

INSTITUTO FEDERAL DO PARANÁ
CAMPUS PARANAVAI
CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE

MATEUS BARBEIRO GARCIA

**TeusControle: Sistema Computacional para
Gerenciamento de uma Loja de Autopeças**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia de Software

Orientador: Prof. Dr. Ayslan Trevizan Possebom

Paranavaí
2022

INSTITUTO FEDERAL DO PARANÁ

Reitor: Prof. Odacir Antonio Zanatta

Diretor do Campus Paranavaí: Prof. José Barbosa Dias Júnior

Coordenador do Curso de Bacharelado em Engenharia de Software: Prof. Me. Rafael
Henrique Dalegrave Zottesso

RESUMO

A gestão de estoque é uma atividade fundamental para uma empresa. Sendo assim, o uso de sistemas de informação se mostra vantajoso. Neste contexto, este trabalho objetiva registrar o estudo e desenvolvimento de uma solução computacional para atender a gestão de estoque utilizando de artifícios da Engenharia de Software e aplicando-o em uma loja de autopeças. A solução resultante apresentada chama-se TeusControle, composta pelos módulos: *REST API* (o projeto que implementa a regra de negócio) e UI (o projeto que implementa a interface do usuário). No trabalho, é apresentado o que foi realizado para construção da solução, ou seja, como foi desenvolvida a modelagem e implementação do TeusControle. Também realizou-se uma validação da solução pelo cliente em potencial, onde foram levantados alguns *feedbacks* e sugestões. Para finalizar, também foram apresentadas algumas sugestões para a evolução e trabalhos futuros.

Palavras-chave: Software. Gerenciamento de estoque. Controle de estoque.

ABSTRACT

The inventory management is a fundamental activity for a company. So the use of information systems is advantageous. In this context, this work aims to record the study and development of a computational solution to attend the inventory management using artifices of Software Engineering and applying it in an auto parts store. The resulting solution presented is called TeusControle, that is composed of the modules: Rest API (the project that implements the business rule) and UI (the project that implements the user interface). In this work, it is presented what was accomplished to build the solution, in other words, how the modeling and implementation of TeusControle were developed. A validation of the solution was also carried out by the potential customer, where some feedbacks and suggestions were collected. Finally, were also presented some suggestions for evolution and future works.

Keywords: Software. Inventory management. Stock control.

LISTA DE FIGURAS

Figura 4.1	Atividades para o desenvolvimento do software TeusControle.....	17
Figura 5.1	Diagrama de Classes.....	22
Figura 5.2	Diagrama de Caso de Uso.	24
Figura 5.3	Diagrama de Arquitetura.	25
Figura 6.1	Estrutura da implementação do serviço <i>Rest API</i>	27
Figura 6.2	Documentação da Api TeusControle - <i>Swagger UI</i>	28
Figura 6.3	Estrutura da implementação do Flutter App.....	30
Figura 6.4	Tela Login - Flutter App.....	31
Figura 6.5	Menu Lateral - Flutter App.....	31
Figura 6.6	Tela Usuários - Flutter App.	32
Figura 6.7	Formulário Cadastro de Usuários - Flutter App.	33
Figura 6.8	Formulário Edição de Usuários - Flutter App.	33
Figura 6.9	Detalhes do Usuários - Flutter App.	34
Figura 6.10	Exclusão do Usuário - Flutter App.	35
Figura 6.11	Tela Produtos - Flutter App.	35
Figura 6.12	Formulário Cadastro de Produtos - Flutter App.	36
Figura 6.13	Formulário Edição de Produtos - Flutter App.	37
Figura 6.14	Detalhes do Produto - Flutter App.....	37
Figura 6.15	Exclusão do Produto - Flutter App.	38
Figura 6.16	Tela Entradas - Flutter App.....	38
Figura 6.17	Formulário Cadastro de Entradas - Flutter App.	39
Figura 6.18	Formulário Edição de Entrada - Flutter App.	40
Figura 6.19	Detalhes da Entrada - Flutter App.	41
Figura 6.20	Exclusão da Entrada - Flutter App.	41
Figura 6.21	Tela Vendas - Flutter App.	42
Figura 6.22	Formulário Cadastro de Vendas - Flutter App.....	42
Figura 6.23	Recibo de Vendas - Flutter App.....	43
Figura 6.24	Formulário Edição de Vendas - Flutter App.	44
Figura 6.25	Detalhes da Venda - Flutter App.....	45
Figura 6.26	Exclusão de Venda - Flutter App.....	45
Figura 6.27	Tela Ponto de Venda - Flutter App.	46
Figura 6.28	Tela Buscar Produto - Flutter App.....	47
Figura 6.29	Informações do Software TeusControle - Flutter App.	48
Figura 6.30	Interface <i>Docker Desktop</i>	48
Figura 6.31	Instalador TeusControle UI.....	49

LISTA DE TABELAS

Tabela 3.1	Ferramentas para gestão e modelagem dos artefatos produzidos.....	14
Tabela 3.2	Ferramentas utilizadas no desenvolvimento do <i>back-end</i>	14
Tabela 3.3	Ferramentas utilizadas no desenvolvimento do <i>front-end</i>	15
Tabela 5.1	Requisitos funcionais.....	19
Tabela 5.2	Requisitos não funcionais.....	20
Tabela 5.3	Requisitos de domínio.	21

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i> ou Interface de Programação de Aplicação
APP	<i>Application</i> ou Aplicação
CLI	<i>Command-Line Interface</i> ou Interface da Linha de Comando
DTO	<i>Data Transfer Object</i> ou Objeto de Transferência de Dados
ERP	<i>Enterprise Resource Planning</i> ou Sistema Integrado de Gestão Empresarial
GPC	<i>Global Product Classification</i> ou Classificação Global de Produtos
GTIN	<i>Global Trade Item Number</i> ou Número Global do Item Comercial
HTTP	<i>Hypertext Transfer Protocol</i> ou Protocolo de Transferência de Hipertexto
IDE	<i>Integrated Development Environment</i> ou Ambiente de Desenvolvimento Integrado
MVP	<i>Minimum Viable Product</i> ou Mínimo Produto Viável
NCM	Nomenclatura Comum do Mercosul
PDF	<i>Portable Document Format</i> ou formato de documento portátil
PDV	Ponto de Venda
RD	Requisito de Domínio
RF	Requisito Funcional
RNF	Requisito Não Funcional
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i> ou Linguagem de Consulta Estruturada
TI	Tecnologia da Informação
UI	<i>User Interface</i> ou Interface do Usuário
UML	<i>Unified Modeling Language</i> ou Linguagem de Modelagem Unificada

SUMÁRIO

1 INTRODUÇÃO	9
2 TRABALHOS E SOLUÇÕES RELACIONADOS.....	11
3 FUNDAMENTAÇÃO TEÓRICA	13
3.1 Sistema de Gestão de Estoque.....	13
3.2 Ferramentas.....	14
4 METODOLOGIA DE DESENVOLVIMENTO DO TEUSCONTROLE	16
5 APRESENTAÇÃO DA SOLUÇÃO TEUSCONTROLE.....	18
5.1 Requisitos Funcionais	18
5.2 Requisitos Não Funcionais	18
5.3 Requisitos de Domínio	19
5.4 Diagrama de Classe.....	20
5.5 Diagrama de Caso de Uso.....	23
5.6 Diagrama de Arquitetura.....	23
5.7 Perfis e Características do Usuário	23
6 RESULTADOS	26
6.1 TeusControle API.....	26
6.1.1 Estrutura interna.....	26
6.1.2 <i>Endpoints</i>	28
6.2 TeusControle UI.....	29
6.2.1 Estrutura interna.....	29
6.2.2 Telas	29
6.2.2.1 Tela Login	29
6.2.2.2 Menu Lateral.....	30
6.2.2.3 Tela Usuários	32
6.2.2.4 Tela Produtos	34
6.2.2.5 Tela Entradas.....	38
6.2.2.6 Tela Vendas	41
6.2.2.7 Tela Ponto de Venda.....	45
6.2.2.8 Tela Buscar Produto.....	46
6.3 Gerenciamento de Configurações.....	47
6.4 Resultados e <i>Feedbacks</i> da Empresa	50
7 CONCLUSÃO	51
7.1 Recomendações para Trabalhos Futuros.....	52
REFERÊNCIAS.....	53

1 INTRODUÇÃO

A gestão de estoque, de acordo com Sebrae (2017), é um fator importante para o sucesso de um empreendimento. Afirmar que, se realizado de modo adequado, a empresa passa a conhecer melhor o seu negócio e entender a situação em que está, e assim, se faz possível evitar desperdícios, aplicar estratégias que possibilitam reduzir custos, e por consequência, aumentar seus lucros. Já Meirelles (2022) enfatiza ainda que existe uma relação positiva entre os custos com investimentos em TI por uma empresa e o crescimento dos seus lucros.

Mediante as duas questões (a gestão de estoque e investimento em TI), este trabalho visa o desenvolvimento de uma aplicação chamada de TeusControle, uma solução computacional para auxiliar empresas que não controlam seus estoques ou ainda as que não possuem sistemas informatizados para gestão de seus estoques. Desta maneira, o software desenvolvido poderá agregar valor à gestão e processos de negócio da empresa.

Como objeto de estudo e a fim de validar o software desenvolvido neste trabalho, foi utilizado a empresa Auto Peças W.R., loja de autopeças situada em Paraíso do Norte no estado do Paraná. Anteriormente ao desenvolvimento deste trabalho, a empresa não recorria a sistemas de informação e, neste contexto, o desenvolvimento do produto se fez necessário para atender às principais necessidades levantadas, como atividades relacionadas a gerenciamento do estoque de produtos e controle de movimentações, tais como entrada e saída.

Sendo assim, o presente trabalho visa documentar o estudo e desenvolvimento do sistema TeusControle. E, para alcançar este objetivo, foram estabelecidos os seguintes objetivos específicos:

- Averiguar e compreender as necessidades do domínio de funcionamento de lojas de autopeças de pequeno porte;
- Pesquisar e analisar abordagens ou soluções tecnológicas similares existentes;
- Realizar a gestão dos requisitos e a modelagem da solução computacional;
- Estudar as ferramentas computacionais necessárias para a construção do sistema;
- Implementar um sistema viável para o uso em uma loja de autopeças, respeitando a delimitação definida;
- Testar e validar a implementação do sistema junto à empresa;
- Documentar o desenvolvimento e os resultados obtidos;

O presente trabalho está organizado em sete seções. A Sessão 2 apresenta três produtos de software relacionados ao tema atual a fim de entender qual a situação externa do mercado de soluções nesta área de aplicação. Na Seção 3, como embasamento para este trabalho é abordado o que consiste um sistema de gestão de estoque, e as ferramentas utilizadas no desenvolvimento do produto. Posteriormente, na Seção 4, é descrito as etapas que foram necessárias para o desenvolvimento da solução. Já nas Seções 5 e 6, é demonstrado a solução TeusControle e os resultados obtidos a partir do desenvolvimento deste trabalho. Por fim, na Seção 7, é apresentado um desfecho dos resultados a fim de validar que objetivos foram realizados e resultados alcançados, além de apontar sugestões para trabalhos futuros.

2 TRABALHOS E SOLUÇÕES RELACIONADOS

Nesta Seção são expostas algumas soluções no mercado que estão em um domínio semelhante ao do sistema proposto neste trabalho. Para a busca de soluções relacionadas foi utilizado a plataforma “Capterra”, uma plataforma que tem por objetivo dar assistência para pessoas e empresas encontrarem soluções de software adequadas às suas necessidades, oferecendo recurso de pesquisa de software para avaliação e seleção de soluções computacionais (CAPTERRA, 2022).

Neste contexto, utilizando das palavras-chave “Sistemas” de Controle de Estoque” e “Software de PDV para Varejo”, ambos aplicando os filtros “Brasil”, “português”, “versão gratuita” e “gestão de estoque” foram encontrados 5 e 4 resultados respectivamente. Um dos produtos esteve presente em ambas as pesquisas, totalizando assim, 8 diferentes produtos.

Como critério de seleção, foram descartados os produtos que não possuíam descrição em português na plataforma Capterra e que menos se encaixavam com o contexto pesquisado. Portanto, foram analisados e selecionados três produtos: Loyverse POS, MarketUP e Odoo.

Segundo Capterra (2022), Loyverse POS se trata de um sistema Ponto de Venda (também conhecido por PDV) ideal para bares, cafés, vendedores de comida de rua, padarias, varejo de alimentos, bicicletarias, cafeterias, saúde e beleza. Além destes empreendimentos mencionados anteriormente, o site Universe (2022) afirma atender também lojas de roupas e varejos. Conta com mais de 10 mil downloads no seu site e mais de 1 milhão de downloads na PlayStore¹, a loja de aplicativos da empresa Google. Na plataforma de buscas de soluções de software Capterra, Loyverse POS possui 183 (cento e oitenta e três) avaliações de usuários, totalizando uma média de 4,8 de 5 pontos possíveis. O sistema está disponível para acesso na Nuvem, IOS e Android.

Com relação ao software MarketUP, ainda de acordo com Capterra (2022), esta ferramenta consiste em um sistema integrado de gestão empresarial (ou ERP) totalmente gratuito, voltado para atender pequenas e médias empresas brasileiras. Segundo o site MarketUP (2022), a plataforma já foi utilizada por mais de 1 milhão de empresas e é um software recomendado pelo Sebrae, o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas. Além disso, conta com mais de 100 mil downloads na PlayStore². No site Capterra (2022) este aplicativo possui apenas 3 (três) avaliações de usuários, totalizando

¹ <https://play.google.com/store/apps/details?id=com.loyverse.sale>

² <https://play.google.com/store/apps/details?id=com.marketup.mobile>

uma média de 5 de 5 pontos possíveis. Além disso, a ferramenta encontra-se disponível para acesso na Nuvem, IOS e Android.

Por fim, o software Odoo, na plataforma Capterra (2022) é apresentado como um software *open source*, ou seja, de código aberto, que pode auxiliar pessoas e empresas de qualquer tamanho, que necessitem desde um blog até recursos empresariais completos. Segundo o site Odoo (2022), cerca de 7 milhões de usuários utilizam do produto para alavancar seus empreendimentos, fazendo uso de muitos dos seus 64 diferentes módulos. Na PlayStore³, este software também conta com mais de 100 mil downloads. Além disso, no site Capterra (2022) conta com 454 (quatrocentos e cinquenta e quatro) avaliações, alcançando a média de 4,1 de 5 pontos possíveis. O software se faz disponível para ser utilizado na Nuvem, Windows, MacOS, IOS e Android.

Apesar dos sistemas apresentados já possuírem um espaço conquistado no mercado, e uma boa cobertura de funcionalidades, o sistema TeusControle surge como uma necessidade apontada pela empresa Auto Peças W.R. que demonstrou não ter encontrado uma solução customizável que atendesse a sua gestão de estoque.

³<https://play.google.com/store/apps/details?id=com.odoo.mobile>

3 FUNDAMENTAÇÃO TEÓRICA

Com o intuito de compreender o que consiste e qual é o papel de um sistema de gestão de estoque, é abordado nesta Seção o conceito de estoque, gestão de estoque e a importância de um sistema de gestão para a atividade. Além disso, também são apresentadas as ferramentas utilizadas durante o desenvolvimento deste trabalho.

