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

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GABRIEL SANTOS SILVA
HEITOR DA SILVA SAQUETO

OTIMIZAÇÃO DO ATENDIMENTO EM CASAS DE CARNE ATRAVÉS DA TECNOLOGIA DA INFORMAÇÃO

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. Fausto Gonçalves Cintra

FRANCA/SP
2024

OTIMIZAÇÃO DO ATENDIMENTO EM CASAS DE CARNE ATRAVÉS DA TECNOLOGIA DA INFORMAÇÃO

Gabriel Santos Silva[1]

Heitor da Silva Saqueto[2]

Fausto Gonçalves Cintra[3]

Resumo

O presente trabalho aborda o desenvolvimento de uma ferramenta digital voltada para o setor de casas de carne, visando a otimização do atendimento ao cliente e a modernização do acesso aos produtos oferecidos. A ferramenta possibilitará que os consumidores consultem o menu de produtos remotamente, realizem pedidos online e agendem retiradas ou entregas, promovendo um atendimento mais ágil e eficiente. A implementação do comércio eletrônico no contexto das casas de carne é analisada sob a perspectiva de sua contribuição para a ampliação do alcance geográfico do estabelecimento e a conveniência oferecida aos clientes, além de representar um diferencial competitivo no mercado. A pesquisa considera também os desafios logísticos associados ao transporte de produtos perecíveis e destaca a importância de investimentos em tecnologia e capacitação para o sucesso da ferramenta digital. Com essa abordagem, o estudo pretende contribuir para a modernização do setor e para uma experiência de consumo mais satisfatória e prática.

Palavras-chave: atendimento. casas de carne. comércio eletrônico. experiência do cliente. tecnologia da informação.

Abstract

This paper addresses the development of a digital tool aimed at optimizing customer service and modernizing product access in the butcher shop sector. The tool will enable consumers to remotely consult the product menu, place online orders, and schedule pickups or deliveries, thus promoting a more agile and efficient service. The implementation of e-commerce within the butcher shop context is analyzed from the perspective of its potential to expand the geographical reach of establishments and enhance customer convenience, while also representing a competitive advantage in the market. The research also considers logistical challenges associated with transporting perishable goods and emphasizes the importance of technology investment and staff

[1] Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: gssilva2711@gmail.com.

[1] Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: heitorsaqueto@hotmail.com.

[1] Docente em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: fausto.cintra@fatec.sp.gov.br.

training for the digital tool's success. This approach aims to contribute to the sector's modernization and provide a more satisfying and practical consumer experience.

Keywords: butcher shops. customer experience. e-commerce. information technology. servisse.

1 Introdução

A carne é um dos alimentos mais antigos consumidos pela humanidade, e seu papel na dieta humana é profundo e essencial. Desde os tempos pré-históricos, o consumo de carne proporcionava nutrientes fundamentais que foram determinantes para o desenvolvimento da espécie. A domesticação de animais e a prática da criação animal trouxeram à sociedade um meio mais sustentável de obter carne, transformando-a em um componente central da alimentação e da cultura humana ao longo dos séculos (Harari, 2013). Com a organização social e econômica, surgiram as casas de carne, estabelecimentos especializados que atuam como ponto de acesso direto entre produtores e consumidores, garantindo cortes de qualidade e a manutenção dos padrões de segurança alimentar.

Em um cenário cada vez mais influenciado pela tecnologia, o comércio eletrônico, ou *e-commerce*, tem se consolidado como uma ferramenta poderosa de transformação, ampliando o alcance de estabelecimentos e oferecendo praticidade e conveniência aos consumidores. Esse tipo de comércio tem mudado a forma como as pessoas consomem, permitindo o acesso remoto a produtos e serviços de maneira rápida e prática (Menezes, 2018). No setor das casas de carne, o *e-commerce* representa uma oportunidade estratégica, permitindo que esses estabelecimentos ampliem seu mercado, alcançando um público mais amplo e diversificado.

Com o avanço do comércio digital, casas de carne têm a possibilidade de expandir seus negócios, permitindo que clientes distantes tenham acesso a cortes de carne frescos e de qualidade sem precisar se deslocar até o local. Esse acesso remoto é particularmente relevante em um contexto onde a conveniência é cada vez mais valorizada e onde o consumidor busca otimizar o tempo e reduzir esforços na hora da compra. A possibilidade de consultar o menu de produtos, visualizar cortes disponíveis e realizar pedidos diretamente pela internet agrega valor ao serviço oferecido pelas casas de carne, criando uma experiência de compra mais satisfatória e adaptada às necessidades contemporâneas.

Além da ampliação do alcance geográfico, a adoção do *e-commerce* pelas casas de carne também traz consigo uma melhoria significativa no

processo de atendimento ao cliente. Em horários de grande movimento, como os finais de semana e feriados, as casas de carne enfrentam desafios para atender a alta demanda, o que pode resultar em insatisfação e até perda de clientes. A possibilidade de realizar pedidos online, com agendamento para retirada ou entrega, permite uma organização mais eficiente do atendimento e reduz o tempo de espera.

O desenvolvimento de uma ferramenta digital destinada ao setor das casas de carne visa, portanto, otimizar o processo de atendimento, possibilitando que clientes consultem o menu de produtos remotamente, façam pedidos e agendem retiradas ou entregas. Esse recurso não só melhora a experiência do consumidor, mas também promove uma operação mais fluida e organizada para o estabelecimento. Em um mercado altamente competitivo, essa adaptação às demandas tecnológicas pode representar um diferencial importante para as casas de carne, atraindo e fidelizando uma base de clientes cada vez mais digital.

É importante observar que a implementação do e-commerce em estabelecimentos que lidam com produtos perecíveis, como as casas de carne, exige um planejamento cuidadoso para garantir a qualidade e a segurança dos produtos até o momento em que chegam ao consumidor. A adaptação das operações logísticas e o rigor no controle de qualidade são fundamentais para o sucesso dessa estratégia, e representam um desafio que precisa ser enfrentado para que a experiência do cliente seja positiva e segura (Pinho, 2021).

Portanto, este trabalho tem como objetivo desenvolver uma ferramenta digital que possa otimizar o atendimento nas casas de carne, permitindo aos clientes uma experiência de compra mais ágil e prática, além de promover maior eficiência no estabelecimento. Com funcionalidades como visualização do cardápio, pedidos online e agendamento de retirada ou entrega, a ferramenta propõe uma nova dinâmica de interação entre o cliente e o estabelecimento, alinhada às tendências tecnológicas e de consumo da atualidade.

A adoção de tecnologias digitais no setor de casas de carne é um passo significativo para modernizar e otimizar o atendimento, trazendo conveniência para o consumidor e melhorando a produtividade e a gestão do

estabelecimento. Essa modernização vai além de uma mudança operacional; ela representa uma transformação cultural, que envolve a capacitação dos funcionários, a implementação de novas práticas de atendimento e a criação de uma experiência mais rica e personalizada para o cliente.

Ao integrar o *e-commerce* no contexto das casas de carne, este trabalho busca também refletir sobre os impactos mais amplos da digitalização no setor alimentício. A tecnologia da informação, como elemento fundamental na organização e na gestão do atendimento, possibilita uma experiência mais dinâmica, confortável e eficiente, reforçando a relevância da transformação digital em segmentos tradicionalmente presenciais.

1.1 Termo da Abertura do Projeto (TAP)

O TAP (Termo de Abertura do Projeto) é um utilizado para formalizar o início de um projeto. Ele define os principais aspectos, como objetivo, escopo, *stakeholders*, entre outros. O TAP (Tabela 1) é geralmente desenvolvido nas fases iniciais do projeto e precisa ser aprovado pelas partes interessadas para garantir o alinhamento sobre o que será entregue.

Segundo o *Project Management Institute* (PMI), o TAP serve como uma ferramenta de comunicação para garantir que todos os envolvidos no projeto estejam cientes das suas expectativas e responsabilidades, ajudando a reduzir o risco de falhas devido à falta de clareza. Além disso, de acordo com Kerzner (2017), o TAP é fundamental para garantir que o projeto seja oficialmente autorizado e que os recursos e esforços sejam formalmente alocados para sua execução.

O TAP define não só o que o projeto pretende entregar, mas também suas premissas, restrições e os critérios de sucesso, estabelecendo as bases para o planejamento subsequente e a gestão do projeto. Assim, sua importância está diretamente relacionada à orientação clara sobre o que será desenvolvido, quem estará envolvido e como o projeto será conduzido até a sua conclusão.

Tabela 1 – Termo de Abertura do Projeto

Seção	Detalhes
Identificação do Projeto	Otimização do atendimento em casas de carne através da tecnologia da informação
Justificativa do Projeto	O gerenciamento de uma casa de carnes sem um sistema eficiente sobrecarrega o atendimento durante horários de pico, o que resulta em filas longas e insatisfação dos clientes. Além disso, o estilo de vida moderno, com a falta de tempo para atividades cotidianas, reforça a necessidade de soluções que permitam aos consumidores realizarem pedidos de forma eficiente, sem a necessidade de comparecimento físico ao estabelecimento. Este projeto também busca resolver problemas como desorganização interna e erros no processamento de pedidos.
Objetivos e Metas	• automatizar o processo de recebimento de pedidos. • proporcionar uma melhor experiência de compra. • reduzir o tempo de espera dos clientes. • facilitar a gestão de pedidos para os funcionários.
Descrição do Projeto	Desenvolver uma ferramenta web que permita a realização de pedidos. A ferramenta deve ser intuitiva, segura e oferecer uma boa experiência de compra para os clientes.
Premissas	• o sistema deve atender às necessidades da casa de carnes. • deve estar em conformidade com as regulamentações vigentes, como a LGPD. • o projeto deve ser entregue conforme o cronograma.
Restrições	• prazo até novembro de 2024. • o escopo foca no desenvolvimento de uma ferramenta web.
Escopo	• ferramenta web para realização de pedidos. • sistema de gerenciamento de pedidos.
Resultados Esperados	• melhoria na organização e processamento de pedidos. • experiência de compra melhorada para o cliente.

Não-escopo	• compra e instalação de equipamentos. • construção de infraestrutura física. • funcionalidades extras como integração com aplicativos de terceiros.
Prazos Previstos	Início: agosto de 2024 Término: novembro de 2025
Custo Previsto	O projeto será desenvolvido utilizando recursos internos da casa de carnes e tecnologias *open-source*. Não estão previstos custos adicionais, além de possíveis gastos com hospedagem do sistema e infraestrutura.

Fonte: Autores

2 Viabilidade do Projeto

Nesta seção, serão abordadas ações estratégicas que sustentam a viabilidade do projeto, que será analisada com base na aplicação do *Business Model Canvas* (BMC). Através dessa ferramenta, será possível mapear os principais elementos que compõem o modelo de negócio.

2.1 Canvas de Negócio (*Business Model Canvas* - BMC)

O Business Model Canvas (BMC) é uma ferramenta estratégica desenvolvida por Alexander Osterwalder e Yves Pigneur, apresentada em seu livro *Business Model Generation*. O BMC auxilia no desenvolvimento e na análise de modelos de negócios, proporcionando uma visão clara e visualmente estruturada dos principais componentes que formam um negócio.

A importância do BMC reside na sua capacidade de oferecer uma compreensão global e integrada de como uma organização cria, entrega e captura valor. Ele é amplamente utilizado para otimizar e inovar modelos de negócios, o que permite que empresas identifiquem áreas de melhoria e façam ajustes estratégicos.

O BMC permite também identificar como a Tecnologia da Informação pode melhorar aspectos chave de um negócio, como o relacionamento com clientes e a eficiência operacional. Sua aplicação facilita a visualização de áreas onde a tecnologia pode agregar valor e promover maior eficácia. A Figura 1 apresenta os blocos que estruturam essa ferramenta.

Figura 1 - Canvas

Fonte: Autores

O modelo é dividido em nove blocos fundamentais, que englobam áreas críticas para o sucesso da empresa:

1- Segmentos de Clientes: identifica os diferentes grupos de clientes que a empresa atende. Entender claramente quem são os clientes permite à empresa oferecer soluções mais precisas e eficientes.

2- Propostas de Valor: define a maneira em que a empresa se diferencia no mercado. É o conjunto de produtos e serviços que atende às necessidades dos clientes de forma exclusiva. Pode envolver inovação, custo-benefício, design superior, conveniência ou qualquer outro aspecto que faça com que os clientes escolham essa empresa em vez de outra. Esse bloco descreve o que torna o negócio relevante e competitivo.

3- Canais: representam as formas como a empresa comunica-se com seus clientes e entrega suas propostas de valor. Isso inclui todos os meios pelos quais os produtos e serviços são divulgados, vendidos e entregues aos consumidores. Podem ser canais físicos, como lojas, ou canais virtuais, como *e-commerce*. A escolha dos canais influencia a experiência do cliente e o sucesso na entrega do valor proposto.

4- Relacionamento com Clientes: descreve o tipo de relacionamento estabelecido entre a empresa e seus clientes. Um bom desenvolvimento do

relacionamento entre ambas as partes contribui para a fidelização e satisfação dos clientes. Diferentes segmentos podem requerer diferentes abordagens.

5- Fontes de Receita: descrevem como a empresa gera dinheiro a partir de cada segmento de clientes. Podem incluir vendas diretas, assinaturas, licenciamento, taxas de uso, aluguel ou comissões. As fontes de receita ajudam a esclarecer quais são os fluxos de receita mais eficazes para sustentar o negócio e como eles estão alinhados à proposta de valor oferecida.

6- Recursos-Chave: são os ativos mais importantes que a empresa necessita para operar com sucesso. Eles podem ser físicos (instalações, equipamentos), intelectuais (patentes, marcas), humanos (equipe qualificada) ou financeiros (capital, crédito). Esses recursos permitem que a empresa crie e entregue sua proposta de valor, mantenha seus canais, gere receita e estabeleça relacionamentos duradouros com os clientes.

7- Atividades-Chave: descrevem ações importantes que a empresa deve realizar para que seu modelo de negócio funcione. Podem incluir produção, marketing, desenvolvimento de produtos ou gestão de operações. A identificação dessas atividades ajuda a focar nos processos que são críticos para o sucesso e a entrega de sua proposta de valor.

8- Parcerias-Chave: incluem alianças com empresas, fornecedores, distribuidores ou parceiros estratégicos que ajudam a empresa a operar e crescer. Essas parcerias podem reduzir riscos, otimizar recursos ou criar novas oportunidades de mercado.

9- Estrutura de Custos: identifica os custos envolvidos na operação do modelo de negócio. Esses custos podem estar associados às atividades-chave, recursos-chave e parcerias-chave. Compreender esse bloco permite que a empresa planeje de forma mais eficiente.

3 Levantamento de Requisitos

A elicitação de requisitos é uma fase crítica do processo de desenvolvimento de sistemas, pois envolve a coleta e identificação das necessidades dos *stakeholders* (pessoas ou grupos que são afetados pelas ações do projeto) para garantir que o sistema atenda às expectativas e funcione de acordo com as especificações.

3.1 Elicitação e especificação dos Requisitos

A importância da elicitação de requisitos no desenvolvimento de software se destacada na capacidade de garantir que as necessidades e expectativas dos *stakeholders* sejam compreendidas corretamente desde o início do projeto. Segundo Sommerville (2011), a elicitação é essencial para evitar problemas de desalinhamento entre o que os usuários esperam e o que é entregue, pois qualquer falha nessa fase pode resultar em um sistema que não atenda adequadamente às necessidades do cliente. Esse processo é fundamental para capturar tanto os requisitos explícitos quanto os implícitos, que muitas vezes não são percebidos ou expressos diretamente pelos usuários (Kotonya; Sommerville, 1998).

Além disso, a elicitação de requisitos ajuda a identificar potenciais riscos e ambiguidades logo no início do projeto. Como apontado por Wiegers (2003), isso facilita a mitigação de problemas e o ajuste das expectativas, permitindo um planejamento mais preciso e eficaz. Ao fornecer uma base sólida para o desenvolvimento, a elicitação garante que o sistema será projetado de acordo com as demandas e limitações reais do negócio, aumentando as chances de sucesso do projeto e reduzindo o retrabalho.

Técnicas de elicitação, como entrevistas, questionários e visitas, foram utilizadas para obter uma compreensão clara e detalhada das necessidades do projeto. Segundo Sommerville (2011), a elicitação de requisitos é essencial para identificar de forma precisa o que os *stakeholders* desejam, evitando problemas de alinhamento entre a expectativa e a entrega do sistema.

A entrevista é uma técnica de elicitação de requisitos que consiste em uma conversa estruturada ou semiestruturada entre o analista de sistemas e os *stakeholders* do projeto, como clientes, usuários finais ou gerentes. O objetivo é obter informações detalhadas sobre as necessidades, problemas e expectativas em relação ao sistema a ser desenvolvido. Essa técnica é amplamente utilizada por sua flexibilidade e capacidade de fornecer uma compreensão profunda das demandas dos envolvidos.

