**APLICAÇÃO DE OBSERVAÇÕES SOBRE CONSTRUÇÃO DE TELAS**

**EM UMA APLICAÇÃO DE GERÊNCIA 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 tela antes de determinar qualquer identidade visual. Este artigo explora as estruturas de tela aplicadas a um aplicativo de gerência financeira, aplicativo este que possui objetivo social e procura causar uma forte identificação visual com um design simples e claro para que os usuários possam interagir e aprender facilmente, também levando em conta as aplicações de qualidade de software em vigor pela ISO 9126.

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

**Abstract**

*The visual identity of an app determines its mental conception by its users. A strong visual identity can be established by several means, be simplistic, minimalistic or complex and broad reaching. However, understanding the layout concepts of a screen before establishing any visual identity can be essential. This article explores the scree structures and applies them to an app for financial management with a social goal and it seeks to have a strong visual identification with a simple and clear design so users can interact and learn easily its functions. Also taking in consideration the quality assurance from ISO 9126.*

**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 reportagem de (SOUZA, 2018), o aplicativo é uma maneira proposta de lidar com esta situação.

Pela pesquisa efetuada propriamente foi descoberto que o tipo mais requisitado de aplicação são aplicações mobile e com integrações para desktop, durante o desenvolvimento do mesmo se focou no paradigma de *‘mobile first’* que visa estabelecer todas as estruturações no mobile e apenas depois traduzi-las para outras fontes, como web e desktop.

Para o desenvolvimento foram utilizadas as linguagens Dart, através do framework Flutter, e python para a relação das APIs, o back-end foi feito utilizando um banco de dados não relacional para que facilitasse a utilização do aplicativo a longo prazo.

Para o melhor desenvolvimento orgânico do aplicativo também será utilizada a teoria própria para o desenvolvimento de telas, baseada e estruturada para uma melhor ergonomia de aplicação, desta forma reduzindo quaisquer resistências a utilização do mesmo.

Através disso será feita uma estrutura cruzada entre o desenvolvimento do aplicativo, sua documentação e os paradigmas que estão sendo seguidos com o objetivo principal de testar as análises próprias feitas para o desenvolvimento de telas.

1. **Referencial Teórico de Engenharia de Software**

Através da análise de telas procura-se proporcionar uma melhoria da experiência do usuário, desta forma reduzindo a resistência dos usuários a utilização do software.

Durante a pesquisa efetuada propriamente utilizando a plataforma google forms a requisição de mobile foi a maior requisição entre os votos, então, a proposta de desenvolvimento, focada no paradigma de *mobile first* procura desenvolver o melhor possível a interface visual de interação.

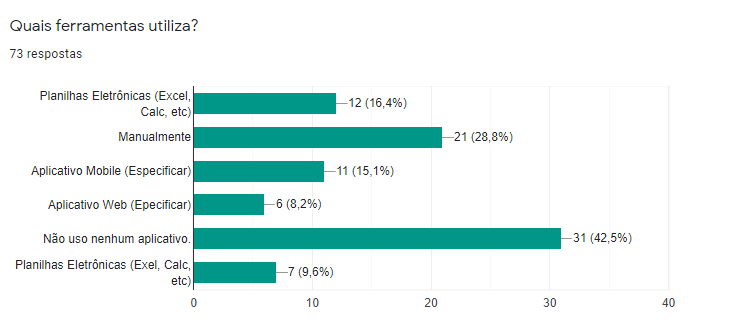
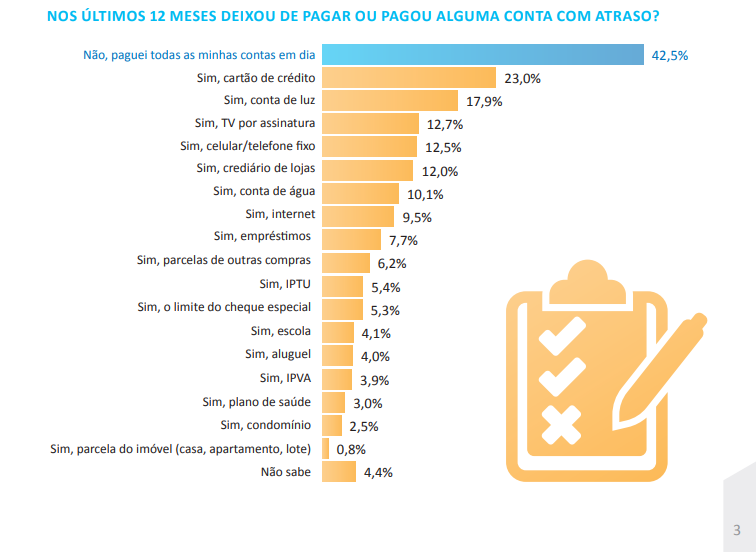


