

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

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**GABRIEL AUGUSTO NUNES
SAMUEL SOUSA PIZA**

GERENCIADOR DE PEDIDOS DOCE MARA

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 Eduardo de França Roland

**FRANCA/SP
2025**

GERENCIADOR DE PEDIDOS DOCE MARA

Gabriel Augusto Nunes¹
Samuel Sousa Piza²
Carlos Eduardo de França Roland³

Resumo

Esse relatório apresenta o desenvolvimento de um sistema web voltado à gestão de encomendas de balas de coco personalizadas, criado para modernizar o processo de gestão de pedidos em uma pequena confeitaria que operava de forma manual. A solução propõe otimizar a organização interna, reduzir erros e proporcionar uma experiência mais ágil e confiável aos clientes e gestores do negócio. O sistema foi desenvolvido com arquitetura web, garantindo acessibilidade e praticidade a todos os usuários. A concepção envolveu etapas de análise de requisitos, modelagem de processos, elaboração de diagramas, definição de metas, riscos e *stakeholders*, além da prototipagem da interface voltada ao uso em dispositivos móveis. Durante o desenvolvimento, foram empregadas ferramentas de código aberto e, ao final, realizadas correções, aprimoramentos e validação junto à cliente. O resultado é um sistema funcional e intuitivo, capaz de centralizar informações, automatizar o acompanhamento de pedidos e melhorar a comunicação com o cliente, o que se espera que contribua para o aumento da eficiência operacional e o fortalecimento do negócio.

Palavras-chave: automação de pedidos; confeitarias; gerenciamento de encomendas; sistema *mobile*; sistema web.

Abstract

This report presents the development of a web-based system for managing orders of personalized coconut candies, created to modernize the order management process in a small confectionery shop that previously operated manually. The solution aims to optimize internal organization, reduce errors, and provide a more agile and reliable experience for customers and business managers. The system was developed with a web architecture, ensuring accessibility and practicality for all users. The design involved stages of requirements analysis, process modeling, diagram creation, definition of goals, risks, and stakeholders, as well as prototyping the interface for use on mobile devices. During development, open-source tools were used, and at the end, corrections, improvements, and validation were carried out with the client. The result is a functional and intuitive system capable of centralizing information, automating order tracking, and improving communication with the customer, which is expected to contribute to increased operational efficiency and strengthen the business.

¹ Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: gabriel.nunes29@fatec.sp.gov.br.

² Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: samuel.piza@fatec.sp.gov.br.

³ Docente em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: carlos.roland@fatec.sp.gov.br

Keywords: *bakeries; order automation; order management; mobile system; web system.*

1 Introdução

Na era digital, o uso de sistemas de gerenciamento integrados se torna uma solução eficaz para pequenos negócios que buscam otimizar suas operações. No Brasil, a tendência de digitalização tem crescido exponencialmente, especialmente em setores que demandam interação constante com o cliente e gestão detalhada de pedidos, como é o caso das confeitarias e lojas de doces especializadas em festas e eventos.

Pesquisa realizada pela Associação Brasileira de Bares e Restaurantes (ABRASEL, 2023), que envolveu mais de 1.500 empresários do setor alimentício no Brasil, revelou que o aumento de produtividade e eficiência impulsionados pelo uso de softwares e ferramentas digitais foi relevante para a sobrevivência desses negócios. A pesquisa apontou que 83% dos bares e restaurantes brasileiros utilizavam algum tipo de software para gerenciamento de vendas.

Para contextualizar, a proposta foi o desenvolvimento de um sistema web projetado para gerenciar encomendas de balas de coco personalizadas. Para Bertalanffy (1968), um sistema é um conjunto de elementos inter-relacionados que trabalham em conjunto para atingir um objetivo comum. Com isso em mente, a ideia para este sistema surgiu após observar os desafios enfrentados pela proprietária de uma pequena loja de doces, que atualmente depende de métodos manuais para anotar e gerenciar encomendas, o que frequentemente leva a erros e ineficiências.

Portanto, o objetivo principal do projeto foi analisar, projetar e implementar uma solução digital que facilitasse a gestão de pedidos, desde o recebimento do pedido, a personalização das balas até a entrega para cliente. Especificamente, o sistema visa proporcionar facilidade de uso para que a proprietária possa registrar, acompanhar e atualizar as encomendas de maneira eficiente e sem erros.

Para iniciar o projeto, foram realizadas entrevistas com a proprietária para entender profundamente suas necessidades, e com clientes frequentes para capturar suas preferências e expectativas. Assim, o sistema iria oferecer funcionalidades como a personalização de pedidos, agendamento de datas para eventos e comunicação direta com os clientes para confirmações e atualizações.

A metodologia empregada incluiu observações diretas na loja para o levantamento de requisitos. A elaboração de Diagramas BPMN e da Unified Modeling Language (UML) foi realizada utilizando os softwares Draw.io e Astah Uml, e a modelagem de processos no site BPMN.io. Para a prototipagem de interfaces, foi utilizado o software Figma. Vale ressaltar que, para o projeto, foram considerados os princípios destacados por Pressman (2005), que enfatiza a importância da compreensão clara dos requisitos do usuário e a integração de *feedback* contínuo no processo de desenvolvimento.

O desenvolvimento do sistema foi realizado com a combinação de HTML, Tailwind CSS e TypeScript. No back-end, a escolha foi por Node.js com um banco de dados MySQL para o armazenamento dos dados. A adoção do TypeScript foi estratégica, pois, como um superconjunto do JavaScript, ele adiciona um sistema de tipagem que detecta erros de forma antecipada, contribuindo para a segurança e manutenibilidade do código em projetos de maior escala. Essa característica foi um fator decisivo na escolha da linguagem, conforme estudos que apontam a sua capacidade de facilitar a transição para desenvolvedores (Bierman *et al.*, 2014).

Embora o foco inicial de implantação seja a loja da proprietária, há a possibilidade de expandir para outras lojas de doces que enfrentem desafios semelhantes. A tecnologia será empregada para criar uma experiência mais dinâmica e sem erros na gestão de encomendas, beneficiando tanto o negócio quanto os clientes.

Com o sistema, espera-se melhorar significativamente a operação da loja de doces, oferecendo um gerenciador de encomendas para a proprietária e um cardápio digital, reduzindo o tempo gasto com gerenciamento manual de encomendas e aumentando a satisfação dos clientes com serviços mais rápidos e personalizados.

1.1 Termo da Abertura do Projeto (TAP)

Criado pelo Project Management Institute e apresentado no guia PMBOK, o Termo de Abertura do Projeto, é um documento fundamental na gestão de projetos que oficializa a existência de um projeto e autoriza o seu início. De acordo com Benzik (2021, p. 13) “O termo permite a compreensão clara de base dos requisitos de projeto ao gerente e permitirá a designação de recursos mais eficientes para as atividades do projeto”. Ele serve para garantir que o projeto tenha um alinhamento estratégico com os objetivos da organização e fornece uma visão geral para todas as partes interessadas.

São apresentadas as fases necessárias para a criação do software e a compreensão das demandas da empresa. A análise de cada tópico será realizada para justificar sua relevância no projeto para os executores.

1.1.1 Situação atual

De acordo com a pesquisa realizada com os *stakeholders*, o processo de administração de pedidos é realizado manualmente, utilizando um caderno para registrar as fases da produção e o andamento das encomendas. Esse modelo, embora mantenha certa simplicidade, apresenta limitações operacionais significativas, afetando o controle eficiente dos pedidos e aumentando a vulnerabilidade a erros humanos. A ausência de um sistema digital dificulta o acompanhamento preciso dos prazos e aumenta o risco de falhas na gestão da informação, como a perda de encomendas ou inconsistências nas entregas.

Outro desafio está na falta de padronização do fluxo de dados, o que pode gerar atrasos e dificultar a comunicação dentro da empresa. Além disso, o tempo gasto para localizar informações e validar etapas no processo manual torna a produção mais lenta e menos eficiente, prejudicando a capacidade da empresa de atender a demandas crescentes com rapidez e precisão.

A cultura organizacional da empresa, por outro lado, é um ponto positivo. Os colaboradores demonstram interesse e são abertos a mudanças, criando um ambiente favorável para adaptação a novas práticas e tecnologias. A flexibilidade da equipe é essencial para a transição para um modelo mais moderno e eficiente, baseado em controles digitais e automatizados. Essa receptividade à inovação é uma vantagem importante para implementar um sistema que otimize o fluxo de informações e automatize etapas críticas, sem comprometer a essência artesanal da produção.

Dada a realidade da empresa, que não utiliza computadores de mesa e opera apenas com *smartphones*, o projeto de modernização precisará considerar essa particularidade. Assim, a plataforma de gerenciamento de pedidos será desenvolvida com foco em dispositivos móveis, garantindo que o administrador e os colaboradores possam acessar o sistema de forma prática e eficiente de qualquer lugar.

Com a adoção desse sistema, a empresa espera reduzir a complexidade operacional, evitar retrabalhos e minimizar o risco de atrasos na entrega. A centralização

das informações e automatização das encomendas também irá aumentar a transparência no processo e melhorar a experiência dos clientes.

Em resumo, o cenário apresenta desafios claros na gestão manual, mas a cultura colaborativa e a abertura para mudanças propiciam a implementação de um sistema que facilite o trabalho interno e eleve o nível de eficiência da produção.

1.1.2 Justificativa

Espera-se que a adoção dessa solução tenha impacto positivo direto na capacidade da empresa de transformar sua produção artesanal em uma operação mais moderna, sem perder a essência de qualidade e tradição. Aumentando a eficiência, portanto a empresa estará alinhada com seus objetivos.

Em resumo, a implementação desse sistema não é apenas uma necessidade operacional, mas também uma estratégia de crescimento, garantindo que a empresa possa atender melhor seus clientes e acompanhar as demandas do mercado, sem perder sua identidade artesanal e compromisso com a qualidade.

1.1.3 Propósito e Metas

As metas principais do projeto incluem a automação e centralização da gestão de pedidos, substituindo o controle manual por um sistema digital eficiente; aumentar a eficiência produtiva sem comprometer a essência artesanal, mantendo o compromisso com a qualidade e tradição; garantir maior satisfação tanto para os colaboradores, com fluxos de trabalho mais organizados, quanto para os clientes, com uma plataforma acessível e transparente; e implementar o sistema no prazo definido, seguindo o cronograma previsto e garantindo uma implantação suave e de fácil manutenção.

1.1.4 Missão, Visão e Valores

Define a razão de ser da empresa, seus objetivos de longo prazo e os princípios que orientam suas ações. É criado com base em entrevistas com o administrador da empresa para garantir alinhamento com a organização e guiar a definição estratégica do projeto, garantindo que ele esteja alinhado com a cultura e objetivos da empresa.

1.1.5 Elicitação de Requisitos

Processo de levantamento das necessidades do cliente e funcionalidades desejadas para o sistema. É criado por meio de entrevistas e questionários eletrônicos, garantindo que todas as expectativas e necessidades sejam compreendidas e documentadas.

1.1.6 Análise SWOT

Permite a identificação de forças, fraquezas, oportunidades e ameaças que influenciam o projeto. Definida a partir de discussões e análise da situação da empresa e do mercado, para auxiliar a definição de estratégias para potencializar pontos fortes e mitigar riscos.

1.1.7 Método 5W2H

Ferramenta para planejar ações com base nas perguntas: O quê, Por quê, Onde, Quando, Quem, Como e Quanto, respondidas com base nas definições de atividades, responsáveis e recursos. Garante clareza e organização no plano de ação.

1.1.8 Business Process Modeling Notation (BPMN)

Diagrama que modela os processos de negócios, mapeando o fluxo das atividades e interações entre diferentes setores ou sistemas. Criado a partir da análise do fluxo atual de trabalho e da identificação de melhorias nos processos. Facilita a compreensão e otimização dos processos internos, garantindo eficiência e clareza no fluxo de trabalho.

1.1.9 Documentação de Requisitos

Documento que formaliza todos os requisitos do sistema, organizando as funcionalidades e restrições, sendo criado a partir das informações coletadas na etapa de elicitação. Serve como referência para desenvolvimento e validação do sistema.

1.1.10 Business Model Canvas (BMC)

Ferramenta de gerenciamento estratégico que representa visualmente os componentes fundamentais de um modelo de negócio em um único quadro com 9 blocos

estruturais, preenchido colaborativamente. Permite uma visão clara e holística do funcionamento do negócio.

1.1.11 Termo de Abertura do Projeto (TAP)

Documento formal que autoriza o início do projeto e define seus objetivos e escopo, criado com a colaboração do cliente e equipe de projeto para garantir alinhamento. Estabelece as bases e responsabilidades para o sucesso do projeto.

1.1.12 Diagrama de Caso de Uso

Representação gráfica das interações entre os atores (usuários) e o sistema. É criado com base nos requisitos e funcionalidades levantadas e no BPMN para auxiliar na compreensão de como o sistema será utilizado e suas principais funcionalidades.

1.1.13 Documentação de Caso de Uso

Documento detalhado que descreve cada caso de uso, incluindo fluxos e exceções. Criado a partir do diagrama de caso de uso e discussões com a equipe. Garante clareza no desenvolvimento e testes do sistema, alinhando expectativas entre as partes envolvidas.

1.1.14 Diagrama de Classes

Representa a estrutura do sistema, com as classes, atributos, métodos e relações entre elas, sendo criado a partir da análise dos requisitos para definir os componentes essenciais do sistema, para garantir uma visão clara da arquitetura do sistema, facilitando o desenvolvimento orientado a objetos.

1.1.15 Diagrama de Atividade

Usado para mapear o fluxo de atividades realizadas no sistema, demonstrando o caminho de execução de processos, é definido a partir do mapeamento dos processos no BPMN. Auxilia na identificação de pontos críticos e otimização dos processos.

1.1.16 Diagrama de Máquina de Estado

Representa os estados possíveis de um objeto ou sistema e as transições entre eles. Criado a partir do mapeamento dos processos no BPMN, é fundamental para entender o comportamento dinâmico do sistema.

1.1.17 Diagrama de Sequência

Mostra a interação entre diferentes objetos ou componentes ao longo do tempo, definido a partir do mapeamento dos processos no BPMN para garantir a visão clara do fluxo de mensagens e interações no sistema.

1.1.18 Prototipagem de Telas

É a criação de esboços das telas da interface do sistema, demonstrando a navegação e funcionalidade. Pensada com base nos requisitos de usabilidade e experiência do usuário (UX), permite testar e validar a interface com o cliente antes do desenvolvimento.

1.1.19 Matriz de Rastreabilidade

Relaciona os requisitos do projeto com seus componentes e testes, garantindo que cada funcionalidade seja rastreada. É criado mapeando a correspondência entre os requisitos e os artefatos produzidos e assegura que todos os requisitos foram implementados e testados.

1.1.20 Documentação de Portabilidade

Documento que descreve como o sistema pode ser transferido e implementado em diferentes ambientes. Criado a partir de testes em múltiplos dispositivos e plataformas, para garantir que o sistema funcione adequadamente em diferentes contextos tecnológicos.

1.1.21 Métricas

São os indicadores quantitativos usados para avaliar o desempenho do sistema e do projeto, calculados com base no Diagrama de Classes do projeto, para permitir a indicação de valores quantitativos para a realização do projeto.

1.1.22 Proposta Comercial

Documento formal que detalha a oferta de produtos ou serviços, incluindo escopo, prazos e valores. É definido com base na análise dos requisitos e expectativas do cliente, e serve como base para alinhamento e formalização do contrato.

1.1.23 Premissas

Sistema intuitivo e de fácil uso: o software deve ser desenvolvido com uma interface simples e intuitiva, visando facilitar a adoção por todos os usuários, tanto clientes quanto o administrador. O desenvolvimento e a implementação do sistema devem seguir os prazos e responsabilidades acordados, garantindo que todas as etapas sejam finalizadas conforme planejado. O sistema deve ser projetado para facilitar a sua implantação e futuras manutenções, minimizando a necessidade de suporte técnico contínuo e agilizando correções ou atualizações. Todos os artefatos e protótipos desenvolvidos devem ser validados regularmente com os integrantes do projeto, para garantir que estejam alinhados com as expectativas. A documentação do projeto deve ser revisada constantemente para garantir seu aprimoramento, clareza e utilidade, facilitando a compreensão e futura referência. Deve-se garantir que o sistema atenda aos requisitos de segurança necessários para proteger os dados das encomendas e das informações dos clientes.

1.1.24 Restrições

Todos os membros da equipe do projeto são alunos e estão em fase de aprendizado, o que pode exigir novos conhecimentos adicionais para garantir a realização do projeto. Os membros da equipe, possuem limitações de disponibilidade, o que impacta o cronograma de desenvolvimento e revisão. A empresa utiliza exclusivamente *smartphones*, sem computadores de mesa, o que limita a implementação de soluções que requerem dispositivos mais potentes. O sistema deve ser totalmente funcional em dispositivos móveis, garantindo usabilidade e eficiência.

1.1.25 Stakeholders

Os *stakeholders* são as partes interessadas que desempenham papéis essenciais na implementação e no sucesso do sistema. No contexto deste projeto, destacam-se dois grupos principais de usuários finais: Administrador da Empresa que é o

principal responsável pela gestão do negócio e pela tomada de decisões estratégicas. Ele tem participação ativa na definição dos requisitos do sistema, garantindo que as funcionalidades estejam alinhadas às necessidades operacionais e objetivos da empresa. Seu principal interesse é otimizar a gestão de pedidos e melhorar a eficiência dos processos internos, permitindo uma operação mais organizada e produtiva.

E os Clientes da empresa, que são usuários finais que acessam a plataforma digital para realizar encomendas e acompanhar o status de seus pedidos em tempo real. A implementação do sistema oferece a esses clientes uma experiência mais ágil e prática, eliminando a necessidade de interações manuais, garantindo maior transparência nos prazos e no andamento das entregas.

Esses *stakeholders* são fundamentais para o sucesso do projeto, pois representam tanto a demanda operacional (Administrador) quanto a experiência final do consumidor (Clientes), garantindo que o sistema atenda plenamente às expectativas de eficiência e satisfação.

1.1.26 Riscos

A qualidade da conexão à internet pode interferir no uso do sistema, especialmente se a rede for instável ou de baixa velocidade. Os membros da equipe podem ter restrições de tempo, afetando o desenvolvimento e implementação dentro dos prazos estipulados. O sistema deve garantir a proteção de dados dos clientes, conforme as exigências da Lei Geral de Proteção de Dados (LGPD). O não cumprimento pode resultar em penalidades legais. Alterações não planejadas durante o projeto podem comprometer prazos. A má comunicação entre a equipe de desenvolvimento e o administrador pode resultar em falhas no projeto.

1.1.27 Marcos

O Quadro 1 mostra os principais marcos do plano de execução do projeto.

2 Viabilidade do Projeto

Como base para o estudo de viabilidade do projeto, foi usado o Business Model Canvas que é uma ferramenta de gestão e planejamento estratégico que auxilia na compreensão, análise e representação visual dos modelos de negócio. Por meio de

um quadro dividido em nove blocos, ele descreve de forma clara como uma organização cria, entrega e captura valor. Essa metodologia, amplamente utilizada em ambientes empresariais e acadêmicos, busca tornar o processo de modelagem de negócios mais dinâmico e colaborativo, facilitando a visualização das relações entre os diferentes elementos que compõem o negócio (Vicelli; Tolfo, 2017).

Quadro 1 – Marcos do projeto

Data	Atividades
Agosto/2024	Definição de projeto Missão, Visão e Valor Levantamento de requisitos com entrevista e questionário
Setembro/2024	BMC Matriz SWOT Método 5W2H BPMN
Outubro/2024	Documento de Requisitos Diagrama de Caso de Uso Documento de Caso de Uso Estrutura Analítica do Projeto Termo de Abertura do Projeto Matriz de Rastreabilidade Diagramas de Sequência Diagramas de Máquina de Estado Diagrama de Sequência Diagrama de Classe
Novembro/2024	Documento de Portabilidade Métricas Proposta Comercial Prototipação inicial das Telas
1ºSemestre/2025	Prototipação final das Telas Desenvolvimento do Sistema
2ºSemestre/2025	Correções e melhoras no Sistema Correções na Documentação

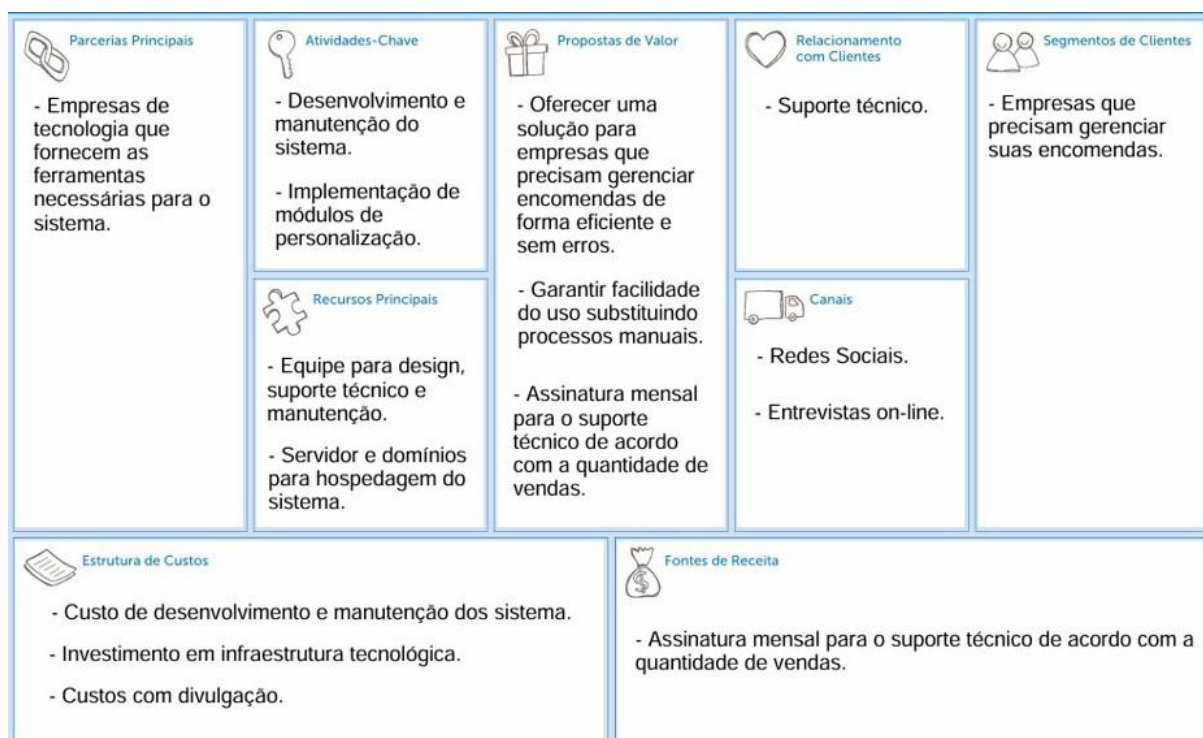
Fonte: os autores

Vicelli e Tolfo (2017) destacam que a competitividade organizacional está frequentemente vinculada à capacidade de criação de valor, o que motiva a adoção de ferramentas que facilitem a proposição de valor no contexto de modelos de negócio, e que a proposta de Osterwalder e Pigneur é amplamente utilizada por sua clareza conceitual. Eles ressaltam que o Canvas é uma ferramenta visual capaz de sistematizar essa lógica por meio de nove blocos interconectados: Segmentos de Clientes, Proposta de Valor, Canais, Relacionamento com Clientes, Fontes de Receita, Recursos Principais, Atividades-Chave, Parcerias Principais, e Estrutura de Custos. Esses blocos são dispostos de modo a visualizar claramente como a empresa gera e entrega valor ao cliente, bem como estrutura seus custos e receitas, permitindo

analisar ou propor modelos de negócio mais coerentes e adaptáveis. A Figura 1 apresenta o Canvas de Negócio.

