NÚCLEO INTERDISCIPLINAR DE PESQUISA

Curso de Sistemas de Informação Artigo Original

A IMPORTÂNCIA DE SISTEMAS DE GERENCIAMENTO DE FINANÇAS PARA A TOMADA DE DECISÕES DE USUÁRIOS DOMÉSTICOS

THE IMPORTANCE OF FINANCIAL MANAGEMENT SYSTEMS FOR HOME USERS' DECISION MAKING

Igor Gabriel Mesquita Araujo[1], **José Bento da Silva Borges**[1]
1 Aluno do Curso de Sistemas de Informação
2 Professor Especialista Curso de Sistemas de Informação

RESUMO

Atualmente, uma questão de grande relevância enfrentada pela população é a dificuldade no controle financeiro pessoal. Um número crescente de brasileiros encontra-se em situação de endividamento, o que reflete a necessidade urgente de encontrar soluções eficientes para lidar com essa problemática. O desafio reside no fato de que muitas pessoas têm dificuldade em gerenciar suas finanças de forma adequada ao longo do tempo. Tendo isto em vista, este trabalho tem como objetivo apresentar o desenvolvimento de um aplicativo de finanças pessoais, que visa auxiliar os usuários no gerenciamento eficiente de suas finanças. O aplicativo oferece recursos como controle de gastos, planejamento financeiro, categorização de despesas e receitas, além de ferramentas de análise e visualização de dados. O desenvolvimento do aplicativo foi realizado utilizando tecnologias modernas e seguindo as melhores práticas de design de interfaces e segurança da informação. O resultado final é um aplicativo intuitivo, fácil de usar e capaz de fornecer aos usuários informações valiosas sobre suas finanças pessoais, promovendo uma maior conscientização e responsabilidade financeira.

Palavras-Chave: Flutter; Dart; Firebase; Gestão de finanças pessoais;

ABSTRACT

Currently, a significant issue faced by the population is the difficulty in personal financial management. An increasing number of Brazilians find themselves in debt, reflecting the urgent need to find efficient solutions to address this problem. The challenge lies in the fact that many individuals struggle to effectively manage their finances over time. With this in mind, this study aims to present the development of a personal finance app designed to assist users in efficiently managing their finances. The app offers features such as expense tracking, financial planning, categorization of expenses and income, as well as data analysis and visualization tools. The development of the app was carried out using modern technologies and following best practices in interface design and information security. The end result is an intuitive and user-friendly app capable of providing users with valuable information about their personal finances, promoting greater awareness and financial responsibility.

Keywords: Flutter; Dart; Firebase; Personal finance management.

Contato: professor@unidesc.edu.br

1. INTRODUÇÃO

Um dos principais problemas enfrentados pela população hoje em dia, é o controle financeiro, grande parte das pessoas atualmente têm dificuldade em gerir suas próprias finanças, tendo em vista que com o decorrer dos anos o número de brasileiros endividados está cada vez maior (REZENDE, 2020).

Com o crescente avanço da tecnologia, o uso de dispositivos móveis e outros aparatos estão vivendo uma revolução tecnológica. A produção em massa de novas aplicações e ferramentas se tornou indispensável nos dias atuais, para facilitar o dia a dia dos usuários, é possível encontrar uma infinidade de aplicações para solucionar seus problemas corriqueiros, lhes trazendo mais conforto e controle (ALVES, 2017).

Aplicações de finanças pessoais vem com objetivo de trazer controle para vida de seus usuários, trazendo maior familiaridade com o assunto, deixando a questão, qual a importância de sistemas de gerenciamento de finanças para a tomada de decisões de usuários domésticos? Tendo em vista que grande parte das aplicações auxiliam os usuários no controle de suas finanças, vale a pena investigar se essas aplicações realmente influenciam o usuário, com suas futuras decisões. Tendo como objetivo o desenvolvimento de uma aplicação nova que possa auxiliar os seus usuários a desenvolver uma maior familiaridade com seus próprios gastos o ajudando a em suas tomadas de decisões futuras de maneira segura.

2. JUSTIFICATIVA

O gerenciamento de finanças vem se tornando cada vez mais necessário, tendo em vista a crise econômica que o Brasil vem passando atualmente. Índice de desemprego alto e preços exorbitantes no mercado do vestuário e alimentação vêm trazendo à tona um cenário onde o controle financeiro é algo imprescindível na vida do brasileiro. Tendo em vista que quando acontecer alguma emergência será melhor ter um dinheiro guardado para não passar por nenhuma dificuldade (CREMONEZI,2015).

O objetivo de se planejar financeiramente, é ter controle sobre seus próprios gastos, mas com isso é feito? A falta da cultura de economizar e do excessivo consumo acaba tornando difícil a organização de suas próprias finanças.

Grande parte da população vive apenas com o básico, usando todo seu rendimento para suprir suas necessidades, e por mais que a tarefa de gerir suas finanças seja uma tarefa importante não se trata de algo simples. O ato de se gerir, e se planejar financeiramente são um trabalho que depende de vários pontos distintos, alguns que às vezes nem mesmo nós podemos controlar. A principal pergunta que recai sobre nós é, como gerir nossas finanças? Pergunta essa que acompanha a humanidade desde os primórdios de sua existência. Com o tempo o homem evoluiu e sua capacidade de se organizar também, desde uma pequena anotação sobre um gasto no mercado à uma planilha elaborada sobre seus gastos mensais, a humanidade cada vez mais vem em busca de uma nova maneira de se gerir (SOUZA,2014).

A revolução tecnológica pelo qual a população tem passado trouxe uma série de ferramentas que os auxiliam no controle financeiro, oferecendo praticidade e facilidade para seus usuários, os aplicativos móveis hoje se tornaram a opção mais adequada para a atual cultura de velocidade, tendo em vista que estão sempre em seu bolso. Porém, mesmo com tantos aplicativos disponíveis no mercado, as pessoas continuam tendo dificuldade para se organizar, principalmente pela produção de novas mercadorias e criação de novas necessidades de consumo (ALVES,2017).

Apesar do recente aumento de brasileiros que realizam controle financeiro, e da vasta gama de mecanismos disponíveis no mercado, grande parte dessas pessoas alega que ainda possuem dificuldades para gerir suas finanças, afirmando que não encontram um mecanismo simples e de fácil entendimento para seu controle pessoal (SOARES,2019).

Aplicações que costumam fazer maior sucesso entre os usuários são aquelas que possuem a interface simples e de fácil entendimento, por serem adequadas a qualquer pessoal, podem ser facilmente integradas a nossa vida.

Usuários que optam por gerenciamento financeiro por meio de aplicativos, tem plena consciência sobre seus gastos e receita, por possuírem um gestor que os acompanha a qualquer lugar, se tornando cada vez mais cientes de qual a melhor hora para se gastar, investir em algo novo ou economizar seu saldo, diminuindo seu risco de se endividar ou não poder atender a algum transtorno financeiro possível (BRANDÃO,2017).

Tendo em vista o anteriormente exposto, o aplicativo desenvolvido tem sua importância justificada no fato de constituir uma ferramenta que contribui para o controle de finanças pessoais para indivíduos com pouco conhecimento sobre o tema.

3. OBJETIVO GERAL

Desenvolver um sistema mobile que auxilie usuários no controle de finanças pessoais.

3.1. Objetivos específicos

- Identificar necessidades reais de possíveis usuários.
- Reunir requisitos funcionais e não funcionais.
- Definir metodologia para desenvolvimento da aplicação.
- Construir um protótipo, com o intuito de testar a usabilidade da aplicação.
- Realizar uma busca sobre aplicações semelhantes para levantamento de novas funcionalidades / comparações

4. METODOLOGIA

Do ponto de vista de sua natureza este trabalho se enquadra como pesquisa aplicada, pois tem como objetivo gerar conhecimento para aplicações práticas, tendo em vista que visará a construção de uma ferramenta para sanar a problematização do tema anteriormente apresentado (WAZLAWICK, 2014).

Em relação aos objetivos o trabalho se caracteriza como de viés exploratório pois sua principal função é desenvolver uma maior familiaridade com o problema tratado, utilizando de técnicas como levantamento bibliográfico e análise de exemplos similares ao problema exposto (RESENDE, 2022).

Sobre os procedimentos técnicos o trabalho utilizará de pesquisas bibliográficas e experimentais, visto que além de se utilizar de materiais bibliográficos como artigos, teses, livros e entrevistas, para formular o corpo do trabalho, ele também apresenta procedimentos técnicos de caráter experimental, pois está relacionado a criação de uma ferramenta *Mobille* que manipula o aspecto de realidade do projeto, ou seja selecionar variáveis que acredita serem capazes de influenciar o objeto de estudos (WAZLAWICK, 2014).

5. FUNDAMENTAÇÃO TEÓRICA

Apesar de ser importante se controlar financeiramente, isso não costuma ser prioridade na vida dos brasileiros. A grande parte deles não possui disciplina para conter os gastos, e acabam ultrapassando seu orçamento, sem se preocupar em administrar o dinheiro adequadamente, não controlando a entrada, saída ou a forma que o dinheiro está sendo aplicado.

5.1 Educação financeira

A educação financeira tem como objetivo principal auxiliar consumidores a gerir, organizar, poupar e investir suas finanças com o intuito de direcionar a receita dos consumidores. Tendo em vista os benefícios pessoais inerentes é imprescindível hoje em dia possuir uma boa administração de suas finanças (VIEIRA; BATAGLIA; SEREIA, 2011).

As finanças são algo de extrema importância para o desenvolvimento crítico do cidadão, diante do consumismo ou da ausência de planejamento financeiro. No caso de educação financeira, desenvolver habilidades acumulando conhecimento financeiro é uma necessidade do mundo moderno. Visando o mundo moderno existem uma infinidade de aplicações que contribuem para o aprendizado dos consumidores, tendo em vista que a utilização de aplicativos para educação financeira é extremamente benéfica visto que sua prática não é isolada e pode ser levada para qualquer lugar (RAMOS; MOURA; LAVOR, 2020).

Dito isto, aplicativos que sirvam para a organização de finanças pessoais também são um método de aprendizado visto que ao praticar a organização recorrentemente os usuários acabam aprendendo na prática, lhes trazendo uma série de benefícios para sua vida financeira.

5.1.1 Benefícios ao usuário

Aplicativos de controle financeiro influenciam diretamente na vida de seus usuários, pois mesmo que indiretamente lhes ensinam educação financeira, são benefícios esses que as aplicações podem alcançar por serem implementadas como uma solução *mobile*.

Os usuários que optam por adicionar o um aplicativo de controle financeiro as suas vidas adquirem plena consistência de seus gastos e sua receita, tendo em vista que com essa adição os consumidores melhoram sua qualidade de consumo, adquirem um melhor planejamento do futuro, tendo em vista que começam a ter mais noção de de seus gastos e investimentos, obtém uma vida mais equilibrada, começam a dar menos valor a bens

materiais, pensando que se importam mais com o gasto de seu dinheiro, e maior conhecimento sobre finanças pessoais(BRANDÃO,2017).

Considerando sobre o uso de aplicações *mobile*, para educação financeira de consumidores sem experiência no assunto, sabe- se que os benefícios citados foram alcançados somente quando o usuário tinha completo entendimento das funcionalidades e interações do aplicativo, só sendo viável quando a interface do próprio era de fácil entendimento.