Imagem 1: Formulário próprio realizado na ferramenta google forms

A Imagem 1 mostra que a maior parte (42.5%) dos perguntados não utilizam nenhuma forma de aplicativo para a gerência financeira pessoal e familiar. Desta forma a criação de uma aplicação que ajude em tal tarefa poderá ter impacto pessoal e social.

Existe evidência que indica 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 até Fevereiro de 2016 (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 de simples uso?

* 1. **Estruturando a formula ‘Mobile First’**

Mobile first é uma metodologia de aplicação no desenvolvimento de softwares que foca no desenvolvimento primariamente para o mobile e, apenas depois, a adaptação para as outras fontes (web e desktop).

Segundo (ANTUNES, 2019) o paradigma de 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.

Pela pesquisa própria realizada também foi encontrada uma preferência clara pela utilização de mobile. Como demonstrado na Imagem 2.

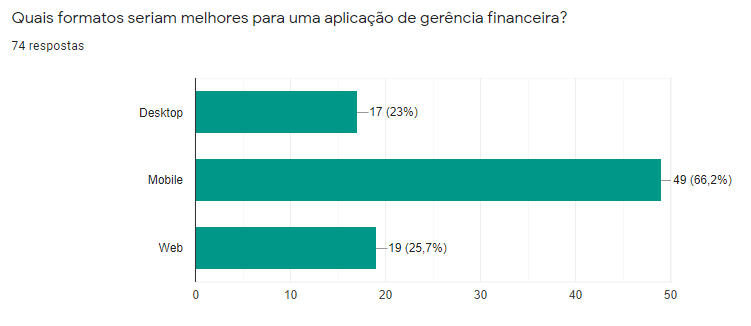


Imagem 2: Preferência por mobile na pesquisa realizada.

A tendência pela preferência mobile foi a maior requisição, em 66,2% a aplicação do paradigma de ‘*Mobile First’* demonstra ser o mais efetivo para atingir os objetivos.

* 1. **Objetivo Social**

O app possui um objetivo social, sua estruturação é feita para atingir o maior numero de pessoas possível gratuitamente. Em uma estrutura simples este APP seria uma forma de se permitir gerência gratuita a todos os usuários. Sem anúncios ou assédio de mensagens para que se compre um pacote prêmium.

Todas as funções do APP devem ser disponibilizadas para todos os usuários 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 de proteção de dados.

* 1. **Análise de conceitos base de telas**

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 estrutural de formas diversas a partir de uma estrutura de análise de telas.

**• Conceito número 1 – Tipos de informações (INFO)**

Existem inúmeras formas nas quais a informação pode ser passada, seja ela imagem, vídeo, gifs, sons, musicas ou diversas outras as pessoas reagem de formas diferentes para cada uma destas informações.

Para efeitos do artigo dividiremos em três tipos de informação. Estática, Dinâmica e Sonora.

o Informações Estáticas (ES): Informações visuais que não tem nenhum tipo de movimento (Imagens, logos, emojis, etc.) E quaisquer tipo de informações textuais.

o Informações Dinâmicas (DI): São informações visuais que compõe as informações de acompanhamento do usuário, possuem movimentação dentro de si, podendo ou não serem atreladas a informações sonoras (Videos, Gifs, Seriados, streamings, etc.)

o Informações Sonoras (SO): São informações que compõe a percepção sonora do usuário. Podem ou não serem atreladas a INFO DI. (Musicas, Vídeos, Podcasts, Reuniões, Etc.)

o Informações Mutáveis (IM): São informações que compõe o feed do usuário, aplicáveis aos chamados feeds do mesmo. Utilizados principalmente em redes sociais como ‘twitter’ e ‘facebook’.