A estrutura do BMC pode ser analisada e compreendida por meio de quatro áreas macro estratégicas inter-relacionadas: O Que será oferecido ao mercado, representado pela Proposta de Valor; Para Quem a solução se destina, englobando os Segmentos de Clientes, os Canais de distribuição e comunicação e o tipo de Relacionamento estabelecido; Como a solução será operacionalizada e entregue, o que constitui a Infraestrutura, formada pelos Recursos, Atividades e Parcerias Chave; e, por fim, a Viabilidade Financeira do modelo, que equilibra as Fontes de Receita com a Estrutura de Custos. Esta divisão permite uma visão sistêmica do modelo de negócio para o aplicativo.

Figura 1 – Canvas de Negócio



Fonte: os autores

O Segmento de Clientes definido compreende pequenas e médias empresas que necessitam gerenciar suas encomendas; a Proposta de Valor central consiste em oferecer uma solução para que elas gerenciem encomendas de forma eficiente e sem erros, substituindo processos manuais por um sistema automatizado e de fácil uso. Adicionalmente, é oferecida uma assinatura mensal para suporte técnico, cujo valor é escalonado de acordo com a quantidade de vendas do cliente, garantindo flexibilidade e suporte adequado ao seu volume de operações; os Canais estabelecidos para

comunicação, divulgação e interação com os clientes são as redes sociais e reuniões on-line. Esses meios digitais permitem um alcance amplo e eficiente, facilitando a prospecção, o atendimento e a manutenção do relacionamento com o público-alvo; o Relacionamento com os clientes é sustentado primordialmente pelo suporte técnico. Esse serviço contínuo visa assegurar a satisfação do cliente, a resolução de problemas e a fidelização, alinhando-se diretamente com o modelo de assinatura mensal.

A principal Fonte de Receita do modelo de negócio advém de uma assinatura mensal para o suporte técnico, cujo valor é variável e proporcional à quantidade de vendas realizadas pelo cliente. Esse modelo de monetização é típico de negócios do tipo Software como Serviço.

Os Recursos Chave essenciais para a operação do negócio incluem uma equipe qualificada para design, suporte técnico e manutenção do sistema, bem como a infraestrutura tecnológica composta por servidor e domínios para hospedagem da plataforma; as Atividades-Chave envolvem o desenvolvimento e a manutenção contínua do sistema, além da implementação de módulos de personalização. Tais atividades são críticas para garantir a funcionalidade, a atualização e a adaptação do *software* às necessidades específicas dos clientes. As Parcerias-Chave, estratégicas, são estabelecidas com empresas de tecnologia que fornecem as ferramentas necessárias para o desenvolvimento e a operação do sistema. Essas parcerias são vitais para acessar tecnologias, ferramentas e expertise que complementam e viabilizam a oferta do serviço.

E a Estrutura de Custos é composta pelos custos de desenvolvimento e manutenção do sistema, investimentos em infraestrutura tecnológica e custos com divulgação. Esses elementos representam os principais gastos operacionais e de investimento inicial para a consolidação e o crescimento do negócio.

2.1 Matriz SWOT

Na análise SWOT realizada (Fig. 2), duas fraquezas principais foram identificadas. Processos lentos para realizar encomendas que pela falta de um sistema automatizado faz com que o processo de encomenda seja demorado, o que pode impactar negativamente a experiência do cliente. Esse problema reduz a agilidade da operação e pode levar a atrasos, afetando a satisfação do cliente e a eficiência do negócio. E a ausência de um sistema automatizado indicando a

dependência excessiva a processos manuais, o que dificulta o atendimento a um volume maior de pedidos e aumenta a chance de erros humanos. Além disso, essa limitação impede a empresa de aproveitar tecnologias que poderiam facilitar a centralização e o controle das encomendas, reduzindo a carga de trabalho e permitindo uma melhor gestão de tempo e recursos.

Essas fraquezas destacam a necessidade de investir em tecnologias que otimizem o processo de encomenda, o que resultaria em melhorias significativas na operação e na satisfação dos clientes.

Figura 2 - Matriz SWOT



Fonte: os autores

2.3 Método 5W2H

A ferramenta 5W2H é uma metodologia de planejamento e gestão que visa ajudar na definição e execução de planos de ação. Conforme Foccoerp (2023) o método “surgiu no Japão durante a década de 1950. Nesse período, o país buscava se posicionar como um líder global na produção e gestão, com as empresas visionárias reconhecendo a importância de aprimorar suas práticas organizacionais”.

A sigla 5W2H vem do inglês e representa sete perguntas essenciais que devem ser respondidas para uma análise completa de um problema ou de uma tarefa: *What* (O que): o que será feito? *Why* (Por quê): por que será feito? *Where* (Onde): onde será feito? *When* (Quando): quando será feito? *Who* (Quem): quem fará? *How* (Como) como será feito? *How Much* (Quanto custa): quanto vai custar?

O método 5W2H é amplamente utilizado em gestão de projetos, qualidade e processos devido à sua simplicidade e eficiência em organizar planos de ação de forma clara e objetiva. Ele oferece uma estrutura que ajuda a garantir que todos os aspectos relevantes de um projeto ou tarefa sejam considerados, promovendo uma execução eficiente e evitando falhas decorrentes de lacunas no planejamento.

Na opinião de Paula (2015):

Para uma empresa que busca crescer de forma saudável e acelerada, sem perder o controle das finanças, ou seja, por meio de um bom processo de Planejamento e Orçamento, a utilização da metodologia 5W2H auxilia a trazer praticidade e organização ao processo de tomada de decisão.

O Quadro 2 apresenta as questões-problema organizadas por meio da matriz 5W2H, facilitando a análise e o planejamento das ações necessárias para o projeto. O plano de ação proposto visa resolver dois problemas principais da loja: processos lentos para realizar encomendas e a falta de um sistema automatizado. Esses problemas comprometem a eficiência operacional e a satisfação dos clientes, afetando diretamente a produtividade e a competitividade da loja.

O plano de ação está bem estruturado, com metas claras e prazos definidos. A automação dos processos e a digitalização da gestão de encomendas são soluções eficazes que aumentam a competitividade da loja, além de melhorar a experiência do cliente. O uso de ferramentas de análise de métricas permitiu o acompanhamento do progresso e o ajuste das estratégias ao longo do tempo, garantindo que as metas fossem alcançadas.

3 Elicitação e Especificação de Requisitos

Para o desenvolvimento do sistema de gestão de pedidos, foram empregadas técnicas de elicitação de requisitos para garantir a compreensão das necessidades e expectativas do cliente.

As principais técnicas empregadas foram entrevistas com o gestor da loja e a aplicação de um questionário para os clientes da empresa.

Quadro 2 - Matriz 5W2H

Questão 1: Processos lentos para realizar encomendas	
O que?	Desenvolver um sistema para digitalizar e automatizar a gestão das encomendas com controle centralizado de pedidos e alertas automáticos
Por quê?	Para reduzir erros, melhorar a organização e aumentar a eficiência no processos, evitar a perda de pedidos e garantir que prazos sejam respeitados
Onde?	O sistema será implementado na loja e acessível via web e integrável em dispositivos móveis
Quando?	Em dezoito dias aproximadamente
Quem?	O gestor da loja e a equipe Fatec
Como?	Criando o sistema para integrar os pedidos e fazer o gerenciamento das encomendas com interface para acompanhar os pedidos e receber alertas
Quanto?	Através da análise de métricas
Questão 2: Falta de sistema automatizado.	
O que?	Criar um sistema online onde os clientes podem fazer encomendas diretamente, e possa haver a comunicação interna entre os setores de produção e atendimento
Por quê?	Para acelerar o processo, permitindo que o cliente realize o pedido sem precisar de atendimento manual, diminuir o tempo de resposta e evitar atrasos no processamento de encomendas
Onde?	No site da loja, com acesso via desktop ou mobile e na loja física
Quando?	Em dezoito dias aproximadamente
Quem?	O gestor da loja e a equipe Fatec
Como?	Criando uma interface intuitiva que permita ao cliente inserir e acompanhar seus pedidos on-line, e a gestão interna com integração de fluxo de trabalho
Quanto?	Através da análise de métricas

Fonte – Os autores

As entrevistas com o administrador da empresa permitiram uma análise detalhada das operações atuais e ajudaram a identificar os principais desafios enfrentados na gestão de pedidos. O gestor foi questionado sobre o fluxo de encomendas, tempos de resposta e etapas críticas que poderiam ser aprimoradas com a digitalização. Para obter uma perspectiva dos clientes e entender melhor suas expectativas, foi elaborado um questionário no Google Forms. Este questionário permitiu coletar dados importantes sobre a preferências dos clientes em relação ao sistema de pedidos, abordando fatores como usabilidade e formas ideais de comunicação sobre o status das encomendas.

A elicitação de requisitos é necessária para o sucesso de um projeto, pois garante que as necessidades reais dos usuários sejam bem compreendidas e traduzidas em funcionalidades precisas. “Ela é uma atividade essencial da ES, pois visa resolver estes problemas, mostrando claramente o que deve ser feito, procurando estabelecer definições claras, precisas e não ambíguas” (Batista, 2003, p. 5).

O levantamento de requisitos com o uso das técnicas descritas permitiu uma visão abrangente e detalhada das necessidades do projeto, orientando o desenvolvimento do sistema. As especificações de requisitos mais relevantes são: a facilidade de registro de pedidos pois o administrador precisa de um sistema rápido e intuitivo para cadastrar novos pedidos e acompanhar o *status*, sem recorrer ao uso de papéis; a facilidade, pelos cliente, de realizar pedidos, que devem poder acessar o sistema para realizar pedidos de forma simples e rápida; e a automação de processos com armazenamento seguro de dados.

A utilização de entrevistas com questionário via Google Forms, e a observação direta contribuíram para um levantamento preciso dos requisitos, alinhado com as expectativas do cliente. Esses elementos formam o escopo do sistema, proporcionando uma solução tecnológica que aprimora o processo de pedidos e promove eficiência, mantendo o compromisso com a qualidade e o atendimento.

Os questionários e suas respostas encontram-se no Apêndice 1 deste documento.

3.1 BPMN

Como aponta TOTVS (2023), o *Business Process Model and Notation* (BPMN) surgiu em agosto de 2000, a partir da iniciativa de líderes de negócios ligados à BPMI, uma organização sem fins lucrativos, com o objetivo de promover uma padronização de processos aplicável a todos os negócios, e atualmente é mantido pela OMG, que se fundiu à BPMI em 2004.

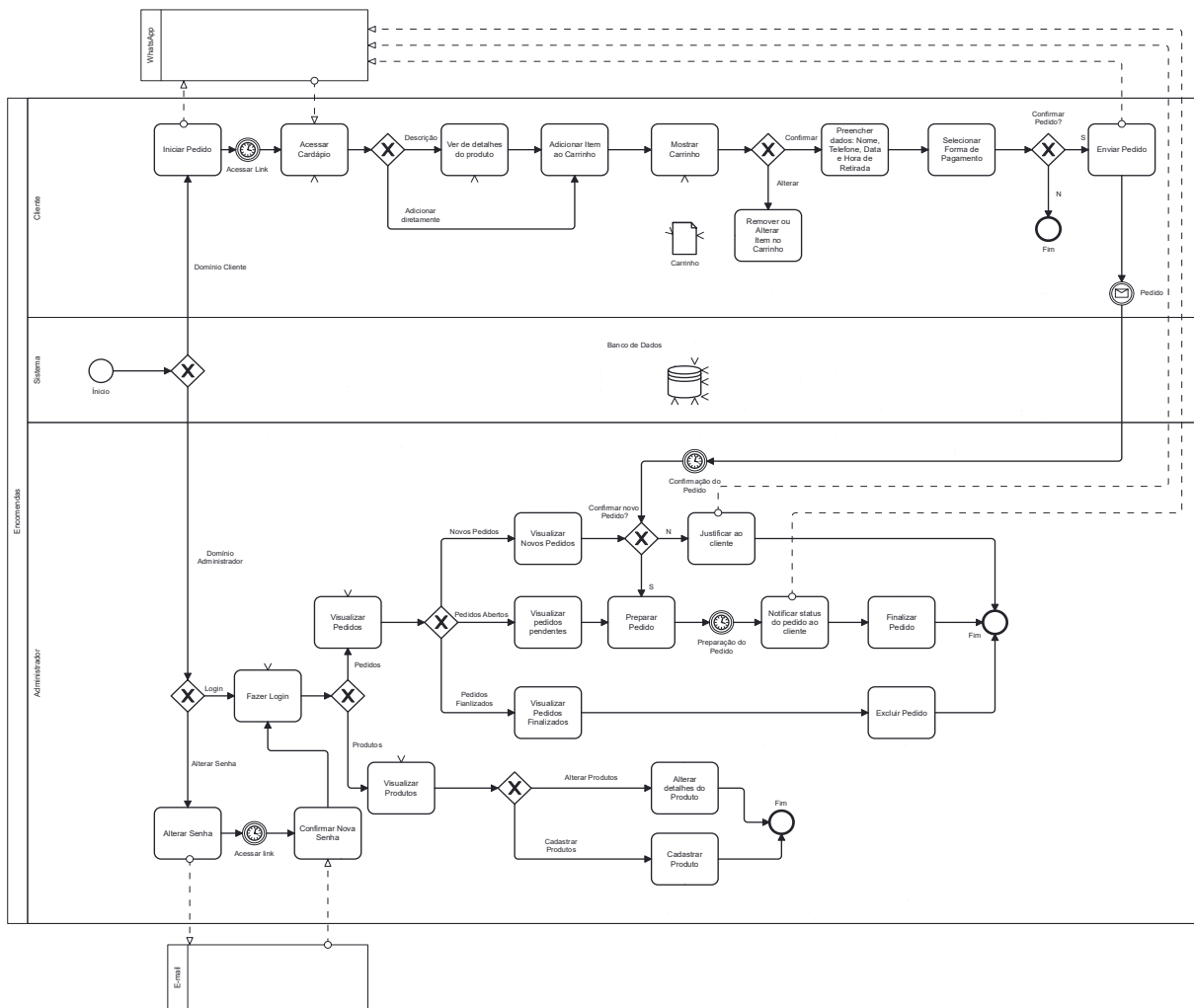
O BPMN é uma notação gráfica frequentemente empregada para ilustrar processos empresariais. Segundo Borges (2023, p. 14) ela “fornece uma notação visual para representar claramente os elementos e as etapas de um processo, incluindo eventos, atividades, decisões, fluxos de sequência e interações entre diferentes partes”.

A ferramenta é especialmente útil para facilitar a documentação, análise e otimização de processos simplificando a comunicação entre equipes de áreas

distintas, promovendo um alinhamento entre a visão do negócio e a execução técnica. A Figura 3 apresenta a modelagem BPMN do projeto.

A notação BPMN não só apoia a transição para uma gestão mais digitalizada como também assegura que a automação proposta esteja alinhada com o fluxo atual de trabalho e com as necessidades do administrador e dos clientes.

Figura 3 – BPMN



Fonte: Os autores

3.2 Requisitos Funcionais (RF)

“De maneira simplificada, o requisito funcional descreve ‘o que’ o sistema deve fazer. Por exemplo, um sistema de cadastro de mercadorias deve permitir que o usuário gere um relatório de itens cadastrados na base” (Mendes, 2023, p. 18).

Os requisitos a seguir exibidos no Quadro 3 foram criados para orientar o desenvolvimento, pois traduzem as expectativas do cliente em funcionalidades concretas.

Quadro 3 – Requisitos Funcionais

ID RF001	Nome do Requisito: Acessar Cardápio
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá mostrar ao cliente as opções de produtos disponíveis.
Informações	Nome, miniatura e preço dos produtos
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF002	Nome do Requisito: Mostrar Detalhes do Produto
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá mostrar ao cliente os detalhes do produto
Informações	Nome, imagem, descrição e preço do produto.
Regra de Negócio	Não há nenhuma regra de negócio.
ID RF003	Nome do Requisito: Adicionar item ao Carrinho
Categoria	Evidente.
Prioridades	Importante.
Descrição	O sistema deverá adicionar produtos ao carrinho de forma com que o cliente possa escolher a quantidade do produto.
Informações	Nome, quantidade, descrição e preço dos produtos.
Regra de Negócio	<ol style="list-style-type: none"> 1. A quantidade total do produto deve ser, no mínimo, 500 gramas. 2. A quantidade total do produto deve ser, no máximo, 50 quilogramas.
ID RF004	Nome do Requisito: Mostrar Carrinho
Categoria	Evidente.
Prioridades	Importante.
Descrição	O sistema deverá mostrar o carrinho ao cliente com todos os produtos adicionados para a encomenda.
Informações	Nome, quantidade, descrição e preço dos produtos.
Regra de Negócio	<ol style="list-style-type: none"> 1. A quantidade total do pedido deve ser, no mínimo, 500 gramas. 2. A quantidade total do pedido deve ser, no máximo, 50 quilogramas.
ID RF005	Nome do Requisito: Remover ou Alterar item do Carrinho
Categoria	Evidente.
Prioridades	Importante.
Descrição	O sistema deverá remover ou alterar as quantidades produto no carrinho caso o cliente queira.
Informações	Nome, quantidade, descrição e preço dos produtos.
Regra de Negócio	<ol style="list-style-type: none"> 1. A quantidade total do produto deve ser, no mínimo, 500 gramas. 2. A quantidade total do produto deve ser, no máximo, 50 quilogramas.
ID RF006	Nome do Requisito: Enviar Pedido
Categoria	Evidente.
Prioridades	Essencial.

Descrição	O sistema deverá gerar um formulário para a coleta de dados sobre o cliente, data para retirada e forma de pagamento, pedir a confirmação do pedido e gerar uma mensagem com as informações do pedido para ser enviada para o perfil do WhatsApp da empresa.
Informações	Nome do cliente, telefone, forma de pagamento, data para retirada, nomes e quantidades dos produtos e preço do pedido.
Regra de Negócio	<ol style="list-style-type: none"> 1. Caso a quantidade total do pedido seja inferior a 4 kg, a data de retirada deve ser, no mínimo, de um dia. Para pedidos entre 5 e 10 kg, a data de retirada deve ser de dois dias. Para pedidos entre 10 kg e 20 kg, a data de retirada deve ser de uma semana. Para pedidos entre 20 kg e 50 kg, a data de retirada deve ser de 15 dias. 2. O horário de retirada deve ser de segunda-feira a sexta-feira das 9h às 17h e no sábado das 9h às 13h. 3. A Contagem total dos dias para retirada deve excluir os sábados e os domingos.
ID RF007	Nome do Requisito: Fazer Login no Sistema
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deve conter um processo de login para o administrador.
Informações	Usuário, senha e Email.
Regra de Negócio	Não há regra de negócio.
ID RF008	Nome do Requisito: Visualizar novos Pedidos
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá mostrar ao administrador os novos pedidos para a sua confirmação.
Informações	Código do pedido, nome do cliente, telefone, forma de pagamento, data para retirada, nome e quantidades dos produtos e preço total da encomenda.
Regra de Negócio	Não há regra de negócio.
ID RF009	Nome do Requisito: Visualizar Pedidos em aberto
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá mostrar ao administrador os pedidos já confirmados.
Informações	Código do pedido, nome do cliente, telefone, forma de pagamento, data para retirada, nome e quantidades dos produtos e preço total da encomenda.
Regra de Negócio	Não há regra de negócio.
ID RF0010	Nome do Requisito: Finalizar Pedido
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá mover pedidos em aberto para pedidos finalizados quando eles forem completados.

Informações	Código do pedido, nome do cliente, telefone, forma de pagamento, data para retirada, nome e quantidades dos produtos e preço total da encomenda.
Regra de Negócio	Não há regra de negócio.
ID RF011	Nome do Requisito: Visualizar Pedidos Finalizados
Categoria	Evidente.
Prioridades	Importante.
Descrição	O sistema deverá mostrar ao administrador os pedidos já finalizados.
Informações	Código do pedido, nome do cliente, telefone, forma de pagamento, data para retirada, nome e quantidades dos produtos e preço total da encomenda.
Regra de Negócio	Não há regra de negócio.
ID RF012	Nome do Requisito: Excluir Pedido
Categoria	Evidente.
Prioridades	Importante.
Descrição	O sistema deverá dar a opção de excluir um pedido ao administrador caso ele tenha sido finalizado.
Informações	Código do pedido, nome do cliente, telefone, forma de pagamento, data para retirada, nome e quantidades dos produtos e preço total da encomenda.
Regra de Negócio	Não há regra de negócio.
ID RF013	Nome do Requisito: Visualizar Produtos
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá ter funcionalidades para o gerenciamento do cardápio, para que o administrador cadastre e altere seus produtos.
Informações	Nome, quantidade, descrição e preço dos produtos.
Regra de Negócio	Não há regra de negócio.
ID RF014	Nome do Requisito: Cadastrar Produtos
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá ter a função de cadastrar produtos para inseri-los no cardápio.
Informações	Nome, descrição e preço dos produtos.
Regra de Negócio	Não há regra de negócio.
ID RF015	Nome do Requisito: Alterar Produtos
Categoria	Evidente.
Prioridades	Essencial.
Descrição	O sistema deverá ter a função de alterar e desativar ou ativar produtos do cardápio.
Informações	Nome, descrição e preço dos produtos.
Regra de Negócio	Não há regra de negócio.
ID RF016	Nome do Requisito: Registrar Dados no Banco de Dados
Categoria	Oculto.
Prioridades	Essencial.

Descrição	O sistema deve conter um banco de dados para armazenar cada processo feito pelo administrador e os pedidos realizados pelos clientes.
Informações	Dados dos pedidos, clientes e produtos armazenados.
Regra de Negócio	Não há regra de negócio.