Durante o levantamento de requisitos, as entrevistas foram conduzidas com os proprietários, funcionários da casa de carne, que lidam diretamente com o atendimento ao cliente e os processos operacionais, e alguns de seus clientes. No método de entrevista foram diagnosticados problemas práticos

enfrentados no dia a dia, como a superlotação em horários de pico, enquanto em outros momentos o movimento era muito baixo, o que gerava uma subutilização dos recursos disponíveis. Essa situação impactava tanto a qualidade do atendimento quanto a eficiência operacional.

Foi também identificado o desejo do cliente de personalizar melhor o seu pedido, permitindo que ele tivesse mais liberdade para customizá-lo e conhecer novos cortes de carne através da plataforma.

Ainda durante o processo de elicitação dos requisitos, foram distribuídas algumas questões com espaço em branco para que os *stakeholders* descrevesse em palavras suas expectativas em relação ao sistema e apontassem sugestões para melhorias no atendimento. As perguntas tinham como foco entender como os *stakeholders* enxergavam a atual operação e quais funcionalidades consideravam essenciais para a otimização do processo.

O método de questionário permite coletar dados de forma estruturada e padronizada, facilitando a análise e comparação das respostas. Os questionários são especialmente úteis quando é necessário obter opiniões ou informações de um grande número de pessoas, ou quando o tempo e a disponibilidade dos *stakeholders* são limitados.

Por fim, o método de observação, que é uma técnica de elicitação de requisitos que consiste em acompanhar as atividades dos usuários em seu ambiente de trabalho. Ele é essencial para identificar práticas e problemas que podem não ser mencionados em entrevistas, mas que impactam diretamente o uso do sistema. A observação ajuda a entender os fluxos de trabalho e as interações dos usuários com os processos e ferramentas existentes.

Nesse projeto, a observação foi usada para analisar o atendimento e o processamento dos pedidos. Durante as visitas, foi possível identificar a superlotação em horários de pico e a baixa demanda em outros momentos, assim como anteriormente mencionado durante o método de entrevista. Essa análise ajudou a encontrar soluções para equilibrar o fluxo de atendimento.

3.2 BPMN

O BPMN (*Business Process Model and Notation*) é uma notação padronizada para modelagem de processos de negócios. Segundo *Object Management Group* (OMG), que desenvolveu o BPMN, ele é uma linguagem

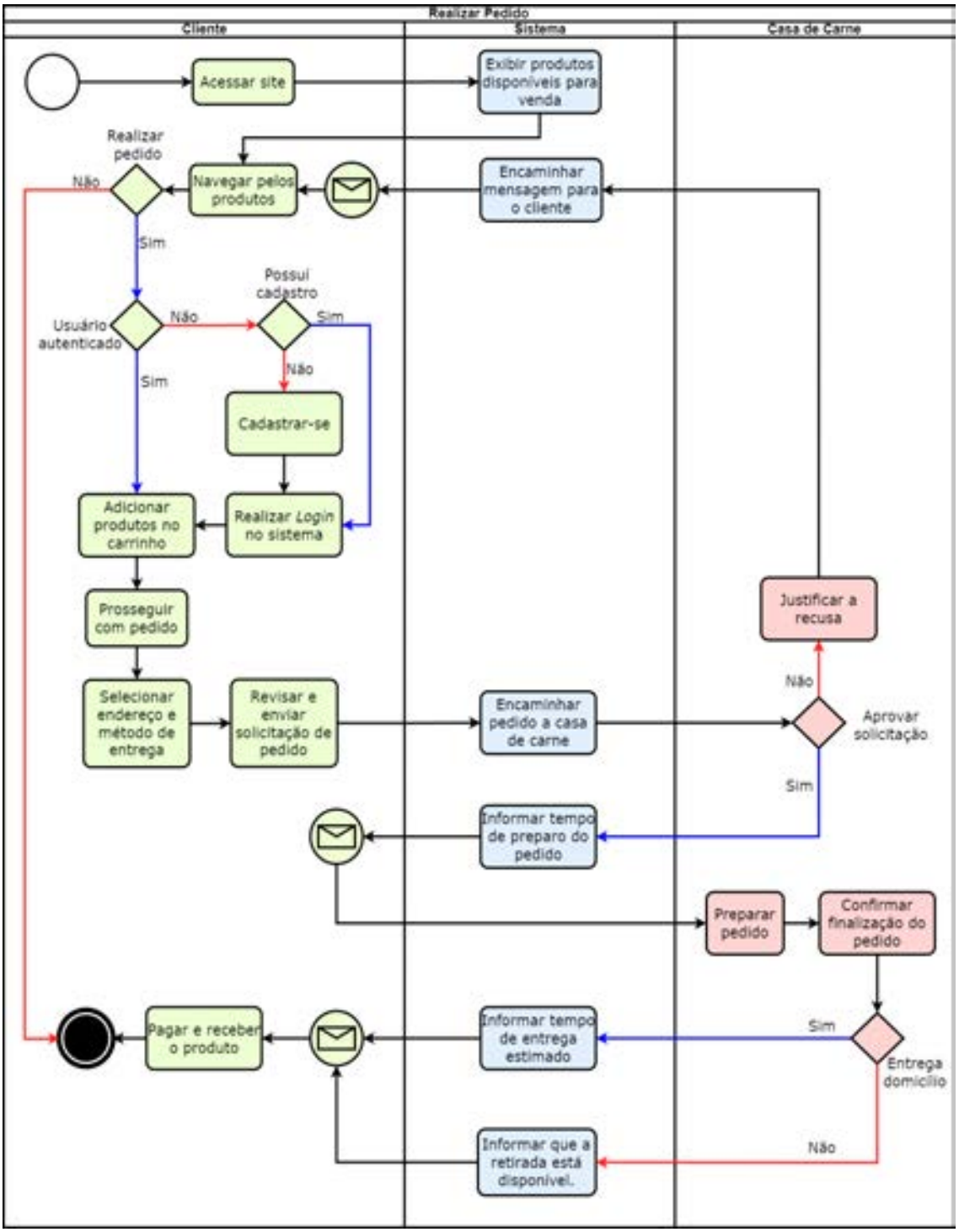
gráfica que facilita a compreensão dos processos tanto para técnicos quanto para profissionais de negócios. Sua estrutura visual é composta por elementos como eventos, atividades, fluxos e gateways, que representam de forma clara o fluxo de atividades dentro de um processo organizacional (OMG, 2011).

O BPMN serve para descrever, modelar e comunicar processos de negócio de forma clara e compreensível. Como apontado por White (2004), essa notação tem o objetivo de criar um entendimento comum entre os diversos *stakeholders* de uma organização, incluindo analistas de processos, desenvolvedores de sistemas e gerentes de negócios. Além de facilitar a comunicação, o BPMN também é utilizado para a análise e melhoria de processos, permitindo a identificação de gargalos, ineficiências e oportunidades de automação.

Para o presente projeto, a utilização de um BPMN permitiu mapear os processos de atendimento e processamento de pedidos de forma visual e estruturada. Isso facilita a compreensão dos fluxos de trabalho atuais e ajuda a identificar onde a tecnologia pode ser aplicada para melhorar a eficiência local. O BPMN também ajudou a garantir o alinhamento entre os envolvidos no que diz respeito aos requisitos levantados.

Os processos de como um pedido será executado, após a implementação do sistema, é foi desenvolvido no BPMN presente na figura 2.

Figura 2 – BPMN Realizar Pedido

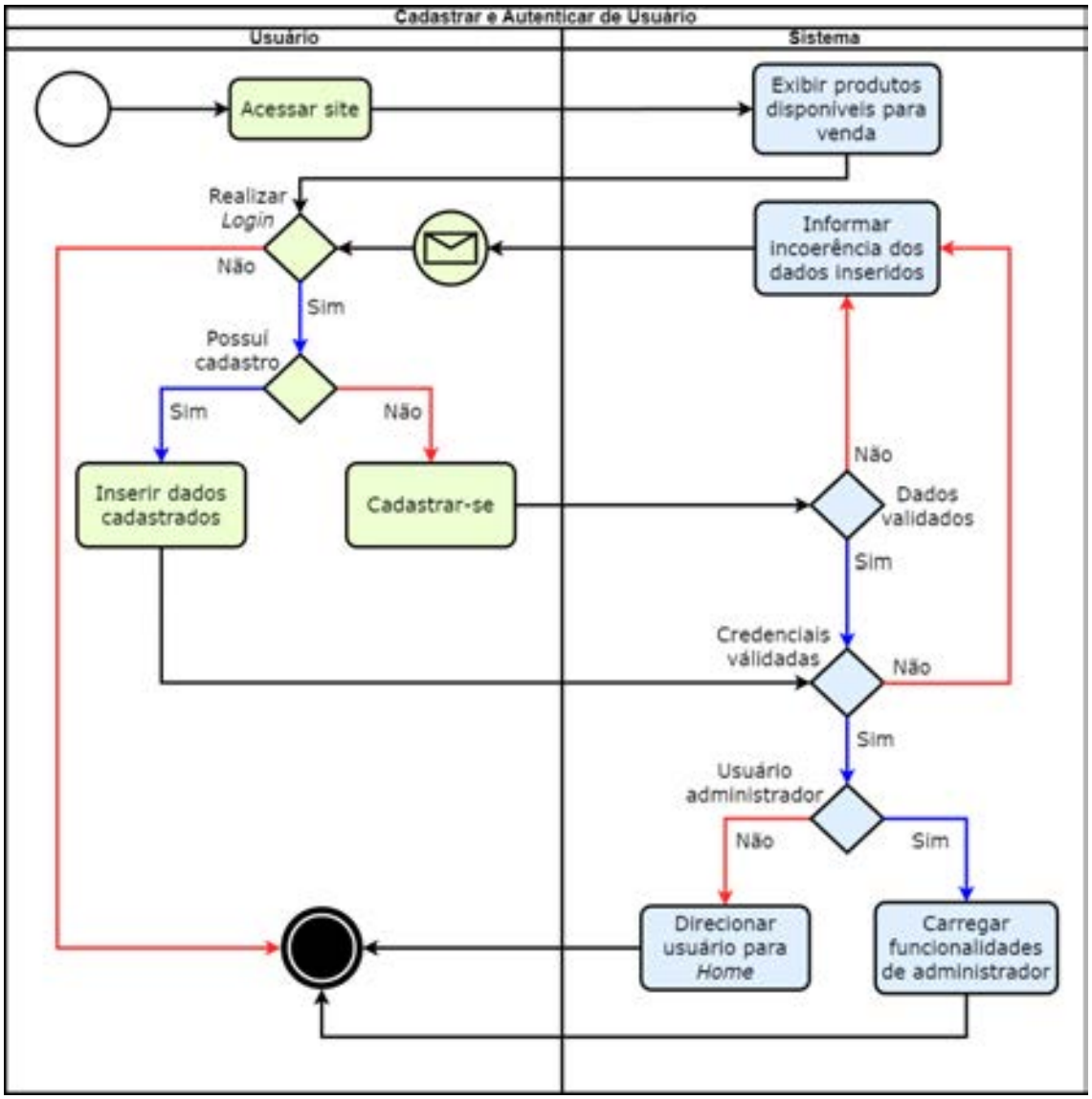


Fonte: Autores

Durante o processo de solicitação do pedido, é notório que ainda existe uma comunicação entre a casa de carnes e o cliente, pois o sistema nessa primeira versão não controlará o estoque da casa de carnes, o que torna preciso uma verificação manual de cada um dos pedidos.

A seguir, a figura 3 demonstra o BPMN do processo de autenticação de usuário, que está presente na figura 2.

Figura 3 – BPMN Cadastrar e Autenticar Usuário

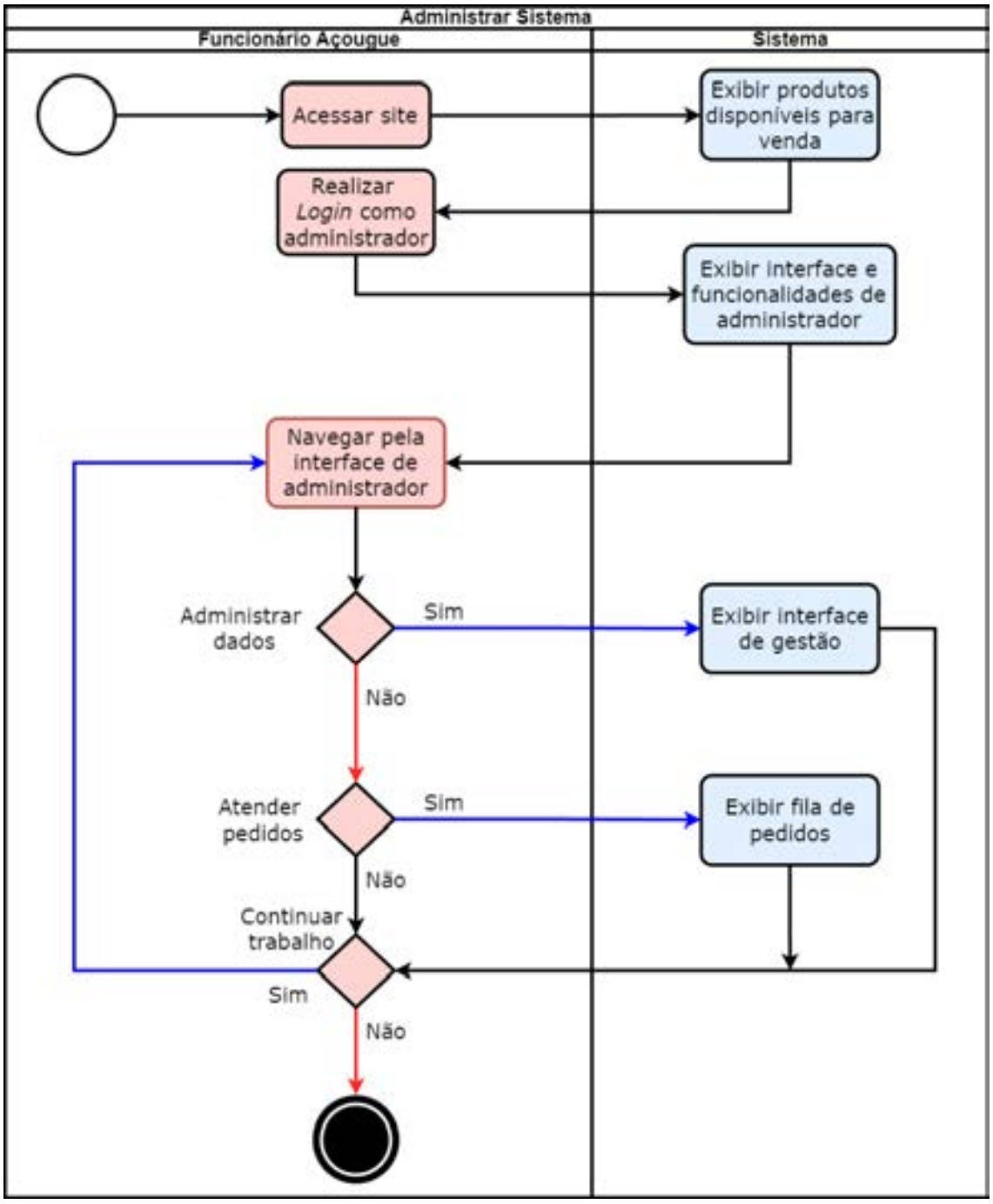


Fonte: Autores

Esse fluxo demonstra como um novo usuário é cadastrado no sistema e também apresenta a divergência de um funcionário comum e um funcionário administrador. Essa separação é importante pois os funcionários administradores possuem a capacidade de gerenciamento das funcionalidades do sistema.

Segue agora, por fim, a figura 4, que representa as funcionalidades disponíveis aos funcionários administradores.

Figura 4 – BPMN Administrar Sistema



Fonte: Autores

Esse último BPMN descreve os procedimentos que podem ser realizados pelos administradores do sistema, donos e/ou funcionários da casa de carnes. Ao ser autenticado pelo *software* o usuário é direcionado para uma interface de gerenciamento para que seu trabalho seja realizado.

3.3 Requisitos Funcionais

Os Requisitos Funcionais, conforme documentado no quadro 1, são descrições das funcionalidades e comportamentos que o sistema deve apresentar para atender às necessidades dos usuários e cumprir os objetivos definidos. Segundo Sommerville (2011), eles especificam o que o sistema deve fazer, incluindo as interações com os usuários, as entradas e saídas de dados, e as respostas a determinadas situações. Esses requisitos definem de forma clara as operações, cálculos, manipulações de dados e qualquer outra funcionalidade essencial para o sistema.

