

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

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**CAROLINE VAZ BARTO
LARISSA GONÇALVES FERREIRA**

SWEET ORDER:

sistema web de encomendas para confeitheiros

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

Sweet Order: sistema web de encomendas para confeiteiros

Caroline Vaz Barto¹

Larissa Gonçalves Ferreira²

Resumo

No contexto atual, é comum encontrar uma grande parcela da administração de negócios no setor de confeitarias operando de maneira amadora, muitas vezes conduzida por um único indivíduo. Essa realidade evidencia a necessidade de soluções que combinem praticidade e rapidez para lidar com esses desafios, por exemplo, a gestão de encomendas. Com o intuito de auxiliar os confeiteiros a superarem essas adversidades, este projeto propõe o desenvolvimento de um sistema *web* especialmente voltado para a gestão comercial das confeitarias, que terá sua conclusão por completo futuramente. Para este trabalho foram realizadas apenas as funcionalidades de cadastro, edição, exclusão e visualização de produtos.

Palavras-chave: Atendimento. Confeitarias. Encomendas. Gerenciamento.

Abstract

In the current context, it is common to find a large portion of business management in the confectionery sector operating in an amateur manner, often conducted by a single individual. This reality highlights the need for solutions that combine practicality and speed to address these challenges, such as order management. To assist confectioners in overcoming these difficulties, this project proposes the development of a web system specifically aimed at the commercial management of confectioneries, which will be fully completed in the future. For this paper, only the functionalities for registering, editing, deleting, and viewing products were implemented.

Keywords: Bakeries. Management. Order. Service.

1 - Introdução

No contexto das confeitarias, os proprietários frequentemente desempenham múltiplas funções como profissionais autônomos, assumindo responsabilidades que vão desde a gestão do negócio até a produção e o atendimento ao cliente. Essa carga de trabalho pode resultar em desafios para oferecer um atendimento adequado aos clientes. Diante dessa realidade, exploramos a possibilidade de como um sistema *web*

¹ Graduanda em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: [carol.vaz2013@gmail.com].

² Graduanda em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: [larissa.goferreira@gmail.com]

pode ser uma ferramenta eficaz para melhorar o gerenciamento e o atendimento nas confeitarias.

Para validar a hipótese proposta, foram conduzidas entrevistas com 11 confeitários por meio do aplicativo *WhatsApp*. Essas entrevistas proporcionaram informações valiosas sobre os desafios enfrentados pelos profissionais do ramo e permitiram a validação da hipótese de que um sistema *web online* poderia ser uma ferramenta eficaz para aprimorar o gerenciamento dos empreendimentos.

O objetivo inicial desse projeto consiste em desenvolver um sistema de fácil acesso, com o propósito de simplificar tanto o processo de vendas quanto o atendimento. Através dele, os confeitários terão a oportunidade de criar suas próprias lojas *online*, realizar o cadastro de seus produtos e disponibilizá-las aos clientes. Por meio desse sistema, os clientes poderão navegar pelos produtos, ler suas descrições e concluir suas compras.

Para esse projeto foi construído um CRUD de gestão de produtos de confeitaria, desenvolvendo as telas e códigos que permitirão o cadastro, edição, visualização e exclusão de produtos, que serão descritos nos próximos capítulos.

Foram utilizadas técnicas como elicitação e análise de requisitos, identificação de pontos fracos, elaboração de diagramas e outros artefatos da engenharia de *software* para criar toda a documentação necessária que sustenta a proposta delineada. Esses processos foram essenciais para garantir um desenvolvimento eficaz do projeto, possibilitando também a revisão e aprimoramento de alguns requisitos ao longo do ciclo de desenvolvimento.

O projeto foi segmentado em dois escopos de desenvolvimento: o *back-end* e o *front-end*. Antes de iniciar o desenvolvimento do *front-end*, realizamos a prototipagem das telas para visualizar a disposição dos elementos. O *back-end* foi desenvolvido em *TypeScript*, empregando conhecimentos avançados em estrutura de dados.

1.1 - Termo da Abertura do Projeto (TAP)

O documento do Termo de Abertura do Projeto tem como objetivo formalizar o início do projeto e descrever todas as suas etapas, justificativas, premissas, riscos, restrições, marcos e partes interessadas (COUTINHO, 2020, *online*).

1.1.1 - Objetivo do Projeto

O objetivo deste projeto consiste na implementação de um sistema *web*, voltado especificamente para a gestão comercial no ramo de confeitarias, proporcionando aos proprietários de estabelecimentos do segmento a facilidade de cadastrar seus produtos de maneira simplificada. Além disso, também viabilizará o acesso aos consumidores interessados em adquirir os produtos das confeitarias cadastradas.

1.1.2 - Justificativa do Projeto

Muitos profissionais do ramo enfrentam obstáculos na gestão de seus empreendimentos, na produção dos itens e no atendimento aos clientes. O propósito deste sistema é simplificar os processos de gestão e atendimento, facilitando as operações comerciais desses profissionais.

1.1.3 - Descrição do Projeto

O projeto abrangerá a análise dos requisitos e a elaboração de toda a documentação essencial para conceber uma solução sistêmica. O sistema será desenvolvido como uma página *web*, permitindo acesso por meio de qualquer dispositivo com um navegador compatível, utilizando tecnologias como *TypeScript*, HTML e CSS.

1.1.4 - Funcionalidades do Projeto

Cadastrar Usuário (proprietário/cliente).

Cadastrar Produto.

Comprar Produtos.

Realizar Pagamentos.

Acompanhar Pedidos.

Para esse projeto foi executado apenas a funcionalidade de cadastrar produto, no qual também permite sua edição, exclusão e visualização. As outras funcionalidades serão desenvolvidas futuramente.

1.1.5 - Premissas do Projeto

Os usuários finais precisam de um dispositivo com conexão de internet.

Os usuários precisarão ter o conhecimento básico de informática.

Os usuários precisarão se cadastrar.

1.1.6 - Riscos do Projeto

Atrasos no cronograma devido a problemas técnicos ou falta de recursos.

Mudanças nos requisitos do projeto durante a implementação.

Falta de aderência dos usuários finais.

2 - Viabilidade do Projeto

A análise de viabilidade de projetos é um tipo de estudo que precede as fases de planejamento de um *software*. Neste capítulo será avaliado se o sistema é realmente viável no contexto de confeitaria, elencando os benefícios que serão alcançados no desenvolvimento.

2.1.1 - Segmento de Clientes

O sistema atenderá a confeitadores profissionais que desejam expandir seus negócios *online*, tal como, clientes em busca de produtos de confeitaria personalizados.

2.1.2 - Proposta de Valor

A proposta de valor consiste na criação de um sistema *web*, com interface simplificada, que oferece ao confeitador a organização de seus pedidos e disponibilização de seus produtos *online*.

2.1.3 - Canais

Serão utilizadas redes sociais, *marketing* digital e anúncios *online*. Através desses canais, espera-se atingir confeitadores de várias localidades.

2.1.4 - Relacionamento

Para melhorias futuras, os confeitadores terão suporte *online* para atender eventuais dúvidas sobre a plataforma. Os clientes finais poderão manter contato com os confeitadores até a entrega de seus pedidos.

2.1.5 - Fontes de Receita

A principal remuneração serão as mensalidades pagas pelos confeitadores para utilizarem a plataforma. Será oferecido diferentes planos, com preços maiores para acessar todas as funcionalidades.

Além disso, será cobrada do cliente final uma pequena taxa por pedido realizado.

2.1.6 - Recursos Chave

Para alcançar os objetivos, o projeto terá recursos-chave, como uma dupla de desenvolvimento, ferramentas como Node.js e *TypeScript*, e um banco de dados Prisma - ORM. Além de recursos físicos como *notebooks* e acesso à *internet* para pesquisas e desenvolvimento.

2.1.7 - Atividades Chave

No que diz respeito às atividades chave, com certeza o ponto focal do projeto será a criação de um sistema *web* para confeitadores. A facilidade de cadastrar e inserir as informações de seus produtos será o diferencial.

O painel que mostra todos os pedidos feitos, organizado por datas e com todas as informações do pedido e cliente, trará ainda mais recursos de gestão de negócio aos empreendedores.

A criação desse sistema trará também ao cliente final uma boa forma de realizar pedidos de produtos de confeitarias, podendo escolher seus produtos e fazer o pagamento em qualquer lugar e a qualquer momento do dia.

2.1.8 - Parcerias Chave

As possíveis parcerias que poderão ajudar o projeto a se tornar um bom produto serão aplicativos de entrega, para proporcionar transporte adequado aos produtos. Bem como, empresas que vendem insumos e suprimentos para confeitaria, essa parceria poderá trazer um diferencial e benefícios aos confeitadores que adquirirem nossos planos.

2.1.9 - Estrutura de Custos

Incluem custos com ferramentas de desenvolvimento, manutenção, *marketing* e futuras parcerias.

2.1.10 – CANVAS

A viabilidade do projeto foi realizada com base no CANVAS, onde foi possível elencar os principais pontos do sistema, que além de analisá-lo também compreende fatores externos facilitando na gestão do negócio, conforme exemplificado por Fermann:

O Business Model Canvas (BMC) é uma ferramenta de Gestão de Negócios que nasceu em 2004 a partir da tese de doutorado de Alex Osterwalder. Em 2010 essa ferramenta se popularizou com a publicação do livro, de sua autoria, intitulado “Business Model Generation”. A grande função do BMC é ajudar na concepção de novos modelos de negócio e refletir sobre sua viabilidade (Fermann, 2020, *online*).

