

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**VINÍCIUS AIELO PEREIRA
ALINE APARECIDA CINTRA SILVA**

APLICATIVO MOBILE DE CONTROLE DE CONTAS

FRANCA/SP

2020

VINÍCIUS AIELO PEREIRA

ALINE APARECIDA CINTRA SILVA

APLICATIVO MOBILE DE CONTROLE DE CONTAS

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Ely Fernando do Prado

FRANCA/SP

2020

Ficha catalográfica

P436a Pereira, Vinícius Aiello e Silva, Aline Aparecida Cintra
Aplicativo mobile de controle de contas / Vinícius Aiello Pereira e
Aline Aparecida Cintra Silva [s.n], 2020

41f.; 30 cm; il

Trabalho de Graduação (Curso Superior de Análise e
Desenvolvimento de Sistemas) Fatec - Faculdade de
Tecnologia "Dr. Thomaz Novelino".

Orientador: Prof.Me. Ely Fernando do Prado

1.Controle Financeiro. 2.Finanças Pessoais. 3.Desenvolvimento
de Sistemas. 4.Mobile.React-Native. I. Autor. II. Título

CDD – 005

VINÍCIUS AIELO PEREIRA

ALINE APARECIDA CINTRA SILVA

APLICATIVO MOBILE DE CONTROLE DE CONTAS

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca – “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Trabalho avaliado e aprovado pela seguinte Banca Examinadora:

Orientador(a) : _____

Nome..... : Orientador: Me. Prof. Ely Fernando do Prado

Instituição : Faculdade de Tecnologia de Franca – “Dr. Thomaz Novelino”

Examinador(a) 1 : _____

Nome..... : Profa Me. Adriana Cristina de Araújo

Instituição : Faculdade de Tecnologia de Franca – “Dr. Thomaz Novelino”

Examinador(a) 2 : _____

Nome..... : Prof. Me. Jorge Luís Takahashi Hattori Instituição

Instituição : Faculdade de Tecnologia de Franca – “Dr. Thomaz Novelino”

Franca, 22 de junho de 2020.

As pessoas costumam dizer que a motivação não dura sempre. Bem, nem o efeito do banho, por isso recomenda-se diariamente.

Zig Zigla

RESUMO

O presente trabalho tem por objetivo principal o processo de desenvolvimento de um aplicativo móvel para o controle de finanças pessoais, disciplina fundamental para aqueles que desejam ter uma boa avaliação de sua saúde financeira. Nesse sistema, o usuário pode contabilizar suas receitas, despesas, investimentos, etc., além de avaliar de maneira mais clara sua real situação financeira. É, portanto, um instrumento de grande importância para aqueles que buscam auxílio para tomada de decisões sobre gestão financeira pessoal. Para tanto, foram efetuadas pesquisas bibliográficas focadas no estudo da gestão de finanças pessoais, de modo que o aplicativo possa atender aos requisitos necessários para auxiliar na gestão dos eventos financeiros. Alinhado a isso, o trabalho apresenta os principais processos de Engenharia de Software, como o levantamento de requisitos, à modelagem e à prototipação do aplicativo, levando em consideração as especificidades de software para o desenvolvimento mobile. O resultado foi a criação de um aplicativo mobile capaz de apresentar relatórios referente aos eventos financeiros lançados pelo usuário, de forma prática, fácil e simples. Conclui-se que o trabalho realizado seja suficiente para a implementação do referido aplicativo, contribuindo para a educação financeira e o crescimento pessoal de quem o utilizar, na qualidade de ferramenta de apoio à gestão financeira.

Palavras-chave: Controle Financeiro. Finanças Pessoais. Desenvolvimento de Sistemas. Mobile. React-Native.

ABSTRACT

This work aims mainly in the process of developing a mobile application for controlling personal finances, fundamental discipline for those who wish to have a good evaluation of their financial health. In this system, the user can account for their revenues, expenses, investments, etc., in addition to assessing more clearly their actual financial situation. It is, therefore, an instrument of great importance for those who seek help to make decisions about personal financial management. Thus, bibliographical research focused on the study of personal finance management, so that the application can meet the necessary requirements to assist in the management of financial events. In line with this, the paper presents the main software engineering processes, such as requirements survey, modeling and prototyping of the application, taking into account the software specificities for mobile device development. The result was a creation of a mobile application capable of presenting a record of events created by the user, in a practical, easy and simple way. As a conclusion, it is considered that the work done is sufficient for the implementation of this application, contributing to the financial education and personal growth of those who use it, as a tool to support financial management.

Keywords: Financial control. Personal finances. Systems development. Mobile. React-Native.

LISTA DE FIGURAS

Figura 1 – Camadas da engenharia de software.....	16
Figura 2 – Modelo espiral	21
Figura 3 – Funcionalidades	25
Figura 4 –Tela inicial	33
Figura 5 – Menu vertical	34
Figura 6 – Cadastro de Eventos	35
Figura 7 – Cadastro de Eventos – LISTA/DATA.....	35
Figura 8 – Consulta de Eventos	36
Figura 9 – Cadastra Cartões	37

LISTA DE QUADROS

Quadro 1 – Requisitos Funcionais.....	32
Quadro 2 – Requisitos Não-Funcionais.....	33

LISTA DE SIGLAS

CLI –*Command Line Interface*

SUMÁRIO

1 INTRODUÇÃO	11
2 DESENVOLVIMENTO	13
2.1 APLICATIVOS MOBILE	13
2.1.1 App.....	13
2.1.2 Tipos de aplicações mobilem	14
2.2 ENGENHARIA DE SOFTWARE	15
2.2.1 Software	16
2.2.2 Processos de software	17
3 O PRODUTO	22
3.1 TECNOLOGIA.....	22
3.2 TEMA	24
3.3 FUNCIONALIDADES	25
3.3.1 Cadastrar Eventos.....	26
3.3.2 Cadastra Contas	27
3.3.3 Transferências.....	27
3.3.4 Cadastra Cartões	28
3.3.5 Consulta Gráfica.....	28
3.3.6 Requisitos.....	29
4 PROTÓTIPO	32
4.1 FUNCIONALIDADES	32
4.1.1 Autenticação no Sistema.....	32
4.1.2 Tela Inicial	32
4.1.3 Cadastro de Eventos.....	34

4.1.4 Consultar Contas.....	37
4.1.5 Consultar Cartões	37
4.1.6 Gráficos.....	38
CONSIDERAÇÕES FINAIS	40
REFERÊNCIAS.....	41

1 INTRODUÇÃO

Este trabalho tem como objetivo principal o processo de desenvolvimento de um aplicativo móvel para o controle de finanças pessoais, disciplina fundamental para aqueles que desejam ter uma boa avaliação de sua saúde financeira. O trabalho está embasado em estudos teóricos explorando os conceitos de finanças pessoais, desenvolvimento de projetos e engenharia de software.

