



**UNIVERSIDADE DO SUL DE SANTA CATARINA**

**GUILHERME DE PELLEGRINI ELIAS**

**APLICATIVO MÓVEL PARA O SERVIÇO DE COMÉRCIO  
HORTIFRUTIGRANJEIRO DE SANTA CATARINA - CEASA**

Florianópolis

2020

**GUILHERME DE PELLEGRINI ELIAS**

**APLICATIVO MÓVEL PARA O SERVIÇO DE COMÉRCIO  
HORTIFRUTIGRANJEIRO DE SANTA CATARINA - CEASA**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Sistemas de Informação da  
Universidade do Sul de Santa Catarina como  
requisito parcial à obtenção do título de  
Bacharel em Sistemas de Informação.

Orientador: Prof. Richard H. de Souza, Dr.

Florianópolis

2020

**GUILHERME DE PELLEGRINI ELIAS**

**APLICATIVO MÓVEL PARA O SERVIÇO DE COMÉRCIO  
HORTIFRUTIGRANJEIRO DE SANTA CATARINA - CEASA**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Sistemas de Informação da  
Universidade do Sul de Santa Catarina como  
requisito parcial à obtenção do título de  
Bacharel em Sistemas de Informação.

Florianópolis, 10 de dezembro de 2020.

---

Professor e orientador Richard H. de Souza, Dr.  
Universidade do Sul de Santa Catarina

---

Prof<sup>ª</sup>. Vera Rejane Niedersberg Schuhmacher, Dr.  
Universidade do Sul de Santa Catarina

---

Prof. Alexandre Vitoreti de Oliveira, Ms.  
Universidade do Sul de Santa Catarina

## RESUMO

As tecnologias de informação se tornaram essenciais na vida das pessoas dado o avanço e a popularização dos dispositivos móveis. É cada vez mais frequente o uso desta tecnologia para a solução de problemas nas mais diversas áreas de atuação e conhecimento. O setor hortifrutigranjeiro é um exemplo que tem se beneficiado de forma direta com a aplicação de tecnologias móveis. Seguindo o contexto, este trabalho descreve as ferramentas e etapas envolvidas na criação do protótipo de um aplicativo móvel. O aplicativo será utilizado para coleta de dados disponibilizados no portal do CEASA/SC. Além da coleta, o software faz o armazenamento de produtos favoritos e possibilita ainda a utilização de filtros temporais para visualização conforme a necessidade do consumidor. Foi utilizado para construção do protótipo *React Native*, e a linguagem Java para programação da lógica do *software*, e *MariaDB Server* para construção e manipulação da base de dados. Como resultado, foi obtido um aplicativo para consulta de produtos por meio de nomenclatura ou por características de produção, tais como: embalagem, quantidade, e/ou formato fornecido.

Palavras-chave: Dispositivos Móveis. Coleta de Dados. Visualização de dados. Setor Hortifrutigranjeiro. Desenvolvimento de *software*.

## **ABSTRACT**

Giving the advance and the popularization of mobile devices, the information technologies has become intrinsic to daily live. The growing of information technology adoption has been expressive in the most diverse areas of expertise and knowledge, e.g, the fruit and vegetable sector which has benefited from the application of mobile technologies directly. Along this context, this work aims to describe the technologies and procedures involved in creating a mobile application, whose will be used to collect available data on the CEASE/SC portal. Beyond daily cotation, the application stores the favorite selections from the user, besides time filtering, allowing the user to check some specific period aligned the buyer necessity. It was obtained a mobile application as a result, capable to search products either by name or such characteristics as: packaging, quantity, and / or format provided.

Keywords: Mobile devices. Data collect. Data visualization. Hortifrutigranjeiro sector. Software development.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Página de busca de arquivos de preços diários CEASA-SC .....	12
Figura 2 – Método Espiral (BOEHM, 1988b).....	19
Figura 3 – Exemplo de visualização da ferramenta Trello .....	25
Figura 4 – Macro fluxo da aplicação .....	31
Figura 5 – Fluxos de obtenção de informação.....	32
Figura 6 – Fluxo de informação via pré-seleção .....	32
Figura 7 – Fluxo de informação via listagem geral de produtos .....	33
Figura 8 – Fluxo de aplicação dos filtros .....	33
Figura 9 – Esquema geral das tabelas do aplicativo.....	39
Figura 10 – Relação dos métodos de buscas da tela inicial do momento inicial da aplicação.....	40
Figura 11 – Classe Produto e seus métodos .....	40
Figura 12 – Métodos de manipulação do produto .....	41
Figura 13 – Exemplo de enumeração: Classificação.....	42
Figura 14 – Tela inicial.....	43
Figura 15 – Tela Produtos .....	44
Figura 16 – Tela Detalhes Produto .....	45
Figura 17 – Tela Detalhes Produto - sobre .....	46
Figura 18 – Tela variação .....	47
Figura 19 – Tela Gráfico .....	48
Figura 20 – Tela busca não encontrado .....	49
Figura 21 – Tela busca encontrada .....	50
Figura 22 – Tela filtro.....	51
Figura 23 – Tela lista filtro aplicado .....	52

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	Application Programming Interface
BPMN	Business Process Modeling Notation
CEASA	Centro de Abastecimento de Santa Catarina
CSV	Comma-separated values
DSDM	Desenvolvimento de Sistemas Dinâmicos
EBC	Empresa Brasil de Comunicação
FDD	Feature-Driven Development
JAVA	Linguagem de Programação criada no início da década de 90.
JVM	Java Virtual Machine
PDF	Portable Document Format
S/A	Sociedade Anônima
SAR	Single-aliquot regenerative-dose
SC	Seleção de Características
SCRUM	Desenvolvimento Ágil, Desenvolvimento de Software.
SQL	Structured Query Language
UI	Interface com usuário do acrônimo em inglês User Interface
XP	eXtreme Programming (Programação Extrema)

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>9</b>
1.1	JUSTIFICATIVA .....	11
1.2	PROBLEMA.....	12
1.3	OBJETIVO .....	12
1.4	OBJETIVOS ESPECIFICOS.....	12
1.5	ESTRUTURA DO PROJETO .....	13
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>14</b>
2.1	METODOLOGIAS DE DESENVOLVIMENTO .....	14
2.1.1	Objetivos específicos .....	14
2.1.2	Metodologias Ágeis.....	17
2.1.3	Método de trabalho .....	21
2.2	CICLOS DE TRABALHO .....	21
2.2.1	Método de trabalho .....	21
2.2.2	Revisão e Monitoramento.....	21
2.3	TECNOLOGIAS APLICADAS .....	23
2.3.1	Git eGithub .....	23
2.3.2	Java.....	23
2.3.3	React Native .....	23
2.3.4	MySQL/MariaDB.....	24
<b>3</b>	<b>METODOLOGIA.....</b>	<b>25</b>
3.1	CARACTERIZAÇÃO DA PESQUISA .....	25
3.2	ETAPAS METODOLÓGICAS .....	25
3.3	DELIMITAÇÕES .....	26
<b>4</b>	<b>MODELAGEM.....</b>	<b>27</b>
<b>5</b>	<b>DESENVOLVIMENTO.....</b>	<b>31</b>
5.1	DESCRIÇÃO UML DO BACKEND .....	37
5.2	PROTÓTIPO.....	39
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>49</b>
	<b>REFERÊNCIAS .....</b>	<b>51</b>
	<b>APÊNDICES .....</b>	<b>54</b>
	<b>APÊNDICE A – Formulário.....</b>	<b>55</b>
	<b>APÊNDICE B – Diagrama UML completo do <i>back-end</i> da aplicação .....</b>	<b>58</b>





# 1 INTRODUÇÃO

Para analisar as características da comercialização de produtos hortifrutigranjeiros assim como qualquer etapa desse fluxo, é necessário o desmembramento dos processos de fornecimento e consumo de frutas, hortaliças etc. Esse setor vem crescendo em importância na dieta do consumidor brasileiro e aumento de estabelecimentos como lanchonetes e restaurantes, principalmente por estar associado a uma possível melhoria da qualidade de vida, bem-estar e melhores condições de saúde (Euromonitor, 2019). Esta importância vem acompanhada de exigências organizacionais de parte dos operadores ao longo de toda a cadeia de abastecimento desses estabelecimentos.

Um dos meios de aquisição dos produtos são pelas centrais de abastecimento determinadas por (LEI ESTADUAL SC 9831- 1995 Art. 71.).

Art. 71. São sociedades de economia mista as seguintes entidades: ( . . . )

X - Centrais de Abastecimento do Estado de Santa Catarina S/A - CEASA-SC, tendo por objetivo: executar a política estadual de abastecimento de hortifrutigranjeiros e de outros produtos alimentícios; constituir, construir, instalar e administrar centrais de abastecimento e mercados. (Redação acrescida pela Lei nº 9904/1995).

A CEASA/SC, segundo a ABRACEN (2020), é uma das 23 centrais de abastecimento do país. Foi fundada em 1978 tem como proposta o abastecimento da população. Apresenta para tal uma infraestrutura para agricultores e para comerciantes do atacado de produtos hortifrutigranjeiros, entre outros produtos alimentícios e não alimentícios. A política de produção e abastecimento de hortifrutigranjeiros, implica em uma empresa de economia mista e está ligada à Secretaria de Estado da Agricultura e da Pesca – SAR (CEASA-SC, 2020).

O objetivo da CEASA/SC é ser o interlocutor entre produtor e consumidor, fazendo a comercialização em atacado e varejo de produtos hortifrutigranjeiros tais como: alimentos, insumos orgânicos, produtos ornamentais, floricultura e artesanais; disponibilizando informações aos agricultores e técnicos envolvidos sobre o setor hortifrutigranjeiro (CEASA-SC, 2020). A fatia mais relevante de clientes que apresentam a necessidade de acesso a valores do mercado atacadista de hortifrutigranjeiros operam na Ceasa/SC, unidade de São José.

A partir dessa busca surgiu a necessidade de identificar o perfil desses consumidores e as estratégias de comercialização que são mobilizadas pelos produtores e pelos demais agentes de mercado. Foram identificados diversos canais de comercialização de produtos hortifrutigranjeiros utilizados pelos agricultores brasileiros, o mesmo acontece no processo de comercialização em Santa Catarina e em especial na região da Grande Florianópolis. Dentre elas é possível citar as

operações de relações entre: o agricultor e o mercado; o agricultor e o intermediário; além da relação direta entre o produtor e o consumidor em feiras livres, na propriedade ou na entrega em domicílio.

Considerando um dos objetivos da CEASA/SC, que é informar sobre o mercado hortifrutigranjeiro, como preços, no tempo adequado, existe o problema da disponibilização limitada de informação. A apresentação desses dados depende do *download* de um arquivo no formato .pdf que contém a cotação diária dos produtos. Para que seja feita a consulta de preços diários é preciso seguir estes cinco passos:

- 1) Acessar o *site* da CEASA/SC;**
- 2) Acessar a Cotação de Preços;**
- 3) Selecionar o ano;**
- 4) Selecionar o mês;**
- 5) Selecionar o dia para *download*.**

Além disto, é necessário procurar na lista a cotação do produto de interesse. Figura 1 a seguir apresenta a interface do *site* da CEASA/SC que deve ser acessado para consulta da cotação de preços.

**Figura 1 – Página de busca de arquivos de preços diários CEASA-SC**



Fonte: <https://www.ceasa.sc.gov.br/index.php/cotacao-de-precos>

Uma maneira de solucionar esse problema é a criação de um aplicativo mobile, que possibilite acesso aos dados com menos passos e sem a necessidade de download de um arquivo individual para cada dia. Tendo funcionalidades como: filtrar o produto de interesse por nome, tipo, faixa de preço entre outros; e a listagem de preço de um único produto num período de dias. Observando o mercado de desenvolvimento e as tecnologias já estabelecidas, é visto que este desenvolvimento é possível e pode trazer melhorias na relação dos produtores e consumidores da CEASA/SC facilitando e tornando mais transparente o processo de informação de valores. Um

diferencial é a possibilidade de levantamento de estatísticas, visto que todas as informações necessárias estarão armazenadas em bancos de dados.

Um diferencial é a forma de visualização das informações, por exemplo, caso um usuário deseje verificar se o valor de algum produto teve grande variação durante a semana, na forma de busca que existente, é necessário o download e consulta individual em quantos documentos .pdf forem necessários. Com o aplicativo é possível a visualização destes dados unificados através dos filtros já mencionados.