5.2 Aplicativos semelhantes

As lojas virtuais atualmente disponibilizam uma série de aplicações semelhantes, para gestão de finanças pessoais, tendo como base o mesmo objetivo principal: gerir os gastos, receitas e investimentos, entretanto cada um possui uma peculiaridade diferente que os torna únicos em relação aos outros. Dentre os aplicativos mais usados, vale a pena realizar uma comparação entre eles, para definir funções e avaliar sua usabilidade.

5.2.1 Mobills

Mobills é um aplicativo de controle financeiro com o objetivo de gerir e organizar seus gastos, é um aplicativo gratuito que administra todas suas finanças, disponível tanto para Android na Play Store quanto para iOS na App Store (MOBILLS, 2022).

Além do controle financeiro, o aplicativo também possui outras funcionalidades, como, o alerta de contas e, leitura automática de SMS e notificações bancárias, fazendo com que o usuário não perca nenhuma informação importante. Como vários outros também tais como a planejamento financeiro com metas, integração com contas bancárias e integração de cartão de crédito (SILVA; CANJÃO; LEAL, 2018).

5.2.2 Guiabolso

Guiabolso também é um aplicativo multiplataforma com o objetivo de gerir finanças pessoais, possibilitando o controle tanto de gastos quanto de receitas. Possibilitando outros serviços como empréstimos, cartões de crédito e débito, e investimentos (GUIABOLSO, 2022).

O aplicativo ainda conta com uma espécie incentivo ao usuário, pois disponibiliza a possibilidade de visualizar sua saúde financeira através de pontuações de acordo com seu fluxo de conta. Quanto mais os números indicam pontuações vermelhas, mais endividado o cliente está e a ideia deste projeto é incentivar o controle do usuário com suas finanças (FERREIRA et al., 2018).

5.2.3 Organizze

Outro aplicativo muito bem avaliado é o Organizze, por também ser multiplataforma estando disponível tanto na App Store quanto no *Play Store* o aplicativo dispõe de uma série de funcionalidades como, acesso a várias contas ao mesmo tempo, podendo analisar todas elas de forma completa e integrada, também recebe um demonstrativo dos lançamentos do mês, podendo organizar por categorias e subcategorias, o que permite entender exatamente o seu orçamento e não sendo necessária a conexão com internet, o tornando uma excelente ferramenta para o dia-a-dia (ORGANIZZE, 2022).

5.2.4 CoinKeeper

Apesar de não ser um aplicativo brasileiro o *CoinKeeper* possui tradução para português. Possuindo um diferencial, focando na interface para além de chamar a atenção do usuário, ser simples e prático de usar, seguindo uma rota única e extremamente visual para alcançar o controle financeiro. Dispondo de um funcionamento semelhante aos demais, seu ponto forte é o sistema de cores que dá liberdade ao usuário de estruturar uma boa organização visual (COINKEEPER, 2022).

5.3 Funcionalidades semelhantes

No Quadro 1 pode se notar a comparação entre os demais aplicativos, previamente expostos, para que seja possível notar quais são os recursos necessários e obrigatórios para que a aplicação possa ser bem sucedida.

Quadro 1: Aplicativos semelhantes x Recursos envolvidos

RECURSOS	Mobills	Guiabolso	Organizze	CoinKeeper
Gerenciamento De Cartão	✔	✔	✔	X
Interface Intuitiva	✔	✔	✔	✔
Sistema De Metas	✔	✔	✔	X
Controle de Receitas e Gastos	✔	✔	✔	✔
Disponível *Offline*	✔	✔	✔	✔
Planejamento Financeiro	✔	✔	✔	✔
Controle Familiar	X	X	X	✔

Fonte: próprio autor

Tendo em vista o levantamento de recursos de aplicações semelhantes, pode-se notar algumas funções básicas, presentes em todos os aplicativos, como o controle de receitas e gastos e o próprio planejamento financeiro.

6. RESULTADOS

6.1 Documentação do Projeto

6.1.1 Planejamentos da aplicação

Para que seja possível uma eficiente aplicação, é necessário que haja um planejamento bem feito. Existe uma série de recursos para o eficiente planejamento da aplicação, como o levantamento de requisitos, diagramas de caso de uso e prototipação.

6.1.1.1 Levantamento de requisitos

O levantamento de requisitos é de suma importância quando se constrói uma aplicação, pois se trata do início de todo o processo de desenvolvimento, pois é aqui que são definidas funcionalidades e os objetivos do sistema (MENDONÇA, 2014).

Ao reunir informações por meio de pesquisas em material bibliográfico e comparação com outras aplicações voltadas à área financeira, foi possível reunir conhecimento para o levantamento de requisitos funcionais e não funcionais.

A aplicação deve possibilitar ao usuário:

- Cadastro: possibilita a criação de uma conta pelo usuário.
- Login: possibilita que o usuário acesse a aplicação por meio de um email e senha já cadastrados.
- Cadastro e visualização de despesas: possibilita o cadastro e visualização das despesas do usuário.
- Cadastro e visualização de receitas: possibilita o cadastro e visualização das receitas do usuário.
- Cadastro e visualização de metas: possibilita o cadastro e visualização das metas do usuário.
- Visualização Relatório de Despesas: Possibilita a visualização de um relatório de despesas e receitas do usuário.

A aplicação também deve possuir:

- Interface intuitiva;
- Layout moderno;
- Usabilidade;

6.1.1.2 Diagrama de caso de uso

Os Diagramas de caso de uso tem por objetivo, apresentar a ideia geral de como o sistema vai se comportar quanto às suas funcionalidades, ou seja ele irá demonstrar de modo visual todo o caminho da aplicação do ponto de vista dos atores, que representam usuário, outros sistemas ou dispositivos (LINS, 2020).

Na figura 1 pode se notar o diagrama de caso de uso responsável pela aplicação a ser tratada no artigo, onde é possível notar todos os casos do aplicativo sendo eles, fazer cadastro, fazer login, inserir dados, consultar relatórios e inserir metas.

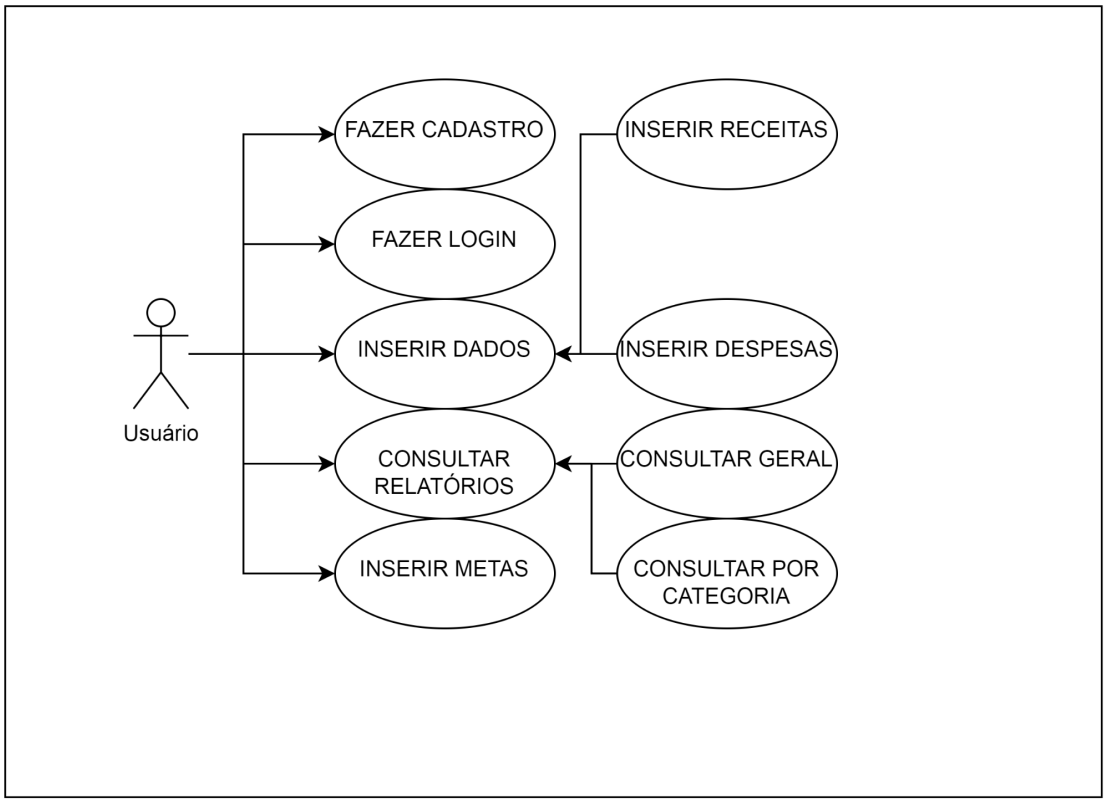


Figura 1: Diagrama de casos de uso

Fonte: próprio autor

Apresentado no Quadro 2 o descrição da ação de fazer cadastro, onde o usuário necessita apenas de um e-mail válido como pré-requisito para que seja possível realizá-lo.

Quadro 2: UC01: Fazer Cadastro

Ator	Usuário do aplicativo
Fluxo Principal	1) Acessar aplicativo; 2) Acessar página de cadastro; 3) Preencher campos obrigatórios (Nome, Email e Senha).
Fluxo Alternativo	1) Deixar de preencher algum campo.
Precondições	1) Possuir e-mail válido.
Pós-Condições	1) O usuário agora poderá acessar o aplicativo.

Fonte: próprio autor

No Quadro 3 está sendo apresentada a ação de login na aplicação. Para realizar o login, o usuário deve possuir um cadastro. Após realizar o login, o usuário poderá acessar a tela principal do aplicativo.

Quadro 3: UC02: Fazer Login

Ator	Usuário do aplicativo
Fluxo Principal	1) Acessar aplicativo; 2) Acessar página de login; 3) Preencher campos obrigatórios (Email e Senha).
Fluxo Alternativo	1) Deixar de preencher algum campo.
Precondições	1) Possuir cadastro no aplicativo.
Pós-Condições	1) O usuário agora estará logado no aplicativo.

Fonte: próprio autor

Está sendo apresentado no Quadro 4 o processo de inserir dados no aplicativo, onde o único pré-requisito para o ato é o usuário estar logado na aplicação, após isso o usuário pode cadastrar suas receitas ou despesas no aplicativo.

Quadro 4: UC03: Inserir Dados

Ator	Usuário do aplicativo
Fluxo Principal	1) Acessar aplicativo; 2) Acessar página de cadastro de dados; 3) Escolher qual tipo de dado (Receita ou Despesa); 4) Preencher campos obrigatórios (Valor, Categoria, descrição e data).
Fluxo Alternativo	1) Deixar de preencher algum campo.
Precondições	1) Estar logado no aplicativo e possuir registros cadastrados.
Pós-Condições	1) O dado será cadastrado.

Fonte: próprio autor

No Quadro 5 está sendo informado o processo de consulta de registro, onde será possível observar suas despesas e receitas, tanto de maneira geral com o valor bruto

quanto por categorias, onde será dividido o valor da despesa para suas respectivas categorias. Os únicos pré-requisitos são, o usuário estar logado na aplicação e possuir algum registro previamente cadastrado.

Quadro 5: UC04: Consultar Relatórios

Ator	Usuário do aplicativo
Fluxo Principal	1) Acessar aplicativo; 2) Acessar página de relatórios; 3) Escolher entre relatório geral ou específico; 4) Selecionar categoria caso deseje consultar o relatório específico.
Fluxo Alternativo	
Precondições	1) Estar logado no aplicativo.
Pós-Condições	1) Usuário poderá consultar seus registros

Fonte: próprio autor

Está presente no Quadro 6, a descrição de como são inseridas metas na aplicação, sendo apresentado como uma espécie de limite para o controle de gastos do usuário. Possuindo apenas um pré-requisito, estar logado no aplicativo, para que seja possível a inclusão de novas metas.

