

PROJETO INTEGRADO

JÚLIA DA ROSA HARLACHER MARQUES

Projeto Integrado

Arroio dos Ratos - RS  
2024

JÚLIA DA ROSA HARLACHER MARQUES

## Projeto Integrado

Trabalho de Projeto Integrado apresentado como requisito parcial para a obtenção de média semestral.

Tutor: Luis Gustavo Cardoso

Arroio dos Ratos - RS  
2024

## SUMÁRIO

1	INTRODUÇÃO .....	3
2	DESENVOLVIMENTO .....	4
	2.1 QUADRO SCRUM.....	4
	2.2 TABELA VERDADE.....	10
	2.3 FUNÇÕES .....	11
	2.4 DIAGRAMA DE CASOS DE USO.....	15
3	CONCLUSÃO.....	17
4	REFERÊNCIAS .....	18

## 1 INTRODUÇÃO

Nesta atividade, foi proposto a criação de um aplicativo de gerenciamento de estoque. O sistema deve ser capaz de auxiliar em funções de cadastro de produtos, atualização de estoque, rastreamento de localização e relatórios. A metodologia Ágil proposta foi o quadro SCRUM, que foi criado no software online Trello.

A fim de elucidar melhor os processos e escolhas de desenvolvimento, foram simuladas datas fictícias para cada Sprint, bem como possíveis empecilhos e estratégias para resolução de problemas no ambiente de desenvolvimento.

Além de apresentação de códigos de funções para o aplicativo, uma tabela verdade sobre o sistema e um diagrama de casos de uso foram desenvolvidos nesta atividade multidisciplinar.

Na primeira parte do trabalho, foi criado o quadro do Scrum no Trello, inicialmente foram identificadas as etapas essenciais do desenvolvimento do projeto com base na premissa fornecida. Em seguida, foram utilizadas as funcionalidades do Trello para criar listas correspondentes a cada etapa do processo Scrum, incluindo “Backlog”, “Sprint atual”, “Em progresso” e “Concluído”. Em cada lista, foram adicionados cards representando as tarefas específicas do projeto, atribuindo datas de entrega e responsáveis a cada uma delas.

## 2 DESENVOLVIMENTO

### 2.1 – Quadro SCRUM

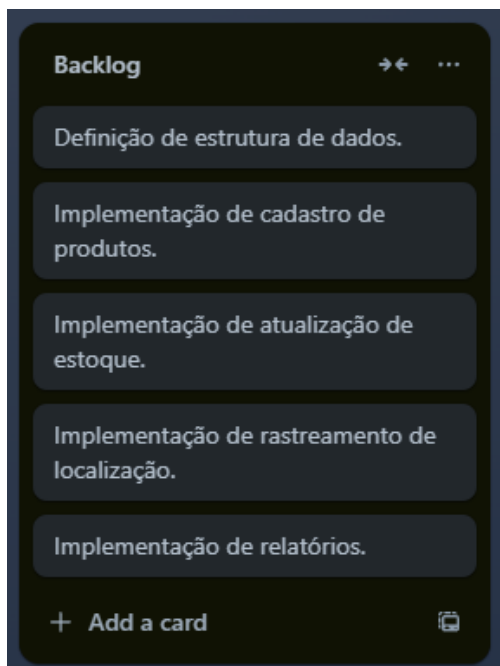
#### Primeira Etapa - Definição de Sprints e datas

Foram definidas 3 Sprints com 2 semanas cada:

Sprint	Período	Objetivo	Tarefas Principais
1	01/05/2024 - 14/05/2024	Configuração Inicial e Estrutura de Dados	Definição de Estrutura de Dados Implementação de Cadastro de Produtos
2	15/05/2024 - 28/05/2024	Atualização de Estoque e Rastreamento de Localização	Implementação de Atualização de Estoque Implementação de Rastreamento de Localização
3	29/05/2024 - 11/06/2024	Relatórios	Implementação de Relatórios

#### Product Backlog

No Trello, foi criado o Backlog com as Sprints necessárias para o desenvolvimento do programa de Estoque.



