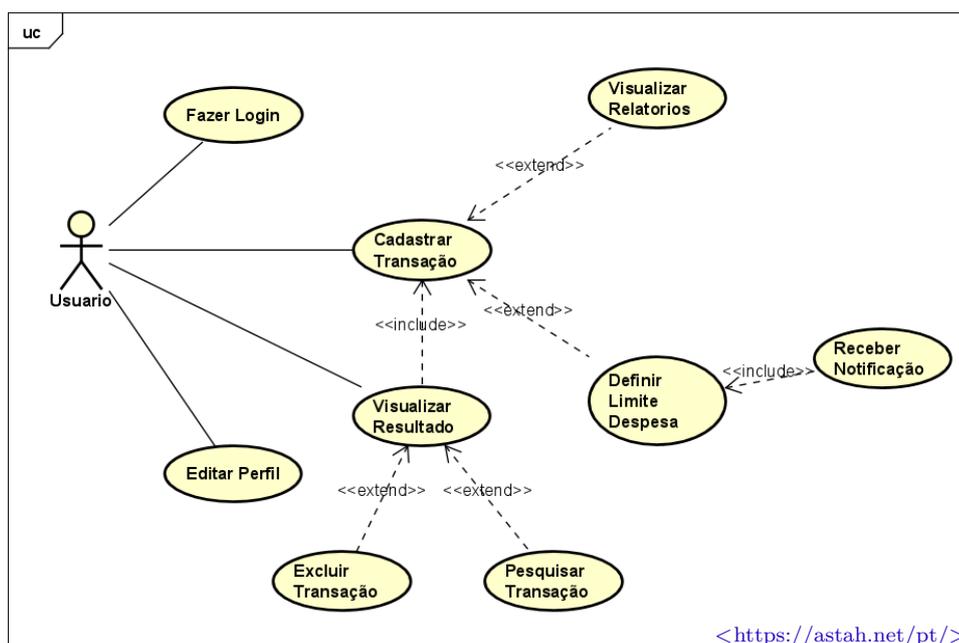


Figura 5 – Diagrama de casos de uso

3.3.1 Casos de Uso

Os casos de uso tratam de estórias textuais que auxiliam na representação do uso da aplicação pelos usuários, tendo um papel importante no levantamento e estruturação dos requisitos do sistema. Nessa seção serão apresentados os diversos casos de uso.

Quadro 1 - Caso de uso - Login

Caso de uso:	Login
Objetivo:	Permite ao usuário validar suas credenciais.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none"> • Aplicativo instalado no smartphone do usuário • Smartphone conectado à internet.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário abre o aplicativo. 2. O usuário acessa sua conta do Google e realiza login. 3. O usuário informa seu e-mail e senha cadastrados e realiza login. 4. O aplicativo redireciona o usuário para a tela principal home. 5. O usuário consegue usar com sucesso as opções do menu do aplicativo.
Extensões:	2a/3a. Smartphone sem conexão com a internet <ul style="list-style-type: none"> • O sistema exibe uma mensagem de erro de conexão.

Quadro 2 - Caso de uso - Cadastro

Caso de uso:	Cadastro
Objetivo:	Permite ao usuário realizar cadastro no sistema.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none">• Aplicativo instalado no smartphone do usuário• Smartphone conectado à internet.
Cenário principal:	<ol style="list-style-type: none">1. O usuário abre o aplicativo.2. O usuário informa nome, e-mail e senha, e ao cadastrar efetua login.3. O usuário é redirecionado para a tela principal do aplicativo (home).
Extensões:	<p>2a. Conexão à internet do smartphone desabilitada.</p> <ul style="list-style-type: none">• O sistema exibe uma mensagem de erro de conexão.

Quadro 3 - Caso de uso - Cadastrar transação

Caso de uso:	Cadastrar transação
Objetivo:	Permite ao usuário cadastrar uma transação do tipo receita/despesa/investimento.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none"> • Aplicativo instalado no smartphone do usuário • Usuário logado com sucesso no aplicativo. • Smartphone conectado à internet.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa o menu de cadastro de transação. 2. O usuário informa o tipo e valor da transação. 3. O usuário opcionalmente informa descrição, seleciona categoria e data. 4. O usuário confirma o registro da transação. 5. O sistema salva a transação corretamente e redireciona o usuário para a tela home.
Extensões:	<p>4a. Conexão à internet do smartphone desabilitada.</p> <ul style="list-style-type: none"> • O sistema espera a internet voltar e grava a transação no banco de dados. <p>2a. Campo valor em branco.</p> <ul style="list-style-type: none"> • O sistema exibe mensagem que deve preencher pelo menos o campo valor.

Quadro 4 - Caso de uso - Visualizar transações e resultado

Caso de uso:	Visualizar transações e resultado
Objetivo:	Permite ao usuário visualizar as transações lançadas, o resultado mensal (receitas - despesas = saldo mensal) e o saldo total na home.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none">• Aplicativo instalado no smartphone do usuário.• Usuário logado com sucesso no aplicativo.
Cenário principal:	<ol style="list-style-type: none">1. O usuário visualiza as transações do mês atual e o resultado do mês atual.2. O usuário visualiza o saldo total lançado no aplicativo.3. O usuário altera o mês e visualiza as transações e resultado para o novo mês definido.

Quadro 5 - Caso de uso - Pesquisar transações

Caso de uso:	Pesquisar transações
Objetivo:	Permite ao usuário visualizar as transações lançadas individualizadas por tipo: receita/despesa/investimento e por descrição/valor.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none"> • Aplicativo instalado no smartphone do usuário. • Usuário logado com sucesso no aplicativo.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário define o tipo de transação. 2. O sistema filtra as transações pelo tipo definido e retorna apenas as desse tipo. 3. O usuário digita uma descrição ou valor de transação. 4. O sistema retorna as transações que correspondem a pesquisa de forma dinamica.
Extensões:	<p>2a/4a. Conexão à internet do smartphone desabilitada.</p> <ul style="list-style-type: none"> • O sistema espera a internet voltar e retorna a consulta.

Quadro 6 - Caso de uso - Excluir transação

Caso de uso:	Excluir transação
Objetivo:	Permite ao usuário excluir uma transação do tipo receita/despesa/investimento.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none">• Aplicativo instalado no smartphone do usuário.• Usuário logado com sucesso no aplicativo.
Cenário principal:	<ol style="list-style-type: none">1. O usuário mantém o dedo pressionado em uma transação na tela home.2. O sistema exibe uma mensagem sobre a confirmação da exclusão da transação.3. O usuário pressiona continuar na mensagem de confirmação.4. A transação é excluída, os valores mensal e total são atualizados.
Extensões:	<p>2a. O usuário clicar em cancelar na mensagem de confirmação de exclusão.</p> <ul style="list-style-type: none">• A transação e os valores mensal e total são mantidos.

Quadro 7 - Caso de uso - Visualizar relatórios de despesas

Caso de uso:	Visualizar relatórios de despesas
Objetivo:	Permite ao usuário visualizar relatórios de despesas por mês, agregados por categoria.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none">• Aplicativo instalado no smartphone do usuário.• Usuário logado com sucesso no aplicativo.
Cenário principal:	<ol style="list-style-type: none">1. O usuário seleciona o menu despesas.2. O sistema trás um relatório de evolução de despesas no mês, dia a dia.3. O usuário clica em categoria.4. O sistema trás um relatório de despesas acumulando-as mensalmente por categoria.

Quadro 8 - Caso de uso - Visualizar relatórios de receitas

Caso de uso:	Visualizar relatórios de receitas
Objetivo:	Permite ao usuário visualizar relatórios de receitas por mês, agregados por categoria.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none">• Aplicativo instalado no smartphone do usuário.• Usuário logado com sucesso no aplicativo.
Cenário principal:	<ol style="list-style-type: none">1. O usuário seleciona o menu receitas.2. O sistema trás um relatório de receitas acumulando-as mensalmente por categoria.3. O usuário clica em evolução.4. O sistema trás um relatório de evolução de receitas no mês, dia a dia.

