

**CENTRO PAULA SOUZA  
FACULDADE DE TECNOLOGIA DE FRANCA  
“Dr. THOMAZ NOVELINO”**

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**FERNANDA SILVA GOMES  
GABRIEL MARIANO TEODORO**

**SOFTWARE PARA GERENCIAMENTO DE ESTOQUE NA INDÚSTRIA  
DE SORVETES**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Alberto Lucas

**FRANCA/SP**

**2024**

# SOFTWARE PARA GERENCIAMENTO DE ESTOQUE NA INDÚSTRIA DE SORVETES

FERNANDA SILVA GOMES<sup>1</sup>  
GABRIEL MARIANO TEODORO<sup>2</sup>

## Resumo

Este projeto teve seu início em fevereiro de 2023, devido a obrigatoriedade de se aplicar os conteúdos estudados no curso. Sendo assim, foi através de uma conversa entre amigos e familiares que surgiu a questão problema deste projeto, qual seja, a necessidade do gerenciamento de estoque de uma indústria de sorvetes. A empresa 'cliente' deste projeto realizava a sua produção e vendas sem nenhum controle, ou às vezes com algumas anotações escritas em algum papel, portanto, surgiu a ideia em solucionar este problema com a criação de um software gerenciador de estoque, que substituirá os processos manuais através de recursos tecnológicos para gerenciamento de produtos e a produção. Algumas funcionalidades foram desenvolvidas visando aprimorar o controle de estoque e proporcionar mais detalhes sobre a produção. Para implementar estas funcionalidades foram utilizadas as tecnologias do React.JS, C# e o banco de dados MySQL. Esta solução visa melhorar significativamente a eficiência operacional da empresa, conferindo-lhe um melhor controle sobre seus recursos e fortalecendo sua posição competitiva no mercado.

**Palavras-chaves:** Estoque. Eficiência Operacional. Gerenciamento. Software. Sorvetes.

## Abstract

*This project began in February 2023, due to the obligation to apply the content studied in the course. Therefore, it was through a conversation between friends and family that the problem of this project arose, namely, the need to manage inventory in an ice cream industry. The 'client' company of this project carried out its production and sales without any control, or sometimes with some notes written on paper, therefore, the idea of solving this problem came up with the creation of inventory management software, which will replace manual processes, offering a computerized system for product and production management. Some features were developed to improve inventory control and provide more details about production. To implement these functionalities, React.JS, C# and the MySQL database technologies were used. This solution aims to significantly improve the company's operational efficiency, giving it better control over its resources and strengthening its competitive position in the market.*

**Keywords:** Ice creams. Management. Operational efficiency. Software. Stock.

---

<sup>1</sup> Graduanda em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: [fernanda.gomes11@fatec.sp.gov.br]

<sup>2</sup> Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: [gabriel.teodoro@fatec.sp.gov.br].

## 1-Introdução

O projeto tem como objetivo apresentar uma proposta de um sistema para uma indústria no ramo de sorvetes, no qual serão informatizados alguns dados de seu estoque e produtos acabados, visando a otimização de sua eficiência operacional e controle de estoque.

Tendo como embasamento para sua criação os pontos fracos identificados nos artefatos criados e aplicados através da engenharia de software como elicitação, observação e brainstorming.

Dentre os pontos fracos, foi identificada a carência de um controle de estoque e produção, no qual é de extrema importância para qualquer empresa, podendo afetar diretamente sua saúde financeira se não existir ou se não for realizado com eficiência.

Diante dessas informações, o objetivo do projeto é o desenvolvimento de um software de controle de estoque que substituirá os processos manuais realizados atualmente.

### 1.1-Termo de Abertura do Projeto (TAP)

De acordo com o Guia PMBOK, o TAP é um dos documentos que faz parte de todo o ciclo de vida de um projeto. Nele, irá constar os objetivos e benefícios do projeto, estudos de viabilidade, as restrições de prazo e orçamento (Artia, 2024, online).

É a partir do TAP que é permitida a autorização de inicialização do projeto.

O quadro 1 apresenta de forma parcial, algumas informações contempladas neste documento tão importante para o sucesso deste projeto.

**Quadro 1:** Termo de Abertura do Projeto (TAP)

Título do projeto: Software para gerenciamento de estoque na indústria de sorvetes
Descrição do projeto: Atualmente, os responsáveis pela produção enfrentam a tarefa de verificar manualmente a quantidade de produtos em estoque em cada câmara fria. O único dado informatizado é a entrada de produtos, registrada manualmente pelo escritório em uma planilha CSV. Este projeto tem como objetivo desenvolver um software de gestão de estoque para otimizar o processo de entrada e saída de produtos na câmara fria. O software permitirá que o administrador crie, ajuste, remova e visualize os produtos. Além disso, os usuários da produção poderão registrar a entrada dos produtos, incluindo quantidade e categoria, enquanto os usuários do estoque poderão efetuar a saída de produtos para a venda. O sistema também contará com recursos adicionais, como alertas automatizados para quando os produtos estão em baixa no estoque. A meta é aumentar a eficiência operacional em 40%, reduzir o desperdício em 30% e aprimorar o controle do estoque em 30%.
Objetivos e metas: A solução do ponto fraco escolhido será a criação de um software para controle de estoque, interligado a um banco de dados (desenvolvido no banco MySQL) para que os usuários tenham acesso na quantidade atualizada de produto em estoque.

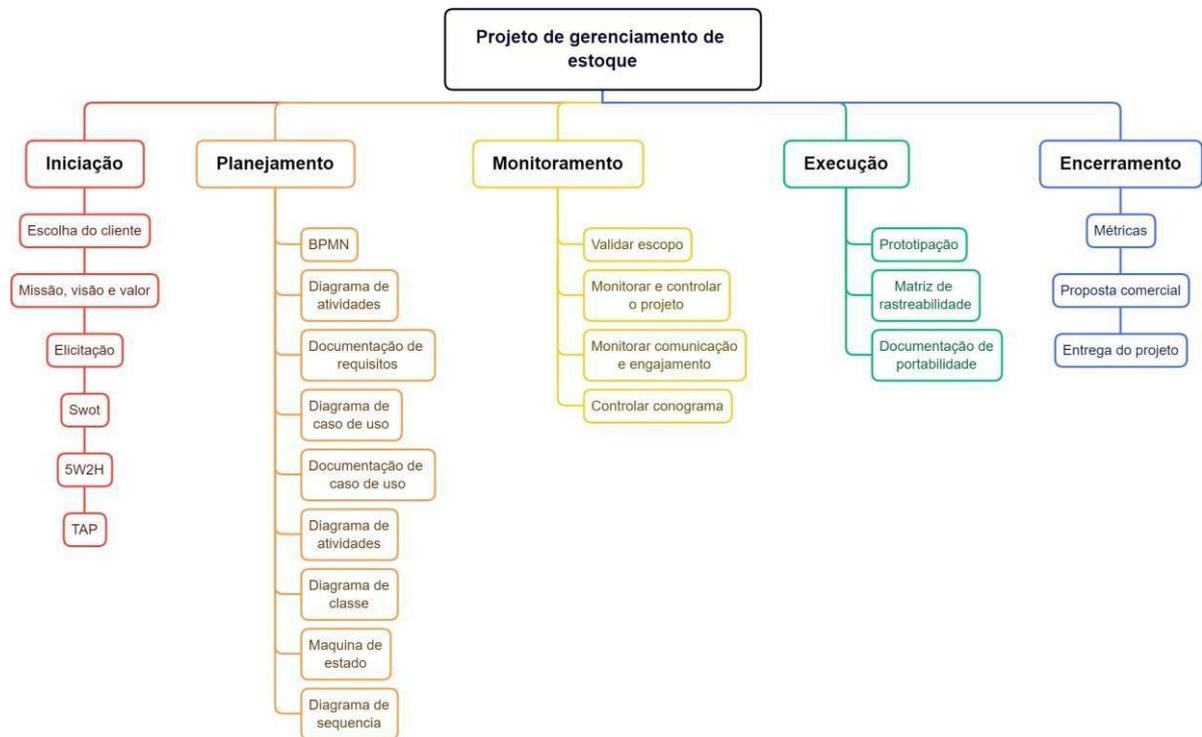
**Fonte:** os autores

## 1.2- Estrutura Analítica do Projeto (EAP)

A EAP é utilizada na organização do escopo de um projeto e a hierarquia das atividades a serem concluídas (Artia, 2024, online).

A figura 1 representa a EAP deste projeto e segue os parâmetros do PMBOK.

**Figura 1:** Estrutura Analítica do Projeto



**Fonte:** os autores

Neste projeto as atividades de iniciação e encerramento são dependentes uma das outras, sendo necessária a finalização de uma para que a outra possa ser iniciada. Já as demais do planejamento, monitoramento e execução podem ser realizadas de forma individualizada e as tarefas são divididas.

