

APLICAÇÃO DE CONCEITOS E PROCESSOS DE PROJETO DE INTERFACES: estudo de caso em aplicativo para gestão financeira pessoal

Vitor Araújo Oliveira
Graduando em Engenharia de Software – Uni-FACEF
vitor.araujo934@hotmail.com

Carlos Eduardo de França Roland
Mestre em Desenvolvimento Regional – Uni-FACEF
roland@facef.br

Resumo

A identidade visual de um aplicativo determina a sua concepção mental pelos usuários. Uma forte identidade visual pode ser determinada de muitas formas, sejam simples, minimalistas ou complexas e abrangentes. Porém, é necessário entender os conceitos estruturais de uma interface antes de determinar qualquer identidade visual. Este artigo apresenta os estudos realizados nos conceitos e fundamentos da disciplina de Interação Humano-Computador, na abordagem de projeto de experiência do usuário (UX/UI Design) explorando as possibilidades de estruturas de tela de um aplicativo para uso em dispositivos móveis para gerência financeira pessoal e familiar. Foi caracterizado o foco social do estudo em função das evidências do endividamento da população brasileira face à pequena oferta de formação neste tema. Foram executados os processos de projeto de software e de interface propostos pelas áreas de conhecimento da computação estudadas e buscou-se projetar a identidade visual do protótipo com um *design* simples e direto para que os usuários possam interagir e aprender facilmente, também levando em conta as definições de qualidade de software propostas pela ISO 9126. Uma versão inicial do protótipo funcional foi gerada e testes restritos de uso apontaram resultados satisfatórios e detalhes que podem melhorar ainda mais a solução em versões futuras.

Palavras-chave: Identidade Visual. Estruturas de Tela. Desenvolvimento Front-End. Análise de Telas.

Abstract

The visual identity of an application determines its mental conception by users. A strong visual identity can be determined in many ways, whether simple, minimalist or complex and comprehensive. However, it is necessary to understand the structural concepts of an interface before determining any visual identity. This article presents the studies carried out on the concepts and fundamentals of Human-Computer Interaction discipline, in the user experience design approach (UX / UI Design) exploring the possibilities of screen structures of an application for use on mobile devices for financial management personal and family. The social focus of the study was characterized due to the evidence of the indebtedness of the Brazilian population in view of the small offer of training on this topic. The software and interface design processes proposed by the computer knowledge areas studied were carried out and we sought to project the visual identity of the prototype with a simple and direct design so that users can interact and learn easily, also taking into account

the software quality definitions proposed by ISO 9126. An initial version of the functional prototype was generated and restricted usage tests showed satisfactory results and details that may further improve the solution in future versions.

Keywords: *Visual Identity. Screen Layout. Front-End Development. Screen analysis.*

1 Introdução

A gerência financeira pessoal e familiar é um problema conhecido entre os brasileiros, como mostrado nas pesquisas de Souza (2018). Ter acesso a ferramentas que possam facilitar o controle de receitas e despesas melhora a qualidade de vida da população e assim, oferecer um aplicativo para dispositivos móveis é a proposta deste estudo para uma possível solução dessa dificuldade.

Este artigo apresenta o projeto de Trabalho de Conclusão de Curso (TCC) que teve como objetivo desenvolver um aplicativo para dispositivos móveis para ser uma ferramenta de gestão que facilite o controle das receitas e despesas financeiras de seus usuários. Como objetivo secundário buscou-se aprofundar estudos sobre projeto de interfaces com vistas à oferta de melhor experiência do usuário com produtos de software.

Para o levantamento dos requisitos do sistema, foi realizada uma pesquisa com potenciais usuários de aplicativos para registro e acompanhamento de movimentos financeiros e ficou evidenciado que o tipo mais requisitado de ferramenta para este fim são aplicações para dispositivos móveis e que ofereçam integrações com aplicações para *desktops*. Durante o desenvolvimento do projeto o foco principal foi na aplicação do paradigma de projeto *mobile first* que visa definir as estruturas arquitetônicas e lógicas do sistema para uso em dispositivos móveis e apenas depois adequá-las para outras plataformas como *web* e *local desktop*.

No desenvolvimento foram utilizadas as linguagens de programação Dart, através do *framework* Flutter, e Python para a implementação de Application Programming Interfaces (APIs). O módulo de *back end* é suportado por banco de dados não relacional para facilitar a utilização do aplicativo a longo prazo.

Para o melhor desenvolvimento orgânico do aplicativo foram aplicados conceitos, métodos e padrões de Interação Humano-Computador (IHC), mais especificamente aspectos de UX Design, que preconiza o foco do projeto na experiência do usuário. Buscou-se estruturar a interface para uma melhor ergonomia de uso, desta forma reduzindo quaisquer resistências à utilização do aplicativo.

Neste contexto, apresenta-se os fundamentos teóricos da Engenharia de Software que nortearam o projeto e o desenvolvimento do aplicativo, sua documentação e os paradigmas que foram adotados.

2 Fundamentação Teórica do Projeto

Nesta seção são apresentadas as justificativas para a adoção do tema e da questão problema de pesquisa que fundamentaram o projeto de TCC, as definições dos elementos estudados, especificações das linguagens de programação e ferramentas utilizadas para o desenvolvimento do protótipo funcional do sistema.

2.1 Gestão Financeira Pessoal e Familiar

Existe evidência que a percepção da qualidade de vida do indivíduo ou das famílias está ligada às suas finanças (ACORDI, 2019), e a maior parte dos brasileiros pagou pelo menos uma conta em atraso como mostra a Figura 1 (SPC, 2016).

Figura 1 - Inadimplência de brasileiros



Fonte: SPC (2016)

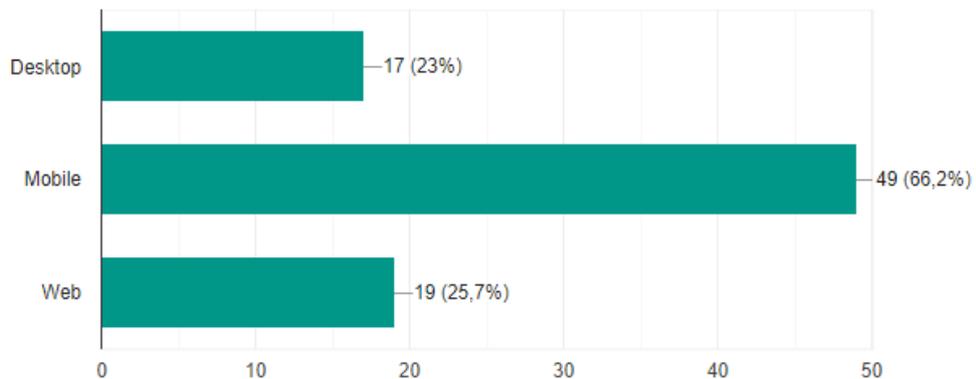
Procurando ser uma solução para os indivíduos e as famílias que têm problemas com a gerência financeira e possivelmente melhorar a qualidade de vida destas, decidiu-se desenvolver um aplicativo *mobile* que resolvesse a questão de pesquisa: como é possível simplificar e viabilizar a gestão financeira pessoal e familiar com um aplicativo fácil de ser usado?

A escolha de se desenvolver uma aplicação *mobile* foi definida por dois motivos: o primeiro foi seguir as tendências internacionais de projeto *mobile first*, e o segundo se deu por causa da pesquisa *online* feita para o levantamento de requisitos com potenciais usuários. Compilando as respostas recebidas, percebeu-se que a preferência dos pesquisados foi pelo formato de aplicações para dispositivos móveis como apresentado na Figura 2.