Quadro 9 - Caso de uso - Editar dados de perfil

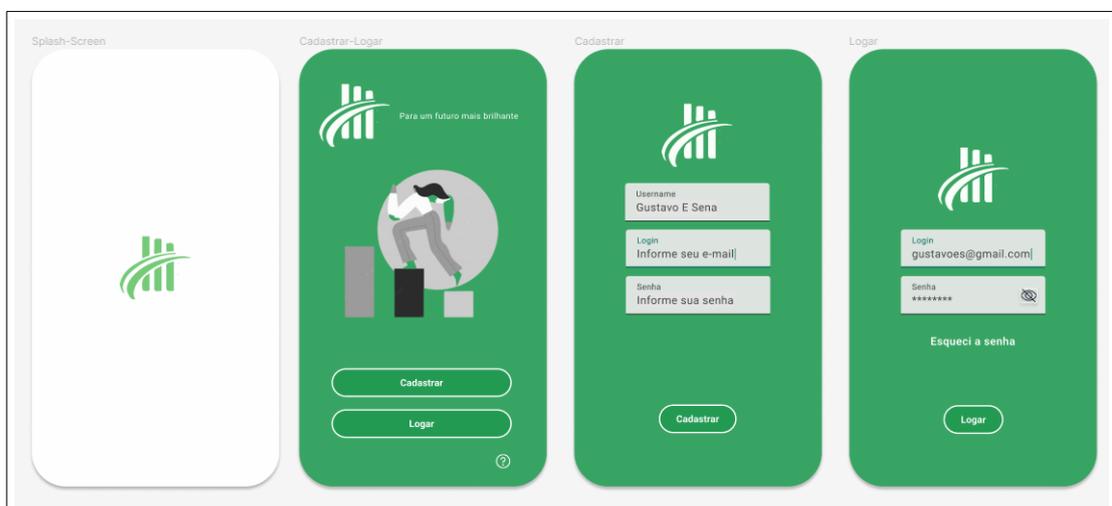
Caso de uso:	Editar dados de perfil
Objetivo:	Permite ao usuário editar seu nome, foto e frase em sua página de perfil.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none">• Aplicativo instalado no smartphone do usuário.• Usuário logado com sucesso no aplicativo.• Smartphone conectado à internet.
Cenário principal:	<ol style="list-style-type: none">1. O usuário acessa o menu perfil pressionando o ícone do usuário no header da home.2. O sistema trás as informações do usuário.3. O usuário altera seu nome e/ou frase e foto e salva.4. O sistema atualiza as informações do usuário.
Extensões:	<p>2a. Conexão à internet do smartphone desabilitada.</p> <ul style="list-style-type: none">• O sistema exibe a mensagem carregando informações, até reconectar à internet

Quadro 10 - Caso de uso - Definir limite mensal para despesas

Caso de uso:	Definir limite mensal para despesas
Objetivo:	Permite ao usuário definir um limite mensal para despesas e uma data para ser lembrado se está gastando acima ou abaixo.
Ator:	Usuário.
Pré-condições:	<ul style="list-style-type: none"> • Aplicativo instalado no smartphone do usuário. • Usuário logado com sucesso no aplicativo.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário seleciona o menu ajustes. 2. O sistema trás as informações de despesas mensais e o limite atual. 3. O usuário altera o limite e define uma data. 4. O sistema atualiza as informações de limite e notifica o usuário na data selecionada.

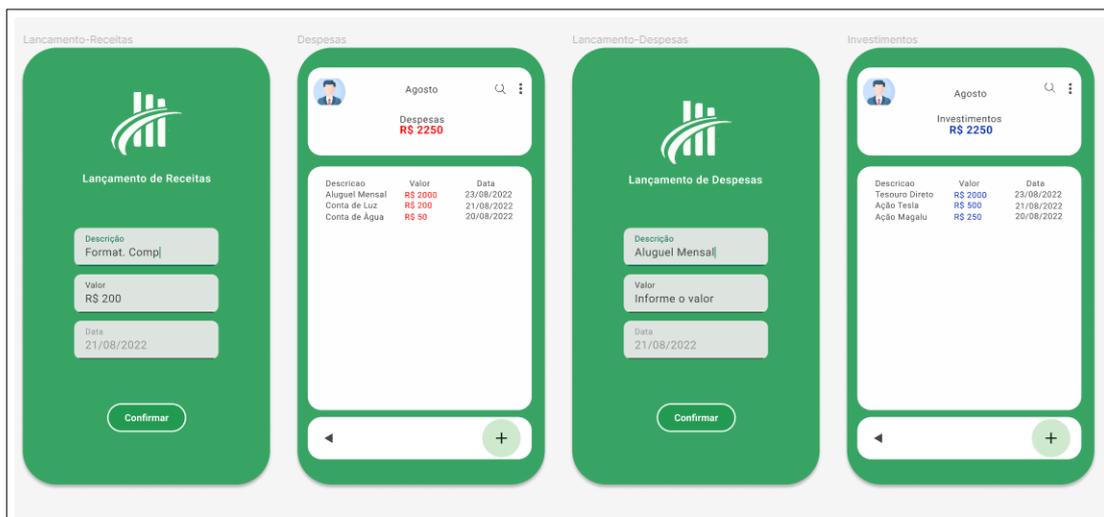
3.4 Prototipação usando Figma

Como maneira de simplificar e agilizar a visualização inicial do *front-end* das telas do sistema, e também de analisar boas práticas de usabilidade e interação com o usuário, foi desenvolvida uma prototipação (FIGMA, 2022) de alta fidelidade com o Figma conforme Figuras 6, 7 e 8.



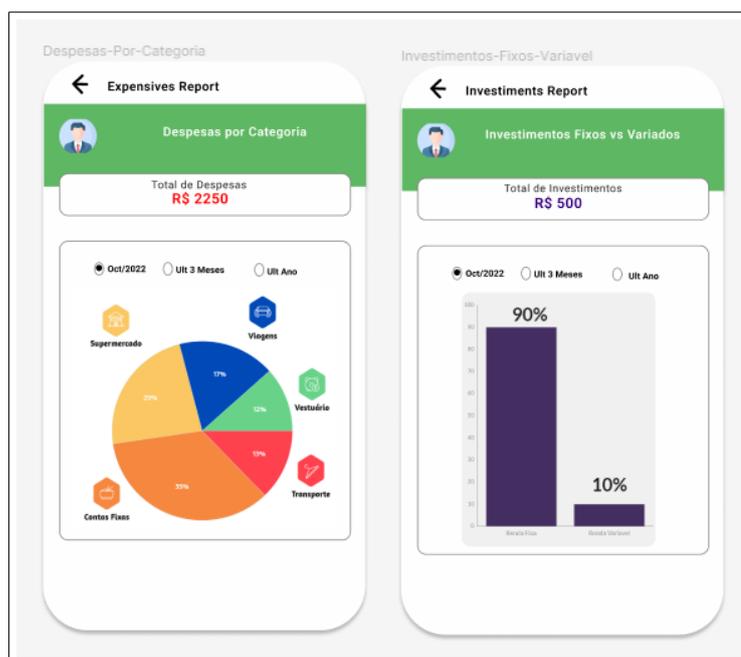
(a) Telas de login e cadastro

Figura 6 – Telas do Simple Finanças no Figma - I



(a) Tela de lançamentos e home

Figura 7 – Telas do Simple Finanças no Figma - II



(a) Relatórios

Figura 8 – Telas do Simple Finanças no Figma - III

3.5 Banco de Dados

O banco de dados escolhido para implementação da aplicação foi o Realtime Database do Firebase Google (FIREBASE, 2023a), o Realtime Database é um banco de dados em nuvem, no modelo NoSQL, que usa arquivos JSON para armazenar informações do tipo chave-valor, o que permite a sincronização de dados em tempo real e a consulta desses dados mesmo com a aplicação off-line. A opção de escolha, entre outros fatores

levou em consideração a facilidade de configuração e conexão com o banco de dados, a velocidade de armazenamento e sincronização de informações, a facilidade de acesso para verificação e manutenção e a possibilidade de escalonamento garantido através dos planos de preço do Firebase.

A estrutura do projeto no firebase individualiza o banco de dados e o módulo de autenticação, o que permite uma maior flexibilidade na análise e uso da aplicação. O banco de dados conta com a definição de regras para validação de segurança, desse forma é possível garantir que apenas as pessoas logadas no sistema possam executar ações de leitura e escrita.

Para prover a autenticação do usuário é preciso acessar a guia *Authentication* no Console do Firebase da Google e definir os métodos de login em *Sign-in*, para essa aplicação foram habilitados e-mail/senha e Google, é importante destacar que para se trabalhar com autenticação via Google é necessário informar um número de *hash SHA*, que corresponde as impressões digitais do projeto no momento de geração da *build*.

A comunicação com o banco de dados é estabelecida através de um arquivo de configuração `firebaseConnection.js`, conforme mostrado na Figura 9, esse arquivo contém os dados de autorização de acesso gerados no momento de configuração do banco para Android SDK dentro do Realtime Database, mais especificamente nos arquivos `google-services.json`, essas configurações são como as indicados na Figura 9 a seguir:

```
import firebase from 'firebase/compat/app';
import 'firebase/compat/auth'; import 'firebase/compat/database';
const firebaseConfig = {
  apiKey: "exemplo",
  authDomain: "exemplo",
  projectId: "exemplo",
  storageBucket: "exemplo",
  messagingSenderId: "exemplo",
  appId: "exemplo",
  databaseURL: "exemplo",
  measurementId: "exemplo"
};
```

Figura 9 – Configuração de conexão com o banco de dados

A estrutura do banco de dados para esse projeto é simples e funcional, favorecendo a leveza da aplicação. Basicamente, quando um usuário realizada cadastro ou login via conta da Google é criado um registro na estrutura **users**, tendo como filho o **userID** único desse usuário, também são cadastradas informações de inicialização para os campos pertinentes a esse usuário, por exemplo, o usuário beta Tulio Henrique, exemplificado na Figura 10 possui os campos: *expense*, *revenue*, *balance*, *email*, *name*, *phrase*, *limit* e

investment, os valores são devidamente inicializados para um novo usuário, e se alteram à medida em que este faz o uso das funções do sistema.



