



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

GIOVANI ALBUQUERQUE VILLARIM DE SIQUEIRA

**Desenvolvimento de um Decision Board com base nas avaliações do Reclame
Aqui para aprimorar a Tomada de Decisão Organizacional**

RECIFE

2023

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

SISTEMAS DE INFORMAÇÃO

GIOVANI ALBUQUERQUE VILLARIM DE SIQUEIRA

**Desenvolvimento de um Decision Board com base nas avaliações do Reclame
Aqui para aprimorar a Tomada de Decisão Organizacional**

TCC apresentado ao Curso de Sistemas de Informação da Universidade Federal de Pernambuco, Centro de Informática, como requisito para a obtenção do título de bacharel em Sistemas de Informação.

Orientador(a): Vinicius Cardoso Garcia.

Coorientador(a): Maria da Conceição Moraes Batista.

**Recife
2023**

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Siqueira, Giovani Albuquerque Villarim de.

Desenvolvimento de um Decision Board com base nas avaliações do Reclame
Aqui para aprimorar a Tomada de Decisão Organizacional / Giovani
Albuquerque Villarim de Siqueira. - Recife, 2023.

80 p. : il., tab.

Orientador(a): Vinicius Cardoso Garcia

Coorientador(a): Maria da Conceição Moraes Batista

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de
Pernambuco, Centro de Informática, Sistemas de Informação - Bacharelado,
2023.

1. Engenharia de Software. 2. Decision Board. 3. Sistema de Apoio à
Decisão. I. Garcia, Vinicius Cardoso. (Orientação). II. Batista, Maria da Conceição
Moraes. (Coorientação). IV. Título.

000 CDD (22.ed.)

GIOVANI ALBUQUERQUE VILLARIM DE SIQUEIRA

Desenvolvimento de um sistema Decision Board com base nas avaliações do Reclame Aqui para aprimorar a Tomada de Decisão Organizacional

TCC apresentado ao Curso de Sistemas de Informação da Universidade Federal de Pernambuco, Centro de Informática, como requisito para a obtenção do título de bacharel em Sistemas de Informação.

Aprovado em: 27/09/2023.

BANCA EXAMINADORA

Profº. Dr. Vinicius Cardoso Garcia (Orientador)
Universidade Federal de Pernambuco

Profº. Dr. Robson do Nascimento Fidalgo (Examinador Interno)
Universidade Federal de Pernambuco

Dedico esse trabalho à minha família, que sempre me apoiou e deu suporte nos momentos mais importantes da minha vida, aos amigos que conheci durante essa jornada, e a mim mesmo, por me manter sempre dedicado, disciplinado e resiliente na busca dos meus objetivos.

AGRADECIMENTOS

Primeiramente gostaria de agradecer profundamente à minha família, em especial ao meu querido Pai Gandhi Villarim de Siqueira, à minha querida mãe Roberta Albuquerque Mello de Castro Silva e ao meu querido irmão e mentor Renan Albuquerque Villarim de Siqueira, sem a base de exemplo, estudo e perseverança de vocês não seria possível concluir uma graduação de ponta como a de Sistemas de Informação na Universidade Federal de Pernambuco.

Em segundo lugar, eu gostaria de agradecer à própria Universidade Federal de Pernambuco, por oferecer toda a estrutura básica e necessária para o desenvolvimento dos discentes. Em conjunto com a universidade, eu gostaria de agradecer também ao Centro de Informática, pois sem o espírito empreendedor e inovador que esse centro trás não seria possível o desenvolvimento de tantos projetos influentes na nossa sociedade nos dias de hoje.

Também gostaria de deixar claro o meu agradecimento a todo o corpo docente que me acompanhou durante todo esse percurso desde 2019, em especial ao meu professor orientador Vinicius Cardoso Garcia e a minha professora coorientadora Maria da Conceição Moraes Batista, que me deram todo o suporte necessário no desenvolvimento desse projeto importante que é o Trabalho de Conclusão de Curso, sem vocês não seria possível o processo criativo e arquitetural de todo esse grande projeto.

Por fim e não menos importante, gostaria de agradecer aos meus amigos que estiveram comigo durante todo esse processo de transformação pessoal e profissional, em especial gostaria de mencionar Bruno Martins Santos, que sempre foi o meu braço direito nessa longa jornada, e a todos que conheci na graduação e no CITi, vocês fizeram e fazem parte da minha história, obrigado!

"Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que há muito a fazer." - Alan Turing.

RESUMO

Um problema das organizações atualmente é conseguir ter um panorama detalhado das reclamações que elas possuem no Reclame Aqui, que é o maior site brasileiro de queixas contra empresas sobre atendimento, compra e venda de produtos e serviços. Além disso, caso uma empresa queira fazer uma leitura dos seus dados que estão armazenados nos servidores do Reclame Aqui com o objetivo de fazer uma análise interna, o preço é extremamente elevado, tornando esse processo ainda mais difícil de ser realizado. Baseando-se nesse contexto, foi observada a oportunidade de explorar os dados do Reclame Aqui de forma gratuita e legítima através de uma técnica chamada Web Scraping, e criar, com isso, um Sistema de Apoio à Decisão Estratégica que irá disponibilizar diversos Decision Boards apoiadores na tomada de decisão das organizações.

Palavras-chave: Engenharia de Software; Decision Board; Sistema de Apoio à Decisão.

ABSTRACT

A well-known problem that the organizations have been going through recently is having detailed data-oriented complaints on Reclame Aqui, the biggest Brazilian website of Customer's Complaint Hub, where they can review others' experiences and see if the desired site is trustworthy. Besides this problem, if some company aims to query its data from the servers of Reclame Aqui, the price is costly, making this entire process even harder. Regarding this scenario, was noticed the opportunity to explore Reclame Aqui's data in a free and legal way using the technique called Web Scraping, creating with that a Decision Support System that will be responsible for making available several Decision Boards, which will be helpful in the process of decision making.

Keywords: Software Engineering; Decision Board; Decision Support System.

LISTA DE FIGURAS

Figura 1 - Fluxo cronológico da revisão literária.

Figura 2 - Exemplo de decision board simples.

Figura 3 - Fluxo do Web Scraping.

Figura 4 - Exemplo visual dos ambientes mobile e web.

Figura 5 - Fluxo cronológico da aplicação da metodologia.

Figura 6 - Mockup da tela Inicial do aplicativo versão mobile.

Figura 7 - Mockup da tela de reclamações do aplicativo versão mobile.

Figura 8 - Mockup da tela de reclamações com a aba lateral aberta do aplicativo versão mobile.

Figura 9 - Mockup da tela de listagem dos decision boards do aplicativo versão mobile.

Figura 10 - Mockup da tela de exibição dos decision boards do aplicativo versão mobile.

Figura 11 - Mockup da tela Inicial do aplicativo versão web.

Figura 12 - Mockup da tela de reclamações do aplicativo versão web.

Figura 13 - Mockup da tela de reclamações com a aba lateral aberta do aplicativo versão web.

Figura 14 - Mockup da tela de listagem dos decision boards do aplicativo versão web.

Figura 15 - Mockup da tela de exibição dos decision boards do aplicativo versão web.

Figura 16 - Ciclo de vida do Scrum.

Figura 17 - Tendência de crescimento do Flutter em comparação ao seu concorrente.

Figura 18 - Demonstração do fluxo de trabalho com a tecnologia git.

Figura 19 - Características de uma boa arquitetura de software.

Figura 20 - Representação visual da arquitetura de software utilizada.

Figura 21 - Representação visual do fluxo de navegação no sistema.

Figura 22 - Visualização prática da tela Inicial do aplicativo versão web.

Figura 23 - Visualização prática da tela de reclamações do aplicativo versão web.

Figura 24 - Visualização prática da tela de reclamações com a aba lateral aberta do aplicativo versão web.

Figura 25 - Visualização prática da tela de listagem dos decision boards do aplicativo versão web.

Figura 26 - Visualização prática da tela de exibição dos decision boards contendo os filtros especializados do aplicativo versão web.

Figura 27 - Visualização prática do filtro especializado dos decision boards na versão web.

Figura 28 - Visualização prática da tela Inicial do aplicativo versão android.

Figura 29 - Visualização prática da tela de reclamações do aplicativo versão android.

Figura 30 - Visualização prática da tela de reclamações com a aba lateral aberta do aplicativo versão android.

Figura 31 - Visualização prática da tela de listagem dos decision boards do aplicativo versão android.

Figura 32 - Visualização prática da tela de exibição dos decision boards do aplicativo versão android.

Figura 33 - Visualização prática do filtro especializado dos decision boards na versão android.

Figura 34 - Visualização prática da tela Inicial do aplicativo versão ios.

Figura 35 - Visualização prática da tela de reclamações do aplicativo versão ios.

Figura 36 - Visualização prática da tela de reclamações com a aba lateral aberta do aplicativo versão ios.

Figura 37 - Visualização prática da tela de listagem dos decision boards do aplicativo versão ios.

Figura 38 - Visualização prática da tela de exibição dos decision boards do aplicativo versão ios.

Figura 39 - Visualização prática do filtro especializado dos decision boards na versão ios.

Figura 40 - Representação do decision board Reclamações por trimestre na versão Web.

Figura 41 - Representação do decision board Reclamações por Estados na versão Web.

Figura 42 - Representação do decision board Status por Reclamação na versão Web.

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais.

Quadro 2 - Requisitos não funcionais.

LISTA DE ABREVIACOES

SAD	Sistema de Apoio à Deciso
TCC	Trabalho de Concluso de Curso
DB	Data Base

SUMÁRIO

1 INTRODUÇÃO	13
1.1 MOTIVAÇÃO	15
1.2 OBJETIVOS	16
1.2.1 OBJETIVO GERAL	16
1.2.2 OBJETIVOS ESPECÍFICOS	16
1.3 ESTRUTURA	16
2. REVISÃO DA LITERATURA	17
2.1 PROCESSO DE TOMADA DE DECISÃO	18
2.2 DECISION BOARD	19
2.3 WEB SCRAPING	20
2.4 AMBIENTE MOBILE E WEB	22
2.5 TRABALHOS RELACIONADOS	24
2.5.1 MICROSOFT POWER BI	24
2.5.2 DATABOX	24
2.5.3 GOOGLE DATA STUDIO	24
2.5.4 PROPOSTA DE VALOR	25
2.6 SUMÁRIO DA SEÇÃO	25
3 METODOLOGIA	26
3.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	27
3.1.1 REQUISITOS FUNCIONAIS	28
3.1.2 REQUISITOS NÃO FUNCIONAIS	29
3.2 PROTOTIPAÇÃO EM ALTA FIDELIDADE	30
3.2.1 RESPONSABILIDADE PARA TELAS MENORES	31
3.2.2 RESPONSABILIDADE PARA TELAS MAIORES	34
3.3 METODOLOGIA ÁGIL	37
3.4 FLUTTER E DART	39
3.5 GIT E GITHUB	40
3.6 CÓDIGO E ARQUITETURA LIMPA	42
3.6.1 CÓDIGO LIMPO	42
3.6.2 ARQUITETURA LIMPA	44
3.7 SUMÁRIO DA SEÇÃO	48
4 RESULTADOS	49
4.1 FLUXO PRINCIPAL	49
4.1.1 WEB	51
4.1.2 ANDROID	55
4.1.3 IOS	58
4.2 DECISION BOARDS	64
4.2.1 CONSTRUÇÃO	65
4.2.1.1 RECLAMAÇÕES POR MESES	65
4.2.1.2 RECLAMAÇÕES POR ESTADOS	66
4.2.1.3 GRÁFICO POR STATUS DA RECLAMAÇÃO	68
4.3 DIFICULDADES ENCONTRADAS	69
4.4 LIÇÕES APRENDIDAS	70

4.5 SUMÁRIO DA SEÇÃO	71
5 CONCLUSÃO	71
6 TRABALHOS FUTUROS	72
REFERÊNCIAS	73

1 INTRODUÇÃO

O projeto desenvolvido neste trabalho de conclusão de curso tem como principal razão viabilizar o processo de tomada de decisão frente a um cenário que não oferece formas simples e baratas de se alcançar isso. Além disto, oferece uma maior possibilidade de utilização, tendo em vista que foi desenvolvido para três ambientes diferentes, sendo eles o Android, IOS e Web, tendo como principal objetivo a democratização do acesso ao suporte na tomada de decisão organizacional.

O processo de tomada de decisão é uma fase pela qual toda e qualquer organização irá passar durante o seu processo de desenvolvimento. Durante esses momentos, é fundamental ter clareza de todos os elementos que irão afetar a sua escolha, pois eles são capazes de impulsionar um negócio (CAPLER, 2020) ou até mesmo de arruinar uma ideia promissora. Seguindo nessa linha de pensamento é que surge a importância de se criar uma prática sistemática no processo de tomada de decisão, objetivando criar parâmetros a serem seguidos, trazendo consigo uma padronização que torne esse momento mais seguro, pois de acordo com Capler: "Isso pode diminuir conflitos, aumentar a confiança e a coesão, contribuir para a funcionalidade geral da equipe e aumentar o consenso" (CAPLER, 2020). É importante mencionar que essa prática é variável de acordo com os diferentes contextos, pois seria fundamentalmente impossível estabelecer padrões a serem seguidos ao mesmo tempo para uma Startup moderna e uma empresa tradicional (KERRIGAN, 2018).

Quando uma prática sistemática é implementada nos processos internos de tomada de decisão, as chances de sucesso aumentam e os riscos de elementos não observados são reduzidos drasticamente, pois o objetivo de todo framework é oferecer uma base confiável para que as aplicações, sejam elas digitais ou não, possam funcionar de forma minimamente estável.

Existem várias práticas que são capazes de dar suporte ao processo de tomada de decisão, dentre elas está a de construção de gráficos - Decision Boards¹ - responsáveis por sumarizar informações que antes seriam difíceis de serem

¹ Definição formal de Decision Boards:

<https://towardsdatascience.com/the-fascinating-science-and-art-of-decision-boards-6b1d5a461132>

cruzadas, pois muitas vezes os dados coletados ainda não foram suficientemente polidos ou filtrados a um nível de suporte à decisão organizacional.

Com isso, é importante analisar a importância que os dados possuem, pois nos dias atuais eles são base para diversos sistemas, sendo estes redes sociais, aplicativos de streaming ou até mesmo grandes negócios (MEIRA, 2021). Tendo em vista que a geração de dados hoje é muito maior do que nas décadas passadas e que a sociedade global hoje está muito mais conectada aos dispositivos com internet (DARÍO, 2015), é natural que o valor sobre esse bem - o dado - venha aumentando de acordo com a sua disponibilidade, pois eles sempre são capazes de trazer insights importantes em diversos cenários diferentes.

Existem os casos em que os dados foram capazes de afetar eleições presidenciais, como no episódio de 2018 em que dados coletados pela Cambridge Analytica foram capazes de influenciar os usuários do Facebook nas propagandas e nos posts que receberam em suas timelines, guiando, possivelmente, os seus respectivos votos (ESCÂNDALO FACEBOOK, 2023).

Também existem os casos em que as empresas se aproveitam desses dados - através dos cookies² - para oferecerem produtos específicos a depender das preferências listadas em suas informações pessoais, aumentando assim exponencialmente a sua possibilidade de sucesso nas vendas para grupos propensos a consumir uma determinada categoria de produtos (SELLEDGE, 2019).

E, finalmente, existem os casos em que os dados são capazes de ajudar na tomada de decisão de uma organização, no qual a junção cruzada de informações específicas apresentadas em Decision Boards - gráficos apoiadores na tomada de decisão - trazem insights valiosos no processo de realizar decisões importantes.

² Definição formal de cookies: [https://pt.wikipedia.org/wiki/Cookie_\(inform%C3%A1tica\)](https://pt.wikipedia.org/wiki/Cookie_(inform%C3%A1tica))

1.1 MOTIVAÇÃO

O estudo de caso base para a idealização desse sistema foi o de reclamações no site Reclame Aqui com relação à empresa Baterias Moura, que é a maior empresa brasileira atuante no mercado de baterias para automóveis da América Latina (BATERIAS MOURA, 2023) e vencedora de quatro premiações anuais na categoria de "Peças e acessórios automotivos" do próprio Reclame Aqui (PREMIO RA, 2022, 2021, 2020, 2019).

Mesmo com esse histórico campeão no atendimento ao cliente dos últimos anos, ainda existem algumas dificuldades sendo enfrentadas no processo de filtragem e entendimento das reclamações no site do Reclame Aqui, o que torna o processo de tomada de decisão sobre a análise desses dados algo complicado de ser realizado, principalmente por serem baseados em duas principais dificuldades, a de Interação e a de Custo:

- **Interação:** A empresa não consegue interagir diretamente com os dados que estão no Reclame Aqui, pois a listagem que existe no site é muito básica e não oferece uma ampla diversidade de filtros que ajudariam à nível de suporte na tomada de decisão.
- **Custo:** A empresa já entrou em contato com o Reclame Aqui para entender o quanto poderia custar uma consulta no banco de dados para retornar as suas informações de reclamações e foi informada que o preço é bastante elevado, passando até mesmo dos R\$25.000,00 (vinte e cinco mil) reais.

Diante dessa situação, como fonte geradora de informações, foi desenvolvido pela professora co orientadora deste trabalho um algoritmo de web scraping capaz de captar todas as informações que já estão públicas na página do Reclame Aqui e juntar todas elas em uma base de dados.

A partir dessa base de dados gerada, o projeto desenvolvido é responsável por sumarizar todas essas informações de forma a facilitar a observação e tomada de decisão de qualquer empresa que venha a utilizar a plataforma, resolvendo não apenas o problema da Baterias Moura, mas também um problema geral que outras empresas sentem ao tentar estudar melhor os seus dados de reclamações na

plataforma do Reclame Aqui.

Em função disso, esse trabalho tem como motivação principal construir um software que seja capaz dar suporte às empresas juntando os conceitos de extração de dados com o da elaboração de ferramentas de apoio à decisão, oferecendo decision boards capazes de tornar esse processo mais dinâmico e menos burocrático.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O objetivo geral deste trabalho de conclusão de curso é criar um SAD - Sistema de Apoio à Decisão - responsável por consumir os dados de uma base de dados, oferecendo para o usuário final interações com os dados, visualizações em Decision Boards e suporte no processo de tomada de decisão estratégico intra organizacional.

1.2.2 OBJETIVOS ESPECÍFICOS

- Tornar o projeto compatível com a base de dados gerada a partir do script de web scraping desenvolvido pela professora co orientadora deste trabalho.
- Criar a plataforma que possibilite a visualização personalizada dos dados oriundos da base de dados;
- Criar Decision Boards interativos capazes de filtrar informações que não são possíveis de serem alcançadas utilizando apenas o sistema Reclame Aqui.
- Tornar o projeto funcional nos ambientes Android, IOS e Web.

1.3 ESTRUTURA

O trabalho está estruturado seguindo a seguinte ordem das seções:

- Revisão de Literatura: Nesta primeira sessão o objetivo será listar as bases e as revisões literárias referenciais do que foi utilizado durante a implementação;
- Metodologia: Todo o método aplicado projeto será descrito na seção de metodologia;
- Resultados: Apresentar uma visão dos resultados obtidos com o desenvolvimento do projeto e o como ele de fato atinge o objetivo central do estudo.
- Conclusão: Na conclusão são feitas as considerações finais sobre os resultados alcançados, a importância dele referente ao cenário que é utilizado, e também lista as possíveis melhorias que podem ser implementadas futuramente.

2. REVISÃO DA LITERATURA

Dentro deste TCC é abordado um fluxo de desenvolvimento de uma solução que parte inicialmente de uma problemática administrativa que possui uma solução tecnológica como explicado na introdução deste trabalho.

Na revisão da literatura será apresentado e detalhado conceitos base que são necessários para o entendimento completo de como a solução foi construída. Desta forma, é importante que o leitor consiga compreender os termos administrativos utilizados ao longo do estudo e as suas aplicações no cenário de tecnologia.