As definições adequadas dos requisitos funcionais guiam o processo de desenvolvimento e servem como base para a implementação do sistema. Como ressaltado por Pressman (2010), a clareza e a precisão desses requisitos são fundamentais para garantir que o sistema final corresponda às expectativas dos *stakeholders*, evitando ambiguidades e mal-entendidos. Além disso, os requisitos funcionais também são usados como referência no processo de testes, ajudando a validar se o sistema atende às suas especificações.

Quadro 1 – Documentação de requisitos funcionais

RF001-Registrar Usuários	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que os usuários se registrem inserindo informações básicas, como nome, e-mail, telefone e endereço.		
RF002-Visualizar menu de produtos	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que os usuários visualizem os produtos inseridos no sistema, com imagem, preço, descrição do produto e demais informações essenciais pertencentes a cada produto previamente inserido.		
RF003-Adicionar itens	Categoria: ( ) Oculto	Prioridade: (X) Altíssima

ao carrinho	(X)Evidente	( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que os usuários insiram os produtos selecionados no carrinho e alterem suas quantidades.		
RF004-Remover itens do carrinho	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que os usuários removam produtos do carrinho.		
RF005-Escolher método de entrega	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que os usuários escolham entre entrega em domicílio ou retirada no local.		
RF006-Receber requisição do cliente	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que os funcionários do estabelecimento recebam as requisições (pedidos) dos clientes para avaliar e aceitar ou recusar.		
RF007-Atualizar status do pedido (funcionário)	Categoria: ( ) Oculto (X)Evidente	Prioridade: ( ) Altíssima (X) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que o funcionário do estabelecimento altere manualmente o status do pedido como: pedido aceito, preparando pedido, pronto para retirada, a caminho, cancelado.		
RF008-Cancelar pedido	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média

		( ) Baixa
Descrição: O sistema deve permitir que os clientes e funcionários do estabelecimento cancelem os pedidos antes que eles comecem a ser preparados pela casa de carnes. Após a preparação do pedido, o cancelamento só deve ser possível mediante autorização do gerente ou supervisor.		
RF009-Manter histórico de pedidos	Categoria: ( ) Oculto (X)Evidente	Prioridade: ( ) Altíssima (X) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve armazenar e disponibilizar o histórico de pedidos dos clientes para consulta e/ou realização de novos pedidos idênticos.		
RF010-Verificar compra	Categoria: ( ) Oculto (X)Evidente	Prioridade: ( ) Altíssima (X) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve, antes de finalizar a compra, mostrar ao usuário detalhadamente o que ele pretende comprar, para que seja feita uma última revisão antes de efetuar o pedido.		
RF011-Notificar pedido	Categoria: ( ) Oculto (X)Evidente	Prioridade: ( ) Altíssima (X) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve, após a confirmação do pedido, notificar o cliente, via e-mail e/ou aplicativo, os detalhes da solicitação realizada, como: hora, produtos, preço, tempo médio de espera e tempo estimado para entrega/retirada.		
RF012-Atualizar status do pedido (cliente)	Categoria: ( ) Oculto (X)Evidente	Prioridade: ( ) Altíssima (X) Alta

		( ) Média ( ) Baixa
Descrição: O sistema deve manter o cliente informado sobre em que estado está o seu pedido, seja ele “pedido aceito”, “preparando pedido”, “pronto para retirada”, “a caminho”, “cancelado”.		
RF013-Gerenciar lista de pedidos recebidos	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve fornecer aos funcionários do estabelecimento uma lista em tempo real de todos os pedidos recebidos, em ordem cronológica.		
RF014-Gerenciar estoque	Categoria: ( ) Oculto (X)Evidente	Prioridade: ( ) Altíssima (X) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve fornecer aos funcionários do estabelecimento um gerenciador de estoque, onde possam visualizar a quantidade dos produtos e modificá-las de acordo com a necessidade. O funcionário deve atualizar o estoque manualmente ou, em caso de venda pelo *software*, o sistema é atualizado automaticamente.		
RF015-Atutenticar usuários	Categoria: (X) Oculto ( )Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve garantir que os clientes não interajam com interface de administrador/funcionário do *software*.		
RF016-Efetuar pedido	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que o cliente realize um pedido para a casa de carnes.		

017-Fazer *login* no *software*	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que o usuário tente realizar um *login* no *software* com um usuário e uma senha cadastrados anteriormente.		
RF018-Cadastrar novos usuário	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que o usuário tente realizar um *login* no *software* com um usuário e uma senha.		
RF019-Verificar credenciais após tentativa de *login*	Categoria: (X) Oculto ( )Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve verificar se o usuário e a senha, inseridos no campo de *login*, coincidem com algum cadastro salvo na base de dados. Caso o *login* seja malsucedido o sistema deve notificar o usuário com a mensagem “usuário ou senha inválidos”		

Fonte: Autores

3.4 Casos de Uso

O Caso de Uso é uma ferramenta utilizada na análise de sistemas para descrever as interações entre usuários e o sistema em desenvolvimento. Segundo Cockburn (2001), ele define uma sequência de ações realizadas por um ator externo (que pode ser uma pessoa ou outro sistema) para alcançar um objetivo específico utilizando o sistema. A ideia central é que os casos de uso detalham como o sistema deve se comportar em resposta às ações dos usuários, fornecendo uma visão clara de suas funcionalidades sob diferentes perspectivas.

A importância do Caso de Uso reside em sua capacidade de facilitar a comunicação entre os *stakeholders* envolvidos no projeto, como apontado por

Jacobson, Booch e Rumbaugh (1999). Ele serve como uma ponte entre os requisitos do cliente e a equipe de desenvolvimento, ajudando a garantir que o sistema final atenda às expectativas do cliente. Além disso, os casos de uso são úteis para a identificação de requisitos funcionais e para o planejamento de testes, uma vez que descrevem as interações esperadas com o sistema de forma detalhada.

3.4.1 Diagrama de Caso de Uso

Para o presente projeto foi desenvolvido o caso de uso apresentado na figura 5. A utilização de casos de uso permitiu descrever como os clientes irão interagir com o sistema de pedidos online, facilitando assim o alinhamento entre os desenvolvedores e o cliente, garantindo que todas as funcionalidades críticas, como a escolha de produtos e a finalização de pedidos, estivessem contempladas nos requisitos.

Figura 5 – Diagrama caso de uso

Fonte: Autores

O diagrama apresentado na figura 5 reúne todas as funcionalidades disponíveis para os usuários do sistema, tanto usuários comuns quanto administradores, entretanto apenas o funcionário administrador tem acesso a interface de gerenciamento.

3.4.2 Documentação de Caso de Uso

A documentação de Caso de Uso (Quadro 2) descreve de forma detalhada como o sistema interage com seus usuários, garantindo que todas as funcionalidades sejam compreendidas por todos os envolvidos no projeto. Conforme Sommerville (2011), essa documentação fornece uma visão clara dos fluxos de trabalho e das ações que os usuários realizam ao interagir com o sistema, o que ajuda a evitar interpretações ambíguas e inconsistentes. Dessa forma, serve como um ponto de referência tanto para a equipe técnica quanto para os *stakeholders*.

Quadro 2 – Documentação caso de uso

Caso de Uso – Visualizar menu de produtos	
ID	UC 001
Descrição	Este caso de uso descreve o processo em que os atores Cliente e Funcionário da Casa de Carnes podem visualizar o menu completo de produtos disponíveis na casa de carnes através do sistema.
Ator Primário	Usuário do sistema
Pré-condição	O sistema estar operacional.
Cenário Principal	• O ator seleciona a opção "Visualizar Menu" na interface do sistema. • O sistema exibe o menu completo de produtos da casa de carnes, incluindo nome do produto, imagem e preço. • O ator navega pelo menu para visualizar os diferentes itens disponíveis.
Pós-condição	O ator visualizou com sucesso o menu completo de produtos da casa de carnes.
Cenário Alternativo	3a. Se o ator desejar, ele pode realizar ações adicionais, como filtrar o menu por categoria ou buscar produtos específicos.

Caso de Uso – Adicionar ou remover itens ao carrinho	
ID	UC 002
Descrição	Este caso de uso descreve o processo em que o ator Cliente pode adicionar ou remover itens do menu ao carrinho de compras.
Ator Primário	Cliente
Pré-condição	Cliente visualizou o menu de produtos.
Cenário Principal	• O ator Cliente navega pelo menu de produtos e seleciona o item desejado. • O sistema exibe os detalhes do item selecionado, incluindo preço e quantidade disponível. • O ator Cliente decide adicionar o item ao carrinho e especifica a quantidade desejada. • O sistema atualiza o carrinho de compras do Cliente com o item selecionado e a quantidade escolhida. • O ator Cliente poderá personalizar seu pedido em um campo de observações.
Pós-condição	O ator Cliente adicionou ou removeu com sucesso itens ao carrinho de compras.
Cenário Alternativo	3a. Se o ator Cliente desejar remover um item do carrinho, ele acessa o carrinho de compras e seleciona a opção de remover o item específico. 3b. Se a quantidade desejada pelo Cliente exceder a disponibilidade do item, o sistema notifica o Cliente e permite que ele ajuste a

	quantidade ou escolha outro item.

Caso de Uso – Fazer *login*	
ID	UC 003
Descrição	Este caso de uso descreve o processo em que os atores Cliente e Funcionário da Casa de Carnes podem fazer *login* no sistema para acessar suas respectivas funcionalidades.
Ator Primário	Usuário do sistema
Pré-condição	O sistema estar operacional.
Cenário Principal	• O ator acessa a tela de *login* na interface do sistema. • O ator insere suas credenciais de *login*, como nome de usuário e senha. • O sistema valida as credenciais fornecidas pelo ator. • Se as credenciais forem válidas, o sistema autentica o ator e concede acesso a outras funcionalidades do sistema, como finalizar pedido.
Pós-condição	O ator fez *login* com sucesso no sistema e pode acessar as funcionalidades disponíveis para seu papel (Cliente ou Funcionário da Casa de Carnes).
Cenário Alternativo	3a. Se as credenciais fornecidas pelo ator forem inválidas, o sistema exibe uma mensagem de erro e solicita que o ator insira credenciais válidas. 3b. Se o ator esquecer suas credenciais, ele pode selecionar a opção de recuperação de senha para redefini-las.

Caso de Uso – Gerenciar estoque	
ID	UC 004
Descrição	Este caso de uso descreve o processo em que o ator Funcionário da Casa de Carnes pode gerenciar o estoque de produtos disponíveis na casa de carnes através do sistema.
Ator Primário	Funcionário da Casa de Carnes
Pré-condição	Sistema estar operacional e o ator Funcionário da Casa de Carnes estar autenticado.
Cenário Principal	• O Funcionário da Casa de Carnes acessa a funcionalidade de gerenciamento de estoque no sistema. • O sistema exibe a lista de produtos disponíveis no estoque da casa de carnes, juntamente com suas quantidades atuais. • O Funcionário da Casa de Carnes atualiza as informações do estoque conforme necessário, incluindo adicionar novos produtos, atualizar quantidades disponíveis ou remover produtos do estoque. • O sistema registra as alterações feitas no estoque e atualiza a base de dados.
Pós-condição	O Funcionário da Casa de Carnes gerenciou com sucesso o estoque de produtos disponíveis na casa de carnes.
Cenário Alternativo	3a. Se um novo produto for adicionado ao estoque, o Funcionário da Casa de Carnes deve fornecer as informações relevantes do produto, como nome, descrição, categoria e quantidade inicial. 3b. Se um produto for removido do estoque, o sistema deve confirmar a ação e atualizar a base de dados.

Caso de Uso – Verificar credenciais	
ID	UC 005
Descrição	Este caso de uso descreve o processo em que o sistema verifica as credenciais fornecidas por um usuário durante o processo de *login*.
Ator Primário	Sistema
Pré-condição	O sistema estar operacional.
Cenário Principal	• O sistema recebe as credenciais fornecidas por um usuário durante o processo de *login*. • O sistema valida as credenciais de acordo com as informações armazenadas em seu banco de dados. • Se as credenciais forem válidas, o sistema autentica o usuário e concede acesso às funcionalidades apropriadas. • Se as credenciais forem inválidas, o sistema nega o acesso e exibe uma mensagem de erro ao usuário.
Pós-condição	O sistema verificou com sucesso as credenciais fornecidas pelo usuário durante o processo de *login* e concedeu o acesso correspondente às credenciais utilizadas.
Cenário Alternativo	3a. Se o sistema detectar múltiplas tentativas de *login* com credenciais inválidas, ele pode bloquear temporariamente o acesso e notificar o administrador do sistema.

Caso de Uso – Exibir Erro de *Login*	
ID	UC 006
Descrição	Este caso de uso descreve o processo pelo qual o sistema exibe uma mensagem de erro ao usuário quando as credenciais fornecidas durante o processo de *login* são inválidas.
Ator Primário	Sistema
Pré-condição	O sistema recebeu credenciais inválidas durante o processo de *login*.
Cenário Principal	• O sistema verifica as credenciais fornecidas pelo usuário durante o processo de *login*. • Se as credenciais forem inválidas, o sistema exibe uma mensagem de erro na tela de *login*. • A mensagem de erro informa ao usuário que as credenciais fornecidas são inválidas e fornece orientações sobre como proceder.
Pós-condição	O sistema exibiu com sucesso uma mensagem de erro ao usuário informando que as credenciais fornecidas durante o *login* são inválidas.
Cenário Alternativo	Nenhum

Caso de Uso – Verificar Compra	
ID	UC 007

Descrição	Este caso de uso descreve o processo pelo qual o ator Cliente verifica o status de uma compra realizada no sistema.
Ator Primário	Cliente
Pré-condição	O sistema estar operacional e o Cliente estar autenticado.
Cenário Principal	• O Cliente acessa a funcionalidade de histórico de compras no aplicativo móvel ou na interface do sistema. • O sistema exibe uma lista das compras previamente realizadas pelo Cliente, incluindo informações como data, status e detalhes do pedido. • O Cliente seleciona a compra específica que deseja verificar. • O sistema exibe os detalhes completos da compra selecionada, incluindo informações sobre os produtos, quantidade, preço total e status do pedido.
Pós-condição	O Cliente verificou com sucesso o status de uma compra realizada no sistema.
Cenário Alternativo	Nenhum

Caso de Uso – Acessar Histórico	
ID	UC 008
Descrição	Este caso de uso descreve o processo pelo qual o ator Cliente acessa o histórico de suas interações ou transações anteriores no sistema.
Ator Primário	Cliente
Pré-condição	O sistema estar operacional e o Cliente estar autenticado.
Cenário Principal	• O Cliente acessa a funcionalidade de histórico no aplicativo móvel ou na interface do sistema. • O sistema exibe uma lista das transações ou interações anteriores do Cliente, como compras, pedidos, pagamentos, entre outros. • O Cliente pode navegar pela lista e selecionar uma transação específica para obter mais detalhes. • O sistema exibe detalhes completos da transação selecionada, incluindo informações relevantes como data, tipo de transação, produtos envolvidos, status e outras informações pertinentes.
Pós-condição	O Cliente acessou com sucesso seu histórico de interações ou transações no sistema.
Cenário Alternativo	Nenhum

Caso de Uso – Realizar Pedido	
ID	UC 009
Descrição	Este caso de uso descreve o processo pelo qual o ator Cliente realiza um pedido de produtos através do sistema.
Ator Primário	Cliente
Pré-condição	O sistema estar operacional e o Cliente estar autenticado.

Cenário Principal	• O Cliente acessa o menu de produtos disponíveis no aplicativo móvel ou na interface do sistema. • O Cliente navega pelo menu para encontrar os produtos desejados. • O Cliente seleciona os produtos desejados e especifica a quantidade para cada item. • O Cliente revisa o carrinho de compras para garantir que os produtos e quantidades estejam corretos. • O Cliente confirma o pedido e escolhe o método de pagamento desejado. • O sistema processa o pedido, registra a transação e envia uma confirmação de pedido para o Cliente.
Pós-condição	O Cliente realizou com sucesso um pedido de produtos através do sistema, e a transação foi registrada no sistema.
Cenário Alternativo	4a. Se o Cliente desejar, ele pode remover itens do carrinho de compras ou ajustar as quantidades antes de confirmar o pedido. 5a. Se o Cliente escolher pagamento online, ele será redirecionado para o sistema de pagamento para concluir a transação. 6a. Se houver algum problema durante o processamento do pedido, o sistema exibirá uma mensagem de erro e fornecerá orientações sobre como proceder.

