

# Aplicação Móvel para Controle Financeiro

Luther Souto Jara<sup>1</sup>, Mirkos Ortiz Martins<sup>1</sup>

<sup>1</sup> Sistemas de Informação - Universidade Franciscana (UFN)  
97010-491 – Santa Maria – RS – Brasil

{luther.souto, mirkos}@ufn.edu.br

**Abstract.** *Given the growing interest of the Brazilian population about financial education and the search for tools that enable control over their finances, this study consists of the proposal to develop a mobile application for financial control, aimed at speeding up the insertion of information relevant to their expenses through the reading of QR CODE contained in Electronic Consumer Invoices. To carry out this project, we chose to use the FDD methodology for structuring the development processes, making use of the Dart language, as well as Firebase for data persistence. Due to technological limitations, a prototype was developed to illustrate the proposed solution.*

**Resumo.** *Dado o crescente interesse da população brasileira a respeito de educação financeira e busca por ferramentas que possibilitem controle sobre suas finanças, este estudo consiste na proposta de desenvolvimento de um aplicação móvel para controle financeiro, destinada a agilizar a inserção das informações pertinentes aos seus gastos através da leitura de QR CODE constantes em Notas Fiscais de Consumidor Eletrônicas. Para a realização desse projeto optou-se por utilizar a metodologia FDD para a estruturação dos processos de desenvolvimento, fazendo uso da linguagem Dart, bem como o Firebase para a persistência de dados. Devido a limitações tecnológicas, foi desenvolvido um protótipo para ilustrar a solução proposta.*

## 1. Introdução

O planejamento financeiro pessoal tem tornado-se uma das ferramentas de maior importância para alcançar os objetivos financeiros. Sem planejamento, muitas empresas por exemplo, iniciam o seu ciclo de vida sem ao menos conhecer seu público-alvo, tais como os fornecedores, a concorrência, os custos fixos e variáveis e acabam fechando as portas antes que o negócio complete dois anos. [Rico, 2018]

Segundo Barbosa (2011), o termo finanças pessoais “refere-se à disciplina que estuda a aplicação de conceitos financeiros e empresariais nas decisões financeiras de um usuário ou uma família”, ou seja, o planejamento financeiro pessoal nada mais é do que definir uma estratégia para tomada de decisões a partir da utilização de ferramentas de controle, podendo fazer uso de plataformas tecnológicas capazes de facilitar a realização dos objetivos financeiros, levando em consideração o perfil e característica pessoal de cada indivíduo.

No âmbito dos investimentos, a organização financeira e o conhecimento do mercado, são os pilares básicos para decisões e ações bem sucedidas, na qual segundo Chaves (2010), “É questão de saber organizar-se e ter disciplina, na qual a maioria das pessoas não

sabem, mas o ciclo de renda não possui nenhuma relação com a quantidade de dinheiro disponível, ou seja, não é somente nas grandes quantias que é possível formar reservas”.

Considerando o volume e disseminação crescente de informações relacionadas a educação financeira, torna-se extremamente útil e conveniente dispormos de ferramentas tecnológicas que auxiliem e agilizem o planejamento financeiro pessoal.

Dado o cenário aqui exposto e observando a grande oferta de plataformas para o controle de finanças (*Apps*, Planilhas, softwares, entre outros), identificou-se uma demanda ainda não atendida por essas plataformas, que apesar de alinhadas com as necessidades atuais da população interessada em consumir esse tipo de tecnologia, ainda podem ser aperfeiçoadas com funcionalidades que permitam mais agilidade e economia de tempo ao usuário.

O projeto proposto consiste em um aplicativo móvel de controle financeiro que tem como diferencial a automatização da inserção de dados à plataforma, através da leitura do *QR CODE* das Notas Fiscais de Consumidor Eletrônicas (NFC-e).

O *QR CODE* consiste em um código de barras bidimensional que permite o escaneamento através da câmera da maioria dos *smartphones* atualmente existentes, propondo uma categorização eficiente e precisa de despesas como alimentação, saúde, vestuário, transporte, educação, entre outros, baseada no código de Nomenclatura Comum do Mercosul (NCM) de cada produto constante na NFC-e.

Tendo em vista que os controles atualmente disponíveis estão limitados até certo ponto, já que as categorizações de suas respectivas integrações, quando automatizadas, são baseadas no ramo de atividade da empresa, isso pode gerar inconsistências de quando, por exemplo, é realizada uma compra de uma peça de vestuário, e itens de alimentação no mesmo estabelecimento como um hipermercado, constando portanto, todos os itens na mesma NFC-e, o que culmina em categorizações genéricas como por exemplo gastos com “mercado”, sem que essas informações reflitam a precisa alocação do recurso por parte do usuário.

Na maioria dos casos, as categorizações hoje incorporadas nas aplicações disponíveis no mercado, possuem uma linha tênue entre de fato serem úteis ou não, conforme o exemplo a seguir:

Em um cenário hipotético, considerando apenas a perspectiva de que ocorreram integrações e sincronizações automáticas da ferramenta de controle junto aos sistemas de bancos e/ou cartões que o usuário possua, ao fim do mês, após diariamente o usuário ter realizado suas operações financeiras, ele tem como fonte de consulta, informações categoricamente dispostas da seguinte forma:

- 45 % Supermercado
- 20 % Transporte
- 5 % Serviços
- 10 % Saúde
- 5 % Vestuário
- 2 % Eletrônicos
- 3 % Lazer
- 4 % Educação
- 6 % Outros

Assume-se então, que o usuário comprou peças de vestuário, 1 livro, 1 caderno e 2 canetas no supermercado, além de alguns alimentos, posteriormente, passou em um posto de gasolina para abastecer o carro, e ao pagar no caixa, resolveu comprar um carregador de celular, que estava a venda no balcão.