Figura 2 – Respostas obtidas sobre a arquitetura mais indicada

Quais formatos seriam melhores para uma aplicação de gerência financeira?

74 respostas



Fonte: os autores

Como é possível ver, na pesquisa é clara a preferência pela aplicação *mobile*, superando em mais de duas vezes a preferência por qualquer das outras formas de aplicativos.

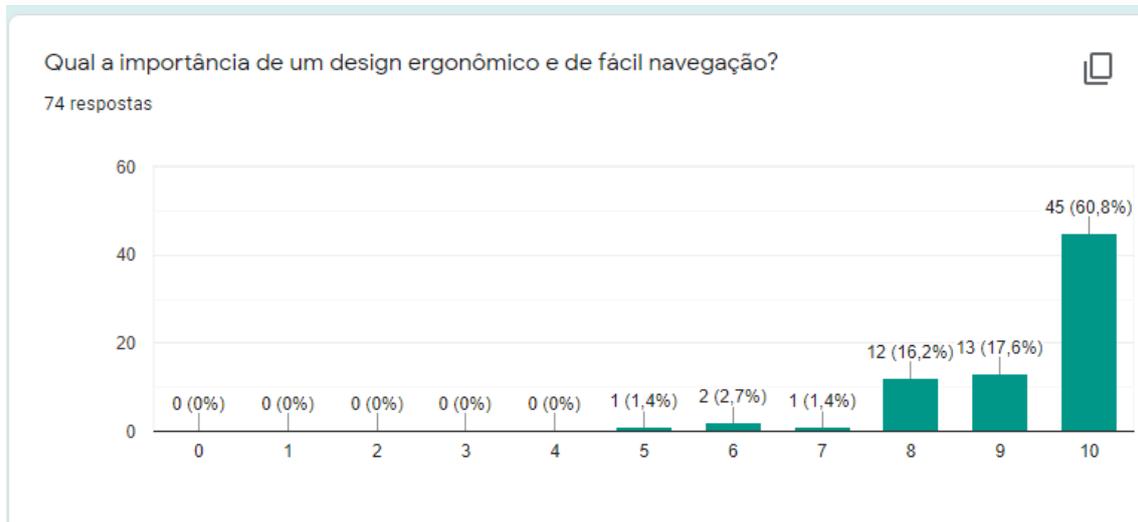
Para atender a este resultado foi decidido usar, para o desenvolvimento do módulo de *front end*, o *framework* Flutter com linguagem de programação Dart, e para o *back end* foi adotada a linguagem de programação Python por preferência e domínio dos autores.

Para o armazenamento dos dados da aplicação foi adotado o Sistema Gerenciador de Bancos de Dados (SGBD) de modelo não relacional para garantir o melhor desempenho de resposta possível.

Para garantir que a aplicação tenha a melhor ergonomia possível utilizaram-se os dados e análises feitas no estudo sobre IHC e UX Design para garantir a usabilidade e a navegabilidade simples pelas funções do aplicativo.

Foram executados os processos de prototipação das telas desde os rabiscoframes, passando pelos *wireframes*, até a prototipação em alta resolução, como serão mostradas na seção de apresentação do projeto. Tais cuidados foram reforçados pelos resultados apontados pela pesquisa sobre a importância dessa característica em aplicativos digitais, como mostrado na Figura 3.

Figura 3 – Importância do *design* ergonômico

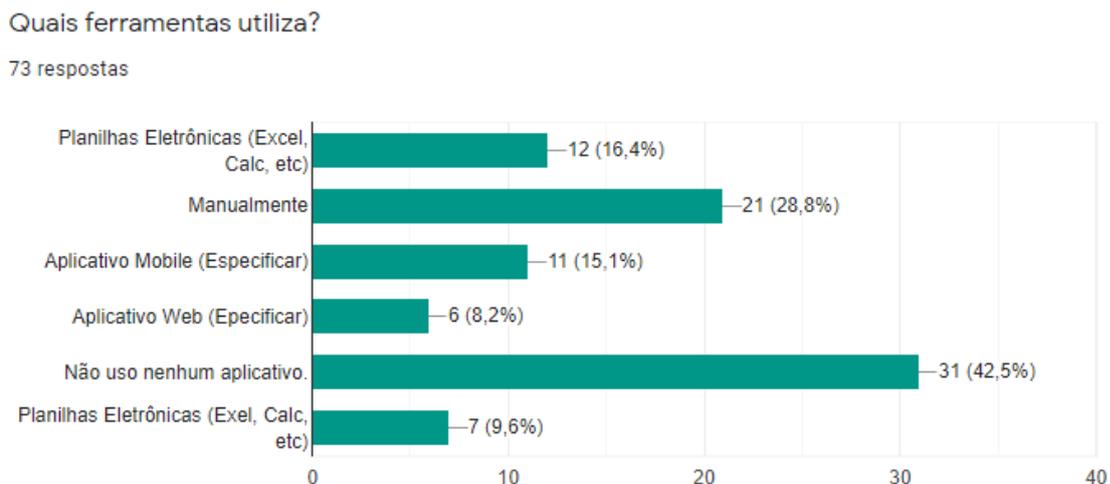


Fonte: os autores

Através da análise de telas procura-se proporcionar uma melhoria da experiência do usuário e desta forma reduzir a resistência à utilização do software.

A pesquisa evidenciou também que 71,3% dos respondentes realiza controles manuais ou não realiza. Tal situação é mostrada na Figura 4.

Figura 4 – Como realizam os controles financeiros



Fonte: os autores

O projeto possui também um objetivo social, tendo sido sua arquitetura feita para alcançar o maior número de pessoas possível gratuitamente. Com uma estrutura simples, o aplicativo pode ser uma forma de facilitar a gerência financeira de usuários, gratuitamente, sem anúncios ou disseminação de mensagens para que se comprem versões mais completas do sistema.

Todas as funções do aplicativo devem ser disponibilizadas para os interessados gratuitamente e livre de quaisquer formas de anúncios. Todos os dados dos usuários serão de total controle dos mesmos, como de acordo com a Lei Geral de Proteção de Dados (LGPD) recentemente implantada no Brasil.

2.2 Projeto com Foco em *Mobile First*

Mobile first é uma metodologia de desenvolvimento de software que foca no desenvolvimento primariamente para interfaces de dispositivos móveis e, apenas depois, sua adaptação para as outras plataformas como *web* e *desktop* (GOBACKLOG, 2019).

Segundo Antunes (2019) o paradigma *mobile first* melhora a experiência geral do usuário, reduz gastos para a manutenção do *website*, melhora o desempenho de carregamento de telas e aumenta a visibilidade e o alcance da aplicação. A adoção do foco em *mobile first* demonstra ser o mais efetivo para atingir os resultados esperados para o projeto.

2.3 Fundamentos de projeto de interface

O primeiro contato do usuário com qualquer software é sua interface visual, como tal, para uma primeira impressão, a interface visual deve ser representativa das funcionalidades da aplicação e, ao mesmo tempo, estruturar a visão do usuário de forma simples para que a usabilidade seja clara à primeira vista.

Por isso as análises dos conceitos básicos que serão feitos a seguir foram aplicadas no protótipo de formas diversas a partir da estrutura das telas.

2.3.1 Tipos de informações em Interfaces

Existem inúmeras formas pelas quais a informação pode ser transmitida, seja ela uma imagem, um vídeo ou um áudio, e as pessoas reagem de formas diferentes para cada uma destas informações.

No processo de análise e projeto de interfaces para comunicação por hipermídia, as informações são classificadas como: estáticas, dinâmicas, sonoras e ou ainda *links* e outras linguagens associativas (PASSOS, 2008).