Figura 10 – Estrutura *users* do banco de dados

De forma análoga, quando um usuário realiza o lançamento de uma transação é gerado um registro na estrutura *trxhistoric*, exemplificado na Figura 11, que contém os campos específicos dessa transação, nesse caso é importante notar que o *userID* encapsula a transação e também é gravado como forma de reforçar o vínculo.



Figura 11 – Estrutura *trxhistoric* do banco de dados

Uma observação importante é que o Realtime Database não impõe restrição rígida aos tipos de dados armazenados, portanto, essa restrição deve ser realizada no *front-end* da aplicação.

O Realtime Database suporta os seguintes tipos de dados básicos:

1. Número: Pode ser inteiro ou ponto flutuante.
2. String: Qualquer sequência de caracteres.
3. Booleano: *true* ou *false*.
4. Objeto: Um conjunto de pares chave-valor, onde as chaves são strings e os valores podem ser qualquer um dos tipos de dados do Firebase (incluindo outro objeto).
5. Array: Uma lista ordenada de valores. No entanto, o Firebase não tem um tipo de dado *array* nativo. Em vez disso, ele usa objetos com chaves numéricas ordenadas para representar arrays.
6. Nulo: Um campo vazio. É usado para representar a ausência de valor.

A manipulação dos dados visualizados nos relatórios de despesa e receita e pesquisas mensais também foi realizada no *front-end*, através da utilização do JavaScript e recursos do Expo e React Native, de forma a simplificar a construção e desenvolvimento do projeto.

3.6 Geração de Build e Versão de Lançamento

Como foi utilizado a ferramenta Expo durante todo o ciclo de desenvolvimento do aplicativo, a geração de build ocorreu seguindo o fluxo do [EAS CLI](#), *configure EAS Build with eas.json* ([EXPO, 2022](#)), esse fluxo é dividido nos seguintes passos:

Passo 1. Configuração do app.json

App.json é o arquivo de configuração do projeto Expo, onde são especificados os detalhes sobre o aplicativo, como nome, versão e outras configurações específicas de plataforma. A Figura 12, mostrada após as informações, ilustra a estrutura básica de um arquivo app.json.

Informações mais importantes desse arquivo:

"name": O nome do aplicativo.

"slug": O slug do aplicativo, usado para gerar a URL do aplicativo na App Store.

"version": A versão do aplicativo.

"android.package": O nome do pacote para o aplicativo Android, que deve ser único na Google Play Store.

"android.versionCode": A versão do código para a sua build Android. Deve ser incrementada a cada nova build.

```
{
  "expo": {
    "name": "SeuAppName",
    "slug": "seu-app-name",
    "version": "1.0.0",
    "platforms": ["ios", "android"],
    "android": {
      "package": "com.seudominio.seuappname",
      "versionCode": 1
    }
  }
}
```

Figura 12 – Estrutura básica de um arquivo app.json

Passo 2. Login no Expo

É necessário estar logado na sua conta do Expo para fazer a build, deve-se usar o seguinte comando para fazer login via linha de comando: `eas login`, Em seguida é necessário informar o usuário e senha cadastrados no site do Expo.

Passo 3. Gerenciamento de credenciais

Para assinar o aplicativo para distribuição na Google Play Store é preciso um arquivo de chave de assinatura contendo entre outros dados a credencial [SHA](#). O Expo pode gerenciar todo o processo ou pode ser feito o gerenciamento manual.

Nesse caso o gerenciamento ficou por conta do Expo, logo, na primeira vez que se tenta buildar o aplicativo com o comando `eas build -profile <profile-name>`, o Expo irá perguntar a plataforma de geração de build se o desenvolvedor gostaria que ele lide com as credenciais. Em caso positivo, o Expo usará as configurações do arquivo `eas.json` representado na Figura 13 para criar um arquivo de chave de assinatura e o armazenará em sua nuvem.

```
{
  "cli": {
    "version": ">= 4.1.2"
  },
  "build": {
    "development": {
      "developmentClient": true,
      "distribution": "internal"
    },
    "preview": {
      "android": {
        "buildType": "apk"
      },
      "distribution": "internal"
    },
    "preview2": {
      "android": {
        "gradleCommand": ":app:assembleRelease"
      }
    },
    "preview3": {
      "developmentClient": true
    },
    "production": {}
  },
  "submit": {
    "production": {}
  }
}
```

Figura 13 – Estrutura básica de um arquivo eas.json

Passo 4. Iniciar a build

Após a configuração do app.json, eas.json e login no Expo, dá-se início ao processo de *build* de uma versão para desenvolvimento, [APK](#) de teste interno ou *deploy*. Sendo que para *deploy* é utilizado o comando: `eas build --profile production`.

É importante notar que durante o desenvolvimento do aplicativo também foram geradas versões para teste com [APK](#) interno: *preview* e para desenvolvimento: *development*.

Passo 5. Acompanhar o progresso da build

O Expo irá empacotar o aplicativo e o desenvolvedor receberá um *link* para acompanhar o progresso. Acesse esse *link* em um navegador para ver o status da sua *build*.

Passo 6. Baixar o arquivo de build

Quando a build for concluída, o desenvolvedor receberá um *link* para baixar o arquivo [AAB](#) gerado. Outras *builds* geradas também podem ser encontradas no site do Expo, sob a seção

de *builds* do seu projeto.

Passo 7. Testar o arquivo de build

Antes de enviar para a Google Play Store, é necessário realizar um teste do [AAB](#) no seu dispositivo ou emulador.

Passo 8. Configurar e publicar o aplicativo na Google Play Store

Envie o arquivo através do Google Play Console, configure os detalhes do aplicativo, como descrição, screenshots, e depois publique o aplicativo.

Passo 9. Baixar e usar o aplicativo da Play Store

Após o aceite do aplicativo pela Google, você pode acessar via Google Play Store, pesquisando por [Simple Finanças](#). Ou se preferir, através do link:

<https://play.google.com/store/apps/details?id=com.gultesguilda.simplefinancas>

Uma vez realizado o download do aplicativo, deve-se abrir e criar uma conta ou logar com o Gmail para iniciar seu uso. A Figura 14 mostra o aplicativo na Google Play Store:

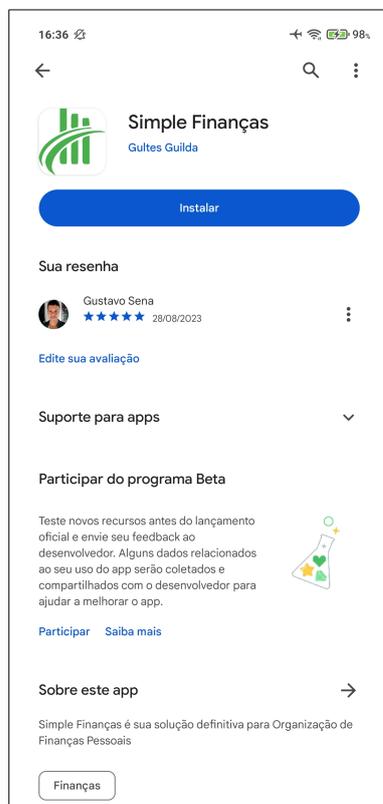


Figura 14 – Simple Finanças na Google Play Store

4 Apresentação e Análise dos Resultados

Neste capítulo serão apresentados os resultados do desenvolvimento do aplicativo Simple Finanças, uma solução para organização financeira pessoal, suas funcionalidades e comportamento em situações gerais de uso, ao final também serão abordados os testes realizados.

4.1 Login e Cadastro

Como primeiro passo, após o usuário baixar, instalar e abrir o sistema, após a splash screen haverá a abertura da tela de login, representada pela Figura 15, onde poderá ser realizado o login com conta Google ou o login utilizando e-mail e senha cadastrados.

Ao clicar em criar uma conta, o usuário será direcionado para a tela cadastrar, representada pela Figura 16, onde informando nome, um e-mail válido e senha, irá logar no sistema. Nessa tela também se encontra a recuperação de senha para um usuário cadastrado.

