

INSTITUTO FEDERAL DO PARANÁ

SOLUÇÃO DIGITAL PARA FLORICULTURA: VITRINE DIGITAL E
GERENCIAMENTO DINÂMICO DE ENTREGAS

UNIÃO DA VITÓRIA, 2025

INSTITUTO FEDERAL DO PARANÁ

LUAN EDUARDO TREUKE

SOLUÇÃO DIGITAL PARA FLORICULTURA: VITRINE DIGITAL E
GERENCIAMENTO DINÂMICO DE ENTREGAS

Trabalho de Conclusão de Curso apresentado ao
Curso Tecnólogo em Análise e Desenvolvimento
de Sistemas, do Instituto Federal do Paraná –
Campus União da Vitória, como requisito parcial
de avaliação.

Orientador: Prof. Fabricio Malta de Oliveira

UNIÃO DA VITÓRIA

2025

RESUMO

Este Trabalho de Conclusão de Curso apresenta o desenvolvimento de uma solução digital voltada à informatização dos processos de vendas e entregas de uma floricultura de pequeno porte localizada em União da Vitória – PR. Trata-se de um estudo de caso com aplicação prática, cujo objetivo foi projetar e implementar uma plataforma web integrada composta por uma vitrine digital de produtos e um painel de controle logístico. A metodologia adotada foi o modelo incremental de desenvolvimento, com ênfase na validação contínua por meio de testes de usabilidade e análise de requisitos técnicos. A solução será desenvolvida com tecnologias modernas de código aberto, utilizando Next.js, NestJS, Tailwind CSS e TypeORM, estruturadas sob a arquitetura cliente-servidor e com separação em camadas. Os resultados esperados são a viabilidade de centralizar a gestão de pedidos e entregas em um único sistema, com redução de falhas manuais, melhor organização da logística e maior controle sobre o catálogo de produtos. A conclusão aponta que a digitalização, ainda que aplicada em um contexto local e específico, pode ser replicada em negócios com demandas operacionais semelhantes, respeitando os limites técnicos e estruturais definidos no escopo do projeto.

ABSTRACT

This undergraduate thesis presents the development of a digital solution aimed at computerizing the sales and delivery processes of a small flower shop located in União da Vitória, Brazil. It is a case study with practical application, whose objective was to design and implement an integrated web platform composed of a digital product catalog and a logistics management panel. The methodology adopted was the incremental development model, with emphasis on continuous validation through usability testing and technical requirements analysis. The solution was developed using modern open-source technologies, including Next.js, NestJS, Tailwind CSS, and TypeORM, structured under a client-server architecture with layered separation. The results demonstrated the feasibility of centralizing order and delivery management in a single system, reducing manual errors, improving logistical organization, and increasing control over the product catalog. The conclusion indicates that digitalization, even when applied in a local and specific context, can be replicated in businesses with similar operational demands, respecting the technical and structural limitations defined in the project scope.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – TELA INICIAL (DESTEFANI)	23
FIGURA 2 – TELA DE VISUALIZAÇÃO DE CATÁLOGO (SICON).....	24
FIGURA 3 – DIAGRAMA DE CASOS DE USO (SICON).....	25
FIGURA 4 – TELA DE VISUALIZAÇÃO DE VENDAS (NUVEMSHOP).....	26
FIGURA 5 – TELA DE VISUALIZAÇÃO DE LAYOUTS (NUVEMSHOP).....	27
FIGURA 6 – MODELO INCREMENTAL DE DESENVOLVIMENTO.....	30
FIGURA 7 – DIAGRAMA DE CASOS DE USO.....	37
FIGURA 8 – DIAGRAMA DE CLASSES.....	42
FIGURA 9 – DIAGRAMA ENTIDADE RELACIONAMENTO.....	43
FIGURA 10 – VITRINE DIGITAL.....	46
FIGURA 11 – VITRINE DIGITAL (VERSÃO MOBILE).....	47
FIGURA 12 – DETALHES DO PRODUTO.....	48
FIGURA 13 – TELA DE LOGIN.....	49
FIGURA 14 – GERENCIAMENTO DE ENTREGAS.....	50
FIGURA 15 – ADIÇÃO DE ENCOMENDA.....	51
FIGURA 16 – CONFIRMAÇÕES DE AÇÃO.....	52
FIGURA 17 – TELA DE GERENCIAMENTO DE CATÁLOGO.....	52
FIGURA 18 – TELA DE CADASTRO DE NOVO PRODUTO.....	53
FIGURA 19 – TELA DE DETALHE E AÇÕES DO PRODUTO.....	54
FIGURA 20 – TELA DE EDIÇÃO DE PRODUTO.....	55
FIGURA 21 – TELA DE GERENCIAMENTO DE IMAGENS DO PRODUTO.....	56
FIGURA 22 – TELA DE GERENCIAMENTO DE CATEGORIAS.....	57
FIGURA 23 – TELA DE ADICIONAR CATEGORIA.....	58
FIGURA 24 – TELA DE GERAÇÃO DE RELATÓRIOS.....	59
FIGURA 25 – TECNOLOGIAS UTILIZADAS.....	60

