

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

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

VICTOR VIEIRA SILVA

CATÁLOGO DIGITAL PARA GESTÃO DE PRODUTOS

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. Dr. Daniel Facciolo Pires

**FRANCA/SP
2025**

CATÁLOGO DIGITAL PARA GESTÃO DE PRODUTOS

Victor Vieira Silva¹
Daniel Facciolo Pires²

Resumo

Este trabalho tem como objetivo o desenvolvimento de um sistema *web* de catálogo digital para a empresa Vital Cosméticos, visando solucionar problemas relacionados à dificuldade na atualização de informações e à dependência de materiais impressos e serviços de terceiros. A motivação para a realização deste projeto surgiu da necessidade de propor uma solução tecnológica que proporcionasse à empresa maior autonomia na gestão de seu portfólio de produtos e melhoria na comunicação com os clientes. A metodologia aplicada envolveu levantamento de requisitos, modelagem de processos, definição da arquitetura do sistema e desenvolvimento *full stack* utilizando tecnologias *web*. O sistema desenvolvido permite o cadastro, edição e exclusão de produtos, categorias e *folders* digitais, além de oferecer uma interface para clientes consultarem produtos, salvarem favoritos e solicitarem orçamentos. Como resultado, o projeto atendeu aos requisitos estabelecidos, entregando uma solução funcional que contribui para a otimização dos processos internos da empresa, redução de custos operacionais e fortalecimento da presença digital.

Palavras-chave: Catálogo Digital. Desenvolvimento Web. Experiência do Cliente. Gestão de Produtos. Otimização de Processos. Solução Tecnológica.

Abstract

This paper aims to develop a web-based digital catalog system for the company, Vital Cosméticos, addressing issues related to the difficulty of updating product information and the dependence on printed materials and third-party services. The motivation for this project arose from the need to propose a technological solution that would provide the company with greater autonomy in managing its product portfolio and improving communication with customers. The applied methodology involved requirements gathering, process modeling, system architecture definition, and full-stack development using web technologies. The developed system allows the registration, editing, and deletion of products, categories, and digital folders, in addition to offering an interface where customers can browse products, save favorites, and request quotes. As a result, the project met the established requirements, delivering a functional solution that contributes to optimizing the company's internal processes, reducing operational costs, and strengthening its digital presence.

Keywords: Customer Experience. Digital Catalog. Process Optimization. Product Management. Solution Development. Web Development.

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

² Docente em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: daniel.pires5@fatec.sp.gov.br.

1 Introdução

Atualmente, a transformação digital é uma necessidade para empresas que buscam se manter competitivas no mercado, especialmente no setor de cosméticos, onde a agilidade na atualização de informações e a qualidade no atendimento ao cliente são diferenciais estratégicos. Nesse contexto, a empresa Vital Cosméticos enfrenta desafios relacionados à gestão manual de seu portfólio de produtos, à dependência de materiais impressos e de serviços de terceiros para atualização de catálogos, o que gera custos operacionais, retrabalho e dificuldades na comunicação eficiente com os clientes.

Diante desse cenário, surge a seguinte questão-problema:

“Como desenvolver uma solução tecnológica que permita à Vital Cosméticos gerenciar seu catálogo de produtos de forma autônoma, ágil e eficiente, proporcionando também uma melhor experiência para seus clientes?”

A partir dessa problemática, o presente trabalho tem como objetivo geral desenvolver um sistema *web* de catálogo digital para a Vital Cosméticos, que permita a gestão eficiente dos produtos, a geração de *folders* digitais personalizados e a disponibilização de informações atualizadas para clientes e parceiros. Como objetivos específicos, destacam-se:

- Permitir o cadastro, edição e exclusão de produtos, categorias e *folders* digitais;
- Proporcionar uma interface intuitiva para que os clientes possam consultar produtos, adicionar aos favoritos e solicitar orçamentos;
- Reduzir a dependência da empresa em relação a materiais impressos e serviços externos;
- Melhorar a comunicação e a experiência do cliente por meio de uma plataforma responsiva e atualizada em tempo real.

A justificativa para a realização deste projeto está fundamentada na necessidade de modernizar os processos internos da empresa, reduzir custos operacionais, otimizar o tempo dos colaboradores e melhorar a experiência dos clientes. Além disso, a adoção de uma solução digital contribui para a sustentabilidade, ao reduzir o uso de papel, e fortalece a presença digital da Vital

Cosméticos no mercado, tornando-a mais competitiva e alinhada às tendências tecnológicas.

Para alcançar os objetivos propostos, este projeto adotou uma metodologia prática e iterativa, que incluiu levantamento de requisitos, modelagem de processos e dados, definição da arquitetura do sistema e desenvolvimento *full stack* utilizando tecnologias *web* modernas. O *frontend* foi desenvolvido como uma aplicação do tipo *Single Page Application* (SPA), utilizando as bibliotecas React e TypeScript, com construção de interface baseada no framework de estilização Tailwind CSS e utilizando Vite como ferramenta de *build* e desenvolvimento. Para o *backend*, foi utilizada a plataforma Supabase, que fornece uma solução *Backend as a Service* (BaaS) integrada com banco de dados relacional PostgreSQL, autenticação, controle de acesso e APIs em tempo real.

Este trabalho está organizado em seis capítulos, cujos conteúdos se resumem a seguir:

- **Capítulo 1 – Introdução**

Apresenta o contexto, a questão-problema, os objetivos, a justificativa e a metodologia adotada.

- **Capítulo 2 – Viabilidade do Projeto**

Analisa a viabilidade técnica, econômica e operacional por meio do Business Model Canvas, da matriz SWOT e de outros estudos de suporte.

- **Capítulo 3 – Levantamento de Requisitos**

Detalha as técnicas de elicitação (entrevistas, questionários, visitas), os requisitos funcionais e não funcionais, as regras de negócio, os diagramas BPMN, casos de uso, diagramas de atividades e a matriz de rastreabilidade.

- **Capítulo 4 – Ferramentas e Métodos**

Descreve as ferramentas de modelagem e prototipação, bem como as tecnologias escolhidas (React, TypeScript, Vite, Tailwind CSS, Supabase /PostgreSQL) e o processo de desenvolvimento iterativo adotado.

- **Capítulo 5 – Desenvolvimento do protótipo de *software***

Expõe as decisões de arquitetura e as etapas de implementação do sistema,

incluindo *backend* (Supabase) e *frontend* (React), gerenciamento de estado, rotas protegidas e integrações.

- **Capítulo 6 – Resultados e Discussão**

Apresenta as telas desenvolvidas (login, catálogo, dashboard, formulários, chat flutuante), métricas observadas e análise crítica em relação aos objetivos propostos.

1.1 Termo de Abertura do Projeto (TAP)

O Termo de Abertura do Projeto (TAP) é um documento fundamental no gerenciamento de projetos, utilizado para formalizar seu início e estabelecer uma visão clara e compartilhada entre todas as partes interessadas. De acordo com o *PMBOK® Guide (PMI, 2017)*, o TAP é desenvolvido na fase de iniciação e serve como um contrato formal que autoriza o projeto, descrevendo informações essenciais como o objetivo, o escopo, os prazos, os recursos necessários, as restrições, os riscos e os responsáveis. Além disso, o TAP funciona como referência ao longo de todo o ciclo de vida do projeto, sendo essencial para garantir o alinhamento entre as expectativas dos *stakeholders* e a entrega dos resultados planejados.

A seguir, apresenta-se o TAP do projeto de desenvolvimento do Catálogo Digital da Vital Cosméticos.

Objetivos do Projeto:

- Desenvolver um catálogo digital de produtos para a distribuidora Vital Cosméticos, com uma interface intuitiva, responsiva e de fácil navegação, que atenda às necessidades dos clientes e otimize o acesso às informações dos produtos.
- Automatizar a gestão e atualização dos produtos, permitindo o cadastro, edição e exclusão de informações, incluindo imagens, descrições detalhadas e códigos de referência de maneira ágil e organizada.
- Melhorar a experiência do cliente por meio de funcionalidades que permitam a visualização dos produtos, favoritos e geração de folders digitais personalizados, facilitando a comunicação entre a empresa e seus clientes.

- Disponibilizar relatórios simples, como quantidade de visualizações e favoritos, que possam auxiliar a empresa na análise de interesse dos clientes sobre os produtos.
- Reduzir a dependência de materiais impressos e de serviços terceirizados, tornando o processo mais eficiente, sustentável e economicamente viável.

Escopo do Projeto:

- Análise e levantamento de requisitos: identificação das necessidades específicas da Vital Cosméticos e definição dos requisitos funcionais e não funcionais do sistema.
- Projeto e desenvolvimento do sistema: elaboração da arquitetura, desenvolvimento dos componentes principais e implementação das funcionalidades, incluindo o cadastro de produtos, categorias, favoritos e geração de folders digitais.
- Testes e validação: realização de testes de funcionamento, usabilidade e responsividade, com validação por parte dos usuários-chave da empresa.
- Implantação e treinamento: publicação do sistema, configuração inicial e treinamento da equipe da Vital Cosméticos para utilização e administração da plataforma.
- Acompanhamento pós-implantação: monitoramento contínuo do desempenho do sistema, com possíveis ajustes, correções e melhorias conforme necessidades identificadas.

Restrições e Dependências:

- Orçamento previamente aprovado para o desenvolvimento do projeto.
- Disponibilidade dos recursos necessários, incluindo equipe de desenvolvimento, ferramentas de *software*, equipamentos e infraestrutura.
- Participação ativa da equipe da Vital Cosméticos para definição, validação de requisitos e realização de testes.
- Cumprimento dos prazos estabelecidos no cronograma do projeto.

Cronograma Preliminar:

O projeto será desenvolvido em cinco etapas, com duração total estimada de seis meses, distribuídas da seguinte forma:

- Análise e levantamento de requisitos: 1 mês;
- Projeto e desenvolvimento: 2 meses;
- Testes e validação: 1 mês;
- Implantação e treinamento: 1 mês;
- Acompanhamento pós-implantação: 1 mês.

Equipe do Projeto:

- Gerente do Projeto: responsável pela coordenação geral, gestão dos recursos e acompanhamento do progresso.
- Analistas de Negócio: responsáveis pela análise dos requisitos e levantamento das necessidades da empresa.
- Equipe de Desenvolvimento: responsável pela construção da interface, lógica de funcionamento e integração dos dados, utilizando tecnologias como React, TypeScript, Vite, Tailwind CSS e Supabase.
- Especialistas em Implantação: responsáveis pela configuração, publicação do sistema e treinamento dos usuários.
- Equipe de Testes: responsável pela garantia da qualidade, realizando os testes funcionais, de usabilidade e responsividade.

Riscos:

- Possíveis atrasos devido a desafios técnicos durante o desenvolvimento da plataforma.
- Eventual necessidade de ajustes para adaptar o sistema às preferências específicas dos usuários.
- Dependência da disponibilidade da equipe da Vital Cosméticos para validação de requisitos, testes e treinamentos.

Aprovação:

- Este Termo de Abertura do Projeto foi revisado e aprovado por todas as partes envolvidas, formalizando assim o início do desenvolvimento do Catálogo Digital da Vital Cosméticos.

2 Viabilidade do Projeto

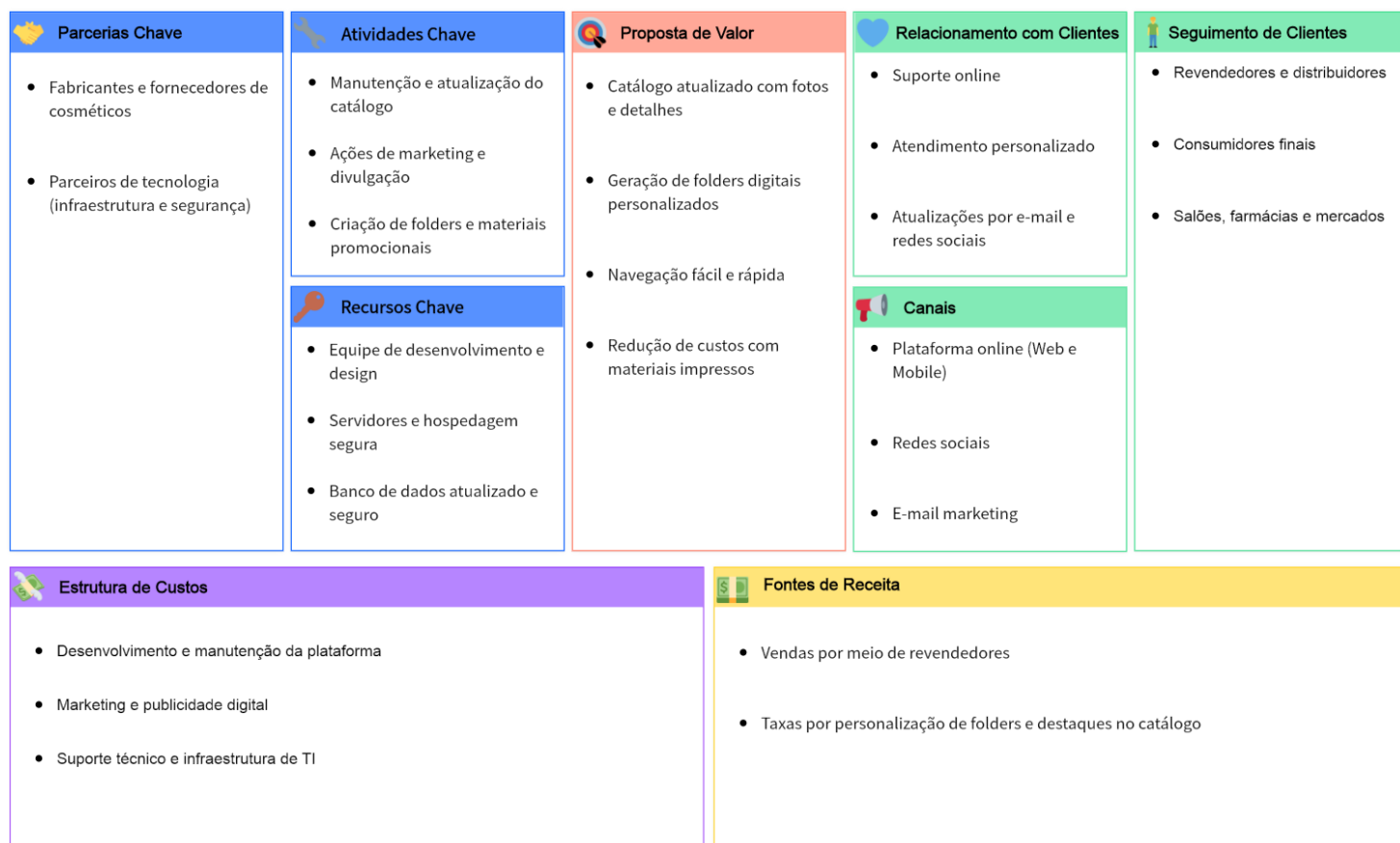
A viabilidade do projeto foi avaliada utilizando o *Business Model Canvas* – BMC, uma ferramenta estratégica de gerenciamento que permite descrever, visualizar, avaliar e alterar modelos de negócio.