Figura 1: CANVAS



Fonte: os autores.

2.2 - Matriz SWOT

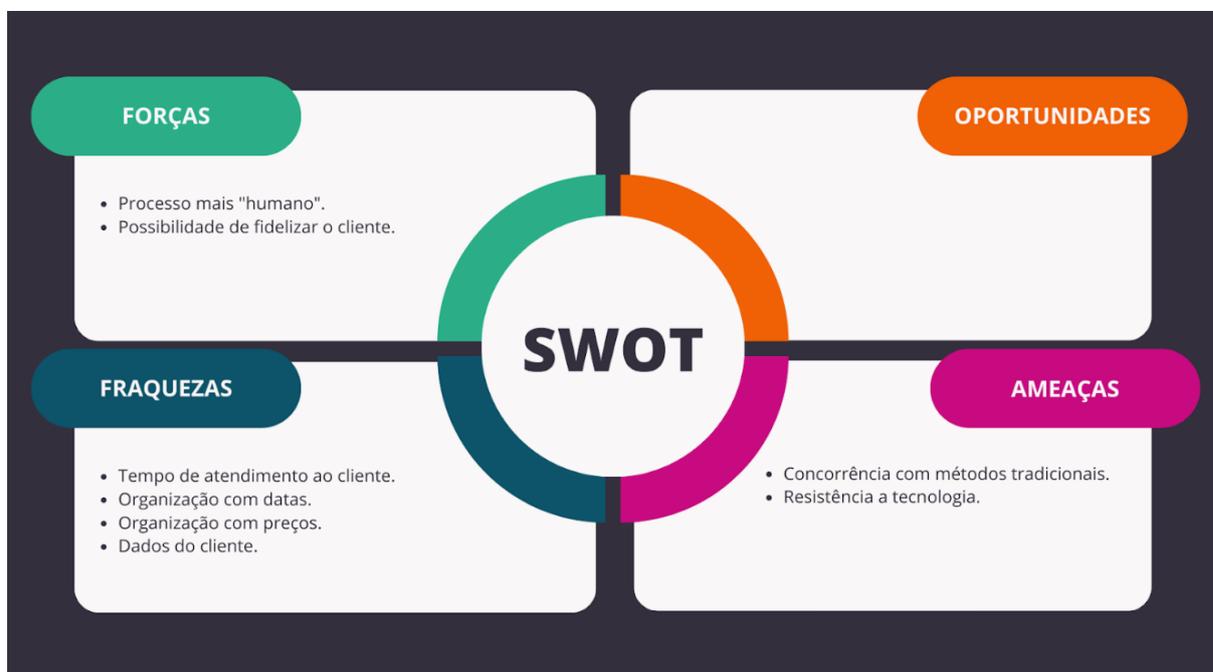
A análise SWOT foi desenvolvida na década de 1960 por Albert S. Humphrey, um consultor de gestão da empresa de consultoria *Stanford Research Institute*, durante um projeto de pesquisa financiado pela *Fortune 500*.

A sigla vem de *Strengths* (forças), *Weaknesses* (fraquezas), *Opportunities* (oportunidades) e *Threats* (ameaças), é uma ferramenta de gestão que possibilita identificar o cenário interno e externo dos negócios. Em português, também é chamada de análise FOFA, correspondendo às iniciais no nosso idioma. (PAULILLO, [s.d.], *online*).

Ao realizar a análise *SWOT* após as entrevistas, foram identificadas algumas fraquezas, como: tempo para atender aos clientes, organização com datas, preços e dados dos clientes. Todas as fraquezas serão trabalhadas no projeto para se tornarem um ponto forte.

Após as análises, identificou-se como fraqueza mais recorrente o tempo de resposta aos clientes, considerando que muitos confeitadores operam de forma independente, incumbidos da gestão integral do negócio, incluindo produção, divulgação e venda. Diante deste acúmulo de tarefas, o atendimento aos clientes pode ser prejudicado, resultando em atrasos na comunicação e, conseqüentemente, desinteresse por parte dos consumidores em adquirir os produtos oferecidos.

Figura 2: *SWOT*



Fonte: os autores

2.3 - Plano de Ação 5W2H do Projeto

A matriz é uma ferramenta de gestão que oferece respostas claras e diretas para sete perguntas: *What* (O que será feito?), *Why* (Por que será feito?), *Who* (Quem fará?), *When* (Quando será feito?), *Where* (Onde será feito?), *How* (Como será feito?) e *How much* (Quanto custará?), ela fornece clareza, foco e direção na implementação de projetos. Ao responder às perguntas, é possível definir as atividades a serem realizadas, os responsáveis, estabelecer prazos e custos.

O plano de ação enfatiza a entrega de um sistema web online para vendas e gerenciamento de negócios de confeitaria. Essa iniciativa visa otimizar os processos e simplificar o trabalho dos confeitários. Para alcançar esse objetivo, foi utilizado um ambiente de desenvolvimento e linguagens como TypeScript, em conjunto com HTML e CSS.

Figura 3: 5W2H

What? (O que?)	Why (Por quê?)	Where? (Onde?)	When? (Quando?)	Who? (Quem?)	How? (Como?)	How much? (Quanto?)
Desenvolvimento de um sistema web para clientes fazerem encomendas e para o gerenciamento das confeitarias.	Para otimizar o processo de gerenciamento de encomendas e produtos, melhorar a eficiência operacional e a experiência do cliente.	Ambiente de Desenvolvimento e Testes	Junho de 2024	Caroline e Larissa	Através do levantamento de Requisitos, utilizando a linguagem TypeScript aliada com HTML e CSS	Orçamento será calculado

Fonte: os autores

3 - Levantamento de Requisitos

O processo de elicitação de requisitos é de suma importância para definir como um sistema deverá ser desenvolvido, pois irá detalhar como será o seu funcionamento, suas necessidades e regras, como pode ser exemplificado por Ian Sommerville:

Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços oferecem e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado engenharia de requisitos (RE, do inglês *requirements engineering*). (SUMMERVILLE, 2011, p.57)

3.1 - Elicitação e especificação dos Requisitos

Para o levantamento de requisitos, foram conduzidas entrevistas com confeitores autônomos que lidam regularmente com reservas. Essa abordagem metodológica permitiu identificar as principais demandas enfrentadas no cotidiano desses profissionais.

Um questionário contendo perguntas abertas foi desenvolvido para conduzir as entrevistas, permitindo que os participantes compartilhassem suas experiências sobre o processo de reservas. Esse método possibilitou o contato direto com os *stakeholders*, o que resultou na identificação dos requisitos essenciais para o desenvolvimento do projeto (SUMMERVILLE, 2011, p.72).

Durante a análise das entrevistas, foram identificadas diversas demandas, sendo a principal delas o atendimento aos clientes. Frequentemente, os clientes entram em contato para realizar reservas, o que consome tempo do profissional, pois é necessário fornecer informações sobre os produtos e, muitas vezes, esclarecer dúvidas que prolongam o processo de atendimento, podendo, eventualmente, resultar em pedidos não finalizados. Outra questão apontada é o gerenciamento de pedidos, onde o profissional precisa registrar as reservas, organizar datas de entregas, controlar o estoque e disponibilidade de insumos.

3.2 - BPMN

O BPMN (*Business Process Model and Notation*) é uma ferramenta de fluxograma que permite demonstrar todas as etapas de um processo, do início ao fim. Esse modelo é construído de forma visual, facilitando o entendimento de todas as partes envolvidas no projeto.

Este fluxograma é indispensável, pois servirá como base para o desenvolvimento de outras ferramentas dentro do projeto, além de orientar as decisões futuras devido à sua estrutura visual de fácil compreensão para os *stakeholders* e os profissionais envolvidos na elaboração do sistema (Lucidchart, [s.d], *online*).

No âmbito deste projeto, foram elaborados dois BPMNs, um sistêmico e outro não sistêmico. O objetivo desses modelos é demonstrar todas as etapas e tomadas de decisões presentes em cada um deles.

3.2.1 BPMN não sistêmico

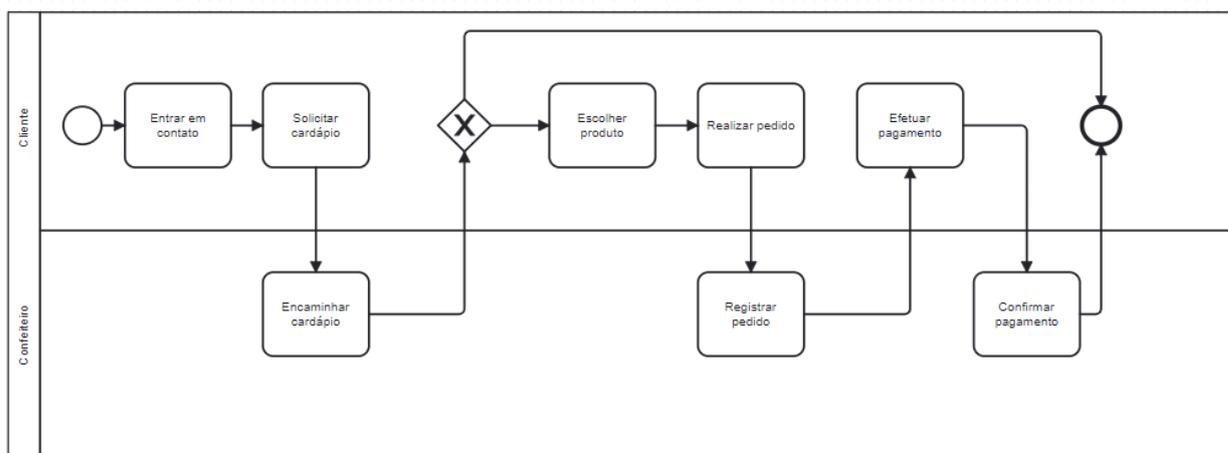
Atualmente o processo de reservas de confeitaria é realizado diretamente com o profissional, geralmente por chamada telefônica ou por mídias sociais, como o *WhatsApp*, *Facebook* ou *Instagram*.