No primeiro deles constam as atividades relativas à realização do Trabalho de Graduação, que estão descritas abaixo:

Tema: Nesta etapa inicial, foram realizados estudos e reflexões para se definir o tema pelo qual guiou-se o trabalho, o escopo e os objetivos do mesmo.

Revisão bibliográfica: Foi necessário realizar um estudo mais aprofundado sobre finanças pessoais e engenharia de software, a fim de se acumular maior conhecimento teórico. O estudo teve base em fontes bibliográficas em livros, artigos e, principalmente, na Internet.

O outro grupo de atividades diz respeito ao desenvolvimento do software, que são divididas em:

Definição das funcionalidades: Por se tratar de um software voltado para as finanças pessoais, foram imprescindíveis as pesquisas sobre a área. Para isso, foi aprofundado o estudo sobre o tema para realizar o levantamento de dados para a composição funcional do sistema.

Estrutura do sistema: Todo o desenvolvimento do sistema foi baseado em um framework criada de forma a agilizar os procedimentos padrões de codificação. Esta estrutura disponibiliza funções pré-estabelecidas que fornece maior velocidade na criação das telas, funções nativas do dispositivo.

Interface Gráfica: Por se tratar de um projeto voltado para o ambiente *mobile*, este item foi trabalhado com uma visão comercial do produto,

agregando valor ao mesmo com uma carga visual mais atraente, clara e funcional.

Protótipo e testes iniciais: Foram realizados testes no sistema a fim de se encontrar erros e possíveis problemas de codificação. Simulou-se o funcionamento em um ambiente real do cotidiano de uma pessoa, utilizando todas as ferramentas disponíveis no software.

Implementação: Após o levantamento inicial dos requisitos do sistema e o término da estrutura básica, construíram-se parcialmente as funcionalidades do sistema.

No decorrer do trabalho houve necessidade de se transitar entre algumas destas etapas, sempre retornando ao estudo bibliográfico.

2 DESENVOLVIMENTO

O presente capítulo irá retratar os conceitos sobre aplicações *mobile* e seus diferentes tipos ressaltando suas características, além dos conceitos da engenharia de software e seus modelos de implementação.

2.1 APLICATIVOS MOBILE

É evidente o quanto os *smartphones* fazem parte do nosso cotidiano, seja para trabalho ou simplesmente para lazer esses dispositivos são considerados, por muitos, um acessório indispensável. Junto a esses aparelhos veem os aplicativos, sistemas que estão cada vez mais populares e que se tornaram importantes na vida dos usuários de *smartphones*.

2.1.1 App

Um aplicativo móvel, conhecido popularmente por seu nome abreviado app, é um software desenvolvido para dispositivos móveis, sejam eles: *smartphone*, *tablet*, *smartwatch* e etc. Tais aplicativos podem ser baixados e instalados no dispositivo através de lojas on-line, tais como Google Play, App Store ou Windows Phone Store.

A sigla "app" é uma abreviatura do termo "aplicação de software". No ano de 2010 foi considerada como "Palavra do Ano" pela American Dialect Society. (American Dialect Society, 2011)

Devido a propagação dos dispositivos móveis e o crescimento da importância dos softwares e suas evoluções, conduziram a expansão dos aplicativos para as mais diversas áreas, como jogos, redes sociais, saúde, banca e negócios, fotografias, entre outros, compondo uma ampla variedade de softwares nas lojas on-line. Tal expansão

teve como consequência à criação de inúmeros Apps com as mais diversas funcionalidades atendendo a cada necessidade de seu utilizador.

2.1.2 Tipos de aplicações mobile

Nesse meio de aplicações moveis, existem três tipos de aplicações, sendo elas: os aplicativos nativos, os *mobile webApps* e os híbridos. Diante dessas opções, este tópico irá mostra as diferenças entre elas.

Aplicativos nativos:

Para PRESSMAN (2015, p. 8) “o termo aplicativo evoluiu para sugerir software projetado especificamente para residir em uma plataforma móvel.”. Os aplicativos nativos são sistemas instaláveis e que podem ser acessados off-line através do dispositivo, podendo ser baixados através de uma loja específica para cada plataforma (como iOS, Android e/ou Windows Phone). Os softwares são desenvolvidos em linguagem nativa o que facilita o acesso a funcionalidades do sistema operacional que atua no dispositivo. Nesse tipo de aplicação, é utilizado o chamado ambiente de desenvolvimento integrado, IDE.

Por tratar diretamente com as bibliotecas do sistema operacional, os aplicativos nativos têm como vantagem a maior facilidade no seu desenvolvimento e otimização dos códigos. Por outro lado, tem como desvantagem uma aplicação que apenas funciona na plataforma na qual foi desenvolvida.

Mobile Web Apps: (sem citações)

Web Apps são na verdade páginas na internet acessadas através de um navegador, popularmente conhecido como websites. Não são realmente aplicativos, e possuem restrições no acesso de algumas funções do dispositivo, o oposto das aplicações nativas.

Esse tipo de aplicação possui um custo bem mais baixo de desenvolvimento e funciona para todas as plataformas por poderem ser acessadas através de um *browser*, compondo assim suas vantagens. Porém, em suas desvantagens encontramos o funcionamento dependente de acesso a internet, limitações ao acesso

das funcionalidades das plataformas dos dispositivos e a não publicação da aplicação em lojas.

Aplicativos híbridos:

Aplicativos híbridos são desenvolvidos com as mesmas tecnologias utilizadas no desenvolvimento de web apps e utilizam frameworks para serem convertidos em um aplicativo instalável no dispositivo. Essas aplicações também são publicadas em lojas online e tem seu funcionamento similar aos nativos. Para Nizamettin Gok e Nitin Khanna (2013, p.2): “Aplicativos “híbridos” são uma categoria especial de aplicativos da Web que ampliam o ambiente de aplicativos por meio do uso de APIs de plataformas nativas disponíveis em um determinado dispositivo.”

Em outras palavras, os aplicativos híbridos são uma mistura dos apps nativos e web, ou seja, usam elementos dos dois tipos de aplicações.

2.2 ENGENHARIA DE SOFTWARE

A engenharia de software é a disciplina responsável de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a sua manutenção.

PRESSMAN (2015, p.15) define a engenharia de software como uma “tecnologia em camadas”. Toda iniciativa de engenharia de software deve ter a qualidade como sua base de sustento. Acima da camada da qualidade encontram-se os processos, logo acima, os métodos e, em seu topo, as ferramentas. Há vários modelos de processos de software, entretanto, nenhum pode ser considerado o ideal devido suas divergências. Para SOMMERVILLE (2003), todos os métodos da engenharia de software compartilham das mesmas atividades fundamentais como especificação dos requisitos, desenvolvimento e implementação, validação e evolução.