Quadro 6: UC05: Inserir Metas

Ator	Usuário do aplicativo
Fluxo Principal	5) Acessar aplicativo; 6) Acessar página de Metas; 7) Preencher os campos necessários para adicionar uma meta nova (Categoria e limite).
Fluxo Alternativo	2) Deixar de preencher algum campo.
Precondições	2) Estar logado no aplicativo.
Pós-Condições	2) Usuário poderá incluir uma meta nova.

Fonte: próprio autor

6.1.2 Interface da aplicação

Pouco tempo atrás os celulares deixaram de serem simplesmente aparelhos para comunicação e passaram a ser ferramentas indispensáveis no dia-a-dia, tendo em vista que te acompanham por todo lado. Como uma série de aplicações, com diversas utilidades, a sua interface é um dos pontos mais importantes, pois está atrelada diretamente à experiência do usuário, que por sua vez se relaciona com a permanência de seu consumidor no aplicativo.

Usabilidade é um conjunto de técnicas, recursos visuais e programação que tornam o site intuitivo, ou seja, que as pessoas entendem facilmente como ele funciona.A Interface do usuário é onde as interações entre o usuário e o aplicativo acontecem. Seu principal objetivo é promover um sistema de fácil uso para os usuários. Quando o aplicativo tem uma boa UI, o usuário tem menos dificuldades ao operar o aplicativo e, consequentemente, sua experiência é otimizada e esses fatores são essenciais para a permanência de um usuário em um aplicativo (NETO, 2013).

Tendo em vista as necessidades e emoções do usuário enquanto eles utilizam a aplicação, sua satisfação com a integração com ele contribuirá para o sucesso ou fracasso do aplicativo. Ao se implementar um aplicativo, deve-se levar em consideração uma série de fatores, como por exemplo o público alvo ao qual ele está direcionado, aplicativos possuem várias finalidades e funcionalidades o ideal é planejar sua produção e encontrar o que melhor se adequa a aplicação (VALENTIM; SILVA; CONTE, 2015).

A para planejar a interface de sua aplicação o melhor a se fazer é criar um protótipo do próprio, onde poderá descobrir o que vai dar certo e o que pode levar ao fracasso da sua interface, acabando assim com a experiência do usuário.

6.1.2.1 Figma

Sabe-se que o design é um importante elemento em qualquer aplicação tendo em vista que ele irá interagir diretamente com o usuário do sistema, pensando nisso é necessário que haja um planejamento sobre, suas utilidades e requisitos, pois serão eles que irão moldar a interface do aplicativo.

Uma ferramenta muito utilizada na prototipagem de aplicações hoje em dia é o *Figma,* ferramenta que nos apresenta um ambiente favorável e de interface simples e bem intuitiva que facilita a criação de novos projetos, por ser implementado em nuvem, permite também colaborações entre outros usuários, tornando os protótipos nele criados ainda mais elaborados (NASCIMENTO et al., 2020).

A tabulação de requisitos é uma das principais vantagens trazidas pelo *Figma* pois com isso é possível obter informações de como cada funcionalidade pode se encaixar na tela do usuário, pensando nisso é indispensável a criação de um protótipo que possa tanto prender a atenção do usuário como ser de fácil uso cotidiano, sabendo que o controle de finanças fará parte de sua rotina(LEPORE, 2021).

6.1.2.2 Prototipação

Do ponto de vista do protótipo em si, a aplicação se iniciará na tela de cadastro do aplicativo, onde o usuário poderá cadastrar suas informações para que seja possível efetuar o login e por sua vez acessar a aplicação.

Na Figura 2 estão presentes tanto a tela de cadastro quanto a de login onde serão apresentados todos os campos necessários para que o usuário possa efetuar o cadastro de suas informações, tais como nome completo, email e senha, para que torne possível o login do usuário.

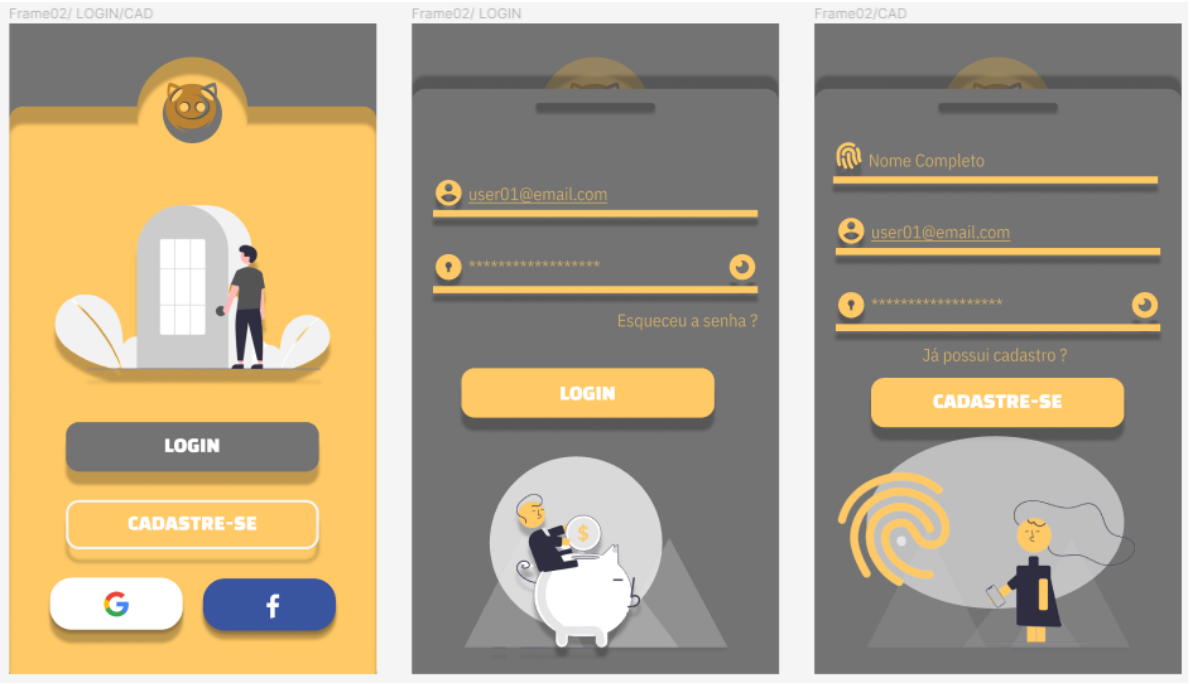


Figura 2: Protótipo das telas de Login e de Cadastro

Fonte: próprio autor

Seguindo o fluxo lógico o próximo passo da aplicação será o acesso da tela principal onde estarão disponíveis as informações iniciais do aplicativo, sendo possível inserir novos dados e consultar relatórios simples sobre as finanças do usuário.

Na figura 3 está disponível a tela principal e a de adição de dados onde será possível consultar suas informações, como receita, despesas e saldo, assim como pequenos gráficos sobre suas finanças. A tela de adição de dados será onde o usuário poderá cadastrar novas receitas ou despesas.

Figura 3: Protótipo das telas de Principal e de Adição

Fonte: próprio autor

A aplicação também disponibilizará uma tela onde será possível acessar um histórico de transações efetuadas pelo usuário onde será possível identificar tanto receitas quanto despesas, e suas respectivas categorias, como apresentado na figura 4.

Figura 4: Protótipo da tela de Transações

Fonte: próprio autor

Também deverá ser possível para o usuário acessar um relatório completo de suas finanças relacionando seus gastos, ganhos e economias, seja mensalmente para obter um relatório mais detalhado ou anualmente para obter um apanhado geral de suas finanças, como pode ser notado na Figura 5.

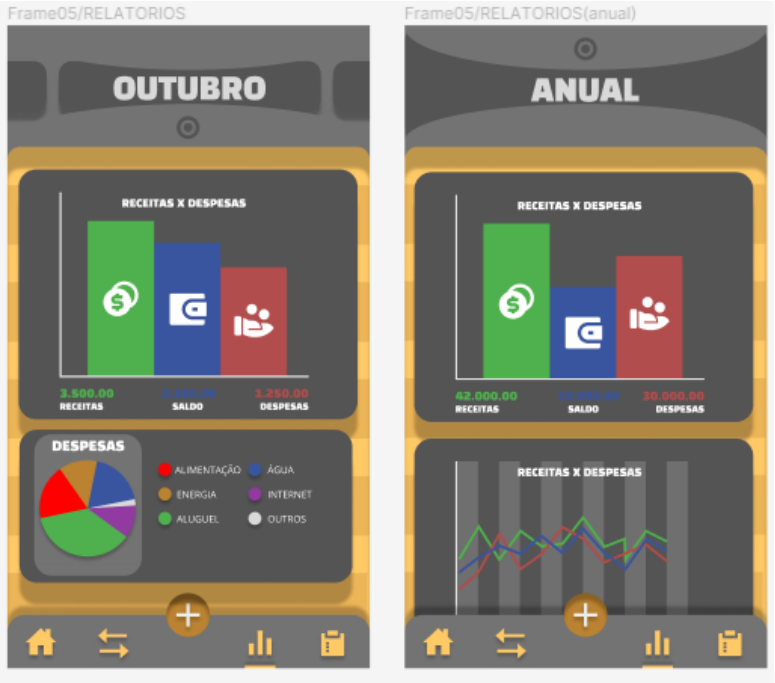


Figura 5: Protótipo das telas de Relatório Mensal e Anual

Fonte: próprio autor

A última função do aplicativo é a inserção de metas que funcionará como uma espécie de limite que norteará as despesas do usuário para indicar quando ele estiver gastando demais em determinada categoria.

Na Figura 6 está disponível a as telas de Metas e Adição de metas onde deverá ser possível para o usuário tanto consultar suas metas e verificar se suas finanças correspondem ao limite, quanto adicionar novas metas.

Figura 6: Protótipo das telas de Metas e Adição de metas

Fonte: próprio autor

6.1.3 Ferramentas

Ao se pensar na criação de um aplicativo primeiro é importante avaliar qual metodologia irá se encaixar adequadamente no projeto. Uma ferramenta muito implementada atualmente para o desenvolvimento de aplicativos é o *framework* Flutter, muito utilizado, pois é de fácil implementação e aprendizado. Outro ponto importante é a redução de custos que a *framework* trás, devido à melhor produtividade da equipe, que pode entregar muito mais em menos tempo. Como se trata de um *framework* cruzado, os desenvolvedores utilizam a mesma base de códigos para ambos os sistemas e não precisam fazer diferentes versões das aplicações, sendo alguns dos principais motivos para a integração das seguintes tecnologias na aplicação.

6.1.3.1 Flutter

Flutter é uma estrutura de código aberto do Google para criar aplicativos multiplataforma bonitos e compilados nativamente a partir de uma única base de código. Utilizando a linguagem Dart, os códigos são compilados diretamente ao ARM nativo e podem acessar API 's de plataformas e serviços (FLUTTER, 2022).

Por ser multiplataforma o Flutter traz consigo outro ponto benéfico, pois sua arquitetura além utilizar a Linguagem Dart para suas implementações também usa outras linguagens, como Kotlin e Swift, para Android e iOS respectivamente (REZENDE, 2020).