LISTA DE QUADROS

QUADRO 1 – QUADRO COMPARATIVO DE FERRAMENTAS SIMILARES.....	27
QUADRO 2 – REQUISITOS FUNCIONAIS.....	34
QUADRO 3 – REQUISITOS NÃO FUNCIONAIS.....	35
QUADRO 4 – CRONOGRAMA DE EXECUÇÃO.....	62

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Problematização	13
2 JUSTIFICATIVA	14
3 OBJETIVOS	15
1.1 Geral	15
1.2 Específicos	15
4 REVISÃO BIBLIOGRÁFICA	16
4.1 Arquitetura de aplicações web modernas	18
4.2 Trabalhos relacionados	21
4.2.1 Fontes literárias	21
4.2.2 Ferramentas similares	22
4.2.2.1 Considerações	28
5 METODOLOGIA	28
5.1 Considerações iniciais	28
5.2 Análise de mercado	30
5.3 Planejamento de Recursos, Custos e Prazos	30
5.4 Análise de requisitos	31
6 DESENVOLVIMENTO	32
6.1 Requisitos do sistema	32
6.1.1 Requisitos de Ambiente (Hardware e Software)	32
6.1.1.1 Requisitos para a Hospedagem	32
6.1.1.2 Requisitos para o usuário	33
6.2 Requisitos funcionais	33
6.3 Requisitos não funcionais	34
6.3.1 Métodos de validação	35
6.4 Modelagem do Sistema	36
6.4.1 Diagrama de Casos de Uso	36
6.4.1.1 Detalhamento dos casos de uso	37
6.4.2 Diagrama de Classes	41
6.4.3 Diagrama Entidade-Relacionamento (DER)	42
6.4.3.1 Detalhamento do modelo de dados e regras de negócio	43
6.5 Prototipagem	45
6.6 Tecnologias Utilizadas	59
6.6.1 Front-end	60
6.6.2 Back-end	61
6.6.3 Banco de Dados	61
6.6.4 Ferramentas de Apoio e Versionamento	62
7 CRONOGRAMA DE EXECUÇÃO	62
8 REFERÊNCIAS	64

1 INTRODUÇÃO

Ao longo da história, a gestão de entregas e o atendimento ao cliente têm sido fundamentais para o sucesso empresarial, especialmente em setores como o da floricultura, onde a agilidade na logística e a precisão nas informações garantem a satisfação do cliente (Mundo Corporativo, 2025). O mercado brasileiro de flores e plantas ornamentais movimentou R\$10,9 bilhões em 2021, representando um crescimento de 15% em relação ao ano anterior (ILIS, 2022). As vendas online e em supermercados contribuíram para esse resultado, especialmente no segmento de flores de corte. A expectativa, segundo Possato Teixeira do Instituto Brasileiro de Floricultura (Ibraflor), é de um crescimento entre 6% e 8% no setor em 2025 (CNA Brasil, 2025).