Quando o cliente entra em contato solicitando informações sobre os produtos disponíveis, o profissional envia o catálogo para que ele possa fazer sua escolha. Nesse momento, o cliente pode optar por realizar um pedido ou não.

Se decidir avançar, ele seleciona os produtos desejados e faz o pedido, que é então registrado pelo confeiteiro.

Em seguida, são fornecidas as opções de pagamento, que devem ser concluídas pelo consumidor para que o pedido seja efetivado.

Figura 4: BPMN não sistêmico



Fonte: os autores

3.2.2 - BPMN sistêmico

O sistema *web* proposto a ser desenvolvido nesse projeto funcionará da seguinte forma: acesso ao painel do confeiteiro e acesso ao painel do cliente, conforme descrito abaixo:

- **Painel do confeiteiro:**

O profissional acessará o sistema e encontrará três opções: consultar pedidos, cadastrar produtos e acessar produtos. A primeira opção dará acesso a todos os pedidos, permitindo a visualização dos produtos e valores das compras. A segunda opção possibilitará ao profissional cadastrar novos produtos, exigindo a inserção das informações necessárias e o armazenamento no banco de dados. Na última opção,

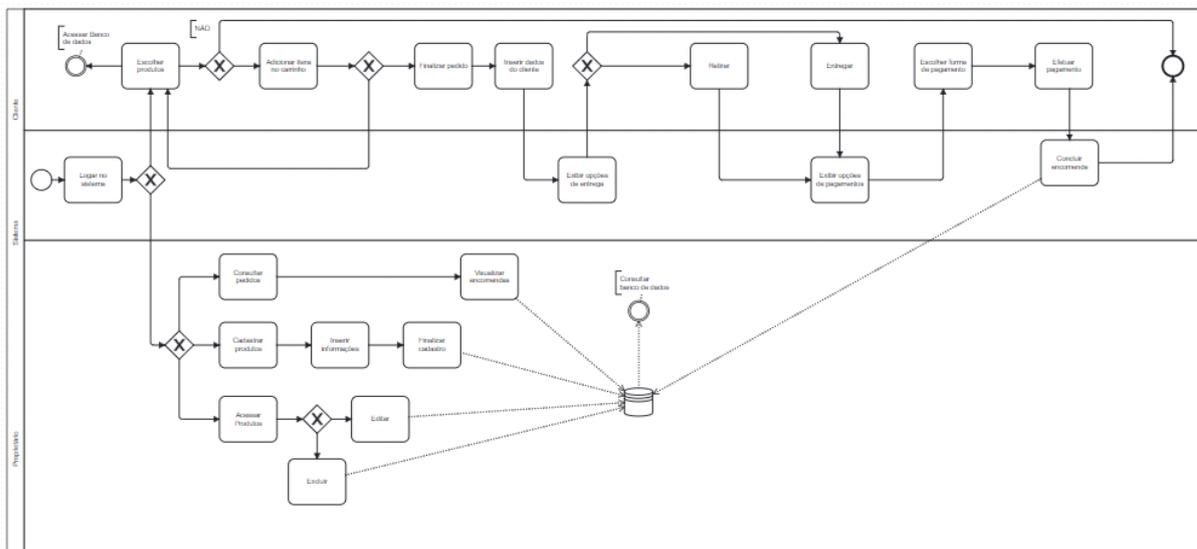
ele poderá acessar todos os produtos já cadastrados, onde será possível editar ou remover um produto do sistema conforme necessário.

- **Painel do cliente:**

O cliente terá acesso aos produtos disponíveis para encomenda e poderá optar por realizar ou não o pedido. Caso decida não prosseguir, ele encerra o processo. Se optar por fazer o pedido, selecionará os produtos desejados e os adicionarão ao carrinho. Neste ponto, ele pode continuar navegando para escolher mais produtos ou finalizar o processo de seleção.

Ao finalizar o pedido com os produtos desejados, o cliente precisará inserir seus dados pessoais para efetuar a compra. Em seguida, serão apresentadas as opções de envio, que incluem retirada ou entrega. Posteriormente, o cliente selecionará o método de pagamento desejado. O pedido só será concluído após o pagamento ser efetuado.

Figura 5: BPMN sistêmico



Fonte: os autores.

3.3 - Requisitos Funcionais

O princípio para elucidar como um sistema deverá ser construído é entender o problema a ser abordado. A partir dessa base, é possível compreender plenamente

as demandas e regras identificadas durante o contato com os *stakeholders*. Esse processo é fundamental para alcançar as metas, necessidades e objetivos da organização.

Para estruturar a base do projeto, os requisitos funcionais são de suma importância, pois demonstram como um sistema deverá se comportar em determinadas situações, descrevendo as ações que deverão ser tomadas e incluindo até mesmo o que não deverá ser feito. (SUMMERVILLE, 2011, p.61)

Quadro 1 – Requisitos Funcionais do sistema

RF001 - Cadastrar Confeiteiro	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá ter a opção do usuário se cadastrar no sistema, através da sua conta google já cadastrada no seu celular ou pelo e-mail. O perfil deverá ser preenchido pelo confeiteiro, contendo informações do seu negócio.		
RF002 - Cadastrar Produto	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá fornecer ao confeiteiro uma funcionalidade para que o mesmo possa realizar o cadastro de seus produtos.		
RF003 - Exibir produtos	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá permitir a visualização do catálogo de produtos pelo cliente final.		
RF004 - Escolher opções	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá permitir que o cliente escolha as opções personalizadas, que serão disponibilizadas para cada tipo de produto.		
RF005 - Adicionar produtos no carrinho	Categoria: (x) Evidente	Prioridade: (X) Altíssima

	() Oculto	() Alta () Média () Baixa
DESCRIÇÃO: O sistema deverá permitir que o cliente adicione ou remova os produtos em seu carrinho.		
RF006 - Finalizar pedido	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá disponibilizar os campos para preenchimento dos dados do cliente, bem como forma de entrega e meios de pagamentos.		
RF007 - Visualizar pedidos	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá exibir ao confeitiro um painel organizado com todos os pedidos recebidos.		
RF008 - Cadastrar cliente	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá permitir o cadastro do cliente.		
RF009 - Exibir meios de pagamentos	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá ter integração com meios de pagamentos para possibilitar o cliente final a realizar o pagamento junto ao pedido.		
RF010 - Exibir forma de entrega	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá exibir ao cliente final as possibilidades de entrega ou retirada de seus pedidos.		
RF011 - Logar no sistema	Categoria: (x) Evidente () Oculto	Prioridade: (X) Altíssima () Alta () Média () Baixa

Descrição: O sistema deverá permitir o cliente final ou o confeitiro a logar no sistema.		
RF012 - Cadastrar formas de pagamentos	Categoria: <input checked="" type="checkbox"/> Evidente <input type="checkbox"/> Oculto	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir ao confeitiro configurar as opções de pagamentos que serão exibidas para o cliente final.		
RF013 - Cadastrar formas de entrega	Categoria: <input checked="" type="checkbox"/> Evidente <input type="checkbox"/> Oculto	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: Deverá possibilitar ao confeitiro cadastrar as formas de entrega e retirada que serão disponibilizadas para o cliente.		
RF014 - Editar produto	Categoria: <input checked="" type="checkbox"/> Evidente <input type="checkbox"/> Oculto	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir a edição dos produtos já cadastrados.		
RF015 - Excluir produto	Categoria: <input checked="" type="checkbox"/> Evidente <input type="checkbox"/> Oculto	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir a exclusão dos produtos.		
RF016 - Consultar pedidos	Categoria: <input checked="" type="checkbox"/> Evidente <input type="checkbox"/> Oculto	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir ao cliente consultar seus pedidos.		

Fonte: os autores.

3.4 - Requisitos Não Funcionais

Os requisitos não funcionais podem ser caracterizados como regras externas que o sistema deve seguir para atender os requisitos funcionais. Podem ser considerados: regras de negócio, restrições de tempo, legislações, atributos de segurança, qualidade, entre outros. (PRESSMAN; MAXIN, 2021).

Quadro 2 – Requisitos Não Funcionais do sistema

RNF001 - Criptografar dados	O sistema deverá criptografar os dados dos clientes e dos confeitadores		() Desejável (X) Obrigatório	(X) Permanente () Transitório
------------------------------------	---	--	----------------------------------	-----------------------------------

Fonte: os autores.

3.5 - Regras de Negócio

As regras de negócios são importantes para definir diretrizes ou restrições da organização. Devem estar associadas a política da empresa, levando em consideração seus valores internos, normas e alinhamento estratégico. Com as regras definidas, o controle do processo e do fluxo de tarefas se torna mais fácil e prático, contribuindo para tomada de decisão. (RIBEIRO, 2020, *online*).