**• Conceito número 2 – Sentido de Leitura (SL)**

O sentido de leitura é ativamente muito importante para entender como as telas são divididas de forma consciente ou inconscientemente. A forma pela qual nós lemos definem os conceitos de importância da tela assim como os conceitos de parâmetros de como as telas são divididas.

Como podemos ver nas diversas telas a seguir.

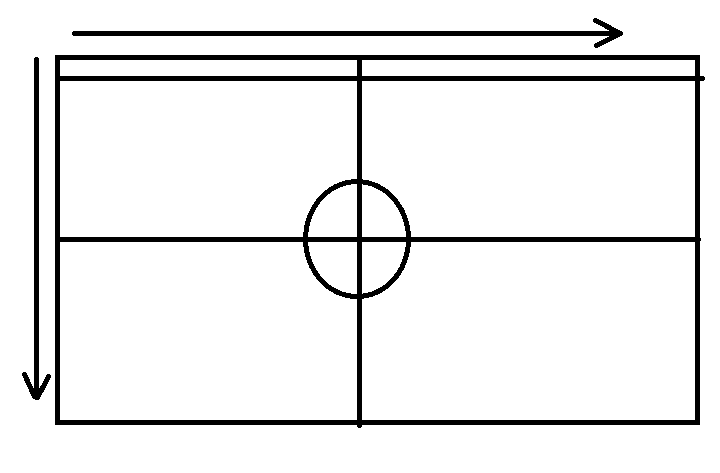


Imagem 1 – Sentido de Leitura Ocidental conceitual a divisão Desktop. (Próprio)

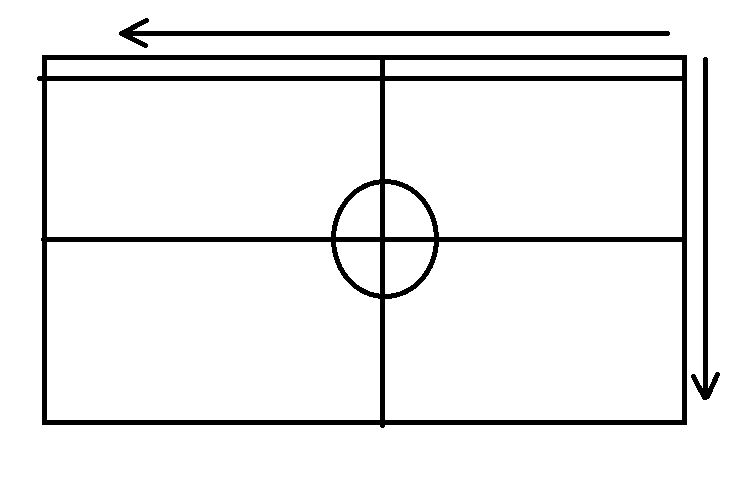


Imagem 2 - Sentido de Leitura Oriental conceitual a divisão desktop (Próprio)

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 de onde a informação dinâmica é mostrada).

Como mostrado nas imagens a seguir sobre análise dos fluxos de leitura e visão:

O sentido de leitura é importante para definir também os conceitos de costume e importância dos aplicativos.

• **Conceito Número 3 – Padrões de importância (IMP)**

Durante os 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 que nos foi ensinado durante os anos de utilização de aparelhos digitais e navegação na internet. Desta forma seguindo o fluxo demonstrado abaixo.