Caso de Uso – Escolher Método de Entrega	
ID	UC 010
Descrição	Este caso de uso descreve o processo pelo qual o ator Cliente escolhe o método de entrega para o pedido realizado através do sistema.
Ator Primário	Cliente
Pré-condição	O sistema estar operacional e o Cliente estar autenticado.
Cenário Principal	• Após confirmar os itens no carrinho de compras e antes de finalizar o pedido, o Cliente acessa a opção de escolha de método de entrega. • O sistema exibe as opções de métodos de entrega disponíveis, que podem incluir entrega domiciliar e retirada no local. • O Cliente seleciona o método de entrega desejado para o pedido atual. • Se o Cliente escolher entrega domiciliar, ele pode inserir o endereço de entrega desejado. • O Cliente confirma a seleção do método de entrega e continua para finalizar o pedido.
Pós-condição	O Cliente selecionou com sucesso o método de entrega desejado para o pedido, e a escolha foi registrada no sistema.
Cenário Alternativo	3a. Se o Cliente escolher entrega domiciliar, mas o endereço fornecido estiver fora da área de cobertura da entrega, o sistema exibirá uma mensagem de aviso e oferecerá opções alternativas, como retirada no local. 4a. Se o Cliente optar por retirada no local, o sistema pode fornecer informações adicionais, como horário de funcionamento e instruções para retirada.

Caso de Uso – Receber Requisição do Cliente	
ID	UC 011
Descrição	Este caso de uso descreve o processo pelo qual o ator Funcionário da Casa de Carnes recebe a requisição de um cliente para preparar e finalizar um pedido.
Ator Primário	Funcionário da Casa de Carnes
Pré-condição	O sistema estar operacional e o Funcionário da Casa de Carnes estar autenticado.

Cenário Principal	• O Funcionário da Casa de Carnes é notificado sobre uma nova requisição de um cliente, seja por meio de uma notificação no sistema ou outro meio de comunicação. • O Funcionário da Casa de Carnes acessa as informações detalhadas do pedido recebido, incluindo os produtos solicitados, a quantidade, o método de entrega selecionado pelo cliente e quaisquer instruções adicionais fornecidas. • O Funcionário da Casa de Carnes verifica a disponibilidade dos produtos solicitados no estoque. • Se todos os produtos estiverem disponíveis, o Funcionário da Casa de Carnes prepara o pedido de acordo com as especificações do cliente. • Após a preparação do pedido, o Funcionário da Casa de Carnes finaliza o pedido no sistema e encaminha para o próximo estágio do processo, que pode incluir a entrega ou notificação ao cliente para retirada no local.
Pós-condição	O Funcionário da Casa de Carnes recebeu e processou com sucesso a requisição do cliente, preparando e finalizando o pedido de acordo com as especificações fornecidas.
Cenário Alternativo	3a. Se algum produto estiver temporariamente indisponível, o Funcionário da Casa de Carnes pode entrar em contato com o cliente para oferecer alternativas ou informar sobre o atraso na entrega.

Caso de Uso – Confirmar Requisição do Cliente	
ID	UC 012
Descrição	Este caso de uso descreve o processo pelo qual o ator Funcionário da Casa de Carnes confirma a requisição de um cliente após receber e verificar os detalhes do pedido.
Ator Primário	Funcionário da Casa de Carnes
Pré-condição	O sistema estar operacional e o Funcionário da Casa de Carnes estar autenticado.
Cenário Principal	• O Funcionário da Casa de Carnes recebe uma requisição de um cliente, contendo os detalhes do pedido. • O Funcionário da Casa de Carnes verifica os detalhes do pedido para garantir sua precisão e disponibilidade de estoque. • Após verificar os detalhes, o Funcionário da Casa de Carnes confirma a requisição no sistema, indicando que o pedido está em processo de preparação. • O sistema registra a confirmação da requisição e atualiza o status do pedido para "Em preparo".
Pós-condição	O Sistema confirmou com sucesso a requisição do cliente, indicando que o pedido está sendo preparado e está em processo de execução.
Cenário Alternativo	2a. Se houver algum problema com os detalhes do pedido, como produtos indisponíveis ou especificações incompletas, o Funcionário da Casa de Carnes pode entrar em contato com o cliente para esclarecimentos ou sugestões de alterações.

Caso de Uso – Notificar Pedido Aceito para o Cliente	
ID	UC 013
Descrição	Este caso de uso descreve o processo pelo qual o sistema notifica o cliente de que o pedido foi aceito e está em processo de preparação.
Ator Primário	Sistema
Pré-condição	O sistema estar operacional e o Funcionário da Casa de Carnes confirmou o pedido.

Cenário Principal	• Após a confirmação do pedido pelo Funcionário da Casa de Carnes, o sistema gera automaticamente uma notificação para o cliente informando que o pedido foi aceito. • A notificação é enviada para o cliente através do aplicativo móvel ou por e-mail, dependendo das preferências de comunicação do cliente. • A notificação contém informações sobre o status atual do pedido, incluindo a confirmação de que foi aceito, o tempo estimado de preparação e quaisquer outras informações relevantes.
Pós-condição	O sistema notificou com sucesso o cliente de que o pedido foi aceito e está em processo de preparação.
Cenário Alternativo	1a. Se o sistema enfrentar problemas técnicos ao enviar a notificação automática, ele pode tentar reenviar a notificação em intervalos regulares até que seja entregue com sucesso.

Caso de Uso – Atualizar Status do Pedido	
ID	UC 014
Descrição	Este caso de uso descreve o processo pelo qual o ator Funcionário da Casa de Carnes atualiza o status de um pedido no sistema, fornecendo informações atualizadas sobre o progresso do pedido.
Ator Primário	Funcionário da Casa de Carnes
Pré-condição	O sistema estar operacional e o Funcionário da Casa de Carnes estar autenticado.
Cenário Principal	• O Funcionário da Casa de Carnes acessa a lista de pedidos em processamento no sistema. • O Funcionário da Casa de Carnes seleciona o pedido que deseja atualizar o status. • O Funcionário da Casa de Carnes atualiza o status do pedido no sistema para refletir o progresso atual, como "Em preparo", "Pronto para entrega", "Entregue" ou outros status relevantes. • O sistema registra a atualização do status do pedido e notifica automaticamente o cliente sobre a mudança de status, se aplicável.
Pós-condição	O Funcionário da Casa de Carnes atualizou com sucesso o status do pedido no sistema, fornecendo informações atualizadas sobre o progresso do pedido.
Cenário Alternativo	3a. Se ocorrerem atrasos ou problemas inesperados no processo de preparação ou entrega do pedido, o Funcionário da Casa de Carnes pode atualizar o status do pedido para refletir essas circunstâncias e notificar o cliente sobre o atraso.

Caso de Uso – Gerenciar Lista de Pedidos Recebidos	
ID	UC 015
Descrição	Este caso de uso descreve o processo pelo qual o ator Funcionário da Casa de Carnes gerencia a lista de pedidos recebidos no sistema, incluindo visualização, seleção e execução de ações sobre os pedidos.
Ator Primário	Funcionário da Casa de Carnes
Pré-condição	O sistema estar operacional e o Funcionário da Casa de Carnes estar autenticado.

Cenário Principal	• O Funcionário da Casa de Carnes acessa a lista de pedidos recebidos no sistema. • O sistema exibe uma lista de todos os pedidos pendentes, incluindo informações como número do pedido, hora do pedido, itens solicitados e status atual. • O Funcionário da Casa de Carnes pode visualizar os detalhes de um pedido selecionado para obter mais informações. • O Funcionário da Casa de Carnes pode selecionar um ou mais pedidos para executar ações, como confirmar, atualizar status, cancelar, ou outras ações relevantes. • Após executar as ações necessárias, o sistema registra as mudanças e atualizações realizadas nos pedidos.
Pós-condição	O Funcionário da Casa de Carnes gerenciou com sucesso a lista de pedidos recebidos, realizando as ações necessárias e mantendo as informações dos pedidos atualizadas no sistema.
Cenário Alternativo	3a. Se houver muitos pedidos na lista, o Funcionário da Casa de Carnes pode utilizar filtros ou ferramentas de busca para encontrar pedidos específicos. 4a. Se ocorrerem atrasos ou problemas com um pedido, o Funcionário da Casa de Carnes pode entrar em contato com o cliente para fornecer atualizações ou resolver problemas.

Caso de Uso – Cancelar Pedido	
ID	UC 016
Descrição	Este caso de uso descreve o processo pelo qual o ator Cliente cancela um pedido realizado através do sistema, desde que o pedido não esteja em processo de preparo.
Ator Primário	Cliente
Pré-condição	O sistema estar operacional, o Cliente estar autenticado e o pedido não estar em processo de preparo.
Cenário Principal	• O Cliente acessa a lista de pedidos realizados no sistema. • O Cliente identifica o pedido que deseja cancelar e verifica seu status atual. • Se o status do pedido permitir o cancelamento (ou seja, não estiver em processo de preparo), o Cliente seleciona a opção de cancelar pedido. • O sistema solicita uma confirmação do Cliente antes de processar o cancelamento do pedido. • Após a confirmação do Cliente, o sistema registra o cancelamento do pedido e atualiza o status para "Cancelado". • O sistema notifica o Cliente sobre o cancelamento bem-sucedido do pedido.
Pós-condição	O Cliente cancelou com sucesso o pedido no sistema, e o status do pedido foi atualizado para refletir o cancelamento.
Cenário Alternativo	3a. Se o pedido estiver em processo de preparo, o sistema exibe uma mensagem informando que o pedido não pode ser cancelado neste momento devido ao status atual. O Cliente pode ser orientado a entrar em contato com a equipe de atendimento ao cliente para obter assistência adicional.

Fonte: Autores

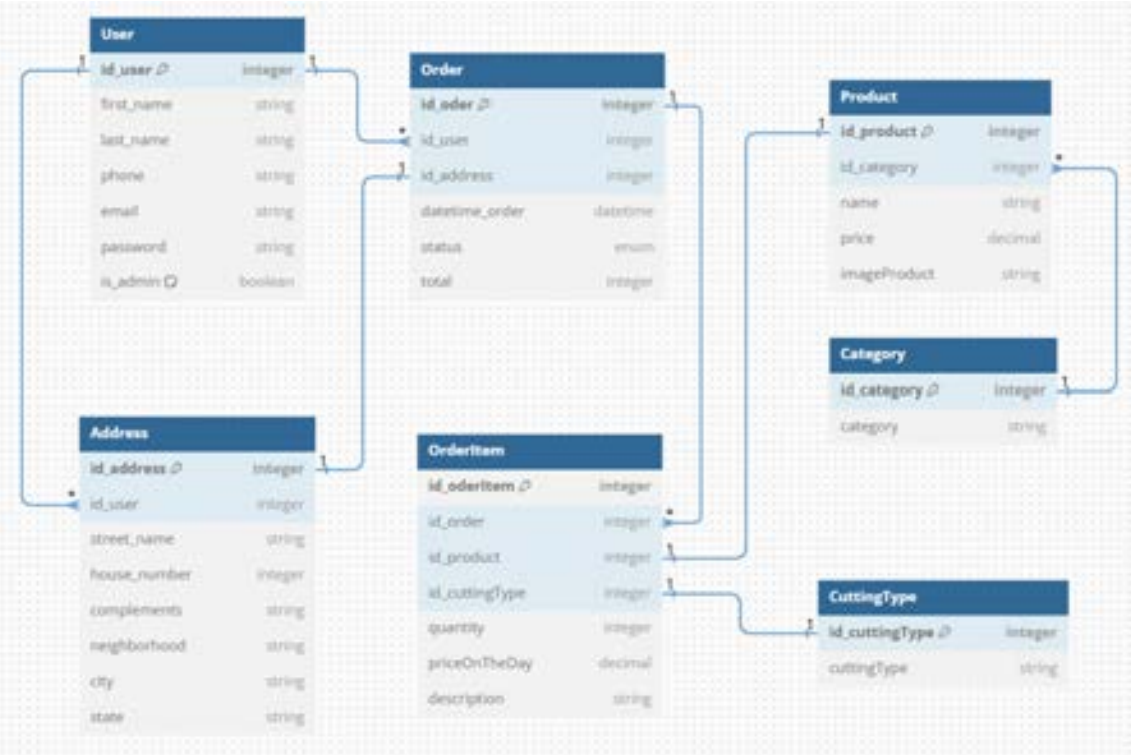
3.5 Diagrama Entidade-Relacionamento

O Diagrama Entidade-Relacionamento (DER), como demonstra a figura 6, é uma representação gráfica que modela a estrutura lógica de um banco de dados, descrevendo as entidades, os atributos e os relacionamentos entre elas.

Conforme Elmasri e Navathe (2011), o DER utiliza símbolos específicos para representar as entidades (objetos ou conceitos do mundo real) e seus relacionamentos, proporcionando uma visão clara e concisa de como os dados estão interligados. As entidades são tipicamente representadas por retângulos, enquanto os relacionamentos são ilustrados por linhas ou losangos conectando as entidades.

A importância do DER está em sua capacidade de simplificar o entendimento do modelo de dados e facilitar a comunicação entre os desenvolvedores, analistas e *stakeholders*. Segundo Coronel e Morris (2016), o DER é crucial no processo de design de bancos de dados, pois garante que as relações entre os dados sejam bem compreendidas antes de sua implementação. Ele ajuda a identificar potenciais problemas, como redundâncias ou inconsistências, além de servir como um guia para a construção do banco de dados físico. Em termos práticos, o DER permite que o sistema de informação seja construído de forma eficiente e organizada, melhorando o desempenho e a integridade dos dados.

Figura 6 – DER



Fonte: Autores

4 Ferramentas e Métodos

As ferramentas e os métodos aplicados na criação deste projeto estão detalhados a seguir.

4.1 Ferramentas

A seleção das ferramentas utilizadas no projeto foi baseada em sua escalabilidade, eficiência e no suporte da comunidade. A escolha também considerou a familiaridade da equipe com essas ferramentas, bem como a experiência anterior no uso delas. Por serem de código aberto, são gratuitas e podem ser empregadas tanto em projetos comerciais quanto pessoais. Além disso, possuem documentação e que facilitam a aprendizagem e implementação das soluções pelos desenvolvedores.

4.1.1 Diagramas - Draw.io

O Draw.io é uma ferramenta gratuita de design online, usada para criar diagramas de fluxo, diagramas UML, organogramas, e outros tipos de diagramas. Sua interface intuitiva permite criar diagramas complexos com facilidade. O Draw.io oferece a possibilidade de colaborar em tempo real, além de integração com serviços de armazenamento como Google Drive, Dropbox e OneDrive. Ele suporta a exportação de diagramas em diversos formatos, incluindo PNG, SVG, PDF e muito mais, sendo amplamente utilizado por equipes para mapeamento de processos e documentação técnica. (Draw.io, 2024)

4.1.2 Canvas - Canva

O Canva é uma plataforma de *design* gráfico que permite a criação de uma variedade de materiais visuais, como protótipos, *banners*, apresentações, e *layouts* para redes sociais. Sua principal vantagem é a simplicidade e acessibilidade, permitindo que pessoas sem habilidades avançadas em *design* possam criar gráficos profissionais usando uma vasta biblioteca de *templates*, imagens, ícones e fontes. O Canva também oferece suporte à colaboração em equipe, onde múltiplos usuários podem trabalhar em um mesmo projeto simultaneamente. (Canva, 2024)

4.1.3 Documentação - Microsoft Word

O Microsoft Word é um dos processadores de texto mais populares no mundo, amplamente utilizado para criação de documentos formais. Parte do pacote Microsoft 365, o Word oferece funcionalidades avançadas, como revisão ortográfica, sugestões de edição e recursos de colaboração em tempo real, onde múltiplos usuários podem editar e comentar documentos simultaneamente. O armazenamento automático no OneDrive garante que as alterações sejam salvas automaticamente na nuvem, proporcionando acessibilidade de qualquer dispositivo. (Microsoft Word, 2024)

4.1.4 Compartilhamento de Documentos - Notion.so

O Notion.so é uma ferramenta completa de produtividade que combina funcionalidades de gerenciamento de tarefas, escrita e organização de informações em uma única plataforma. Ele permite a criação de documentos colaborativos, que podem ser compartilhados com outros membros da equipe. Com recursos como blocos de notas, listas de tarefas, calendários e bases de dados, o Notion se destaca pela flexibilidade, oferecendo uma maneira eficiente de gerenciar a documentação do projeto e a colaboração entre os desenvolvedores. (Notion, 2024)

4.1.5 DBDiagrams.io

O DBDiagrams.io é uma ferramenta online gratuita usada para criação de diagramas de entidade-relacionamento (DER). Permite a criação rápida de esquemas de banco de dados, com suporte a diversas linguagens de modelagem, incluindo SQL. A ferramenta possibilita o compartilhamento e a colaboração em tempo real, o que facilita o trabalho em equipe e a revisão de estruturas de banco de dados. (DBDiagrams.io, 2024)

4.1.6 Desenvolvimento

4.1.6.1 Visual Studio Code (VSCode)

O Visual Studio Code (VSCode) é um editor de código-fonte leve, mas poderoso, desenvolvido pela Microsoft. Ele é amplamente utilizado por desenvolvedores por oferecer suporte a uma grande variedade de linguagens de programação, como JavaScript, Python, C++ e outras. O VSCode conta com uma vasta biblioteca de extensões, que podem ser instaladas para adicionar funcionalidades como depuração de código, controle de versão com Git, e *auto-complete* de código. A flexibilidade e o suporte à personalização

tornam o VSCode uma escolha popular entre desenvolvedores. (VSCode, 2024)