Segundo Osterwalder e Pigneur (2010), criadores do BMC, essa ferramenta auxilia no entendimento de como uma organização cria, entrega e captura valor, permitindo uma visão ampla e estruturada dos principais elementos que compõem o negócio.

De acordo com Analista Modelos de Negócios (2022), o BMC é composto por nove blocos fundamentais, que estão distribuídos em quatro áreas principais: clientes, oferta, infraestrutura e viabilidade financeira. Essa abordagem possibilita uma comunicação clara entre todos os envolvidos no projeto e permite identificar melhorias e alinhar estratégias.

Neste projeto, o *Business Model Canvas* foi aplicado para representar o modelo de negócio do catálogo digital de produtos da empresa Vital Cosméticos, conforme demonstrado na Figura 1.

Figura 1 – Business Model Canvas (BMC)



Fonte: dados da pesquisa

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

A seguir, são apresentados os detalhes de cada um dos nove blocos que compõem o BMC do projeto:

2.1.1 Segmento de Clientes

- Revendedores e distribuidores;
- Consumidores finais;
- Salões, farmácias e mercados.

2.1.2 Proposta de Valor

- Catálogo atualizado com fotos e detalhes;
- Geração de folders digitais personalizados;
- Navegação fácil e rápida;

- Redução de custos com materiais impressos.

2.1.3 Canais

- Plataforma online *web* e *mobile*;
- Redes sociais;
- E-mail marketing.

2.1.4 Relacionamento com Clientes

- Suporte online;
- Atendimento personalizado;
- Atualizações por e-mail e redes sociais.

2.1.5 Fontes de Receita

- Vendas por meio de revendedores;
- Taxas por personalização de folders e destaques no catálogo.

2.1.6 Recursos-Chave

- Equipe de desenvolvimento e design;
- Servidores e hospedagem segura;
- Banco de dados atualizado e seguro.

2.1.7 Atividades-Chave

- Manutenção e atualização do catálogo;
- Ações de marketing e divulgação;
- Criação de folders e materiais promocionais.

2.1.8 Parcerias-Chave

- Fabricantes e fornecedores de cosméticos;
- Parceiros de tecnologia (infraestrutura e segurança).

2.1.9 Estrutura de Custos

- Desenvolvimento e manutenção da plataforma;
- Marketing e publicidade digital;

- Suporte técnico e infraestrutura de TI.

Dessa forma, o modelo de negócio proposto demonstra ser viável, atendendo às necessidades da Vital Cosméticos e promovendo eficiência na gestão do portfólio, redução de custos, melhoria na comunicação e fortalecimento da presença digital da empresa no mercado.

2.2 Matriz SWOT

A análise SWOT é uma ferramenta de planejamento estratégico utilizada para avaliar os ambientes interno e externo de uma organização. O termo é um acrônimo em inglês que representa quatro elementos principais: *strengths* (forças), *weaknesses* (fraquezas), *opportunities* (oportunidades) e *threats* (ameaças).

Segundo Porter (1986), a análise dos ambientes interno e externo é essencial para que as empresas desenvolvam estratégias competitivas, identifiquem seus diferenciais e fortaleçam sua posição no mercado frente às ameaças e oportunidades. Essa ferramenta possibilita uma compreensão clara dos fatores que impactam diretamente o desempenho organizacional, permitindo que as empresas atuem de forma proativa e estratégica.

No desenvolvimento deste projeto, a matriz SWOT foi fundamental para compreender os desafios enfrentados pela Vital Cosméticos, além de evidenciar seus pontos fortes e as oportunidades de crescimento.

Na Figura 2, é apresentada a matriz SWOT elaborada para a empresa Vital Cosméticos, considerando seu contexto atual.

Figura 2: Matriz SWOT



Fonte: o autor

Por meio da análise SWOT, foram identificadas algumas fraquezas que merecem atenção. A principal delas refere-se à dificuldade na gestão e atualização do catálogo de produtos, que atualmente depende de processos externos e manuais, resultando em lentidão e custos adicionais.

Diante desse cenário, a proposta deste projeto visa à implementação de um catálogo digital, que permitirá uma gestão mais eficiente dos produtos, atualização rápida das informações, redução de custos operacionais e, principalmente, o fortalecimento da presença digital da Vital Cosméticos. Essas ações são fundamentais para melhorar a comunicação com clientes, atrair novos mercados e garantir o crescimento sustentável da empresa.

3 Levantamento de Requisitos

O levantamento de requisitos é uma etapa fundamental no desenvolvimento de sistemas, pois tem como objetivo identificar, compreender e documentar as

necessidades dos usuários e *stakeholders*, bem como as funcionalidades e restrições do sistema a ser desenvolvido. Essa fase é essencial para garantir que o sistema final atenda de forma eficaz às expectativas, otimize os processos e resolva os problemas identificados no contexto da organização.

No desenvolvimento do sistema de catálogo digital para a Vital Cosméticos, foi realizado um processo de elicitação de requisitos com o intuito de entender profundamente as dificuldades enfrentadas pela empresa na gestão manual de produtos e na comunicação com clientes. Essa etapa permitiu transformar essas necessidades em especificações claras e objetivas que orientaram o desenvolvimento do sistema.

Na sequência, são apresentadas as técnicas utilizadas para a elicitação dos requisitos, sua importância no contexto da engenharia de *software* e a descrição detalhada das informações coletadas.

3.1 Elicitação e especificação dos Requisitos

A elicitação de requisitos é um processo essencial na engenharia de *software*, pois é a partir dela que se obtêm informações precisas sobre as necessidades dos usuários e os objetivos do projeto.

Para este projeto, foram aplicadas as seguintes técnicas de elicitação:

- **Entrevistas:** foi realizada uma entrevista com o proprietário da empresa Vital Cosméticos, buscando entender os principais desafios enfrentados na gestão do catálogo de produtos, além de compreender suas expectativas em relação ao sistema.
- **Questionários:** foi elaborado um questionário direcionado a alguns clientes e representantes comerciais, com o objetivo de entender suas necessidades quanto ao acesso às informações dos produtos, busca por novidades e facilidade na comunicação com a empresa.
- **Visitas às Instalações:** foram realizadas visitas à empresa para observar os processos internos relacionados ao controle de produtos, atendimento aos clientes e métodos utilizados atualmente para compartilhar informações. Isso proporcionou uma compreensão mais ampla e precisa do ambiente e das necessidades reais da empresa.

A partir dessas atividades, foi possível especificar um conjunto de requisitos que orientaram o desenvolvimento do sistema, garantindo que ele atendesse de forma eficaz às demandas tanto da empresa quanto dos clientes e parceiros.

As perguntas e respostas coletadas durante a entrevista estão apresentadas no Apêndice 2 no final deste documento.

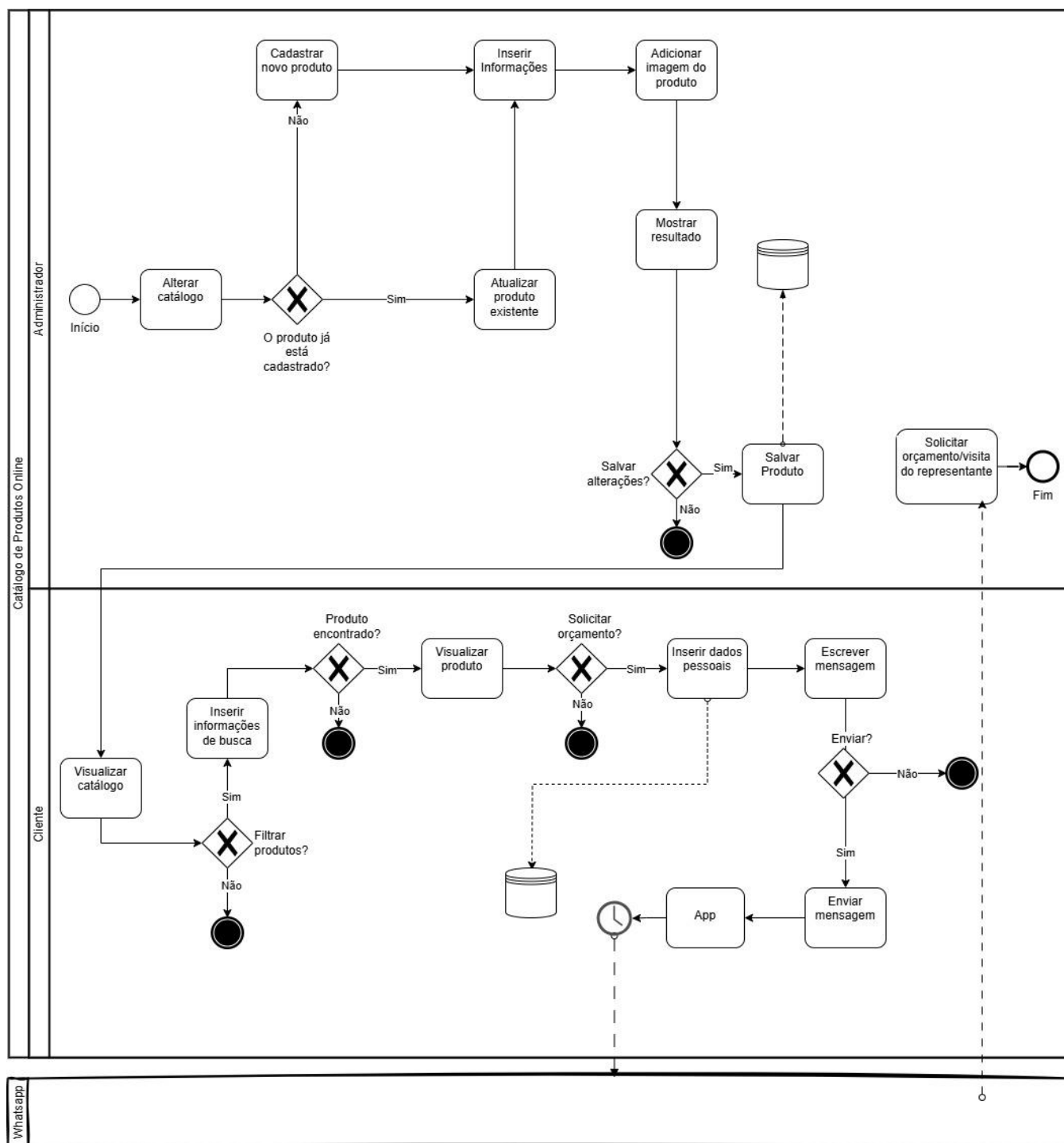
3.2 *Business Process Model and Notation (BPMN)*

O *Business Process Model and Notation (BPMN)* é uma notação gráfica padrão que permite a representação de processos de negócio de maneira clara e consistente. Segundo Pressman (2016), o BPMN tem como objetivo criar uma ponte entre o *design* do processo e sua implementação, sendo compreensível tanto para analistas de negócio quanto para desenvolvedores técnicos. Por meio desta notação, é possível documentar os processos de maneira padronizada, facilitando a comunicação entre diferentes áreas da organização, bem como a análise, entendimento e melhoria dos fluxos operacionais.

Essa notação contribui diretamente para que as empresas possam alinhar seus processos às suas estratégias, tornando-os mais eficientes, transparentes e eficazes. Além disso, auxilia na identificação de gargalos e oportunidades de otimização, garantindo maior controle sobre as operações.

No contexto do sistema para a Vital Cosméticos, o modelo de processo foi elaborado com base nas informações obtidas durante a elicitação de requisitos por meio de entrevistas, questionários e visitas presenciais. Ele representa o fluxo das atividades relacionadas ao gerenciamento dos produtos no catálogo digital, desde o cadastro, atualização de informações e geração de *folders* personalizados, até o acesso dos clientes às informações e a solicitação de orçamentos. O diagrama BPMN desenvolvido para o projeto é apresentado na Figura 3.