Nesse cenário, nenhuma das operações será adequadamente categorizada, visto que os itens mencionados não estarão na categoria Lazer (Livro), Educação (caderno e canetas) e Eletrônicos (carregador de celular) já que a atividade econômica das empresas que comercializam esses produtos/serviços é mais abrangente e essas diferenças não são expostas ao cliente na relação entre as empresas e suas respectivas operadoras de máquinas de cartão.

Por exemplo, na operação do posto de combustíveis, independente da mercadoria ou serviço que se consuma, poderá constar no extrato do cliente tipificado simplesmente como “Serviços”, ou em casos mais graves, todas as mercadorias adquiridas em um supermercado estarem categorizadas como alimentação.

Portanto, tendo em vista essas cotidianas inconsistências, se tornou conveniente o desenvolvimento desse projeto, construindo uma ferramenta que contribua à rotina de seus usuários, auxiliando-os para uma melhor análise e tomada de decisão a respeito de suas finanças pessoais.

## 2. Referencial Teórico

Hoje existem diversas formas de criar aplicativos *mobile*, onde o mais usual é a utilização da Linguagem Java/Kotlin no Android e Swift/Objective-C para o iOS.

Embora esse cenário represente um padrão atual para o mercado de desenvolvimento, para programação com essas tecnologias é necessário ter profissionais que demandam conhecimentos das duas plataformas, podendo existir outras dificuldades como sincronizar as *features* que existem em uma aplicação de uma plataforma em outra e então conseguir evoluir o código base de forma conjunta [Souto, 2019].

Uma alternativa para isso seria ter uma base de código única que executasse em ambas as plataformas, diante disso surgiu a ideia de usar *WebViews*. Por meio dessa ideia, cria-se *interfaces* e é possível executar um código apenas, *cross platform*. Com os adventos das *SPAs*, ou seja, *frameworks front end* que facilitaram trazer páginas dinâmicas baseadas em dados vindos de *APIs* e transições de tela mais próximas dos aplicativos nativos, isso foi ganhando cada vez mais popularidade principalmente para projetar *MVPs* ou mesmo aplicativos completos. Muitas tecnologias têm surgido no ecossistema das *WebViews* dando um suporte melhor para tudo isso [Souto 2019].

Portanto, considerando as informações apresentadas, para o desenvolvimento do projeto, que consiste em uma aplicação inicialmente para a plataforma Android será feito uso da Linguagem Dart, através do *framework cross platform* à ser utilizado na implementação, o Flutter, flexibilizando assim, uma futura disponibilização para outras plataformas, já que tecnologias como essa, permitem que um sistema seja executado em diferentes sistemas operacionais com apenas um único código-base. Além disso, o banco de dados será desenvolvido com Firebase. Como ambiente de desenvolvimento será utilizado a IDE Android Studio.

## 2.1. Android

Android é atualmente o sistema operacional da Google para dispositivos móveis baseados no Linux. A loja virtual *Play Store* tem uma diversidade de aplicativos, tanto pagos quanto gratuitos, para os *smartphones* e *tablets* com Android. Os sistemas operacionais são programas que gerenciam todas as tarefas de um dispositivo e fornecem uma *interface* visual para que se possa interagir com um sistema eletrônico, sem necessariamente saber o que acontece no interior dele [Cidral 2011]

## 2.2. Linguagem Dart

Dart é uma linguagem de *script* voltada à web desenvolvida pela Google, lançada na GOTO Conference 2011, que aconteceu em outubro de 2011 em Aarhus, na Dinamarca, em um projeto fundado pelos desenvolvedores Lars Bark e Kasper Lund. O objetivo da linguagem Dart foi inicialmente a de substituir a JavaScript como a linguagem principal embutida nos navegadores [Velooso 2011].

A maioria das arquiteturas atuais faz uso de plataformas e linguagens diferentes em ambos os lados cliente e servidor, o que muitas vezes leva a código duplicado e complexidade aumentada, isso sem falar dos custos de manutenção e declínio da produtividade da equipe e foi pensando nisso que a Google criou a linguagem de programação Dart, com o intuito de facilitar a forma em que é realizada a integração da aplicação como um todo [Sampaio, 2015].

A Dart foi desenhada para facilmente escrever ferramentas de desenvolvimento para aplicações *web* modernas e capacitadas para ambientes de alta performance [Dart 2019].

Considerando a tendência de uma crescente adoção do uso dessa linguagem, a mesma foi definida para codificação desse projeto.

## 2.3. Firebase

Segundo Orlandi (2019), o Firebase é um Baas (*Backend as a Service*) para aplicações Web e Mobile do Google, foi lançado em 2004 e com o passar dos anos cresceu muito, se tornando uma ferramenta que hoje para alguns projetos é a melhor opção, devido a quantidade de serviços oferecidos por ele, além da facilidade de implementação.