Justamente neste cenário de expansão, os desafios logísticos tornam-se ainda mais evidentes. Estudos sobre logística em redes varejistas de flores mostram que ajustes como a escolha de fretes especializados, roteirização de entregas e compras locais permitiram reduzir em cerca de 80% o tempo total entre o pedido e a entrega (*lead time*), contribuindo para maior flexibilidade e capacidade de resposta à demanda sazonal (Mello et al., 2016). Embora essas práticas envolvam aspectos externos ao escopo deste trabalho, elas reforçam a importância de se investir na organização e no controle das etapas logísticas.

Entretanto, empresas menores desse segmento ainda enfrentam dificuldades operacionais, como catálogos desatualizados, atendimento demorado e sistemas manuais de gestão de entregas. Em floriculturas de menor porte, como a analisada neste trabalho, o processo de vendas online e logística ainda é baseado em conversas pelo WhatsApp e registros físicos improvisados. Isso resulta em atrasos, erros de comunicação e falhas na entrega, especialmente em datas comemorativas com alta demanda.

1.1 Problemática

A concepção deste projeto tem como base a "Floricultura Quatro Estações", um negócio de pequeno porte localizado em União da Vitória, Paraná, empresa cuja eficiência é limitada por desafios operacionais diários. Através da observação prática, constatou-se que ela não possui um sistema informatizado, realizando toda

a sua gestão de vendas e logística de forma manual, com o auxílio de ferramentas informais como anotações em papel e o aplicativo WhatsApp.

Este método de trabalho resulta em uma série de problemas que afetam a eficiência e a imagem profissional do negócio:

- Perda de informações: Detalhes de pedidos, como mensagens de cartão ou endereços, podem ser facilmente perdidos ou transcritos incorretamente.
- Desorganização logística: A ausência de um painel de controle centralizado dificulta o acompanhamento do status das entregas, levando a atrasos e falhas na comunicação com os clientes e entregadores.
- Catálogo desatualizado: A dificuldade em manter uma vitrine de produtos consistente e atualizada compromete a experiência de compra do cliente final.

Outro problema que surge como consequência da falta de um sistema web é o gargalo operacional durante períodos de alta demanda, como o dia das mães e o dia dos namorados. Nesses picos, embora o número de funcionários seja temporariamente ampliado, a capacidade de registrar encomendas permanece restrita a apenas dois aparelhos com acesso ao WhatsApp da empresa. A inviabilidade de instalar o aplicativo de negócios nos dispositivos de múltiplos atendentes temporários cria um ponto de estrangulamento no atendimento, resultando em esperas, potencial perda de vendas e um aumento no risco de erros de anotação.

Diante disso, o problema central que este trabalho se propõe a resolver é como o desenvolvimento de uma plataforma web integrada pode centralizar a gestão de uma pequena floricultura, a fim de solucionar as ineficiências do processo manual e melhorar a exposição online da empresa.

2 JUSTIFICATIVA

Conforme dito anteriormente, o mercado de flores e plantas ornamentais no Brasil apresenta crescimento constante, impulsionado por datas comemorativas, mudanças no comportamento de consumo e o fortalecimento das vendas online (ILIS, 2022). E como consequência, torna-se urgente a modernização de processos nas pequenas floriculturas.

A apresentação dos produtos impacta diretamente na experiência do

consumidor, catálogos desatualizados, com imagens de baixa qualidade, dificultam a decisão de compra e comprometem a imagem da empresa. Segundo Kotler, Kartajaya e Setiawan (2017), na era digital, o comportamento do consumidor é influenciado por experiências visuais, interfaces amigáveis e conectividade com a marca, tornando plataformas intuitivas e visualmente atrativas um diferencial competitivo.

Diante desse contexto, justifica-se o desenvolvimento de uma solução digital voltada à realidade das pequenas floriculturas, unindo um catálogo virtual funcional a um sistema de gerenciamento dinâmico de entregas. Essa solução permitirá ganhos operacionais, como o registro padronizado de pedidos, a melhor organização do fluxo de entregas e a atualização facilitada dos produtos. Ao propor uma ferramenta acessível e personalizável, este trabalho busca atender não apenas a uma necessidade técnica, mas também contribuir para a profissionalização do setor, proporcionando eficiência, organização e melhoria no atendimento ao cliente.