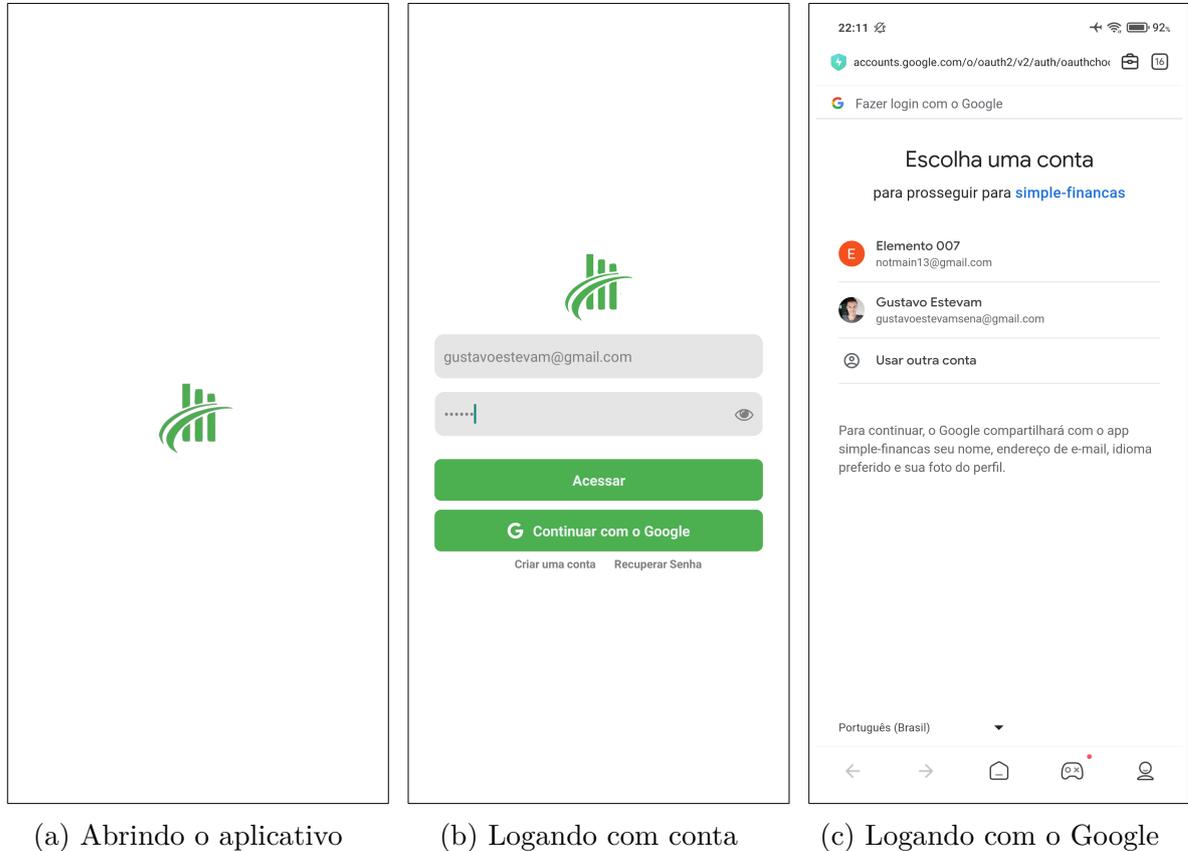


Figura 15 – Tela de login

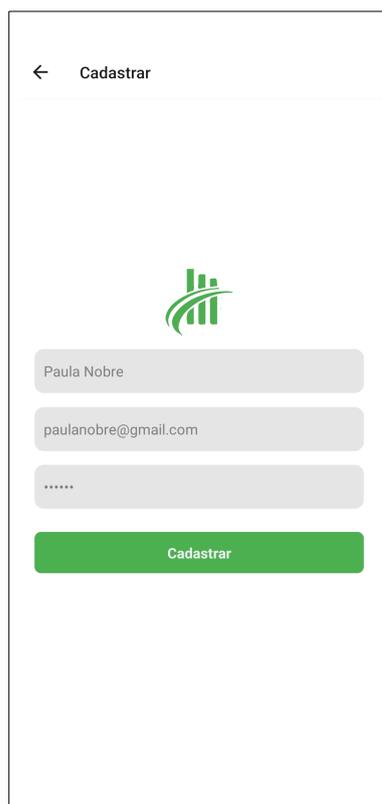


Figura 16 – Tela de cadastro de usuário

4.2 Home, Pesquisa e Exclusão de Transação

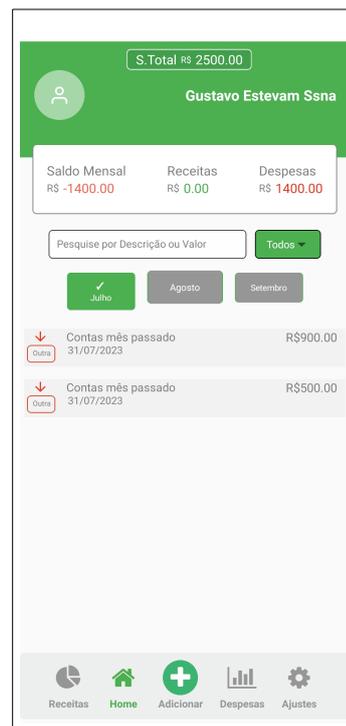
Após o login bem sucedido, na tela principal do aplicativo, a home, é possível visualizar as transações recentemente lançadas, assim como o resultado para o mês atual, isto é, o saldo mensal, receitas e despesas, também é possível visualizar o saldo total lançado na conta, e alternar entre os meses, fato que fica mais claro ao observarmos a Figura 17.

Da home e de todas as telas, é possível navegar pelas variadas funções do sistema, de forma intuitiva com o auxílio do menu no rodapé da tela, ou através do ícone de perfil no cabeçalho para perfil.

Além disso é possível pesquisar por tipo de transação, isto é, apenas receita, apenas despesa ou apenas investimento. Ou ainda, pesquisar por uma transação específica de forma dinâmica através de sua descrição ou valor, conforme demonstrado na Figura 18.



(a) Visualizando transações de Agosto

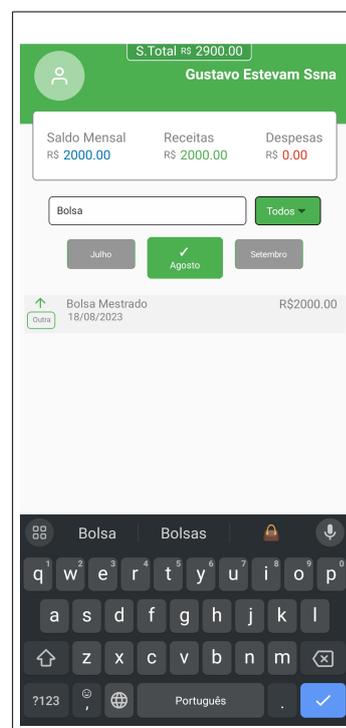


(b) Visualizando transações de Julho

Figura 17 – Tela home - Visualização de transações



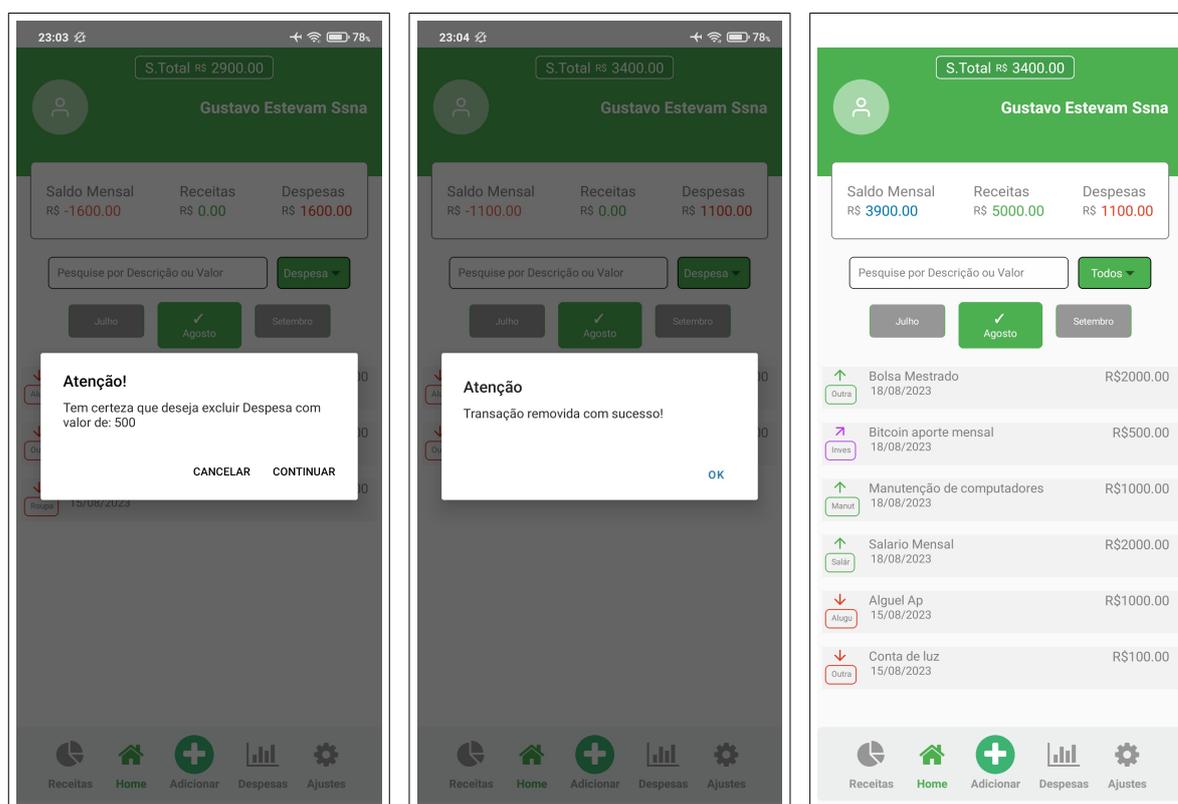
(a) Pesquisando por tipo: despesa



(b) Pesquisando por descrição ou valor

Figura 18 – Tela home - Pesquisa de transações

Por fim, nessa tela é possível realizar a exclusão de uma determinada transação através de pressionar e segurar com o dedo acima do seu registro, em caso de confirmação, os valores serão atualizados no cabeçalho, e o usuário receberá *feedback* adequado. Essa funcionalidade é identificada na Figura 19.