O Firebase oferece uma gama de serviços que podem ser utilizados, eles são separados por 4 grandes categorias, sendo elas Analytics, Develop, Grow e Earn.

BaaS ou BackEnd As A Service (MBaaS = *Mobile BackEnd As A Service*) é um serviço disponibilizado em que toda a estrutura do backend como: configuração de servidor, integração com banco de dados, sistema de *push notification* e outros serviços, que fazem parte do *backend*, estão completamente prontos para se integrar com o seu aplicativo.

## 2.4. Flutter

Flutter é um *framework* para desenvolvimento híbrido de aplicativos utilizando Dart como a linguagem base para a codificação dos aplicativos.

Diferente do *framework React Native* por exemplo, que possui um intermediário (*Bridge*) entre a UI (*User Interface* ou Interface do usuário) e o dispositivo, o Flutter fica

na camada do *UI* e não na chamada dos componentes nativos do SO e é desenhado diretamente em um canvas, que consiste em um elemento para delimitação de área destinada a renderização dinâmica, que aumenta a performance e fluidez a nível de um aplicativo desenvolvido exclusivamente de forma nativa.

Além do ganho com performance, o Flutter é uma tecnologia recente e com *features* em desenvolvimento a serem disponibilizadas para a *Web* e *Desktop*, ou seja, será possível construir um aplicativo que poderá ser reaproveitado para *Web* e também para *Desktop* caso necessário, com apenas uma base de código para toda a camada de apresentação [Santana 2019].

Pensando nessas vantagens e na fase de pleno crescimento da tecnologia, que o Flutter foi definido como o *framework* a ser utilizado nesse projeto.

## **2.5. Android Studio**

O Android Studio é chamado de Ambiente de Desenvolvimento Integrado (ou *IDE*, sigla em inglês para *Integrated Development Environment*), um programa de computador que reúne as características e ferramentas de apoio para a criação de aplicativos para dispositivos móveis para Android [Harada 2019].

## **3. Trabalhos Relacionados**

Os trabalhos apresentados a seguir foram obtidos e analisados através de estudo realizado sobre as ferramentas de controle financeiro disponíveis no mercado, com o objetivo de elucidarem o cenário atual de consumo dos usuários por esse tipo de aplicação, para então, evidenciar o diferencial proposto nesse projeto.

### **3.1. Sobcontrole - Sistema de Controle de Finanças Pessoais**

O artigo elaborado por Brandão (2017), busca fornecer uma ferramenta de auxílio ao planejamento financeiro doméstico/familiar, permitindo que o usuário tenha noção clara do seu fluxo de entradas e saídas financeiras, promovendo o hábito de poupar e se planejar para o futuro.

O autor percebeu que a falta de planejamento financeiro da população, é conseqüentemente, um dos causadores do grande volume de endividamento atualmente existente na vida da maioria dos brasileiros, agravado principalmente pelas altas taxas de juros praticadas no país.

Diante desse cenário, foi criado um aplicativo que ajude as pessoas a controlar suas vidas financeiras. Sob Controle é um aplicativo de controle financeiro, onde o usuário poderá lançar suas despesas e receitas, tendo o controle de toda a sua vida financeira.

O trabalho foi realizado através de uma pesquisa exploratória com análise bibliográfica e estudo de caso através de um questionário aplicado a um grupo de pessoas de ambos os sexos, com idades entre 18 e 67 anos, da cidade de Manhuaçu e região. O questionário foi feito em um formulário online para resultados mais amplos.

### **3.2. MyFinances: Aplicativo Móvel para o Gerenciamento de Finanças Pessoais**

O projeto desenvolvido por Cremonesi (2015) teve como intuito criar um aplicativo de gerenciamento de finanças para apoiar os usuários de forma fácil e intuitiva, na qual deverão

despender de apenas alguns minutos do seu dia para controlar seus gastos. Considerando que pequenas quantidades de dinheiro do dia a dia têm grande importância no controle financeiro e na formação de reservas, identificou-se a necessidade de disponibilizar algo prático e acessível que possa auxiliar as pessoas na organização financeira e que elas possam utilizar em qualquer momento do seu dia.

Foi realizada uma pesquisa sobre outros aplicativos com características semelhantes e disponíveis em lojas para download de aplicativos móveis. Após os estudos sobre o domínio, estudou-se qual modelo de processo da Engenharia de Software poderia ser empregado para realizar a concepção e desenvolvimento do aplicativo proposto. Técnicas de levantamento de requisitos com o emprego de entrevistas e a elaboração de um documento de requisitos nortearam as etapas de concepção e desenvolvimento.

Protótipos de média fidelidade foram elaborados e avaliados por voluntários representantes do domínio em diferentes momentos do desenvolvimento do aplicativo. Para o armazenamento de dados locais gerados pelo aplicativo foi utilizado o banco de dados SQLite.

Quanto ao *layout* do aplicativo, esse foi elaborado seguindo padrões de *interface* sugeridos pela Google, como o *Material Design*.

### **3.3. Aplicativo Mobills – Controle Financeiro Pessoal**

A ideia nasceu depois que os dois idealizadores, Carlos Terceiro e David Mosiah, começaram a perceber que, por diversas vezes, acabava o dinheiro, mas não acabava o mês. Viram também que isto é um problema comum de muita gente. No fim, nunca era possível dizer ao certo com o que foi gasto o dinheiro de cada um.

