

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

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**HUGO HENRIQUE LOURENÇO
JOÃO VICTOR MENEZES LEAL**

VITRINE VIRTUAL

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. Alexandre Gomes da Silva

**FRANCA/SP
2025**

VITRINE VIRTUAL

Hugo Henrique Lourenço¹
João Victor Menezes Leal²

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação para supermercados, com o objetivo de facilitar as compras online por meio de uma vitrine virtual que permite aos clientes navegarem por produtos, gerenciar a lista de compras personalizada. A proposta surgiu da análise de deficiências em aplicativos existentes, como catálogos incompletos, divergências de preços e falta de funcionalidades intuitivas, que prejudicam a experiência do usuário. O projeto foi estruturado em etapas de iniciação, planejamento, execução, monitoramento e finalização, utilizando técnicas como análise SWOT e matriz 5W2H para priorizar a automatização de listas de compras, incluindo recursos como criação, edição e salvamento de múltiplas listas. Foram elaborados artefatos como diagramas BPMN e UML, além de protótipos de telas. As principais restrições, como tempo limitado e capacitação da equipe, foram gerenciadas com metodologias ágeis e ferramentas como Figma e StarUML. A solução busca beneficiar tanto os clientes, oferecendo praticidade, quanto os supermercados, ampliando suas vendas. O cronograma, executado em cinco meses, com potencial para expansão mobile e para demais ramos de vendas e varejo além de mercados.

Palavras-chave: Comércio eletrônico. Experiência do usuário. Lista de compras. Metodologias ágeis. Supermercados.

Abstract

This work presents the development of an application for supermarkets, aimed at facilitating online shopping through a virtual showcase that allows customers to browse products and manage personalized shopping lists. The proposal emerged from an analysis of deficiencies in existing applications, such as incomplete catalogs, price discrepancies, and lack of intuitive features, which impair user experience. The project was structured into stages of initiation, planning, execution, monitoring, and completion, using techniques like SWOT analysis and the 5W2H matrix to prioritize the automation of shopping lists, including features such as creating, editing, and saving multiple lists. Artifacts such as BPMN and UML diagrams were developed, along with screen prototypes. Key constraints, including limited time and team expertise, were managed through agile methodologies and tools like Figma and StarUML. The solution aims to benefit both customers, by offering convenience, and supermarkets, by expanding their sales. The five-month development schedule includes potential for mobile expansion and adaptation to other retail sectors beyond supermarkets.

Keywords: E-commerce. User experience. Shopping list. Agile methodologies. Supermarkets.

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

² Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: joao.leal4@fatec.sp.gov.br

1 Introdução

No cenário contemporâneo, o comércio eletrônico tornou-se uma parte integral do cotidiano das pessoas, proporcionando conveniência e facilidade nas compras, abrangendo diversos setores como alimentação, ferramentas, vestuário e dispositivos tecnológicos. Em resposta a essa tendência crescente, o projeto "Vitrine Virtual" visa desenvolver uma aplicação web para supermercados, funcionando como uma plataforma digital para facilitar as compras online.

A proposta de desenvolvimento da "Vitrine Virtual" surgiu da análise e identificação de deficiências significativas em aplicativos de compras online existentes no mercado, especialmente na cidade de Franca, interior de São Paulo, onde poucos estabelecimentos locais possuem sistemas de vendas online realmente eficientes. As principais fragilidades observadas nesses aplicativos incluíam:

- Catálogos incompletos de produtos.
- Divergências de preços em relação às lojas físicas.
- Falta de funcionalidades intuitivas, o que prejudica a experiência do usuário.
- Falhas na comunicação entre o supermercado e o cliente.
- Atrasos na entrega dos pedidos.
- Dificuldades no processamento de pagamentos.
- Ausência de funcionalidades básicas, como a possibilidade de salvar listas de compras.

Uma pesquisa realizada com *stakeholders* por meio de formulários e a aplicação de técnicas de elicitación de requisitos, como a análise SWOT, corroboraram essas deficiências. A matriz SWOT, em particular, revelou vulnerabilidades significativas em quatro aspectos principais: a logística de entregas, as opções de pagamento e a curva de aprendizado do usuário. Essas fragilidades impactavam diretamente a eficiência operacional dos supermercados e a experiência do cliente.

A "Vitrine Virtual" surge como uma solução abrangente para abordar e superar essas deficiências, com o objetivo principal de simplificar a experiência do usuário e, ao mesmo tempo, ampliar as vendas para os estabelecimentos. Para isso, o projeto se focou na automatização e personalização das listas de compras, definida como uma funcionalidade essencial para a solução sistêmica.

Entre as funcionalidades essenciais oferecidas pela "Vitrine Virtual", destacam-se:

- Navegação por uma ampla gama de produtos e visualização de seus detalhes.
- Criação, edição, salvamento e gerenciamento de múltiplas listas de compras personalizadas, incluindo a opção de renomeá-las e adicionar/remover itens.
- Um processo de *checkout* simples e seguro.
- Opção de receber os pedidos em casa ou retirá-los diretamente na loja.
- Acompanhamento do status dos pedidos em tempo real para os clientes.
- Para os funcionários do supermercado, a plataforma oferece recursos para atualizar a lista de produtos na base de dados, listar, aceitar, informar sobre a separação e notificar o andamento dos pedidos.

Desenvolvida com base em metodologias ágeis e técnicas de engenharia de software, a "Vitrine Virtual" foi concebida para ser uma plataforma intuitiva e eficiente. A solução busca beneficiar tanto os clientes, oferecendo praticidade e uma experiência de compra mais rápida e personalizada, quanto os supermercados, otimizando suas operações e expandindo seu alcance de mercado e vendas.

1.1 Termo da Abertura do Projeto (TAP)

Segundo o livro *Project Management Body of Knowledge*, o Termo de Abertura do Projeto (TAP) é um documento crucial que formaliza a autorização do projeto, concedendo ao gerente de projetos organizar suas atividades (PROJECT MANAGEMENT INSTITUTE; Ahmad S Al-Shamsi, 1996). Esse artefato estabelece as bases para o êxito do projeto ao definir seus objetivos, restrições, premissas, responsabilidades e expectativas das partes interessadas, conforme destacado por (Kerzner, 2021).

No contexto do desenvolvimento de software, o TAP detalha as etapas necessárias para a execução do projeto, reunindo informações relevantes, como justificativa de negócio, requisitos iniciais e riscos identificados. Segundo Pressman (2021), esse documento proporciona uma visão abrangente do que será necessário para alcançar os objetivos propostos, servindo como um guia estratégico desde o início.

A relevância do TAP é amplamente reconhecida na literatura de gerenciamento de projetos. Para Heldman (2006), esse documento é fundamental para assegurar

uma compreensão clara do escopo, metas e expectativas do projeto, reduzindo desvios e garantindo alinhamento com as necessidades do negócio. O TAP é um dos processos-chave na fase de iniciação, sendo essencial para o planejamento eficaz e a gestão adequada de projetos.

1.1.1 Situação Atual

As compras online têm se tornado cada vez mais presentes no cotidiano das pessoas, abrangendo diversos setores, como alimentação, ferramentas, vestuário e dispositivos tecnológicos (Kotler, 2021).

No entanto, apesar do crescimento do e-commerce, muitos estabelecimentos locais ainda não exploram todo o potencial desse modelo de negócio.

Na cidade de Franca, interior de São Paulo, há uma grande quantidade de mercados, supermercados e atacados, mas poucos possuem aplicativos de vendas online realmente eficientes. De acordo com uma análise realizada, muitos desses aplicativos apresentam falhas, como a ausência de catálogos completos de produtos ou divergências de preços em relação às lojas físicas. Essas deficiências impactam negativamente a experiência do usuário e limitam o crescimento do comércio eletrônico local.

Em uma pesquisa realizada com *stakeholders* por meio de formulários, identificou-se que os principais problemas relatados estão relacionados a falhas na comunicação, atrasos na entrega, dificuldades no processamento de pagamentos e a falta de funcionalidades básicas, como a possibilidade de salvar lista de compras. Essas questões reforçam a necessidade de melhorias na usabilidade e na eficiência dos aplicativos, visando atender melhor às expectativas dos consumidores.

1.1.2 Justificativa do Projeto

Por meio da aplicação de técnicas de elicitação de requisitos, como formulários estruturados, e da análise SWOT, foi possível identificar deficiências críticas nos aplicativos de supermercado existentes. Conforme destacado por Wieggers e Beatty (2013), a elicitação sistemática de requisitos é fundamental para compreender as necessidades reais dos usuários. A matriz SWOT, por sua vez, revelou vulnerabilidades significativas em quatro aspectos principais: logística de entregas,

opções de pagamento, curva de aprendizado do usuário e funcionalidades de listas de compras automatizadas - problemas que corroboram as descobertas de Kotler (2021) sobre as expectativas dos consumidores no comércio eletrônico.

Essas fragilidades, conforme observado por Nielsen (2000) em seus estudos sobre usabilidade, impactam diretamente tanto a eficiência operacional quanto a experiência do cliente. No entanto, como ressaltado por Laudon e Traver (2020), tais gaps representam oportunidades estratégicas para desenvolvimento de soluções inovadoras. A implementação de um aplicativo otimizado, com foco nestas áreas problemáticas, pode oferecer benefícios mútuos: maior conveniência para os consumidores, conforme previsto no modelo de tecnologia aceitação de Davis (1989) e aumento no volume de vendas para os estabelecimentos - um alinhamento perfeito com os princípios de vantagem competitiva propostos (Porter, 1999).

1.1.3 Propósito do Projeto e Metas

O aplicativo de compras para supermercados visa simplificar a experiência do usuário, tornando o processo mais prático e conveniente, ao mesmo tempo em que amplia as vendas para os estabelecimentos. Com base em estudos de usabilidade de Nielsen (2000) e princípios de design centrado no usuário de Norman (2006), o projeto tem como principal objetivo automatizar e personalizar as listas de compras, permitindo funcionalidades como adicionar múltiplos itens, salvar listas, gerenciar várias listas e renomeá-las. Essa abordagem, alinhada com teorias de adoção tecnológica de Rogers (2003) e estratégias de comércio eletrônico de Chaffey (2019), busca melhorar a eficiência e a satisfação do consumidor, promovendo maior engajamento e fidelização.

1.1.4 Descrição do Projeto

O projeto foi dividido em cinco etapas: iniciação, planejamento, execução, monitoramento e finalização. Todas as etapas passaram em paralelo pela etapa de monitoramento, que será descrita a seguir. Em conjunto da equipe de desenvolvimento, houve participação direta do orientador/PO do projeto, auxiliando na criação e validação de todos os artefatos para a solução sistêmica.

Na primeira etapa 'Iniciação', foi definida equipe participante, selecionado o gerente, escolhido a ideia principal do projeto, realizado pesquisas e estudos sobre o segmento abordado e definida proposta de criar um formulário de pesquisa para melhor entendimento das necessidades do cliente e do negócio.

Na segunda etapa 'Planejamento', criamos o princípio de conduta definido pela missão, visão e valor de nossa empresa. Geramos também dois formulários destinados aos mercados e potenciais clientes, onde através das respostas foi possível identificar responsabilidades, requisitos funcionais e não funcionais, atores internos e externos para o sistema. Após análise da coleta e reunião com stakeholders, levantamos pontos de fraqueza, força, oportunidade e ameaça. Desses pontos identificados realizamos uma diagramação da matriz SWOT para maior clareza do negócio. Identificamos quatro pontos fracos para a proposta: o sistema de entregas, a plataforma de pagamento, o aprendizado do usuário e a automatização da lista de pedidos. Exploramos estas fraquezas criando questões problema para possível solução com o desenvolvimento da aplicação. Cada questão problema recebeu um tratamento de acordo com a matriz 5W2H. Após análise e extração de ideias das matrizes foi fechado o escopo do projeto e definido uma funcionalidade essencial para a solução sistêmica: permitir a personalização da lista de compras para os clientes.

Na terceira etapa 'Execução' foram criados 14 artefatos, para futuro auxílio na produção do código fonte do sistema, interação com usuários do sistema e alinhamentos comerciais. Inicialmente foi desenvolvido o BPMN (*Business Process Model and Notation*) para descrição geral de todos os processos envolvidos na aplicação, nesta etapa foram estabelecidas funcionalidades essenciais para o fluxo de execução do sistema, como por exemplo, personalizar lista de compras. Em sequência a documentação de requisitos e o diagrama de casos de uso. De forma intercalada, a documentação de casos de uso, diagramas de atividade, máquina de estado, sequência e classes, todos embasados no padrão internacional estabelecido pela UML. Finalizado o diagrama de classes, foram desenvolvidas as métricas, proposta comercial e prototipação de telas. Em seguida gerada matriz de rastreabilidade e pôr fim a documentação de portabilidade. Todos os artefatos passaram por validação do orientador/PO do projeto.

A quarta etapa 'Monitoramento', foi realizada de forma cíclica em conjunto com as demais etapas. Sendo feito o acompanhamento das tarefas, o controle de qualidade, mudanças e o controle do cronograma.

A quinta etapa 'Finalização' passou pelos processos de validação final, entrega e apresentação.

O sistema segue dois fluxos de execução onde os usuários do mercado e os clientes do mercado contarão com funcionalidades distintas, mas que se complementam. Ambos os usuários terão cadastro e vão precisar logar no sistema. Para os clientes do mercado existem funções para acompanhamento de pedido e gerenciamento de listas como selecionar, criar, editar, salvar e excluir. Na interface dos usuários do mercado existem funções para atualizar lista de produtos e gerenciamento de pedidos como listar pedidos, aceitar e notificar andamento do pedido.

1.1.5 Premissas

- Constante monitoramento e validação para criar um software eficiente e eficaz, intuitivo e com boa usabilidade;
- Desenvolvimento de uma solução sistêmica potável para plataforma web e com posterior evolução do projeto implementar plataforma mobile.
- Seguir o cronograma estipulado pela equipe, realizando entregas dentro dos prazos, separando todo o processo em pequenas atividades exequíveis para ganho de produtividade e tempo disponível para ajustes.

1.1.6 Restrições

- Tempo disponível dos membros da equipe. A equipe é formada por dois membros que trabalham 8 horas por dia e cursam presencial noturno o curso de ADS, com tempo reduzido para dedicação às atividades do projeto;
- Capacitação dos integrantes da equipe de desenvolvimento. Os integrantes da equipe de desenvolvimento estão cursando faculdade e ainda no processo de aprendizado da maioria dos artefatos produzidos na solução;
- Consideração de recursos disponíveis pela contratante e da equipe de desenvolvimento;

- Adaptabilidade (treinamento adequado) ao novo sistema implantado por parte dos funcionários do mercado.

1.1.7 Stakeholders

- Equipe do mercado, operador de caixa, operador de reposição/separação, motorista, gerente de caixa, proprietário do mercado;
- Clientes do mercado usuários do aplicativo;
- Equipe de desenvolvimento e orientador PO, professores de disciplinas congruentes ao desenvolvimento do sistema;
- IAs para orientação e instrumento de busca de temas técnicos.

1.1.8 Riscos

- Segurança envolvendo dados sensíveis;
- Qualidade, eficiência e eficácia do desenvolvimento de todos os artefatos da solução sistêmica;
- Desligamento por motivos diversos de integrante da equipe;
- Concorrência de mercado;

1.1.9 Responsabilidades

Todos os membros da equipe de desenvolvimento colaboraram ativamente no levantamento dos processos do BPMN, na identificação dos pontos críticos da matriz SWOT e na posterior elaboração da matriz 5W2H, que foi construída a partir das fragilidades detectadas na análise. O trabalho foi distribuído de forma estratégica entre os integrantes, garantindo a cobertura completa de todas as etapas necessárias para o sucesso da iniciativa.

Os autores assumiram diversas responsabilidades cruciais para o andamento do projeto. Além de serem os responsáveis pela redação inicial deste Termo de Abertura, a equipe dedicou-se à criação das métricas do sistema e ao desenvolvimento do diagrama de sequência, elementos fundamentais para a modelagem das interações entre os componentes da aplicação. Ambos os membros também atuaram na elaboração do gráfico da matriz SWOT, na prototipação das telas

do sistema e na construção dos diagramas de atividade e máquina de estado, essenciais para o entendimento do fluxo operacional.

Os envolvidos no projeto ainda desempenharam um papel importante no acompanhamento geral, monitorando todas as etapas, participando ativamente das validações com stakeholders, controlando o cronograma e contribuindo para o processo criativo de todos os artefatos desenvolvidos.

Os membros responsáveis concentraram seus esforços em áreas igualmente estratégicas. Encarregados pelo desenvolvimento do diagrama de casos de uso e pela respectiva documentação técnica, que servem como base para o entendimento das funcionalidades do sistema. Ambos participaram intensamente do processo de prototipação das telas, garantindo que a interface do usuário atendesse aos requisitos estabelecidos, e elaboraram a minuciosa documentação de portabilidade. Os colaboradores ainda desenvolveram os dois formulários de pesquisa utilizados no projeto, criaram o diagrama BPMN que mapeia os processos do negócio, preencheram toda a documentação de requisitos e elaboraram o importante diagrama de classes, além de terem contribuído significativamente para o diagrama de sequência.

A equipe trabalhou de forma coesa e integrada, com um sistema constante de revisões e validações que assegurou o perfeito alinhamento de todos os artefatos produzidos com os requisitos do projeto e as expectativas dos stakeholders. Essa abordagem colaborativa e meticulosa resultou no desenvolvimento de uma solução robusta e bem-estruturada, que se encontra agora pronta para avançar para a fase de implementação prática. A distribuição equilibrada de tarefas e a complementaridade das habilidades dos membros da equipe foram fatores determinantes para o sucesso alcançado nesta etapa do projeto.

1.2 EAP

A EAP (Estrutura Analítica de Projeto) organiza de forma hierárquica as etapas a serem desenvolvidas de forma visual, com o objetivo de promover uma visão total, simples e organizada do projeto.

Figura 1 – Estrutura Analítica do Projeto.



Fonte: Elaborado pelos autores.

2 Viabilidade do Projeto

A viabilidade do projeto foi analisada sob diferentes perspectivas, para garantir que a solução proposta seja exequível e atenda às necessidades identificadas. Para isso, foram utilizadas ferramentas como formulários de pesquisa e a matriz SWOT, que permitiram validar as demandas do mercado e avaliar os pontos críticos a serem abordados. A combinação desses métodos forneceu dados concretos para embasar as decisões de desenvolvimento, assegurando que o sistema proposto não apenas resolva problemas existentes, mas também apresente vantagens competitivas.

2.1 Formulários

A ideia de construir formulários de busca e alinhamento com possíveis clientes e mercados, veio da necessidade de entender melhor o negócio e o que é esperado das partes envolvidas, tanto do mercado quando os clientes do mercado. Foram criadas 15 perguntas para o formulário dos clientes (<https://forms.office.com/r/qZpgda58FR>) e 12 perguntas para o formulário do mercado (<https://forms.office.com/r/YGU0kT5FBr>). Através dos formulários, foram coletadas as respostas e elaborados gráficos, para uma melhor visualização dos dados([https://drive.google.com/drive/folders/1RuP8I0edEqjxIQC8xsTARQ2Pvz1Klx9O?usp=drive link](https://drive.google.com/drive/folders/1RuP8I0edEqjxIQC8xsTARQ2Pvz1Klx9O?usp=drive_link)).