(a) Antes de confirmar

(b) Exclusão Confirmada

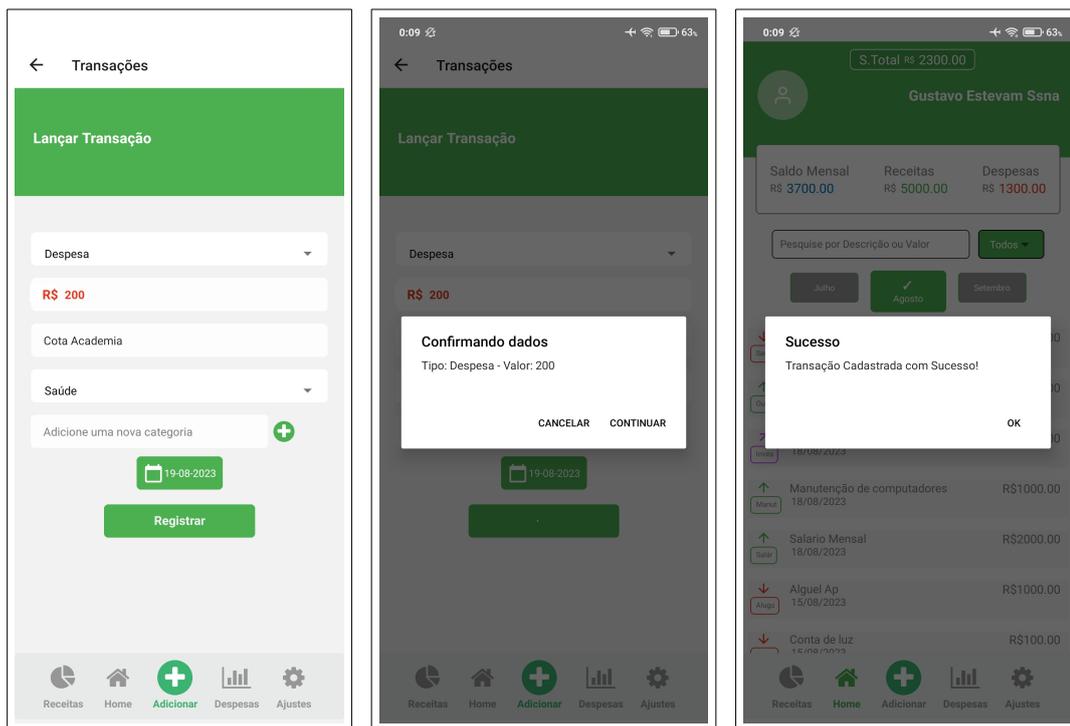
(c) Saldo atualizado

Figura 19 – Tela home - Excluindo transação

4.3 Lançar Transação

Ao selecionar o menu adicionar, é possível acessar a tela lançar transação, representada pela Figuras 20 e 21.

Na tela de lançamento de transação é possível cadastrar uma despesa, receita ou investimento, aqui damos destaque ao valor, que terá sua formatação alterada a depender do tipo lançado. Também é possível selecionar uma categoria ou incluir uma nova, sendo as seis primeiras da lista nativa da aplicação. Por fim é possível escolher a data de lançamento, após a confirmação os totais serão atualizados na home, para onde o usuário será redirecionado.

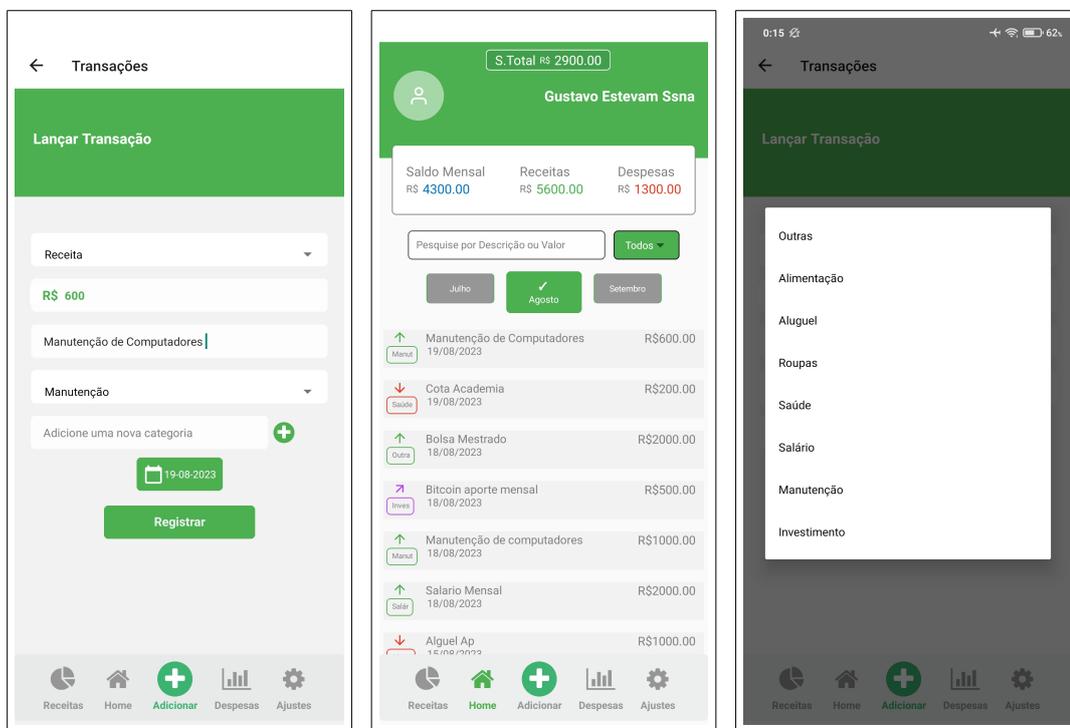


(a) Antes de confirmar

(b) Confirmando

(c) Transação Confirmada

Figura 20 – Tela de Lançamento de transação - Despesa



(a) Antes de confirmar

(b) Transação Confirmada

(c) Categorias

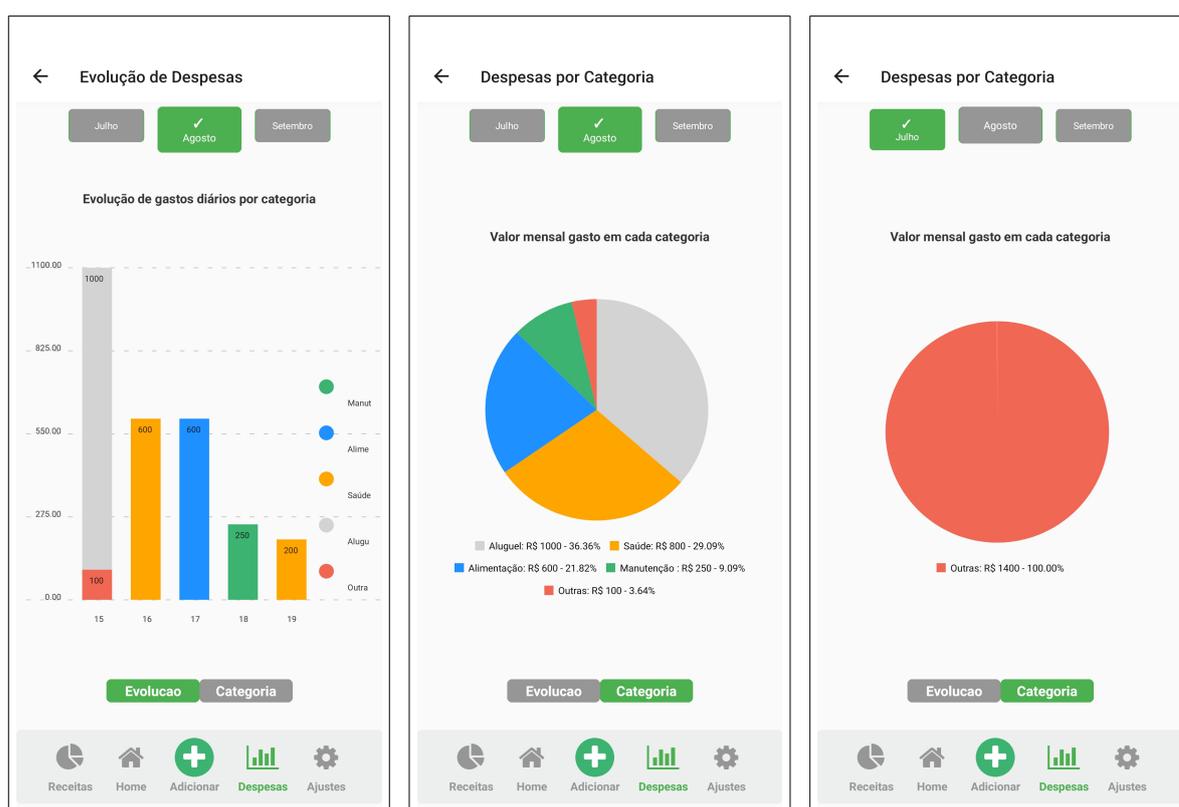
Figura 21 – Tela de Lançamento de transação - Receita