Fonte: Os autores

Esses requisitos tornam-se uma referência para os desenvolvedores e para o próprio cliente, assegurando que o sistema final entregue as funcionalidades necessárias para suportar as operações de forma eficiente e confiável.

3.3 Requisitos Não Funcionais (RNF)

Requisitos Não Funcionais referem-se aos critérios que especificam como o sistema deve operar, em vez de descrever as funções que ele executa. Eles abordam atributos de qualidade, como desempenho, segurança, usabilidade, confiabilidade e portabilidade. Como destaca Mendes (2023, p. 19) “Basicamente, esse tipo de requisito descreve “como” o sistema desempenhará as funcionalidades solicitadas”. O Quadro 4, mostram os RNF do sistema.

Quadro 4 – Requisitos Não Funcionais

ID RNF001	Nome do Requisito: Login
Categoria	Segurança.
Prioridades	Essencial.
Descrição	O sistema deve conter um processo de login para o administrador, incluindo a opção de recuperar a senha.
Informações	Usuário, senha, Email.
Regra de Negócio	Não há regra de negócio.
ID RNF002	Nome do Requisito: Sistema Web
Categoria	Compatibilidade.
Prioridades	Essencial.
Descrição	O sistema deve ser construído para ser executado em navegadores web.
Informações	Compatível com navegadores modernos (Google Chrome, Firefox, Microsoft Edge).
Regra de Negócio	Não há regra de negócio.

Fonte: Os autores

Os RNF apresentados, asseguram que o sistema será confiável, fácil de usar e compatível com as limitações de infraestrutura da empresa.

3.4 Casos de Uso

Um Caso de Uso é uma técnica de modelagem usada para descrever como um sistema interage com seus usuários, os atores, para realizar uma função específica. “Um caso de uso pode ser tomado como um cenário simples que descreve o que o usuário espera de um sistema” (Sommerville, 2011, p.86 *apud* Moreno, Masson, e Caraviéri, 2022, p. 6).

A importância dos casos de uso no desenvolvimento de sistemas é substancial, pois eles promovem uma compreensão clara dos requisitos e das interações fundamentais entre usuários e sistema.

O objetivo dos casos de uso é a identificação das funcionalidades requeridas para diagramas de casos de uso o sistema. Assim, os casos de uso incluem-se na fase de ANÁLISE DE REQUISITOS, a fase em que procuramos identificar, da melhor forma possível, o que é que o nosso sistema (frequentemente um ou mais programas) deve realmente ser capaz de fazer. (Barros, 2009, p.1).

A Figura 4 exibe o diagrama de caso de uso, destacando as principais interações entre os usuários e o sistema, permitindo uma visão clara das funcionalidades e requisitos do projeto.

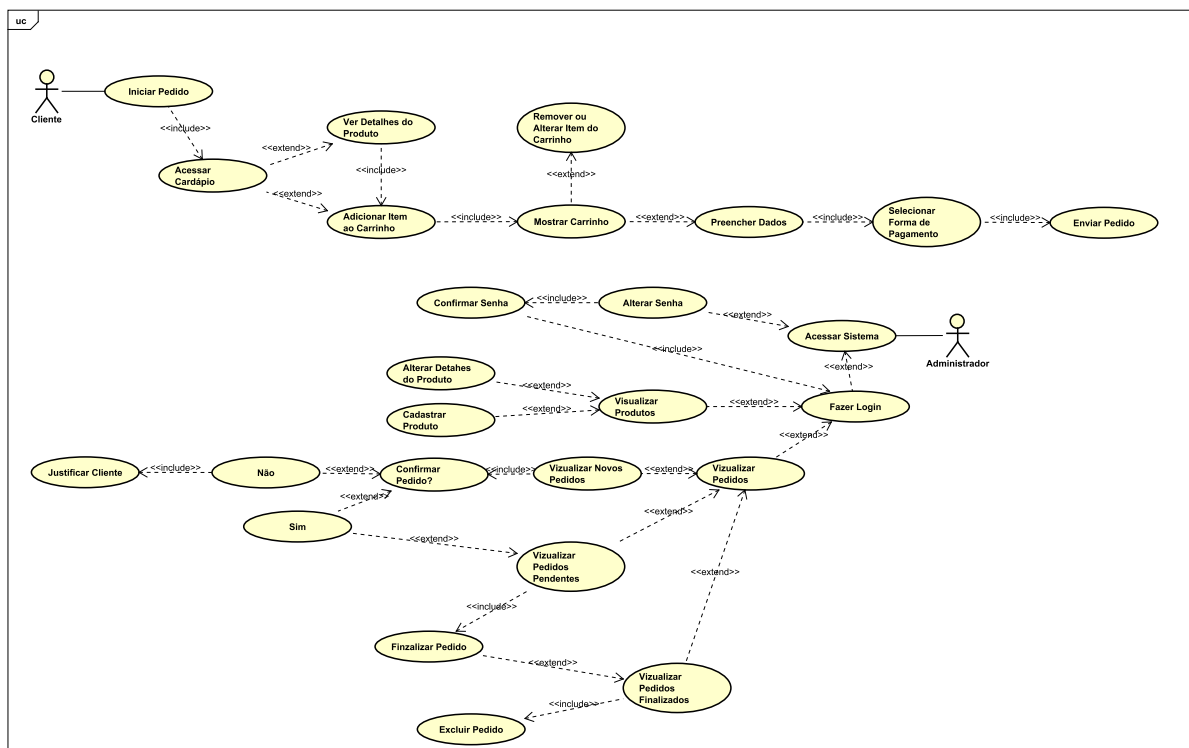
O Quadro 5, apresenta detalhadamente os casos de uso, descrevendo as interações, objetivos, pré-condições, fluxos de eventos e resultados esperados para cada funcionalidade do sistema.

3.5 Diagrama de Classes

Um Diagrama de Classes é um tipo de diagrama estrutural usado na modelagem orientada a objetos para representar a estrutura de um sistema, ilustrando suas classes, atributos, métodos e as relações entre elas. De acordo com Felisbino (2017, p. 20) “Este diagrama apresenta uma visão estática de como as classes estão organizadas, preocupando-se em definir a estrutura lógica das mesmas.

O valor do diagrama de classes reside na sua capacidade de proporcionar uma visão detalhada da estrutura interna do sistema e de suas interações. “O Diagrama de Classes serve como base para a construção da maior parte dos demais diagramas da UML” (Felisbino, 2017, p. 20).

Figura 4 – Diagrama de Caso de Uso



Fonte: Os autores

Quadro 5 – Descrições dos Casos de Uso

Caso de Uso – Iniciar Pedido	
ID	UC 001
Descrição	Este caso de uso tem como objetivo iniciar um pedido.
Ator Primário	Cliente
Pré-condição	Não possui.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o cliente solicita o cardápio por meio do WhatsApp. 2. Uma mensagem automática é enviada ao cliente com o link para acessar o cardápio. 3. O cliente aperta no link.
Pós-condição	Não possui.
Cenário Alternativo	Não possui.
Inclusão	Não possui.
Extensão	Não possui.
Caso de Uso – Acessar Cardápio	
ID	UC 002
Descrição	Este caso de uso tem como objetivo acessar o cardápio.
Ator Primário	Cliente
Pré-condição	Ter solicitado o link do cardápio.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o cliente acessa o link do cardápio. 2. O cliente é redirecionado ao cardápio com os produtos disponíveis. 3. Quando item selecionado o sistema deve mostrar os detalhes do produto: preço e descrição.

Pós-condição	Não possui.
Cenário Alternativo	3a – Caso o cliente clique diretamente no ícone o caso UC 003 iniciará.
Inclusão	UC 001 – Iniciar Pedido
Extensão	Não possui.
Caso de Uso – Uso Ver Detalhes do Produto	
ID	UC 003
Descrição	Este caso de uso tem como objetivo mostrar o cardápio.
Ator Primário	Cliente
Pré-condição	Deve haver produtos cadastrados
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o cliente acessa o clica sobre o produto. 2. O cliente é redirecionado à tela que mostra detalhes do produto (nome, imagem, preço e descrição). 3. O cliente seleciona a opção de adicionar
Pós-condição	Não possui.
Cenário Alternativo	3a – o cliente clica no botão de voltar
Inclusão	Não possui
Extensão	UC 002 – Acessar Cardápio
Caso de Uso – Adicionar Item ao Carrinho	
ID	UC 004
Descrição	Este caso tem como objetivo adicionar um item ao carrinho.
Ator Primário	Cliente.
Pré-condição	Deve haver produtos cadastrados.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso se inicia quando o cliente seleciona a opção adicionar pela tela de detalhes ou pelo ícone no cardápio. 2. O cliente seleciona a quantidade. 3. O sistema adiciona o item ao carrinho. 4. O sistema redireciona o cliente para o carrinho
Pós-condição	Não possui.
Cenário Alternativo	<p>2a – O cliente pode clicar no botão de voltar.</p> <p>2b – O sistema pode barrar o cliente de prosseguir, caso a regra de negócio (RF003) não seja satisfeita.</p>
Inclusão	UC 003 – Ver detalhes do Produto
Extensão	UC 002 – Acessar Cardápio
Caso de Uso – Mostrar Carrinho	
ID	UC 005
Descrição	Este caso de uso tem como objetivo mostrar o carrinho com os itens do pedido.
Ator Primário	Cliente.
Pré-condição	Não possui.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o cliente seleciona a opção ver carrinho. 2. O sistema carrega todos os itens do pedido com quantidade e preço final. 3. O cliente seleciona continuar. 4. O sistema redireciona o cliente para preencher os dados.

Pós-condição	Não possui.
Cenário Alternativo	1a – O cliente pode acessar o carrinho pela tela do cardápio, caso já haja algum item nele. 2a – O cliente pode editar a quantidade de produto no carrinho. 2b – O cliente pode remover o item do carrinho. 3a – O sistema pode barrar o cliente de prosseguir, caso a regra de negócio (RF004) não seja satisfeita. 3b – O cliente poder clicar do botão voltar
Inclusão	UC 004 - Adicionar Item ao Carrinho
Extensão	Não Possui
Caso de Uso – Remover ou Alterar Item do Carrinho	
ID	UC 006
Descrição	Este caso de uso tem como objetivo remover ou alterar um item do carrinho.
Ator Primário	Cliente.
Pré-condição	Ter itens adicionado ao carrinho.
Cenário Principal	1. O caso de uso inicia quando o cliente seleciona o carrinho. 2. O sistema mostra o item adicionado, com valor e quantidade. 3. O cliente seleciona o ícone remover item. 4. O sistema retira o item do carrinho.
Pós-condição	Não possui.
Cenário Alternativo	3a – O cliente seleciona o ícone de alterar item no carrinho. 3b – O sistema pode barrar o cliente de prosseguir, caso a regra de negócio (RF003) não seja satisfeita. 3c – O cliente clicar do botão de voltar. 4a – O sistema altera a quantidade do item.
Inclusão	Não possui.
Extensão	UC 005 – Mostrar Carrinho
Caso de Uso – Preencher Dados	
ID	UC 007
Descrição	Este caso de uso tem como objetivo informar os dados para realizar o pedido.
Ator Primário	Cliente.
Pré-condição	Ter itens adicionado ao carrinho.
Cenário Principal	1. O caso de uso inicia quando o cliente seleciona a opção continuar. 2. O sistema carrega o formulário para a preencher os dados. 3. O cliente preenche os campos nome, telefone e informa a data e hora de retirada (disponível de acordo com a quantidade). 4. O cliente clica em continuar. 5. O sistema redireciona o cliente para pagamento.
Pós-condição	Não possui.
Cenário Alternativo	3a – O sistema pode barrar o cliente de prosseguir, caso a regra de negócio (RF006) não seja satisfeita. 4a - O sistema pode barrar o cliente de prosseguir, caso o formulário não seja completamente preenchido.

	4b – O cliente pode clicar no botão de voltar.
Inclusão	Não possui.
Extensão	UC 005 – Mostrar Carrinho
Caso de Uso – Selecionar Forma de Pagamento	
ID	UC 008
Descrição	Este caso de uso tem como objetivo informar qual será a forma de pagamento do pedido.
Ator Primário	Cliente.
Pré-condição	Ter preenchido os dados do cliente.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o cliente seleciona a opção selecionar forma de pagamento. 2. O sistema mostra as opções de pagamento disponíveis. 3. O cliente seleciona a opção desejada. 4. O sistema mostra a opção selecionada. 5. O cliente confirma.
Pós-condição	Não possui.
Cenário Alternativo	<p>5a- O cliente clica do botão de voltar.</p> <p>5b -O sistema pode barrar o cliente de prosseguir, caso a forma de pagamento não seja selecionada.</p>
Inclusão	UC 007 – Preencher dados
Extensão	Não possui.
Caso de Uso – Enviar Pedido	
ID	UC 009
Descrição	Este caso de uso tem como objetivo enviar o pedido para a loja.
Ator Primário	Cliente.
Pré-condição	Ter preenchido os dados do cliente e de pagamento.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o cliente seleciona a opção continuar. 2. O sistema pergunta ao cliente se ele deseja confirmar o pedido 3. O cliente confirma o pedido 4. O sistema guarda o pedido 5. O sistema produz um resumo do pedido, abre o WhatsApp do cliente e o cliente envia o resumo para o perfil do administrador
Pós-condição	A loja deve confirmar o pedido.
Cenário Alternativo	<p>3a – O cliente cancela o pedido.</p> <p>5a – O cliente não envia o resumo do pedido para o perfil no WhatsApp do administrador</p>
Inclusão	UC 008 – Selecionar Forma de Pagamento
Extensão	Não possui.
Caso de Uso – Acessar o Sistema	
ID	UC 010
Descrição	Esse caso de uso tem como objetivo acessar o sistema.
Ator Primário	Administrador.
Pré-condição	Não possui.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o Administrador inicia o sistema. 2. O sistema inicia com as opções de login e alterar senha.

Pós-condição	Não possui.
Cenário Alternativo	Não possui.
Inclusão	Não possui.
Extensão	Não possui.
Caso de Uso – Alterar Senha	
ID	UC 011
Descrição	Este caso de uso tem como objetivo alterar a senha do administrador.
Ator Primário	Administrador.
Pré-condição	Possuir uma conta cadastrada.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção alterar senha. 2. O sistema abre um campo para inserir o e-mail no qual a conta está cadastrada. 3. O administrador informa o e-mail para alterar a senha. 4. O sistema verifica se o e-mail informado possui cadastro. 5. O sistema confirma que o e-mail possui cadastro. 6. O sistema envia um e-mail para realizar a alteração da senha.
Pós-condição	O Administrador deve acessar o e-mail para confirmar a nova senha.
Cenário Alternativo	3a- O e-mail informado não possui cadastro.
Inclusão	Não possui.
Extensão	UC 010 – Acessar Sistema
Caso de Uso – Confirmar Senha	
ID	UC 012
Descrição	Este caso de uso tem como objetivo confirmar a alteração da senha.
Ator Primário	Administrador.
Pré-condição	Ter solicitado a alteração da senha.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador acessa o link para alteração da senha. 2. O sistema abre um formulário com os campos para a alteração da senha. 3. O administrador preenche os campos nova senha e confirmação da nova senha. 4. O sistema verifica se a nova senha e a confirmação são iguais. 5. O sistema valida a nova senha. 6. O sistema altera a senha.
Pós-condição	Não possui.
Cenário Alternativo	3a Sistema não valida a senha . 5a Sistema não valida a confirmação da senha.
Inclusão	UC 011 – Alterar senha
Extensão	Não possui.
Caso de Uso – Fazer Login	
ID	UC 013
Descrição	Este caso de uso tem como objetivo fazer login no sistema.
Ator Primário	Administrador.

Pré-condição	Possuir uma conta cadastrada.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador abre o sistema. 2. O sistema solicita os dados de login. 3. O administrador preenche os campos usuário e senha. 4. O sistema verifica se o usuário e senha estão corretos. 5. O sistema valida usuário e senha. 6. O sistema realiza o login.
Pós-condição	Não possui.
Cenário Alternativo	5a- O usuário ou a senha informada está incorreto.
Inclusão	UC 012 – Confirmar Senha
Extensão	UC 010 – Acessar Sistema
Caso de Uso – Visualizar Produtos	
ID	UC 014
Descrição	Este caso de uso tem como objetivo visualizar os produtos cadastrados.
Ator Primário	Administrador.
Pré-condição	Não possui.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção produtos. 2. O sistema mostra os produtos cadastrados. 3. O sistema disponibiliza a opção de alterar produtos ou cadastrar novos produtos.
Pós-condição	Não possui.
Cenário Alternativo	a* - O administrador pode clicar do botão de sair
Inclusão	Não possui.
Extensão	UC 013 - Fazer Login
Caso de Uso – Alterar Detalhes do Produto	
ID	UC 015
Descrição	Este caso de uso tem como objetivo alterar as informações e a presença do produto no cardápio.
Ator Primário	Administrador.
Pré-condição	Possuir produtos cadastrados.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona o ícone de alterar detalhes do produto. 2. O sistema abre os detalhes do produto. 3. O administrador seleciona qual informação deseja alterar (nome, imagem, preço ou descrição). 4. O sistema informa as alterações que serão feitas e pede confirmação. 5. O administrador confirma. 6. O sistema salva as alterações.
Pós-condição	Não possui.
Cenário Alternativo	<p>1a – O administrador seleciona o ícone de desativar o produto do cardápio.</p> <p>1b – O administrador seleciona o ícone para ativar o produto.</p> <p>2a – O sistema pergunta se realmente é para desativar o produto.</p> <p>2b – O sistema pergunta se realmente é para ativar o produto.</p>

	3a – O administrador clica do botão de voltar 5a – O administrador cancela a desativação do produto. 5b – O administrador cancela a ativação do produto.
Inclusão	Não possui.
Extensão	UC 014 – Visualizar Produtos
Caso de Uso – Cadastrar Produto	
ID	UC 0016
Descrição	Este caso de uso tem como objetivo cadastrar um novo produto.
Ator Primário	Administrador.
Pré-condição	Não possui.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção cadastrar produto. 2. O sistema carrega o formulário para cadastro de produto. 3. O administrador informa o nome do produto. 4. O administrador preenche as informações do produto. 5. O sistema mostra as informações do cadastro e pede confirmação. 6. O administrador confirma. 7. O sistema cadastra o novo produto.
Pós-condição	Não possui.
Cenário Alternativo	6a- O clica do botão de voltar.
Inclusão	Não possui.
Extensão	UC 014 – Visualizar Produtos
Caso de Uso – Visualizar Pedidos	
ID	UC 017
Descrição	Este caso de uso tem como objetivo visualizar os pedidos.
Ator Primário	Administrador.
Pré-condição	Não possui.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção pedidos. 2. O sistema disponibiliza as opções visualizar novos pedidos, visualizar pedidos em aberto e visualizar pedidos finalizados.
Pós-condição	Não possui.
Cenário Alternativo	a* - O administrador pode clicar do botão de sair
Inclusão	Não possui.
Extensão	UC 013 – Fazer Login
Caso de Uso – Visualizar Novos Pedidos	
ID	UC 018
Descrição	Este caso de uso tem como objetivo mostrar os novos pedidos.
Ator Primário	Administrador.
Pré-condição	Não possui.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção visualizar novos produtos. 2. O sistema carrega a aba novos pedidos. 3. O sistema informa a opção “confirmar pedido?” Para cada pedido.

Pós-condição	Não possui.
Cenário Alternativo	a* - O administrador pode clicar do botão de sair
Inclusão	Não possui.
Extensão	UC 017 – Visualizar Pedidos
Caso de Uso – Confirmar Pedido	
ID	UC 019
Descrição	Este caso de uso tem como objetivo informar se o pedido será confirmado ou não.
Ator Primário	Administrador.
Pré-condição	Ter recebido um novo pedido.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção “deseja confirmar do pedido?”. 2. O sistema informa os detalhes do pedido (produto, quantidade, valor, data de retirada e dados do cliente). 3. O sistema disponibiliza a opção “confirmar pedido?”. 4. O sistema disponibiliza as opções “Sim”, “Não” e “Cancelar”. 5. O administrador confirma o pedido. 6. O sistema o move o pedido para a aba pendentes.
Pós-condição	Não Possui.
Cenário Alternativo	<p>5a- O administrador não confirma o pedido.</p> <p>5b – O administrador clica em “Cancelar”</p> <p>6a – O sistema exclui o pedido</p>
Inclusão	UC 018 – Visualizar Novos Pedidos
Extensão	Não possui.
Caso de Uso – Justificar Cliente	
ID	UC 020
Descrição	Este caso de uso tem como objetivo lembrar ao administrador de justificar ao cliente caso o pedido seja recusado
Ator Primário	Administrador.
Pré-condição	Ter recusado um pedido.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção “Não” na confirmação do pedido. 2. O sistema mostra uma mensagem requisitando que o administrador justifique ao cliente o motivo de não ser aceito. 3. O sistema exclui o pedido. 4. O administrador envia mensagem ao cliente justificando o motivo do pedido não ser aceito por meio do WhatsApp.
Pós-condição	Não possui.
Cenário Alternativo	Não possui.
Inclusão	Não possui.
Extensão	UC 019 – Confirmar Pedido
Caso de Uso – Visualizar Pedidos Pendentes	
ID	UC 021
Descrição	Este caso de uso tem como objetivo visualizar os pedidos já confirmados, mas não finalizados.
Ator Primário	Administrador.
Pré-condição	Ter aceitado algum pedido.