3.1 Sistema de Gestão de Estoque

De acordo com Silva e Madeira (2004), o estoque diz respeito aos bens e insumos armazenados por uma organização com o objetivo de suprir demandas futuras e garantir segurança operacional, capaz de alimentar a produção sem interrupção, incrementando a eficiência no processo produtivo; como também de apoiar as vendas, entregando um melhor serviço ao cliente, agregando maior valor ao produto, com um atendimento mais personalizado e com maior agilidade.

Silva e Madeira (2004) também afirmam que o controle ou gestão de estoque envolve atividades como procedimentos e estratégias que têm por finalidade proporcionar a qualidade, quantidade e tempo certo de cada produto em estoque, ou seja, uma administração eficaz do estoque, buscando garantir a satisfação pronta do cliente e uma melhor otimização dos custos. Para os autores, quanto mais estoque, maiores são os custos com aluguéis e armazenagem, pessoas e equipamentos, maiores chances de obsolescência, e maiores chances de furtos, roubos e perdas.

Levando em conta o cenário atual, empresas que possuem mais informações disponíveis a respeito do seu negócio, estão em vantagem no mercado (BEZERRA, 2015). Para a gestão do estoque, esta vantagem implica na redução do desperdício, redução no volume em estoque e consequentemente na redução de custos, além de uma melhoria significativa no serviço prestado (PINTO et al., 2013).

Para possibilitar uma vantagem, um sistema de informação se faz necessário. Um sistema de informação tem como propósito fazer proveito do caráter estratégico da informação a fim de entregar valor, podendo apoiar e aprimorar a produtividade nos processos da empresa ou organização, graças à combinação de “pessoas, dados, processos, interfaces, redes de comunicação e tecnologia” que compõem um sistema da informação (BEZERRA, 2015).

3.2 Ferramentas

Nesta subseção são demonstradas as ferramentas utilizadas para o desenvolvimento deste trabalho. Entende-se por ferramentas as tecnologias, *frameworks*, softwares e serviços que apoiam a realização das atividades da implementação de um sistema.

Na Tabela 3.1 estão discriminadas as ferramentas utilizadas para gestão e modelagem da solução, uma breve descrição e o propósito para o seu uso.

Tabela 3.1: Ferramentas para gestão e modelagem dos artefatos produzidos.

<i>Ferramenta</i>	<i>Descrição</i>	<i>Propósito</i>
Jira Software	Software da família de produtos da empresa Atlassian criado com o objetivo de apoiar o gerenciamento de todo tipo de trabalho, em especial para equipes ágeis (JIRA, 2022).	Gestão dos requisitos
Lucidchart	Software online para criação de diagramas e recursos de comunicação visual (LUCIDCHART, 2022b).	Modelagem da solução
Google Drive	Serviço de armazenamento de arquivos na nuvem da empresa Google (GOOGLE, 2022b).	Armazenamento dos artefatos
Git	Sistema de código aberto para controle de versionamento de código distribuído, criado para lidar com projetos de qualquer dimensão (GIT, 2022).	Armazenamento e versionamento do código
GitHub	Plataforma colaborativa de desenvolvimento, construída com base no Git (GITHUB, 2022).	Armazenamento e versionamento do código

Fonte: Elaborado pelo autor

Na Tabela 3.2 estão indicadas as ferramentas utilizadas para a construção da solução TeusControle API, uma breve descrição e o propósito para o seu uso.

Tabela 3.2: Ferramentas utilizadas no desenvolvimento do *back-end*.

<i>Ferramenta</i>	<i>Descrição</i>	<i>Propósito</i>
Visual Studio Community	Ambiente de desenvolvimento integrado (IDE) gratuito da Microsoft (STUDIO, 2022b).	Desenvolvimento do código fonte da aplicação
MySQL Workbench	Ferramenta visual para modelagem, arquitetura, desenvolvimento de SQL para base de dados MySQL (MYSQL, 2022).	Acesso a base de dados
C#	Linguagem de programação moderna, orientada a objeto e de tipagem forte (MICROSOFT, 2022b).	Desenvolvimento da aplicação
.NetCore	Plataforma de desenvolvedor gratuita, multiplataforma de código aberto para diferentes tipos de aplicações (MICROSOFT, 2022a).	Desenvolvimento da aplicação
MySQL	Sistema de gerenciamento de banco de dados (SGBD) relacional mais popular de código aberto no mundo (ORACLE, 2022).	SGBD utilizado
Docker	Plataforma construída para auxiliar desenvolvedores a construir, compartilhar e executar aplicações (DOCKER, 2022).	Implantação da aplicação

Fonte: Elaborado pelo autor

Na Tabela 3.3 estão discriminadas as ferramentas utilizadas para a construção da

solução TeusControle UI, uma breve descrição e o propósito para o seu uso.

Tabela 3.3: Ferramentas utilizadas no desenvolvimento do *front-end*.

<i>Ferramenta</i>	<i>Descrição</i>	<i>Propósito</i>
Android Studio	Ambiente de desenvolvimento integrado (IDE) da Google, baseado no IntelliJ IDEA (STUDIO, 2022a).	Desenvolvimento do código fonte da aplicação
Dart	Linguagem de programação criada pela empresa Google, baseado em classes, com herança única e orientada a objetos (DART, 2022).	Desenvolvimento da aplicação
Flutter	<i>Framework</i> de código aberto criado pela empresa Google para desenvolvimento de aplicações <i>client side</i> multiplataforma (FLUTTER, 2022).	Desenvolvimento da aplicação
Inno Setup	Um instalador desenvolvido por Jordan Russell e Martijn Laan para programas do sistema operacional Windows (RUSSELL; LAAN, 2022).	Criação do instalador

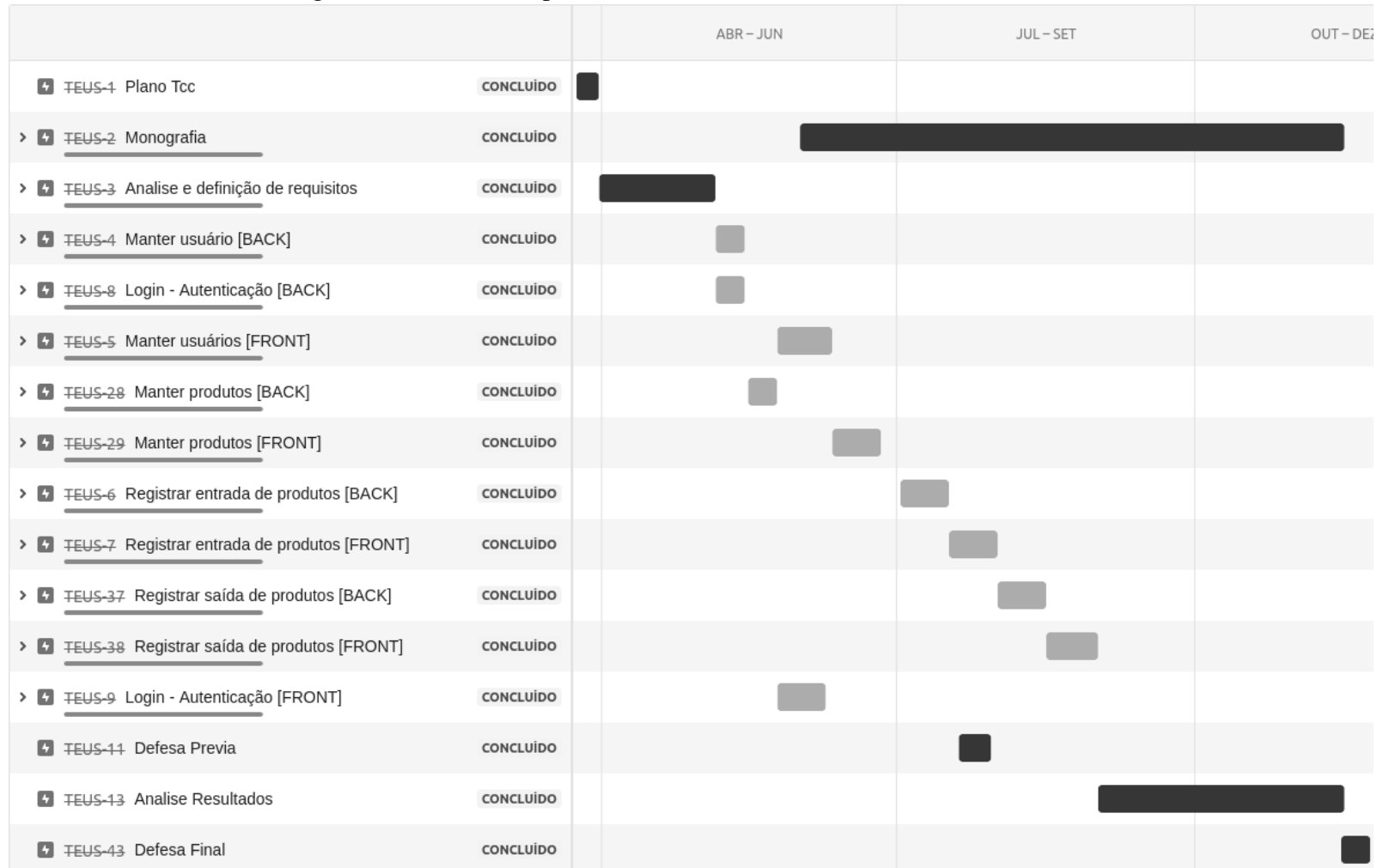
Fonte: Elaborado pelo autor

4 METODOLOGIA DE DESENVOLVIMENTO DO TEUSCONTROLE

De acordo com Sommerville (2018), “um processo de software é um conjunto de atividades relacionadas que levam a produção de um sistema de software”. Afirmar ainda que todo processo de software deve compreender atividades relacionadas a especificação, desenvolvimento, validação e evolução, atividades estas, fundamentais da engenharia de software. Assim, para o desenvolvimento deste trabalho, se fez necessário passar pelas seguintes etapas:

- **Especificação:** durante esta etapa, se deu início a concepção do trabalho. Iniciando com conversas com o potencial cliente (empresa estudada), levantando suas necessidades, que se traduziram nos requisitos do sistema (funcionalidades e limitações) e diagramas, a fim de apoiar as próximas etapas do processo. Além disso, foi elaborado um cronograma de entrega das atividades. Na Figura 4.1, no canto esquerdo, são apresentadas todas as atividades, identificadas pelo prefixo “TEUS”, seguido por um número, também pelo seu título e status, onde é possível observar que todos estão “concluído”. Ao lado, é apresentado o intervalo de tempo dedicado a cada atividade, onde em preto é identificadas as atividades associadas a escrita e especificação deste trabalho e em cinza as atividades relacionadas ao desenvolvimento da solução.
- **Desenvolvimento:** nesta etapa deu-se início ao desenvolvimento dos componentes do sistema. Iniciou-se com a escolha das tecnologias, seguido da estruturação dos projetos de *back-end* e *front-end*, buscando o reaproveitamento de código e preparando para sustentar futuras funcionalidades. Após a execução das atividades, seguiu a sequência tal qual apresentada na Figura 4.1.
- **Validação:** nesta fase do projeto, buscou-se validar junto ao proprietário da empresa se o software desenvolvido atende às suas expectativas e necessidades. Pretendia-se, após um período de uso, recolher pontos positivos, pontos negativos, sugestões de melhorias e impactos na rotina da empresa.
- **Evolução:** neste estágio é onde as mudanças são planejadas e executadas, visando atender novas necessidades do cliente. Este trabalho não abordará esta etapa, pois, tem como foco apresentar o processo desde a concepção, até a primeira entrega do produto. Entretanto, será apresentado algumas recomendações de trabalhos futuros.

Figura 4.1: Atividades para o desenvolvimento do software TeusControle.



Fonte: Elaborado pelo autor

5 APRESENTAÇÃO DA SOLUÇÃO TEUSCONTROLE

Neste capítulo será apresentada a proposta da solução computacional por meio dos requisitos e diagramas levantados na etapa de especificação do sistema. De acordo com Sommerville (2018), “requisitos de um sistema são as descrições dos serviços que o sistema deve prestar e as restrições à sua operação”. Já os diagramas são representações gráficas que tem como objetivo demonstrar o esquemático simplificado ou uma abstração geral sobre um assunto.

Neste contexto, para a modelagem de sistemas, faz-se uso da Linguagem de Modelagem Unificada também conhecida pelo acrônimo UML, um padrão de modelagem visual comum para descrever comportamentos e estruturas com uma semântica e sintaxe rica (GUEDES, 2018).

5.1 Requisitos Funcionais

Nesta Subseção, serão apresentados os requisitos funcionais do sistema. Os requisitos funcionais dizem respeito às funcionalidades que o sistema deve fornecer e como deve se comportar em determinadas situações (SOMMERVILLE, 2018).

A Tabela 5.1 descreve os requisitos funcionais. Nesta tabela tem-se o ID (código identificador do requisito), seguido pelo seu nome, descrição e nível de prioridade. Estes detalhes são importantes para manter a rastreabilidade e controle de requisitos durante o processo de desenvolvimento do sistema.

5.2 Requisitos Não Funcionais

Os requisitos não funcionais dizem respeito às limitações aplicadas às funcionalidades, e geralmente se aplicam ao sistema como um todo (SOMMERVILLE, 2018).

A Tabela 5.2 descreve os requisitos não funcionais. Nesta tabela tem-se o ID (código identificador do requisito), seguido pelo seu nome, descrição e nível de prioridade. Estes detalhes são importantes para manter a rastreabilidade e controle de requisitos durante o processo de desenvolvimento do sistema.

Tabela 5.1: Requisitos funcionais.

<i>ID</i>	<i>Nome</i>	<i>Descrição</i>	<i>Prioridade</i>
RF-001	Cadastro de usuários	O sistema deve permitir o cadastro de usuários que terão acesso ao sistema.	Alta
RF-002	Listagem de usuários	O sistema deve permitir a listagem de usuários que terão acesso ao sistema.	Alta
RF-003	Alteração de usuários	O sistema deve permitir a alteração de usuários que terão acesso ao sistema.	Alta
RF-004	Exclusão de usuários	O sistema deve permitir a exclusão de usuários que terão acesso ao sistema.	Alta
RF-005	Cadastro de produtos	O sistema deve permitir o cadastro de produtos que serão mantidos em estoque.	Alta
RF-006	Listagem de produtos	O sistema deve permitir a listagem de produtos que serão mantidos em estoque.	Alta
RF-007	Alteração de produtos	O sistema deve permitir a alteração de produtos que serão mantidos em estoque.	Alta
RF-008	Exclusão de produtos	O sistema deve permitir a exclusão de produtos que serão mantidos em estoque.	Alta
RF-009	Registrar entrada de produtos	O sistema deve permitir realizar a movimentação de entrada de produtos a serem repostos e incrementar sua quantidade em estoque.	Alta
RF-010	Registrar venda de produtos	O sistema deve permitir realizar a movimentação de venda de produtos e decrementar sua quantidade em estoque.	Alta
RF-010	Autenticação e autorização	O sistema deve permitir a autenticação e autorização dos usuários previamente cadastrados.	Alta
RF-011	Relatório de usuários	O sistema deve possibilitar a emissão de relatório de todos os usuários cadastrados.	Média
RF-012	Relatório de produtos	O sistema deve possibilitar a emissão de relatório de todos os produtos cadastrados e sua respectiva quantidade em estoque.	Média
RF-013	Relatório de entradas de produtos	O sistema deve possibilitar que seja emitido relatório de movimentações de entrada de produtos.	Média
RF-014	Relatório de vendas	O sistema deve possibilitar que seja emitido relatório de movimentações de venda.	Média

Fonte: Elaborado pelo autor

5.3 Requisitos de Domínio

Os requisitos de domínio ou regra de negócios são requisitos que refletem a maneira que uma empresa executa seu negócio. De acordo com Leite e Leonardi (1998) compreendem as políticas, os procedimentos, as leis ou convenções que são implementados em um sistema de software a fim de atender o domínio do negócio.