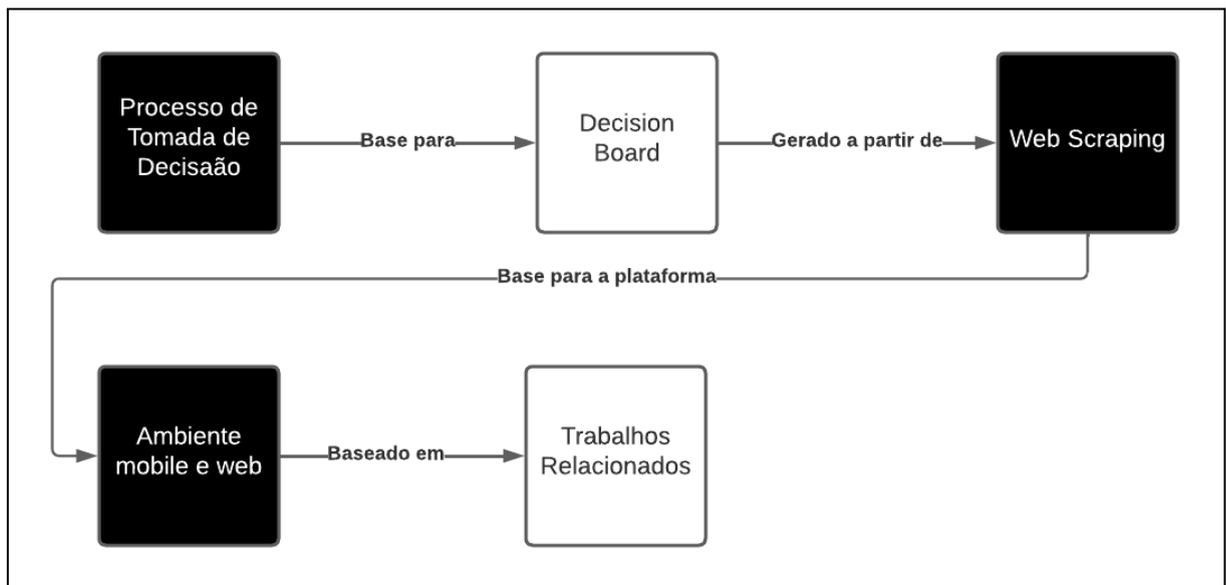
Seguindo uma ordem lógica, primeiro será apresentado o que é o **processo de tomada de decisão**, pois ele é o fundamento principal para o qual os **decision boards** foram construídos (este que é de fato a solução tecnológica para a problemática).

Analogamente, após o entendimento base desses dois conceitos, será fundamental entender o que é a técnica de coleta de dados **web scraping**, pois ela é o motor gerador de todos os dados utilizados na aplicação construída.

Seguindo o fluxo da construção das ideias bases deste trabalho, também é fundamental o entendimento aprofundado do que são os **ambientes mobile e web** de desenvolvimento de software, tendo em vista que a plataforma foi desenvolvida com o objetivo de suprir as necessidades tanto para usuários de dispositivos móveis quanto para aqueles que utilizam computadores e notebooks.

Por fim, é de suma importância ter em mente quais são os **trabalhos relacionados** ao que foi desenvolvido neste trabalho, pois dessa forma é possível notar os diferenciais que a plataforma desenvolvida possui sobre os demais e também entender em como eles foram bases referenciais para a construção do software (Figura 1).

Figura 1 - Fluxo cronológico da revisão literária.



Fonte: O autor (2023)

2.1 PROCESSO DE TOMADA DE DECISÃO

O processo de tomada de decisão é um processo formal que toda organização passa para poder definir quais caminhos serão seguidos a partir de uma escolha executada. Muitas vezes esse processo fundamental no desenvolvimento de qualquer organização pode ser ou não suportado por ferramentas que dão apoio nesse momento importante, sendo elas chamadas de apoiadores da tomada de decisão.

Não existe exatamente uma classificação formal e limitada de ferramentas que são consideradas apoiadores no processo de tomada de decisão, porém é sabido que tudo aquilo que oferece insights à cúpula estratégica de uma organização e que, com isso, traz apoio na decisão tomada embasada em dados e informações, pode de fato ser considerado uma ferramenta de apoio a tomada de decisão (ANDERSEN, 2019).

Tendo isso em vista, atualmente grande parte das ferramentas de apoio à decisão são desenvolvidas no cenário digital, pois é nele que está a maior parte das informações concretas que são fundamentais para uma tomada de decisão embasada na realidade atualizada. Os sistemas desenvolvidos nesse ambiente são chamados de SAD - Sistema de Apoio à Decisão - pois o seu objetivo central é sintetizar os dados para oferecer de forma visual um suporte no entendimento do que aquelas informações estão passando para a organização. A forma com o qual o sistema pode apresentar esse apoio na decisão pode ser através de transcrições subjetivas e resumidas, imagens, quadros e principalmente gráficos, os quais podem ser chamados de Decision Boards para casos em que eles tenham como objetivo central ajudar no processo de tomada de decisão (VOROBEEV, 2023).

2.2 DECISION BOARD

O termo "Decision Board" vem em tradução direta do inglês para o português como "Quadro de Decisão", o que significa ser um local no qual é possível adquirir informações úteis para o processo de tomada de decisão. Os sistemas Decision Board possuem como premissa principal oferecer - através de ferramentas digitais - fatores que ajudem quem está utilizando a ampliar a sua visão sobre determinado assunto e, conseqüentemente, receber suporte para melhorar o processo decisivo na tomada de decisão (ANDERSEN, 2019).

Apesar de atualmente existir uma base sólida para soluções baseadas em decision boards no cenário digital, essa é uma estratégia e modelo de solução que pode ser usada até mesmo de forma física em quadros, separando, alinhando e cruzando informações importantes que fornecerão juntas uma visão holística sobre a decisão e problemática abordada. Porém, essas são soluções simples que possuem limitações físicas na elaboração de ferramentas mais complexas como a de gráficos complexos e interativos em escala maior (Figura 2).

Figura 2 - Exemplo de decision board simples.

		<div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; text-align: center;"> <small>HINT: KEEP IT SHORT AND CONCISE</small> </div>	
STEP 1: Core Information			
Decision name	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>Name the decision</i></div>		
Description	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>What are the main reasons?</i></div>		
Collaborators	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>Collaborator 1</i></div>	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>Collaborator 2</i></div>	
Stakeholders	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>Stakeholder 1</i></div>	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>Stakeholder 2</i></div>	
Due date	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>MM/DD/YYYY</i></div>		
Status	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>Done / Started / Paused / Rejected</i></div>		
Outcome	<div style="background-color: #c6e0b4; padding: 5px; text-align: center;"><i>What conclusions have you reached?</i></div>		

Fonte: Devpost (2021)

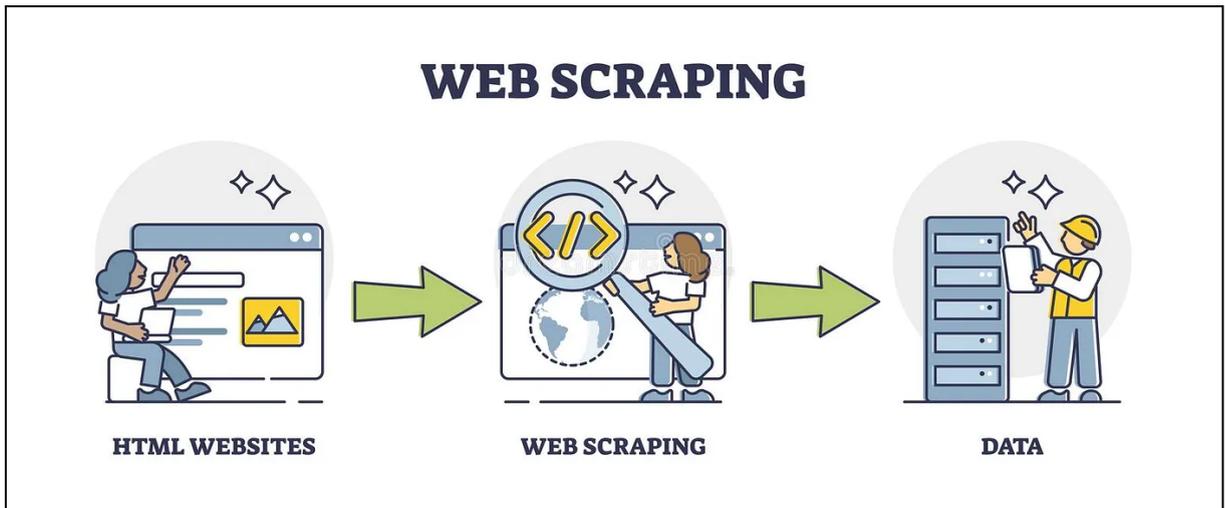
No sistema desenvolvido neste TCC, as ferramentas desenvolvidas para dar suporte ao processo decisivo foram os gráficos interativos, tendo como base principal as suas motivações iniciais de dar suporte às organizações que estão buscando entender e estudar mais a fundo o cruzamento de informações chaves de suas bases de dados (DABAB, 2021).

2.3 WEB SCRAPING

Web scraping³ vem do inglês "Raspagem da web", significando que o script irá fazer uma leitura e, simultaneamente, colhendo as informações que estejam disponíveis na página web em que está sendo executado. Essa é uma técnica utilizada amplamente na internet com o objetivo principal de coletar informações que estão públicas na web e utilizá-las para análise futuras (Figura 3).

³ Endereço eletrônico para web scraping: https://pt.wikipedia.org/wiki/Coleta_de_dados_web

Figura 3 - Fluxo do Web Scraping.



Fonte: Medium - Web Scraping em páginas dinâmicas com Python e Selenium (2022)

No caso do projeto desenvolvido nesse TCC, o web scraping faz parte da fonte de alimentação dos dados que vão diretamente na forma crua para o aplicativo, onde eles são tratados, formatados e apresentados de uma forma visível e palpável.

O script de web scraping usado para gerar as bases de dados que são utilizadas pela aplicação foi desenvolvido pela professora co orientadora antes do início do desenvolvimento deste trabalho, o qual foi configurado para fazer a raspagem dos dados do site Reclame Aqui com o objetivo de extrair as informações e poder colocá-los em um sistema capaz de gerir todos esses dados, oferecendo insights para quem esteja em busca de entender melhor os feedbacks dos clientes sobre as reclamações disponibilizadas publicamente no Reclame Aqui.

O output⁴ do script é uma database no formato CSV⁵ separado em 8 principais campos, sendo eles e em ordem:

1. Identificador: É o identificador único de cada reclamação.
2. Título: Título da reclamação que fica em destaque.
3. Datahora: Data e hora do momento exato em que a reclamação foi feita.
4. Local: Cidade e estado de onde a reclamação foi feita.

⁴ A saída, o produto da operação.

⁵ Significado de CSV: https://pt.wikipedia.org/wiki/Comma-separated_values

5. Status: Status da reclamação, podendo variar de não respondida, respondida, resolvida, em réplica e não resolvido.
6. Texto: Corpo base do texto da reclamação que o usuário fez.
7. Resposta: Caso a empresa tenha dado alguma resposta, esse campo vem em um formato parecido com o do Texto da reclamação do usuário.
8. Página: Simboliza em qual página estava aquela determinada reclamação no reclame aqui.

Para efeitos práticos, no aplicativo desenvolvido neste trabalho foram utilizados apenas os campos do número 2 até o 7 com o objetivo principal de lidar apenas com dados que possam gerar informações úteis à nível de apoio a decisão organizacional.

2.4 AMBIENTE MOBILE E WEB

Todo software é desenvolvido em algum ambiente específico que faça mais sentido para a sua aplicabilidade, e de forma geral existe o ambiente mobile caracterizado pelos dispositivos móveis como os smartphones, smartwatches e tablets, e o ambiente web, que é o utilizado em computadores com telas maiores, como os desktops e notebooks (Figura 4).

O ambiente mobile é, atualmente, o que mais cresce em número de usuários pela facilidade de acesso e valor de aquisição do dispositivo consideravelmente mais baixo. Com base nisso, diversas soluções digitais e empreendedoras vêm surgindo nesse setor, como o caso do Ifood, Uber e Nubank. Uma característica muito forte dos sistemas desenvolvidos para esse ambiente é que sejam leves, intuitivos e que possuam um cuidado especial na sua prototipação UX/UI (DESIGN THINKING, 2022), tendo em vista que os usuários só possuem uma pequena tela touchscreen para interagir com o sistema.

Já o ambiente web é tido como base de tudo que foi criado ao longo dos anos desde que a internet foi amplamente utilizada ao redor do mundo. Nesse tipo de ambiente é mais comum existirem sistemas que sejam mais complexos de se utilizar, pois o usuário geralmente vai dispor de um ferramental maior para a utilização do sistema, como teclado, mouse, monitores maiores e um hardware mais potente.

No sistema desenvolvido neste TCC foi escolhido desenvolver ele primeiramente para o cenário mobile utilizando da técnica "mobile first" (DUÓ, 2022), no qual as interfaces são criadas primeiramente para os dispositivos móveis e, posteriormente, corrigidas suas respectivas responsabilidades para telas maiores. O principal motivo pelo qual o sistema foi desenvolvido para os dois cenários pode ser dividido em suas respectivas vantagens:

- Mobile: Facilidade e velocidade no acesso dos dados. Ampla gama de possíveis usuários. Portabilidade de um decision board, o que o torna uma ferramenta bastante útil para uso no dia a dia.
- Web: Disponibilização de um espaço maior na interface, pois se trata de uma plataforma que possui gráficos, o que facilita a sua usabilidade. Ampliar a base de usuários, pois existe uma grande quantidade de pessoas que utilizam decision boards digitais.

Figura 4 - Exemplo visual dos ambientes mobile e web.



Fonte: Fission Labs (2019)

Desta forma, é esperado que com uma maior disponibilidade, também aumente a quantidade de organizações que possam usufruir do sistema para receberem um suporte no processo de decisão.

2.5 TRABALHOS RELACIONADOS

A seção de trabalhos relacionados tem como objetivo principal listar outros trabalhos que também são atuantes no cenário de sistemas de apoio à decisão baseado em dados.

2.5.1 MICROSOFT POWER BI

O Power BI é a ferramenta de Business Intelligence mais utilizada no mercado de sistemas de apoio à decisão orientada a dados (XP EDUCAÇÃO, 2022). Isso se dá pelo fato dela ser a mais completa em número de funcionalidades e interações, oferecer versão mobile, integração com diversos tipos de base de dados diferentes e, por fim, um dashboard completo com as mais recentes funcionalidades necessitadas pelo mercado. Ela foi desenvolvida pela Microsoft com o objetivo de revolucionar a forma com que as empresas desenvolvem as suas gestões baseadas em dados (POWER BI, 2023).

2.5.2 DATABOX

O Databox é uma ferramenta freemium⁶ que possui os mesmo objetivos do Power BI, porém preza pela simplicidade na interação com o sistema como um todo (DATABOX, 2023). Conta com diversas integrações para tipos de bases de dados diferentes e também possui ferramentas de interação com os dashboards, porém o usuário só consegue extrair o máximo de insights possíveis assinando o plano premium.

2.5.3 GOOGLE DATA STUDIO

Por fim, o Google Data Studio, que é a quarta plataforma de BI⁷ mais utilizada no mercado (XP EDUCAÇÃO, 2022) e isso se dá pelo fato dela ser completamente grátis e oferecer muitas integrações por parte da base de dados. O Google desenvolveu essa plataforma para, primeiramente, oferecer um nível de concorrência para o Power BI da Microsoft e, posteriormente, ter a sua participação nesse mercado de Business Intelligence que vem crescendo muito nos últimos anos (GOOGLE DATA STUDIO, 2023). Recentemente teve o seu nome alterado para Looker Studio, porém a ideia do sistema ainda se mantém a mesma: Ser uma

⁶ Freemium é um modelo de negócio no qual um produto ou serviço possui uma parte gratuita “free” e outra paga “premium”.

⁷ BI é a abreviação de Business Intelligence.

plataforma de BI gratuita capaz de, a partir de uma base de dados, criar relatórios e gráficos capazes de gerar insights valiosos para o tomador de decisão que estiver utilizando o sistema.

2.5.4 PROPOSTA DE VALOR

A proposta de valor deste trabalho de graduação se refere principalmente sobre a sua especificidade de utilização, ou seja, ele foi feito para gerar insights para um cenário específico, que é o das reclamações oriundas do Reclame Aqui. Além disso, foi desenvolvido com o objetivo de ser uma ferramenta grátis, de fácil acesso e como opção alternativa para que os tomadores de decisão possuam possibilidades diversas de interação e estudo dos seus dados oriundos do Reclame Aqui.

2.6 SUMÁRIO DA SEÇÃO

Com base no que foi apresentado nesta seção, espera-se que o leitor tenha compreendido todas as bases fundamentais que o projeto desenvolvido neste trabalho possui como referência literária. Seguindo a ordem de como foi dividida esta seção, inicialmente o objetivo foi demonstrar do que se trata o processo de tomada de decisão e sua importância, tendo em vista que o motivo principal desse trabalho é a melhoria e suporte nos processos decisórios organizacionais. Posteriormente, foi explorado o que são os Decision Boards, os seus motivos e aplicações na vida real, pois eles são os vetores do suporte à tomada de decisão buscada neste trabalho. Como fonte geradora das informações, foi fundamental se aprofundar no significado de Web Scraping, pois essa é uma técnica amplamente utilizada no mercado de software e serve principalmente para ajudar aqueles que buscam conhecimento e significado nos dados disponíveis na internet. Além disso, foi explicado o que são os ambientes web e mobile, pois estas são as bases em que o projeto foi desenvolvido, demonstrando as suas diferenças e potenciais únicos de cada um. Por fim, foram listados os trabalhos relacionados referenciais que serviram de farol estratégico no que se trata da direção para a qual o sistema seria desenvolvido.

Com base nas referências literárias, na próxima seção será explorada a aplicação dos conceitos fundamentais no caso de estudo deste trabalho, que é o desenvolvimento do sistema de apoio à decisão baseado em decision boards.

3 METODOLOGIA

Nesta seção serão explicados cada fase do processo de desenvolvimento do projeto, tendo como objetivo central oferecer ao leitor a possibilidade de entendimento e compreensão de todas as etapas do trabalho (Figura 5).

O ponto de partida inicial foi o desenvolvimento do script de web scraping que é a base fundamental geradora de todos os dados que o sistema utiliza para a geração dos decision boards.

Após isso foi observado a necessidade de se elicitar os requisitos funcionais e não funcionais do sistema, com o objetivo central de listar e organizar quais funcionalidades deveriam ou não ser desenvolvidas para o projeto.

Posteriormente, foi necessário se criar um protótipo em alta fidelidade para tornar o trabalho de codificação mais palpável e preciso, tendo em vista que construir telas sem nenhum parâmetro visual é um trabalho bastante complicado e oferece diversos riscos quanto à usabilidade do usuário final.

Após isso, com uma base mais sólida do que iria ser construído durante o desenvolvimento do software, foi decidido em qual metodologia de desenvolvimento de projetos iria ser escolhida, tendo como objetivo a agilidade e simplicidade no processo.

Com toda a base já preparada para a documentação e desenvolvimento dos processos, a tecnologia e linguagem de programação foram escolhidas, sendo elas as principais ferramentas de construção do projeto em si.

Junto com as tecnologias de desenvolvimento, foi necessário também escolher as ferramentas de versionamento do código, estas que são responsáveis por atualizar de forma padronizada todas as atualizações que forem feitas durante o processo de desenvolvimento do software.