## **Sprint 1: Configuração Inicial, Estrutura de Dados e Cadastro de produtos**

Período: 01/05/2024 – 14/05/2024

Objetivo: Configurar o quadro no Trello, definir a estrutura de dados e implementar o cadastro de produtos.

### **Tarefas:**

- Definição de Estrutura de Dados
- Implementação de Cadastro de Produtos

### **Processos:**

#### **01/05/2024:**

Foi feita a reunião de planejamento da Sprint a fim de alinhar os processos e procedimentos.

#### **02/05/2024 - 13/05/2024:**

#### **Definição de Estrutura de Dados:**

Nesta etapa, o foco foi em criar classes e relacionamentos no código. O card "Definição de Estrutura de Dados" foi movido para "Em Progresso" em 02/05/2024. A conclusão desta etapa foi em 05/05/2024 e seu card foi movido para "Concluído".

#### **Implementação de Cadastro de Produtos:**

O foco era desenvolver a funcionalidade de cadastro. O card "Implementação de Cadastro de Produtos" foi movido para "Em Progresso" em 06/05/2024 e, então concluído e movido para "Concluído" em 13/05/2024.

#### **14/05/2024:**

#### **Revisão da Sprint 1**

Nesta data, que marca o final na primeira sprint, foi feita a avaliação do que foi concluído através de discussões sobre os desafios e planejamento de melhorias.

## **Relatório primeira Sprint**

Data de Início: 01/05/2024

Data de Término: 14/05/2024

### **Tarefas Concluídas:**

- Definição de Estrutura de Dados
- Implementação de Cadastro de Produtos

### **Desafios:**

- Ajustes na definição das relações entre categorias e produtos.
- Necessidade de refinar a estrutura para futuras funcionalidades.

### **Sucessos:**

- Estrutura de dados definida eficientemente.
- Cadastro de produtos implementado e testado com sucesso.

### **Melhorias para Próxima Sprint:**

- Melhor comunicação sobre requisitos técnicos.
- Ajustar tempo de estimativas para tarefas futuras.

## **Sprint 2: Atualização de Estoque e Rastreamento de Localização**

**Período: 15/05/2024 - 28/05/2024**

Objetivo: Implementar a atualização de estoque e o rastreamento de localização dos produtos.

### **Tarefas:**

- Implementação de Atualização de Estoque
- Implementação de Rastreamento de Localização

**Processos:****15/05/2024:**

Reunião de Planejamento da Sprint 2 foi realizada a fim de definir metas e distribuir tarefas entre as equipes.

**16/05/2024 - 27/05/2024:****Implementação de Atualização de Estoque:**

Nesta etapa, foi desenvolvida a funcionalidade de atualização de estoque, O card relativo foi movido para o campo "Em Progresso" em 16/05/2024. Em 20/05/2024 o desenvolvimento da tarefa foi concluído e seu card movido para "Concluído".

**Implementação de Rastreamento de Localização:**

Nesta fase, a equipe desenvolveu a funcionalidade de rastreamento. O card respectivo à funcionalidade foi movido para "Em Progresso" em 21/05/2024 e, posteriormente, movido para "Concluído" em 27/05/2024.

**28/05/2024:**

A equipe SCRUM, junto ao cliente, realizou a Revisão da Sprint 2 para avaliar o que foi concluído, discutir desafios e planejar melhorias.

**Relatório da Sprint 2**

Data de Início: 15/05/2024

Data de Término: 28/05/2024

**Tarefas Concluídas:**

- Implementação de Atualização de Estoque
- Implementação de Rastreamento de Localização

**Desafios:**

- Integração das novas funcionalidades com a estrutura existente.
- Otimização do código para melhor desempenho.



**Sucessos:**

- Atualização de estoque funcionando conforme esperado.
- Rastreamento de localização implementado e integrado com sucesso.

**Melhorias para Próxima Sprint:**

- Melhor documentação do código.
- Realizar mais testes automatizados.

**Sprint 3: Relatórios****Período: 29/05/2024 - 11/06/2024**

Objetivo: Desenvolver funcionalidades de geração de relatórios e criar a tabela verdade.