O Flutter possui uma estrutura construída por *widgets,* que são as ferramentas que constroem a interface da aplicação, possuindo inspiração direta no React*,* cada *widget* é uma declaração imutável de parte da interface do usuário eles formam uma hierarquia com base na composição fazendo com que cada *widget* se aninha dentro de seu pai e podendo receber contexto do pai. Além disso, os *widgets* podem ser definidos tanto como estáticos, onde todas as suas configurações são feitas durante a inicialização, quanto dinâmico, onde além da configuração inicial, também possuem um *State*, que define seu estado atual (CRUZ, 2022).**.**

6.1.3.2 Linguagem de programação Dart

O Dart foi criado pelo Google com a intenção de substituir o Javascript e ser a mais utilizada pelos navegadores, Dart é uma linguagem de desenvolvimento otimizada para multiplataforma, tendo como objetivo oferecer uma linguagem produtiva, e com tempo de execução flexível que possa combinar com frameworks de desenvolvimento de aplicativos (DART, 2022).

O Dart também possui diferentes plataformas e disponibiliza duas distintas soluções, como o Dart Native*,* utilizado para o desenvolvimento *mobile*, *desktop* e de servidor e o Dart Web*,* possui seu foco voltado para o desenvolvimento de aplicações web (CRUZ, 2022).

6.1.3.3 Firebase

Outro recurso muito utilizado, na criação de aplicativos, é a implementação da ferramenta Firebase*.* O Firebase é uma plataforma de desenvolvimento de aplicativos que ajuda você a criar e desenvolver aplicativos e jogos, apoiado pelo Google e confiável por milhões de empresas em todo o mundo (FIREBASE, 2022).

Seu principal objetivo é melhorar o desempenho de apps com a junção de várias funcionalidades numa plataforma só. Tais funcionalidades contribuem para o

desenvolvimento e para a monetização de apps, já que é possível utilizá-lo também no Marketing Digital. Por também dispor de um serviço BaaS (*Backend-as-a-Service*), acaba diminuindo muito o tempo de desenvolvimento da aplicação, reduzindo o tempo gasto com desenvolvimento de um backend próprio em sua hospedagem (SOARES, 2019).

No aplicativo desenvolvido, o Firebase é utilizado para realizar cadastramento e autenticação de usuários, recuperação de senha e realizando o armazenamento de dados tais como valor, descrição, data e tipo das movimentações realizadas pelo usuário.

6.1.3.3.1 Firebase Authentication

O aplicativo utiliza o Firebase Authentication para realizar o cadastro, autenticação e recuperação de senha dos usuários dando a possibilidade de autenticação de usuários por vários meio dos serviços do próprio Firebase, através do login com endereço de e-mail e senha. O SDK do Firebase Authentication fornece métodos que permitem aos usuários fazer login com as Contas do Google, Facebook e outras maneiras. Entretanto, a aplicação utilizará apenas a autenticação de cadastro por e-mail e senha, tal qual o login por meio das Contas do Google e Facebook.

6.1.3.3.2 Firebase Cloud Firestore

O Cloud Firestore é um banco de dados não relacional hospedado em nuvem que disponibiliza estruturas de dados hierárquicos flexíveis. O controle dos dados se baseia em documentos, organizados em coleções, que funcionam como se fossem as tabelas presentes em um banco de dados relacional. Os documentos podem conter objetos aninhados complexos, além de subcoleções.

A imagem abaixo representa um exemplo da interface presente dentro da plataforma do Firebase, onde é possível consultar todos os dados inseridos no sistema.

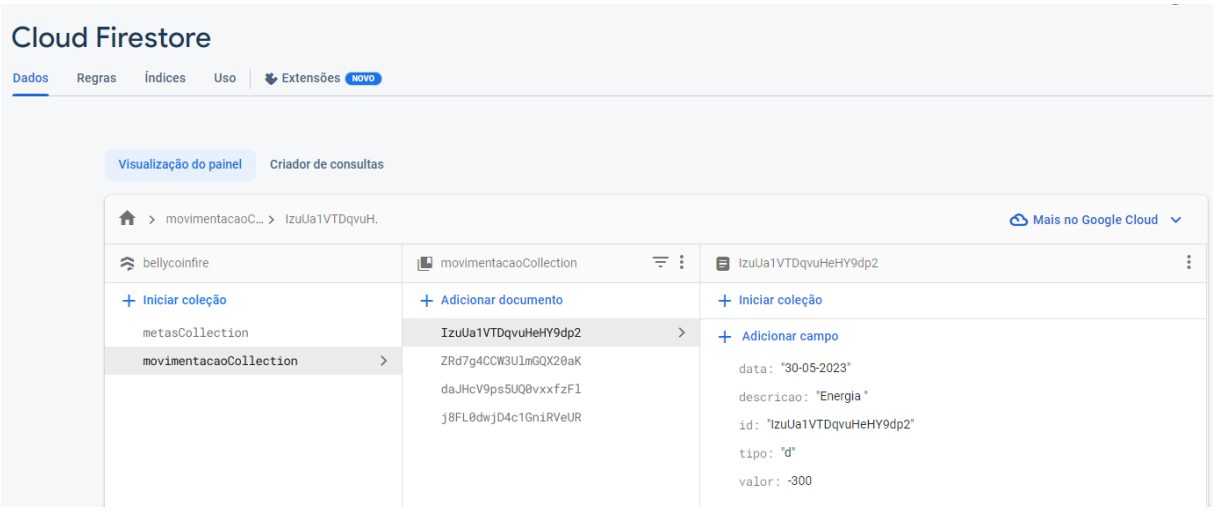


Figura 7: Tela da plataforma de Cloud Firestore

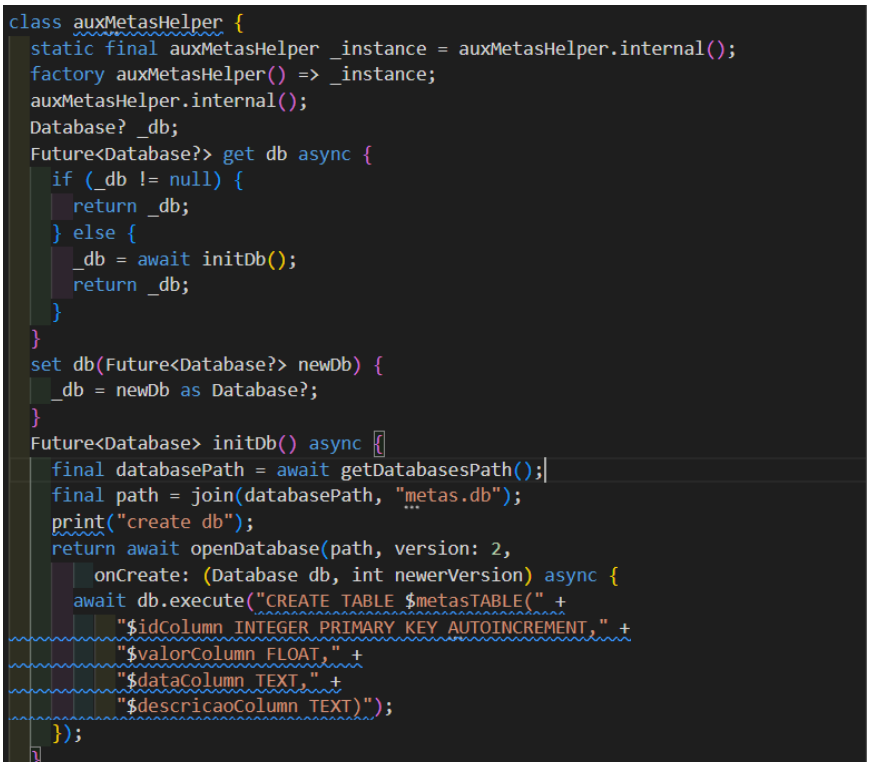
Fonte: próprio autor

6.1.3.4 SQLite

Além do serviço presente do Cloud Firestore, a aplicação utiliza um banco de dados local, pois é interessante que o usuário não seja tão dependente de conexão com a internet. Por este motivo foi implementada uma solução com o banco de dados SQLite, que funcionará como uma auxiliar secundário para realizar transações e tratar dados enquanto a aplicação não estiver conectada à internet.

A implementação é feita por meio do Sqflite uma biblioteca presente no próprio Flutter que permite ao usuário a utilização do banco de dados SQLite, o Sqflite disponibiliza todos os recursos de um banco de dados SQL comum, permitindo maior flexibilidade na construção de consultas, tabelas, triggers, procedures, dentre outros recursos (ALVES,2022).

A seguir segue um exemplo de como é realizada a criação de e utilização do Sqflite na aplicação.



```
class auxMetasHelper {
  static final auxMetasHelper _instance = auxMetasHelper.internal();
  factory auxMetasHelper() => _instance;
  auxMetasHelper.internal();
  Database? _db;
  Future<Database?> get db async {
    if (_db != null) {
      return _db;
    } else {
      _db = await initDb();
      return _db;
    }
  }
  set db(Future<Database?> newDb) {
    _db = newDb as Database?;
  }
  Future<Database> initDb() async {
    final databasePath = await getDatabasesPath();
    final path = join(databasePath, "metas.db");
    print("create db");
    return await openDatabase(path, version: 2,
        onCreate: (Database db, int newerVersion) async {
      await db.execute("CREATE TABLE $metasTABLE(" +
          "$idColumn INTEGER PRIMARY KEY AUTOINCREMENT," +
          "$valorColumn FLOAT," +
          "$dataColumn TEXT," +
          "$descricaoColumn TEXT)");
    });
  }
```

Figura 8: Criação do banco de dados SQLite

Fonte: próprio autor

6.1.3.3.3 Dicionário de Dados

Adiante será apresentado o dicionário de dados presente na aplicação informações que serão salvas no serviço de dados do Firebase o Cloud Firestore assim como no banco auxiliar do SQLite presente na aplicação.

O quadro a seguir apresenta o dicionário de dados referente a coleção de Movimentações do Cloud Firestore, onde são indicados todas as especificações referentes aos dados de Movimentações.

Quadro 7: DD 001: Dicionário de dados de Movimentações

Identificador	Descrição			
DD 001	**Dados de Movimentações**			
Identificador	**Nome da Variável**	**Tipo**	**Descrição**	**Valor**
ID da Movimentação	ID	String	Armazena o ID das Movimentações	
Data da Movimentação	Data	String	Armazena a Data das Movimentações	DD-MM-YYYY
Valor da Movimentação	Valor	Double	Armazena o Valor das Movimentações	000001-999999
Tipo da Movimentação	Tipo	String	Armazena o Tipo das Movimentações	r-d
Descrição da Movimentação	Descrição	String	Armazena a Descrição das Movimentações	

Fonte: próprio autor

A seguir estão dispostas as informações referentes ao dicionário de dados da coleção de Metas do Cloud Firestore, onde são indicadas todas as especificações referentes aos dados de Metas.

Quadro 8: DD 002: Dicionário de dados de Metas

Identificador	Descrição			
DD 002	Dados de Metas			
Identificador	**Nome da Variável**	**Tipo**	**Descrição**	**Valor**
ID de Metas	ID	String	Armazena o ID das Metas	
Data de Metas	Data	String	Armazena a Data das Metas	DD-MM-YYYY
Valor de Metas	Valor	Double	Armazena o Valor das Metas	000001-999999
Descrição de Metas	Descrição	String	Armazena a Descrição das Metas	

Fonte: próprio autor

6.2 Implementações

A seguir serão apresentadas às telas da aplicação em sua primeira versão, mostrando todas as características e utilidades da ferramenta especificando o método de construção.