A Figura 1 representa a tecnologia em camadas.

Figura 1 – Camadas de engenharia de software



Fonte: PRESSMAN; MAXIM, 2015, p.16.

Foco na qualidade: camada base que sustenta a engenharia de software

Processo: O processo define a metodologia que deve ser estabelecida para o projeto. Para PRESSMAN (1995), o processo constitui a base para o controle de gerenciamento de projetos de software e estabelece os métodos, nessa etapa são produzidos produtos de trabalhos (modelos, documentos, dados, relatórios, formulários, etc.).

Métodos: Traz consigo as informações de “como fazer” para desenvolver o software. “(PRESSMAN, 2015, p.16). “Os métodos envolvem um conjunto de tarefas, que incluem: comunicação, análise de requisitos, modelagem de projeto, construção de programa, testes e suporte”.

Ferramentas: Seguindo os conceitos de PRESSMAN (1995), a última camada é responsável por fornecer suporte automatizado ou semiautomatizado para o processo e para os métodos.

2.2.1 Software

A engenharia de software está fortemente relacionada ao software, pois ela torna capaz o desenvolvimento de sistemas complexos e com alta qualidade,

respeitando o prazo definido. São esses os aspectos responsáveis pelo sucesso de um projeto.

Conhecido como a parte lógica da computação, o software tinha uma importância relativamente pequena no início da era tecnológica, porém, com os avanços tecnológicos, aumentando a dependência de um sistema com funcionalidades específicas, sua importância foi se tornando cada vez maior. PRESSMAN (2015, p. 4) define o software como sendo:

1. Instruções que quando executadas produzem a função e o desempenho desejados.
2. Estruturas de dados que possibilitam que os programas manipulem adequadamente a informação.
3. Documentos que descrevem a operação e uso dos programas.

O software difere do hardware basicamente por uma única palavra: físico. O hardware é um produto tangível, ou seja, possui partes físicas, é possível tocar e vê-lo. Para o software há um nível de abstração maior, é um elemento de sistema lógico.

Segundo PRESSMAN (1995), “software não se desgasta”. Por não ser um produto com características físicas, como o hardware, o software não sofrerá um desgaste, mas, por outro lado existe a depreciação, onde o software sofre alterações em relação à mudança do seu escopo que podem acabar comprometendo a sua usabilidade e desempenho.

Outra característica do software é que são feitos sob encomenda, que faz do software uma peça moldada para ser aplicada em um processo específico ou no auxílio na realização de determinadas atividades. No entanto, atualmente é comum a reusabilidade dos códigos, processo com a finalidade de otimizar a criação de novos softwares e reaproveitamento do mesmo, dividindo um software em diferentes módulos(modularização).

2.2. 2. Processos de software

Processos de software são utilizados para controlar e coordenar projetos de desenvolvimento de softwares. Na engenharia de software os processos podem ser definidos para atividades como desenvolvimento, manutenção, aquisição e contratação de software.

Para SOMMERVILLE (2011), um processo de software pode ser definido como um conjunto de atividades e resultados associados que conduzem à produção de um software. Existe uma grande diversidade de processos de software, onde nenhum é o ideal, desenvolvidos de maneiras diferentes por cada organização de acordo com suas necessidades.

No entanto, em todo processo de software existem atividades fundamentais comuns, definidas por SOMMERVILLE (2011):

1. Especificação do software - as funcionalidades e as restrições de operação do software são definidas:

- Engenharia de Sistema: estabelecer uma solução geral para o problema, envolvendo questões além do software.
- Análise de Requisitos: levantamento dos requisitos do software. Tal etapa tem como objetivo produzir uma especificação de requisitos no qual o sistema deverá atender, essa especificação pode ser considerada uma documentação do sistema.
- Especificação de Sistema: descrição das funcionalidades do sistema. As descrições poderão incluir um plano de testes para verificar adequação do sistema e a conformidade dos os requisitos estabelecidos na etapa anterior.

2. Projeto e implementação - o software é codificado e implementado de acordo com as especificações:

- Projeto Arquitetural: desenvolvimento de um modelo conceitual para o sistema, onde descreve a arquitetura do sistema dividido em módulos.
- Projeto de Interface: cada módulo que compõem a arquitetura do projeto tem sua interface definida.
- Projeto Detalhado: os módulos em si são definidos de forma detalhada e tem seu algoritmo descrito de forma genérica, pseudocódigo;
- Codificação: a implementação do sistema em uma linguagem computacional.

3. Validação do software - o software é avaliado para verificar se está em conformidade com as necessidades do cliente:

- Teste de Unidade e Módulo: realização de testes para verificar possíveis erros e integridade das funções do sistema;
- Integração: união dos módulos em um só produto de software, e a verificação da interação entre eles quando operando em conjunto.

4. Evolução do software - o software é modificado para atender as novas exigências do cliente:

- O software é submetido a todas as fases anteriores.

Essas atividades definem o “ciclo de vida” do software. Existem diversos modelos de processo de software. Os mais conhecidos são:

Modelo sequencial linear ou modelo em cascata

O modelo em cascata organiza as atividades de forma sequencial. Para SOMMERVILLE (2011, p.20) "O modelo em cascata é um exemplo de um processo dirigido a planos". Cada fase envolve o planejamento e programação das atividades que devem ser aprovados antes de se iniciar a fase seguinte. Sendo assim, uma fase só deve ser iniciada após a conclusão daquela que a antecede.

Os principais estágios do modelo em cascata é a Análise e definição dos requisitos; Projeto de sistemas e software: onde é definido a arquitetura geral do sistema, analisando os requisitos tanto do software quando do hardware; Implementação e teste unitário: são feitos testes unitários de cada módulo gerado em determinadas etapas do projeto, nesse estágio, geralmente, permite-se um retorno à fase anterior para a correção de erros encontrados; Integração e teste: é feito a integração das partes produzidas do sistema e realizados teste como um sistema completo; Operação e manutenção: o sistema é colocado em uso e são feito as manutenções de erros que não foram detectados nos teste iniciais e a ampliação do sistema. Os documentos produzidos em cada etapa devem ser alterados para refletirem as mudanças implementadas do sistema final.

Modelo Incremental

Para PRESSMAN (2015) o modelo incremental é indicado no caso de sistemas muito extenso, pois é um modelo que disponibiliza rapidamente uma versão para o usuário. O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial e coletar um *feedback* dos usuários para aperfeiçoamento do sistema e prosseguir por meio de criação de várias versões até que um sistema adequado seja desenvolvido.

Segundo SOMMERVILLE (2011, p.22):

Cada incremento ou versão do sistema incorpora alguma funcionalidade necessária para o cliente. Frequentemente, os incrementos iniciais incluem a funcionalidade mais importante ou mais urgente. Isso significa que o cliente pode avaliar o sistema em um estágio relativamente inicial do desenvolvimento para ver se ele oferece o que foi requisitado. Em caso negativo, só o incremento que estiver em desenvolvimento no momento precisará ser alterado e, possivelmente, nova funcionalidade deverá ser definida para incrementos posteriores.