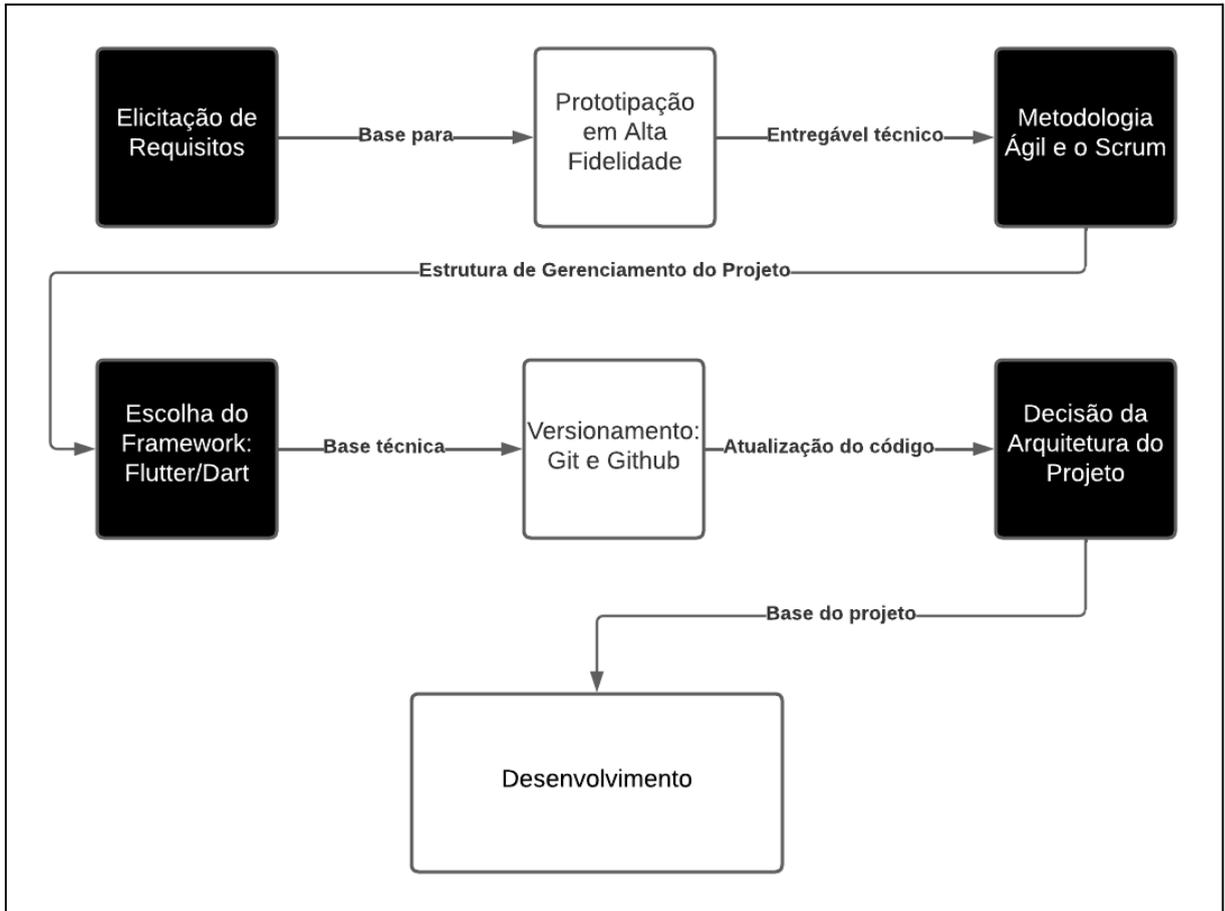
Por fim, foi decidido o tipo de arquitetura e as suas especificações técnicas sobre o qual o software será desenvolvido, com o objetivo de construir um projeto limpo, duradouro e capaz de receber atualizações ao longo do tempo.

Como base para apoio em todo o processo de desenvolvimento do projeto, foi utilizado o Whatsapp para comunicação direta, Google Meet para reuniões mais específicas, Google Drive para armazenamento de documentos, e a VPN⁸ do Centro

⁸ Definição formal para VPN: https://pt.wikipedia.org/wiki/Rede_privada_virtual

de Informática com o objetivo de acessar boas referências nas bibliotecas digitais, como a Science Direct⁹ e a IEEE Xplore¹⁰ por exemplo.

Figura 5 - Fluxo cronológico da aplicação da metodologia.



Fonte: O autor (2023)

3.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Antes do início do processo de desenvolvimento é necessário fazer a elicitação dos requisitos do sistema, tendo como objetivo central o estudo dos requisitos funcionais e não funcionais. Eles são úteis para documentar todas as ferramentas que serão desenvolvidas na plataforma final, dando ao leitor a possibilidade de entendimento do processo desde a sua concepção.

Por causa dessa necessidade, foi observado que, além de servir como uma ferramenta de documentação do projeto, também seria um artefato importante para

⁹ Endereço eletrônico para biblioteca digital Science Direct: <https://www.sciencedirect.com/>

¹⁰ Endereço eletrônico para biblioteca digital IEEE Xplore: <https://ieeexplore.ieee.org/Xplore/home.jsp>

guiar o processo de desenvolvimento do software, tornando mais claro o motivo principal de cada funcionalidade do sistema.

O objetivo final na construção dos requisitos do sistema é o de amadurecer e melhorar a qualidade final do software que será construído, pois *"It is critical because the quality of the systems is strappingly affected by the quality of the requirements"* (TEJAS, 2016).

3.1.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais são: "Em engenharia de software, um requisito funcional define uma função de um sistema de software ou seu componente." (REQUISITO FUNCIONAL, 2023). Ou seja, toda aquela funcionalidade chave que torne possível a realização do projeto será um requisito funcional para a construção da plataforma.

Quadro 1 - Requisitos funcionais.

ID	Requisitos Funcionais
RF 01	Compatibilidade com o script de Web Scraping
RF 02	Interação com a Database
RF 03	Listar Reclamações
RF 04	Listar Decision Boards
RF 05	Exibir Decision Boards
RF 06	Interagir com os Decision Boards

Fonte: O autor (2023)

- RF 01: A compatibilidade com o script de web scraping diz respeito à tornar o sistema desenvolvido capaz de receber as bases de dados que são geradas ao fim de sua execução;
- RF 02: A adição da base de dados é o requisito pelo qual o sistema deve ser capaz de interagir com a database, fazendo possível a leitura das informações contidas nela;

- RF 03: A listagem das reclamações é o requisito que se faz necessário que o sistema seja capaz de, após a interação inicial com a database, ele consiga tratar e tornar elas legíveis para o usuário final.
- RF 04: Listar Decision Boards é o requisito necessário para que seja possível fazer a listagem de todos os decision boards desenvolvidos no sistema.
- RF 05: Exibir Decision Boards é o requisito necessário para que seja possível ser feita a visualização prática dos decision boards desenvolvidos.
- RF 06: Interagir com os Decision Boards é o requisito necessário para que seja possível a interação em tempo real com os decision boards desenvolvidos.

3.1.2 REQUISITOS NÃO FUNCIONAIS

Já os requisitos não funcionais são responsáveis por listar de forma geral as necessidades que os usuários venham a ter em relação à segurança, desempenho, disponibilidade, manutenção entre outros fatores que possibilitem a utilização fidedigna de um software (REQUISITO NÃO FUNCIONAL, 2019).

Quadro 2 - Requisitos não funcionais.

ID	Requisitos não funcionais
RNF 01	O sistema deve possuir responsividade para diferentes tamanhos de tela
RNF 02	O sistema deve oferecer uma boa experiência de usabilidade e compreensão
RNF 03	O sistema deve possuir uma boa escrita de código e arquitetura para possíveis futuras manutenções
RNF 04	O sistema deve ser capaz de aumentar ou diminuir conforme necessário

Fonte: O autor (2023)

- RNF 01: O sistema deverá ser capaz de oferecer uma usabilidade estável e responsiva para os diferentes tamanhos de telas e dispositivos, oferecendo dessa forma uma versatilidade na utilização do software.

- RNF 02: Independentemente se o sistema estará sendo utilizado no ambiente mobile ou web, ele deverá oferecer uma boa interface e experiência de usabilidade e compreensão para que seja possível uma boa utilização.
- RNF 03: O código e a arquitetura do projeto devem ser bem elaboradas e documentadas para que seja possível a contribuição futura e uma boa longevidade do projeto.
- RNF 04: A arquitetura do projeto deve ser capaz de crescer ou diminuir em número de funcionalidades sem oferecer riscos à sua estabilidade, garantindo dessa forma um possível crescimento futuro do projeto.

3.2 PROTOTIPAÇÃO EM ALTA FIDELIDADE

A prototipação faz parte de um grande ciclo de concepção de produtos e artefatos digitais, que é o Design Thinking. Nessa metodologia o foco é imergir no problema, criar ideias a partir dos insights gerados, prototipar e, por fim, realizar o que foi imaginado (DESIGN THINKING, 2022).

O foco neste trabalho é sobre a terceira fase do Design Thinking, a de prototipação, que pode ser dividida em duas principais, a de baixa e alta fidelidade, no qual a escolha vai depender principalmente da situação em que o projeto se encontra. No caso deste trabalho, o tempo é um dos principais fatores críticos de sucesso para o desenvolvimento, portanto a escolha foi feita na direção do protótipo de alta fidelidade, com o objetivo de ter uma projeção em protótipo bastante aproximada da versão final o mais rápido possível.

O software base utilizado para criar todo o protótipo foi o figma (FIGMA, 2023), que é grátis e oferece todas as ferramentas necessárias para criar protótipos de interface para dispositivos mobile e web.

Seguindo o primeiro requisito não funcional definido na fase anterior da metodologia do trabalho, o sistema precisa ser responsivo¹¹ para diferentes tamanhos de telas, principalmente focando nos dispositivos móveis, como os smartphones por exemplo, e para os dispositivos maiores, como notebooks e monitores de computador.

Por conta dessa necessidade, a prototipação foi dividida em duas principais dimensões, a de telas menores (que contém principalmente os dispositivos móveis

¹¹ Definição formal para Responsividade: https://pt.wikipedia.org/wiki/Web_design_responsivo

como smartphones), e a de telas maiores (que contém os dispositivos como tablets, notebooks e computadores de forma geral).

3.2.1 RESPONSABILIDADE PARA TELAS MENORES

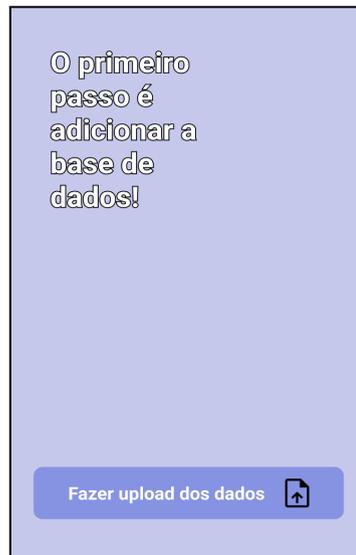
O sistema foi feito com base na estratégia mobile-first (PIMENTA, 2021) de desenvolvimento, portanto o fluxo mobile foi o que recebeu o foco principal no processo de prototipação e desenvolvimento.

O objetivo nas telas menores é conseguir oferecer ao usuário a maior facilidade no alcance dos elementos que estão na tela (PIMENTA, 2021), por conta disso a estratégia foi colocar sempre os elementos em forma de ícones em locais da tela que facilite o toque com uma mão.

O fluxo do sistema seguiu a seguinte ordem:

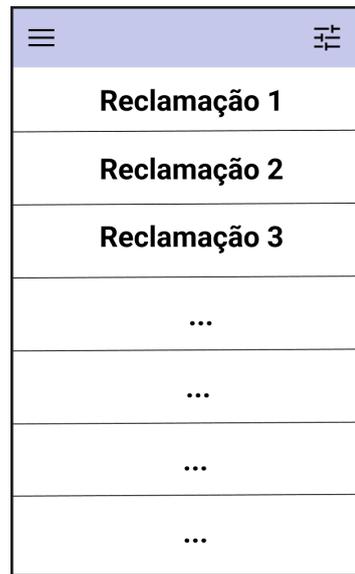
- Tela de adição da database;
- Tela de Listagem de reclamações;
- Listagem dos decision boards;
- Acesso e Interação com os decision boards.

Figura 6 - Mockup da tela Inicial do aplicativo versão mobile.



Fonte: O autor (2023)

Figura 7 - Mockup da tela de reclamações do aplicativo versão mobile.

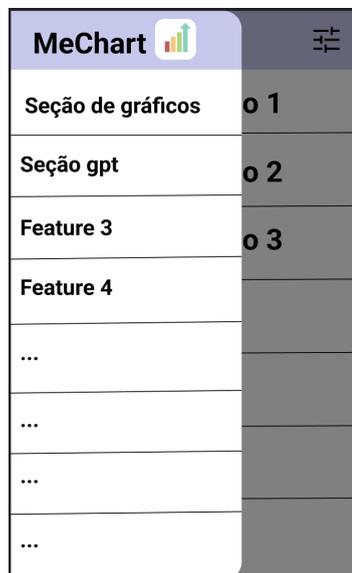


The mockup shows a mobile application interface for a complaints screen. At the top, there is a purple header bar containing a hamburger menu icon on the left and a list icon on the right. Below the header, the screen displays a vertical list of complaint items. Each item is contained within a white rectangular box with a thin black border. The first three items are labeled 'Reclamação 1', 'Reclamação 2', and 'Reclamação 3' in bold black text. The remaining four items are represented by three dots '...', indicating a scrollable list.

Reclamação 1
Reclamação 2
Reclamação 3
...
...
...
...

Fonte: O autor (2023)

Figura 8 - Mockup da tela de reclamações com a aba lateral aberta do aplicativo versão mobile.

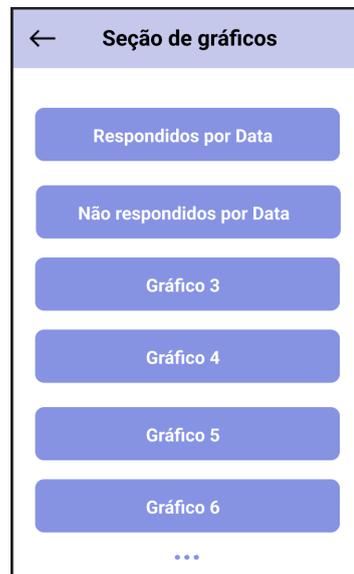


The mockup shows a mobile application interface with a side menu open. The top header bar is purple and contains the text 'MeChart' followed by a bar chart icon and a list icon on the right. The side menu is a white vertical bar on the left side of the screen, containing a list of menu items: 'Seção de gráficos', 'Seção gpt', 'Feature 3', 'Feature 4', and four items represented by three dots '...'. To the right of the side menu, the main content area is a dark gray vertical bar containing a list of items labeled 'o 1', 'o 2', and 'o 3', with the remaining items represented by three dots '...'. The side menu and main content area are separated by a vertical line.

Seção de gráficos	o 1
Seção gpt	o 2
Feature 3	o 3
Feature 4	
...	
...	
...	
...	

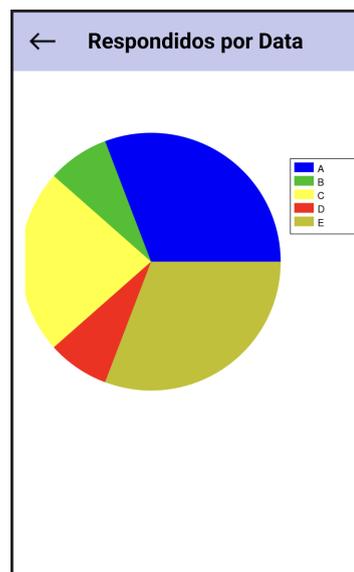
Fonte: O autor (2023)

Figura 9 - Mockup da tela de listagem dos decision boards do aplicativo versão mobile.



Fonte: O autor (2023)

Figura 10 - Mockup da tela de exibição dos decision boards do aplicativo versão mobile.



Fonte: O autor (2023)

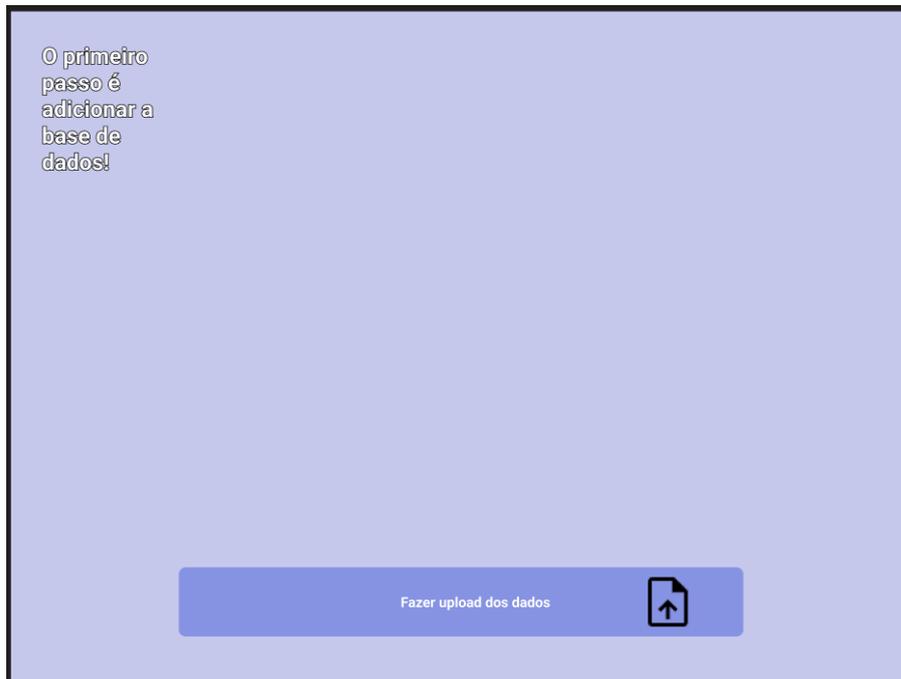
3.2.2 RESPONSABILIDADE PARA TELAS MAIORES

O objetivo principal na criação de telas responsivas para telas maiores foi no da ampliação da possibilidade de uso, tendo em vista que muitos usuários preferem interagir com gráficos em telas maiores, possibilitando uma usabilidade mais ampla pela maior quantidade de espaço livre na tela (PIMENTA, 2021).

A estratégia por trás da responsividade para telas maiores foi baseada em manter a lógica estrutural da versão mobile, e, posicionar os elementos de forma mais estratégica para as telas maiores. Porém, a estrutura básica não foi alterada, pois o objetivo é apenas fazer com que os elementos não fiquem fora de posição quando o tamanho da tela aumentar.

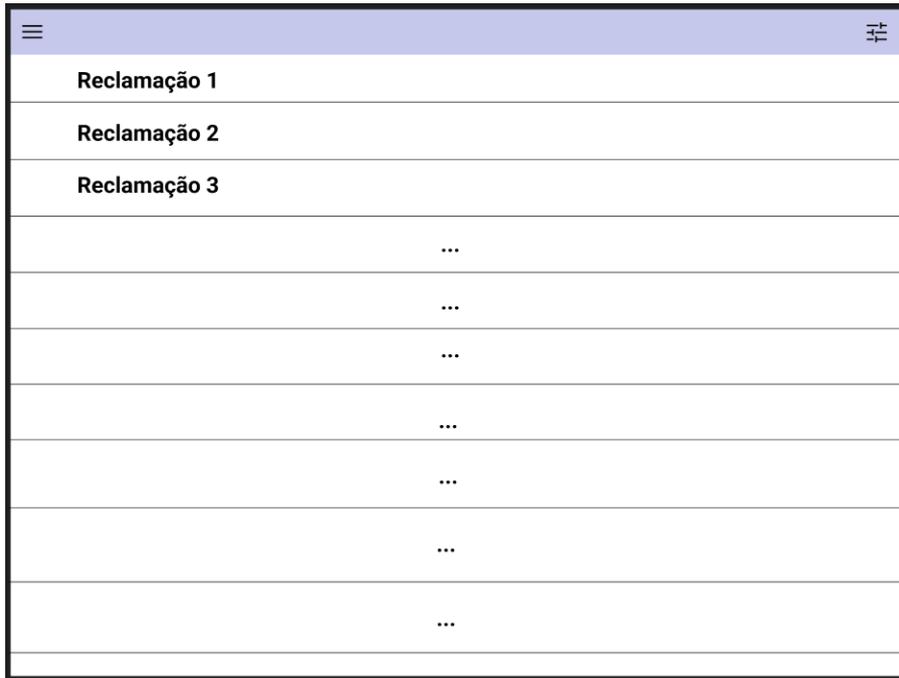
Apesar de não ser obrigatório, o fluxo de telas se manteve igual ao mobile, pois o objetivo é de integrar mais a usabilidade tanto para o usuário que irá utilizar a versão mobile quanto para o que irá utilizar a versão web.