**Tarefas:**

- Implementação de Relatórios
- Criação da Tabela Verdade

**Processo****29/05/2024:**

Reunião de Planejamento da Sprint 3: Definir metas e distribuir tarefas.

**30/05/2024 - 10/06/2024:****Implementação de Relatórios:**

Etapa dedicada ao desenvolvimento da funcionalidade de geração de relatórios.

O card de referência foi movido para "Em Progresso" em 30/05/2024. Esta etapa foi concluída em 07/06/2024 e seu card movido para "Concluído".

## Revisão da Sprint 3

Foi avaliado tudo o que foi concluído, a equipe discutiu sobre os desafios e planejamento melhorias práticas.

## Relatório da Sprint 3

Data de Início: 29/05/2024

Data de Término: 11/06/2024

Tarefas Concluídas:

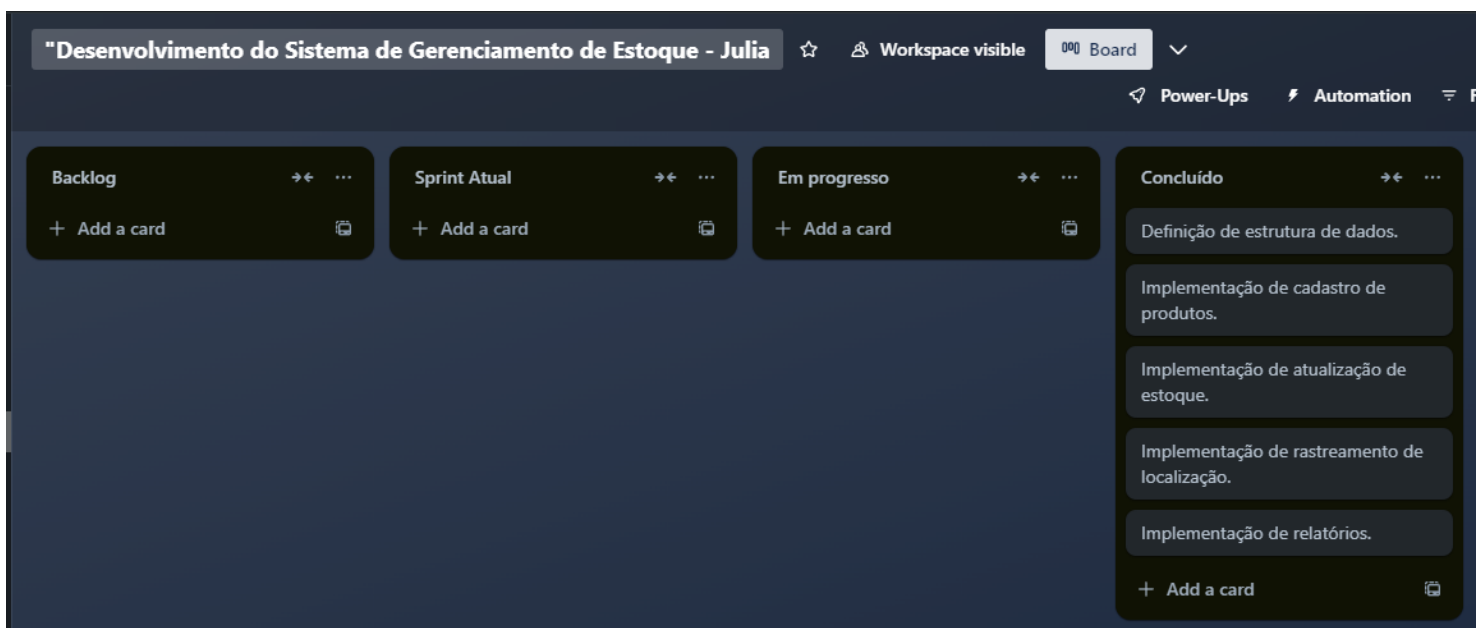
- Implementação de Relatórios
- Avaliação dos processos e procedimentos de teste e desenvolvimento

Desafios:

- Garantir que os relatórios atendam a todas as necessidades de análise.
- Garantir a qualidade do produto e implementação dentro do prazo

Sucessos:

- Relatórios implementados com filtros avançados.
- Acessibilidade e facilidade de uso geral.