Então com o objetivo de fornecer uma ferramenta útil e acessível para controle financeiro pessoal, desenvolveu-se uma aplicação móvel que serve de ferramenta para controle financeiro pessoal, possibilitando vinculação de cartões, cadastro de metas, orçamentos, além de fornecer gráficos interativos. Também oferece sincronização das informações na nuvem e também identifica os gastos informando a geolocalização, além de uma versão *web* e outras funcionalidades que são oferecidas conforme o plano de assinatura desejado [Mobills 2019].

### **3.4. Aplicativo Guia Bolso – Simplificando sua vida com o dinheiro**

Segundo o fundador, Thiago Alvarez, a desorganização financeira da população, desconhecimento da real situação financeira em que o cidadão está inserido é o que em muitas vezes o impede de atingir seus objetivos financeiros.

Com o objetivo de fornecer uma ferramenta que organize os gastos de forma descomplicada e automatizada, desenvolveu-se uma aplicação com funções semelhantes ao Mobills, tendo como atual diferencial, pareceria com grandes bancos e financeiras para oferecimento de empréstimos, além de disponibilizar a situação do CPF do usuário através de parceria com a empresa Boa Vista, empresa de informações de crédito com abrangência nacional.

### **3.5. Conclusão sobre os trabalhos relacionados**

Conclui-se portanto, que através do sistema Sobcontrole de Brandão (2017), o usuário passa a ter maior controle de sua vida financeira. Ressaltando que, apenas com o bom uso

da ferramenta e com o lançamento de todas as suas receitas e despesas, que a mesma irá proporcionar tudo aquilo que é capaz de lhe oferecer, tornando o planejamento financeiro, mais claro.

Por outro lado, segundo o trabalho desenvolvido por Cremonezi (2015), conclui-se que as pessoas precisam adquirir uma vida financeira saudável e, para isso, precisam adotar um controle detalhado de suas finanças. Esse controle permitirá avaliar as despesas e receitas, bem como deixá-las mais equilibradas. Com a ajuda de tecnologias que estão cada vez mais presentes na vida das pessoas como os smartphones, este trabalho apresentou uma solução para ajudar essas pessoas na organização de suas finanças e que faz uso dos referidos dispositivos.

Nota-se que o aplicativo Mobills cumpre com o que promete, fornece dados em tempo real, com uma interface moderna e intuitiva, design adequado, e bem completo, e frequentemente adicionando novas funcionalidades.

Por fim, a respeito do aplicativo Guia Bolso, na qual trata-se de um aplicativo robusto, o mesmo oferece diversos serviços e funcionalidades através de sua plataforma, sendo a aplicação mais completa entre as mencionadas, vindo ao encontro da necessidade atual de controle financeiro pessoal, atendendo plenamente a demanda e satisfazendo inclusive os usuários mais exigentes, cobrindo uma parcela relevante desse mercado

Evidencia-se a partir do exposto, a conveniência que o projeto proposto neste trabalho agregará aos controles financeiros do usuário, possuindo como diferencial a agilidade ao inserir informações no sistema, bastando apenas realizar a leitura de um *QR CODE* de uma NFC-e.

#### **4. Metodologia**

Com o objetivo de atingir o melhor resultado possível, se fará uso da metodologia *Feature-Driven Development (FDD)* para o desenvolvimento do projeto.

FDD significa desenvolvimento orientado à funcionalidades, ou seja, por um requisito funcional do sistema. É prático para o trabalho com projetos iniciais ou projetos com codificações existentes. Assim como XP, ASD, Scrum e AUP, faz parte das metodologias ágeis originais, sendo este um modelo incremental e iterativo do processo de desenvolvimento de software que, segundo Heptagon (2017), tem como lema "Resultados frequentes, tangíveis e funcionais".

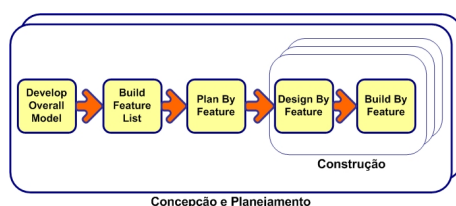
Entre todas as práticas oferecidas pela metodologia, não existem regras que definam quais são as mais apropriadas. Cabe ao gerente do projeto e a equipe definirem quais se adequam mais a cada situação [Aranha, 2014].

Concebido originalmente por Jeff de Luca, o FDD surgiu em Singapura, entre os anos de 1997 e 1999 com base no método Coad (Criado por Peter Coad – 1980/1990) e nos processos iterativos e lean já utilizados por Jeff de Luca. [Rocha, 2013]

A metodologia FDD possui cinco processos básicos [Aranha, 2014]:

- Desenvolver um modelo abrangente: Pode envolver desenvolvimento de requisitos, análise orientada por objetos, modelagem lógica de dados e outras técnicas para entendimento do domínio de negócio em questão. O resultado é um modelo de objetos (e/ou de dados) de alto nível, que guiará a equipe durante os ciclos de construção.