Quadro 3 – Regras de Negócio do sistema.

RN001 - Cadastrar usuário
Descrição: O usuário deverá realizar o cadastro para utilizar o sistema.
RN002 - Armazenar informações do negócio
Descrição: O sistema deverá ter cadastrado as informações básicas do negócio.
RN003 - Cadastrar nome e preço
Descrição: É obrigatório o cadastro de nome e preço do produto.
RN004 - Exibir catálogo
Descrição: O sistema deverá exibir um catálogo visualmente atraente de todos os produtos de confeitaria disponíveis.
RN005 - Produtos com estoque
Descrição: Se o produto não tiver estoque, não deverá ser possível adicionar no carrinho.
RN006 - Inserir campos obrigatórios
Descrição: O cliente deverá preencher todos os campos para finalizar o pedido.

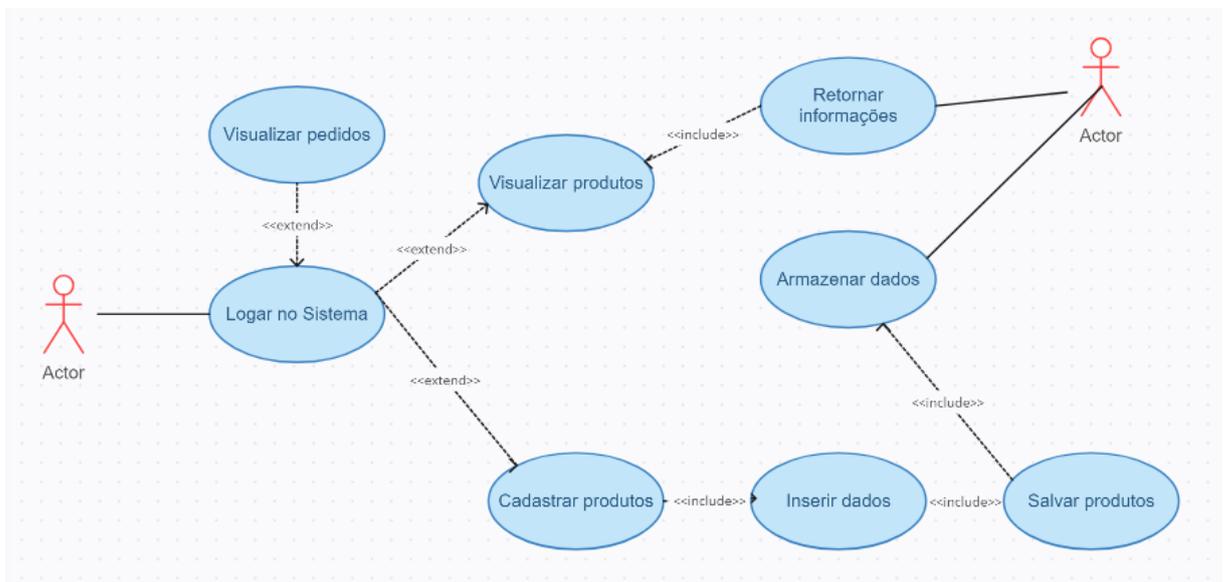
Fonte: os autores.

3.6 - Casos de Uso

O caso de uso tem como objetivo definir como um sistema irá se comportar através das ações de um usuário para atingir suas metas. Nesse documento será demonstrado todas as interações e processos que o usuário pode fazer, constando também os fluxos alternativos e como o sistema lida com essas mudanças (PRESS-MAN; MAXIN, 2021).

Com esse documento, é possível coletar informações importantes das funcionalidades disponíveis no sistema para os usuários, pois produz um conhecimento detalhado do sistema, como a relação entre o usuário e o sistema e o comportamento esperado pelo sistema. (IBM, 2021, *online*).

Figura 6: Diagrama de Caso de Uso



Fonte: os autores.

A documentação de Caso de Uso é essencial para demonstrar o cenário no qual vai ser usado, juntamente com as regras em que o sistema deverá seguir com outras opções por meio da escolha do usuário.

Serão descritos os eventos necessários para efetuar as ações no cenário principal e alternativo, considerando pré-condições ou aquelas que forem posteriores a sua execução.

Novamente são destacados os atores, parâmetros a que devem ser fornecidos, restrições e validações.

Quadro 4: Documentações de Casos de Uso

Caso de Uso – Cadastrar Confeiteiro	
ID	UC 001
Descrição	Este caso de uso tem por objetivo cadastrar uma conta para o confeiteiro.
Ator Primário	Confeiteiro
Pré-condição	Cadastrar no sistema
Cenário Principal	1.O use case inicia quando o usuário seleciona a opção cadastrar conta. 2.O sistema carrega o formulário de cadastro. 3.O usuário preenche com seus dados básicos obrigatórios (nome, contato, e-mail, endereço e senha). 4.Validação dos dados obrigatórios. 5.O usuário deverá salvar os dados de cadastro. 6.O sistema armazena as informações no banco de dados.
Pós-condição	Usuário cadastrado
Cenário Alternativo	1.a - O usuário tentar logar sem cadastro. 1.b - O sistema irá redirecionar para a tela de cadastro. 3.a - O usuário não informa os dados obrigatórios.
Caso de Uso – Cadastrar Produto	
ID	UC 002
Descrição	Este caso de uso tem por objetivo cadastrar produtos
Ator Primário	Confeiteiro
Pré-condição	Logar no sistema
Cenário Principal	1. O use case inicia quando o usuário seleciona a opção cadastro de produtos. 2. O sistema carrega o formulário de cadastro de produtos. 3. O usuário deverá informar os dados obrigatórios do produto (nome, preço, quantidade, opções de sabor). 4. O sistema deverá validar se os dados obrigatórios foram preenchidos. 5. O usuário deverá salvar o produto. 6. Após validação, o sistema deverá armazenar as informações no banco de dados.
Pós-condição	Cadastro de produtos
Cenário Alternativo	3.a - O usuário deverá fornecer os dados obrigatórios.
Caso de Uso – Visualizar Produtos	
ID	UC 003
Descrição	Este caso de uso tem por objetivo visualizar os produtos cadastrados
Ator Primário	Confeiteiro
Pré-condição	Logar no sistema e produtos cadastrados
Cenário Principal	1. O use case inicia quando o usuário seleciona a opção visualizar de produtos. 2.O sistema exibe os produtos cadastrados e as opções de visualizar, editar ou excluir produto.
Pós-condição	Acessar produtos.
Cenário Alternativo	2.a - Se não houver produto cadastrado, o sistema exibirá um alerta de que não há produtos cadastrados.
Caso de Uso – Visualizar Pedidos	
ID	UC 003
Descrição	Este caso de uso tem por objetivo visualizar pedidos.
Ator Primário	Confeiteiro
Pré-condição	Logar no sistema

Cenário Principal	1. O use case inicia quando o usuário seleciona a opção visualizar os pedidos. 2. O sistema exibe uma lista com todos os pedidos. 3. O usuário seleciona o pedido desejado. 4. O sistema carrega e exibe todos os detalhes do pedido (dados do cliente e do pedido).
Pós-condição	Visualizar pedidos.
Cenário Alternativo	3.a - O usuário escolhe não exibir os detalhes do pedido.

Fonte: os autores.

3.7 - Diagrama de Classe

O diagrama de classe tem como objetivo descrever uma modelagem de programação orientada a objetos. Os componentes que o compõem, representam as classes que serão desenvolvidas, os objetos ou as interações entre si. (LUCID-CHART, [sd], *online*).

Esse modelo é de suma importância, pois identifica os objetos e os agrupam em suas determinadas classes, facilitando na construção da estrutura do banco de dados, uma vez que cada classe representará uma tabela. (DEV MEDIA, [sd], *online*).

O diagrama do projeto possui quatro entidades, sendo elas:

- Confeiteiro:

Atributos: ID, nome_conf, nome_empresa, CNPJ, endereco, email e telefone.

- Produto:

Atributos: ID, nome_product, tipo_valor, sabor_massa, sabor_recheio, preco e quantidade.

- Pedidos:

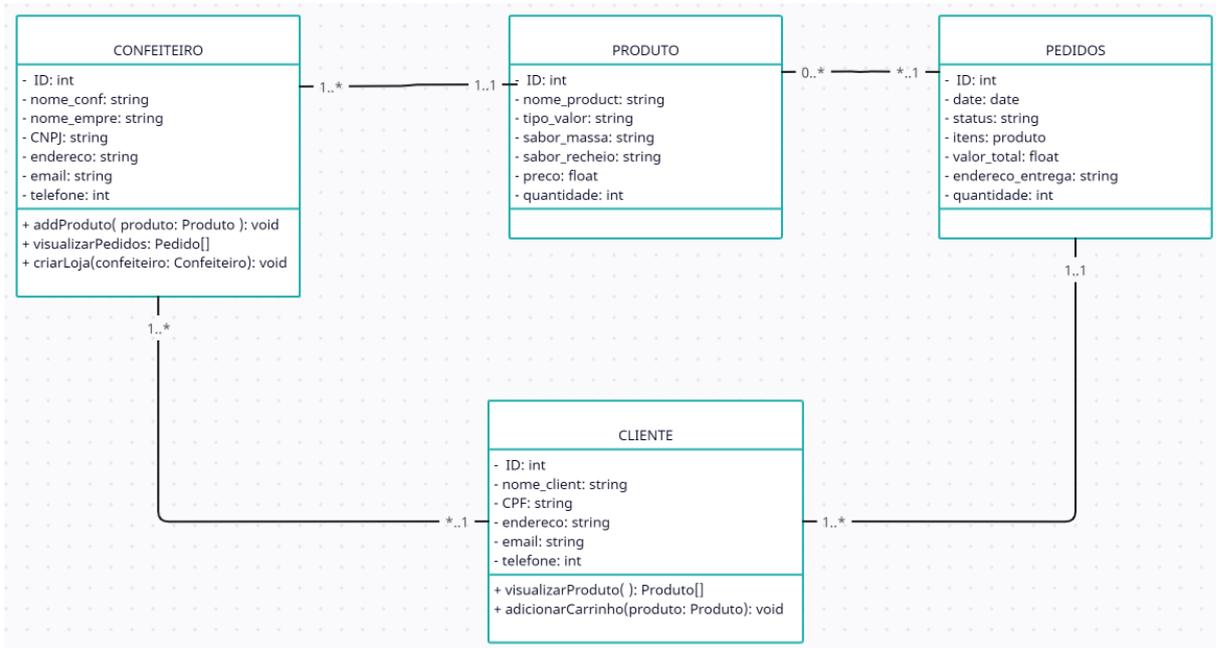
Atributos: ID, date, status, itens, valor_total, endereco, quantidade.