Figura 3: Business Process Model and Notation (BPMN)



Fonte: o autor

3.3 Requisitos Funcionais

Na Engenharia de Software, os Requisitos Funcionais correspondem às descrições das funcionalidades, serviços e comportamentos que o sistema deve executar. Segundo Sommerville (2011), esses requisitos especificam de forma clara

quais ações e operações o software deve ser capaz de realizar para atender às necessidades dos usuários e dos processos de negócio.

Esses requisitos definem como o sistema deve se comportar perante determinadas entradas, interações e demandas dos usuários, detalhando os recursos oferecidos. De maneira prática, os requisitos funcionais respondem o que o sistema deve fazer, servindo como base para as etapas de projeto, desenvolvimento e validação do *software*.

O Quadro 1 representa os principais requisitos funcionais deste sistema.

Quadro 1: Requisitos Funcionais do Sistema

ID: RF001	Nome do Requisito: Cadastrar novo produto
Descrição →	O sistema permitirá que o administrador cadastre um novo produto no catálogo, incluindo nome, descrição, categoria, código, referência, imagens e status de estoque.
Categoria: Evidente	Prioridades: Essencial
Informações →	Nome, descrição, categoria, código, referência, imagens, status do estoque.
Regra de Negócio →	Política de produtos e estoque.
ID: RF002	Nome do Requisito: Atualizar produto
Descrição →	O sistema permitirá que os administradores atualizem informações de um produto já cadastrado, incluindo alteração de dados e status de estoque.
Categoria: Evidente	Prioridade: Importante
Informações →	Nome, descrição, categoria, código, referência, imagens, status do estoque.
Regra de Negócio →	Atualização regular de conteúdo.
ID: RF003	Nome do Requisito: Filtrar produtos
Descrição →	O sistema permitirá que os usuários filtrem os produtos disponíveis no catálogo com base em critérios como categoria e disponibilidade de estoque.
Categoria: Evidente	Prioridade: Importante
Informações →	Categoria, status de estoque.
Regra de Negócio →	Busca e filtragem de produtos.
ID: RF004	Nome do Requisito: Gerar folder digital
Descrição →	O sistema permitirá que os administradores gerem folders digitais personalizados com produtos

	selecionados, com opções de exportar em PDF e enviar por e-mail ou WhatsApp.
Categoria: Evidente	Prioridades: Essencial
Informações →	Produtos selecionados, dados do cliente, formato de exportação.
Regra de Negócio →	Utilização de <i>folders</i> digitais.
ID: RF005	Nome do Requisito: Sistema de Favoritos
Descrição →	O sistema permitirá que os clientes adicionem produtos aos favoritos e visualizem uma lista personalizada dos produtos de seu interesse.
Categoria: Evidente	Prioridades: Importante
Informações →	Produtos favoritos, identificação do usuário.
Regra de Negócio →	Gestão de favoritos.
ID: RF006	Nome do Requisito: Visualizar catálogo
Descrição →	O sistema permitirá que os usuários visualizem detalhes completos dos produtos disponíveis no catálogo, incluindo nome, descrição, imagens, código e referência.
Categoria: Evidente	Prioridades: Essencial
Informações →	Nome, código, descrição, imagens, categoria, status.
Regra de Negócio →	Atualização regular de conteúdo.
ID: RF007	Nome do Requisito: Sistema de autenticação
Descrição →	O sistema permitirá que usuários façam login e logout, diferenciando permissões entre clientes e administradores.
Categoria: Evidente	Prioridades: Essencial
Informações →	E-mail, senha, tipo de usuário.
Regra de Negócio →	Controle de acesso.
ID: RF008	Nome do Requisito: Gerenciar contatos
Descrição →	O sistema permitirá que administradores visualizem e gerenciem solicitações de contato dos clientes, incluindo status de atendimento.
Categoria: Evidente	Prioridades: Importante
Informações →	Nome, e-mail, telefone, mensagem, produto relacionado, status.
Regra de Negócio →	Atendimento ao cliente.
ID: RF009	Nome do Requisito: Dashboard administrativo

Descrição →	O sistema permitirá que administradores visualizem estatísticas do catálogo, incluindo total de produtos, visualizações, contatos e produtos mais acessados.
Categoria: Evidente	Prioridades: Importante
Informações →	Estatísticas de produtos, contatos, visualizações.
Regra de Negócio →	Controle de acesso.

Fonte: o autor

3.4 Requisitos Não Funcionais

Na Engenharia de Software, os Requisitos Não Funcionais são responsáveis por definir as características de qualidade que o sistema deve possuir, não estando diretamente ligados às funcionalidades específicas.

Diferente dos requisitos funcionais, que tratam sobre o que o sistema deve fazer, os requisitos não funcionais estão relacionados a como o sistema deve se comportar, estabelecendo critérios como desempenho, segurança, usabilidade, confiabilidade, disponibilidade e eficiência.

Esses requisitos são fundamentais para garantir uma boa experiência para os usuários, além de assegurar que o sistema opere de maneira estável, segura e com qualidade adequada às necessidades do negócio.

O Quadro 2 representa os principais requisitos não-funcionais deste sistema.

Quadro 2: Requisitos Não-Funcionais do Sistema

ID: RNF001	Nome do Requisito: Segurança dos dados
Descrição →	O sistema deverá garantir a segurança dos dados pessoais dos clientes e funcionários, incluindo dados de login, informações de contato e histórico de interações no catálogo.
Categoria: Segurança	Prioridades: Essencial
Informações →	N/A
Regra de Negócio →	Política de privacidade
ID: RNF002	Nome do Requisito: Responsividade
Descrição →	O site deverá ser responsivo e funcionar perfeitamente em diferentes dispositivos e navegadores, mantendo a mesma qualidade de experiência do usuário.

Categoria: Compatibilidade	Prioridades: Importante
Informações →	N/A
Regra de Negócio →	Responsividade do sistema
ID: RNF003	Nome do Requisito: Redirecionamento para o WhatsApp
Descrição →	O sistema deverá ser capaz de redirecionar os clientes para uma conversa no WhatsApp com o administrador para solicitar orçamentos e tirar dúvidas sobre produtos.
Categoria: Compatibilidade	Prioridades: Importante
Informações →	N/A
Regra de Negócio →	Atendimento ao cliente
ID: RNF004	Nome do Requisito: Performance do sistema
Descrição →	O sistema deverá ser capaz de carregar o catálogo e produtos em no máximo 3 segundos para garantir uma experiência de navegação fluida e evitar abandonos por lentidão.
Categoria: Desempenho	Prioridades: Essencial
Informações →	N/A
Regra de Negócio →	Desempenho do sistema
ID: RNF005	Nome do Requisito: Usabilidade
Descrição →	O sistema deverá ser intuitivo e de fácil navegação, permitindo que os administradores gerenciem produtos e os clientes naveguem pelo catálogo sem dificuldades.
Categoria: Usabilidade	Prioridades: Importante
Informações →	N/A
Regra de Negócio →	Busca e filtragem de produtos
ID: RNF006	Nome do Requisito: Backup regular
Descrição →	O sistema deverá realizar backups automáticos regulares dos dados de produtos e clientes para evitar perda de informações importantes.
Categoria: Segurança	Prioridades: Essencial
Informações →	N/A
Regra de Negócio →	Backup e recuperação

Fonte: o autor

3.5 Regras de Negócio

As regras de negócio representam um conjunto de diretrizes, critérios e restrições que definem como uma empresa conduz suas operações e toma decisões no dia a dia. Conforme definido por Navathe e Elmasri (2011), as regras de negócio são políticas, procedimentos ou princípios dentro de uma organização que são cruciais para a modelagem de dados, pois descrevem as restrições e a lógica que governam os processos de negócio. Elas são responsáveis por orientar os processos internos, padronizar procedimentos e garantir que as ações estejam alinhadas com os objetivos estratégicos da organização e com as necessidades dos clientes.

Essas regras documentam as políticas, normas, restrições e práticas específicas aplicáveis ao contexto do negócio, servindo como base para o funcionamento correto dos processos. Elas podem abranger aspectos como políticas de vendas, precificação, descontos, regras de cadastro, critérios de aprovação, validações de dados, cálculos operacionais e exigências legais ou de conformidade.

Na sequência, o Quadro 3 apresenta as regras de negócio definidas para o desenvolvimento do catálogo digital da empresa Vital Cosméticos, descrevendo suas principais diretrizes.

Quadro 3: Regras de Negócio

RN001 - Política de produtos e estoque
Descrição: O catálogo exibirá produtos com status "Disponível", "Esgotado" ou "Em breve", permitindo que os clientes vejam a situação atual do estoque e possam planejar suas compras.
RN002 - Política de privacidade
Descrição: O site incluirá uma política de privacidade clara, informando aos clientes como seus dados serão utilizados e protegidos pela Vital Cosméticos, em conformidade com a LGPD (Lei Geral de Proteção de Dados).
RN003 - Atualização regular de conteúdo
Descrição: O conteúdo do catálogo, incluindo informações sobre produtos, será atualizado regularmente pela equipe de marketing para garantir que as informações estejam sempre corretas e refletindo as características reais dos produtos.
RN004 - Utilização de <i>folders</i> digitais
Descrição: A Vital Cosméticos permitirá que os funcionários utilizem folders digitais como uma ferramenta de marketing para divulgar produtos em campanhas específicas, com geração automática de PDF e envio por e-mail ou WhatsApp.
RN005 – Busca e filtragem de produtos
Descrição: Os resultados da filtragem de produtos no catálogo deverão ser claros e organizados, permitindo que os clientes visualizem as informações dos produtos de forma conveniente através de

filtros por categoria e disponibilidade.
RN006 – Gestão de favoritos
Descrição: A empresa utilizará o sistema de favoritos para permitir que os clientes salvem produtos de interesse, facilitando futuras consultas e possibilitando análises de preferências para melhorar as campanhas de marketing.
RN007 – Controle de acesso
Descrição: O sistema diferenciara permissões entre clientes e administradores, garantindo que apenas usuários autorizados tenham acesso às funcionalidades administrativas de gerenciamento de produtos e contatos.
RN008 – Atendimento ao cliente
Descrição: Todas as solicitações de contato dos clientes serão registradas no sistema com status de acompanhamento, garantindo que nenhuma solicitação seja perdida e permitindo um atendimento organizado via WhatsApp.
RN009 – Responsividade do sistema
Descrição: O sistema deverá adaptar-se automaticamente a diferentes dispositivos e tamanhos de tela, garantindo experiência consistente em desktops, tablets e smartphones.
RN010 – Desempenho do sistema
Descrição: O sistema deverá manter tempos de carregamento otimizados para garantir uma experiência fluida, com carregamento de páginas em no máximo 3 segundos.
RN011 – Backup e recuperação
Descrição: O sistema realizará backups automáticos regulares de todos os dados importantes para garantir a continuidade do negócio e recuperação em caso de falhas.

Fonte: o autor

3.6 Casos de Uso

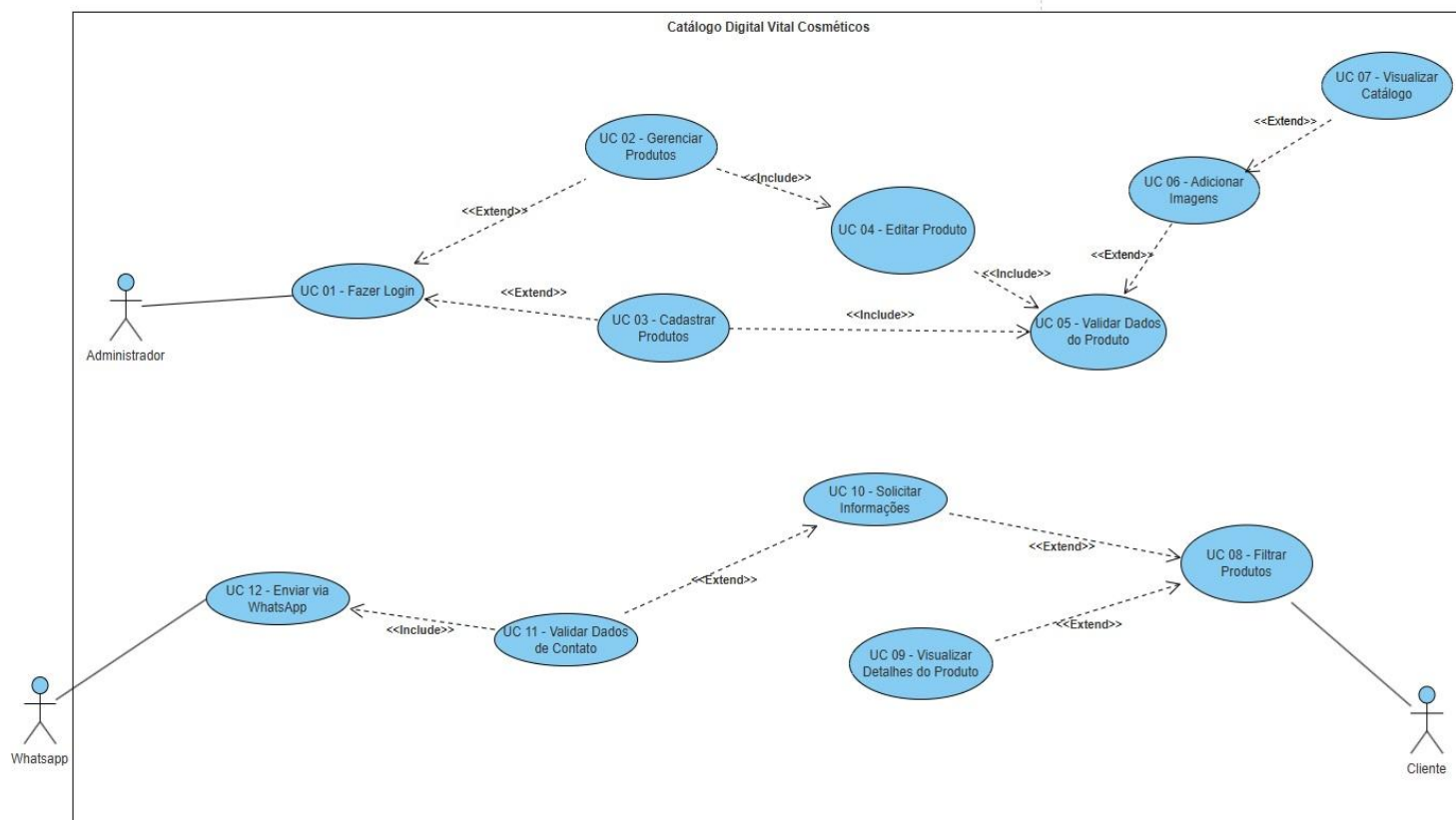
Um Caso de Uso é uma técnica bastante utilizada na Engenharia de Software para representar e descrever as interações funcionais que ocorrem entre os usuários, chamados de atores, e o sistema. De acordo com Sommerville (2011), um caso de uso detalha um cenário específico em que um ator interage com o sistema para atingir um determinado objetivo, sendo uma ferramenta fundamental para a comunicação de requisitos funcionais.