## 2-Viabilidade do Projeto

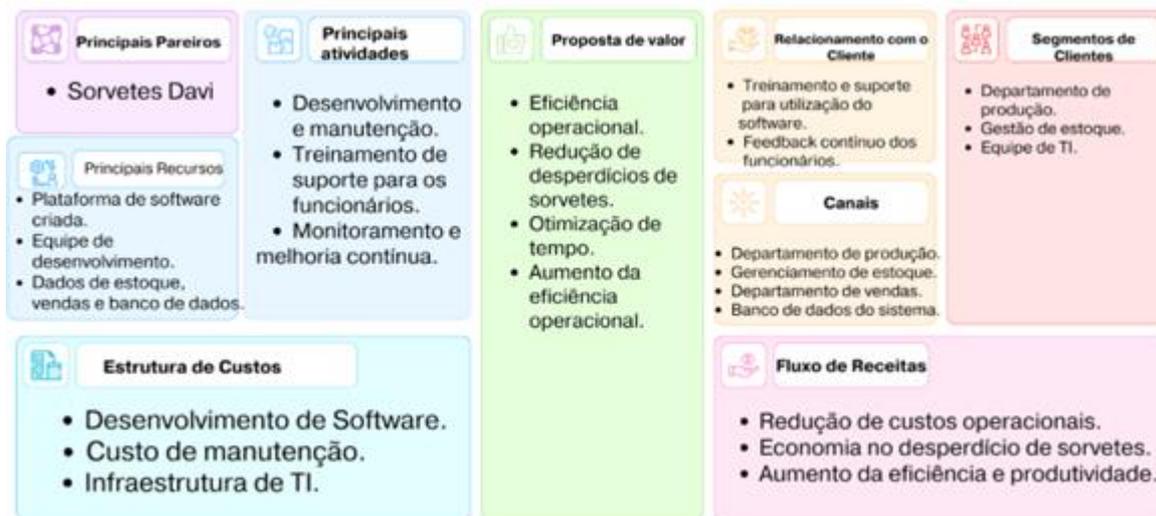
### 2.1-Canvas

O Canvas de Negócio (Business Model Canvas - BMC) tem sido responsável por uma revolução na forma como empreendedores e empresas constroem seus novos negócios. Alexander Osterwalder, o criador do quadro, é o principal responsável por essa transformação.

O Canvas é uma ferramenta que possibilita a criação de representações visuais de modelos de negócios, sejam eles novos ou existentes (Sebrae, 2024, online). A figura 2 ilustra o modelo de negócio do software de gerenciamento e seu papel fundamental na

transformação da maneira como empreendedores e empresas desenvolvem seus novos empreendimentos.

**Figura 2: Business Model Canvas**



Fonte: os autores

## 2.2-Matriz SWOT

Segundo o site AGENDOR, a matriz SWOT é uma ferramenta de gestão estratégica utilizada em projetos, na qual funciona como um “raio-X” da empresa, permitindo uma análise aprofundada da empresa, destacando pontos fortes, fraquezas, oportunidades e ameaças (Agendor, 2024, online).

Para iniciação do produto foca-se nas fraquezas a serem solucionadas.

A figura 3 representa a SWOT deste projeto, retratando apenas os atributos das fraquezas.

**Figura 3: Matriz SWOT**

FRAQUEZAS
Falta de servidor para melhor funcionamento do sistema.
A produção não possui controle de estoque.
Vendas para distribuidores não têm acesso ao estoque.
A loja não registra a retirada de produto.
Planejamento de produção não é informatizado, apesar de ter estatísticas de vendas.

**Fonte:** os autores

Após análise abrangente da situação atual da empresa, a tomada de decisão para o produto deste projeto foi na fraqueza: a produção não possui controle de estoque, visto que o controle de estoque é importante em qualquer organização em produção de produtos físicos.

Com um controle de estoque em funcionamento, outras demandas serão diretamente afetadas, como: planejamento de produção e compras de matéria-prima, redução de custos, controle de desperdícios, eficiência operacional e tomada de decisões.

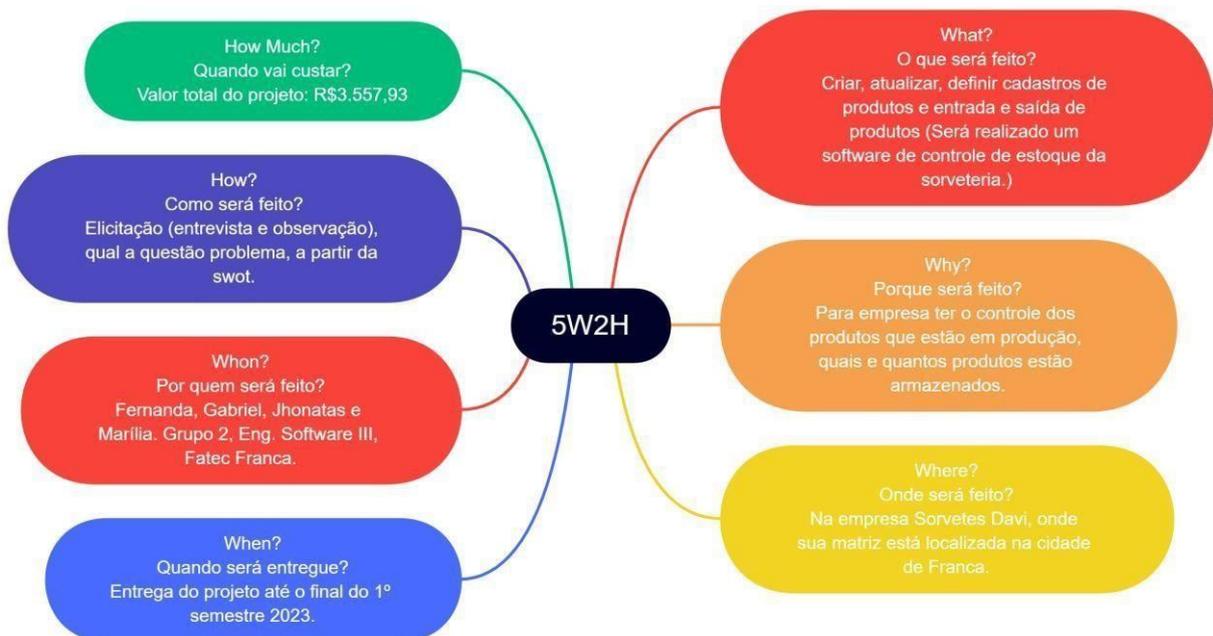
### 2.3-Matriz 5W2H

A matriz 5W2H é uma ferramenta de gestão de projeto que envolve 7 perguntas, tendo a sua sigla em inglês, as perguntas: What? (O que?), Why (Por quê), Where? (Onde?), When? (Quando?), Who? (Por quem?), How? (Como?) e How much? (Quanto?).

Com a utilização desta ferramenta o plano de ação e o foco do projeto ficam cada vez mais claro e há uma diminuição na ociosidade ou tarefas desnecessárias que fogem do escopo do projeto (Treasy, 2024, online).

A figura 4 apresenta o plano de ação deste projeto através de um mapa mental.

**Figura 4:** Matriz 5W2H



**Fonte:** os autores

Com a criação e verificação da matriz 5W2H, pode-se entender qual o foco principal

e inicial para o controle de estoque da empresa e qual o valor médio para a produção do sistema de gerenciamento.

### **3-Levantamento de Requisitos**

#### **3.1 Elicitação**

No desenvolvimento do sistema operacional, é crucial compreender as necessidades e expectativas dos usuários para o sucesso de um projeto. A elicitação de requisitos é o processo de levantamento e a documentação dessas necessidades, são fundamentais nessa jornada (Logap, 2024, online).

Para coleta das informações foram utilizadas técnicas de elicitação de requisitos, sendo utilizados os métodos de conversação, entrevistas, questionário e observação.

Após análise minuciosa dos dados obtidos através dos métodos, pode-se identificar os requisitos necessários para otimização do trabalho através de um software.

Iniciada com a observação, criamos o cenário de como é realizado o processo atual na empresa de estocagem e produção, sendo a maior parte manualmente.

Conversando com funcionários podemos saber mais sobre a empresa, não apenas em sua parte informatizada, mas sua missão, visão e valores (ao final deste documento no Anexo 1).

Posteriormente foram utilizados questionários (todas as perguntas encontram-se no final deste documento no Apêndice 1).

Primeiro de uma forma generalizada, para conhecer os processos que são informatizados e os que poderiam ser, a disponibilidades de computadores e infraestrutura, assim foram feitas perguntas como: “Há estatística para a quantidade de produtos vendidos?” e “É feito o backup dos dados da empresa?”

Obtida as respostas dessas perguntas, foi verificado a necessidade de novas perguntas, desta vez mais direcionadas aos problemas que o sistema poderia resolver, por exemplo: “Qual a forma de armazenamento de estoque?” e “Como é a entrada e saída de produto?”.

Com os resultados dessa elicitação, pode-se saber quais os requisitos e começar o esboço em diagramas e prototipação do projeto.

