

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

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**GIOVANI FALEIROS LOPES  
JULIANA CRISTINA MACHADO**

**SISTEMA DE VENDAS COM INTEGRAÇÃO DE BALANÇA PARA  
SERVIÇOS DE *SELF-SERVICE***

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  
2023**

# SISTEMA DE VENDAS COM INTEGRAÇÃO DE BALANÇA PARA SISTEMA SELF-SERVICE

Giovani Faleiros Lopes<sup>1</sup>

Juliana Cristina Machado<sup>2</sup>

## Resumo

Este estudo tem como objetivo criar uma ferramenta para auxiliar no comércio de vendas de açaí e sorvetes, com intuito de facilitar o gerenciamento de vendas que possui o modelo autoatendimento (*Self-service*). O sistema *self-service* é um modelo de negócio que permite aos clientes escolherem e servirem a si mesmos uma variedade de sabores de açaí, sorvetes e acompanhamentos que estão disponíveis em uma estação de serviço. Normalmente, o cliente paga por peso, ou seja, pelo valor total do peso dos produtos selecionados. Esse tipo de serviço proporciona aos clientes uma experiência única, permitindo que eles possam personalizar o seu produto, reduzir seu tempo de espera para atendimento, pois não existe a necessidade de aguardar demais, o cliente é quem determina quando se servir, além de reduzir custos com funcionários já que eles são direcionados a outras atividades. O modelo em questão é relativamente fácil de gerenciar, podendo ter uma margem de lucro generosa, desde que todos os custos com ingredientes e as manutenções sejam controladas de forma efetiva.

**Palavras-chave:** Autoatendimento. Consumo consciente. Lucratividade.

## Abstract

*This study aims at creating a tool to assist in the sales of açaí and ice cream, with the intention of facilitating the management of sales that has the Self-service model. The self-service system is a business model that allows customers to choose a variety of flavors of açaí, ice cream, and toppings that are available at the drop-in and serve themselves. Normally, the customer pays by weight, that is, for the total weight value of the selected products. This type of service provides customers with a unique experience, allowing them to personalize their product, reduce their waiting time for service, as there is no need to wait too long, the customer determines when to serve themselves, and reduces costs with employees since they are directed to other activities. This type of service is relatively easy to manage and can have a generous profit margin, as long as all the costs with ingredients and maintenance are effectively controlled.*

**Keywords:** *Self-service. Conscious consumption. Profitability.*

---

<sup>1</sup> Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: agokdroid@gmail.com.

<sup>2</sup> Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: juhcrmachado@gmail.com.

## **1 Introdução**

O sistema de atendimento *self-service* ou autosserviço é um sistema onde o cliente se serve sozinho, ou seja, não existe a necessidade de um funcionário para montar o seu prato ou copo, isto é, o cliente escolhe o tamanho do copo ou prato, a quantidade de produtos e os seus acompanhamentos, caso queira.

Trata-se de um sistema bem diferente do conhecido a lá carte, onde os pratos são apresentados em um menu que é previamente definido, com valores e quantidades já determinados.

No Self Service os valores são medidos pela quantidade de produto que o cliente irá consumir, ou seja, quanto mais produtos inseridos no copo ou prato maior será o peso, elevando assim o valor a ser pago (SEBRAE, 2022).

Como possui diversas vantagens os sistemas de *self-service* se popularizou no mundo todo, ele proporciona aos estabelecimentos um aumento significativo da eficiência em seus serviços, isso é possível, pois a função de servir que antes seria exercida por um garçom passa a ser realizada pelo próprio cliente, que determinará quais produtos serão consumidos, isso inevitavelmente agilizará o processo de consumo de cada cliente, reduz custos já que existe a redução da folha de pagamento e um melhor aproveitamento dos funcionários, pois eles são direcionados a atividades específicas com o intuito de melhorar ainda mais as experiências de consumo de cada cliente, outra vantagem é a possibilidade de redução de tempo de espera por parte dos clientes, pois não existe a necessidade de aguardar o atendimento de um garçom, o cliente é quem determina quando se servir.

O objetivo deste estudo visa desenvolver uma ferramenta para auxiliar no comércio de vendas de açaí e sorvetes, com intuito de facilitar o gerenciamento de vendas que possui o modelo autoatendimento maximizando os benefícios proporcionados por este tipo de serviço.

## **2 Viabilidade do Projeto**

O Estudo de viabilidade do projeto tem como principal objetivo analisar a possibilidade da realização do projeto, ele será responsável pela tomada de decisão sobre o início ou a continuidade do projeto, dizendo se o investimento é de fato recomendado.

Na realização do estudo serão levantadas questões que precisam ser respondidas para compreender como será o processo de execução, trazendo mais segurança e credibilidade as aplicações e se ele é condizente com todo o esforço empenhado no projeto.

Diversas etapas são analisadas no desenvolvimento do estudo, sendo algumas delas: os custos, as despesas, os investimentos, a projeção de receitas entre outras.

Apresentação da viabilidade do projeto elaborado via Canvas, representado na Figura 1, descreve os principais parceiros, que poderão fazer uso do sistema bem como os principais recursos, custos e fontes de receita provenientes da aplicação

**Figura 1 - Estudo de Viabilidade**

Principais Parceiros	Principais Atividades	Proposição de Valor	Relacionamento	Segmentos de clientes
 Empresas de comercio de açai e sorvetes que utilizam a modalidade de atendimento selfservice e produtos a lá carte	 Desenvolver um novo sistema que fara a gestão do comercio, incluindo todo o processo de venda e produção	 Auxiliar os administradores no processo de gestão da empresa, contribuindo na tomada de decisão em relação a produção de acordo com os níveis de estoque de cada produto.	 Suporte técnico e assistência personalizada para melhorias do software.	 Micro e pequenos empreendedores; Pequenas empresas com problemas de gestão de vendas e produção;
	<b>Principais Recursos</b>  Computadores, Desenvolvedores, Banco de dados para armazenamento de dados.	Facilitar o processo de vendas com uma única ferramenta	<b>Canais</b>  Sites; Redes Sociais; Materiais Gratuitos.	
	8	7	2	4
				1
<b>Estrutura de Custo</b>  15 mil Reais		<b>Fontes de Receita</b>  Venda do software e taxas de manuntenção		
		9		
		5		

Fonte: Autores

### 3 Levantamento de Requisitos

#### 3.1 Elicitação e Especificação dos Requisitos

A elicitação de requisitos consiste no processo de reunir as informações acerca do sistema que será desenvolvido, envolvendo durante o processo de captura de informações todos os que de alguma forma possam contribuir com a descoberta dos requisitos, os chamados *Stakeholders*.

A captura das informações pode ser realizada de diversas maneiras, podendo ser utilizada não somente uma, mas sim, técnicas variadas para um melhor entendimento das necessidades seja de fato eficiente.

Dentre a mais conhecida das técnicas está a entrevista, que consiste em uma conversa formal ou informal com os *Stakeholders*, nesta conversa são levantadas as informações sobre as perspectivas do sistema que será desenvolvido. Essas entrevistas podem ser realizadas de forma fechada, onde cada *Stakeholders* responde algumas perguntas que foram previamente definidas pelos engenheiros de software. Já na entrevista aberta a equipe vai ao longo do diálogo explorando as questões e necessidades dos *Stakeholders* buscando absorver e compreender as informações.

Outro processo de elicitação que pode ser aplicado é o baseado em cenários, com início em um resumo da iteração, e ao longo do processo são incluídos os detalhes e assim ocorre a descrição completa de todo o processo de iteração do sistema.

Diferente dos processos citados anteriormente, a etnografia consiste na técnica de observação, esta é uma das técnicas que demandam mais tempo, pois nesta técnica o analista é inserido no ambiente de trabalho do *Stakeholders*. Será realizado um trabalho diário de observação e anotações para compreender os requisitos sociais e organizacionais da empresa.

Independentemente do método de elicitação escolhido, é necessário compreender a importância desta etapa, nela serão pré-estabelecidos os processos e as necessidades que o cliente possui e de que maneira os problemas serão solucionados.

Na etapa inicial da elicitação dos requisitos foi realizada uma reunião entre os desenvolvedores do projeto para a elaboração de um questionário relacionado ao comércio em questão, foram elaboradas 12 questões que serão descritas abaixo, indagações essas que nos permitiu compreender, identificar e analisar os problemas efetivos da empresa, e quais eram as necessidades e expectativas do comércio.

Posteriormente, em uma visita técnica realizamos a entrevista com o proprietário do estabelecimento, na entrevista foram apresentadas as questões levantadas anteriormente.

O comércio em questão trata-se de uma empresa familiar que precisa realizar tanto o controle de venda bem como o controle de produção dos produtos (Açaí e

Soverte). A empresa surgiu no ano de 2016, a cidade escolhida para abertura do comércio foi Ituverava, pois a cidade possui um clima quente na maior parte do ano. Nos primeiros anos da empresa o processo de venda foi controlado por planilhas de Excel e um *software* de prateleira que não atende a todas as necessidades do estabelecimento.

Em virtude disto viu-se a exigência de devolver um sistema que fosse personalizado e atendesse a todas as suas necessidades, o local trabalha com vendas em duas modalidades sendo elas: a lá carte e *self-service*, por possuir uma forma diversificada de atendimento o sistema atual não é viável.

A seguir estão as questões citadas:

- Descreva todo o processo de vendas.
- Quais os produtos/materiais são comercializados?
- Como são realizadas as cobranças aos clientes
- Como é feita a escolha dos produtos à lá carte (produto Pronto)?
- Você pode falar um pouco sobre a cultura da empresa?
- O que te levou a atuar(empreender) nesta área?
- Como é a gestão da sua empresa?
- Existe algum controle na empresa, seja ele de vendas ou produção?
- Qual a sua expectativa em relação ao desenvolvimento de um novo sistema, o que você espera que seja melhorado?
- Como é analisado a produtividade e os resultados da empresa?
- Quais são os pontos fortes e pontos fracos da sua empresa? Riscos e Oportunidades dela?
- Existe algum sistema de gerenciamento de vendas na empresa? Ele atende todas as suas necessidades?
- Todo processo de vendas é automatizado? Quais os processos são realizados de forma manual?
- Qual tipo de sistema seria mais útil para o seu segmento, sistema Desktop ou sistema Web?
- Quantos equipamentos você tem disponível na empresa? E quais as especificações? (Armazenamento, memória, processador).
- Conte-nos a história da sua empresa.

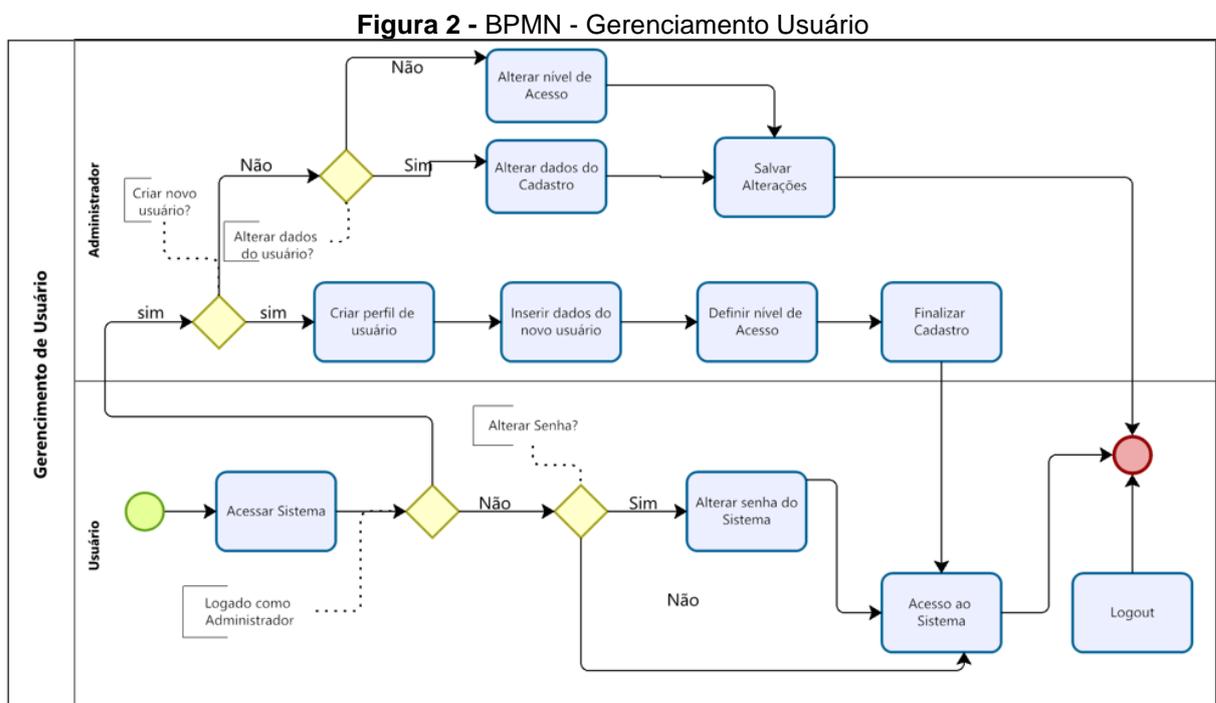
### 3.2 BPMN

BPMN (*Business Process Modeling Notation*) ou notação de modelagem de processos de negócios é um modelo de notação gráfica que foi padronizado com intuito de representar os processos de negócios de uma empresa ou sistema por meio de diagramas.