2.2 Matriz SWOT

A matriz SWOT é uma ferramenta de análise de negócio que consiste em avaliar Strengths (Forças) e Weaknesses (Fraquezas), referentes ao funcionamento interno da equipe e as Opportunities (Oportunidades) e Threats (Ameaças) da empresa, referentes ao mercado. A importância da matriz SWOT está em identificar as fraquezas que podem ser tratadas com uma solução tecnológica, neste caso.

Foram identificadas 4 fraquezas relacionadas ao escopo da solução sistêmica: Sistema de entregas, Plataforma de pagamentos, Aprendizado do usuário e automatizar a lista de pedidos. Todas essas fraquezas foram identificadas pelo formulário destinado aos clientes dos supermercados, o que demonstra claramente os problemas mais presentes em aplicações de mesmo nicho.

Figura 2 – Matriz SWOT



Fonte: Elaborado pelos autores.

2.3 Plano de Ação 5W2H do Projeto

O 5W2H é uma matriz que tem como finalidade a otimização de visibilidade de plano de ação, formada de 7 perguntas (*What, Why, Where, Who, When, How e How Much*). Essa metodologia ajuda a detalhar e esclarecer tarefas, projetos ou processos, garantindo uma compreensão abrangente e alinhamento entre as partes envolvidas. Sua importância reside na capacidade de estruturar informações de forma clara e objetiva, facilitando a execução e monitoramento das atividades. A partir da construção desta matriz, conseguimos criar um plano de ação alinhado com os requisitos da aplicação.

Foram levantadas seis questões problemas a partir das fraquezas identificadas na matriz swot, entretanto devido a proposta geral do projeto, o fechamento de escopo se fez necessário, onde o grupo envolvido focaria em tratar apenas uma questão problema.

O plano de ação selecionado foca no tratamento da lista de compras para os clientes do mercado, fornecendo algumas funcionalidades a criação e personalização de listas de compras.

Figura 3 – Matriz 5W2H

Questão Problema: Automatizar lista de pedidos (mercado/cliente)							
Ação	O que? (What?)	Por que? (Why?)	Onde (Where?)	Quem? (Who?)	Quando? (When?)	Como? (How?)	Quanto? (How much?)
1	Permitir a personalização da lista de compra	Para uma experiência mais prática ao usuário	No aplicativo de vitrine virtual.	Equipe envolvida no projeto.	Março a Junho de 2024.	Implementando funcionalidades para adicionar x unidades, remover itens, salvar lista, criar mais de uma lista, renomear a lista.	Calculo será realizado em Junho de 2024.
2	Automatizar a lista de compras que chega no mercado.	Para otimizar os processos de aceite, separação, finalização do pedido.	No aplicativo de vitrine virtual.	Equipe envolvida no projeto.	Março a Junho de 2024.	Implementando um módulo no aplicativo para o mercado onde serão listados os produtos permitindo aprovação do pedido, dar um check nos itens selecionados, botão para notificar compra separada, botão para confirmar a disponibilidade do envio.	Calculo será realizado em Junho de 2024.

Fonte: Elaborado pelos autores.

3 Levantamento de Requisitos

O levantamento de requisitos foi conduzido com rigor metodológico para garantir que todas as necessidades dos stakeholders fossem capturadas e traduzidas em especificações técnicas precisas. Para isso, adotou-se uma abordagem multifacetada, combinando métodos quantitativos e qualitativos que permitiram mapear tanto as expectativas explícitas quanto as demandas latentes dos usuários finais.

3.1 Elicitação e especificação dos Requisitos

A elicitação de requisitos se deu por meio dos formulários de busca e alinhamento, *brainstorming* e reuniões periódicas entre os participantes do projeto. Através destas técnicas foi possível extrair quais as funcionalidades esperadas pelos possíveis stakeholders, incluindo clientes dos mercados e os funcionários dos mercados. O formulário destinado aos consumidores dos supermercados foi compartilhado entre o círculo social de convívio dos participantes do grupo e o formulário destinado aos mercados foi disseminado pontualmente em locais selecionados pelos integrantes do projeto, possíveis clientes potenciais. Os dados coletados foram analisados para definir requisitos funcionais e não funcionais da aplicação.

As perguntas e respostas levantadas nos formulários estão disponíveis no Apêndice 2 [no final].

3.2 BPMN

O BPMN (*Business Process Model and Notation*) é uma metodologia padronizada para modelagem de processos de negócios, utilizando diagramas gráficos para representar de forma visual e clara os fluxos de trabalho. Ele proporciona uma linguagem comum que pode ser compreendida por todos os *stakeholders*, incluindo analistas de negócios que criam e refinam os processos, técnicos que implementam as soluções e gestores que monitoram e gerenciam os processos.

O BPMN serve para documentar, analisar e melhorar processos de negócios, facilitando a comunicação e colaboração entre diferentes áreas da organização. Ele é utilizado para mapear processos complexos, identificar ineficiências e implementar melhorias, garantindo que os processos estejam alinhados com os objetivos estratégicos da empresa.

Conforme o levantamento de requisitos, o fluxo de processo pode ser visualizado de maneira clara e detalhada no diagrama BPMN. Devido ao seu tamanho, o link do diagrama completo está incluído no Anexo 1, que está junto a esta documentação.

3.3 Requisitos Funcionais

Os requisitos funcionais, tem como objetivo descrever as funcionalidades específicas que um sistema deve executar para atender às necessidades dos usuários e stakeholders. Eles descrevem o comportamento do sistema e as ações que ele deve ser capaz de realizar. Segundo Pressman (2021), "requisitos funcionais são declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em situações particulares". Wieggers e Beatty (2013) também destacam que esses requisitos são essenciais para garantir que o sistema atenda aos seus propósitos de negócio e forneça valor aos usuários.

Eles abrangem requisitos de usuário, de sistema, de domínio, evidentes e ocultos. A seguir apresentamos o índice dos requisitos funcionais identificados para a aplicação e a posterior sua documentação.

- RF 001: Cadastrar clientes
- RF 002: Validar cadastro
- RF 003: Logar no sistema
- RF 004: Validar login
- RF 005: Atualizar lista de produtos
- RF 006: Listar pedidos
- RF 007: Aceitar pedido
- RF 008: Informar entrega
- RF 009: Abrir menu cliente
- RF 010: Ver cadastro
- RF 011: Editar cadastro
- RF 012: Atualizar cadastro
- RF 013: Visualizar setores
- RF 014: Buscar Produtos
- RF 015: Adicionar no carrinho
- RF 016: Criar lista
- RF 017: Visualizar listas
- RF 018: Selecionar lista
- RF 019: Personalizar lista
- RF 020: Renomear lista
- RF 021: Adicionar itens
- RF 022: Remover Itens
- RF 023: Salvar Lista
- RF 024: Revisar pedido
- RF 025: Fazer pedido
- RF 026: Enviar pedidos
- RF 027: Ver pedido
- RF 028: Informar pedido aceito
- RF 029: Carregar plataforma mercado
- RF 030: Carregar plataforma cliente
- RF 031: Abrir menu mercado
- RF 032: Finalizar pedido
- RF 033: Atualizar lista de pedidos
- RF 034: Atualizar status pedido

- RF 035: Selecionar plataforma
- RF 036: Enviar form cadastro
- RF 037: Enviar form login
- RF 038: Separar produtos

Quadro 1 – Requisitos Funcionais

RF001 - Cadastrar Usuários	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir aos usuários do mercado e usuários clientes do mercado, criar cadastro para acesso a plataforma.		
RF002 - Validar cadastro	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá validar todos os novos cadastros realizados, permitindo ou não prosseguir.		
RF003 - Logar no sistema	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O usuário deverá logar no sistema para ter acesso aos recursos da plataforma.		
RF004 - Validar login	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá verificar validade do login acessando a base de dados, liberar ou não o acesso.		
RF005 - Atualizar lista de produtos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o usuário do mercado realize cadastro e atualização de produtos na base de dados.		
RF006 - Listar pedidos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o usuário do mercado liste os pedidos recebidos.		
RF007 - Aceitar pedido	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o usuário do mercado aceite pedidos recebidos.		
RF008 - Informar entrega	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média

		() Baixa
Descrição: O sistema deverá enviar notificação para a plataforma do usuário cliente após confirmado que os produtos foram separados e estão prontos para entrega. RF038.		
RF009 - Abrir menu cliente	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá exibir a tela de menu inicial após carregada plataforma cliente RF030.		
RF010 - Ver cadastro	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o usuário cliente veja os seus dados cadastrados.		
RF011 - Editar cadastro	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o cliente faça a edição dos dados cadastrados.		
RF012 - Atualizar cadastro	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá atualizar a base de dados do usuário após evento de click no botão opção confirmar do formulário de editar dados do cadastro.		
RF013 - Visualizar setores	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o cliente carregue a lista de setores.		
RF014 - Buscar Produtos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir a busca de produtos pelo usuário cliente.		
RF015 - Adicionar no carrinho	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir o cliente selecionar produtos e sua quantidade.		
RF016 - Criar lista	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir o usuário cliente, criar listas a partir dos produtos selecionados/carrinho de compras.		
RF017 - Visualizar listas	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o usuário cliente veja suas listas criadas.		

RF018 - Selecionar lista	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o usuário cliente selecione uma lista pronta e continue a compra ou edite a lista.		
RF019 - Personalizar lista	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o usuário cliente personalize uma lista selecionada.		
RF020 - Renomear lista	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir que o usuário cliente renomeie uma lista selecionada.		
RF021 - Adicionar itens	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve permitir que o usuário cliente adicione itens em uma lista selecionada.		
RF022 - Remover Itens	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve permitir que o usuário cliente remova produtos da lista.		
RF023 - Salvar Lista	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir ao cliente salvar a lista personalizada.		
RF0024 - Revisar pedido	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir ao cliente revisar a lista de produtos antes de confirmar o pedido.		
RF025 - Fazer pedido	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir o usuário cliente enviar/fazer o pedido, após a revisão da lista.		
RF026 - Enviar pedidos	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá enviar notificação de pedido contendo o pedido para plataforma do mercado.		
RF027 - Ver pedidos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta

		() Média () Baixa
Descrição: O sistema deverá permitir que o usuário cliente acompanhe o status do pedido.		
RF028 - Informar pedido aceito	Categoria: (X) Oculto () Evidente	Prioridade: () Altíssima () Alta (X) Média () Baixa
Descrição: O sistema deverá enviar notificação para plataforma do usuário cliente assim que o mercado aceitar o pedido.		
RF029 - Carregar plataforma mercado	Categoria: (X) Oculto () Evidente	Prioridade: () Altíssima () Alta (X) Média () Baixa
Descrição: O sistema deverá carregar a página do mercado após validação de login e seleção de plataforma.		
RF030 - Carregar plataforma cliente	Categoria: (X) Oculto () Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá carregar a página do cliente após validação de login e seleção de plataforma.		
RF031 - Abrir menu mercado	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá exibir a tela de menu inicial após carregada plataforma mercado.		
RF032 - Finalizar pedido	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá permitir ao mercado finalizar o pedido, após receber o comprovante de pagamento.		
RF033 - Atualizar lista de pedidos	Categoria: (X) Oculto () Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá atualizar a lista de pedidos sempre que for gerado um novo pedido de cliente.		
RF034 - Atualizar status pedido	Categoria: (X) Oculto () Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá atualizar o status do pedido de acordo com notificação do mercado.		
RF035 - Selecionar plataforma	Categoria: (X) Oculto () Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá selecionar a plataforma para ser carregada de acordo com o login validado.		
RF036 - Enviar form cadastro	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta

		() Média () Baixa
Descrição: O sistema deverá permitir o usuário enviar os dados de cadastro para validação.		
RF037 - Enviar form login	Categoria: (X) Oculto () Evidente	Prioridade: () Altíssima (X) Alta () Média () Baixa
Descrição: O sistema deverá permitir o usuário enviar os dados de login para o sistema validar.		
RF038 - Separar produtos	Categoria: () Oculto (X) Evidente	Prioridade: () Altíssima () Alta (X) Média () Baixa
Descrição: O mercado irá separar os produtos e deverá informar pelo aplicativo que o pedido foi separado, a previsão de entrega e o nome do entregador. RF008.		

3.4 Casos de Uso

Um Caso de Uso descreve como um usuário interage com um sistema para atingir um objetivo. Ele define uma sequência de ações, mostrando a interação entre o ator (usuário ou outro sistema) e o sistema. De acordo com (Cockburn, 2000) *Writing Effective Use Cases*), "um caso de uso é uma lista de ações ou passos tipicamente definidos através de uma interação entre um papel (ou ator) e um sistema para alcançar um objetivo".

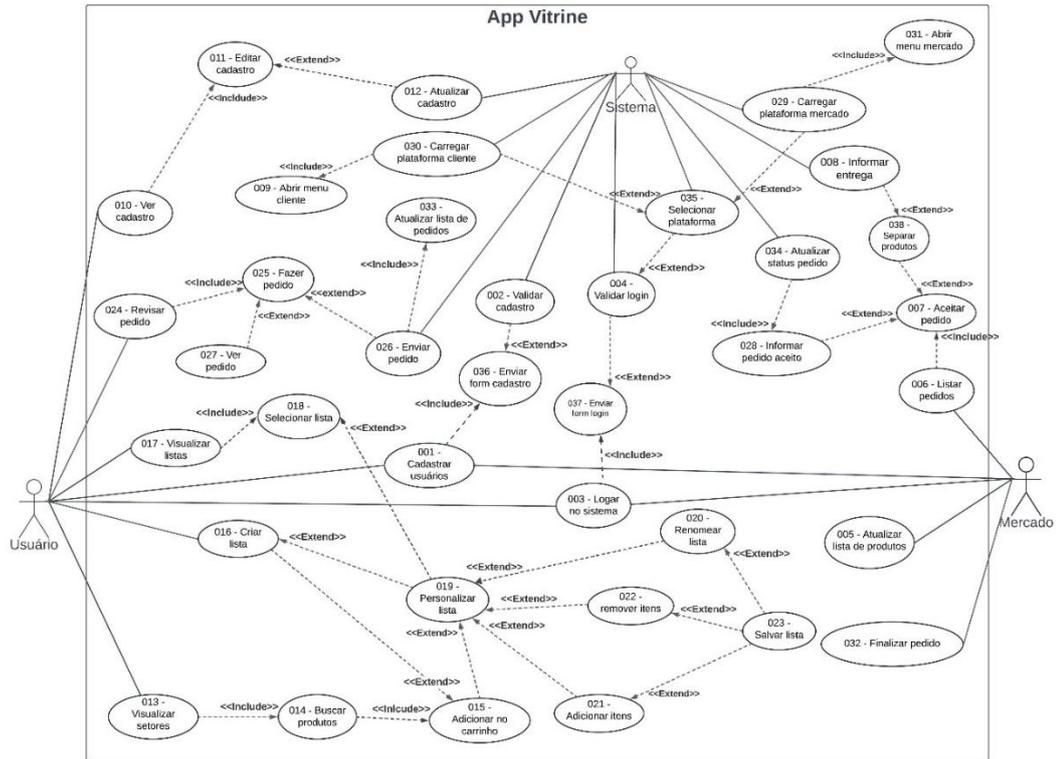
Os Casos de Uso são cruciais para o desenvolvimento de software porque ajudam a esclarecer os requisitos do sistema e a entender o comportamento esperado sob diferentes condições. Eles facilitam a comunicação entre desenvolvedores, analistas e stakeholders, assegurando que todos tenham uma compreensão clara dos requisitos e funcionalidades. Além disso, ajudam na identificação de possíveis falhas e na definição de testes de aceitação, conforme mencionado por Jacobson (2005), que destacou que "os casos de uso são fundamentais para capturar os requisitos funcionais e fornece uma base para o design e testes de sistemas". Em resumo, Casos de Uso são ferramentas essenciais para assegurar que o sistema desenvolvido realmente atenda às necessidades dos usuários e aos objetivos do negócio.

A seguir, apresentamos o diagrama de casos de uso e os cenários identificados, organizados por atividades, o que facilita a leitura e a análise. Em sequência a documentação dos casos de uso.

- UC 001: Cadastrar clientes
- UC 002: Validar cadastro
- UC 003: Logar no sistema
- UC 004: Validar login

- UC 005: Atualizar lista de produtos
- UC 006: Listar pedidos
- UC 007: Aceitar pedido
- UC 008: Informar entrega
- UC 009: Abrir menu cliente
- UC 010: Ver cadastro
- UC 011: Editar cadastro
- UC 012: Atualizar cadastro
- UC 013: Visualizar setores
- UC 014: Buscar Produtos
- UC 015: Adicionar no carrinho
- UC 016: Criar lista
- UC 017: Visualizar listas
- UC 018: Selecionar lista
- UC 019: Personalizar lista
- UC 020: Renomear lista
- UC 021: Adicionar itens
- UC 022: Remover Itens
- UC 023: Salvar Lista
- UC 024: Revisar pedido
- UC 025: Fazer pedido
- UC 026: Enviar pedidos
- UC 027: Ver pedido
- UC 028: Informar pedido aceito
- UC 029: Carregar plataforma mercado
- UC 030: Carregar plataforma cliente
- UC 031: Abrir menu mercado
- UC 032: Finalizar pedido
- UC 033: Atualizar lista de pedidos
- UC 034: Atualizar status pedido
- UC 035: Selecionar plataforma
- UC 036: Enviar form cadastro
- UC 037: Enviar form login
- UC 038: Separar produtos

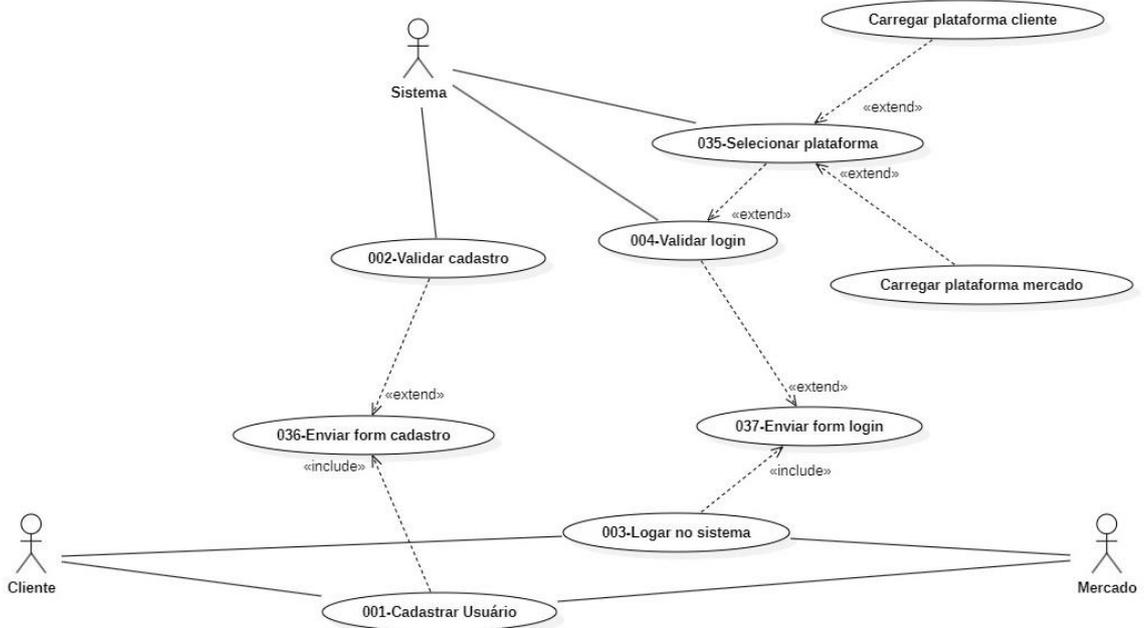
Figura 4 – UC Visão geral do sistema



Fonte: Elaborado pelos autores.