4.4 Visualizar Gráfico de Despesa ou Receita

Ao selecionar o menu despesas ou receitas, Figuras 22 e 23, é possível verificar gráficos para cada um desses tipos de transação, separados por categoria. O ícone dos menus são diferentes para referenciar a existência de mais de um tipo de gráfico.

Na tela de qualquer um dos gráficos é possível alternar entre: por categoria e por evolução, o gráfico de categorias exibe o valor mensal gasto ou ganho em cada categoria, em reais e percentual, e o gráfico por evolução exibe o valor gasto ou ganho dia a dia para aquela determinada categoria dentro do mês. É possível visualizar para qualquer mês desejado.

Por fim, caso não haja receitas ou despesas no mês desejado é exibida uma mensagem para o usuário, informando que se não houver dados suficientes para gerar um gráfico, deve-se realizar um lançamento.



(a) Evolução diária de despesas por categoria (b) Acumulado mensal de despesas por categoria (c) Visualização em mês anterior

Figura 22 – Tela de gráficos - Despesas

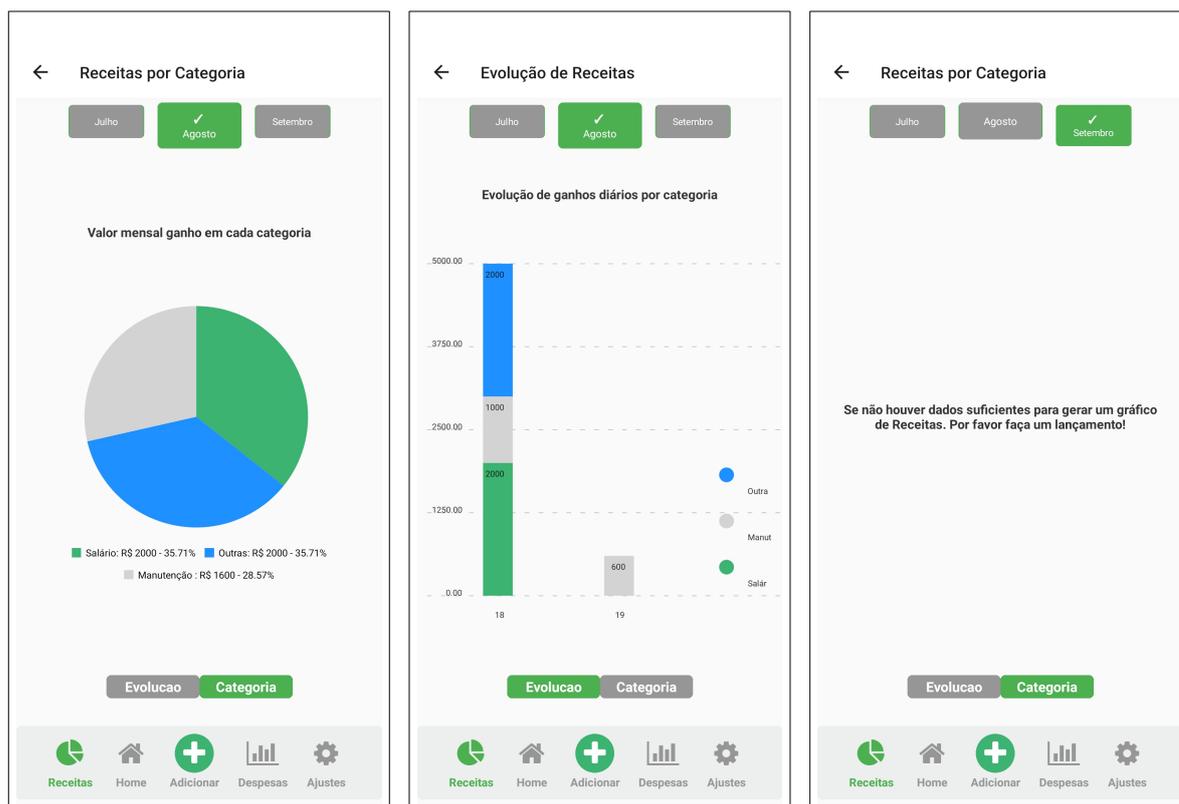


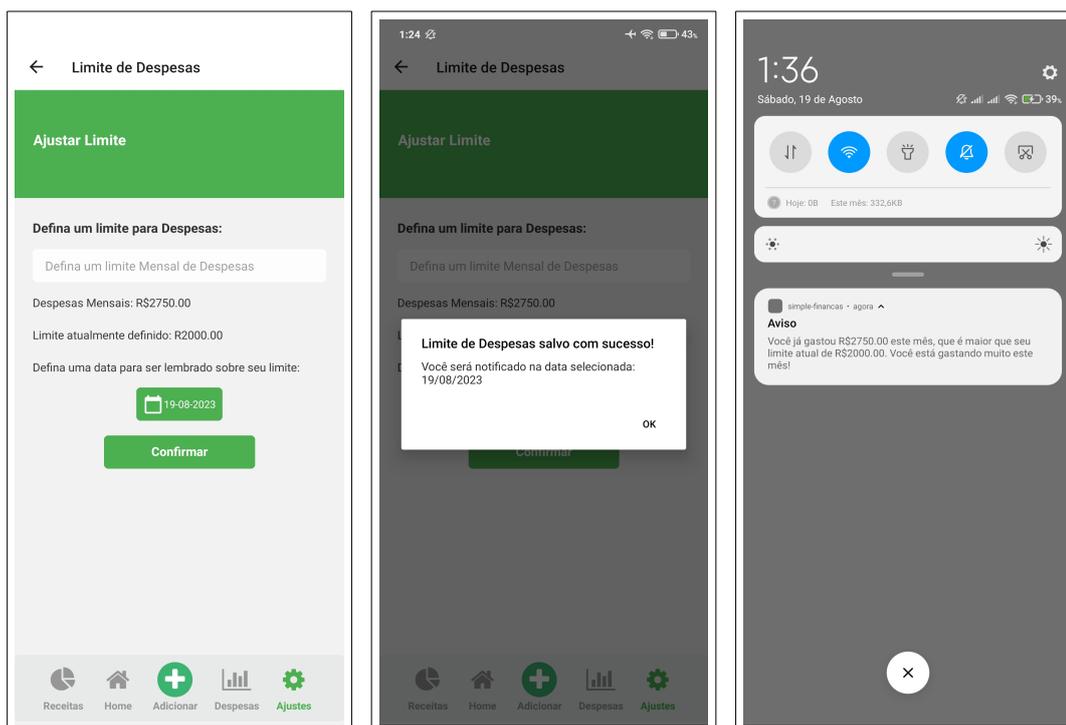
Figura 23 – Tela de gráficos - Receitas

4.5 Ajustar Limite de Despesas

Ao selecionar o menu ajustes, Figuras 24 e 25, é possível acessar a tela de definição de limite para despesas, com isso um usuário que tenha despesas lançadas poderá colocar um valor e uma data para ser lembrado sobre esse limite.