O principal objetivo de um Caso de Uso é representar a sequência de atividades e interações necessárias para que um ator alcance sua meta dentro do sistema. Essa

representação inclui os passos realizados pelo usuário, os dados inseridos, os resultados obtidos e também possíveis exceções ou situações alternativas que possam ocorrer durante o processo.

A Figura 4 representa o diagrama de caso de uso do sistema.

Figura 4: Diagrama de Caso de Uso



Fonte: o autor

A documentação de Casos de Uso é fundamental para representar o cenário em que o sistema será utilizado, juntamente com as regras que ele deve atender, além das possíveis escolhas realizadas pelo usuário.

São apresentados os eventos necessários para a realização das ações tanto no fluxo principal quanto nos alternativos, considerando as condições que devem estar atendidas antes da execução, assim como aquelas que ocorrem após.

Nesse contexto, também são destacados os atores envolvidos, os parâmetros que precisam ser informados, além das restrições e regras de validação que o sistema deve cumprir.

O Quadro 4 representa os principais casos de uso deste sistema.

Quadro 4: Principais Casos de Uso do Sistema

Caso de Uso – Fazer login	
ID	UC 01
Descrição	Este caso de uso terá por objetivo permitir que usuários (administradores e clientes) realizem login no sistema.
Ator Primário	Usuário (Administrador/Cliente)
Pré-condição	O usuário deve possuir credenciais válidas no sistema.
Cenário Principal	1. O usuário acessará a página de login. 2. O sistema exibirá os campos para inserção do e-mail e senha. 3. O usuário deverá inserir suas credenciais e confirmar o login. 4. O sistema validará as credenciais e autenticará o usuário. 5. O usuário será redirecionado para a página inicial ou para sua área específica conforme seu perfil.
Pós-condição	O usuário está autenticado e pode acessar funcionalidades do seu perfil.
Cenário Alternativo	4a – Credenciais inválidas: O sistema exibe mensagem de erro. 4a.1 – O usuário pode tentar novamente ou recuperar senha.
Extend	UC 02 – Gerenciar produtos UC 03 – Cadastrar produtos
Caso de Uso – Gerenciar produtos	
ID	UC 02
Descrição	Este caso de uso terá por objetivo permitir o gerenciamento completo do catálogo de produtos no sistema.
Ator Primário	Administrador
Pré-condição	O usuário deverá estar autenticado como administrador no sistema.
Cenário Principal	1. O administrador acessará a seção de gerenciamento de produtos no sistema. 2. O sistema exibirá uma lista de produtos cadastrados e opções de gerenciamento. 3. O administrador poderá escolher entre criar novo produto, editar existente ou excluir produto. 4. O sistema processará a ação selecionada pelo administrador.
Pós-condição	O catálogo será atualizado conforme a ação realizada.
Cenário Alternativo	3a – O administrador poderá optar por cancelar a operação a qualquer momento.
Include	UC 004 – Editar Produto
Caso de Uso – Cadastrar produto	
ID	UC 03
Descrição	Este caso de uso terá por objetivo permitir o cadastro de novos produtos no sistema.
Ator Primário	Administrador
Pré-condição	O usuário deverá estar autenticado como administrador no sistema.
Cenário Principal	1. O administrador acessará a opção de cadastro de novo produto. 2. O sistema exibirá um formulário para preenchimento com os dados do produto (nome, descrição, categoria, código, referência, status, imagens). 3. O administrador preencherá o formulário com as informações do produto. 4. O sistema validará os dados inseridos.

	5. O sistema salvará o produto no banco de dados.
Pós-condição	O produto será cadastrado no sistema e ficará disponível no catálogo.
Cenário Alternativo	4a – O sistema exibirá mensagens de erro se os dados estiverem incorretos ou incompletos.
Include	UC 05 – Validar dados do produto
Caso de Uso – Editar produto	
ID	UC 04
Descrição	Este caso de uso terá por objetivo permitir a edição de produtos já existentes no sistema.
Ator Primário	Administrador
Pré-condição	O usuário deverá estar autenticado como administrador e ter selecionado um produto para edição.
Cenário Principal	1. O administrador selecionará um produto da lista para edição. 2. O sistema carregará o formulário com os dados atuais do produto. 3. O administrador modificará as informações desejadas. 4. O sistema validará os dados alterados. 5. O sistema atualizará o produto no banco de dados.
Pós-condição	O produto terá suas informações atualizadas no sistema.
Cenário Alternativo	4a – O sistema exibirá mensagens de erro se os dados estiverem incorretos. 3a – O administrador poderá cancelar a edição sem salvar.
Include	UC 005 – Validar Dados do Produto
Caso de Uso – Validar dados do produto	
ID	UC 05
Descrição	Este caso de uso terá por objetivo validar os dados inseridos durante o cadastro ou edição de produtos.
Ator Primário	Sistema
Pré-condição	Dados do produto devem ter sido inseridos ou modificados.
Cenário Principal	1. O sistema verificará se todos os campos obrigatórios foram preenchidos. 2. O sistema validará o formato dos dados inseridos. 3. O sistema confirmará que os dados estão corretos e completos. 4. O sistema permitirá prosseguir com o salvamento.
Pós-condição	Os dados estarão validados e prontos para serem salvos no banco de dados.
Cenário Alternativo	2a – O sistema exibirá mensagens de erro específicas para campos inválidos ou obrigatórios não preenchidos.
Extend	UC 006 – Adicionar Imagens
Caso de Uso – Adicionar imagens	
ID	UC 06
Descrição	Este caso de uso terá por objetivo permitir a adição de imagens aos produtos durante o cadastro ou edição.
Ator Primário	Administrador
Pré-condição	O administrador deverá estar no processo de cadastro ou edição de produto.
Cenário Principal	1. O administrador inserirá URLs de imagens do produto. 2. O sistema validará se as URLs são válidas. 3. O administrador confirmará a adição das imagens. 4. O sistema exibirá preview das imagens adicionadas.

	5. O sistema associará as imagens ao produto.
Pós-condição	As imagens estarão associadas ao produto e disponíveis para visualização.
Cenário Alternativo	2a – O sistema exibirá erro se a URL for inválida. 4a – O administrador poderá remover imagens da lista.
Extend	UC 07 – Visualizar catálogo
Caso de Uso – Visualizar catálogo	
ID	UC 07
Descrição	Este caso de uso terá por objetivo permitir que o administrador visualize gerencie o catálogo de produtos do ponto de vista administrativo.
Ator Primário	Administrador
Pré-condição	O administrador deverá estar autenticado no sistema.
Cenário Principal	1. O administrador acessará a seção de visualização do catálogo no painel administrativo. 2. O sistema exibirá todos os produtos cadastrados com informações completas e status. 3. O administrador poderá visualizar produtos como os clientes veem. 4. O administrador poderá acessar opções de gerenciamento para cada produto. 5. O sistema permitirá pré-visualização de como os produtos aparecem no catálogo público.
Pós-condição	O administrador terá uma visão completa do catálogo e poderá realizar ações de gerenciamento.
Cenário Alternativo	2a – Se não houver produtos cadastrados, o sistema exibirá mensagem informativa e opção para cadastrar novo produto.
Caso de Uso – Filtrar produtos	
ID	UC 08
Descrição	Este caso de uso terá por objetivo permitir que clientes filtrem produtos por categoria e disponibilidade.
Ator Primário	Cliente
Pré-condição	O cliente deverá estar visualizando o catálogo de produtos.
Cenário Principal	1. O cliente utilizará os filtros disponíveis na barra lateral. 2. O cliente selecionará categoria e/ou status de disponibilidade desejado. 3. O sistema aplicará os filtros em tempo real. 4. O sistema exibirá apenas produtos que atendem aos critérios selecionados.
Pós-condição	A lista de produtos será filtrada conforme os critérios selecionados.
Cenário Alternativo	4a – Se nenhum produto atender aos critérios, o sistema exibirá mensagem "Nenhum produto encontrado".
Extend	UC 09 – Visualizar detalhes do produto UC 10 – Solicitar informações
Caso de Uso – Visualizar detalhes do produto	
ID	UC 09
Descrição	Este caso de uso terá por objetivo permitir que clientes visualizem informações detalhadas de um produto específico.
Ator Primário	Cliente
Pré-condição	O cliente deverá estar no catálogo e selecionar um produto específico.
Cenário Principal	1. O cliente clicará em um produto do catálogo. 2. O sistema incrementará o contador de visualizações do produto. 3. O sistema exibirá modal com informações completas do produto.

	4. O cliente poderá visualizar todas as informações e imagens disponíveis. 5. O cliente poderá interagir com opções do produto.
Pós-condição	O cliente terá acesso completo às informações do produto selecionado.
Cenário Alternativo	3a – Se houver erro no carregamento, o sistema exibirá mensagem de erro.
Caso de Uso – Solicitar informações	
ID	UC 10
Descrição	Este caso de uso terá por objetivo permitir que clientes solicitem informações sobre produtos através do sistema de contato.
Ator Primário	Cliente
Pré-condição	O cliente deverá estar interessado em um produto específico ou navegando no catálogo.
Cenário Principal	1. O cliente clicará no botão de solicitação de informações ou chat flutuante. 2. O sistema exibirá formulário de contato. 3. O cliente preencherá seus dados pessoais e mensagem. 4. O sistema processará a solicitação de contato.
Pós-condição	O sistema iniciará o processo de contato conforme solicitado pelo cliente.
Cenário Alternativo	3a – O cliente poderá cancelar a solicitação a qualquer momento.
Extend	UC 11 – Validar dados de contato
Caso de Uso – Validar dados de contato	
ID	UC 11
Descrição	Este caso de uso terá por objetivo validar os dados de contato inseridos pelo cliente.
Ator Primário	Sistema
Pré-condição	O cliente deverá ter preenchido o formulário de contato.
Cenário Principal	1. O sistema verificará se todos os campos obrigatórios foram preenchidos. 2. O sistema validará o formato do e-mail inserido. 3. O sistema verificará se o telefone possui formato válido. 4. O sistema confirmará que todos os dados estão corretos.
Pós-condição	Os dados de contato estarão validados e prontos para salvamento.
Cenário Alternativo	2a – O sistema exibirá erro se o e-mail estiver em formato inválido. 1a – O sistema solicitará preenchimento de campos obrigatórios vazios.
Include	UC 12 – Enviar via WhatsApp
Caso de Uso – Enviar via WhatsApp	
ID	UC 12
Descrição	Este caso de uso terá por objetivo redirecionar o cliente para o WhatsApp com mensagem pré-formatada.
Ator Primário	Cliente
Ator Secundário	Whatsapp
Pré-condição	O cliente deverá ter solicitado informações sobre algum produto.
Cenário Principal	1. O sistema formatará mensagem com dados do cliente e produto de interesse. 2. O sistema gerará URL do WhatsApp com a mensagem pré-formatada. 3. O sistema abrirá WhatsApp em nova aba/aplicativo. 4. O cliente poderá revisar e enviar a mensagem formatada.

Pós-condição	O cliente terá acesso direto ao WhatsApp para finalizar o contato com o administrador.
Cenário Alternativo	4a – O cliente poderá optar por não enviar a mensagem e fechar o WhatsApp. 3a – Se o WhatsApp não estiver disponível, o sistema exibirá mensagem de erro.