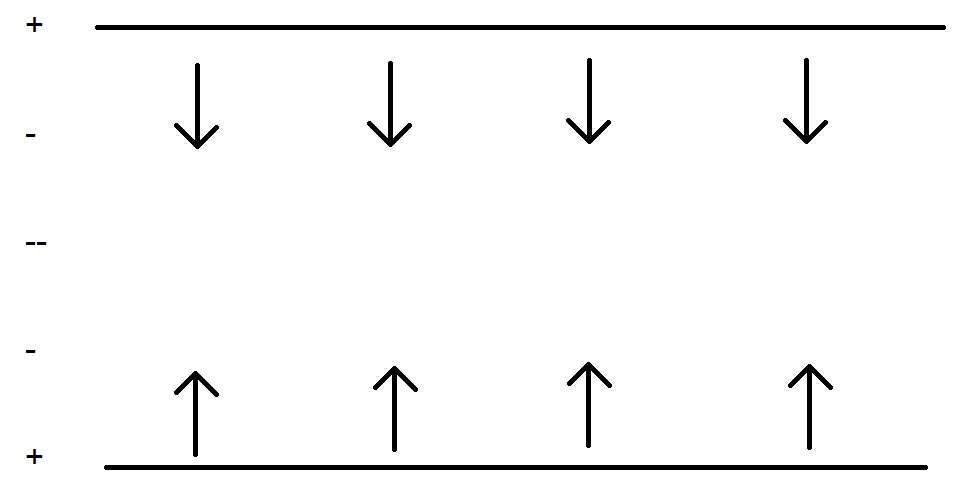


Imagem 3 – Orientações de importância do software (Próprio)

Os padrões de importância devem ditar a orientação do seu design de forma a analisar 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 deve-se colocar as informações mais importantes e/ou mais utilizadas devem vir primeiro.

Para informações mutáveis o sentido de importância indica o fluxo de notificações e novas interações de seu feed 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’ poderá ser uma boa prática.

Este é utilizado também para definir a estrutura geral de cada página e tela. Como observado na imagem 4, abaixo.

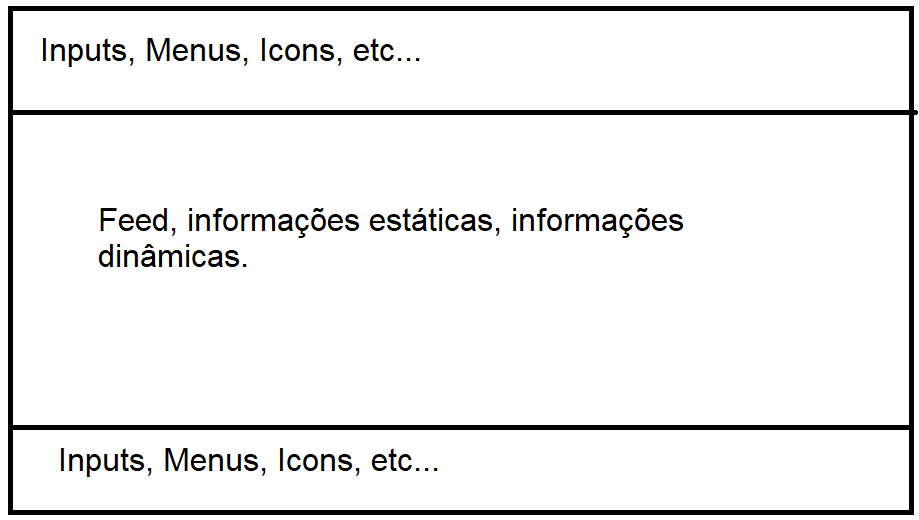


Imagem 4 –Divisão de estrutura básica (Próprio)

Estruturalmente, a análise leva a crer que as aplicações atuais seguem a estruturação da Imagem 4. Seguindo assim os índices de importância do aplicação. Os itens mais utilizados se localizam usualmente nas regiões demarcadas como Menus e Inputs enquanto as informações de leitura e visualização se localizam ao centro da tela.

Esta estrutura foi lentamente definida com a evolução dos websites e aplicações digitais e, atualmente, estamos acostumados a estas.