O princípio é que a cada ciclo ou iteração, uma nova versão do sistema é desenvolvida e entregue para uso ou avaliação do cliente. Por tanto, os requisitos devem ser levantados. Devido as características do modelo incremental, o software produzido é constatado como um sistema modular, de modo que se possa planejar o desenvolvimento em incrementos.

Dentre as vantagens do modelo incremental, podem ser citadas:

- Menor custo e menos tempo são necessários para se entregar a primeira versão.
- Entrega e implementação rápida de um sistema funcional e útil para o cliente.
- Agilidade na obtenção de *feedback* do cliente sobre o sistema desenvolvido.

Como desvantagens, podemos citar:

- Requisitos não estáveis ou completos podem fazer com que alguns incrementos sejam bastante alterados;
- A gerência do projeto é mais complexa, pois precisa-se de entregas regulares para o progresso do desenvolvimento.

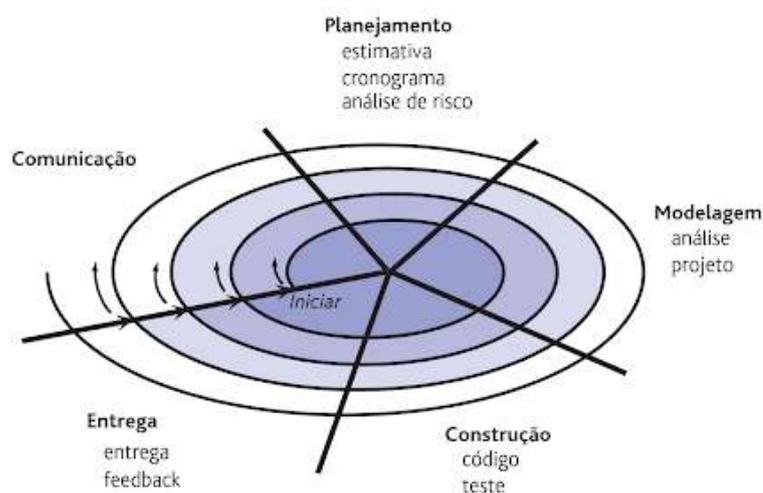
Modelo Espiral

Para PRESSMAN (2015, p.47) "O modelo espiral pode ser adaptado para ser aplicado ao longo de todo o ciclo de vida de uma aplicação, desde o desenvolvimento de conceitos até sua manutenção."

Proposto por Barry Boehm, o modelo espiral é indicado para sistemas e softwares de larga escala. O sistema é desenvolvido em ciclos, sendo que no primeiro pode ser realizado a especificação do produto ou um estudo de viabilidade. O planejamento é revisto a cada passagem na região destinada ao planejamento e o custo e cronograma são recalculados conforme o *feedback* do cliente. Ao longo da espiral podem ser desenvolvidos protótipos, que são usados como mecanismo na redução de riscos.

A figura 2 representa o modelo espiral.

Figura 2 – Modelo Espiral



Fonte: PRESSMAN; MAXIM, 2015, p.48.

O modelo espiral pode ser adaptado para os ciclos de manutenção do software, fazendo com que o modelo contemple todo o ciclo de vida do software.

3 O PRODUTO

O produto desenvolvido é um software, estruturado sobre plataforma *mobile* com a tecnologia de aplicativos híbridos, ou seja, é um aplicativo instalável no dispositivo do usuário. A interação será possível através do *smartphone* do usuário.

O software será utilizado com a finalidade de informatizar procedimentos contábeis para auxiliar o usuário a obter seu controle financeiro e permitir a visualização de seus gastos no qual poderá auxiliar nas tomadas de decisões.

3.1 TECNOLOGIA

Em um projeto como esse, um ponto muito importante é a escolha da tecnologia empregada para o desenvolvimento. Devido aos custos para o desenvolvimento de uma aplicação, tomou-se a decisão de basear o desenvolvimento em tecnologias denominada “livres”, para as quais não são necessárias aquisições de nenhum software comercial.

Para a programação será utilizado o React Native, framework desenvolvido pela equipe do Facebook, que possibilita o desenvolvimento de aplicações mobile, tanto para Android, como para iOS, utilizando apenas JavaScript, uma linguagem de script de código fonte livre e de uso geral, muito utilizada e especialmente criada para o desenvolvimento de aplicações Web. Atualmente, o React Native oferece suporte a iOS e Android e tem o potencial de se expandir para plataformas futuras também.

Com React Native as aplicações são escritas usando uma mistura de JavaScript e XML, conhecida como JSX. Segundo Bonnie Eisenman (2016, p.1): ” O “*bridge*” React Native invoca as APIs de renderização nativas em Objective-C (para iOS) ou Java (para Android)”.

Bridge, ou ponte para português, é responsável por realizar essa comunicação do JavaScript com as APIs nativas, diferencial que destaca o React Native dos demais frameworks do mercado. Outra vantagem é a utilização de recursos da plataforma, como câmera ou GPS, tornando se possível pela exposição da linguagem JavaScript para APIs nativas. Portanto, o aplicativo será renderizado utilizando componentes reais da interface do usuário móvel.

Para a utilização do React Native é essencial a instalação de algumas dependências para o seu funcionamento, entre elas: Node.js, CLI do React-Native e ferramentas de edições de código. Como ferramenta para auxílio de edição de código foi utilizado o Visual Studio Code, ferramenta *free* com suporte embutido as linguagens de programação web.

O banco de dados escolhido é o Firebase, uma plataforma de desenvolvimento adquirida pela Google no ano de 2014. Tal ferramenta foi criada com a finalidade de ser o *back-end* completo e de fácil usabilidade, disponibilizando diversos serviços que auxiliam no desenvolvimento e gerenciamento de aplicações.

O Firebase oferece serviços de fácil implementação, e a maioria deles podem ser usados de forma gratuita. As vantagens de implementar tais funções são inúmeras.

Os serviços utilizados serão:

Realtime Database:

Banco de dados que sincroniza os dados com os dispositivos em tempo real. Regras de segurança podem ser configuradas para definir quem tem acesso a quais dados.

Authentication:

Possibilita autenticação através de contas do Google, Facebook, Twitter, Github ou um sistema de contas próprio.

Há diversas outras ferramentas que são disponibilizadas pelo Firebase, como o *Analytics*, *Hosting*, *Cloud Firestore*, entre outros... Suas definições podem ser consultadas através da documentação do Firebase.