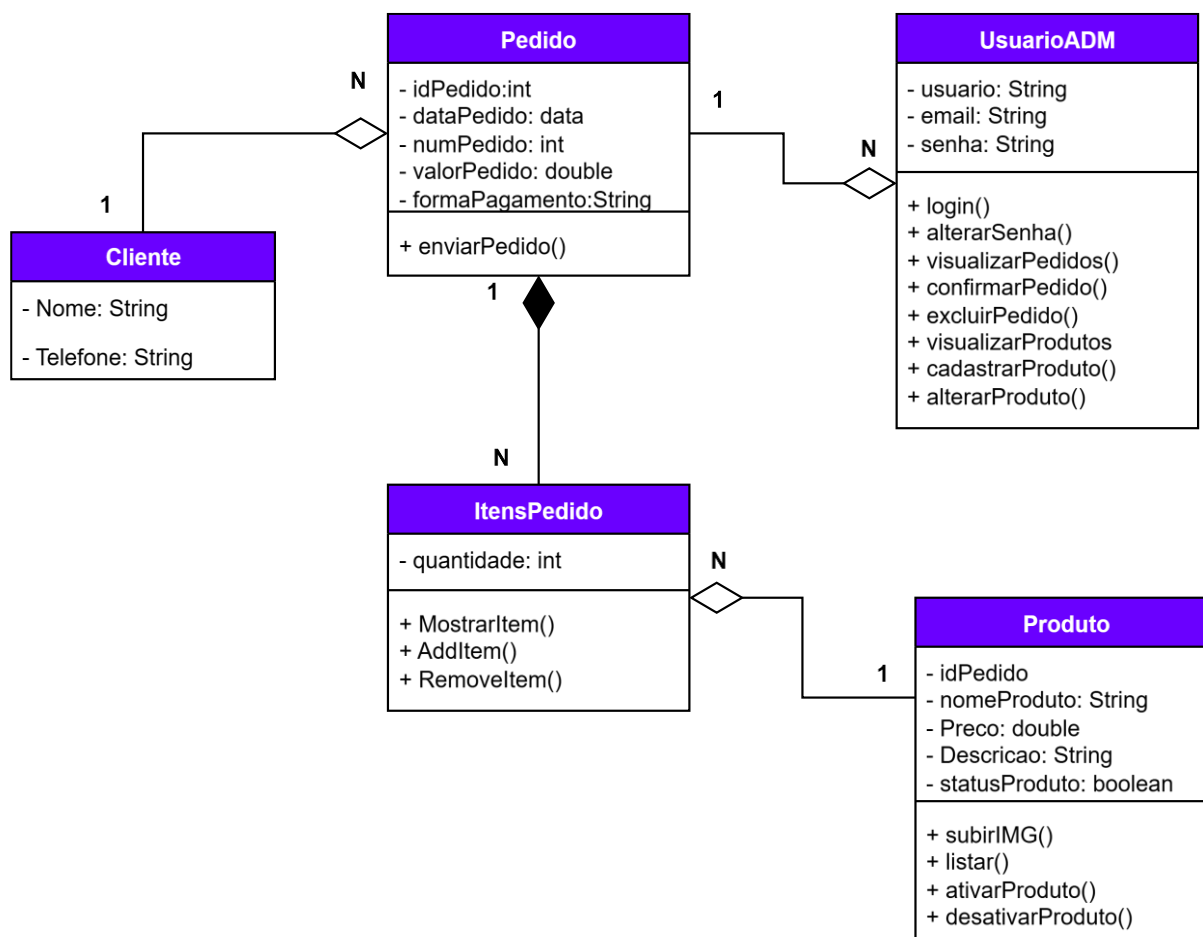
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a aba visualizar pendentes. 2. O sistema carrega os pedidos pendentes. 3. O sistema informa os detalhes do pedido (quantidade, data de entrega e dados do cliente). 4. O sistema disponibiliza a opção de finalizar o pedido.
Pós-condição	Não possui.
Cenário Alternativo	a* - O administrador pode clicar do botão de sair
Inclusão	Não possui.
Extensão	UC 017 – Visualizar Pedidos
Caso de Uso – Finalizar Pedido	
ID	UC 022
Descrição	Este caso de uso tem como objetivo finalizar os pedidos prontos.
Ator Primário	Administrador.
Pré-condição	Nenhuma.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção finalizar pedido. 2. O sistema informa uma mensagem perguntando se realmente deseja finalizar o pedido. 3. O administrador confirma. 4. O sistema finaliza o pedido, movendo-o para aba de finalizados.
Pós-condição	Não possui.
Cenário Alternativo	3a- O administrador não confirma a finalização do pedido.
Inclusão	UC 021 – Visualizar Pedidos Pendentes
Extensão	Não possui.
Caso de Uso – Visualizar Pedidos Finalizados	
ID	UC 023
Descrição	Este caso de uso tem como objetivo visualizar os pedidos já finalizados.
Ator Primário	Administrador.
Pré-condição	Não possui.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a aba visualizar pedidos finalizados. 2. O sistema mostra os pedidos finalizados. 3. O sistema disponibiliza a opção de excluir pedido.
Pós-condição	Não possui.
Cenário Alternativo	a* - O administrador pode clicar do botão de sair
Inclusão	Não possui.
Extensão	UC 017 – Visualizar Pedidos. UC 022 – Finalizar Pedido.
Caso de Uso – Excluir Pedido	
ID	UC 024
Descrição	Este caso de uso tem como objetivo excluir os pedidos já finalizados.
Ator Primário	Administrador.
Pré-condição	Ter pedidos finalizados.
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o administrador seleciona a opção excluir pedido.

	<ol style="list-style-type: none"> 2. O sistema informa uma mensagem perguntando se realmente deseja excluir o pedido. 3. O administrador confirma a exclusão. 4. O sistema exclui o pedido do sistema.
Pós-condição	Não possui.
Cenário Alternativo	3a- O administrador não confirma a exclusão.
Inclusão	UC 023 – Visualizar Pedidos Finalizados
Extensão	Não possui.

Fonte: os autores

A Figura 5 ilustra o diagrama de classes, detalhando a estrutura do sistema, suas entidades, atributos, métodos e as relações entre elas, facilitando o entendimento da arquitetura e do design do software.

Figura 5 – Diagrama de Classe



Fonte: os autores

Essas classes representam as principais entidades e funcionalidades de um sistema de gestão de pedidos, onde o administrador (UsuarioADM) pode gerenciar produtos e pedidos, e os clientes podem realizar pedidos de produtos, os quais são detalhados por meio de itens com suas respectivas quantidades e informações.

3.6 Diagrama de Atividades

Um Diagrama de Atividades é um tipo de diagrama de modelagem que representa o fluxo de atividades e decisões dentro de um processo ou sistema, permitindo visualizar a sequência lógica e o comportamento dos processos. Como aponta Parker (2024) “Diagrama de Atividades é basicamente um fluxograma para representar o fluxo de uma atividade para outra atividade. A atividade pode ser descrita como uma operação do sistema.

A importância de um Diagrama de Atividades está em sua capacidade de fornecer uma visão clara e detalhada do fluxo de trabalho, identificando todas as etapas e pontos de decisão de forma visual. Isso facilita o entendimento dos processos, tanto para desenvolvedores quanto para stakeholders, garantindo que todos compreendam o comportamento do sistema. “O objetivo básico dos diagramas de atividades é capturar o comportamento dinâmico do sistema. Também é chamado de fluxograma orientado a objetos” Parker (2024).

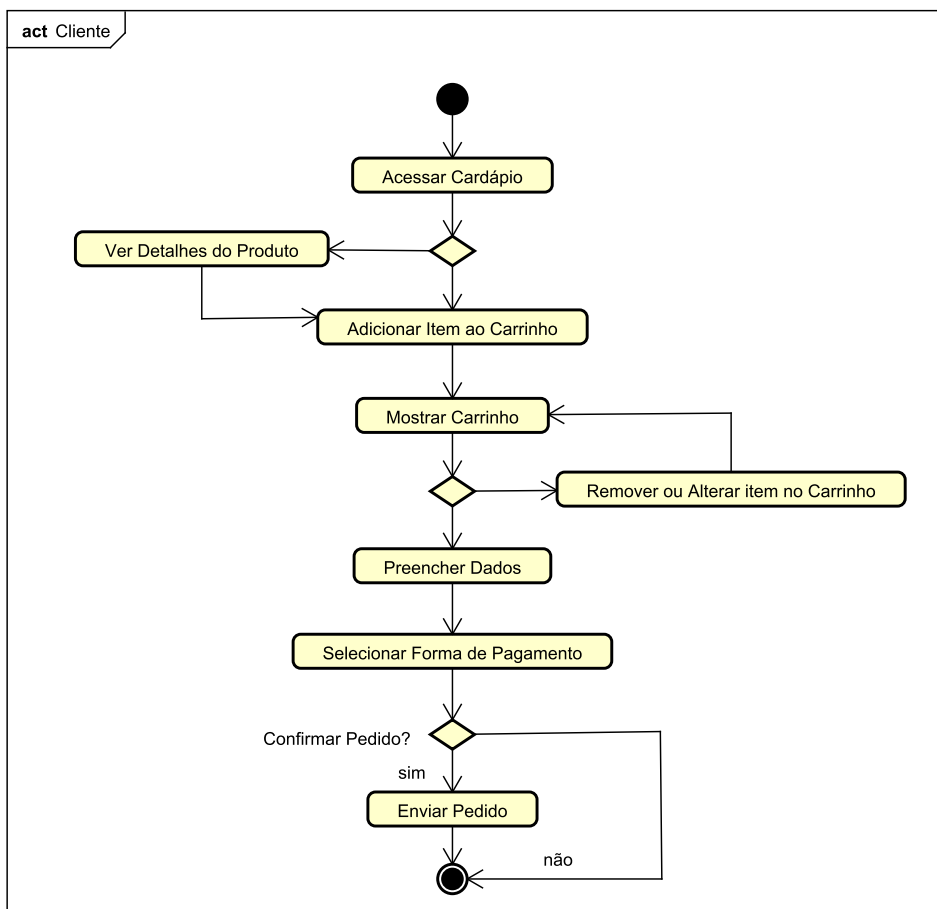
3.6.1 Diagramas de Atividades

As Figuras 6 a 8 mostram os Diagramas de Atividade criados.

O fluxo do Cliente começa com o ele acessando o cardápio; o cliente pode optar tanto por ver os detalhes do produto, quanto por já adicioná-lo diretamente no carrinho; o usuário acessa o carrinho onde ele pode alterar ou remover os itens; confirmando, o cliente preencherá o formulário de dados e selecionará a forma de pagamento; após continuar o cliente confirmará o pedido encerrando o fluxo.

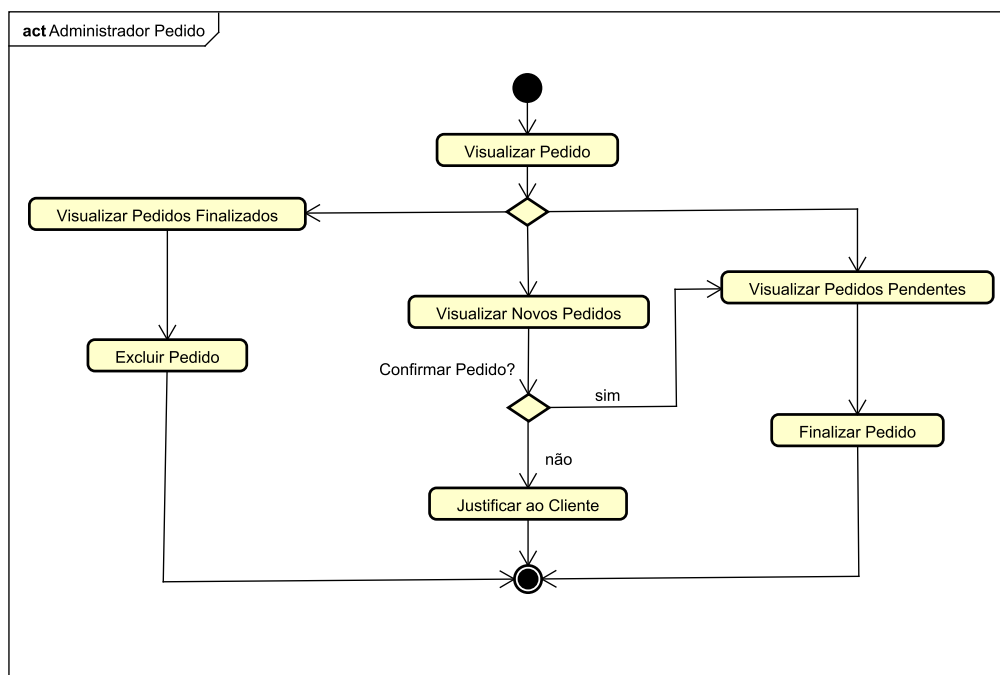
O fluxo da Gestão de Pedidos se inicia com a visualização dos pedidos pelo administrador; o administrador pode visualizar os pedidos finalizados e optar por excluí-los, visualizar pedidos em aberto para iniciar a preparação, ou visualizar novos pedidos para confirmá-los ou com o sistema lembrando-o de justificar ao cliente se necessário; o fluxo segue para a preparação do pedido e, finalmente, ser finalizado.

Figura 6 – Diagrama de Atividade Cliente



Fonte: os autores

Figura 7 – Diagrama de Atividade Gestão de Pedidos

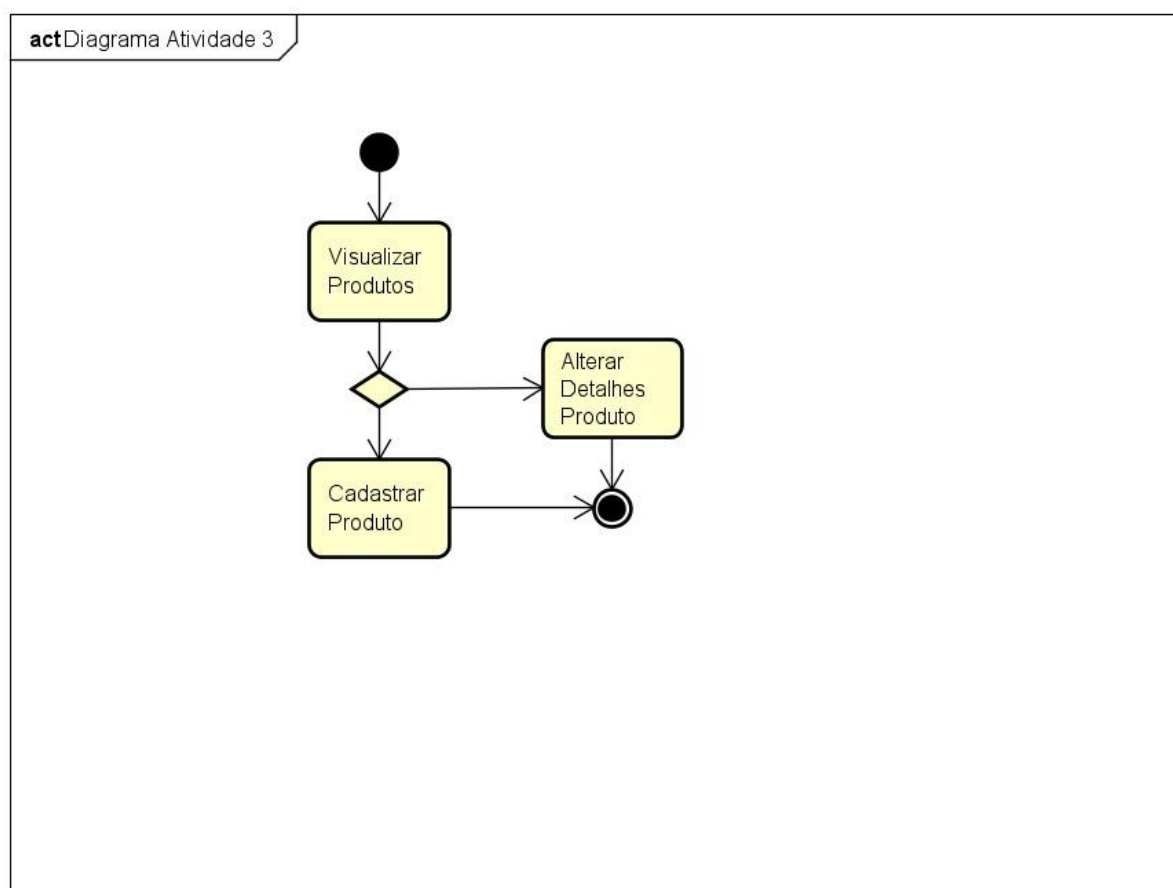


Fonte: Os autores

No fluxo de Gestão de Produtos o administrador tem a opção de visualizar, cadastrar ou alterar os detalhes de um produto (incluindo seu estado cardápio); após a visualização, ele pode decidir alterar as informações ou cadastrar novos produtos, garantindo que o catálogo esteja sempre atualizado e refletindo as mudanças necessárias.

Os diagramas de atividade feitos proporcionam uma visão detalhada das interações e decisões que o sistema deve realizar para garantir a operação eficiente do processo de pedidos e produtos, atendendo às necessidades de administração interna e interação com o cliente.

Figura 8 – Diagrama de Atividade Gestão de Produtos



Fonte: os autores

3.7 Diagrama de Estados

Um Diagrama de Estados, ou Diagrama de Máquina de Estados, é uma representação gráfica que ilustra as diversas condições que um objeto ou sistema

pode experimentar durante sua existência. Ele demonstra a conexão entre esses estados através de transições, provocadas por eventos ou circunstâncias específicas.

Nas palavras de Parker (2024b) “Diagrama de Estado são usados para capturar o comportamento de um sistema de software. Os diagramas de máquina de estados UML podem ser usados para modelar o comportamento de uma classe, um subsistema, um pacote ou até mesmo um sistema inteiro.

As Figuras 9 a 11 ilustram os diagramas de estado, apresentando as principais transições e estados relevantes para o sistema em análise.

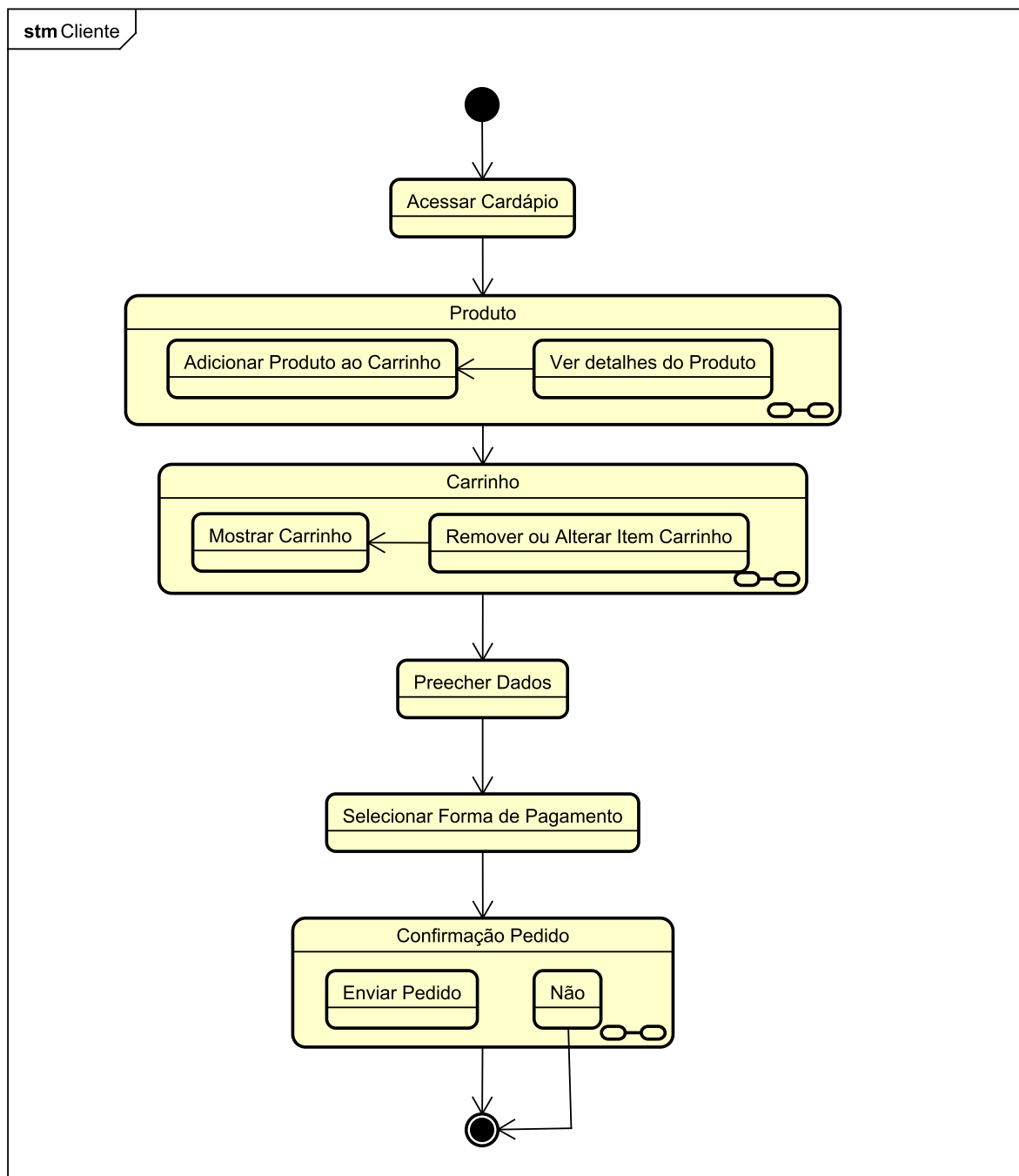
Cada evento e condição no sistema assegura a correta execução do fluxo de trabalho, prevenindo inconsistências e proporcionando uma compreensão precisa de como cada objeto deve se comportar em variados contextos. Esta estrutura simplifica o processo de criação e teste, assegurando que o sistema se comporte conforme o esperado em cada condição e mudança possível.

3.8 Diagrama de Sequência

Um Diagrama de Sequência é um tipo de diagrama que ilustra a interação entre objetos em um sistema de forma temporal. Ele foca em representar como os objetos se comunicam por meio de mensagens ao longo do tempo, ajudando a entender o fluxo de execução de um processo específico. Na visão de Creately (2021) “Em palavras mais simples, um diagrama de sequência mostra diferentes partes de um sistema trabalhando em uma ‘sequência’ para se fazer algo.

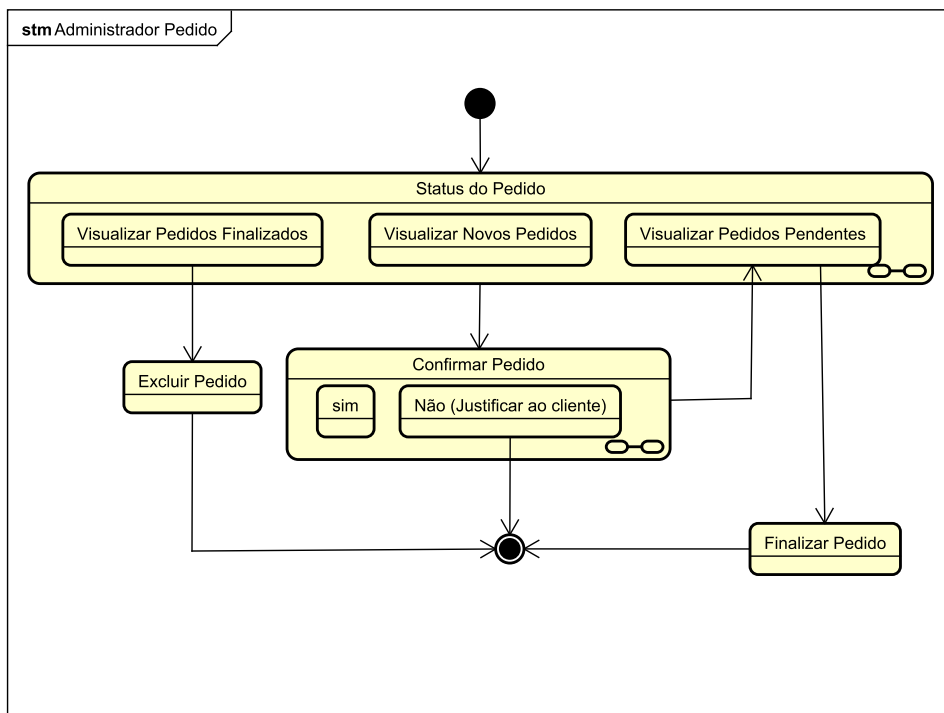
O valor de um Diagrama de Sequência reside em sua capacidade de modelar e visualizar a dinâmica de um sistema, proporcionando uma visão clara de como os objetos e componentes interagem durante um processo. Eles possibilitam que todos os envolvidos no projeto tenham uma compreensão alinhada das operações do sistema, ajudando na validação dos requisitos e na redução de erros e retrabalhos na fase de desenvolvimento.