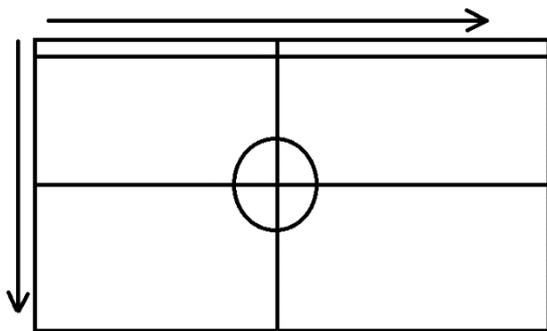
- **Informações Estáticas:** são informações visuais que não tem nenhum tipo de movimento tais como imagens, logos, emojis, etc., e as informações textuais.
- **Informações Dinâmicas:** são as que compõe as informações visuais de acompanhamento do usuário, apresentam algum tipo de movimentação, podendo ou não ser atreladas a informações sonoras (videos, *gifs* animados, *streamings*, etc.)

- **Informações Sonoras:** são informações que compõe a percepção sonora do usuário podem ou não ser atreladas às dinâmicas (músicas, *podcasts*, gravações de reuniões, etc.)

2.3.2 Sentido de Leitura

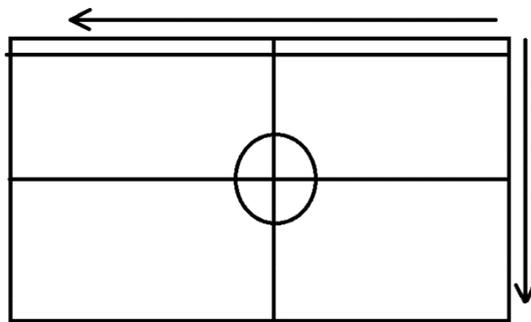
O sentido de leitura orienta a divisão das telas para inserção de conteúdo. A forma pela qual a leitura é realizada define a importância das regiões da tela. As Figuras 5 e 6 destacam as principais orientações de leitura no mundo.

Figura 5 - Sentido de Leitura Ocidental



Fonte: os autores

Figura 6 - Sentido de Leitura Oriental



Fonte: os autores

A ideia do sentido de leitura se aplica diferentemente a informações estáticas e a informações dinâmicas. Enquanto observando informações estáticas o usuário tenderá a seguir o sentido de leitura que está acostumado. Porém ao seguir informações dinâmicas o foco estará sempre nas partes centrais da tela ou onde a informação dinâmica é mostrada. O sentido de leitura é importante para definir também os conceitos de costume e importância para projeto de aplicativos.

2.3.3 Padrões de importância

Durante anos as pessoas foram treinadas pelo uso da internet, e como tal, algumas estruturas se tornaram comuns e quase padrão. Dessa forma o conceito de importância das informações se apresenta.

Informações ou itens mais importantes ou mais utilizados devem seguir o fluxo padrão adotado nos projetos de interfaces durante os anos de utilização de aparelhos digitais e navegação na internet como o fluxo mostrado na Figura 7.

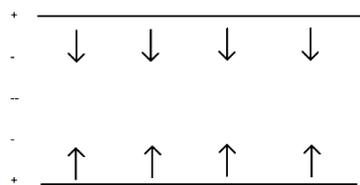
Os padrões de importância devem ditar a orientação do *design* da interface de forma a definir estruturalmente sua funcionalidade. Tais padrões se aplicam diferentemente a informações estáticas e mutáveis. Para informações estáticas o sentido de importância é o mesmo sentido de leitura, portanto as informações mais importantes e/ou mais utilizadas devem ser as primeiras a serem disponibilizadas. Para informações

mutáveis o sentido de importância indica o fluxo de notificações e novas interações, por exemplo, em *feeds* de notícias.

Em ambos os casos o sentido de importância deve ser considerado de acordo com o sentido de leitura em conjunto. Perguntar-se como o usuário irá chegar a este ponto da interface é uma boa prática de projeto.

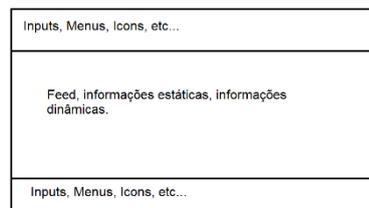
O sentido de leitura é utilizado também para definir a estrutura geral de cada tela como mostra a Figura 8. Estruturalmente este conceito sugere que as aplicações devam seguir a distribuição mostrada, associando os índices de importância da informação com os mais utilizados localizados nas regiões demarcadas como Inputs, Menus, Ícone, etc., enquanto as informações de leitura e visualização se localizam ao centro da tela. A adoção dessas orientações resultaram na divisão de áreas da interface do aplicativo deste projeto como a mostrada na Figura 9.

Figura 7 – Orientações de importância do software



Fonte: os autores

Figura 8 – Divisão de estrutura básica



Fonte: os autores

Figura 9 – Divisão de estrutura por leitura e Importância

Ícones Entradas	Inputs, Menus, Abas, Ícones	Ícones entradas
Menus	Feed Visual Informações	Menus
Ícones	Inputs, Mensagens, Menus, Footers.	Ícones

Fonte: os autores

2.3.4 Usabilidade e acessibilidade

Atualmente acessibilidade é uma necessidade. Existem diversas condições médicas, psicológicas e sociais que podem causar impedimentos durante a utilização de aplicações. Desta forma aplicações devem manter estruturas gerais de acessibilidade e facilidade.

Para tal existem algumas situações às quais o desenvolvedor deve atentar. Existem de fato inúmeras condições que podem causar dificuldades na utilização de um aplicativo, sendo as mais comuns a serem observadas:

- Idade
- Nível de escolaridade
- Dislexia
- Daltonismo
- Deficiência visual
- Deficiência auditiva

A relação com as condições sociais e físicas dos usuários é essencial para manter uma aplicação de grande abrangência. Para lidar com diferentes níveis de acessibilidade são necessárias diferentes formas de se apresentar as informações. É recomendado que, na função de Configurações do aplicativo se tenha pelo menos a

possibilidade de se configurar a interface para deficiência visual, dislexia e daltonismo, pois estas são as menos complexas para implementação. Ferramentas de software para *design* gráfico aceitam *plugins* que adicionam recursos que ajudam a projetar *layouts* com melhor acessibilidade (MAR, 2020). Em casos extremos como por exemplo de aplicativos para uso por cegos, a abordagem deve ser mais radical, buscando-se recursos sonoros para possibilitar o uso do sistema.

As deficiências auditivas devem ser consideradas quando parte considerável de seu conteúdo tem formato de áudio. Oferecer funções de apresentação de legendas ou outras formas distintas de áudio pode ser essencial para incluir pessoas portadoras de deficiência auditiva.