Com o aplicativo a CEASA/SC pode se valer de um banco de informações mais consistente e íntegro, proporcionando maior qualidade na prestação deste serviço. Ainda fica na possibilidade de haver expansões conforme haja necessidade de demanda por novas funcionalidades. Para possibilitar o desenvolvimento do aplicativo serão utilizados conceitos de metodologias ágeis, e tecnologias mais recentes, porém estáveis, para que não haja uma degradação e desatualização tão acelerada da proposta.

Entretanto, o interesse neste projeto é propor um meio que facilite a visualização dos valores de comercialização dos produtos fornecidos pela central de abastecimento da Ceasa/SC.

## 1.1 JUSTIFICATIVA

Nos últimos anos, o uso de dispositivos móveis no Brasil vem aumentando consideravelmente (EBC, 2019). Uma parcela significativa da população atualmente possui um smartphone, e fazem uso desse dispositivo em diversos lugares. É perceptível, ao andar pelas ruas que sempre há alguém utilizando um dispositivo móvel.

Essa tecnologia acabou se tornando indispensável. Com o aumento do uso de dispositivos móveis, é notório a popularização do uso de aplicativos. As pessoas fazem uso de vários aplicativos com variadas finalidades, de acordo com o seu interesse e suas necessidades particulares. Da mesma forma acontece com a consulta do preço dos produtos hortifrutigranjeiros. Sendo oferecida pouca, ou quase nenhuma alternativa para facilitar esse processo.

Em resumo, a razão que motiva a realização deste trabalho é o interesse em usufruir de uma tendência tecnológica de sistemas de informação que possa gerir uma solução alternativa simples. Essa solução visa atender a demanda por praticidade na visualização da variação dos preços dos produtos. Além disso, aumentar a eficiência como esse processo é realizado, encurtando o caminho para tal e possibilitando eliminar o gasto com papéis e impressões, oferecendo uma maior autonomia ao usuário na visualização, além de oferecer uma ferramenta intuitiva aos que necessitam a consulta diária sinta prazer em usar no dia a dia.

## 1.2 PROBLEMA

Considerando um dos objetivos da CEASA/SC, que é informar sobre o mercado hortifrutigranjeiro, surgiu o problema da disponibilização limitada dos preços dos produtos vendidos pela mesma, sendo a apresentação desses dados dependente do *download* de um arquivo, para que isso seja feito é preciso acessar o *site* da CEASA/SC, onde são necessários cinco passos:

- 1. Entrar no *site* da CEASA/SC**
- 2. Entrar em cotação de preços;**
- 3. Selecionar o ano;**
- 4. Selecionar o mês;**
- 5. Selecionar o dia para efetuar o download;**

Além disto, é necessário procurar na lista a cotação do produto de interesse. A disponibilização de preços pela CEASA/SC é feita de forma manual e não prática, tornando o processo de consulta menos eficiente do que deveria.

## 1.3 OBJETIVO

Tem-se por objetivo substituir o método convencional de visualização dos dados por meio do uso da tecnologia, com o desenvolvimento de um protótipo de aplicativo mobile, que tornará a visualização dos valores mais rápida e precisa. Além disso, com a informatização deste processo será também possível adicionar outras funcionalidades como, por exemplo, pesquisa por produto específico, filtro de tipo, origem, etc.

Será também objetivo a disponibilidade de cotação por período configurado pelo usuário, tornando a visualização mais objetiva. Sendo assim, todo o processo de coleta, até a alimentação dos *softwares* com os dados coletados.

## 1.4 OBJETIVOS ESPECIFICOS

De forma conjunta com o objetivo, algumas metas podem ser citadas, tais como:

- **Permitir ao usuário buscar produtos hortifrutigranjeiros;**
- **Facilitar ao usuário visualizar cotação dos valores de produtos de sua preferência;**
- **Abordar os processos de desenvolvimento de *software*;**

- **Desenvolver um protótipo de um aplicativo para auxiliar na visualização dos valores dos produtos.**

## 1.5 ESTRUTURA DO PROJETO

No capítulo de Introdução<sup>1</sup> é feita uma simples apresentação do contexto que será trabalhado 1.1 que contém a identificação do problema 1.2 mostra a justificativa para a realização de tal projeto, com quais são os objetivos 1.3 e 1.4. Brevemente é apresentado 1.5 a estrutura do projeto. Finalizando o primeiro capítulo, são apresentadas as metodologias (1.6).

Na sequência, o capítulo de Referencial Teórico traz alguns conceitos, em (2.1) que são fundamentais para a compreensão acerca do desenvolvimento do aplicativo. Posteriormente em (2.2) é apresentada toda a estrutura do sistema, mostra quais foram os documentos produzidos para o projeto, como foi feita a construção utilizando as tecnologias de desenvolvimento de aplicativos móveis.

Posteriormente em (2.2) é apresentada toda a estrutura do sistema, mostra quais foram os documentos produzidos para o projeto, como foi feita a construção utilizando as tecnologias de desenvolvimento de aplicativos móveis.

No terceiro capítulo, é apresentada a metodologia. O capítulo 4 apresenta a modelagem do projeto seguido do capítulo 5, que relata o desenvolvimento do aplicativo explicando a metodologia utilizada, descrevendo a plataforma de desenvolvimento e esclarecendo onde e como funcionam as classes e pacotes.

Por fim, há o capítulo de Considerações finais (6) onde são feitas as conclusões a acerca do trabalho, ressaltando quais foram as dificuldades e propõe futuras melhorias para o produto desenvolvido.

## 2 REFERENCIAL TEÓRICO

Neste capítulo é abordada a revisão bibliográfica do trabalho, ou seja, o fundamento que foi tomado como base para a realização desse projeto. Ele está dividido em três seções principais: Metodologia de desenvolvimento (seção 2.1), Método de Trabalho (seção 2.2), Tecnologias Aplicadas (seção 2.3).

### 2.1 METODOLOGIAS DE DESENVOLVIMENTO

A presente pesquisa visa apresentar alternativas e reconhecer uma metodologia que melhor concilie o planejamento de requisitos com o escopo do projeto. Servirá como um guia para implantação de tecnologias e processos.

#### 2.1.1 Objetivos específicos

Tendo sido um personagem importante no desenvolvimento de metodologias de projeto de *software*, Boehm (1988a) elenca as principais funções de um processo de desenvolvimento de *software* que são: determinar a ordem (evolução) dos estágios; e estabelecer os critérios para transição entre os estágios. Almeida (2017) relaciona essas funções com duas premissas básicas de projeto de *software*: “O que devemos fazer?”, e “o que é necessário para completar o que estamos fazendo?”

Em (BOEHM, 1988b) são apresentadas duas evoluções desse método, *stagewise* e cascata (do inglês “*Waterfall*”), respectivamente, respectivamente. O primeiro implanta o conceito de que o desenvolvimento de *software* deve ser realizado em estágios sucessivos, tais como: planejamento operacional, especificações operacionais (requisitos), especificações de código, codificação, teste de parametrização, teste de montagem, prototipagem e avaliação do sistema.

O método Cascata, traz um refino do método anterior, e que veio a se tornar muito influente para sua década e as posteriores. Conforme sumarizado por as mudanças foram:(ALMEIDA, 2017)

- 1) Reconhecimento de *loops de feedback* entre estágios, como uma linha guia para confinar os *loops* de retorno apenas aos estágios sucessivos entre si, o que minimiza o retrabalho em retornar muitos estágios;**
- 2) Uma incorporação inicial da prototipação no ciclo de vida do *software*, em paralelo com a análise de requisitos e *design* (também no início).**

Almeida (2017) continua citando dois pontos principais que contribuíram para a disseminação com método Cascata de Boehm: simplicidade de explicar e lembrar da ordem das fases, enquanto outros métodos pré-existentes eram mais complexos, o incremental, por exemplo; e o processo passa impressão de estar ordenado, contável, e mensurável uma vez que possui

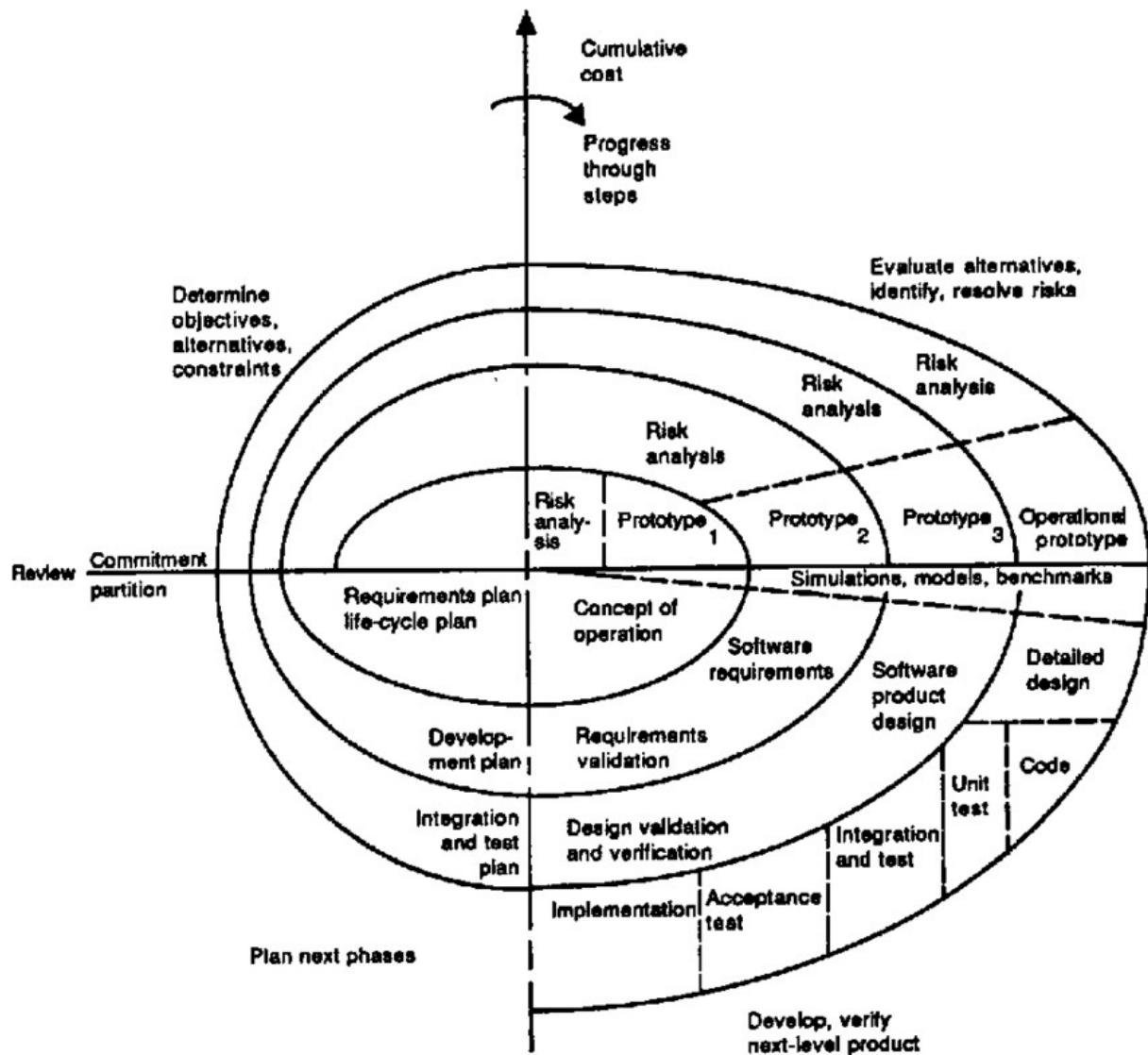
marcos alcançáveis (*milestones*) e documentados. Apesar do fluxo mais bem estruturado, a execução das fases do modelo cascata ocorre de maneira sequencial e sistemática. Estas características, aliadas ao foco na documentação do projeto, fazem com que o desenvolvimento de projetos demande um tempo maior para sua conclusão e a forma sequencial do projeto contribui para a perda de foco (PRESSMAN, 2011). Em um projeto, a interação entre as fases diminui o número de erros gerados pelas mesmas, visto que o cliente nem sempre consegue expressar de maneira satisfatória suas necessidades antes de visualizar a aplicação ou interagir com as primeiras versões do sistema (SOMMERVILLE, 2011) (PRESSMAN, 2011) que pode acabar gerando grandes quantidades de código não utilizado no final do projeto .