Figura 11 - Mockup da tela Inicial do aplicativo versão web.



Fonte: O autor (2023)

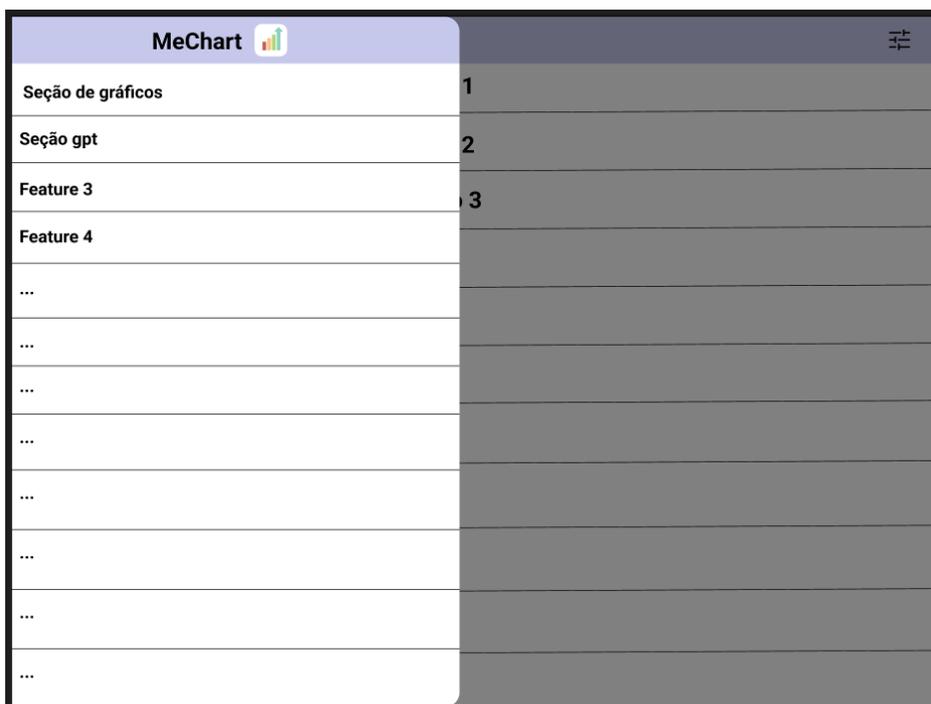
Figura 12 - Mockup da tela de reclamações do aplicativo versão web.



Reclamação 1
Reclamação 2
Reclamação 3
...
...
...
...
...
...
...
...

Fonte: O autor (2023)

Figura 13 - Mockup da tela de reclamações com a aba lateral aberta do aplicativo versão web.



MeChart	
Seção de gráficos	1
Seção gpt	2
Feature 3	3
Feature 4	
...	
...	
...	
...	
...	
...	
...	
...	
...	
...	

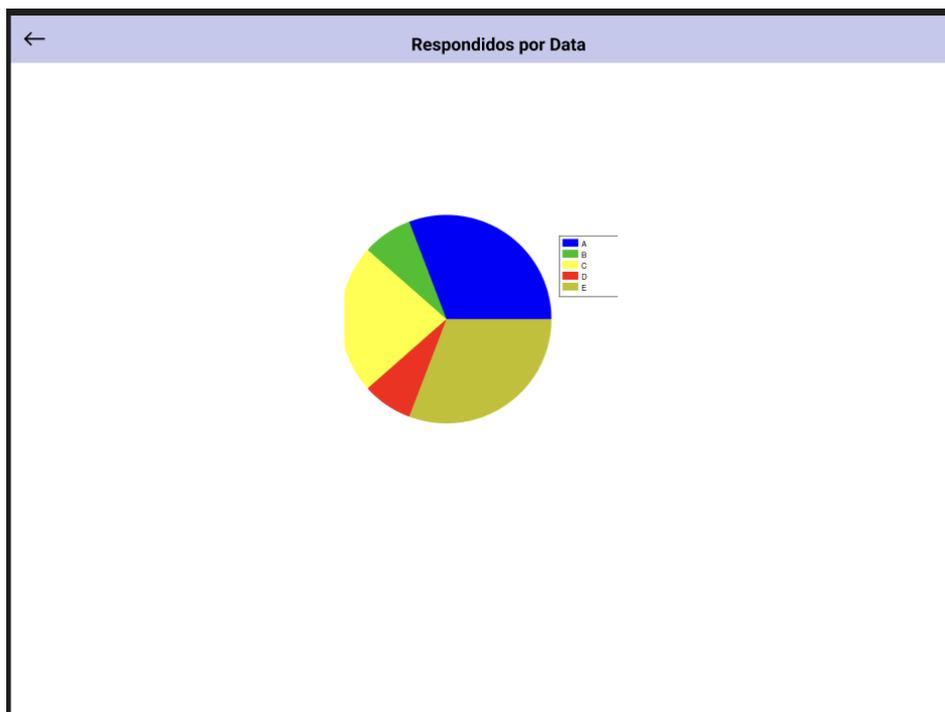
Fonte: O autor (2023)

Figura 14 - Mockup da tela de listagem dos decision boards do aplicativo versão web.



Fonte: O autor (2023)

Figura 15 - Mockup da tela de exibição dos decision boards do aplicativo versão web.



Fonte: O autor (2023)

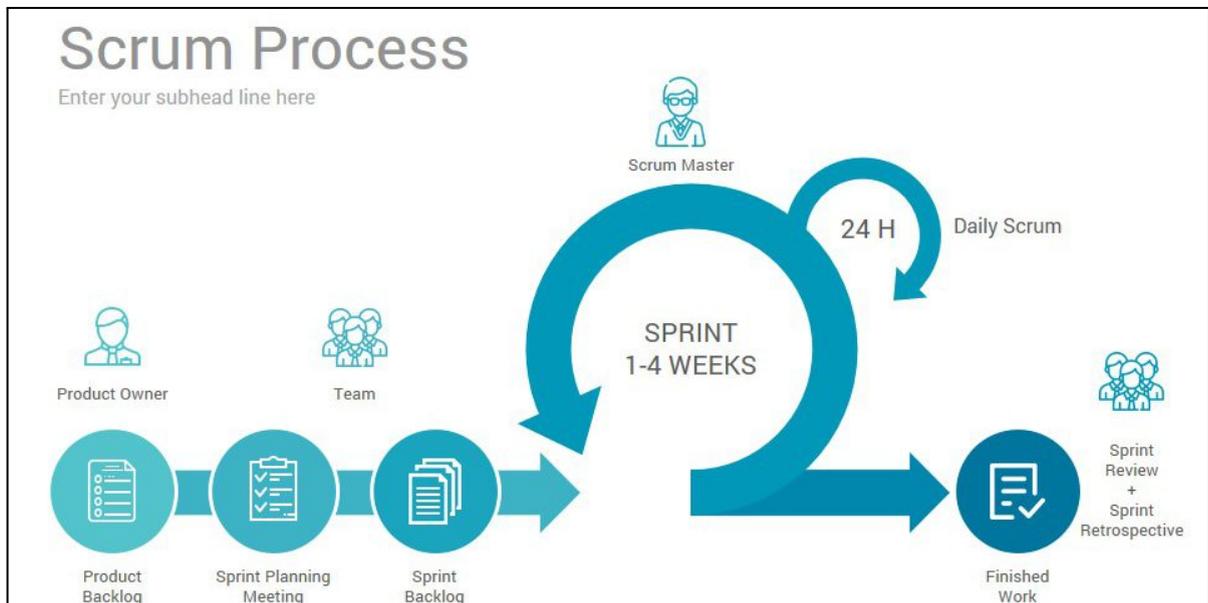
3.3 METODOLOGIA ÁGIL

A metodologia ágil de desenvolvimento de software é responsável por "é uma disciplina que estuda um conjunto de comportamentos, processos, práticas e ferramentas utilizados para a criação de produtos" (DESENVOLVIMENTO ÁGIL DE SOFTWARE, 2023), ou seja, é a base conceitual sobre o qual um time de desenvolvimento de software pode se basear para conseguir fazer a entrega bem sucedida de um produto.

Dentro da metodologia ágil existem frameworks¹² que ajudam nesse processo de concepção de produtos, e um dos mais famosos e utilizados no mundo todo é o Scrum, que oferece diversas metodologias de micro e macro gerenciamento nas equipes de engenharia de software (SCRUM, 2023).

O scrum possui um ciclo de vida subdividido em sprints, que marcam os pontos de entrega que podem variar de acordo com a necessidade do cliente ou com o poder de desenvolvimento da equipe desenvolvedora (Figura 16).

Figura 16 - Ciclo de vida do Scrum.



Fonte: Training Education Services (2018)

Porém, no caso do projeto desenvolvido neste trabalho, o ciclo do scrum foi alterado e simplificado para servir às necessidades do projeto. É importante ressaltar

¹² Definição formal de framework: <https://pt.wikipedia.org/wiki/Framework>

que essa é uma das vantagens do framework scrum de métodos ágeis, pois ele é capaz de se moldar de acordo com a necessidade da equipe, variando de grandes projetos de longo prazo de entrega para pequenos projetos com tempo de desenvolvimento curto, como é o caso deste trabalho de graduação.

As principais ferramentas utilizadas nesse processo interno de gerenciamento de projeto foi o whatsapp¹³ como meio de troca de informações e reuniões assíncronas e trello¹⁴ para documentação das atividades. A descrição de cada processo do ciclo de vida do scrum do projeto ficou detalhado da seguinte forma:

- Product backlog: Definido no início do projeto, o backlog total ficou documentado na plataforma Trello.
- Sprint Planning Meeting: A planning foi realizada antes do início de toda sprint, definindo todos os pontos que iriam ser desenvolvidos naquele sprint.
- Sprint Backlog: O backlog da sprint foi realizado logo após as definições da planning, tendo como objetivo central separar quais atividades seriam desenvolvidas na sprint seguinte.
- Sprint: A sprint foi definida para entregas semanais, com as dailies definidas para acontecer apenas no final da entrega com o objetivo de atualizar tudo o que foi desenvolvido durante a sprint.
- Finished Work: O trabalho finalizado era repassado via mensagem e atualizado no repositório do GitHub¹⁵, que será mais profundamente definido na seção 3.5 deste trabalho.

As principais alterações feitas na estrutura do ciclo de vida do scrum neste projeto foram a da redução na frequência das dailies, pois o nome indica que devem ocorrer diariamente, porém, por causa do curto tempo de desenvolvimento, ficaram ocorrendo semanalmente ao final da sprint, podendo ser chamada de "*weekly*". Além dessa alteração, a maioria das reuniões ocorreram de forma assíncrona pelo whatsapp, apenas as primeiras mais voltadas para a concepção da solução ocorreram de forma presencial síncrona e digital síncrona.

¹³ Endereço eletrônico para a ferramenta whatsapp: https://www.whatsapp.com/?lang=pt_BR

¹⁴ Endereço eletrônico para a ferramenta trello: <https://trello.com/pt-BR>

¹⁵ Endereço eletrônico para o Github: <https://github.com/>

3.4 FLUTTER E DART

O flutter é um framework cross-platform¹⁶ open source¹⁷ mantido pela google capaz de oferecer ao desenvolvedor de software um ambiente completo para a construção de frontends¹⁸, seja ele para dispositivos portáteis, como smartphones e tablets, ou para dispositivos maiores, como notebooks e computadores.

O fato do flutter ser capaz de criar aplicações que funcionam, simultaneamente e sobre um mesmo código base, tanto para o ambiente mobile quanto web fez com que a escolha do framework tornasse possível o desenvolvimento, em tempo hábil, de todas as funcionalidades que foram listadas nos requisitos funcionais e não funcionais descritos na seção 3.1. Além dessa versatilidade, ele conta com uma documentação completa (FLUTTER, 2023) que serve tanto para desenvolvedores experientes quanto para iniciantes, o que amplia ainda mais a possibilidade de contribuições futuras no projeto.

A linguagem de programação que o framework flutter utiliza como base de desenvolvimento é o dart, que é uma linguagem fortemente tipada, estática e inferida (DART LINGUAGEM DE PROGRAMAÇÃO, 2023), garantindo que os tipos das variáveis declaradas serão verificadas em tempos de compilação, e não em tempo de execução, evitando assim uma gama de possíveis erros que poderiam a vir ocorrer como acontecem em projetos baseados na linguagem javascript e python por exemplo.

O dart surgiu inicialmente com o objetivo de substituir o javascript, porém com o surgimento de diversos frameworks, bibliotecas e adaptação do mercado e da comunidade com a linguagem javascript, o dart acabou não conseguindo realizar essa substituição. Porém, anos depois em 2018, com o lançamento da versão 2.0, a linguagem dart ganhou destaque principalmente por causa da sua integração com o framework flutter, que vem crescendo rapidamente nos últimos anos devido a sua estrutura limpa, fortemente orientada a objetos e as rápidas atualizações frente às necessidades dos desenvolvedores e dos usuários.

Além disso o dart possui, similarmente ao flutter, uma documentação (DART DOCUMENTATION, 2023) rica em informações que são úteis para o desenvolvimento dos programadores, o que facilita no surgimento de novas

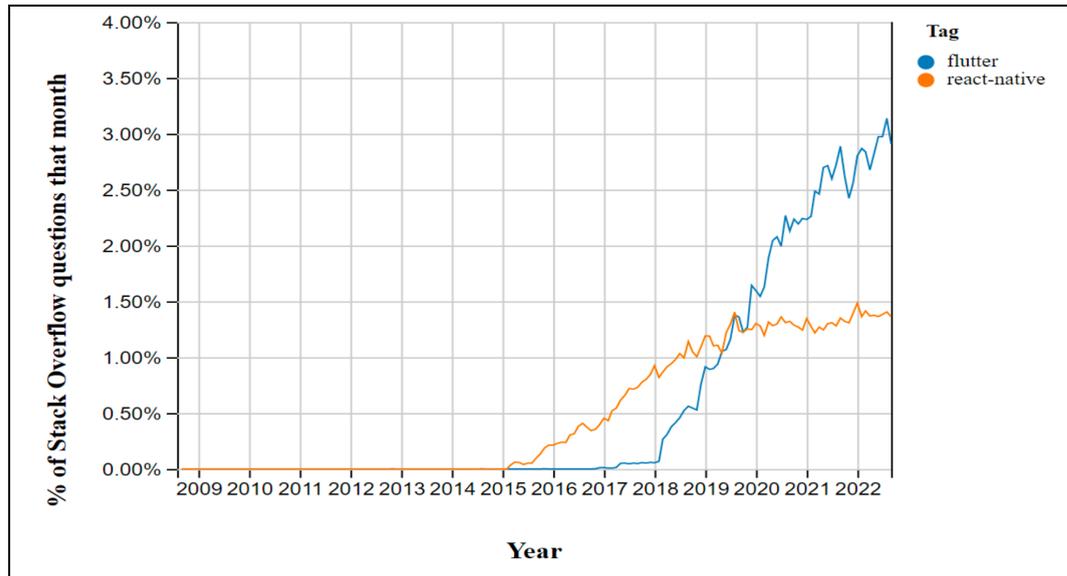
¹⁶ Definição formal de cross-platform: https://en.wikipedia.org/wiki/Cross-platform_software

¹⁷ Definição formal de open source: https://en.wikipedia.org/wiki/Open_source

¹⁸ Definição formal de frontend: https://en.wikipedia.org/wiki/Frontend_and_backend

bibliotecas e repositórios, fazendo com que a comunidade em torno da stack¹⁹ flutter/dart venha se tornando mais forte a cada ano²⁰ (Figura 17).

Figura 17 - Tendência de crescimento do Flutter em comparação ao seu concorrente.



Fonte: creative-hustlers²¹ (2022)

3.5 GIT E GITHUB

Uma parte fundamental no processo de criação de software é o versionamento do código, que tem por objetivo oferecer ao time de desenvolvimento a possibilidade de atualizar o software, simultaneamente e para diferentes destinos, sem oferecer riscos de perda ou desorganização no processo de merge²².

Atualmente existem diversas tecnologias de versionamento de código, porém a stack que tem mantido a sua posição como a mais atualizada nos últimos anos é a do Git (GIT, 2023) com o Github (GITHUB, 2023), no qual o primeiro é a CLI²³ para

¹⁹ Definição formal para stack de programação: <https://www.devmedia.com.br/quem-quer-ser-um-programador-fullstack/38786#:~:text=No%20contexto%20do%20desenvolvimento%20de,frameworks%20e%20bancos%20de%20dados.>

²⁰ Na figura a seguir é apresentado a linha de crescimento do Flutter em comparação ao seu principal concorrente de mercado, o react-native.

²¹ Tendência de crescimento do flutter para 2023: <https://www.creative-hustlers.com/post/will-flutter-still-be-popular-in-2023>

²² Merge vem do inglês e significa imergir, fundir. No caso se refere ao processo de junção de códigos.

²³ Command-Line Interface, significa Interface de linha de comando, que são os softwares que funcionam por terminal como o Bash do Linux ou PowerShell do Windows.

controle das versões do código e o segundo é o HUB²⁴ online que serve como repositório digital para armazenamento de projetos.

Como decisão do projeto, a stack Git e Github foi a escolhida para ser utilizada no processo de atualização e armazenamento do código fonte do sistema, tendo como base toda a comunidade e informações disponíveis na internet para a solução de possíveis problemas que pudessem surgir no andamento do desenvolvimento do software.

O fluxo de trabalho com essa stack se baseia na atualização de um ramo principal - conhecido popularmente como branch - chamado por default²⁵ de main. A ideia é que um time de desenvolvimento possua diferentes branches e, que no final do processo, façam o merge no ramo principal main para atualizar tudo o que foi desenvolvido sem oferecer os riscos de perder alterações importantes nos conflitos que possam surgir.

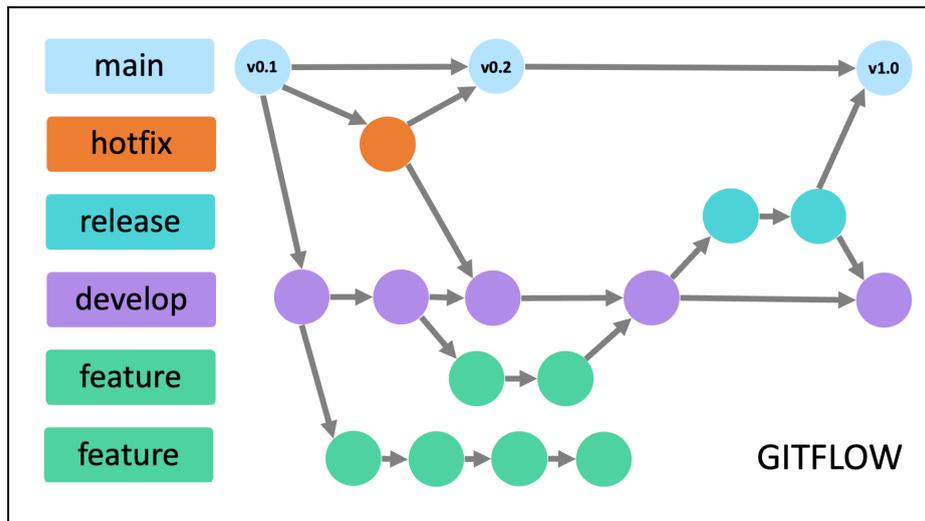
No caso deste trabalho de conclusão de curso, o time de desenvolvimento era de apenas uma pessoa, o que acabou não demonstrando a necessidade de criar complexos procedimentos prévios no processo de merge das alterações, com o trabalho sendo feito completamente apenas no ramo principal (Figura 18).

O projeto está disponível neste [repositório](#) público do github, possui uma documentação em inglês para ampliar a possibilidade de contribuintes e possui um passo a passo de como realizar os procedimentos de replicação do funcionamento do projeto.

²⁴ HUB é o ponto central, no caso se refere ao ponto central onde os repositórios online são armazenados.