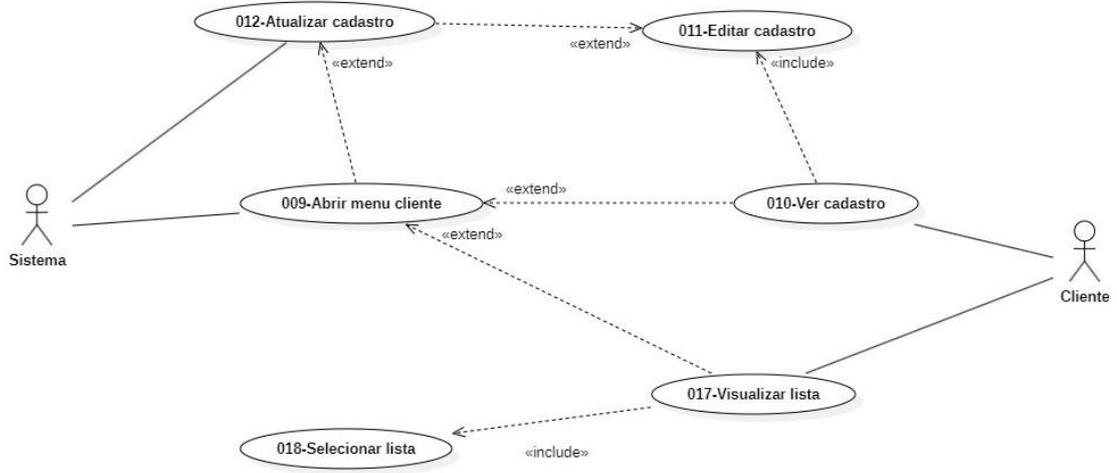
Imagem 5 – UC Acesso ao sistema

Acesso ao sistema



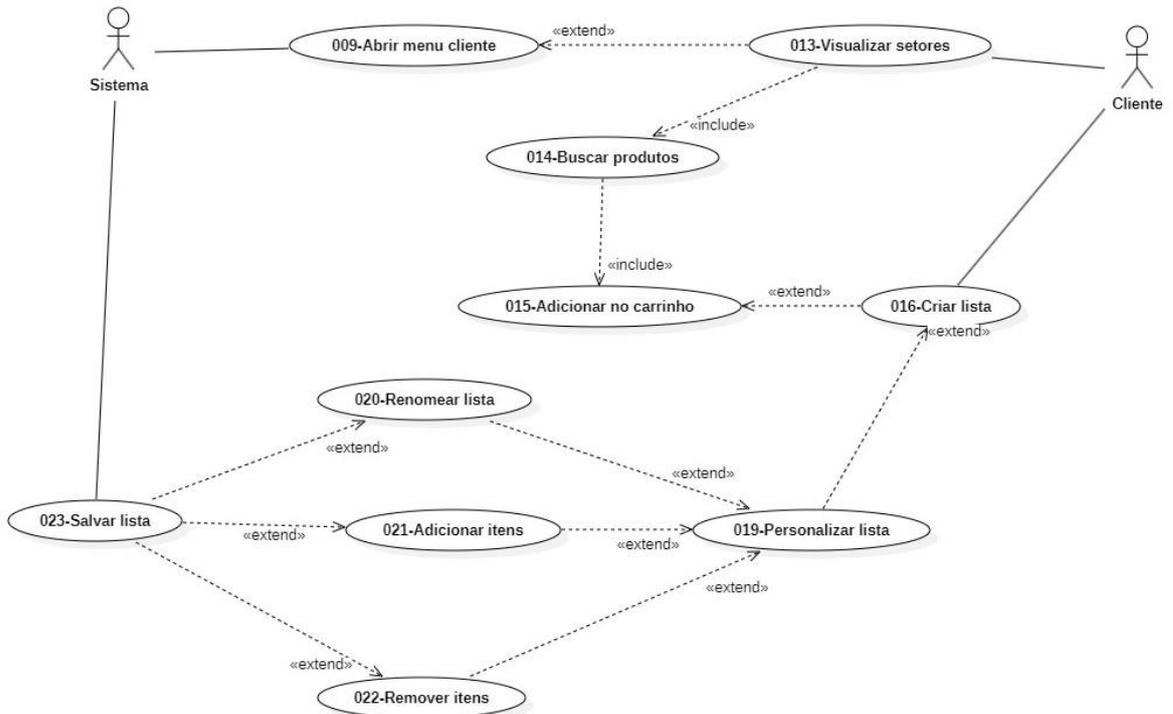
Fonte: Elaborado pelos autores.

Figura 6 – UC Editar cadastro, UC Selecionar lista
 EditarCadastro-SelecionarLista



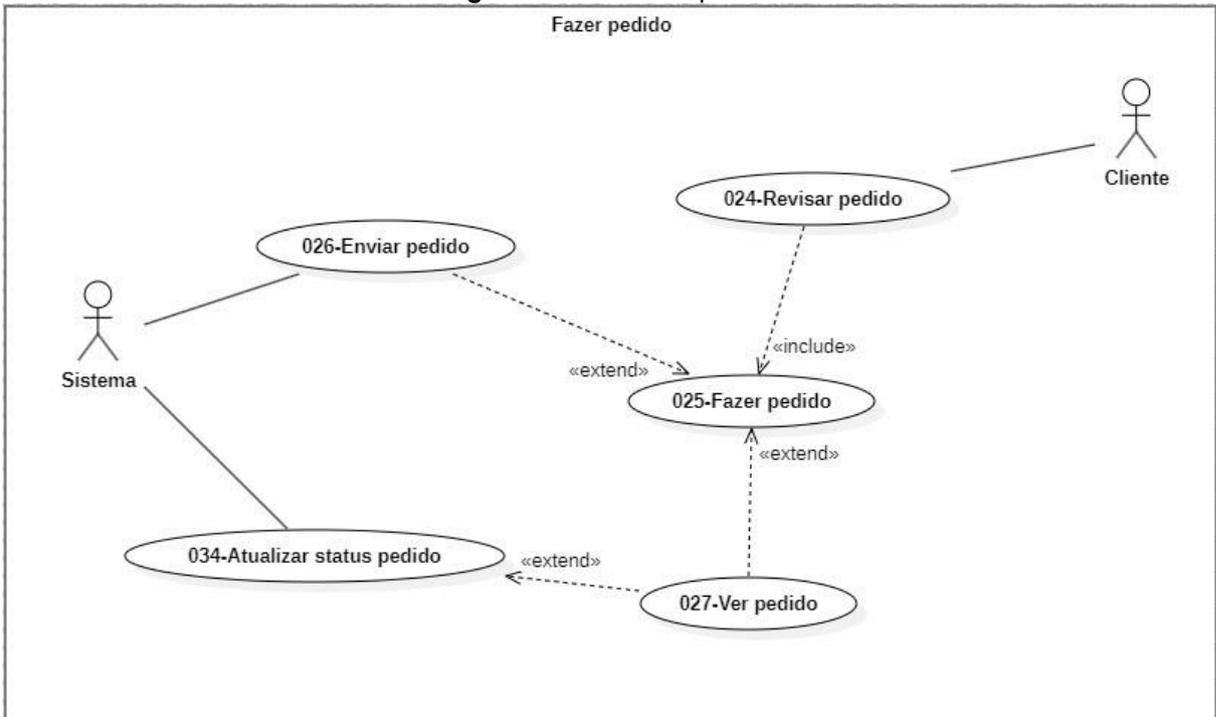
Fonte: Elaborado pelos autores.

Figura 7 – UC Criar lista, UC Personalizar lista
 CriarLista-PersonalizarLista



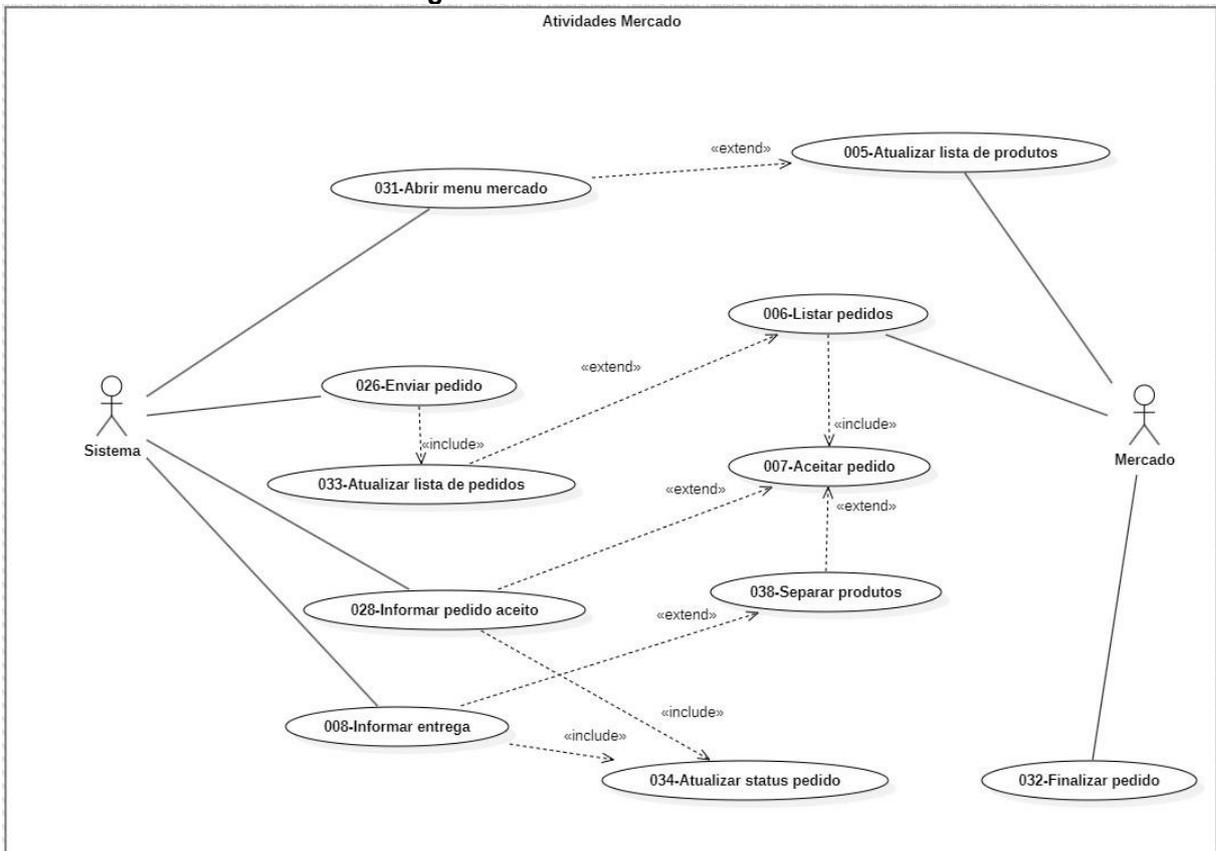
Fonte: Elaborado pelos autores.

Figura 8 – UC Fazer pedido



Fonte: Elaborado pelos autores.

Figura 9 – UC Atividades Mercado



Fonte: Elaborado pelos autores.

Quadro 2 – Documentação de casos de uso

Caso de uso – Cadastrar clientes	
ID	UC 001
Descrição	Os usuários, mercado ou cliente preencherão formulário de cadastro atendendo ao RF 001.
Ator primário	Cliente ou mercado.
Ator secundário	Sistema
Pré-condição	Não ser cadastrado.
Cenário principal	<ol style="list-style-type: none"> 1. O cliente acessará o sistema e seleciona opção de realizar cadastro. 2. O sistema carregará um formulário de cadastro. 3. O cliente preencherá os campos do formulário CPF(Cliente) ou CNPJ(Mercado), nome, endereço (logradouro, bairro, cep, cidade), telefone, Email (opcional) e sua senha.
Pós-condição	O usuário enviar o formulário.
Cenário alternativo	1. O ator principal sair do sistema.
Inclusão	UC 036 – Enviar form cadastro.
Extensão	Não há.
Caso de uso – Validar cadastro	
ID	UC 002
Descrição	O sistema validará as informações do cadastro inseridas pelo cliente atendendo o RF 002.
Ator primário	Sistema.
Pré-condição	Ter sido preenchido e enviado o formulário de cadastro.
Cenário principal	<ol style="list-style-type: none"> 1. O sistema verificará se os dados presentes no formulário (nome, CPF/CNPJ, celular, e-mail, senha, logradouro, bairro, cep, cidade) já existem no banco de dados. 2. Se eles já estiverem cadastrados, uma mensagem de conta já cadastrada será mostrada ao cliente. 3. Se eles não estão presentes no banco de dados um novo perfil será criado e uma mensagem é mostrada ao ator. 4. O sistema redirecionará o cliente para a página de login.
Pós-condição	O ator preencher os campos de login.
Cenário alternativo	Não há.
Inclusão	Não há.
Extensão	UC 036 – Enviar form cadastro.
Caso de uso – Logar no sistema	
ID	UC 003
Descrição	Os usuários irão logar na plataforma atendendo ao RF 003.
Ator primário	Cliente ou mercado.
Pré-condição	Ter um cadastro no aplicativo.
Cenário principal	<ol style="list-style-type: none"> 1. Os usuários selecionarão a opção logar ao acessar o sistema ou será redirecionado para a página de login. 2. O sistema carregará o formulário de login. 3. Os usuários preencheram os campos do formulário CPF(Cliente) ou CNPJ(Mercado), e sua senha.
Pós-condição	O usuário enviar o formulário.

Cenário alternativo	1. O cliente sair do aplicativo
Inclusão	UC 037 – Enviar form login.
Extensão	Não há.
Caso de uso – Validar login	
ID	UC 004
Descrição	O sistema validará o login atendendo o RF 004.
Ator primário	Sistema.
Pré-condição	Ter sido gerado um formulário de login.
Cenário principal	<ol style="list-style-type: none"> 1. O sistema validará os dados do formulário de login (CPF/CNPJ e senha) com o banco de dados. 2. Se os dados não estão presentes no banco de dados uma mensagem de login inválido é mostrada ao cliente. 3. Se os dados estão presentes no banco de dados uma mensagem de confirmação de login é mostrada ao cliente. 4. O sistema liberará o acesso para o cliente usar sua conta.
Pós-condição	O sistema irá selecionar a plataforma do cliente ou do mercado de acordo com a validação do login realizado.
Cenário alternativo	<ol style="list-style-type: none"> 1. Os dados estiverem incorretos. 2. O perfil já estar logado.
Inclusão	Não há.
Extensão	UC 037 – Enviar form login. UC 035 – Selecionar plataforma.
Caso de uso – Atualizar lista de produtos	
ID	UC 005
Descrição	O mercado atualiza as informações dos produtos atendendo o RF 005.
Ator primário	Mercado.
Pré-condição	O sistema ter carregado a plataforma do mercado.
Cenário principal	<ol style="list-style-type: none"> 1. O mercado enviará as informações dos seus produtos como nome, quantidade, imagem e preço no banco de dados. 2. Uma mensagem de atualização é mostrada após o envio. 3. As informações serão atualizadas na plataforma.
Pós-condição	O usuário mercado retornar para o menu.
Cenário alternativo	1. Não houver produtos para serem cadastrados.
Inclusão	Não há.
Extensão	Não há.
Caso de uso – Listar pedidos	
ID	UC 006
Descrição	O mercado listará os pedidos dos clientes atendendo o RF 006.
Ator primário	Mercado.
Pré-condição	O cliente ter feito um pedido.
Cenário principal	1. O mercado anotar as informações do dono do pedido como nome solicitante, endereço, telefone, valor pedido, horário pedido, observação de entrega e status do pedido (pendente, em processo, concluído).

Pós-condição	O mercado aceitar o pedido.
Cenário alternativo	Não há.
Inclusão	UC 007 – Aceitar pedido.
Extensão	Não há.
Caso de uso – Aceitar pedido	
ID	UC 007
Descrição	O mercado irá aceitar o pedido e o sistema irá modificar o status do pedido para o cliente atendendo o RF 007.
Ator primário	Mercado.
Pré-condição	O mercado ter listado o pedido.
Cenário principal	1. O mercado irá aceitar o pedido após verificação e confirmação pelo botão aceitar.
Pós-condição	Após evento de aceitar o pedido, o sistema irá modificar o status (pedido aceito) do pedido na plataforma do cliente. O mercado separa os produtos.
Cenário alternativo	1. Não há produtos suficientes para o pedido.
Inclusão	Não há.
Extensão	UC 028 – Informar pedido aceito. UC 038 – Separar produtos.
Caso de uso – Informar entrega	
ID	UC 008
Descrição	O sistema informará o cliente sobre o pedido atendendo o RF 008.
Ator primário	Sistema.
Pré-condição	O pedido ter sido aceito e separado.
Cenário principal	1. O sistema avisará o cliente através de uma mensagem que seu pedido foi separado e está pronto para ser entregue. 2. O sistema informará ao cliente o tempo estimado da entrega e o nome do entregador.
Pós-condição	O mercado realizará entrega e o cliente receberá o pedido.
Cenário alternativo	1. Problemas com a logística. 2. O cliente não estar em casa.
Inclusão	Não há.
Extensão	UC 038 – Separar produtos
Caso de uso – Abrir menu cliente.	
ID	UC 009
Descrição	O sistema abrirá o menu do cliente atendendo o RF 009.
Ator primário	Sistema.
Pré-condição	O sistema carregar a plataforma do cliente.
Cenário principal	1. Após o cliente logar com sua conta o sistema inicializará o menu cliente por onde ele irá navegar pelo app.
Pós-condição	1. O cliente pode escolher entre ver o seu cadastro, visualizar setores, visualizar listas, revisar pedido ou ver o pedido.
Cenário alternativo	Não há.
Inclusão	Não há.
Extensão	Não há.
Caso de uso – Ver cadastro	
ID	UC 010
Descrição	O cliente poderá ver o seu cadastro atendendo o RF 010.
Ator primário	Cliente.