Também podemos ampliar o conceito, aplicando o conceito de leitura a este logo nos encontramos com a seguinte análise estrutural em regiões. Como demonstrado na Imagem 5.

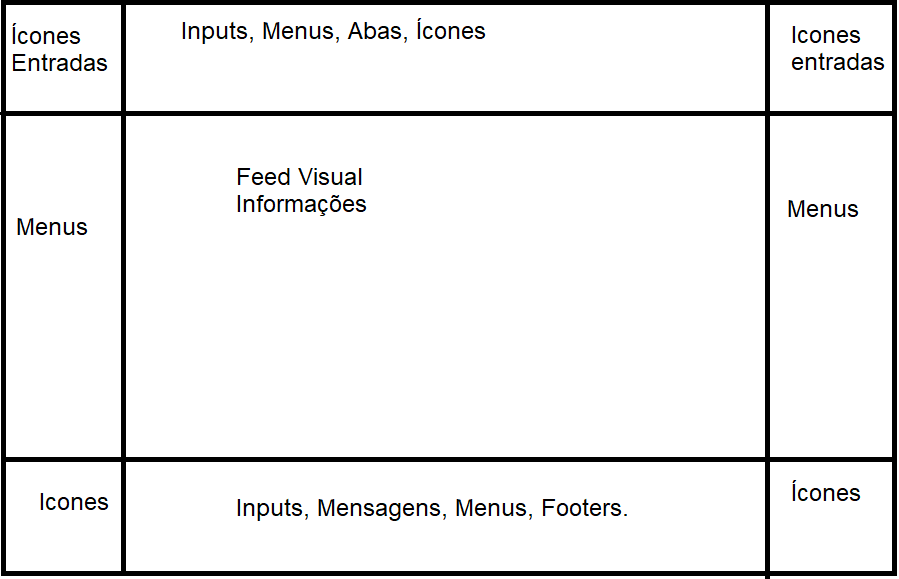


Imagem 5 –Divisão de estrutura por leitura e Importância (Própria)

Analisando a Imagem 5 a estrutura se monta de forma melhor para visualização completa. Lembrando que as divisões presentes podem mudar com formas dinâmicas (Como menus ‘scaffold’ que se minimizam após o uso) ou com estruturas diferentes, mas, a estruturação geral ainda se mantém, mesmo que as proporções mudem.

**• Conceito número 4 – Facilidade 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 ao qual o desenvolvedor deve ficar atento principalmente. Existem, de fato, inúmeras condições que podem causar problemas na utilização de um aplicativo. Porém estas são as mais comuns a serem observadas.

* Dislexia
* Daltonismo
* Deficiência visual
* Deficiência auditiva
* Pessoas com baixa ou sem educação formal.
* Pessoas acima de 50 anos.

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 aproximar os assuntos. É recomendado que, nas configurações de seu produto se tenham pelo menos uma configuração de dislexia e de daltonismo. Pois estas são as menos intrusivas no código fonte e que o sistema seja desenhado da forma mais orgânica possível para facilitar quaisquer usuários não acostumados com tecnologia de utilizarem.

Daltonismo e dislexia são os menos intrusivos pois, como mostrado em Mar (2020) já existem métodos não intrusivos de aplicação para facilitar o acesso.

Quanto a deficiências visuais e baixa educação formal são soluções semelhantes. Existem vários níveis de deficiência visual que podem atrapalhar ou completamente negar a utilização de APPs. Cegueira completa é um exemplo em que uma aproximação distinta é necessária, como, por exemplo, interfaces sonoras.

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

* 1. **Análise de telas específicas para Mobile**

Aplicando os conceitos construídos na parte anterior para especificamente a plataforma mobile temos de fazer algumas observações específicas.

Mobile, geralmente, se aplica em duas formas diferentes de adaptação. Celulares e Tablets. Isso é importante não pela ideia de código, mas pela ideia de proporções e espaço disponível.

Mostrando na Imagem 6 é possível melhor visualizar as proporções aproximadas de uma tela mobile celular e uma tela mobile tablet.