## 2.2 - Tabela Verdade

No segundo passo da atividade proposta, foi solicitada a criação de uma tabela verdade sobre o desenvolvimento de um sistema de gerenciamento de estoque. Os requisitos do sistema foram representados através da definição de variáveis booleanas para mostrar as possíveis combinações de valores no sistema.

As variáveis foram definidas conforme exigência da atividade, sendo elas:

- P: Cadastro de Produtos
- E: Atualização de Estoque
- L: Rastreamento de Localização

P	E	L	R	Solução Completa ( $P \wedge E \wedge L \wedge R$ )
F	F	F	F	F
F	F	F	T	F
F	F	T	F	F
F	F	T	T	F
F	T	F	F	F
F	T	F	T	F
F	T	T	F	F
F	T	T	T	F
T	F	F	F	F
T	F	F	T	F
T	F	T	F	F
T	F	T	T	F
T	T	F	F	F
T	T	F	T	F
T	T	T	F	F
T	T	T	T	T

A coluna “Solução Completa” será verdadeira (T) apenas quando todas as variáveis (P, E, L, R) forem verdadeiras, ou seja, todos os requisitos forem atendidos.

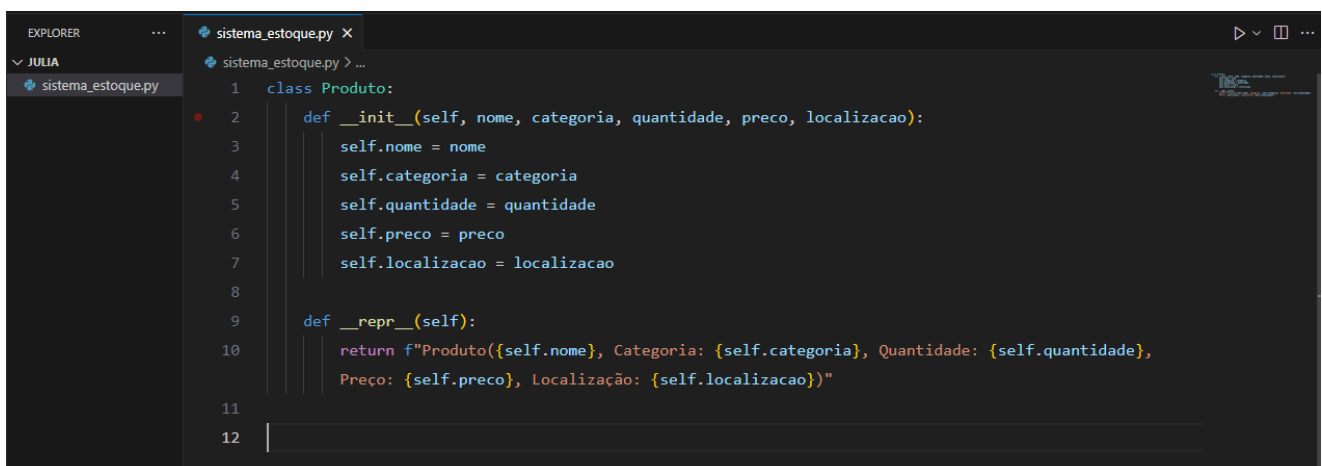
## 2.3 Funções

Neste subcapítulo, foram desenvolvidas funções para definição de estrutura de dados. A linguagem Python foi escolhida por ser altamente eficaz neste tipo de trabalho, além da maior facilidade de desenvolvimento.