#### **3.2-Business Process Model and Notation (BPMN)**

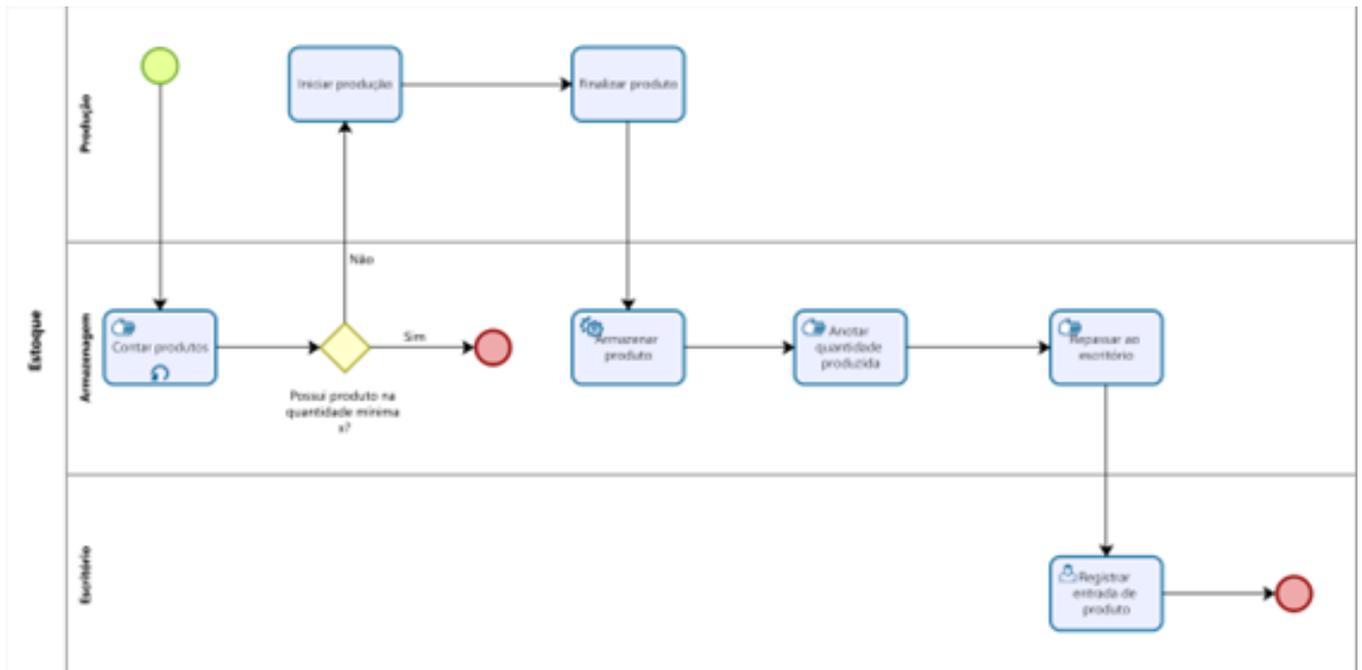
O BPMN é a modelagem processual ou sistêmica de como é ou deve ser uma empresa e seu sistema, utilizado para facilitar o entendimento do usuário sobre seu funcionamento.

A utilização do BPMN visa entender o cenário atual, para que assim sejam descobertos novos processos e otimizações que podem ser realizadas.

Ao realizar o esboço do BPMN, pode-se verificar na figura 5 que atualmente 95% do processo de estoque e produção é feito manualmente, desde a contagem do produto ao controle de produção, inserido as informações manualmente.

A figura 5 ilustra o funcionamento atual da empresa 'cliente'.

**Figura 5: BPMN - funcionamento atual**

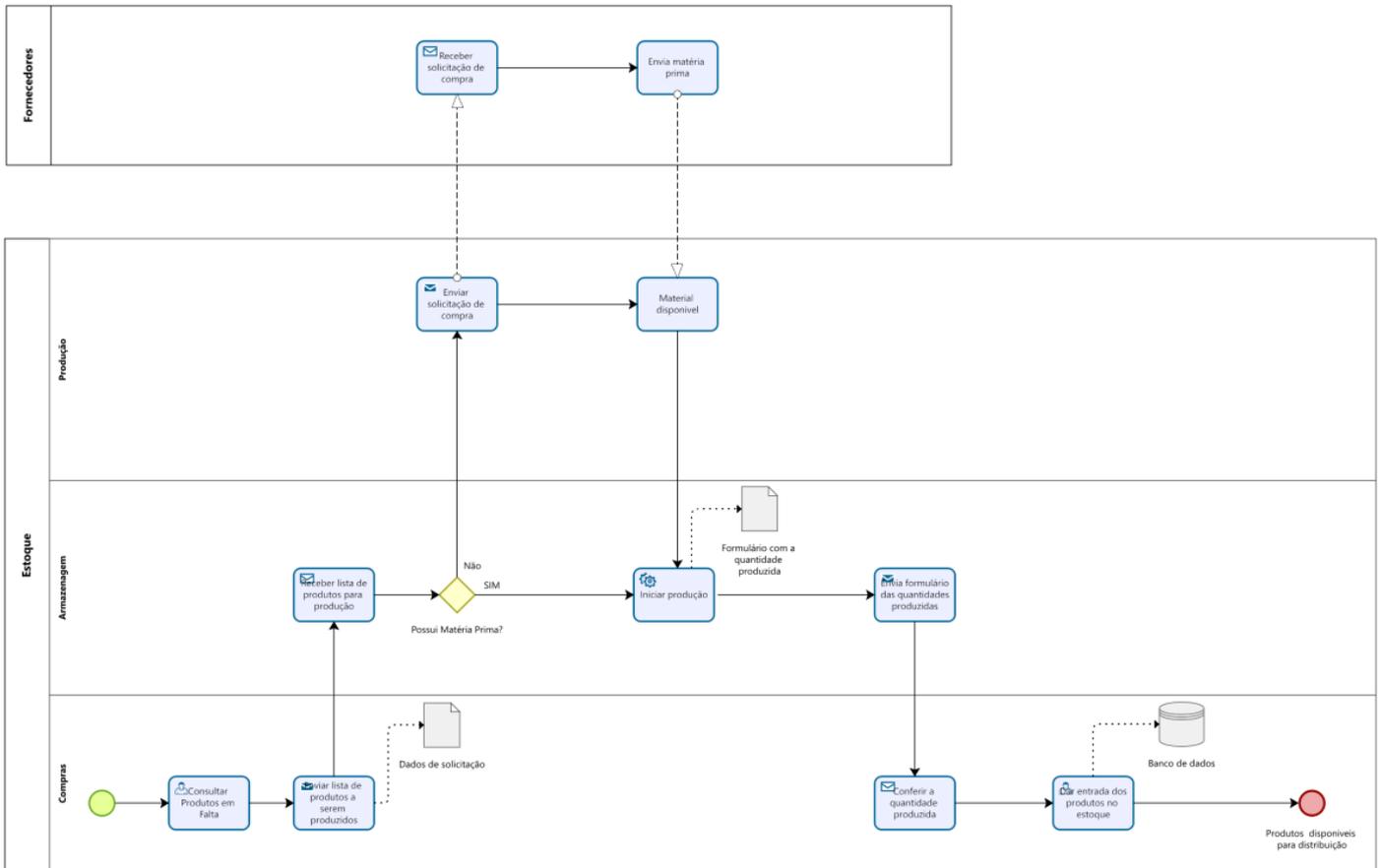


**Fonte:** os autores

A figura 6 ilustra os processos da empresa 'cliente'.

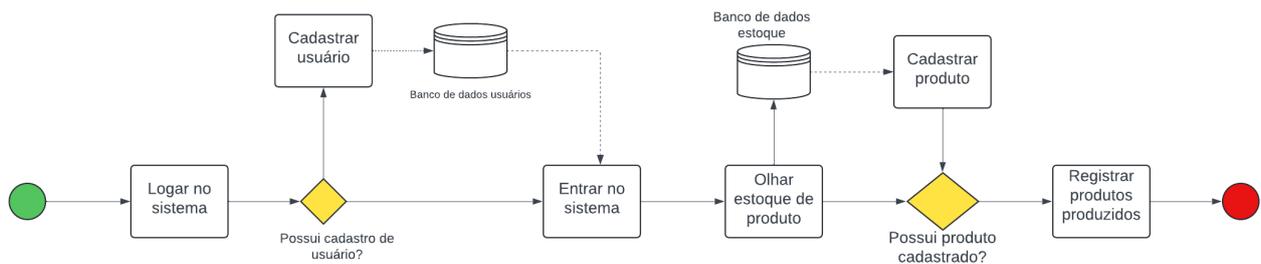
**Figura 6: BPMN Processual**

Fonte: os autores



A figura 7 ilustra o BPMN sistêmico elaborado pelos autores deste projeto.

Figura 7: BPMN Sistêmico



Fonte: os autores

Desta forma, podemos verificar melhorias a serem implementadas, onde o programa economizará tempo dos processos manuais.

A entrada dos produtos após a finalização da produção será feita sistemicamente através de relatórios, assim, com a movimentação de saída de produto no estoque o sistema notificará quando cada produto atingir a quantidade mínima pré-estabelecida fazendo com que a produção deste seja iniciada e siga seu loop de produção física e informação de

quantidade ao estoque.

### 3.3-Requisitos Funcionais

Os requisitos funcionais são todas as funcionalidades, necessidades ou características esperadas que o software deverá apresentar.

Para este projeto alguns dos requisitos funcionais são: efetuar login, controlar acesso, registrar e controlar estoque, emitir relatório, prever demanda, programar e gerenciar produção e controlar insumo.

No quadro 2 temos a representação de quatro requisitos funcionais deste projeto.