A Tabela 5.3 descreve os requisitos de domínio levantados. Nesta tabela tem-se o ID (identificador do requisito), seguido pelo seu nome, descrição e nível de prioridade. Estes detalhes são importantes para manter a rastreabilidade e controle de requisitos durante o processo de desenvolvimento do sistema.

Tabela 5.2: Requisitos não funcionais.

<i>ID</i>	<i>Nome</i>	<i>Descrição</i>	<i>Prioridade</i>
RNF-001	Integração API de catálogo de produtos	O sistema deve integrar com a API de catálogo de produtos da Cosmos (2022) para busca de informações.	Alta
RNF-002	Ambiente de execução	O sistema deve rodar em ambiente local, no sistema operacional <i>Windows 10</i> e versões mais recentes.	Alta
RNF-003	Linguagens do <i>back-end</i>	O <i>back-end</i> deve ser desenvolvido C#.	Alta
RNF-004	Frameworks do <i>back-end</i>	O <i>back-end</i> deve ser desenvolvido com o <i>framework .net core</i> .	Alta
RNF-005	Linguagens do <i>front-end</i>	O <i>front-end</i> deve ser em Dart.	Alta
RNF-006	Frameworks do <i>front-end</i>	O <i>front-end</i> deve ser desenvolvido com o <i>framework Flutter</i> .	Alta
RNF-007	Interação com banco de dados	O sistema deve se comunicar com o MySQL.	Alta
RNF-008	Exclusão de registros	Todas as operações de exclusão deverão ser realizadas de forma lógica, ou seja, não serão apagados fisicamente, mas não serão apresentadas em tela.	Alta

Fonte: Elaborado pelo autor

5.4 Diagrama de Classe

O diagrama de classes é um dos mais utilizados da UML e é responsável por trazer uma visão estratégica das classes que compõem o sistema e tem como objetivo representar seus atributos, métodos e como estas se relacionam (GUEDES, 2018).

Na fase de especificação, foi construído um diagrama de classe conceitual que se preocupa em dar foco às informações que faz necessário para o sistema, neste contexto representado pelos atributos e relacionamentos.

Desta forma, na Figura 5.1 temos representados os atributos e os relacionamentos das classes *User*, *Product*, *Disposal*, *ProductDisposal*, *Entry* e *ProductEntry*.

A classe *User* é utilizada para representar os usuários que terão acesso ao sistema e possibilitar o controle de acesso das operações internas. Além disso, para possibilitar a rastreabilidade das operações realizadas, esta classe possui um auto-relacionamento e também relacionamentos com todas as demais classes, ficando salvo assim o responsável pela inserção de cada um dos registros.

A classe *Product* representa os itens que fazem parte do estoque gerenciado, ou seja, os produtos. Um item terá seu código de barras armazenado no atributo “*gtin*” (acrônimo em inglês do termo *Global Trade Item Number*, que significa Número Global do Item Comercial), quantidade que consta em estoque no atributo “*inStock*”, seu preço de saída representados por “*unitPrice*”, e informações para sua classificação nos campos

Tabela 5.3: Requisitos de domínio.

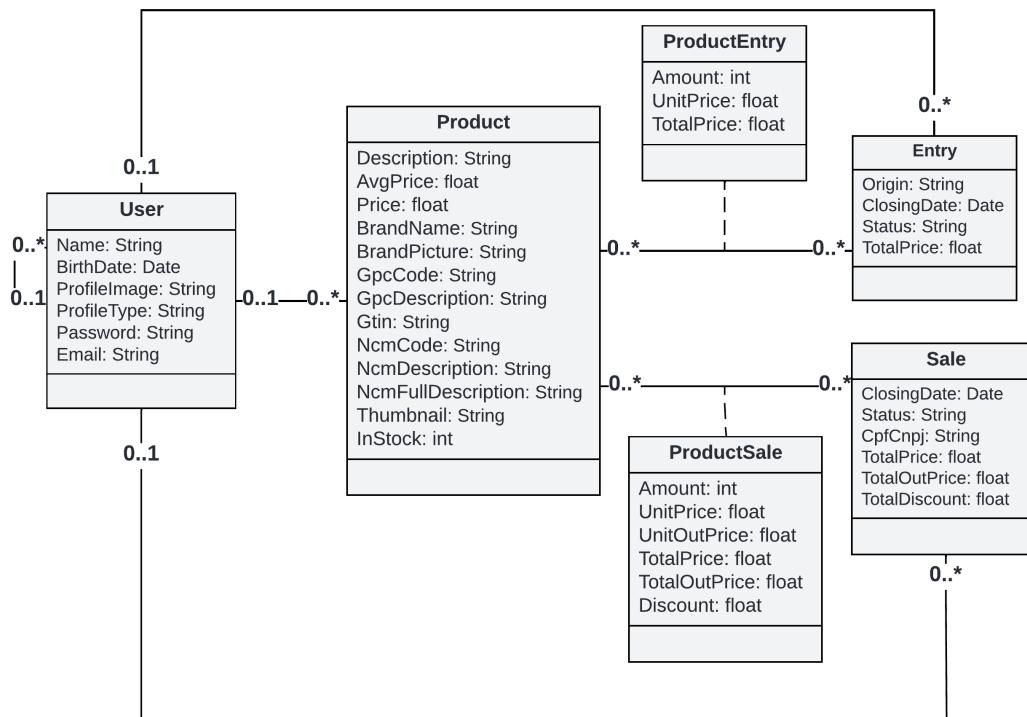
<i>ID</i>	<i>Nome</i>	<i>Descrição</i>	<i>Prioridade</i>
RD-001	Permissão para cadastro de entradas	Apenas usuários autenticados com papel “Administrador” poderão realizar o cadastro de entrada de produtos no sistema.	Alta
RD-002	Permissão para cadastro de vendas	Apenas usuários autenticados com papel “administrador” ou “vendedor” poderão realizar o cadastro de vendas de produtos no sistema.	Alta
RD-003	Movimentação de saída do estoque	A quantidade em estoque do produto só será atualizada e decrementada após o registro de venda estar com <i>status</i> “fechado”.	Alta
RD-004	Movimentação de entrada do estoque	A quantidade em estoque do produto só será atualizada e incrementada após o registro de entrada de produtos estar com <i>status</i> “fechado”.	Alta
RD-005	Venda com <i>status</i> fechada	Uma vez que um registro de venda estar com <i>status</i> “fechado”, o mesmo não poderá sofrer alterações.	Alta
RD-006	Entrada com <i>status</i> fechada	Uma vez que um registro de entrada estar com <i>status</i> “fechado”, o mesmo não poderá sofrer alterações.	Alta
RD-007	Fechar uma movimentação de venda	Apenas usuários com papel “administrador” poderão realizar o cadastro de vendas de produtos e mantê-las com <i>status</i> “aberto”. Vendas realizadas por usuários com papel “vendedor” serão fechadas assim que a venda for confirmada.	Alta
RD-008	Aplicar desconto a produtos	Apenas usuários com papel “administrador” poderão aplicar descontos a produtos ao realizar uma venda.	Alta
RD-009	Exclusão de entradas	Apenas entradas com <i>status</i> diferente de “fechado” poderão ser excluídas.	Alta
RD-010	Exclusão de vendas	Apenas vendas com <i>status</i> diferente de “fechado” poderão ser excluídas.	Alta
RD-011	Exclusão de usuários	Apenas usuários com papel “administrador” poderão executar a exclusão de usuários. Sendo que, o usuário autenticado não poderá realizar a exclusão de seu próprio registro.	Alta

Fonte: Elaborado pelo autor

“*gpc*” (do inglês *Global Product Classification*, que significa Classificação Global de Produtos) e “*ncm*” (acrônimo do termo Nomenclatura Comum do Mercosul) e por fim, atributos para representar características gerais tais como marca, descrição e imagens. A classe *Product* se relaciona com as classes *Sale* e *Entry* através das classes associativas *ProductEntry* e *ProductSale*.

A classe *Sale* representa uma movimentação de venda. O registro de uma saída só será efetivado após finalizado, e para este controle o atributo “*status*” se faz necessário. Além disso, é possível a inclusão do documento para identificação do cliente, representado pelo campo “*document*”. Para possibilitar o relacionamento muitos para muito que

Figura 5.1: Diagrama de Classes.



Fonte: Elaborado pelo autor

uma saída exige (uma venda possui muitos produtos e um produto pode estar em muitas vendas), se faz necessário a criação de uma classe associativa para este relacionamento, representado pela classe “*ProductSale*” já mencionado anteriormente. Em “*ProductSale*”, além dos identificadores da venda e produto, as informações do valor de venda da unidade do item é representado pelo atributo “*unitPrice*”, a quantidade que está saindo por “*amount*”, o valor total por “*totalPrice*”, o valor do desconto aplicado ao item em “*discount*”, o valor total dos produtos aplicado o desconto em “*totalOutPrice*”.

A classe *Entry* é utilizada para representar uma movimentação de entrada, ou seja, a reposição do estoque. Análoga a classe *Sale*, só será atualizado o estoque após a sua finalização, desta forma também possuindo um atributo “*status*”. O atributo “*origin*” foi criado para registrar a procedência desta movimentação. Além disso, para possibilitar o relacionamento muitos para muito que uma entrada exige (uma entrada possui muitos produtos e um produto pode estar em muitas entradas), se faz necessário a criação de uma classe associativa para este relacionamento, representado pela classe “*ProductEntry*” já mencionado anteriormente. Em “*ProductEntry*”, além dos identificadores da entrada e produto, as informações do valor pago por unidade do item é representado pelo atributo “*unitPrice*”, a quantidade adquirida por “*amount*”, e o valor total por “*totalPrice*”.

5.5 Diagrama de Caso de Uso

O diagrama de caso de uso é mais um diagrama da UML. Utiliza uma linguagem simples e de fácil compreensão do comportamento do sistema. Identifica graficamente os atores que interagem com o sistema, ou seja, usuários ou outros sistemas, e os casos de uso, que representa as funcionalidades que o sistema oferece aos atores, através de uma visão geral e informal. (GUEDES, 2018).

Como mostra a Figura 5.2, temos como atores no sistema, o usuário vendedor e o usuário administrador. O usuário vendedor terá permissão para efetuar registros de saída de produtos do estoque, ou seja, realizar uma venda, onde poderá ou não ser emitido um recibo. Além disso, poderá realizar consultas de informações dos produtos cadastrados. O usuário administrador terá todas as permissões que o usuário vendedor possui, e demais operações gerenciais, sendo elas: manter registros de produtos, manter registros de usuários, registrar entradas, e poderá gerar relatórios simplificados destes registros. Entende-se como “manter”, operações relacionadas a cadastro, edição, exclusão e consulta.

5.6 Diagrama de Arquitetura

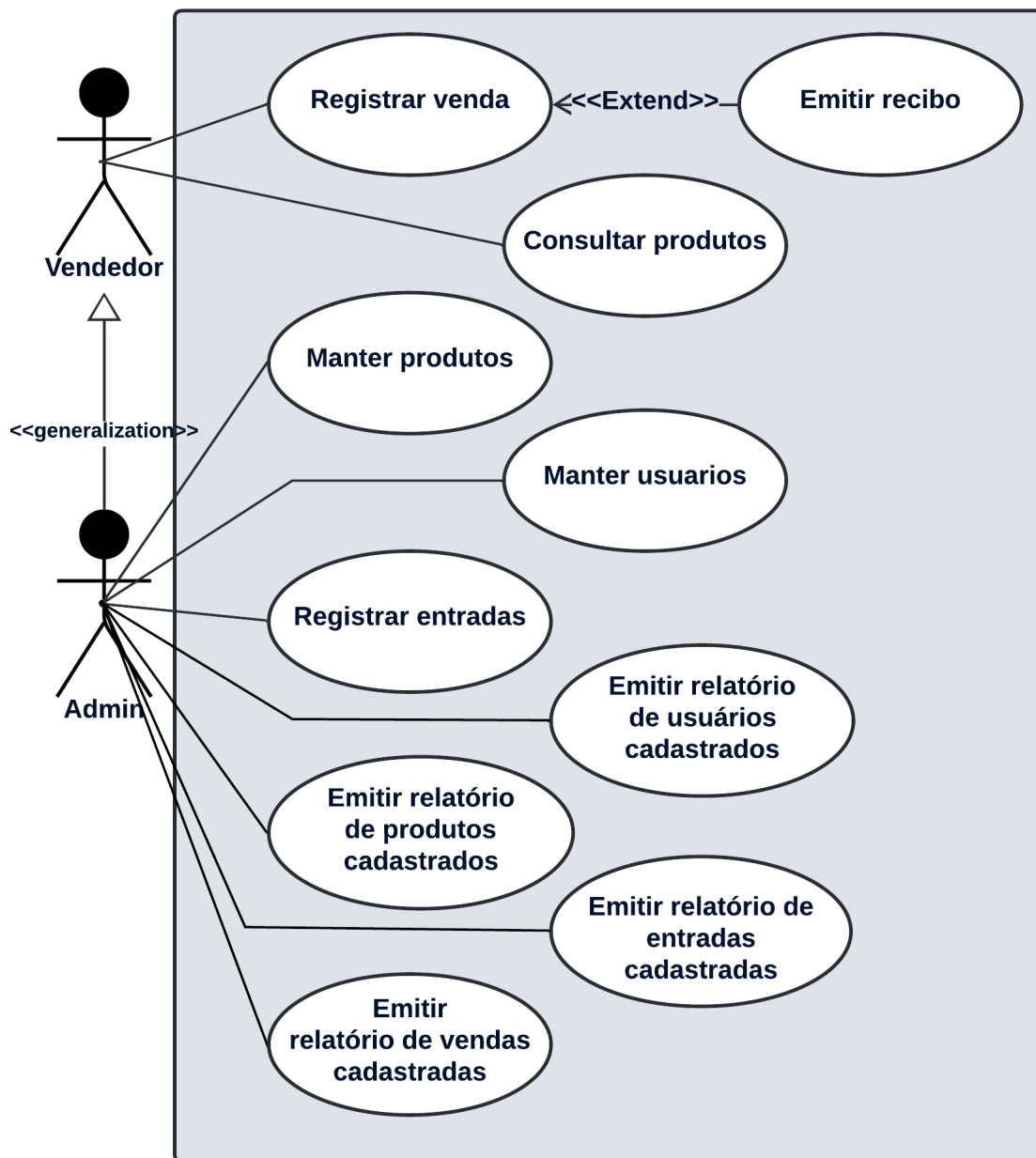
Segundo Lucidchart (2022a), o diagrama de arquitetura é uma representação gráfica que mostra como a implementação física dos componentes de software estão estruturados e associados, além de suas limitações e limites entre eles.

Na Figura 5.3, podemos observar como os diferentes elementos do sistema interagem entre si. Desta forma, para este trabalho, temos que o ator usuário interage com uma interface gráfica construída com Flutter que por sua vez consome um serviço externo de catálogo de produtos e também da *REST API* construído com o .NetCore e, por sua vez, faz uso de um banco de dados MySQL para a persistência dos dados. Além disso, fica evidente que todos os componentes estarão hospedados em um mesmo ambiente.

5.7 Perfis e Características do Usuário

O sistema é voltado para pequenas e médias empresas, que necessitam de um gerenciamento informatizado de estoque e de vendas. Desta forma, tem como principal

Figura 5.2: Diagrama de Caso de Uso.