Com a função de mostrar de forma clara e fácil até mesmo os processos mais complexos, o BPMN apresenta o passo a passo da lógica dos problemas, facilitando a compreensão e a visualização, utilizando-se de ícones para simbolizar as etapas do processo (SYDLE, 2023).

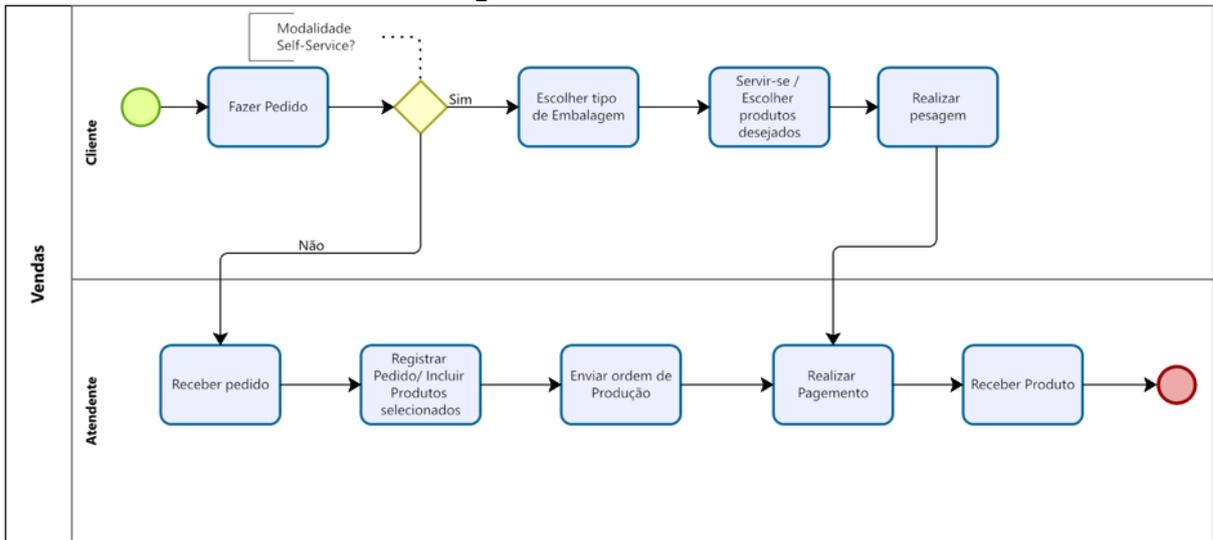
A partir dos dados do negócio é realizado a modelagem e posteriormente a automatização do processo. Em torno do desenho da modelagem é possível identificar as falhas de processo, realizar os devidos ajustes ao longo do projeto visando sempre sua otimização, evitando desperdício.

O BPMN do projeto em questão pode ser visto nas Figuras 2, 3 e 4, onde são explicados o passo a passo do processo de criação de usuários, todas as suas etapas até, bem como as atividades e processos das ações de venda até a finalização do processo de retirada ou armazenamento do estoque.



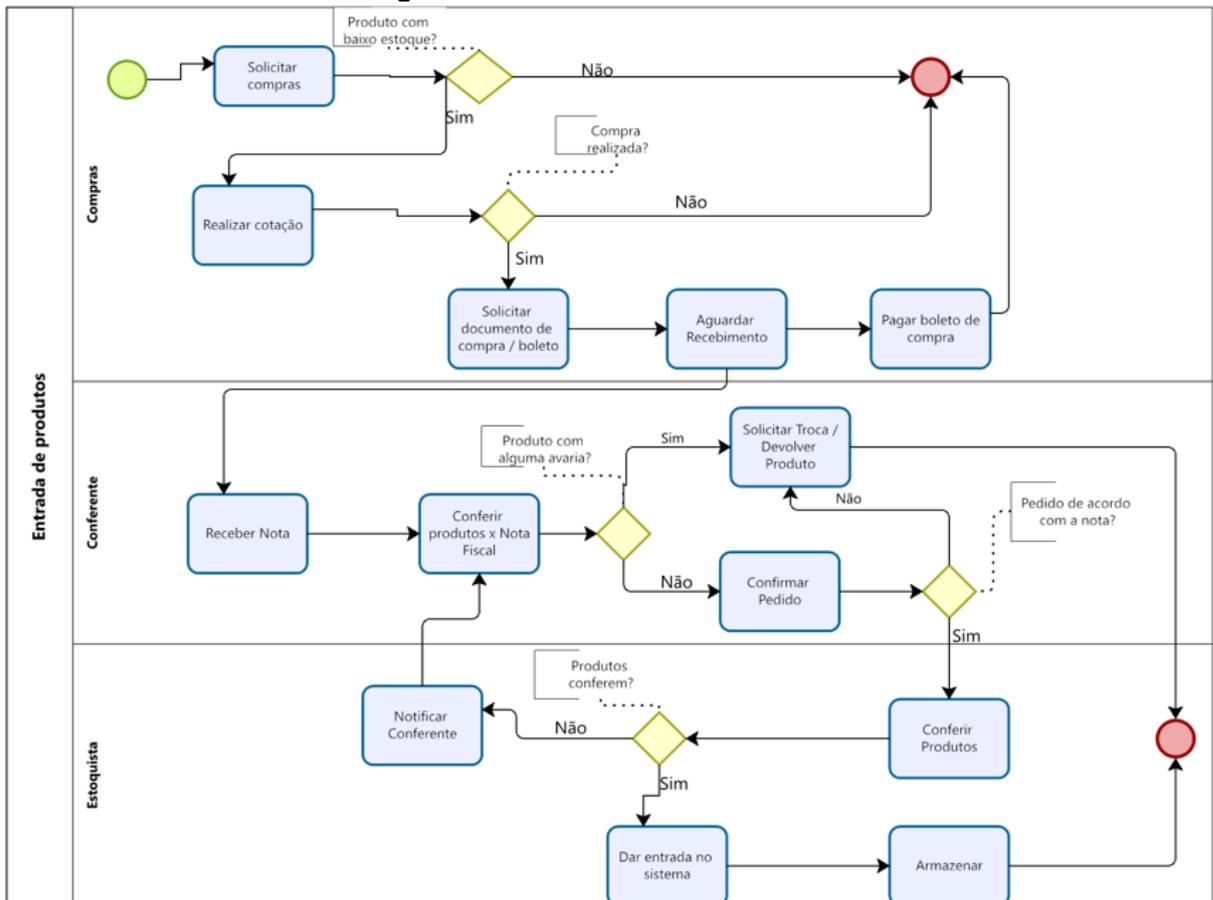
Fonte: Autores

**Figura 3 - BPMN - Vendas**



Fonte: Autores

**Figura 4 - BPMN - Entrada de Produtos**



Fonte: Autores

### 3.3 Requisitos Funcionais

Segundo Sommerville (2011, p.59):

Requisitos funcionais. São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer.

Os requisitos funcionais vão descrever de forma detalhada as funções que o sistema irá executar, abrangendo requisitos gerais ou até mesmo as funcionalidades mais específicas de cada sistema.

Os requisitos referentes ao projeto estão descritos no Quadro 1.

**Quadro 1 - Requisitos Funcionais**

<b>RF001-</b> Cadastrar Usuários	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve cadastrar usuários de acesso, separando-os por níveis de acesso de acordo com a função na empresa.		
<b>RF002-</b> Cadastrar nível de acesso	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve cadastrar níveis de acesso dos usuários		
<b>RF003-</b> Alterar usuários Cadastrados	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir a alteração de usuários cadastrados		
<b>RF004-</b> Consultar usuários Cadastrados	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir a consulta de usuários cadastrados		
<b>RF005-</b> Desabilitar Usuário	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir que o administrador desabilite usuários do sistema		
<b>RF006-</b> Cadastrar tipos de produtos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir o cadastro de tipos de produtos		

<b>RF007</b> -Cadastrar novos Produtos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir o cadastro de novos produtos, contendo itens como: descrição, quantidade e valor		
<b>RF008</b> - Editar Produtos Cadastrados	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir a edição de produtos já cadastrados no sistema		
<b>RF009</b> -Consultar Produtos Cadastrados	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir que o usuário logado consiga pesquisar os produtos no sistema, mostrando as informações de quantidade disponível.		
<b>RF010</b> -Salvar histórico de alterações	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve armazenar o histórico de alteração realizadas no produto bem o usuário responsável.		
<b>RF011</b> -Registrar Vendas	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve registrar as vendas efetuadas		
<b>RF012</b> -Cadastrar Clientes	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir o cadastro de clientes		
<b>RF013</b> -Cadastrar Fornecedor	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir o cadastro de fornecedores		
<b>RF014</b> -Cadastrar Pedidos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve permitir o cadastro de pedidos		
<b>RF015</b> -Gerenciar Pedidos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa

**Descrição:** O sistema deve permitir o gerenciamento dos pedidos, controlando o estoque à medida que os pedidos forem sendo realizados.

**Fonte:** Autores

### 3.4 Requisitos Não Funcionais

Ainda segundo Sommerville (2011, p.59):

Requisitos não funcionais: São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.

Os requisitos não funcionais na maioria das vezes está direcionado as necessidades do sistema em si, ou seja, não está conectado com as funções, como exemplo de propriedades não funcionais podemos citar tempo de resposta, segurança entre outros.

Os requisitos não funcionais estão especificados no Quadro 2.

**Quadro 2 - Requisitos Não Funcionais**

<b>RNF001-</b> Login	O sistema deve restringir o acesso ao sistema conforme os níveis de acesso	Tipo - Evidente	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF002-</b> Recuperação de senha	O sistema deve permitir que os usuários cadastrados recuperem as senhas esquecidas	Tipo - Evidente	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF003-</b> Gerenciar Usuários	O sistema deve permitir a parametrização dos níveis de acesso de cada usuário.	Tipo - Evidente	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF004-</b> Usabilidade	O usuário deve estar apto a usar todas as funções do sistema após a realização do treinamento.	Tipo - oculto	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF005-</b> Segurança das senhas	O sistema aceitara apenas senha que contenha letras, números e caracteres com no mínimo 8 dígitos..	Tipo - evidente	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF006-</b> Cores da Empresa	O sistema deverá conter predominantemente as cores características da empresa, Roxo e amarelo.	Tipo - evidente	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF007-</b> Funcionalidade	O funcionamento do sistema será realizado no modo on-line.	Tipo - evidente	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF008-</b> Tempo de Execução	Os processos de solicitação do sistema deverão ter o tempo de resposta de no máximo 1 minuto	Tipo - oculto	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório

**Fonte:** Autores

### 3.5 Regras de Negócio

As Regras de Negócios são as instruções estipuladas pela empresa com o objetivo de definir ou restringir algumas ações, dando direcionamento para a realização das operações. Definir bem essas diretrizes proporcionam a empresa olhar mais claro de como os processos devem ser realizados.

As regras deste projeto podem ser visualizadas no Quadro 3.

**Quadro 3 - Regras de Negócio**

<b>RN001 – Cadastro de usuários</b>
<b>Descrição:</b> Só serão permitidos cadastros e gerenciamento de usuários quem acessar o sistema com nível de administrador do sistema.
<b>RN002 – Cadastro de Produtos</b>
<b>Descrição:</b> Apenas serão aceitos alteração manual de cadastro de produtos, o usuário que possuir tal nível de acesso.
<b>RN003 – Consulta de Produtos</b>
<b>Descrição:</b> Serão permitidas consultas aos produtos somente no modo leitura.
<b>RN004 – Alteração de senha</b>
<b>Descrição:</b> Serão solicitadas alteração de senhas a cada 30 dias. Para o caso de esquecimento, necessário realizar o processo de reset e alteração de senha.
<b>RN005 – Realização de Pedidos</b>
<b>Descrição:</b> Serão permitidas somente um tipo de pedido, <i>self-service</i> ou a lá carte, caso queira os dois, será necessário realizar outro pedido.