- Construir a lista de funcionalidades: Decomposição funcional do modelo do domínio, em três camadas típicas: áreas de negócio, atividades de negócio e passos automatizados da atividade (funcionalidades). O resultado é uma hierarquia de funcionalidades que representa o produto a ser construído (também chamado de *Product Backlog*, ou lista de espera do produto).
- Planejar por funcionalidade: Abrange a estimativa de complexidade e dependência das funcionalidades, também levando em consideração a prioridade e valor para o negócio/cliente. O resultado é um plano de desenvolvimento, com os pacotes de trabalho na seqüência apropriada para a construção.
- Detalhar por funcionalidade: Já dentro de uma iteração de construção, a equipe detalha os requisitos e outros artefatos para a codificação de cada funcionalidade, incluindo os testes. O projeto para as funcionalidades é inspecionado. O resultado é o modelo de domínio mais detalhado e os esqueletos de código prontos para serem preenchidos.
- Construir por funcionalidade: Cada esqueleto de código é preenchido, testado e inspecionado. O resultado é um incremento do produto integrado ao repositório principal de código, com qualidade e potencial para ser usado pelo cliente/usuário.



**Figura 1. Processo de Desenvolvimento FDD (PIMENTEL, 2010)**

#### 4.1. Desenvolver um modelo abrangente

Este processo é o que abrange todo o projeto, e é realizado por membros do domínio do negócio juntamente com os desenvolvedores [FDD 2018]. Realiza-se então um estudo dirigido sobre o escopo do sistema e seu contexto. Após, são realizados estudos com melhor detalhamento sobre o domínio do negócio para cada área a ser modelada. Deste modo, se construirá um modelo que satisfaça o domínio em questão [FDD, 2018].

Os modelos, ou o modelo, que forem eleitos para orientar o trabalho a partir de então será utilizado por todos durante o desenvolvimento do projeto [Aranha, 2014].

#### 4.2. Planejar por funcionalidade

Atividade que trata do estudo detalhado ante às funcionalidades do sistema analisando duas dependências e complexidade para que se consiga realizar uma estimativa de tempo de desenvolvimento de cada funcionalidade. Dá-se ênfase na construção de uma ordem de planejamento de implementação analisando a complexidade e nível de importância da mesma para o projeto [FDD 2018].

Na Tabela 1, verifica-se o planejamento para o desenvolvimento das funcionalidades que se fizeram necessárias neste projeto.



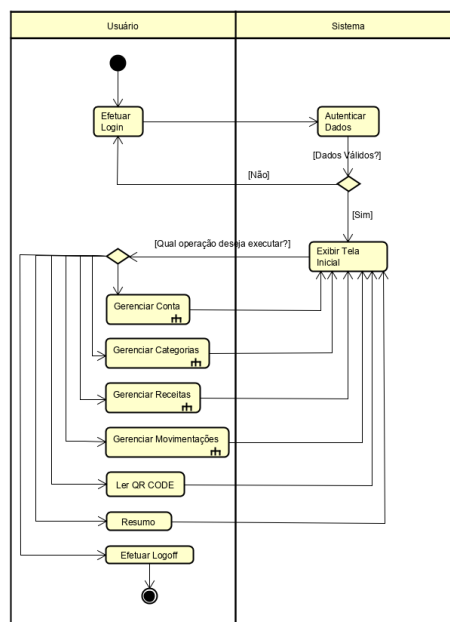
**Tabela 1. Planejamento por funcionalidades**

Ordem	Funcionalidade	Tempo de Desenvolvimento Estimado
1	RF1. Gerenciar Conta	20 dias
2	RF2. Gerenciar Categorias	15 dias
3	RF3. Gerenciar Movimentações	20 dias
4	RF4. Ler QR CODE	90 dias
5	RF5. Extrato Mensal	25 dias

## 5. Detalhar por funcionalidade

Esta etapa da metodologia tem por finalidade permitir que o fluxo de execução do sistema seja entendido, bem como as atividades que podem ser realizadas e o respectivo funcionamento das mesmas pelo sistema. [Heptagon, 2017]

Na Figura 4 é apresentado o Diagrama de Atividades sob a perspectiva do usuário final da aplicação.



**Figura 2. Diagrama de Atividade perspectiva usuário**

O Diagrama de Classes, conforme apresentado na Figura 6, representa as classes a serem implementadas no sistema, com o objetivo de ilustrar os objetos, bem como seus respectivos atributos e métodos.

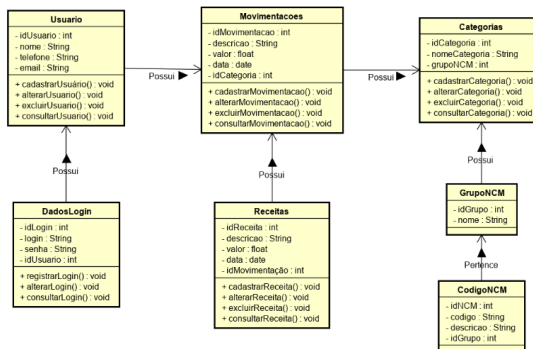


Figura 3. Diagrama de Classes

### 5.1. Protótipos de interfaces

Com o intuito de ilustrar a aplicação a ser desenvolvida, foram criados alguns protótipos de interfaces, para apresentar de forma aproximada, como será a experiência do usuário ao utilizar a ferramenta.

Na Figura 7, é apresentada a interface inicial, onde o usuário realiza o Login e é redirecionado para a tela principal da aplicação.