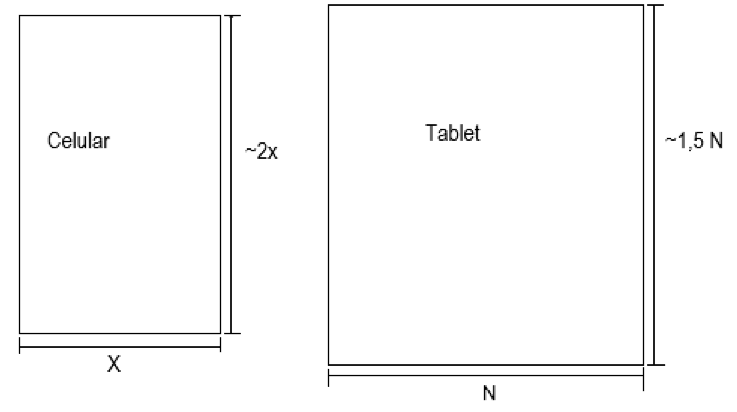


Imagem 6: Proporções Celular e Tablet. (X e N podem mudar de acordo com o aparelho, mas as proporções tendem a permanecer semelhantes e N > X)

Conhecendo as proporções que se tem trabalho e aplicando os conceitos anteriores chegamos as seguintes observações sobre a estrutura mobile.

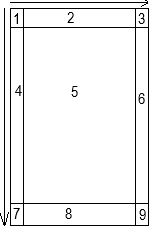


Imagem 7: Estruturas Prévias no mobile

Aplicando os sentidos de leitura e organização da tela ao mobile temos um total de 9 divisões que seguem o sentido de leitura ao qual se está aplicando. Para estas divisões aplicamos suas maiores funções. Para tal, de acordo com a maioria dos APPs as funções das divisões são.

1 – Logos e Acesso ao menu

2 – Nome da tela, menus e nome do app..

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

Tudo isso se aplica também ao conceito do tamanho dos itens e ícones clicáveis. 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 o APP a ser desenvolvido. Considerando o tipo de informações que serão aplicadas.

Com tudo isso em mente seguimos para a estruturação geral do APP.

* 1. **Estruturação geral do APP**

Imagens 8 e 9 demonstram a estruturação do aplicativo e de seu menu principal respectivamente.

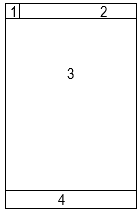


Imagem 8 – Estruturação geral (Própria)

Vemos aqui a estruturação geral da aplicação, seguindo os modelos anteriores onde 1 é o acesso ao menu, 2 é a informação de pagina, 3 é o conteúdo de página e 4 são menus e botões de *footer*.

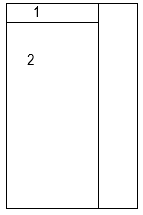


Imagem 9 – Estruturação Menu.

A estruturação do menu é a mais simples possível 1 é o header e 2 são os itens de menu para navegação.

* 1. **Sobre Flutter**

Flutter 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 Google (2018), 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. Também para designers que desejam mostrar suas formas em alta fidelidade.

Devido a fácil modularização de Flutter e sua estruturação direta o framework foi escolhido para ser aplicado ao aplicativo mobile.

* 1. **Sobre Python**

Python 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 pode ser utilizado como linguagem simples, de orientação a objeto ou de design gráfico, embora as opções de design da linguagem sejam extremamente limitadas.

Devido a simplicidade da linguagem e a abundância de bibliotecas Python foi escolhido para realizar a comunicação entre a aplicação e o servidor de backups.

* 1. **Sobre Banco de dados não relacional**

Bancos de dado não relacionais ou NoSQL são bancos de dados baseados em uma estrutura Json e, segundo Ianni (2012) Bancos não relacionais são bem variados e utilizados por diversas empresas para Big Data devido a sua versatilidade.

Estruturar uma aplicação ao redor de um banco não relacional tem suas diferenças, vantagens e desvantagens e devem ser obtidos de acordo com a necessidade da aplicação, como mencionado em De Oliveira (2014), as aplicações rodando em bancos de dado não relacionais tem uma grande melhora em performance, porém não foram criados para substituir bancos relacionais.