**Fonte:** Autores

### 3.6 Casos de Uso

Como descrito por Sommerville (2011), o diagrama de caso de uso é uma das técnicas utilizadas para elicitación de requisitos e mostra como é a interação entre os usuários e o sistema em si.

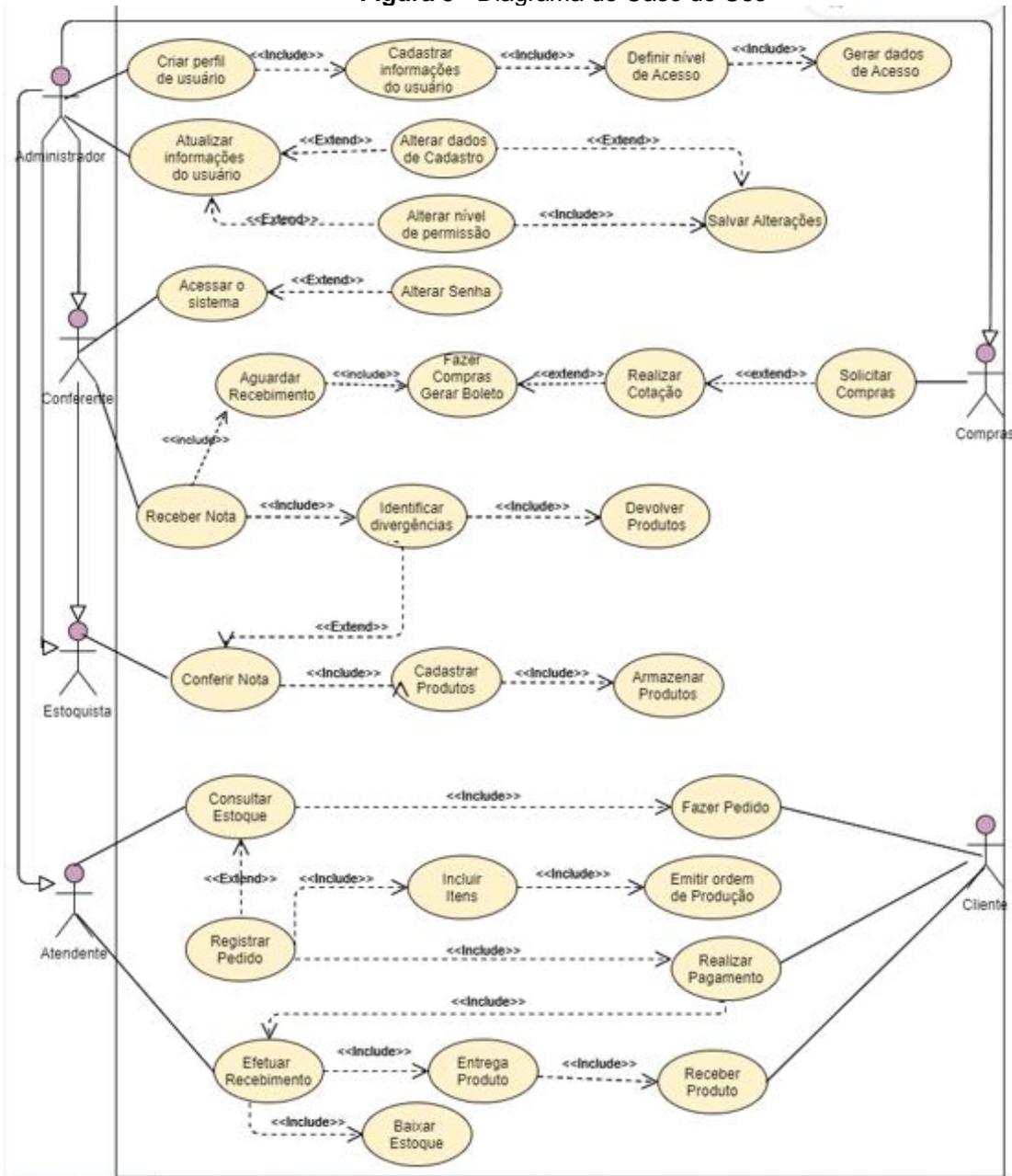
Considerado um diagrama de alto nível, ele vai descrever todas as ações realizadas tanto pelos atores quanto pelo sistema dentro de um cenário. Desenhar este diagrama é uma das formas de definir as funcionalidades do sistema além de ajudar a identificar as possíveis restrições, sendo utilizado durante o ciclo de vida de um software, ou seja, ele está presente desde a especificação dos requisitos até a validação do sistema.

O diagrama de caso de uso deste projeto está representado na Figura 5, que apresenta todas as iterações entre os atores e os processos do sistema, que vai desde a criação do tipo de acesso até a entrega do produto final ao cliente.

Índice de casos de uso:

- UC 001 - Criar Perfil de Usuário
- UC 002 - Cadastrar informações do usuário
- UC 003 - Definir nível de Acesso
- UC 004 - Gerar dados de acesso
- UC 005 - Atualizar Informações Do Usuário
- UC 006 - Alterar Dados de Cadastro
- UC 007 - Alterar Nível de Permissão
- UC008 - Salvar Alterações
- UC 009 - Acessar o Sistema
- UC 010 - Alterar Senha
- UC 011 - Solicitar Compra
- UC 012 - Realizar Cotação
- UC 013 - Fazer Compras/ Pedido
- UC 014 - Receber Nota
- UC 015 - Identificar Divergências
- UC 016 - Devolver Produto
- UC 017 - Conferir Nota
- UC 018 - Cadastrar Produtos
- UC 019 - Armazenar Produto
- UC 020 - Fazer Pedido
- UC 021 - Consultar Estoque
- UC 022 - Registrar Pedido
- UC 023 - Emitir Ordem de Produção
- UC 024 - Realizar Pagamento
- UC 025 - Baixar Estoque
- UC 026 - Entregar Produto
- UC 027 - Receber Produto

Figura 5 - Diagrama de Caso de Uso



Fonte: Autores

Todas as especificações do caso de uso, descrições, atores e possíveis cenários estão descritos nos Quadros 4 a 40.

**Quadro 4 - Caso de Uso: Criar Perfil de Usuário**

<b>Caso de Uso – Criar Perfil de Usuário</b>	
<b>ID</b>	UC 001
<b>Descrição</b>	Este caso de uso tem por objetivo criar um perfil de usuário para acesso ao sistema
<b>Ator Primário</b>	Administrador
<b>Pré-condição</b>	Estar logado com o usuário de Administrador
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o administrador seleciona o campo de criar perfil usuário.</li> <li>2. O sistema carrega o formulário para o perfil de usuário.</li> <li>3. O administrador informa o nome de usuário do colaborador.</li> <li>4. O sistema consulta o banco de dados e verifica.</li> <li>5. O sistema valida o usuário.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	1a – O administrador pode a qualquer momento sair do sistema 4a – O usuário informado já consta no banco de dados

**Fonte:** Autores

**Quadro 5 - Caso de Uso – Cadastrar informações do usuário**

<b>Caso de Uso – Cadastrar informações do usuário</b>	
<b>ID</b>	UC 002
<b>Descrição</b>	Este caso de uso tem por objetivo cadastrar as informações de usuário no sistema.
<b>Ator Primário</b>	Administrador
<b>Pré-condição</b>	Ter passado pelo UC 001 – Criar Perfil do Usuário
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o administrador seleciona o campo de cadastrar informações de usuário.</li> <li>2. O sistema carrega o formulário para o cadastramento das informações de usuário.</li> <li>3. O administrador preenche os campos de nome completo, data de nascimento, RG, CPF, endereço, número, cep, cidade, estado, telefone.</li> <li>4. O sistema grava a data de inclusão.</li> <li>5. Uma mensagem de confirmação do cadastro informações usuário é mostrada.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	1a – O administrador pode a qualquer momento sair do sistema 5a – O CPF já consta nos cadastros

**Fonte:** Autores

**Quadro 6 - Caso de Uso – Definir nível de Acesso**

<b>Caso de Uso – Definir nível de Acesso</b>	
<b>ID</b>	UC 003
<b>Descrição</b>	Este caso de uso tem por objetivo definir o nível de permissão que o usuário tem no sistema.
<b>Ator Primário</b>	Administrador

<b>Pré-condição</b>	Ter passado pela UC 001 – Criar Perfil Usuário, UC 002 – Cadastrar Informações Usuário.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o administrador finalizar o cadastrado do usuário no sistema.</li> <li>2. O sistema carregará a tela com os níveis de permissão.</li> <li>3. O administrador selecionara um nível de permissão para o usuário.</li> <li>4. O administrador salvará o nível de permissão ao usuário.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	<p>1a – O administrador pode a qualquer momento sair do sistema</p> <p>3a – Usuário não pode ficar sem o nível de permissão atrelada.</p>

**Fonte:** Autores

**Quadro 7 - Caso de Uso – Gerar dados de acesso**

<b>Caso de Uso – Gerar dados de acesso</b>	
<b>ID</b>	UC 004
<b>Descrição</b>	Este caso de uso tem por objetivo criar um relatório, com dados de usuário ao sistema.
<b>Ator Primário</b>	Administrador
<b>Pré-condição</b>	Ter passado pela UC 001 – Criar Perfil Usuário, UC 002 – Cadastrar Informações Usuário e UC 003 – Definir Nível de Permissão.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o administrador finalizar o cadastro do usuário no sistema.</li> <li>2. O sistema gerar um relatório contendo os dados de acesso.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	1a – O administrador pode a qualquer momento sair do sistema

**Fonte:** Autores

**Quadro 8 - Caso de Uso – Atualizar Informações Do Usuário**

<b>Caso de Uso – Atualizar Informações Do Usuário</b>	
<b>ID</b>	UC 005
<b>Descrição</b>	Este caso de uso tem o objetivo de atualizar as informações de acesso do usuário
<b>Ator Primário</b>	Administrador
<b>Pré-condição</b>	Estar logado com usuário Administrador
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o administrador seleciona o campo de atualizar informações de usuário.</li> <li>2. O sistema carrega a lista de usuários existentes.</li> <li>3. O administrador seleciona o usuário que deseja realizar as alterações desejadas.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	1ª – O administrador pode sair do sistema a qualquer momento

**Fonte:** Autores

**Quadro 9 - Caso de Uso – Alterar Dados de Cadastro**

<b>Caso de Uso – Alterar Dados de Cadastro</b>	
<b>ID</b>	UC 006
<b>Descrição</b>	Este caso de uso tem o objetivo de atualizar os dados de cadastro de um usuário selecionado.

<b>Ator Primário</b>	Administrador
<b>Pré-condição</b>	Ter passado pela UC 005 – Atualizar Informações de Acesso
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o administrador seleciona o campo de alterar dados de cadastro.</li> <li>2. O sistema carrega o usuário selecionado, na UC anterior.</li> <li>3. O sistema abrirá um formulário para que possa fornecer os novos dados.</li> <li>4. O administrador irá preencher que poder ser alterados com os campos de endereço, número, cep, cidade, estado, telefone.</li> <li>5. O administrador salvará as informações.</li> <li>6. O sistema salvará no banco de dados.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	<p>1a - Em qualquer momento o administrador pode sair do sistema.</p> <p>4a – Caso fique algum campo em branco, o sistema não deixa salvar, sendo necessário a verificação dos campos que estão em brancos ou com dados incoerentes. O sistema abrirá um formulário para que possa fornecer os novos dados.</p>

**Fonte:** Autores

**Quadro 10 - Caso de Uso – Alterar Nível de Permissão**

<b>Caso de Uso – Alterar Nível de Permissão</b>	
<b>ID</b>	UC 007
<b>Descrição</b>	Este caso de uso tem o objetivo alterar a permissão do usuário selecionado.
<b>Ator Primário</b>	Administrador
<b>Pré-condição</b>	Ter passado pela UC 005 – Atualizar Informações de Acesso
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o administrador seleciona o campo de altera nível de permissão.</li> <li>2. O sistema carregará a tela com os níveis de permissão.</li> <li>3. O administrador alterar o nível de permissão para o usuário, conforme a necessidade solicitada.</li> <li>4. O administrador salvará o nível de permissão ao usuário.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	<p>1a - Em qualquer momento o administrador pode sair do sistema.</p> <p>3a – O sistema não deixará que usuário fique sem permissão caso desmarque o um dos campos de permissão.</p>

**Fonte:** Autores

**Quadro 11 - Caso de Uso – Salvar Alterações**

<b>Caso de Uso – Salvar Alterações</b>	
<b>ID</b>	UC 008
<b>Descrição</b>	Este caso de uso tem o objetivo de salvar as alterações realizadas nos cadastros de usuários.
<b>Ator Primário</b>	Administrador
<b>Pré-condição</b>	Ter passado pela UC 006 – Alterar Dados de Cadastro ou pela UC 007 – Alterar Nível de Permissão.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o administrador seleciona o campo de salvar alterações.</li> <li>2. O sistema irá aplicar a alteração no banco e no sistema, após inserção de novas informações.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	1a - Em qualquer momento o administrador pode sair do sistema.