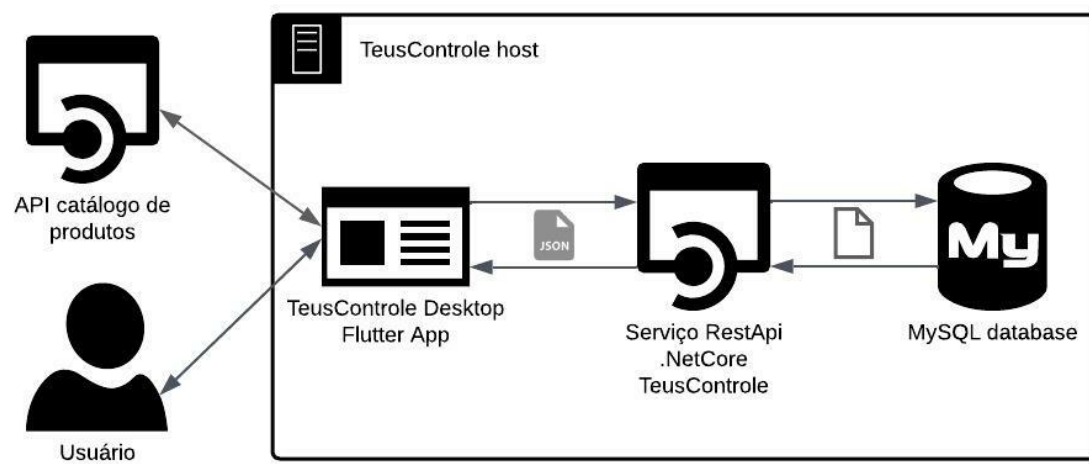


Fonte: Elaborado pelo autor

público pequenos e médios empreendedores e funcionários que atuam no comércio e não utilizam nenhuma ferramenta informatizada para esta função.

No sistema existem dois diferentes níveis de acesso: administrador e vendedor. Na Seção 5.5 é apresentada as responsabilidades de cada tipo de acesso, representado pelos atores.

Figura 5.3: Diagrama de Arquitetura.



Fonte: Elaborado pelo autor

6 RESULTADOS

Nesta seção é apresentado o resultado do software TeusControle, onde o produto TeusControle API e TeusControle UI serão detalhados. Além disso, é apresentado *feedbacks* recolhidos em conjunto com a empresa Auto Peças W.R., buscando validar que o sistema construído é capaz de apoiar a sua gestão de estoque.

6.1 TeusControle API

O projeto *back-end* é onde se encontra a implementação da regra de negócio da aplicação. Como *back-end* do TeusControle foi criado uma *REST API* em C# fazendo uso da plataforma *.NetCore*. Em seguida será apresentado a solução criada e como está estruturado a API do TeusControle.

6.1.1 Estrutura interna

Na Figura 6.1 é apresentado como ficou a estrutura interna da solução, onde existem cinco pastas no diretório raiz, sendo elas: *Core*, *Data*, *Manager*, *Tests* e *WebApi*. A seguir será abordado cada uma das pastas.

- **Core:** Contém as classes que representam as entidades dentro do projeto de classes *Core* e as classes que representam os *DTOs* no projeto de classes *Core.Shared*.
- **Data:** No projeto de classes *Data*, estão as estruturas que possibilitam acesso e manipulação de dados. Encontramos aqui os diretórios: *Configurations*, *Context*, *Migrations*, *Repository* e por fim *Services*.
- **Manager:** Nesta pasta encontram-se as classes que encapsulam as regras de negócio. Aqui temos os diretórios: *Implementation*, *Interfaces*, *Mapping*, *Utils* e *Validators*.
- **Tests:** Contém projetos de testes unitários para as camadas *Manager*, *Repository* e *WebApi*, além do projeto de classe *FakeData* responsável por encapsular as classes que permitem a criação de objetos populados a fim de possibilitar a execução dos testes.
- **WebApi:** Contém a classe principal do projeto que possibilita a construção de um

serviço *HTTP REST*. Além disso, possui os diretórios: *Properties*, *Configurations* e *Controllers*.

Figura 6.1: Estrutura da implementação do serviço *Rest API*.

```

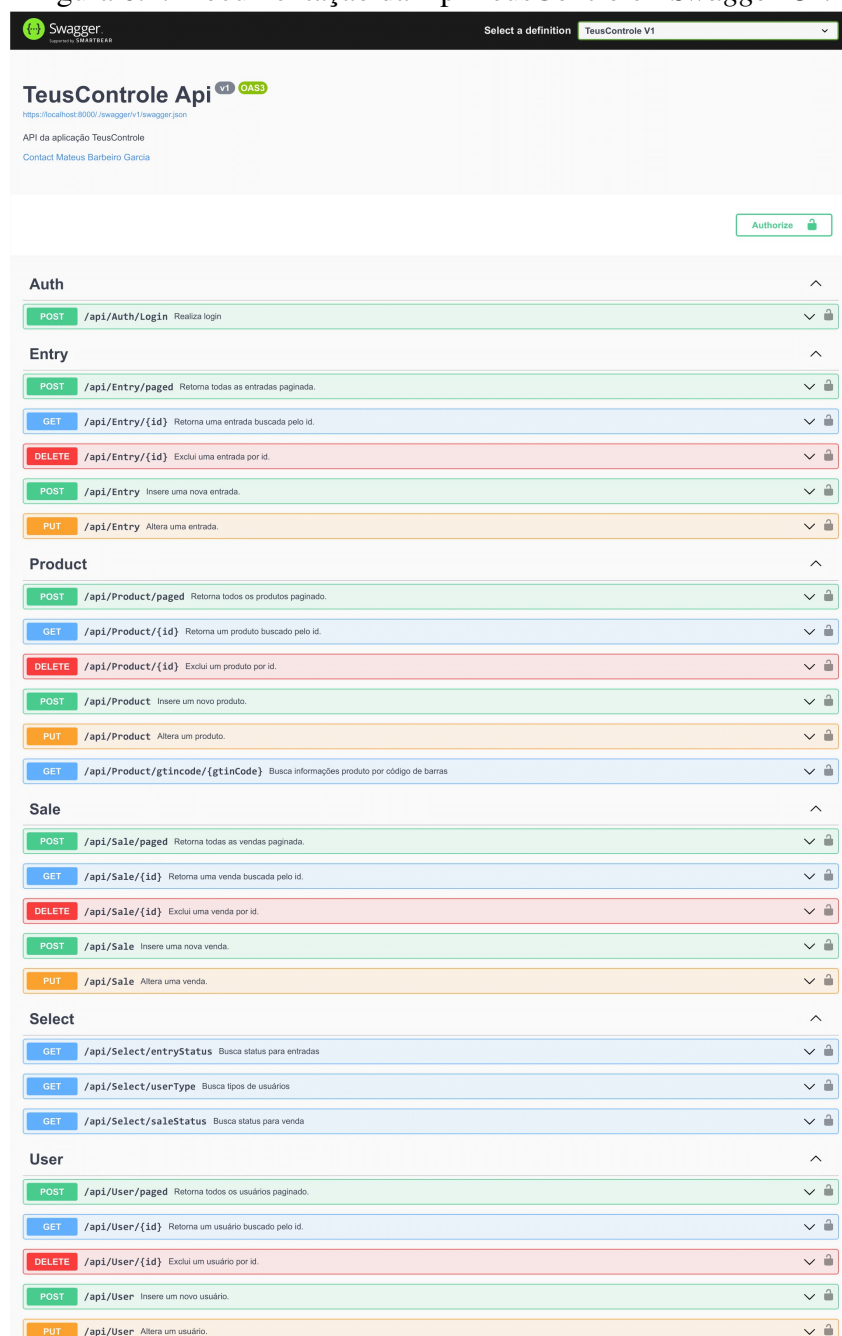
TeusControle
├─ Core
│   └─ Domain
│       └─ Base
├─ Core.Shared
│   └─ Models
│       ├── Entry
│       ├── Enums
│       ├── Product
│       ├── Request
│       ├── Responses
│       ├── Sale
│       └─ User
├─ Data
│   ├── Configurations
│   ├── Context
│   ├── Migrations
│   ├── Repository
│   │   └─ Base
│   └─ Services
├─ FakeData
│   ├── ProductData
│   └─ UserData
├─ Manager
│   ├── Implementation
│   │   └─ Base
│   ├── Interfaces
│   │   ├── Managers
│   │   │   └─ Base
│   │   ├── Repositories
│   │   │   └─ Base
│   │   └─ Services
│   ├── Mapping
│   │   ├── Entry
│   │   ├── Product
│   │   ├── Sale
│   │   └─ User
│   ├── Utils
│   └─ Validator
│       ├── Entry
│       ├── Product
│       └─ User
├─ Manager.Tests
│   └─ Managers
├─ Repository.Tests
│   └─ Repositories
├─ WebApi.Tests
└─ TeusControleApi
    ├── Configuration
    ├── Controllers
    └─ Properties
  
```

Fonte: Elaborado pelo autor

6.1.2 Endpoints

Para documentar os *endpoints*, ou seja os pontos de interação finais do sistema, foi utilizado o *Swagger*, um *framework* que provê ferramentas para visualizar, descrever, e consumir serviços de uma *API REST* (SMARTBEAR, 2022). A seguir na Figura 6.2 é apresentado o resultado.

Figura 6.2: Documentação da Api TeusControle - *Swagger UI*.



Fonte: Elaborado pelo autor

Na Figura 6.2 temos os *endpoints* para autenticação do usuário em *Auth*, opera-

ções relacionadas a entrada de produtos no estoque em *Entry*, operações relacionadas ao cadastro de produtos em *Product*, operações relacionada a movimentações de venda de produtos em *Sale*, consultas de itens de seleção em *Select*, e, por fim, operações relacionadas ao cadastro de usuários em *User*.

6.2 TeusControle UI

O projeto *front-end* é onde se encontra a implementação da interface de usuário do TeusControle. Como *front-end*, foi criado um projeto *Desktop* em *Dart*, fazendo uso do *framework Flutter* utilizando do padrão de design desenvolvida pelo Google (2022a), o Material Design.

6.2.1 Estrutura interna

Na Figura 6.3 é apresentado como ficou a estrutura interna da solução, onde existem duas pastas no diretório “lib”, sendo elas: *core* e *UI*. A seguir será abordado cada uma das pastas.

- **Core:** Contém as pastas *apis*, *models* e *services*. Nestas pastas estão os arquivos .dart que possibilitam o consumo das chamadas da *REST API* do TeusControle e classes para converter informações recebidas e a serem enviadas para a API.
- **Ui:** Contém as pastas *shared* e *views*. Nelas estão os arquivos .dart que implementam os *widgets*, telas e estilos da aplicação.

6.2.2 Telas

O Software desenvolvido possui as telas: *login*, usuários, produtos, entradas e vendas. A seguir, serão apresentadas e descritas a sua composição.

6.2.2.1 Tela Login

Para acessar o sistema, se faz necessário informar um e-mail e senha previamente cadastrado.

Figura 6.3: Estrutura da implementação do Flutter App.

```

lib
├── core
│   ├── apis
│   ├── models
│   │   ├── entry
│   │   ├── paginated
│   │   │   └── enums
│   │   ├── product
│   │   ├── sale
│   │   ├── select
│   │   └── user
│   └── services
└── ui
    ├── shared
    │   ├── styles
    │   ├── utils
    │   └── widgets
    │       ├── buttons
    │       ├── default
    │       ├── dialogs
    │       ├── drawer
    │       ├── inputs
    │       └── tables
    │           └── paginated
    └── views
        ├── entry
        │   └── widgets
        ├── home
        ├── landing
        ├── login
        │   └── widgets
        ├── point_of_sale
        │   └── widgets
        ├── product
        │   └── widgets
        ├── sale
        │   └── widgets
        └── user
            └── widgets

```

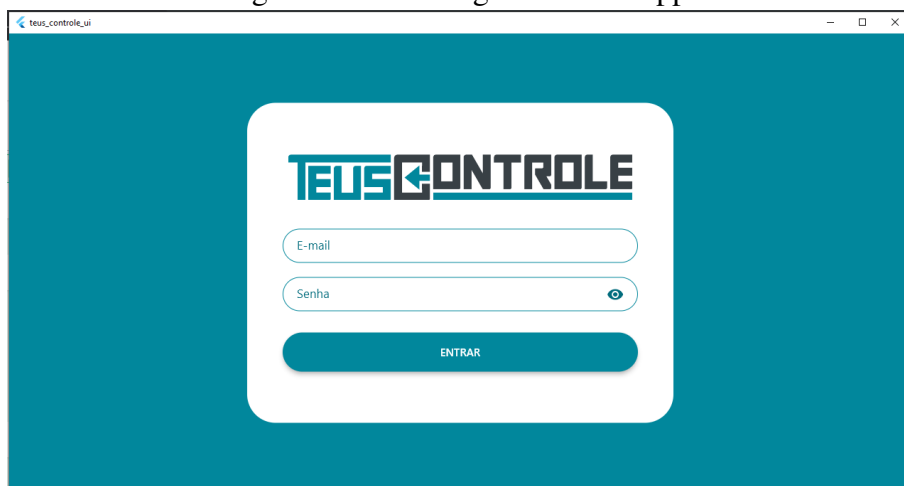
Fonte: Elaborado pelo autor

Na Figura 6.4 é exibida a tela de *login* onde o usuário irá se autenticar. Ao informar os dados e pressionar o botão “ENTRAR”, se forem informado credenciais válidas, o usuário terá acesso às demais telas e funcionalidades do sistema do qual possui permissão.

6.2.2.2 Menu Lateral

Ao estar autenticado no sistema com um usuário com o perfil “administrador”, no canto esquerdo é apresentado o menu lateral fixo com informações referente ao usuário

Figura 6.4: Tela Login - Flutter App.

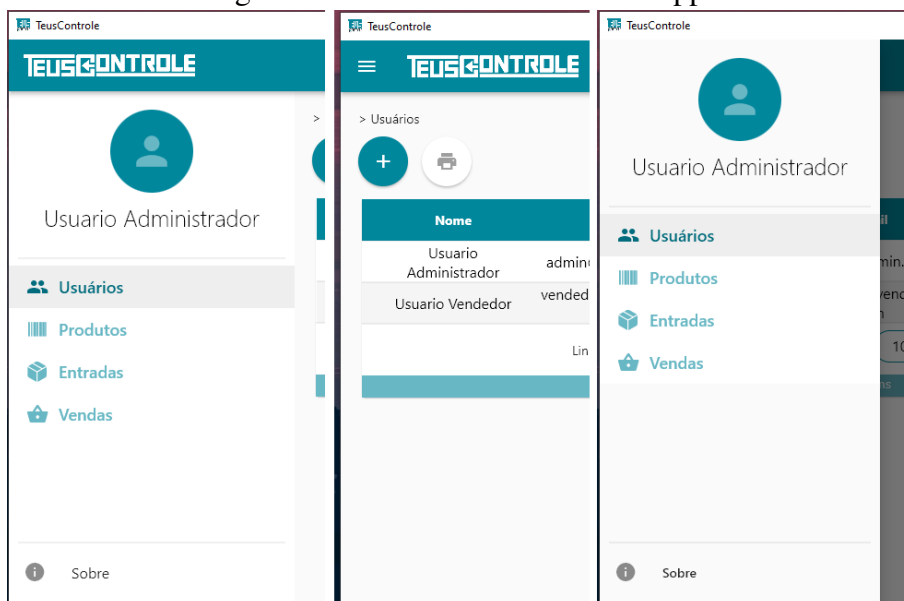


Fonte: Elaborado pelo autor

autenticado, acesso a demais telas e informações do sistema. Caso o tamanho da janela não seja suficiente para apresentar o menu e a respectiva tela, o menu lateral será escondido, podendo ser acessado através do ícone de menu ao lado da logo do TeusControle.