Ainda em Boehm (1988a) é apresentado o modelo espiral, que é baseado nas experiências e refinamentos feitos no modelo cascata, e traz como principal diferença a orientação a riscos para o processo de desenvolvimento do *software*, em contrapartida, a orientação a documentação ou codificação.

Conforme é ilustrado na Figura 2, o método espiral traz os elementos do método em cascata, porém, dispostos de forma a criarem ciclos que permitem um melhor acompanhamento dos estágios e compreensão dos custos envolvendo retrabalho, além dos envolvidos terem papéis bem definidos.



Figura 2 – Método Espiral



Fonte: (BOEHM, 1988b)

De forma resumida Almeida (2017) apresenta como características de metodologias clássicas:

- 1) O desenvolvimento guiado pelo ciclo de vida, como o modelo cascata e o modelo espiral, com tarefas e executores bem definidos;
- 2) O processo gera uma grande quantidade de documentação, que descreve processos e requisitos, além de tem um produto com código funcional;
- 3) A comunicação entre os participantes do projeto é formalizada, e o cliente tem papel importante na definição de especificações (apenas);

- 4) **Essas metodologias são apropriadas tanto para paradigmas orientados a objetos, quanto a mais simples como procedural ou estruturas monolíticas;**
- 5) **A abordagem é racionalizada, pautada em engenharia, apresentando planejamento extensivo nas fases iniciais e mediador durante o desenvolvimento, o que pode gerar problemas de controle durante o ciclo de vida de desenvolvimento.**

Desta forma, metodologias clássicas são defendidas por usuários que defendem um planejamento mais extensivo, e processos codificados e rigorosos, no intuito de tornar o desenvolvimento preditivo e eficiente. Embora as recentes dinâmicas postas por requisitos de sistema dinâmico, padrões de mercado e mudança no perfil dos times de desenvolvimento fez emergir um novo estilo de desenvolvimento: o estilo ágil (ALMEIDA, 2017).

### **2.1.2 Metodologias Ágeis**

Dado o final da década de 80, houve um crescimento significativo de desenvolvedores que simpatizam com a ideia de metodologias que focam na dinâmica com os novos problemas do mercado, isso é comprovado pela vasta criação de metodologias auto intituladas ágeis, como, *Scrum* (MAR; SCHWABER, 2002); *XP eXtreme Programming* (BECK, 1999); *Crystal* (COCKBURN, 2001); *FDD (Feature-Driven Development)* (COAD;PALMES,2002); *DSDM (Dynamic Software Development)* (STAPLETON, 1997); *Adaptative Software Development* (HIGHSMITH; HIGHSMITH, 2002); *Lean* (POPPENDIECK, M; POPPENDIECK, T., 2003)

De acordo com (SOARES, 2004), as metodologias ágeis têm sido preferidas frente a metodologias clássicas porque há projetos que possuem grande mutabilidade e cujos requisitos são passíveis a mudanças de forma inesperada, dois fatores principais que influenciam de forma negativa as metodologias clássicas que possuem um caráter mais engessado.

Por outro lado, fica entendido, ainda em (SOARES, 2004), que para projetos que venham a possuir um escopo bem determinado com requisitos mais firmes e mudanças previsíveis (projetos de expansão), as metodologias clássicas ainda são bem vistas dentro de ambientes empresariais, embora a tendência pela adoção completa a metodologias ágeis seja um futuro próximo.

Segundo Sommerville (2011) gerenciamento ágil de projeto é denominado como um conjunto de princípios de desenvolvimento, os quais buscam facilitar a criação de sistemas. Com o crescimento acelerado das metodologias alguns representantes e outros simpatizantes, que percebiam a necessidade de uma alternativa as metodologias clássicas e engessadas, reuniram-se para formalizar as bases do que seria considerado Desenvolvimento Ágil, entram, de concordância surgiu o “Manifesto Ágil” (AGILE ALLIANCE, 2001):

## “Manifesto para Desenvolvimento Ágil de software

Estamos descobrindo maneiras melhores de desenvolver softwares, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar: indivíduos e interações mais que processos e ferramentas; software em funcionamento mais que documentação abrangente; colaboração com o cliente mais que negociação de contratos; responder a mudanças mais que seguir um plano. Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.”

Destes pontos é possível concluir algumas bases ideológicas do desenvolvimento ágil que são:

- **Apoio nas interações entre indivíduos, facilitando o compartilhamento de informações e acelerando mudanças no processo (permite uma minimização da documentação);**
- **Colaboração com o cliente, o que coloca este no mesmo time que usuários e desenvolvedores, trazendo mais rapidamente a mudança de requisitos;**
- **Funcionamento do *software* dentro de um prazo minimizado permite medir quão eficaz a equipe é, além de permitir mensurar os tempos de respostas a problemas;**
- **Responder às mudanças que podem ocorrer com frequência dentro do projeto é de mais-valia que obedecer a um plano com alta mutabilidade.**

É apresentado por (ALMEIDA, 2017) que mesmo todas as metodologias ágeis compartilhando pontos em comum, não há padrão para a definição dos processos que são considerados ágeis, criando assim espaço para compará-las entre si. A seguir uma breve explicação das metodologias mais comuns:

- **Crystal: Cockburn, em seu livro intitulado *Agile Software Development* (COCKBURN, 2001), compara a família de processos *Crystal* com um cristal propriamente dito. A base para a comparação está no fato de que essa forma geométrica, sob a influência da luz, pode mudar de cor conforme a intensidade. A metodologia *Crystal* funciona de forma semelhante: projetos maiores, que exigem maior coordenação e comunicação, correspondem a cores mais escuras (branco, amarelo, laranja, vermelho e assim por diante). Podendo adequar-se à dificuldade e tamanho de um projeto, caso esses aspectos variem. Se um projeto aumenta de tamanho ou de nível crítico, mais processos *Crystal* podem ser adicionados a ele, e vice-versa. Tem base em sete características: entrega frequente, melhoria reflexiva, comunicação próxima, segurança pessoal, foco, fácil acesso a usuários especialistas, e definição de requisitos para o ambiente técnico;**

- **DSDM:** O DSDM (*Dynamic System Development Model* - Desenvolvimento de Sistemas Dinâmicos em português) (MEDEIROS, 2020). É um projeto compreendido em três fases: pré-projeto, ciclo de vida, e pós projeto. Tem como nove pilares que montam sua base: envolvimento do usuário, autonomia do time de projeto, entrega frequente, abordagem das necessidades atuais, desenvolvimento iterativo e incremental, permissão para mudanças reversíveis, escopo de alto-nível determinado a priori do projeto, testes durante o ciclo de vida e comunicação eficiente efetiva;
- **FDD:** Desenvolvido por Peter Coad e Jeff De Luca, foi inicialmente publicado em 1999 no livro “*Java Modeling in Color with UML*”, sendo que uma das principais obras, que possui uma versão completa da metodologia foi publicada em 2002. *Feature-driven development* (FDD) (AMBLER, 2005), ou Desenvolvimento Dirigido por Funcionalidades, é um método leve e iterativo que combina modelagem com desenvolvimento ágil com ênfase no modelo, divisão do trabalho em funcionalidades, e *design* iterativo de cada funcionalidade. Cada interação das funcionalidades consiste em duas fases: *design* e desenvolvimento.
- **Desenvolvimento *Lean* de Software:** A Toyota em meio ao crescimento e notando que não conseguiria entrar no mercado no mesmo ritmo que os concorrentes, criou o *Lean Thinking* (FADEL; SILVEIRA, 2010), que foca naquilo que é necessário, no local e no momento certo e eliminando desperdícios. Segundo o (LEAN INSTITUTE BRASIL, 2018), a metodologia *Lean* é uma estratégia de negócios que busca aumentar a satisfação do cliente através de um melhor aproveitamento dos recursos. A gestão *Lean* busca oferecer aos clientes um valor com o custo mais baixo dos seus produtos (propósito) através de melhorias contínuas dos seus fluxos de valor primário e de suporte (processos) através de pessoas com iniciativa, motivadas e qualificadas (pessoas). Adotada como Filosofia, que consiste em sete princípios: eliminar o desperdício, amplificar o aprendizado, decidir o quão tarde foi possível, entregar o quão rápido for possível, autonomia ao time, construir com integridade, e observar o todo;
- **SCRUM:** Foca no gerenciamento de projetos em situações em que é difícil de se planejar a frente, conta com mecanismos de controle empírico de processos, em que *loops de feedback* são um dos elementos principais (SUTHERLAND, 2017). Os times são autogeridos em incrementos começando pelo planejamento e terminando com uma revisão. Um dos personagens chaves nesse processo é o *scrum master* que tem

como principal objetivo resolver os problemas que geram impeditivos para o grupo de trabalho eficiente.

- **XP, XP2:** Esse método é focado nas melhores práticas de desenvolvimento, sendo doze delas: planejamento estratégico, pequenos lançamentos, metáforas, *design* simples, teste refatoração, programação pareada, propriedade coletiva, integração contínua, 40h por semana, clientes in loco, e padrões de codificação. O XP2 é uma revisão do XP que traz “práticas primárias” do XP<sup>1</sup> e “práticas corolárias” como: sentar-se juntos, visão de equipe, trabalho informativo e energizado, uso de estórias, ciclos (semanal, trimestral).

De forma resumida, metodologias ágeis constam de:

- Ciclos de desenvolvimento curtos e interativos orientados pelos recursos alocados, períodos de reflexão e tomada de decisão colaborativa;
- Presença de subprojetos dentro de um projeto, tendo eles tipicamente planejamento, desenvolvimento, integração, teste e entrega;
- Capacidade de lidar com imprevisibilidade uma vez que a confiança é nas pessoas (e sua criatividade), em detrimento ao processo;
- Favorecimento de um estilo de gerenciamento liderança-e-colaboração, tornando o gestor um facilitador ou coordenador;
- Cada ciclo de desenvolvimento é finalizado com código funcional que pode ser usado pelo cliente;
- Membros das equipes são rotativos com a finalidade de evitar monopólio de informação, visto que a documentação é focada apenas no código.

Cada uma dessas práticas contribui para o aumento da qualidade do software e ajuda a garantir que o produto final agregue valor e atenda as necessidades do negócio (PRIKLADNICKI; WILLI; MILANI, 2014).

- **Kanban:** O Método *Kanban* foi formulado por David J. Anderson e é baseado em práticas *Lean*, visando aperfeiçoar o processo de desenvolvimento de *software* pré-existente. Este método limita o trabalho em progresso, apresentando a evolução de forma visual, em quadros e cartões, tornando os problemas evidentes, proporcionando uma cultura de melhoria contínua (ANDERSON, 2010).

---

<sup>1</sup> XP, *eXtreme Programming*

### 2.1.3 Método de trabalho

Em conformidade com o que fora anteriormente visto, optou-se por experimentar uma variação do que se conhece como ‘métodos ágeis de desenvolvimento de software’. Diferentes opções dos atuais métodos utilizados pela comunidade foram experimentadas e adaptadas conforme a necessidade, no intuito de conhecer tais métodos e encontrar um (ou a mistura de mais de um deles) que suprisse as necessidades. Para a elaboração dos métodos de trabalho vistos neste projeto, foram experimentadas variações e adaptações dos métodos: *Scrum*, *XP2* e do *Kanban*.

## 2.2 CICLOS DE TRABALHO

Tendo por base o método *Scrum*, que consiste na divisão da tarefa em ciclos de atividades, mensurando o nível de dificuldade da tarefa e o tempo a ser administrado para alcançar sua conclusão, foi idealizado um plano de trabalho em que houvesse a definição do que precisaria ser feito e sua duração. De acordo com o tempo necessário e o disponível, assim como o conjunto de tarefas propostas, foram apresentadas as necessidades de quatro etapas de trabalho, conhecidas como *sprints* no modelo *Scrum*. Cada um desses ciclos retornaria um conjunto de informações a serem analisadas formando indicadores: do avanço em direção aos objetivos, o que estava fora de conformidade e onde eram necessários maiores atuação e foco.