Figura 9 – Diagrama de Estado Carrinho de Compras



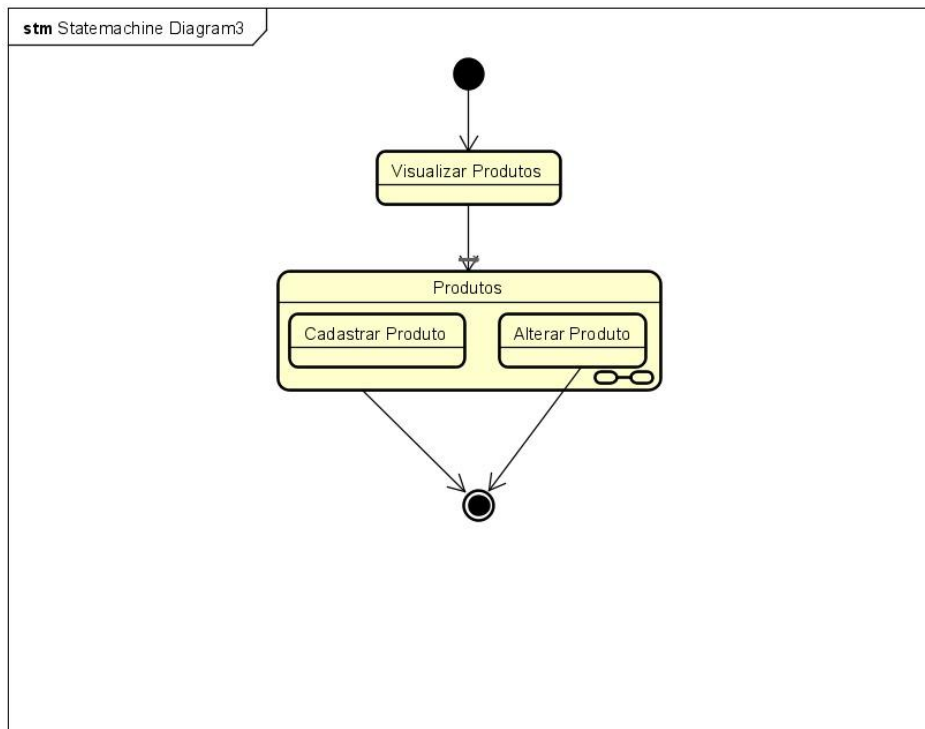
Fonte: os autores

Figura 10 – Diagrama de Estado Gestão de Pedidos



Fonte: os autores

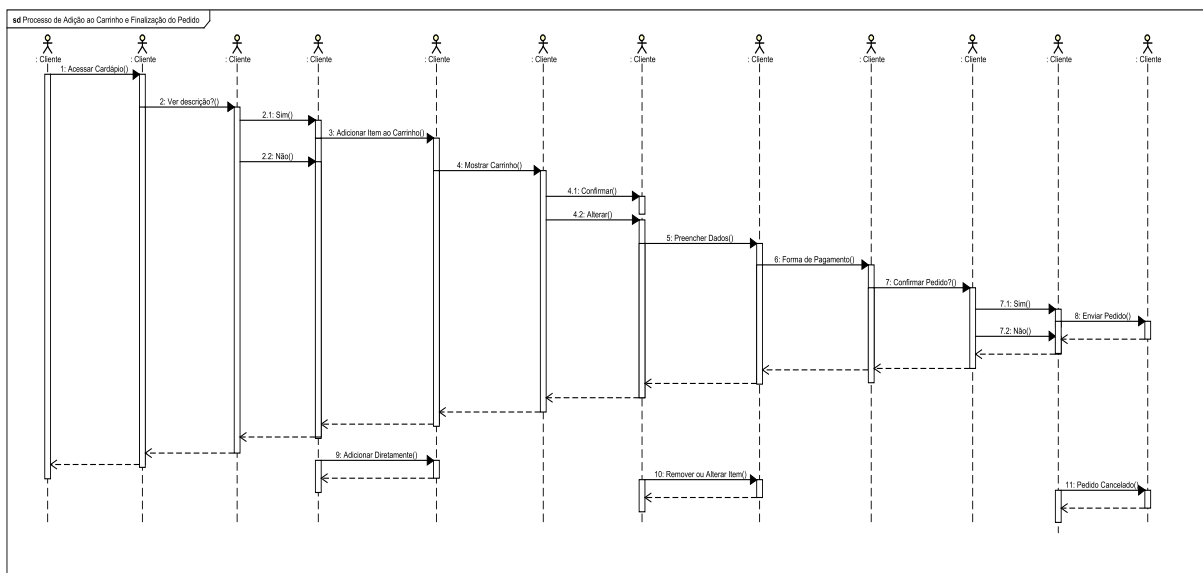
Figura 11 – Diagrama de Estado Gestão de Produtos



Fonte: os autores

As Figuras 12 a 14 ilustram os diagramas de sequência, detalhando as interações entre o cliente e o sistema.

Figura 12 – Processo de Adição ao Carrinho e Finalização do Pedido



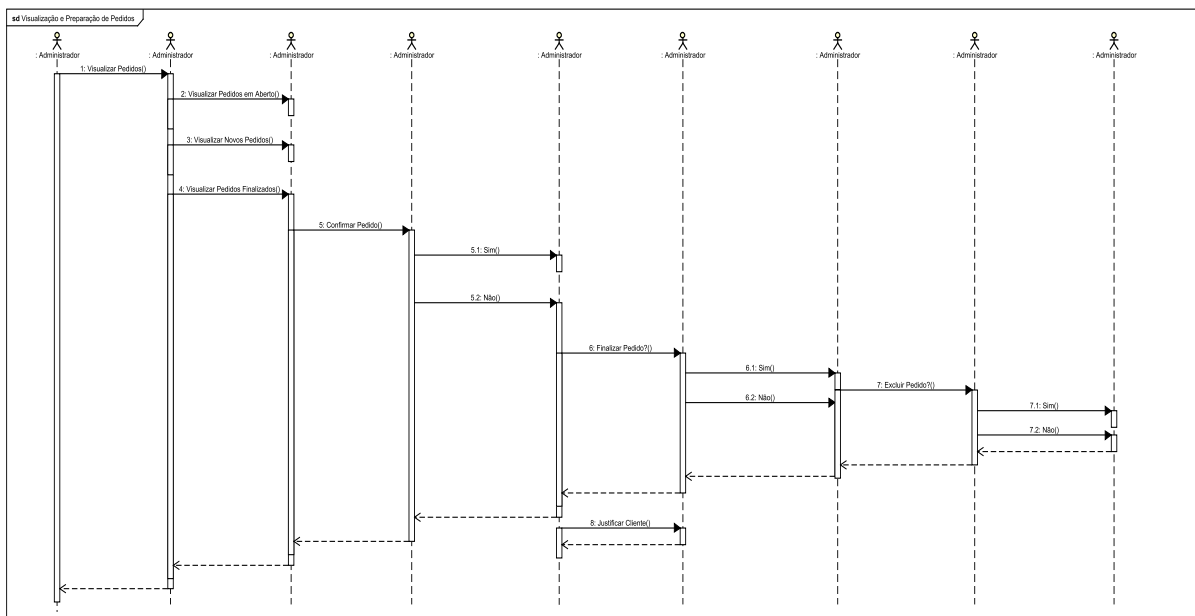
Fonte: os autores

Esse diagrama ilustra o processo de um cliente ao adicionar itens ao carrinho e finalizar uma compra na plataforma da confeitaria. O fluxo inicia quando o cliente acessa o cardápio (1), onde pode visualizar a descrição de um produto (2) ou adicioná-lo ao carrinho (3). Em seguida, o cliente acessa o carrinho (4), onde tem a possibilidade de remover itens (4.1) ou alterar as quantidades (4.2).

Após revisar o pedido, o cliente segue para o preenchimento dos dados necessários (5) e a escolha da forma de pagamento (6). Na etapa seguinte, deve confirmar o pedido (7.1). Uma vez confirmado, o sistema envia o pedido (8).

Este diagrama representa o fluxo de trabalho do administrador no sistema da confeitaria para o gerenciamento de pedidos. O administrador pode visualizar os produtos (1), os pedidos em aberto (2), os novos pedidos (3) e os pedidos finalizados (4). Após revisar essas informações, ele pode confirmar um pedido (5), escolhendo entre prosseguir (5.1 – Sim) ou recusar (5.2 – Não).

Figura 13 – Visualização e Preparação de Pedidos

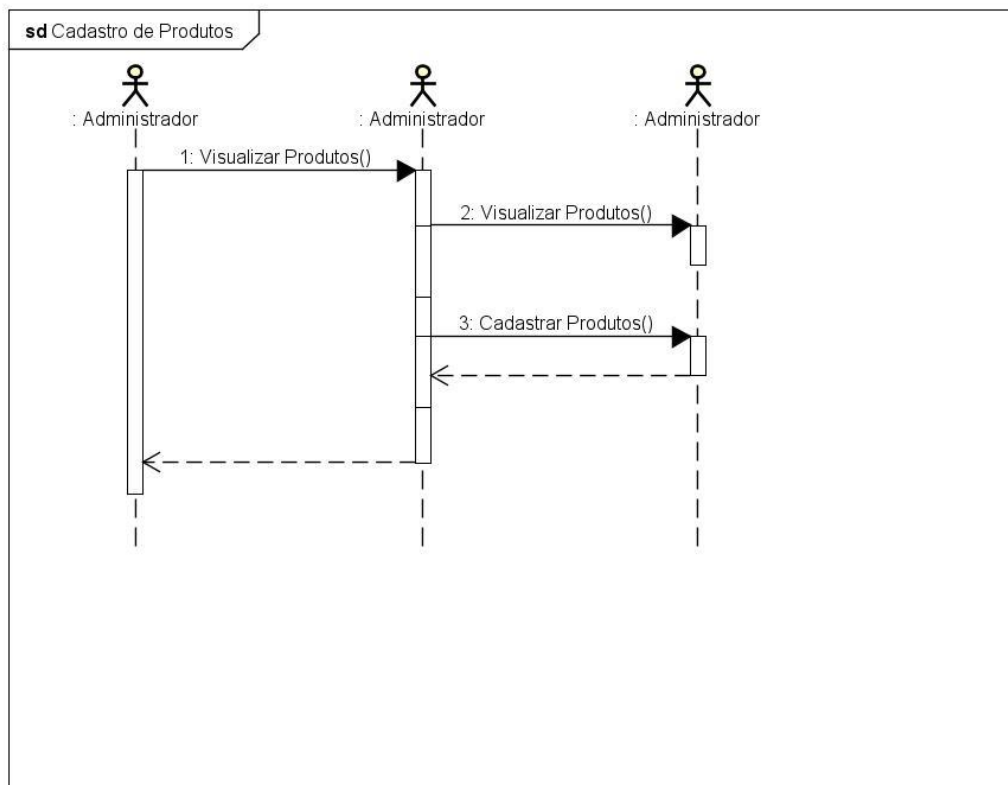


Fonte: os autores

Quando o pedido é confirmado, o administrador segue para a preparação e finalização (6). Caso necessário, ele também tem a opção de excluir o pedido (7).

Esse fluxo garante que o administrador mantenha controle total sobre o estado dos pedidos, assegurando organização e eficiência em toda a operação.

Figura 14 – Cadastro de Produtos



Fonte: os autores

Esse diagrama descreve o processo de cadastro de novos produtos no sistema pelo administrador. Inicialmente, o administrador visualiza os produtos já cadastrados (1), com a opção de visualizar novamente caso seja necessário (2). Em seguida, ele realiza o cadastro de novos produtos (3), completando o processo. Esse fluxo simples permite que o administrador adicione itens ao catálogo da confeitaria, garantindo que o cardápio esteja atualizado para os clientes.

3.9 Matriz de Rastreabilidade

A Matriz de Rastreabilidade é uma ferramenta essencial em projetos de desenvolvimento de sistemas, usada para garantir que os requisitos especificados estejam sendo atendidos em todas as etapas do projeto. Ela auxilia no processo de validação e verificação, tornando o projeto mais transparente e permitindo uma melhor comunicação entre as partes envolvidas. Essa visibilidade é fundamental para a gestão de projetos, pois reduz o risco de falhas de requisitos não atendidos, o que pode acarretar retrabalhos e custos adicionais

Como aponta Espinha (2020), “A matriz de rastreabilidade ajuda a fazer um bom gerenciamento e controle das mudanças no escopo e é essencial para que o projeto seja bem-sucedido e atenda às expectativas dos *stakeholders*”.

O Quadro 6 apresenta a Matriz de Rastreabilidade, estabelecendo a relação entre os requisitos não funcionais do sistema e as regras de negócio.

Quadro 6 – Matriz de Rastreabilidade

ID da Regra de Negócio	Descrição da Regra de Negócio	ID do Requisito Funcional	Nome do Requisito Funcional
RN-RF003.1	A quantidade total do produto deve ser, no mínimo, 500 gramas.	RF003	Adicionar item ao Carrinho
RN-RF003.2	A quantidade total do produto deve ser, no máximo, 50 quilogramas.	RF003	Adicionar item ao Carrinho
RN-RF004.1	A quantidade total do pedido deve ser, no mínimo, 500 gramas.	RF004	Mostrar Carrinho
RN-RF004.2	A quantidade total do pedido deve ser, no máximo, 50 quilogramas.	RF004	Mostrar Carrinho
RN-RF005.1	A quantidade total do produto deve ser, no mínimo, 500 gramas.	RF005	Remover ou Alterar item do Carrinho
RN-RF005.2	A quantidade total do produto deve ser, no máximo, 50 quilogramas.	RF005	Remover ou Alterar item do Carrinho
RN-RF006.1	Se a quantidade total do pedido for inferior a 4 kg, a retirada deve ser em um dia. Entre 5 e 10 kg, dois dias. Entre 10 e 20 kg, uma semana. Entre 20 e 50 kg, 15 dias.	RF006	Enviar Pedido

RN-RF006.2	O horário de retirada deve ser de segunda-feira a sexta-feira, das 9h às 17h, e no sábado, das 9h às 13h.	RF006	Enviar Pedido
RN-RF006.3	A contagem de dias para retirada deve excluir sábados e domingos.	RF006	Enviar Pedido

Fonte: os autores

Com essa matriz, é possível acompanhar o progresso e garantir que nenhuma regra de negócio seja negligenciada minimizando riscos e retrabalho.

4 Ferramentas e Métodos

As ferramentas utilizadas para o desenvolvimento do projeto foram escolhidas com base em sua eficiência, escalabilidade e facilidade de uso. Além de serem amplamente adotadas no mercado, essas ferramentas possuem documentação robusta, tutoriais acessíveis e uma comunidade ativa que facilita o aprendizado e a implementação. A escolha também levou em conta a experiência prévia da equipe e a disponibilidade de versões gratuitas ou com licenças acessíveis para estudantes. Foram adotadas as ferramentas descritas no Quadro 7.

Quadro 7 – Ferramentas utilizadas

Ferramenta	Descrição
Microsoft Word	Office 365 (Licença Estudante), licença acadêmica gratuita; foi usado para registrar Missão, Visão e Valores (Apêndice 2), o TAP (Termo de Abertura do Projeto), a Documentação de Requisitos, a Matriz de Rastreabilidade, a Documentação de Portabilidade e a Proposta Comercial. MS Word é uma ferramenta robusta para a criação e formatação de documentos, facilitando a edição colaborativa, a revisão e o compartilhamento de arquivos
Google Forms	licença gratuita (inclusa no Google Workspace); utilizado para a Elicitação de Requisitos, permitindo coletar respostas de forma rápida e organizada dos stakeholders; permite criar questionários facilmente e obter feedback em tempo real, com funcionalidades de análise automática de respostas
Paint	licença gratuita (incluída nativamente no Windows); pré-instalado no sistema operacional Windows; usado para a criação e edição do Business Model Canvas (BMC); permite edição direta de imagens e adição de texto de forma simples e intuitiva, atendendo às necessidades básicas de customização do canvas
Canva	licença gratuita com opções pagas; foi usado para desenvolver a Análise SWOT; excelente para criar gráficos e visualizações atraentes de forma rápida, sem necessidade de conhecimento avançado em design
Bpmn.io	utilizada a versão Web-based; open source; para criar o diagrama BPMN; ferramenta específica para modelagem de processos com BPMN, sendo intuitiva e gratuita
Astah	utilizada versão v.41 (Licença Estudante); acadêmica gratuita; para criar os diagramas de Caso de Uso, de Atividade, de Máquina de Estado e de Sequência; uma das melhores ferramentas para modelagem UML, com uma interface amigável e suporte para múltiplos diagramas
Draw.io	licença open source; usada para criar o Diagrama de Classes; simples, intuitiva e gratuita, ideal para diagramas de arquitetura de software
Microsoft Excel	MS Office 365 (Licença Estudante) acadêmica gratuita; usado para desenvolver as Métricas a partir de um modelo pré pronto

Ferramenta	Descrição
Figma	licença gratuita com opções pagas; utilizado para a prototipagem de telas; permite a criação de protótipos interativos e colaborativos em tempo real, facilitando a validação de design com stakeholders
Node.js	licença open source (MIT License); serviu como ambiente de execução para o backend da aplicação, possibilitando a implementação das rotas da API; foi escolhido por ser multiplataforma, amplamente usado no mercado e também porque a equipe teve contato inicial com Node.js em sala de aula, facilitando sua aplicação prática no projeto
TypeScript	licença open source (Apache License 2.0); utilizado para o desenvolvimento do código do frontend e do backend, adicionando tipagem estática ao JavaScript; a tipagem do TypeScript ajuda a reduzir erros em tempo de desenvolvimento, tornando o código mais seguro e fácil de manter, além de oferecer melhor suporte em IDEs
Tailwind CSS	licença open source (MIT License); aplicado para a estilização da interface da aplicação; foi escolhido por ser mais simples e produtivo que escrever CSS puro, permitindo padronização visual rápida e consistente
Next.js	licença open source (MIT License); framework principal para o desenvolvimento do frontend e da lógica de rotas; foi escolhido por ser moderno, robusto e por já ter sido estudado pela equipe em sala de aula, além de oferecer suporte nativo a Server-Side Rendering (SSR) e Static Site Generation (SSG)
MySQL	licença open source (GPL 2.0); Sistema Gerenciador de Bancos de Dados Relacionais (SGBDR) utilizado para armazenar os dados de clientes, produtos e pedidos; foi escolhido por ser amplamente adotado, gratuito e já conhecido pela equipe por meio das aulas, facilitando a implementação prática
Prisma ORM	licença open source (Apache License 2.0); ORM utilizado para realizar o mapeamento dos dados do banco MySQL e facilitar a manipulação das tabelas no código; foi escolhido por simplificar a comunicação entre a aplicação e o banco de dados, reduzindo a complexidade da escrita de queries SQL
Navegadores Chromium e Extensões	Google Chrome e Microsoft Edge foram utilizados para acessar e testar a aplicação web durante o desenvolvimento, garantindo compatibilidade com diferentes ambientes e permitindo a inspeção de elementos, depuração de código e análise de desempenho; foram instaladas as extensões U-Eyes, Mobile Device Simulator que simula interface de dispositivos móveis, como iPhone 13 Pro Max, e ferramentas de desenvolvedor nativas do Chromium; a escolha pelos navegadores Chromium se deu devido à ampla adoção no mercado, compatibilidade com padrões web modernos e suporte a ferramentas de desenvolvimento. As extensões foram empregadas para simular dispositivos móveis e testar responsividade, permitindo que a equipe validasse a experiência do usuário sem a necessidade de múltiplos dispositivos físicos

Fonte: os autores

O desenvolvimento da interface para o projeto foi realizado utilizando Figma, que permitiu a prototipagem rápida de telas interativas e validou a experiência do usuário com o cliente antes da implementação final. Segundo Theis *et al.* (2021, p. 4):

Desenhar a ideia (ou ideação) é a estratégia mais comum nos procedimentos de design, uma prototipagem em forma de elementos simples, esquemas, diagramas e gráficos para que todos os envolvidos no processo possam visualizar e compartilhar os achados.

A prototipagem pode variar desde esboços simples (*wireframes*) até protótipos interativos de alta fidelidade que simulam a experiência de uso do software. Ferramentas são amplamente utilizadas para esse processo, permitindo que

desenvolvedores, designers e *stakeholders* visualizem e validem a interface antes da codificação.

Os métodos empregados incluíram a abordagem Centrada no Usuário, utilizada para entender as necessidades do cliente e propor soluções inovadoras para melhorar a experiência do usuário; e definição do estilo da interface, com foco no layout, cores, tipografia e elementos visuais para garantir uma interface agradável e intuitiva.

O uso dessas ferramentas e métodos garantiu um fluxo de trabalho eficiente, colaborativo e alinhado com as expectativas do cliente, resultando em um sistema bem documentado e pronto para implementação.

5 Desenvolvimento

O desenvolvimento de sistemas é um processo estruturado que abrange concepção, projeto, implementação, testes e manutenção, de modo a converter requisitos em artefatos de software funcionais e confiáveis. Segundo a IBM, “o desenvolvimento de software se refere a um conjunto de atividades de ciências da computação dedicadas ao processo de criação, design, implementação e suporte de software” (IBM, s.d.)

5.1 Sistema Gerenciador de Bancos de Dados Relacionais (SGBDR)

Para o *schema* do banco de dados foi utilizado Prisma ORM com MySQL. A configuração emprega um *generator* para Prisma Client e um *datasource* conectado por variável de ambiente. Foram definidas, na modelagem, as entidades: clientes, usuários administrativos, produtos e pedidos.