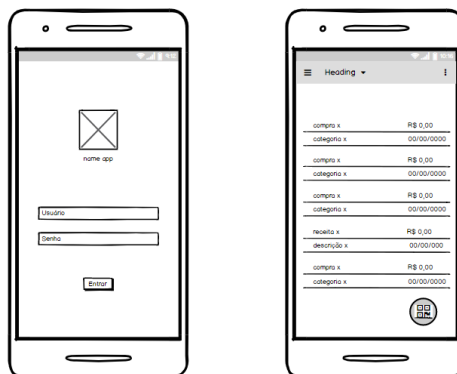


Figura 4. Interface de login e principal

Na interface apresentada através da Figura 8, é ilustrado a disposição das funcionalidades no menu da aplicação. Além disso, é ilustrado também a interface de resumo das movimentações, com a consolidação dos valores através de um gráfico.



Figura 5. Interface de menu e resumo

## 6. Implementação

O protótipo foi construído a fim de ilustrar a solução proposta. Para isso, iniciou-se configurando o projeto para comunicar-se com a base de dados definida para utilização, no caso o Firebase. Após testado o efetivo funcionamento e comunicação, prosseguiu-se com a implementação da tela inicial da aplicação na qual consiste em uma listagem de todos os produtos cadastrados na base de dados. Nessa listagem, os produtos apresentam sua respectiva descrição e preço total, além de mostrar a categoria em qual o mesmo se enquadra e a data efetiva de compra.

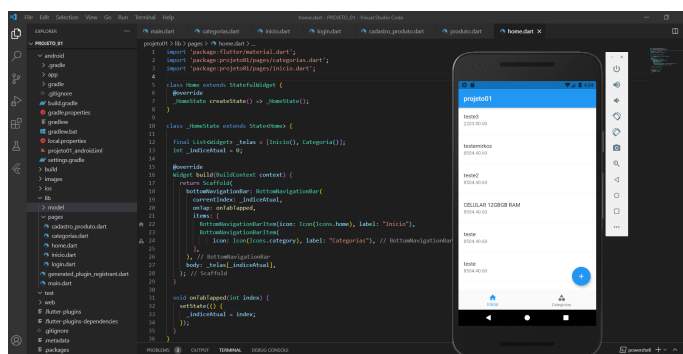


Figura 6. Ambiente de Desenvolvimento do Projeto

Com a tela inicial implementada, se deu início à codificação de uma tela de cadastro de produtos, para representar a inserção das informações, que por motivos de limitação de recursos tecnológicos, não foram possíveis de se automatizar conforme o planejamento original da solução.

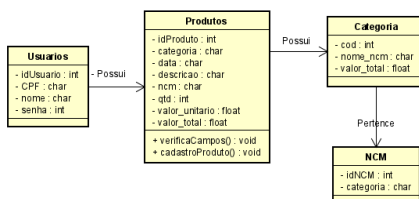


Figura 7. Diagrama de Classes Protótipo

Após implementar o cadastro de produtos, foi criada a tela de categorias, onde a aplicação vincula o NCM do produto cadastrado à respectiva categoria, para então consolidar no somatório total de todos os produtos cadastrados em suas categorias.

```
Run | Debug | Profile
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();

  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: Home(),
  )); // MaterialApp
}
```

Figura 8. Inicialização Firebase

Concluída a implementação das funcionalidades mencionadas, foi desenvolvido uma tela de extrato mensal, a fim de permitir que o usuário visualize o seu gasto total filtrando por mês.

Após todas a funcionalidades principais estarem criadas, se desenvolveu uma tela de acesso à aplicação para o usuário autenticar-se.

Como já mencionado anteriormente, foi utilizado o Firebase no qual tem sua inicialização declarada na classe *Main* da aplicação, conforme mostra a Figura 8.

Já na figura 9, podemos observar o trecho de código onde é realizada a busca das informações na base de dados e retornado para a aplicação através de uma lista com o conteúdo buscado, no caso a lista de produtos cadastrados.

```
body: StreamBuilder(
  stream: firebaseFirestore.instance.collection(
    "produtos").snapshots(),
  builder: (ctx, snap) {
    switch (snap.connectionState) {
      case ConnectionState.none:
        return Text("Sem dados");
      case ConnectionState.waiting:
        return CircularProgressIndicator();
      default:
        return ListView.builder(
          itemCount: (snap.data as QuerySnapshot).docs.length,
          itemBuilder: (ctx, index) {
            return Column(
              children: [
                ListTile(
                  subtitle: Text((snap.data as QuerySnapshot).docs[index]["ncm"].toString()),
                  title: Text((snap.data as QuerySnapshot).docs[index]["descricao"].toString()), // Listfile
                ),
              ],
            ); // Column
          }, // Listview builder
        ); // StreamBuilder // Scaffold
    }
  });
```

Figura 9. Leitura das informações da base de dados

Nas Figuras 10 e 11, pode visualizar-se a classe *cadastroProdutos*, bem como quando chamada para efetiva inserção das informações na base de dados.

```
213 cadastroProduto(Produto p) {
214   FirebaseFirestore.instance.collection("produtos").add({
215     "categoria": categoria,
216     "descricao": p.descricaoGet,
217     "valor": p.valorGet,
218     "valor_unitario": p.valorUnitGet,
219     "qtd": p.qtdGet,
220     "ncm": p.ncmGet
221   });
222   calculaValorTotalPorCartegoria(p.valorGet, p.ncmGet);
223 }
```

Figura 10. Classe para cadastro de produtos

```

if (verificaCampos() == true) {
    Produto p = Produto("", "", 0, 0, 0);
    p.ncmSet = controleNcm.text;
    p.descricaoSet = controleDescricao.text;
    p.valorSet = double.parse(
        converterDouble(controleValor.text.toString()));
    p.valorUnitSet = double.parse(converterDouble(
        controleValorUnitario.text.toString()));
    p.qtdSet = double.parse(
        converterDouble(controleQtd.text.toString()));
    cadastroProduto(p);
}

```

Figura 11. Chamada da classe de cadastro

Nas Figuras 12 e 13, pode visualizar-se a versão final da aplicação ilustrado pelas telas do mesmo.