4.1.6.2 Node.js

O Node.js é um ambiente de execução de JavaScript que permite executar código no lado do servidor. Baseado no motor V8 do Google Chrome, o Node.js se destaca pela sua alta performance em operações de I/O e pela escalabilidade, tornando-o ideal para o desenvolvimento de aplicações web em tempo real, como APIs e micro-serviços. Com uma vasta comunidade de desenvolvedores e uma biblioteca robusta de pacotes disponíveis através do npm (Node *Package Manager*), o Node.js é uma escolha eficiente para o desenvolvimento *back-end* de sistemas modernos. (Node.js, 2024)

4.1.6.3 JavaScript (JS)

O JavaScript é uma das linguagens de programação mais populares do mundo, utilizada tanto no desenvolvimento *front-end* quanto *back-end* de aplicações web. No lado do cliente, o JavaScript é responsável por tornar as páginas web interativas, manipulando o DOM (Document Object Model). No lado do servidor, com ferramentas como Node.js, o JavaScript pode ser utilizado para construir APIs, gerenciar bancos de dados e processar dados do servidor. Sua ampla adoção e suporte em todos os principais navegadores tornam o JavaScript uma escolha essencial para o desenvolvimento web. (JavaScript, 2024)

4.1.6.4 React

O React é uma biblioteca JavaScript de código aberto criada pelo Facebook, que permite a construção de interfaces de usuário eficientes e reutilizáveis. O principal conceito do React é a criação de componentes, que são blocos independentes que podem ser combinados para construir interfaces dinâmicas e interativas. A renderização eficiente, por meio do Virtual DOM, torna o React ideal para o desenvolvimento de SPAs (Single Page Applications), onde o desempenho e a experiência do usuário são críticos. (React, 2024)

4.1.6.5 Material-UI (MUI)

O Material-UI (MUI) é uma biblioteca de componentes para React que implementa o design system Material Design, criado pela Google. O MUI oferece uma ampla variedade de componentes de interface, como botões, diálogos, tabelas e formulários, que seguem as diretrizes de design do Material Design. Essa biblioteca facilita a criação de interfaces de usuário atraentes e consistentes, otimizando o desenvolvimento front-end ao fornecer componentes prontos para uso. (Material-UI, 2024)

4.1.6.6 Supabase

O Supabase é uma plataforma de código aberto que oferece uma solução completa de *back-end* com funcionalidades como banco de dados PostgreSQL, autenticação de usuários, armazenamento de arquivos e APIs em tempo real. Ele se destaca pela simplicidade de implementação, permitindo que os desenvolvedores criem rapidamente *back-ends* escaláveis sem precisar configurar servidores ou sistemas complexos. O Supabase é frequentemente comparado ao Firebase, mas com a vantagem de ser totalmente *open-source.* (Supabase, 2024)

4.1.6.7 Prisma

O Prisma é um ORM (Object-Relational Mapping) moderno que permite a manipulação de bancos de dados relacionais em aplicações Node.js. Ele facilita a comunicação entre o código JavaScript e o banco de dados, permitindo a criação de consultas otimizadas e seguras. O Prisma é compatível com diversos bancos de dados, incluindo PostgreSQL, MySQL e SQLite, e é conhecido por sua facilidade de uso e integração com o TypeScript, além de gerar automaticamente um modelo de dados que reflete o estado atual do banco. (Prisma, 2024)

4.2 Métodos

Os artefatos desenvolvidos ao longo do projeto seguiram métodos estruturados para assegurar clareza, organização e alinhamento com os objetivos da solução. A escolha dos métodos foi guiada por sua capacidade de melhorar a eficiência e a transparência no processo de desenvolvimento, conforme detalhado abaixo.

4.2.1 TAP (Termo de Abertura do Projeto)

O Termo de Abertura do Projeto (TAP) foi utilizado para formalizar os objetivos principais, definir o escopo e identificar as partes interessadas. Esse documento inicial serviu como uma base de referência para a equipe, garantindo que todos os envolvidos estivessem alinhados quanto aos resultados esperados e aos critérios de sucesso do projeto.

4.2.2 BPMN (*Business Process Model and Notation*)

A modelagem dos processos de negócios foi realizada utilizando BPMN, uma notação padronizada que permitiu a visualização clara e detalhada das operações e fluxos de trabalho. Através dessa ferramenta, foram identificadas oportunidades para otimização e melhorias nos processos, assegurando que as interações entre usuários e o sistema fossem representadas de maneira precisa e eficiente.

4.2.3 Requisitos Funcionais

Os requisitos funcionais foram levantados e documentados para definir com precisão as funcionalidades que o sistema deveria oferecer. Esse processo foi essencial para garantir que o desenvolvimento estivesse alinhado com as expectativas dos usuários e que todas as funcionalidades necessárias fossem implementadas conforme planejado, sem desvios ou lacunas.

4.2.4 Documentação e Diagramação de Casos de Uso

A documentação e a diagramação dos casos de uso foram realizadas para descrever as interações entre os diferentes tipos de usuários e o sistema. Esse método permitiu identificar todas as possíveis ações e cenários, ajudando a garantir que o sistema fosse desenvolvido para atender às necessidades reais dos usuários, com um foco específico na clareza e na cobertura completa de todos os fluxos.

4.2.5 Portabilidade

Durante o desenvolvimento, a portabilidade do sistema foi considerada para assegurar que a solução pudesse ser executada em diferentes plataformas e ambientes, como sistemas operacionais distintos e dispositivos variados. Esse cuidado visou ampliar a flexibilidade e a acessibilidade do sistema, tornando-o utilizável em uma gama diversificada de contextos tecnológicos.

4.2.6 Métricas

Métricas foram definidas para monitorar e avaliar o desempenho do sistema em termos de usabilidade, confiabilidade, segurança e eficiência. Esses indicadores de desempenho foram acompanhados ao longo do desenvolvimento e após a implementação, garantindo que a solução atendesse aos padrões de qualidade esperados e permitisse ajustes proativos quando necessário.

Todos esses artefatos e métodos desempenharam um papel importante no processo de desenvolvimento, fornecendo estrutura, documentação e direcionamento, para garantir a entrega da solução de acordo com as necessidades dos *stakeholders* e os objetivos do projeto.

5 Desenvolvimento

O desenvolvimento de *software* envolve não apenas a criação do código, mas também a integração de práticas que garantem a qualidade, segurança e eficiência do sistema. Desde a análise inicial até a implementação final, cada fase é projetada para alinhar o produto às necessidades do usuário e às metas do projeto, enquanto se mantém flexível para futuras melhorias. Esse processo é essencial para que o *software* seja escalável, adaptável e capaz de acompanhar mudanças tecnológicas e novas demandas do mercado.

5.1 *Back-end*

O *back-end* é a parte do sistema que opera nos servidores e lida com a lógica de negócios, processamento de dados e integração com o banco de dados. Ele é responsável por receber, processar e enviar dados de forma segura e eficiente para o *front-end*.

5.1.1 PrismaDB

O PrismaDB é configurado para conectar o projeto ao banco de dados PostgreSQL, definindo um mapeamento ORM que facilita o gerenciamento dos dados e possibilita a criação de queries de forma segura e padronizada.

O arquivo "*schema.prisma*" contém a configuração do Prisma, com dois elementos principais: o "*generator*", que especifica o cliente Prisma em

JavaScript, e o "*datasource*", que conecta a aplicação ao banco de dados. Em seguida, definimos os modelos que representam as tabelas, como "*Product*", "*Category*", "*User*", "*Address*", "*Order*", entre outros, que formam a base de dados.

O modelo "*Product*" contém informações dos produtos, como "*name*", "*price*", "*imageProduct*", "*category*", "*availability*" e o relacionamento com o modelo "*Category*" por meio de "*id_category*". Essa relação permite vincular cada produto a uma categoria específica, utilizando o campo "*category*" para associá-los conforme definido no banco de dados.

O modelo "*User*" armazena dados dos usuários com os campos "*first_name*", "*last_name*", "*phone*", "email", "*password*" e "*is_admin*", que sinaliza se o usuário possui permissões de administrador. Além disso, ele possui relacionamentos com "*Address*" e "*Order*", que permitem conectar os usuários aos endereços cadastrados e ordens realizadas.

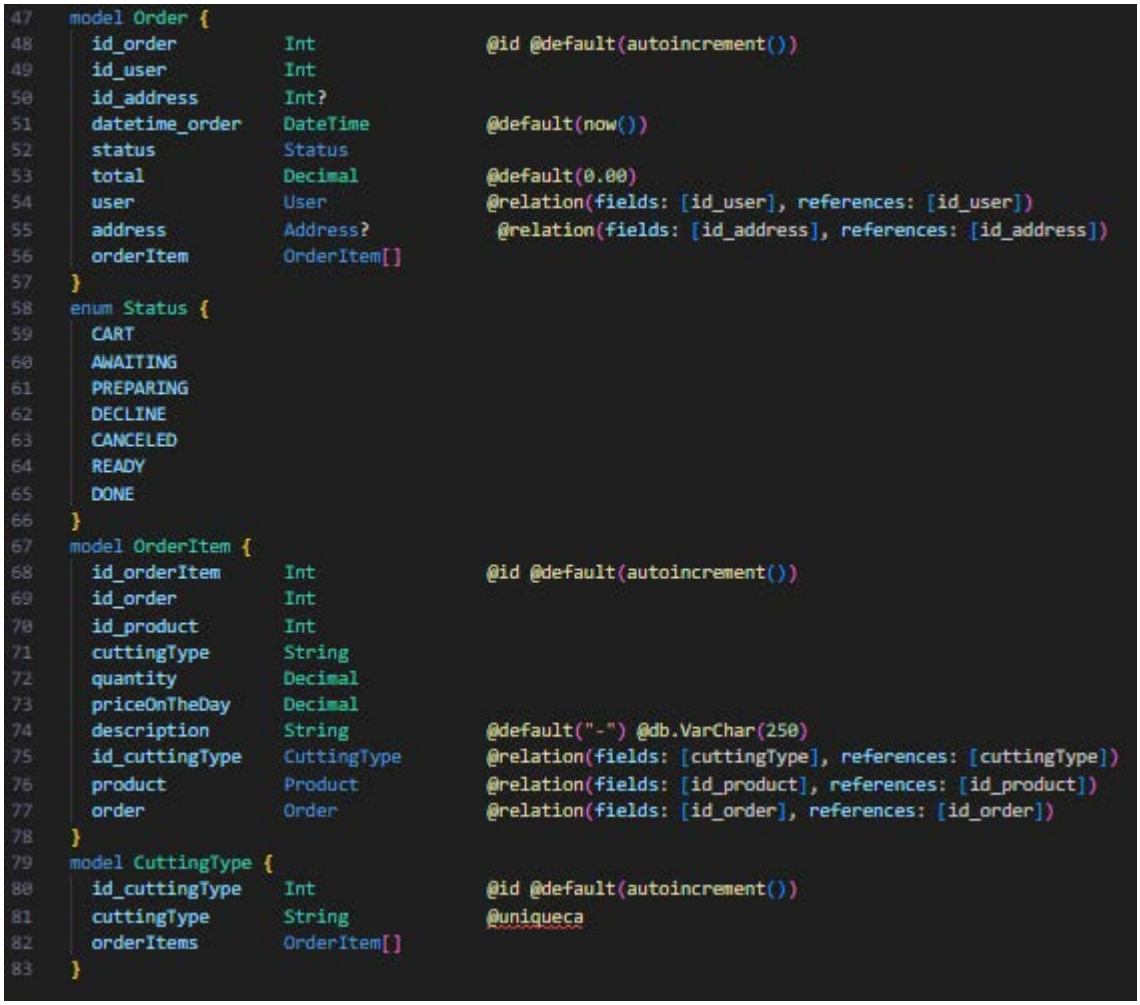
Assim como vemos nas figuras 7 e 8, esse trecho do código contém a configuração inicial e os primeiros modelos.

Figura 7 – Arquivo “*schema.prisma*” parte 1

```
generator client {
  provider = "prisma-client-js"
}
datasource db {
  provider = "postgres"
  url      = env("DATABASE_URL")
}
model Product {
  id_product        Int               @id @default(autoincrement())
  name              String            @unique
  price             Decimal
  imageProduct      String            @unique
  category          String
  availability      Boolean           @default(true)
  id_category       Category          @relation(fields: [category], references: [category])
  orderItem         OrderItem[]
}
model Category {
  id_category       Int               @id @default(autoincrement())
  category          String            @unique

  product           Product[]
}
model User {
  id_user           Int               @id       @default(autoincrement())
  first_name        String
  last_name         String
  phone             String
  email             String            @unique
  password          String
  is_admin          Boolean           @default(false)
  address           Address[]
  order             Order[]
}
model Address {
  id_address        Int               @unique @default(autoincrement())
  id_user           Int
  street_name       String
  house_number      Int
  complements       String            @default("-")
  neighborhood      String
  city              String
  state             String
  user              User              @relation(fields: [id_user], references: [id_user])
  order             Order []
}
```

Fonte: Autores

Figura 8 – Arquivo “*schema.prisma*” parte 2



```
model Order {
  id_order          Int              @id @default(autoincrement())
  id_user           Int
  id_address        Int?
  datetime_order    DateTime         @default(now())
  status            Status
  total             Decimal          @default(0.00)
  user              User             @relation(fields: [id_user], references: [id_user])
  address           Address?          @relation(fields: [id_address], references: [id_address])
  orderItem         OrderItem[]
}
enum Status {
  CART
  AWAITING
  PREPARING
  DECLINE
  CANCELED
  READY
  DONE
}
model OrderItem {
  id_orderItem      Int              @id @default(autoincrement())
  id_order          Int
  id_product        Int
  cuttingType       String
  quantity          Decimal
  priceOnTheDay     Decimal
  description       String           @default("-") @db.VarChar(250)
  id_cuttingType    CuttingType      @relation(fields: [cuttingType], references: [cuttingType])
  product           Product          @relation(fields: [id_product], references: [id_product])
  order             Order            @relation(fields: [id_order], references: [id_order])
}
model CuttingType {
  id_cuttingType    Int              @id @default(autoincrement())
  cuttingType       String           @uniqueca
  orderItems        OrderItem[]
}
```

Fonte: Autores

5.1.2 *Controllers*

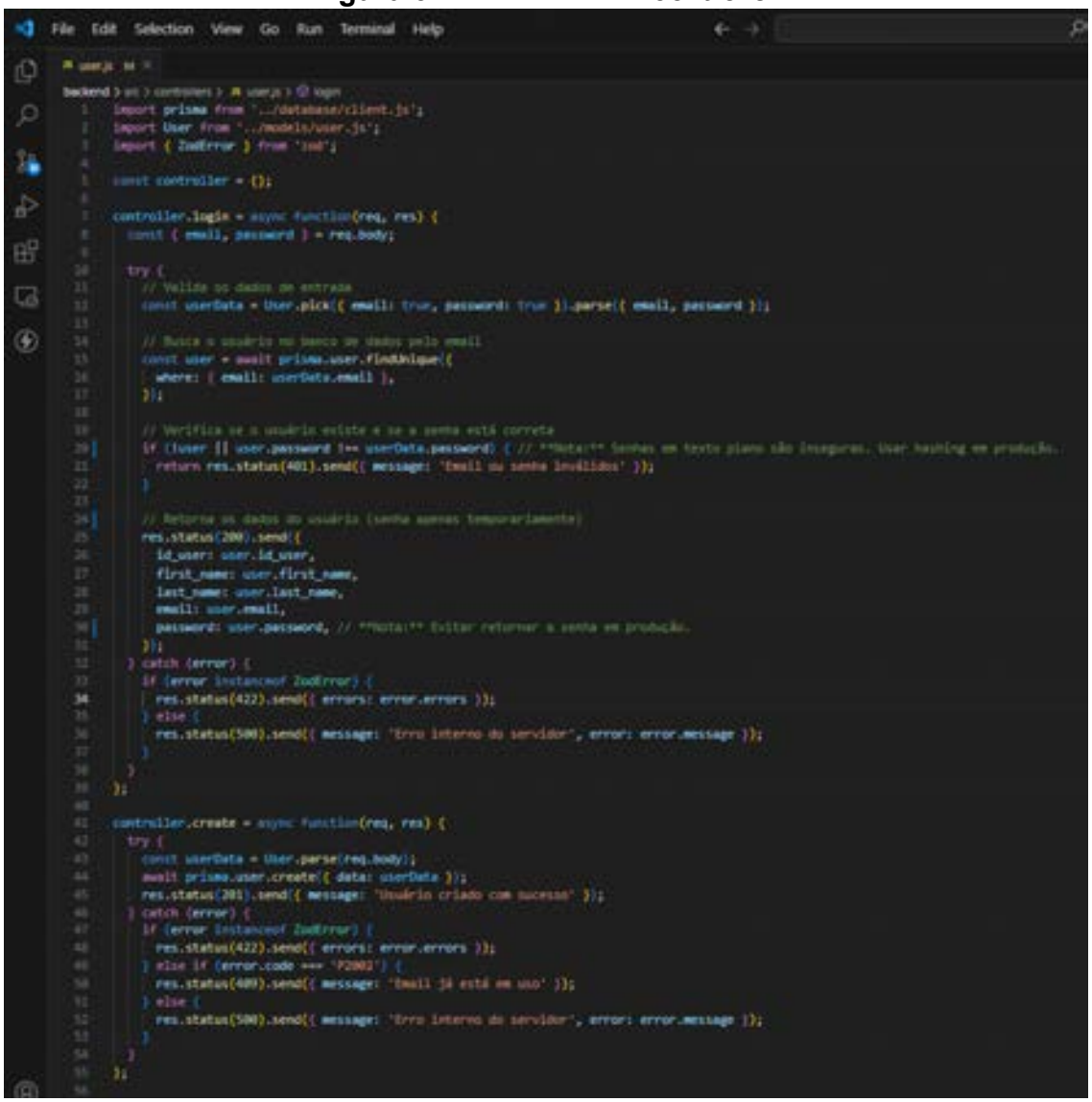
Os *Controllers* servem para organizar e encapsular a lógica de negócios da aplicação. Através deles, são processadas as requisições recebidas, e as respostas são enviadas de volta ao cliente. Cada função dos *controllers* executa uma operação específica, como a criação de um usuário ou a atualização de uma ordem.