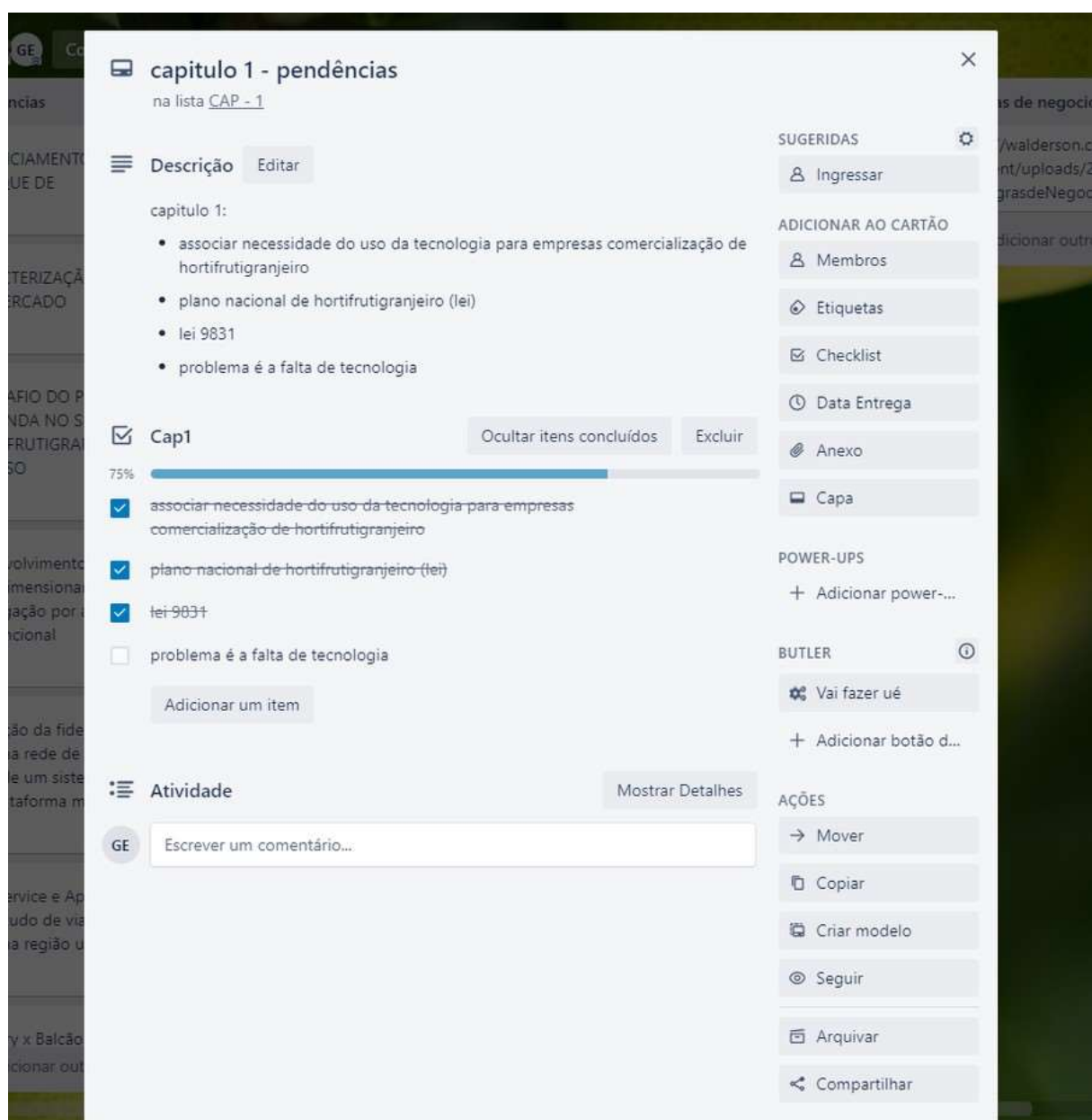
### 2.2.1 Método de trabalho

No objetivo de propiciar maior dinamismo e organização nas práticas dos ciclos citados anteriormente, optou-se por uma variante baseada no método de programação extrema (XP).

### 2.2.2 Revisão e Monitoramento

Para a revisão conjunta e o acompanhamento das tarefas foi utilizada uma alternativa que emula a ferramenta da qualidade denominada *Kanban*. A ferramenta *online Trello* disponibiliza um conjunto de cartões com diferentes opções de parametrização de cores e conteúdos, entre listas, imagens, *links* e demais anexos, proporcionando uma organização para o desenvolvimento de projetos solo ou em equipe. As tarefas inerentes ao trabalho descrito nesse artigo foram dispostas e acompanhadas na ferramenta *Trello*, simulando um *Kanban2*, onde cada tarefa e/ou etapa concluída tinha seu *status* e cor modificados. As dúvidas, problemas e dificuldades também eram descritas nessa plataforma, assim como sua solução também era comentada, formando assim também uma base de conhecimento para esse trabalho. A figura 3 traz um exemplo de cartão de controle usado nessa ferramenta.

Figura 3 – Exemplo de visualização da ferramenta *Trello* – um modelo de *Kanban*



Autor

## 2.3 TECNOLOGIAS APLICADAS

Nas próximas seções são abordados um conjunto de conhecimentos que possibilitam a criação de um aplicativo mobile, e algumas nuances da sua aplicação no contexto que está sendo trabalhado.

### 2.3.1 *Git e Github*

O *Git* (CHACON; STAUB, 2020) é um sistema de controle de versão para desenvolvimento de *softwares*. Ele tem como características principais a velocidade, integridade dos dados e suporte para fluxos de trabalho distribuídos e não-lineares. O *GitHub* (GITHUB, 2020), por sua vez, é um sistema *online* para hospedagem de repositórios *Git*. Esse sistema é muito popular entre desenvolvedores que pretendem disponibilizar o código fonte de seus projetos na *internet* para que outras pessoas possam colaborar.

### 2.3.2 **Java**

Java é uma linguagem de programação orientada a objetos, multiplataforma, e executável em qualquer sistema operacional aonde se tenha um interpretador instalado. Ao contrário de linguagens como o C, que ao serem compiladas geram um binário o qual pode ser executado diretamente pelo computador, o JAVA gera um *bytecode* que deverá ser interpretado por uma máquina virtual (*Java Virtual Machine*, a JVM).

Atualmente o *Java* vem sendo fortemente utilizado no setor corporativo, principalmente em sistemas e servidores *web*. Além disso, (DEITEL; DEITEL, 2011) abordam que o *Java* é a principal linguagem utilizada atualmente no desenvolvimento dos aplicativos móveis.

### 2.3.3 *React Native*

O *React Native* é um *framework* criado pelo *Facebook* para o desenvolvimento de aplicações móveis multiplataforma. Seus autores alegam que é possível fazer *apps* nativos utilizando *JavaScript* e *React* (sendo ambos a base do *framework*). De fato, isto é possível pois



*React Native* utiliza os mesmos blocos de código de interface de usuário (i.e., UI) que as plataformas Android e iOS (FACEBOOK, ). Ademais, é possível integrar códigos nativos (escritos em *Java*, *Objective-C* ou até *Swift*) ao projeto com o *framework*. Uma vantagem da ferramenta é o fato de que os aplicativos são nativos, o que reduz gargalos de desempenho e discrepâncias entre componentes. O diferencial no desenvolvimento com o *framework React Native* é a não necessidade de uma camada de renderização baseado em Web, utilizando componentes internos para a tarefa (LEITE *et al.*, 2018).

#### **2.3.4 MySQL/MariaDB**

*MariaDB Server* é um dos bancos de dados relacionais de código aberto mais populares. É feito pelos desenvolvedores originais do *MySQL* e tem garantia de código aberto. É parte da maioria das ofertas de nuvem e o padrão na maioria das distribuições Linux. É construído sobre os valores de desempenho, estabilidade e abertura, e a Fundação *MariaDB* garante que as contribuições serão aceitas por mérito técnico (MARIADB FOUNDATION, 2020).

### 3 METODOLOGIA

Neste capítulo são pontuados os meios pelo qual foram elaborados os princípios dessa pesquisa, assim como as etapas que envolvem o tipo de metodologia utilizados e as limitações enfrentadas durante o processo de pesquisa. Tendo em mente que a presente pesquisa visa o atendimento de uma necessidade da população, em específico dos produtores e usuários da CEASA, foi adotada a metodologia de pesquisa qualitativa e bibliográfica, tendo, contudo, um aplicativo para melhora da disponibilização de preços, bem como a consulta, como objetivo do trabalho.

Para a criação desse projeto serão utilizados os conhecimentos já explorados no capítulo de referencial teórico, vale salientar que o produto apresentado ao final desta monografia não representa o mesmo aplicativo utilizado pelo consumidor final, visto que é necessária aprovação do cliente.

#### 3.1 CARACTERIZAÇÃO DA PESQUISA

Partindo do objetivo desta pesquisa – que é substituir o método convencional de visualização dos dados por meio do uso da tecnologia –, fora optado por adotar o método de pesquisa qualitativa e bibliográfica que se mostrou o mais apropriado para o tipo de análise proposta. Em relação à abordagem, pesquisa qualitativa, de acordo com Richardson (1999) é capaz de “[...] descrever a complexidade de determinado problema, analisar a interação de certas variáveis, compreender e classificar processos dinâmicos vividos por grupos sociais[...]” e ainda que a pesquisa qualitativa pode “[...]contribuir com uma mudança no processo de determinado grupo, possibilitar maior nível de profundidade e o entendimento de particularidades de comportamentos dos indivíduos[...]”.

O modelo de investigação escolhido para a realização da pesquisa qualitativa enquadra-se como aplicada. Ela tem como objetivo provocar conhecimento para a aplicação prática conduzindo a resolução de problemas específicos, relaciona verdades e Interesses locais (SILVA, 2001). Quanto aos procedimentos de pesquisa, ou seja, a forma como se conduz os estudos e obtêm os dados esta pesquisa se classifica como bibliográfica. Segundo Gil (2002, p. 44) “é desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos”.

#### 3.2 ETAPAS METODOLÓGICAS

Em conformidade com o que fora anteriormente visto, optou-se por experimentar uma

variação do que se conhece como ‘métodos ágeis de desenvolvimento de *software*’. Diferentes opções dos atuais métodos utilizados pela comunidade foram experimentadas e adaptadas conforme a necessidade, no intuito de conhecer tais métodos e encontrar um (ou a mistura de mais de um deles) que suprisse as necessidades. Para a elaboração dos métodos de trabalho vistos neste projeto, foram experimentadas variações e adaptações dos métodos *Scrum*, *XP2* e do *Kanban*.

Após a análise do formulário Google que se dispunha, foi definido que para ter um desenvolvimento lógico eram necessários que os ciclos fossem divididos nas etapas listadas a seguir (apêndice A):

- a) Definição do problema: etapa que foi definida a problematização a ser tratada;**
- b) Revisão bibliográfica: etapa de levantamento de conhecimento teórico e técnico possibilitando o desenvolvimento da solução pensada para a proposta;**
- c) Prototipagem da solução: etapa na qual o levantamento de requisitos e funcionalidades do aplicativo são levantadas;**
- d) Desenvolvimento: etapa que compreende o desenvolvimento da solução proposta, também está inclusa nessa etapa o teste do aplicativo;**
- e) Conclusão: etapa de avaliação final do aplicativo e identificação de objetivos que foram alcançados, também ajustes finais a ferramenta e finalização do trabalho.**

### 3.3 DELIMITAÇÕES

Não foi possível fazer o aplicativo em tempo real, devido a não ter acesso ao banco de dados de produção, ou seja, foi necessária a criação de um banco de dados próprio para a aplicação, e a criação deste envolveu etapas a mais como impeditivos para prosseguimento do projeto, tais como: aquisição dos dados, que também ocorreu de forma manual devido problemas desenvolvimento de um drive que permitir-se *web scrapping* na página da CEASA; modelagem dos dados obtidos, uma vez que a forma de apresentação, os arquivos pdf, possuíam inconsistências na construção do documento; e por fim, a os ajustes mais finos dos dados para que fosse possível ter todas as informações necessárias para a aplicação. O capítulo seguinte vai apresentar como se deu a modelagem do problema dadas as limitações apresentadas.

## 4 MODELAGEM

O presente capítulo apresenta a modelagem de um aplicativo para a consulta dos preços por dia, utilizando filtros pertinentes como tipo de produto. A proposta conta com soluções de *back-end*, *front-end* e banco de dados. Estas soluções estão planejadas e organizadas de forma a montar fluxos de utilização e transição da informação dentro do aplicativo e na forma de interação do usuário com o aplicativo.