6.2.1 Tela Inicial e OnBoarding Page

O fluxo da aplicação se inicia com a tela de carregamento do aplicativo que nos apresenta a logomarca do mesmo, possuindo um *CircularProgressIndicator,* que funciona como uma espécie de temporizador para que a aplicação seja carregada.

Em seguida, após o carregamento da tela a aplicação nos direciona para a *onBordingPage*, página que nos apresenta a aplicação, possuindo, também, a logo do aplicativo e um *Button,* que nos direciona para a página seguinte.

Relembrando que dentro do Flutter existem dois tipos principais de de *Widgets* que são os construtores da interface da aplicação, a onBoarding Page é um exemplo de uma tela que foi construída por meio do método estático, pois não muda o seu estado durante sua execução.

Na imagem abaixo está um exemplo de como é a realizada a construção de uma tela por meio do *StatelessWidget,* o método estático.

```dart
import 'package:bellycoin_app/common/app_text_styles.dart';
import 'package:flutter/material.dart';
import '../../common/routes.dart';
final loading = ValueNotifier<bool>(false);
class OnbordingPage extends StatelessWidget {
  final bool isLoggedIn;
  const OnbordingPage({Key? key, required this.isLoggedIn}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFFFFC966),
      body: Column(
        children: [
          Expanded(
            flex: 4,
            child: Container(
              decoration: new BoxDecoration(
                  color: Color(0xFF737373),
                  borderRadius: BorderRadius.only(
                    bottomLeft: const Radius.circular(40.0),
                    bottomRight: const Radius.circular(40.0),
                  ), // BorderRadius.only
                  boxShadow: [
                    BoxShadow(
                      color: Color(0xFF303030),
                      blurRadius: 25.0,
                      spreadRadius: 8.0,
                      offset: Offset(2.0, 5.0)), // BoxShadow
                  ]), // BoxDecoration
```

Figura 9: Criação de um StatelessWidget

Fonte: próprio autor

Na figura a seguir está sendo apresentada a interface da tela de boas vindas.

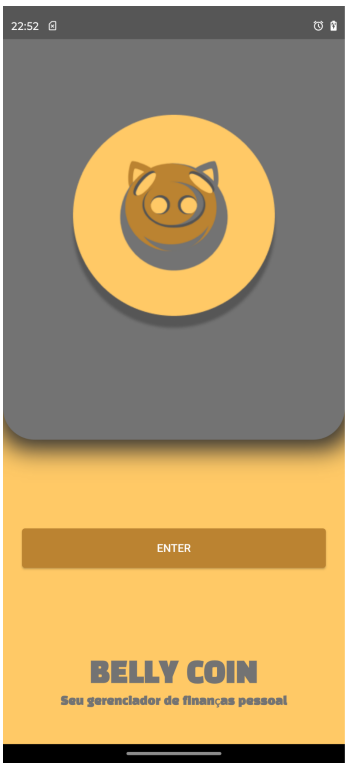


Figura 10: Tela de Boas Vindas

Fonte: próprio autor

6.2.2 Telas de Login e Cadastro de Usuário

A tela login é onde o usuário poderá se cadastrar ou logar caso já possua uma conta, dando-lhe a possibilidade de escolher entre três maneiras diferentes de login, sendo o da própria plataforma, o login realizado pela conta do Google ou o pela conta do Facebook do usuário.

A tela foi construída pelo *StatefulWidget*, pois utiliza de um *showModalBottomSheet,* que serve como uma espécie de aba, onde quando o usuário toca o botão desejado ele é direcionado para uma tela de login ou de cadastro que lhe permite realizar a autenticação do seu email.

A seguir são apresentadas as telas de login e cadastro do usuário, que são compostas pelos seguintes componentes:

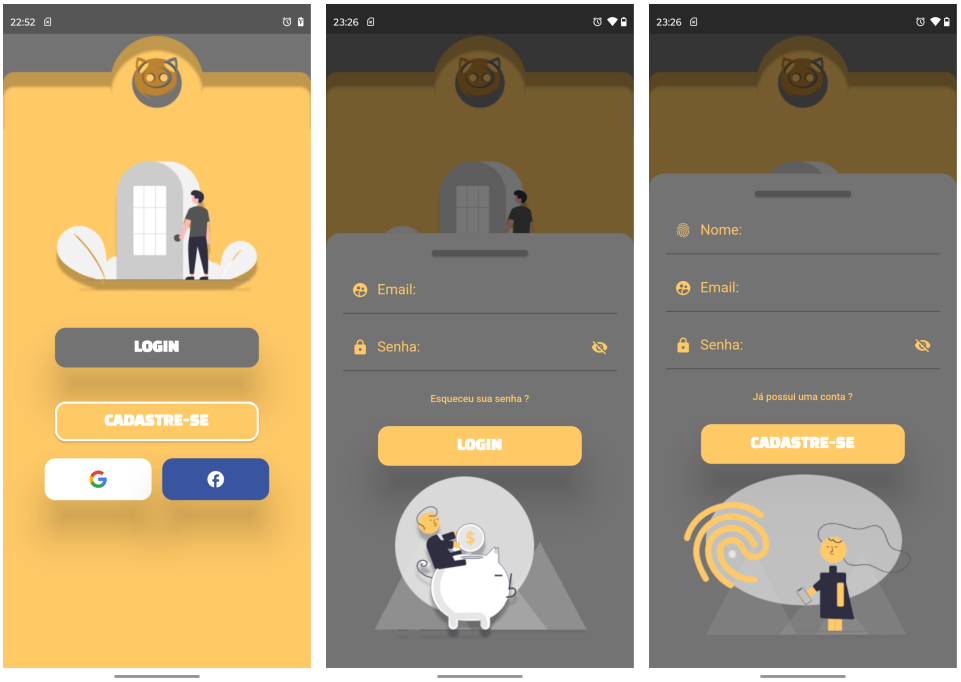


Figura 11: Telas de Cadastro e Login

Fonte: próprio autor

A seguir são apresentados os componentes presentes nas telas de login e cadastro:

- E-mail: Campo que o usuário informa um e-mail já cadastrado;
- Senha: Campo que o usuário informa a senha correspondente ao e-mail cadastrado;

- Login: Botão que ao ser pressionado irá realizar a validação dos campos de e-mail e senha. Caso sejam validados, o usuário será redirecionado para a Tela Principal; caso contrário, será exibida uma mensagem contendo o erro;
- Esqueci minha senha: Botão que ao ser pressionado realiza a validação do campo de e-mail e envia o mesmo o procedimento de recuperação de senha;
- Nome: Campo que o usuário informa um nome, presente apenas quando é realizado o cadastro do usuário.

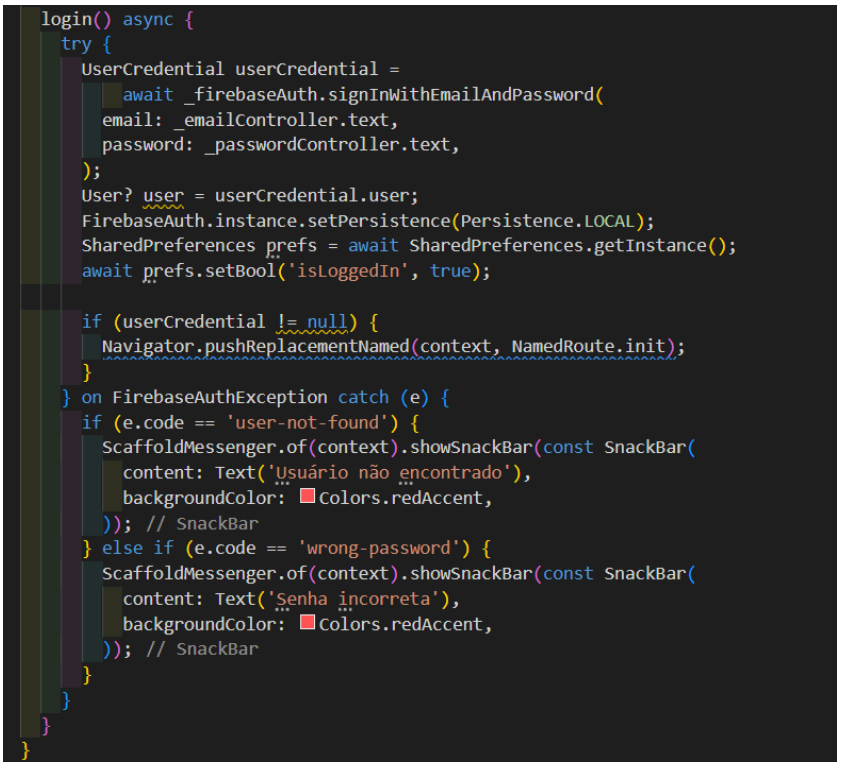
Além disso é nessas telas que é realizada a autenticação do usuário por meio do Firebase Authentication, como pode ser notado no exemplo abaixo.

```
cadastrar() async {
  try {
    UserCredential userCredential =
        await _firebaseAuth.createUserWithEmailAndPassword(
            email: _emailController.text, password: _passwordController.text);
    if (userCredential != null) {
      userCredential.user!.updateDisplayName(_nameController.text);
      CustomCircularProgressIndicator;
      Navigator.pushReplacementNamed(context, NamedRoute.init);
    }
  } on FirebaseAuthException catch (e) {
    if (e.code == 'weak-password') {
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
        content: Text('Crie uma senha mais forte'),
        backgroundColor: Colors.redAccent,
      )); // SnackBar
    } else if (e.code == 'email-already-in-use') {
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
        content: Text('Este email já foi cadastrado'),
        backgroundColor: Colors.redAccent,
      )); // SnackBar
    }
  }
}
```

Figura 12: Cadastro de Usuários

Fonte: próprio autor

A seguir está presente como é realizada a autenticação dos usuários cadastrados na plataforma.



```
login() async {
  try {
    UserCredential userCredential =
        await _firebaseAuth.signInWithEmailAndPassword(
      email: _emailController.text,
      password: _passwordController.text,
    );
    User? user = userCredential.user;
    FirebaseAuth.instance.setPersistence(Persistence.LOCAL);
    SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.setBool('isLoggedIn', true);

    if (userCredential != null) {
      Navigator.pushReplacementNamed(context, NamedRoute.init);
    }
  } on FirebaseAuthException catch (e) {
    if (e.code == 'user-not-found') {
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
        content: Text('Usuário não encontrado'),
        backgroundColor: Colors.redAccent,
      )); // SnackBar
    } else if (e.code == 'wrong-password') {
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
        content: Text('Senha incorreta'),
        backgroundColor: Colors.redAccent,
      )); // SnackBar
    }
  }
}
```

Figura 13: Cadastro de Usuários

Fonte: próprio autor

A tela de Login também pode ser usada como tela de recuperação de senha, e permite ao usuário iniciar o processo de redefinição de senha. O usuário deve informar seu e-mail no campo de e-mail no Login e pressionar o botão “Esqueci minha senha”, e será encaminhado uma mensagem para o email do usuário onde será possível redefinir a senha.

A seguir segue uma imagem da tela de redefinição de senha e o e-mail encaminhado para o usuário.