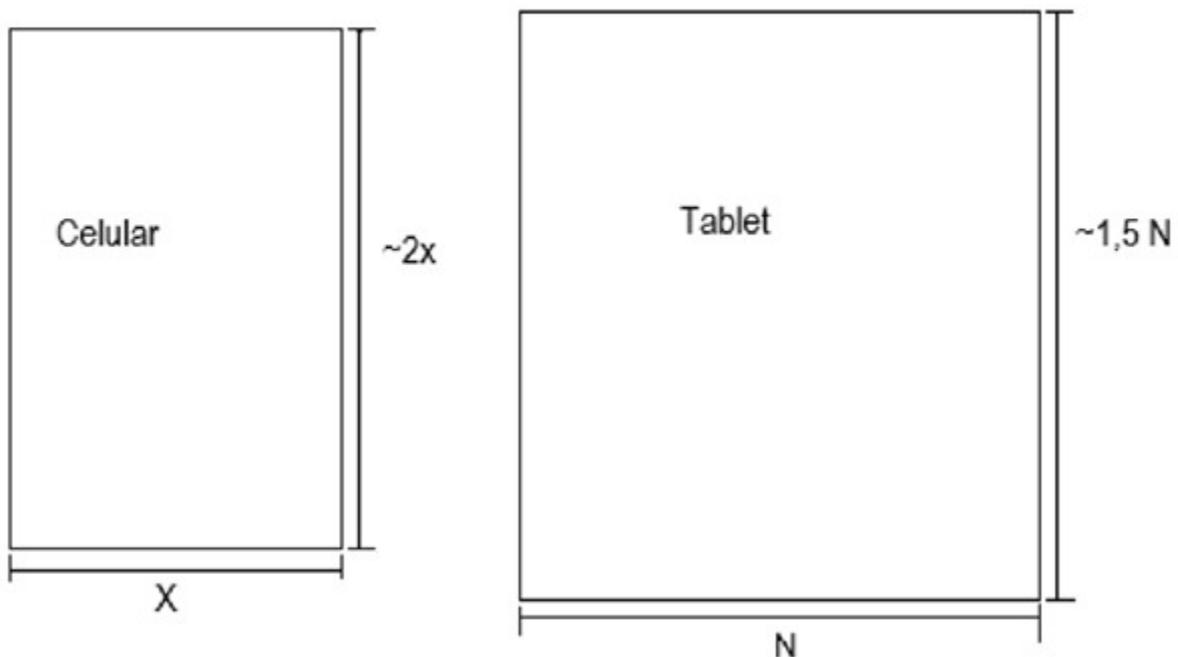
2.3.5 Interfaces para Dispositivos Móveis

Aplicando os conceitos apresentados especificamente para uso por dispositivos móveis algumas observações específicas são apresentadas.

Mobile geralmente se aplica em duas formas diferentes de dispositivos: *smartphones* e *tablets*. Essas diferenças são relevantes não só na implementação do código, mas principalmente pelas proporções da interface e o espaço disponível para apresentação de dados e informações.

Pela Figura 10 é possível visualizar as proporções aproximadas de uma tela de celulares e uma tela de *tablets*. As dimensões X e N podem mudar de acordo com o aparelho, mas as proporções tendem a permanecer semelhantes.

Figura 10 - Proporções Celular e Tablet



Fonte: os autores

Conhecendo as proporções das áreas de trabalho e aplicando os conceitos anteriores chega-se às seguintes observações sobre a estrutura de interfaces *mobile*.

Aplicando os sentidos de leitura e organização da tela ao *mobile* tem-se 9 divisões que seguem o sentido de leitura e importância vistos:

- 1 – Logos e Acesso ao menu
- 2 – Nome da tela, menus e nome do aplicativo
- 3 – Logos e Acesso ao menu
- 4 – Entradas menu minimalista ou expansão de 5
- 5 – Demonstração de dados
- 6 – Entradas menu minimalista ou expansão de 5
- 7 – Logos, botões de menu ou expansão de 8
- 8 – Ícones de menu
- 9 – Logos, botões de menu ou expansão de 8

As correspondências dessas divisões em telas são apresentadas na Figura 11. Tudo isso se aplica também ao conceito do tamanho dos componentes clicáveis nas telas do aplicativo. Botões muito pequenos podem causar frustração no usuário enquanto botões muito grandes quebram as proporções e podem parecer estranhos. Cabe ao desenvolvedor aplicar a melhor estruturação possível para a interface do aplicativo em desenvolvimento.

2.3.6 Estruturas de Interface *Mobile* Adotada

As Figuras 12 e 13 mostram a estruturação adotada para a interface do aplicativo e de seu menu principal, respectivamente. Vê-se a estrutura adotada para a interface da aplicação, seguindo os conceitos estudados onde a área 1 é para acesso ao menu, a 2 é a para informação de página, a 3 é para conteúdo de página e a 4 para menu e botões de *footer* (Figura 12). A estruturação do menu foi definida da forma mais simples possível com o *header* definido na área 1 e os itens de menu para navegação na 2 (Figura 13).

2.4 Infraestrutura de Desenvolvimento

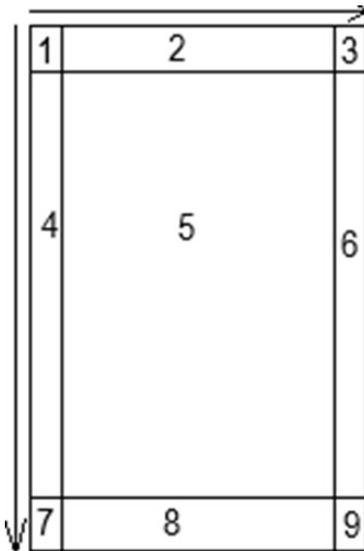
Nesta seção são apresentadas as ferramentas adotadas como infraestrutura de desenvolvimento do protótipo funcional do aplicativo.

Para o desenvolvimento do código do *front end* do projeto foi adotada a plataforma Flutter que é um kit de desenvolvimento de interface de usuário criado pelo Google e possui código aberto. Este permite a criação de aplicativos compilados nativamente nos sistemas Android e iOS.

Segundo Flutterdev (*sd online*), Flutter é um *framework* de acesso grátis e de código aberto para desenvolvedores que desejam unificar o desenvolvimento *web*, *mobile* e *desktop* com um único investimento. Pode ser usado também por *designers* que desejam mostrar seus projetos em alta fidelidade. Devido sua fácil modularização e sua

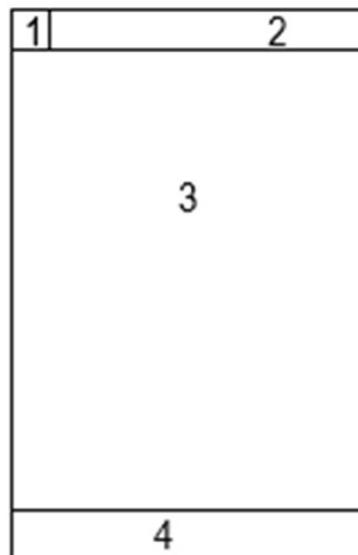
estruturação direta o *framework* foi escolhido para ser usado no desenvolvimento do aplicativo *mobile*.

Figura 11 - Estruturas Prévias no mobile



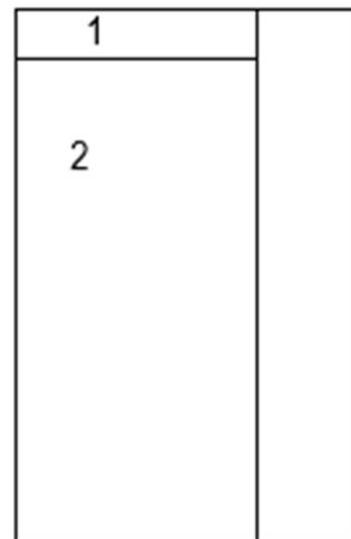
Fonte: os autores

Figura 12 - Estruturação geral



Fonte: os autores

Figura 13 - Estruturação Menu



Fonte: os autores

Para o *back end* do sistema, foi adotada a linguagem de programação Python que é uma linguagem de programação interpretada, dinâmica e de alto nível. Segundo Lima (2019) a linguagem foi criada no final da década de 80 por Guido Van Rossum. Este pesquisador desenvolveu a linguagem pelo paradigma de melhoria constante.