**Fonte:** Autores

**Quadro 12 - Caso de Uso – Acessar o Sistema**

<b>Caso de Uso – Acessar o Sistema</b>	
<b>ID</b>	UC 009
<b>Descrição</b>	Este caso de uso tem o objetivo de acessar o sistema de gerenciamento de estoque.
<b>Ator Primário</b>	Conferente
<b>Pré-condição</b>	Possuir usuário e senha para acessar sistema
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>3. O use case inicia quando o conferente abre o software.</li> <li>4. O sistema carrega a tela de login.</li> <li>5. O usuário insere o usuário e a senha.</li> <li>6. O sistema valida as informações inseridas.</li> <li>7. O sistema abre a interface de sistema.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	<p>4a – Caso o usuário ou a senha estiver errada, o sistema informará que os dados estão divergentes.</p> <p>4b – Caso o usuário não soubesse o usuário ou senha, deverá procurar o administrador para restaurar os dados de acesso.</p>

**Fonte:** Autores**Quadro 13 - Caso de Uso – Alterar Senha**

<b>Caso de Uso – Alterar Senha</b>	
<b>ID</b>	UC 010
<b>Descrição</b>	Este caso de uso tem objetivo de realizar a troca de senha do usuário.
<b>Ator Primário</b>	Conferente
<b>Pré-condição</b>	Ter passado pela UC 009 – Acesso ao Sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o conferente seleciona o campo de alterar senha.</li> <li>2. O sistema abrirá um formulário.</li> <li>3. O usuário irá preencher os campos de senha atual, nova senha, repetir senha nova.</li> <li>4. O sistema vai validar as informações inseridas.</li> <li>5. O sistema vai salvar a nova senha no banco.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	<p>3a – Caso o usuário não lembre a senha atual, deverá entrar em contato com o administrador para realizar o reset da senha.</p> <p>3b – Caso a senha não seja preenchida com os requisitos de no mínimo de oito caracteres, contendo letras maiúsculas e minúsculas, números e caracteres especiais, a senha não será trocada e as informações não serão salvas</p>

**Fonte:** Autores**Quadro 14 - Caso de Uso – Solicitar Compra**

<b>Caso de Uso – Solicitar Compra</b>	
<b>ID</b>	UC 011
<b>Descrição</b>	Este caso de uso tem por objetivo fazer a solicitação de compra ao identificar que o estoque está baixo
<b>Ator Primário</b>	Compras
<b>Pré-condição</b>	Estar logado como o usuário que possua permissão de solicitar compras
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>6. O use case inicia quando o compras identifica que algum produto está com estoque baixo</li> </ol>

	7. O compras faz a contagem do produto
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	1a – nenhum produto possui o estoque baixo

**Fonte:** Autores

**Quadro 15 - Caso de Uso – Realizar Cotação**

<b>Caso de Uso – Realizar Cotação</b>	
<b>ID</b>	UC 012
<b>Descrição</b>	Este caso de uso tem objeto realizar a cotação do produto com baixo estoque
<b>Ator Primário</b>	Compras
<b>Pré-condição</b>	Ter passado pela UC 011 – Solicitar Compras
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o compras fez a contagem do produto e constata que o nível está baixo</li> <li>2. O compras seleciona os fornecedores do produto</li> <li>3. O compras faz a cotação dos produtos em falta</li> <li>4. O compra escolhe o melhor orçamento</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	2a – Lista de fornecedores está desatualiza 3a – Os fornecedores possuem os mesmos valores

**Fonte:** Autores

**Quadro 16 - Caso de Uso – Fazer Compras/ Pedido**

<b>Caso de Uso – Fazer Compras/ Pedido</b>	
<b>ID</b>	UC 013
<b>Descrição</b>	Este caso de uso tem objeto realizar a compra, solicitação do produto em baixo estoque
<b>Ator Primário</b>	Compras
<b>Pré-condição</b>	Ter passado pela UC 011 – Solicitar Compras e UC 012 – Realizar cotação
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o compras faz o pedido dos produtos ao fornecedor</li> <li>2. O compras informa as quantidades que deseja</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	2a – Produto está em falta no fornecedor

**Fonte:** Autores

**Quadro 17 - Caso de Uso – Receber Nota**

<b>Caso de Uso – Receber Nota</b>	
<b>ID</b>	UC 014
<b>Descrição</b>	Este caso de uso tem por objetivo receber a nota e identificar alguma divergência que possa constar na nota fiscal
<b>Ator Primário</b>	Conferente
<b>Pré-condição</b>	Estar logado como o usuário que possua permissão de conferente
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando conferente recebe a nota fiscal</li> <li>2. O sistema carrega o formulário.</li> <li>3. O conferente preenche os campos de número de nota, entregador, quantidade de produto, numeração, peso, marca etc.</li> </ol>

	4. O conferente salvar as informações obtidas da nota e produto. 5. O sistema valida as informações inseridas.
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	3a – Todos os campos do formulário e obrigatório, e caso não seja preenchido o sistema indicará os campos que estão em branco ou incorreto.

**Fonte:** Autores

**Quadro 18 - Caso de Uso – Identificar Divergências**

<b>Caso de Uso – Identificar Divergências</b>	
<b>ID</b>	UC 015
<b>Descrição</b>	Este caso de uso tem por objetivo identificar quais são as divergências que constam na nota recebida
<b>Ator Primário</b>	Conferente
<b>Pré-condição</b>	Estar logado como o usuário que possua permissão de conferente
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o conferente identifica quais são as divergências existentes</li> <li>2. O sistema carrega o formulário.</li> <li>3. O conferente informa a nota fiscal e o produto que está divergente.</li> <li>4. O conferente salva as divergências.</li> <li>5. O sistema validar as informações inseridas.</li> <li>6. O sistema reporta que é necessário a devolução do produto.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	3a – Os campos de nota fiscal e produto devem ser devidamente preenchidos, para que o sistema possa rejeitar o pedido.

**Fonte:** Autores

**Quadro 19 - Caso de Uso – Devolver Produto**

<b>Caso de Uso – Devolver Produto</b>	
<b>ID</b>	UC 016
<b>Descrição</b>	Este caso de uso tem por objetivo realizar a devolução dos produtos com divergência identificadas.
<b>Ator Primário</b>	Conferente
<b>Pré-condição</b>	Ter passado pela UC 012 - Identificar Divergências.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o conferente seleciona o campo devolver nota fiscal.</li> <li>2. O sistema verifica as informações da nota fiscal e produto.</li> <li>3. O sistema realiza a remoção dos produtos no estoque contidos na nota.</li> <li>4. O sistema rejeita nota fiscal.</li> <li>5. O sistema reporta o conferente, que pode ser devolvido os produtos, por conter divergências</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	nenhum

**Fonte:** Autores

**Quadro 20 - Caso de Uso – Conferir Nota**

<b>Caso de Uso – Conferir Nota</b>	
<b>ID</b>	UC 017
<b>Descrição</b>	Este caso de uso tem objetivo de conferir se todos os produtos que constam na nota foram de fato recebidos
<b>Ator Primário</b>	Estoquista

<b>Pré-condição</b>	Ter passado pela UC 011 – Receber Nota
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando estoquista seleciona o campo conferir nota.</li> <li>2. O sistema carrega as informações da nota.</li> <li>3. O estoquista confirma se a nota está correta, se equivale com os produtos encaminhados.</li> <li>4. O sistema valida a conferência.</li> <li>5. O sistema salva as informações no banco.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	3.a Se a nota estiver divergente dos produtos encaminhados, sistema reportar ao estoquista, que é necessário especificar qual é a divergência e devolver os produtos

**Fonte:** Autores

**Quadro 21 - Caso de Uso – Cadastrar Produtos**

<b>Caso de Uso – Cadastrar Produtos</b>	
<b>ID</b>	UC 018
<b>Descrição</b>	Este caso de uso tem objetivo de cadastrar produtos novos
<b>Ator Primário</b>	Estoquista
<b>Pré-condição</b>	Estar logado como o usuário que possua permissão de Estoquista
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando estoquista seleciona o campo de cadastrar produtos.</li> <li>2. O sistema carrega o formulário de cadastro de produto.</li> <li>3. O fiscal preenche os campos de nome do produto, marca, quantidade, data de fabricação, data de vencimento, número de lote, valor unitário, fabricante etc.</li> <li>4. O estoquista salva as informações do novo produto.</li> <li>5. O sistema valida as informações.</li> <li>6. O sistema salva as informações inserida no banco.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	3a – O sistema verifica se todos os campos foram preenchidos, caso não são marcados em vermelhos para ser corrigidos ou preenchidos.

**Fonte:** Autores

**Quadro 22 - Caso de Uso – Armazenar Produto**

<b>Caso de Uso – Armazenar Produto</b>	
<b>ID</b>	UC 019
<b>Descrição</b>	Este caso de uso tem objetivo alocar os produtos recebidos
<b>Ator Primário</b>	Estoquista
<b>Pré-condição</b>	Ter passado pela UC015 – Cadastrar Produtos
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o estoquista termina o cadastro e seleciona onde serão armazenados os produtos.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	nenhum

**Fonte:** Autores

**Quadro 23 - Caso de Uso – Fazer Pedido**

<b>Caso de Uso – Fazer Pedido</b>	
<b>ID</b>	UC 020
<b>Descrição</b>	Este caso de uso tem objetivo de registrar um pedido de um cliente

<b>Ator Primário</b>	Cliente
<b>Pré-condição</b>	Solicitar a um atendente que esteja logado para registrar o pedido
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o cliente solicita ao atendente um produto</li> <li>2. O sistema carrega o formulário de venda.</li> <li>3. O cliente informa os produtos desejados</li> <li>4. O sistema exibe o resumo do pedido do cliente, incluindo valores.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	3a - Todos os campos do formulário e obrigatório, e caso não seja preenchido o sistema indicará os campos que estão em branco ou incorreto.

**Fonte:** Autores

**Quadro 24 - Caso de Uso – Consultar Estoque**

<b>Caso de Uso – Consultar Estoque</b>	
<b>ID</b>	UC 021
<b>Descrição</b>	Este caso de uso tem objetivo de verificar se o produto
<b>Ator Primário</b>	Vendedor
<b>Pré-condição</b>	Ter logado com nível de permissão de vendedor.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o vendedor seleciona o campo de consultar estoque.</li> <li>2. O sistema carrega a lista de produtos existentes.</li> <li>3. O vendedor seleciona o produto que deseja.</li> <li>4. O sistema mostra a quantidade disponível e valor.</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	nenhuma

**Fonte:** Autores

**Quadro 25 - Caso de Uso – Registrar Pedido**

<b>Caso de Uso – Registrar Pedido</b>	
<b>ID</b>	UC 022
<b>Descrição</b>	Este caso de uso tem objetivo de registrar um pedido de venda.
<b>Ator Primário</b>	Atendente
<b>Pré-condição</b>	Ter passado pela UC 021 – Consultar Estoque.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o atendente seleciona o campo de registrar pedido.</li> <li>2. O sistema carrega o formulário de venda.</li> <li>3. O atendente preenche os campos nome do cliente, data de venda, produto, quantidade, valor total, forma de pagamento etc.</li> <li>4. O atendente salva o pedido.</li> <li>5. O sistema valida as informações.</li> <li>6. O sistema salva as informações no banco.</li> <li>7. O sistema gera as informações de pagamento</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	nenhuma

**Fonte:** Autores

**Quadro 26 - Caso de Uso – Emitir Ordem de Produção**

<b>Caso de Uso – Emitir Ordem de Produção</b>	
<b>ID</b>	UC 023
<b>Descrição</b>	Este caso de uso tem por objetivo emitir a ordem de produção do pedido realizado
<b>Ator Primário</b>	Atendente
<b>Pré-condição</b>	Ter passado pela UC 022 – Registrar Pedido
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o atendente finaliza o processo de registrar o pedido</li> <li>2. O sistema faz a impressão do pedido</li> <li>3. O atendente encaminha para a produção</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	nenhum

**Fonte:** Autores

**Quadro 27 - Caso de Uso – Realizar Pagamento**

<b>Caso de Uso – Realizar Pagamento</b>	
<b>ID</b>	UC 024
<b>Descrição</b>	Este caso de uso tem por objetivo permitir que o cliente realize o pagamento do pedido solicitado
<b>Ator Primário</b>	Cliente
<b>Pré-condição</b>	Ter passado pela UC 022 – Registrar Pedido
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o sistema gera as informações de pagamento.</li> <li>2. O atendente informa os valores e as forma de pagamento disponíveis</li> <li>3. O cliente informa a forma de pagamento.</li> <li>4. O cliente realiza o pagamento</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	4a – Caso cliente possua cupom de desconto deve ser inserido no momento de finalizar o pagamento.