O primeiro passo para a modelagem de uma aplicação é o levantar os requisitos do sistema, sendo estes funcionais ou não funcionais. Durante o processo de levantamento de requisitos surgem problemas que nem sempre são solucionados. Os requisitos de um sistema são descrições dos serviços fornecidos e as suas restrições operacionais. Esses requisitos podem refletir nas necessidades dos clientes, por exemplo, controlar ou enviar um pedido por um dispositivo.

Segundo Sommerville (2011) a engenharia de software tem como objetivo apoiar o desenvolvimento do sistema, mais do que a programação individual, incluindo técnicas que apoiam as especificações, projeto e evolução de programas. Assim como todas as documentações necessárias, para fazer um programa funcionar corretamente.

A partir dessas informações, nota-se que a engenharia de software, precisa passar pelo levantamento de informações através de requisitos. É por isso que se entende a importância de realizar um levantamento de requisitos, os quais podem ser funcionais e não funcionais, que são apresentados, respectivamente, nas duas listagens a seguir

- **Requisitos funcionais**

RF-01 - Buscar produtos hortifrutigranjeiros deverá filtrar produto;

RF-02 - Buscar produtos hortifrutigranjeiros deverá filtrar variedade;

RF-03 - Buscar produtos hortifrutigranjeiros deverá filtrar classificação;

RF-04 - Buscar produtos hortifrutigranjeiros deverá filtrar tipo;

RF-05 - Buscar produtos hortifrutigranjeiros deverá filtrar origem;

RF-06 - Buscar produtos hortifrutigranjeiros deverá filtrar embalagem;

RF-07 - Exibição dos produtos filtrados por variedade;

RF-08 - Exibição detalhada dos produtos hortifrutigranjeiros;

RF-09 - Exibição detalhada da cotação de valores de determinado período selecionado pelo usuário;

RF-10 - Exibição de valores de um produto por um período em forma de lista e gráfico.

- **Requisitos não funcionais**

RNF-01 - O *backend* deve ser feito em *Spring boot*;

RNF-02 - O aplicativo deve ser feito em *React Native*;

RNF-03 - O banco de dados deve ser *MariaDB*;

RNF-04 - O sistema deve ter uma interface de fácil uso.

Uma vez levantados e entendidos os requisitos, o próximo passo é determinar os fluxos que ditaram o comportamento do sistema. A aplicação proposta tem o intuito de ser de simples acesso e consulta, sendo necessária apenas a consulta ao aplicativo para que o consumidor tenha acesso à informação desejada, conforme é ilustrado no macro fluxo do aplicativo (figura4)

**Figura 4 – Macro fluxo da aplicação**

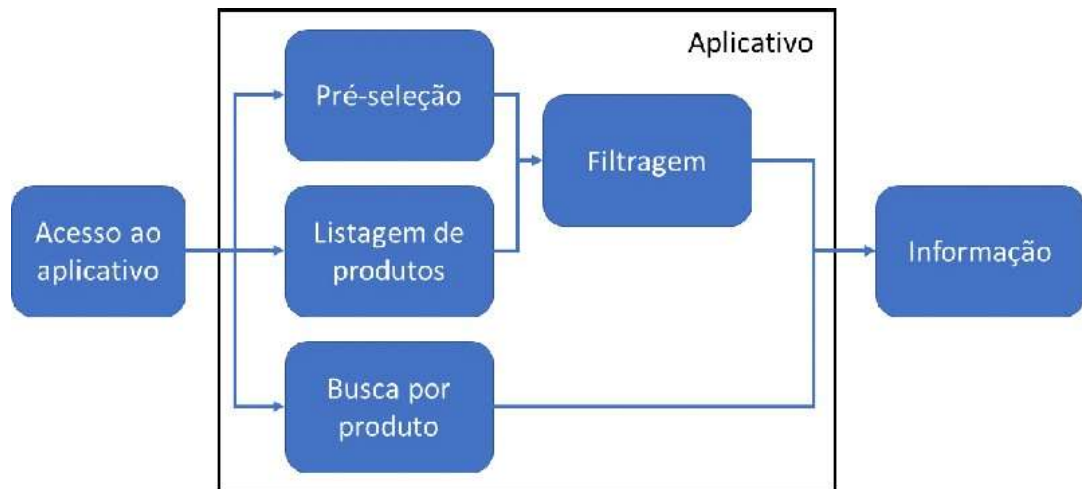


Autor

O fluxo do aplicativo por sua vez começa com a disponibilização de opções para interação com os dados sendo elas: filtros pré-prontos (Pré-seleção), listagem de produtos, e busca por produto. Uma vez que o usuário opte com seleção de produto ele terá acesso direto a informação que ele deseja, como por exemplo: caso ele busca por Abacate, ele já terá a informação direta.

Porém também é possível que o usuário opte ou por verificar todas as opções disponíveis (listagem de produtos), ou listar apenas produtos orgânicos (Pré-seleção) por exemplo. Dadas essas opções para que haja acesso à informação final é necessário um processo de filtragem que pode ser feito através do próprio aplicativo, ou simplesmente uma busca pelo usuário. A figura 5 ilustra o fluxo que ocorre no aplicativo com base nessas possibilidades.

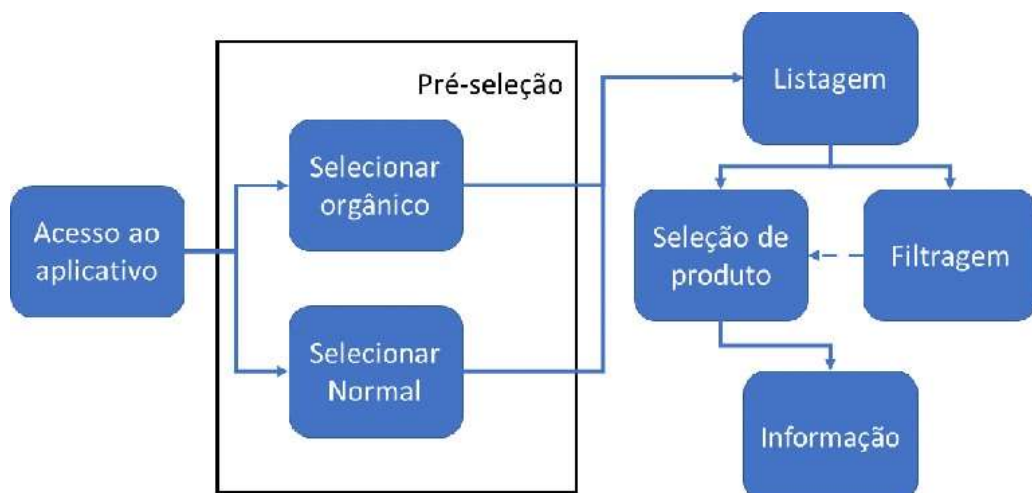
**Figura 5 – Fluxos de obtenção de informação**



A figura 6 traz de forma mais detalhada o fluxo de uso da opção de pré-seleção. São disponibilizadas duas opções principais para a seleção de produtos por classe: orgânicos e normais. O que estas opções realizam são requisições para o *back-end* da aplicação para exibição da listagem de produtos que estão marcadas como cada opção respectivamente.

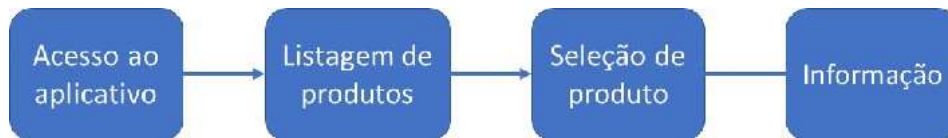
Uma vez que os produtos são listados para que o usuário chegue a informação desejada podem ocorrer dois processos bem simples, ou há uma seleção do produto de forma manual (o usuário procurando dentro da lista apresentada), ou aplicação de filtros, como valor, peso e tipo, para que seja reduzida as opções de produtos e o usuário chegue a informação final a respeito do produto de interesse, conforme é ilustrado na figura 6.

**Figura 6 – Fluxo de informação via pré-seleção**



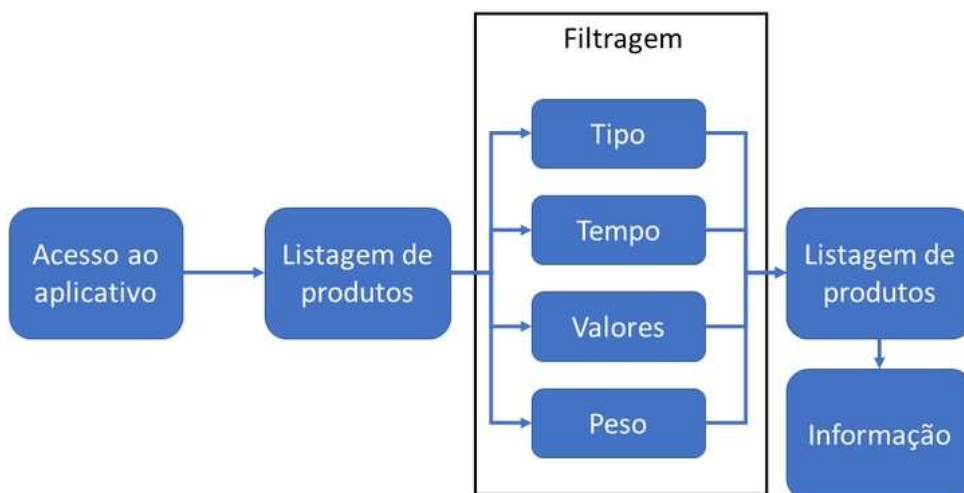
Outro fluxo bem parecido é de listagem geral de produtos (figura 7), no qual o usuário é apresentado a todos os produtos presentes na plataforma, e então ele faz a seleção do produto, seja de forma manual seja através dos filtros disponíveis.

**Figura 7 – Fluxo de informação via listagem geral de produtos**



A figura 8 por sua vez, ilustra o processo que ocorre na aplicação quando há o uso dos filtros disponíveis. De forma resumida, dada uma listagem de produtos é possível que esta seja filtrada baseada em informações como: tipo, tempo (seleção de um intervalo de tempo para busca), valores, além do peso. É possível a aplicação individual ou combinada desses filtros, resultando em uma segunda listagem dos produtos que atendem as condições dos filtros aplicados, e por meio dessa listagem é possível o usuário chegar a informação final, que seria a cotação do produto, e suas informações estatísticas disponíveis na aplicação.

**Figura 8 – Fluxo de aplicação dos filtros**



O próximo capítulo apresenta os requisitos levantados baseados nesses fluxos e o protótipo alcançado.

## 5 DESENVOLVIMENTO

Dado que não foi disponibilizado o acesso às informações dos produtos da CEASA seja por acesso direto a um banco de dados, seja via alguma API (*Application Programming Interface*), ou seja, uma interface de aplicação que possua acesso àquele na forma de requisições web, a única forma de acesso livre aos dados é pelo *site* da CEASA/SC.

No *site* a informação está disponível na forma de arquivos PDF, sendo necessário o *download* de toda a massa de arquivos para ter os dados. A estratégia adotada foi a conversão dos arquivos PDF em arquivos de dados .csv, para realizar essa conversão foi utilizado um script auxiliar que faz a leitura de um documento .pdf e o interpreta como um arquivo tabulado, tendo como base a tabulação do arquivo tornando possível montar uma estrutura similar à de um csv. Considerando o volume expressivo de documentos e informações, foi adotado o Python 3.7 para a criação deste visto que teria um desempenho mais interessante para o desenvolvimento.