Para a aplicação foi escolhido o Sembast para a persistência de dados locais e o MongoDB para guardar os dados no backend, Sembast é um banco de dados não relacional para flutter, o qual pode ser facilmente adicionado a qualquer projeto através do link https://pub.dev/packages/sembast.

MongoDB é uma plataforma gratuita de armazenamento de dados NoSQL. Este possui recursos pagos também como processamento paralelo e estruturação de dados.

A utilização de bancos não relacionais foi decidida pelo fato de que, idealmente, usuários vão utilizar a aplicação durante muito tempo e o volume de dados irá crescer com cada novo usuário que decidir integrar aplicações ou fazer backups. Desta forma a utilização do banco não relacional se deve ao fato de que manter o desempenho da aplicação é essencial, assim como reduzir erros que podem vir do tipo de dados que estará sendo alimentado.

1. **Referencial Teórico de um tema específico da Engenharia de Software**

Pelo que foi estabelecido previamente é possível observar o grande foco na experiência do usuário, a experiência do usuário é um fator determinante na qualidade do software.

A qualidade de software nesta aplicação está sendo utilizada como uma maneira de garantir a melhor experiência possível para o usuário no requerimento. Tal experiência é essencial para manter o usuário utilizando o APP.

A utilização constante do APP será necessária para uma melhor estruturação financeira. Desta forma, um software de alta qualidade causara uma resistência menor ao usuário utilizador.

* 1. **Sobre Qualidade de Software**

Segundo a ISSO 9126/NBR 13596 a qualidade de software possui algumas características, sendo estas Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade, Portabilidade. Suas divisões internas são mostradas na Imagem 10.

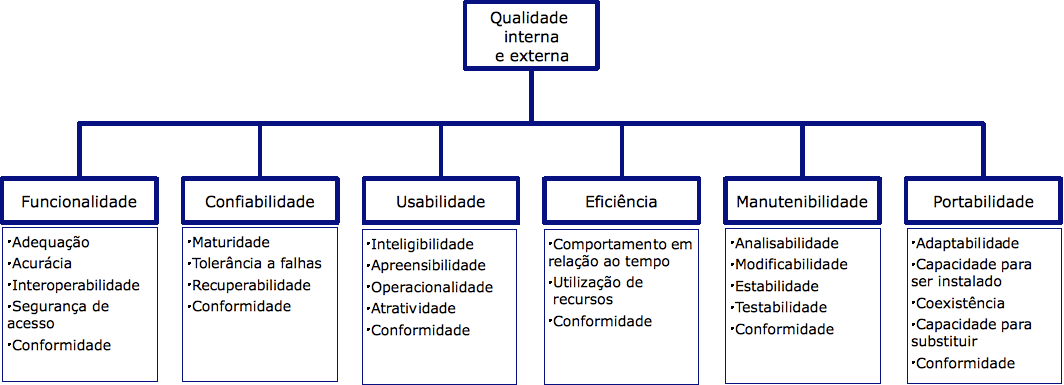


Imagem 10 – Definição de qualidade externa e interna (WIKIPEDIA, 2020)

Durante o desenvolvimento da aplicação o foco principal se foi nas características de Usabilidade e Eficiência da aplicação. Pois, em sua concepção o APP é uma demonstração de como melhor aplicar conceitos de tela ao desenvolvimento.

De acordo com a ISO 9126, nas características de Usabilidade:

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

- Apreensibilidde: A capacidade do software de ser aprendida pelo usuário.

- Operacionalidade: A capacidade do usuário de operar e controlar as funcionalidades do software.

- Estética/Atratividade: O quão atrativo são os visuais de seu software, o quão forte é sua identidade visual.

- Acessibilidade: A prática de tornar o software acessível ao maior número de pessoas possíveis, incluindo pessoas portadoras de deficiência.

- Conformidade: A capacidade do software de se apresentar em padrões estabelecidos.