**Quadro 2: Requisitos Funcionais**

ID: RF001	Nome: Efetuar login
Descrição: O sistema deve permitir que os usuários acessem as funcionalidades.	
Categoria (x) Evidente ( ) Oculto	Prioridades (x) Essencial ( ) Importante ( ) Desejável
Informações: login e senha já criados previamente.	
Regra de negócio: RN007	
ID: RF002	Nome: Controlar acesso
Descrição: O sistema deve liberar as funcionalidades de acordo com o perfil logado.	
Categoria ( ) Evidente (x) Oculto	Prioridades (x) Essencial ( ) Importante ( ) Desejável
Informações: login vinculado há um perfil.	
Regra de negócio: RN007	
ID: RF003	Nome: Registrar estoque
Descrição: O sistema deve permitir registrar novos produtos e manter o que já estão, com detalhes e quantidade em estoque	
Categoria ( ) Evidente (x) Oculto	Prioridades ( ) Essencial (x) Importante ( ) Desejável
Informações: nome, categoria, tipo, ingredientes necessários, quantidade mínima e outros dados do produto.	
Regra de negócio: RN004	
ID: RF004	Nome: Contar estoque
Descrição: O sistema deve permitir uma contagem periódica e visualização da mesma, para comparação da quantidade registrada e o estoque físico.	
Categoria (x) Evidente ( ) Oculto	Prioridades ( ) Essencial (x) Importante ( ) Desejável
Informações: dados de produtos já cadastrados e sua quantidade atualizada em estoque.	
Regra de negócio: RN004	
ID: RF005	Nome: Cadastro de usuário
Descrição: O sistema deve permitir que o usuário administrador realize o cadastro de outros usuários.	
Categoria (X) Evidente ( ) Oculto	Prioridades (x)Essencial ( )Importante ( ) Desejável
Informações: login do usuário.	
Regra de negócio: RN005	
ID: RF005	Nome: Liberar permissões para usuário

Descrição: o sistema deve permitir que o usuário administrador libere permissões de menus para outros usuários.	
Categoria: <input checked="" type="checkbox"/> Evidente <input type="checkbox"/> Oculto	Prioridades <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
Informações: nome do menu e liberado.	
Regra de negócio: RN006	
ID: RF005	Nome: Mostrar produtos em baixa quantidade
Descrição: o sistema deve mostrar os produtos em baixa quantidade.	
Categoria: <input checked="" type="checkbox"/> Evidente <input type="checkbox"/> Oculto	Prioridades <input type="checkbox"/> Essencial <input checked="" type="checkbox"/> Importante <input type="checkbox"/> Desejável
Informações: nome e quantidade do produto.	

**Fonte:** os autores

### 3.4-Requisitos Não-Funcionais

Os requisitos não funcionais exemplificam de qual maneira o sistema deverá fazer cada função estabelecida. Isso inclui seu desempenho, disponibilidade, segurança e integridade.

A partir disto, os requisitos não funcionais estabelecidos foram: validar credenciais, recebimento de insumo, acompanhamento, responsividade e desenvolvimento.

#### Quadro 3: Requisitos Não-Funcionais

ID: RNF001	Nome: Validar credencial
Descrição: O sistema deverá autenticar os dados informados no login se está correto ou existe.	
Categoria <input type="checkbox"/> Evidente <input checked="" type="checkbox"/> Oculto	Prioridades <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
Tipo: Organizacional	
Tipo: Entrega	
ID: RNF002	Nome: Responsividade
Descrição: O sistema deverá ser responsivo a diferentes tipos de dispositivos	
Categoria <input type="checkbox"/> Evidente <input checked="" type="checkbox"/> Oculto	Prioridades <input type="checkbox"/> Essencial <input checked="" type="checkbox"/> Importante <input type="checkbox"/> Desejável
Tipo: Portabilidade	
ID RNF003	Nome: Desenvolvimento
Descrição: O sistema deverá ser desenvolvido em C# (backend) + React (frontend).	
Categoria <input type="checkbox"/> Evidente <input checked="" type="checkbox"/> Oculto	Prioridades <input type="checkbox"/> Essencial <input type="checkbox"/> Importante <input checked="" type="checkbox"/> Desejável
Tipo: Implementação	
ID RNF004	Nome: Criptografar senha
Descrição: O sistema deverá criptografar, como segurança, as senhas dos usuários.	
Categoria <input type="checkbox"/> Evidente <input checked="" type="checkbox"/> Oculto	Prioridades <input type="checkbox"/> Essencial <input type="checkbox"/> Importante <input checked="" type="checkbox"/> Desejável
Tipo: Organizacional	

**Fonte:** os autores

### 3.5-Regras do Negócio

As regras do negócio são diretrizes e restrições que implicam em um negócio, ou seja, definem como uma organização opera e toma decisões. Deste modo, as regras são aplicadas no ambiente e tem como objetivo principal a execução das atividades de modo correto dentro das políticas, regulamentação e saneamento estabelecido.

As regras de negócio foram elaboradas para serem aplicadas na utilização do sistema desenvolvido.

O quadro 4 apresenta as regras do negócio da empresa 'cliente'.

**Quadro 4:** Regras de Negócio

RN001 – Registro divergente
Descrição: O sistema deverá registrar a quantidade de produtos produzidos diferenciando o tipo de cada um.
RN003 - Login
Descrição: O sistema dará acesso ao histórico de registro de entrada e saída de produtos se o usuário estiver logado.

**Fonte:** os autores

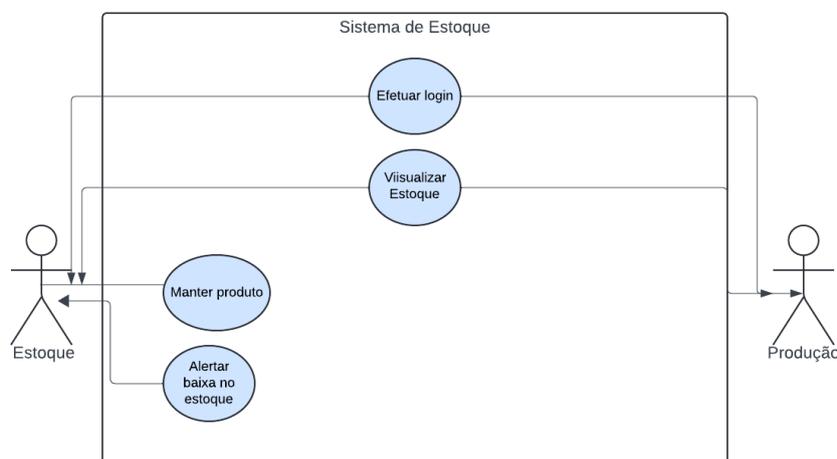
### 3.6-Casos de Uso

Para descrição e visualização dos atores em um sistema e seus usuários é utilizado o diagrama de Caso de Uso, no qual possui atores, fluxo principal e alternativo, pré e pós condições, inclusões e extensões (Rodojacto, 2024, online).

Desta forma, a utilização do Caso de Uso na execução de um projeto é de extrema importância, pois a partir dele podemos entender os requisitos que o sistema possui, a comunicação e objetivos de cada funcionário e também serve de guia para o desenvolvimento de um sistema eficaz.

A figura 8 ilustra o Diagrama de Caso de Uso deste projeto.

**Figura 8:** Caso de Uso



**Fonte:** os autores

O caso de uso do controle de estoque, possui 2 atores, no qual eles compartilham dois casos em comum, efetuar login e visualizar estoque, os outros são separados de acordo com a sua autorização de acesso.

Ao estoque seus casos são manter produto, alertar a baixa no estoque e emitir relatório de produção, já na produção será iniciar a produção com extensão para solicitar a matéria-prima e emitir relatório de entrada de produto.

### 3.6.1-Documentação de Caso de Uso

Após a criação do diagrama de caso de uso é feita a documentação, que trata de todos os detalhes dos atores envolvidos no sistema, bem como o cenário principal e alternativo, a descrição e as condições.

**Quadro 5 – Documentação de Caso de Uso**

Caso de Uso – Efetuar Login	
ID	UC 001
Descrição	Este caso de uso tem por objetivo efetuar o login dos usuários do sistema.
Ator Primário	Usuários (produção e estoque)
Pré-condição	Ter um login e senha já cadastrados.
Cenário Principal	O use case inicia quando usuário abre o sistema de estoque. O sistema exibe a página de login, que contém campos para o nome de usuário e senha. O usuário insere seu login e senha nos campos apropriados. O usuário clica no botão “Entrar”. O sistema verifica as credenciais informadas em relação aos dados armazenados. Caso as credenciais constarem no banco de dados e estiverem corretas, o sistema autentica o usuário e libera o acesso. O sistema exibe a página inicial. O use case é encerrado com o usuário logado no sistema.
Pós-condição	O usuário está autenticado no sistema e tem acesso às funcionalidades permitidas de acordo com o seu nível de permissão.
Cenário Alternativo	A1 - Informações erradas A1.1 - No passo 6 do cenário principal, caso seja inserido alguma informação errada no login ou senha o sistema exibirá uma mensagem de dados inválidos. A1.2 - O usuário informa os dados corretamente. A2 - Usuário não encontrado A2.1 - Verificar se o login está correto A2.2 - Em casos de usuário correto, verificar se o usuário existe
Inclusão	Não possui
Extensão	Não possui
Caso de Uso – Manter Produto	
ID	UC 002
Descrição	Este caso de uso tem por objetivo cadastrar, excluir, editar e visualizar os produtos do catálogo.
Ator Primário	Administrador