Python é ser uma linguagem de programação de sintaxe simples, com orientação a objeto e com recursos para *design* gráfico, embora as opções de *design* da linguagem sejam limitadas. Devido à simplicidade da linguagem e a abundância de bibliotecas disponíveis ela foi escolhida para implementar a comunicação entre a aplicação e o servidor de serviços de armazenamento de dados. O Sistema Gerenciador de Bancos de Dados (SGBD) adotado foi de arquitetura NoSQL em função das características do projeto.

SGBDs não relacionais ou NoSQL são bancos de dados baseados em uma estrutura JSON e, segundo Fernandes (2017) são bem variados e utilizados por diversas empresas para, principalmente, aplicações de Big Data devido a sua versatilidade.

Estruturar uma aplicação sobre um banco não relacional tem suas diferenças, vantagens e desvantagens e devem ser avaliadas de acordo com a necessidade da aplicação. Como menciona Oliveira (2014), as aplicações rodando em bancos de dados não relacionais têm uma grande melhora em performance, porém não foram criados para substituir SGBDs relacionais.

Para a operação da aplicação foi escolhido o Sembast para a persistência de dados no *front end* e o MongoDB para armazenar os dados no *back end*. Sembast é um banco de dados não relacional para Flutter, o qual pode ser facilmente adicionado a qualquer projeto, e MongoDB é uma plataforma gratuita de armazenamento de dados NoSQL.

A utilização de bancos não relacionais foi decidida pelo fato de que, idealmente, usuários vão utilizar a aplicação durante muito tempo, o volume de dados deverá crescer e manter o desempenho e a segurança dos dados da aplicação com facilidade de escalabilidade é essencial.

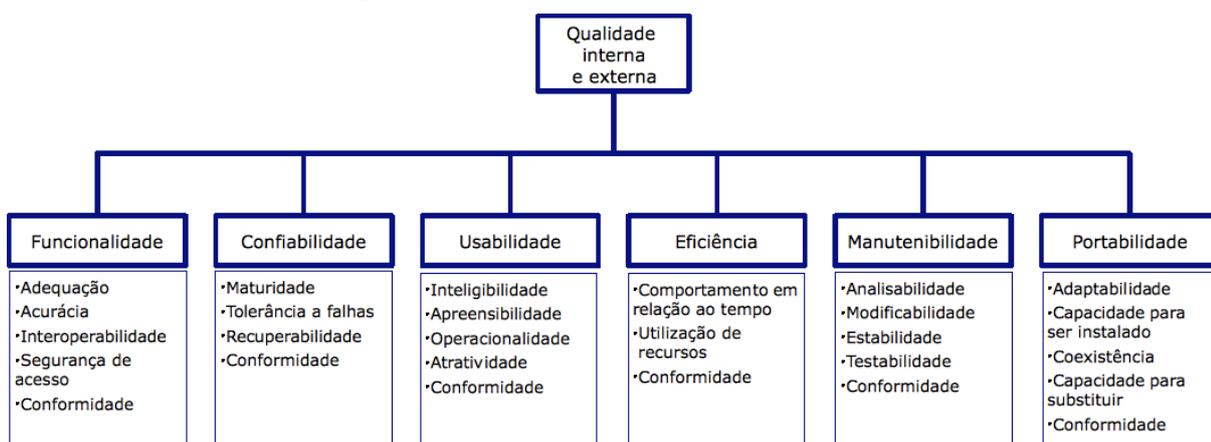
3 Qualidade de Software e Integração de Aplicações

Pelo que foi apresentado é possível se considerar que o foco na experiência do usuário deva ser um fator determinante de qualidade do software.

A Qualidade de Software nesta aplicação foi tratada como uma maneira de garantir a melhor experiência possível para o usuário no uso do aplicativo. Tal experiência é essencial para manter o usuário utilizando o produto. O engajamento do usuário com o produto proporciona sua viabilidade tanto operacional quanto financeira e software de alta qualidade minimiza a resistência do usuário utilizador.

Segundo a ABNT (2003) que publica a norma ISO 9126/NBR 13596, Qualidade de Software possui algumas características: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade. Detalhes de cada uma são mostrados na Figura 14.

Figura 14 - Características da Qualidade de Software



Fonte: ABNT (2003)

Durante o desenvolvimento da aplicação o foco principal foi nas características da interface do software, assim foram tratados no projeto os aspectos de Usabilidade:

- Inteligibilidade: capacidade do software de ser compreendido pelo usuário e se o mesmo pode atender suas expectativas.

- Apreensibilidade: capacidade do software de ser aprendido pelo usuário.
- Operacionalidade: facilidade do usuário de operar e controlar as funcionalidades do software.
- Estética/Atratividade: quão atrativo são os visuais do software, o quão forte é sua identidade visual.
- Acessibilidade: prática de tornar o software acessível ao maior número de pessoas possíveis, incluindo pessoas portadoras de deficiências.
- Conformidade: capacidade do software de se apresentar em padrões estabelecidos.

E os de Eficiência:

- Comportamento em Relação ao tempo: eficiência de resposta do aplicativo em relação ao tempo.
- Utilização de Recursos: medidas de utilização de hardware que o software requer.
- Conformidade: capacidade do software de se apresentar em padrões estabelecidos.

Contextualizando a aplicação destes conceitos ao protótipo desenvolvido, buscou-se estabelecer uma boa interação visual com o usuário. E, para isso, as características de usabilidade foram aplicadas em maior ou menor nível. Avaliações de acessibilidade do protótipo não foram realizadas nas primeiras versões devido à necessidade da participação de usuários testadores para auxiliar sua realização. Desta forma a acessibilidade completa deve ser implementada com estudos futuros.

4 Projeto e Implementação

Para o projeto do aplicativo foram realizados os processos e aplicados os métodos propostos pela Engenharia de Software. Foi percorrido o Ciclo de Vida do Desenvolvimento de Software (CVDS) com os processos: levantamento e documentação e análise de Requisitos, projeto baseado em diagramas como o BPMN e os UML, além de modelagem e prototipação de bancos de dados.

Como o foco do trabalho de TCC foi aprofundar os conhecimentos sobre conceitos de Interação Humano-Computador, na fase de prototipação da interface do protótipo, foi dada especial atenção aos aspectos da experiência do usuário.

4.1 UX/UI Design

A área de conhecimento da computação que trata aspectos de sucesso de produtos pela aceitação por usuários tem sido designada UX/UI Design. Neste contexto considera-se que o objetivo principal de todo produto e serviço é resolver problemas das pessoas. Como um produto os resolve pode fazer a pessoa se sentir tanto mais satisfeita e exaltada quanto frustrada e desapontada. O propósito do projeto baseado na

experiencia do usuário (UX Design do termo em inglês *User Experience Design*) é fazer as pessoas se sentirem bem usando produtos que facilitem suas vidas (PALMER, 2020).

Experiência de usuário é sobre sentimentos das pessoas. Assim, os projetistas de experiência do usuário que desenvolvam uma alta inteligência emocional têm mais facilidade de prever como seus produtos farão as pessoas se sentirem. Deve-se antecipar e prever com precisão onde e como o produto será usado, e se ele resolverá ou não os problemas dos usuários. O trabalho de um UX Designer implica na compreensão das necessidades dos usuários, suas motivações, frustrações e objetivos. É necessário se antecipar e prever as variáveis relacionadas aos problemas dos usuários: o que, porque, quem, onde, quando e como (5W1H dos termos em inglês *What, Why, Who, When, Where e How*). Contextos físicos de uso definem como pessoas encontram e usam a informação presente em aplicativos digitais. Contextos ambientais, emocionais e cognitivos de utilização por usuários devem orientar as soluções de projetos de interfaces (PALMER, 2020).