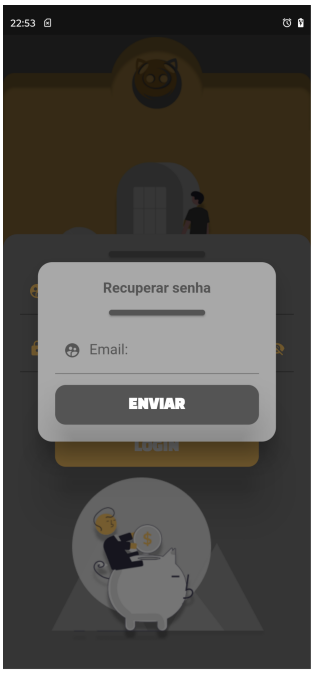


Figura 14: Tela de redefinição de senha

Fonte: próprio autor

E abaixo é apresentado o email recebido pelo usuário para realização do cadastro de uma nova senha.

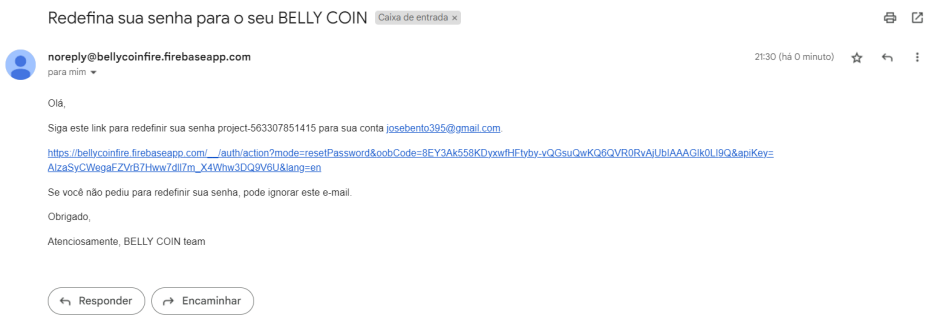


Figura 15: E Mail de redefinição de senha

Fonte: próprio autor

6.2.3 Tela Home

A próxima figura apresenta a tela Home da aplicação. Nesta tela pode se notar a presença de três *labels* principais que representam respectivamente os valores de saldo, receitas e despesas do usuário, além de histórico de transações realizadas durante o mês e a disponibilidade de navegar entre os meses que ocorreram algum tipo de transação.

Além disso, a página também permite que o usuário acesse um histórico específico com apenas receitas ou descrições ao tocar no respectivo ícone.

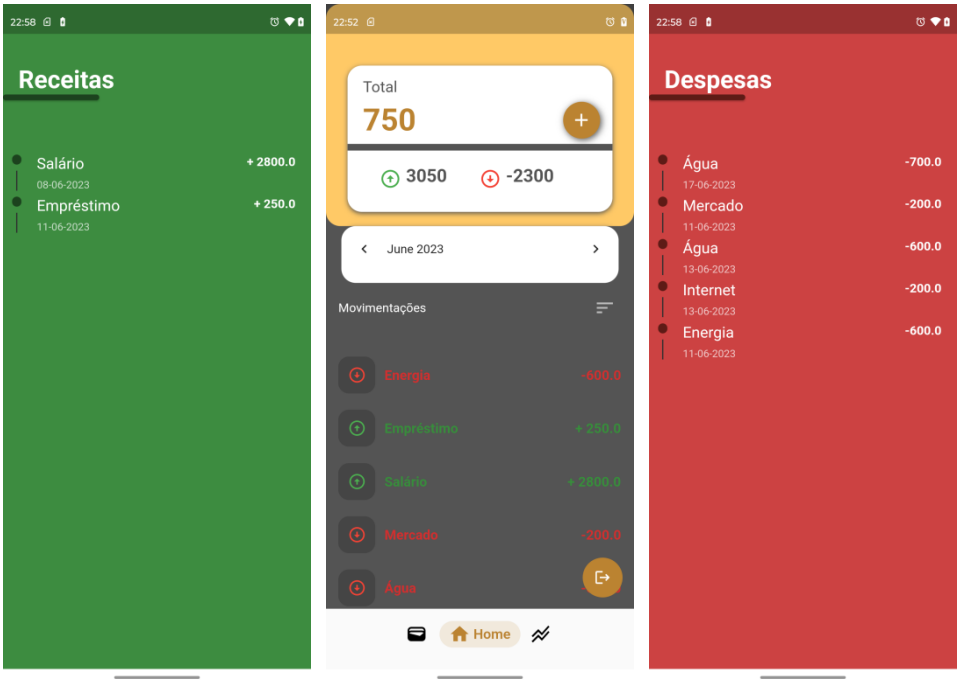


Figura 16: Tela Principal e de Histórico

Fonte: próprio autor

A tela é construída por meio de um *StatefulWidget,* como pode ser notado na figura a seguir, esse método de construção para essa tela deve-se ao fato dela possuir diferentes estados durante sua execução tornando possível a atualização dos valores expostos na tela. A tela é dividida em dois principais Containers, que a separa entre os valores totais e o histórico de transações.

```dart
import '../../../common/models/widgets/CardMovimentacoesItem.dart';
import '../../../common/models/widgets/CustomDialog.dart';
import '../../../common/routes.dart';
import '../../../repositories/Transactions.dart';
import '../../../repositories/auxTransactionsSQL.dart';

class HomeSc extends StatefulWidget {
  @override
  _HomeScState createState() => _HomeScState();
  const HomeSc({super.key});
}
```

Figura 17: Criação de um StatefulWidget

Fonte: próprio autor

Além disso, a tela aplicação também permite que usuário possa inserir ou editar novos dados, por meio de um *Button,* presente no canto superior da tela.

A seguir são apresentados como os dados são inseridos na aplicação.

Figura 18: Tela de adição de dados

Fonte: próprio autor

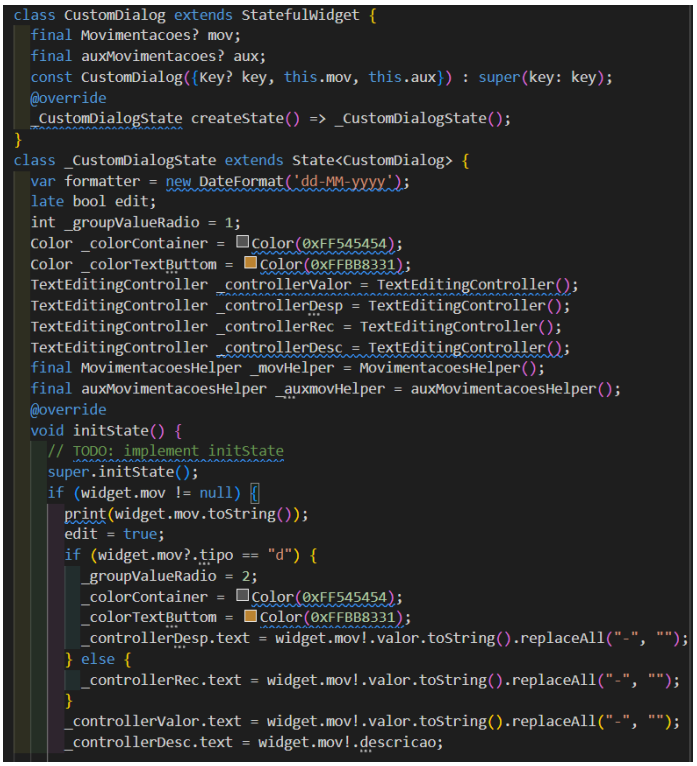
A seguir o são apresentados os componentes que estão presentes no no widget de inserir e alterar transações:

- Valor: Campo que o usuário informa um valor de receitas ou despesas;
- Tipo: *Radio* que o usuário informa se o valor é uma despesa ou receita;
- Descrição: Campo que o usuário informa o que foi a transação realizada;
- Confirmar/Editar: Botão que ao ser pressionado realiza confirma ou edita transação realizada;
- Cancelar: Botão que ao ser pressionado cancela a transação a ser realizada;

O Flutter possui uma função que permite que o desenvolvedor crie seus próprios widgets e implemente na tela em uma espécie de função, que o permite customizar e implementar uma lógica nova e única para a função.

O Button presente na tela home é um exemplo dessa prática, ele foi um *widget* desenvolvido por meio de um *StatefulWidget,* pois ele altera o estado da aplicação ao inserir novos dados ou alterá-los.

A imagem a seguir apresenta um exemplo de como é realizada a construção do *widget.*

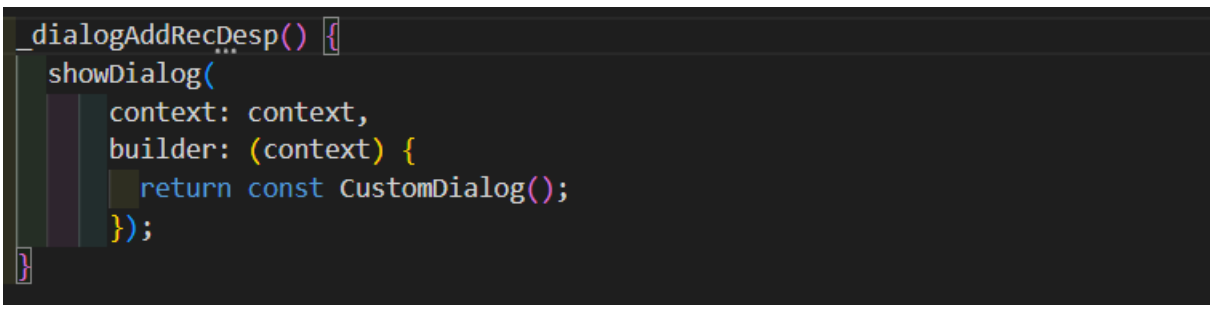


```
class CustomDialog extends StatefulWidget {
  final Movimentacoes? mov;
  final auxMovimentacoes? aux;
  const CustomDialog({Key? key, this.mov, this.aux}) : super(key: key);
  @override
  _CustomDialogState createState() => _CustomDialogState();
}
class _CustomDialogState extends State<CustomDialog> {
  var formatter = new DateFormat('dd-MM-yyyy');
  late bool edit;
  int _groupValueRadio = 1;
  Color _colorContainer = Color(0xFF545454);
  Color _colorTextButtom = Color(0xFFBB8331);
  TextEditingController _controllerValor = TextEditingController();
  TextEditingController _controllerDesp = TextEditingController();
  TextEditingController _controllerRec = TextEditingController();
  TextEditingController _controllerDesc = TextEditingController();
  final MovimentacoesHelper _movHelper = MovimentacoesHelper();
  final auxMovimentacoesHelper _auxmovHelper = auxMovimentacoesHelper();
  @override
  void initState() {
    // TODO: implement initState
    super.initState();
    if (widget.mov != null) {
      print(widget.mov.toString());
      edit = true;
      if (widget.mov?.tipo == "d") {
        _groupValueRadio = 2;
        _colorContainer = Color(0xFF545454);
        _colorTextButtom = Color(0xFFBB8331);
        _controllerDesp.text = widget.mov!.valor.toString().replaceAll("-", "");
      } else {
        _controllerRec.text = widget.mov!.valor.toString().replaceAll("-", "");
      }
      _controllerValor.text = widget.mov!.valor.toString().replaceAll("-", "");
      _controllerDesc.text = widget.mov!.descricao;
```

Figura 19: Criação do CustomDialog

Fonte: próprio autor

E a seguinte mostra como é realizada a chamada da função na tela home.



```
_dialogAddRecDesp() {
  showDialog(
      context: context,
      builder: (context) {
        return const CustomDialog();
      });
}
```

Figura 20: Chamada do CustomDialog

Fonte: próprio autor

A aplicação ainda possui na parte inferior da tela, um menu que permite o usuário navegar entre as telas principais do aplicativo, podendo acessar a tela Home, de Status e Metas ao tocar no respectivo ícone.