3 OBJETIVOS

1.1 Geral

Projetar e implementar uma solução web integrada que atue em duas frentes de uma floricultura: vitrine digital e um painel administrativo.

1.2 Específicos

- Realizar uma revisão bibliográfica sobre sistemas de e-commerce e gestão logística para pequenas empresas.
- Levantar e documentar os requisitos funcionais e não funcionais da plataforma web.
- Modelar a arquitetura do sistema, incluindo seus casos de uso, diagramas de classe e o modelo de dados.
- Criar um protótipo de alta fidelidade da aplicação para validar os fluxos de navegação e a experiência do usuário.
- Implementar uma plataforma web, desenvolvendo as funcionalidades de gestão de catálogo e controle de entregas.

- Validar a plataforma desenvolvida por meio da verificação dos requisitos e da realização de testes de usabilidade com o usuário-alvo.
- Realizar o refinamento final do projeto e a sua implantação em um ambiente de produção na nuvem.

4 REVISÃO BIBLIOGRÁFICA

A eficiência na distribuição é fator determinante em um cenário de competição, conforme indicado por estudo recente sobre e-commerce de flores, agilidade na entrega é prioridade para 56 % dos consumidores, com 64 % influenciados pelo prazo estimado. No setor de flores, perecibilidade e ocasiões especiais intensificam essa pressão. O uso de hubs logísticos, rastreamento em tempo real e automação otimizam prazos e preservam a qualidade dos produtos (Capterra; Ibraflor; Clóvis Souza, 2025).

No entanto, observam-se ainda floriculturas que utilizam processos manuais e sistemas básicos para o controle de vendas, o que compromete a criação de bases de dados e a integração com fornecedores. Esse uso limitado de tecnologia gera barreiras na adoção de ferramentas digitais que poderiam melhorar a gestão e a eficiência operacional. Um estudo recente de Mantilla (2025), voltado ao setor florícola em países emergentes, destaca que “a implantação de tecnologias da Indústria 4.0¹ encontra entraves, incluindo custos elevados e falta de expertise técnica, limitando o uso de ferramentas inteligentes no controle operacional”. Além disso, a logística da última milha² é identificada como um grande desafio nas vendas via internet, impactando a etapa decisiva da entrega, que deve ser cumprida com competência (Menin, 2019). Nesse contexto, cabe analisar de que forma a adoção de um catálogo digital aliado a um sistema estruturado de entregas pode contribuir para a superação dos desafios logísticos e para a elevação da eficiência operacional e da satisfação do cliente.

Primeiramente, é necessário analisar o papel do catálogo digital. Segundo

¹ termo que se refere à quarta revolução industrial, caracterizada pela integração de tecnologias como inteligência artificial, internet das coisas e automação avançada nos processos produtivos.

² etapa final da entrega de um produto ao consumidor, geralmente do centro de distribuição até o destino final, sendo considerada a mais complexa e custosa da logística.

(Liu et al., 2022), ele é uma ferramenta essencial para apresentar os produtos de forma organizada, clara e acessível, permitindo que o usuário compreenda rapidamente as opções disponíveis e tome decisões com base em informações visuais e descritivas. a estruturação de um catálogo digital melhora a navegação do usuário e facilita a localização de conteúdos relevantes, contribuindo para uma experiência mais objetiva.

Para que essa ferramenta de gestão se alinhe à experiência do cliente, o foco recai sobre a Experiência do Usuário (UX) e a usabilidade. Segundo Sharp (2013), a usabilidade se refere à qualidade da experiência ao interagir com um sistema ou site e é fundamental para o sucesso no comércio eletrônico. Portanto, um catálogo online deve ser intuitivo e fácil de usar, permitindo que os usuários encontrem a informação desejada, algo que a falta de usabilidade em muitos sites impede, levando à perda de visitantes (Silva, 2023).