Fonte: o autor

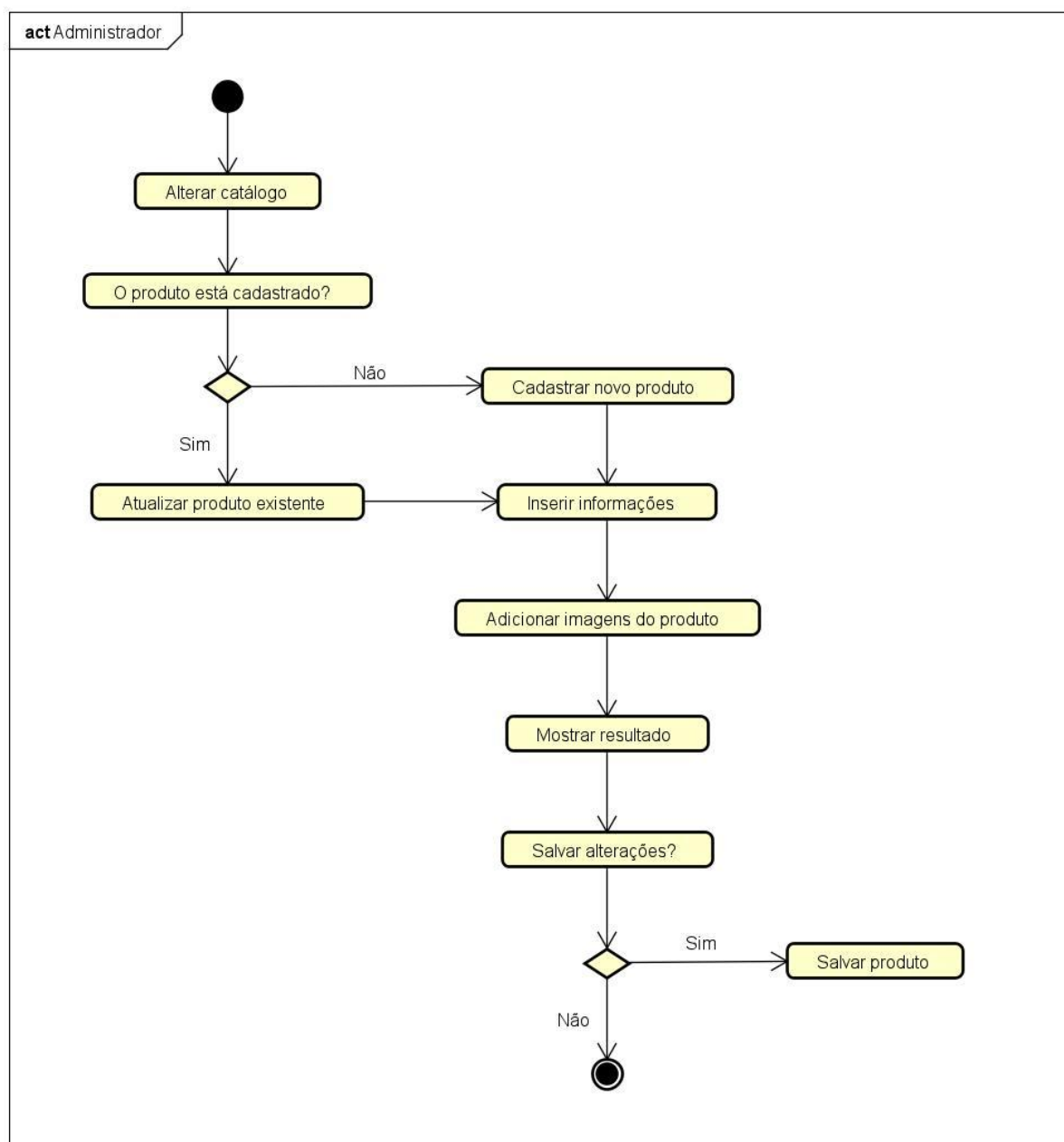
3.7 Diagrama de Atividades

O Diagrama de Atividades é uma ferramenta da *Unified Modeling Language* (UML) que permite representar graficamente o fluxo de atividades de um sistema, descrevendo o comportamento dinâmico de processos e suas interações. Segundo Pressman (2016), esse diagrama modela o fluxo de controle, mostrando a sequência de ações, decisões e paralelismos dentro de um processo.

A utilização desse diagrama é fundamental, pois proporciona uma visão clara e objetiva sobre como as atividades ocorrem, quais são os pontos de decisão e quais são os caminhos possíveis dentro de um processo. Além disso, contribui significativamente para a melhoria da comunicação entre os membros da equipe de desenvolvimento, facilitando a compreensão das regras de negócio e dos fluxos operacionais Sommerville (2011).

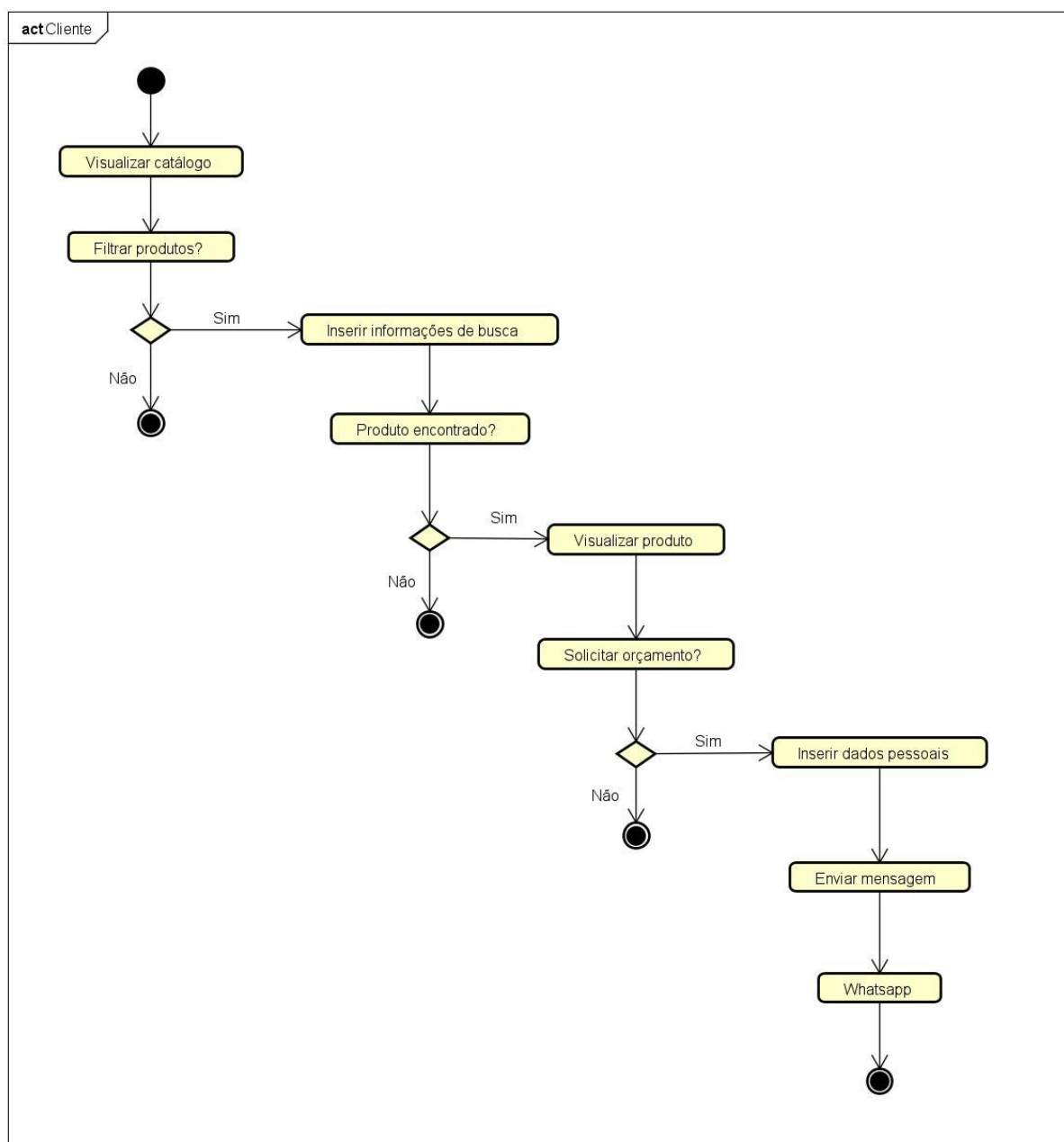
A seguir, são apresentados os Diagramas de Atividades desenvolvidos para o sistema de catálogo digital da empresa Vital Cosméticos, contemplando os principais fluxos operacionais tanto da área administrativa quanto da área do cliente.

Na Figura 5, é apresentado o Diagrama de Atividades referente ao fluxo de cadastro e atualização de produtos realizado pelo administrador do sistema.

Figura 5: Diagrama de Atividades Administrador

Fonte: o autor

Na Figura 6, é apresentado o Diagrama de Atividades referente ao fluxo de interação do cliente com o catálogo de produtos.

Figura 6: Diagrama de Atividades Cliente

Fonte: o autor

3.8 Matriz de Rastreabilidade

A Matriz de Rastreabilidade é uma ferramenta essencial no desenvolvimento de *software*, pois permite acompanhar e validar se todos os requisitos definidos estão devidamente alinhados às regras de negócio e às regras sistêmicas do sistema.

Segundo Pressman (2016), a rastreabilidade assegura que cada requisito seja devidamente ligado às necessidades do negócio, permitindo uma gestão eficaz de mudanças, verificações e validações. De acordo com Sommerville (2011), essa prática é fundamental para garantir que o sistema permaneça consistente com os objetivos da organização ao longo de seu ciclo de vida.

Essa matriz desempenha um papel fundamental no controle de qualidade e no acompanhamento do projeto, pois permite:

- Validar a cobertura dos requisitos, assegurando que todos estejam associados às regras de negócio e aos comportamentos esperados do sistema.
- Facilitar o rastreamento e o controle de mudanças, possibilitando rápida identificação dos impactos de alterações em regras ou requisitos.
- Apoiar na verificação e validação do sistema, assegurando que nenhuma necessidade ou diretriz do negócio seja esquecida durante o desenvolvimento.

A Tabela 1 apresenta a matriz que relaciona os requisitos funcionais (RF) e não funcionais (RNF) com as regras de negócio (RN) da empresa. Nela, a marcação “X” indica que um determinado requisito está diretamente vinculado a uma ou mais regras de negócio.

Tabela 1: Matriz de Rastreabilidade – Requisitos x Regras de Negócio

	RN001	RN002	RN003	RN004	RN005	RN006	RN007	RN008	RN009	RN010	RN011
RF001	X										
RF002			X								
RF003					X						
RF004				X							
RF005						X					
RF006			X								
RF007							X				
RF008								X			
RF009							X				
RNF001		X									
RNF002									X		
RNF003								X			
RNF004										X	
RNF005					X						
RNF006											X

Fonte: o autor

A Tabela 2 apresenta a matriz de rastreabilidade que relaciona os requisitos com as regras sistêmicas, derivadas dos casos de uso e dos comportamentos esperados do sistema. Esta matriz assegura que todos os requisitos estejam adequadamente implementados dentro da lógica do sistema.

Tabela 2: Matriz de Rastreabilidade – Requisitos x Regras Sistêmicas

	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09	UC10	UC11	UC12
RF001		X	X		X	X	X					
RF002		X		X	X	X	X					
RF003								X	X			
RF004		X										
RF005									X			
RF006							X	X	X			
RF007	X											
RF008										X	X	X
RF009							X					
RNF001	X	X	X	X				X		X		
RNF002							X	X	X	X		
RNF003												X
RNF004							X	X	X			
RNF005							X	X	X	X		
RNF006		X	X	X								

Fonte: o autor

4 Ferramentas e Métodos

As ferramentas selecionadas para o desenvolvimento do projeto foram escolhidas levando em consideração critérios como desempenho, escalabilidade e ampla aceitação pela comunidade de desenvolvedores.

Além disso, todas contam com documentação completa, recursos didáticos e uma vasta quantidade de tutoriais disponíveis, o que facilita tanto o aprendizado quanto a aplicação das soluções no desenvolvimento do sistema.

Outro fator relevante na escolha foi a afinidade da equipe de desenvolvimento, que já possuía experiência prévia no uso de algumas dessas ferramentas, o que contribuiu para otimizar o processo de desenvolvimento.

A maioria das ferramentas utilizadas possui licença de código aberto ou versão gratuita, permitindo sua utilização tanto em projetos acadêmicos quanto profissionais e comerciais sem custos adicionais.

4.1 Ferramentas Utilizadas

DOCUMENTAÇÃO E RELATÓRIOS:

MICROSOFT WORD

- **Versão:** Microsoft 365 (2023)
- **Licença:** proprietária (comercial)
- **Site Oficial:** <https://www.microsoft.com/word>

O Microsoft Word foi utilizado para elaboração de toda a documentação textual do projeto, como TAP, documentação geral e relatórios. Foi escolhido por ser um processador de texto robusto, amplamente utilizado, com recursos avançados de formatação, revisão e integração com outros aplicativos do pacote Office.

MATRIZES:

MICROSOFT EXCEL

- **Versão:** Microsoft 365 (2025)
- **Licença:** proprietária (comercial)
- **Site Oficial:** <https://www.microsoft.com/excel>

O Excel foi utilizado na criação das matrizes de rastreabilidade. Sua escolha se deu pela facilidade na criação de tabelas, fórmulas, organização de dados e estruturação visual das informações.