Pré-condição	O cliente estar logado.
Cenário principal	1. O cliente selecionará uma opção que lhe mostra as informações de sua conta como nome, endereço, telefone, CPF e senha.
Pós-condição	O cliente poderá editar o cadastro.
Cenário alternativo	1. O cliente sai do aplicativo.
Inclusão	UC 011 – Editar cadastro.
Extensão	Não há.
Caso de uso – Editar cadastro	
ID	UC 011
Descrição	O cliente editará o seu cadastro atendendo o RF 011.
Ator primário	Cliente.
Pré-condição	O cliente escolheu editar o cadastro.
Cenário principal	1. O cliente informará seus dados nos campos do formulário aberto pelo sistema. 2. O cliente seleciona o botão para confirmar atualização de dados. 3. O sistema solicita que o usuário confirme os dados inseridos.
Pós-condição	O sistema atualizará o cadastro.
Cenário alternativo	1. O cliente informar dados inválidos ou incompletos.
Inclusão	Não há.
Extensão	UC – 012 Atualizar cadastro.
Caso de uso – Atualizar cadastro	
ID	UC 012
Descrição	O sistema atualizará o cadastro do cliente atendendo o RF 012.
Ator primário	Sistema.
Pré-condição	O cliente ter preenchido o formulário de cadastro.
Cenário principal	1. O sistema receberá o formulário que foi preenchido pelo cliente e aplica no banco de dados. 2. O sistema alterará os dados de acordo com o formulário. 3. Uma mensagem de alteração será enviada ao cliente.
Pós-condição	O cliente retornar ao menu.
Cenário alternativo	Não há.
Inclusão	Não há.
Extensão	UC011 – Editar cadastro
Caso de uso – Visualizar setores	
ID	UC 013
Descrição	O cliente visualizará os setores atendendo o RF 013.
Ator primário	Cliente.
Pré-condição	O cliente escolheu visualizar o setor.
Cenário principal	1. O cliente escolherá uma opção onde o sistema carrega a lista de setores para ele. 2. O sistema deverá mostrar os setores do mercado como hortifruti, enlatados, grãos, massas, limpeza, padaria e açougue.
Pós-condição	O cliente buscar por itens.
Cenário alternativo	1. O cliente sai do aplicativo.
Inclusão	UC 014 – Buscar produtos
Extensão	Não há.
Caso de uso – Buscar produtos	

ID	UC 014
Descrição	O cliente buscará por produtos atendendo o RF 014.
Ator primário	Cliente.
Pré-condição	O cliente visualizou o setor.
Cenário principal	1. O cliente acessará uma barra de pesquisa onde pode buscar por produtos do mercado.
Pós-condição	O cliente selecionar produtos.
Cenário alternativo	1. O produto está em falta. 2. O cliente sai do aplicativo.
Inclusão	UC 015 – Adicionar no carrinho.
Extensão	Não há.
Caso de uso – Adicionar no carrinho	
ID	UC 015
Descrição	O cliente selecionará o produto e sua quantidade atendendo o RF 015.
Ator primário	Cliente.
Pré-condição	O cliente ter buscado um produto.
Cenário principal	1. O cliente selecionará o produto desejado e informa sua quantidade.
Pós-condição	O cliente adicionar itens ao carrinho.
Cenário alternativo	1. O cliente não selecionar produtos.
Inclusão	Não há.
Extensão	UC 016 – Criar lista. UC019 – Personalizar lista
Caso de uso – Criar lista	
ID	UC 016
Descrição	O cliente criará uma lista atendendo o RF 016.
Ator primário	Cliente.
Ator secundário	Sistema.
Pré-condição	O cliente ter selecionado mais de um produto.
Cenário principal	1. O cliente seleciona produtos. 2. O cliente visualiza o carrinho e tem a opção de criar lista. 3. Quando uma lista for criada ela será adicionada ao banco de dados.
Pós-condição	O cliente poderá personalizar uma lista ou continuar adicionando itens nela.
Cenário alternativo	Não há.
Inclusão	Não há.
Extensão	UC 019 – Personalizar lista. UC015 – Adicionar no carrinho.
Caso de uso – Visualizar listas	
ID	UC 017
Descrição	O cliente poderá visualizar suas listas atendendo o RF 017.
Ator primário	Cliente.
Pré-condição	O cliente selecionará a opção de visualizar listas
Cenário principal	1. O cliente selecionará uma opção onde ele pode visualizar suas listas salvas.
Pós-condição	O cliente poderá selecionar uma lista.
Cenário alternativo	1. Não há listas criadas.
Inclusão	UC 018 – Selecionar lista
Extensão	Não há.
Caso de uso – Selecionar lista	
ID	UC 018

Descrição	O cliente poderá selecionar uma lista de compras atendendo o RF 018.
Ator primário	cliente.
Pré-condição	O cliente visualizar uma lista.
Cenário principal	1. O cliente irá escolher uma lista.
Pós-condição	O cliente personalizar a lista.
Cenário alternativo	1. Não há listas para selecionar.
Inclusão	Não há.
Extensão	UC 019 – Personalizar lista.
Caso de uso – Personalizar lista	
ID	UC 019
Descrição	O cliente poderá editar suas listas atendendo o RF 019.
Ator primário	Cliente.
Pré-condição	O cliente ter selecionado uma lista.
Cenário principal	1. O cliente poderá renomear e salvar suas listas ou adicionar itens e remover itens.
Pós-condição	O cliente poderá renomear lista, salvar lista, adicionar itens ou remover itens.
Cenário alternativo	1. O cliente saí do aplicativo.
Inclusão	Não há.
Extensão	UC 020 – Renomear lista, UC 022 – Remover itens, UC 021 – Adicionar itens e UC 015 – Adicionar no carrinho.
Caso de uso – Renomear lista	
ID	UC 020
Descrição	O cliente poderá renomear suas listas atendendo o RF 020.
Ator primário	Cliente.
Pré-condição	Existir uma lista.
Cenário principal	1. O cliente poderá renomear uma lista de compras.
Pós-condição	Salvar a lista.
Cenário alternativo	1. O cliente saí do aplicativo.
Inclusão	Não há.
Extensão	UC 023 – Salvar lista. UC019 – Personalizar lista.
Caso de uso – Adicionar itens	
ID	UC 021
Descrição	O cliente poderá adicionar itens as listas atendendo o RF 021.
Ator primário	Cliente.
Pré-condição	Existir uma lista.
Cenário principal	1. O cliente poderá adicionar o item em uma lista.
Pós-condição	Salvar a lista.
Cenário alternativo	O cliente saí do aplicativo.
Inclusão	Não há.
Extensão	UC 023 – Salvar lista. UC019 – Personalizar lista.
Caso de uso – Remover itens	
ID	UC 022
Descrição	O cliente poderá remover itens de sua lista atendendo o RF 022.
Ator primário	Cliente.
Pré-condição	Existir uma lista.
Cenário principal	1. O cliente poderá remover itens adicionados de uma lista.

Pós-condição	Salvar a lista.
Cenário alternativo	1. O cliente sai do aplicativo.
Inclusão	Não há.
Extensão	UC 023 – Salvar lista. UC019 – Personalizar lista.
Caso de uso – Salvar lista	
ID	UC 023
Descrição	O cliente poderá salvar uma lista atendendo o RF 023.
Ator primário	Cliente.
Pré-condição	Existir uma lista.
Cenário principal	1. O cliente salvará sua lista para finalizar o pedido.
Pós-condição	O cliente revisar o pedido.
Cenário alternativo	1. O cliente sai do aplicativo.
Inclusão	Não há.
Extensão	UC 020 – Renomear lista, UC 022 – Remover itens, UC 021 – Adicionar itens.
Caso de uso – Revisar pedido	
ID	UC 024
Descrição	O cliente revisará o pedido atendendo o RF 024.
Ator primário	Cliente.
Pré-condição	O cliente salva uma lista.
Cenário principal	1. O cliente revisará o pedido pelo botão revisar pedido. 2. Após a revisão ele poderá fazer o pedido
Pós-condição	O cliente poderá fazer o pedido.
Cenário alternativo	1. O cliente revisa a lista e retorna para editá-la.
Inclusão	UC 025 – Fazer pedido.
Extensão	Não há.
Caso de uso – Fazer pedido	
ID	UC 025
Descrição	O cliente poderá fazer um pedido atendendo o RF 025.
Ator primário	Cliente.
Pré-condição	O cliente ter revisado o pedido.
Cenário principal	1. Após revisado pedido o cliente apertará o botão fazer pedido.
Pós-condição	O sistema gerar e enviar o pedido para o mercado.
Cenário alternativo	1. O cliente não faz o pedido.
Inclusão	Não há.
Extensão	UC 026 – Enviar pedido e UC 027 – Ver pedido.
Caso de uso – Enviar pedido	
ID	UC 026
Descrição	O sistema enviará o pedido ao mercado atendendo o RF 026.
Ator primário	Sistema.
Pré-condição	O cliente ter feito um pedido.
Cenário principal	1. O sistema enviará o pedido do cliente ao mercado. 2. O cliente receberá uma mensagem de que o pedido foi enviado. 3. O mercado será notificado do pedido.
Pós-condição	O mercado receber e ver o pedido.
Cenário alternativo	Não há.

Inclusão	UC 033 – Atualizar lista de pedidos
Extensão	UC025 – Fazer pedido
Caso de uso – Ver pedidos	
ID	UC 027
Descrição	O cliente acompanhará o pedido atendendo o RF 027.
Ator primário	Cliente.
Pré-condição	O cliente fazer um pedido.
Cenário principal	1. O cliente acompanhará o andamento/status do pedido.
Pós-condição	O cliente receberá o pedido.
Cenário alternativo	1. Problemas de logística. 2. O cliente não estar em casa.
Inclusão	Não há.
Extensão	UC025 – Fazer pedido.
Caso de uso – Informar pedido aceito	
ID	UC 028
Descrição	O sistema deverá notificar a mensageria que o pedido foi aceito atendendo o RF 028.
Ator primário	Sistema.
Pré-condição	O mercado ter aceitado um pedido.
Cenário principal	1. O mercado aceita o pedido. 2. O sistema envia notificação para mensageria atualizar o status do pedido.
Pós-condição	O sistema atualiza o status do pedido.
Cenário alternativo	1. Problemas de logística.
Inclusão	Não há.
Extensão	UC 007 – Aceitar pedido.
Caso de uso – Carregar plataforma mercado.	
ID	UC 029
Descrição	O sistema carregará a plataforma do mercado atendendo o RF 029.
Ator primário	Sistema.
Pré-condição	A validação do login do mercado.
Cenário principal	1. O sistema carregará a plataforma do mercado.
Pós-condição	O sistema abrir o menu do mercado.
Cenário alternativo	1. Erro na validação do mercado.
Inclusão	UC 031 – Abrir menu mercado.
Extensão	Não há.
Caso de uso – Carregar plataforma cliente	
ID	UC 030
Descrição	O sistema carregará a plataforma do cliente atendendo o RF 030.
Ator primário	Sistema.
Pré-condição	A validação do login do cliente.
Cenário principal	1. O sistema carregará a plataforma do cliente.
Pós-condição	O sistema abrir o menu do cliente.
Cenário alternativo	1. Erro na validação do cliente.
Inclusão	UC 009 – Abrir menu cliente.
Extensão	Não há.
Caso de uso – Abrir menu mercado	
ID	UC 031
Descrição	O sistema exibirá o menu do mercado atendendo o RF 031.
Ator primário	Sistema.

Pré-condição	O sistema ter carregado a plataforma do mercado
Cenário principal	1. O sistema carregará o menu do mercado. 2. O mercado terá a opção de escolher entre atualizar lista de produtos ou listar pedidos.
Pós-condição	O mercado atualizará produtos ou listará os pedidos.
Cenário alternativo	1. O mercado saí do aplicativo.
Inclusão	Não há.
Extensão	Não há.
Caso de uso – Finalizar pedido	
ID	UC 032
Descrição	O mercado finalizará o pedido atendendo o RF 032.
Ator primário	Mercado.
Pré-condição	O cliente ter recebido e pagado pelo pedido.
Cenário principal	1. O mercado receberá o comprovante do pagamento. 2. O mercado finalizará o pedido no sistema.
Pós-condição	O sistema atualiza o status do pedido para finalizado.
Cenário alternativo	1. O mercado não recebe o comprovante. 2. O cliente não paga pelo pedido.
Inclusão	Não há.
Extensão	Não há.
Caso de uso – Atualizar lista de pedidos	
ID	UC 033
Descrição	A plataforma do mercado receberá uma notificação do sistema de novo pedido atendendo o RF026.
Ator primário	Sistema.
Pré-condição	O cliente ter realizado um pedido.
Cenário principal	1. O sistema enviará notificação de novo pedido. 2. A plataforma do mercado atualizará a lista de pedidos.
Pós-condição	O mercado listar pedidos.
Cenário alternativo	1. O mercado não recebe o pedido.
Inclusão	Não há.
Extensão	Não há.
Caso de uso – Atualizar status pedidos	
ID	UC 034
Descrição	A plataforma do mercado enviará notificação contendo status do pedido atendendo o RF034.
Ator primário	Sistema.
Pré-condição	O mercado ter informado atualização no pedido;
Cenário principal	3. O mercado enviará uma notificação para o cliente informando pedido aceito ou separado e pronto para entrega. 4. O status do pedido é atualizado para o cliente.
Pós-condição	O cliente ver o status do pedido.
Cenário alternativo	2. O mercado não recebe o pedido.
Inclusão	UC 028 – Informar pedido aceito.
Extensão	Não há.
Caso de uso – Selecionar plataforma	

ID	UC 035
Descrição	O sistema irá selecionar a plataforma mercado ou cliente de acordo com login validado, atendendo o RF035.
Ator primário	Sistema.
Pré-condição	O sistema ter validado login de acesso;
Cenário principal	<ol style="list-style-type: none"> 1. O sistema validará as entradas do login consultando no banco de dados. 2. O sistema irá recuperar do banco de dados o tipo de usuário (mercado ou cliente). 3. O sistema direcionará para carregar a plataforma do mercado ou do cliente.
Pós-condição	O sistema irá carregar a plataforma de acordo com o tipo de usuário (mercado ou cliente).
Cenário alternativo	1. O cliente não ter cadastro.
Inclusão	Não há.
Extensão	UC 004 – Validar login, UC 029 – Carregar plataforma mercado e UC 030 – carregar plataforma cliente.
Caso de uso – Enviar form cadastro	
ID	UC 036
Descrição	Os usuários, cliente ou mercado irão enviar os dados preenchidos no formulário de cadastro atendendo o RF036.
Ator primário	Cliente ou mercado.
Pré-condição	O usuário ter preenchido os dados de cadastro.
Cenário principal	<ol style="list-style-type: none"> 1. O usuário preenche os dados no formulário e realiza o envio. 2. O sistema receberá os dados do cadastro. 3. O sistema enviará os dados para a validação.
Pós-condição	O sistema validará o cadastro.
Cenário alternativo	Não há.
Inclusão	Não há.
Extensão	UC 002 – Validar cadastro
Caso de uso – Enviar form login	
ID	UC 037
Descrição	O sistema enviará os dados de login para a validação atendendo o RF 037.
Ator primário	Sistema.
Pré-condição	O usuário ter preenchido o cadastro.
Cenário principal	<ol style="list-style-type: none"> 1. O usuário preenche os dados no formulário e realiza o envio. 2. O sistema receberá os dados de login. 3. O sistema enviará os dados para a validação.
Pós-condição	O sistema validará o login.
Cenário alternativo	Não há.
Inclusão	Não há.
Extensão	UC 004 – Validar login.
Caso de uso – Separar produtos	
ID	UC 038
Descrição	O mercado separará os produtos atendendo o RF 038.
Ator primário	Mercado.
Pré-condição	O mercado ter aceitado o pedido.
	<ol style="list-style-type: none"> 1. O mercado receberá o pedido. 2. O mercado separará os produtos.

Cenário principal	3. O mercado irá informar o cliente que o pedido foi separado.
Pós-condição	O mercado informar que o pedido foi preparado e está pronto para entrega.
Cenário alternativo	Não há.
Inclusão	Não há.
Extensão	UC 008 – Informar entrega, UC 007 – Aceitar pedido.

3.5 Diagrama de Classes

Um Diagrama de Classes é uma representação gráfica que detalha a estrutura de um sistema orientado a objetos, mostrando suas classes, atributos, métodos e os relacionamentos entre os objetos. Segundo Gilleanes Guedes (2018) "o Diagrama de Classes é uma ferramenta essencial da UML (*Unified Modeling Language*) que permite a modelagem dos aspectos estáticos de um sistema". Ele ajuda a ilustrar como os diferentes componentes do sistema interagem entre si e como os dados são estruturados e manipulados. Fowler (2004) explica que "um diagrama e classes é a principal construção usada no design orientado a objetos para mostrar as classes de um sistema e seus relacionamentos".

A importância de um Diagrama de Classes reside na sua capacidade de fornecer uma visão clara e organizada da arquitetura do sistema (Larman, 2002).

Destaca que "um diagrama de classes fornece uma visão global da arquitetura de um sistema, essencial para o entendimento e implementação do design orientado a objetos". Essa visualização facilita a comunicação entre os membros da equipe de desenvolvimento, tornando mais fácil identificar e resolver problemas de design. Além disso, ajuda a garantir que todos os requisitos funcionais e não funcionais do sistema sejam atendidos. Conforme Guedes (2018), "os diagramas de classes são fundamentais para a análise e design de sistemas complexos, proporcionando uma base sólida para a codificação e manutenção do *software*". Portanto, um Diagrama de Classes é uma ferramenta indispensável para o desenvolvimento de software eficiente e bem-estruturado.

O projeto consiste em várias classes interconectadas, sendo a classe "Pedido" o ponto principal do sistema, responsável por gerenciar os pedidos dos clientes. A classe "Pedido" possui atributos como código do pedido, nome do cliente, endereço, telefone, valor total, horário, observação, status do pedido e lista de produtos. Esta

classe inclui métodos para atualizar o status do pedido e está diretamente relacionada a várias outras classes, facilitando a integração completa do sistema.

A classe "Usuario" é fundamental para a autenticação e autorização no sistema, permitindo o cadastro e a validação de usuários, login no sistema e envio de formulários de cadastro e login. Existem subclasses específicas para clientes ("Cliente") e mercados ("Mercado"), que herdam de "Usuario" e adicionam funcionalidades específicas, como ver e editar cadastro, listar e aceitar pedidos, e finalizar pedidos.