Código 1. Código auxiliar para a conversão dos arquivos PDF para arquivos CSV

```
1. import tabula
2. import json
3. import pandas
4. # readinf the PDF file that contain Table Data
5. # you can find find the pdf file with complete code in below
6. # read_pdf will save the pdf table into Pandas Dataframe

7. import os, fnmatch

8. def find_files(directory, pattern):
9.     for root, dirs, files in os.walk(directory):
10.         for basename in files:
11.             if fnmatch.fnmatch(basename, pattern):
12.                 filename = os.path.join(root, basename)
13.                 yield filename

14. for filename in find_files('ceasa', '*.pdf'):
15.     print (f'Found pdf file:{filename}')

16. collumns = ['Produto Variedade', 'Classificação',
```



```

17. 'Tipo','Origem','Emabalagem','Conv. Kg',
18. 'Minimo','Comum','Máximo','Comum Kg']

19. df2 = tabula.read_pdf(filename, area=(130,0,760,600),
20. encoding='ANSI',stream=True,pages='all',
21. multiple_tables=True,pandas_options={'header':None})

22. aux_df = []
23. for item in df2:
24.     if item.shape[1] != 10:
25.         if item.shape[1] > 10:
26.             item = item.drop(1, axis=1)
27.         else:
28.             item.loc[:,10] = 'NaN'
29.             item = item[[0,10,1,2,3,4,5,6,7,8]]
30.     item.columns= columns
31.     aux_df.append(item)
32. else:
33.     item.columns= columns
34.     aux_df.append(item)
35. # print('\n')

36. new_DF = pandas.concat(aux_df).to_csv(filename[:-3]+
    'csv',index=False,sep=';',encoding='utf-8')
37. print (f'file converted:{filename}')

```

Apesar do script ter um desempenho satisfatório na conversão dos arquivos, os PDFs obtidos no site da CEASA apresentavam inconsistências na sua formatação, causando erros que comprometiam a integridade dos dados, uma vez que existe mais de uma formatação para a tabulação do PDF, portanto a interpretação não se deu uniformemente, sendo necessário o ajuste reformatação do documento, trecho do código entre a linha 22 e linha 34, no qual a coluna extra que aparece é ignorada e a coluna que contem a informação correta é realocada, de forma a criar um padrão único nos documentos.

Uma segunda conversão foi realizada, desta vez entre registros (linhas do arquivo .csv) para comandos de inserção do SQL, para realizar a alimentação dos dados no banco. Para esta segunda conversão foi utilizado um *script* em Java (mesma tecnologia utilizada no *back-end*), isto porque o mesmo seria utilizado por parte da CEASA (ou Gestor de Informação) embutido na aplicação responsável pelo gerenciamento do banco de dados utilizado pela aplicação proposta. Contudo, não sendo o foco deste trabalho o gerenciamento por parte da CEASA das informações, foi explorado e desenvolvido apenas a aplicação para o consumidor final.

Código 2. Código auxiliar para conversão de CSV para SQL

```
public static final void CsvToSQL(Enters enters) throws IOException {  
    BufferedReader bufferedReader = new BufferedReader(new FileReader(enters.getSourceFile()));  
    String year = enters.getSourceFile().substring(53, 57);  
    String month = enters.getSourceFile().substring(58, 60);  
    String day = enters.getSourceFile().substring(61, 63);  
    String date = year + "-" + month + "-" + day;  
    FileWriter fileWriter = new FileWriter(date + ".sql");  
    PrintWriter printWriter = new PrintWriter(fileWriter);  
  
    List<String> lines = new ArrayList<>();  
  
    bufferedReader.readLine();  
    while (bufferedReader.ready()) {  
        String str = bufferedReader.readLine();  
        if (str.split(enters.getSplitRegex()).length != 0) {  
            lines.add(str);  
        }  
    }  
  
    String[][] lineVector = new String[lines.size()][];  
  
    for (int i = 0; i < lines.size(); i++) {  
        String[] lineSplit = lines.get(i).split(enters.getSplitRegex());  
        if (lineSplit[0].isEmpty()) {  
            boolean test = false;  
            int j = 1;  
            while (test) {  
                if (lineVector[i - j][1] != null) {  
                    j--;  
                } else {  
                    lineVector[i - 1][1] += " " + lines.get(i).split(enters.getSplitRegex())[1];  
                    lineVector[i] = new String[1];  
                    test = true;  
                }  
            }  
        }  
    }  
}
```

```

    }
}

} else {
    lineVector[i] = lines.get(i).split(enter.getSplitRegex());
}
}

for (String[] strings : lineVector) {
    if (strings != null) {
        for (int j = 0; j < strings.length; j++) {
            if (strings.length == 1) {
                break;
            }

            if (j == 0) {
                printWriter.print("INSERT INTO " + enter.getTableName() + " (produto, classificacao, tipo,
origem, embalagem, conv_kg, minimo, comum, maximo, medio_kg, data) VALUES (");
            }

            String string = strings[j];
            Object object = Functions.convertVariableType(string);

            if (object instanceof String) {
                if (!string.isEmpty()) {
                    string = string.replaceAll("\s{2,}", " ").trim();
                    if (string.matches("[a-zA-Z\u00C0-\u024F ()/0-9-]+")) {
                        printWriter.print("'" + string + "'");
                    } else {
                        printWriter.print(string);
                    }
                } else {
                    printWriter.print("null");
                }
            } else if (object instanceof Boolean || object instanceof Number) {
                if (object instanceof Double) {
                    if (!((Double) object).isNaN()) {
                        printWriter.print(string.replace(",", "."));
                    } else {
                        printWriter.print("null");
                    }
                } else {
                    printWriter.print(string);
                }
            }
        }
    }
}

```

```

    }
}

if (j != (strings.length - 1)) {
    printWriter.print(", ");
}

if (j + 2 > strings.length) {
    printWriter.println(", " + date + "");
}
}
}
}
bufferedReader.close();
printWriter.close();
}

```

O primeiro processo, código 1, foi aplicado a toda a massa de arquivos obtidos do *site* da CEASA/SC disponibilizando ao final uma massa de documentos CSV. O segundo momento foi de conversão da massa de tabelas de dados para a estrutura do comando *insert* do SQL, para isso foi utilizado o código 2 que realiza a leitura dos arquivos CSV e montava os comandos *insert* conforme o desenvolvimento inicial do banco de dados. Após isso ter sido feito, foi necessário mais um ajuste para se adequar a estrutura do banco, que envolve duas tabelas sendo elas produto e classificação. A tabela produto possui os dados embalagem, origem, produto, tipo, variedade; e a tabela classificação apenas a classificação, foram tirados da parte textual dos dados da tabela e passado todos para *enum*, exceto data, para reduzir o tamanho.

Às duas tabelas que existem no banco estão vinculadas conforme é apresentado na figura 9, sendo a TB\_PRODUTO os dados que são específicos de cada produto, em geral, não compartilhados entre eles. Ambas as tabelas possuem “*ids*” próprios para como chave primaria para buscas, e estão conectadas pela chave estrangeira “*produto\_id*”, que permite associar a classificação informada ao produto correto.

Figura 9 – Esquema geral das tabelas do aplicativo



Autor

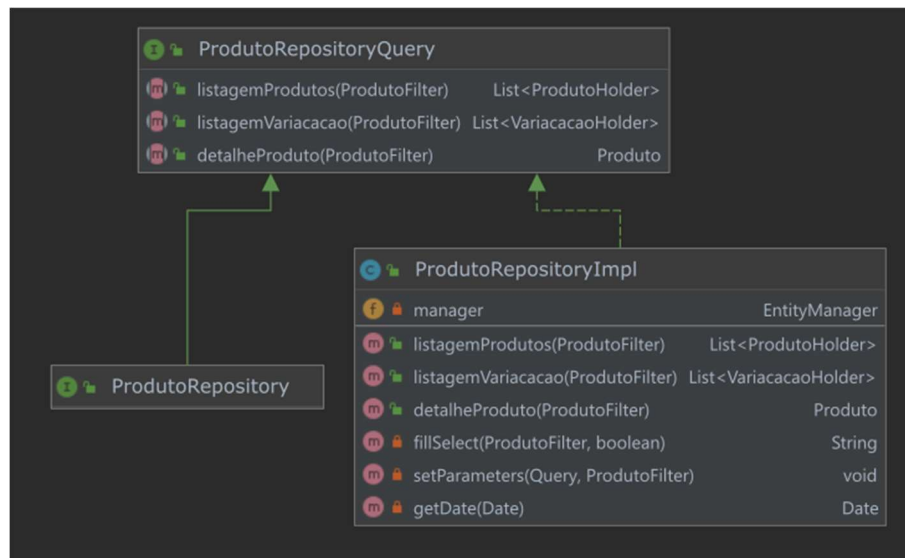
A TB\_CLASSIFICAÇÃO por sua vez, armazena as informações que podem ser comuns a vários produtos, como por exemplo a característica lavada ou grande, e assim com as *tags* nome e variedade também são *enums*, para otimizar o armazenamento de informação no banco de dados. O *enum* comentado é a estratégia de criar no *back-end* um dicionário de valores a serem interpretados, seja na leitura, seja na escrita no banco de dados. Consiste em uma equivalência de uma sequência numérica com um valor qualquer, como, por exemplo, é possível associar o valor “0” com a descrição “ABACAXI”. Desta forma é possível dividir e simplificar o armazenamento de informações no banco de dados, deixando este com as informações citadas acima, vinculados a descrição do item que está no *back-end*. Desta forma é possível remontar toda a informação durante o processo de requisição feito pelo *front-end* ao *back-end*.

Uma vez construído o banco de dados, é possível começar a trabalhar as partes de *back-end* e *front-end*, para a construção do protótipo da aplicação, que conta com um total de oito telas sendo elas de interação e visualização de informações. As seções a seguir trazem mais informações a respeito do processo de desenvolvimento da aplicação.

## 5.1 DESCRIÇÃO UML DO BACKEND

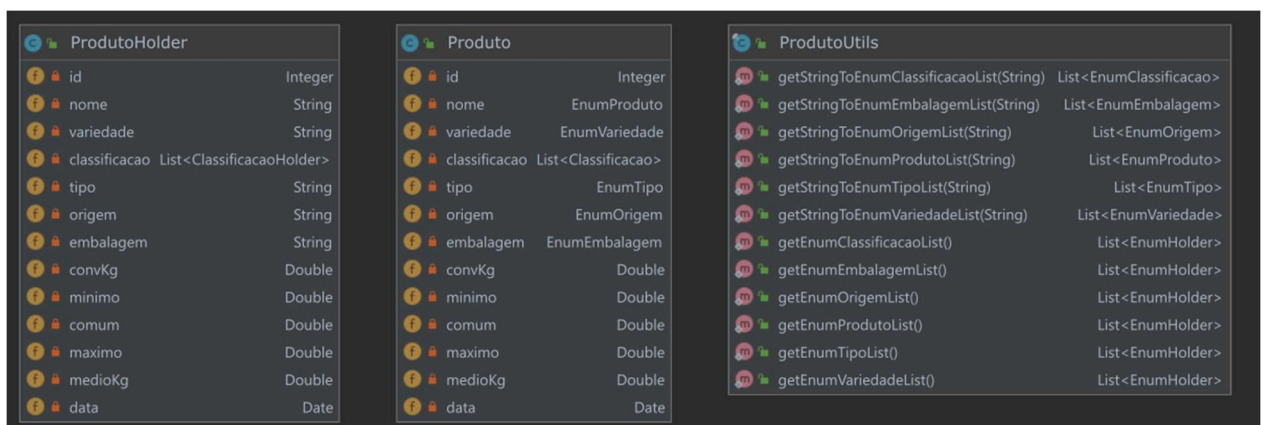
Também parte do desenvolvimento da aplicação, o diagrama UML (*Unified Modeling Language*) representa a descrição e documentação da relação de objetos de uma aplicação, no caso do presente trabalho, vai descrever os métodos e os objetos que constituem a aplicação, conforme os fluxos apresentados no Capítulo 4. Sendo o padrão de projeto adotado uma versão modificada do MVC (*Model-View-Controller*), os diagramas a seguir descreveram a estruturação do *back-end* para a adequação dos fluxos citados.

**Figura 10 – Relação dos métodos de buscas da tela inicial do momento inicial da aplicação**



Para a etapa de busca de informações, são disponíveis e implementados os métodos presentes na caixa inferior esquerda que recebem as informações do *front-end*, então montam as informações para recuperar as informações do banco de dados utilizando os métodos da caixa superior, Figura 10, que monta os comandos necessários do SQL.