Martins e Almeida, (2022), destacam a importância da Arquitetura da Informação, definida por Cruz e Bicudo (2010) como a competência responsável pela disposição e organização objetiva das informações. Segundo Kim e Lee (2021), o uso de um Sistema de Gerenciamento de Conteúdo (CMS) permite administrar todas as etapas, da criação à publicação de conteúdo, separando o gerenciamento da sua apresentação. Em um catálogo digital, isso se traduz na facilidade de descrever itens (catalogação), rotular conteúdo com palavras-chave (indexação ou tags), e permitir a busca e recuperação de informações (Kim; Lee, 2021). Ferramentas como barras de pesquisa e nuvens de tags podem melhorar significativamente a navegabilidade. Uma boa prática sugere lançar o catálogo quando estiver cerca de 80% pronto, fazendo ajustes baseados no feedback do cliente (Nielsen; Budiu, 2021). Ferreira, Chauvel e Silveira (2006) também enfatizam a importância dos requisitos não funcionais de usabilidade, que têm forte impacto na qualidade de sistemas de informação com interação direta com clientes.

Em relação à gestão de entregas, ela aborda um dos gargalos operacionais do setor, especialmente considerando o desafio da última milha (Menin, 2019). A falta de controle sobre as operações logísticas, evidenciada pela baixa utilização de tecnologia para acompanhar fluxos de compra e venda, leva à desorganização e perda de informações.

Um sistema de gerenciamento de entregas pode mitigar esses problemas

logísticos ao otimizar processos e substituir métodos manuais, que frequentemente causam perda de informações e ineficiência (Menin, 2019).

Finalmente, a proposta de uma plataforma digital integrada está alinhada com a transformação digital e os princípios do Marketing 4.0. Kotler, Kartajaya e Setiawan (2017), assim como Lima e Queiroz (2018), observam que o avanço tecnológico influencia diretamente o comportamento do consumidor, que utiliza cada vez mais meios digitais no processo de compra. Em um mercado onde a experiência digital é importante, plataformas com boa usabilidade e interfaces intuitivas tornam-se um diferencial competitivo (Loureiro; Barbosa, 2022). Conhecer o consumidor online e suas interações com a plataforma torna-se relevante neste contexto. A otimização da experiência do usuário pode ser medida por métricas de atendimento ao cliente, como CSAT³, NPS⁴, CES⁵, tempo de resposta e também por uma métrica de autoatendimento, como pesquisas realizadas pelos próprios usuários (Métricas de Atendimento, 2019).

4.1 Arquitetura de aplicações web modernas

Além dos aspectos de negócio e usabilidade, a construção de uma solução digital depende de uma arquitetura de software bem definida. A arquitetura estabelece a estrutura do sistema, ditando como seus componentes se organizam e se comunicam para atender aos requisitos funcionais e não funcionais do projeto (SOMMERVILLE, 2019). Para aplicações web contemporâneas, dois padrões arquitetônicos se destacam por sua eficácia, escalabilidade e aderência aos princípios da Engenharia de Software: a Arquitetura em Camadas e o modelo Cliente-Servidor.

A Arquitetura em Camadas (Layered Architecture) é um estilo arquitetural amplamente adotado em sistemas de informação, pois organiza a aplicação em camadas horizontais, cada uma responsável por uma função específica. De acordo com Fowler (2002), uma estrutura comum envolve três camadas principais: a camada de apresentação (interface com o usuário), a camada de lógica de negócio

³ métrica que avalia o nível de satisfação do cliente com um serviço ou produto, geralmente por meio de uma pergunta direta após a interação.

⁴ indicador que mede a probabilidade de um cliente recomendar a empresa, classificando-a de 0 a 10, e identificando promotores, neutros e detratores.

⁵ métrica que avalia o esforço necessário por parte do cliente para resolver um problema ou concluir uma ação com a empresa.

(onde são processadas as regras de negócio) e a camada de acesso a dados (responsável pela comunicação com o banco de dados). Essa separação favorece o princípio da separação de interesses (separation of concerns), promovendo modularidade, facilidade de manutenção e testabilidade (FOWLER, 2002; SOMMERVILLE, 2019).