3.2 TEMA

O tema a ser escolhido para o desenvolvimento de uma aplicação foi finanças pessoais, disciplina que estuda os conceitos financeiros aplicados na vida de uma pessoa ou de uma família. A escolha do tema deu-se pelo fato da importância de se obter um controle financeiro pessoal, para que possa obter estabilidade financeira e dispensar endividamentos desnecessários.

Segundo Comparcida (2015, p. 12), “em finanças pessoais são considerados todas as características da família e os diversos eventos financeiros que esta atravessa, bem como a sua fase de vida, de modo a proporcionar um planejamento financeiro adequado às suas necessidades e prioridades”

Inicialmente é feito um diagnóstico da atual situação financeira e na definição de metas e prioridades, o que é possibilitado através da elaboração do orçamento doméstico. O principal objetivo das finanças pessoais é auxiliar no uso adequado do dinheiro, permitindo a satisfação das necessidades de acordo com suas prioridades. O equilíbrio no orçamento e o atingimento de metas, sejam eles de curto ou longo-prazo é possibilitado através do comprometimento e responsabilidade ao tratar dos recursos financeiros.

Em outras palavras, o objetivo das finanças pessoais é permitir que cada indivíduo ou família obtenha uma vida financeira controlada podendo, assim, superar complicações financeiras no cotidiano sem frustrações, bem como possibilitar alcances a outros objetivos maiores como comprar uma casa, um carro ou iniciar o próprio negócio, por exemplo.

Aplicando os conceitos do orçamento doméstico, torna-se possível acompanhar e verificar onde está sendo aplicado o dinheiro. Verifica-se as despesas de todos os gastos, até mesmo dos que são julgados como “irrelevantes” por serem de valores pequenos, pois gastos de baixo valor podem compor juntos uma grande fatia do orçamento. Controlar as contas fixa é uma tarefa mais simples comparada as despesas variáveis, considerada mais complexa por ter que registrar todos os eventos que ocorrerem durante o dia.

Em outras palavras, o orçamento doméstico não é mais do que uma planificação dos rendimentos que uma pessoa ou família obtém e do destino que lhes dá num determinado período de tempo (BARBOSA, 2011).

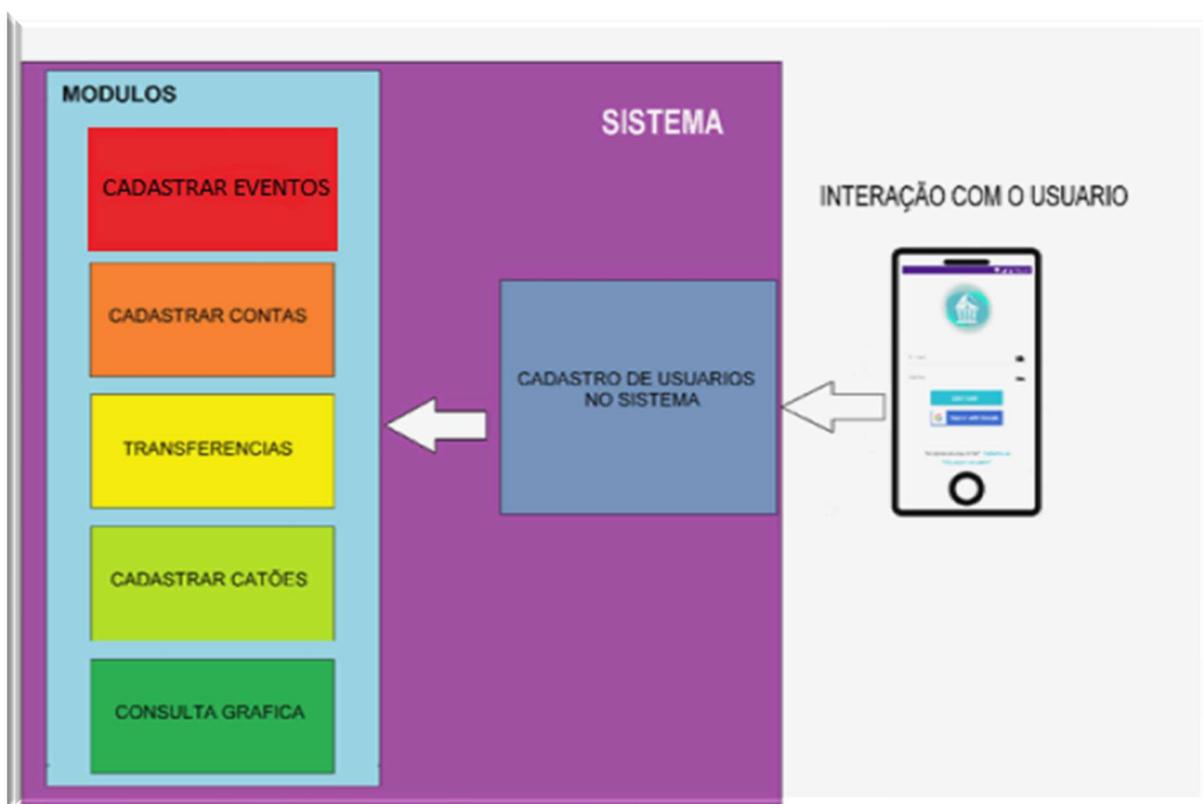
Após todas as anotações é possível ter uma visão ampla da situação financeira e verificar se está gastando além do que seria permitido, podendo assim, auxiliar nas decisões a serem tomadas.

3.3 FUNCIONALIDADES

Diante a engenharia de software as funcionalidades do sistema podem ser reconhecidas nos requisitos funcionais, parte no qual descreve as funções que o sistema deve executar. Os requisitos funcionais fazem parte do processo de licitações, implementado sobre o tema Finanças Pessoais.

A figura a seguir representa de forma estática as funcionalidades do sistema.

Figura 3 – Funcionalidades



Fonte: Autor.

Seguem abaixo as descrições das funcionalidades implementas no produto:

Um dos principais objetivos abordado sobre o tema em questão é ajudar a família a acompanhar e controlar os seus gastos financeiros, utilizando recursos do orçamento, ferramenta indispensável no mundo empresarial.

3.3.1 Cadastrar Eventos

Tradicionalmente, a forma de acompanhamento de gastos é feita através das anotações de todos os eventos recorrentes em um papel ou em uma planilha. Para Jorge e Ferreira (2011) os eventos devem estar associados a uma descrição, para que seja possível relembrá-lo no futuro e, também, a uma categoria. Associar a uma categoria permite a identificação das categorias que provavelmente requer uma ação

de controle. Com base nisso, foi levantado os requisitos necessários para tal funcionalidade.

Cadastrar Eventos é a Funcionalidade responsável por enviar ao banco de dados os eventos financeiros relatados pelo o usuário. É imprescindível que o usuário informe corretamente os dados nesse processo, pois informações erradas podem alterar drasticamente o resultado no momento de retirar o relatório dos gastos ou receitas, trazendo uma visão distorcida da realidade. Nessa etapa, torna-se importante também que o usuário sempre relate os eventos financeiros que ocorrem durante o dia.