O *controller* de usuário, por exemplo, contém as funções "*login*", "*create*", "*retrieveAll*", "*retrieveOne*", "*update*" e "*delete*". A função "*login*" é responsável por autenticar o usuário e garantir que o "email" e "senha" inseridos estejam corretos. Em "*create*", é possível adicionar um novo usuário, validando os dados de entrada e assegurando que o "email" seja único. Já "*retrieveAll*" e "*retrieveOne*" realizam consultas para recuperar todos os

usuários ou apenas um específico, enquanto "*update*" e "delete" permitem a modificação e exclusão de registros, respectivamente.

Como mostra a figura 9, abaixo, o código mostra um exemplo do *controller* que realiza essas operações sobre os dados do usuário, incluindo a validação e o tratamento de erros com a biblioteca "Zod".

Figura 9 –Trecho de um *controller*



Fonte: Autores

5.1.3 Rotas

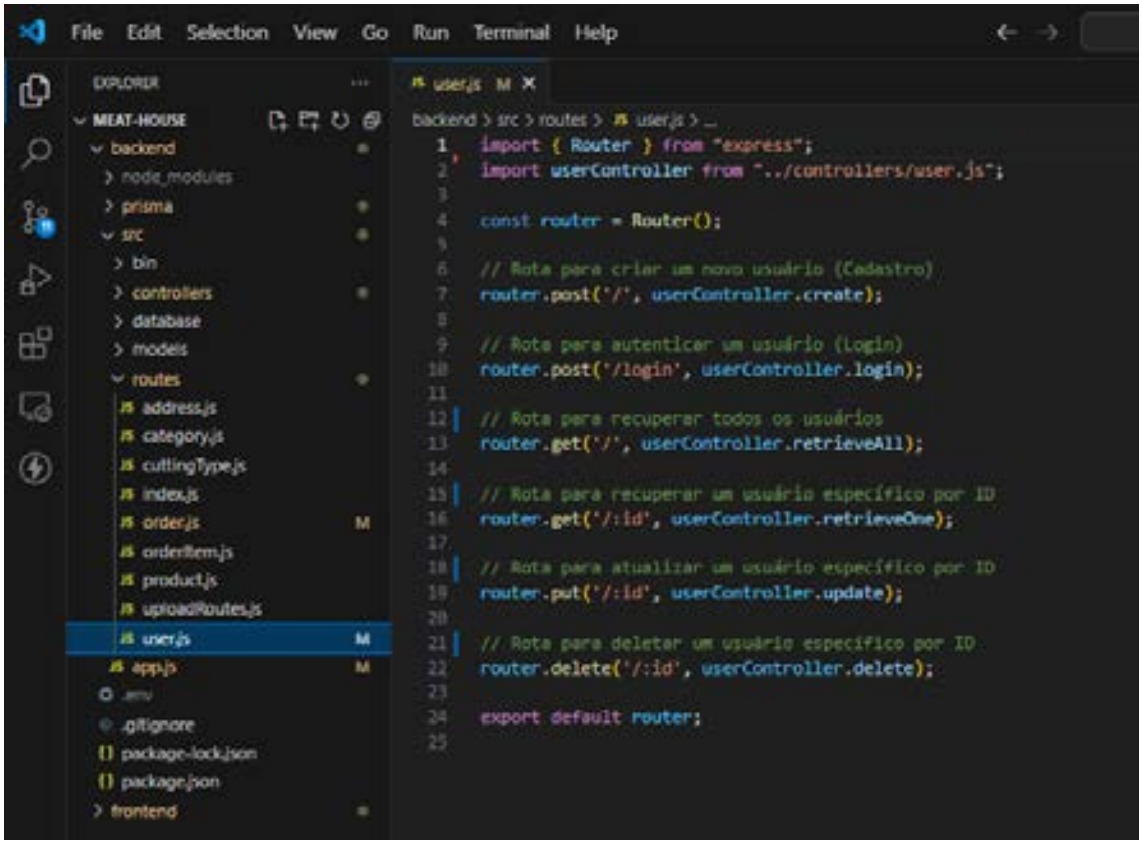
As Rotas definem a URL e método HTTP que será usado para acessar cada função nos *controllers*, sendo o ponto de entrada para todas as

operações na API. Elas organizam os *endpoints* e os associam às funções que manipulam os dados.

Por exemplo, no arquivo de rotas do usuário ("*routes*/*user*.js"), definimos rotas como "*POST* /" para criação de um novo usuário, "*POST* /*login*" para autenticação, "*GET* /" para recuperação de todos os usuários e "*GET* /:id" para recuperação de um usuário específico. As rotas "*PUT* /:id" e "*DELETE* /:id" permitem atualizar e deletar um usuário pelo "ID".

Assim como na figura 10, o código a seguir mostra essas rotas definidas e como cada URL é associada ao método correspondente no controller.

Figura 10 – Exemplo de rotas



Fonte: Autores

5.1.4 App.js

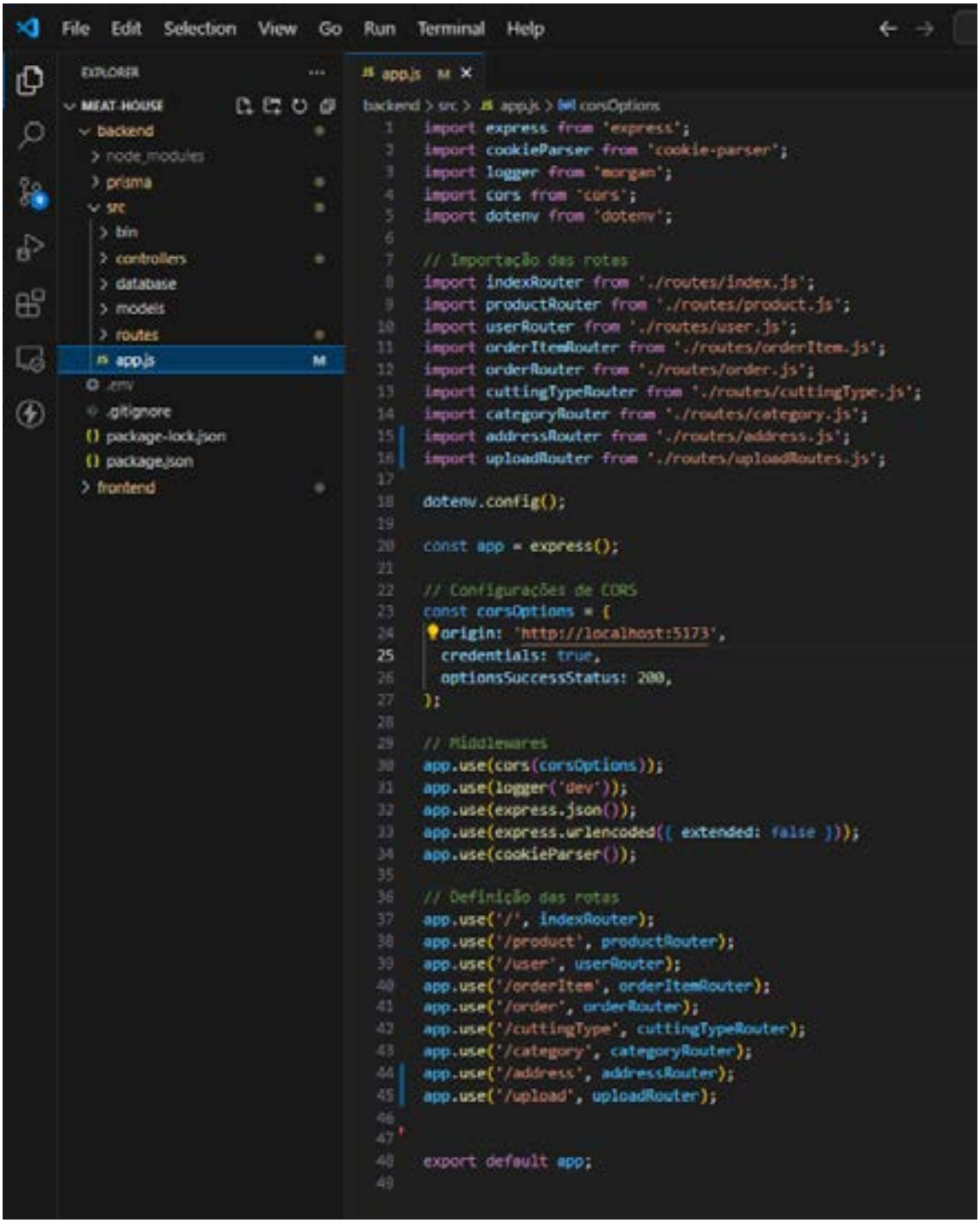
O arquivo "app.js" contém as configurações centrais e é a base do servidor "*Express*". Ele é responsável por carregar os módulos e *middlewares*, como "*express*.json" para interpretar JSON, "*cookieParser*" para manipulação de *cookies*, "cors" para configurações de segurança, e "*logger*" para *logar* as

requisições no servidor. Além disso, as rotas são configuradas, permitindo o acesso a cada *endpoint* da API.

No trecho a seguir, também vemos como o arquivo importa as rotas criadas anteriormente e as associa aos seus respectivos caminhos. Cada recurso principal da aplicação, como "*product*", "*user*", "*orderItem*", entre outros, é representado por uma rota, permitindo uma organização modular da API. O "app.js" também configura "CORS" para permitir a comunicação com o *front-end*, definindo a origem permitida e outros parâmetros.

Como ilustrado na figura 11, o trecho a seguir exibe o início do "app.js", com importações e configurações das rotas.

Figura 11 – Arquivo “App.js”



Fonte: Autores

5.2 *Front-end*

O *front-end* é a parte do sistema que interage diretamente com o usuário e é responsável pela interface visual e pela experiência do usuário. Ele exibe dados, captura interações e envia solicitações ao *back-end*. Essas solicitações

são recebidas e tratadas para que as informações sejam apresentadas ao usuário final para que sejam interpretadas.

5.2.1 Componentes e Funcionalidades

Esses seguintes códigos criam dois componentes importantes para o presente projeto, são eles: "*CartModal*" e "*ProductModal*". Esses componentes são fundamentais para a funcionalidade de um carrinho de compras, onde o usuário pode visualizar, editar e remover produtos do carrinho, além de adicionar novos produtos com detalhes personalizados como o tipo de corte e a quantidade.

5.2.1.1 *CartModal*

O componente *CartModal* é responsável por exibir todos os itens atualmente adicionados ao carrinho de compras. Ele usa *localStorage* para armazenar os dados do carrinho, garantindo que eles estejam disponíveis em sessões futuras e proporcionando persistência, uma vez que o estado dos dados do carrinho é mantido entre atualizações de página. Alguns trechos do código de maior relevância são:

a) *useState*: Define estados para itens do carrinho (*cartItems*), produto selecionado (*selectedProduct*), controle de abertura do modal de produto (*isProductModalOpen*) e carregamento (*loading*)

b) *useNavigate*: Usado para navegação de rota quando o usuário deseja concluir o processo de compra. A figura 12 mostra a estrutura inicial do componente, destacando os *hooks* usados para gerenciar estados e navegação.

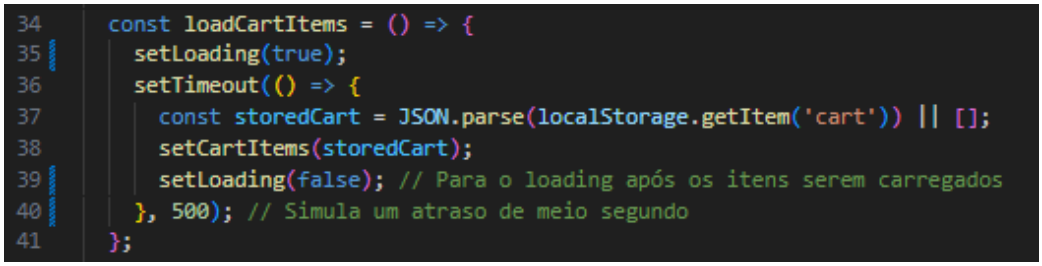
Figura 12 – Inicio do componente

```
25  const [cartItems, setCartItems] = useState([]);
26  const [selectedProduct, setSelectedProduct] = useState(null);
27  const [isProductModalOpen, setIsProductModalOpen] = useState(false);
28  const [loading, setLoading] = useState(true);
29  const navigate = useNavigate();
```

Fonte: Autores

c) *loadCartItems*: Carrega os itens do carrinho a partir do *localStorage* e simula um atraso para *renderizar* um *spinner* de carregamento. A figura 13 demonstra o uso de uma função assíncrona, essencial para garantir que o carregamento dos itens ocorra antes da exibição.

Figura 13 – "*loadCartItems*"



Fonte: Autores

d) *handleRemoveItem*: Remove um item do carrinho e atualiza o *localStorage* em tempo real, garantindo que o estado persistente esteja sincronizado. A figura 14 demonstra como é implementada a lógica do *handleRemoveItem*, *handleEditItem* e *handleUpdateCart*.

e) *handleEditItem*: Abre o *ProductModal* para permitir a edição do item selecionado.

f) *handleUpdateCart*: Atualiza o carrinho quando uma alteração é feita em um item específico e salva as mudanças no *localStorage*.

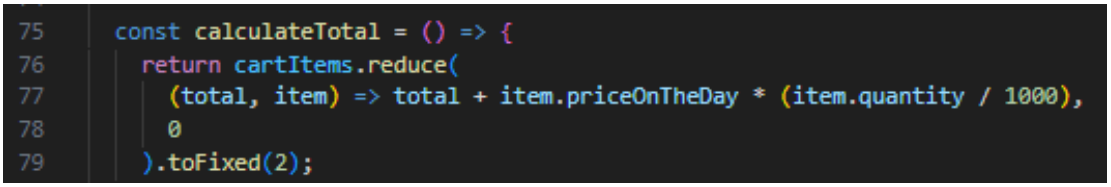
Figura 14 – Lógicas de remoção, adição e edição.

```
const handleRemoveItem = (id_product) => {
  const updatedCart = cartItems.filter(
    (item) => item.product.connect.id_product !== id_product
  );
  setCartItems(updatedCart);
  localStorage.setItem('cart', JSON.stringify(updatedCart));
  notifySuccess('Item removido do carrinho!');
};

const handleEditItem = (product) => {
  setSelectedProduct(product);
  setIsProductModalOpen(true);
};

const handleUpdateCart = (updatedProduct) => {
  const updatedCart = cartItems.map((item) =>
    item.product.connect.id_product === updatedProduct.product.connect.id_product
      ? updatedProduct
      : item
  );
  setCartItems(updatedCart);
  localStorage.setItem('cart', JSON.stringify(updatedCart));
  notifySuccess('Item atualizado no carrinho!');
  setIsProductModalOpen(false);
};
```

Fonte: Autores

g) Calcular total: *calculateTotal* soma o valor total do carrinho com base no preço e quantidade de cada item. Esse cálculo, como apresenta a figura 15, mostra o valor total ao usuário em tempo real.