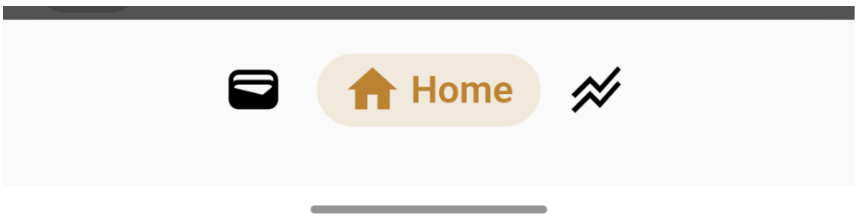


Figura 21: Menu da Aplicação

Fonte: próprio autor

6.2.4 Tela de Status

Ao clicarmos no ícone direito do menu a aplicação nos permite acessar uma tela de estatísticas, que representa a relação de todos os dados lançados no aplicativo, como pode ser notado na imagem a seguir.

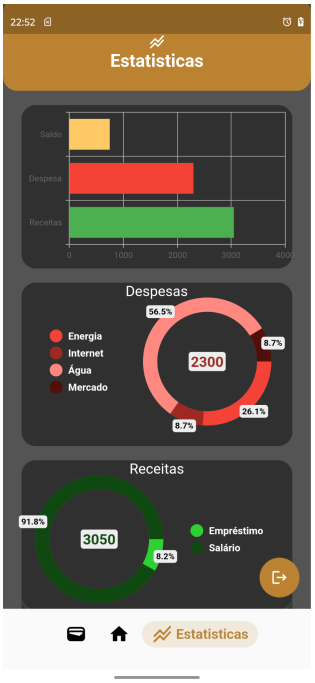


Figura 22: Tela de Estatísticas

Fonte: próprio autor

A tela é constituída basicamente de três gráficos, um que relaciona as receitas com as despesas e mais um gráfico que especifica quais as áreas de cada lançamento, os gráficos foram construído por uma biblioteca externa ao flutter, '*syncfusion_flutter_charts',* é uma biblioteca que possibilita ao desenvolvedor construir uma série de gráficos interativos, e com designe arrojado para facilitar a visualização de informações.

Os gráficos foram instanciados em um widget próprio onde foi possível realizar as devidas customizações e aprimoramento de funcionalidades, permitindo que fossem inserido na tela de status com maior facilidade como pode ser notado no exemplo abaixo.

```
@override
Widget build(BuildContext context) {
  List<_SalesData> salesDataList = [
    _SalesData(rec, totalR, Colors.green),
    _SalesData(desp, totalD, Colors.red),
    _SalesData(saldo, total, Color(0xFFFFC966)),
  ];
  return SfCartesianChart(
    primaryXAxis: CategoryAxis(),
    series: <BarSeries<_SalesData, String>>[
      BarSeries<_SalesData, String>(
        dataSource: salesDataList,
        xValueMapper: (_SalesData data, _) => data.tipo,
        yValueMapper: (_SalesData data, _) => data.valor,
        pointColorMapper: (_SalesData data, _) => data.cor,
      ), // BarSeries
    ], // <BarSeries<_SalesData, String>>[]
  ); // SfCartesianChart
}
}
```

Figura 23: Criação do BarChart

Fonte: próprio autor

O objetivo principal dessa tela é facilitar o entendimento do usuário na compreensão do lançamento de dados realizados pelo próprio, tornando assim sua experiência mais fácil e interativa

6.2.5 Tela de Metas

Em sequência ao tocar no ícone do lado esquerdo o usuário será direcionado para a tela de metas, tela essa que possui a finalidade de tentar restringir o usuário da aplicação de extrapolar em suas divididas lhe dando de maneira mais detalhada os lançamentos realizados por cada área especifica e dando a possibilidade de de estipular uma meta para o respectivo dado.

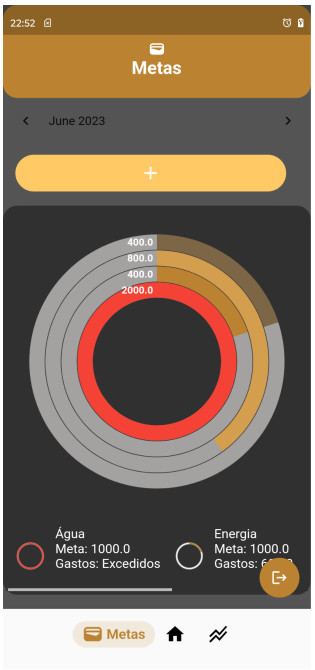


Figura 24: Tela de Estatísticas

Fonte: próprio autor

Esta tela também faz uso ‘*syncfusion_flutter_charts’* como biblioteca para que seja possível ilustrar os lançamentos realizados pelo usuário. A tela é composta principalmente por três mecanismos:

- Adicionar: Um botão na parte superior da tela dela que possui a função de chamar widget semelhante ao widget que realiza lançamentos na tela home;
- Calendário: Um calendário que permite ao usuário navegar entre os meses que possuem lançamentos;
- Gráfico de metas : Um gráfico que serve para mostrar ao usuário o andamento de cada uma de suas metas de maneira intuitiva.

Dentre os widgets lançados na tela de metas, o referente ao gráfico, é o principal da tela. O widget foi construído de maneira isolada ao código tela, e é um *StatefulWidget*, pois também altera o estado da aplicação a lançar ao lançar novos dados.

Essa página na aplicação faz o uso de uma nova coleção de dados que servem para armazenar quais seriam as metas que o usuário pretende alcançar durante o mês, os dados presentes nessa coleção se relacionam com as informações da coleção de movimentações que são apresentados em um *RadialBar* presente na tela.

O Flutter permite que sejam inseridas uma série de dependências diferentes para serem implementadas na aplicação, sendo instanciadas por meio de um arquivo *‘pubspec.yaml’,* que é crucial no gerenciamento de dependências do projeto e na

definição de configurações específicas do aplicativo. Como pode ser notado na figura a seguir.

```
dependencies:
  flutter:
    sdk: flutter

  table_calendar: ^3.0.8
  intl: ^0.17.0
  page_transition: ^2.0.9
  modal_bottom_sheet: ^2.1.2
  get_it: ^7.2.0
  graphql_flutter: ^5.1.2
  flutter_secure_storage: ^8.0.0
  cloud_functions: ^4.0.13
  firebase_auth: ^4.4.0
  firebase_core: ^2.13.0
  firebase_database: ^10.2.2
  google_sign_in: ^6.1.0
  sqflite: ^2.2.6
  shared_preferences: ^2.1.1
  cupertino_icons: ^1.0.2
  cloud_firestore: ^4.4.3
  firebase_auth_web: ^5.3.2
  path_provider: ^2.0.14
  uuid: ^3.0.0
  fl_chart: ^0.36.4
  charts_flutter: ^0.12.0
  pie_chart: ^5.3.2
  syncfusion_flutter_charts: ^21.2.4
```

Figura 25: Dependências da aplicação

Fonte: próprio autor

As informações são extraídas das coleções por meio de funções desenvolvidas para específicas para cada tratamento selecionado pelo sistema como pode ser notado no exemplo a seguir.

```
class MetasHelper {
  static final MetasHelper _instance = MetasHelper.internal();

  factory MetasHelper() => _instance;

  MetasHelper.internal();

  final FirebaseFirestore _firestore = FirebaseFirestore.instance;

  Future<void> saveMetas(Metas metas) async {
    print("chamada save");
    CollectionReference metasRef = _firestore.collection(metasCollection);
    metas.id = metasRef.doc().id;
    await metasRef.doc(metas.id).set(metas.toMap());
  }

  Future<Metas?> getMetas(String id) async {
    DocumentSnapshot snapshot =
        await _firestore.collection(metasCollection).doc(id).get();
    if (snapshot.exists) {
      return Metas.fromMap(snapshot.data() as Map<String, dynamic>);
    } else {
      return null;
    }
  }

  Future<void> updateMetas(Metas metas) async {
    print("chamada update");
    print(metas.toString());
    await _firestore
        .collection(metasCollection)
        .doc(metas.id)
        .update(metas.toMap());
  }
```

Figura 26: Criação de uma coleção do Cloud Firestore

Fonte: próprio autor

Os dados são inserido por meio de um novo *CustomDialog,* que exerce a mesmo função do presente na tela home, possuindo basicamente os mesmos componentes presentes:

Figura 18: Tela de adição de dados

Fonte: próprio autor

- Valor: Campo que o usuário informa um valor de receitas ou despesas;
- Descrição: Campo que o usuário informa o que foi a transação realizada;
- Confirmar/Editar: Botão que ao ser pressionado realiza confirma ou edita transação realizada;
- Cancelar: Botão que ao ser pressionado cancela a transação a ser realizada;

6.2 Comparação com aplicações semelhantes

Levando em consideração o anteriormente tratado pode se observar que a pesquisa logrou êxito em sua proposta, pois, além de cumprir com todos os objetivos que o trabalho buscava alcançar, foi possível notar que a natureza da aplicação se relaciona com todo material bibliográfico usado como fonte de pesquisa.

Um ponto importante levantado durante a pesquisa foi referente à identidade visual da aplicação, que quanto mais elaborada era aplicação, mais difícil poderia ser o seu

entendimento, ponto este que foi exemplificado ao analisar e considerar outras aplicações presentes no mercado, que são efetivas, porém podem causar o estranhamento e afastamento de alguns usuários, pela burocracia necessária para o uso da plataforma. O aplicativo desenvolvido opta pelo uso de uma identidade visual relativamente simples em relação a outras soluções, entretanto este é um ponto tendo em vista que um de seus objetivos é criar uma maior familiaridade do usuário para com suas próprias finanças e o próprio aplicativo.

Outra característica importante a se levar em conta está relacionada às funcionalidades da aplicação. Com base no levantamento de requisitos, o aplicativo cumpre com tudo que está disposto a alcançar.

Quadro 9: Aplicativos desenvolvido x Soluções de Mercado

RECURSOS	Belly Coin	Mobills	Guiabolso	Organizze	CoinKeeper
Gerenciamento De Cartão	X	✔	✔	✔	X
Interface Intuitiva	✔	✔	✔	✔	✔
Sistema De Metas	✔	✔	✔	✔	X
Controle de Receitas e Gastos	✔	✔	✔	✔	✔
Disponível *Offline*	✔	✔	✔	✔	✔
Planejamento Financeiro	✔	✔	✔	✔	✔
Controle Familiar	X	X	X	X	✔

Fonte: próprio autor

Como pode se perceber no quadro acima a aplicação, apesar de estar em uma primeira versão, cumpre com as principais características e funções de outras soluções presentes no mercado. Mesmo ainda não conseguindo rivalizar em nível de

funcionalidade com estas outras aplicações, presentes a bem mais tempo e com mais recursos, a aplicação de mostra eficiente entrega tudo que é prometido ao usuário, uma navegação simples e entendimento direto em relação a suas principais funções.

7. CONSIDERAÇÕES FINAIS

Este projeto apresentou o desenvolvimento da primeira versão de um aplicativo denominado Belly Coin, abordando a justificativa para a implementação, a documentação geral do aplicativo, o levantamento dos requisitos, as ferramentas usadas para o seu desenvolvimento e os resultados obtidos. Tendo como objetivo desenvolver um aplicativo móvel que facilitará o gerenciamento do financeiro pessoal de forma simples, além de também confeccionar a sua documentação.