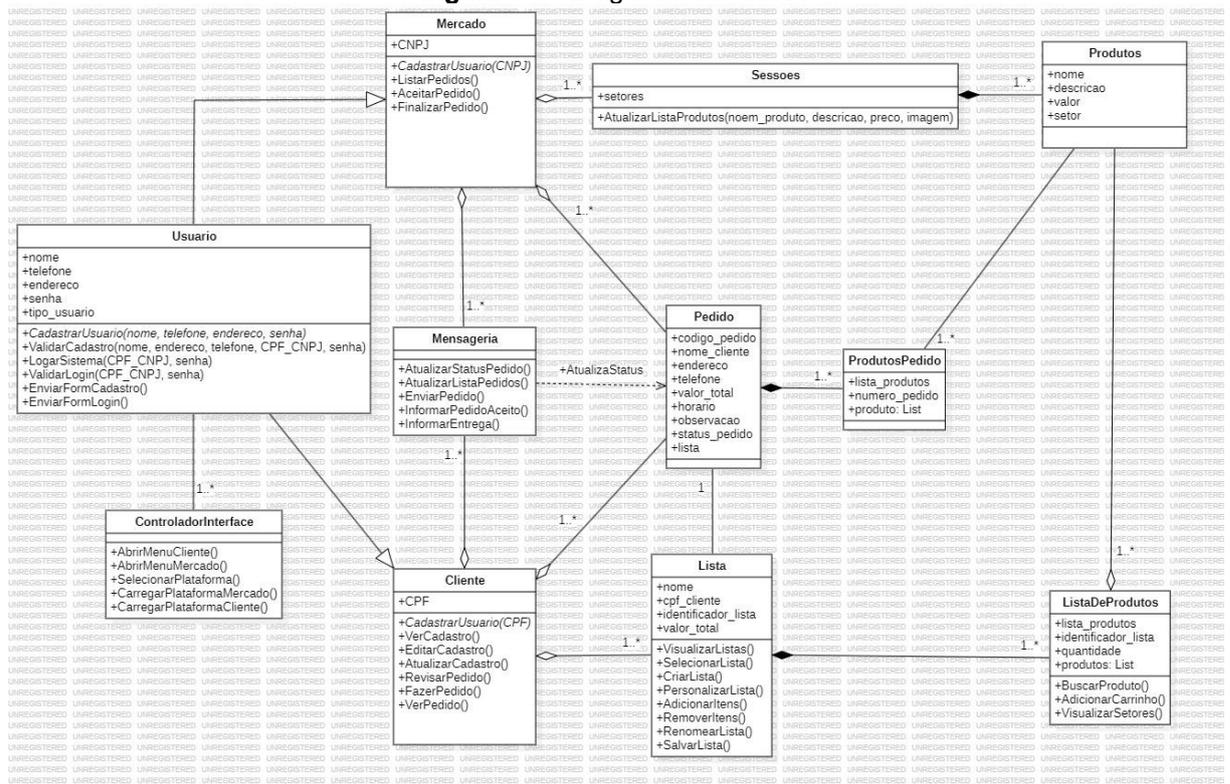
A classe "Mensageria" desempenha um papel crucial na comunicação do sistema, com métodos para atualizar o status dos pedidos, atualizar listas de pedidos, enviar pedidos e informar sobre aceitação e entrega de pedidos. Além disso, a classe "Produtos" e "ProdutosPedido" gerenciam os detalhes dos produtos, incluindo nome, descrição, valor, setor e lista de produtos incluídos em cada pedido.

A classe "Lista" permite a criação e gestão de listas de compras, com métodos para visualizar, selecionar, criar, personalizar, adicionar itens, remover itens e salvar listas. A "ListaDeProdutos" complementa essa funcionalidade, facilitando a busca de produtos, adição ao carrinho e visualização de setores.

A classe "Sesoes" organiza os produtos em setores e atualiza a lista de produtos com detalhes como nome, descrição, preço e imagem.

Por fim, a classe "ControladorInterface" gerencia a interface do usuário, fornecendo métodos para abrir menus de cliente e mercado, selecionar a plataforma e carregar as plataformas específicas para cada tipo de usuário. Ela possui uma relação com a classe "Usuario" e inclui métodos como `abrirMenuCliente()`, `abrirMenuMercado()`, `selecionarPlataforma()`, `carregarPlataformaMercado()` e `carregarPlataformaCliente()`, facilitando a navegação e interação dos usuários com o sistema.

Figura 10 – Diagrama de Classe



Fonte: Elaborado pelos autores.

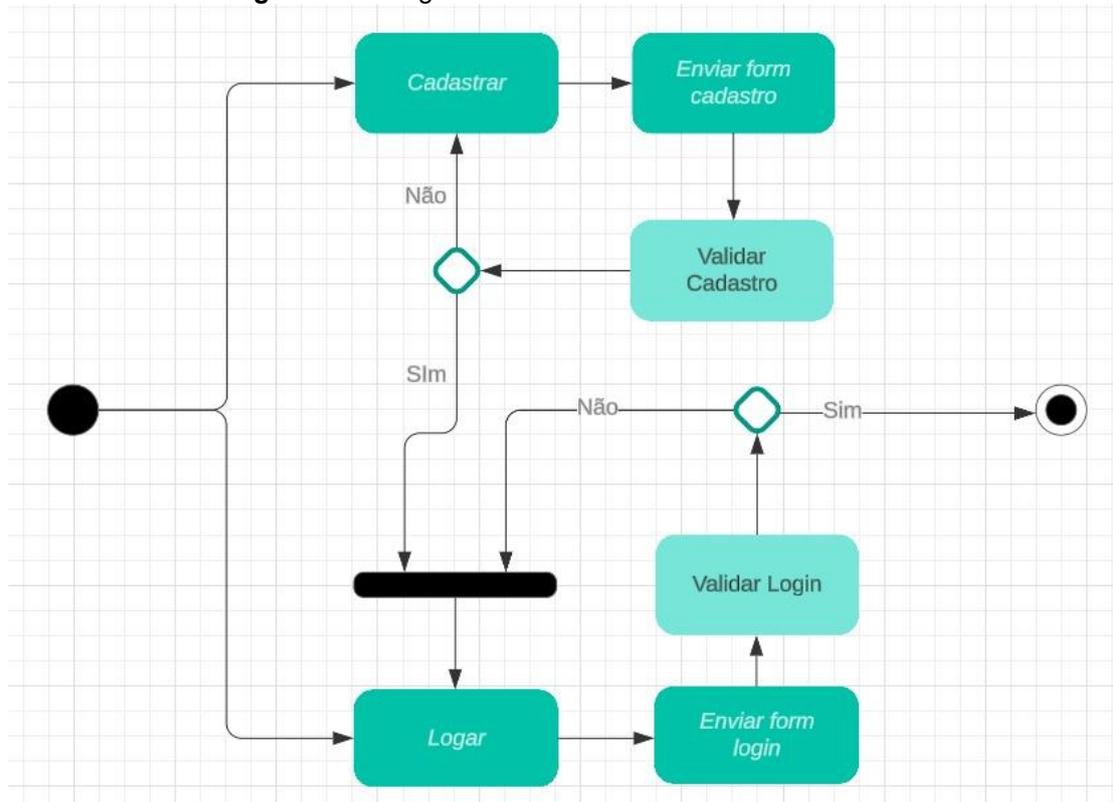
3.6 Diagrama de Atividades

A importância de um Diagrama de Atividades reside na sua capacidade de proporcionar uma visão clara e detalhada dos processos internos do sistema, facilitando a identificação de possíveis melhorias e gargalos. Este diagrama é essencial para a modelagem de fluxos de trabalho complexos, permitindo aos desenvolvedores e analistas de sistemas entenderem e comunicarem o comportamento do sistema de maneira eficaz. Ele ajuda a alinhar os requisitos de negócio com as implementações técnicas, garantindo que todos os stakeholders tenham uma compreensão comum dos processos do sistema (Larman, 2002).

Um Diagrama de Atividades é um tipo de diagrama da UML (*Unified Modeling Language*) que representa o fluxo de trabalho ou o processo de atividades no sistema, mostrando a sequência de ações, decisões e a interação entre diferentes partes do sistema. Este diagrama é particularmente útil para visualizar o comportamento dinâmico do sistema, ilustrando como as atividades se desenrolam ao longo do tempo (Booch; Rumbaugh; Jacobson, 2005).

O seguinte diagrama representa o processo de atividades decorrentes do acesso ao sistema, onde é demonstrado as funções de cadastro e login de usuários.

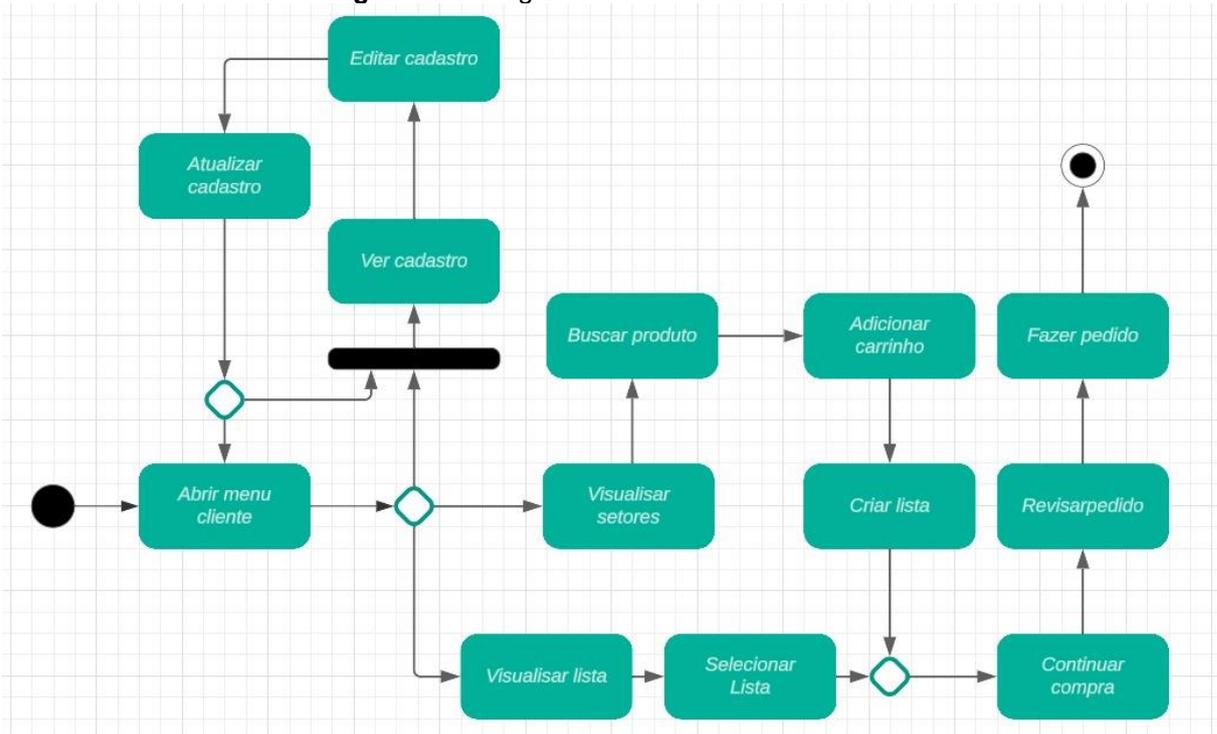
Figura 11 – Diagrama de Atividade - Acesso ao sistema



Fonte: Elaborado pelos autores.

O próximo diagrama representa o fluxo de trabalho que são executados pelos usuários determinados clientes. São representadas as sequencias de ações para editar cadastro, selecionar listas, adicionar produtos no carrinho e realizar pedido.

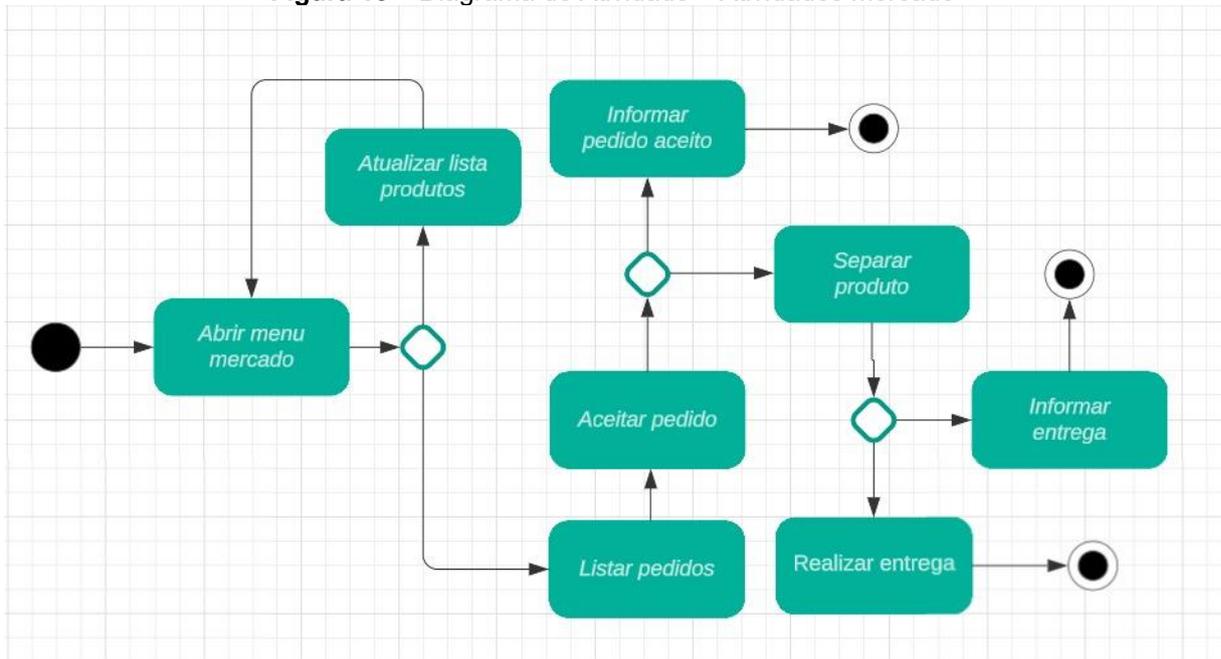
Figura 12 – Diagrama de Atividade – Atividades cliente



Fonte: Elaborado pelos autores.

O diagrama a seguir representa as atividades relacionadas ao mercado, que realiza as ações de atualizar os produtos disponíveis no app e a tratativa de pedidos, listando, aceitando, alterando o status dos pedidos. Possui três saídas do fluxo relacionadas a comunicação com a plataforma do cliente para atualização do status do pedido.

Figura 13 – Diagrama de Atividade – Atividades mercado



Fonte: Elaborado pelos autores.

3.7 Diagrama de Estados

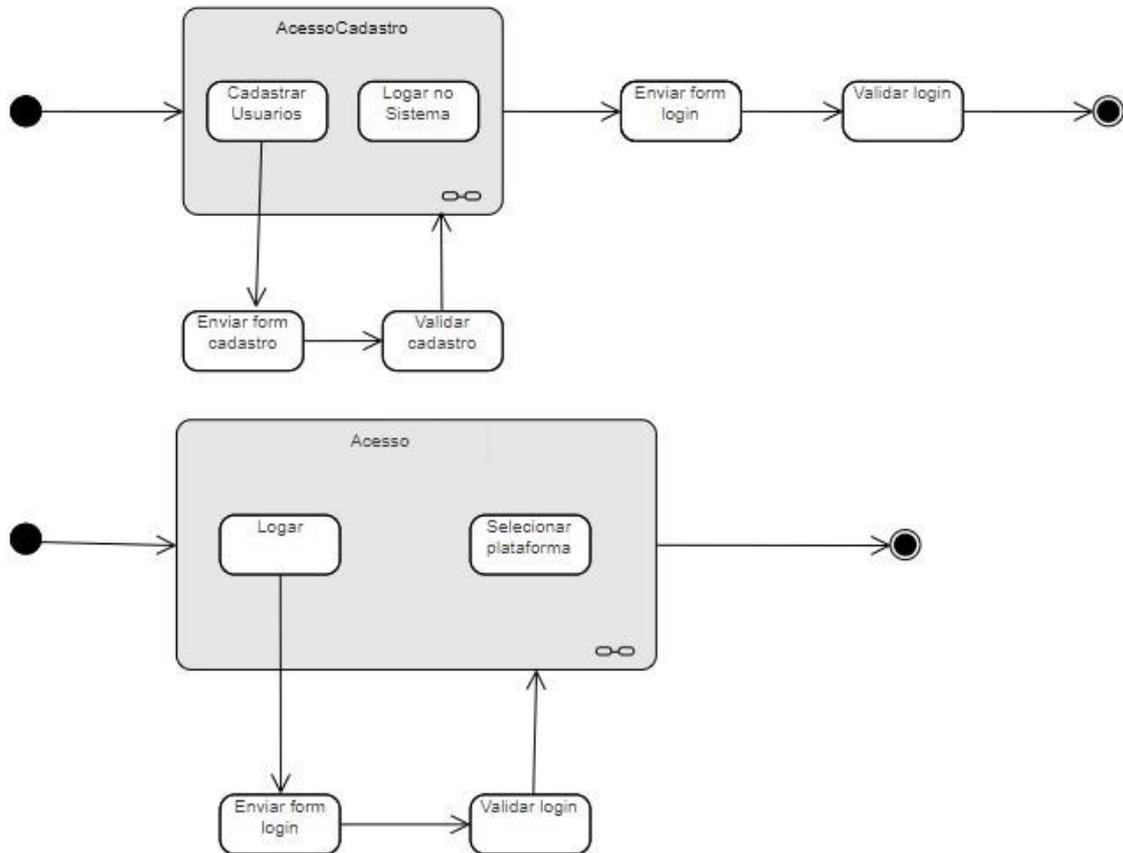
Um Diagrama de Estados, também conhecido como Diagrama de Máquina de Estados, é um tipo de diagrama da UML que descreve os estados pelos quais um objeto ou um sistema pode passar ao longo de seu ciclo de vida, bem como os eventos que causam essas mudanças de estado. Este diagrama ilustra como um objeto reage a eventos e transita de um estado para outro, ajudando a modelar o comportamento dinâmico do sistema. A partir das classes do diagrama, podemos inferir o estado do objeto pedido como “Acesso ao sistema” e “Fluxo mercado”. Abaixo, os dois diagramas de máquina de estado.

A importância de um Diagrama de Estados está em sua capacidade de fornecer uma visão clara e precisa do comportamento de um sistema ou objeto em resposta a diferentes eventos. Este diagrama é particularmente útil para modelar sistemas reativos onde a resposta a eventos é crítica.

Ele ajuda os desenvolvedores a entenderem e especificarem os requisitos de comportamento do sistema, facilitando a detecção de estados redundantes ou não alcançáveis, o que pode levar a um design mais robusto e eficiente (Larman, 2002).

O diagrama de estado representado a seguir apresenta o estado do objeto usuário ao realizar o acesso no sistema. Foi representado os processos de cadastro e login.