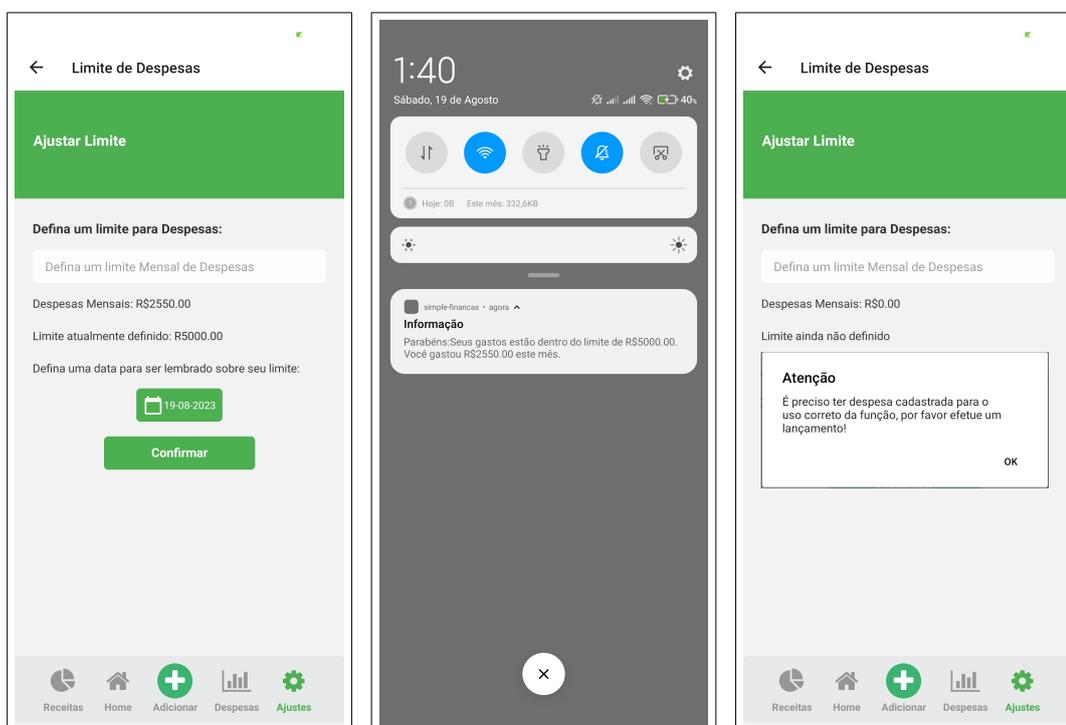
Na data selecionada pelo usuário, o sistema fará uma validação entre as despesas lançadas até aquela data e o limite configurado no dia que este foi definido. O valor das despesas mensais sendo maior ou menor que esse limite, irá gerar um *Push Notification* personalizado na data selecionada, um lembrete sobre os gastos.

Há um tratamento para exibir “Limite ainda não definido” caso haja despesas sem limite definido, e “Não há despesas lançadas” para quando o usuário ainda não realizou nenhum lançamento. Não é possível definir uma data no passado para essa função.



(a) Limite abaixo das despesas mensais (b) Limite salvo - Confirmação (c) Notificação recebida na data salva

Figura 24 – Tela de ajuste de limite para despesas - I

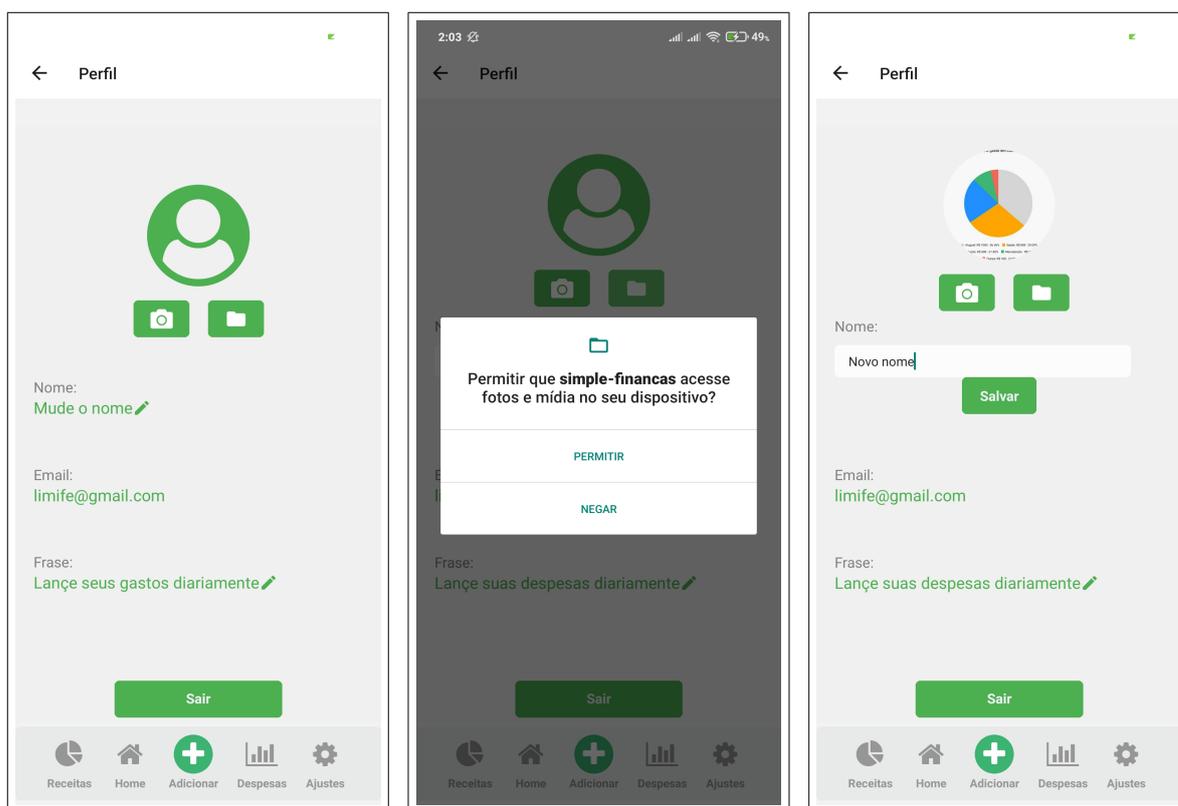


(a) Limite acima das despesas mensais (b) Notificação recebida na data salva (c) Quando não há dados para uso da função

Figura 25 – Tela de ajuste de limite para despesas - II

4.6 Editar dados de Perfil e Fazer Logout

Ao pressionar o ícone de usuário através do cabeçalho da tela principal do aplicativo, a home, este tem acesso a seus dados, onde pode trocar sua foto, nome e frase conforme mostrado nas Figuras 26 e 27. O nome e a foto serão atualizados também na tela home após a alteração. Vale observar que o usuário poderá selecionar uma foto da galeria ou tirar uma selfie. Ao selecionar uma foto e realizar logout essa foto não será perdida.

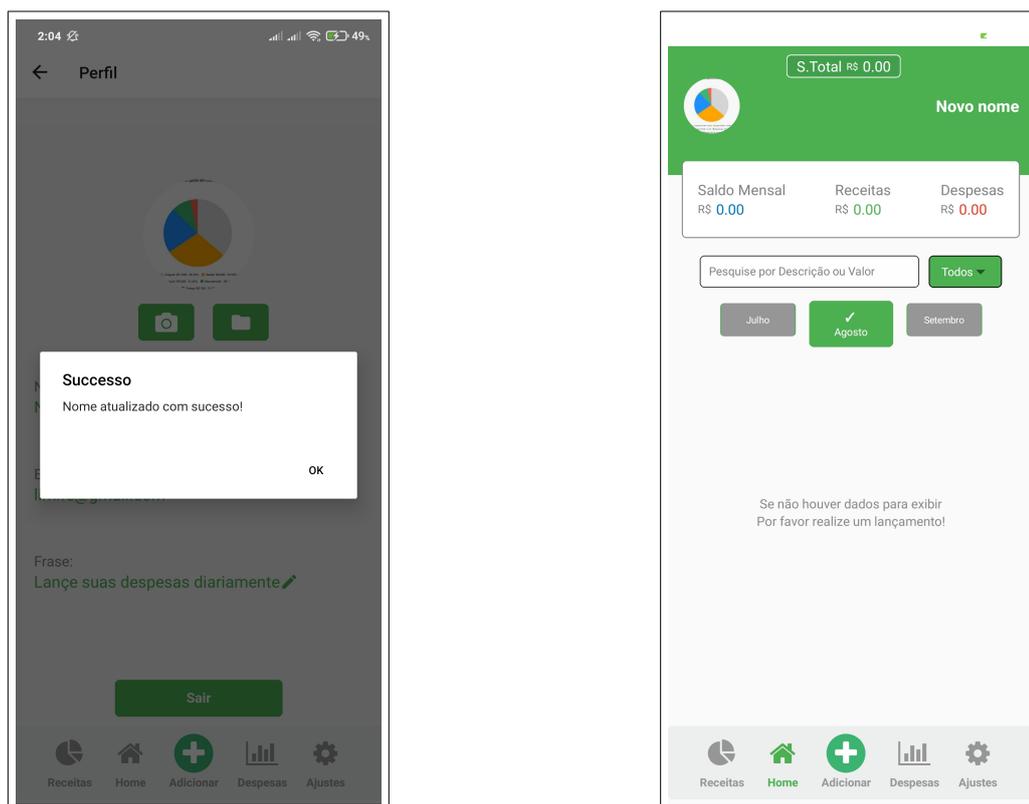


(a) Perfil com dados atuais

(b) Permissão para galeria

(c) Alterando nome e foto

Figura 26 – Tela de perfil do usuário - I



(a) Nome alterado com sucesso

(b) Nome e foto de perfil alterados na home

Figura 27 – Tela de perfil do usuário - II

4.7 Testes realizados

Afim de melhorar a experiência do usuário e verificar o funcionamento do Simple Finanças, foram realizados testes em alguns smartphones xiaomi: redmi note 8, redmi mi note10 e o xiaomi poco x3 pro, o primeiro fazendo uso do android 10, o segundo do android 14 e o terceiro do android 12, em todos os casos a performance do aplicativo ficou dentro do esperado, conferindo um aspecto de leveza em seu uso.