Pré-condição	O administrador precisa estar logado no perfil “administrador”, no qual ele terá acesso às funcionalidades.
Cenário Principal	<p>O use case se inicia quando o usuário seleciona a opção “Cadastro de produto”</p> <p>O sistema abre a tela de gerenciamento de produtos.</p> <p>O usuário seleciona a opção “Cadastrar Produto”.</p> <p>Se selecionada a opção “Cadastrar Produtos”</p> <p>O sistema exibe um formulário de cadastro de produtos com campos como ....</p> <p>O administrador preenche os campos com as informações do novo produto.</p> <p>O administrador clica no botão “Salvar” para adicionar o produto ao sistema.</p> <p>O sistema valida os dados inseridos e armazena o novo produto no banco de dados.</p> <p>O sistema exibe uma mensagem de confirmação que o produto foi adicionado com sucesso..</p> <p>Se selecionada a opção “Atualizar Produto”:</p> <p>O usuário seleciona o produto desejado.</p> <p>O sistema exibe um formulário com os dados atuais do produto.</p> <p>O administrador atualiza os campos desejados com as novas informações.</p> <p>O administrador clica no botão “Salvar” para adicionar o produto</p>
Pós-condição	Os produtos são adicionados, atualizados ou excluídos de acordo com as ações realizadas pelo usuário.
Cenário Alternativo	<p>A1 - Campos não preenchidos</p> <p>A1.1 - Nos passos 4 ou 5, se algum dos campos obrigatórios não forem preenchidos corretamente, o sistema exibe uma mensagem de erro indicando quais não foram preenchidos.</p> <p>A1.2 - O usuário preenche os dados não preenchidos corretamente.</p>
Inclusão	Não possui
Extensão	Não possui
Caso de Uso – Visualizar Estoque	
ID	UC 003
Descrição	Este caso de uso tem por objetivo visualizar a quantidade de produtos atualizadas no estoque
Ator Primário	Usuários (produção e estoque)
Pré-condição	Estar logado em um perfil com a permissão de visualização.
Cenário Principal	<p>O use case se inicia quando o usuário seleciona a opção “Contagem de estoque”</p> <p>O sistema exibe uma lista de produtos em estoque, com detalhes como nome e quantidade disponível.</p> <p>O usuário pode usar filtros ou pesquisar por um produto específico.</p> <p>O sistema atualiza a exibição de acordo com as seleções feitas pelo usuário.</p> <p>O sistema permite que o usuário retorne à lista de produtos em estoque ou faça outras ações, como adicionar uma nova entrada de estoque.</p>
Pós-condição	<p>O usuário tem visão geral do estoque atual, com os produtos disponíveis e suas quantidades.</p> <p>O usuário pode utilizar as informações para tomadas de decisão, como reposição de estoque e outras.</p>
Cenário Alternativo	<p>A1 - Quantidade divergente</p> <p>A1.1 - Se o usuário detectar que a quantidade do produto está divergente do estoque físico, pode solicitar uma verificação manual do estoque.</p> <p>A1.2 - Confirmada divergência com o estoque físico, poderá ser alterada a quantidade manualmente do produto.</p>
Cenário Alternativo	<p>A1 - Quantidade divergente</p> <p>A1.1 - Se o usuário detectar que a quantidade do produto está divergente do estoque físico, pode solicitar uma verificação manual do estoque.</p> <p>A1.2 - Confirmada divergência com o estoque físico, poderá ser alterada a quantidade manualmente do produto.</p>
Inclusão	Não possui
Extensão	Não possui
Caso de Uso – Alertar Baixo Estoque	

ID	UC 004
Descrição	Este caso de uso tem por objetivo enviar um alerta quando os produtos chegarem em uma quantidade mínima estipulada.
Ator Primário	Estoque
Pré-condição	Estar logado em um perfil com a permissão de visualização. Ter definido a quantidade mínima de um produto no UC002.
Cenário Principal	O use case se inicia quando o usuário seleciona a opção “Alertas” O sistema monitora constantemente as quantidades de produtos em estoque. O sistema verifica regularmente se a quantidade de um produto específico está abaixo do nível mínimo definido. Se a quantidade do produto atingir ou cair abaixo do nível mínimo, o sistema gera um alerta de estoque baixo. O sistema exibe o alerta para o usuário, destacando o produto específico, sua quantidade atual e qual a quantidade mínima definida.
Pós-condição	O administrador é notificado sobre os produtos com baixo estoque e envia relatório de produção. Os alertas de baixo estoque são atualizados conforme o estoque é ajustado com entrada ou saída de produtos.
Cenário Alternativo	A1 – Alerta divergente ao estoque físico. A1.2 – O usuário administrador deve alterar a quantidade em estoque para que o item saia do alerta.
Inclusão	Não possui
Extensão	Não possui
Cenário Alternativo	A1 - Quantidade divergente A1.1 - Se o usuário detectar que a quantidade do produto está divergente do estoque físico, pode solicitar uma verificação manual do estoque. A1.2 - Confirmada divergência com o estoque físico, poderá ser alterada a quantidade manualmente do produto.
Inclusão	Não possui
Extensão	Não possui
<b>Caso de Uso – Alertar Baixo Estoque</b>	
ID	UC 004
Descrição	Este caso de uso tem por objetivo enviar um alerta quando os produtos chegarem em uma quantidade mínima estipulada.
Ator Primário	Estoque
Pré-condição	Estar logado em um perfil com a permissão de visualização. Ter definido a quantidade mínima de um produto no UC002.
Cenário Principal	O use case se inicia quando o usuário seleciona a opção “Alertas” O sistema monitora constantemente as quantidades de produtos em estoque. O sistema verifica regularmente se a quantidade de um produto específico está abaixo do nível mínimo definido. Se a quantidade do produto atingir ou cair abaixo do nível mínimo, o sistema gera um alerta de estoque baixo. O sistema exibe o alerta para o usuário, destacando o produto específico, sua quantidade atual e qual a quantidade mínima definida.
Pós-condição	O administrador é notificado sobre os produtos com baixo estoque e envia relatório de produção. Os alertas de baixo estoque são atualizados conforme o estoque é ajustado com entrada ou saída de produtos.
Cenário Alternativo	A1 – Alerta divergente ao estoque físico. A1.2 – O usuário administrador deve alterar a quantidade em estoque para que o item saia do alerta.
Inclusão	Não possui
Extensão	Não possui

**Fonte:** os autores

### 3.7-Diagrama de Classes

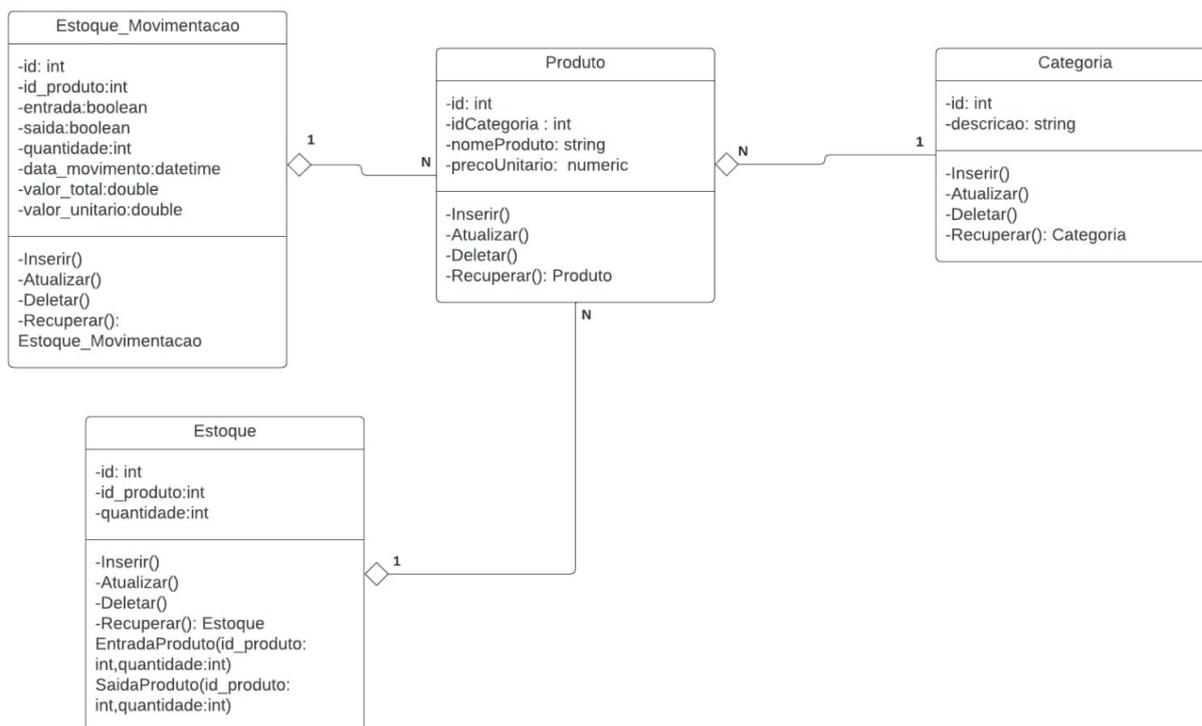
Um diagrama de classe é a representação visual da estrutura e relações entre as

tabelas e atributos de um software. Devendo conter classes do sistema, atributos, cardinalidades e relacionamento entre elas.

Utilizada para obter uma visão clara da organização do sistema, ajudando a entender a estrutura e facilitar a comunicação entre os membros da equipe.

A figura 8 representa o diagrama de classe de 'produto' elaborado e definido para este projeto.

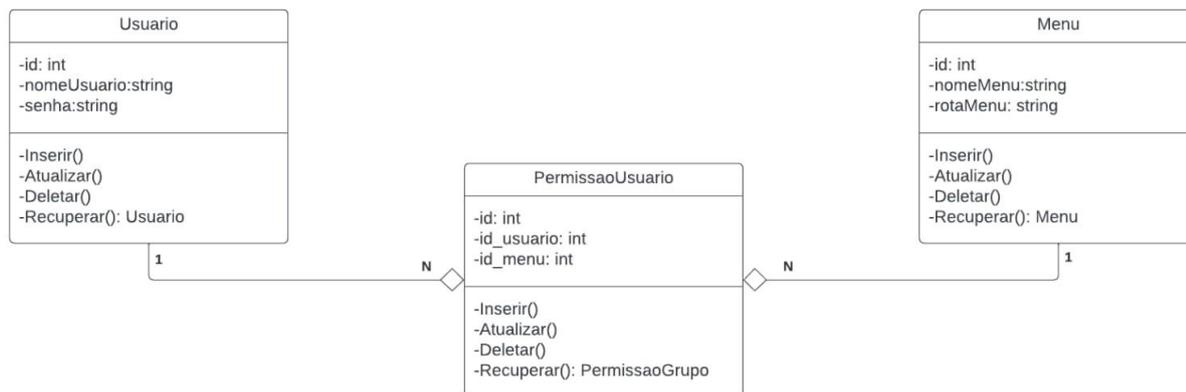
**Figura 8 – Diagrama de Classe de produtos**



**Fonte:** os autores

A figura 9 representa o diagrama de classe de 'usuários' elaborado e definido para este projeto.