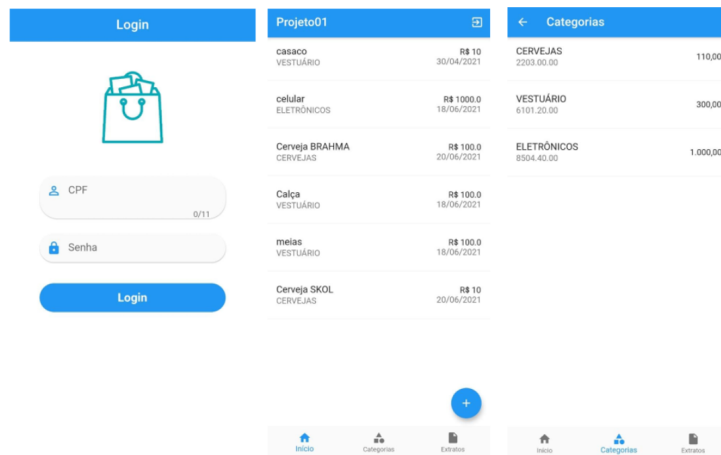


Figura 12. Telas da versão final da aplicação

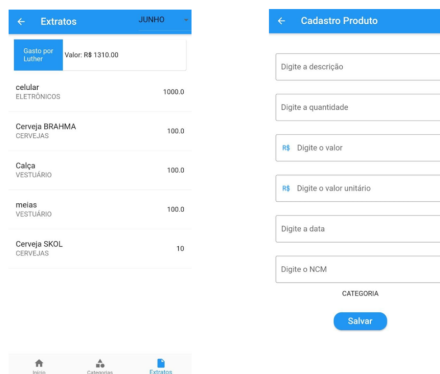


Figura 13. Telas da aplicação

## 7. Resultados

Durante o período/processo de pesquisa e levantamento das tecnologias às quais seriam necessárias para implementação do projeto proposto, foram surgindo dificuldades para viabilização da solução.

Inicialmente, o planejamento previa a utilização de recursos modernos que facilitariam e agilizariam todo o fluxo de execução da aplicação.

A ideia inicial era usufruir de um acesso à uma API que oferecesse a possibilidade de consumir dados anonimizados que compõe uma nova fiscal, tais como, produto, quantidade, valor unitário, NCM, entre outros.

Diante disso, iniciou-se um período de pesquisas com o intuito de identificar uma API que fornecesse tais dados, no entanto, sem sucesso.

Em busca de uma abordagem mais objetiva para a solução necessitada, optou-se por realizar um questionamento direto à instituição responsável pelo processamento dos dados de uma nota fiscal no estado do Rio Grande do Sul, no caso, a Procergs, que processa, inclusive, as Notas Fiscais de outros 13 Estados da Federação.

No contato realizado, foi informado por parte da Procergs, que a mesma não possui API para disponibilização das informações pretendidas. Como consequência dessa situação, optou-se então, pela busca de outras tecnologias que permitissem viabilizar o projeto.

Conforme conhecimento prévio, as Notas Fiscais ao Consumidor Eletrônicas (NFC-e), em seu modelo atual, tem por obrigatoriedade, desde abril/2016, a impressão de um QR CODE, no qual permite ao consumidor consultar instantaneamente informações completas sobre a compra que realizou, confirmando assim, a autenticidade do documento fiscal, bem como do seu respectivo registro no banco de dados da Receita Estadual, podendo também verificar de forma imediata a colocação do CPF na NFC-e, quando solicitado pelo próprio consumidor.

Essa consulta é viabilizada pelo fato de que a informação contida no QR CODE é a url que, ao acessá-la, redireciona o consumidor para a versão digital completa da NFC-e direto no navegador de internet do dispositivo em uso.

Visto a ocorrência da renderização dessas informações em uma página da internet, durante as pesquisas, encontrou-se algumas opções de tecnologias que realizam a captura de informações de um documento HTML. Diversas abordagens dessa tecnologia foram verificadas, no entanto, nenhuma se provou totalmente funcional durante as tentativas de implementações, que pelo fato de estarem disponíveis de forma pública na comunidade, acabam por necessitarem de complementações e adequações conforme o objetivo de sua aplicação, devendo assim, levar em conta todo o universo de soluções a serem integradas, no qual teve por consequência, muitos erros que inviabilizaram a implementação. Diante de todas essas intercorrências, decidiu-se junto ao orientador apresentar a solução de forma simulada à situação originalmente proposta, devido aos obstáculos tecnológicos intercorridos por tecnologias de terceiros.