- Cliente:

Atributos: ID, nome_client, CPF, endereco, email e telefone.

Abaixo foram descritos os métodos e o relacionamento entre cada entidade.

Figura 7: Diagrama de Classe



Fonte: os autores.

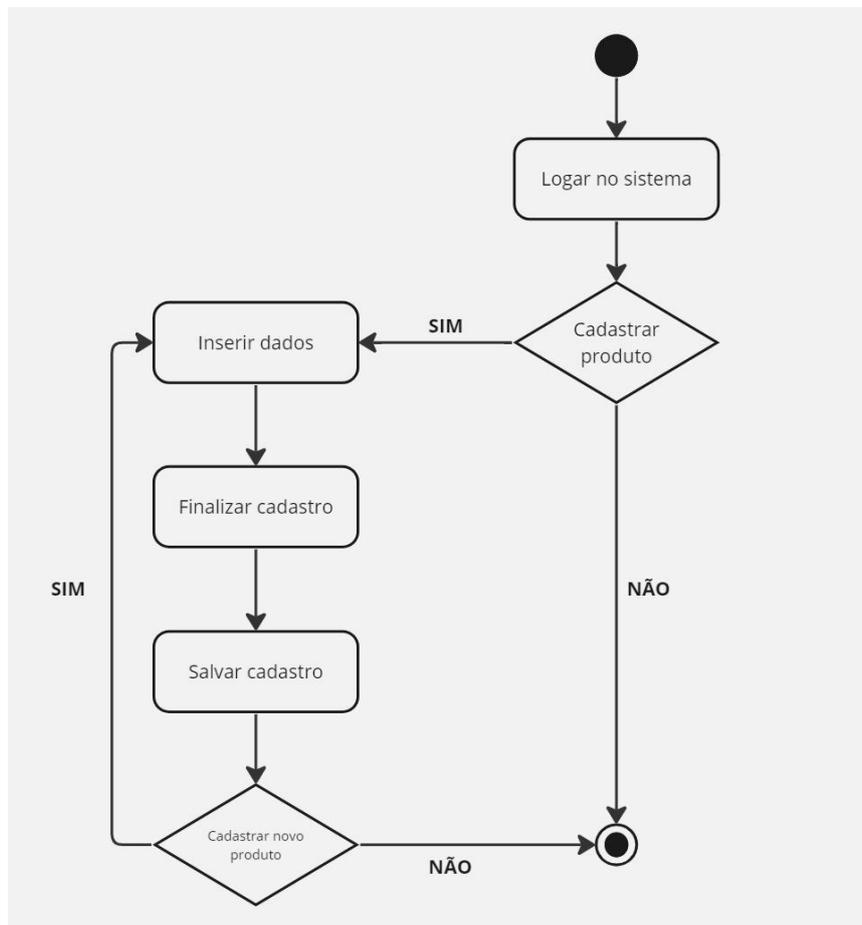
3.8 - Diagrama de Atividade

O diagrama de atividade tem como objetivo demonstrar fluxos de interações com o sistema em cenários específicos, onde há informações que poderão ser acessadas ou manipuladas pelo usuário. Geralmente é construído em fluxos que possuem tomadas de decisão, pois esses eventos se caracterizam como transições de objeto para objeto. (PRESSMAN; MAXIN, 2021).

Esse modelo permite a visualização de como o sistema poderá se comportar, exibindo a sequência dos fluxos de ações necessárias para atingir algum objetivo durante esse processo. (IBM, 2021, *online*).

O fluxo selecionado para o Diagrama de Atividade refere-se à tomada de decisão relacionada ao cadastro de produtos. Após o usuário fazer *login* no sistema, ele tem a escolha de cadastrar ou não um produto. Se optar por prosseguir com o cadastro, ele inserirá os dados, finalizará e salvará as informações. Em seguida, terá a opção de cadastrar novos produtos ou finalizar o processo, conforme ilustrado na imagem abaixo.

Figura 8: Diagrama de Atividade



Fonte: os autores.

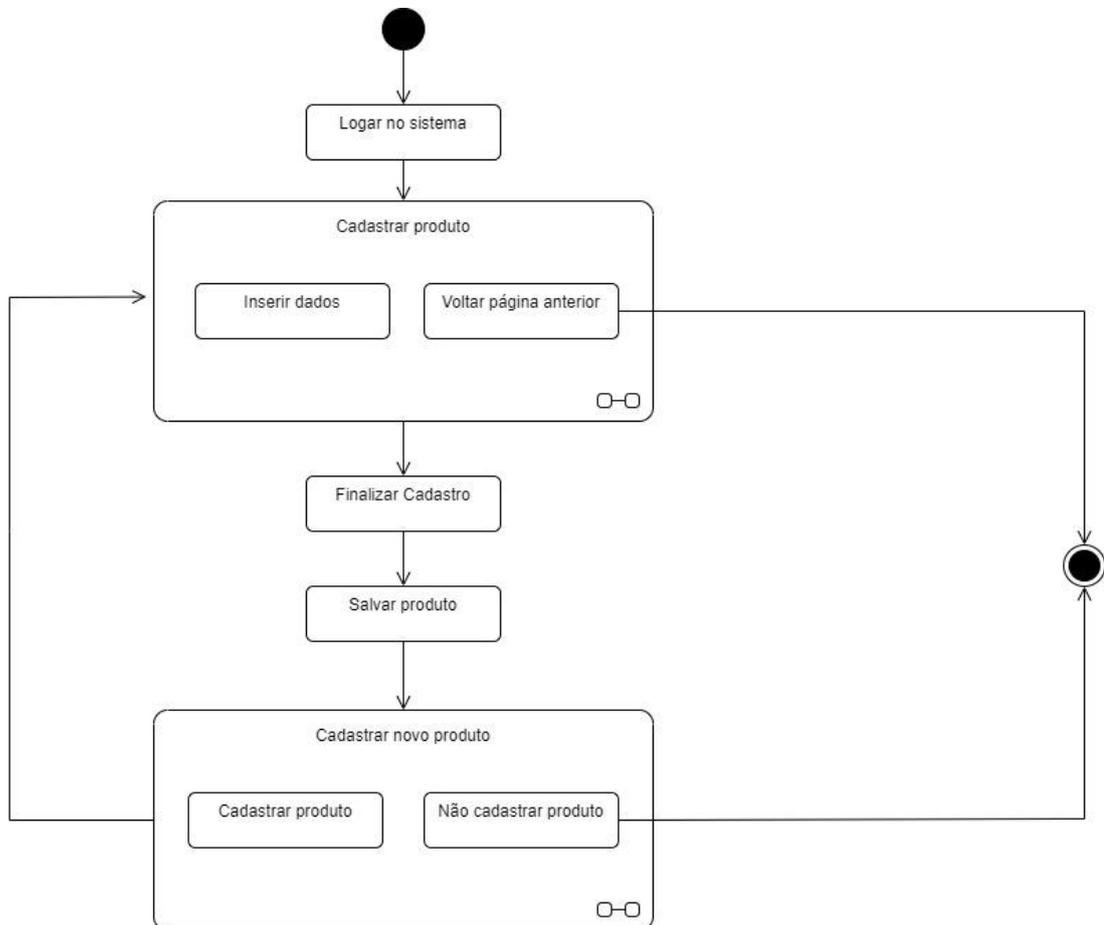
3.9 - Diagrama de Estado

O Diagrama de Estado é uma representação gráfica que demonstra os diferentes tipos de estados que um objeto pode conter em sua vida útil, permitindo a visualização da transição entre os estados. Essa ferramenta facilita a compreensão do comportamento dinâmico do sistema, permitindo a identificação e análise de forma mais precisa de como as mudanças de estado afetam o funcionamento do sistema como um todo (LUCIDCHART, [sd], *online*).

Para compreender este diagrama, escolhemos o fluxo de cadastro de produtos. Inicialmente, quando o usuário acessa o sistema, é redirecionado para a lista de produtos, onde tem a opção de cadastrar um novo item. Se optar por não cadastrar, a ação é concluída. Porém, se escolher cadastrar, será solicitada a inserção dos dados do produto, seguida pelo encerramento do cadastro e sua confirmação. Após essa

etapa, o usuário pode decidir entre adicionar outro produto, repetindo o fluxo anterior, ou concluir.