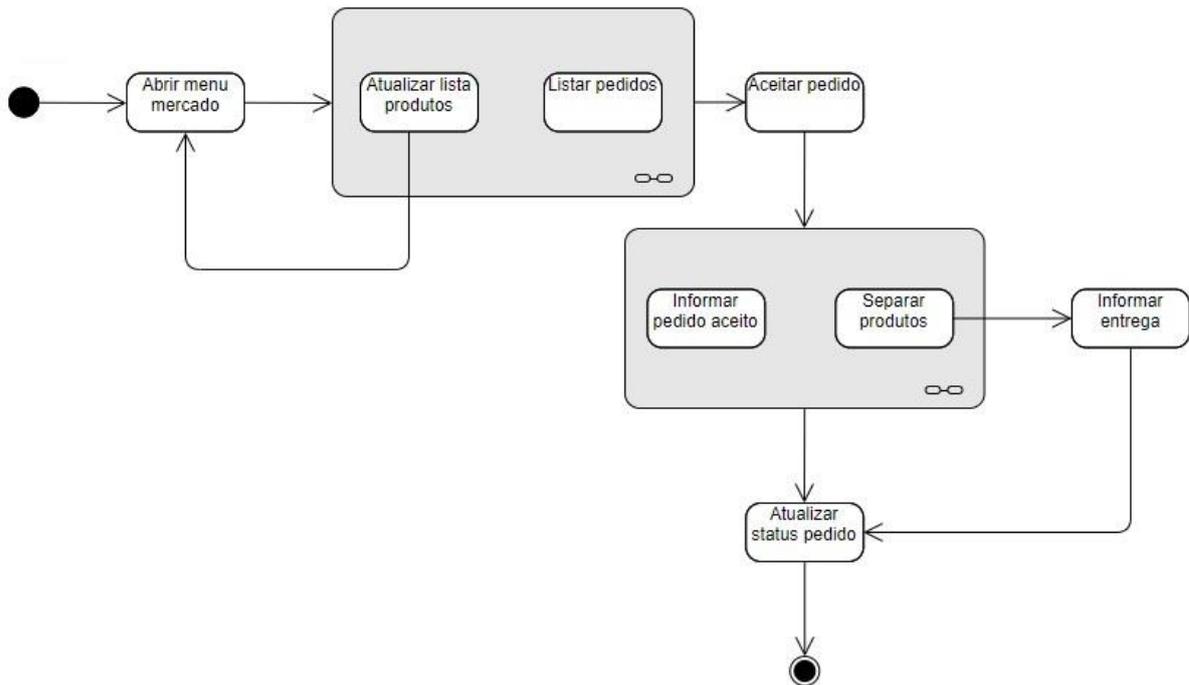
Figura 14 – Diagrama de Estado - Acesso ao sistema



Fonte: Elaborado pelos autores.

O diagrama a seguir apresenta os estados do objeto usuário do mercado durante o uso da plataforma. O fluxo segue até a segunda máquina de estado onde é alterado o contexto do usuário do mercado para o controle do sistema fazer a notificação para o usuário.

Figura 15 – Diagrama de Estados – Fluxo mercado



Fonte: Elaborado pelos autores.

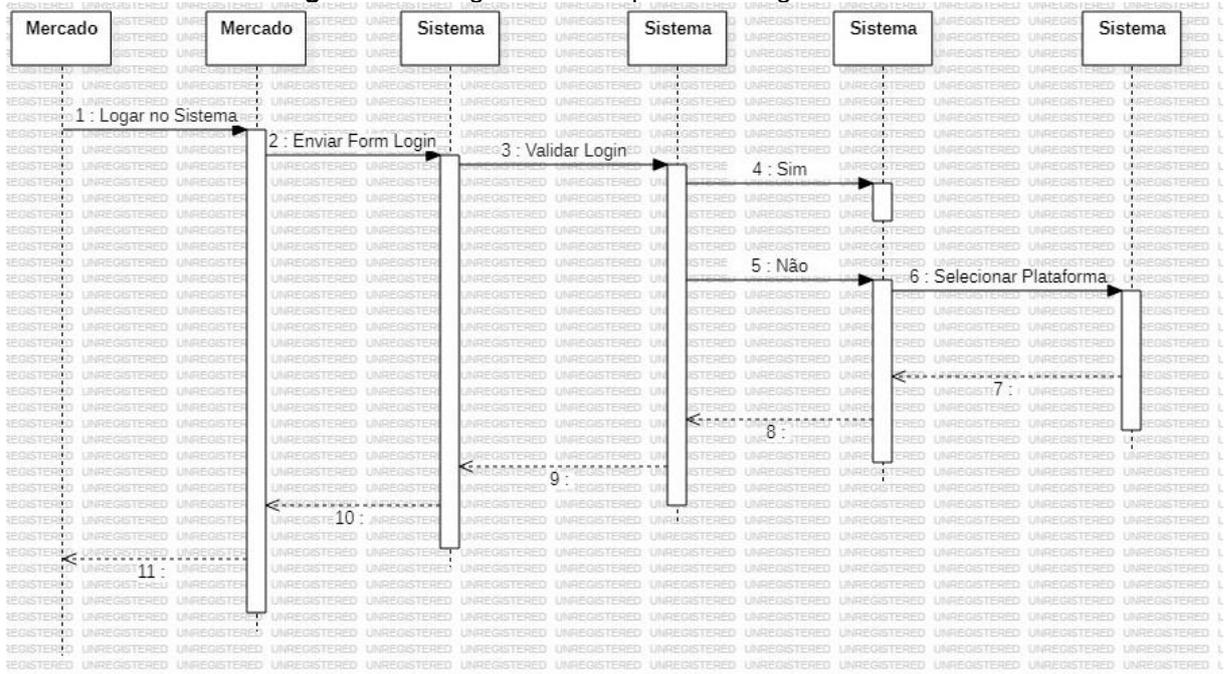
3.8 Diagrama de Sequência

Um Diagrama de Sequência é um modelo de diagrama de interação da UML que demonstra como os objetos interagem em um cenário de tempo. Ele representa os objetos envolvidos na interação e a sequência de mensagens trocadas entre eles para realizar uma funcionalidade ou um processo específico. No diagrama, os objetos são dispostos horizontalmente no topo e o tempo progride verticalmente, de cima para baixo, mostrando o fluxo de controle entre os objetos (Fowler, 2004).

A importância de um Diagrama de Sequência está na sua capacidade de capturar a lógica de tempo e a ordem de interações entre os objetos em um sistema. Este diagrama é essencial para a modelagem dinâmica do sistema, permitindo que os desenvolvedores e analistas compreendam como as funcionalidades são realizadas em termos de comunicação entre objetos. Ele ajuda a identificar os pontos de integração e as dependências entre diferentes componentes do sistema, o que é crucial para garantir a coesão e a corretude do design (Booch; Rumbaugh; Jacobson, 2005)

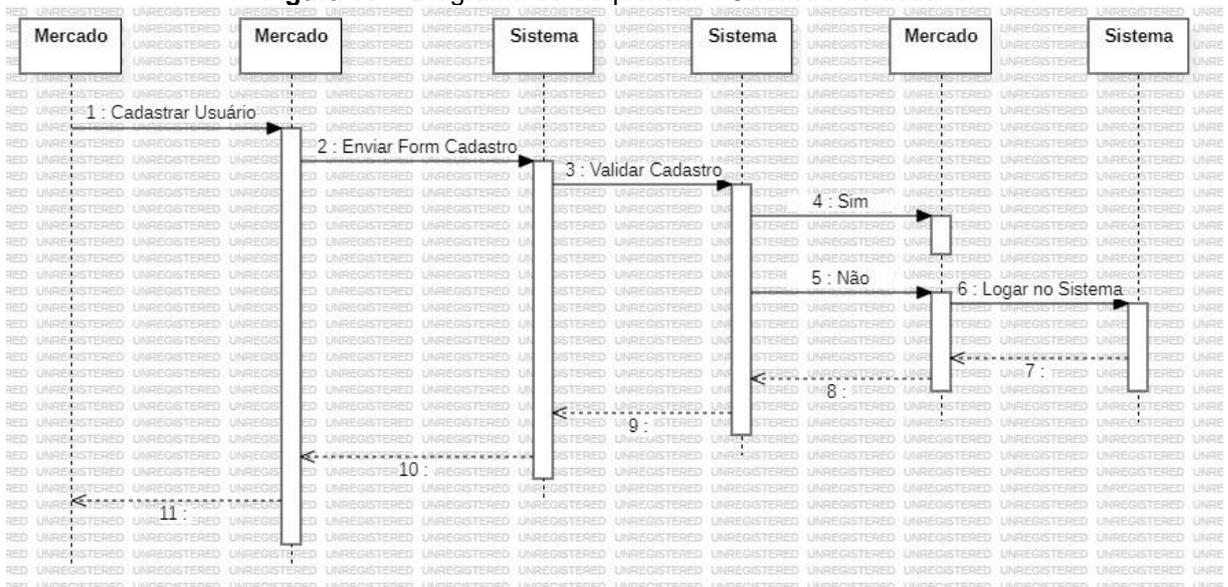
Foram desenvolvidos três diagramas de sequência, buscando representar contextos específicos do sistema. São eles os cenários de login no sistema, cadastro de usuário e o ciclo dos usuários do mercado no processo de aceitar um pedido até que o pagamento dele seja efetuado.

Figura 16 – Diagrama de Sequência – Login no sistema



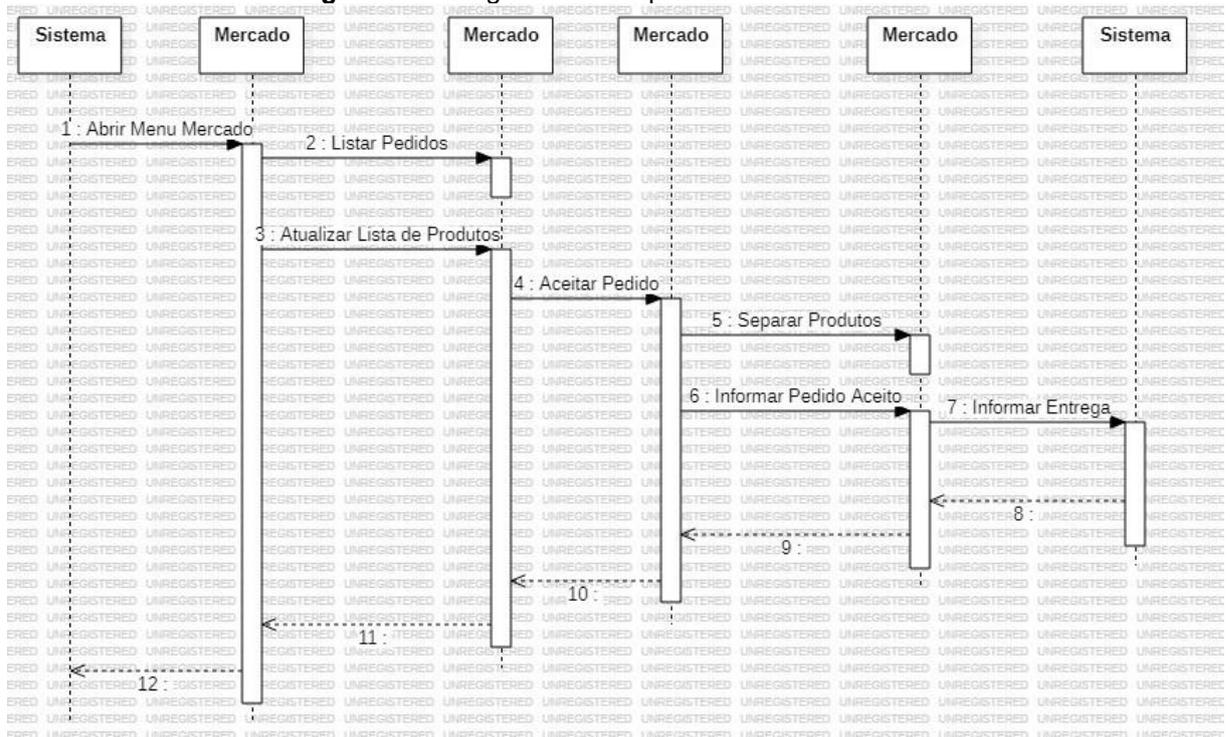
Fonte: Elaborado pelos autores.

Figura 17 – Diagrama de Sequência – Cadastro de usuário



Fonte: Elaborado pelos autores.

Figura 18 – Diagrama de Sequência – Ciclo mercado



3.9 Diagrama Entidade-Relacionamento

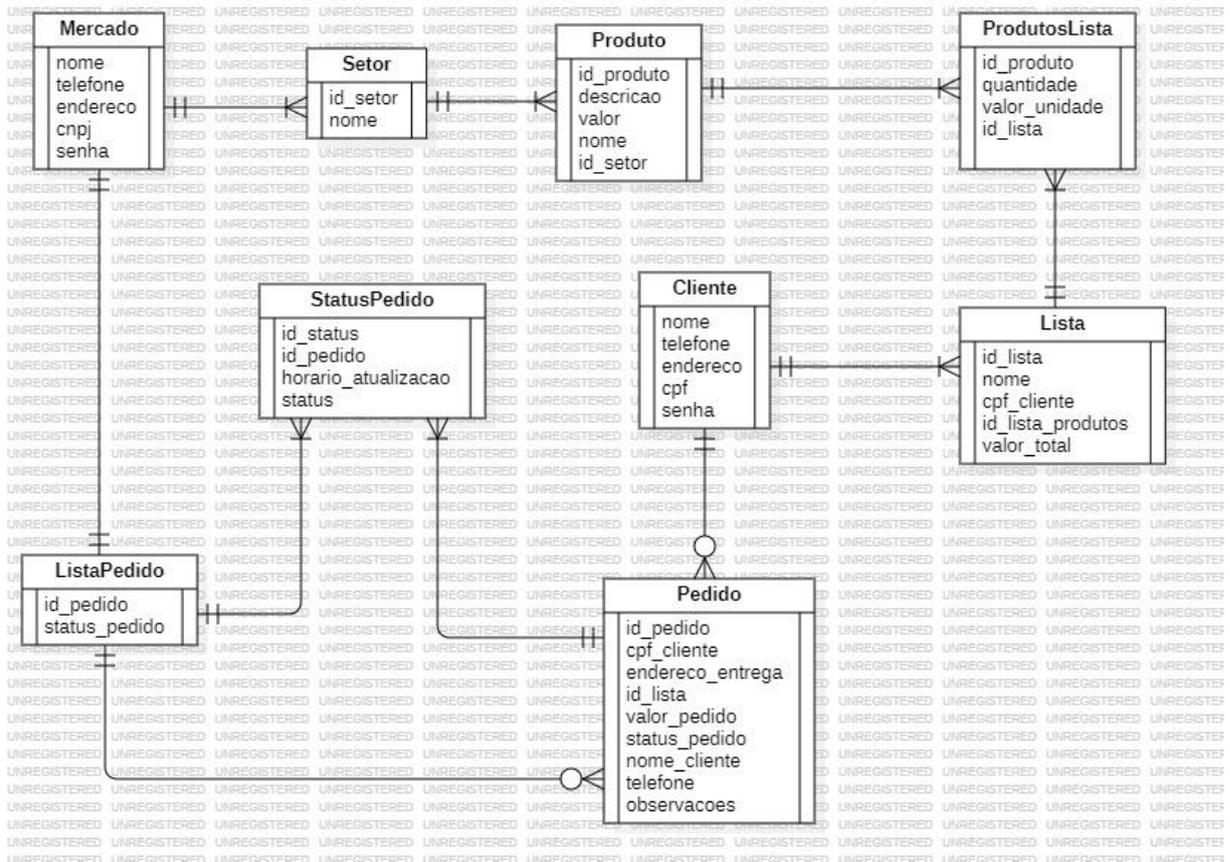
O Diagrama Entidade-Relacionamento (DER) constitui uma ferramenta essencial no processo de modelagem conceitual de bancos de dados, servindo como ponte entre os requisitos do sistema e sua implementação técnica. Desenvolvido como linguagem visual para representar a estrutura de dados, o DER permite a abstração dos elementos fundamentais de um domínio de negócios por meio de três componentes principais: entidades (objetos ou conceitos relevantes), atributos (características desses objetos) e relacionamentos (associações entre as entidades). Esta modelagem independe de plataformas tecnológicas específicas, focando na representação fiel da realidade organizacional que será informatizada. Sua importância está na capacidade de oferecer uma perspectiva clara e concisa da estrutura de dados antes mesmo da fase de implementação, facilitando a comunicação entre analistas, desenvolvedores e os usuários finais, além de prevenir inconsistências que poderiam surgir em fases avançadas do desenvolvimento.

A construção de um DER eficaz segue um processo sistemático que começa com a identificação das entidades principais do domínio em estudo, como clientes, produtos ou pedidos no contexto de um sistema comercial. Em seguida, definem-se

os atributos relevantes para cada entidade e estabelecem-se os relacionamentos entre elas, com suas respectivas cardinalidades (um-para-um, um-para-muitos ou muitos-para-muitos). Esta representação gráfica documenta a estrutura de dados e revela as regras de negócio implícitas nas relações entre as entidades. No contexto do projeto em análise, o DER demonstra utilidade para modelar as relações entre clientes, pedidos e produtos, podendo estruturar hierarquias de categorias e fluxos de atualização de estoque. Quando adequadamente elaborado, o diagrama serve como base sólida para a posterior transformação em modelo relacional, garantindo a integridade, eficiência e escalabilidade do banco de dados implementado.

O diagrama a seguir apresenta a estrutura de banco de dados, organizando os dados em entidades principais como Mercado, Produto, Cliente, ListaPedido, StatusPedido e ProductsLista.

Figura 19 – Diagrama de Entidade-Relacionamento

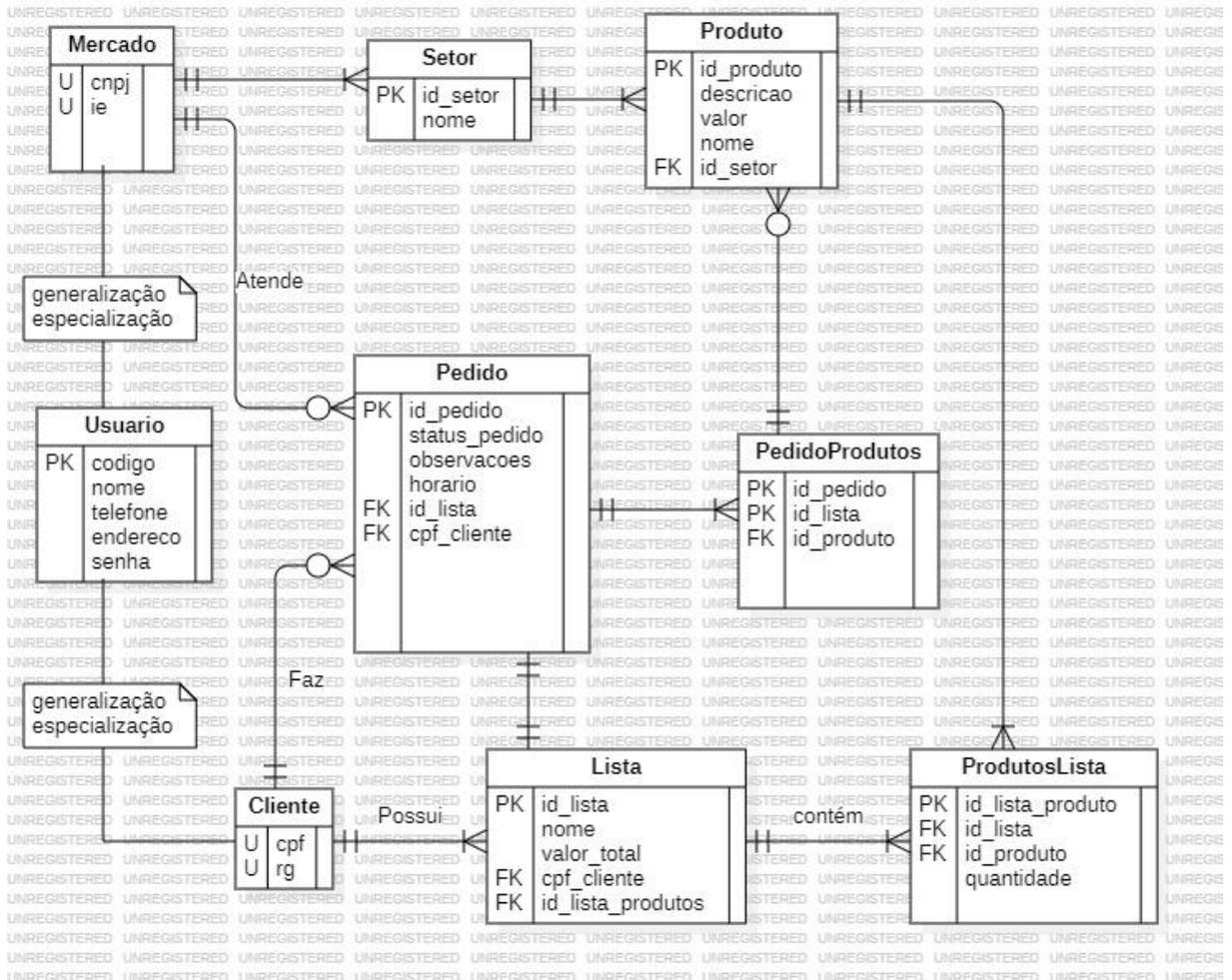


Fonte: Elaborado pelos autores.