**Figura 9 – Diagrama de Classe de Usuários**



**Fonte:** os autores

No diagrama apresentado, foram consideradas as discussões e observações feitas nos outros diagramas, resultando em um modelo de diagrama de classe que inclui tabelas, atributos, métodos e cardinalidade.

Na seção de estoque, são representados o produto, a movimentação do estoque e o próprio estoque. O produto pode ser cadastrado e adicionado ao estoque. Caso haja retirada ou produção de produtos, é registrada uma movimentação.

Na seção de usuário, é estabelecido que cada usuário possui permissões específicas para acessar determinadas telas.

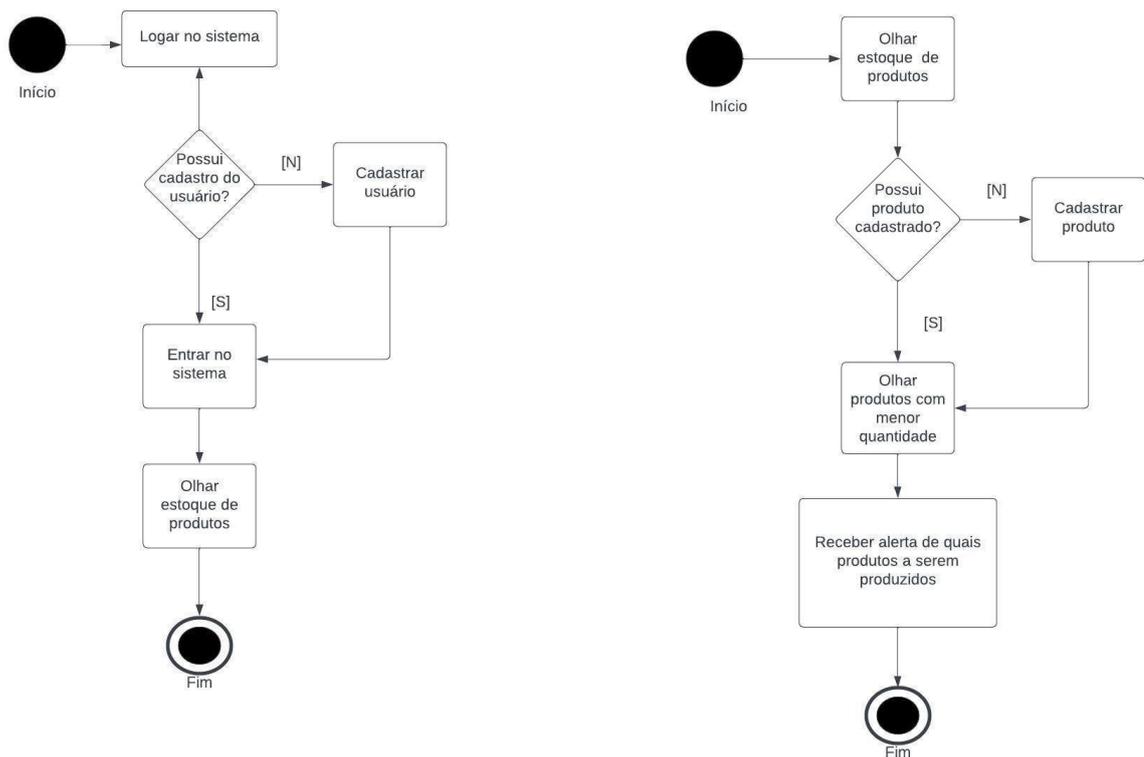
### 3.8-Diagrama de Atividades

Para demonstrar a sequência de ações, decisões e fluxos de controle em um processo ou atividade de uma para outra, é utilizado o diagrama de atividades. Essencial para demonstrar a lógica que o algoritmo terá, podendo identificar o seu comportamento.

O diagrama de atividade pode conter dois estados, simples e compostos e pode incorporar elementos que representam a execução de ações, pontos de decisão e finalização do processo.

A figura 10 representa dois diagramas de atividades (logar) e (estoque).

**Figura 10** – Diagramas de Atividade



**Fonte:** os autores

O diagrama de atividade foi desenvolvido diante do BPMN Processual pelo qual retrata o fluxo de controle de um ponto inicial a um ponto final, aos vários caminhos a serem percorridos enquanto a atividade está sendo executada, como também as decisões ou ações a serem tomadas.

Neste diagrama foi dividido em duas partes: A primeira parte do diagrama retrata o caminho que o usuário realizará ao fazer login no sistema e, caso não tenha um, realizar o seu cadastro. Logo, iniciará as funcionalidades do sistema que é a verificação de produtos em estoque.

A segunda parte do diagrama é a continuação do primeiro diagrama de atividade, em que retrata as funcionalidades que o sistema realizará como: a possibilidade de cadastramento de produtos (se não houver), verificar os produtos em menor quantidade no estoque e informar quais deverão ser produzidos.

### 3.9-Diagrama de Estados

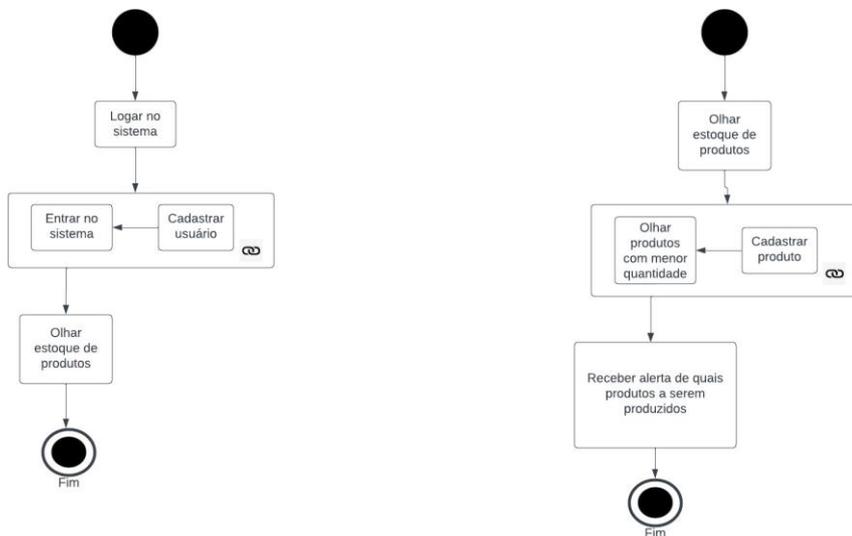
O diagrama de máquina de estado contém em sua representação gráfica a retratação das transições que o sistema passará, ou seja, as mudanças de um estado para outro.

Os elementos que compõem um diagrama de estado são: Sinal (evento pra

representar estímulo assíncrono), Região (parte de um estado ou de uma máquina), Bifurcação (divide uma transição em várias outras), Junção (une várias transições em uma única).

A figura 11 representa dois diagramas de estado (login) e (estoque).

**Figura 11 – Diagramas de Estado**



**Fonte:** os autores

O diagrama foi desempenhado posteriormente ao diagrama de atividade, deste modo as seqüências de estados foram desenvolvidos de acordo com as quantidades de getters (as tomadas de decisões ) que continham no diagrama anterior, por isso foi dividido em duas partes. Ele se inicia com um círculo preenchido e se encerra com um círculo duplo também preenchido.

A primeira parte do diagrama condiz com a ação do usuário realizar o login no sistema, se possui ou não um cadastro e após isto utilizar o sistema.

A segunda parte do diagrama é sobre os estados que o sistema realiza para ser funcional e validando os estados que passará.

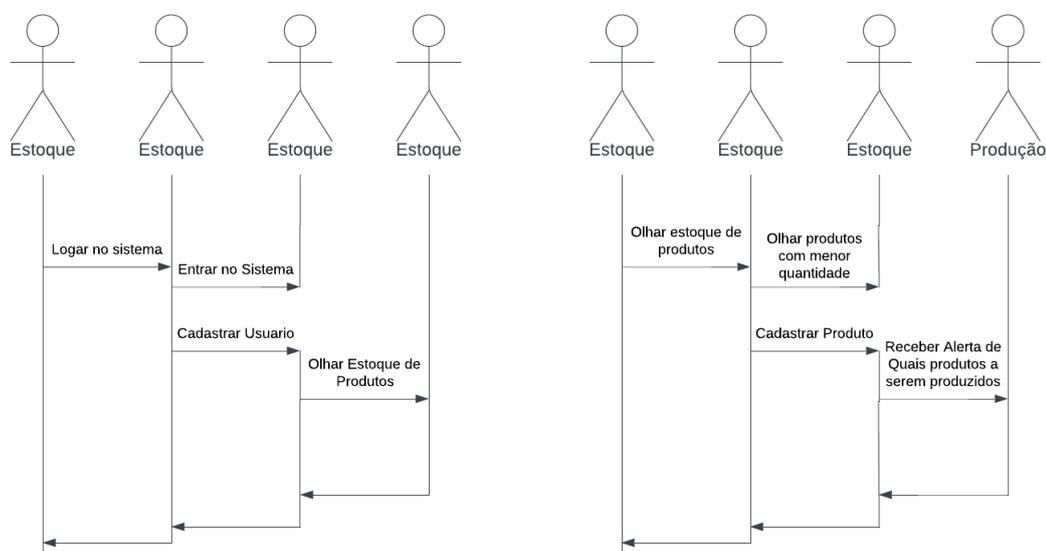
### 3.10-Diagrama de Sequência

Um diagrama de seqüência mostra a interação entre objetos em um sistema, representando a ordem das mensagens trocadas.