Figura 9: Diagrama de Estado



Fonte: os autores.

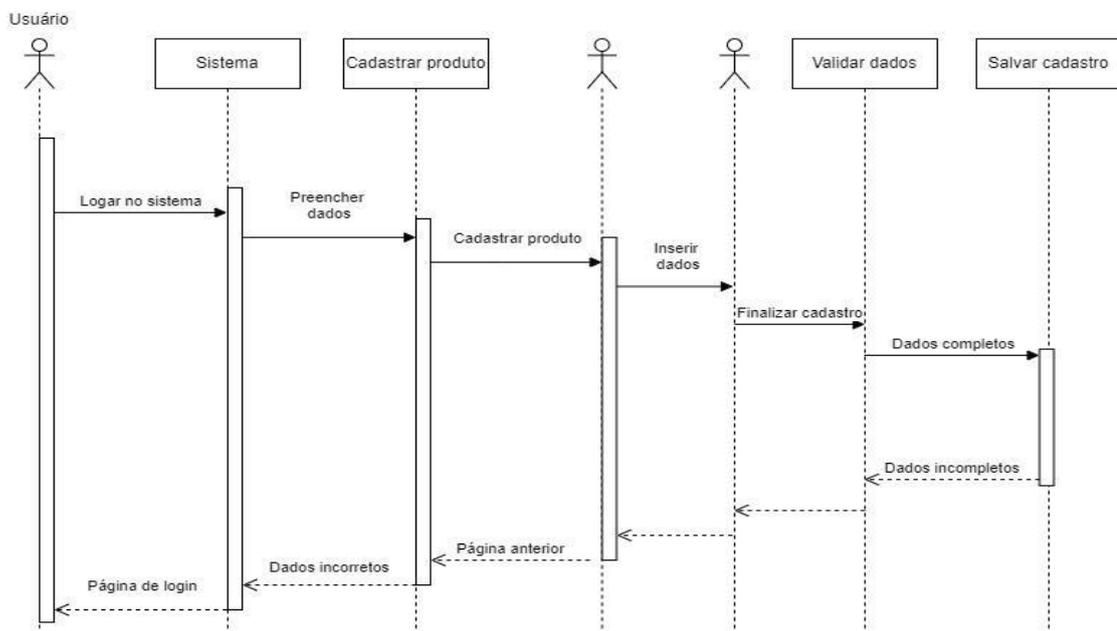
3.10 - Diagrama de Sequência

O diagrama de sequência demonstra as mensagens que são trocadas entre os objetos e as interações realizadas. São construídos a partir de linhas de vida, que retratam os atores envolvidos no processo, que podem ser: o usuário, o banco de dados, o sistema e outros, nos quais participam de interações que resultam em mensagens que são trocadas entre eles para execução de alguma funcionalidade (IBM, 2021, *online*).

No diagrama de sequência a seguir, é demonstrado o processo de cadastro de produtos pelo usuário. Na parte superior, estão destacadas todas as ações que o usuário pode realizar no sistema, enquanto na parte inferior são apresentadas as possíveis mensagens que o sistema envia ao cliente.

Inicialmente, o usuário acessa o sistema e realiza o *login*. Se os dados inseridos estiverem incorretos, uma notificação é exibida e o usuário será redirecionado à página de anterior para tentar novamente. Uma vez que o *login* seja efetuado com êxito, o usuário pode prosseguir para cadastrar um novo produto, inserindo todas as informações necessárias e garantindo sua validade. Caso haja alguma informação obrigatória não preenchida, o sistema solicitará ao usuário que a insira antes de prosseguir. Após a inserção correta de todos os dados do produto, o usuário pode salvar as informações e retornar à página inicial do sistema.

Figura 10: Diagrama de Sequência



Fonte: os autores.

3.11 - Matriz de Rastreabilidade

A matriz de rastreabilidade desempenha um papel crucial, pois cataloga os requisitos funcionais e os vincula às regras de negócio. Essa prática oferece uma visão abrangente do funcionamento do negócio e simplifica a introdução de mudanças, possibilitando uma análise minuciosa das implicações de cada ação, o que ajuda a prevenir possíveis erros, conforme explicitado por Espinha:

Matriz de rastreabilidade de requisitos é uma ferramenta que explicita a relação direta dos requisitos entre si ou com os outros componentes do projeto. Assim, caso alguma alteração seja feita no projeto, sabe-se quais requisitos serão afetados com tal mudança. Com uma matriz de rastreabilidade, é possível rastrear a origem dos requisitos, os elos que os ligam com outros elementos do projeto, encontrar inconsistências nos requisitos e garantir que os diferentes níveis do projeto estejam alinhados com os mesmos. (ESPINHA, 2020, *online*).

Quadro 5: Matriz de Rastreabilidade

	RN001	RN002	RN003	RN004	RN005	RN006
RF001	X					
RF002			X			
RF003				X		
RF004					X	
RF006						X
RF008	X					
RF011	X					

Fonte: os autores

4 - Ferramentas e Métodos

Para consolidar o projeto, foi essencial empregar ferramentas que facilitassem sua gestão e organização, utilizando-as para várias áreas como prototipação, estruturação de *software* e desenvolvimento, sendo elas:

4.1.1 - BPMN.io

Versão: 2.0

Licença: *Web*, livre.

Sobre: é um sistema *web* que permite criar diagramas BPMN e possui interface de fácil acesso. *Software* gratuito.

4.1.2 - Figma

Versão: 127.2

Licença: gratuita, porém possui planos pagos.

Sobre: é uma aplicação gráfica que permite a criação de interfaces gráficas interativas para sites e aplicativos. A sua versão gratuita permitiu a revisão do projeto e compartilhamento de dados a qualquer momento.

4.1.3 - Visual Studio Code

Versão: 1.88

Licença: Livre

Sobre: é um *software* que permite a utilização de várias linguagens de programação, além de possuir várias extensões que possibilitam sua execução em qualquer sistema operacional.

4.1.4 - Lucidchart

LucidChart Free: versão gratuita

Sobre: aplicativo *web* de criação de diagramas e gráficos que permite o compartilhamento, edição e revisão.

4.1.5 - Draw.io

Versão: 24.2.5

Licença: Livre

Sobre: *software* de desenho gráfico multiplataforma, pode ser utilizado para criar diagramas e fluxogramas.

4.1.6 - Node.js

Versão: 18.17

Sobre: *software* de código aberto e multiplataforma, possui uma arquitetura assíncrona e permite a execução do *JavaScript* fora de um navegador *web*.

4.1.7 - TypeScript

Versão: 5.0.4

Sobre: linguagem de programação de código aberto desenvolvido pela Microsoft, que permite a sua tipagem.

4.1.8 - Microsoft Office

Versão: 360

Licença: acesso por *e-mail* institucional

Sobre: pacote de aplicativos desenvolvido pela Microsoft, contendo programas de planilhas, processadores de texto, apresentações gráficas e outros.

4.1.9 - Insomnia

Versão: 2023.5.8

Licença: Livre

Sobre: aplicativo *desktop* de código aberto que permite a interação, depuração e testes de APIs.

4.1.10 - Prisma

Versão: 4.13

Sobre: é um ORM (*Object Relational Mapping*) que tem como objetivo facilitar a interação entre o código e o banco de dados relacional, dispensando a necessidade de realizar consultas manuais em SQL.

4.1.11 - Fastify

Versão: 4.17

Licença: Livre

Sobre: *framework back-end* para *JavaScript* baseado em *plugins*.

4.1.12 - Creately

Licença: Livre

Sobre: Creately é uma plataforma de colaboração visual que torna mais fácil para sua equipe colaborar e trabalhar em conjunto com mais eficiência.

5 - Desenvolvimento

O objetivo central deste projeto é criar um sistema *web* de encomendas para confeitários. Para esse trabalho, foi desenvolvido uma parte do *software* que consiste no painel de controle de produtos, onde permitirá o cadastro, edição e remoção de produtos. Também foi criado o painel de visualização dos produtos cadastrados. Posteriormente, o *software* será construído por completo para comercialização.

Para realizar esse desenvolvimento, todo o processo de estruturação da Engenharia de *Software* foi de suma importância, pois a partir dela foi elencado como o sistema deverá ser construído, definindo seus requisitos, regras de negócios entre outros.

O nome escolhido para o sistema foi o *Sweet Order*, cujo objetivo é transmitir a ideia de realizar pedidos de doces de confeitaria em geral, oferecendo uma pronúncia fácil e trazendo referências ao aspecto lúdico da experiência.

No desenvolvimento desta aplicação *web* foram utilizadas as ferramentas Node.js, *TypeScript*, HTML e CSS. O *TypeScript* é uma extensão do *JavaScript* que permite adicionar tipagem ao código. Foi necessário instalar o Node.js para a criação do projeto e compilação do *TypeScript* em *JavaScript*, uma vez que o Node não o reconhece diretamente, criando o ambiente de *back-end* do projeto.

Para construção do *front-end* foram utilizados o HTML que é uma linguagem de marcação e definição de estrutura essencial para páginas *web*, onde foram definidos todos os elementos necessários para a interface. Com o objetivo de garantir a personalização das páginas de acordo com a prototipagem inicial, foram aplicados os recursos do CSS, responsável pela estilização dos elementos HTML.