²⁵ Por padrão, selecionado previamente sem intervenção de terceiros.

Figura 18 - Demonstração do fluxo de trabalho com a tecnologia git.



Fonte: KINTO Technologies (2023)

3.6 CÓDIGO E ARQUITETURA LIMPA

No processo de desenvolvimento de um software existem dois pontos que são fundamentais para que se obtenha sucesso, que são os de escolha da arquitetura sobre o qual o software será construído e a metodologia de escrita de código.

Ambos são fundamentais para garantir que o projeto possua uma maior longevidade, qualidade, manutenibilidade e, por fim, maior capacidade de contribuição por parte da comunidade de desenvolvimento (ROBERT, 2008, 2017).

3.6.1 CÓDIGO LIMPO

"Mesmo um código ruim pode funcionar. Mas se ele não for limpo, pode acabar com uma empresa de desenvolvimento." (ROBERT, 2008). Esse é o resumo do pensamento que Robert C. Martin defende sobre a importância da escrita de códigos verdadeiramente limpos na escrita de qualquer software.

O desenvolvimento de código limpo significa seguir conceitos e estratégias que façam aquele determinado código mais legível, auto explicativo, funcional, declarativo e conciso.

Escrever códigos limpos podem necessitar de mais tempo para serem desenvolvidos, tendo em vista que será necessário pensar mais na hora de escrever uma linha de código. Porém, como defendido por Robert C. Martin em sua obra Clean Code (ROBERT, 2008), o projeto será capaz de escalar e de ser atualizado ao

longo de vários anos, garantindo dessa forma uma longevidade para o time ou empresa que está diretamente relacionada com o software.

Para que um código seja verdadeiramente limpo, é importante que ele siga alguns padrões, que, juntos, são capazes de fazer o software funcional e limpo:

- Palavras com significados concretos: Todas as declarações como de variáveis, funções e métodos devem ser escritas de forma que sejam auto declarativas ao invés de letras ou apenas códigos que não expliquem explicitamente o seu motivo (exemplo: `phoneNumber` seria um nome de variável melhor para se referir a um número de telefone ao invés de `x` ou `var1`).
- Funções e métodos bem definidos: As funções e métodos muitas vezes são ferramentas fundamentais para o funcionamento de um código, portanto eles devem ser sempre responsáveis por uma única atividade e não podem criar efeitos colaterais em locais fora de seu escopo.
- Comentários: Os comentários só devem existir no código caso ele seja conciso e extremamente necessário (para casos em que a lógica seja muito rebuscada e que a leitura do código em si não consiga explicar todos os seus detalhes). A ideia por trás desse critério é que a nomeação correta das variáveis e funções, junto com outras técnicas de escrita de código façam dele ser auto explicativo.
- Formatação: A formatação do código se refere a organização, indentação²⁶, quebra de linha e tudo o que se refere à organização na escrita, similarmente como a formatação de documentos, livros e trabalhos acadêmicos.
- Controle de erros: Os erros devem ser gerenciados através de cláusulas try-catch²⁷, previstos e dinamicamente delineados, tornando o projeto de software menos suscetível a erros que travam completamente o seu funcionamento.

²⁶ Espaçamento na lateral direita dos códigos escritos, com o objetivo de deixar mais claro a ordem da chamada de cada código, no qual o que mais possui esse espaçamento esteja dentro do que possua menos.

²⁷ Cláusulas formais e globais a todas as linguagem de programação, onde o código tenta "try" realizar determinada operação e, caso ocorra alguma falha, o código pega "catch" esse erro com segurança sem quebrar a compilação ou o funcionamento do sistema.

- Testes: É fundamental que o projeto de software esteja coberto de testes unitários²⁸ e de integração²⁹, garantindo de forma automatizada todas as funcionalidades desenvolvidas.

No projeto desenvolvido nesse trabalho de conclusão de curso foram utilizados os métodos de escrita de código clean code para garantir que, junto com a arquitetura limpa - assunto que será discutido no próximo tópico número 3.6.2 - o projeto possua um bom grau de manutenibilidade, escalabilidade e leitura ao longo de seu ciclo de vida.

3.6.2 ARQUITETURA LIMPA

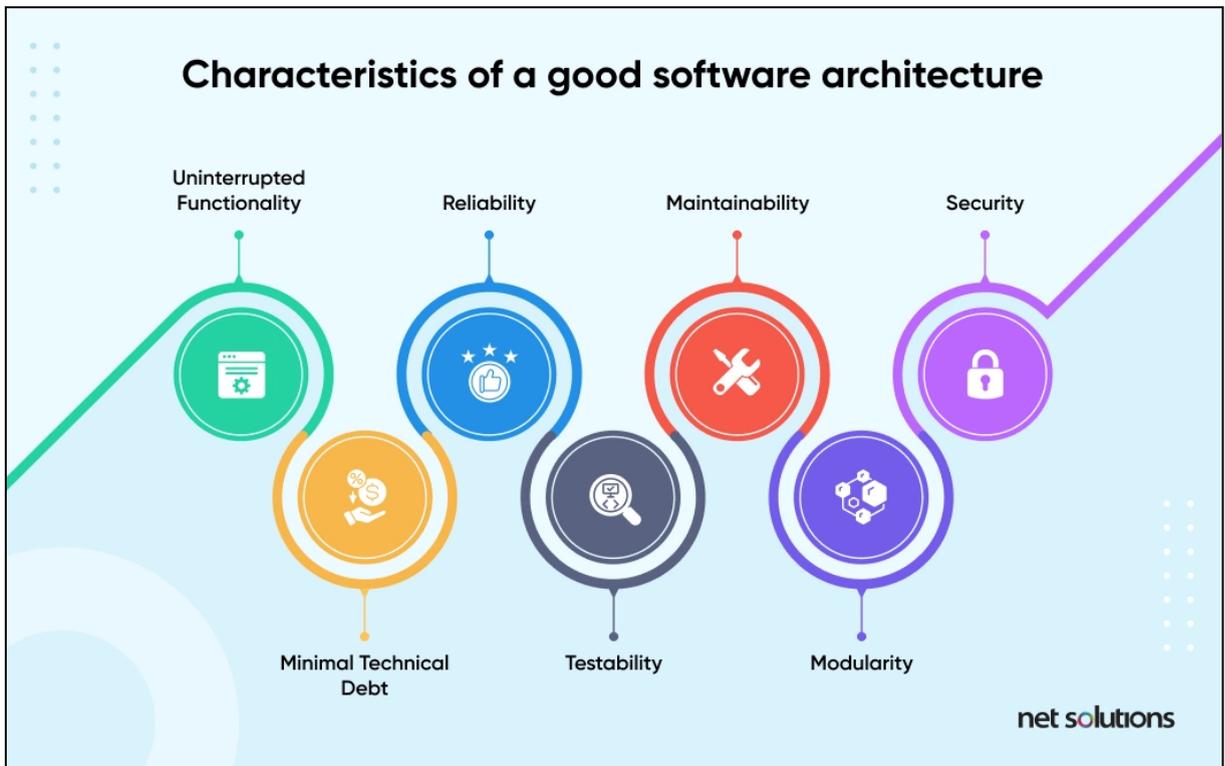
Em todo software construído existe uma base estrutural sobre o qual o programa é escrito, sendo popularmente chamada de "arquitetura", fazendo alusão com a área de construção civil, na qual a base arquitetônica é utilizada para se construir todas as etapas de uma casa ou prédio.

Essa arquitetura de software é fundamental para se estabelecer os padrões e regras de desenvolvimento do sistema com o objetivo de garantir mais organização, estabilidade e longevidade do código escrito (ARQUITETURA DE SOFTWARE, 2020). Além dessas vantagens, atualmente é imprescindível que um projeto possua uma base arquitetural (Figura 19), pois dessa forma é possível que novos contribuintes consigam se adaptar ao modo de operação interna do software mais rapidamente, trazendo mais celeridade e qualidade na entrega de novas funcionalidades, afetando positivamente o mercado de tecnologia mundialmente.

²⁸ Testes unitários se dedicam em garantir que uma funcionalidade em específico está funcionando de acordo com o esperado.

²⁹ Testes de integração se dedicam em garantir que um fluxo inteiro de funcionalidades estejam funcionando de acordo com o esperado.

Figura 19 - Características de uma boa arquitetura de software.



Fonte: Net Solutions (2023)

Dentro da área de arquitetura de software existe um caminho a ser seguido para construir boas arquiteturas que consigam entregar tudo o que a figura X apresenta, sendo conhecida na comunidade e introduzida por Robert C. Martin³⁰ como "Arquitetura Limpa".

O principal objetivo dessa metodologia de desenvolvimento é, como o próprio Robert descreve em sua obra Clean Architecture: "Você deve ser capaz de entender o funcionamento de todo um sistema olhando apenas para o seu código base" (ROBERT, 2017). A ideia por trás dessa afirmação advém da aplicação de outros conceitos em conjunto, como o SOLID³¹ e DRY³² na construção da base arquitetônica do software, pois dessa forma o código do sistema é escrito de uma maneira em que o desenvolvedor consegue compreender a lógica da manipulação

³⁰ Grande, importante e influente personalidade na área de desenvolvimento de software mundialmente. Endereço eletrônico para descrição detalhada de seu perfil: https://pt.wikipedia.org/wiki/Robert_Cecil_Martin

³¹ Definição formal de SOLID: <https://medium.com/desenvolvendo-com-paixao/o-que-%C3%A9-solid-o-guia-completo-para-voc%C3%AA-entender-os-5-princ%C3%ADpios-da-poo-2b937b3fc530>

³² Definição forma de DRY: https://en.wikipedia.org/wiki/Don%27t_repeat_yourself

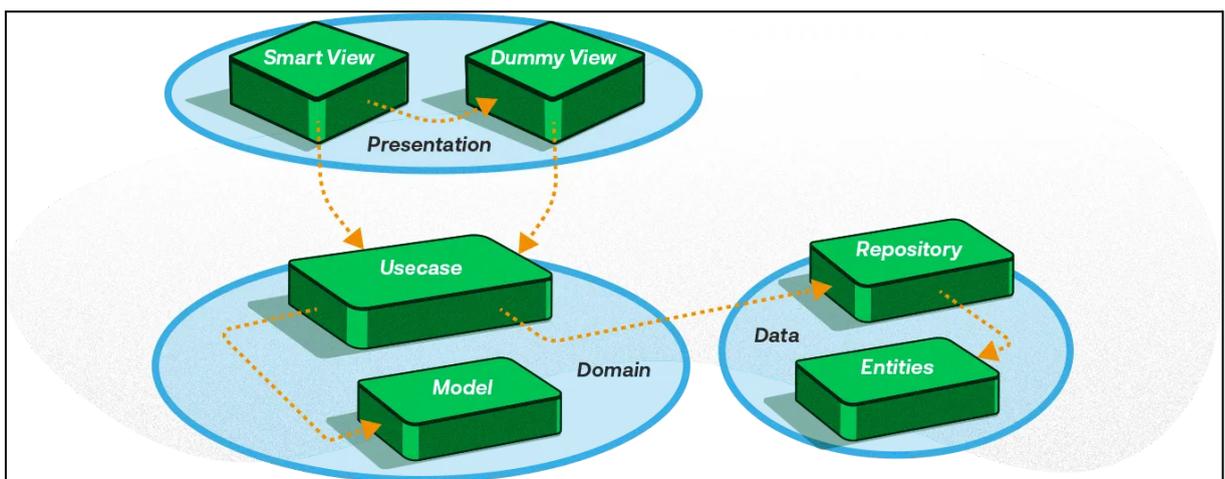
de todos os componentes da aplicação observando apenas os pontos-chaves do código fonte.

No projeto desenvolvido nesse trabalho de conclusão de curso o objetivo foi construir a base da arquitetura baseada na metodologia da clean architecture introduzida anteriormente. Kainã Terto explica detalhadamente em seu artigo "*A Viable Architecture choice for real world projects in Flutter*" (TERTO, 2021) o processo de escolha e decisão para a organização dos padrões que são utilizados nessa arquitetura limpa (Figura 20).

Os componentes do sistema ficam divididos em módulos, que internamente são subdivididos em três principais camadas:

- **Dados/Data:** Responsável por realizar as comunicações com os servidores e tratar todas as informações que vem dele, com o objetivo principal de enviar para a camada de domínio os dados que serão utilizados pela aplicação.
- **Domínio/Domain:** Responsável por tratar todos os casos de uso da aplicação e enviar todas as informações preparadas para a camada de apresentação.
- **Apresentação/Presentation:** Apresentar todas as informações que foram tratadas nas camadas anteriores e oferecer a interface de interação com o usuário.

Figura 20 - Representação visual da arquitetura de software utilizada.



Fonte: Capyba Software Studio (2021)

Dentro da camada de **presentation** é possível observar dois elementos, que são as smart e dummy views. As smartviews são responsáveis por controlar toda a navegação do módulo em que está localizada e ela funciona como a interface sobre a qual os outros módulos do sistema irão interagir. Já as dummy views são responsáveis apenas por apresentar visualmente a interface com o qual o usuário irá interagir, ou seja, enquanto uma se dedica na interação à nível de código, a outra se baseia na interação à nível usuário-sistema.

Paralelamente, na camada **domain** existe o usecase, que é o núcleo das interações com a regra de negócio da aplicação, e ele pode ser subdivido em três elementos, que são os de evento, estado e usecase. Os eventos são os responsáveis por disparar dentro do usecase a função requerida pelo usuário na interface. Os estados são responsáveis por manter e gerenciar todas as variáveis que são utilizadas naquele módulo. E, por fim, o usecase é o ponto central para onde os eventos enviam as requisições que foram feitas pelo usuário na interface e também é o local onde o estado é atualizado de acordo com o processamento das operações realizadas pelos eventos. Ainda no módulo de usecase existem as models, que são representações definitivas das entidades - que serão descritas no próximo parágrafo a respeito da camada de dados - e protegidas contra nullables³³, possuindo como principal objetivo oferecer ao usecase apenas as informações que fazem sentido para o módulo funcionar.

Por fim, existe também o módulo de **data**, que se subdivide em dois elementos, as entities e os repositórios. As entities são as entidades representativas dos dados que estão entrando na aplicação e seu principal objetivo é documentar e padronizar os tipos e chaves dos json³⁴ que foram recebidos pelos repositórios. Já os repository são responsáveis por fazerem as requisições que irão trazer as informações cruas, passar elas para o formato das entidades daquele módulo e, por fim, passar para o formato de uma modal para finalmente ser enviada para o usecase.

Dessa forma, o projeto consegue crescer em número de módulos indefinida e organizadamente, trazendo escalabilidade e estabilidade para o crescimento do

³³ Nullables são, em tradução direta, possíveis de se tornarem nulos. Isso se refere a um possível problema que todo sistema de software passa por, no qual é esperado um valor para uma variável, porém é retornado null por vários fatores que é impossível de ser documentado genericamente.

³⁴ JSON: Javascript Object Notation é um padrão de transferência de dados baseado em chave e valor, trazendo mais simplicidade e dinamismo no envio e recebimento de informações na internet mundialmente.

software. No escopo desse projeto, que inicialmente não demonstra a necessidade de tamanha escalabilidade, foi utilizada essa arquitetura para garantir uma longevidade e possibilidade de crescimento no número de novas funcionalidades que podem surgir de acordo com o amadurecimento do projeto.

3.7 SUMÁRIO DA SEÇÃO

O objetivo central desta seção foi detalhar e explicar todas as ferramentas e metodologias utilizadas ao longo do projeto, tendo como visão dar ao leitor a possibilidade de entender a nível técnico quais foram as estratégias adotadas para se alcançar o resultado final. Seguindo a ordem de como foi dividida esta seção, inicialmente foram explicados os requisitos funcionais e não funcionais responsáveis por listar as necessidades principais que o sistema requisitará para o seu bom funcionamento. Posteriormente, foi apresentada a prototipação em alta fidelidade das versões web e mobile, artefato este que serviu de base visual para a construção das telas no frontend com o Flutter. Seguindo a ordem, foi apresentado o como que foi aplicada a metodologia ágil de desenvolvimento de software no projeto, objetivando e dinamizando as entregas em pequenas etapas ao longo do processo. Após isso, foram apresentadas as ferramentas Dart e Flutter, que compõem o ecossistema da Google para construção de front-ends cross-platforms nativos utilizados na construção do sistema desenvolvido neste trabalho de graduação. Posteriormente, foram apresentadas as tecnologias de versionamento de código Git e GitHub, facilitadores no processo de atualização e gerenciamento de repositórios e versões de projetos, tornando possível o compartilhamento dinâmico da mesma base de código pela internet. Por fim, fundamentalmente foi explicada a arquitetura utilizada no software construído neste projeto, subdividindo as suas responsabilidades e tornando o entendimento das estratégias por trás da arquitetura mais claras para o leitor interessado compreender.

Com o entendimento de toda a metodologia aplicada no projeto e explicada nesta seção, espera-se que o leitor consiga entender os resultados alcançados que serão explorados na seção seguinte.

4 RESULTADOS

Como resultado final da implementação do sistema decision board proposto nesse trabalho de conclusão do curso, foi possível desenvolver um aplicativo funcional e responsivo para os ambientes IOS, Android e Web.

O aplicativo conta com todos os requisitos funcionais e não funcionais documentados na elicitação levantada no início do desenvolvimento do projeto, além de que também consegue trazer a visualização de dados necessários para serem gerados insights que dão suporte ao processo de tomada de decisão.

Além do desenvolvimento do sistema em si, toda a documentação necessária para se entender a arquitetura, decisões de projeto, e outros arquétipos da engenharia de software utilizada neste projeto estão documentados neste documento de TCC e no repositório³⁵ do github, pois o objetivo é que o leitor seja capaz de replicar ou entender todo o fluxo de decisões que foram tomadas ao longo do desenvolvimento tanto a nível teórico quanto prático.

Com base nisso, nas próximas seções serão explorados os fluxos desenvolvidos nos três principais ambientes IOS, Android e Web, demonstrando as funcionalidades desenvolvidas na aplicação.

4.1 FLUXO PRINCIPAL

O fluxo principal é igual nos três ambientes, alterando apenas detalhes visuais e de usabilidade, seguindo a seguinte ordem (Figura 21):

- Upload da Base de dados: Primeira etapa do fluxo, local no qual o usuário irá fazer o upload da sua base de dados das reclamações geradas pelo algoritmo de web scraping;
- Listagem das Reclamações formatadas: Nessa segunda etapa do fluxo principal, o app é responsável por pegar os dados que vem da base de dados de forma crua, trata, adiciona os tipos³⁶ e apresentá eles na tela;
- Listagem dos Decision Boards: A terceira etapa é a da listagem dos gráficos, sendo ela fundamental para a escolha de que gráfico irá ser analisado;
- Apresentação dos Decision Boards: Por fim, a visualização final dos Decision Boards, momento no qual será possível gerar insights e aprimorar o processo

³⁵ Endereço eletrônico para o repositório: https://github.com/Giovaniavs/decision_board_system

³⁶ Orientação a Objeto em Dart - Tipos: <https://dart.dev/language/type-system>

de tomada de decisão de acordo com a compreensão dos dados apresentados.

Uma feature que foi desenvolvida, porém, não listada no fluxo base é a de “Dados Persistentes”, significando que uma vez que o usuário faz o upload da sua database, após ele entrar no aplicativo novamente ela já estará pré-carregada, evitando a necessidade de fazer o upload sempre que fechar o aplicativo. Os dados ficam armazenados no próprio smartphone no flutter secure storage³⁷, que é um local seguro e criptografado que apenas o sistema consegue acessar.