É importante porque fornece uma visão clara do comportamento do sistema, ajuda a identificar interações e ordem das mensagens, auxilia os desenvolvedores a entender o fluxo de execução, facilita a comunicação entre as equipes e serve como documentação para referência futura e teste do sistema.

A figura 12 representa dois diagramas de sequencia (logar) e (estoque).

**Figura 12 – Diagramas de Sequência**



**Fonte:** os autores

### 3.11-Matriz de Rastreabilidade

A matriz de rastreabilidade é utilizada para visualização de diferentes requisitos e seus cruzamentos, promovendo relacionamento entre requisitos e casos de uso ou requisitos e regras de negócio (Artia, 2024, online).

Sua principal importância é no gerenciamento e controle de mudanças no escopo, para análise de quais as consequências que essas alterações poderão gerar.

O quadro 6 representa a matriz de rastreabilidade deste projeto, indicando o cruzamento dos Requisitos x Regras do Negócio.

**Quadro 6 – Matriz Requisitos x Regras do Negócio**

	RF 001	RF 002	RF 003	RF 004	RF 005
RN 001					X
RN 002		X			
RN 003	X				
RN 004			X	X	

**Fonte:** os autores

#### **4-Ferramentas e Métodos**

Para efetividade do projeto foram selecionadas ferramentas nas quais seu uso é gratuito, porém completas o suficiente para o projeto.

Para os diagramas foi utilizado o Lucid, que permite as criações dos arquivos, com quantidade limitada de recursos na sua versão para estudantes.

O Bizagi é um dos softwares mais utilizados no mapeamento de processos, assim foi utilizado para a criação dos BPMN's.

A prototipação foi realizada no Figma, ferramenta mais utilizada por quem trabalha com prototipação, design, UX e entre outros, que para a necessidade do trabalho foi utilizado a sua versão gratuita

Para o gerenciamento do projeto foi utilizado o Trello e para a documentação final o editor Word.

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e suporte à comunidade e foram o React.JS, C# e o banco de dados MySQL.

#### **5-Desenvolvimento**

Os artefatos utilizados e o código escrito foram desenvolvidos através da disciplina de engenharia de software III.

A figura 13 exemplifica o controller que é utilizado no backend. Ela é aplicada a partir do conceito de injeção de dependência para utilizar os métodos da 'Service'.

A controller é utilizada apenas para ser o endpoint (uma das extremidades de uma conexão de API,) dos métodos de API (Application Programming Interface). Nesta camada não é aplicado nenhum código.

**Figura 13 - Controller**

```
[ApiController]
1 referência
public class CategoriaController : ControllerBase
{
    private readonly ICategoriaService _CategoriaService;

    0 referências
    public CategoriaController(ICategoriaService CategoriaService)
    {
        _CategoriaService = CategoriaService;
    }

    [HttpGet()]
    0 referências
    public ActionResult<IEnumerable<string>> Categoria() =>
        Ok(_CategoriaService.Categoria());

    [HttpPost]
    0 referências
    public ActionResult<IEnumerable<string>> InserirCategoria(Categoria categoria) =>
        Ok(_CategoriaService.InserirCategoria(categoria));

    [HttpPut]
    0 referências
    public ActionResult<IEnumerable<string>> AtualizarCategoria(Categoria categoria) =>
        Ok(_CategoriaService.AtualizarCategoria(categoria));

    [HttpDelete]
    0 referências
    public bool DeletarCategoria(int id) =>
        _CategoriaService.DeletarCategoria(id);
}
```

Fonte: os autores

A figura 14 é um exemplo de serviço, que emprega o método de injeção de dependência para acessar os métodos do repositório.

Um serviço é empregado para incluir quaisquer regras adicionais além das expectativas originais do método.

Figura 14 – Service

```

2 referências
public class EstoqueService : IEstoqueService
{
    private readonly IEstoqueRepository _EstoqueRepository;

    0 referências
    public EstoqueService(IEstoqueRepository EstoqueRepository)
    {
        _EstoqueRepository = EstoqueRepository;
    }

    2 referências
    public object Estoque(string nomeProduto, int idCategoria) =>
        _EstoqueRepository.Estoque(nomeProduto, idCategoria);

    2 referências
    public object AtualizarEstoque(PutEstoque putEstoque)
    {
        var estoque = _EstoqueRepository.EstoquePorIdProduto(putEstoque.idProduto);

        if(putEstoque.operacao == "A")
        {
            estoque.Quantidade = (estoque.Quantidade + putEstoque.quantidade);
            _EstoqueRepository.Update(estoque);
        }
        else
        {
            if(putEstoque.quantidade > estoque.Quantidade)
            {
                throw new Exception($"Não há quantidade suficiente para retirar essa quantidade de produto");
            }
            estoque.Quantidade = (estoque.Quantidade - putEstoque.quantidade);
            _EstoqueRepository.Update(estoque);
        }
        return estoque;
    }
}

```

**Fonte:** os autores

A figura 15 ilustra o repositório utilizado no contexto. O repositório emprega o EntityFramework para realizar operações relacionadas ao banco de dados (no caso específico, o MySQL). Todos os métodos do repositório são operações de busca, já que outras operações são herdadas da classe BaseRepository, onde os demais métodos são implementados.

**Figura 15 - Repositório**

```
4
5 namespace SorvetesApi.Repositories
6 {
7     2 referências
8     public class EstoqueRepository : BaseRepository<Estoque>, IEstoqueRepository
9     {
10         0 referências
11         public EstoqueRepository(SorvetesDaviContext context) : base(context)
12         {
13         }
14
15     2 referências
16     public object Estoque(string nomeProduto, int idCategoria)
17     {
18         using var db = new SorvetesDaviContext();
19
20         var lQuery = (from Estoque in db.Estoque.AsNoTracking()
21             join Produto in db.Produto.AsNoTracking() on Estoque.IdProduto equals Produto.Id
22             join Categoria in db.Categoria.AsNoTracking() on Produto.IdCategoria equals Categoria.Id
23             where (string.IsNullOrEmpty(nomeProduto) || Produto.NomeProduto == nomeProduto)
24             && (idCategoria == 0 || Produto.IdCategoria == idCategoria)
25             select new
26             {
27                 id = Estoque.Id,
28                 idCategoria = Produto.IdCategoria,
29                 NomeProduto = Produto.NomeProduto,
30                 descricaoCategoria = Categoria.Descricao,
31                 Quantidade = Estoque.Quantidade
32             }).OrderBy(x => x.NomeProduto);
33
34         return lQuery.ToList();
35     }
36 }
```

Fonte: os autores

A figura 16 representa parte do código e desta forma, foi criada uma classe em C# para que o EntityFramework a reconheça como uma tabela no banco de dados.

Figura 16 – código de uma classe

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Runtime.Serialization;

namespace SorvetesApi.Domain
{
    5 referências
    public class Estoque
    {
        [DataMember(Order = 1, Name = "id")]
        [Key]
        [Column("ID")]
        1 referência
        public int Id { get; set; }

        [DataMember(Order = 2, Name = "idProduto")]
        [Column("ID_PRODUTO")]
        4 referências
        public int IdProduto { get; set; }