Figura 15 – "*calculateTotal*"



```
const calculateTotal = () => {
  return cartItems.reduce(
    (total, item) => total + item.priceOnTheDay * (item.quantity / 1000),
    0
  ).toFixed(2);
```

Fonte: Autores

5.2.1.2 *ProductModal*

O *ProductModal* permite ao usuário configurar detalhes do produto, como o tipo de corte, quantidade e uma breve descrição. Esse componente é integrado ao *CartModal*, permitindo ao usuário adicionar produtos personalizados ao carrinho. Seus trechos de maior importância são:

a) *useEffect*: Carrega dados do corte ao abrir o modal e configura os campos de descrição e quantidade caso o item já esteja no carrinho.

O uso do *useEffect*, figura 16, é crítico para sincronizar os dados entre sessões e modais.

Javascript

Figura 16 – "*useEffectt*"

```
42
43  useEffect(() => {
44    if (product && open) {
45      const fetchData = async () => {
46        try {
47          const cutsData = await myfetch.get('cuttingType');
48          setCuts(cutsData);
49        } catch (error) {
50          console.error(error);
51        }
52      };
53
54      fetchData();
55    }
56  }, [product, open]);
57
```

Fonte: Autores

b) *handleAddToCart*, figura 17, adiciona um novo produto ao carrinho ou atualiza o item, salvando-o no *localStorage* para garantir persistência.

Figura 17 – "*handleAddToCart*"

```
const handleAddToCart = async () => {
  if (!selectedCut) {
    notifyError('Por favor, selecione o tipo de corte');
    return;
  }

  try {
    const selectedCuttingType = cuts.find(cut => cut.id_cuttingType === selectedCut);
    const selectedProduct = {
      quantity,
      description,
      priceOnTheDay: product.price,
      cuttingType: {
        connect: { id_cuttingType: selectedCut },
        name: selectedCuttingType ? selectedCuttingType.cuttingType : ''
      },
      product: {
        connect: { id_product: product.id_product },
        name: product.name
      },
    };

    const existingCart = JSON.parse(localStorage.getItem('cart')) || [];
    const existingProductIndex = existingCart.findIndex(
      (item) => item.product.connect.id_product === product.id_product
    );

    if (existingProductIndex !== -1) {
      // Atualiza o produto existente no carrinho
      existingCart[existingProductIndex] = selectedProduct;
      notifySuccess('Item atualizado no carrinho!');
    } else {
      // Adiciona um novo produto ao carrinho
      existingCart.push(selectedProduct);
      notifySuccess('Item adicionado ao carrinho!');
    }

    localStorage.setItem('cart', JSON.stringify(existingCart));
    onAddToCart(selectedProduct);
    resetFields();
  } catch (error) {
    console.error('Error adding item to cart:', error);
    notifyError('Erro ao adicionar item ao carrinho');
  }
};
```

Fonte: Autores

Este código utiliza o *localStorage* para manter a persistência dos itens no carrinho de compras, o que garante que o estado seja mantido em diferentes sessões.

5.3 Telas

A seguir será descrito a interação usuário com o sistema para realização de um pedido até usa finalização.

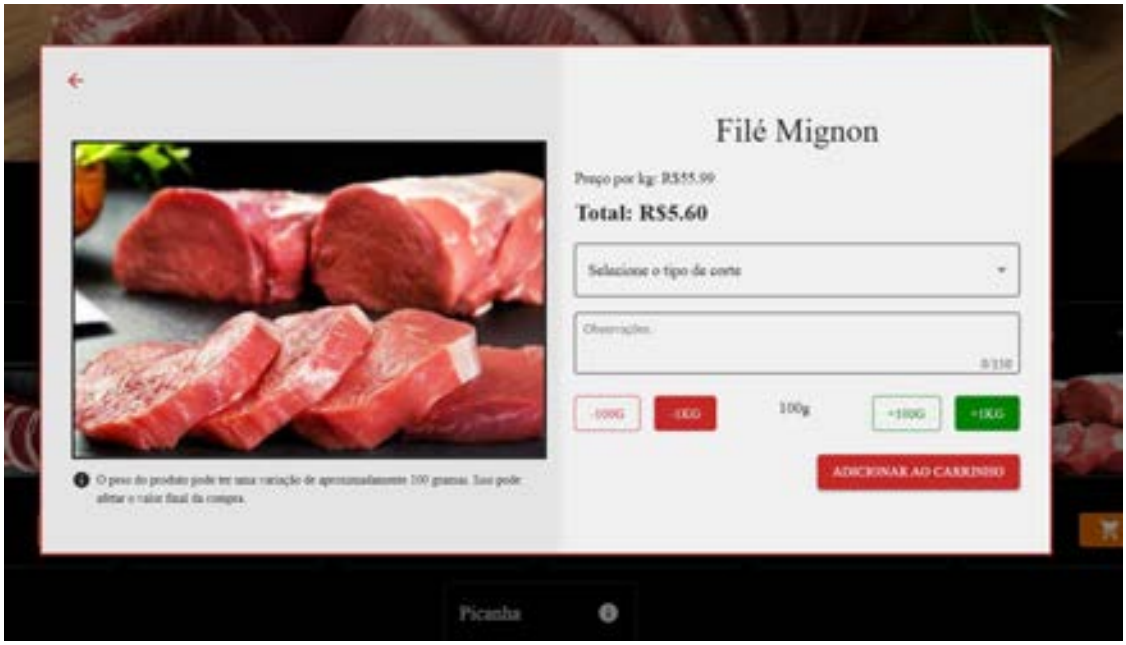
A figura 18 representa a página principal da do projeto, onde os produtos estão organizados por suas categorias, o que facilita navegação e escolha dos clientes. O objetivo do layout é atrair e facilitar a experiência do usuário. Vale ressaltar que as imagens foram geradas por inteligência artificial (IA) e a logo criada pelos autores.

Figura 18 - Tela principal

Fonte: Autores

Caso o usuário selecione algum produto uma modal (figura 19) é aberta para que sejam especificadas a quantidade e a maneira que o cliente deseja que seu seja preparado.

Figura 19 - Modal produto



Fonte: Autores

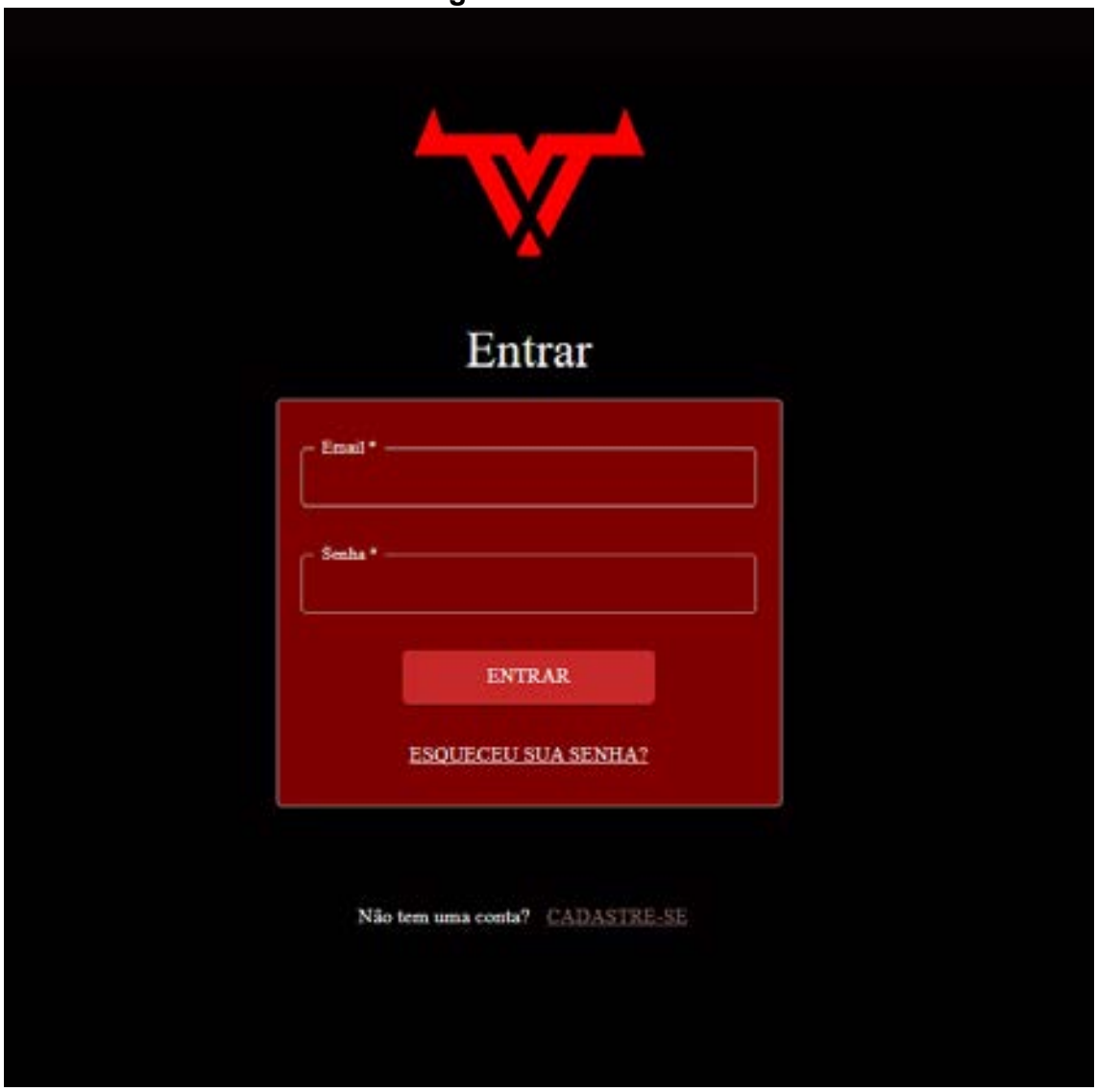
Ao escolher seus produtos o cliente pode modificar seu pedido ou completá-lo através do carrinho, figura 20.

Figura 20 - Carrinho

Fonte: Autores

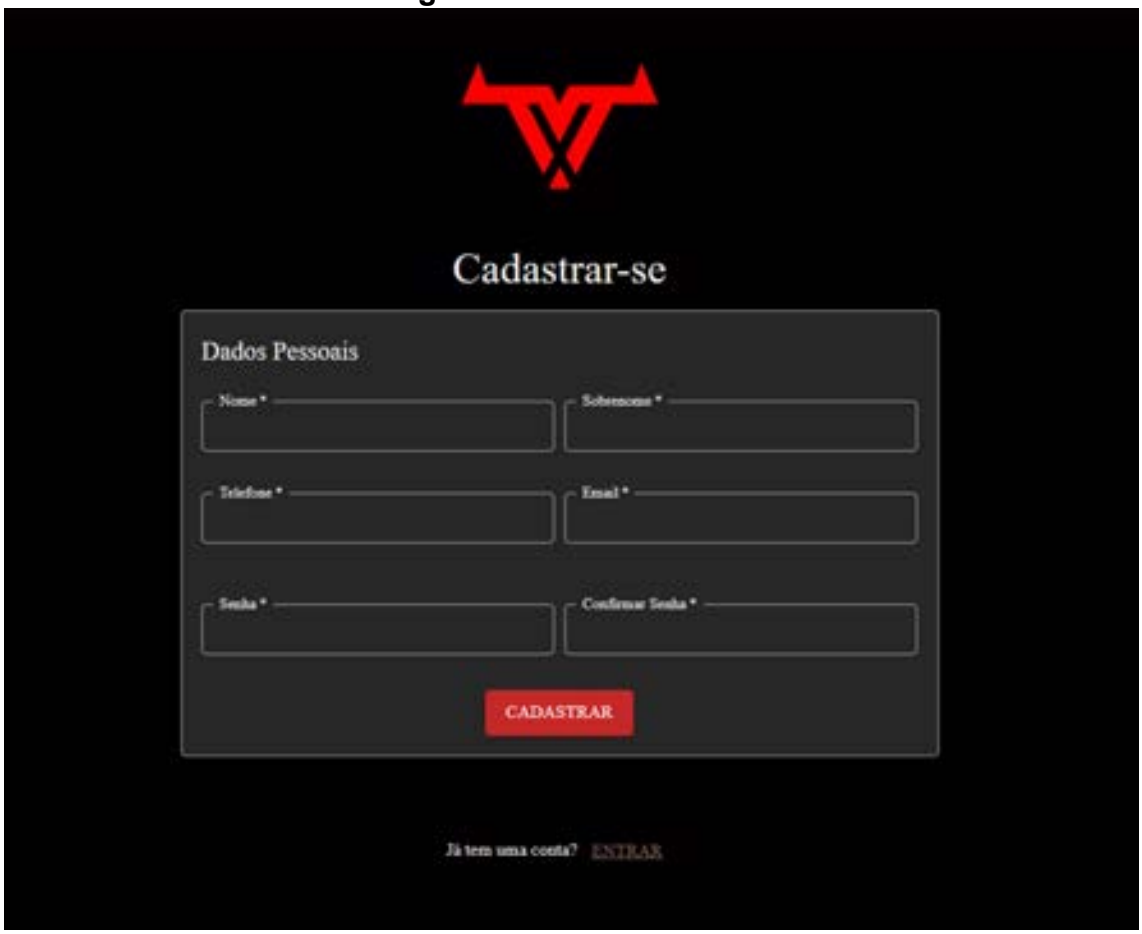
O cliente tem a opção de finalizar sua compra, mas para que essa ação seja concluída o usuário deve realizar um cadastro, caso não possua, como demonstrado nas figuras 21 e 22.

Figura 21 – Entrar



Fonte: Autores

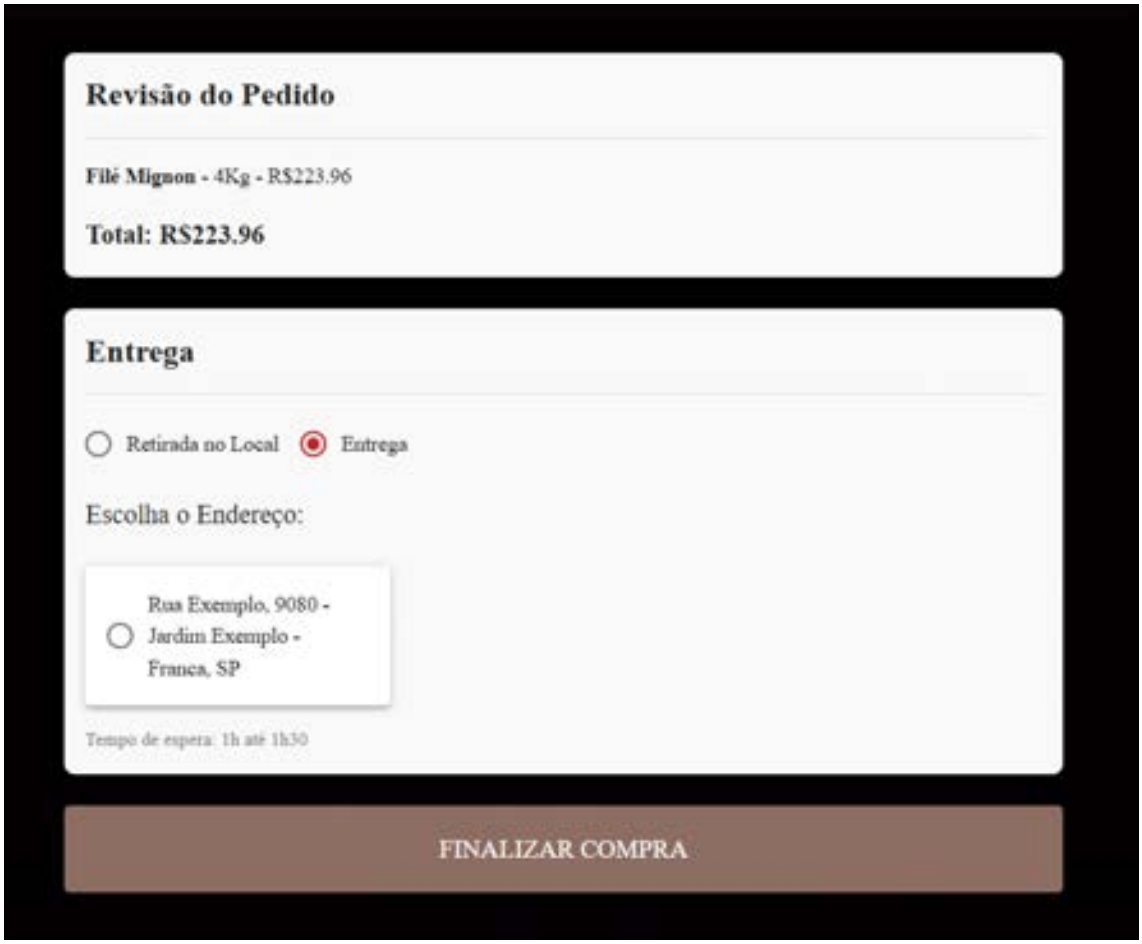
Figura 22 – Cadastrar-se



Fonte: Autores

Após o passo de autenticação o usuário é direcionado para página em que ele pode revisar seu pedido e escolher o método de entrega (figura 23). O cliente tem as opções “Retirar no Local” ou “Entrega”, caso “Entrega” seja selecionada o sistema mostrará todos os endereços cadastrados por ele. Em caso que não exista nenhum endereço o usuário tem a opção de criá-lo para dar prosseguimento no pedido.