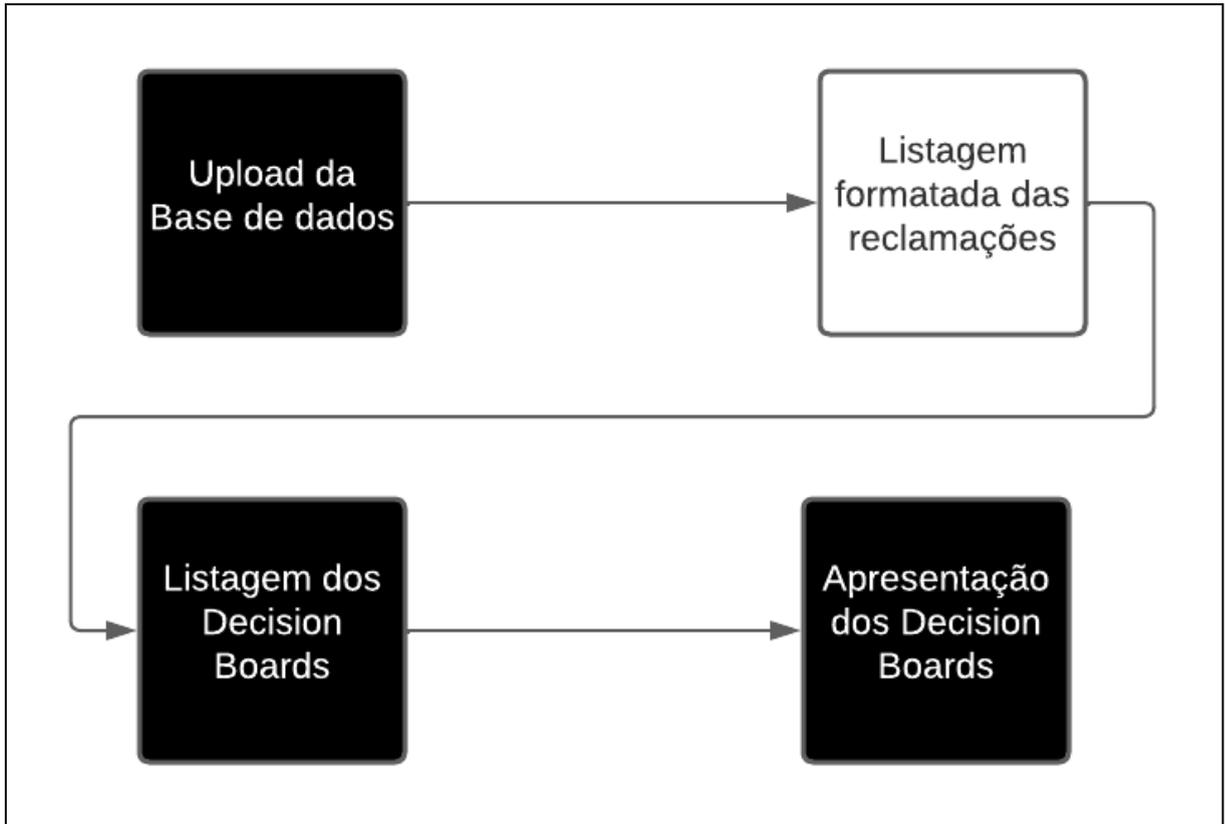
Na versão web não foi possível manter essa funcionalidade de persistência de dados pelo fato de que os navegadores não permitem um armazenamento de dados tão grande em seus “local storages”³⁸ como nos aplicativos móveis. Nesse caso, sempre que o usuário web quiser fazer a visualização dos seus dados nos Decision Boards, será necessário fazer o upload do arquivo na tela inicial sempre que abrir a plataforma em seu navegador.

Outro detalhe fundamental que foi desenvolvido é o processo de tratamento e falha do upload da base de dados, pois o app foi desenvolvido para trabalhar apenas com a database gerada pelo script de web scraping, ou seja, caso o usuário tente fazer o upload de algum arquivo que não esteja conforme a estrutura base do .csv que é gerado ao final do script, o aplicativo irá disparar erros tratados para o usuário o avisando disso.

³⁷ Endereço eletrônico para flutter secure storage: https://pub.dev/packages/flutter_secure_storage

³⁸ Local storage vem da tradução direta do inglês para português como “Armazenamento Local”, é uma forma que os navegadores possuem de liberar um espaço da memória para o salvamento de informações simples, tais como credenciais e identificadores únicos.

Figura 21 - Representação visual do fluxo de navegação no sistema.

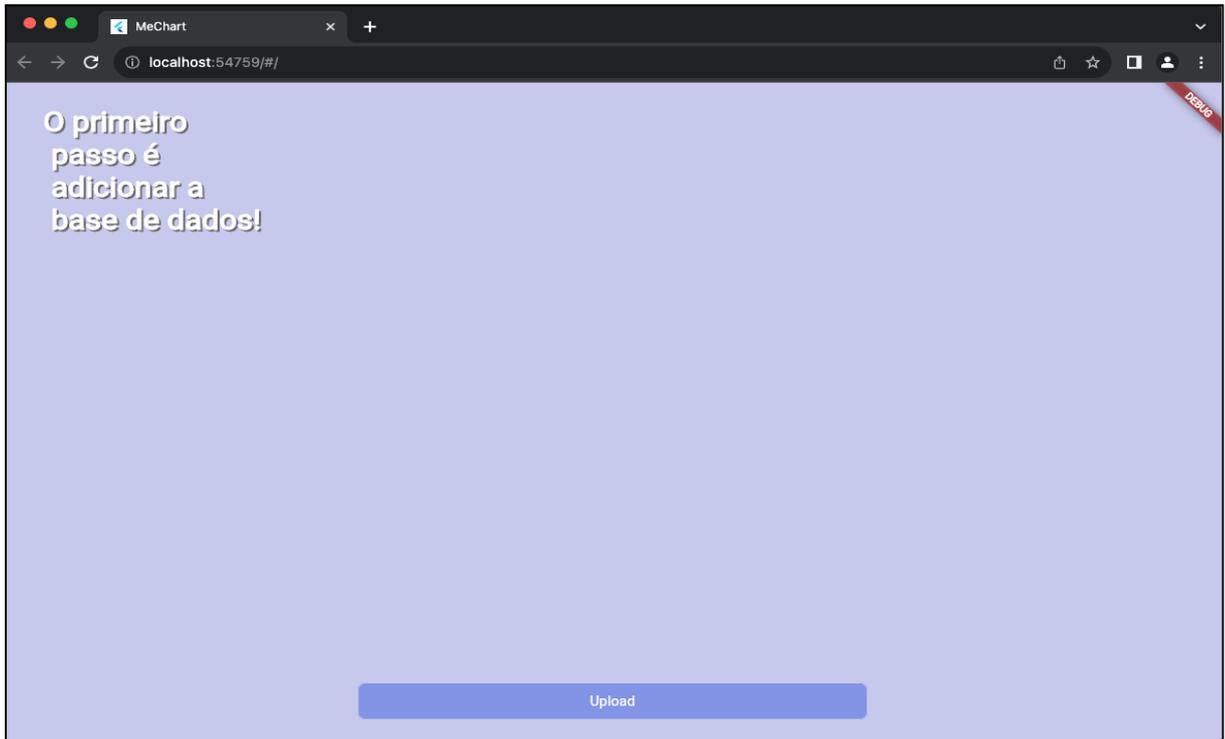


Fonte: O Autor (2023)

4.1.1 WEB

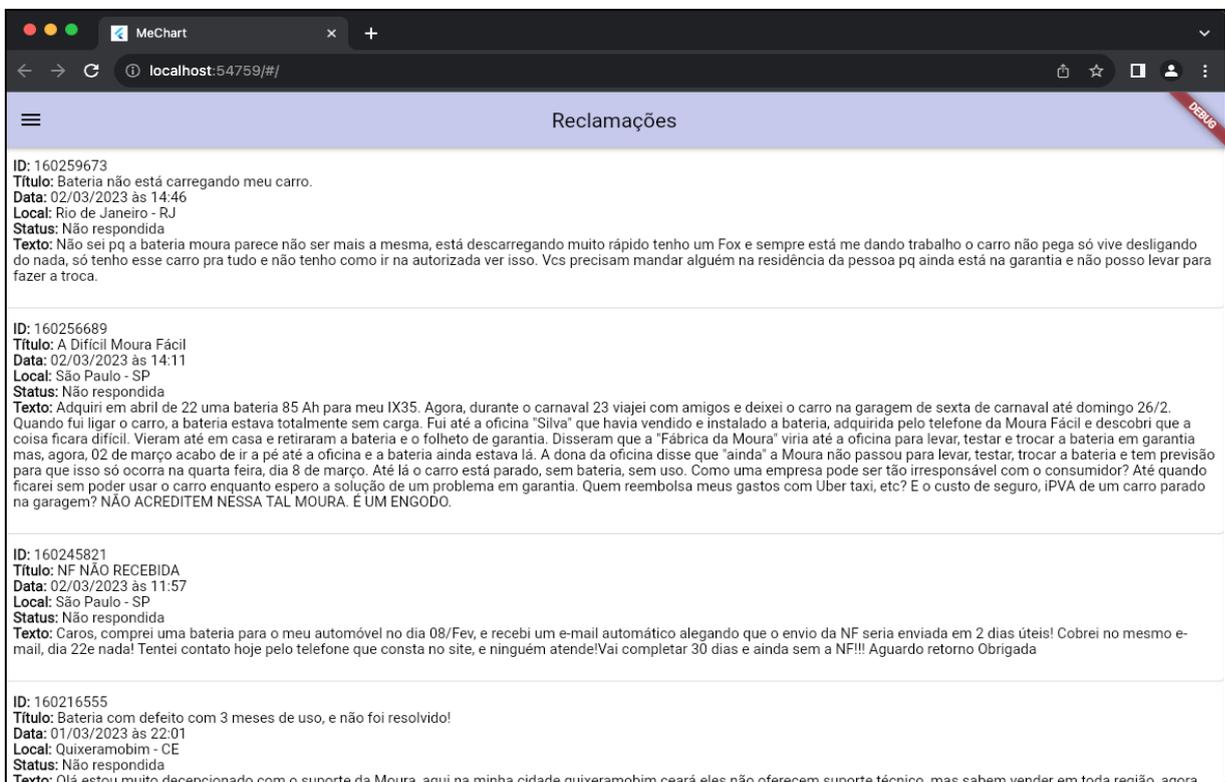
Demonstração prática do fluxo principal da aplicação no sistema Web utilizando o Google Chrome como navegador (Figura 22 à 27):

Figura 22 - Visualização prática da tela Inicial do aplicativo versão web.



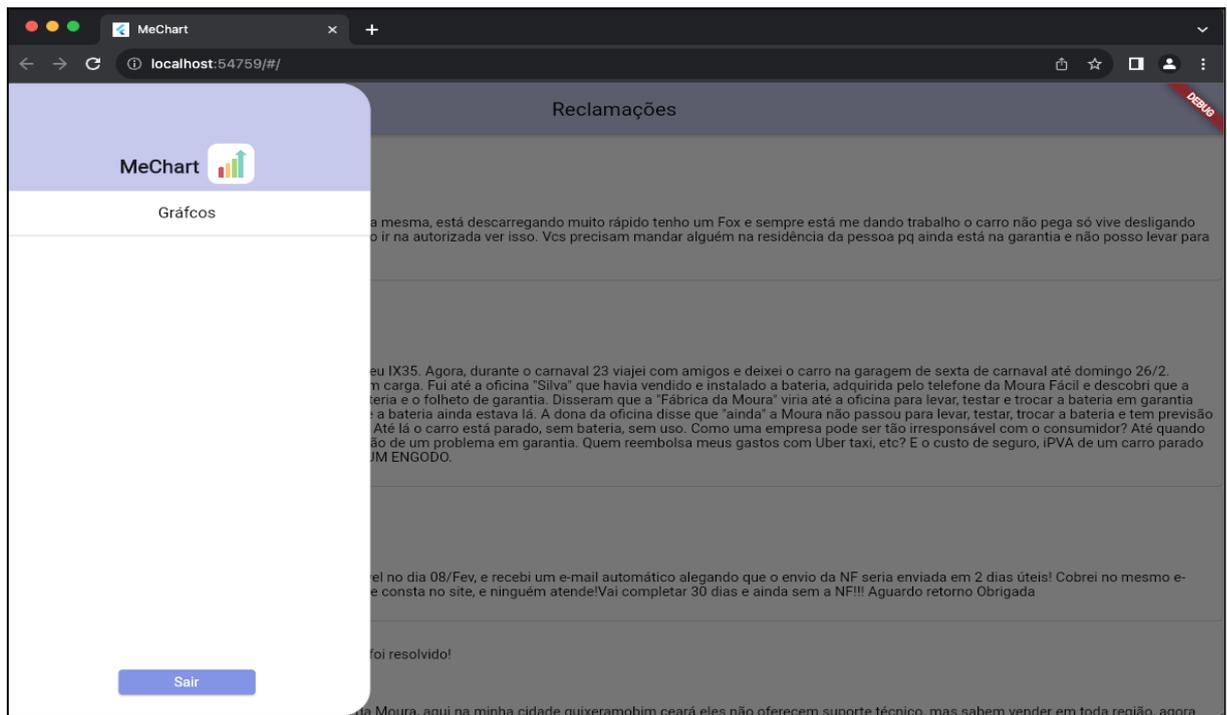
Fonte: O Autor (2023)

Figura 23 - Visualização prática da tela de reclamações do aplicativo versão web.



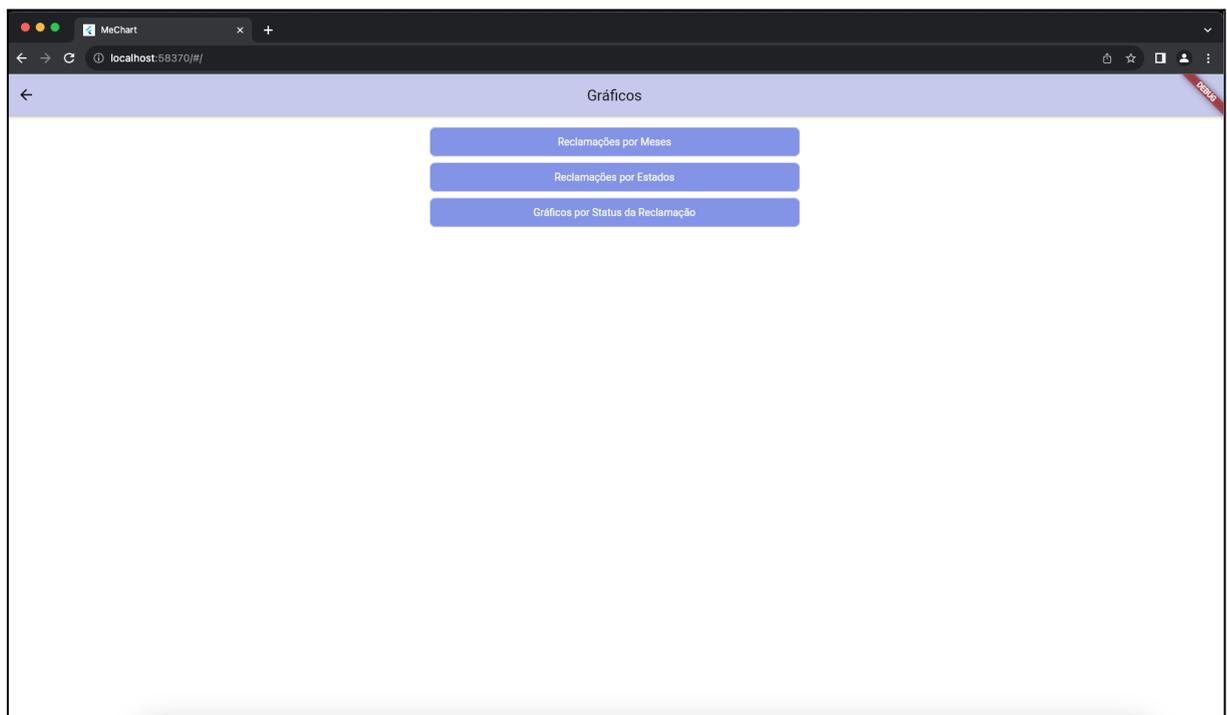
Fonte: O Autor (2023)

Figura 24 - Visualização prática da tela de reclamações com a aba lateral aberta do aplicativo versão web.



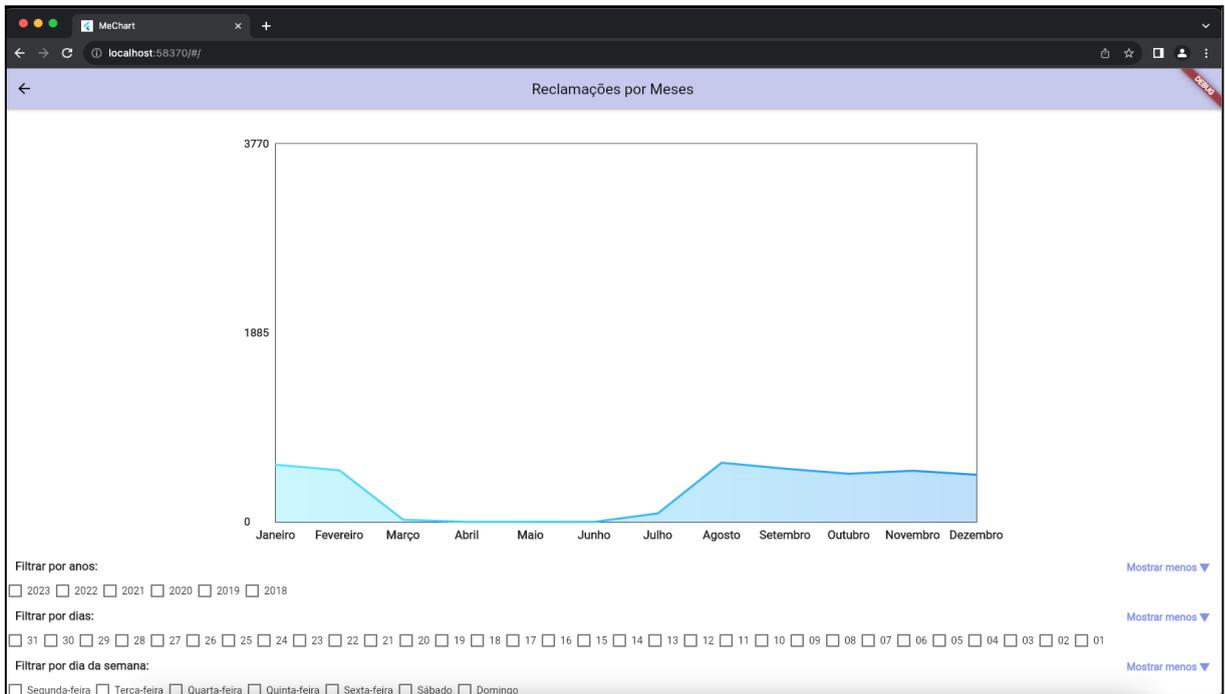
Fonte: O Autor (2023)

Figura 25 - Visualização prática da tela de listagem dos decision boards do aplicativo versão web.