        [DataMember(Order = 3, Name = "quantidade")]
        [Required(ErrorMessage = "O campo Quantidade é obrigatório.")]
        [Column("QUANTIDADE")]
        9 referências
        public int Quantidade { get; set; }
    }
}
```

Fonte: os autores

Para a conclusão do projeto foram criadas algumas telas, ou seja, o desenvolvimento de suas funcionalidades.

Na figura 16 temos a representação da tela de Login. Nessa tela é possível clicar em “Esqueceu Senha?”, caso o usuário não lembre a senha de acesso.

Figura 16 – Tela de Login

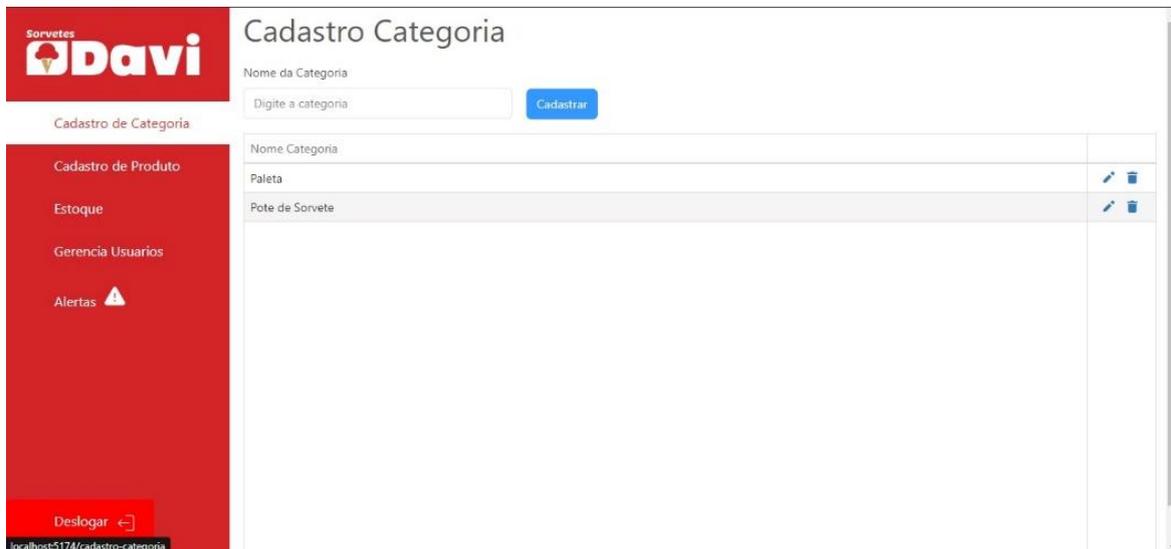
The image shows a login form with the following elements:

- Title: Login
- Label: Nome de Usuario
- Input field: A text box with a vertical cursor.
- Label: Senha
- Input field: A password field.
- Buttons: A blue 'Entrar' button and a text link 'Esqueceu Senha?'.

Fonte: os autores

A figura 17 ilustra o cadastro de categoria que permite ao usuário cadastrar as categorias de produtos de sorvetes no sistema, assim como editar e excluí-lo.

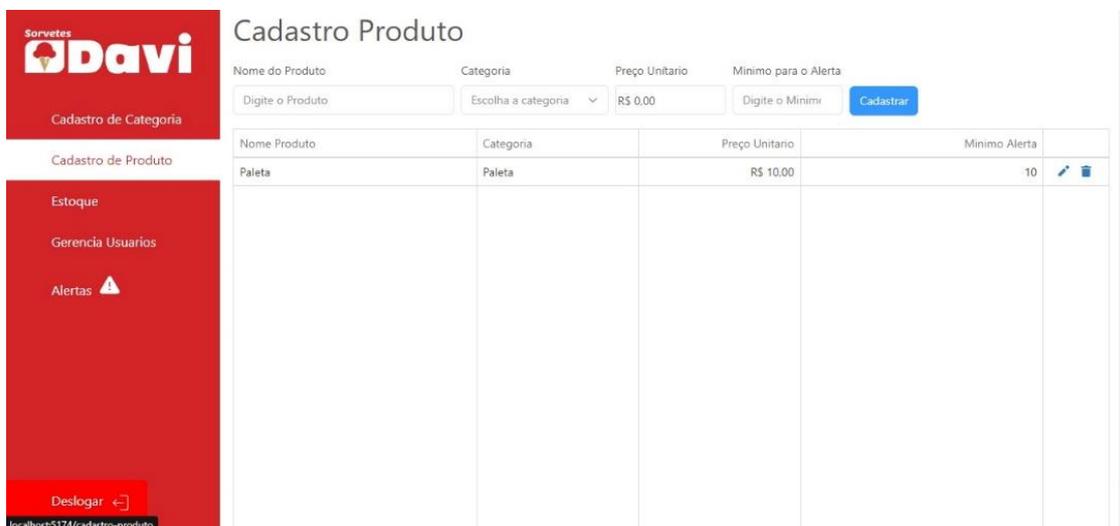
**Figura 17 – Tela de Cadastro de Categoria**



**Fonte:** os autores

A tela de cadastro de produto permite aos usuários inserir novos produtos no sistema com alguns itens obrigatórios para preenchimento como: nome do produto, categoria e o preço unitário. Logo, ela também possibilita poder editar e excluir o item cadastrado, conforme ilustração da figura 18.

**Figura 18 - Tela de Cadastro de Produto**



**Fonte:** os autores

A tela de entrada de estoque fornece para o usuário a possibilidade de uma pesquisa diretamente por nome do produto, com as informações principais do item a ser inserido no sistema.

A figura 19 ilustra a tela de estoque deste projeto.

**Figura 19 – Estoque**

Nome Produto	Categoria	Quantidade
Paleta	Paleta	9

**Fonte:** os autores

## 6-Resultados e Discussão

A solução proposta foi o desenvolvimento de um software personalizado para controle do estoque de produtos da matéria prima e os processados, especificamente para atender as necessidades da empresa 'cliente'.

Esta solução alcançou o objetivo pois foram criados os processos [CRUD] para gerenciar o estoque de produtos por categorias, o cadastro dos usuários e o estoque de produtos processados.

Como o principal problema encontrado foi a falta de controle com relação aos itens retirados do estoque e submetidos para produção final, esta solução também ajudará na organização deste processo.

As principais funcionalidades do sistema são: cadastrar os itens necessários para produção do sorvete; cadastrar estes mesmos itens por categoria; cadastrar os usuários e suas movimentações no sistema e cadastrar os produtos produzidos/processados.

Cientes do contexto atual, onde os processos exigem qualidade e agilidade, este projeto desenvolveu uma solução sistêmica que aborda de maneira eficaz o desafio da gestão do estoque de matéria prima e produtos produzidos no ramo de sorveteria, proporcionando uma solução eficiente.

### **Considerações finais**

O projeto teve como premissa solucionar um dos pontos fracos identificados na empresa, que era a falta de um controle de estoque e produção. Durante o desenvolvimento do mesmo, foram feitas análises da situação atual e o que poderia ser realizada, chegando a criação de um software que substituirá as ações realizadas manualmente.

Algumas funcionalidades que permitem criar, atualizar, deletar e visualizar os produtos, enquanto os usuários da produção poderão registrar a entrada dos produtos, incluindo quantidade no estoque.

Há outros processos que ajudarão na otimização e interação deste sistema operacional, como o setor de compras, vendas, entregas e outros.

Desta forma, o projeto de gerenciamento de produção e estoque da empresa do ramo de sorvetes foi bem-sucedida em sua proposta e criação.

### **Referências**

AGENDOR. Matriz SWOT. Disponível em: <https://www.agendor.com.br/blog/matriz-swot-como-fazer/>. Acesso em: 04 março 2024.

ARTIA. Como fazer EAP na gestão de projetos. Disponível em: <https://artia.com/blog/como-fazer-eap-na-gestao-de-projetos/>. Acesso em: 15 jun. 2024.

ARTIA. Matriz de Rastreabilidade. Disponível em: <https://artia.com/blog/matriz-de-rastreabilidade/>. Acesso em: 16 março 2024.

ARTIA. Termo de Abertura do Projeto. Disponível em: <https://artia.com/blog/termo-de-abertura-do-projeto/>. Acesso em: 25 abril 2024.

LOGAP. Prototipação. Disponível em : <https://logap.com.br/blog/o-que-e-prototipacao-software>. Acesso em: 1 de maio de 2024

RODOJACTO. Sistema de controle de estoque. Disponível em: <https://transporte.rodjacto.com.br/>. Acesso em: 20 abril 2024.

SEBRAE. Canvas. Disponível em: <https://digital.sebraers.com.br/blog>. Acesso em: 1 de maio de 2024.

SEBRAE. Como melhorar a gestão de produtos no varejo. Disponível em:

<https://sebrae.com.br/sites/PortalSebrae/artigos>. Acesso em: 15 jun. 2023.

TREASY. 5W2H. Disponível em: <https://www.treasy.com.br/blog/5w2h/>. Acesso em: 15 março 2024.

## Anexo 01 – Missão, Visão e Valores da empresa ‘cliente’

Missão: Nos atentamos em ser um time que agregue as necessidades de novos clientes, no preço acessível para todos e no produto de qualidade.

Visão: Sermos reconhecidos pela boa qualidade e ter acesso aos pequenos e grandes negócios.

Valores: Deus acima de tudo, honra, respeito, dedicação, honestidade, trabalho e qualidade.

## Anexo 02 – Termo de autorização de uso de imagem e nome da Empresa

Figura 19 – Termo de autorização de uso de imagem

**Fatec**  
Franca  
Dr. Thomaz Novelino

### FACULDADE DE TÉCNOLOGIA DO ESTADO DE SÃO PAULO - FATEC TERMO DE AUTORIZAÇÃO DE USO DE NOME E IMAGEM

Eu, Vanessa Lopes da Silva, nacionalidade brasileira estado civil casada, responsável pelo corpo administrativo da empresa Sorvetes Davi portador da Cédula de identidade RG nº 45.787.601-5, AUTORIZO o uso da imagem da empresa no trabalho acadêmico de Engenharia de Software III da Instituição Tecnológica de Franca/SP e somente para esta finalidade.

Fica ainda **autorizada**, de livre e espontânea vontade, para os mesmos fins, a cessão de direitos da veiculação das imagens não recebendo para tanto qualquer tipo de remuneração e expresso a vontade do uso da imagem para uso exclusivo da instituição para fins acadêmicos e somente.

Por esta ser a expressão da minha vontade declaro que autorizo o uso acima descrito sem que nada haja a ser reclamado a título de direitos conexos à imagem ou a qualquer outro.

Franca, dia 08 de junho de 2023.

Wanyllima  
(Assinatura)

Nome:  
Telefone p/ contato:

**35.485.195/0001-91**  
J. D. DA SILVA IND. E  
COM. DE SORVETES ME  
Rua Madre Maria Vilac, 2010  
Jd. Aeroporto III CEP 14.404-260  
**FRANCA - SP**

Fonte: os autores

## **Apêndice 01 – Entrevista Geral**

Quais processos deveriam ser informatizados?

Já possui áreas informatizadas? Caso a resposta seja positiva, estas possuem algum problema?

Quantas pessoas têm acesso ao sistema?

Quantos aparelhos a empresa possui?

Há controle de estoque? Se sim, como é feito?

Como é feita a distribuição dos produtos?

Há venda mínima? É por lote?

Há estatística para a quantidade de produtos vendidos?

Quais os cargos que utilizam o sistema?

É feito o backup dos dados da empresa?

Como são organizadas as câmaras?

Qual a forma de armazenamento de estoque?

Como é o controle de validade/lote?

Como é feita a decisão de produção?

Qual a medida que cada produto utiliza? (litros/kg)

Como é a entrada e saída de produto?