3.3.2 Cadastrar Contas

O cadastro de contas tem o papel de gravar as contas, seja elas bancárias ou não, do usuário. Tal funcionalidade facilitará os lançamentos que ocorrerem na funcionalidade anterior, ou seja, se for lançado um evento de despesa o sistema automaticamente debitará o valor da conta a qual foi associada a despesa.

Esse procedimento de informar qual a conta associada ao evento traz uma visão mais detalhada do resultado, mostrando, por exemplo, onde o dinheiro da conta salário está sendo mais aplicado.

Vale ressaltar que o sistema não terá vínculos com contas bancárias e nem realizará débito automático nas mesmas, as contas cadastradas no sistema será apenas um ambiente virtual que imitará os eventos que estiverem ocorrendo na conta real do usuário.

3.3.3 Transferências

Está associada a funcionalidade anterior, pois é responsável por efetuar transferências de valores no saldo em contas, ou seja, é capaz de movimentar o saldo

entre contas (Ex. da conta Banco para conta Carteira), efetuando o débito do valor solicitado na conta de origem e o depositando na conta de destino.

As transferências em contas, mesmo pertencentes a mesma pessoa, também devem ser relatadas, pois para se obter um controle adequado deve-se saber qualquer movimentação financeira que ocorrer.

3.3.4 Cadastrar Cartões

Tal funcionalidade é responsável por armazenar dados relacionados ao cartão como: limite, data de pagamento da fatura e outras informações relevantes para o bom funcionamento da função. Ao associar uma fatura ao cartão o sistema automaticamente irá efetuar o lançamento das parcelas como despesa, organizando-as através da data de fechamento e validade da fatura.

3.3.5 Consulta Gráfica

A análise dos resultados é a etapa na qual irá auxiliar na determinação das ações necessárias para que a situação financeira comece a melhorar e, a melhor forma de se analisar dados é optar por gráficos, pois representações visuais facilitam a análise e a interpretação do conjunto de dados, podendo economizar tempo na visualização das informações. Com base nisso, optou-se por usar gráficos, construindo tal funcionalidade.

Responsável por apresentar os dados de forma gráfica, tal funcionalidade é imprescindível para o usuário, pois é ela quem trará um relatório no qual posteriormente poderá auxiliá-lo na tomada de decisões. Para isso, após analisá-las é preciso, também, questioná-las: os gastos com produtos estão condizentes com o que realmente precisa? Os investimentos estão tendo retornos favoráveis?

Como auxílio na visualização do conjunto de dados, o usuário terá opções de períodos, sendo elas mensais ou anuais, para geração dos gráficos, além dos três tipos de gráficos disponíveis no sistema.

3.3.6 Requisitos

Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos.

Requisitos são importantes para:

- Obter uma documentação que estabelece uma concordância entre o cliente e o desenvolvedor sobre as funções que o software deverá executar;
- Oferecer uma referência para que seja feita uma validação do sistema;
- Redução de custo no desenvolvimento. Pois requisitos mal definidos no processo de licitação podem causar retrabalho o que acarreta um maior custo no desenvolvimento.

Entendida a definição sobre requisitos, é preciso conhecer seus tipos.

Requisitos funcionais:

São requisitos de software que especifica uma função na qual o sistema deverá executar. Estes definem o comportamento do sistema, ou seja, o processo ou transformação que os componentes, sejam eles de software ou hardware, efetuam sobre as entradas para gerar as saídas. Em outro ponto de vista, são as funcionalidades do sistema.

Requisitos não funcionais:

São requisitos que expressam condições que o software deve atender ou qualidades específicas que o software deve ter. Os requisitos não funcionais colocam restrições no sistema. Um exemplo disso é: O software deve ser compatível com *smartphones* com Android 5.0 ou superior.

Para PRESSMAN (2015, p.141):” Pode ser descrito como um atributo de qualidade, de desempenho, de segurança ou como uma restrição geral em um sistema”

Abaixo estão representados, em quadros, os requisitos do sistema referente ao trabalho.

Quadro 1 –Requisitos Funcionais.

RF 001 – Adicionar Receita	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve ser capaz de adicionar as eventuais receitas cadastradas pelo usuário.		
RF 002 – Adicionar Despesa	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve ser capaz de adicionar as eventuais despesas cadastradas pelo usuário.		
RF 003 – Selecionar categoria	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve ter um campo com todas as categorias já cadastradas para que o usuário possa selecionar.		
RF 004 – Cadastrar categoria	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O Sistema deverá ser capaz de armazenar as categorias criadas pelo usuário.		
RF 005 – Consultar Gráficos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O Sistema deve gerar gráficos mensais, bimestrais, semestrais e anuais através dos eventos registrados pelo usuário.		
RF 006 – Adicionar Cartão de Crédito.	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O Sistema deve ser capaz de receber dados sobre cartões de crédito.		

RF 007 – Adicionar Conta ao Cartão de Crédito.	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O Sistema deve ser capaz de receber dados sobre a conta, relacioná-lo ao cartão de crédito e lançar, automaticamente, os valores da fatura as despesas.		
RF 008 – Adicionar Descrição.	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O Sistema deve ser capaz de armazenar uma descrição dos eventos lançados pelo usuário.		
RF009 – Adicionar Conta.	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O Sistema deve ser capaz de receber dados sobre um tipo de conta do usuário.		
RF 010 – Transferência.	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O Sistema deve ser capaz de realizar a transferências dos dados sobre o saldo relacionado as contas solicitadas.		

Quadro 2–Requisitos Não-Funcionais.

Requisitos não funcionais				
Nome	Restrição	Categoria	Obrigatoriedade	Permanência
RNF 1.1 Fácil manuseio.	O usuário deve ser capaz de realizar as atividades em menos de 60 segundos.	Performance.	<input type="checkbox"/> Desejável <input checked="" type="checkbox"/> Obrigatório	<input checked="" type="checkbox"/> Permanente <input type="checkbox"/> Transitório
RNF 1.2 Login de acesso	O sistema deve possuir login para acesso das funcionalidades.	Interface	<input type="checkbox"/> Desejável <input checked="" type="checkbox"/> Obrigatório	<input type="checkbox"/> Permanente <input checked="" type="checkbox"/> Transitório
RNF 1.3 Plataforma mobile.	O sistema deve ser operável em <i>smartphones</i> .	Interface	<input type="checkbox"/> Desejável <input checked="" type="checkbox"/> Obrigatório	<input checked="" type="checkbox"/> Permanente <input type="checkbox"/> Transitório

4 PROTÓTIPO

De acordo com o escopo do trabalho, está previsto o desenvolvimento de uma aplicação *mobile*. Serão listadas abaixo as funcionalidades presentes, ilustradas com figuras capturadas da própria tela do sistema.