A entidade `UsuarioADM` possui atributos únicos para usuário e e-mail. A entidade `Cliente` mantém uma relação *one-to-many* com pedidos. A entidade `Produto` inclui um campo booleano `statusProduto` para controle de disponibilidade, sem exclusão física. Sua relação com pedidos é feita através da tabela `ItemPedido`.

```

Produto {
  id          Int          @id @default(autoincrement())
  nomeProduto String
  preco       Float
  descricao   String
  imagem      String
  itens       ItemPedido[]
  statusProduto Boolean    @default(true) // true = ativo, false = de-
sativado
}

```

O fluxo de pedidos utiliza uma enumeração com os estados: NOVO, PENDENTE, ENTREGUE e CANCELADO. A entidade Pedido registra dados da transação, como data, valor total, forma de pagamento, e seus relacionamentos com o cliente e os itens. A relação *many-to-many* entre pedidos e produtos é implementada pela entidade ItemPedido, que armazena a quantidade de cada produto.

```
enum StatusPedido {
  NOVO
  PENDENTE
  ENTREGUE
  CANCELADO
}

model Pedido {
  id          Int          @id @default(autoincrement())
  dataPedido  DateTime
  valorPedido Float
  clienteId   Int
  cliente     Cliente     @relation(fields: [clienteId], references:
[id])
  itens       ItemPedido[]
  status      StatusPedido @default(NOVO)
  formaPagamento String
}
```

Os relacionamentos entre entidades são garantidos através de chaves estrangeiras e restrições de integridade referencial. A tabela ItemPedido atua como pivô na relação *many-to-many* entre Pedido e Produto, contendo referências para ambas as entidades e a quantidade de cada item. Todas as relações utilizam as diretivas de relação do Prisma, assegurando consistência nas operações de banco de dados.

```
model ItemPedido {
  id          Int          @id @default(autoincrement())
  pedidoId    Int
  produtoId   Int
  quantidade  Int
  pedido      Pedido      @relation(fields: [pedidoId], references: [id])
  produto     Produto     @relation(fields: [produtoId], references: [id])
}
```

5.2 Autenticação do Administrador

A interface de login usa React com TypeScript. O formulário gerencia email, senha e a opção de lembrar credenciais com useState. Ao ser submetido, faz uma

requisição POST para a API de autenticação usando fetch. Inclui tratamento de erros para feedback visual.

```
const [email, setEmail] = useState("");
const [senha, setSenha] = useState("");
const [lembrar, setLembrar] = useState(false);
const [erro, setErro] = useState("");

async function handleSubmit(event: React.FormEvent) {
  event.preventDefault();
  try {
    const response = await fetch("/api/login", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ email, senha, lembrar }),
      credentials: 'include'
    });
    // Tratamento da resposta
  } catch (error) {
    setErro("Erro de conexão com o servidor");
  }
}
```

A API de login valida as credenciais consultando o usuário por email via Prisma. A senha é comparada com o hash armazenado usando bcrypt. Após a validação, gera um token JWT com ID e email do usuário, definido em um cookie HTTPOnly com opções de segurança.

```
const usuario = await prisma.usuarioADM.findUnique({ where: { email } });
const senhaValida = await bcrypt.compare(senha, usuario.senha);
const token = await generateToken({ id: usuario.id, email: usuario.email });

response.cookies.set({
  name: "authToken",
  value: token,
  httpOnly: true,
  secure: process.env.NODE_ENV === "production",
  sameSite: "strict",
  maxAge: lembrar ? 30 * 24 * 60 * 60 : 3600,
});
```

O fluxo de recuperação de senha verifica o email cadastrado e gera um token com expiração de uma hora. Um link com token e ID do usuário é enviado por email. A página de redefinição valida o token e o ID antes de permitir a alteração. A nova senha é criptografada com bcrypt antes do armazenamento.

```
const token = await generateToken({ id: usuario.id, email: usuario.email });
```

```
const resetLink = `${process.env.BASE_URL}/adm/login/alterar-senha/${usuario.id}?token=${encodeURIComponent(token)}`;

// Na redefinição:
const decoded = decodeToken(token);
if (decoded.id !== userIdFromUrl) {
  throw new Error("ID do token não corresponde");
}
```

Um middleware intercepta requisições para rotas administrativas. Rotas de login e recuperação de senha são públicas; as demais exigem o cookie de autenticação. Para redefinição de senha, valida o token nos parâmetros da URL. O redirecionamento é feito via `NextResponse`.

```
export async function middleware(request: NextRequest) {
  const path = request.nextUrl.pathname;

  if (publicRoutes.some(route => {
    const regex = new RegExp(`^${route.replace('(.*)', '.*')}.$`);
    return regex.test(path);
  })) {
    return NextResponse.next();
  }

  if (!request.cookies.get("authToken")) {
    return NextResponse.redirect(new URL("/adm/login", request.url));
  }
}
```

O encerramento de sessão remove o cookie de autenticação usando dois métodos: a função `delete` e definição de `maxAge` como zero. A resposta segue um formato padronizado para sucesso e falha.

```
export async function POST() {
  const response = NextResponse.json({ message: "Logout realizado com sucesso" });
  response.cookies.delete("authToken");
  response.cookies.set({
    name: "authToken", value: "", httpOnly: true,
    secure: process.env.NODE_ENV === "production",
    maxAge: 0, path: "/",
  });
  return response;
}
```

Esta estrutura proporciona um sistema de autenticação completo, abrangendo desde a interface do usuário até a segurança das rotas, com tratamento adequado de credenciais e recuperação de acesso.

5.3 Produtos

A página de produtos ordena os itens priorizando os ativos. Usa os hooks `useState` e `useEffect` para buscar os dados da API e ordená-los pelo campo `statusProduto`. Produtos ativos aparecem primeiro.

```
// Função personalizada para ordenação de negócio
const ordenarProdutos = (lista: Produto[]) => {
  return [...lista].sort((a, b) => (b.statusProduto ? 1 : 0) - (a.status-
  Produto ? 1 : 0));
};

// Uso no useEffect para aplicar a ordenação
useEffect(() => {
  const fetchProdutos = async () => {
    const res = await fetch('/api/produtos');
    if (res.ok) {
      const data: Produto[] = await res.json();
      setProdutos(ordenarProdutos(data));
    }
  };
  fetchProdutos();
}, []);
```

O sistema de upload emprega `FormData` para envio. A API processa a imagem, salva na pasta `public/uploads` com um nome baseado em timestamp e armazena a URL no banco de dados.

```
// Processamento de upload de imagem na API
const bytes = await imagem.arrayBuffer();
const buffer = Buffer.from(bytes);

const uploadDir = path.join(process.cwd(), "public/uploads");
const imageName = `${Date.now()}-${imagem.name}`;
const imagePath = path.join(uploadDir, imageName);

await writeFile(imagePath, buffer);
const imagemPath = `/uploads/${imageName}`;
```

A edição mantém a imagem existente se nenhuma nova for enviada. Usa o operador `spread` e avaliação condicional para construir o objeto de atualização, incluindo a nova imagem apenas quando fornecida.

```
// Update condicional - só atualiza imagem se houver nova
const produtoAtualizado = await prisma.produto.update({
  where: { id: parseInt(params.id) },
  data: {
    nomeProduto,
    preco,
    descricao,
  },
});
```

```

        ...(imagemPath && { imagem: imagemPath }), // Spread condicional
    },
});

```

A alternância de status usa exclusão lógica. O sistema busca o estado atual, inverte o valor booleano de statusProduto e atualiza o registro. O produto permanece no banco, mas é ocultado da visualização.

```

// Exclusão lógica com busca prévia do estado atual
const produtoAtual = await prisma.produto.findUnique({ where: { id } });
if (!produtoAtual) {
    return NextResponse.json({ error: "Produto não encontrado" }, { status:
404 });
}

// Alterna o status (exclusão lógica)
const produtoAtualizado = await prisma.produto.update({
    where: { id },
    data: { statusProduto: !produtoAtual.statusProduto },
});

```

A aplicação usa TypeScript para definir interfaces. A interface Produto define a estrutura dos objetos com tipos específicos, indicando propriedades que aceitam valores nulos.

```

// Interface TypeScript para estrutura de dados
interface Produto {
    id: number;
    nomeProduto: string;
    preco: number;
    descricao: string | null;
    imagem: string;
    statusProduto: boolean;
}

```

Todas as operações da API incluem tratamento de erros. Validam dados de entrada, verificam a existência de recursos e gerenciam exceções. As respostas retornam códigos HTTP e mensagens adequadas.

```

// Estrutura padrão de tratamento de erros
try {
    // Operação principal com Prisma
    return NextResponse.json(produtoAtualizado, { status: 200 });
} catch (error) {
    console.error("Erro ao atualizar produto:", error);
    return NextResponse.json({ error: "Erro ao atualizar produto" }, { sta-
tus: 500 });
}

```

A interface gerencia estados para operações assíncronas, incluindo carregamento, confirmações modais e dados do formulário. Isso fornece *feedback* visual e impede ações simultâneas.

```
// Estados para gestão da interface
const [produtos, setProdutos] = useState<Produto[]>([]);
const [modalAberto, setModalAberto] = useState(false);
const [produtoSelecionado, setProdutoSelecionado] = useState<Produto |
null>(null);
const [loading, setLoading] = useState(true);
```

5.3 Pedidos

O módulo de pedidos foi organizado em três seções distintas baseadas no status: Novos, Pendentes e Finalizados. Cada seção utiliza componentes de página separados que compartilham a mesma estrutura base, diferenciando-se pelo parâmetro de *status* na requisição à API. A navegação entre as seções é realizada através de *links* que mantêm o usuário na mesma estrutura de *layout* enquanto alterna entre os diferentes estados dos pedidos.

```
// Estrutura de navegação entre status
<div className="flex space-x-6">
<Link href="/adm/pedidos" className={` ${jaro.className} font-bold text-
[#4B1F1F] text-xl border-b-2 border-[#4B1F1F] pb-1`} >
Novos
</Link>
<Link href="/adm/pedidos/pendentes" className={` ${jaro.className} text-
gray-600 text-xl hover:text-[#4B1F1F]`} >
Pendentes
</Link>
<Link href="/adm/pedidos/finalizados" className={` ${jaro.className} text-
gray-600 text-xl hover:text-[#4B1F1F]`} >
Finalizados
</Link>
</div>
```

Cada página de pedidos implementa estados locais para gerenciar a lista de pedidos, modais de confirmação e o pedido selecionado. A busca é realizada através de `useEffect` que faz requisições à API filtrando pelo *status* correspondente. O tratamento de estados de carregamento fornece *feedback* visual durante a obtenção dos dados, enquanto a atualização local da lista após cada ação garante a consistência da interface.

```
// Busca de pedidos por status
useEffect(() => {
```

```

const fetchPedidos = async () => {
  try {
    const response = await fetch('/api/pedidos?status=NOVO');
    const data = await response.json();
    setPedidos(data);
  } catch (error) {
    console.error('Erro ao buscar pedidos:', error);
  } finally {
    setLoading(false);
  }
};
fetchPedidos();
}, []);

```

Os itens dos pedidos são processados para converter quantidades de gramas para quilogramas quando necessário, melhorando a legibilidade. As informações de data e hora são formatadas para o padrão brasileiro, e as formas de pagamento são representadas visualmente através de ícones correspondentes.

```

// Formatação de quantidades e exibição de itens
pedido.itens.map((item, index) => {
  const isKg = item.quantidade >= 1000;
  const quantidadeFormatada = isKg
    ? `${item.quantidade / 1000}kg`
    : `${item.quantidade}g`;

  return (
    <p key={index} className={` ${jaldi.className} text-[15px] text-
[#126C00] font-bold leading-tight mb-1`} >
    • {quantidadeFormatada} {item.produto.nomeProduto}
    </p>
  );
})}

```

O sistema implementa um fluxo de trabalho completo para os pedidos, com transições de *status* gerenciadas através de requisições PATCH. Pedidos novos podem ser confirmados (movendo para Pendentes) ou recusados (excluídos), pedidos pendentes podem ser finalizados (movendo para Entregues), e pedidos finalizados podem ser arquivados (excluídos). Cada transição é precedida por confirmação modal para prevenir ações acidentais.

```

// Transição de status - Exemplo: Confirmar pedido
const handleConfirmarPedido = async () => {
  if (!selectedPedidoId) return;
  try {
    const response = await fetch(`/api/pedidos?id=${selectedPedidoId}`,
{
    method: 'PATCH',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ status: 'PENDENTE' }),

```

```

    });
    if (response.ok) {
      setPedidos(pedidos.filter((pedido) => pedido.id !== selectedPe-
        didoId));
    }
  } catch (error) {
    console.error('Erro ao confirmar pedido:', error);
  }
};

```

A API de pedidos oferece operações CRUD com filtragem por *status*. A criação de pedidos envolve registrar o cliente, calcular o valor total com base nos produtos e quantidades, e inserir os itens. A exclusão remove primeiro os itens do pedido e depois o pedido, mantendo a integridade referencial.

```

// Cálculo do valor total na criação do pedido
const valorPedido = itens.reduce((total: number, item: any) => {
  const produto = produtos.find(p => p.id === item.produtoId);
  const preco = produto?.preco || 0;
  return total + ((item.quantidade/1000) * preco);
}, 0);

```

Esta estrutura proporciona um sistema completo de gestão de pedidos, desde a criação pelo cliente até o processamento administrativo final, com controle preciso do fluxo de trabalho e manutenção da integridade dos dados.

5.4 Cardápio

A página do cardápio foi desenvolvida para exibir os produtos disponíveis, filtrando apenas aqueles com status ativo. A interface utiliza o estado local para armazenar a lista de produtos e o carrinho, com dados sendo carregados inicialmente via `useEffect`. A listagem é apresentada em *cards* que incluem imagem, nome, preço e um botão para adicionar ao carrinho, redirecionando para a página de seleção de quantidade. O design emprega fontes personalizadas e cores específicas para manter a identidade visual do estabelecimento.

```

// Carregamento de produtos e estado do carrinho
useEffect(() => {
  (async () => {
    try {
      const res = await fetch("/api/cardapio");
      const data = await res.json();
      setProdutos(data);
    } catch (error) {
      console.error("Erro ao carregar produtos:", error);
    }
  })();
}

```

```

    })();

    const loadCarrinho = () => {
      const carrinhoStr = localStorage.getItem("carrinho");
      const parsed = carrinhoStr ? JSON.parse(carrinhoStr) : [];
      setCarrinho(normalizeCarrinho(parsed));
    };
    loadCarrinho();
  }, []);

```

O carrinho é gerenciado no `localStorage` para persistência dos dados entre sessões. A função `normalizeCarrinho` é utilizada para padronizar os itens do carrinho, convertendo quantidades de gramas para quilogramas quando necessário e garantindo que os preços e quantidades sejam números válidos. A atualização do carrinho é feita em tempo real, escutando eventos de `storage` para sincronizar abas diferentes. O cálculo do total é realizado somando o preço de cada item multiplicado pela quantidade em quilogramas.

```

// Normalização e cálculo do carrinho
function normalizeCarrinho(raw: unknown): { produtoId: number; preco: number; quantidade: number }[] {
  if (!Array.isArray(raw)) return [];
  return raw.map((it: ItemCarrinhoLS) => {
    const preco = toNumber(it.preco);
    let quantidade = toNumber(it.quantidade);
    // Se quantidade aparenta estar em gramas (>= 100), converte para
    kg
    if (quantidade > 50) quantidade = quantidade / 1000;
    return {
      produtoId: typeof it.produtoId === "string" ? Number(it.produtoId) : (it.produtoId as number),
      preco,
      quantidade,
    };
  });
}

const total = carrinho.reduce((sum, item) => sum + item.quantidade * item.preco, 0);

```

A página de detalhes de um produto específico é acessada via rota dinâmica, buscando os dados do produto por ID na API. A página exibe imagem em tamanho ampliado, nome em destaque sobre fundo bege, descrição completa e preço. O botão de voltar no canto superior esquerdo permite retornar ao cardápio principal, enquanto o botão "Adicionar" redireciona para a página de seleção de quantidade. A estrutura mantém consistência visual com o restante da aplicação.

```

// Estrutura da página de detalhes
<div className="w-full max-w-[430px] bg-white rounded-[40px] shadow-xl flex
flex-col overflow-hidden relative p-6">
  <Link href="/cardapio" className="absolute top-5 left-5 flex items-cen-
ter z-10 group">
    <Image src="/arrow.svg" alt="Voltar" width={24} height={24} />
    <span className={` ${jaldi.className} text-black font-bold`} >Vol-
tar</span>
  </Link>

  <div className="flex justify-center mt-10">
    <img src={produto.imagem} alt={produto.nomeProduto} className="w-
[300px] h-[300px] object-cover rounded-[20px]" />
  </div>

  <div className="bg-[#F1EBE1] text-center py-3 mt-6">
    <h1 className={` ${jaques_Francois.className} text-3xl font-
bold`} >{produto.nomeProduto}</h1>
  </div>
</div>

```

As APIs do cardápio foram implementadas para retornar a lista de produtos ativos e os detalhes de um produto específico. A rota `/api/cardapio` retorna todos os produtos com `statusProduto` verdadeiro, enquanto a rota `/api/cardapio/[id]` retorna um produto específico, também verificando se está ativo antes de disponibilizar os dados.

```

// API para listagem de produtos ativos
export async function GET() {
  try {
    const produtos = await prisma.produto.findMany({
      where: { statusProduto: true }
    });
    return NextResponse.json(produtos);
  } catch (error) {
    return NextResponse.json({ error: "Erro ao buscar cardápio" }, {
      status: 500 });
  }
}

// API para detalhes de produto específico
export async function GET(_req: NextRequest, { params }: { params: { id:
string } }) {
  try {
    const id = Number(params.id);
    const produto = await prisma.produto.findFirst({
      where: { id, statusProduto: true },
    });
    if (!produto) {
      return NextResponse.json({ error: "Produto não encontrado" }, {
        status: 404 });
    }
    return NextResponse.json(produto, { status: 200 });
  } catch (error) {
    return NextResponse.json({ error: "Erro ao buscar produto" }, {
      status: 500 });
  }
}

```

```

    }
  }
}

```

A interface do cardápio utiliza componentes de navegação entre páginas. O rodapé com total e botão do carrinho é exibido condicionalmente quando há itens adicionados. A tipografia emprega as fontes Italianno para o título e Jaques Francois para os nomes dos produtos, definindo a hierarquia visual.

```

// Estrutura do rodapé condicional
{carrinho.length > 0 && (
  <footer className="p-4 flex justify-between items-center bg-[#FFF4E3]
border-t border-gray-300">
  <span className={`${jaques_Francois.className} text-xl sm:text-
2xl`} >{formatBRL(total)}</span>
  <Link href="/carrinho">
    <button className={`${jaro.className} text-xl sm:text-2xl px-6
py-2 rounded-[20px] text-white bg-[#4B1F1F] hover:bg-[#5a2a2a] transition-
colors`} >
      Ver Carrinho
    </button>
  </Link>
</footer>
)}

```

Essa aplicação oferece uma navegação completa pelo cardápio, permitindo ao usuário visualizar os produtos, acessar seus detalhes e gerenciar o carrinho, garantindo uma integração eficiente entre o *front-end* e as APIs do *back-end*.

5.5 Carrinho

A página do carrinho implementa um sistema de gerenciamento de estado que utiliza o `localStorage` para persistência dos dados entre sessões. O componente emprega o *hook* `useState` para controlar a lista de itens e o estado de carregamento, enquanto `useEffect` é utilizado para carregar os dados iniciais e sincronizar mudanças entre abas através do evento *storage*. A função `carregarCarrinho` realiza a leitura e validação dos dados armazenados, garantindo a integridade das informações exibidas.

```

const carregarCarrinho = () => {
  const carrinho = localStorage.getItem("carrinho");
  if (carrinho) {
    try {
      const dadosCarrinho = JSON.parse(carrinho);
      const itensFormatados = dadosCarrinho.map((item: any) => ({
        produtoId: item.produtoId || "",
        nomeProduto: item.nomeProduto || item.nome || "Produto",

```

```

        preco: item.preco || 0,
        imagem: item.imagem,
        quantidade: item.quantidade || 0
    }));
    setItens(itensFormatados);
  } catch (error) {
    console.error("Erro ao analisar dados do carrinho:", error);
    setItens([]);
  }
} else {
  setItens([]);
}
setLoading(false);
};

```

O sistema implementa funções específicas para cálculo de preços e formatação de quantidades. A função `calcularPrecoItem` converte a quantidade em gramas para quilogramas e calcula o valor proporcional baseado no preço por quilo. Já a função `formatarQuantidade` formata a exibição para mostrar valores inteiros ou decimais conforme necessário, melhorando a legibilidade para o usuário final.

```

const formatarQuantidade = (gramas: number) => {
  const kg = gramas / 1000;
  return kg % 1 === 0 ? kg.toString() : kg.toFixed(1);
};

const calcularPrecoItem = (precoPorKg: number, gramas: number) => {
  const kg = gramas / 1000;
  return precoPorKg * kg;
};

const totalCarrinho = itens.reduce(
  (total, item) => total + calcularPrecoItem(item.preco, item.quantidade),
  0
);

```

A interface apresenta cada item do carrinho em *cards* individuais contendo imagem, nome, quantidade formatada e preço calculado. Cada item inclui botões para edição (que redireciona para a página de seleção de quantidade) e exclusão. O rodapé condicional exibe o total geral e um botão para continuar com o pedido, aparecendo apenas quando há itens no carrinho. A navegação é facilitada pelo botão de voltar no canto superior esquerdo.

```

{itens.map((item) => (
  <div key={item.produtoId} className="bg-[#F1EBE1] rounded-[20px] p-4 border border-gray-200">
    <div className="flex items-center">
      <div className="mr-4">

```


O carrinho atua como ponto central no fluxo de compras, conectando a seleção de produtos com o processo de finalização do pedido. O botão Continuar no rodapé redireciona para a página de formulário de pedido, carregando todos os itens e o total calculado. A estrutura de navegação permite retornar facilmente ao cardápio para adicionar mais produtos ou ajustar quantidades, criando um fluxo intuitivo e eficiente para o usuário.

```
{itens.length > 0 && (
  <footer className="bg-[#F8F3EC] p-4 flex justify-between items-center
border-t border-gray-300">
  <span className={` ${jaques_Francois.className} text-2xl`} >
    R$ {totalCarrinho.toFixed(2)}
  </span>
  <Link href="/formulario">
    <button className={` ${jaro.className} text-xl sm:text-2xl px-10
py-2 rounded-[20px] text-white bg-[#4B1F1F] hover:bg-[#5a2a2a] transition-
colors`} >
      Continuar
    </button>
  </Link>
</footer>
)}
```

Esta implementação do carrinho proporciona uma experiência completa de gestão de itens, com cálculos precisos, persistência de dados e navegação fluida, servindo como ponte essencial entre a seleção de produtos e a finalização do pedido.