Os processos sugeridos pela UX Design compreendem a criação de um *briefing* do projeto onde são alinhadas as necessidades e expectativas dos envolvidos na solução do problema, seguido da definição do plano de ação 5W1H. O passo seguinte é a caracterização das personas usuárias do produto, seguido da listagem de suposições acerca das possíveis soluções do problema, formalização de hipóteses de solução, priorização delas e a validação da escolhida junto aos potenciais usuários do produto. Para consolidar a proposta de projeto, passa-se à realização de *benchmark* de produtos concorrentes para então se desenvolver as jornadas de usuários que resultarão no primeiro esboço, em baixa resolução, da interface do aplicativo, chamado rabiscoframes.

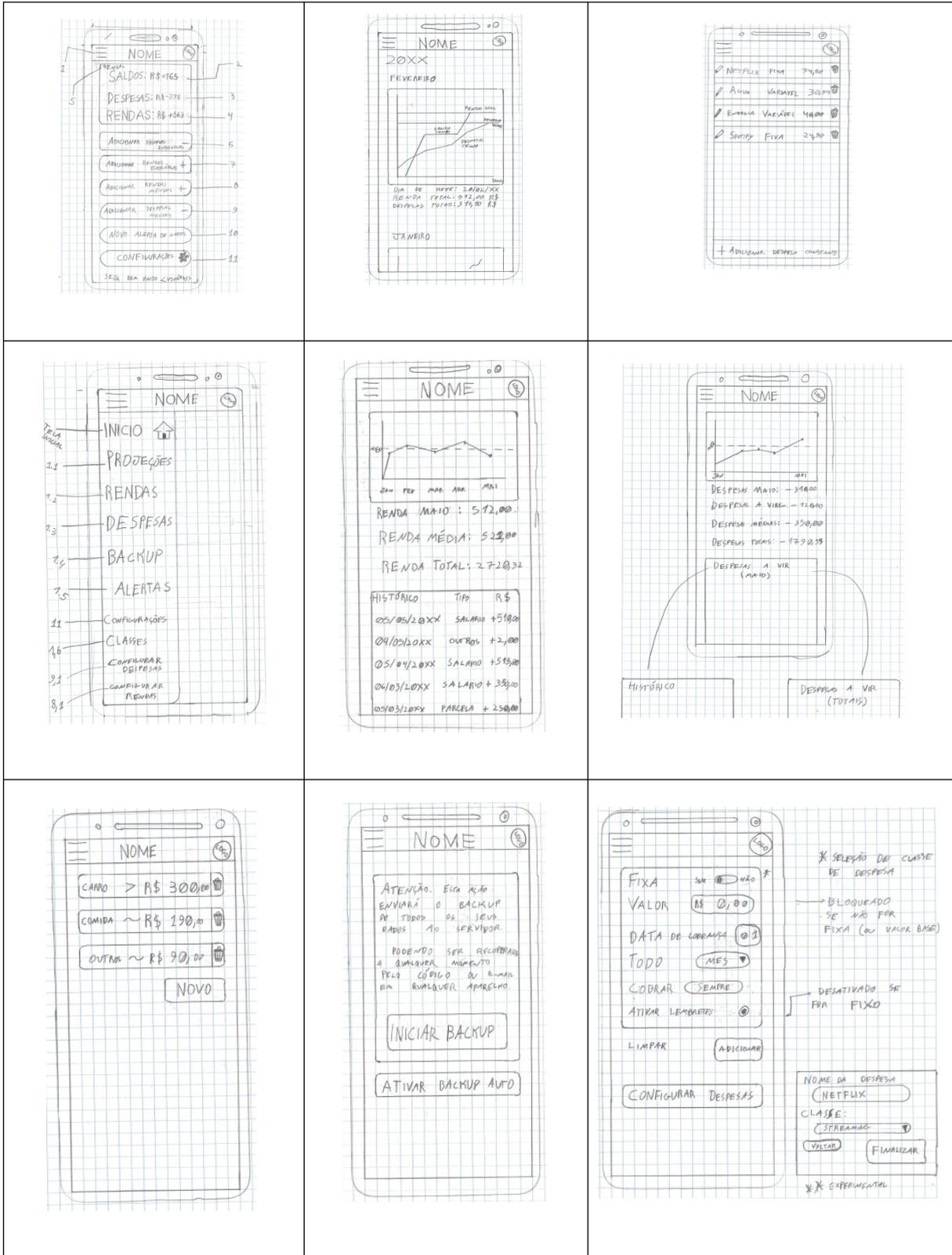
A partir de nova rodada de validação com usuários são criados os *wireframes* que representam a adequação dos rabiscoframes aos *feedbacks* recebidos, num nível melhor de resolução agora incorporando aspectos de navegabilidade da interface. Nesse momento ainda não se preocupa com detalhes gráficos. O passo seguinte é a definição dos parâmetros gráficos que nortearão a implementação da interface com a geração do *styleguide* onde são especificados os padrões de cores (paleta), tipografia, forma dos elementos (botões, campos de formulários, ícones, etc.), a navegabilidade e a usabilidade da interface, bem como estruturas de apresentação de textos, menus, formulários, e aspectos de tempo de resposta das funcionalidades (BABICH, 2020).