O diagrama a seguir apresenta a utilização do conceito de generalização e especialização, partindo da entidade "Usuário" como base, que se desdobra em

"Cliente", incorporando atributos específicos como CPF e RG. Além disso, o sistema é composto pelas entidades "Setor", "Produto", "Pedido" e "Lista", que estabelecem relações entre si por meio de chaves primárias e estrangeiras. Tabelas de junção, como "PedidoProdutos" e "ProductosLista".

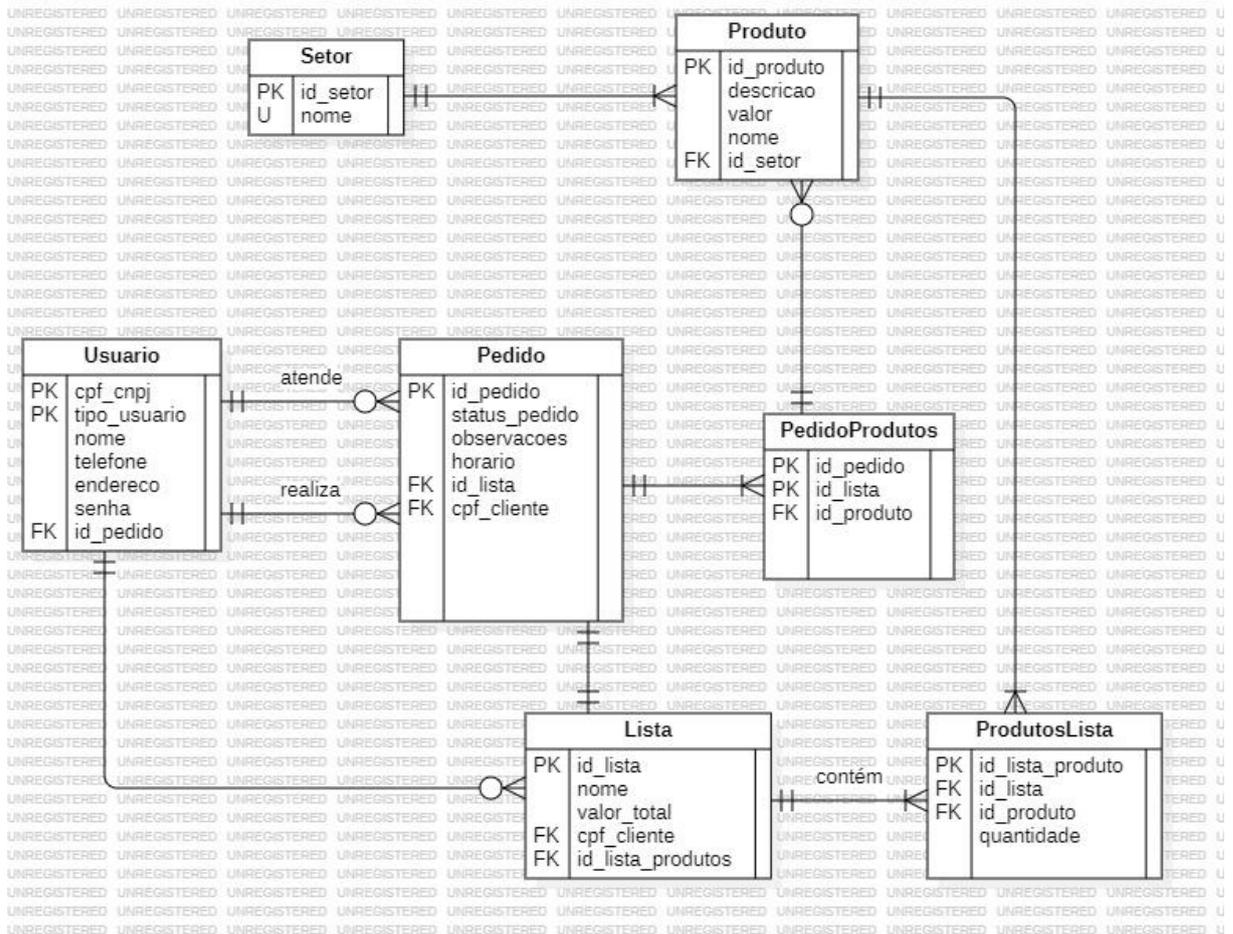
Figura 20 – Segunda figura Diagrama de Entidade-Relacionamento



Fonte: Elaborado pelos autores.

O diagrama a seguir apresenta uma estrutura com entidades principais que refletem as operações essenciais. A modelagem utiliza as chaves primárias e relacionamentos, organizando-os desde o cadastro de usuários até o processamento de pedidos.

Figura 21 – Terceira figura Diagrama de Entidade-Relacionamento



Fonte: Elaborado pelos autores.

4 Ferramentas e Métodos

O desenvolvimento do projeto seguiu metodologias com fundamentos na Engenharia de Software, utilizando técnicas para eliciação de requisitos, mapeamento de questões problema, matrizes de rastreabilidade e diagramação fundamentada no padrão UML. O uso do Kanban auxiliou no gerenciamento do fluxo de trabalho, garantindo que as atividades fossem divididas em tarefas menores e exequíveis, promovendo maior produtividade e permitindo ajustes contínuos durante o processo de desenvolvimento. Para o desenvolvimento do protótipo de *software*, foram utilizadas diversas ferramentas que cobrem diferentes aspectos do ciclo de vida do projeto, desde o gerenciamento até a documentação, diagramação e prototipação.

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e suporte à comunidade. Além disso, as presentes ferramentas citadas, possuem uma documentação abrangente, tutoriais e recursos

que estão disponíveis na comunidade de desenvolvedores, o que torna mais fácil para os desenvolvedores aprenderem e implementarem as soluções. A escolha dessas ferramentas também foi influenciada pela preferência pessoal da equipe de desenvolvimento e experiência prévia no uso delas.

4.1 Ferramentas para Gerenciamento do Projeto

1. **Miro**: Plataforma colaborativa online para criação de quadros visuais, utilizada para brainstorming, mapeamento de processos e organização de ideias em equipe.
2. **Microsoft Word**: Processador de textos essencial para elaboração de documentos formais, relatórios e toda a documentação textual do projeto.
3. **Microsoft Excel**: Planilha eletrônica utilizada para criação de cronogramas, matrizes de análise e controle de atividades.
4. **Git e GitHub**: Sistema de controle de versão e plataforma de hospedagem de código, essenciais para versionamento de documentos e gerenciamento do código fonte.
5. **Bloco de Notas**: Editor de texto simples utilizado para anotações rápidas e registro de ideias durante o desenvolvimento.

4.2 Ferramentas para Diagramação

1. **Camunda**: Plataforma para modelagem de diagramas BPMN com precisão técnica.
2. **Miro**: Utilizado para criação de diagramas colaborativos nos estágios iniciais.
3. **StarUML**: Ferramenta de modelagem UML para criação de diagramas de classes e casos de uso.
4. **Lucidchart**: Aplicativo web para criação de diagramas profissionais e fluxogramas.
5. **Visual Paradigm**: Suite completa para modelagem UML e BPMN.

4.3 Ferramentas para Prototipação

1. **Canva**: Plataforma de design gráfico para criação de wireframes iniciais.
2. **Figma**: Ferramenta profissional para desenvolvimento de protótipos interativos.

4.4 Metodologias Utilizadas

1. **Técnicas de Elicitação de Requisitos**: Abordagens sistemáticas para identificação dos requisitos.
2. **Mapeamento de Questões Problema**: Metodologia para análise estruturada de problemas.
3. **Matrizes de Rastreabilidade**: Técnica para garantir atendimento aos requisitos.
4. **Diagramação UML**: Padrão internacional para modelagem visual de sistemas.
5. **Kanban**: Método ágil para gestão visual do fluxo de trabalho.

4.5 Ferramentas de Desenvolvimento

1. **MySQL**: Neste projeto, foi utilizado o banco de dados MySQL para armazenamento e gerenciamento das informações. A escolha pelo MySQL se deu por ser uma solução robusta, amplamente utilizada no mercado, com código aberto e excelente desempenho para aplicações de pequeno a grande porte. Além disso, sua compatibilidade com diversas linguagens de programação e ferramentas de desenvolvimento facilitou a integração com o sistema proposto, garantindo maior eficiência no desenvolvimento e na manipulação dos dados.
2. **Java Script**: No desenvolvimento do projeto, foram utilizadas as tecnologias HTML, CSS e JavaScript para o front-end. O HTML foi responsável pela estruturação das páginas, o CSS pela estilização e definição do layout visual, enquanto o JavaScript permitiu a criação de interações dinâmicas, proporcionando uma experiência mais fluida e intuitiva ao usuário. No back-end, o JavaScript foi utilizado por meio do Node.js, possibilitando a construção de uma aplicação robusta e escalável. A escolha por essas tecnologias se deu por

sua ampla adoção no mercado, facilidade de aprendizado, grande comunidade de suporte e a possibilidade de utilizar JavaScript tanto no cliente quanto no servidor, promovendo maior integração e agilidade no desenvolvimento.

5 Desenvolvimento

Foram desenvolvidas quatro telas nesta fase do projeto. Abaixo elas estão apresentadas, juntamente as suas respectivas funcionalidades.

Figura 22 – Tela cadastro

Cadastre-se

Se você já possui uma conta conosco, por favor, faça seu login na página de login

Informações da Conta

Nome CPF

Celular

E-mail

Senha Confirmar senha

Logradouro

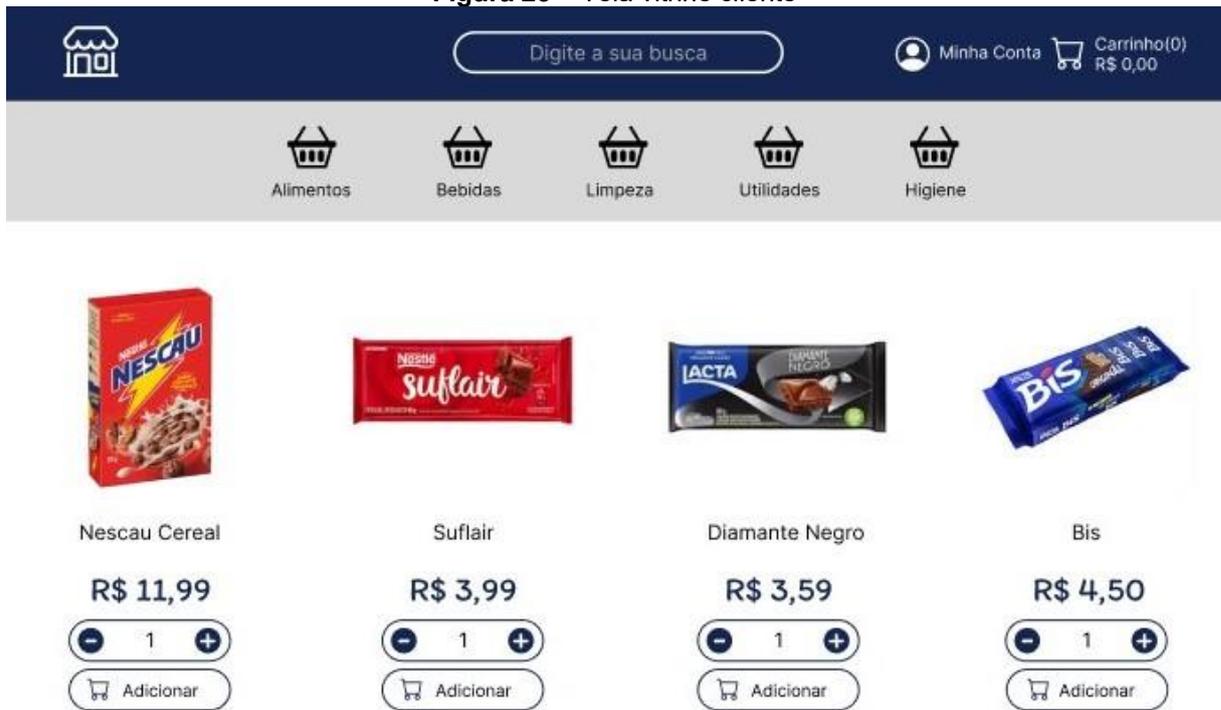
Bairro CEP Cidade

Finalizar cadastro

Fonte: Elaborado pelos autores.

Tela de cadastro: 10 campos de entrada de dados que serão validados e redirecionarão o usuário ao login do sistema.

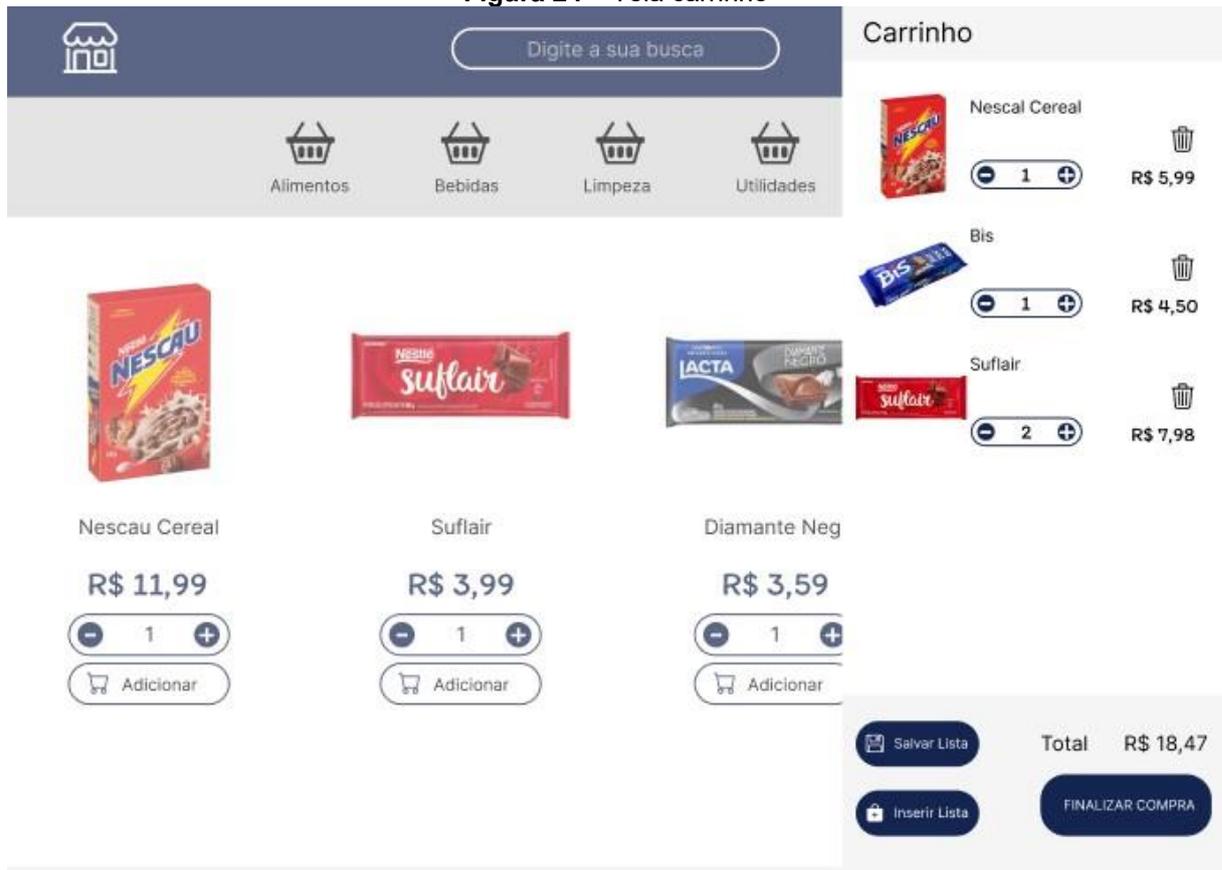
Figura 23 – Tela vitrine cliente



Fonte: Elaborado pelos autores.

Tela de Vitrine Cliente: apresenta no índice opções de acesso a busca de produtos, conta e carrinho de compras. Possui um subtítulo com setores Alimentos, Bebidas, Limpeza, Utilidades, Higiene. No corpo central da página é apresentada uma vitrine de produtos em células com imagem do produto, nome, preço unitário, opção de adicionar e retirar e adicionar no carrinho.

Figura 24 – Tela carrinho



Fonte: Elaborado pelos autores.

Tela de carrinho: Se apresenta no canto direito da página, mostrando a lista de produtos, com sua imagem, nome, quantidade e opção para alterar, opção de excluir o produto e no final da página as opções para salvar lista, Inserir Lista, valor total e Finalizar Compra.