5.6 Modal de Confirmação

O desenvolvimento do sistema priorizou a criação de componentes reutilizáveis, como o ModalConfirmacao. Este componente é responsável por exibir confirmações interativas, adaptando-se a diferentes contextos através de propriedades customizáveis.

```
{/* Modal */}
<div
  className="fixed top-1/2 left-1/2 transform -translate-x-1/2 -trans-
late-y-1/2
  bg-[#4B1F1F] rounded-[20px] shadow-2xl z-[1000] min-w-
[300px] max-w-[90%]
  overflow-hidden border"
>
  <div className="p-6">
    {/* Conteúdo */}
    <p className={` ${jaro.className} text-white mb-6 text-2xl text-cen-
ter`} >
      {title}
    </p>
  </div>
</div>
```

```

</p>

{/* Botões */}
<div className="flex justify-center space-x-4">
  {/* Botão Não */}
  <button
    onClick={onClose}
    className={`{jaro.className} px-6 py-2 rounded-[15px]
text-white bg-red-700 hover:bg-red-600 transition-colors`}
  >
    {cancelText}
  </button>

  {/* Botão Sim */}
  <button
    onClick={onConfirm}
    className={`{jaro.className} px-6 py-2 rounded-[15px]
text-white ${confirmColor} hover:bg-green-600 transition-colors`}
  >
    {confirmText}
  </button>

  {/* Botão Cancelar (neutro, só se for habilitado) */}
  {showNeutralButton && (
    <button
      onClick={onNeutral}
      className={`{jaro.className} px-6 py-2 rounded-[15px]
text-white bg-yellow-600 hover:bg-yellow-500 transition-colors`}
    >
      {neutralText}
    </button>
  )}
</div>
</div>
</div>

```

A lógica central do modal gerencia sua visibilidade condicional e ações do usuário. O *overlay* de fundo permite o fechamento ao ser clicado, enquanto os botões de ação acionam funções específicas passadas via *props*. A estrutura flexível suporta um botão neutro opcional, expandindo os fluxos de interação disponíveis.

5.8 Regra de negócio

Sob a orientação do nosso orientador, optou-se por implementar as regras de negócio no *front-end*, uma vez que tal abordagem se mostrou mais simples e prática de aplicar. Considerando a natureza relativamente simples do sistema, não se justificava a imposição dessas regras diretamente no banco de dados, pois eventuais violações não comprometeriam significativamente a integridade ou o funcionamento geral da aplicação.

```

    //regras RN-RF006.1 e RN-RF006.3
    const hoje = new Date();
    let diasMinimos = 1;

    if (totalG < 4000) diasMinimos = 1;
    else if (totalG < 10000) diasMinimos = 2;
    else if (totalG < 20000) diasMinimos = 7;
    else diasMinimos = 15;

    // Função que avança apenas em dias úteis
    const calcularDataMinima = (data: Date, dias: number) => {
        const novaData = new Date(data);
        let adicionados = 0;

        while (adicionados < dias) {
            novaData.setDate(novaData.getDate() + 1);
            const diaSemana = novaData.getDay();
            if (diaSemana !== 0 && diaSemana !== 6) {
                adicionados++;
            }
        }

        return novaData;
    };

    const dataFinal = calcularDataMinima(hoje, diasMinimos);

    const yyyy = dataFinal.getFullYear();
    const mm = String(dataFinal.getMonth() + 1).padStart(2, "0");
    const dd = String(dataFinal.getDate()).padStart(2, "0");

    setDataMinima(`${yyyy}-${mm}-${dd}`);
}
}, []);

```

No módulo de preenchimento dos dados do formulário para a data de retirada, implementamos a regra de negócio relacionada ao prazo mínimo de retirada, conforme definido na matriz de rastreabilidade (Quadro 6). A regra RN-RF006.1 determina que o tempo de preparo do pedido depende do peso total: pedidos inferiores a 4 kg devem ser retirados em 1 dia; entre 4 e 10 kg, em 2 dias; entre 10 e 20 kg, em 7 dias; e entre 20 e 50 kg, em 15 dias.

Além disso, a regra RN-RF006.3 estabelece que a contagem de dias deve excluir os finais de semana, garantindo que apenas dias úteis sejam considerados para a data mínima de retirada. A implementação utiliza o *front-end* para calcular dinamicamente a data mínima com base no total de produtos no carrinho, fornecendo ao usuário um *feedback* imediato sobre o prazo de retirada permitido. Este exemplo mostra como implementar regras de negócio no *front-end*, validando interações do usuário de forma imediata e garantindo cumprimento de algumas das regras de negócio do requisito RF006 – Enviar Pedido.

6 Resultados e Discussão

Um Documento de Portabilidade é um registro que descreve as características necessárias para que um sistema ou aplicação possa ser executado em diferentes tipos de dispositivos e ambientes tecnológicos. Ele especifica os requisitos mínimos de hardware e software para garantir a funcionalidade do sistema, considerando tanto a infraestrutura disponível quanto as condições ideais para um desempenho satisfatório.

A criação de um Documento de Portabilidade é fundamental para garantir que o sistema seja acessível e funcional em várias plataformas, o que maximiza o alcance e a eficiência de uso. Esse documento ajuda a prever possíveis desafios ao uso da aplicação em diferentes dispositivos e ambientes, além de facilitar a implementação de futuras atualizações. O Quadro 8 mostra as características do sistema.

Quadro 8 – Características do sistema

Item	Característica
Dados da Empresa	Nome da Empresa: Doce Mara Nome do Contato: Lucimara Sousa Piza Telefone: (16) 99459-8079
Infraestrutura	Rede de Dados: não há rede interna ou infraestrutura de servidores dedicada. A conexão à internet é feita por meio de redes móveis (2G a 5G), disponíveis para smartphones. Rede Elétrica: não existem requisitos especiais além de energia elétrica comum para recarregar o dispositivo móvel. Eventuais instantabilidades na rede podem impactar a operação se os <i>smartphones</i> não estiverem carregados ou conectados à rede elétrica.
Hardware	Dispositivo móvel (<i>smartphone</i>) J5 é o único recurso tecnológico disponível. A empresa não possui computadores de mesa. Periféricos: nenhum periférico específico, como impressoras ou <i>scanners</i> , é utilizado no processo produtivo.
Software	Antivírus Norton Mobile. Navegadores padrão e aplicativos de comunicação (como WhatsApp).
Recursos Humanos	Quantidade de usuários: um administrador principal (gestor da loja). Clientes diversos realizando pedidos e consultas on-line. Nível de Alfabetização Digital: o administrador possui familiaridade básica com <i>smartphones</i> e navegação web. A plataforma foi projetada com interface intuitiva para facilitar o uso por clientes com diferentes níveis de conhecimento digital. Mudança constante de usuário: não há rotatividade interna significativa. O gestor será o principal operador do sistema.
Configuração Mínima	Dispositivos Móveis: Sistema operacional: Android 8.0+ ou iOS 12+ Navegador: Google Chrome, Safari, ou outros navegadores atualizados Conectividade: Conexão 3G/4G ou Wi-Fi estável Autonomia de bateria: Mínimo de 6 horas de uso Computadores ou Desktops: Sistema operacional: Windows 10+, macOS 10.12+, ou distribuições Linux recentes Navegador: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge (versões atualizadas)

Item	Característica
	Memória RAM: 4 GB ou superior Processador: Dual-core 2.0 GHz ou superior Conectividade: Internet banda larga para garantir acesso estável

Fonte: os autores

Com essas configurações mínimas, o sistema será plenamente acessível em dispositivos móveis e computadores, atendendo tanto às necessidades do gestor da loja quanto dos clientes. A portabilidade e flexibilidade garantem que a plataforma funcione de forma consistente, independentemente do tipo de dispositivo utilizado, com foco na experiência fluida e eficiente.

6.1 Métricas

Métricas são indicadores quantitativos usados para medir e avaliar o desempenho, progresso ou qualidade de um determinado processo, produto ou sistema. Elas fornecem dados que ajudam a entender como um projeto está progredindo em relação aos objetivos estabelecidos.

Como aponta Pinheiro (2015, p. 81):

As métricas de software servem para realizar a tarefa fundamental do gerenciamento de projetos que é o planejamento. É com uma métrica que se pode identificar a quantidade de esforço necessário para o desenvolvimento, o custo para tal e o prazo para entrega dos artefatos.

A relevância das métricas reside no fato de que elas servem como uma base para tomada de decisões informadas. Sem métricas claras, é difícil identificar problemas, avaliar melhorias, ou determinar se um projeto está no caminho certo para atingir suas metas. As métricas permitem monitorar o desempenho em tempo real, ajustando as estratégias quando necessário para garantir o sucesso do projeto. Elas também ajudam a justificar investimentos, alinhar expectativas dos *stakeholders* e assegurar que os recursos sejam utilizados de forma eficaz.

As Figuras 15 e 16 apresentam as métricas do projeto.

6.2 Proposta Comercial

A proposta comercial é um documento formal utilizado para apresentar os detalhes de um projeto ou serviço a um cliente potencial, com o objetivo de firmar uma parceria ou fechar um contrato.

Figura 15 – Métricas

Função	Nº de ocorrência	Complexidade	Peso	Resultado	Nível de Influência do Sistema (0 a 5)	
entradas	5	Simple	3	0	Comunicação de dados	5
		Médio	4	20	Performance	5
		complexo	6	0	Volume de transações	5
saídas	5	Simple	4	4	Eficiência do usuário final	5
		Médio	5	25	Processamento complexo	5
		complexo	7	0	Facilidade de implantação	5
					Múltiplos locais	5
consultas	6	Simple	3	0	Processamento distribuído	5
		Médio	4	24	Utilização de equipamento	5
		complexo	6	0	Entrada de dados on-line	5
arquivos	5	Simple	7	35	Atualização on-line	5
		Médio	10	0	Reutilização de código	5
		complexo	15	0	Facilidade operacional	5
					Total de NI	70
interfaces	8	Simple	5	30	Facilidade de mudanças	5
		Médio	7	0		
		complexo	10	0		
Total de FP'b				138	FA = multiplicar o NI pela taxa real = 0,65*(0,01*70)	1,35

Fonte: os autores

Figura 16 – Continuação das Métricas

		FP'r = multiplicar o FP'b pelo FA	186
		Informe o nº de LOC da Linguagem	15
Estimativas do número médio de LOC por FP		KLOC = Multiplicar o FP'r pelo tipo de linguagem	
Cobol	100	Total de KLOC	2795
Pascal	90		
Linguagens Orientadas a Objeto (C++)	30	Informe o tipo de sistema	3.300
Java / Delphi / Visual Basic / C#	20		
Geradores de Código (SQL + HTML + RUBY + PYTHON + DEMAIS)	15	PRAZO (dividir o KLOC pelo tipo de sistema)	
		(RESULTADO DA DIVISÃO) QTDD DE MESES	0,84682
Tipo de Sistema	Produ - Kloc/Loc /mês	QTDD DE DIAS (% x 22)	18,48
Sistema Comercial	2.500	QTDD DE HORAS (% x 6)	2,88
Comércio Eletrônico	3.600	QTDD DE MINUTOS (% x 60)	52,8
Sistema Web	3.300		
CUSTO - INVESTIMENTO			
PRAZO		Informe o valor da hora de trabalho	R\$ 20,00
DIAS/MÊS	22	ISO (NORMA INTERNACIONAL) HORAS/MÊS =	132
HORAS/DIA	6	FÓRMULA = 132 * RESULTADO DA DIVISÃO * VALOR DA HORA	
MINUTOS/HORA	60		
CONFIRMAÇÃO (RESULTADO * 132 * Vr DA HORA)		VALOR TOTAL DO PROJETO =	R\$ 2.235,60

Fonte: os autores

Esse documento costuma descrever os produtos ou serviços oferecidos, os prazos, os custos, os termos de pagamento, as condições de entrega e outras informações essenciais para alinhar as expectativas entre as partes envolvidas.

A importância de uma Proposta Comercial é significativa, especialmente em negociações que envolvem projetos personalizados ou serviços que precisam ser adaptados às necessidades do cliente. “A proposta comercial é peça fundamental para o nascimento de um projeto, pois dá início às discussões sobre seu planejamento, além de auxiliar na execução e na entrega do mesmo” LEONARDO (2017).

Com base nos objetivos identificados, nossa proposta visa implementar uma solução que modernize o processo de gestão de encomendas da sua loja de balas de coco. Atualmente, o processo é manual e suscetível a falhas, o que limita a eficiência e aumenta o risco de erros. Essa iniciativa permitirá que a empresa otimize o controle de pedidos, melhore a eficiência operacional e ofereça uma experiência mais aprimorada e transparente para os clientes.

A justificativa para a implementação deste sistema é clara: os processos atuais apresentam lentidão, especialmente na administração de encomendas. Essa limitação será solucionada através de um sistema digital que automatiza o gerenciamento de pedidos e centraliza as informações em uma única plataforma. Conforme identificado na análise, a solução proposta inclui a digitalização do processo de encomendas, no qual os clientes poderão fazer pedidos diretamente on-line, o que agiliza o atendimento e torna o processo mais prático e acessível; e a automação do gerenciamento de encomendas, pois o sistema oferece controle em tempo real para o administrador e a equipe, reduzindo a possibilidade de erros humanos e permitindo uma gestão mais precisa dos pedidos.

A solução inclui módulo de Gestão de Encomendas que possibilita o controle completo e centralizado dos pedidos pelo administrador; módulo Cliente que permite fazer pedidos de forma on-line e automatizada, com uma experiência aprimorada e maior transparência em todo o processo; e a centralização de informações reduzindo falhas e melhorando o fluxo de dados para produção, otimizando a comunicação entre as etapas do processo.

A implantação do sistema está prevista para ser concluída em aproximadamente 18 dias, seguindo um cronograma detalhado para garantir uma transição suave e bem-sucedida.

O investimento necessário para uso do sistema é de R\$ 2.235,60, abrangendo os custos de configuração, treinamento e suporte inicial.

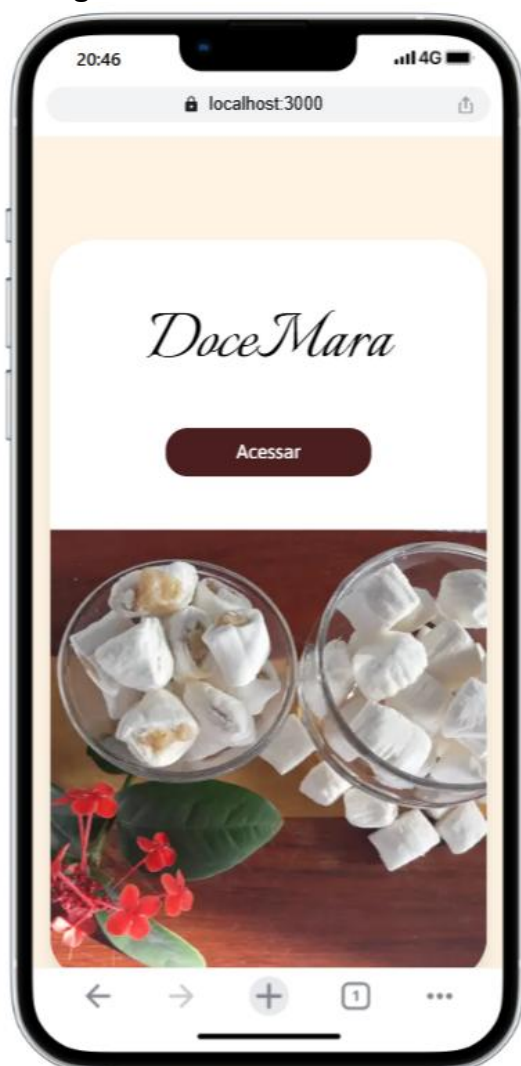
Em resumo, a implementação deste sistema é um passo essencial para o futuro da loja. Ele não apenas otimizará o trabalho interno e reduzirá os riscos operacionais, mas também proporcionará uma experiência de atendimento mais ágil e confiável, atendendo às expectativas dos clientes e abrindo novas possibilidades de crescimento.

6.3 Interface implementada

As telas do sistema foram capturadas do acesso com Microsoft Edge, utilizando a extensão *U-Eyes: Mobile Device Simulator*, que simula a interface do *smartphone* iPhone 13 Pro Max.

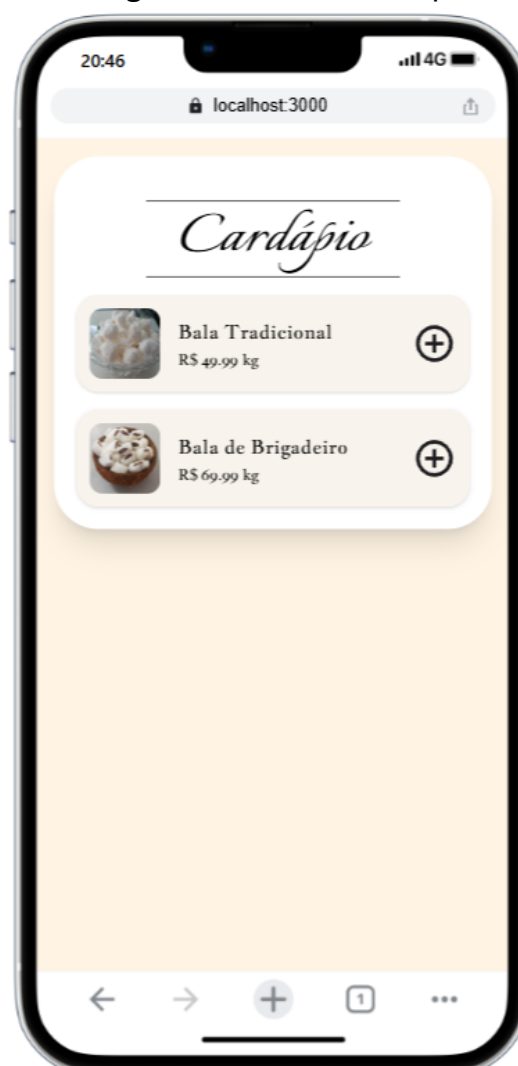
A Figura 17 exibe a tela de boas-vindas que direciona o cliente ao Cardápio (Figura 18).

Figura 17 – Tela de Boas-Vindas



Fonte: os autores

Figura 18 – Tela Cardápio



Fonte: os autores

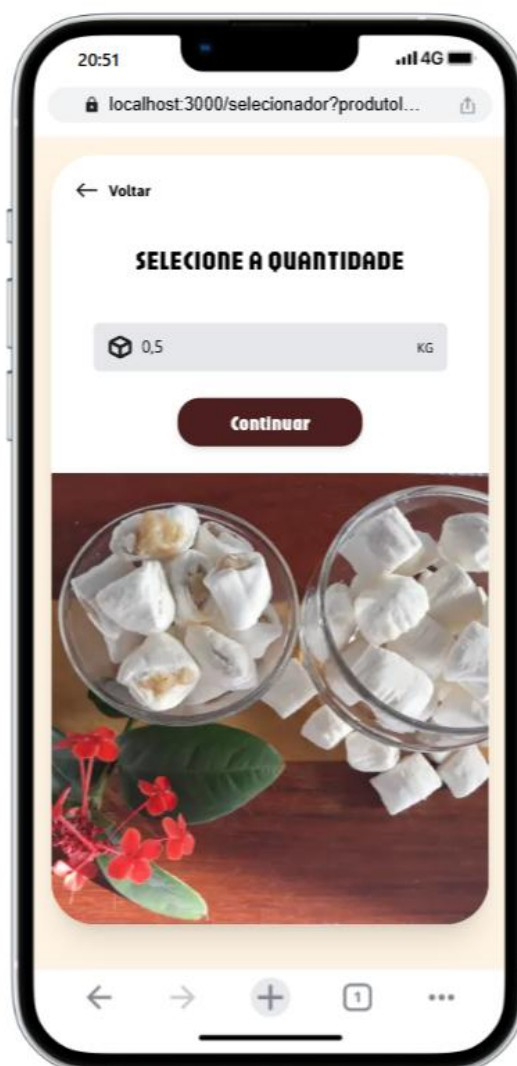
A Figura 19 mostra a tela de detalhes do produto e a Figura 20 o selecionador de quantidade do produto.

Figura 19 – Tela Detalhes do Produto



Fonte: os autores

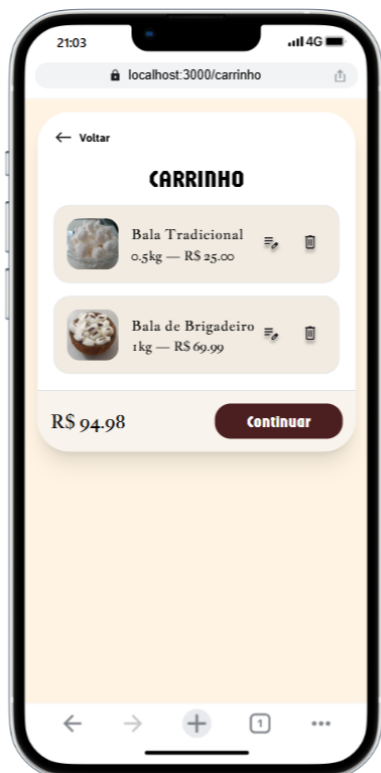
Figura 20 – Tela Selecionador



Fonte: os autores

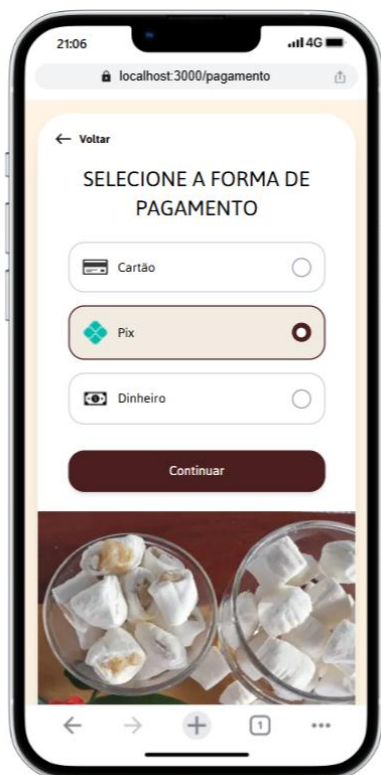
A Figura 21 exhibe o carrinho com os produtos selecionados pelo cliente; a 22 apresenta a tela para o cliente preencher os dados (Nome, Telefone, Data de Retirada e Horário de Retirada); a 23 mostra a tela na qual o cliente deve selecionar a forma de pagamento entre as opções disponíveis; e a 24 apresenta a tela na qual o cliente deve confirmar o pedido e enviá-lo ou cancelá-lo.