De acordo com a ISO 9126, nas características de Eficiência

- Comportamento em Relação ao tempo: Eficiência de resposta do aplicativo em relação ao tempo.

- Utilização de Recursos: As medidas de utilização de hardware que o software requer.

- Conformidade: A capacidade do software de se apresentar em padrões estabelecidos.

* 1. **Qualidade de Software aplicada a construção de telas e ao APP**

Uma vez com o contexto fornecido na parte 3.1, é possível iniciar sua contextualização em relação ao APP. Como declarado nas sessões anteriores o APP tenta estabelecer uma boa interação visual com o usuário. E, para isso, todas as características de usabilidade devem ser aplicadas em maior ou menor nível.

* 1. **Integração de aplicações**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

1. **Resultados da Análise e Projeto**
   * 1. **Rabiscoframes**

| **Figuras X –** Estudos de interface realizados por Rabiscoframes | | |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Fonte:** os autores

.

* 1. **Telas da Aplicação**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

* 1. **Possibilidades futuras**

A aplicação produzida teve uma estrutura visual satisfatória porém ainda possui muitas possibilidades de expansão e melhorias ao todo. Melhor integração entre aplicações, adicionar uma estrutura web junto a mobile até mesmo integração com APIs de outros lugares como Bancos e APPs de pagamento pode ser possível.

Olhando para as possibilidades de escalonamento e expansão do aplicativo as prioridades a serem desenvolvidas seriam.

1 – Expansão e integração Web.

2 – Integração segura com Apps de terceiros (Bancos e Apps de pagamento como PicPay)

3 – Implementação do modo avançado, este que iria incluir integração de crédito, seguros, inflação e juros.

4 – Implementação de recursos para deficientes visuais e pessoas com pouca educação formal.

Uma vez que estas três expansões primárias estejam implementadas a manutenção geral da aplicação será necessária e, para reduzir manutenção ao máximo é recomendado que a mesma seja o mais minimalista o possível, requerendo a menor quantidade de servidor o possível.

1. **Resultados da Implementação**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

* 1. **Assunto 1**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

* 1. **Assunto 2**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

1. **Conclusão**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

**Referências**

MOBILE FIRST: CONCEITO E RAZÕES PARA PENSAR POR ESTA PERSPECTIVA. Gobacklog, 16 jul. 2019. Disponível em: https://gobacklog.com/blog/mobile-first/. Acesso em: 10 set. 2020.

SOUZA, Ludimilla. Pesquisa revela que 58% dos brasileiros não se dedicam às próprias finanças. Agência Brasil, 28 mar. 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.

MAR, Joseana. Acessibilidade Visual: Interfaces amigáveis para daltônicos e disléxicos. Medium, 19 fev. 2020. Disponível em: https://medium.com/@joseanasmar/acessibilidade-visual-e2afe67888a4. Acesso em: 10 set. 2020.

FLUTTER FAQ. Flutter dev: Google, 2018. Disponível em: https://flutter.dev/docs/resources/faq#what-is-flutter. Acesso em: 12 set. 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.

LIMA, Gustavo. O que é Python e pra que serve?. Voitto, 7 out. 2019. Disponível em: https://www.voitto.com.br/blog/artigo/python. Acesso em: 15 set. 2020.

IANNI, Vinicius'. Entendendo o conceito do NoSQL. Dev Media, 2012. Disponível em: https://www.devmedia.com.br/introducao-aos-bancos-de-dados-nosql/26044. Acesso em: 15 set. 2020.

DE 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 publica do Amapá, Revista da escola de administração publica do Amapá, v. 2, n. 1, p. 184–194, 1 dez. 2014. Disponível em: https://www2.unifap.br/oliveira/files/2016/02/35-124-1-PB.pdf. Acesso em: 16 set. 2020.

WIKIPEDIA. ISO/IEC 9126. Wikipedia, 23 abr. 2020. Disponível em: https://pt.wikipedia.org/wiki/ISO/IEC\_9126. Acesso em: 16 set. 2020.