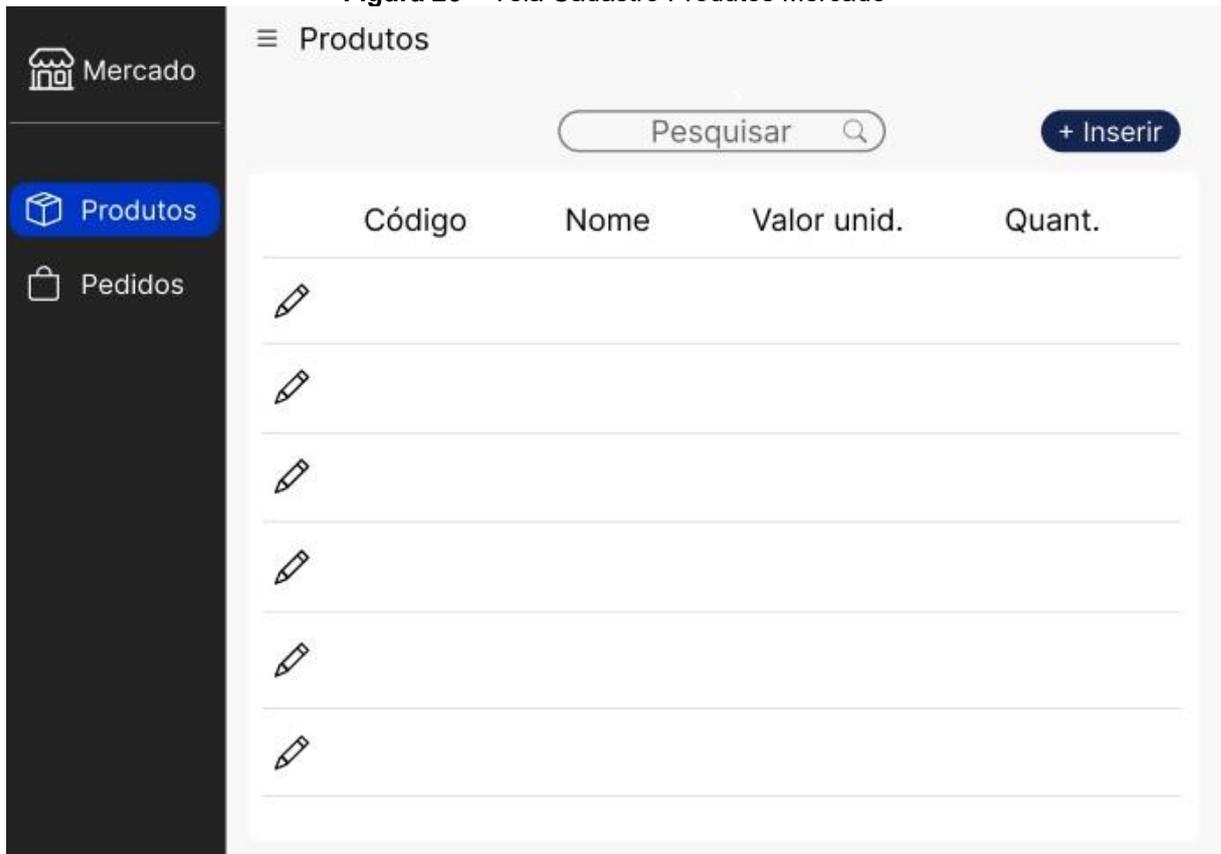
Figura 25 – Tela de listas criadas



Fonte: Elaborado pelos autores.

Tela de Listas Criadas: Ela apresenta o índice e o subtítulo da página da Vitrine Cliente, seguida de um menu no canto esquerdo com as opções de Listas, que representa a tela atual, e Minha Conta que redireciona para o cadastro do usuário. No corpo central é apresentado o botão para inserir listas, uma barra de busca para produtos e a listagem em vertical das listas criadas, com seu nome, valor total, opção de edição e enviar para o carrinho.

Figura 26 – Tela Cadastro Produtos Mercado



Fonte: Elaborado pelos autores.

Tela de Cadastro Produtos Mercado: Tela onde os dados de produtos serão armazenados, atualizados. Ela apresenta um menu no canto esquerdo com as opções de Produtos, que representa a tela atual, e Pedidos que redireciona para o gerenciamento de pedidos. Tem uma barra de busca de produtos, botão para inserir e uma lista de produtos em forma de tabela com os índices, Código, Nome, Valor unid. E Quant., os produtos são listados na vertical com botão para editar no canto esquerdo.

6 Resultados e Discussão

Formulário de Análise de Portabilidade

Configuração mínima

Hardware e Software do usuário final:

Desktop ou Notebook: equipamento com capacidade de processamento adequada para executar navegadores web modernos sem

lentidão (Arquitetura x86-64bit, Dual-Core de 2GHz ou superior, 4GB RAM ou superior);

Navegadores suportados: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, outros navegadores baseados no Chromium também são suportados;

Sistema Operacional: Windows 10 ou mais recente, MacOS 10.14 ou mais recente, Linux distribuições modernas (Ubuntu 20.04+);

Resolução de Tela: recomendado mínimo 1280x768 pixels;

Conectividade com Internet: conexão banda larga com velocidade mínima de 20Mbps para download e upload (valores inferiores a 20Mbps pode funcionar, porém com intermitência).

Configuração do Servidor:

Arquitetura: x86-64bit;

Processador: mínimo 4 núcleos ou superior;

RAM: mínimo 8GB ou superior;

Armazenamento: mínimo 200GB, preferencialmente SSD;

Servidor Web: Nginx ou Apache;

Servidor de Aplicação: Flask (Python), SpringBoot (Java);

Banco de Dados: MYSQL ou PostgreSQL. Instância com pelo menos 2 núcleos e 4GB RAM dedicada. Configuração de Backup automáticos semanal;

Servidor de arquivos: configurado para armazenar arquivos estáticos e upload (imagens de produtos). Mínimo 50GB inicial;

6.1 Métricas

As métricas apresentadas seguem o método de Análise de Pontos de Função (APF ou FPA - *Function Point Analysis*), uma técnica consagrada na engenharia de software para medição do tamanho funcional de sistemas. Desenvolvida por Allan Albrecht na IBM em 1979, esta abordagem é reconhecida pela ISO (ISO/IEC 20926) e amplamente utilizada

Estas métricas constituem um modelo quantitativo robusto quando adequadamente calibrado, fornecendo insumos valiosos para o planejamento e

controle do projeto de software. Esta metodologia usada, embora apresente pequenas inconsistências nos cálculos finais, apresenta maturidade na aplicação de técnicas de engenharia de software e estimativas realistas.

Figura 27 – Métricas 1

Função	Nº de ocorrência	Complexidade	Peso	Resultado	Nível de Influência do Sistema (0 a 5)	
entradas	0	Simple	3	0	Comunicação de dados	5
	7	Médio	4	28	Performance	5
	0	complexo	6	0	Volume de transações	5
					Eficiência do usuário final	5
saídas	1	Simple	4	4	Processamento complexo	5
	7	Médio	5	35	Facilidade de implantação	5
	0	complexo	7	0	Múltiplos locais	5
					Processamento distribuído	5
consultas	0	Simple	3	0	Utilização de equipamento	5
	6	Médio	4	32	Entrada de dados on-line	5
	0	complexo	6	0	Atualização on-line	5
					Reutilização de código	5
arquivos	0	Simple	7	0	Facilidade operacional	5
	7	Médio	10	70	Facilidade de mudanças	5
	0	complexo	15	0		
					Total de NI	70
interfaces	1	Simple	5	5		
	7	Médio	7	49		
	0	complexo	10	0		
Total de FP'b				223	FA = multiplicar o NI pela taxa real =0,65+(0,01*119)	1,35

Fonte: Elaborado pelos autores.

Figura 28 – Métricas 2

		FP'r = multiplicar o FP'b pelo FA	301
		Informe o nº de LOC da Linguagem	17,5
Estimativas do número médio de LOC por FP		KLOC = Multiplicar o FP'r pelo tipo de linguagem	5268,375
	Cobol	100	Total de KLOC
	Pascal	90	5268
Linguagens Orientadas a Objeto (C++)		30	Informe o tipo de sistema
	Java / Delphi / Visual Basic / C#	20	3.300
Geradores de Código (SQL + HTML + RUBY + PYTHON + DEMAIS)		15	PRAZO (dividir o KLOC pelo tipo de sistema)
		(RESULTADO DA DIVISÃO)QTDD DE MESES	1,60
Tipo de Sistema	Produ - Kloc/Loc /mês		QTDD DE DIAS (% x 22)
Sistema Comercial	2.500		13,20
Comércio Eletrônico	3.600		QTDD DE HORAS (% x 6)
Sistema Web	3.300		1,20
			45,6
CUSTO - INVESTIMENTO			
PRAZO			Informe o valor da hora de trabalho
DIAS/MÊS	22		R\$ 25,00
HORAS/DIA	6		ISO (NORMA INTERNACIONAL)
MINUTOS/HORA	60		HORAS/MÊS =
			132
			FÓRMULA = 132 * RESULTADO DA DIVISÃO * VALOR DA HORA
CONFIRMAÇÃO (RESULTADO * 132 * Vr DA HORA)		VALOR TOTAL DO PROJETO =	R\$ 5.268,38

Fonte: Elaborado pelos autores.

6.2 Proposta comercial

A implementação do aplicativo de compras para supermercados se baseia na ideia de simplificar o processo de ida até o estabelecimento, de forma prática, conveniente e confiável. Além de atender às demandas do cliente final, também poderá ampliar o alcance de vendas e lucro para os prestadores de serviços e podendo receber uma futura implementação para outros ramos varejistas.

O principal objetivo do projeto está em automatizar a lista de pedido do cliente, que foi definido após fechamento de escopo. A proposta consiste em permitir a personalização da lista de compra, para uma experiência mais prática ao usuário, implementando funcionalidades para adicionar 'x' unidades, remover itens, salvar lista, criar mais de uma lista, e renomear a lista.

Através da elicitação de requisitos por um formulário e da análise da matriz SWOT, foram identificados pontos fracos significativos que afetam a eficiência e a experiência dos usuários em aplicativos de supermercados, destacando-se o sistema de entregas, a forma/plataforma de pagamento, o aprendizado do usuário e a automatização da lista de compras. Esses pontos representam oportunidades para melhorias substanciais por meio da implementação de um aplicativo de compras para supermercados, trazendo assim maior praticidade e conveniência para os clientes e mais vendas para os mercados.

O sistema foi projetado para atender a dois perfis de usuários principais: os clientes e os funcionários do supermercado. Ambos precisam realizar um cadastro inicial e fazer login para acessar as funcionalidades da plataforma, garantindo segurança e personalização.

Para os clientes, o sistema oferece ferramentas práticas que facilitam suas compras. Eles podem acompanhar o status de seus pedidos em tempo real, criar e editar listas de compras personalizadas, salvar suas listas mais usadas e remover aquelas que não são mais necessárias. Essas funcionalidades foram pensadas para tornar a experiência de compra mais ágil e conveniente.

Do lado dos funcionários do supermercado, a plataforma disponibiliza recursos importantes para o gerenciamento das vendas. Eles podem atualizar o estoque de produtos disponíveis, visualizar todos os pedidos recebidos, confirmar esses pedidos e manter os clientes informados sobre cada etapa do processo, desde a confirmação até a entrega.

Essa estrutura dupla permite que o sistema atenda às necessidades tanto dos clientes, que buscam praticidade e comodidade, quanto dos supermercados, que precisam de eficiência no gerenciamento de suas vendas online. As duas partes funcionam de maneira integrada, criando um fluxo de trabalho completo que beneficia todos os envolvidos no processo de compra e venda.

Desenvolver um sistema web para gerenciamento de vendas e operações em supermercados, que inclui funcionalidades de gerenciamento de listas de compras para clientes e cadastro de produtos para funcionários. A Estrutura Analítica do Projeto para a aplicação, proporciona uma visão clara e organizada das atividades e entregáveis do projeto, facilitando o planejamento, execução, monitoramento e finalização. Com essa estrutura, o projeto é decomposto em partes gerenciáveis, permitindo uma melhor alocação de recursos e um controle mais eficaz do progresso e da qualidade do trabalho.

Principais entregáveis:

- Interface do Usuário (UI): Interface web responsiva para desktop.
- Módulo de Autenticação: Registro, login e gerenciamento de sessões.
- Módulo de Gestão de Produtos: Cadastro, edição e exclusão de produtos.
- Módulo de Lista de Compras: Criação, visualização e edição de listas de compras pelos clientes.

- Banco de dados: Estrutura de banco de dados para armazenar informações de produtos, usuários, pedidos e listas de compras.
- Segurança: Implementação de HTTPS, controle de acesso baseado em funções de proteção contra vulnerabilidades comuns.
- Documentação: Documentação técnica e de usuário.

Seguindo o plano de desenvolvimento estabelecido pelo cálculo das métricas, obtivemos um prazo de 36 dias úteis, para entrega da aplicação sendo escrita na linguagem Java Script. Este período estipulado não leva em consideração o uso de tecnologias containerização, padrão de arquitetura como micro serviços, dentre outros conceitos que levariam a aplicação se tornar mais escalável e modular, facilitando a manutenção a longo prazo.

O investimento necessário para adquirir nossa aplicação está detalhado abaixo:

- Oferta para os 10 primeiros clientes:
- Licença de uso vitalícia: R\$500,00 (pagamento único);
- Mensalidades: Primeira mensalidade somente após 3 meses de uso, por R\$ 50,00 (para manutenção, suporte e pequenas alterações);

Os gastos com hospedagem e infraestrutura não estão inclusos no cálculo acima. Entretanto, exploramos as configurações mínimas para o sistema, na documentação de portabilidade.

Considerações finais

As considerações finais do projeto "Vitrine Virtual" consolidam o desenvolvimento de uma aplicação web destinada à supermercados, atuando como uma vitrine digital para otimizar as compras online e o gerenciamento de listas personalizadas. Este sistema foi concebido a partir de uma análise detalhada das deficiências em aplicativos de compras online existentes, especialmente na cidade de Franca/SP, onde foram identificadas falhas como catálogos incompletos, divergências de preços, falta de funcionalidades intuitivas e problemas de comunicação, entrega e processamento de pagamentos, além da crucial ausência de funcionalidade para salvar listas de compras. Pesquisas com *stakeholders* e uma análise SWOT confirmaram essas vulnerabilidades, destacando a logística de entregas, opções de

pagamento, curva de aprendizado do usuário e, primordialmente, a carência de automação para listas de compras.

A "Vitrine Virtual" foi projetada como uma solução abrangente para superar esses desafios, com o objetivo principal de simplificar a experiência do usuário e ampliar as vendas para os estabelecimentos. O foco central do projeto foi a automatização e personalização das listas de compras, permitindo aos clientes criar, editar, salvar e gerenciar múltiplas listas de forma prática. O desenvolvimento seguiu uma estrutura organizada em cinco etapas – iniciação, planejamento, execução, monitoramento e finalização – utilizando metodologias ágeis e técnicas de engenharia de software. Artefatos como diagramas BPMN e UML, além de protótipos de telas, foram essenciais para guiar a produção, e restrições como tempo e capacitação da equipe foram gerenciadas com ferramentas como Kanban, Figma e StarUML.

O sistema foi concebido para atender a dois perfis de usuários: clientes do supermercado e funcionários do supermercado, cada um com acesso personalizado após cadastro e login. Para os clientes, a aplicação permite a navegação por produtos, visualização de detalhes, adição ao carrinho, um processo de *checkout* simples e seguro, escolha entre recebimento em casa ou retirada na loja, e acompanhamento em tempo real do status dos pedidos. Eles podem ainda gerenciar suas informações pessoais, endereços de entrega, métodos de pagamento e histórico de compras. Para os funcionários, a plataforma oferece recursos para atualizar o catálogo de produtos na base de dados e gerenciar os pedidos, incluindo listagem, aceitação e notificação dos clientes sobre o andamento e a separação dos itens para entrega. Essa dualidade funcional assegura que o sistema beneficie ambos os lados, otimizando as operações e aprimorando a satisfação do consumidor.

A primeira versão do sistema, com seu escopo focado na simplicidade e na gestão de listas de compras, demonstra a maturidade na aplicação de técnicas de engenharia de software e estimativas realistas para o desenvolvimento. No entanto, o projeto possui um grande potencial para expansão e aprimoramento, com propostas claras para futuras iterações.

Ideias para Projetos Futuros incluem a expansão para plataformas móveis (mobile), o que já é previsto e ampliaria significativamente o alcance da aplicação. Para garantir escalabilidade e modularidade a longo prazo, a modelagem de uma arquitetura de microsserviços é uma proposta de expansão essencial, facilitando a manutenção e a evolução de funcionalidades independentes. Embora a logística de

entregas e as opções de pagamento tenham sido identificadas como fraquezas, projetos futuros poderiam aprofundar o desenvolvimento de módulos robustos para otimização de rotas e integração com múltiplos *gateways* de pagamento. A otimização da curva de aprendizado do usuário poderia ser abordada com tutoriais interativos e interfaces mais adaptativas. Além disso, a flexibilidade do conceito da "Vitrine Virtual" permite sua adaptação para outros setores de vendas e varejo, além de supermercados. Funcionalidades avançadas para clientes, como recomendações de produtos personalizadas e integração com programas de fidelidade, bem como a gestão de estoque e integração com sistemas ERP dos supermercados, representam um vasto horizonte de crescimento e inovação para o futuro do projeto.

Referências

- AHMAD S AL-SHAMSI, Mohammed; *PROJECT MANAGEMENT INSTITUTE. Project Management Body of Knowledge*. 1. ed. [S. l.: s. n.], 1996. 176 p.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. *Unified Modeling Language User Guide*. 2. ed. [S. l.]: Addison-Wesley, 2005. 475 p.
- CHAFFEY, Dave; HEMPHILL, Tanya; EDMUNDSON-BIRD, David. *Digital Business and E-commerce Management*. 1. ed. [S. l.]: Pearson Education, Limited, 2019. 680 p.
- COCKBURN, Alistair. *Writing Effective Use Cases*. 1. ed. [S. l.]: Pearson Education, 2000. 304 p.
- DAVIS, F. D. *Perceived usefulness, perceived ease of use, and user acceptance of information technology*. *MIS Quarterly*, 1989.
- FOWLER, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. [S. l.]: Addison-Wesley, 2004. 175 p.
- GUEDES, Gilleanes. *UML 2 - Uma Abordagem Prática*. [S. l.]: Novatec Editora, 2018. 496 p.
- HELDMAN, Kim. *Project Management JumpStart*. 1. ed. [S. l.]: Wiley, 2006. 356 p.
- KERZNER, Harold. *Gerenciamento de projetos*. 2. ed. [S. l.]: Blucher, 2021. 782 p.
- KOTLER, Philip. *Marketing 5.0*. [S. l.]: Sextante, 2021. 256 p.
- LAUDON, Kenneth; TRAVER, Carol. *E-Commerce 2020-2021: Business, Technology and Society, Global Edition*. [S. l.]: Pearson Education, 2020. 913 p.

LARMAN, Craig. ***Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and the Unified Process***. [S. l.]: Prentice Hall PTR, 2002. 627 p.

NIELSEN, Jakob. **Projetando websites**. 1. ed. [S. l.]: ELSEVIER EDITORA, 2000. 416 p.

NORMAN, Donald. **O design do dia a dia**. [S. l.]: Rocco, 2006. 272 p.

PORTER, Michael. **Competição: estratégias competitivas essenciais**. [S. l.]: CAMPUS, 1999. 515 p.

PRESSMAN, Roger. **Engenharia de software**. 9. ed. [S. l.: s. n.], 2021. 704 p.

ROGERS, Everett. ***Diffusion of Innovations***. 5. ed. [S. l.]: Free Press, 2003. 576 p.

WIEGERS, Karl; BEATTY, Joy. ***Software Requirements***. 1. ed. [S. l.]: Pearson Education, 2013. 672 p.

Anexo 1

Link para acessar a imagem do BPMN:
(https://drive.google.com/file/d/1gpl55KgZp_tbQ9wft0ybgeUfOeeS-IEN/view?usp=drive_link).