Com relação ao layout, foram realizados testes com a ajuda de usuários beta em aparelho Motorola, modelo Moto g52, com Android 10 e Samsung, modelo A10s, com Android 11. Após análise e correção de posição de itens e alinhamentos pode-se obter um layout satisfatório para todos os modelos verificados, tornando o aplicativo mais responsivo, além disso, sempre que possível propriedades CSS foram alteradas para percentual ao invés de pixel, o que é uma prática conhecida para auxiliar na responsividade.

Com relação as funcionalidades, foram realizados testes de lançamento de transações com datas retroativas, ou seja, data de mês passado, que é um caso clássico em que o usuário esquece de lançar alguma despesa, nesse cenário os valores mensais são exibidos corretamente na página

principal do programa, e o valor total também é ajustado de acordo. O login se mostrou persistente ao fechar e abrir o aplicativo, ou mesmo reiniciar o smartphone em que este foi instalado.

O uso do hardware se mostrou consistente entre os diferentes smartphones apresentando boa velocidade para uso das funcionalidades do sistema, mesmo em dispositivo com 2GB de memória ram e Snapdragon Quad Core.

Por fim, testes com queda de conexão de internet também foram realizados, nesses casos o sistema teve o comportamento correto e esperado, uma vez que foi desenvolvido com React Native, Expo e Firebase, considerando a capacidade de combinação intrínseca entre essas tecnologias.

5 Conclusão e Trabalhos Futuros

Com base no trabalho desenvolvido pode-se concluir que através da união das tecnologias modernas utilizadas em sua confecção foi possível desenvolver um sistema móvel eficiente para organização de finanças pessoais. Uma solução de baixo custo, em um tempo satisfatório e com equipe de desenvolvimento reduzida, o aplicativo Simple Finanças é uma nova opção que surge de forma gratuita em um ambiente de mercado popular, que é o de aplicativos Android, apresentando alta acessibilidade para o usuário brasileiro que deseja uma ferramenta para auxiliar na manutenção de sua saúde financeira.

A utilização do Firebase para o *back-end* da aplicação, em adição ao Real Time Database e módulo de autenticação tornou o desenvolvimento mais ágil e o monitoramento do aplicativo no futuro mais fácil e orientado através de suas métricas, ao ponto que se houver um grande crescimento do seu uso é viável realizar essa expansão sem grandes complicações. A performance da aplicação se mostrou satisfatória em todos os aparelhos em que foi executado, mostrando-se uma solução leve e confiável.

A principal mudança em relação ao escopo da aplicação foi trabalhar com o uso de datas mensais, uma vez que inicialmente pensou-se em trimestral e anual, porém como é um aplicativo para organização diária de finanças, a ênfase no escopo mensal da aplicação faz mais sentido e torna seu uso simplificado e direto.

Os trabalhos futuros dessa solução podem incluir:

- Aperfeiçoamento do aplicativo com relação à forma de captação de dados, por exemplo, permitir um recurso de integração de cartões e leitura de informações de transações via QRCode.
- Aperfeiçoamento e lançamento de uma versão do aplicativo para IOS, o que permitirá atingir um número maior de usuários.
- Elaboração de um guia de uso dentro do próprio aplicativo, para tornar ainda mais orientado seu uso.
- Elaboração de personalização de tema escuro.
- Inclusão de uma tela de ativos/carteira, permitindo ao usuário centralizar a informação de seus recursos financeiros guardados.

Por fim, é possível concluir que todos os objetivos mencionados nesse trabalho foram concluídos com sucesso e as sugestões de trabalho futuros incrementam seus benefícios de forma contínua.

Referências

DUECK, C.; MUCHINSKI, C. A.; OLIVEIRA, L. C. d. *Financapp: aplicação para gestão de finanças pessoais*. Dissertação (Mestrado) — Universidade Federal do Paraná, Curitiba, 2021. Disponível em: <https://acervodigital.ufpr.br/bitstream/handle/1884/73089/G_R_TCC_Financapp_aplicacao_para_gestao_de_financas_pessoais.pdf?sequence=1&isAllowed=y>. Citado na página 17.

EXAME. *Toro compra fintechs Mobills e Monetus de olho no open banking*. 2021. Disponível em: <<https://exame.com/invest/minhas-financas/toro-compra-fintechs-mobills-e-monetus-de-olho-no-open-banking/>>. Citado na página 17.

EXPO. *Configure EAS Build with eas.json*. 2022. Disponível em: <<https://docs.expo.dev/build/eas-json/>>. Citado na página 36.

EXPO. *Camera*. 2023. Disponível em: <<https://docs.expo.dev/versions/latest/sdk/camera/>>. Citado na página 19.

EXPO. *Documentação oficial*. 2023. Disponível em: <<https://docs.expo.dev/>>. Nenhuma citação no texto.

EXPO. *Notifications*. 2023. Disponível em: <<https://docs.expo.dev/versions/latest/sdk/notifications/>>. Citado na página 19.

FIGMA. *Prototipação para o Aplicativo Simple Finanças*. 2022. Disponível em: <<https://www.figma.com/proto/M2KQUQEf1R0XRIfPVYLkes/Simple-Financas-Mobile?type=design&node-id=11-9&t=7kkPQCmWrCeBiDpF-1&scaling=min-zoom&pageid=0%3A1&starting-point-node-id=11%3A9&show-proto-sidebar=1&mode=design>>. Citado na página 32.

FIREBASE, G. *Documentação do Realtime Database*. 2023. Disponível em: <<https://firebase.google.com/docs/database?hl=pt-br>>. Citado na página 33.

FIREBASE, G. *Página inicial*. 2023. Disponível em: <<https://firebase.google.com/?hl=pt>>. Citado na página 19.

IBGE, I. B. D. G. E. E. *População cresce, mas número de pessoas com menos de 30 anos cai 5,4% de 2012 a 2021*. 2021. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-denoticias/noticias/37237-de-2010-a-2022-populacao-brasileira-cresce-6-5-e-chega-a-203-1-milhoes>>. Citado na página 13.

JANONE, L.; ALPACADA, N. H. *Três em cada dez famílias estão endividadas no Brasil, aponta CNC*. CNN Brasil, 2022. Disponível em: <<https://www.cnnbrasil.com.br/economia/tres-em-cada-dez-familias-estao-endividadas-no-brasil-aponta-cnc/>>. Citado na página 13.

MOBILLS. *Sobre o Mobills, Olhar o presente, ver o futuro*. 2023. Disponível em: <<https://www.mobills.com.br/sobre/>>. Citado na página 17.

- NATIVE, R. *Getting Started*. 2023. Disponível em: <<https://reactnative.dev/docs/getting-started>>. Nenhuma citação no texto.
- NPM. *@react-native-community/datetimesticker*. 2023. Disponível em: <<https://www.npmjs.com/package/@react-native-community/datetimesticker>>. Citado na página 18.
- ORGANIZZE. *Quem somos*. 2023. Disponível em: <<https://www.organizze.com.br/quem-somos/>>. Citado na página 16.
- REACTNATIVE. *Showcase*. 2023. Disponível em: <<https://reactnative.dev/showcase>>. Citado na página 18.
- ROCKETSEAT. *Expo & React Native*. 2018. Disponível em: <<https://blog.rocketseat.com.br/expo-react-native/>>. Citado na página 19.
- SOUSA, A. F. d. *Mouros, Planejamento Financeiro Pessoal e Gestão do patrimônio*. 2. ed. São Paulo: Editora Manole, 2018. 504 p. Citado na página 16.
- STATCOUNTER. *OS Market Share Mobile Brazil*. 2022. Disponível em: <<https://gs.statcounter.com/os-market-share/mobile/brazil/#monthly-202207-202307>>. Citado na página 13.
- VARGAS, F. G. *Retrospectiva 2021: Brasil tem dois dispositivos digitais por habitante, revela pesquisa FGV*. 2021. Disponível em: <<https://portal.fgv.br/noticias/retrospectiva-2021-brasil-tem-dois-dispositivos-digitais-habitante-revela-pesquisa-fgv>>. Citado na página 13.