4.1 FUNCIONALIDADES

4.1.1 Autenticação do Sistema

Antes de ser liberada a utilização do sistema, o usuário deverá informar o seu e-mail e senha, para que sejam validados e, posteriormente, permitidos ou vetado o acesso. Esse procedimento é denominado “*Login*”.

4.1.2 Tela Inicial

Após a validação dos dados de *login*, o usuário é direcionado à tela principal do sistema. Nesta tela serão exibidos alguns ícones que direcionaram o usuário para algumas telas como Consulta Gráfica, Cadastro de Receitas/Despesas, Contas e Cartões, ou seja, são atalhos rápidos das funcionalidades que serão mais utilizados pelo usuário, facilitando e agilizando sua navegação no sistema. A qualidade visual é um ponto que deve ser ressaltado. Pretende-se agradar os usuários com uma imagem trabalhada e de cores diferentes, trazendo tonalidades específicas em cada tela do sistema. Também podemos observar o menu de opções, disposto verticalmente, de forma oculta, na lateral esquerda da tela que pode ser acionado ao clicar no menu hambúrguer ou simplesmente deslizando a tela da esquerda para a direita. Ali são listadas todas as funcionalidades presentes no sistema, como demonstram as figuras abaixo.

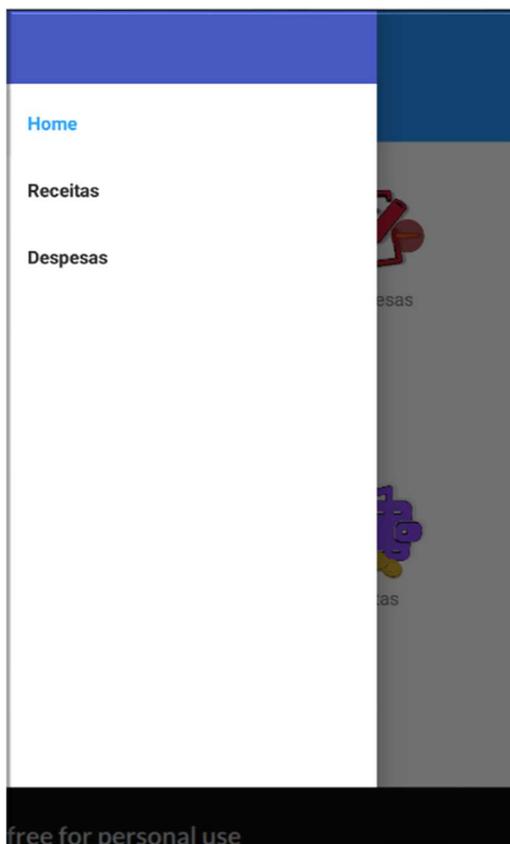
A figura 4 representa a tela inicial do aplicativo.

Figura 4–Tela Inicial



A figura a seguir representa o menu vertical do aplicativo.

Figura 5–Menu vertical.



4.1.3 Cadastro de Eventos

Ao navegar para as telas de cadastro, seja elas de receitas ou despesas, o sistema carregará um breve formulário para ser preenchido, onde o usuário deverá informar o valor do evento e, opcionalmente, uma descrição. O sistema listará, em um *DropDown* as categorias cadastradas no qual o usuário deverá associar ao evento e informar a conta na qual sofrerá alterações. Além disso, a tela contará com um campo de data, onde o usuário deverá informar a data em decorrência do evento.

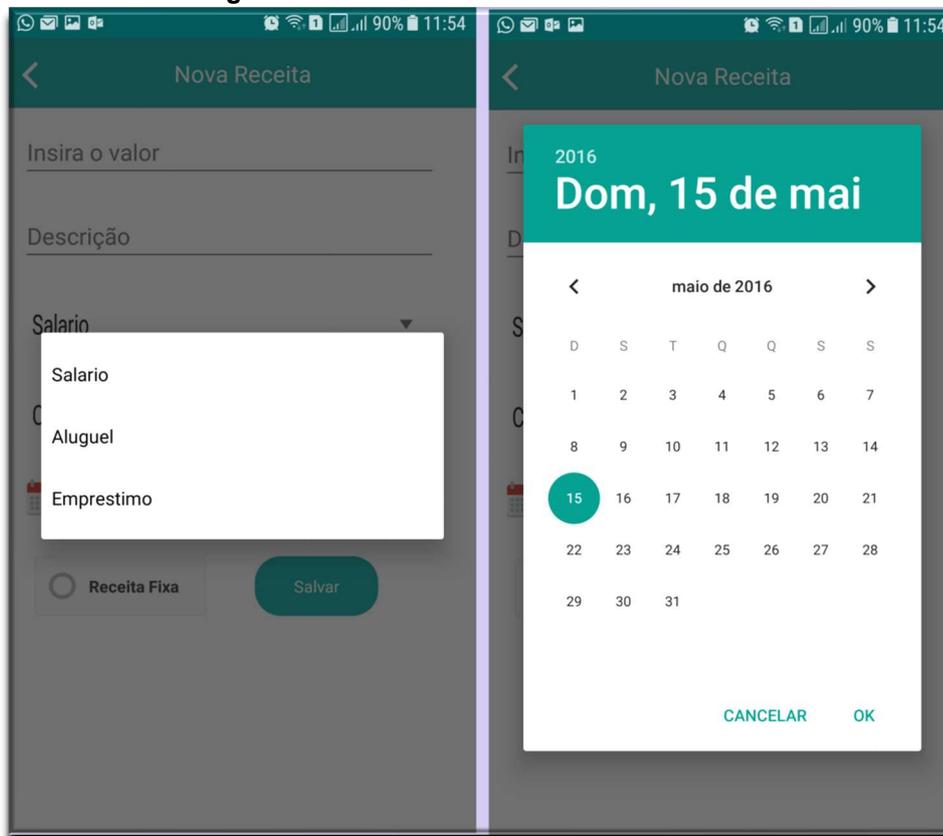
A figura 6 representa o cadastro de eventos.

Figura 6—Cadastro de Eventos

The image displays two side-by-side mobile application screens for recording financial events. The left screen, titled "Nova Despesa" (New Expense), has a red header and contains a form with the following fields: "Insira o valor" (Enter the value), "Descrição" (Description), a dropdown menu with "Alimentação" (Food) selected, another dropdown menu with "Carteira" (Wallet) selected, a date picker showing "15-05-2016", and a radio button labeled "Despesa Fixa" (Fixed Expense). A red "Salvar" (Save) button is positioned at the bottom right. The right screen, titled "Nova Receita" (New Receipt), has a teal header and contains a form with the following fields: "Insira o valor", "Descrição", a dropdown menu with "Salario" (Salary) selected, another dropdown menu with "Carteira" selected, a date picker showing "15-05-2016", and a radio button labeled "Receita Fixa" (Fixed Receipt). A teal "Salvar" button is positioned at the bottom right. Both screens show a status bar at the top with 93% battery and the time 11:09 on the left and 11:08 on the right.