### Definição de Estruturas de Dados:

#### Produto

Cada produto contém informações como nome, categoria, quantidade em estoque, preço e localização no depósito. O produto é representado por uma classe, como na amostra a seguir:

A screenshot of a code editor window titled 'sistema\_estoque.py'. The editor shows the definition of a Python class named 'Produto'. The class has two methods: '\_\_init\_\_' and '\_\_repr\_\_'. The '\_\_init\_\_' method takes six parameters: 'self', 'nome', 'categoria', 'quantidade', 'preco', and 'localizacao', and assigns them to instance variables. The '\_\_repr\_\_' method returns a string representation of the object in a dictionary-like format. The code is as follows:

```
1 class Produto:
2     def __init__(self, nome, categoria, quantidade, preco, localizacao):
3         self.nome = nome
4         self.categoria = categoria
5         self.quantidade = quantidade
6         self.preco = preco
7         self.localizacao = localizacao
8
9     def __repr__(self):
10        return f"Produto({self.nome}, Categoria: {self.categoria}, Quantidade: {self.quantidade},
11        Preço: {self.preco}, Localização: {self.localizacao})"
12
```

Foi definida uma nova classe chamada **Produto**. O método `__init__`, que é o construtor da classe, é chamado automaticamente quando um objeto da classe é criado, neste caso são nome, categoria, quantidade, preço e localização.

O parâmetro `self` refere-se à instância da classe atual. Ele é necessário para acessar as variáveis e métodos de um objeto.

Desta forma, qndo um objeto da classe for criado, o método `__init__` será automaticamente chamado e os valores fornecidos ("Notebook", "Eletrônicos" etc.) serão atribuídos aos atributos do objeto **produto1**.

Atributos do Objeto: cada instância de **Produto** terá seus próprios valores para nome, categoria etc.

Método `__repr__`: Define como o objeto será exibido quando impresso, retornando uma string que mostra todos os atributos de forma legível.

## Algoritmos de Cadastro e Consulta

```

12 # Função para cadastrar produtos
13 estoque = []
14
15 def cadastrar_produto(nome, categoria, quantidade, preco, localizacao):
16     produto = Produto(nome, categoria, quantidade, preco, localizacao)
17     estoque.append(produto)
18     return f"Produto {nome} cadastrado com sucesso."
19
20 # Função para consultar produtos

```

## Algoritmos de Movimentação

### Função para registrar entrada de produtos

```

# Função para registrar entrada de produtos

movimentacoes = []

from datetime import datetime

def registrar_entrada(nome, quantidade):
    produto = consultar_produto(nome)
    if isinstance(produto, Produto):
        produto.quantidade += quantidade
        movimentacao = Movimentacao(produto, 'entrada', quantidade, datetime.now())
        movimentacoes.append(movimentacao)
        return f"Entrada registrada para {nome}, nova quantidade: {produto.quantidade}."
    return produto # Retorna a mensagem de erro da consulta

```

## Função para registrar saída de produtos

```
# Função para registrar saída de produtos

def registrar_saida(nome, quantidade):
    produto = consultar_produto(nome)
    if isinstance(produto, Produto):
        if produto.quantidade >= quantidade:
            produto.quantidade -= quantidade
            movimentacao = Movimentacao(produto, 'saída', quantidade, datetime.now())
            movimentacoes.append(movimentacao)
            return f"Saída registrada para {nome}, nova quantidade: {produto.quantidade}."
        else:
            return f"Quantidade insuficiente em estoque para {nome}."
    return produto # Retorna a mensagem de erro da consulta
```

## Relatórios e Consultas

### Função para gerar relatório de produtos com estoque baixo

```
# Função para gerar relatório de produtos com estoque baixo

def gerar_relatorio_estoque_baixo(limite):
    produtos_baixo_estoque = [p for p in estoque if p.quantidade <= limite]
    if produtos_baixo_estoque:
        return produtos_baixo_estoque
    return "Nenhum produto com estoque baixo."
```

### Função para consultar o histórico de movimentações

```
# Função para consultar o histórico de movimentações

def consultar_historico_movimentacoes(nome):
    historico = [m for m in movimentacoes if m.produto.nome == nome]
    if historico:
        return historico
    return f"Sem movimentações registradas para {nome}."
```

## Saídas Esperadas

Após cadastrar os produtos, será possível ver a listagem correta do inventário. As movimentações de entrada e saída irão alterar a quantidade do estoque.