Complementarmente, o modelo cliente-servidor define uma arquitetura distribuída, em que o cliente solicita recursos e o servidor os fornece. Segundo Silva e Almeida (2022), essa estrutura fundamenta a maioria das aplicações web atuais. O servidor concentra a lógica de negócio e o acesso aos dados, enquanto o cliente, geralmente uma aplicação web ou mobile, gerencia a interação com o usuário. A comunicação entre essas camadas ocorre via APIs, que expõem os serviços do sistema de forma padronizada e reutilizável (Silva; Almeida, 2022).

Entre os padrões de API mais comuns destaca-se o modelo RESTful (Representational State Transfer), definido por Richardson e Ruby (2020). Nesse modelo, os recursos do sistema são manipulados por meio dos verbos do protocolo HTTP (GET, POST, PUT, DELETE), esses verbos são comandos utilizados para definir a ação que o cliente deseja realizar sobre um recurso em uma aplicação web. No modelo REST, os principais são: GET, para obter dados; POST, para criar novos registros; PUT, para atualizar um recurso existente; e DELETE, para removê-lo. Esses métodos padronizam a comunicação entre cliente e servidor e orientam o comportamento das requisições em sistemas web. promovendo simplicidade, escalabilidade e interoperabilidade. O uso de APIs REST é uma prática usada na criação de sistemas desacoplados, permitindo que diferentes clientes (como uma aplicação web) consumam os mesmos serviços back-end (Richardson; Ruby, 2020; Foreach Partners, 2025).

A integração desses dois modelos, arquitetura em camadas e cliente-servidor, representa uma prática recomendada no desenvolvimento de soluções web modernas. Ela possibilita a criação de sistemas escaláveis, organizados e sustentáveis, fundamentais para projetos que demandam evolução contínua e manutenção ao longo do tempo, como o proposto neste trabalho (SOMMERVILLE, 2019).

Para que o sistema esteja disponível continuamente ao usuário final, é necessário utilizar um serviço de hospedagem, que consiste na alocação dos

arquivos e dados da aplicação em servidores conectados à internet, permitindo o acesso remoto e ininterrupto ao website (IBM, 2023). Conforme afirma Leonardi (2023), a hospedagem web viabiliza a publicação de aplicações online, garantindo desempenho, segurança e tempo de atividade compatíveis com a experiência esperada pelo usuário.

A arquitetura de aplicações web modernas normalmente adota a separação entre as responsabilidades do front-end e do back-end como uma prática consolidada, pois isso favorece a modularidade e facilita a manutenção e evolução dos sistemas (SOMMERVILLE, 2019). O front-end é responsável pela interface gráfica e pela experiência de interação do usuário, enquanto o back-end gerencia a lógica de negócio e o acesso aos dados da aplicação (FIELDING, 2000).

Entre os frameworks modernos utilizados para construção de interfaces web, destaca-se o Next.js, ele é baseado no React.js e se destaca por oferecer renderização no lado do servidor (Server-Side Rendering – SSR), geração de páginas estáticas (Static Site Generation – SSG), divisão automática de código e roteamento orientado a arquivos, o que melhora o desempenho e a indexação da aplicação em motores de busca (NEXT.JS, 2025). O React, por sua vez, é uma biblioteca declarativa mantida pela Meta (antigo Facebook), cuja estrutura baseada em componentes reutilizáveis permite o desenvolvimento de interfaces complexas de forma eficiente e organizada (REACT, 2025).

A estilização de interfaces em projetos modernos pode ser realizada com Tailwind CSS, um framework CSS utilitário adotado por sua produtividade e consistência visual (TAILWIND CSS, 2025). Ele utiliza classes utilitárias diretamente na marcação HTML, o que reduz a necessidade de escrever CSS personalizado e aumenta a consistência visual do projeto (TAILWIND CSS, 2025).

No contexto de aplicações back-end escaláveis, o NestJS destaca-se como opção, baseada em TypeScript, com suporte à modularidade, injeção de dependências e organização em camadas (NESTJS, 2025). O NestJS se fundamenta em princípios de programação orientada a objetos e funcional, e organiza o código em módulos, controladores e serviços, promovendo coesão e baixo acoplamento (NESTJS, 2025). Segundo a documentação oficial, o NestJS emprega o padrão de injeção de dependência, o que favorece a reutilização de código e a escalabilidade da aplicação (NESTJS, 2025).