Durante a realização do projeto ocorreram algumas dificuldades que podem ser notadas pelas diferenças entre a primeira concepção desenvolvida pelo Figma e o resultado final, as maiores dificuldades encontradas foram a falta de familiaridade com o desenvolvimento com o framework Flutter, que por mais já tivesse tido contato com a ferramenta nunca tinha ido tão a fundo ao desenvolvimento de uma solução com essa metodologia, fora carga e o cansaço que um projeto dessa importância carrega consigo.

Dito isso, pode se dizer que os resultados obtidos foram satisfatórios, uma vez que o aplicativo foi desenvolvido com o uso de uma linguagem nativa, e resultou em uma aplicação simples, eficaz, robusta, e de fácil entendimento aos usuários, que eram os principais objetivos ao desenvolvê-la.

8. FUTUROS PROJETOS E IMPLEMENTAÇÕES

Após o desenvolvimento do aplicativo, surgiram novas ideias de melhorias e funcionalidades para facilitar a vida dos usuários e tornar o aplicativo um pouco mais comercial e ter maior valor de mercado. Ideias essa que não foram implementadas nesse primeiro momento pois o tempo para realizar as melhorias era escasso e projeto poderia não ser encerrado dentro do prazo estipulado.

Dentre as melhorias que pretendo lançar futuramente, todas estão voltadas principalmente para a automatização da aplicação, como por exemplo:

- A conexão do aplicativo com as contas de cartão de crédito do usuário;
- Uma espécie de rastreio de CPF, para que assim que seja lançada uma nota ou compra no nome do usuário seja lançado automaticamente na aplicação;
- Parcelamento de despesa, para que, ao ser informado pelo usuário, as parcelas das despesas sejam lançadas automaticamente ao longo dos meses no sistema;

REFERÊNCIAS

ALVES, Max Braynner Menezes. **Bolso Virtual: Aplicação Web Móvel para controle de finanças pessoais.** 2017. Disponível em: http://ri.ufs.br/jspui/handle/riufs/9103. Acesso em: 18 de Ago. de 2022

ALVES, JOSÉ GUILHERME. **COLOCAGE: PLATAFORMA WEB PARA AUXÍLIO NO ENSINO-APRENDIZAGEM DE ESPECIFICIDADES DAS LÍNGUAS PORTUGUÊS E INGLÊS. 2022.** Disponivel em: https://repositorio.unisagrado.edu.br/handle/handle/1250. Acesso em: 06 de Jun. de 2023

BRANDÃO, Roney Soares. **SOBCONTROLE–SISTEMA DE CONTROLE DE FINANÇAS PESSOAIS.** Repositório de Trabalhos de Conclusão de Curso, 2018. Disponível em: http://pensaracademico.facig.edu.br/index.php/repositoriotcc/article/view/672/583. Acesso em: 18 de Ago. de 2022

COINKEPPER. **Right financial decisions**. Every day. 2022. Disponível em: https://about.coinkeeper.me/eng#features. Acesso em: 27 de Set. de 2022

CREMONEZI, ANDRÉ LUIZ. MYFINANCES: **Aplicativo móvel para o gerenciamento de finanças pessoais**. Centro Universitário de Araraquara (Uniara), 2015. Disponível em: :https://m.uniara.com.br/arquivos/file/cca/artigos/2015/andre-luiz-cremonezi.pdf. Acesso em: 18 de Ago. de 2022

CRUZ, Gabriel Mendes. **Desenvolvimento de um aplicativo para gestão de vendas: estudo de caso em vendas de calçados**. 2022. Disponível em: https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/4369/1/TCC%20II%20-%20Gabriel%20Mendes%20Cruz.pdf. Acesso em: 20 de Set. de 2022

DART. **Dart overview**. Dart dev. 2022. Disponível em: https://dart.dev/overview. Acesso em: 27 de Set. de 2022

DE CASTRO LINS, Hertz Wilton. **UML–Diagramas de caso de uso**. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/35725428/BD_-UML_-_Diagramas_de_caso_de_uso-with-cover-page-v2.pdf?Expires=1665782448&Signature=Dj1KZKwLVEtoRQBOabkKXcbVBeHV8sjfz1wvLZCZF29ayywSe-ZLxMkTDC14pAMbzRkA5nza-zU0gH01qnblj7aa7epoho7vl3GitDAvF3nW1b5QwSf7VAnhYLlcvUPbT~ehWkd84rUK5HhEHROdvxKIaK0AVpzDYihuMrNkm-J1t3BqcUwj0fTexpI7w1iF5~HkIesytye3VECeV0P881aQcGLvFV5fnDdVX~5dwdikdYkENLpit8R0HTqLgMYn5x9aUxTwFldtJCNKO6rxTIhts2ZzDWZZYR1486cYXIhx0a~t5vkrVQXqydhv32Xn-pHOvTNuIa09eoM-z8kByQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA. Acesso em: 14 de Out. de 2022

DE MENDONÇA, Ricardo Augusto Ribeiro. **Levantamento de requisitos no desenvolvimento ágil de software**. Semana da Ciência e Tecnologia da PUC Goiás, p. 12, 2014. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/35529111/Levantamento_de_requisitos_no_desenvolvimento_agil_de_software-with-cover-page-v2.pdf?Expires=1665779703&Signature=AosBegAPMG0arW8O26xM5E4rqTQrHJEwfFuwfR95G8jkOAfY~tRNc51wViq6NWNq1hSkab7nGOC6vZU868ue9QyI269TtvrW~wiA4eKd9rIfPyyLOgVN7sfieC3rp-QKrirUTVBm5z8jCZqzUwpo-tZX~FPe2f2ODsJSL~Sx3rD7um4calBns2y4uAMlZJE6Qs1fkVcmVvjfIwZDwHGVM0~BEDCmUM44MkFwfneAT8w1V1hKqy9JTXRJ0YBAgnwCxhlQnOvF~tqhA~tVprlWQAXZ9-77niigNLP9QXsY-rl8k4BOjeLiXkF~YrfBrblMlWcUuz~QU6YlPBk-p4sFFg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA. Acesso em: 14 de Out. de 2022

DO NASCIMENTO, Karla Angélica Silva et al. **Ferramenta de Prototipagem para Criação de um Aplicativo para o Ensino na Saúde**. In: Anais do XXVI Workshop de Informática na Escola. SBC, 2020. p. 509-513. Disponível em: https://sol.sbc.org.br/index.php/wie/article/view/12658/12521. Acesso em: 21 de Set. de 2022

FERREIRA, Caroline Agostinho et al. **Novas evoluções do mercado de crédito: Uma análise sobre as Fintechs**. Revista de Iniciação Científica da Universidade Vale do Rio Verde, v. 9, n. 1, 2018. Disponível em: http://periodicos.unincor.br/index.php/iniciacaocientifica/article/view/5221. Acesso em: 27 de Set. de 2022

FIREBASE. **Aprenda os fundamentos. Google**. 2022. Disponível em: https://firebase.google.com/docs?hl=pt. Acesso em: 22 de Set. de 2022

FLUTTER. **Learn Flutter any way you want.** Flutter dev.2022. Disponível em: https://flutter.dev/learn. Acesso em: 22 de Set. de 2022

GUIABOLSO. **O Guiabolso é o seu app de gestão financeira.** 2022. Disponível em: https://www.guiabolso.com.br/aplicativo. Acesso em: 27 de Set. de 2022

LEAL, Lennylde Cantanhede do Vale Ferreira. Raimundo Nonato Lima da Silva Mykhael Marinho Canjão. **Tópicos em Administração Volume 12, p. 38**. Disponível em: https://www.poisson.com.br/livros/adm/volume12/Topicos_em_Administracao_vol12.pdf#page=38. Acesso em: 14 de Out. de 2022

LEPORE, Giampaolo de Araujo. **Levantamento de requisitos para um sistema de apoio às disciplinas de PBL com a aplicação de Learning Analytics.** 2021. Disponível em: https://bdm.unb.br/handle/10483/29685. Acesso em: 21 de Set. de 2022

MOBILLS. **Guia Completo.** 2022. Disponível em: https://www.mobills.com.br/. Acesso em: 27 de Set. de 2022

NETO, Machado; JOSÉ, Olibario. **Usabilidade da interface de dispositivos móveis: heurísticas e diretrizes para o design.** 2013. Tese de Doutorado. Universidade de São Paulo. Disponível em: https://www.teses.usp.br/teses/disponiveis/55/55134/tde-07012014-110754/publico/dissertacaoOlibario.pdf. Acesso em: 20 de Set. de 2022

ORGANIZZE. **Quem somos.** 2022. Disponível em: https://www.organizze.com.br/quem-somos. Acesso em: 27 de Set. de 2022

RAMOS, Maria do Socorro Ferreira; DE SOUZA MOURA, Patrícia; LAVOR, Otávio Paulino. **Educação financeira: Sequência didática com o aplicativo "Minhas Economias". Revista de Investigação e Divulgação em Educação Matemática**, v. 4, n. 1, 2020. Disponível em: https://periodicos.ufjf.br/index.php/ridema/article/view/32047/21697. Acesso em: 20 de Set. de 2022

RESENDE, Laís Lopes. **Mani: aplicativo gerenciador de finanças pessoais utilizando react native.** 2022. Disponível em: https://repositorioinstitucional.uniformg.edu.br:21074/xmlui/handle/123456789/859. Acesso em: 4 de Out. de 2022

REZENDE, Lucas Rodrigues. **Sistema de controle financeiro.** 2020. Disponível em: https://repositorio.pucgoias.edu.br/jspui/handle/123456789/708. Acesso em: 20 de Set. de 2022

SOARES, Adriano Guimarães. **PoupaGrana: aplicativo gerenciador de finanças pessoais com interface conversacional. 2019**. Disponível em : https://www.lume.ufrgs.br/handle/10183/198485. Acesso em: 18 de Out. de 2022

SOUZA, Jéssica Colombo de. **Manual de finanças pessoais: maneiras de gerenciamento das finanças pessoais para a formação de patrimônio.** 2014. Disponível em: http://repositorio.unesc.net/handle/1/3200. Acesso em: 18 de Out. de 2022

VALENTIM, Natasha M. Costa; SILVA, Williamson; CONTE, Taiana. **Avaliando a Experiência do Usuário e Usabilidade de um Aplicativo Web Móvel: Um Relato de Experiência. Em: Cibe . 2015. pág. 788.** Disponível em: https://www.researchgate.net/profile/Natasha-Valentim/publication/281874420_Evaluating_the_user_experience_and_the_usability_of_a_mobile_web_application_An_experience_report/links/569d121008ae78356e563e9e/Evaluating-the-user-experience-and-the-usability-of-a-mobile-web-application-An-experience-report.pdf. Acesso em: 20 de Set. de 2022

VIEIRA, Saulo Fabiano Amancio; BATAGLIA, Regiane Tardiolle Manfre; SEREIA, Vanderlei José. **Educação financeira e decisões de consumo, investimento e poupança: uma análise dos alunos de uma universidade pública do norte do Paraná. Revista de Administração** Unimep, v. 9, n. 3, p. 61-86, 2011. Disponível em: http://www.raunimep.com.br/ojs/index.php/rau/article/view/345. Acesso em: 20 de Set. de 2022

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação.** 2. ed. Campus, 2014. 168 p. Disponível em: https://www.travessa.com.br/metodologia-de-pesquisa-para-ciencia-da-computacao-2-ed-2014/artigo/40b269e3-2592-41f2-bde3-5bac07bb5e37. Acesso em: 4 de Out. de 2022