Na Figura 6.5 podemos observar à esquerda o menu lateral fixo, ao centro o menu ocultado e o ícone ao lado da logo, e à direita o menu lateral expandido após pressionado o botão de menu.

Figura 6.5: Menu Lateral - Flutter App.



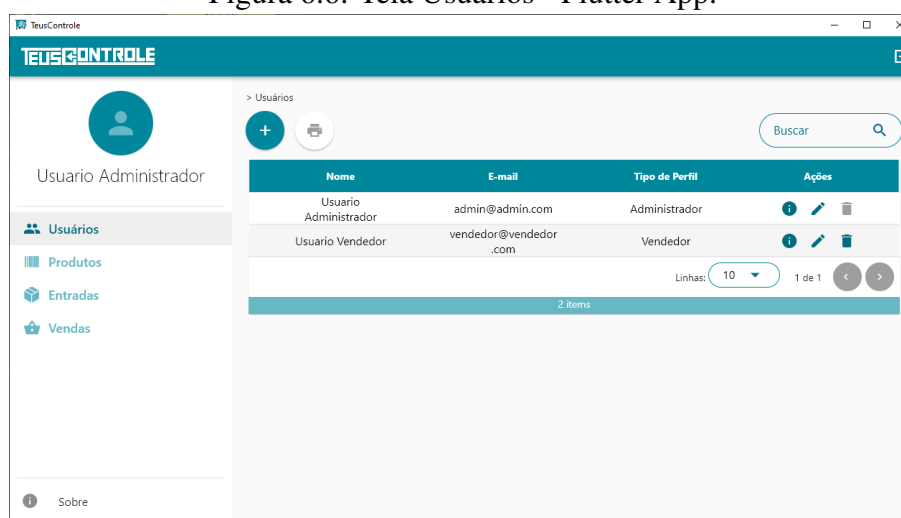
Fonte: Elaborado pelo autor

6.2.2.3 Tela Usuários

Para ter acesso ao sistema e possibilitar o controle de acesso de operações, se faz necessário manter registros de usuários. Buscando satisfazer esta necessidade, é necessário ter armazenado tais informações.

Na Figura 6.6 é exibido a tela de usuário, onde as operações de cadastro, listagem, atualização e exclusão podem ser realizadas. Além disso, é importante ressaltar que apenas o usuário com tipo de perfil “administrador” poderá acessar esta tela.

Figura 6.6: Tela Usuários - Flutter App.



Fonte: Elaborado pelo autor

6.2.2.3.1 Formulário de Cadastro de Usuários

Ao pressionar o botão com ícone de adição da tela de usuários (Figura 6.6), uma janela flutuante com título “cadastro” com os campos de entrada para registro de usuário irá aparecer.

Na Figura 6.7 é possível observar os campos de entrada “nome completo”, “nascimento”, “imagem de perfil”, “tipo de perfil”, “e-mail” e “senha”, que compõem o formulário de cadastro de usuários. Após preenchido as informações, torna-se possível salvar o registro, ao pressionar o botão “confirmar” no rodapé da janela. Também possibilita cancelar a operação de cadastro pressionando no rodapé o botão “cancelar” ou ainda no ícone com “X” ao lado do título da janela.

6.2.2.3.2 Edição do Registro de Usuários

Figura 6.7: Formulário Cadastro de Usuários - Flutter App.

Fonte: Elaborado pelo autor

Ao pressionar o botão com ícone de lápis da coluna “ações” de um registro da listagem na tela de usuários (Figura 6.6), uma janela flutuante com título “edição” com os campos de entrada irá aparecer com as informações do registro a ser editado.

Na Figura 6.8, assim como no formulário de cadastro, os mesmos campos de entrada são apresentados. Os botões de “confirmar”, “cancelar” e “X” também se fazem presente, possibilitando a confirmação ou cancelamento da operação de editar o registro.

Figura 6.8: Formulário Edição de Usuários - Flutter App.

Fonte: Elaborado pelo autor

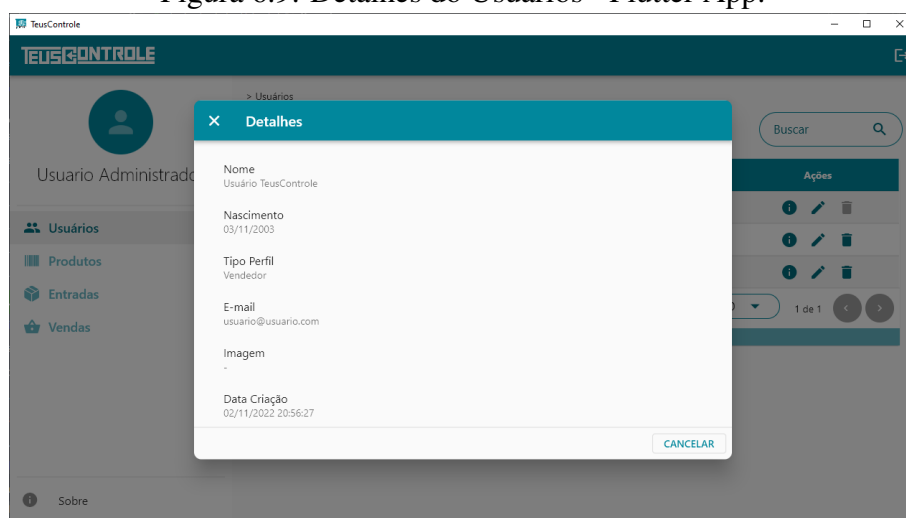
6.2.2.3.3 Detalhes do Registro de Usuários

Na tabela da listagem na tela de usuários (Figura 6.6), são apresentadas apenas al-

gumas informações. Para visualizar todas as informações de determinado registro, existe o botão com ícone “i” da coluna “ações” para todos os registros.

A Figura 6.9 mostra a janela com título “detalhes” com as informações do respectivo registro. Na janela são apresentadas todas as informações que foram enviadas no formulário de cadastro (com exceção do campo “senha”), além da data de criação e de última alteração do registro. Algumas informações estão ocultas na Figura 6.9 pois ficaram fora da área visível da área de rolagem.

Figura 6.9: Detalhes do Usuários - Flutter App.



Fonte: Elaborado pelo autor

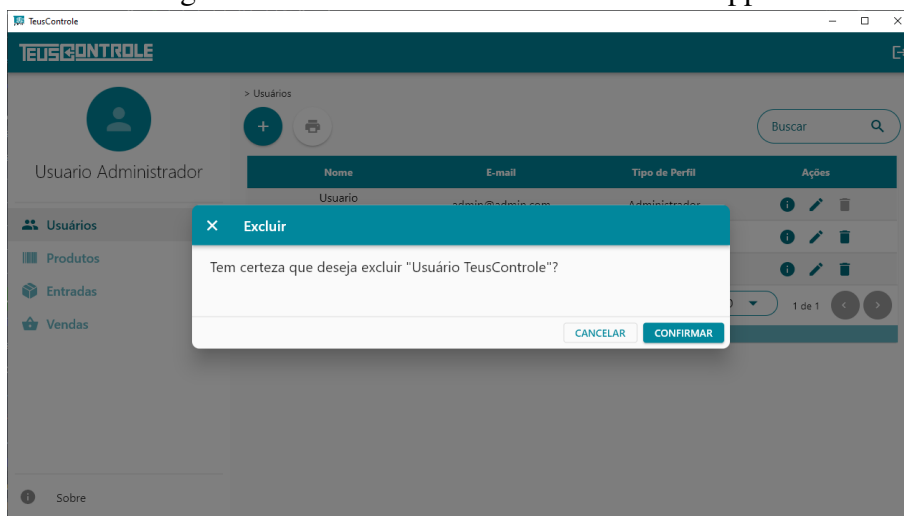
6.2.2.3.4 Exclusão do Registro de Usuários

Para possibilitar a exclusão de registros de usuário, na tabela da listagem, existe na coluna “ações” o ícone de lixeira. Ao pressioná-lo, uma janela de confirmação como na Figura 6.10 é aberta. Na janela, os botões de “confirmar”, “cancelar” e “X” também se fazem presente, possibilitando a confirmação ou cancelamento da operação de exclusão do registro.

6.2.2.4 Tela Produtos

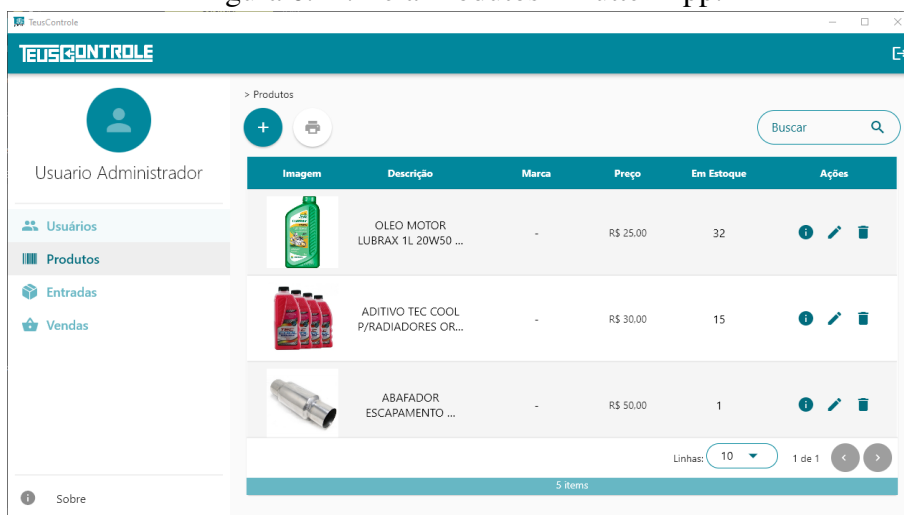
Para o controle do estoque dos produtos, se faz necessário manter registros de produtos. Para isso, na Figura 6.11 é exibido a tela de produtos, onde as operações de cadastro, listagem, atualização e exclusão podem ser realizadas. Para esta tela, apenas o usuário “administrador” tem acesso. O usuário “vendedor” poderá consultar produtos de uma forma resumida em outra tela que ainda será abordada.

Figura 6.10: Exclusão do Usuário - Flutter App.



Fonte: Elaborado pelo autor

Figura 6.11: Tela Produtos - Flutter App.



Fonte: Elaborado pelo autor

6.2.2.4.1 Formulário de Cadastro de Produtos

Ao pressionar o botão com ícone de adição da tela de produtos (Figura 6.11), uma janela flutuante com título “cadastro” com os campos de entrada para registro de produtos irá aparecer.

Na Figura 6.12 é possível observar os campos de entrada “código de barras”, “descrição”, “preço”, “em estoque”, “preço médio”, “marca”, “imagem da marca”, “imagem do produto”, campos para informações da Classificação Global de Produtos (GPC) e Nomenclatura Comum do Mercosul (NCM) que compõem o formulário de cadastro de produtos. Após preenchido as informações, é possível salvar o registro ao pressionar o botão

“confirmar” no rodapé da janela. Também é possível cancelar a operação de cadastro pressionando no rodapé o botão “cancelar” ou ainda no ícone com “X” ao lado do título da janela. Algumas informações estão ocultas na Figura 6.12 pois ficaram fora da área visível da área de rolagem.

Figura 6.12: Formulário Cadastro de Produtos - Flutter App.

Fonte: Elaborado pelo autor

6.2.2.4.2 Edição do Registro de Produtos

Ao pressionar o botão com ícone de lápis da coluna “ações” de um registro da listagem na tela de produtos (Figura 6.11), uma janela flutuante com título “edição” com os campos de entrada irá aparecer com as informações do registro a ser editado.

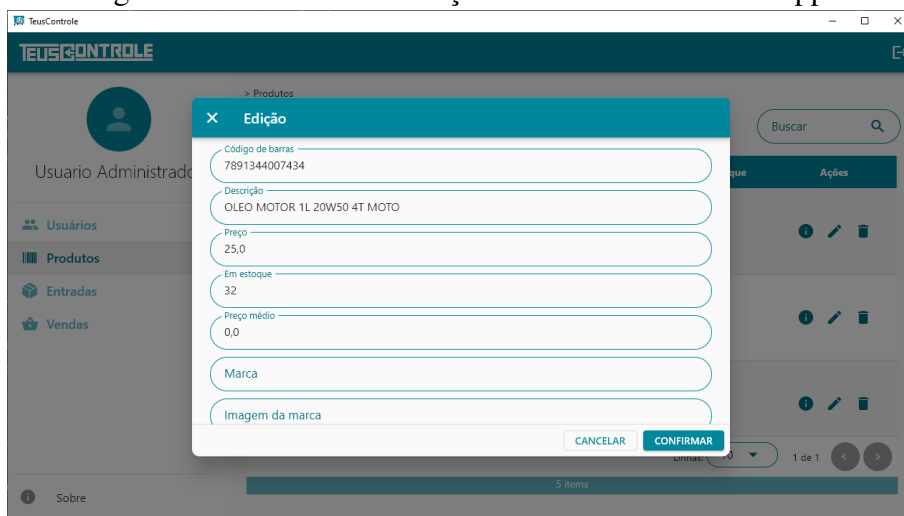
Na Figura 6.13, assim como no formulário de cadastro, os mesmos campos de entrada são apresentados. Os botões de “confirmar”, “cancelar” e “X” também se fazem presente, possibilitando a confirmação ou cancelamento da operação de editar o registro.

6.2.2.4.3 Detalhes do Registro de Produtos

Na tabela da listagem na tela de produtos (Figura 6.11), são apresentadas apenas algumas informações. Para visualizar todas as informações de determinado registro, existe o botão com ícone “i” da coluna “ações” para todos os registros.

Na Figura 6.14 é apresentada a janela com título “detalhes” com as informações do respectivo registro. Na janela são apresentadas todas as informações que foram enviadas no formulário de cadastro, além da data de criação e de última alteração do registro.

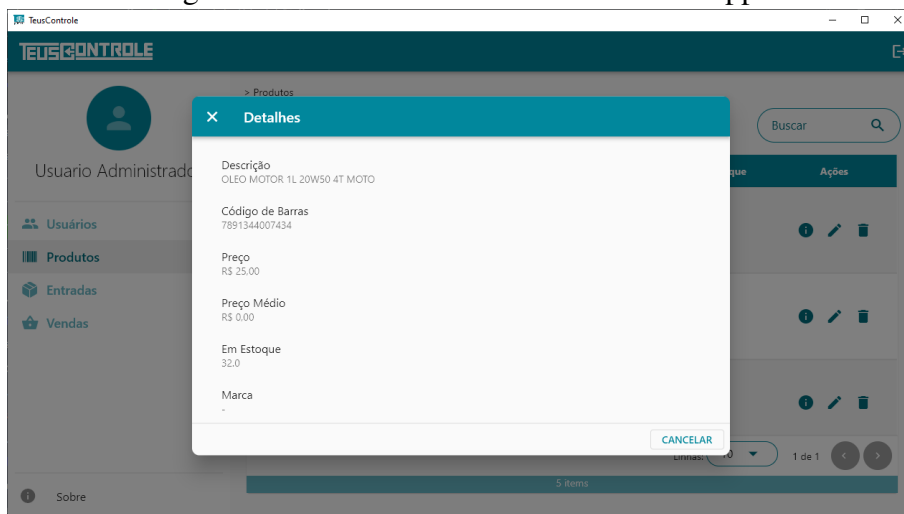
Figura 6.13: Formulário Edição de Produtos - Flutter App.