Como base de dados, foi escolhida a ferramenta Prisma ORM (*Object-Relational Mapping*) integrada ao SQLite (Structured Query Language Injection). Essa escolha possibilita a utilização de uma API intuitiva para interação com o banco de dados através da linguagem de programação, eliminando a necessidade de lidar diretamente com SQL.

5.1 - Back-end

A estruturação do Prisma foi realizada no arquivo *schema.prisma*, onde foi definido o SQLite como banco de dados criando a tabela de produto, no qual contém os atributos necessários e seus tipos de acordo com o Diagrama de Classe apresentado anteriormente no capítulo 3.7. Abaixo está demonstrado o trecho de código utilizado:

Figura 11: Código de definição de banco de dados

```
schema.prisma x
backend > prisma > schema.prisma
1 // This is your Prisma schema file,
2 // learn more about it in the docs: https://pris.ly/d/prisma-schema
3
4 generator client {
5   provider = "prisma-client-js"
6 }
7
8 datasource db {
9   provider = "sqlite"
10  url      = env("DATABASE_URL")
11 }
12
13 model Product {
14   id          String   @id @default(uuid())
15   description String
16   price       Decimal
17   quantity    Int
18   type_price  String
19   massa       String
20   sabor       String
21   created_at  DateTime
22   @@map("product")
23 }
24
25 model Login {
26   id          String   @id @default(uuid())
27   user        String
28   password    String
29 }
```

Fonte: os autores.

Para realizar as requisições API foi utilizado o *framework Fastify*, no qual iniciou-se como servidor para ser utilizado nas rotas. Essas foram definidas no arquivo *routes.ts*, onde consta todas as requisições de API que foram utilizadas nesse projeto.

Na criação das rotas, utilizamos a biblioteca ZOD que define o esquema de dados de forma declarativa, onde irá implementar ao objeto o tipo de dado definido e realizar as validações necessárias. A imagem abaixo é um exemplo de rota GET, onde é possível verificar a importação do ZOD e do *Fastify*, há dois tipos de consultas de produtos, sendo elas pelo seu id e uma lista que contém todos os cadastros:

Figura 12: Definição das rotas de API

```
routes.ts x
backend > src > routes.ts > AppRoutes
1 // importa a dependência ZOD
2 import {z} from 'zod'
3 import {prisma} from './lib/prisma'
4 import { FastifyInstance } from 'fastify'
5 export async function AppRoutes(server: FastifyInstance){
6
7 // rota para consultar todos os produtos no banco de dados
8 server.get('/products', async () => {
9   const products = await prisma.product.findMany()
10  return products
11 })
12
13 // rota para consultar os produtos que iniciam a descrição com uma palavra
14 server.get('/product/:id', async (request) => {
15
16   const idParam = z.object({
17     id: z.string()
18   })
19
20   const {id} = idParam.parse(request.params)
21   // faz a consulta no banco de dados
22   const product = prisma.product.findFirst({
23     where: {
24       id
25     }
26   })
27   return product
28
29 })
```

Fonte: os autores.

As funções responsáveis por realizar todo o processo de consulta a API, foram centralizadas em um único arquivo, e possuem como objetivo: coletar dados, chamar as rotas e realizar as alterações solicitadas pelo usuário. Entre elas estão as funções: cadastrar, editar, remover e listar produtos. Na imagem abaixo, exemplo de função de remoção de produtos com base em seu ID e posteriormente a chamada da função de exibir a lista:

Figura 13: Código de função de remoção de produto

```
64 async function remover(id){
65   const confirma = confirm('Deseja remover esse produto?')
66   if(!confirma){
67     return
68   } else {
69     await fetch(`http://localhost:3333/product/id/${id}`, {
70       method: 'DELETE'
71     })
72     .then(resposta=>{
73       alert('Remoção com sucesso')
74     })
75     .catch(erro=>{
76       alert('Problema na remoção')
77     })
78   }
79   //mostra a nova lista de produtos atualizado
80   consultaProdutos()
81 }
```

Fonte: os autores.

5.2 – Front-end

A estruturação das páginas *web* foram feitas por arquivos HTML e estilizadas com CSS, permitindo uma interação com o usuário fácil e intuitiva. Essa questão pode ser exemplificada na página de lista de produtos cadastrados, onde são exibidos juntamente com seus dados e possibilita ao usuário adicionar um novo produto, excluir ou editar algum já cadastrado.

Figura 14: Página de lista de produtos

Produto	Preço R\$	Quantidade	Tipo Preço	Sabor Massa	Sabor Recheio		
Bolo	35.9	10	kg	baunilha	brigadeiro		
Docinhos	50	100	cento	chocolate	brigadeiro		
Cupcake	6.5	35	unidade	morango	brigadeiro		

Fonte: os autores.

Na tela de lista de produtos, foi utilizado o HTML para estruturar os elementos da página e o CSS para personalizá-los, conforme demonstrado nas duas imagens seguintes:

Figura 15: Código HTML da lista de produtos

```
<div class="title">
  <div class="container">
    <table class="table mt-4">
      <thead>
        <tr>
          <th scope="col">Produto</th>
          <th scope="col"> Preço R$</th>
          <th scope="col"> Quantidade </th>
          <th scope="col"> Tipo Preço</th>
          <th scope="col"> Sabor Massa</th>
          <th scope="col"> Sabor Recheio</th>
        </tr>
      </thead>
      <tbody id="linhastabela"></tbody>
    </table>
  </div>
</div>
```

Fonte: os autores.

Figura 16: Código de estilização em CSS

```
22  .menu-left {
23      width: 200px;
24      height: 100vh;
25      flex-shrink: 0;
26      border-radius: 0px 20px 20px 0px;
27      background: #D99F7E;
28  }
29
30  .menu-option,
31  li {
32      top: 10px;
33      list-style: none;
34      gap: 10px;
35      display: flex;
36      flex-direction: column;
37      padding: 8px;
38      color: rgba(89, 25, 2, 1);
39      font-family: 'Roboto', serif;
40      font-size: 20px;
41      cursor: pointer;
42  }
43
44  .menu-option li:hover {
45      background-color: rgba(89, 25, 2, 0.493);
46      border-radius: 8px;
47      color: white;
48  }
49
```

Fonte: os autores.

Quando o usuário clicar para cadastrar um novo produto, será aberta uma nova página com um formulário para ser preenchido com os dados, após isso será necessário salvar as informações, que será exibido uma mensagem de retorno para validar se esse processo foi feito corretamente.

Figura 17: Tela de cadastro de produto

Cadastrar Produto

Nome do Produto:

Quantidade:

Valor

Tipo de Valor:

Sabor Massa

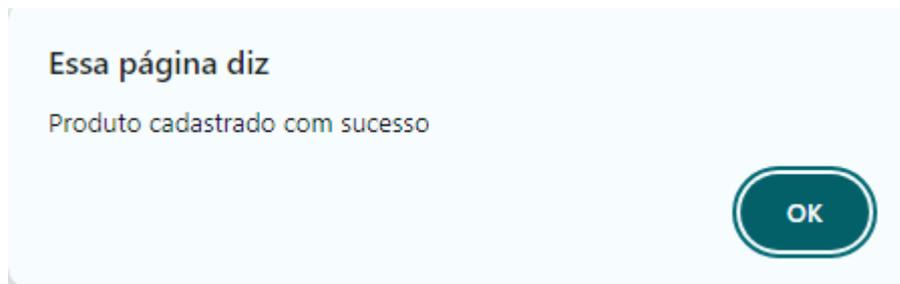
Sabor Recheio

Confirmar

Fonte: os autores.

Caso o usuário opte por gravar o produto, será exibido um alerta para confirmar o cadastro.

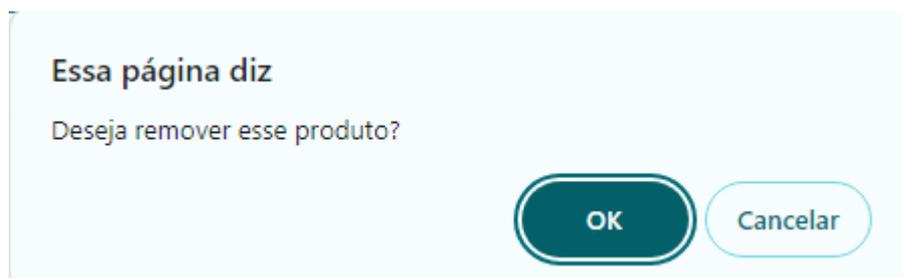
Figura 18: Alert de sucesso de cadastro de produto



Fonte: os autores.

Caso o usuário opte por remover o produto, será exibido um alerta para confirmar ou não a ação, se confirmada, o produto é excluído.

Figura 19: Alert de confirmação de exclusão de produto



Fonte: os autores.

Para modificar um produto, será necessário selecionar a opção de editar que fica ao final da tabela na lista, onde será encaminhado para outra página que exibirá os dados atuais e possibilidade de alterá-los. Após será necessário confirmar os novos dados e será exibido se as informações foram salvas corretamente ou não.