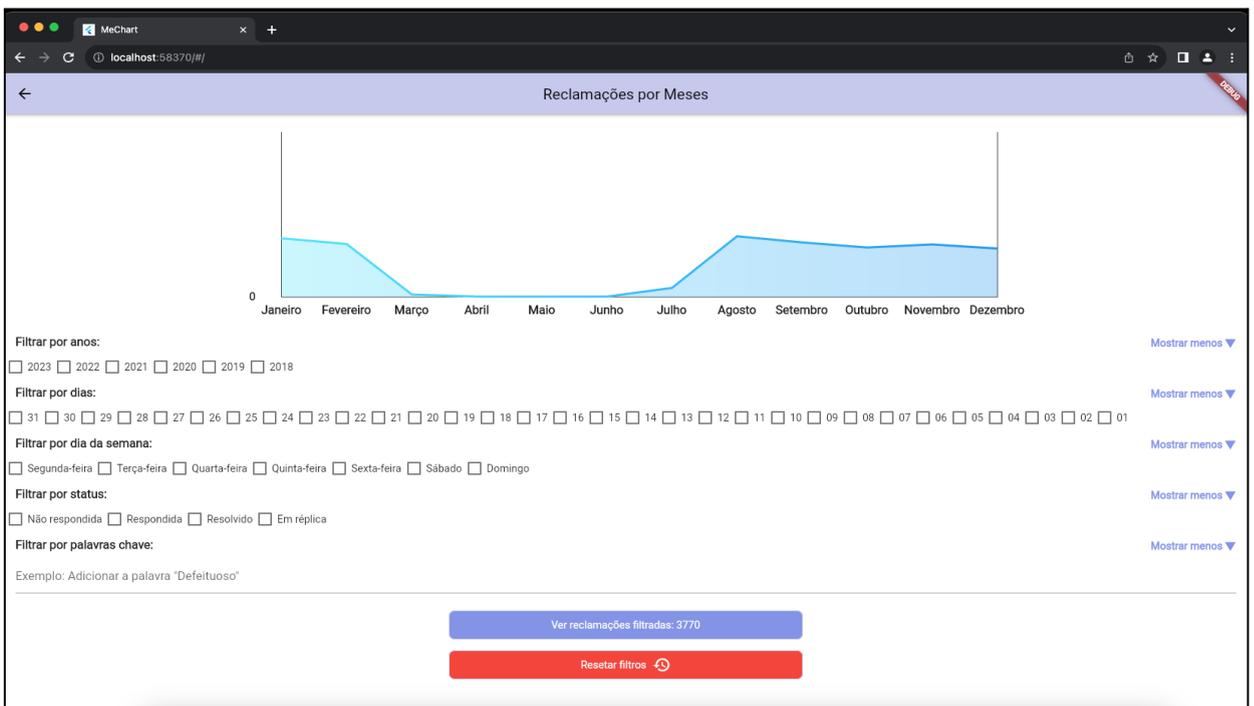
Fonte: O Autor (2023)

Figura 26 - Visualização prática da tela de exibição dos decision boards do aplicativo versão web.



Fonte: O Autor (2023)

Figura 27 - Visualização prática do filtro especializado dos decision boards na versão web.



Fonte: O Autor (2023)

4.1.2 ANDROID

Demonstração prática do fluxo principal da aplicação no sistema Android utilizando o emulador do Pixel 4 do Android Studio³⁹ (Figura 28 à 33):

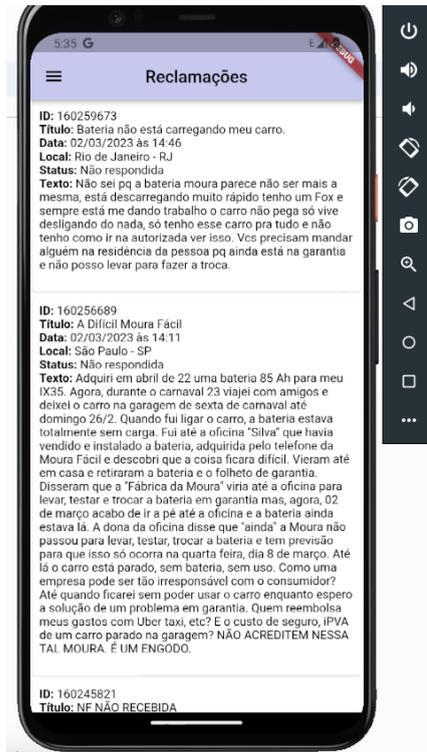
Figura 28 - Visualização prática da tela Inicial do aplicativo versão android.



Fonte: O Autor (2023)

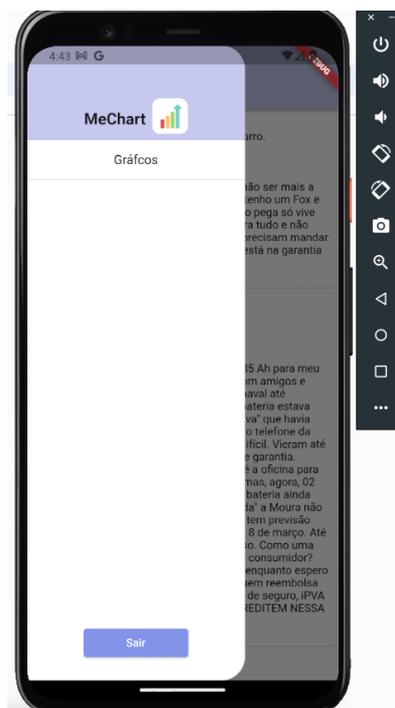
³⁹ Endereço eletrônico para Android Studio: <https://developer.android.com/studio>

Figura 29 - Visualização prática da tela de reclamações do aplicativo versão android.



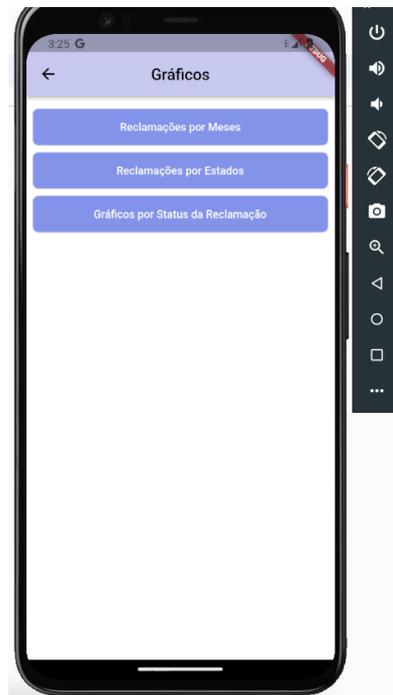
Fonte: O Autor (2023)

Figura 30 - Visualização prática da tela de reclamações com a aba lateral aberta do aplicativo versão android.



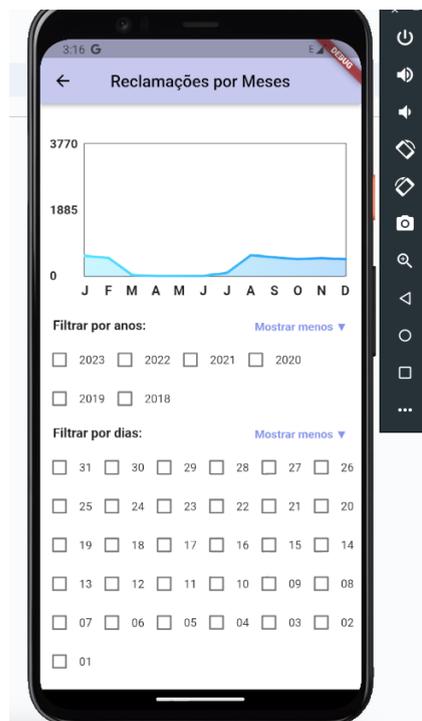
Fonte: O Autor (2023)

Figura 31 - Visualização prática da tela de listagem dos decision boards do aplicativo versão android.



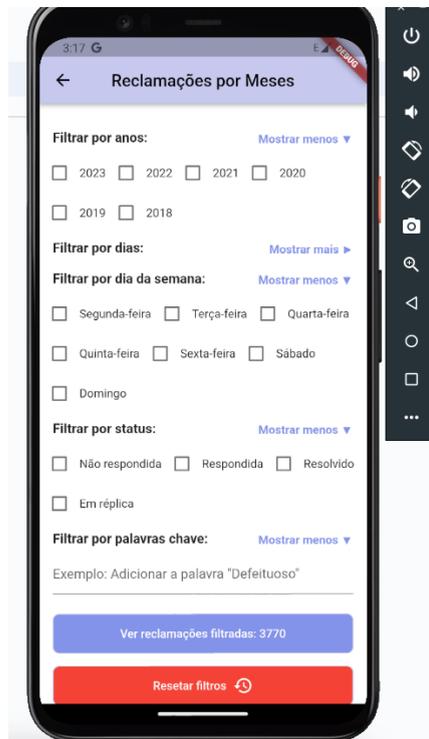
Fonte: O Autor (2023)

Figura 32 - Visualização prática da tela de exibição dos decision boards do aplicativo versão android.



Fonte: O Autor (2023)

Figura 33 - Visualização prática do filtro especializado dos decision boards na versão android.



Fonte: O Autor (2023)

4.1.3 IOS

Demonstração prática do fluxo principal da aplicação no sistema IOS utilizando o emulador do Iphone 14 Pro Max do Xcode⁴⁰ (Figura 34 à 39):

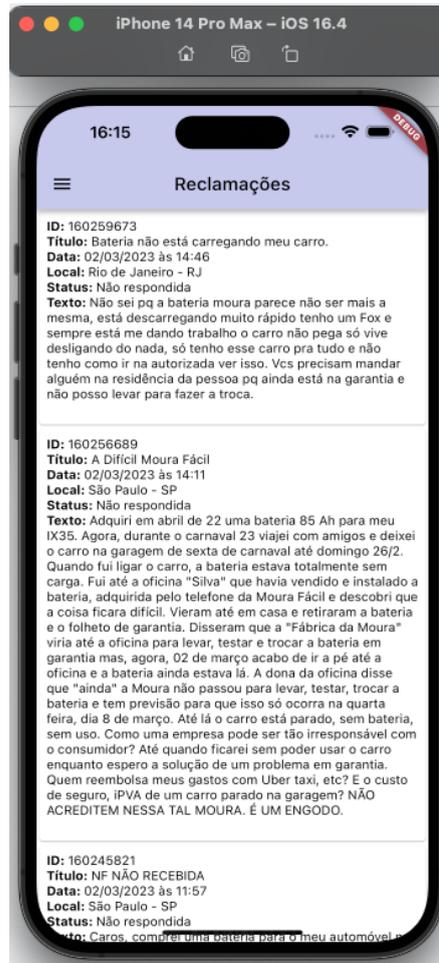
⁴⁰ Endereço eletrônico para Xcode: <https://developer.apple.com/xcode/>

Figura 34 - Visualização prática da tela Inicial do aplicativo versão ios.



Fonte: O Autor (2023)

Figura 35 - Visualização prática da tela de reclamações do aplicativo versão ios.



Fonte: O Autor (2023)

Figura 36 - Visualização prática da tela de reclamações com a aba lateral aberta do aplicativo versão ios.



Fonte: O Autor (2023)

Figura 37 - Visualização prática da tela de listagem dos decision boards do aplicativo versão ios.



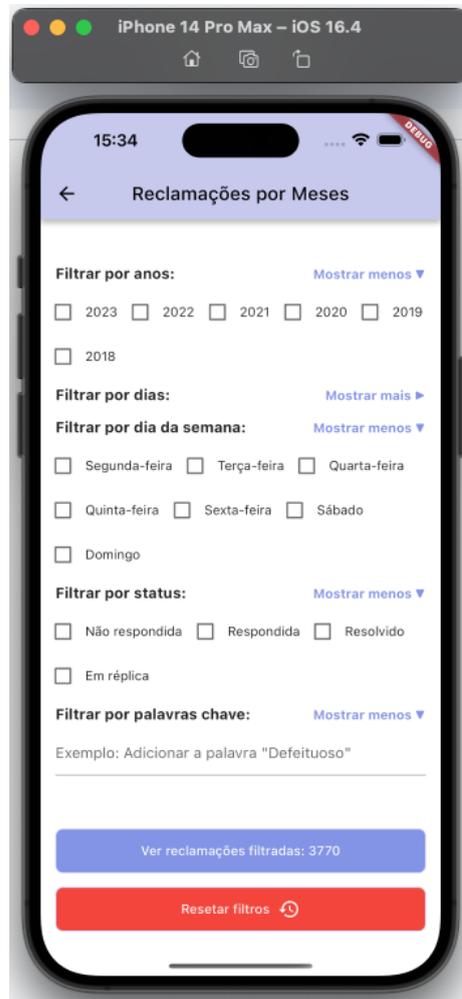
Fonte: O Autor (2023)

Figura 38 - Visualização prática da tela de exibição dos decision boards do aplicativo versão ios.



Fonte: O Autor (2023)

Figura 39 - Visualização prática do filtro especializado dos decision boards na versão iOS.



Fonte: O Autor (2023)

4.2 DECISION BOARDS

Foram construídos no total três Decision Boards base para suporte à tomada de decisão organizacional e eles se baseiam em três principais grandezas, sendo elas Tempo, Local e Status:

- Tempo: Na grandeza tempo foram utilizados como critério os anos e trimestres como base para construção dos gráficos;
- Local: Na grandeza local foi utilizado os estados do Brasil como critério para construção dos gráficos;

- Status: Na grandeza status foi utilizado o estado da reclamação, ou seja, se ela foi respondida, resolvida, não resolvida ou em réplica para a construção dos gráficos.

Juntamente com os decision boards, foi desenvolvido também um filtro especializado de interação em tempo real com o gráfico, oferecendo ao tomador de decisão a possibilidade de cruzar as informações de **anos, meses, dias, status da reclamação, dias da semana e palavras chaves** em cada decision board. Dessa forma, as possibilidades e insights que são gerados no momento da interação com os decision boards são diversas, ampliando ainda mais o suporte à decisão.

Após a aplicação do filtro também é possível navegar para as reclamações que foram selecionadas, tornando a análise mais específica e declarativa, acentuando os problemas que antes seriam extremamente difíceis - inviáveis por não possuírem um filtro específico como o desenvolvido nesse trabalho - de serem percebidas utilizando apenas a ferramenta do Reclame Aqui.

Juntando esses três decision boards juntamente com o filtro é possível que a organização que venha fazer uso desse software consiga ter uma visão crítica e holística sobre diferentes aspectos das reclamações, podendo tomar decisões a nível operacional, tático ou estratégico a depender do insight que seja gerado a partir da leitura dos gráficos.

4.2.1 CONSTRUÇÃO

Nesta seção será apresentado a ideia por trás da construção de cada gráfico, a motivação de cada cruzamento de informações e, por fim, a demonstração gráfica na plataforma web de cada decision board desenvolvida.

4.2.1.1 RECLAMAÇÕES POR MESES

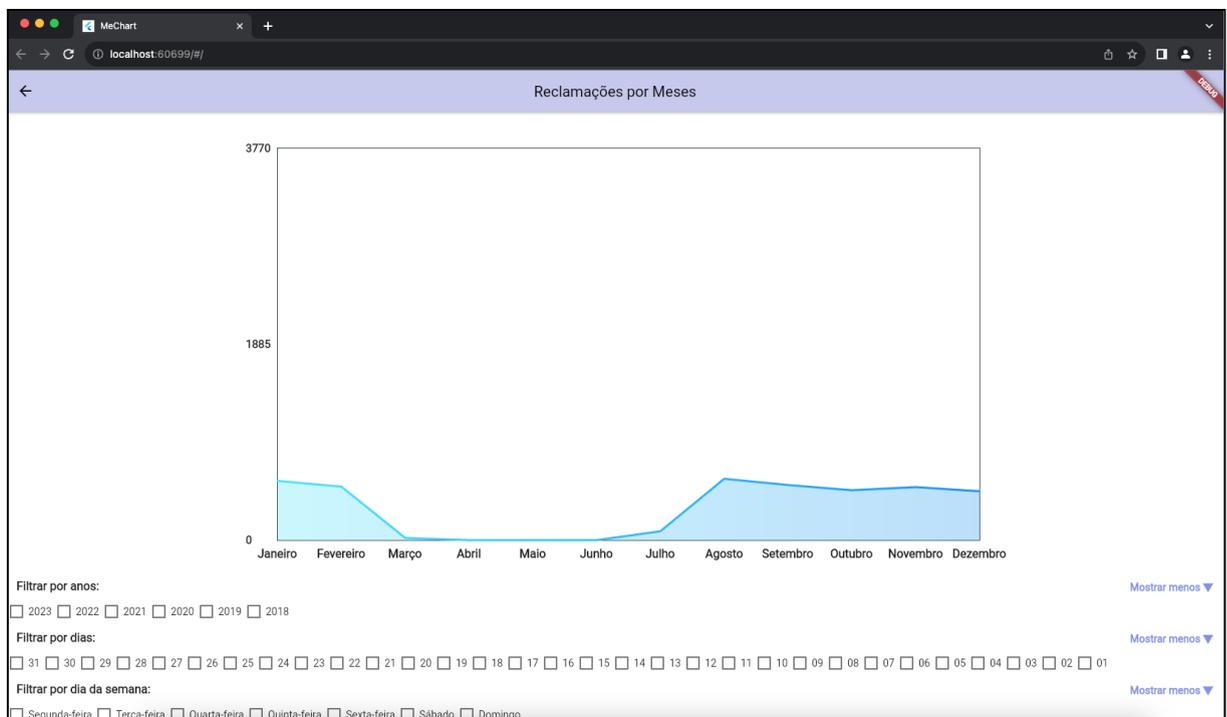
No gráfico de “Reclamações por meses” foram cruzadas as informações de **número de reclamações totais** com os **meses** em que elas aconteceram, trazendo dessa forma uma correlação de quais momentos do ano a maior parte das reclamações acontecem.

Com base nesse gráfico é possível notar em quais fases do ano o maior e menor número de reclamações acontecem, trazendo insights valiosos como em

quais períodos o time de CS⁴¹ deve estar mais atento ou não às reclamações que irão chegar no Reclame Aqui.

Além disso, com a aplicação do filtro o tomador de decisão será capaz de cruzar as informações de anos, dias, status da reclamação, dias da semana e palavras chaves, tornando essa leitura trimestral um decision board capaz de demonstrar em quais momentos do ano as reclamações acontecem com mais frequência de acordo com os filtros que foram selecionados (Figura 40).

Figura 40 - Representação do decision board Reclamações por meses na versão Web.



Fonte: O Autor (2023)

4.2.1.2 RECLAMAÇÕES POR ESTADOS

No gráfico “Reclamações por Estados” foram cruzadas as informações de **número de reclamações totais** com a de **localidade** com o objetivo de se gerar insights sobre quais estados as reclamações mais acontecem em todo o Brasil.

Com base nesse decision board, a organização pode tomar decisões sobre em quais estados ela deve focar mais no suporte ao cliente tanto à nível digital

⁴¹ CS - Customer Success: Time responsável por lidar com o sucesso do cliente, suprindo todas as suas necessidades possíveis.

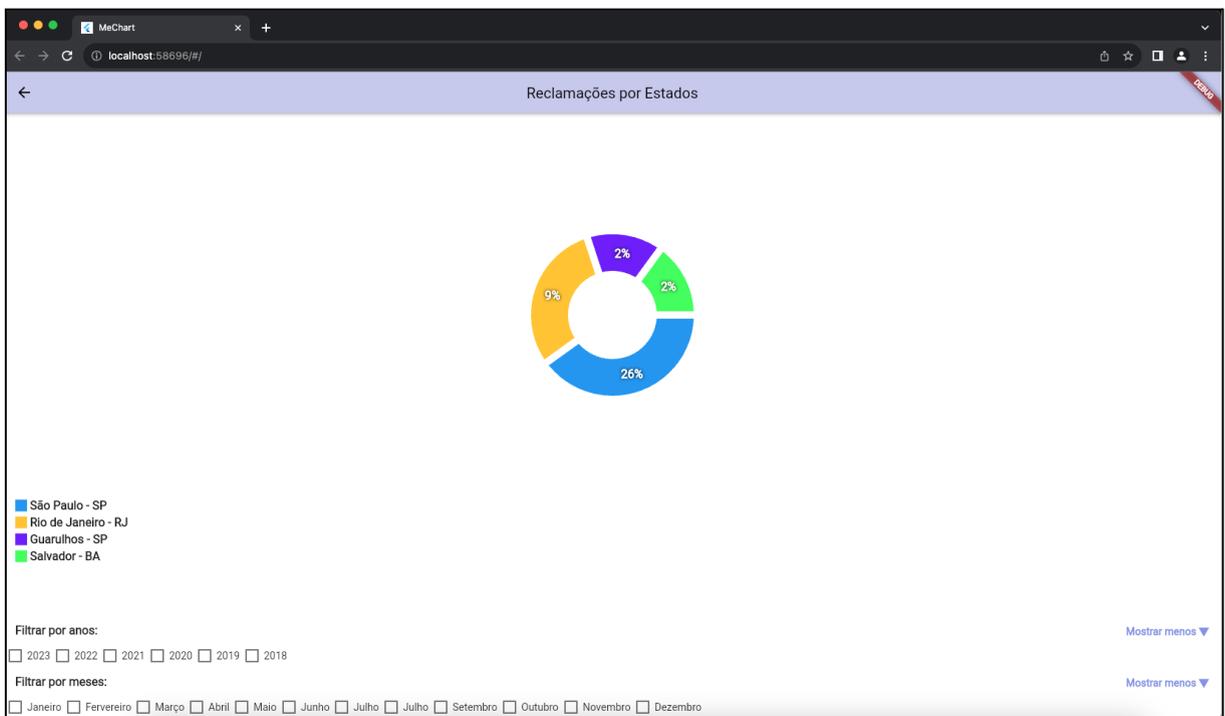
quanto físico, construindo novos estabelecimentos ao longo desses locais para que os clientes consigam, com mais facilidade, resolver os seus problemas.