**Fonte:** Autores

**Quadro 28 - Caso de Uso – Baixar Estoque**

<b>Caso de Uso – Baixar Estoque</b>	
<b>ID</b>	UC 025
<b>Descrição</b>	Este caso de uso tem objetivo de atualizar as informações do estoque.
<b>Ator Primário</b>	Atendente
<b>Pré-condição</b>	Ter passado pela UC 022 – Registrar Pedido ou UC 023 – Emitir Ordem de Produção
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o atendente finaliza o pedido do cliente e emite a ordem de produção</li> <li>2. O sistema atualiza a quantidade de produtos retirando a quantidade de produtos que constam no pedido do cliente.</li> <li>3. O sistema salva a informação no banco</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	1a – O cliente pode cancelar o pedido

**Fonte:** Autores

**Quadro 29 - Caso de Uso – Entregar Produto**

<b>Caso de Uso – Entregar Produto</b>	
<b>ID</b>	UC 026
<b>Descrição</b>	Permitir que o atendente entregue o produto que o cliente fez no pedido
<b>Ator Primário</b>	Atendente
<b>Pré-condição</b>	Ter passado pela UC 023 – Emitir Ordem de Produção
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o atendente entrega a ordem de produção para a cozinha.</li> <li>2. A cozinha monta o produto do cliente com os produtos selecionados</li> <li>3. A Cozinha entrega o produto pronto para o atendente</li> <li>4. A atendente vai até o cliente e faz a entrega do produto</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	4a – Se o pedido for realizado de forma online, atendente entrega pedido para o entregador

**Fonte:** Autores

**Quadro 30 - Caso de Uso – Receber Produto**

<b>Caso de Uso – Receber Produto</b>	
<b>ID</b>	UC 027
<b>Descrição</b>	Permitir que o cliente receba o produto que foi pedido
<b>Ator Primário</b>	Cliente
<b>Pré-condição</b>	Ter passado pela UC 023 - Fazer Pedido
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O use case inicia quando o cliente faz o pedido</li> <li>2. A Atendente registra o pedido e emite ordem de pagamento</li> <li>3. O Cliente realiza o pagamento do pedido e emite ordem de produção</li> <li>4. A Atendente encaminha para produção</li> <li>5. A Cozinha produz o pedido e devolve para a atendente</li> <li>6. A Atendente entrega o produto ao cliente</li> <li>7. O Cliente recebe o produto</li> <li>8. A Atendente confirma a entrega no sistema</li> </ol>
<b>Pós-condição</b>	nenhuma
<b>Cenário Alternativo</b>	nenhum

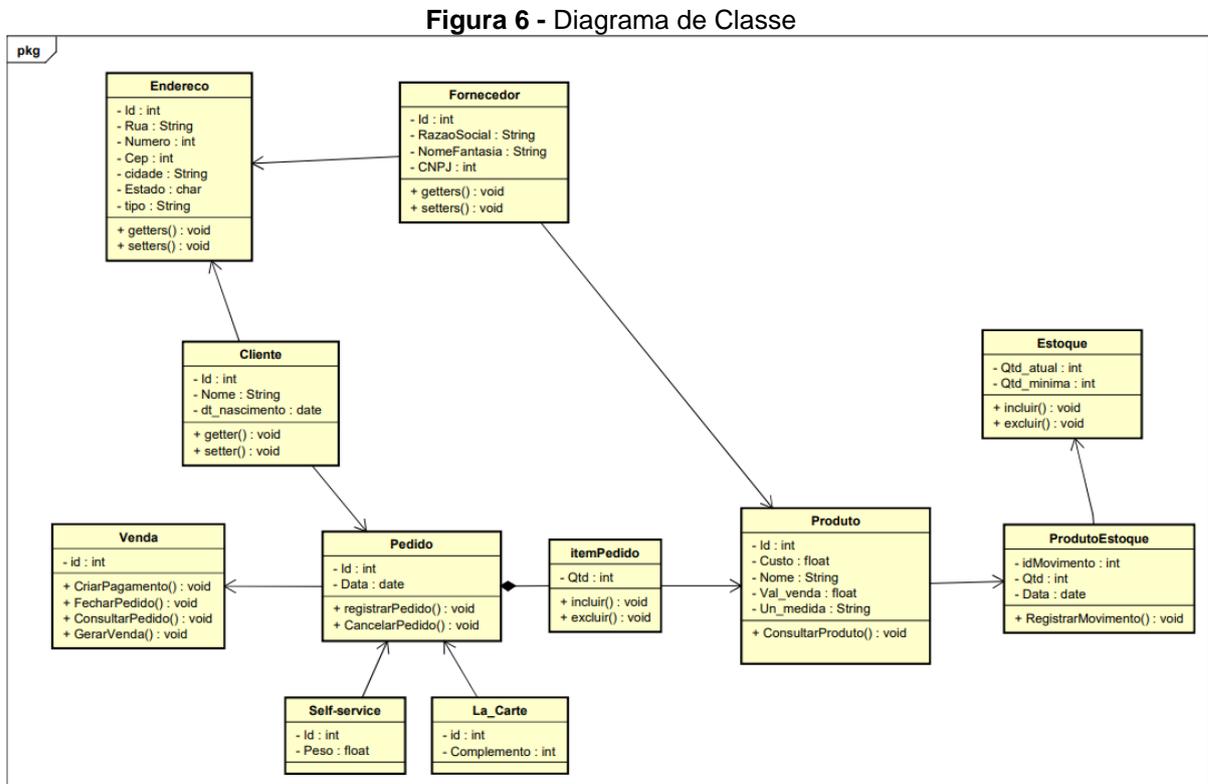
**Fonte:** Autores

### 3.7 Diagrama de Classes

Considerado um dos mais importantes diagramas e o mais utilizado no diagrama da UML, o diagrama de classe está focado em permitir a visualização das classes que constituirão o sistema. Através dele conseguimos ter uma visão dos atributos e métodos em que o sistema consiste, bem como expressar como é o relacionamento entre as classes.

O diagrama em si serve como base para que os demais diagramas da UML sejam construídos, ele mostra uma visão estática de como as classes serão organizadas e é basicamente composto por classes e associações entre elas (GUEDES, 2018).

A Figura 6 apresenta o diagrama de classe desenvolvido para este projeto, representando as classes existentes, seus relacionamentos bem como seus atributos e métodos.



Fonte: Autores

### 3.8 Diagrama de Atividades

O Diagrama de atividade possui ênfase na descrição dos passos que serão percorridos para a conclusão de uma atividade específica, e as condições para organizar comportamentos de baixo nível.

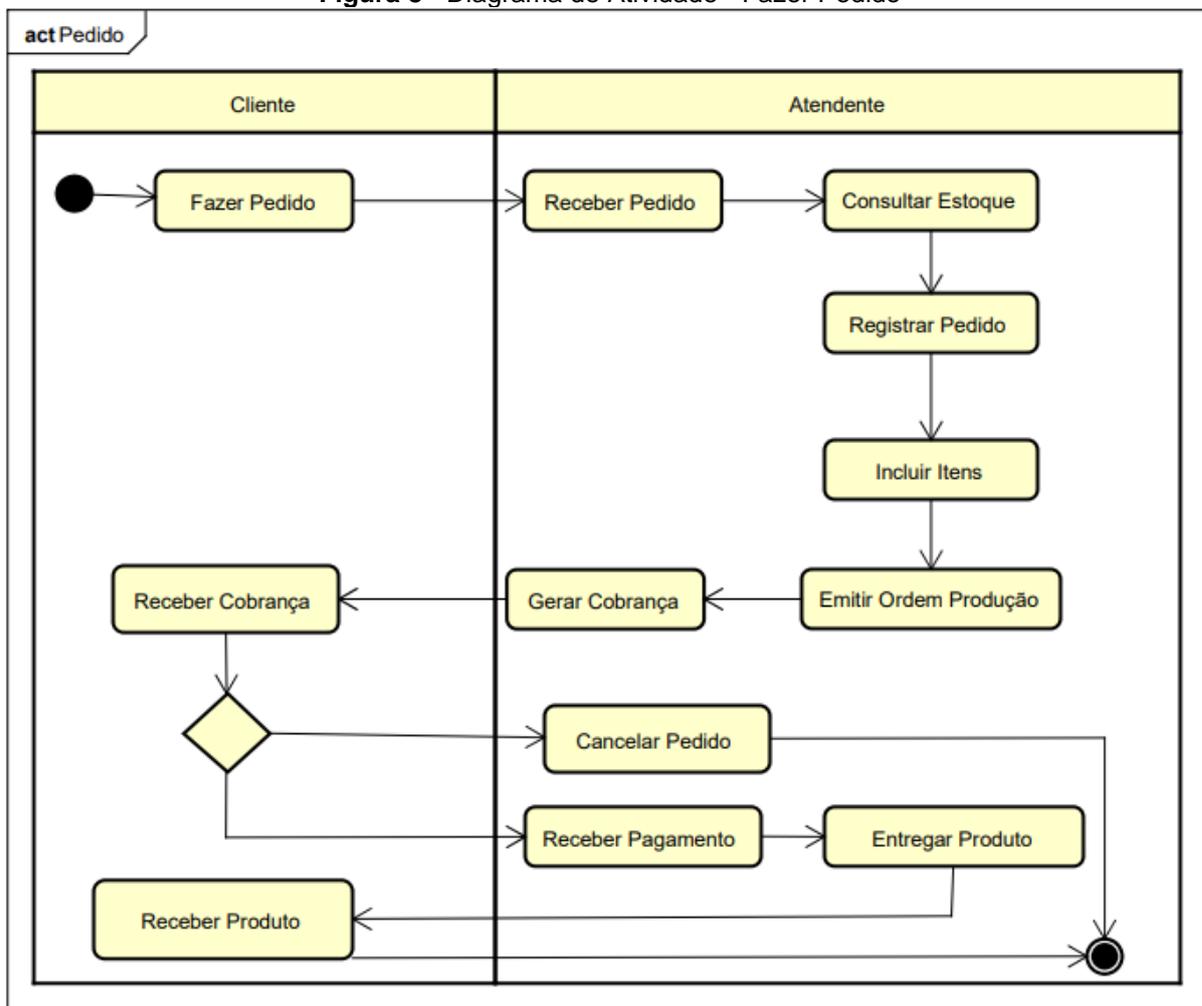
Este diagrama apresenta os fluxos de controle e de objetos entre as ações, que são representadas por símbolos geométricos, podendo incluir nós de tomada de decisão, bem como junções, bifurcações e sincronizações desde o início até o fim.

Para este projeto foram desenvolvidos os diagramas que mostram as atividades de efetuar o login no sistema e o processo de vendas, que estão representados nas Figuras 7 e 8, representando os passos que serão percorridos caso um login seja realizado de forma incorreta, e as ações que deverão ser executadas (GUEDES, 2018).

Na Figura 8 o processo apresentado é o de venda onde são descritos os passos que vão desde a realização do pedido, passa pela ação de atendimento até chegar a ação de entrega/receber produto.