Além dos obstáculos mencionados, durante a implementação do protótipo, também houveram dificuldades quanto da utilização do *Framework* escolhido, na qual ocorreram diversas inconsistências no ambiente de desenvolvimento, devido à configurações de plu-

gins, comunicação com a base de dados originalmente escolhida, que seria o SQLite, bem como inconsistências de versionamento de plugins e bibliotecas utilizadas, que foram contornadas por uma nova configuração de ambiente desde o início, realizando as devidas instalações e integrações após formatar a máquina utilizada.

Com isso, o propósito final desse projeto infere em uma abordagem sugestiva da solução original mediante a disponibilização de tecnologia adequada para o tratamento proposto das informações a serem objeto desse trabalho, por parte do detentor das mesmas, no caso a Procergs.

## **8. Considerações Finais**

Conforme as explicações no presente trabalho, o mesmo propõe o desenvolvimento e implementação de uma aplicação móvel para gestão e controle financeiro doméstico, afim de disponibilizar ao usuário final uma ferramenta que permita saber com precisão onde seus gastos estão sendo alocados.

Durante a realização das pesquisas sobre as ferramentas atualmente existentes, não identificou-se nenhum sistema de gestão financeira pessoal que realize o mesmo tratamento das informações pertinentes as movimentações de consumo dos usuários, conforme o que é proposto neste projeto, elevando o apelo à conveniência dessa solução.

A metodologia FDD foi escolhida levando em consideração a flexibilidade para com as mudanças que se façam necessária, tratando-se de uma metodologia ágil, sendo então, a melhor alternativa para o projeto, visto inclusive, a melhor qualidade de gestão e planejamento para o processo de desenvolvimento que a mesma permite.

## **9. Sugestões para o seguimento do trabalho**

Como sugestão para o seguimento do trabalho, entende-se que o ideal é obter acesso a base de dados da Secretaria da Fazenda via API, viabilizando assim, automatizar e por em prática a ação de leitura do QR Code de uma NFC-e, consumindo suas respectivas informações e então tratando-as com o objetivo de trazer clareza e transparência do controle financeiro doméstico, bem como agregando estatísticas e relatórios que possibilitem o pleno entendimento da situação financeira do usuário, permitindo ao mesmo, planejar e definir objetivos, baseado em informações precisas.

## **Referências**

- Aranha, Frederico. (2014). Feature Driven Development – Artigo 1 de 2. Disponível em: <https://sitecampus.com.br/feature-driven-development-artigo-1-de-2/> Acesso: Outubro/2019.
- Brandão, R. S. (2018). Sobcontrole-Sistema de controle de finanças pessoais. Repositório de Trabalhos de Conclusão de Curso e Monografias. Disponível em: <http://pensaracademico.unifacig.edu.br/index.php/repositoriottcc/article/viewFile/672/583> Acesso: Setembro/2019.
- Cidral, Beline. (2012). Afinal, o que é Android? Disponível em: <http://www.techtudo.com.br/artigos/noticia/2011/01/afinal-o-que-e-android.html> Acesso: Setembro/2019.
- Cremonesi, A. L. (2015). Myfinances: Aplicativo Móvel para o Gerenciamento de finanças Pessoais. Disponível em: <https://m.uniara.com.br/arquivos/file/cca/artigos/2015/andre-luiz-cremonesi.pdf> Acesso: Setembro/2019.
- Dart. (2019). Disponível em: <https://dart.dev/> Acesso: Outubro/2019.
- Harada, Eduardo. (2019). O que é o Android Studio, ferramenta criada para criar apps mobile. Disponível em: <https://www.tecmundo.com.br/software/146361-o-android-studio-ferramenta-criada-desenvolver-apps-mobile.htm> Acesso: Outubro/2019.
- Heptagon. (2017). Disponível em: <http://heptagon.com.br/fdd/> Acesso: Outubro/2019.
- Pimentel, Manoel. (2010). Fluxo de Processos da FDD. Disponível em: <https://visãoagil.wordpress.com/2010/02/25/fluxo-de-processos-da-fdd/> Acesso: Setembro/2019.
- Rico. (2018). O Guia Prático para um Planejamento Financeiro Pessoal Poderoso. Disponível em: <https://blog.rico.com.vc/planejamento-financeiro-pessoal-poderoso> Acesso: Setembro/2019.
- Rocha, Fábio. (2013). Introdução ao FDD – Feature Driven Development. Disponível em: <https://www.devmedia.com.br/introducao-ao-fdd-feature-driven-development/27971> Acesso: Outubro/2019.
- Sampaio, Julio. (2015). Primeiros Passos com a Google Dart. Disponível em: <https://www.devmedia.com.br/primeiros-passos-com-a-google-dart/32954> Acesso: Setembro/2019.
- Santana, Fabiano. (2019). Flutter: Porque Você Deveria Apostar nessa Tecnologia. Disponível em: <https://medium.com/tableless/flutter-porque-voc%C3%AA-deveria-apostar-nesta-tecnologia-94a510fffd18> Acesso: Outubro/2019.
- Souto, Mario. (2019). Conhecendo o Flutter, e uma Visão do Desenvolvimento Mobile. Disponível em: <https://blog.caelum.com.br/conhecendo-o-flutter-e-uma-visao-do-desenvolvimento-mobile/> Acesso: Outubro/2019.
- Veloso, Thássius. (2011). Dart, A Linguagem de Programação para Web do Google. Disponível em: <https://tecnoblog.net/79214/google-dart-linguagem/> Acesso: Outubro/2019.