CANVAS E SWOT:

CANVA

- **Versão:** *online* (2025)
- **Licença:** gratuito com recursos limitados (com opção de assinatura para recursos avançados)
- **Site Oficial:** <https://www.canva.com>

O Canva foi escolhido para a elaboração do *Business Model Canvas* e da análise SWOT, além da prototipação das telas do sistema. A ferramenta oferece uma interface intuitiva, com uma ampla variedade de modelos, elementos gráficos e facilidade na criação de artefatos visuais de alta qualidade.

BPMN:

DRAW.IO

- **Versão:** *online* (2025)
- **Licença:** código aberto
- **Site Oficial:** <https://www.drawio.com>

O Draw.io foi utilizado para a modelagem de processos de negócio através do diagrama BPMN. A escolha foi motivada pela praticidade, acesso gratuito, integração com armazenamento na nuvem e suporte a diversos tipos de diagramas.

DIAGRAMA DE CASO DE USO:

VISUAL PARADIGM ONLINE

- **Versão:** *online* (2025)
- **Licença:** gratuito para uso básico, com opções de assinatura para recursos avançados
- **Site Oficial:** <https://online.visual-paradigm.com>

O Visual Paradigm Online foi utilizado na criação do Diagrama de Casos de Uso, por ser uma ferramenta intuitiva, com suporte ao padrão UML e modelos prontos que facilitam a elaboração de diagramas de forma rápida e organizada.

DIAGRAMA DE ATIVIDADES E MÁQUINA DE ESTADOS:

ASTAH COMMUNITY

- **Versão:** 8.5 (2025)
- **Licença:** gratuito para uso não comercial (versão Community)
- **Site Oficial:** <https://astah.net/>

O Astah foi utilizado para a elaboração do Diagrama de Atividades e Diagrama de Máquina de Estados. Foi escolhido por sua simplicidade, precisão na representação dos fluxos e conformidade com os padrões UML.

PROTOTIPAÇÃO DE TELAS:

CANVA

- **Versão:** *online* (2023)
- **Licença:** gratuito com recursos limitados (com opção de assinatura para recursos avançados)
- **Site Oficial:** <https://www.canva.com>

O Canva também foi utilizado para a prototipação das telas da aplicação, como a tela de login, catálogo de produtos, favoritos e *dashboard* administrativo. A

escolha se deu pela interface amigável e pela vasta biblioteca de elementos gráficos, que permite criar protótipos com aparência profissional, sem necessidade de conhecimento técnico em design.

4.2 Tecnologias Utilizadas no Sistema

As tecnologias utilizadas no desenvolvimento do sistema foram selecionadas por oferecerem escalabilidade, facilidade de manutenção, robustez e excelente performance para aplicações *web* modernas.

LINGUAGEM DE PROGRAMAÇÃO:

TYPESCRIPT

- **Versão:** 5.5.3
- **Licença:** MIT (código aberto)
- **Site Oficial:** <https://www.typescriptlang.org>

O TypeScript foi escolhido por adicionar tipagem estática ao JavaScript, oferecendo mais robustez, segurança e legibilidade no código, além de facilitar a detecção de erros durante o desenvolvimento.

FRAMEWORK DE *FRONTEND*:

REACT.JS

- **Versão:** 18.3.1
- **Licença:** MIT (código aberto)
- **Site Oficial:** <https://reactjs.org>

O React.js é uma biblioteca JavaScript utilizada na construção da interface de usuário. Foi escolhido por sua capacidade de criar aplicações *Single Page Application* (SPA), modularidade, escalabilidade e ampla adoção pela comunidade.

FERRAMENTA DE *BUILD*:

VITE

- **Versão:** 5.4.2
- **Licença:** MIT (código aberto)
- **Site Oficial:** <https://vitejs.dev>

O Vite foi utilizado como ferramenta de *build* por sua alta velocidade e simplicidade, tanto em ambiente de desenvolvimento quanto na geração da versão de produção da aplicação.

FRAMEWORK DE ESTILIZAÇÃO:

TAILWIND CSS

- **Versão:** 3.4.1
- **Licença:** MIT (código aberto)
- **Site Oficial:** <https://tailwindcss.com>

Tailwind CSS foi utilizado para a criação de uma interface responsiva, moderna e personalizável, baseada em classes utilitárias que aceleram o desenvolvimento e garantem padronização visual.

GERENCIAMENTO DE ROTAS:

REACT ROUTER DOM

- **Versão:** 6.21.0
- **Licença:** MIT (código aberto)
- **Site Oficial:** <https://reactrouter.com>

O React Router Dom foi utilizado para gerenciar as rotas da aplicação, permitindo navegação eficiente e dinâmica entre as diferentes páginas dentro do modelo SPA.

BACKEND COMO SERVIÇO:

SUPABASE

- **Versão:** 2.50.0
- **Licença:** MIT (código aberto)
- **Site Oficial:** <https://supabase.com>

O Supabase foi utilizado como *Backend as a Service* (BaaS), oferecendo autenticação de usuários, banco de dados PostgreSQL, armazenamento de arquivos e APIs automáticas em tempo real, eliminando a necessidade de um *backend* tradicional.

BANCO DE DADOS:

POSTGRESQL (via Supabase)

- **Versão:** Incorporado no Supabase

- **Licença:** PostgreSQL License (código aberto)
- **Site Oficial:** <https://www.postgresql.org>

O PostgreSQL foi utilizado como sistema gerenciador de banco de dados relacional (SGBD), garantindo escalabilidade, segurança e confiabilidade no armazenamento dos dados.

BIBLIOTECAS E FERRAMENTAS ADICIONAIS:

- Lucide React – biblioteca de ícones vetoriais;
- React Hook Form – gerenciamento eficiente de formulários;
- React Select – componentes avançados de seleção;
- html2canvas – captura de tela de elementos HTML;
- jsPDF – geração de documentos PDF diretamente no navegador;
- ESLint e TypeScript ESLint – ferramentas para análise estática de código, garantindo qualidade, padronização e boas práticas.

CONTROLE DE VERSÃO:

GIT

- **Versão:** 2.40.0
- **Licença:** GPL (código aberto)
- **Site Oficial:** <https://git-scm.com>

O Git foi utilizado para o controle de versão, permitindo organização, rastreabilidade, histórico de alterações e colaboração entre desenvolvedores, com os repositórios hospedados no GitHub.

4.3 Métodos de Execução do Projeto da Interface

Para a construção do protótipo da interface do sistema de catálogo digital da Vital Cosméticos, foram aplicados os seguintes métodos:

Prototipação: a etapa de prototipação foi essencial para definir a estrutura visual e a navegação da interface *User Interface* (UI) do sistema. Por meio da ferramenta Canva, foram desenvolvidos protótipos das principais telas, como login, listagem de produtos, área de favoritos e painel administrativo. Esses protótipos passaram por revisões constantes baseadas no retorno dos *stakeholders*, assegurando que o layout e a usabilidade estivessem alinhados às expectativas dos usuários.

Desenvolvimento Iterativo: a interface foi desenvolvida utilizando uma abordagem iterativa, baseada em ciclos de desenvolvimento incremental. A cada ciclo, foram implementadas funcionalidades específicas, começando pelas de maior prioridade, como autenticação, exibição de produtos e gerenciamento de favoritos. Após cada entrega, foram realizados testes, ajustes e incrementos, visando melhorias contínuas.

Design Responsivo: uma das prioridades durante o desenvolvimento foi garantir que o sistema funcionasse adequadamente em diferentes tamanhos de tela, como desktops, tablets e smartphones. Para isso, foram empregadas boas práticas de CSS, o framework Tailwind CSS e tecnologias do React.js, que oferecem recursos que facilitam o desenvolvimento de interfaces adaptáveis e responsivas.

Testes de Usabilidade: durante o desenvolvimento, foram conduzidos testes de usabilidade com usuários representativos, que ajudaram a identificar possíveis melhorias na navegação, acessibilidade e interação. As correções feitas a partir desses testes contribuíram diretamente para aperfeiçoar a experiência do usuário.

Integração Contínua: adotou-se o conceito de integração contínua ao longo do desenvolvimento, garantindo que todo o código produzido fosse constantemente sincronizado e versionado. O uso do GitHub, aliado ao Git, facilitou a organização, rastreabilidade das mudanças e o trabalho colaborativo entre os membros da equipe, evitando conflitos e retrabalho.

5 Desenvolvimento do protótipo de software

A metodologia de desenvolvimento focou na utilização de tecnologias modernas para criar uma aplicação *web* robusta, interativa e de fácil manutenção. O processo iniciou-se com a definição da arquitetura do sistema, onde a principal decisão arquitetônica foi a adoção de um *Backend as a Service* (BaaS), o que permitiu agilizar o desenvolvimento ao abstrair grande parte da complexidade de infraestrutura de servidor.

A escolha recaiu sobre a plataforma Supabase, que oferece banco de dados PostgreSQL, autenticação segura e APIs prontas para uso. O desenvolvimento do *frontend* foi baseado na biblioteca React.js em conjunto com TypeScript, garantindo uma base de código fortemente tipada, modular e escalável. Para o ambiente de

desenvolvimento e *build* da aplicação, foi utilizado o Vite.js, conhecido por sua alta performance e configuração simplificada.

5.1 *Backend* e Persistência de Dados com Supabase

Diferente de uma abordagem tradicional com um servidor customizado, o catálogo utiliza o Supabase como sua espinha dorsal de *backend*. Esta plataforma oferece um banco de dados PostgreSQL, um sistema de autenticação seguro e APIs *RESTful* auto-geradas, que são consumidas diretamente pelo *frontend*. A conexão com o Supabase é estabelecida através de um cliente JavaScript, que utiliza as chaves de API para autenticar as requisições de forma segura, como demonstrado na Figura 9.

Figura 9: Configuração do cliente Supabase para conexão com o *backend*

```
import { createClient } from '@supabase/supabase-js';

const supabaseUrl = 'https://rmajyriqfpbtzigxuvca.supabase.co';
const supabaseAnonKey = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...';

export const supabase = createClient(supabaseUrl, supabaseAnonKey);
```

Fonte: o autor

5.1.1 Implementação de API para Cadastro de Produtos

O sistema utiliza as APIs *RESTful* automaticamente geradas pelo Supabase para realizar operações de cadastro, leitura, atualização e exclusão de produtos. A seguir, são apresentados exemplos práticos da implementação da API de cadastro de produtos, demonstrando como a lógica de negócio é integrada com o *backend*.

Figura 10: Implementação da função de cadastro de produtos

```
const addProduct = async (productData) => {
  try {
    // 1. Buscar categoria relacionada
    const { data: categoryData } = await supabase
      .from('categories')
      .select('id')
      .eq('name', productData.category)
      .single();

    if (!categoryData) {
      throw new Error('Categoria não encontrada');
    }

    // 2. Inserir novo produto
    const { data, error } = await supabase
      .from('products')
      .insert([
        {
          name: productData.name,
          description: productData.description,
          category_id: categoryData.id,
          code: productData.code,
          reference: productData.reference,
          stock: productData.stock,
          images: productData.images
        }
      ])
      .select('*', categories(name))
      .single();

    // 3. Tratamento de erros e atualização do estado
    if (error) throw new Error(error.message);

    setProducts(prev => [data, ...prev]);
    return data;
  } catch (error) {
    console.error('Erro ao cadastrar produto:', error);
    throw error;
  }
};
```

Fonte: o autor

Figura 11: Integração da API no componente de formulário

```
const onSubmit = async (data) => {
  setIsLoading(true);

  try {
    // 1. Validação dos dados obrigatórios
    if (!data.name.trim() || !data.category || !data.code.trim()) {
      throw new Error('Campos obrigatórios não preenchidos');
    }

    // 2. Chamada da API de cadastro
    await addProduct({
      name: data.name,
      description: data.description,
      category: data.category,
      code: data.code,
      reference: data.reference,
      stock: data.stock,
      images: data.images
    });

    // 3. Feedback positivo e reset do formulário
    showToast('Produto cadastrado com sucesso!', 'success');
    reset();
    onClose();

  } catch (error) {
    console.error('Erro ao salvar produto:', error);
    showToast(`Erro ao cadastrar: ${error.message}`, 'error');
  } finally {
    setIsLoading(false);
  }
};
```

Fonte: o autor

Figura 12: Exemplo de resposta da API RESTful para consulta de produtos

Fonte: o autor

5.1.2 Características da Implementação

Comunicação com o Backend: O sistema utiliza o cliente JavaScript oficial do Supabase (@supabase/supabase-js) para estabelecer comunicação com o *backend*. As requisições HTTP são automaticamente geradas para as APIs RESTful do Supabase, eliminando a necessidade de configuração manual de *endpoints*. A biblioteca oferece suporte nativo a todas as operações *CRUD* através de uma interface fluente e intuitiva.

Validação e Segurança: A implementação incorpora múltiplas camadas de validação, iniciando no *frontend* com o React Hook Form, que realiza validações em tempo real dos campos obrigatórios. No *backend*, o Supabase aplica *Row Level Security* (RLS) configurado no banco de dados, garantindo que apenas usuários autorizados possam realizar operações específicas. A tipagem TypeScript adiciona segurança de tipos em tempo de compilação.

Tratamento de Erros: O sistema implementa uma estratégia robusta de tratamento de erros em múltiplas camadas. Erros são capturados tanto na comunicação com a API quanto na lógica de negócio, com mensagens descritivas para diferentes tipos de erro. O sistema inclui *fallbacks* para situações inesperadas, garantindo estabilidade da aplicação.