Além disso, existem problemas que podem estar relacionados a locais específicos, como lotes defeituosos ou com validade expirada. Dessa forma, é possível analisar com mais eficiência e com uma outra visão os problemas relacionados a um estado que possui muitas reclamações.

Nesse decision board o objetivo foi listar apenas os quatro estados com o maior número de reclamações, pois dessa forma o espaço amostral fica reduzido apenas aos locais com maior criticidade em número de reclamações, ajudando a tomada de decisão ser mais direta.

Com a aplicação do filtro que varia entre os anos, meses, dias, dias da semana e palavras chaves o tomador de decisão é capaz de notar quais cidades possuem os maiores e menores índices de reclamações, sendo essas quatro cidades variáveis que podem mudar em tempo real de acordo com a seleção dos filtros (Figura 41).

Figura 41 - Representação do decision board Reclamações por Estados na versão Web.



Fonte: O Autor (2023)

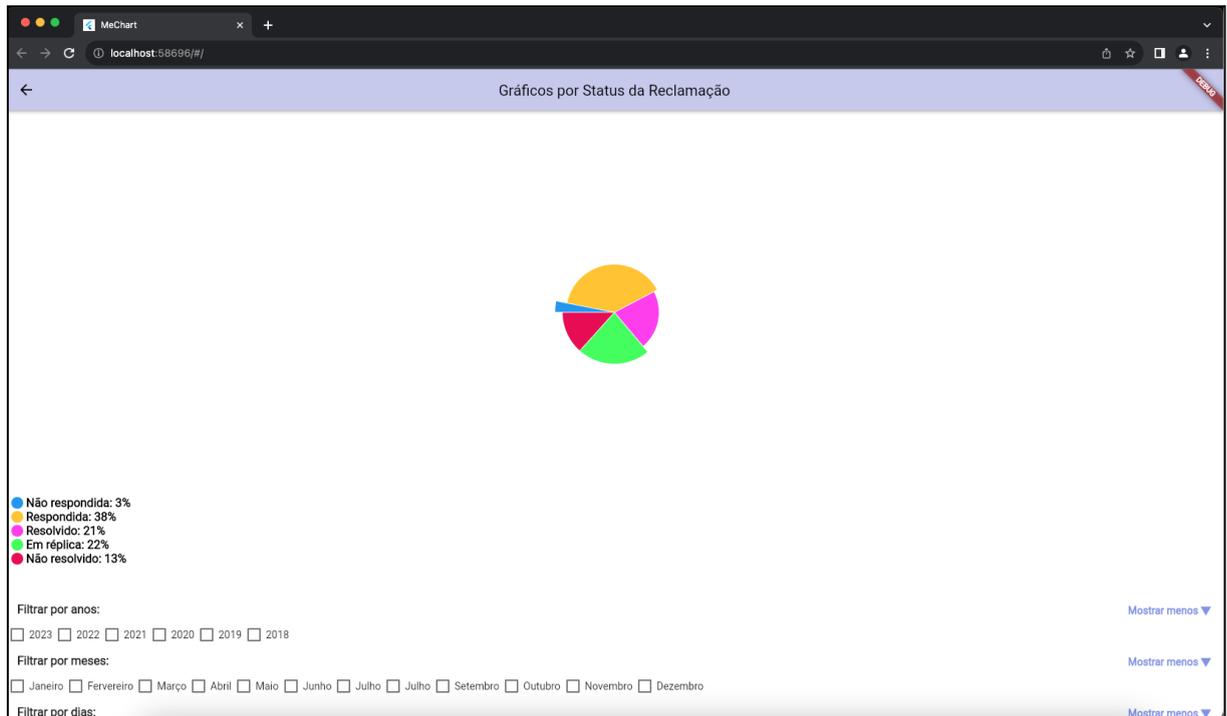
4.2.1.3 GRÁFICO POR STATUS DA RECLAMAÇÃO

No gráfico “Gráfico por Status da Reclamação” são cruzadas as informações de **status da reclamação** com o **número total de reclamações** com o objetivo central de analisar, quantificar e qualificar o status geral de todas as reclamações presentes na base de dados.

Com base nesse decision board, a organização será capaz de identificar e medir o nível de eficácia do time de Customer Success na plataforma do Reclame Aqui. Além disso, também poderá tomar decisões estratégicas sobre o plano de ação em resolver esses problemas a depender da vacância na porcentagem de reclamações respondidas ou resolvidas.

Juntamente com o filtro que possui os anos, meses, dias, dias da semana e palavras chaves disponíveis para serem selecionados o tomador de decisão pode observar, em tempo real, o status das reclamações nos filtros que foram selecionados. Portanto, será possível entender que em certos anos ou meses a taxa pode ter aumentado ou reduzido por fatores que valem a pena ser analisados para garantir uma assertividade maior na tratativa das reclamações do Reclame Aqui (Figura 42).

Figura 42 - Representação do decision board Status por Reclamação na versão Web.



Fonte: O Autor (2023)

4.3 DIFICULDADES ENCONTRADAS

As dificuldades encontradas ao longo deste TCC podem ser divididas em **logísticas** e **técnicas**, e todas elas se caracterizam como barreiras que fizeram desse processo de conclusão de curso um processo de aprendizado seja ele positiva ou negativamente.

A dificuldade **logística** mais evidente foi o tempo, já que o período letivo foi reduzido devido aos efeitos da pandemia, fazendo com que o trabalho tivesse que ser feito com menos dois meses totais. Porém, como uma forma de reduzir esses danos, esse trabalho foi iniciado um mês antes do início do período, fazendo com que fosse possível desenvolver com mais afinco os detalhes do projeto.

Além disso, uma outra dificuldade encontrada ao longo desenvolvimento deste trabalho foi o de gerenciar o foco entre desenvolvimento do software e da monografia, pois eram dois artefatos fundamentais para a entrega completa do projeto.

Já à nível **técnico**, as maiores dificuldades foram lidar com os três ambientes de desenvolvimento simultaneamente, sendo eles o IOS, Android e Web, já que mesmo o flutter sendo um cross-platform framework e lida perfeitamente com o build para esses cenários, o desenvolvedor sempre precisará cuidar de alguns detalhes de usabilidade e setup para que o software ofereça uma experiência de uso similar em todos os ambientes.

Além dos ambientes, o cruzamento das informações juntamente com a implementação dos gráficos e do filtro foi um grande desafio, já que lidar com esse tipo de lógica em tempo reduzido não é algo trivial de ser alcançado.

4.4 LIÇÕES APRENDIDAS

A maior lição aprendida que pude notar ao longo de todo o desenvolvimento do projeto foi a importância da **disciplina**, sem ela seria impossível conciliar o desenvolvimento de todo o TCC com o meu trabalho de desenvolvedor de software somado com mais outras duas disciplinas que cursei juntamente com esta disciplina.

Além disso, notei o quão importante é possuir bons guias no caminho pelo refinamento do conhecimento, sem as ideias chaves do meu professor orientador Vinicius Garcia junto com as especificações técnicas e científicas da minha professora co orientadora Maria da Conceição o caminho seria sem sombra de dúvidas mais árduo.

À nível acadêmico pude ter uma experiência real do que é redigir uma monografia científica, apesar de ter tido várias experiências ao longo da graduação como um todo, nenhuma foi tão marcante como o trabalho de conclusão do curso, pois precisei estudar detalhes aprofundados de escrita, explicação e referenciação que antes não tinham sido requisitados na mesma intensidade.

À nível técnico sem dúvidas tive um crescimento considerável, pois precisei trabalhar mais a fundo sobre scripts e integrações com base de dados que não tinha lidado ainda no ambiente Flutter. Além disso, precisei olhar de forma mais madura para níveis de arquitetura de software, pois para tornar o desenvolvimento desse software possível de ser realizado dentro do tempo

limite, foram necessárias tomadas de decisões cruciais quanto à organização e codificação do projeto.

4.5 SUMÁRIO DA SEÇÃO

O objetivo desta seção foi demonstrar de forma prática e aplicada os resultados alcançados ao longo do desenvolvimento do projeto. Seguindo a ordem de como foi dividida esta seção, inicialmente é explorado o fluxo principal da aplicação nos três ambientes desenvolvidos web, ios e android. Após isso, são detalhados e explicados os decision boards juntamente com os seus respectivos filtros, dando exemplos de possíveis insights que poderiam ser gerados com as suas utilizações. Após isso, são explicadas as dificuldades encontradas ao longo do desenvolvimento do projeto, que serviram de aprendizados sólidos desenvolvidos nas lições aprendidas, que em suma significam crescimento profissional, acadêmico e pessoal para o autor deste trabalho.

Com os resultados obtidos esclarecidos para o leitor, na próxima seção é explorada as conclusões obtidas e os trabalhos futuros propostos para ampliar ainda mais as capacidades do projeto.

5 CONCLUSÃO

Este trabalho demonstrou todo o processo que foi utilizado para desenvolver a aplicação responsável por resolver os problemas de leitura dinâmica dos dados oriundos do Reclame Aqui.

Através do sistema desenvolvido, qualquer organização que possua seus dados disponíveis no Reclame Aqui será capaz de estudar, compreender e extrair informações que antes eram impossíveis utilizando apenas as ferramentas disponíveis no site.

Além disso, serve como uma solução gratuita e de fácil acesso para os tomadores de decisão, podendo ser utilizada juntamente com os trabalhos relacionados a fim de gerar insights valiosos nos momentos de tomada de decisão organizacional.

No entanto, é válido ressaltar que a plataforma possui uma limitação quanto ao tipo da base de dados aceita, precisando ser no formato .csv e ter sido gerada a

partir do script de web scraping desenvolvido pela professora co orientadora deste trabalho. Isso ocorre pelo fato de que este projeto é uma extensão gráfica/visual do script desenvolvido anteriormente ao seu início.

Juntamente com isso, não foram feitos testes de usabilidade da plataforma, portanto é possível que existam filtros especiais ou decision boards a serem adicionados ou removidos a depender dos feedbacks que o sistema venha a receber após estas testagens.

Como conclusão, o projeto desenvolvido neste trabalho contribui para resolver pontualmente o problema que se propôs inicialmente, apoiando diretamente a organização no seu processo de tomada de decisão a partir da leitura, interação e interpretação dos decision boards desenvolvidos.

6 TRABALHOS FUTUROS

Os possíveis trabalhos futuros são, primordialmente, realizar testagem em campo com organizações que possuem páginas na plataforma do Reclame Aqui, como a Baterias Moura, por exemplo. Ampliar a integração do script de web scraping com o sistema desenvolvido, tornando todo o processo focado apenas no aplicativo e fazendo com que o usuário interaja apenas com a plataforma, não precisando executar o script separadamente para gerar a base de dados. Criar novos decision boards e adicionar novos filtros para tornar a possibilidade de geração de insights ainda mais promissora. Integrar os decision boards com o chat gpt, criando interações com a inteligência artificial a partir do resumo das reclamações, tornando possível o mapeamento do humor dos clientes. Por fim, realizar o lançamento da aplicação nas lojas de aplicativos, pois dessa forma é possível amadurecer a ideia do projeto e colocá-lo como um potencial competidor das plataformas atuantes nesse mercado de BI.

REFERÊNCIAS

ANDERSEN, J.A. (2019), "Owners vs executives and decisions vs control", *Corporate Governance*, Vol. 19 No. 3, pp. 458-470. Disponível em: <<https://doi.org/10.1108/CG-04-2018-0158>>. Acesso em: 19 jun. 2023.

ARQUITETURA DE SOFTWARE. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2020. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Arquitetura_de_software&oldid=60084267>. Acesso em: 24 dez. 2020.

BATERIAS MOURA. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Baterias_Moura&oldid=65881090>. Acesso em: 16 mai. 2023.

CAPLER, J. (2020), "A descriptive case study of effective decision-making of a local government organization in the United States of America", *International Journal of Public Leadership*, Vol. 16 No. 3, pp. 279-297. Disponível em: <<https://doi.org/10.1108/IJPL-12-2019-0090>>. Acesso em: 28 jun. 2023.

DABAB, M.; FREILING, M.; SAGALOWICZ, D. "Expanding Horizons in Business Decision Making," 2021 IEEE Technology & Engineering Management Conference - Europe (TEMSCON-EUR), Dubrovnik, Croatia, 2021, pp. 1-8, doi: 10.1109/TEMSCON-EUR52034.2021.9488619. Disponível em: <<https://ieeexplore.ieee.org/document/9488619>>. Acesso em: 19 jun. 2023.

DARÍO Rodríguez, CAROLINA Busco, RODRIGO Flores, Information technology within society's evolution, *Technology in Society*, Volume 40, 2015, Pages 64-72, ISSN 0160-791X, Disponível em: <<https://doi.org/10.1016/j.techsoc.2014.08.006>>. Acesso em: 28 jun. 2023.

DART DOCUMENTATION, Dart. Documentação Dart. Disponível em: <<https://dart.dev/language>>. Acesso em: 25 jul. 2023.

DART LINGUAGEM DE PROGRAMAÇÃO. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <[https://pt.wikipedia.org/w/index.php?title=Dart_\(linguagem_de_programa%C3%A7%C3%A3o\)&oldid=65358339](https://pt.wikipedia.org/w/index.php?title=Dart_(linguagem_de_programa%C3%A7%C3%A3o)&oldid=65358339)>. Acesso em: 23 fev. 2023.

DATABOX, In: Databox Documentation. Disponível em: <<https://databox.com/>>. Acesso em: 17 set. 2023.

DESENVOLVIMENTO ÁGIL DE SOFTWARE. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Desenvolvimento_%C3%A1gil_de_software&oldid=65245537>. Acesso em: 5 fev. 2023.

DESIGN THINKING, In: SEBRAE, Entenda o conceito de design thinking e como aplicá-lo aos negócios, 2022. Disponível em: <<https://sebrae.com.br/sites/PortalSebrae/artigos/design-thinking-inovacao-pela-criacao-de-valor-para-o-cliente,c06e9889ce11a410VgnVCM1000003b74010aRCRD>>. Acesso em: 29 jun. 2023.

DUÓ, M. In: KINSTA, O Guia para Iniciantes em Design Responsivo da Web (Amostras de Código e Exemplos de Layout), 2022. Disponível em: <<https://kinsta.com/pt/blog/design-responsivo-web/>>. Acesso em 29 jun. 2023.

ESCÂNDALO DE DADOS FACEBOOK–CAMBRIDGE ANALYTICA. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Esc%C3%A2ndalo_de_dados_Facebook%E2%80%93Cambridge_Analytica&oldid=65987808>. Acesso em: 2 jun. 2023.

FIGMA, Figma. Documentação do Figma. Disponível em: <<https://www.figma.com/>>. Acesso em: 19 jun. 2023.

FLUTTER, Flutter. Documentação Flutter. Disponível em: <<https://flutter.dev/>>. Acesso em: 19 jun. 2023.

GIT, Git. Documentação Git. Disponível em: <<https://git-scm.com/>>. Acesso em: 27 jul. 2023.

GITHUB, Github. Documentação Github. Disponível em: <<https://github.com/>>. Acesso em: 27 jul. 2023.

GOOGLE DATA STUDIO, Google Data Studio. Disponível em: <<https://lookerstudio.google.com/>>. Acesso em: 17 set. 2023.

KERRIGAN R., Harvard Business School. STARTUP VS. CORPORATE: WHICH CULTURE IS THE RIGHT FIT FOR YOU?, 2018. Disponível em: <<https://online.hbs.edu/blog/post/startup-vs-corporate-culture>>. Acesso em: 28 jun. 2023.

MEIRA, S., Fundações para os futuros digitais. TDS.company, [49pp], 2021, Disponível em: <bit.ly/futurosdigitais>. Acesso em: 28 jun. 2023

PIMENTA, M. In: ROCKCONTENT, Layout mobile ou responsivo: entenda as diferenças que existem entre eles, 2021. Disponível em: <<https://rockcontent.com/br/blog/layout-mobile/>>. Acesso em: 19 jul. 2023.

POWER BI, In: Documentação Microsoft Power BI. Disponível em: <<https://powerbi.microsoft.com/pt-br/>>. Acesso em: 17 set. 2023.

PRÊMIO RA, Premium Ra. Premiação anual reclame aqui, 2022. Disponível em: <<https://www.reclameaqui.com.br/premio/resultado>>. Acesso em: 27 jun. 2023.

PRÊMIO RA, Premium Ra. Premiação anual reclame aqui, 2021. Disponível em: <<https://www.reclameaqui.com.br/premio/anosanteriores/2021>>. Acesso em: 27 jun. 2023.

PRÊMIO RA, Premium Ra. Premiação anual reclame aqui, 2020. Disponível em: <<https://www.reclameaqui.com.br/premio/anosanteriores/2020>>. Acesso em: 27 jun. 2023.

PRÊMIO RA, Premium Ra. Premiação anual reclame aqui, 2019. Disponível em: <<https://www.reclameaqui.com.br/premio/anosanteriores/2019>>. Acesso em: 27 jun. 2023.

RECLAME AQUI. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Reclame_Aqui&oldid=65895789>. Acesso em: 18 mai. 2023.

REQUISITO FUNCIONAL. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Requisito_funcional&oldid=66062733>. Acesso em: 15 jun. 2023.

REQUISITO NÃO FUNCIONAL. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2019. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Requisito_n%C3%A3o_funcional&oldid=56780994>. Acesso em: 21 nov. 2019.

ROBERT C. Martin. 2017. Clean Architecture: A Craftsman's Guide to Software Structure and Design (1st. ed.). Prentice Hall Press, USA.

ROBERT C. Martin. 2008. Clean Code: A Handbook of Agile Software Craftsmanship (1st. ed.). Prentice Hall PTR, USA.

SCRUM. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <<https://pt.wikipedia.org/w/index.php?title=Scrum&oldid=65386485>>. Acesso em: 27 fev. 2023.

SELLEDGE, Lead Generation Institute. HOW ARE LEADS TRACKED VIA FIRST, SECOND AND THIRD PARTY COOKIES?, 2019. Disponível em: <<https://leadgenerationinstitute.com/how-are-leads-tracked-via-first-second-and-third-party-cookies/>>. Acesso em: 28 jun. 2023.

TEJAS Shah, SV Patel, A Novel Approach for Specifying Functional and Non-functional Requirements Using RDS (Requirement Description Schema), Procedia Computer Science, Volume 79, 2016, Pages 852-860, ISSN 1877-0509, Disponível em: <<https://doi.org/10.1016/j.procs.2016.03.083>>. Acesso em: 27 jun. 2023.

TERTO , K. In: Medium: Capyba, A Viable Architecture choice for real world projects in Flutter, 2021. Disponível em: <<https://medium.com/capyba/a-viable-architecture-choice-for-real-world-projects-in-flutter-579c8f715a1f>>. Acesso em: 31 jul. 2023.

VOROBÉV A.; ELESIN I.; SHEVADRONOV, A."On-Board Intelligent Decision Support System Development for Aviation Complexes Using Machine-Oriented Methods," 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russian Federation, 2023, pp. 1-5, doi: 10.1109/REEPE57272.2023.10086920. Disponível em: <<https://ieeexplore.ieee.org/document/10086920>>. Acesso em: 12 set. 2023.

XP EDUCAÇÃO, In: Xp Educação, "6 ferramentas de BI mais utilizadas no mercado: como escolher a melhor?", 2022. Disponível em: <<https://blog.xpeducacao.com.br/ferramentas-de-bi/>>. Acesso em: 17 set. 2023.