**Figura 8 - Diagrama de Atividade - Fazer Pedido**



Fonte: Autores

### 3.9 Diagrama de Estados

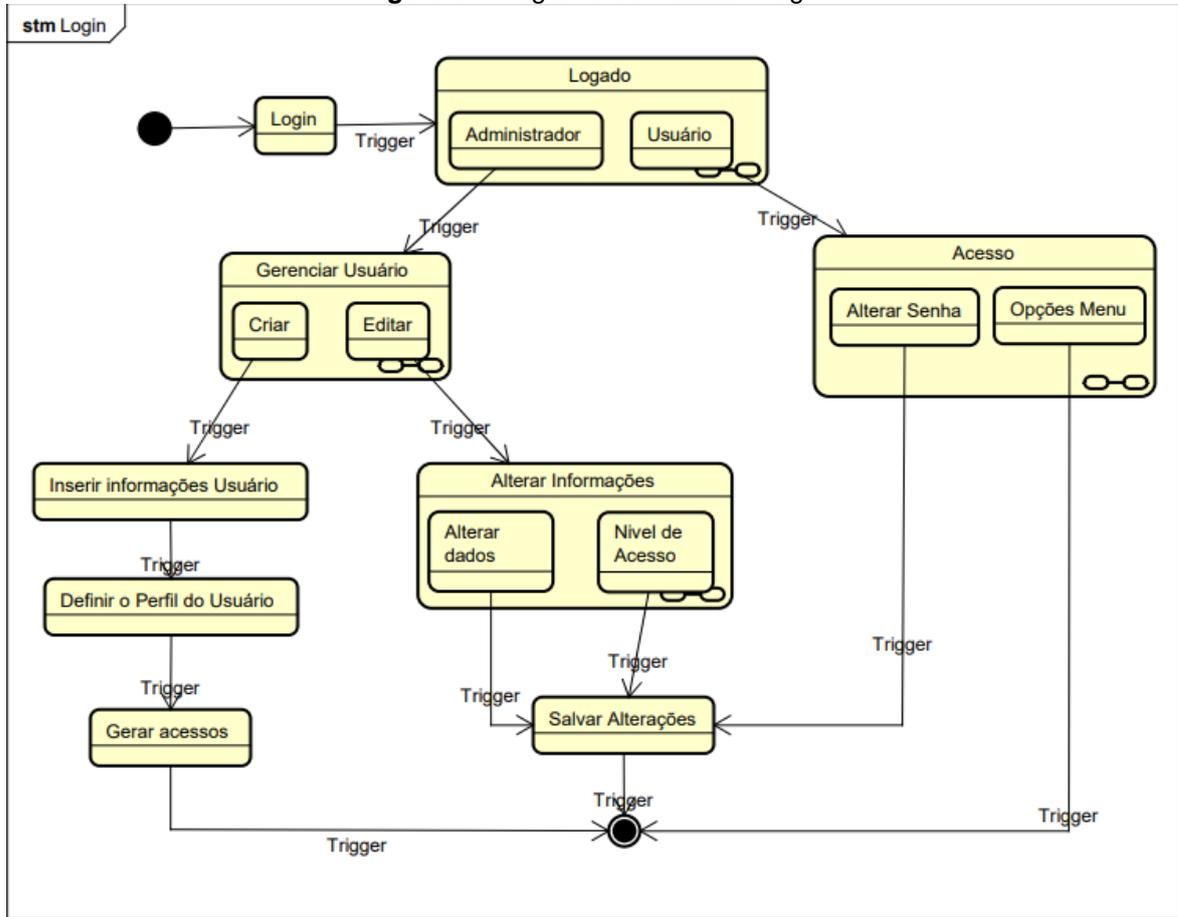
O Diagrama de Máquina de Estado, assim como o diagrama de sequência, é um diagrama comportamental, que demonstra a conduta de um elemento através das transições entre os estados de um objeto.

Usado para modelar o comportamento dinâmico de vários elementos do modelo ou uma parte do sistema, é composto por estados, eventos, transições e ações, entre outros. Onde um estado representa a situação que o objeto se encontra, o evento é o impulso que gera a transição, a transição é a alteração do estado do objeto e a ação é a atividade que acontece no decorrer de uma transição ou um estado (GUEDES, 2018).

As Figuras 9 e 10 apresentam os diagramas de estados login e pedidos respectivamente, que representam os processos de transição dos estados quando é

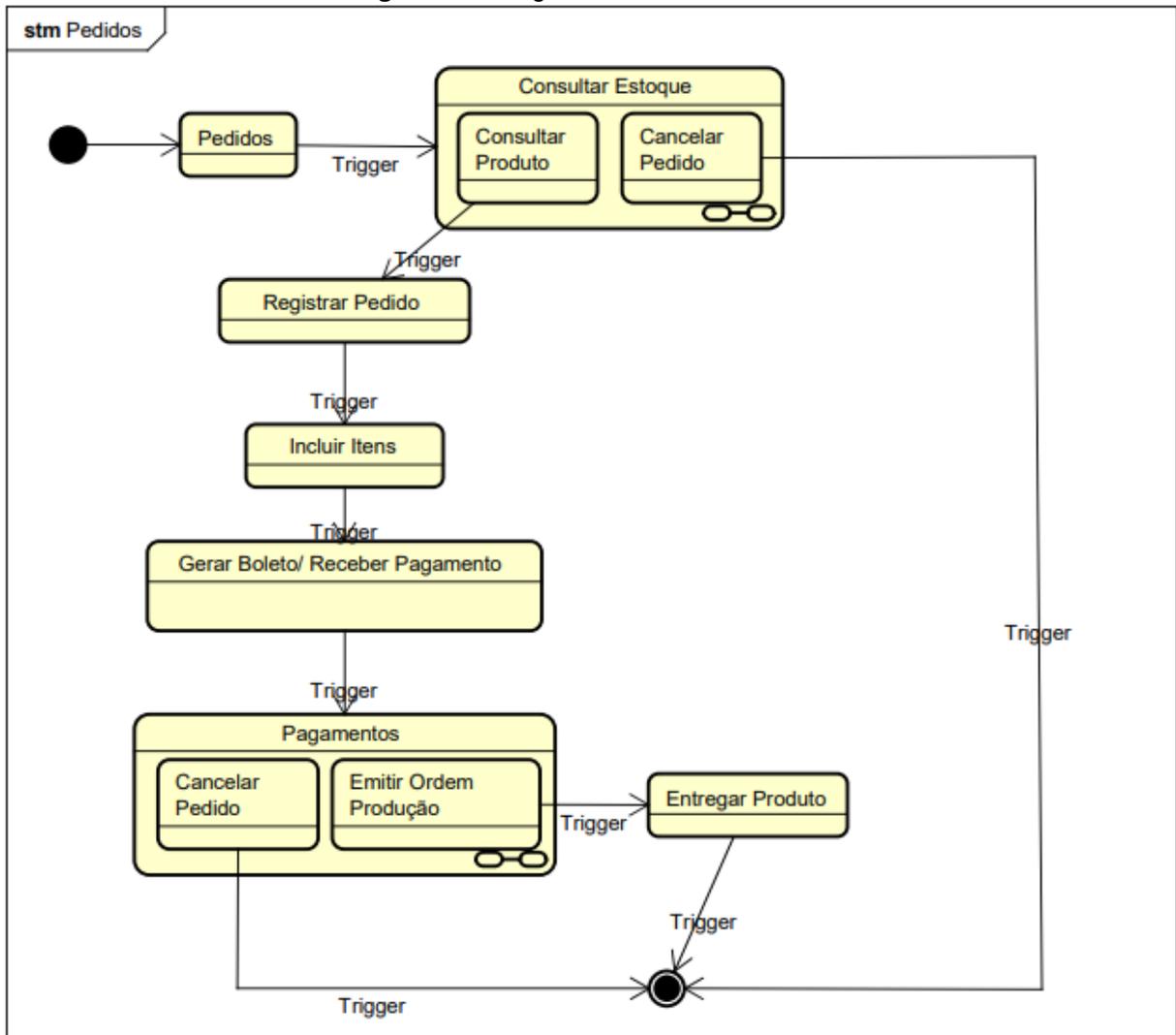
realizado o login no sistema, ou a alteração de senha por parte do cliente, e as transições que acontecem no processo de solicitação de pedido até a entrega do produto ao cliente.

Figura 9 - Diagrama de Estado - Login



Fonte: Autores

Figura 10 - Diagrama de Estado - Pedido



Fonte: Autores

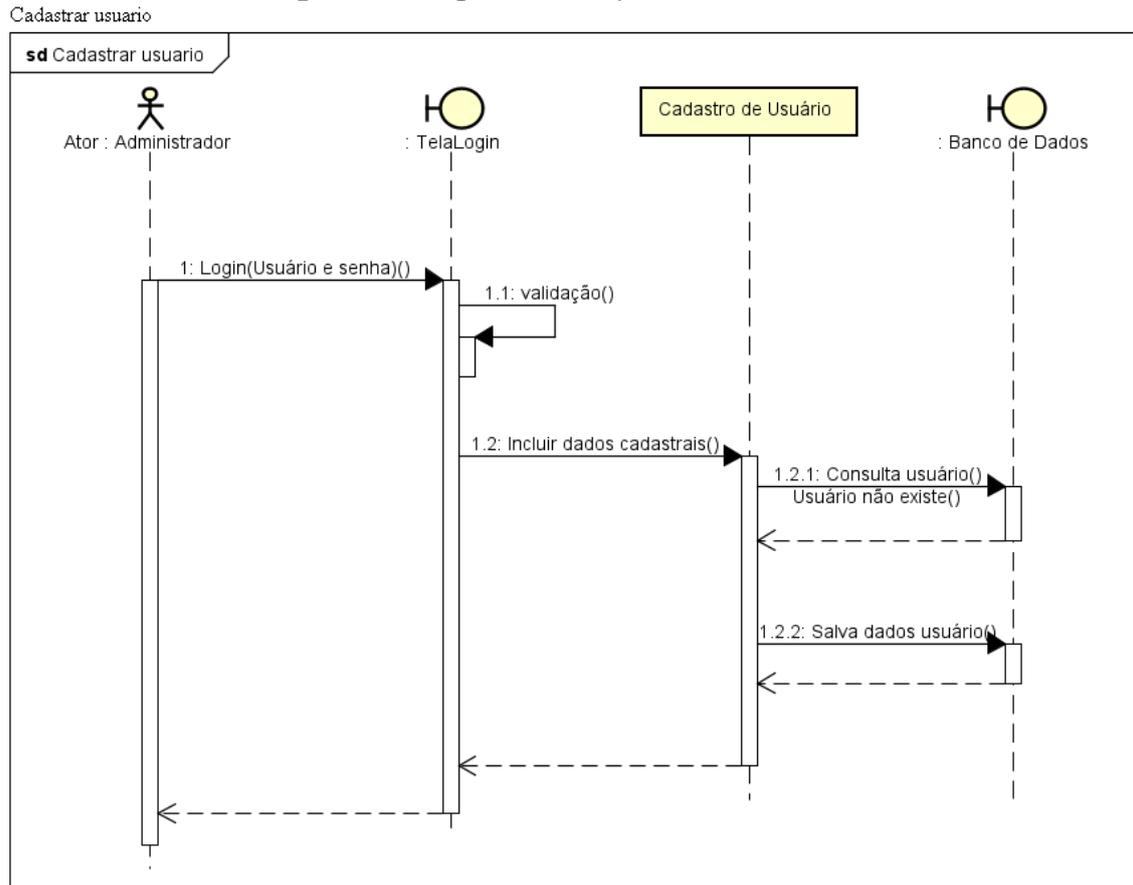
### 3.10 Diagrama de Sequência

O diagrama de sequência tem como função exibir a interação que ocorre entre os objetos do sistema de forma temporal, demonstrando a ordem cronológica do fluxo de mensagens trocadas entre os objetos, bem como sua colaboração. Tem como base geral os diagramas de caso de uso, juntamente com os diagramas de classe.

Este diagrama tem como utilidade modelar o comportamento dinâmico do sistema, bem como os atores que são responsáveis por tais eventos, determinando como o processo se desenrola e é concluído através das chamadas de mensagens trocadas entre os objetos (GUEDES, 2018).