**Figura 11 – Classe produto e seus métodos**

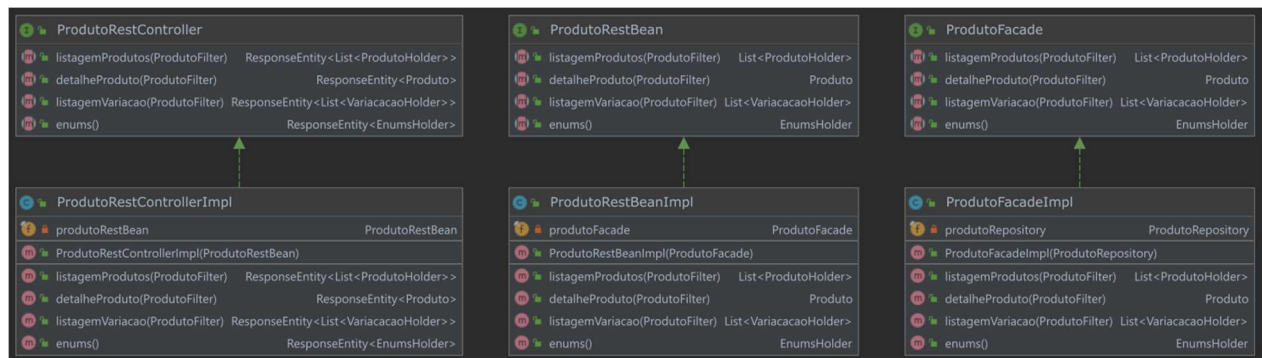


Feita a consulta no banco de dados para a recuperação da informação, a classe produto e seus métodos (figura 11) passam a fazer a descrição da informação no banco e sua relação com o *backend*, e como reestruturar a informação, que ainda passará por etapas de filtragem e

manipulação das informações dependendo da tela na qual o produto ou sua informação estão sendo exibido.

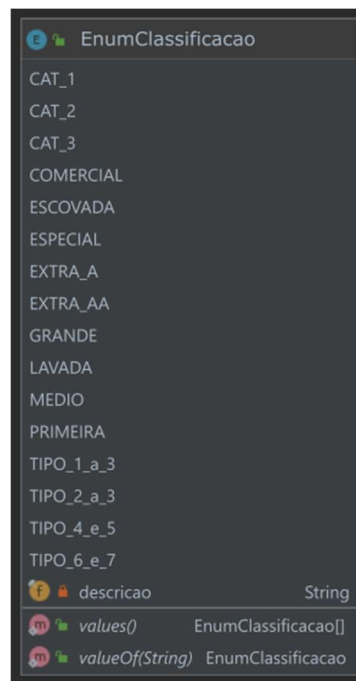
Uma vez recuperada as informações do banco de dados e quais as informações que precisam ser exibidas na tela que o usuário estiver, os métodos da figura 12 são acionados para formar a informação final a ser apresentada para o usuário. Estes métodos também são os responsáveis por remontar as informações que foram anteriormente traduzidos em *enums*<sup>2</sup>.

**Figura 12 – Métodos de manipulação do produto**



De exemplo, a figura 13 ilustra o que é um *enum* responsável por fazer a tradução os valores da TB\_CLASSIFICACAO para a exibição da classificação do produto, na tela de detalhes do produto por exemplo, que será apresentada junto das demais telas do sistema na seção seguinte. O diagrama UML completo do sistema está no apêndice do trabalho.

**Figura 13 – Exemplo de enumeração: Classificação**



<sup>2</sup> Enum é uma estratégia de modelagem de dados no qual é criado um dicionário de chave:valor que gera um significado entre *back-end* e banco de dados, por exemplo Enum\_produto = {1: "Abacaxi"}

## 5.2 PROTÓTIPO

A tela inicial do aplicativo, figura 14, tem objetivo de direcionar o usuário para a listagem de produtos, com quatro opções possíveis para abrir a listagem sendo dois deles filtro por tipo (Orgânico e Normal), e os outros dois, filtros pelo nome do produto além de acessar a lista sem filtro

**Figura 14 – Tela inicial**



A listagem de produtos (figura 15) tem como objetivo exibir a lista de produtos (filtrados ou não) exibindo a cada linha o nome do produto, tipo e valor médio do último dia que tem valores coletados/registrados. Além de possuir um botão que leva para uma tela de pesquisa (figura 22).



Figura 15 – Tela Produtos

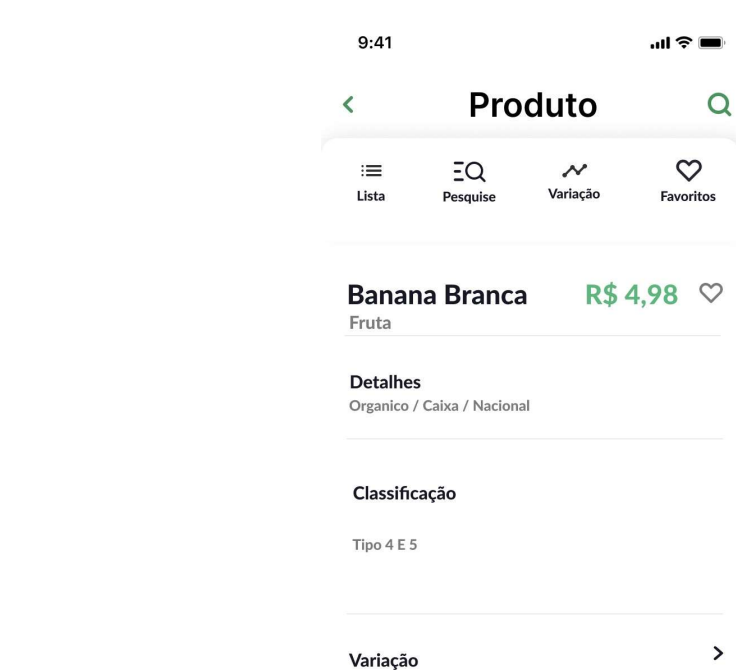


The screenshot shows a mobile application interface for a product catalog. At the top, the status bar displays the time 9:41 and signal strength. The app's title 'Produtos' is centered at the top with a back arrow on the left and a search icon on the right. Below the title is a navigation bar with four icons: a list icon labeled 'Lista', a magnifying glass labeled 'Pesquise', a line graph labeled 'Variação', and a heart icon labeled 'Favoritos'. The main content area is a list of products, each with a heart icon, the product name, its type, and its price. The products are: Abacate (organico, R\$ 4,98), Abacaxi (organico, R\$ 4,98), Abobora (organico, R\$ 4,98), Banana (organico, R\$ 4,98), Mamão (normal, R\$ 4,98), Mamão (normal, R\$ 4,98), Banana Prata (organico, R\$ 4,98), Maçã (normal, R\$ 4,98), Mamão (normal, R\$ 4,98), Pêssego (organico, R\$ 4,98), and Melão (organico, R\$ 4,98). The price for the last Mamão entry is highlighted in red.

Produto	Tipo	Preço
Abacate	organico	R\$ 4,98
Abacaxi	organico	R\$ 4,98
Abobora	organico	R\$ 4,98
Banana	organico	R\$ 4,98
Mamão	normal	R\$ 4,98
Mamão	normal	R\$ 4,98
Banana Prata	organico	R\$ 4,98
Maçã	normal	R\$ 4,98
Mamão	normal	R\$ 4,98
Pêssego	organico	R\$ 4,98
Melão	organico	R\$ 4,98

As figuras (figura 16 e 17) ilustram a forma de apresentação dos detalhes dos produtos, sendo eles: nome, valor médio, tipo, embalagem, origem e classificação.

Figura 16 – Tela Detalhes Produto



**Figura 17 – Tela Detalhes Produto - sobre**



Variação do produto (figura 18) tem como base a listagem de um único produto a partir de um período selecionado pelo usuário, sendo exibindo quatro valores para cada dia selecionado pelo filtro sendo eles mínimo, comum, máximo e comum/kg.

Figura 18 – Tela variação

9:41

<

Variação

Q

Abóbora

De

Até

MM / DD / YYYY

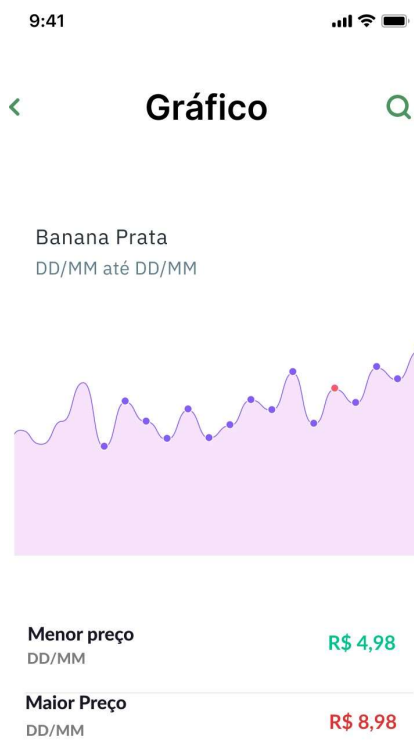
MM / DD / YYYY

Gráfico >

	Min	Comum	Max	Comum /kg
DD / MM	12,00	12,00	15,00	12,00
DD / MM	2,30	2,40	2,50	2,30
DD / MM	7,00	7,00	7,00	4,40,
DD / MM	4,40	4,40	4,40	4,40
DD / MM	4,40	4,40	4,40	4,40
DD / MM	4,40	4,40	4,40	4,40
DD / MM	4,40	4,40	4,40	4,40

Gráfico, figura 19, exhibe apenas um dos valores (mínimo, comum, máximo e comum/kg) de um período.

**Figura 19 – Tela Gráfico**



O aplicativo possui uma tela de busca, figura 21, seu funcionamento principal é facilitar a busca rápida caso o usuário deseje buscar um produto específico de forma rápida, caso o produto não exista é mostrado um *feedback* que o mesmo não existe (figura 20).

**Figura 20 – Tela busca não encontrado**

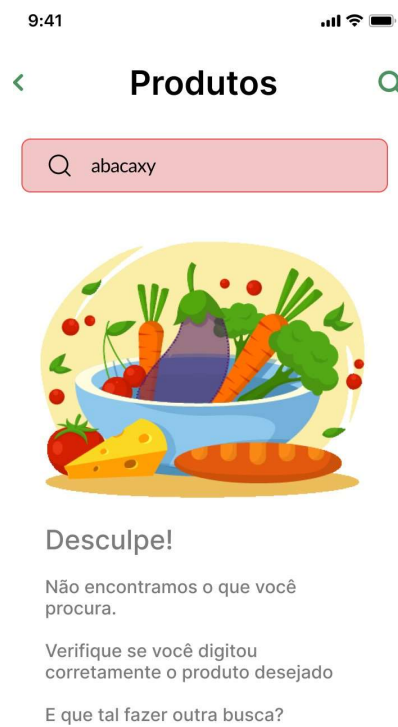


Figura 21 – Tela busca encontrada



Figura 22 – Tela filtro

9:41

<

Filtro

Limpar Filtro

Origem do produto

☒ Nacional

☐ Internacional

Produção

☐ Orgânica

☐ Normal

Embalagem

☐ Caixa

☐ Kilo

☐ Maço

☐ Molho

☐ Bandeja

☐ Dúzia

☐ Litro

☐ Fardo

Classificação

☐ Tipo 1 a 3

☐ Tipo 2 a 4

☐ Tipo 2 a 3

☐ Tipo 4 e 5

☐ Tipo 6 e 7

☐ Especial

☐ Escovada

☐ Primeira

☐ Extra A

☐ Extra AA

☐ Cat 1

☐ Cat 2

☐ Cat 3

☐ Medio (a)

☐ Grande

Aplicar Filtro



Caso o usuário tem como objetivo filtrar a lista de resultado, por exemplo produto do tipo orgânico, é possível que isso seja feito (figura 23).

**Figura 23 – Tela lista filtro aplicado**



## 6 CONSIDERAÇÕES FINAIS

Partindo do problema inicial proposto, que era o método ultrapassado de visualização dos dados e valores dos produtos da CEASA/SC, foram realizados estudos sobre a aplicabilidade de tecnologias móveis para a solução de problemas. Com base em estudos de diversos trabalhos semelhantes, notou-se que os resultados obtidos através da aplicabilidade da tecnologia móvel são sempre positivos, o que demonstrou e justificou a necessidade e viabilidade da construção desse aplicativo.

Relembrando do fator limitante ligado ao banco de dado, o qual não se obteve acesso, os scripts utilizados para tratamento dos dados presentes nos PDFs conseguiram suplantam essa questão, permitindo a criação de um banco de dados integro com todos os dados disponíveis no site. Com a criação desse banco de dados, o desenvolvimento da aplicação também passou por facilidades, visto que foi possível modelar ambos para que houvesse melhor desempenho no uso geral, dada a versatilidade com a qual poderiam ser construídas as tabelas e também os *enums* (dicionários) da aplicação.