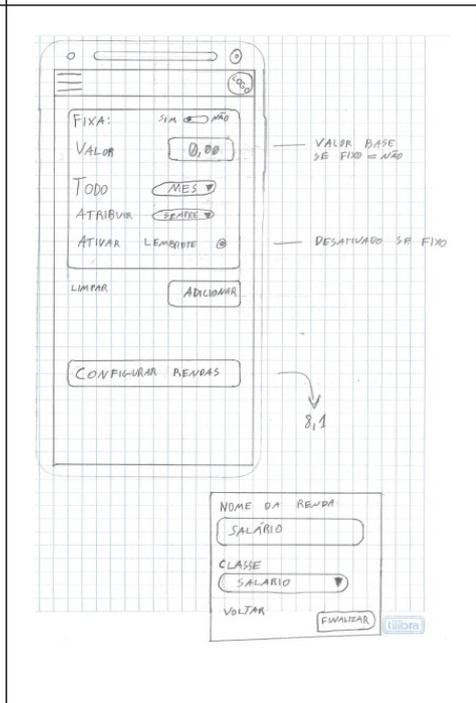
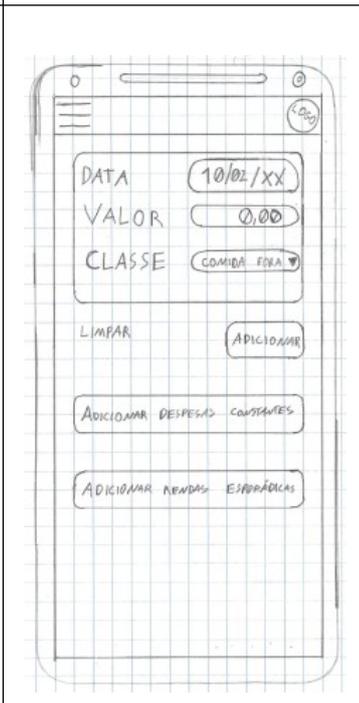
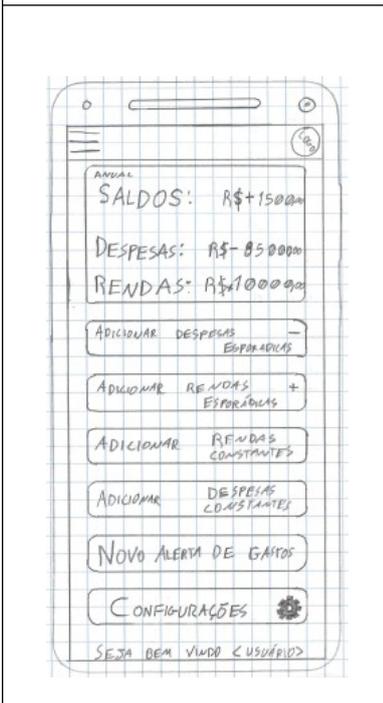
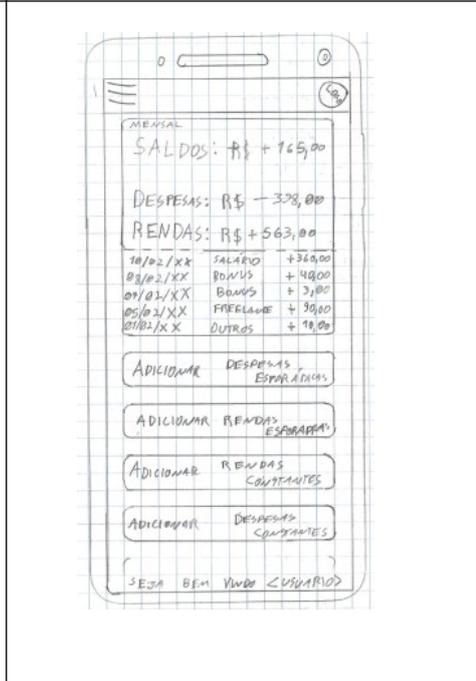
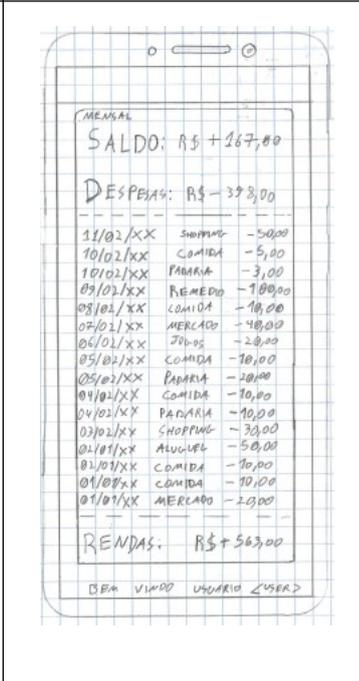
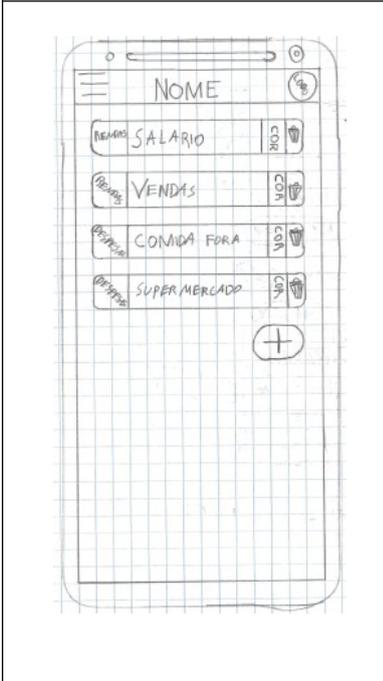
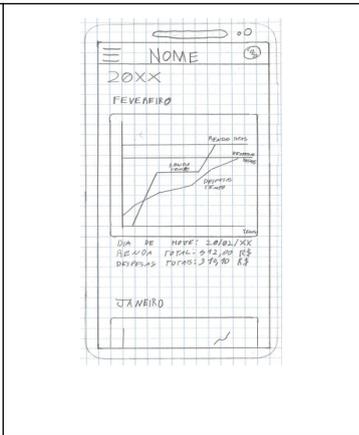
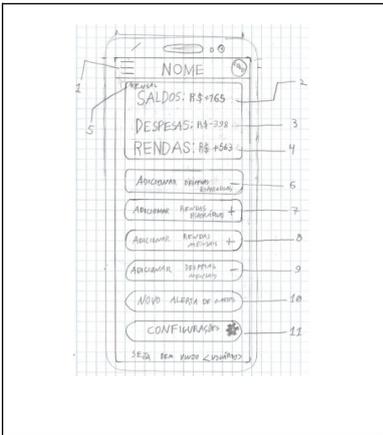
4.2 Rabiscoframes

O Quadro 1 apresenta os estudos da interface realizados por rabiscoframes.

Usando papel quadriculado e lápis, o projetista esboça as primeiras ideias de como os usuários irão interagir com a solução, com o objetivo de ter uma visão inicial, básica, do fluxo das tarefas que são realizadas nos processos envolvidos no problema de projeto. A partir desse exercício, novas percepções são geradas evidenciando detalhes que no levantamento inicial não foram percebidos.

Quadro 1 – Estudos de interface realizados por Rabiscoframes





Fonte: os autores

4.2 Controle de Versionamento e Hospedagem do protótipo

A documentação do sistema pode ser encontrada no repositório: GitHub Projeto de Aplicação de gerência financeira. 0.1.0. GitHub, 2020. Disponível em: <<https://github.com/VitorAraujoOliveira/TCC>>.

5 Resultados da Implementação

O protótipo funcional foi implementado e detalhes de projeto foram adequados em função das ferramentas de desenvolvimento adotadas.

O Quadro 2 apresenta algumas das telas do aplicativo.

O protótipo produzido alcançou uma estrutura visual satisfatória, porém ainda foram notadas possibilidades de expansão e melhorias. Para uma utilização mais abrangente do sistema deve-se oferecer a possibilidade de acesso do *front end* por navegadores web operados em *notebooks* ou *desktops*. Percebeu-se também a possibilidade de integração com APIs de outros sistemas como os bancários e aplicativos de pagamentos para troca de dados digitais, minimizando a necessidade de digitação de operações.

Uma vez que estas expansões primárias forem disponibilizadas a manutenção geral da aplicação será necessária e, para sua viabilidade operacional é recomendado que seja o mais minimalista possível consumindo a menor quantidade possível de recursos de hospedagem em servidor.

6 Conclusão

Estabelecer uma boa imagem visual pode ser um desafio e tanto para desenvolvedores. Neste estudo foram aplicados conceitos e fundamentos de Engenharia de Software e de Interação Humano-Computador, na abordagem de UX/UI Design, para a definição do *layout* da interface.