Fonte: Elaborado pelo autor

Algumas informações estão ocultas na Figura 6.14 pois ficaram fora da área visível da área de rolagem.

Figura 6.14: Detalhes do Produto - Flutter App.

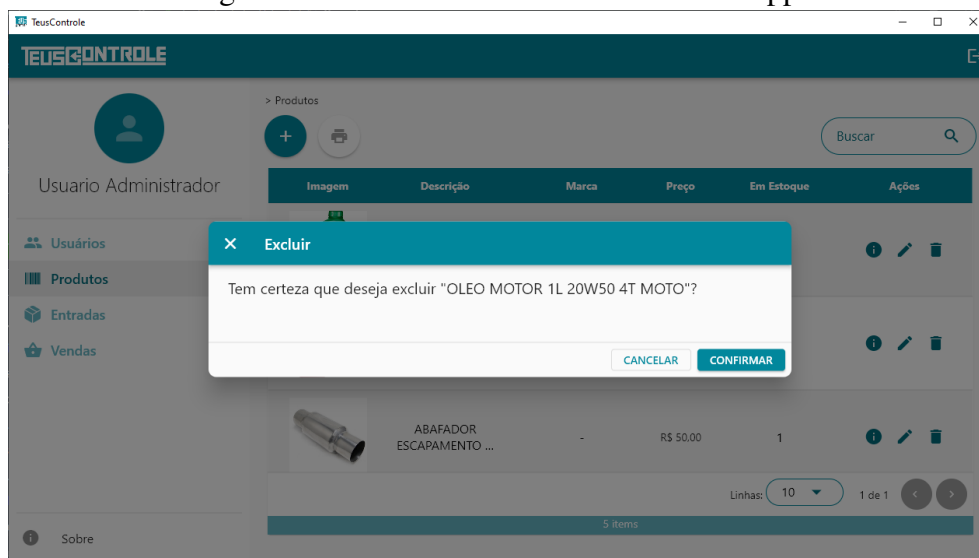


Fonte: Elaborado pelo autor

6.2.2.4.4 Exclusão do Registro de Produtos

Para possibilitar a exclusão de registros de produtos, na tabela da listagem, existe na coluna “ações” o ícone de lixeira. Ao pressioná-lo, uma janela de confirmação como na Figura 6.15 é aberta. Na janela, os botões de “confirmar”, “cancelar” e “X” também se fazem presente, possibilitando a confirmação ou cancelamento da operação de exclusão do registro.

Figura 6.15: Exclusão do Produto - Flutter App.

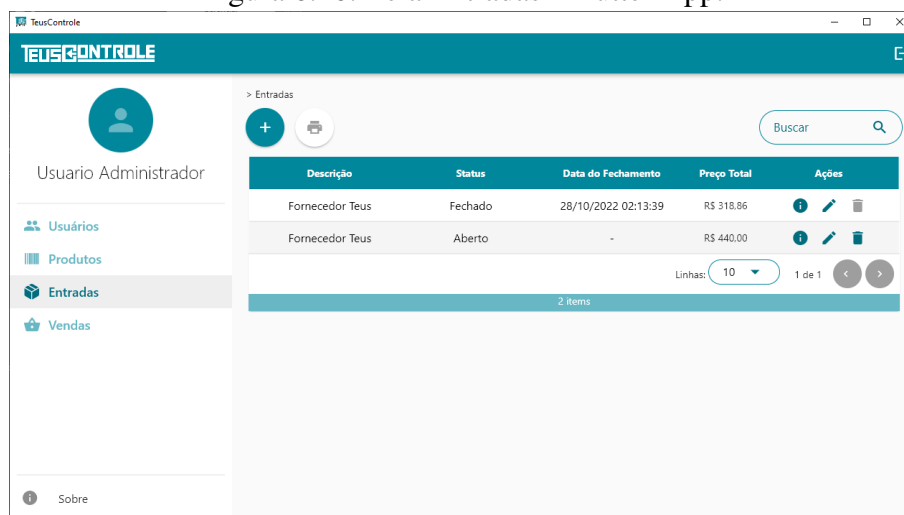


Fonte: Elaborado pelo autor

6.2.2.5 Tela Entradas

Com a finalidade de repor produtos em estoque, o cadastro de entradas de mercadorias se faz necessário. Na Figura 6.16 é exibido a tela de entradas onde as operações de cadastro, listagem, atualização e exclusão podem ser realizadas. Para esta tela e funcionalidade, apenas o usuário “administrador” tem a permissão de acesso.

Figura 6.16: Tela Entradas - Flutter App.



Fonte: Elaborado pelo autor

6.2.2.5.1 Formulário de Cadastro de Entradas

Ao pressionar o botão com ícone de adição da tela de entradas (Figura 6.16), uma

janela flutuante com título “cadastro” com os campos de entrada para registro de entradas de produtos irá aparecer.

Na Figura 6.17 é possível observar os campos de entrada “descrição”, “código”, “preço” e “quantidade” que compõem o formulário de cadastro de entrada de produtos. Também é visível o botão com rótulo “adicionar”, o campo de seleção “*status*” (a fim de satisfazer os requisitos de domínio RD-004 e RD-006) e a tabela de produtos da entrada.

Figura 6.17: Formulário Cadastro de Entradas - Flutter App.

Fonte: Elaborado pelo autor

Assim, após buscar pelo produto, digitando pelo código ou selecionando pela tela “buscar produto” (acessada ao pressionar o campo “produto” - Figura 6.28), e preenchidos os demais campos, ao pressionar o botão “adicionar”, o item é inserido na tabela.

Por fim, após a finalização da inserção de produtos, é possível salvar o registro ao pressionar o botão “confirmar” no rodapé da janela. Também é possível cancelar a operação de cadastro pressionando no rodapé o botão “cancelar” ou ainda no ícone com “X” ao lado do título da janela.

6.2.2.5.2 Edição do Registro de Entradas

Ao pressionar o botão com ícone de lápis da coluna “ações” de um registro da listagem na tela de entradas (Figura 6.16), uma janela flutuante com título “edição” com os campos de entrada irá aparecer com as informações do registro a ser editado. Conforme indica o requisito de domínio RD-006, apenas os registros com *status* “aberto” podem ser alterados.

Na Figura 6.18, assim como no formulário de cadastro, os mesmos campos de

entrada e tabela são apresentados. Também é possível observar que na tabela, existem na coluna “ações” os ícones de lápis e lixeira, que permitem respectivamente, a edição e exclusão dos produtos da entrada. Os botões de “confirmar”, “cancelar” e “X” também estão presentes, possibilitando a confirmação ou cancelamento da operação de editar o registro.

Figura 6.18: Formulário Edição de Entrada - Flutter App.

Imagem	Descrição	Preço Unitário	Quantidade	Preço Total	Ação
	ADITIVO P/ RADIADORES ORG ...	R\$ 5,00	22,0	R\$ 110,00	
	FILTRO DE AR FXX / GXX / VXXXXX - ...	R\$ 15,00	22,0	R\$ 330,00	

Fonte: Elaborado pelo autor

6.2.2.5.3 Detalhes do Registro de Entradas

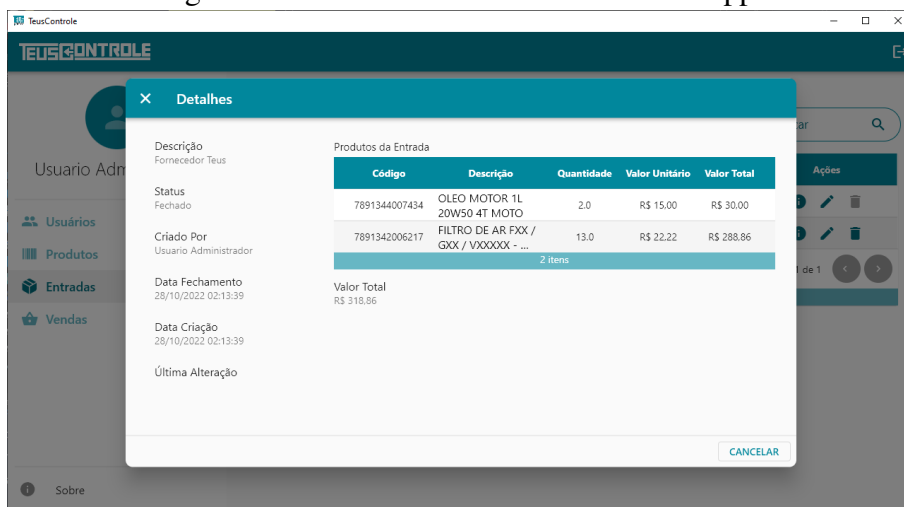
Na tabela da listagem na tela de entradas (Figura 6.16), são apresentadas apenas algumas informações. Para visualizar todas as informações de determinado registro, existe o botão com ícone “i” da coluna “ações” para todos os registros.

Na Figura 6.19 é apresentada a janela com título “detalhes” com as informações do respectivo registro. Na janela são apresentadas todas as informações que foram enviadas no formulário de cadastro, além da data de criação e de última alteração do registro.

6.2.2.5.4 Exclusão do Registro de Entradas

Para possibilitar a exclusão de registros de entradas, na tabela da listagem, existe na coluna “ações” o ícone de lixeira. Ao pressioná-lo, uma janela de confirmação como na Figura 6.20 é aberta. Na janela, os botões de “confirmar”, “cancelar” e “X” também se fazem presente, possibilitando a confirmação ou cancelamento da operação de exclusão do registro. Conforme indica o requisito de domínio RD-009, apenas os registros com

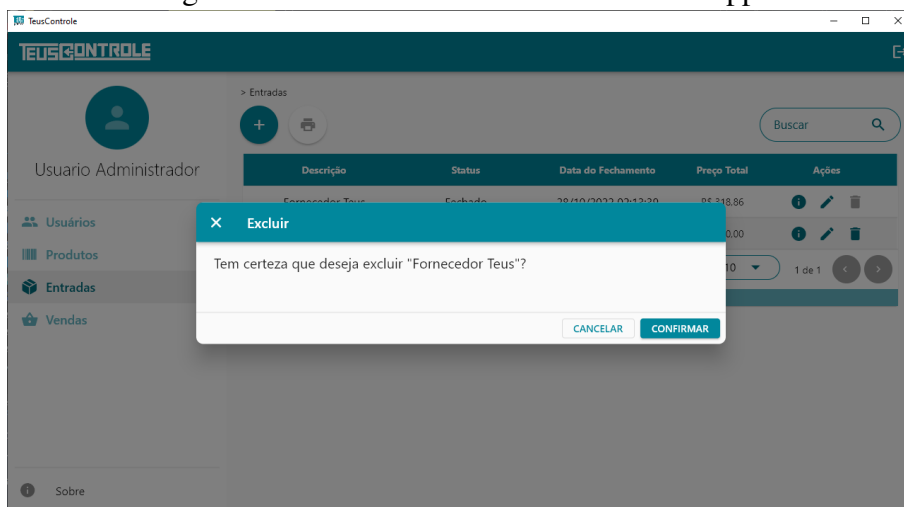
Figura 6.19: Detalhes da Entrada - Flutter App.



Fonte: Elaborado pelo autor

status diferente de “fechado” podem ser excluídos.

Figura 6.20: Exclusão da Entrada - Flutter App.

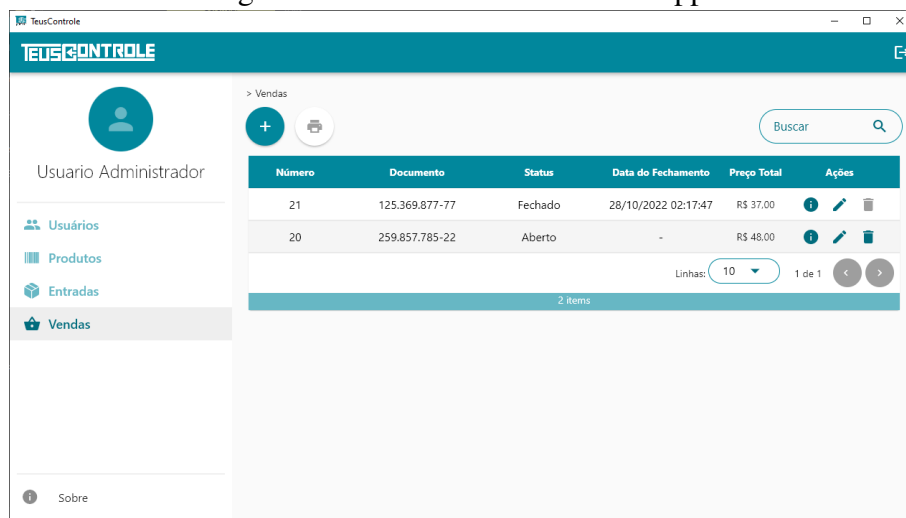


Fonte: Elaborado pelo autor

6.2.2.6 Tela Vendas

Com a finalidade de remover produtos do estoque, o cadastro de vendas torna-se necessário. Na Figura 6.21 é exibido a tela de vendas onde as operações de cadastro, listagem, atualização e exclusão podem ser realizadas. Para esta tela e funcionalidade, apenas o usuário “administrador” tem permissão de acesso. O usuário “vendedor” terá acesso apenas a tela de cadastro de vendas, que ainda será abordada neste trabalho.

Figura 6.21: Tela Vendas - Flutter App.



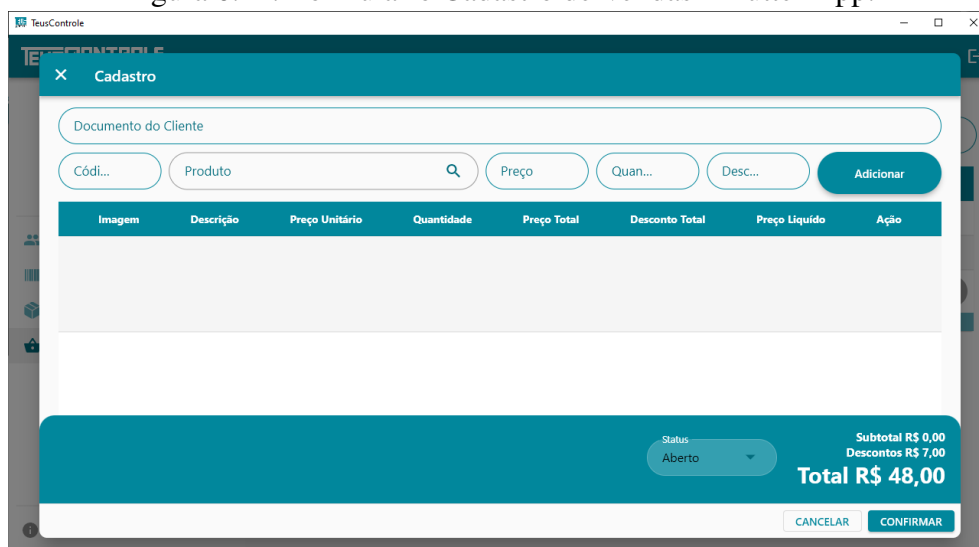
Fonte: Elaborado pelo autor

6.2.2.6.1 Formulário de Cadastro de Vendas

Ao pressionar o botão com ícone de adição da tela de vendas (Figura 6.21), uma janela flutuante com título “cadastro” com os campos de entrada para registro de venda de produtos irá aparecer.