A figura 7 representa o cadastro de eventos.

Figura 7–Cadastro de Eventos – LISTA/DATA.



Através do menu lateral é possível acessar a tela de consulta dos eventos cadastrados, podendo filtrar a consulta por seu respectivo mês e ano.

A figura a seguir representa a consulta de eventos.

Figura 8–Consulta de Eventos



4.1.4 Consultar Contas

Com o atalho rápido da tela inicial é possível navegar até a tela de consulta de contas, tal tela é responsável por listar todas as contas cadastradas pelo usuário. No canto inferior direito da tela é possível encontrar um ícone no qual abre um *PopUp* com um pequeno formulário para cadastrar novas contas.

Ao clicar sobre uma conta o sistema carrega suas informações e torna possível a edição ou exclusão do cadastro.

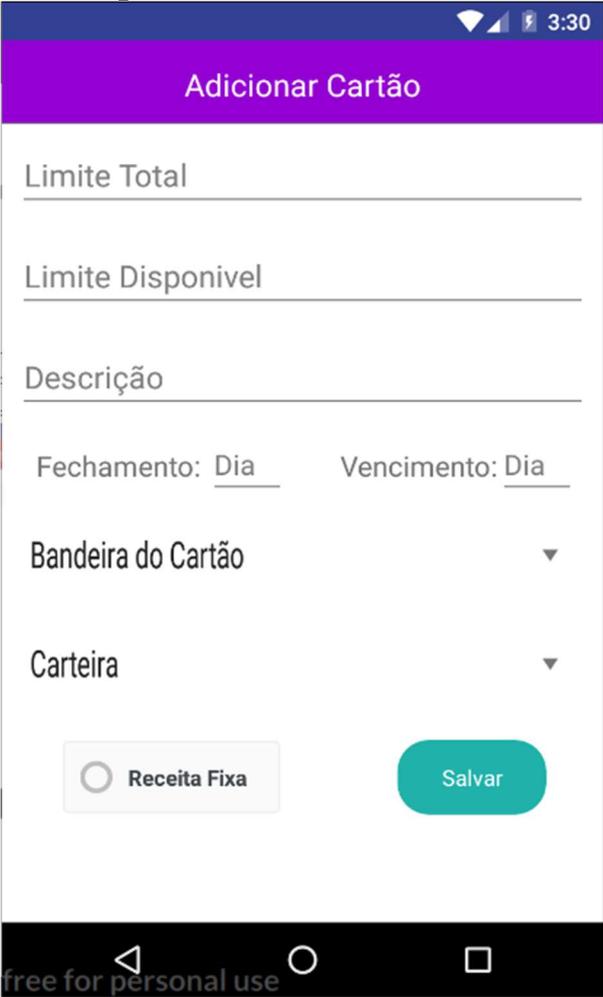
4.1.5 Consultar Cartões

A partir do atalho de acesso rápido na tela inicial, o módulo consultar cartões traz uma tela com todos os cartões listados, semelhante ao módulo “Consultar Contas”.

Há um ícone no canto inferior direito que ao ser acionado irá navegar para a tela de cadastro de novos cartões, tela no qual trará um formulário para receber dados referentes ao cartão.

A figura 9 representa o cadastramento de cartões do usuário.

Figura 9–Cadastra Cartões



Adicionar Cartão

Limite Total

Limite Disponível

Descrição

Fechamento: Dia Vencimento: Dia

Bandeira do Cartão ▼

Carteira ▼

Receita Fixa Salvar

free for personal use

Ao clicar sobre o cartão listado, será renderizado uma tela com as informações do respectivo cartão, podendo ser editado ou adicionado uma conta a ele. A Exclusão do cartão só será permitida se não houver nenhuma fatura pendente já lançada no sistema. Esse requisito de segurança permite a integridade dos dados no relatório final.

4.1.6 Gráficos

Podendo ser acessado através dos atalhos rápidos da tela inicial, o módulo de consulta gráfica traz os dados referentes aos eventos lançados. Os gráficos poderão ser gerados por:

- Evento: Trará as informações divididas entre receitas de despesas.
- Categorias: Tal função irá gerar os gráficos usando as categorias do evento como legenda, mas para isso o usuário deverá escolher o evento no qual terá os dados representados.
- Período: O gráfico irá conter apenas os dados referentes ao período mencionado. Tal funcionalidade é aplicável nas duas opções anteriores.

CONSIDERAÇÕES FINAIS

O objetivo principal o processo de desenvolvimento de um aplicativo móvel para o controle de finanças pessoais, disciplina fundamental para aqueles que desejam ter uma boa avaliação de sua saúde financeira. Algumas das funções são: lançamento de receitas e despesas, cadastro de cartões, cadastro de contas bancárias do usuário e geração de gráficos para análise dos resultados. O app permite a separação dos gastos por categoria (compras, casa, lazer, transporte, e etc.), o que permite ao usuário enxergar para onde está indo o dinheiro. Ao controlar as finanças não corre o risco de ser pego de surpresa, pois é possível saber quanto e quando as dívidas vencem.

A organização do orçamento pode nos ajudar a perceber os excessos, o que pode e deve ser cortado para que possamos alcançar com mais facilidade os nossos sonhos, sejam eles lugares que queremos conhecer, bens que queremos adquirir, investir em um negócio, está tudo atrelado ao controle e liberdade financeira. O devido planejamento é a base para o sucesso da vida financeira.

Na elaboração desse projeto tivemos algumas dificuldades, como na pesquisa e aprofundamento no assunto de finanças pessoais, visto que o tempo disponível para estudo era escasso. Tivemos dificuldade para aprender a linguagem utilizada na codificação do sistema, na elaboração de engenharia de software e levantamento de dados para a produção do mesmo.

Como objetivos futuros pretendemos realizar melhorias incrementais no sistema, desenvolver as funcionalidades restantes e disponibilizar para as lojas de aplicativos de smartphones. Desta forma poderá ser realizado um levantamento sobre o quanto o aplicativo consegue ajudar as pessoas a estimular sua educação financeira e administrar melhor o seu dinheiro.

REFERÊNCIAS

COMPARCIDA, Cleide. Administração financeira: 3º ano técnico. Colégio aplicação. Disponível em: <<http://www.colegioaplicacao.com.br/apostilas/ADM%20FINANCEIRA%20%203%20ANO.pdf>>. Acesso em: 01 jun. 2015

EISENMA, Bonnie. **Learning React Native: Building Native Mobile Apps**. 2ed. United States of America: O'Reilly, 2017

PRESSMAN, Roger S. **Engenharia de software**. 8 ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. Person, 2013.