Figura 20: Tela de atualização de produto

Atualizar Produto

Nome do Produto:

Quantidade:

Valor:

Tipo de Valor:

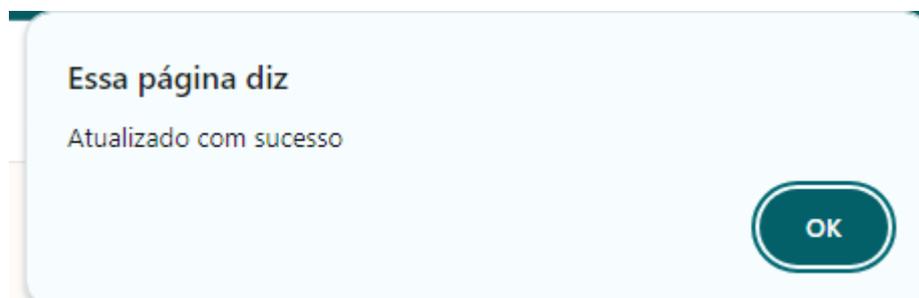
Sabor Massa:

Sabor Recheio:

Fonte: os autores.

Caso o usuário opte por gravar o produto, será exibido um alerta para confirmar a atualização do cadastro.

Figura 21: Alert de sucesso de atualização de produto



Fonte: os autores.

6 - Resultados e Discussão

A princípio foi realizada uma pesquisa sobre o segmento de confeitaria, pois o conhecimento da área é fundamental para estruturar o projeto e pontuar as principais necessidades e melhorias que podem ser feitas nos processos já usados.

Para coleta de dados das experiências dos *stakeholders*, foi utilizado a técnica da entrevista via *WhatsApp*, que permitiu conhecer os processos diários e como a tecnologia poderia ajudar na melhoria do controle de encomendas. Através das entrevistas foi possível coletar os pontos necessários para início do projeto. Logo após iniciou-se a criação da matriz *SWOT* e o *5W2H*, possibilitando determinar as principais dificuldades para elencar prioridades.

A principal função da análise *SWOT* é avaliar resultados do ambiente externo e interno da empresa, desenvolvendo assim, informações necessárias para identificação dos pontos fracos e fortes.

Através dela, foi desenvolvido o Termo de Abertura do Projeto (TAP), onde destacou-se a situação atual, as justificativas do projeto, os objetivos, as metas e a descrição. Propondo uma resolução sistêmica para os problemas e descrevendo as etapas necessárias para que esse projeto seja construído.

Com a documentação dos requisitos funcionais e não funcionais através da elicitação, resultou-se no máximo de informações sobre o objeto em questão, pois é a primeira atividade no processo de engenharia de requisitos, na qual se busca entender quais são as necessidades do usuário que devem ser atendidas pelo *software* que será desenvolvido.

Com a criação do BPMN foi possível desenhar o Diagrama de Caso de Uso, que demonstra os objetivos e interações entre os usuários e o sistema, assim como os processos exigidos pelo *software* para executar esses objetivos.

Em seguida, foram desenvolvidos o Diagrama de Atividade representando o fluxo de relacionamento entre os processamentos, o Diagrama de Máquina de Estado que representa o estado em que o objeto se encontra durante o processo de execução do sistema e o Diagrama de Sequência que representa o fluxo dos processos, principalmente a comunicação entre eles.

A próxima etapa foi a documentação da Matriz de rastreabilidade, pois ela interliga os elementos dos requisitos funcionais, casos de uso e regras de negócio, evidenciando em caso de modificação do projeto, o que poderia ser afetado e o que deve ser alterado em conjunto.

Partindo para uma das últimas etapas de criação dos artefatos de engenharia de *software*, foi criado o diagrama de classe e o banco de dados a ser estruturado.

Em seguida, foram discutidos como o projeto deveria ser realizado, fazendo esboços e rascunhos. Para tomar decisões de forma eficiente, antes do desenvolvimento do *software*, foram realizadas prototipagens de um sistema *web* que permitiu criar um protótipo funcional, evitando erros futuros ou inconformidade com o projeto tratado. Após essa etapa, o tempo e a divisão de trabalhos foram organizados, aprofundando os conhecimentos nas linguagens de programação escolhidas.

Assim, deu-se início à fase de desenvolvimento dos códigos das funcionalidades criadas, bem como os testes a fim de assegurar que o funcionamento esperado estava ocorrendo e caso contrário, as regras do negócio e requisitos criados eram revisados para alinhar e corrigir os problemas.

Considerações finais

O objetivo central deste projeto foi desenvolver um sistema *web* para gestão de encomendas para confeitários. Com a sua concretização, espera-se que esse processo seja mais prático, rápido e eficaz, ao centralizar todas as ferramentas e informações relevantes em um único *software*.

Inicialmente o profissional precisará cadastrar seus produtos e dados, para que esses sejam disponibilizados em seu catálogo. Em seguida, o cliente final terá acesso a esses itens e poderá escolher entre suas opções, avançando para finalização do pedido, onde será possível escolher a forma de pagamento e entrega desejada. Após a confirmação da compra, o pedido estará disponível ao vendedor que poderá consultá-lo e organizar a sua produção.

Este projeto foi concebido com base nas principais demandas identificadas durante as entrevistas, com o intuito de simplificar o dia a dia dos profissionais por meio de um sistema de encomendas. Seu foco principal foi eliminar a necessidade de contato direto entre cliente e profissional, economizando tempo que pode ser direcionado para outras atividades.

Para melhorias futuras, considera-se a realização do desenvolvimento completo do sistema para sua comercialização, incluindo a disponibilização de uma versão *mobile*. Isso proporcionará acesso facilitado em qualquer lugar e a qualquer momento.

Durante o processo de desenvolvimento deste projeto, verificou-se que o sistema poderá desempenhar um papel significativo na gestão de encomendas, visto que

é uma aplicação funcional que atende às demandas dos *stakeholders*. Portanto, para concluir com sucesso este projeto, foi de suma importância seguir a metodologia aplicada, o que incluiu um estudo aprofundado das ferramentas e métodos necessários para garantir a qualidade do *software*.

Em suma, este projeto representa não apenas uma solução atual para atender às demandas dos *stakeholders*, mas também estabelece uma sólida base para futuros empreendimentos comerciais. Com a possibilidade de implementar melhorias contínuas e adicionar novas funcionalidades, o sistema não só atenderá às necessidades atuais do negócio, mas também poderá se adaptar às mudanças e crescimento futuro, proporcionando uma plataforma robusta para o sucesso a longo prazo.

Referências

COUTINHO, T. **Termo de abertura do projeto: saiba o que é e como fazer.** Disponível em: <<https://www.voitto.com.br/blog/artigo/o-que-e-termo-de-abertura-do-projeto>>. Acesso em: 14 abr. 2024.

DEVMEDIA. **Orientações básicas na elaboração de um diagrama de classes.** Disponível em: <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 30 mar. 202

ESPINHA, R.G. **Matriz de Rastreabilidade de Requisitos: saiba como gerenciar as mudanças no escopo.** Disponível em: <<https://artia.com/blog/matriz-de-rastreabilidade/#:~:text=Matriz%20de%20rastreabilidade%20de%20requisitos%20%C3%A9%20uma%20ferramenta%20que%20explicita,ser%C3%A3o%20afetados%20com%20tal%20mudan%C3%A7a>>. Acesso em: 01 mai. 2024.

FERMANN, R. **Tudo sobre Business Model Canvas.** Disponível em: <<https://medium.com/coproducto/tudo-sobre-business-model-canvas-31a3daac44f8>>. Acesso em: 23 abr. 2024.

IBM. **Casos de uso.** Disponível em: < <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-cases> >. Acesso em: 27 mar. 2024.

IBM. **Diagramas de Atividades.** Disponível em: < <https://www.ibm.com/docs/pt-br/rational-soft-arch/9.7.0?topic=diagrams-activity> >. Acesso em: 30 mar. 2024.

IBM. **Diagramas de Sequência.** Disponível em: <<https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=uml-sequence-diagrams>>. Acesso em: 30 mar. 2024.

LUCIDCHART. **O que é BPMN?** Disponível em: < <https://www.lucidchart.com/pages/pt/o-que-e-bpmn> > Acesso em: 26 mar. 2024.

LUCIDCHART. **O que é um diagrama de classe UML?** Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>>. Acesso em: 30 mar. 2024.

LUCIDCHART. **O que é um diagrama de máquina de estado?** Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-maquina-de-estados-uml>>. Acesso em: 04 mai. 2024.

PAULILLO, G **Matriz SWOT: o que é e como usar na análise estratégica da empresa?** Disponível em: <<https://www.agendor.com.br/blog/matriz-swot-como-fazer/>>. Acesso em: 01 mai. 2024.

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

RIBEIRO, Renato. **Entenda o que são as regras de negócio e como ajudam seu software!** Disponível em: <<https://www.iugu.com/blog/regras-de-negocio>>. Acesso em: 27 mar. 2024.

SOMMERVILLE, Ian. **Engenharia de Software.** 9. ed. São Paulo: Pearson, 2011. 529 p. Disponível em: <<https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>>. Acesso em: 03 mar 2024.

APÊNDICE 1 – Questionário de entrevista com *Stakeholders*

1. Como é realizado o processo de pedidos?
2. Atualmente, qual a sua maior dificuldade?
3. Como é realizado a gerência dos pedidos?
4. Como é realizado o atendimento ao cliente?
5. Você acredita que pode perder vendas por não conseguir atender os clientes imediatamente?
6. Quanto tempo de duração tem em média um atendimento?
7. Atualmente, qual a sua maior dificuldade?
8. Você tem facilidade para utilizar plataformas digitais?
9. Você acredita que seus clientes comprariam seus produtos através de uma loja online?
10. Você acha que um sistema de encomendas e gerenciamento poderia facilitar o dia a dia?