O relatório de estoque baixo mostrará produtos com quantidades menores ou

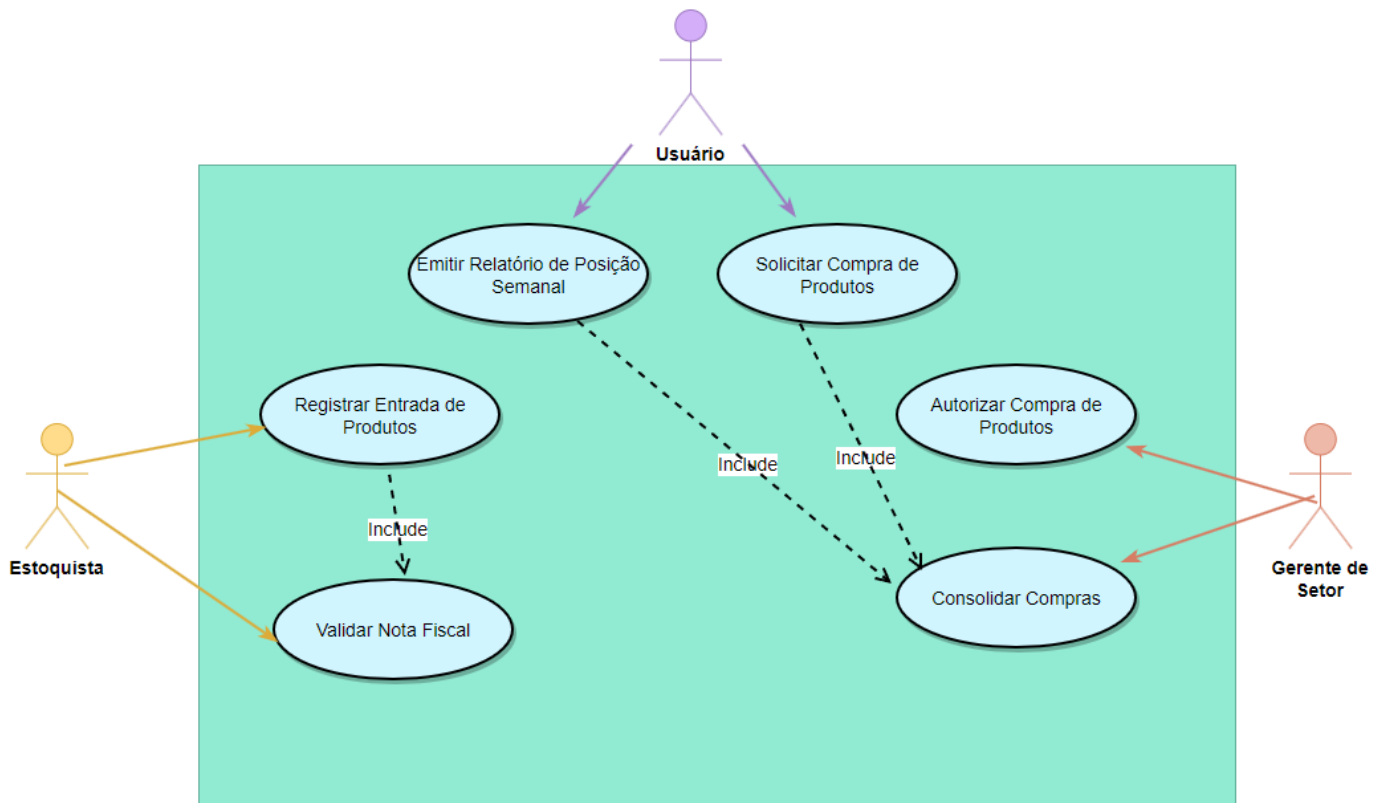
iguais ao limite especificado.

O histórico de movimentações mostrará todas as entradas e saídas de um produto específico.

Essas funções cobrem as funcionalidades básicas de cadastro, consulta, movimentação e relatórios de produtos para o sistema de gerenciamento de estoque.