Figura 23 – Revisão do pedido



Fonte: Autores

6 Resultados e Discussão

6.1 PORTABILIDADE

A adoção de uma ferramenta digital para casas de carne visa proporcionar uma experiência de compra eficiente, segura e prática para clientes e funcionários, seja em dispositivos móveis ou computadores. A portabilidade do sistema torna-se fundamental para que a ferramenta possa ser usada em diferentes ambientes e dispositivos, ampliando seu alcance e garantindo que o sistema funcione de maneira adequada para atender as demandas de forma otimizada.

6.1.1 Configuração mínima para o sistema de gerenciamento de pedidos

Para garantir a funcionalidade da ferramenta de atendimento e menu remoto em casas de carne, foram definidas configurações mínimas que asseguram o bom desempenho do sistema. Essas especificações permitem o uso eficaz do sistema, evitando lentidão e proporcionando uma experiência satisfatória ao usuário.

a. Processador: Intel Core i3 ou equivalente.
b. Memória RAM: 4 GB.
c. Armazenamento: Pelo menos 128 GB de armazenamento em SSD ou disco rígido.
d. Sistema Operacional: Windows 10, Linux ou sistemas móveis com Android 9.0 ou superior e iOS 12.0 ou superior.
e. Navegador: As versões mais recentes do Google Chrome, Mozilla Firefox, Safari ou Microsoft Edge.
f. Banco de Dados: MySQL ou PostgreSQL.

Essas configurações foram estabelecidas para suportar o volume médio de transações e acessos simultâneos, bem como a execução de operações durante horários de pico. A adoção desses parâmetros minimiza o risco de problemas operacionais e garante uma experiência fluida para o usuário.

6.2 PROPOSTA COMERCIAL

A implementação de um sistema digital para casas de carne visa otimizar o processo de atendimento ao cliente, proporcionando uma experiência satisfatória. Esse sistema automatiza a apresentação do menu de produtos, permitindo que os clientes acessem as opções disponíveis de forma remota, selecionem produtos, e realizem pedidos com praticidade. A ferramenta busca melhorar a organização e o fluxo de pedidos, reduzindo o tempo de espera, agilizando o atendimento e garantindo maior precisão nas operações.

Essa proposta comercial visa atender ao aumento da demanda por soluções tecnológicas no setor alimentício, focando especialmente nas necessidades das casas de carne. A digitalização dos processos representa um diferencial competitivo, atendendo às expectativas dos clientes por uma experiência de compra moderna.

6.2.1 Solução da proposta

A solução proposta é uma ferramenta digital direcionada ao setor de casas de carne, que visa otimizar o atendimento ao cliente e modernizar o acesso ao portfólio de produtos oferecidos. Essa plataforma permitirá que os consumidores consultem o menu de produtos de forma remota, façam pedidos online e agendem a retirada ou entrega dos itens, proporcionando uma experiência de atendimento mais ágil e eficiente. Com essa abordagem, A ferramenta contribui para que o estabelecimento amplie seu alcance geográfico, permitindo a conquista de uma base de clientes mais diversificada e o atendimento a consumidores que, de outra forma, poderiam enfrentar dificuldades de acesso físico.

A solução se propõe a transformar a experiência de compra, oferecendo conveniência e flexibilidade ao consumidor, que pode acessar o menu a qualquer hora e de qualquer local, e se planejar de acordo com suas necessidades. A possibilidade de agendamento para retiradas ou entregas dos produtos adiciona uma camada de praticidade ao processo de compra, com foco na redução de tempo de espera e na satisfação do cliente. Esse diferencial de atendimento contribui para que as casas de carne se posicionem competitivamente no mercado.

Considerando os desafios logísticos que envolvem o transporte de produtos perecíveis, a ferramenta digital incorpora práticas específicas para assegurar a qualidade e a segurança dos alimentos, desde o armazenamento até o momento da entrega. Para viabilizar o funcionamento da plataforma de forma eficaz e segura, a solução demanda investimentos em tecnologias adequadas e na capacitação de funcionários. Com esses requisitos, a implementação da ferramenta se alinha com as necessidades atuais de inovação e digitalização no setor, e visa contribuir para a modernização do atendimento e para uma experiência de consumo mais prática e satisfatória.

Considerações finais

O objetivo inicial deste trabalho foi desenvolver uma solução digital para melhorar a gestão e o atendimento em uma casa de carne, proporcionando aos clientes uma experiência prática e intuitiva de consulta aos produtos, realização de pedidos e escolha entre opções de entrega ou retirada. Com objetivo de

modernizar, a ferramenta atende ao crescimento da demanda por conveniência e praticidade, permitindo que o cliente visualize o cardápio de produtos disponíveis e finalize o pedido, acompanhando o status do mesmo até o recebimento.

Ao longo do desenvolvimento, a aplicação foi estruturada de modo a integrar funcionalidades como a exibição e atualização do catálogo de produtos e o controle de pedidos desde o recebimento até a finalização. A ferramenta ainda possibilita que os funcionários acompanhem as solicitações em tempo real para o gerenciamento de diferentes estágios dos pedidos, o que contribui para a organização interna do estabelecimento. A gestão do sistema ocorre por meio de um painel idealizado para simplificar a organização dos pedidos.

Embora o pagamento online fosse uma funcionalidade planejada na concepção do projeto, ela foi deixada para futuras atualizações, assim como o lançamento de promoções diretamente na plataforma e a coleta de *feedback* dos clientes para aprimoramento contínuo do serviço. A solução demonstrou potencial de expansão, tanto em termos de funcionalidades — como a introdução de controle de estoque e gerenciamento de entrada e saída de produtos — quanto em novas áreas, como a ampliação do catálogo para incluir itens complementares à venda de carnes. A organização da interface de usuário também foi um ponto de atenção, buscando sempre uma navegação intuitiva, capaz de oferecer uma experiência positiva aos clientes da casa de carne.

Entre os principais desafios enfrentados, destaca-se a necessidade de adaptar o protótipo a um ambiente de produção, garantindo a robustez e segurança necessárias para operações comerciais. A execução local da aplicação permitiu simulações completas do processo de compra, mas há planos de expandir o projeto para uma infraestrutura de produção que suporte maior volume de acessos e transações, consolidando o sistema como uma solução confiável e escalável para o mercado.

No aspecto técnico, o desenvolvimento do protótipo envolveu a utilização de ferramentas como *React* e Material-UI, que garantiram uma interface moderna. A integração com o Prisma permitiu um gerenciamento eficiente dos dados, facilitando o controle das entidades envolvidas, como produtos, pedidos, categorias, etc.

Para o futuro, o projeto visa implementar a integração com métodos de pagamento online diretamente na plataforma, proporcionando mais praticidade aos clientes. Há também planos para a criação de funcionalidades avançadas de análise e relatório, que oferecerão ao gestor da casa de carne informações estratégicas sobre as vendas e o comportamento dos consumidores e adição de um sistema de controle de estoque, visando o registro de entradas e saídas de produtos e reduzindo o desperdício que resultaria em prejuízo para o estabelecimento.

Este trabalho ofereceu novas experiências, tanto no que diz respeito ao desenvolvimento de soluções tecnológicas aplicadas ao varejo quanto na organização de processos e atendimento a requisitos de negócios específicos. O projeto contribuiu para fortalecer habilidades em gestão de tempo, estudo contínuo, consistência nas entregas, trabalho em equipe e resiliência frente aos desafios, refletindo práticas e responsabilidades frequentemente encontradas no mercado de trabalho.

Assim, o desenvolvimento desta ferramenta digital para a casa de carne representa um avanço significativo para o setor, oferecendo uma solução prática para consumidores e contribuindo para a eficiência e competitividade do estabelecimento. A modernização do setor de carnes se torna uma realidade com este projeto, que almeja não apenas o aprimoramento do atendimento ao cliente, mas também a integração e digitalização de processos operacionais, adaptando-se às demandas atuais e abrindo caminho para um futuro sustentável e digitalmente conectado.

Referências

CANVA. Canva. 2024. Disponível em: https://www.canva.com. Acesso em: 03/11/2024.

CODEPROJETS. Visual representation of SQL joins. 10/01/2015. Disponível em: http://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins. Acesso em: 05/10/2015.

COCKBURN, Alistair. **Writing Effective Use Cases**. Boston: Addison-Wesley, 2001.

CORONEL, Carlos; MORRIS, Steven. **Database Systems: Design, Implementation, & Management**. 12. ed. Boston: Cengage Learning, 2016.

DATE, C. J. **Introdução a sistemas de banco de dados**. 8. ed. Rio de Janeiro: Elsevier, 2003.

DBDIAGRAMS.IO. DBDiagrams.io. 2024. Disponível em: https://dbdiagrams.io. Acesso em: 03/11/2024.

DE PAULA MENEZES, Vinícius. **O impacto do e-commerce no comportamento do consumidor brasileiro**. Revista Brasileira de Marketing, São Paulo, v. 17, n. 2, p. 103-119, 2018.

DRAW.IO. Draw.io. 2024. Disponível em: https://draw.io. Acesso em: 03/11/2024.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson, 2011.

HARARI, Yuval Noah. **Sapiens: uma breve história da humanidade**. São Paulo: Companhia das Letras, 2013. p. 240-245.

IBICT. Instituto Brasileiro de Informação em Ciência e Tecnologia. **Bibliografia Brasileira de Ciência da Informação: 2004/2006**. Brasília: IBICT, 2007.

JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. **The Unified Software Development Process**. Reading, MA: Addison-Wesley, 1999.

KERZNER, Harold. **Project Management: A Systems Approach to Planning, Scheduling, and Controlling**. 12. ed. New York: John Wiley & Sons, 2017.

KOTONYA, Gerald; SOMMERVILLE, Ian. **Requirements Engineering: Processes and Techniques**. New York: John Wiley & Sons, 1998.

MICROSOFT WORD. Microsoft Word. 2024. Disponível em: https://www.microsoft.com. Acesso em: 03/11/2024.

NODE.JS. Node.js. 2024. Disponível em: https://nodejs.org. Acesso em: 03/11/2024.

NOTION.SO. Notion.so. 2024. Disponível em: https://www.notion.so. Acesso em: 03/11/2024.

OBJECT MANAGEMENT GROUP. Business Process Model and Notation (BPMN) Version 2.0. 2011. Disponível em: https://www.omg.org/spec/BPMN/2.0. Acesso em: 03/11/2024.

OSTERWALDER, Alexander; PIGNEUR, Yves. **Business Model Generation: Inovação em Modelos de Negócios**. Rio de Janeiro: Alta Books, 2011.

PINHO, Francisco Vinicius Bezerra de. **Desafios logísticos e de qualidade no e-commerce de produtos perecíveis**. 2021. 132 f. Trabalho de Conclusão de

Curso (Graduação em Logística) – Faculdade de Administração e Negócios, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2021.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2010.

PROJECT MANAGEMENT INSTITUTE. **Um guia do conhecimento em gerenciamento de projetos** (Guia PMBOK®). 6. ed. Pennsylvania: Project Management Institute, 2017.

PRISMA. Prisma. 2024. Disponível em: https://www.prisma.io. Acesso em: 03/11/2024.

REACT. React. 2024. Disponível em: https://react.dev. Acesso em: 03/11/2024.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

SUPABASE. Supabase. 2024. Disponível em: https://supabase.com. Acesso em: 03/11/2024.

VSCODE. Visual Studio Code (VSCode). 2024. Disponível em: https://code.visualstudio.com. Acesso em: 03/11/2024.

WIEGERS, Karl. **Software Requirements**. 2. ed. Redmond: Microsoft Press, 2003.

WHITE, Stephen A. **Introduction to BPMN**. IBM Cooperation, 2004. Disponível em: https://www.omg.org/bpmn. Acesso em: 03/11/2024.

APÊNDICE A – DESCRIÇÃO CASA DE CARNES BOI GORDO

1. CASA DE CARNES BOI GORDO

A Casa de Carnes Boi Gordo é uma empresa local focada em fornecer cortes de carne de alta qualidade para a comunidade, atendendo às necessidades de clientes que buscam produtos frescos, selecionados e adequados para diferentes tipos de receitas e ocasiões. Com uma equipe qualificada e comprometida, a Casa de Carnes Boi Gordo busca garantir a satisfação dos clientes por meio de um atendimento personalizado, produtos rigorosamente inspecionados e uma experiência de compra confiável e conveniente.

2.1 MISSÃO, VISÃO E VALORES

2.1.2 Missão

Oferecer aos nossos clientes carnes de alta qualidade, frescas e cuidadosamente selecionadas, proporcionando uma experiência de compra completa e personalizada que valoriza a segurança alimentar e a satisfação do cliente.

2.2.2 Visão

Ser a referência em qualidade e atendimento no setor de carnes da região, reconhecida por nossos produtos frescos e serviços que atendem às expectativas e necessidades dos consumidores com excelência e inovação.

2.2.3 Valores

a. Qualidade e frescor – Garantia de produtos sempre frescos e de procedência confiável.
b. Atendimento personalizado – Valorização de cada cliente, com um atendimento próximo e atencioso.
c. Compromisso com a segurança alimentar – Rigor nos processos de manipulação e armazenamento dos produtos.
d. Inovação e adaptação – Implementação de novas tecnologias e práticas para melhorar continuamente a experiência de compra.
e. Respeito ao meio ambiente – Práticas sustentáveis para reduzir o impacto ambiental e apoiar a produção responsável.
f. Valorização da comunidade local – Contribuição para o desenvolvimento da economia local, oferecendo produtos e empregos de qualidade.

APÊNDICE B – ENTREVISTA

PERGUNTAS DA ENTREVISTA	RESPOSTAS DA ENTREVISTA
a. O que você quer que o sistema faça?	Quero um sistema para meus clientes fazerem pedidos pela internet
b. Quais tipos de produtos estarão disponíveis?	Apenas as carnes que vendemos

c. Quais tipos de animais que vocês trabalham?	Atualmente trabalhamos com bovinos, suínos e frango, mas nosso objetivo é expandir e trabalhar com outros tipos de animais também, como ovino e caprino.
d. Como é o processo desde o recebimento do pedido até sua finalização?	Nós ouvimos e anotamos o que o cliente ter a dizer, levamos até o açougueiro que verifica e prepara o pedido, a pessoa que preparou o pedido nos chama e nós fazemos a finalização do pedido, que é o pagamento.
e. Como é feito o controle de estoque atualmente?	Os açougueiros organizam os cortes de maneira que sempre sabem o que ainda possuímos em estoque apenas numa rápida ida ao freezer.
f. Quais os tipos de pagamentos aceitos?	Dinheiro, pix, cartões de crédito, débito, e alimentação.
g. Como será feito o cadastro de usuários? Qualquer indivíduo pode se cadastrar ou apenas vocês podem cadastrar os clientes que vem aqui presencialmente?	Qualquer um pode se cadastrar, desde que tenha no mínimo o número de telefone para entrar em contato caso aconteça algum problema.
h. Caso alguem deseje cancelar o pedido, como vocês gostaria que o sistema lidasse com isso?	O sistema pode mostrar o número daqui para o cliente ligar e falar diretamente com a gente, assim temos uma chance de renegociar o pedido dele.
i. Geralmente em quanto tempo um pedido leva para ser preparado?	Leva em torno de cinco a quinze minutos, talvez mais dependendo da quantidade de pessoas na fila.

j. Vamos mudar um pouco a maneira de como são realizadas as perguntas, eu vou sugerir como será o sistema e você e você me diz se estou entendendo corretamente. O sistema deverá possuir um catálogo de produtos para os clientes escolherem e realizaram um pedido?	Sim.
k. Vocês, funcionários, terão um local onde possam inserir e modificar os tipos de produtos disponíveis no local?	Sim
l. Vocês, funcionários, terão uma interface de recebimento de pedidos, simulando uma fila de clientes?	Sim.
m. Vocês, funcionários, realizarão o controle de estoque pelo sistema?	Não. O controle de estoque não será necessário no momento.
n. Os clientes poderão realizar o pagamento diretamente pelo sistema?	Não, o peso do produto pode variar, então o preço também pode. O cliente só saberá o real preço quando o produto estiver pronto.
o. O peso pode variar em aproximadamente quantas gramas pra mais e pra menos?	Em média, por volta de cem gramas.
p. Quantos endereços um cliente pode ter?	No máximo três endereços
q. Alguem sem cadastro pode realizar um pedido pelo	Não, apenas os cadastrados.

sistema?	
r. Quanto tempo eu posso sugerir para entrega e retirada do produto?	Pode ser quarenta minutos para retirada e uma hora e meia para entrega.