Otimização de Performance: A implementação utiliza *joins* otimizados com a tabela de categorias para minimizar requisições ao banco. O estado local é atualizado imediatamente após operações bem-sucedidas, proporcionando feedback instantâneo ao usuário. Para grandes volumes de dados, o sistema implementa *lazy loading* e paginação.

Experiência do Usuário: Durante operações assíncronas, o sistema exibe *loading states* para informar o progresso das ações. O feedback visual é fornecido através de *toast notifications* que informam sobre sucessos ou falhas. Os formulários incluem validação em tempo real, permitindo correção de erros antes do envio.

5.1.3 Arquitetura de Dados

O banco de dados PostgreSQL é estruturado com tabelas relacionais que mantêm a integridade referencial. A tabela *products* possui relacionamento com *categories* através de chave estrangeira, permitindo queries eficientes e consistência de dados. O Supabase gerencia automaticamente *timestamps* de criação e atualização, além de implementar triggers para operações específicas como incremento de visualizações.

Esta implementação demonstra como o Supabase abstrai a complexidade de infraestrutura de *backend*, permitindo que o desenvolvedor foque na lógica de negócio e na experiência do usuário, mantendo a robustez e segurança necessárias para um sistema de produção. A arquitetura *Backend as a Service* elimina a necessidade de desenvolvimento e manutenção de servidor customizado, reduzindo significativamente o tempo de desenvolvimento e os custos operacionais.

5.2 Desenvolvimento do *Frontend* e Gerenciamento de Estado

O *frontend* foi construído com React.js, permitindo a criação de uma interface de usuário reativa e baseada em componentes. O uso de TypeScript adicionou segurança de tipo, reduzindo erros em tempo de execução e melhorando a legibilidade do código. Para a estilização, foi adotado o framework Tailwind CSS, que possibilita a criação de interfaces complexas de forma rápida e consistente através de classes utilitárias.

O roteamento da aplicação é gerenciado pela biblioteca React Router DOM, que implementa a navegação no modelo de *Single Page Application* (SPA). Foram definidas rotas públicas (catálogo, favoritos, login) e rotas privadas para o painel administrativo. O acesso às rotas administrativas é controlado por um componente de ordem superior (ProtectedRoute), que verifica o status de autenticação do usuário antes de renderizar o conteúdo, como demonstrado na Figura 13.

Figura 13: Implementação de rota protegida para a área administrativa

```
function ProtectedRoute({ children }: { children: React.ReactNode }) {
  const { user, loading } = useAuth();

  if (loading) {
    return (
      <div className="min-h-screen flex items-center justify-center">
        <div className="text-center">
          <div className="w-8 h-8 border-4 border-[#183263] border-t-transparent
            rounded-full animate-spin mx-auto mb-4"></div>
          <p className="text-gray-600">Verificando acesso...</p>
        </div>
      </div>
    );
  }

  if (!user || user.role !== 'admin') {
    return <Navigate to="/login" replace />;
  }

  return <>{children}</>;
}
```

Fonte: o autor

5.2.1 Gerenciamento de Estado e Custom Hooks

A lógica de negócios e as interações com o Supabase foram encapsuladas em Custom Hooks do React, como useAuth e useProducts. Essa abordagem promove a

reutilização de código e a separação de responsabilidades. O estado global, como informações de notificação e *toasts*, é gerenciado através da Context API do React.

A Figura 14 apresenta o componente *ProductForm*, responsável pelo cadastro e edição de produtos, demonstrando o uso de estado local *useState* e a integração com a biblioteca *React Hook Form* para um gerenciamento eficiente e performático dos formulários.

Figura 14: Estrutura do componente de formulário para cadastro e edição de produtos

```
export function ProductForm({ product, isOpen, onClose }: ProductFormProps) {
  const { addProduct, updateProduct, categories } = useProducts();
  const { showToast } = useToast();
  const [isLoading, setIsLoading] = useState(false);

  const { register, handleSubmit, formState: { errors }, reset } = useForm<ProductFormData>({
    defaultValues: product ? {
      name: product.name,
      description: product.description,
      category: product.category,
      code: product.code,
      reference: product.reference,
      stock: product.stock,
      images: product.images,
    } : {
      name: '', description: '', category: '', code: '',
      reference: '', stock: 'available', images: []
    }
  });

  // Implementação do onSubmit conforme Figura 11
  const onSubmit = async (data: ProductFormData) => { /* ... */ };

  return (
    // Estrutura do formulário com validações e campos
  );
}
```

Fonte: o autor

5.2.2 Arquitetura de Componentes

A aplicação segue uma arquitetura baseada em componentes reutilizáveis, organizados em uma estrutura hierárquica clara. Os componentes são categorizados em:

- **Componentes de Layout:** responsáveis pela estrutura geral da aplicação (Header, Footer, AdminLayout)

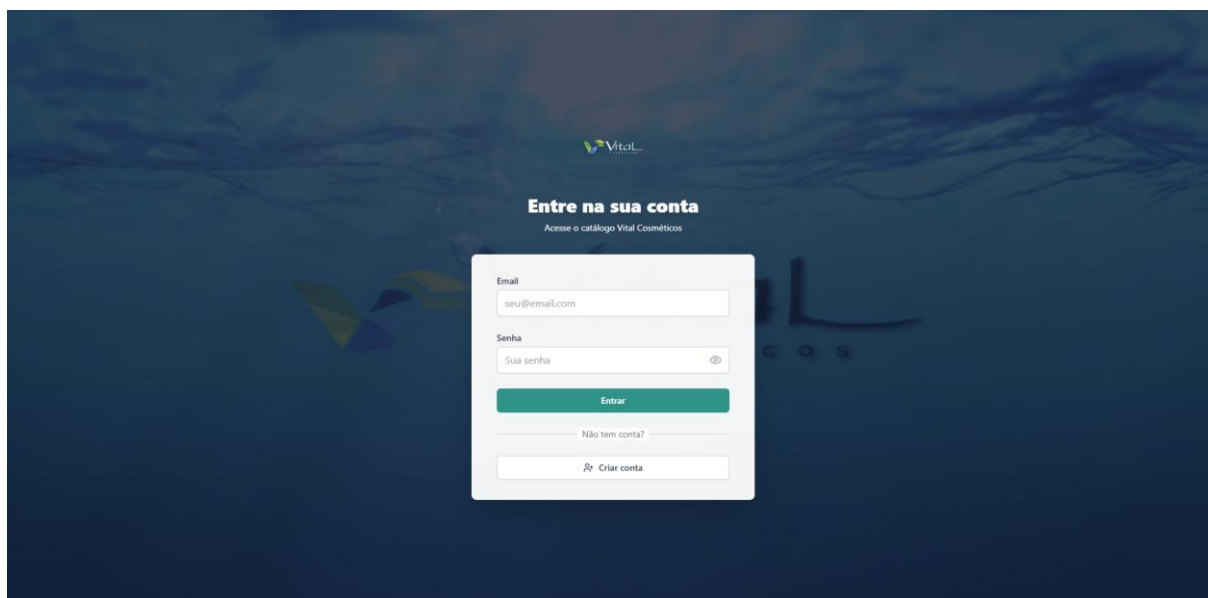
- **Componentes de Interface:** elementos de interação do usuário (ProductCard, ProductModal, FloatingChat)
- **Componentes Administrativos:** funcionalidades específicas do painel administrativo (ProductForm, Dashboard, Contacts)
- **Componentes de Estado:** gerenciamento de contexto e estado global (AppContext, FavoritesContext, ToastContext)

Esta organização promove a facilidade de manutenção, a implementação de testes e a escalabilidade do código, seguindo as melhores práticas de desenvolvimento React.js e garantindo uma arquitetura robusta para futuras expansões do sistema.

6 Resultados e Discussão

Neste capítulo, são apresentados os resultados do desenvolvimento do protótipo do catálogo digital de produtos, com foco nas principais interfaces e funcionalidades implementadas. Cada tela foi projetada para ser intuitiva e funcional, atendendo aos requisitos de um sistema de catálogo digital moderno.

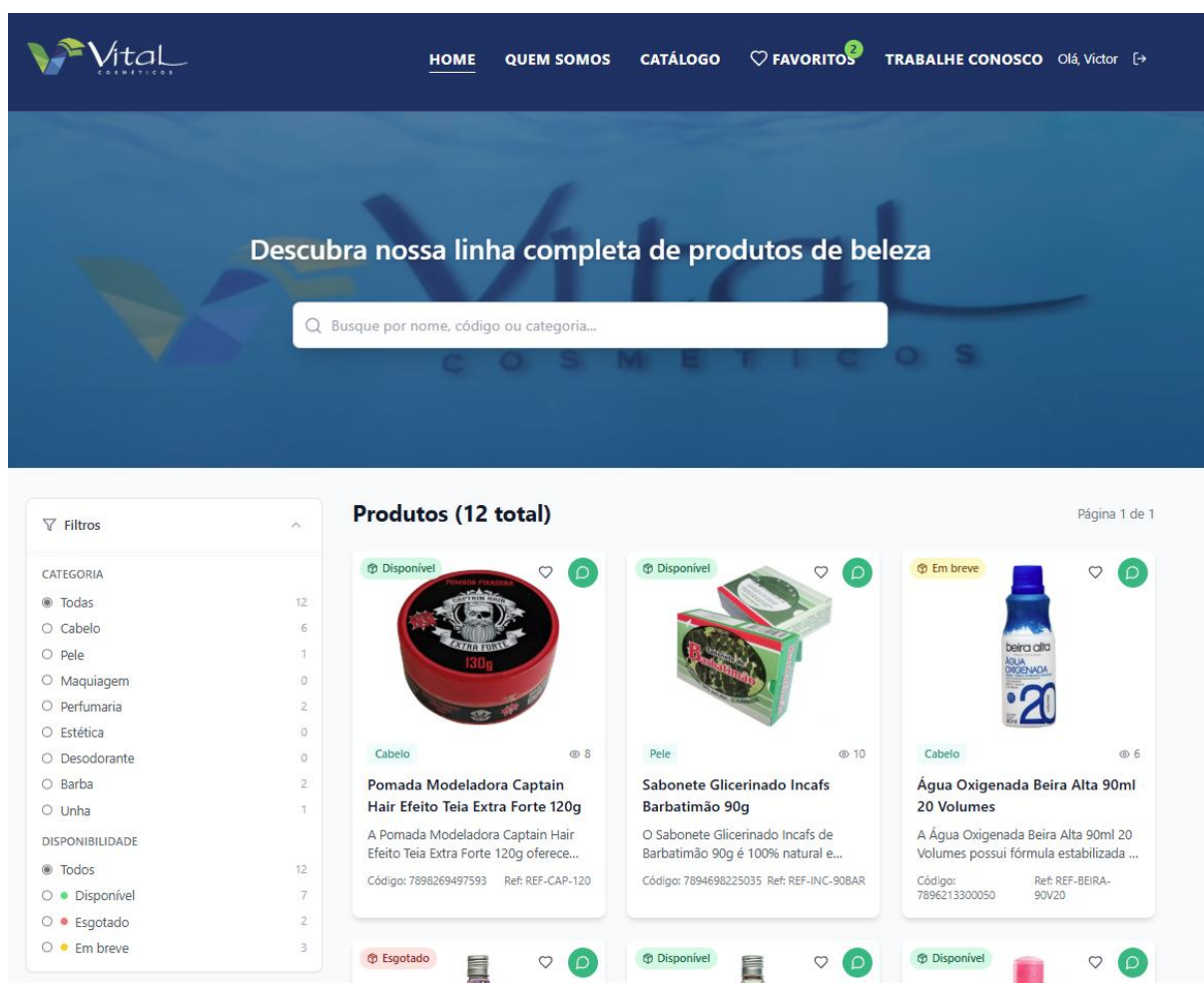
A Figura 15 apresenta a tela de autenticação do sistema. Esta página funciona como um portal de entrada unificado tanto para clientes quanto para administradores. A interface foi projetada de forma minimalista, com campos para e-mail e senha, e a lógica de autenticação é gerenciada pelo serviço Supabase Auth, o que assegura um processo de login seguro e robusto. Após a validação das credenciais, o sistema identifica o papel do usuário e o redireciona para a área correspondente: o painel de gerenciamento para os administradores ou as funcionalidades personalizadas, como a lista de favoritos, para os clientes.

Figura 15: Página de autenticação

Fonte: o autor

A Figura 16 demonstra a tela principal do sistema, que funciona como o catálogo de produtos público. Nela, os produtos são exibidos em um *layout* de grade, permitindo fácil visualização e navegação. A interface inclui componentes essenciais como uma barra de busca e filtros de categoria, permitindo ao usuário encontrar itens específicos de forma eficiente. Esta tela é o coração da experiência do usuário, construída com React para ser dinâmica e responsiva, utilizando Tailwind CSS para a estilização.