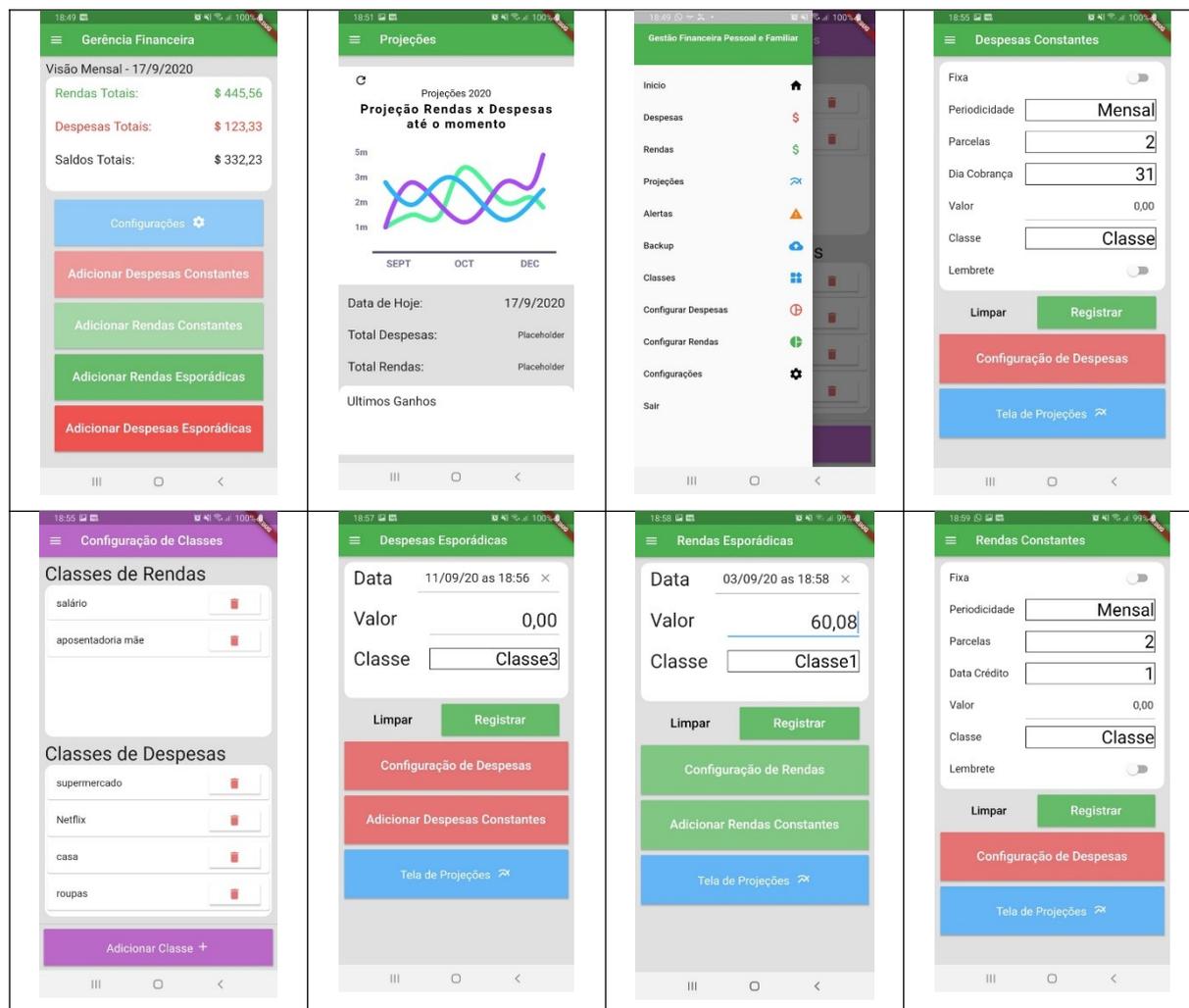
O objetivo inicial de exercitar os processos de projeto de interfaces foi atingido tendo sido aplicados os métodos e utilizadas as ferramentas de UX/UI Design para a definição da interface de um aplicativo móvel para gestão de finanças pessoais e familiares.

Avaliando as possibilidades de escalabilidade e expansão do aplicativo as prioridades a serem desenvolvidas em versões futuras poderiam ser:

- Expansão e integração Web;
- Integração segura com Apps de terceiros (Bancos e Apps de pagamento como PicPay);
- Implementação do modo avançado, que incluiria integrações de crédito, seguros, inflação e juros;
- Implementação de recursos para deficientes visuais e pessoas com pouca educação formal.

Espera-se que este trabalho seja referência para uma melhor forma de se projetar aplicações *mobile* através das análises feitas e resultados apresentados e que o aplicativo, que se pretende disponibilizar de forma gratuita em versão melhorada, possa contribuir para a solução da gestão financeira pessoal e familiar que impacta parte significativa da população brasileira.

Quadro 2 – Interface implementada no aplicativo



Fonte: os autores

Referências

ABNT. NBR ISO/IEC 9126-1 Engenharia de Software – Qualidade de produto. 2003. Disponível em: <<https://www.jkolb.com.br/wp-content/uploads/2014/02/NBR-ISO-IEC-9126-1.pdf>>. Acesso em: 16 set. 2020.

ACORDI, F. P. C. **Finanças Pessoais, Endividamento Familiar e Qualidade de vida do Servidor.** 2019. Disponível em: <http://www.profiap.org.br/profiap/tcfs-dissertacoes-1/utfpr/2019/dissertacao_francine-patricia-costa_2019.pdf>. Acesso em 25 mar.2020

ANTUNES, A. **Mobile First: conceito e razões para pensar por essa perspectiva.** 2019. Disponível em: <<https://gobacklog.com/blog/mobile-first/>>. Acesso em: 25 mar.2020.

BABICH, N. **Sketch, Wireframe, Mockup, and Prototype: Why, When and How.** 2020. Disponível em: <<https://uxplanet.org/sketch-wireframe-mockup-and-prototype-why-when-and-how-29a25b3157c4>>. Acesso em: 12 set.2020.

FERNANDES, C. V. **Modelagem de Banco de Dados Não Relacional em Plataforma Big Data Visando Dados de Internet Das Coisas.** 2017. Disponível em: <https://bdm.ufmt.br/bitstream/1/267/1/TCCP_2017_Cleiton%20Vitor%20Fernandes.pdf>. Acesso em: 12 set.2020.

FLUTTERDEV. **FAQ.** *sd online.* Disponível em: <<https://flutter.dev/docs/resources/faq#what-is-flutter>>. Acesso em: 12 set. 2020.

GOBACKLOG. **MOBILE FIRST: Conceito e Razões para Pensar por esta Perspectiva.** 2019. Disponível em: <<https://gobacklog.com/blog/mobile-first/>>. Acesso em: 10 set. 2020.

LIMA, Gustavo. **O que é Python e pra que serve?.** 2019. Disponível em: <<https://www.voitto.com.br/blog/artigo/python>>. Acesso em: 15 set. 2020.

MAR, Joseana. **Acessibilidade Visual: Interfaces amigáveis para daltônicos e disléxicos.** 2020. Disponível em: <<https://medium.com/@joseanasmar/acessibilidade-visual-e2afe67888a4>>. Acesso em: 10 set. 2020.

OLIVEIRA, Samuel Silva. **Bancos de Dados Não-Relacionais: um novo paradigma para armazenamento de dados em sistemas de ensino colaborativo.** Revista da Escola de Administração Pública do Amapá, v. 2, n. 1, p. 184–194, 2014. Disponível em:<<https://www2.unifap.br/oliveira/files/2016/02/35-124-1-PB.pdf>>. Acesso em: 16 set. 2020.

PALMER, Alex. **UX Design Fundamentals: Learn the Basics of IA, UX, UI.** 2020. Springboard. Disponível em: <https://ddf46429.springboard.com/uploads/resources/1548810145_Ebook_UX_Fundamentals_Learn_the_Basics_of_IA_UX_and_UI_Design.pdf>. Acesso em 16.ago.2020.

PASSOS, Ravi Figueiredo. **O Design da Informação em Interfaces de Hipermídias**. 2008. São Paulo: Universidade Anhembi Morumbi. Disponível em: <<https://ppgdesign.anhembi.br/wp-content/uploads/dissertacoes/14.pdf>>. Acesso em: 13.set.2020.

SOUZA, Ludimilla. **Pesquisa revela que 58% dos brasileiros não se dedicam às próprias finanças**. Agência Brasil, 2018. Disponível em: <<https://agenciabrasil.ebc.com.br/economia/noticia/2018-03/pesquisa-revela-que-58-dos-brasileiros-nao-se-dedicam-proprias-financas>>. Acesso em: 31 ago. 2020.

SPC. **O Conceito do Endividamento e as Consequências da Inadimplência**. 2016. Disponível em: <https://www.spcbrasil.org.br/uploads/st_imprensa/analise_educacao_financeira_dividas.pdf>. Acesso em 12.abr. 2020.