Na Figura 6.22 é possível observar os campos de entrada “documento do cliente”, “código”, “produto”, “preço”, “quantidade” e “desconto” que compõem o formulário de cadastro de venda de produtos. Também é visível o botão com rótulo “adicionar”, o campo de seleção “status” (a fim de satisfazer os requisitos de domínio RD-003 e RD-005) e a tabela de produtos da venda.

Figura 6.22: Formulário Cadastro de Vendas - Flutter App.



Fonte: Elaborado pelo autor

Assim, após buscar pelo produto, digitando pelo código ou selecionando pela tela “buscar produto” (acessada ao pressionar o campo “produto” - Figura 6.28), e preenchidos demais campos, ao pressionar o botão “adicionar”, o item é inserido na tabela.

Por fim, após a finalização da inserção de produtos, torna-se possível salvar o registro ao pressionar o botão “confirmar” no rodapé da janela. Também é possível cancelar a operação de cadastro pressionando no rodapé o botão “cancelar” ou ainda no ícone com “X” ao lado do título da janela.

Ao confirmar uma venda com o *status* “fechado”, é possível emitir um relatório com os dados da respectiva venda. Na Figura 6.23 é apresentado a janela com o título “pré-visualizar” onde uma visualização previa do documento é exibida. No rodapé da janela, existem os ícones de impressora e compartilhamento, que possibilitam, respectivamente, o envio do documento para uma impressora e a geração do recibo como *pdf*.

Figura 6.23: Recibo de Vendas - Flutter App.



Fonte: Elaborado pelo autor

6.2.2.6.2 Edição do Registro de Vendas

Ao pressionar o botão com ícone de lápis da coluna “ações” de um registro da listagem na tela de vendas (Figura 6.21), uma janela flutuante com título “edição” com os campos de entrada irá aparecer com as informações do registro a ser editado. Conforme indica o requisito de domínio RD-005, apenas os registros com *status* “aberto” podem ser alterados.

Na Figura 6.24, assim como no formulário de cadastro, os mesmos campos de entrada e tabela são apresentados. Também é possível observar que na tabela, existem

na coluna “ações” os ícones de lápis e lixeira, que permitem respectivamente, a edição e exclusão dos produtos da venda. Os botões de “confirmar”, “cancelar” e “X” também estão presentes, possibilitando a confirmação ou cancelamento da operação de editar o registro.

Figura 6.24: Formulário Edição de Vendas - Flutter App.

Imagem	Descrição	Preço Unitário	Quantidade	Preço Total	Desconto Total	Preço Líquido	Ação
	OLEO MOTOR 1L 20W50 4T MOTO	R\$ 25,00	1,0	R\$ 25,00	R\$ 7,00	R\$ 18,00	
	ADITIVO P/ RADIADORES ORG ...	R\$ 30,00	1,0	R\$ 30,00	R\$ 0,00	R\$ 30,00	

Subtotal R\$ 55,00
Descontos R\$ 7,00
Total R\$ 48,00

Status: Aberto

CANCELAR CONFIRMAR

Fonte: Elaborado pelo autor

6.2.2.6.3 Detalhes do Registro de Vendas

Na tabela da listagem na tela de vendas (Figura 6.21), são apresentadas apenas algumas informações. Para visualizar todas as informações de determinado registro, existe o botão com ícone “i” da coluna “ações” para todos os registros.

Na Figura 6.25 é apresentada a janela com título “detalhes” com as informações do respectivo registro. Na janela são exibidas todas as informações que foram enviadas no formulário de cadastro, além da data de criação e de última alteração do registro. Além do final de uma venda (Figura 6.23), na tela de detalhes, existe um botão com rótulo “gerar recibo”, que permite reemitir o relatório com as informações da respectiva venda.

6.2.2.6.4 Exclusão do Registro de Vendas

Para possibilitar a exclusão de registros de vendas, na tabela da listagem, existe na coluna “ações” o ícone de lixeira. Ao pressioná-lo, uma janela de confirmação como na Figura 6.26 é aberta. Na janela, os botões de “confirmar”, “cancelar” e “X” também estão presentes, possibilitando a confirmação ou cancelamento da operação de exclusão

Figura 6.25: Detalhes da Venda - Flutter App.

Código	Descrição	Quantidade	Valor Unitário	Valor Total	Desconto Total
7891344007434	OLEO MOTOR 1L 20W50 4T MOTO	1,0	R\$ 25,00	R\$ 25,00	R\$ 7,00
7898307292401	ADITIVO P/ RADIADORES ORG ...	1,0	R\$ 30,00	R\$ 30,00	R\$ 0,00
2 itens					

Valor Total: R\$ 55,00
Desconto Total: R\$ 7,00
Valor Liquido: R\$ 48,00

Gerar Recibo

CANCELAR

Fonte: Elaborado pelo autor

do registro. Conforme indica o requisito de domínio RD-010, apenas os registros com *status* diferente de “fechado” podem ser excluídos.

Figura 6.26: Exclusão de Venda - Flutter App.

Excluir

Tem certeza que deseja excluir "259.857.785-22"?

CANCELAR CONFIRMAR

Fonte: Elaborado pelo autor

6.2.2.7 Tela Ponto de Venda

Ao se autenticar no sistema com um usuário com tipo de perfil “vendedor”, apenas a tela com o formulário venda será visualizada. Na Figura 6.27 é apresentado a tela de ponto de venda. A tela se assemelha com a tela de venda do qual o usuário “administrador” possui acesso (Figura 6.21), com exceção do campo descontos, que para o usuário “vendedor” deve estar desabilitado.

Sendo assim, após buscar pelo produto, digitando pelo código ou selecionando

Figura 6.27: Tela Ponto de Venda - Flutter App.

Fonte: Elaborado pelo autor

pela tela “buscar produto” (acessada ao pressionar o campo “produto” - Figura 6.28), e preenchidos demais campos, ao pressionar o botão “adicionar”, o item é inserido na tabela.

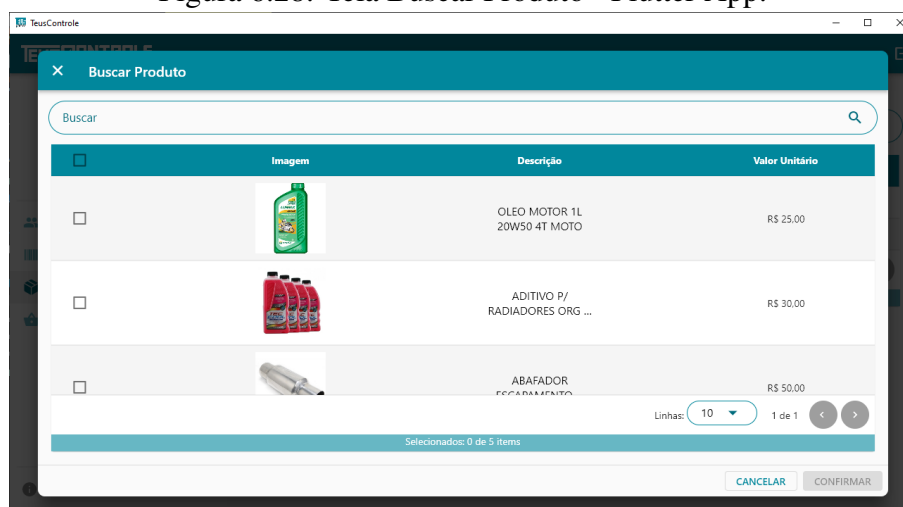
Por fim, após a finalização da inserção de produtos da venda, torna-se possível salvar o registro ao pressionar o botão “confirmar” no rodapé da janela. Conforme apresenta o requisito de domínio RD-007, ao confirmar uma venda realizada pelo usuário “vendedor”, a venda será fechada. Sendo assim, é possível emitir o relatório com os dados da respectiva venda como mostra a Figura 6.23. Também possibilita cancelar a operação de cadastro pressionando no rodapé o botão “cancelar”, que realiza a exclusão de todas as informações previamente inseridas nos campos e tabela.

6.2.2.8 Tela Buscar Produto

A busca dos produtos para serem registrados nas entradas ou vendas podem ser realizadas a partir do código de barras, porém, caso não seja possível (como por problemas na etiqueta do código de barras), também possibilita buscar o produto pela tela buscar produto.

Na Figura 6.28 é apresentado uma busca paginada dos produtos cadastrados, onde é possível pesquisar pelo item no campo “buscar”. Para retornar para a tela onde o produto se faz necessário, é preciso selecionar um item, marcando a caixa de seleção, e pressionar o botão “continuar”. Na janela, os botões “cancelar” e “X” também estão presentes, possibilitando o cancelamento da busca pelo produto.

Figura 6.28: Tela Buscar Produto - Flutter App.



Fonte: Elaborado pelo autor

6.3 Gerenciamento de Configurações

De acordo com Aiello e Sachs¹ (apud SOMMERVILLE, 2018), o gerenciamento de configurações diz respeito “às políticas, aos processos e as ferramentas para o gerenciamento de sistema de software que mudam”. Para Sommerville (2018), o gerenciamento de configurações compreende as atividades relacionadas ao controle de versão, construção de sistema, gerenciamento de mudanças e gerenciamento de lançamentos.

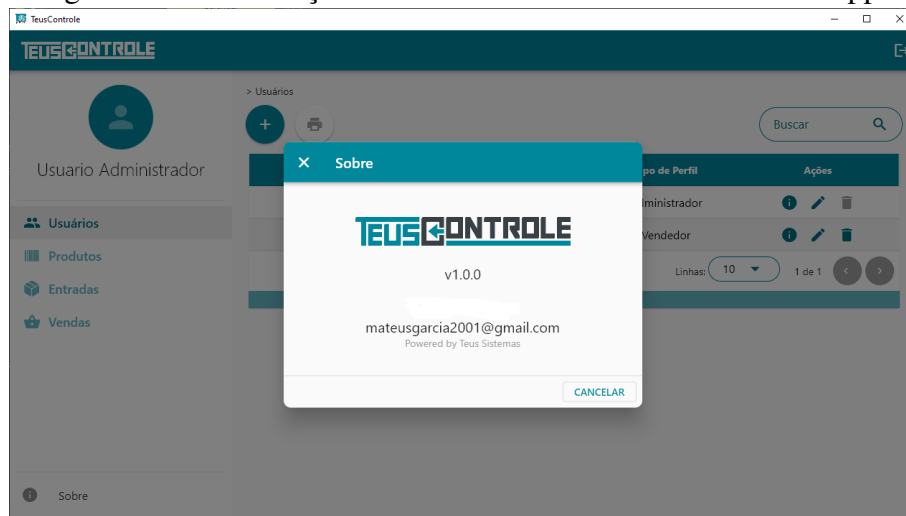
Para este trabalho, não foram definidos e discutidos todos os processos que trata do gerenciamento de configurações. Levando em consideração o atual estágio que o software TeusControle se encontra e o curto período de realização, neste aspecto, fica evidente que o software desenvolvido representa um MVP, ou seja, um mínimo produto viável, que para Ries (2009), consiste na versão de um novo produto que permite a equipe apanhar o máximo de aprendizado validado pelos clientes, com um menor esforço.

Sendo assim, fica definido que a versão do software apresentado neste trabalho é a *v1.0.0*, ou seja, a primeira versão construída e a ser disponibilizada para validação. Na Figura 6.29 é apresentada informações do sistema, como o identificador da versão e algumas informações de contato, que podem ser acessadas ao pressionar o item “sobre” no menu lateral da aplicação.

Para construir a versão a ser lançada do projeto *back-end*, foi utilizado do Docker (2022), uma plataforma que possibilita construir, executar e compartilhar aplicações em ambientes isolados ou também conhecidos por *containers*. Para compor a solução, foi

¹ AIELLO, B.; SACHS, L. **Configuration Management Best Practices: Practical Methods that Work in the Real World**. [S.l.]: Pearson Education, 2010.

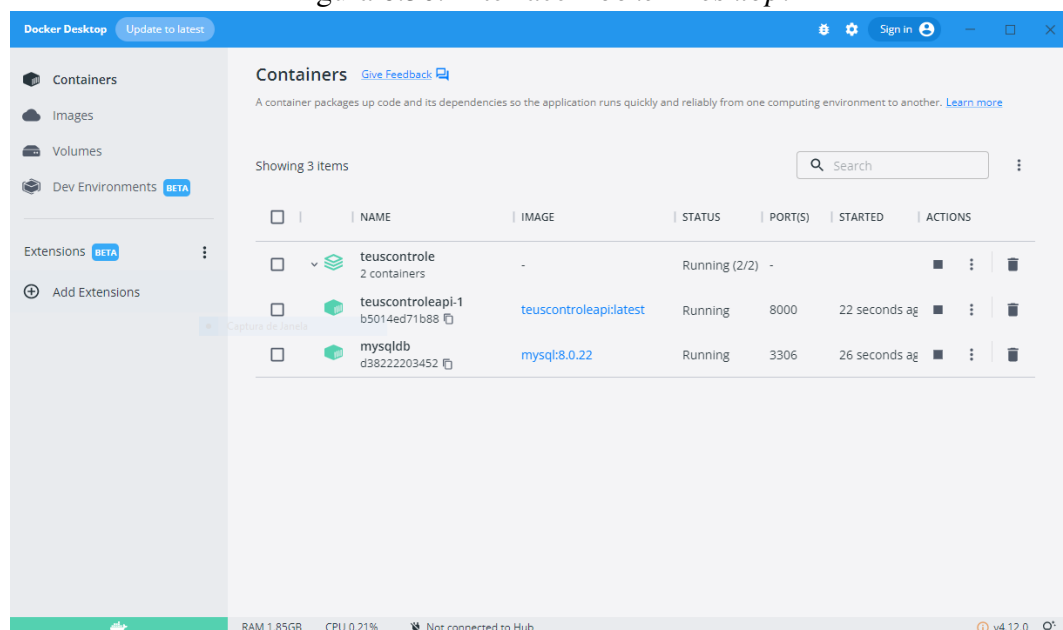
Figura 6.29: Informações do Software TeusControle - Flutter App.



Fonte: Elaborado pelo autor

utilizado do *Docker Compose*, uma ferramenta que permite executar aplicações composta por vários *containers*, que neste caso, compreendem a imagem do TeusControle API e a imagem da base de dados em *MySQL*.

Na Figura 6.30 é apresentado a API construída e a base de dados sendo executadas dentro de *containers*, presentes dentro da plataforma disponibilizada pelo *Docker*. Na figura, é apresentado a interface do *Docker Desktop*, uma aplicação que oferece uma interface simples para permitir o gerenciamento dos *containers*, imagens e aplicações sem a necessidade de utilizar do CLI oferecido.