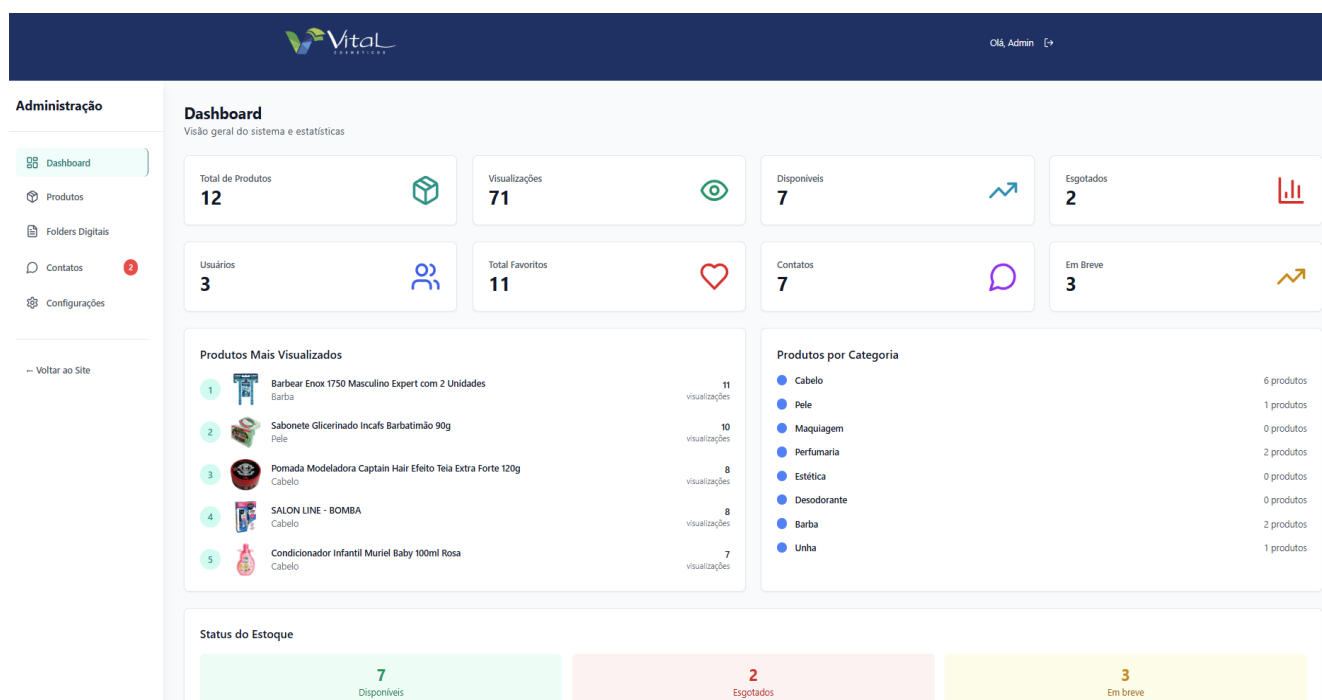
Figura 16: Tela principal do catálogo de produtos



Fonte: o autor

A Figura 17 exibe o *dashboard*, a tela principal da área administrativa do sistema. Esta interface oferece uma visão geral e consolidada das principais métricas e estatísticas do catálogo. Através de cartões informativos, o administrador pode monitorar em tempo real dados cruciais como o número total de produtos, o volume de visualizações, a quantidade de contatos recebidos e o status de estoque (disponíveis, esgotados, em breve). Além disso, o painel apresenta listas dinâmicas com os produtos mais populares e a distribuição de itens por categoria, fornecendo informações para a tomada de decisão estratégica e o gerenciamento do negócio.

Figura 17: Painel de gerenciamento de produtos



Fonte: o autor

A Figura 18 detalha o formulário de cadastro e edição de produtos, acessado a partir do painel administrativo. Esta interface permite que o administrador insira todas as informações pertinentes a um produto, como nome, descrição, categoria, código, status de estoque e imagens. O formulário foi desenvolvido com a biblioteca React Hook Form para validações em tempo real e gerenciamento eficiente do estado, e se integra ao Supabase Storage para o upload e armazenamento das imagens. Esta tela é central para a alimentação do catálogo, garantindo que os dados inseridos sejam completos e consistentes.

Figura 18: Formulário de cadastro e edição de produto

Fonte: o autor

A Figura 19 ilustra a funcionalidade de contato direto, implementada através de um componente de chat flutuante. Quando um usuário demonstra interesse em um produto, este componente é ativado, abrindo um formulário para que o cliente insira suas informações de contato (nome e telefone). A mensagem já vem pré-preenchida com o nome do produto de interesse, agilizando o processo de comunicação. Esta funcionalidade representa um canal direto de captura de *leads*, conectando o interesse do cliente a uma ação de contato de forma prática e imediata.

Figura 19: Funcionalidade de contato por produto

Fonte: o autor

Considerações finais

O desenvolvimento do sistema de catálogo digital para a Vital Cosméticos alcançou os objetivos propostos no início do projeto. Durante todo o processo, foram superados desafios tanto técnicos quanto relacionados à usabilidade, que contribuíram significativamente para o aperfeiçoamento da aplicação, resultando em uma solução prática, intuitiva e eficiente para os usuários.

O projeto foi desenvolvido em etapas bem definidas, passando pela elaboração dos artefatos de análise e modelagem, construção do banco de dados e desenvolvimento das funcionalidades essenciais para a gestão dos produtos, pastas digitais e informações da empresa. Todo o processo foi realizado utilizando tecnologias modernas e alinhadas às necessidades atuais do mercado.

Através desse sistema, a empresa agora dispõe de uma ferramenta que permite organizar e divulgar seu portfólio de produtos de forma mais eficiente, ágil e profissional, além de reduzir a dependência de materiais impressos e processos

manuals. A plataforma proporciona uma experiência mais dinâmica tanto para os administradores da empresa quanto para os clientes, que podem acessar os produtos, visualizar informações e até salvar seus itens favoritos.

Acreditamos que, com as futuras melhorias e atualizações, o sistema poderá se tornar ainda mais robusto e adaptável, contribuindo diretamente para o crescimento digital da Vital Cosméticos e oferecendo uma solução que acompanhe a evolução das demandas do mercado e da própria empresa.

Referências

ASTAH. **Astah Community – Software de modelagem UML**. Disponível em: <<https://astah.net/>>. Acesso em: 20.ago.2024.

ANALISTA MODELOS DE NEGÓCIOS. **O que é o Business Model Canvas?**, 2022. Disponível em: <<https://analistamodelosdenegocios.com.br/o-que-e-o-business-model-canvas>>. Acesso em: 22.ago.2024.

CANVA. **Plataforma de design gráfico e prototipação de interfaces**. Disponível em: <<https://www.canva.com>>. Acesso em: 02.set.2024.

DRAW.IO. **Ferramenta de criação de diagramas online**. Disponível em: <<https://www.diagrams.net>>. Acesso em: 20.ago.2024.

GIT. **Git - Distributed version control system**. Disponível em: <<https://git-scm.com>>. Acesso em: 25.ago.2024.

INSTITUTE OF PROJECT MANAGEMENT. **A guide to the project management body of knowledge (PMBOK® Guide)**. 6. ed. Newtown Square, PA: Project Management Institute, 2017.

NAVATHE, S.; ELMASRI, R. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson, 2011.

PORTER, Michael E. **Competição: estratégias competitivas essenciais**. 2. ed. Rio de Janeiro: Campus, 1986. 459 p.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL: The world's most advanced open source relational database**. Disponível em: <<https://www.postgresql.org>>. Acesso em: 05.mai.2025.

PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional**. 8. ed. São Paulo: McGraw Hill Brasil, 2016.

REACTJS. React – **Uma biblioteca JavaScript para criar interfaces de usuário**. Disponível em: <<https://react.dev>>. Acesso em: 10.mai.2025.

SOMMERVILLE, I. **Software Engineering**. 9ª Edição. Boston, MA: Pearson Education, 2011.

SUPABASE. **Plataforma Backend as a Service (BaaS)**. Disponível em: <<https://supabase.com>>. Acesso em: 01.jun.2025.

TAILWIND CSS. **Framework utility-first CSS**. Disponível em: <<https://tailwindcss.com>>. Acesso em: 03.jun.2025.

TYPESCRIPT. **TypeScript: JavaScript with syntax for types**. Disponível em: <<https://www.typescriptlang.org>>. Acesso em: 05.jun.2025.

VITE. **Vite: Next Generation Frontend Tooling**. Disponível em: <<https://vitejs.dev>>. Acesso em: 10.jun.2025.

VISUAL PARADIGM. **Visual Paradigm Online – Ferramenta de modelagem UML**. Disponível em: <<https://online.visual-paradigm.com>>. Acesso em: 25.ago.2024.

Anexo 1

Missão:

Conquistar e manter uma relação onde possamos satisfazer as expectativas de nossos clientes com diferencial no atendimento, fortalecendo a cada dia nosso relacionamento e confiança.

Visão:

Ser reconhecida como líder em inovação e qualidade no mercado de cosméticos, oferecendo produtos diferenciados e soluções sustentáveis que atendam de forma eficaz às necessidades dos nossos clientes.

Valores:

Inovação, compromisso com a qualidade, excelência no atendimento, respeito ao cliente e responsabilidade social.

Apêndice 2

- **Há quanto tempo a sua empresa está em operação?**

8 anos.

- **Quem é o público-alvo principal dos seus produtos?**

Supermercados, farmácias e salões de beleza que procuram produtos de qualidade, com bom custo-benefício e que tragam inovação.

- **Quais são os maiores desafios que vocês enfrentam na gestão de produtos?**

Manter o catálogo sempre atualizado. Hoje a gente depende de agência pra isso, o que acaba gerando custos mais altos e, muitas vezes, demora no processo.

- **Vocês utilizam algum sistema digital para gerenciar produtos atualmente?**

Não, hoje não usamos nenhum sistema. Tudo é feito por meio da agência, então não temos autonomia pra atualizar quando precisa.

- **O que seria ideal para melhorar esse processo de gerenciamento?**

Ter um sistema próprio, onde a gente consiga cadastrar, atualizar e gerenciar os produtos de forma simples e rápida, sem depender de terceiros.

- **Existe alguma funcionalidade que você considera essencial no sistema?**

Sim, seria muito bom se tivesse uma função pra gerar *folders* digitais, pra gente conseguir enviar pras lojas, distribuidores e clientes direto por WhatsApp ou e-mail.

- **Quem na empresa seria o responsável por manter o catálogo atualizado?**

A equipe de marketing. Eles já conhecem bem os produtos, então seriam eles que iriam cuidar dessa parte.

- **O catálogo seria para uso interno ou os clientes também teriam acesso?**

Os dois. A gente quer que o cliente tenha acesso pra consultar os produtos, olhar os detalhes, favoritar e, se quiser, já chamar no WhatsApp pra falar com a gente.

- **É importante que o catálogo funcione em celular e tablet?**

Com certeza. Hoje todo mundo usa mais o celular, então tem que rodar bem tanto no desktop quanto no celular e tablet.

- **Você tem alguma preferência sobre o design do catálogo?**

Gosto de uma pegada mais moderna, limpa e bem fácil de navegar. Nada muito poluído, que confunda quem tá usando.

- **Quais informações precisam aparecer em cada produto?**

Nome, descrição bem feita, ingredientes, benefícios, fotos de boa qualidade e, se possível, o preço sugerido também.

- **Como os produtos devem ser organizados no catálogo?**

Por categorias. Tipo linha capilar, corporal, kits... E se tiver filtros pra facilitar a busca, melhor ainda.

- **Existe algum padrão ou regra para as fotos dos produtos?**

Sim, as fotos precisam ser em alta qualidade e seguir nosso padrão visual, pra ficar tudo alinhado com a identidade da empresa.

- **Você gostaria de ter uma opção de contato direto no catálogo, como WhatsApp?**

Com certeza. Ter um botão pro cliente chamar direto no WhatsApp seria perfeito, fica muito mais prático.

- **O que seria, na sua opinião, a experiência ideal para o cliente que vai acessar esse catálogo?**

Que ele entre, ache rápido o que procura, consiga ver todos os detalhes do produto e, se quiser, já tenha como falar com a gente.

- **Você gostaria de acompanhar algum tipo de relatório sobre o uso do catálogo?**

Seria ótimo. Coisa do tipo: quais produtos são mais vistos, quanto tempo o cliente fica no site... essas informações ajudam muito no marketing.

- **Com que frequência vocês costumam atualizar o catálogo?**

Mais ou menos duas vezes por mês. Às vezes mais, dependendo se tem lançamento ou alguma mudança.

- **Qual é a expectativa de vocês com esse catálogo digital?**

Que ele agilize muito nosso trabalho, reduza custo com agência e ainda melhore a experiência dos nossos clientes, com um catálogo sempre atualizado.

- **Vocês gostariam de incluir banners de promoções ou lançamentos?**

Sim, acho uma boa. Sempre tem uma promoção rolando ou um produto novo chegando, então ter um espaço pra isso é bem interessante.

- **Sobre o treinamento para a equipe usar o sistema, como você imagina?**

Sendo fácil de mexer, acho que um manual simples e talvez um vídeo curto explicando já tá ótimo.

- **Existe alguma necessidade de cadastrar ou destacar produtos que estão em promoção ou são lançamentos?**

Sim, seria ótimo. Assim que chega novidade ou quando a gente faz alguma promoção, seria bom conseguir destacar isso no catálogo.

- **Vocês pensam em futuramente integrar esse catálogo com algum sistema de vendas *online*, como e-commerce?**

Olha, por enquanto a ideia é só catálogo mesmo. Mas não descarto, lá na frente, quem sabe, transformar isso num e-commerce.

- **Existe alguma preocupação com segurança, como controle de quem pode alterar os dados do catálogo?**

Com certeza. A gente quer que só o pessoal autorizado, principalmente o marketing, consiga mexer. Cliente só consulta, não altera nada.