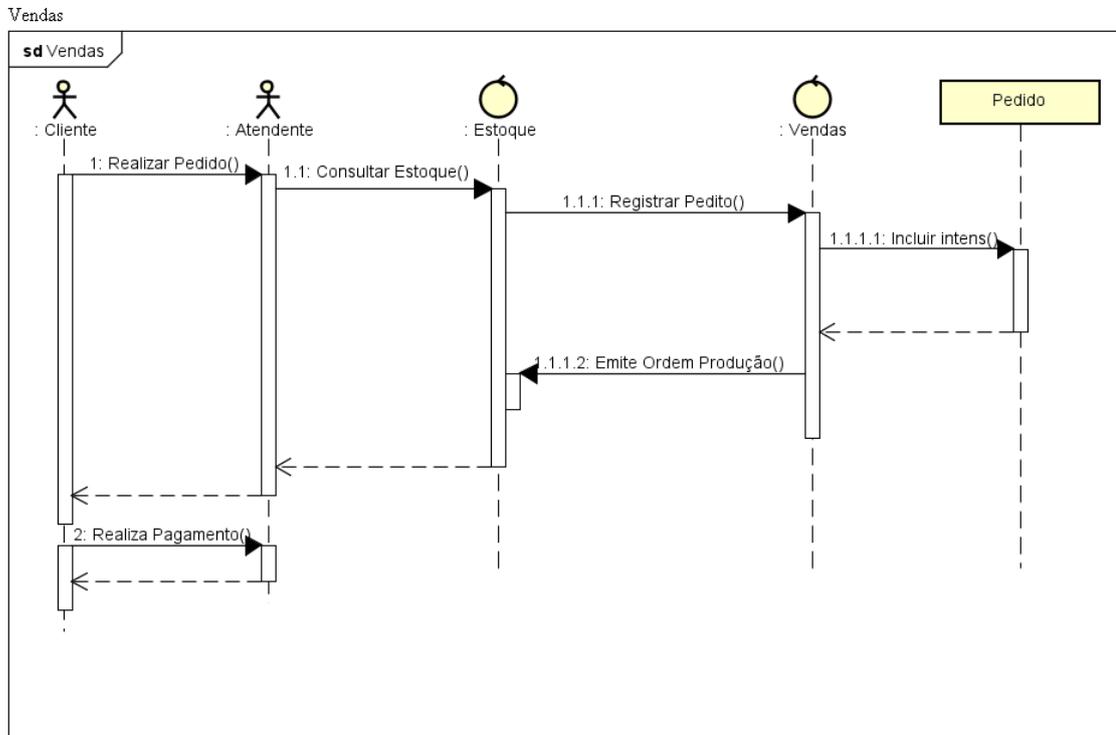
As sequências de cadastro de usuários e vendas foram representadas respectivamente nas Figuras 11 e 12, representando a ordem do fluxo da realização do login, desde o processo inicial até o armazenamento no banco de dados, representa também o fluxo de venda partindo da realização do pedido, os atores responsáveis pelas ações até a realização do pagamento do pedido.

**Figura 11 - Diagrama de Sequência - Cadastrar usuário**



Fonte: Autores

**Figura 12 - Diagrama de Sequência – Vendas**



**Fonte:** Autores

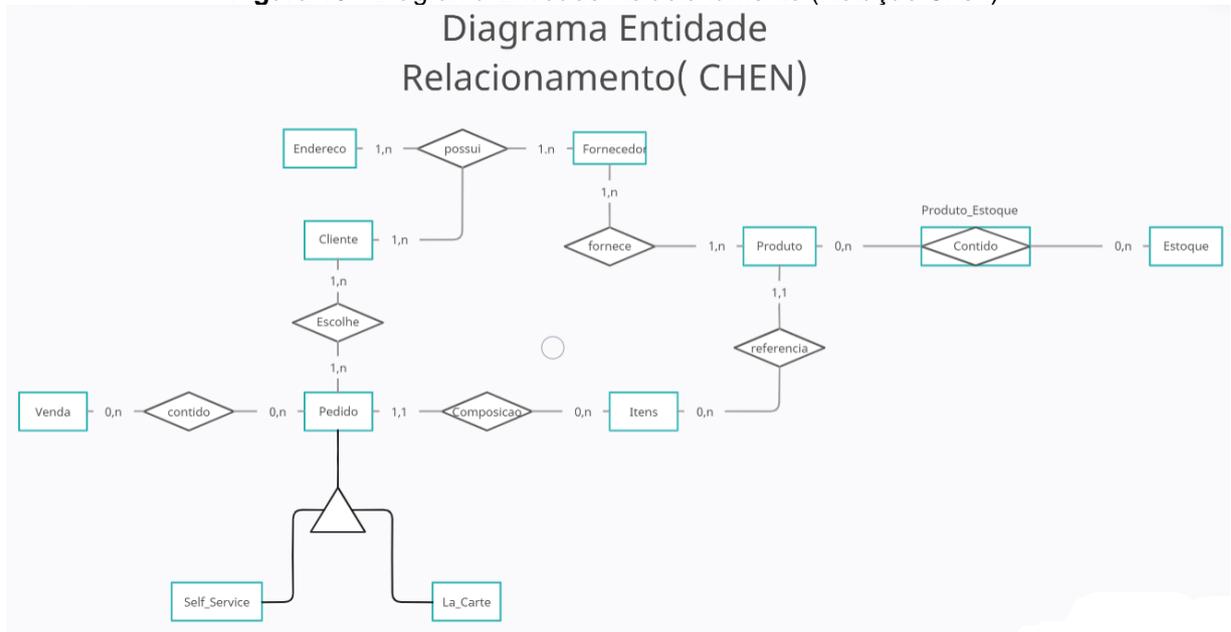
### 3.11 Diagrama Entidade Relacionamento

O Diagrama Entidade Relacionamento (DER) tem a função de descrever a estrutura conceitual e lógica de um banco de dados de forma gráfica, representando as entidades, os relacionamentos e os atributos de um sistema.

Sua modelagem deve ser mais próxima possível do mundo real, pois seu modelo será utilizado como base no desenvolvimento do projeto lógico e físico do banco de dados, é suma importância que possua uma metodologia simples, porém deve ser precisa e eficiente (MACHADO, 2020).

A Figura 13 apresenta o diagrama entidade relacionamento do projeto segundo a notação Chen, representando da forma mais real possível como é a ligação entre as entidades, suas cardinalidades.

**Figura 13 - Diagrama Entidade Relacionamento (Notação Chen)**



**Fonte:** Autores

## 4 Ferramentas e Métodos ou Desenvolvimento

### 4.1 Ferramentas

Como servidor de versionamento de código utilizamos o GitHub, além de ser uma ferramenta gratuita é possível acessá-la de diversos dispositivos. O uso do GitHub facilita o compartilhamento de código entre os integrantes do grupo, além de garantir integração contínua do projeto.

A interface para desenvolvimento do código utilizada foi o *Visual Studio Code*, trata-se de um editor de código *open source* criado pela Microsoft. Nele foi desenvolvido todo o código-fonte do sistema, utilizando ferramentas de autocomplete que auxiliam na agilidade da escrita e na supressão de erros, e ferramentas de integração com Git facilitando o versionamento do código.

Para implantação do código *backend* foi utilizado o *Docker*, um software *open source*, que utiliza contêineres para execução do código criando um ambiente virtual

específico para a aplicação, possibilitando que esta aplicação esteja funcional em qualquer outro ambiente, diferente de onde ele foi desenvolvido.

PostgreSQL foi o SGBD (Sistema Gerenciador de Banco de Dados) escolhido para criação do bando de dados, sua escolha se deve ao fato de ser uma ferramenta também *open source* e de fácil manuseio. Trata-se de um banco de dados relacional, considerado estável e confiável, possui suporte a diversos tipos de dados, incluindo dados diferentes do padrão SQL.

Para a estruturação da página, foi utilizada a linguagem de marcação HTML (*HyperText Markup Language*) criando todo o conjunto de regras e organizando todos os componentes visuais. O HTML é uma linguagem básica de fácil sintaxe com intuito de criar de páginas da *web*, onde são definidos o cabeçalho da página, contendo informações sobre a configuração da página e importação de *scripts*, e o corpo, onde são alocados os elementos.

Juntamente ao HTML também adicionamos ao código a linguagem CSS (*Cascading Style Sheets*) responsável pela estilização da página, que não houve utilização de bibliotecas ou *frameworks* referentes aos estilos que constam no projeto. Inserir a linguagem CSS trouxe diversas vantagens ao projeto, sendo o primeiro delas a economia no tempo de desenvolvimento e um código mais limpo. A utilização do CSS para estilização do projeto permite o reaproveitamento das linhas de código.

No desenvolvimento do *backend* da aplicação, optamos pela linguagem *Python*, uma linguagem de alto nível e de fácil entendimento, qualidades que proporcionam um código mais limpo e claro, utilizamos ainda *Django Rest*, um dos vários *frameworks* do *Python* que auxilia no desenvolvimento das APIs (*Application Programming Interface*) e bem como na criação do banco de dados.

As funcionalidades e lógica da página foram desenvolvidas, com a utilização da linguagem *JavaScript*, incluindo o framework *VueJS*. O *VueJS* aprimora o desenvolvimento com a componentização do código, de maneira a facilitar a reutilização de funções, formulários e demais itens que se repetem no código.

A criação da interface da aplicação foi elaborada com base nos desenhos produzidos no Figma, uma ferramenta de design e prototipação de projetos que nos permitiu ter uma visão de como seriam as telas, deixando um visual bonito e de fácil utilização.

Assim como a ferramenta citada anteriormente, o Canva também é uma plataforma de design gráfica on-line, utilizada por grandes empresas para criação de

visuais de comunicação que podem ser publicadas em diversos locais. Esta ferramenta teve um papel importante no desenvolvimento do nosso projeto, com ela elaboramos o nosso estudo de viabilidade do projeto, mostrando de forma versátil e sucinta os pontos favoráveis para o desenvolvimento da aplicação e onde ela poderia se enquadrar.

Por fim a diagramação de todo projeto foi realizada utilizando três ferramentas, sendo elas: AstaUML, BizAge Modeler e VisualParadigm, com elas conseguimos mostrar através dos diagramas todo o fluxo de funcionamento do nosso sistema.

#### 4.2 Métodos ou Desenvolvimento

O processo de desenvolvimento iniciou-se na documentação do projeto, com foco principal no levantamento dos requisitos e nas regras de negócio, possibilitando o entendimento geral da necessidade da criação do sistema. Ao concluir o levantamento dos requisitos passamos para a diagramação de todo o sistema para facilitar a etapa de codificação.

Em segundo momento iniciou-se a escolha das ferramentas e linguagens que seriam utilizadas para desenvolvimento e codificação do sistema. A prioridade foi a prototipação das telas, destacando suas funcionalidades, cores e dimensionamentos, garantindo que o visual estivesse de acordo os requisitos do cliente.

Por fim passou-se para a fase de codificação de um CRUD (*Create, Read, Update, Delete*), onde foi implementada as funcionalidades básicas do sistema. Durante essa fase, a cada passo concluído foi realizado os testes para garantir que as operações estavam corretas, seguindo as especificações descrita na documentação. Assim foi possível analisar o processo de manipulação dos dados em banco de dados. A finalização do projeto se deu com a conclusão da documentação e a finalização do CRUD, ocorrendo ambas as etapas simultaneamente.

### 5 Resultados e Discussão

A Figura 14 apresenta a tela inicial do sistema que contém uma imagem que representa a empresa com fundo amarelo, à esquerda da tela foi inserida um menu lateral com fundo roxo contendo algumas funcionalidades do sistema, esse menu possui um sensor que apresenta as opções do menu assim que ocorre o clique sobre os ícones.

**Figura 14** - Tela Inicial



**Fonte:** Autores

A Figura 15 mostra também a tela principal, porém com a descrição das opções que constam na barra de menu, ao clicar em alguma das opções listadas outra tela será apresentada.

**Figura 15** - Tela Inicial com menu



Fonte: Autores

A Figura 16 mostra a tela de usuários, que é composta por um formulário com os campos para cadastro e visualização dos dados das credenciais de acesso ao sistema para cada pessoa que irá acessá-lo. Nessa tela é executado todo o processo de CRUD dos usuários, a consulta é realizada ao digitar um código e passar para o próximo campo, ou simplesmente sair da edição de código.

Figura 16 - Tela de Cadastro de Usuários

The screenshot shows a web interface for user management. The title 'Usuários' is prominently displayed at the top center. Below the title, there are several input fields: 'Código', 'Nome', 'Usuário', and 'Senha'. There are also fields for 'Nível de acesso'. At the bottom of the form, there are four buttons: 'Cancelar', 'Excluir', 'Limpar', and 'Salvar'. The interface has a yellow background and a dark purple sidebar menu on the left.