Figura 21 – Tela Carrinho



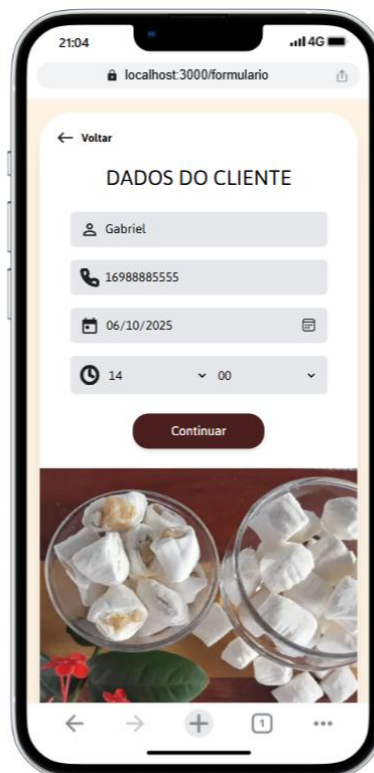
Fonte: os autores

Figura 23 – Tela Pagamento



Fonte: os autores

Figura 22 – Tela Formulário



Fonte: os autores

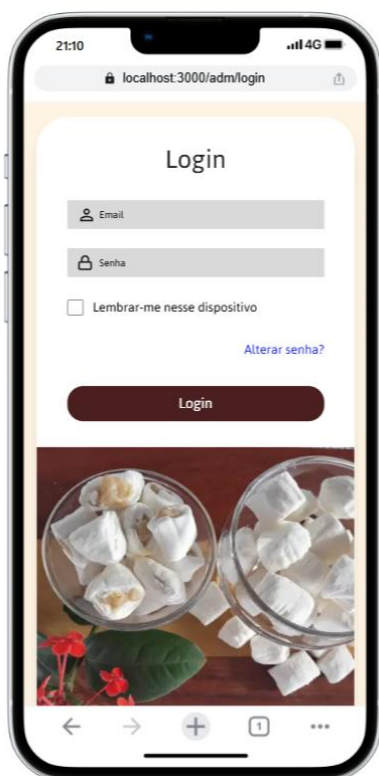
Figura 24 – Tela Finalização



Fonte: os autores

A Figura 25 apresenta a tela na qual o administrador deve digitar suas credenciais para acessar o sistema; a 26 a tela para alterar a senha; a 27 a tela para visualização dos produtos cadastrados e seus *status* do cardápio (ativo ou inativo) e alterá-los; a 28 apresenta a tela na qual o administrador preenche os dados para cadastrar um novo produto e também para alterar esses dados; a 29 apresenta a tela para visualização de novos pedidos; a 30 mostra a tela na qual o administrador pode aceitar ou recusar pedido com o modal de confirmação; e a 31 apresenta a tela que o administrador recebe o lembrete de justificar ao cliente caso pedido recusado, e sua exclusão.

Figura 25 – Tela Login



Fonte: os autores

Figura 26 – Tela Alterar Senha



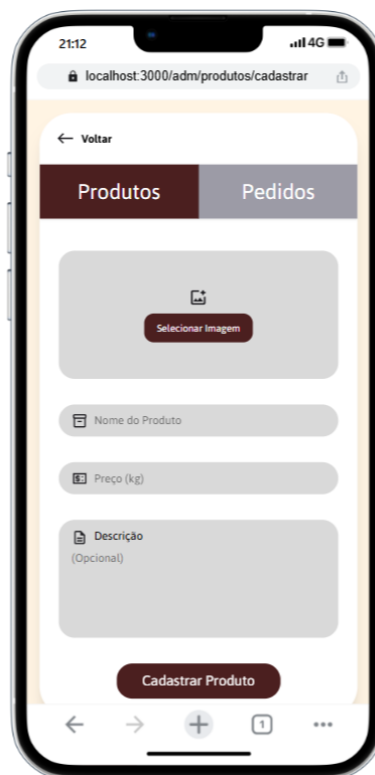
Fonte: os autores

Figura 27 – Tela Produtos



Fonte: os autores

Figura 28 – Tela Formulário Produto



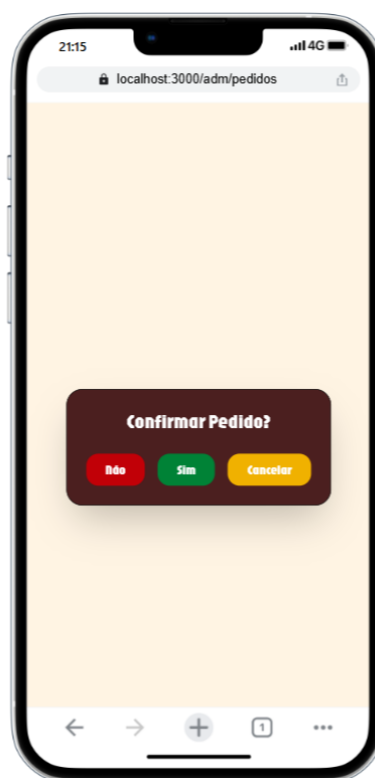
Fonte: os autores

Figura 29 – Tela Novos Pedidos



Fonte: os autores

Figura 30 – Tela Confirmar Pedido



Fonte: os autores

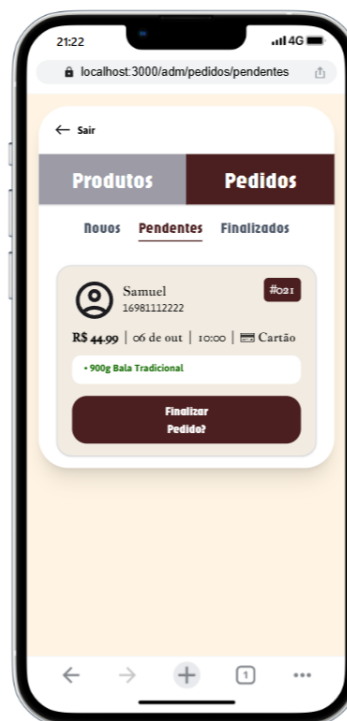
A Figura 32 mostra a tela onde o administrador pode ver os pedidos pendentes; e a 33 a tela que o administrador consulta pedidos finalizados.

Figura 31 – Tela Justificar ao Cliente



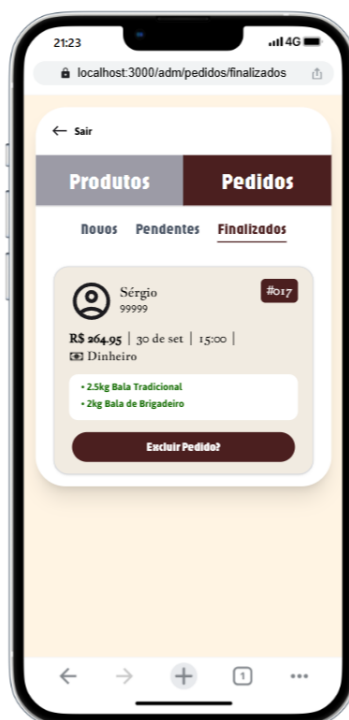
Fonte: os autores

Figura 32 – Tela Pedidos Pendentes



Fonte: os autores

Figura 33 – Tela Pedidos Finalizados



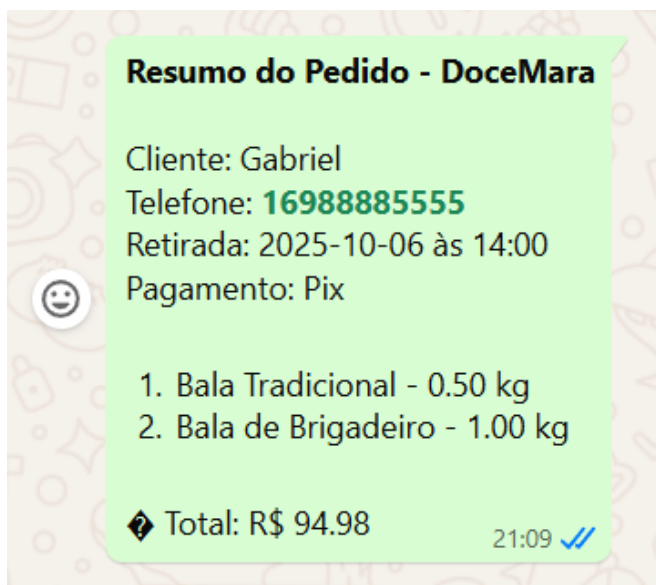
Fonte: os autores

6.4 Telas complementares para operação do sistema

O uso do sistema envolve a comunicação por mensagens importantes que acontecem fora sistema, mas que fazem parte do seu funcionamento.

A Figura 34 apresenta um exemplo de mensagem que o sistema gera para o cliente para ser enviada para o Administrador. Essa mensagem é o comprovante do pedido do cliente.

Figura 34 – Mensagem *WhatsApp*



Fonte: os autores

Considerações Finais

Durante o desenvolvimento do projeto, enfrentamos algumas dificuldades, principalmente relacionadas às correções de fluxo no lado do cliente, que exigiram ajustes constantes para garantir uma experiência de usuário coerente e funcional.

Observamos também que algumas decisões poderiam ter sido melhor planejadas, especialmente na estrutura das tabelas do banco de dados; devido ao planejamento inicial, tivemos que adotar soluções alternativas para contornar limitações e atender aos requisitos do sistema. Um exemplo foi a necessidade de impedir a exclusão de produtos ligados a pedidos, permitindo apenas sua desativação.

Entre as melhorias futuras, destaca-se a implementação de um processo de notificação ao cliente mencionado no BPMN (figura 3), que aumentaria a eficiência da comunicação e a experiência de uso da plataforma por parte do administrador.

Apesar do conhecimento básico da equipe em algumas tecnologias, conseguimos desenvolver o projeto dentro do escopo definido, atendendo aos

requisitos funcionais e implementando as regras de negócio de forma prática e eficaz. Como resultado, o requerente do sistema aprovou a solução proposta em sua fase final.

Conclui-se, portanto, que a decisão de trabalhar com um escopo sucinto e objetivo foi sábia, permitindo cumprir os prazos e entregar um sistema funcional e alinhado às expectativas.

Referências

ASSOCIAÇÃO BRASILEIRA DE BARES E RESTAURANTES(ABRASEL). **Bares e restaurantes estão mais digitais na gestão e mais presentes nas redes sociais, diz pesquisa da Abrasel**. [S. l.], 18 jul. 2023. Disponível em: <https://mg.abrasel.com.br/noticias/noticias/bares-e-restaurantes-estao-mais-digitais-na-gestao-e-mais-presentes-nas-redes-sociais-diz-pesquisa-da-abrasel/>. Acesso em: 30 out. 2024.

BARROS, J. P. **Caso de Uso e Respectivos Diagramas**. 2009. 4 p. Escola Superior de Tecnologia e Gestão Instituto Politécnico de Beja. 2009. Acesso em: <http://gres.uninova.pt/~jpb/textos/useCaseDiagrams.pdf>. Acesso em 08 nov. 2024.

BATISTA, E. A. **Uma taxonomia facetada para técnicas de elicitação de requisitos**. 2003. 150 p. Dissertação (Mestrado em Engenharia de Computação) – Universidade Estadual de Campinas, Instituto de Computação, Campinas, 2003. Disponível em: <https://repositorio.unicamp.br/acervo/detalhe/302022>. Acesso em: 07 nov. 2024.

BENCZIK, E. F. **Modelo De Abertura Para Projetos Industriais**. 2021. 58 p. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Automotiva) – Universidade Federal de Santa Catarina, Joinville, 2021. Disponível em: <https://repositorio.ufsc.br/handle/123456789/228262>. Acesso em: 06 nov. 2024.

BERTALANFFY, L. V. **General System Theory: Foundations, Development, Applications**. New York: George Braziller, 1968.

BIERMAN, G.; ABADI, M.; TORGENSEN, M. Understanding TypeScript. In: JONES, R. (ed.). **ECOOP 2014 – Object-Oriented Programming**. Berlim: Springer, 2014. p. 1-25. (Lecture Notes in Computer Science, v. 8586). DOI: 10.1007/978-3-662-44202-9_1. Disponível em: https://doi.org/10.1007/978-3-662-44202-9_1. Acesso em: 28 ago. 2025.

BORGES, R. F. **Uso de BPMN para definição de planos de cuidado em saúde na plataforma Takere**. 2023. 53 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2023. Disponível em: <https://lume.ufrgs.br/handle/10183/274057>. Acesso em: 07 nov. 2024.

CREATELY. **Tutorial do Diagrama de Sequência: Guia completo com exemplos.** [S. I.], 29 jan. 2021. Disponível em: <https://creately.com/blog/pt/diagrama/tutorial-do-diagrama-de-sequencia/>. Acesso em: 09 nov. 2024.

ESPINHA, R. G. **Matriz de Rastreabilidade de Requisitos: saiba como gerenciar as mudanças no escopo.** [S. I.], 30 jul. 2020. Disponível em: <https://artia.com/blog/matriz-de-rastreabilidade/>. Acesso em: 09 nov. 2024.

FELISBINO, Cláudio Marcio. **Ferramenta para o apoio ensino-aprendizagem do modelo orientado a objetos durante a construção do diagrama de classes.** 2017. 118 f. Dissertação (Mestrado em Computação Aplicada) - Universidade Tecnológica Federal do Paraná, Curitiba, 2017. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/2848>. Acesso em 08 nov. 2024.

FOCCOERP. **5W2H: o que é, qual sua importância e como usar?**. [S. I.], 01 jul. 2023. Disponível em: <https://foccoerp.com.br/blog/5w2h/>. Acesso em: 06 nov. 2024.

IBM. **Desenvolvimento de software.** Disponível em: <https://www.ibm.com/br-pt/think/topics/software-development>. Acesso em: 5 out. 2025.

LEONARDO, A. **Tipos de proposta comercial: modelo para desenvolvimento de software.** [S. I.], 23 fev. 2017. Disponível em: <https://negocieapp.com.br/tipos-de-proposta-comercial-modelo-para-desenvolvimento-de-software/>. Acesso em: 11 nov. 2024.

MENDES, F. C. **Sistema de acompanhamento de egressos UFO: levantamento e análise de requisitos.** 2023. 76 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal de Uberlândia, Uberlândia, 2023. Disponível em: <https://repositorio.ufu.br/handle/123456789/39896>. Acesso em: 07 nov. 2024.

MORENO, G. D.; MASSON, P. H. S.; CARAVIERI, F. P. M. **MK Manager: software para cadastro e gerenciamento de produtos.** 2022. Artigo de Graduação (Tecnologia em Análise e Desenvolvimento de Sistemas) – Faculdade de Tecnologia Prof. José Camargo, Jales, 2022. Artigo apresentado no VI Simpósio de Tecnologia da Fatec Jales – SITEF, 2022, Jales-SP. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/11397>. Acesso em: 08 nov. 2024.

PARKER, A. **Diagrama de atividades em UML: símbolo, componentes e exemplo.** [S. I.], 26 set. 2024a. Disponível em: <https://www.guru99.com/pt/uml-activity-diagram.html>. Acesso em: 09 nov. 2024.

PARKER, A. **Diagrama de máquina de estado e diagrama de gráfico de estado em UML.** [S. I.], 26 set. 2024b. Disponível em: <https://www.guru99.com/pt/state-machine-transition-diagram.html>. Acesso em: 09 nov. 2024.

PAULA, G. B. D. **O que é 5W2H: reduza incertezas, ganhe produtividade e aprenda como fazer um plano de ação.** [S. I.], 08 nov. 2015. Disponível em: <https://www.treasy.com.br/blog/5w2h/>. Acesso em: 06 nov. 2024.

PINHEIRO, A. F. **Fundamentos da Engenharia de Software Análise, Modelagem e Métricas**. 1ª ed. Recife: Publicação Independente, 2015. 108 p. Disponível em: https://www.researchgate.net/profile/Alvaro-Pinheiro-2/publication/341327200_Analysis_Modeling_and_Metrics/links/602fc553a6fdcc37a8381f5e/Analysis-Modeling-and-Metrics.pdf. Acesso em 10 nov. 2024.

PRESSMAN, R. S. **Engenharia de Software**: uma abordagem profissional. 6ª ed. São Paulo: McGraw-Hill, 2005.

THEIS, M. R. et al. A IMPORTÂNCIA DA PROTOTIPAGEM NO PROCESSO DE DESIGN E SUAS RELAÇÕES COMO MÍDIA DO CONHECIMENTO. **Anais do Congresso Internacional de Conhecimento e Inovação – ciki**, [S. l.], v. 1, n. 1, 2022. DOI: 10.48090/ciki.v1i1.1058. Disponível em: <https://proceeding.ciki.ufsc.br/index.php/ciki/article/view/1058>. Acesso em: 12 nov. 2024.

TOTVS. **BMPN: entenda o que é a modelagem de processos de negócios, como fazer e sua importância!**. [S. l.], 01 mar. 2023. Disponível em: <https://www.totvs.com/blog/gestao-industrial/bpmn/>. Acesso em: 07 nov. 2024.

VICELLI, B.; TOLFO, C. **Um estudo sobre valor utilizando o Business Model Canvas**. Revista Espacios, v. 38, n. 3, p. 14, 2017. Disponível em: <https://www.revistaespacios.com/a17v38n03/a17v38n03p14.pdf> . Acesso em: 07/10/2025.

Apêndice 1

Questionário e respostas da entrevista com o Administrador

1 – Quem serão os usuários do sistema?

R: Os clientes em geral e a proprietária da loja.

2 – Como você atualmente gerencia as encomendas?

R: Anoto cada encomenda em um caderno, registrando as informações necessárias, e risco as encomendas quando são concluídas.

3 – Quais são os principais desafios que você enfrenta no processo de gestão de encomendas?

R: O principal desafio é o tempo gasto para responder às perguntas dos clientes sobre as encomendas e registrá-las manualmente no caderno.

4 – Como você prefere receber as encomendas?

R: Prefiro recebê-las em um único local centralizado, onde eu possa visualizá-las facilmente para gerenciar de forma eficiente.

5 – Quais métodos de pagamento você aceita em sua loja?

R: Aceitamos pagamentos em dinheiro, Pix, cartão de débito e cartão de crédito.

6 – Quais informações você precisa para marcar uma encomenda?

R: Preciso da data da encomenda, nome do cliente, telefone, quantidade desejada e tipo de bala.

7 - Como você faz o acompanhamento das encomendas em andamento?

R: Faço o acompanhamento pelo caderno, onde todas as encomendas estão anotadas.

8 - Você usa algum software ou sistema atualmente? Se sim, quais?

R: Atualmente, utilizo apenas o WhatsApp para receber pedidos de encomenda por mensagem.

9 - Qual é o volume médio de encomendas que você recebe por semana?

R: Recebo em média 20 encomendas por semana.

10 - Como funciona a escolha de sabores personalizados?

R: Os sabores personalizados são feitos sob encomenda e devem ser solicitados a partir de 500g.

11 - Como você imagina a interface ideal do sistema?

R: A interface ideal seria de fácil acesso e intuitiva, permitindo uma navegação simples e eficiente.

12 - Que tipo de informações e relatórios você gostaria de acessar facilmente no sistema?

R: Gostaria de acessar informações detalhadas dos clientes, como nome, telefone, histórico de encomendas, além de relatórios sobre o total de encomendas feitas na semana.

13 - Você tem alguma preferência por dispositivos (computador, tablet, smartphone) para acessar o sistema?

R: Prefiro acessar o sistema pelo smartphone.

14 - Você oferece serviço de entrega a domicílio ou pretende oferecer?

R: No momento, não oferecemos serviço de entrega a domicílio e não há planos para isso.

15 - Qual o objetivo principal do sistema?

R: O principal objetivo do sistema é gerenciar as encomendas feitas pelos clientes de maneira mais eficiente.

Questionário e respostas da entrevista com os Clientes via Google Forms

1 - Como você prefere fazer uma encomenda de bala de coco?

Online: 74,2%

Por telefone: 12,9%

Não tenho preferência: 12,9%

Presencialmente: 0%

2 - Com que antecedência você faria uma encomenda de balas de coco?

No mesmo dia: 12,9%

Com 1 ou 2 dias de antecedência: 45,2%
De 3 a 7 dias de antecedência: 35,5%
Com mais de uma semana de antecedência: 6,4%

3 – Você acha útil receber uma confirmação da encomenda por mensagem caso fazê-la online?

Sim por mensagem (WhatsApp): 96,8%
Não, não preciso de confirmação: 3,2%

4 – Você já encomendou balas de coco personalizadas em termos de sabor?

Sim: 35,5%
Não: 32,3%
Nunca pensei nisso: 32,3%

5 - Você gostaria de receber atualizações em tempo real sobre o andamento da sua encomenda?

Sim, gostaria de receber por mensagem (WhatsApp): 67,7%
Não vejo necessidade de receber atualizações: 32,3%

6 - O que é mais importante para você ao encomendar balas de coco?

Qualidade do produto: 83,9%
Preço acessível: 9,7%
Facilidade no processo de encomenda: 3,2%
Variedade de sabores: 0%
Atendimento ao cliente: 3,2%

7 - Com que frequência você encomenda balas de coco para eventos ou ocasiões especiais?

Raramente: 29,9%
Algumas vezes ao ano: 45,2%
Mensalmente: 0%
Nunca encomendei: 25,8%

8 - O que faria você recomendar a loja para amigos ou familiares?

Qualidade das balas de coco: 90,3%
Bom atendimento: 2,7%
Facilidade de encomenda: 7%
Variedade de sabores: 0%

9 - Há alguma funcionalidade específica que tornaria sua experiência mais agradável ou acessível?

- A) Poder repetir a última encomenda feita com apenas um clique.
- B) Uma forma de acelerar o processo de pedido na próxima vez.
- C) Rapidez no atendimento
- D) Realizar o pagamento após a encomenda através de um QR code / link para pagamento via pix ou cartão
- E) Menu com variedades de sabores.
- F) Uma interface amigável seria agradável.
- G) Ter um feedback dos usuários sobre sabores, cálculo de quantidade para uma festa, embalagens personalizadas.

- H) Eu acredito que uma interface intuitiva com ilustrações ajudaria bastante os usuários na seleção dos produtos.
- I) Facilidade de escolher sabores, facilidade de ver valores que o pagamento seja feito fora do aplicativo poderia ter algum lugar com a informação de quais formas de pagamentos são aceitas.
- J) Mostrar imagem da bala de coco a ser encomendada e especificações do peso e tamanho da unidade.
- K) Cardápio disponível.
- L) Informar os ingredientes das balas recheadas, para informar as pessoas com intolerância a lactose ou alergias.
- M) Entrega em embalagem especial.
- N) Fotos reais de como são.

Apêndice 2

Missão: criar balas de coco artesanais que tragam felicidade e conexão com tradições familiares.

Visão: ser reconhecida como a melhor bala de coco de Franca e região, transformando doces em experiências emocionais e significativas.

Valores: compromisso com a qualidade artesanal e transformar simplicidade em algo especial e significativo