Também devido a isso foi possível trafegar dentro da aplicação com a quantidade exata de informação entre casa uma das telas, permitindo assim que a solução tenha um comportamento suave e sem onerar desnecessariamente do dispositivo mobile, dado que as requisições não trazem informações desnecessárias para as telas. Além disso a visualização dos dados dos produtos, se provou também mais efetiva e estatística, visto que é possível comprar cotações de um período e ter essa informação tanto em lista quanto em apresentação gráfica.

Ouro ponto bem executado na aplicação é a sua navegação, ou seja, a integração do usuário final com esta se dá de forma intuitiva, uma vez que as telas não trazem elementos desnecessários, facilitando assim seu uso tanto por consumidores que já possuem alta adaptabilidade com tecnologias móveis, quanto por consumidores que não possuem tanta afinidade com o uso de *smartphones*.

Como apresentado, foi possível a construção de uma solução para o problema proposto. O protótipo de aplicativo é funcional e pode atender às necessidades aos interessados para a visualização dos dados tabulados. Finalmente, para trabalhos futuros, podem ser sugeridas algumas melhorias para um funcionamento completo do protótipo.

Dentre elas, por exemplo, expansões na base de dados para melhor adequação a demanda caso haja necessidade, revisão da modelagem para garantir melhor desempenho caso possível. Ao nível de integração entre *back-end* e *front-end* melhorias no fluxo de informações e requisições entre *front-end* e *back-end* para tornar o aplicativo fluido e sem contratempos. Ao nível de *front-end* mudanças no visual e experiência do usuário, conforme necessidade do cliente final.

Além das expansões dentro da própria aplicação para torná-la mais adequada ao cenário de aplicação, é possível também desenvolver todo o cenário de catálogo e *e-commerce*, usando como base a aplicação atual que já conta com uma base de dados e serviço de catálogo. Para esta evolução dois pontos seriam cruciais, primeiro o desenvolvimento de uma área de administração, responsável pela injeção diária de informações e cotações; e segundo a integração com algum meio de pagamento, tais como *maxipago* ou *pagseguro*.

## REFERÊNCIAS

- ABRACEN. **Sobre a Abracen - Abracen**. 2020. Disponível em: <https://abracen.org.br/sobre/>. Acesso em: 20 de julho de 2020.
- ALMEIDA, G. A. M. de. **Fatores de escolha entre metodologias de desenvolvimento de software tradicionais e ágeis**. 2017. Tese (Doutorado) — Universidade de São Paulo.
- AMBLER, S. W. **Feature Driven Development (FDD) and Agile Modeling**. 2005. Disponível em: <http://agilemodeling.com/essays/fdd.htm>. Acesso em: 18 maio 2020.
- ANDERSON, D. J. **Kanban: successful evolutionary change for your technology business**. [S.l.]: Blue Hole Press, 2010.
- BECK, K. Embracing Change with Extreme Programming. **IEEE Computer**, v. 32, n. 10, p. 70 – 77, 1999. Disponível em: <http://doi.ieeecomputersociety.org/10.1109/2.796139>.
- BOEHM, B. W. A spiral model of software development and enhancement. **Computer**, IEEE, v. 21, n. 5, p. 61 – 72, 1988a.
- BOEHM, B. W. **The Spiral Model of software development**. 1988b. Disponível em: [https://www.researchgate.net/figure/The-Spiral-Model-of-software-development-from-Boehm-1988\\_fig2\\_2497256](https://www.researchgate.net/figure/The-Spiral-Model-of-software-development-from-Boehm-1988_fig2_2497256). Acesso em: 20 de Novembro de 2020.
- CEASA-SC. **A Ceasa**. 2020. Disponível em: <https://www.ceasa.sc.gov.br/index.php/institucional/a-ceasa>.
- CHACON, S.; STAUB, B. **Pro Git**. 2020. Disponível em: <https://git-scm.com/book/en/v2>. Acesso em: 18 de Novembro de 2020.
- COCKBURN, A. **Agile Software Development: Software Cockburn, A., Agile Software Development: Software through People**. 2001.
- DEITEL, P.; DEITEL, H. **Java Como Programar**. 8. ed. São Paulo: Pearson, 2011. ISBN 978-85-7605-563-1.
- EBC. Empresa Brasileira de Comunicação. **Empresa Brasil de Comunicação**, 2019. Disponível em: <http://www.ebc.com.br/>. Acesso em: 18/02/2019.
- FACEBOOK.ReactNative.Disponível em: <https://developers.facebook.com/docs/react-native>.
- FADEL, A. C.; SILVEIRA, H. da M. **Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean**. 2010. 26 p. Dissertação (Gestão e Qualidade de Projetos) — UNICAMP. Disponível em: [https://www.academia.edu/617187/Metodologias\\_%C3%A1geis\\_no\\_contexto\\_de\\_desenvolvimento\\_de\\_software\\_XP\\_Scrum\\_e\\_Lean](https://www.academia.edu/617187/Metodologias_%C3%A1geis_no_contexto_de_desenvolvimento_de_software_XP_Scrum_e_Lean). Acesso em: 19 nov 2019.
- GITHUB. **GitHub is how people build software**. 2020. Disponível em: <https://github.com/about>. Acesso em: 18 de Novembro de 2020.

HIGHSMITH, J. A.; HIGHSMITH, J. **Agile Software Development Ecosystems**. Addison-Wesley, 2002. 404 p. (Agile software development series). ISSN 9780201760439. ISBN 9780201760439. Disponível em: <https://books.google.com.br/books?id=uE4FGFOHs2EC>.

LEAN INSTITUTE BRASIL. Definição Filosofia Lean. 2018. Disponível em: <https://www.lean.org.br/o-que-e-lean.aspx>. Acesso em: 17/03/2018. MAR, K.;

SCHWABER, K. Scrum with XP. **Informit. com**, 2002.

MARIADB FOUNDATION. **About MariaDB**. Disponível em: <https://mariadb.org/>.

MEDEIROS, H. **Modelos de processos ágeis: conceitos e princípios**. Disponível em: <https://www.devmedia.com.br/modelos-de-processos-ageis-conceitos-e-principios/30059>.

PRESSMAN, R. **Engenharia de Software - Uma Abordagem Profissional**. 7. ed. [S.l.]: Editora ARMED, 2011.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. Rio de Janeiro: McGraw-Hill, 2006.

PRIKLADNICKI, R.; WILLI, R.; MILANI, F. **Métodos Ágeis para desenvolvimento de software**. Porto Alegre: Bookman, 2014.

RICHARDSON, R. J. **Pesquisa social: métodos e técnicas**. Atlas, 1999. ISBN 9788522421114. Disponível em: <https://books.google.com.br/books?id=fO0QAAAACAAJ>.

SOARES, M. dos S. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. **Revista Eletrônica de Sistemas de Informação**, v. 3, n. 1, 2004.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 544 p. ISBN 978-85-7936-108-1.

SUTHERLAND, K. S. e J. **Scrum**. 2017. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>.



## **APÊNDICES**

## APÊNDICE A – Formulário

20/11/2020

Aplicativo Ceasa/SC

## Aplicativo Ceasa/SC

\* Required

1. Qual o seu sexo? \*

Mark only one oval.

☐ Feminino☐ Masculino

2. Qual sua faixa etária? \*

Mark only one oval.

☐ 18 a 24☐ 25 a 35☐ 36 a 45☐ 46 a 55☐ 56+

3. Você possui um Smartphone? \*

Mark only one oval.

☐ Sim☐ Não

4. O que você acha sobre a facilidade de encontrar os valores dos produtos do CEASA/SC por um aplicativo? \*

Mark only one oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente



20/11/2020

Aplicativo Ceasa/SC

5. Você tem dificuldades para localizar e visualizar esse tipo de informação hoje? \*

Mark only one oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

6. Quais funções você acha que o aplicativo deve ter? \*

---

---

---

---

---

7. Você conhece algum serviço para este mesmo fim? \*

Mark only one oval.

- ☐ Sim  
☐ Não

8. Caso conheça algum serviço, qual?

---

9. Com sua frequência de consulta a esses dados \*

Mark only one oval.

- ☐ 1 a 3 vezes no dia  
☐ 3 ou mais vezes no dia  
☐ 1 a 3 vezes na semana  
☐ Não consulto

20/11/2020

Aplicativo Ceasa/SC

10. Você atua em qual segmento \*

*Mark only one oval.*

- ☐ Supermercado
- ☐ Restaurante
- ☐ Profissional
- ☐ Consumidor final

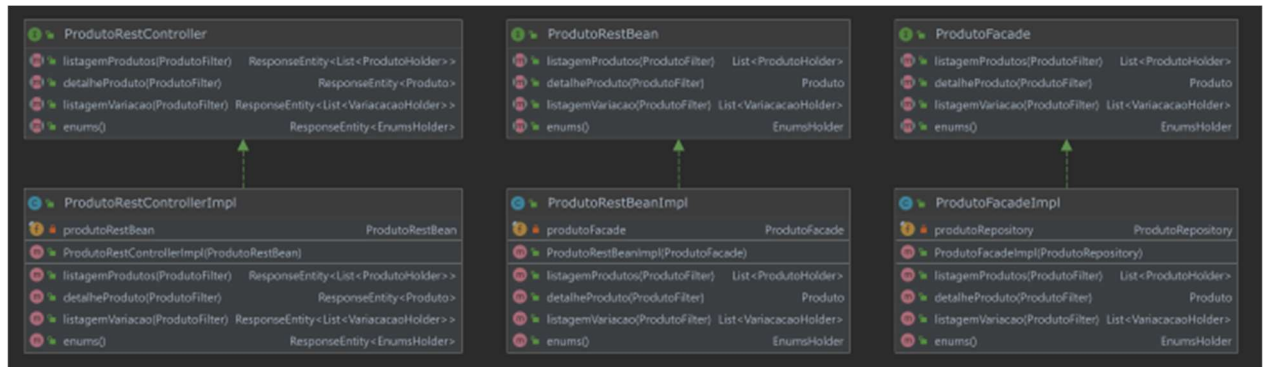
---

This content is neither created nor endorsed by Google.

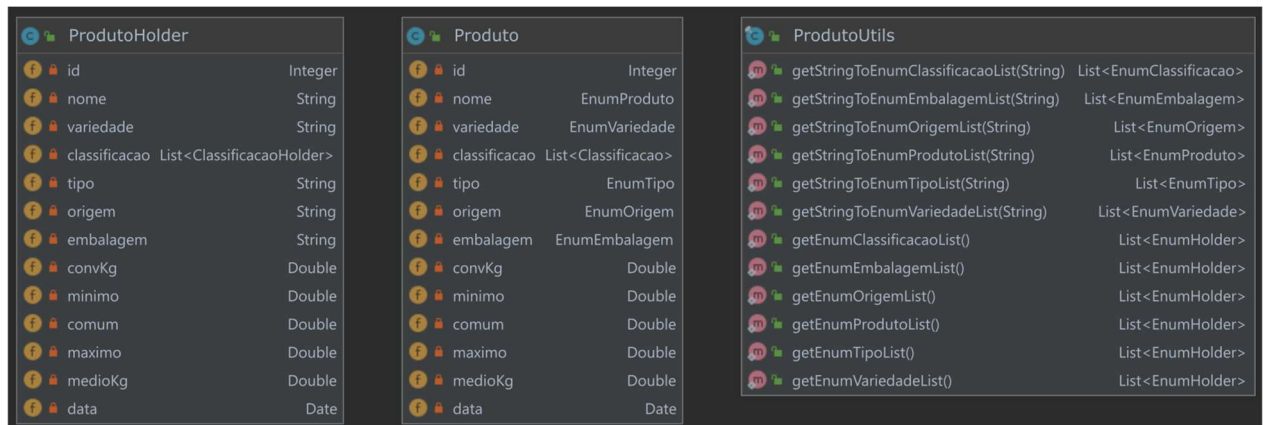
Google Forms

## APÊNDICE B – Diagrama UML completo do *back-end* da aplicação

*Figura. Camada de comando e comunicação com o banco de dados*



*Figura. Modelo de produto e seus métodos*



*Figura. Enums de Origem e Tipo*