## 2.4 Diagrama de Casos de Uso

O diagrama foi criado no aplicativo online Visual Paradigm, contendo os seguintes atores e casos de uso



### Detalhamento:

#### Atores:

**Estoquista:** Conectado aos casos de uso "Registrar Entrada de Produtos" e "Validar Nota Fiscal".

**Usuário:** Ligado aos casos de uso "Emitir Relatório de Posição Semanal" e "Solicitar Compra de Produtos".

**Gerente de Setor:** Relacionado corretamente aos casos de uso "Autorizar Compra de Produtos" e "Consolidar Compras".



Casos de Uso: As elipses que representam os casos de uso estão bem colocadas e nomeadas de acordo com as funcionalidades descritas:

Registrar Entrada de Produtos

Validar Nota Fiscal

Emitir Relatório de Posição Semanal

Solicitar Compra de Produtos

Consolidar Compras

Autorizar Compra de Produtos

Para realizar os casos “Emitir Relatório de Posição Semanal” e “Solicitar Compra de Produtos”, realizados pelo usuário, fica estabelecida a dependência do caso “Consolidar Compras”, realizada pelo gerente de setor.

Este tipo de dependência ocorre, também, entre os casos “Registrar Entrada de Produtos” e “Validar Nota Fiscal”, realizadas pelo estoquista.

### 3 CONCLUSÃO

Este trabalho multidisciplinar evidencia e simula o cotidiano no desenvolvimento de um software de gerenciamento de estoque. Durante o seu desenvolvimento, foi possível utilizar várias ferramentas em cada etapa de sua construção, proporcionando contato com várias tecnologias úteis para modelar, organizar, definir e entregar o produto dentro do prazo estipulado.

A utilização de boas práticas de desenvolvimento, como a definição clara de requisitos, a aplicação de metodologias ágeis e a criação de diagramas visuais, tornou o processo mais organizado e previsível. Além disso, o uso da linguagem Python para a implementação de algoritmos básicos do sistema reforçou a importância de uma linguagem de programação versátil e poderosa para a automação de processos complexos.

Por fim, o projeto ilustra como a combinação de planejamento, modelagem e implementação prática cria a base para um sistema de gerenciamento de estoque eficiente, que pode ser continuamente aprimorado e adaptado conforme novas necessidades surgirem.

## 4 REFERÊNCIAS

### Sites:

PYTHON. Python documentation. Disponível em: <https://www.python.org/doc/>. Acesso em: 27 set. 2024.

TRELLO. Trello Boards. Disponível em: <https://trello.com/u/lucaslima75330318/boards>. Acesso em: 27 set. 2024.

VISUAL PARADIGM. Diagramas Online. Disponível em: <https://online.visual-paradigm.com/pt/>. Acesso em: 27 set. 2024.

DEVMEDIA. O que é UML e diagramas de caso de uso: introdução prática à UML. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>. Acesso em: 27 set. 2024.

TODA MATÉRIA. Tabela Verdade. Disponível em: <https://www.todamateria.com.br/tabela-verdade/>. Acesso em: 27 set. 2024.

### Livros:

GIDO, Jack; CLEMENTS, Jim; BAKER, Rose. Gestão de projetos. Tradução da 7ª ed. norte-americana. São Paulo: Cengage, 2018.

LARSON, Erik W.; GRAY, Clifford F. Gerenciamento de projetos. Porto Alegre: AMGH, 2016.

LAYTON, Mark C.; OSTERMILLER, Steven J. Gerenciamento Ágil de Projetos para Leigos. Rio de Janeiro: Alta Books, 2019.

CAMARGO, Robson Alves de; RIBAS, Thomaz. Gestão ágil de projetos. São Paulo: Saraiva Educação, 2019.

MAXIMIANO, Antonio Cesar A.; VERONEZE, Fernando. Gestão de Projetos: Preditiva, Ágil e Estratégica. Barueri: Atlas, 2022.

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de software: uma abordagem profissional. Porto Alegre: AMGH, 2021.