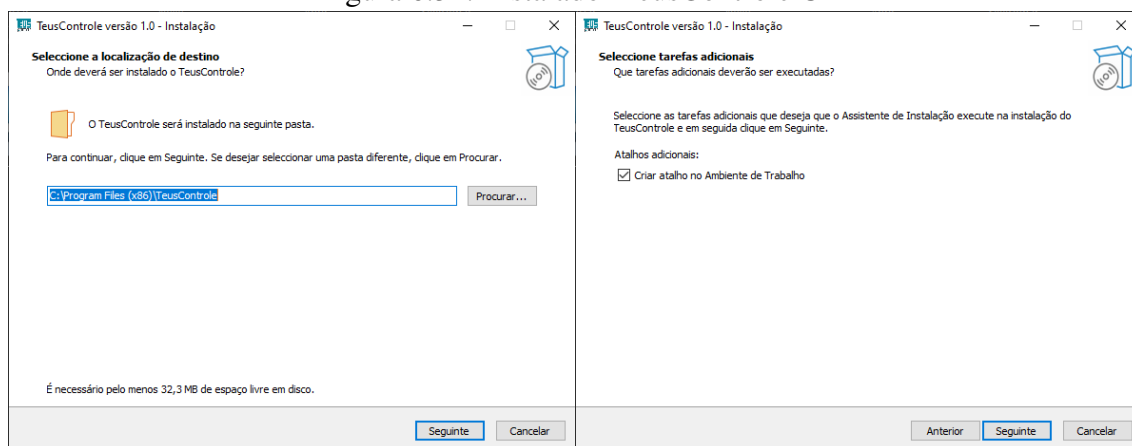
Figura 6.30: Interface *Docker Desktop*.

Fonte: Elaborado pelo autor

Quanto ao projeto TeusControle UI, o próprio *Flutter* disponibiliza ferramentas para gerar os arquivos das dependências da aplicação e o arquivo executável para o ambiente *Windows*. Com o objetivo de gerar o instalador com todos os arquivos gerados previamente, foi utilizado do Inno Setup, um criador de instalador gratuito dirigido a *scripts* (RUSSELL; LAAN, 2022).

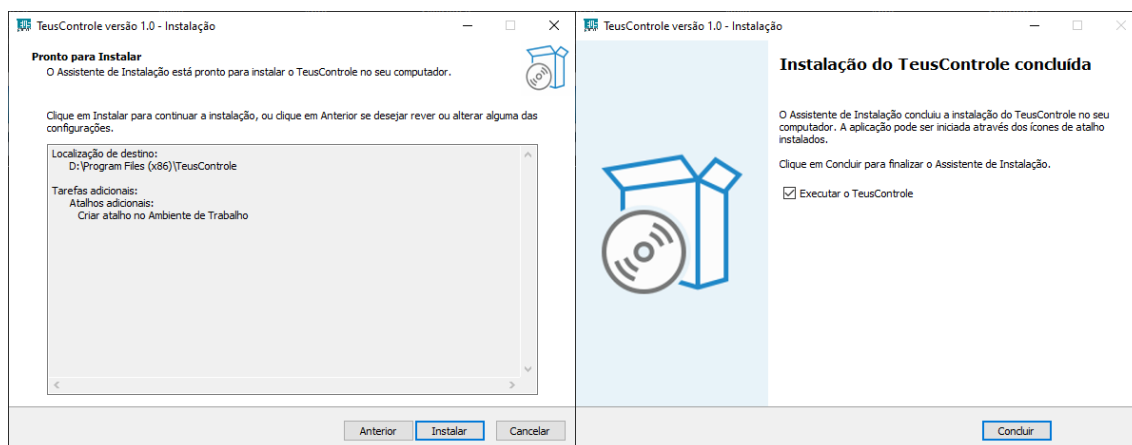
Na Figura 6.31 é apresentado as quatro etapas da instalação. Na Figura 6.31a é apresentado a escolha do local da pasta onde ficará os arquivos da aplicação, na tela da Figura 6.31b se faz possível marcar a opção para gerar o atalho na área de trabalho, na Figura 6.31c é apresentado um resumo do que será realizado e por fim na Figura 6.31d a tela de confirmação da instalação.

Figura 6.31: Instalador TeusControle UI



(a) Diretório de instalação

(b) Tarefas adicionais



(c) Resumo da instalação

(d) Confirmação da instalação

Fonte: Elaborado pelo autor

6.4 Resultados e *Feedbacks* da Empresa

Antes da implantação do sistema na loja de autopeças Auto Peças W.R, foi apresentado o sistema e recolhido alguns *feedbacks* e resultados preliminares. Sendo assim, após mostrar a atual versão do TeusControle, o proprietário da empresa reconheceu que o sistema atende à necessidade identificada. Porém, algumas funcionalidades e recursos não previamente estipulados foram levantados como necessários, sendo eles:

- Mapeamento de ações e atalhos para manusear o sistema com maior agilidade;
- Inclusão de dicas (*tooltips*) para botões e ícones do sistema;
- Adequação das telas de detalhes e tabelas de listagem dos cadastros a fim de possibilitar a cópia do conteúdo para área de transferência;
- Implementação da funcionalidade de devolução de produtos da venda;
- Implementação de um *dashboard* com principais informações e relatórios gerenciais;

Em outro momento, sucedeu-se a tentativa de implantação do sistema. Pretendia-se configurar os artefatos apenas para uso em ambiente local, entretanto não se fez possível por motivos não previamente identificados e previstos.

Ao tentar instalar o *Docker* para hospedar o TeusControle API, ocorreu uma falha na instalação de um de seus pré-requisitos. Analisando brevemente o problema, foi identificado um provável corrompimento no sistema operacional da máquina a ser utilizada. Sendo assim, não foi possível colocar a aplicação para ser testada pela loja por um período maior, como se pretendia. Não possibilitando assim, uma análise dos impactos e valores agregados pelo uso do sistema, consequentemente não atendendo na íntegra um dos objetivos propostos pelo trabalho: Testar e validar a implementação do sistema.

7 CONCLUSÃO

Esse trabalho buscou apresentar o estudo e desenvolvimento do software TeusControle, um sistema de controle de estoque que busca agregar valor à gestão e processos de negócio à empresas que não fazem o controle de seus estoques. Para o seu desenvolvimento se fez necessário realizar atividades relacionadas à “comunicação, planejamento, modelagem, construção e entrega”, que Maxim e PRESSMAN (2021) afirmam serem tarefas gerais da engenharia de software. Além disso, foram elencados sete objetivos específicos, dos quais destes, um não foi totalmente atendido.

Inicialmente, buscando entender a necessidade da empresa Auto Peças W.R., foram realizadas conversas com o proprietário do negócio e levantado suas principais necessidades, desta forma, atendendo ao primeiro objetivo específico apresentado.

Para compreender a situação no mercado, foram analisadas soluções semelhantes, como propôs o segundo objetivo específico.

Em seguida, para satisfazer o terceiro objetivo, foi realizada a construção da proposta da solução, e gerenciamento dos requisitos com auxílio de ferramentas digitais.

E, com a finalidade de iniciar as atividades relacionadas ao desenvolvimento da solução, foram selecionadas e estudadas as ferramentas computacionais para apoiar a construção do TeusControle, atendendo assim o quarto objetivo.

Para satisfazer ao quinto objetivo, foram implementados os cadastros, movimentações e relatórios necessários, buscando satisfazer os requisitos funcionais, requisitos não funcionais e requisitos de domínio, apresentados, respectivamente nas Tabelas 5.1, 5.2 e 5.3.

Também foi apresentado a solução para o proprietário da empresa, e recolhido alguns *feedbacks* de melhorias futuras. Sendo assim, atendendo parcialmente o sexto objetivo, visto que não foi possível disponibilizar a aplicação para ser utilizada e validada em um ambiente real.

Além disso, fica evidente que o processo de desenvolvimento e resultados alcançados foram devidamente documentados, como propõe o sétimo e último objetivo específico estabelecido.

Por fim, era esperado que, ao concluir este trabalho, a solução TeusControle estivesse disponível e em uso pela empresa, possibilitando agregar valor à sua gestão e processos internos, mas como observado, isto não foi possível. Dessa maneira, será proposto alguns trabalhos futuros que buscam atender o que não foi possível realizar.

7.1 Recomendações para Trabalhos Futuros

Durante a realização deste trabalho, foram identificados dificuldades na implantação do sistema para o cliente em potencial. Desta forma, como um futuro trabalho a ser realizado, se mostra importante preparar um ambiente propício para validar a atual versão e versões com futuras correções e novas funcionalidades que possam ser desenvolvidas. Como também analisar e avaliar a possibilidade de hospedar o serviço TeusControle API em alguma plataforma que ofereça serviços de computação em nuvem.

Levando em consideração as aplicações existentes, apresentadas na Seção 2, e a enorme quantidade de clientes atendidos por estas aplicações; implementar os recursos e funcionalidades trazidas pela empresa e registrado na Subseção 6.4 se mostra uma atividade futura importante a ser realizada para ganhar espaço no mercado competitivo de soluções semelhantes. Além disso, aplicar conceitos administrativos da gestão de estoque (como cálculo do ponto de pedido, análise ABC e outros métodos), apresenta-se como oportunidades para agregar valor às informações coletadas e registradas pelo TeusControle.

Também, buscando alcançar uma maior diversidade de público, criar versões da aplicação para outras plataformas, se mostra uma oportunidade de trabalho futuro relevante para o software desenvolvido. Em conjunto com a disponibilização do serviço TeusControle API em nuvem, criar versões do TeusControle para *web* e *mobile* se mostra uma possibilidade.

Outra possibilidade futura é a implementação de uma funcionalidade para lançamentos de divergências do estoque no sistema com o estoque físico. Como erros em decorrência do registro de movimentações de estoque equivocadas e a inutilização de produtos são situações possíveis, esta funcionalidade se mostra bastante importante para acompanhar os gastos e adequar a discrepância do estoque.

Os trabalhos futuros também podem explorar o desenvolvimento de um estudo de caso relacionado ao uso do sistema pela empresa estudada ou ainda outras empresas, do mesmo ou outros ramos de atuação. Desta forma, pode-se, levantar maiores informações quanto a viabilidade e ganhos proporcionados pelo sistema desenvolvido, além de levantar pontos críticos, acertos e erros na implementação. Tais informações se mostram relevantes para a perpetuação e maturidade do produto.

REFERÊNCIAS

- AIELLO, B.; SACHS, L. **Configuration Management Best Practices: Practical Methods that Work in the Real World**. [S.l.]: Pearson Education, 2010.
- BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. [S.l.]: Elsevier Rio de Janeiro, 2015.
- CAPTERRA. **Sobre nós - Capterra Brasil 2022**. 2022. Last access: 26-06-2022. Available from Internet: <<https://www.capterra.com.br/company/about>>.
- COSMOS, B. **Catalogo Online de Produtos**. 2022. Last access: 29-10-2022. Available from Internet: <<https://cosmos.bluesoft.com.br/>>.
- DART. **Dart programming language**. 2022. Last access: 04-07-2022. Available from Internet: <<https://dart.dev/>>.
- DOCKER. **Docker: Accelerated, Containerized Application Development**. 2022. Last access: 29-10-2022. Available from Internet: <<https://www.docker.com/>>.
- FLUTTER. **Flutter - Build apps form many screens**. 2022. Last access: 04-07-2022. Available from Internet: <<https://flutter.dev/>>.
- GIT. **Git**. 2022. Last access: 03-07-2022. Available from Internet: <<https://git-scm.com/>>.
- GITHUB. **GitHub: Where the world builds software**. 2022. Last access: 03-07-2022. Available from Internet: <<https://github.com/>>.
- GOOGLE. **Material Design**. 2022. Last access: 29-10-2022. Available from Internet: <<https://material.io/design/usability/accessibility.html>>.
- GOOGLE. **Plataforma pessoal de armazenamento na nuvem e compartilhamento de arquivos do Google**. 2022. Last access: 03-07-2022. Available from Internet: <<https://www.google.com/intl/pt-BR/drive/>>.
- GUEDES, G. T. A. **UML 2 - Uma Abordagem Prática**. 3. ed. [S.l.]: Novatec Editora, 2018.
- JIRA. **Visão geral do Jira**. 2022. Last access: 03-07-2022. Available from Internet: <<https://www.atlassian.com/br/software/jira/guides/getting-started/overview>>.
- LEITE, J. C. S. do P.; LEONARDI, M. C. Business rules as organizational policies. In: IEEE. **Proceedings Ninth International Workshop on Software Specification and Design**. [S.l.], 1998. p. 68–76.
- LUCIDCHART. **Como fazer cinco tipos de diagrama de arquitetura de sistema**. 2022. Last access: 26-06-2022. Available from Internet: <<https://www.lucidchart.com/blog/pt/como-fazer-diagramas-de-arquitetura-de-sistema>>.
- LUCIDCHART. **Software online de diagramas e comunicação visual**. 2022. Last access: 03-07-2022. Available from Internet: <<https://www.lucidchart.com/pages/pt>>.

MARKETUP. **Plataforma de Gestão e Vendas Grátis Para PMEs - MarketUP**. 2022. Last access: 26-06-2022. Available from Internet: <<https://marketup.com/>>.

MAXIM, B.; PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. [S.l.]: Porto Alegre:[sn], 2021.

MEIRELLES, F. S. Pesquisa do uso da ti - tecnologia de informação nas empresas, fgvcia. **FGV EAESP**, v. 33, 2022. Available from Internet: <https://eaesp.fgv.br/sites/eaesp.fgv.br/files/u68/fgvcia_pes_ti_2022_-_relatorio.pdf>.

MICROSOFT. **NET (e .NET Core) – introdução e visão geral - Microsoft Learn**. 2022. Last access: 13-11-2022. Available from Internet: <<https://learn.microsoft.com/pt-br/dotnet/core/introduction>>.

MICROSOFT. **Um tour do C# - visão geral**. 2022. Last access: 06-07-2022. Available from Internet: <<https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>>.

MYSQL. **MySQL :: MySQL Workbench**. 2022. Last access: 03-07-2022. Available from Internet: <<https://www.mysql.com/products/workbench/>>.

ODOO. **Open Source ERP and CRM | Odoo**. 2022. Last access: 26-06-2022. Available from Internet: <https://www.odoo.com/pt_BR>.

ORACLE. **What is MySQL? | Oracle**. 2022. Last access: 13-11-2022. Available from Internet: <<https://www.oracle.com/mysql/what-is-mysql/>>.

PINTO, R. A. Q. et al. Gestão de estoque e lean manufacturing: estudo de caso em uma empresa metalúrgica. **Revista Administração em Diálogo-RAD**, v. 15, n. 1, 2013.

RIES, E. Minimum viable product: a guide. **Startup lessons learned**, v. 3, p. 1, 2009.

RUSSELL, J.; LAAN, M. **Inno Setup**. 2022. Last access: 03-11-2022. Available from Internet: <<https://jrsoftware.org/isinfo.php>>.

SEBRAE. **E Book Mantendo o estoque em dia**. [s.n.], 2017. Last access: 08-07-2022. Available from Internet: <https://www.sebrae.com.br/Sebrae/Portal/%20Sebrae/Arquivos/ebook_mantendo-o-estoque-em-dia.pdf>.

SILVA, K. B. A. da; MADEIRA, G. J. Gestão de estoques e lucro da empresa. In: **Anais do Congresso Brasileiro de Custos-ABC**. [S.l.: s.n.], 2004.

SMARTBEAR. **API Documentation & Design Tools for Teams | Swagger**. 2022. Last access: 10-07-2022. Available from Internet: <<https://swagger.io/>>.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. [S.l.]: Pearson Education do Brasil Ltda., 2018.

STUDIO, A. **Conheça o Android Studio**. 2022. Last access: 03-07-2022. Available from Internet: <<https://developer.android.com/studio/intro>>.

STUDIO, V. **Visual Studio 2022 Community Edition**. 2022. Last access: 03-07-2022. Available from Internet: <<https://visualstudio.microsoft.com/pt-br/vs/community/>>.

UNIVERSE, L. **Programa para Controle de Vendas e Gestão do Estoque**. 2022. Last access: 26-06-2022. Available from Internet: <<https://loyverse.com/br>>.