Fonte: Autores

A Figura 17 mostra a tela de clientes para cadastro dos clientes que frequentam o estabelecimento.

**Figura 17 - Tela de Cadastro de Clientes**

**Clientes**

Código	Nome	Data de nascimento
<input type="text"/>	<input type="text"/>	<input type="text" value="dd/mm/aaaa"/>
Rua	Número	CEP
<input type="text"/>	<input type="text"/>	<input type="text"/>
Cidade	Estado	Tipo
<input type="text"/>	<input type="text"/>	<input type="text"/>

**Fonte:** Autores

A Figura 18 apresenta a tela para a realização do cadastro de produtos que serão comercializados na empresa.

**Figura 18 - Tela de Cadastro de Produtos**

**Produtos**

Código	Nome	Descrição
<input type="text"/>	<input type="text"/>	<input type="text"/>
Unidade de medida	Preço de custo	Preço de venda
<input type="text"/>	<input type="text"/>	<input type="text"/>
Último fornecedor		
<input type="text"/>		

**Fonte:** Autores

A Figura 19 apresenta a tela de fornecedores de produtos, que posteriormente serão vinculados a cada produto no campo de último fornecedor.

Figura 19 - Tela de Cadastro de Fornecedores

The image shows a web form titled 'Fornecedores' on a yellow background. On the left, there is a dark purple sidebar with icons. The form contains the following fields:

- Código
- CPF/CNPJ
- Razão Social
- Nome Fantasia
- Rua
- Número
- CEP
- Cidade
- Estado
- Tipo

At the bottom of the form, there are four buttons: 'Cancelar', 'Excluir', 'Limpar', and 'Salvar'.

Fonte: Autores

A Figura 20 apresenta o trecho do código contendo os *models* de endereço, telefone e tipo de contato. *Models* são representações de tabelas do banco de dados.

Figura 20 - *Models* da área de contatos

```
from django.db import models
from django.contrib.contenttypes.fields import GenericForeignKey
from django.contrib.contenttypes.models import ContentType

class ContactType(models.TextChoices):
    RESIDENTIAL = "R", "Residencial"
    COMMERCIAL = "C", "Comercial"

class Address(models.Model):
    #Fields
    type = models.TextField(choices=ContactType.choices)
    street = models.TextField(max_length=100)
    number = models.IntegerField()
    complement = models.TextField(max_length=50)
    postal_code = models.TextField(max_length=8)
    city = models.TextField(max_length=50)
    state = models.TextField(max_length=2)

class Telephone(models.Model):
    #Fields
    type = models.TextField(choices=ContactType.choices)
    area_code = models.IntegerField()
    number = models.TextField(max_length=9]
```

Fonte: Autores

A Figura 21 mostra um trecho de código contendo o *endpoint* de fornecedores. No *framework Django*, os *endpoints* são criados através de *viewsets*, que são utilizados para os processamentos das requisições enviadas pelo *front-end*.

Figura 21 - Endpoint Fornecedores

```
1 from rest_framework import viewsets, status
2 from .models import Provider
3 from .serializers import ProviderSerializer
4 from contacts.models import (
5     Address,
6     Telephone,
7     ContactType
8 )
9 from django.http import JsonResponse
10 from rest_framework.response import Response
11
12 class ProviderViewSet(viewsets.ModelViewSet):
13     queryset = Provider.objects.all()
14     serializer_class = ProviderSerializer
15
16     def get_queryset(self):
17         queryset = self.queryset
18         if self.request.GET.get("name"):
19             queryset = queryset.filter(
20                 name=self.request.GET.get("name")
21             )
22         if self.request.GET.get("id"):
23             queryset = queryset.filter(
24                 id=self.request.GET.get("id")
25             )
26         if self.request.GET.get("corporate_name"):
27             queryset = queryset.filter(
28                 corporate_name=self.request.GET.get("corporate_name")
29             )
30         if self.request.GET.get("cnpj"):
31             queryset = queryset.filter(
32                 cnpj=self.request.GET.get("cnpj")
33             )
34         return queryset
35
36     def create(self, request):
37         addresses_data = request.data.pop("address")
38         telephones_data = request.data.pop("telephone")
39         provider = Provider.objects.create(**request.data)
40         for address_data in addresses_data:
41             address_data["type"] = ContactType.objects.get(id=address_data["type"])
42             Address.objects.create(content_object=provider, **address_data)
43         for telephone_data in telephones_data:
44             telephone_data["type"] = ContactType.objects.get(id=telephone_data["type"])
45             Telephone.objects.create(content_object=provider, **telephone_data)
46         serializer = self.get_serializer(data=request.data)
47         serializer.is_valid()
48         headers = self.get_success_headers(serializer.data)
49         return JsonResponse(serializer.data, status=status.HTTP_201_CREATED, headers=headers)
50
51     def delete(self, request):
52         provider = Provider.objects.get(
53             id=request.GET.get("id")
54         )
55         provider.delete()
56         return Response(status=status.HTTP_200_OK)
57
```

Fonte: Autores

A Figura 22 mostra o código do serializer de produtos, ele define quais dados serão enviados e recebidos entre os *models* e *endpoint*.

Figura 22 - Serializer de Produtos

```
1  ✓ from rest_framework.serializers import ModelSerializer
2    from .models import Product
3    from providers.serializers import ProviderSerializer
4
5  ✓ class ProductSerializer(ModelSerializer):
6      provider = ProviderSerializer()
7  ✓   class Meta:
8          ordering = ["id"]
9          model = Product
10  ✓   fields = [
11           "id",
12           "provider",
13           "name",
14           "description",
15           "cost_value",
16           "sell_value",
17           "measurement_unit",
18       ]
19
```

Fonte: Autores

A Figura 23 mostra uma parte do código que gera o componente *sidebar*, utilizado como menu de navegação entre os módulos do sistema.

Figura 23 - Componente *Sidebar*

```
1 <template>
2   <div class="sidebar" :class="{ collapsed: isOpen }">
3     <ul>
4       <li class="menu-button" v-for="menuItem in menuItems" :key="menuItem.id"
5         @click="toggleSidebar(menuItem.action)">
6         <i class="material-icons-outlined">{{ menuItem.icon }}</i>
7         <span class="button-text" :class="{ hidden: !isOpen }">{{ menuItem.text }}</span>
8       </li>
9     </ul>
10  </div>
11 </template>
12
13 <script>
14 import { EventBus } from "@eventBus";
15
16 export default (await import('vue')).defineComponent({
17   name: "side-bar",
18   props: {
19     menuItems: {
20       type: Object,
21       required: true
22     },
23   },
24   data() {
25     return {
26       isOpen: false,
27     };
28   },
29   methods: {
30     toggleSidebar(action) {
31       if (this.isOpen && action) {
32         EventBus.emit("optionSelected", action);
33       }
34       this.isOpen = !this.isOpen;
35     },
36     closeSidebar() {
37       this.isOpen = false;
38     }
39   },
40   mounted() {
41     document.addEventListener("click", e => {
42       if (!this.$el.contains(e.target)) {
43         this.closeSidebar();
44       }
45     });
46   }
47 });
48 </script>
49
```

Fonte: Autores

A Figura 24 mostra o trecho do código contendo o componente que monta a página para exibição do formulário, onde ocorrerá a exibição, criação, exclusão e alteração dos dados do cliente.

Figura 24 - Componente Página de Clientes

```

1 <template>
2   <ContentPage :form-elements=formElements page-name="Clientes" endpoint="clients/"></ContentPage>
3 </template>
4
5 <script>
6   import ContentPage from "@/components/ContentPage.vue"
7   export default (await import('vue')).defineComponent({
8     name: "clients-page",
9     data() {
10      return {
11        formElements: [
12          {id:"id", name:"Código", type:"number"},
13          {id:"name", name:"Nome", type:"text"},
14          {id:"birth_date", name:"Data de nascimento", type:"date"},
15          {id:"postal_code", name:"CEP", type:"text"},
16          {id:"street", name:"Rua", type:"text"},
17          {id:"number", name:"Número", type:"number"},
18          {id:"city", name:"Cidade", type:"text"},
19          {id:"state", name:"Estado", type:"text"},
20          {id:"address_type", name:"Tipo de endereço", type:"text"},
21          {id:"telephone", name:"Telefone", type:"text"},
22          {id:"telephone_type", name:"Tipo de telefone", type:"text"}
23        ],
24      }
25    },
26    components: {
27      ContentPage
28    },
29    methods: {
30      getContactTypes() {
31        // const address_type = document.getElementById("address_type")
32        // const telephone_type = document.getElementById("telephone_type")
33        this.$http.options('contacts/types/')
34        .then(response => {
35          console.log(response)
36        })
37        .catch(error => {
38          console.log(error)
39        })
40      },
41    },
42    beforeMount() {
43      this.getContactTypes()
44    }
45  })
46 </script>

```

Fonte: Autores

Os trechos de código apresentados nas Figuras 20 a 24 apresentam uma linguagem pura do Framework *Django*, pertencente à linguagem de programação Python. Neles não existem associações com qualquer linguagem de marcação, como HTML ou CSS, embora elas também façam parte do projeto. Como essas marcações são comumente utilizadas para desenvolvimento de projetos, não se fez necessária explorá-las neste contexto.

## Considerações Finais

O incentivo para o desenvolvimento deste projeto, surgiu da necessidade destacada pelo nosso cliente, que alega não encontrar softwares no mercado que atendam devidamente a necessidade de sua açaiteria/sorveteria.

A ideia principal do projeto, era entregar um sistema completamente funcional, com integração com uma balança digital, para agilizar o processo de vendas self-service, onde o preço do produto cadastrado como venda por peso, seria aplicado diretamente na tela de vendas, multiplicado pelo peso presente na balança.

Ao longo do projeto surgiram as dificuldades para adquirir uma balança que fosse compatível com a aplicação para que pudéssemos integrá-la ao sistema e assim seguir com o desenvolvimento e testes, a partir contratempo optamos por fazer um sistema para controle de vendas, porém sem a opção da balança, pois trata-se de um modulo com grau de complexidade elevada, o que por tomaria muito tempo de desenvolvimento, para a realização da integração seria necessário modelos específicos de balança que possuem um valor elevado, o que acarretaria no aumento significativo nos custos do desenvolvimento do projeto.

Nosso maior desafio dentre todas as dificuldades encontradas ao longo do projeto foi a falta de experiência dos integrantes em relação à codificação do *front-end*, o que exigiu maior empenho na execução e um grande esforço durante o processo de aprendizagem.

O *Back-end* foi uma das partes menos trabalhosas, isso deve-se ao fato de já existir experiências anteriores com esse tipo de codificação, sendo assim o protótipo está sendo entregue com os módulos de APIs principais funcionando.

O Protótipo apresentado ainda não possui o modulo de vendas, entretanto já é possível demonstrar algumas funções.

## Referências

GUEDES, Gilleanes Thorwald Araujo. **UML 2 - Uma Abordagem Prática**. 3ª ed. São Paulo: Novatec, 2018.

LIMA, Paula. Artigo Para que serve o Estudo de Viabilidade? Disponível em: <https://www.linkedin.com/pulse/para-que-serve-o-estudo-de-viabilidade-paula-lima/?originalSubdomain=pt>. Acesso em: 06.mar.2023.

MACHADO, Felipe Nery Rodrigues. **Banco de dados: Projeto e Implementação** 4ª ed. São Paulo: Erica, 2020.

SEBRAE. **Alimentos e Bebidas: Como montar um restaurante self-service:** Disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/ideias/como-montar-um-restaurante-self-service,8c287a51b9105410VgnVCM1000003b74010aRCRD#apresentacao-de-negocio>>. Acesso em: 22.set.2022.

SILVA, Paulo César Barreto da. **Artigo SQL Magazine 63 – Utilizando UML: Diagrama de Classes.** Disponível em: <<http://www.devmedia.com.br/artigo-sql-magazine-63-utilizando-uml-diagrama-de-classes/12251>>. Acesso em: 18.abr.2023.

SOMMERVILLE, Ian. **Engenharia de Software.** 9ª ed. Tradução Kalinka Oliveira e Ivan Bosnic. São Paulo: Pearson Prentice Hall, 2011.

SYDLE. **Artigo Notação BPMN: Como aplicar para modelar Processos? Entenda etapas:** Disponível em: <<https://www.sydle.com/br/blog/notacao-bpmn-5ef510823130175de40cc4c2>>. Acesso em: 23.fev.2023.