A camada de persistência de dados em aplicações modernas pode ser gerenciada por ferramentas como o TypeORM, um ORM⁶ que permite o mapeamento entre objetos e tabelas relacionais, facilitando a manipulação e a consistência dos dados (TYPEORM, 2024). Ele permite o uso de classes como entidades, realizando o mapeamento automático para as tabelas do banco de dados e facilitando a manipulação dos registros sem a necessidade de comandos SQL diretamente no código (TYPEORM, 2024).

Para garantir a disponibilidade contínua do sistema em ambiente de produção, muitas ferramentas utilizam o PM2, um gerenciador de processos para aplicações Node.js. De acordo com a documentação da Keymetrics (2025), o PM2 mantém a aplicação em execução mesmo após falhas inesperadas e oferece suporte ao modo cluster, que permite utilizar múltiplos núcleos da CPU, otimizando o desempenho e a estabilidade do sistema.

4.2 Trabalhos relacionados

Este trabalho propõe o desenvolvimento de uma solução digital para floriculturas de pequeno porte, com foco na criação de um catálogo virtual e sistema de gerenciamento de entregas. Nesse contexto, é importante destacar plataformas, ferramentas e trabalhos acadêmicos que abordam temas semelhantes, seja no âmbito da informatização de floriculturas, da organização logística em pequenos negócios, ou da experiência do usuário em vitrines digitais. A seguir, apresentam-se estudos e sistemas que influenciam e justificam a proposta deste projeto.

4.2.1 Fontes literárias

A literatura aponta que pequenos negócios, como floriculturas, ainda utilizam métodos informais de atendimento e registro de vendas, o que compromete a eficiência e a escalabilidade das operações (Menin, 2019). A ausência de um catálogo digital e de um sistema logístico estruturado compromete a experiência do cliente, a confiabilidade das entregas e a imagem da marca.

O estudo de Mantilla (2025) evidencia que melhorias nos processos logísticos podem reduzir significativamente o tempo de entrega (lead time), resultando em ganhos operacionais e maior satisfação do cliente. Já Santos e Lima

⁶ técnica que permite mapear objetos da programação orientada a objetos para tabelas de bancos de dados relacionais, facilitando o acesso e a manipulação dos dados sem uso direto de SQL.

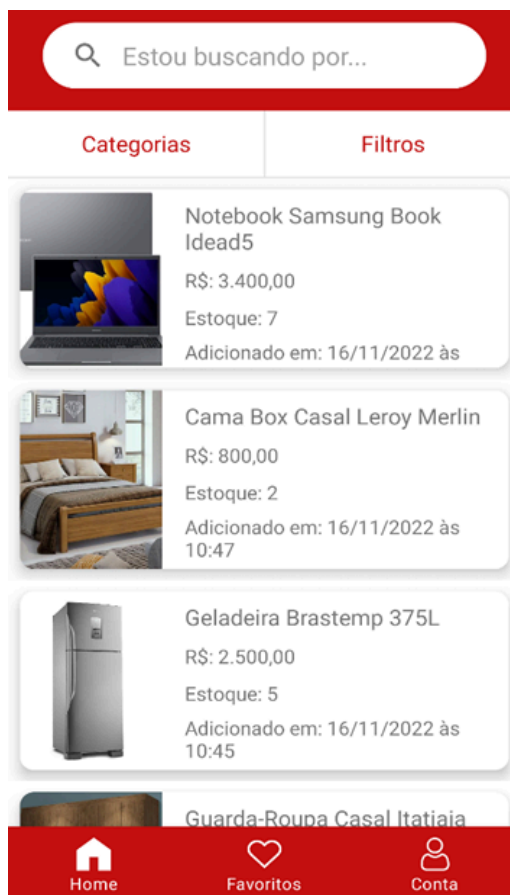
(2022), ressaltam que a usabilidade em sistemas digitais voltados ao comércio eletrônico é essencial para o sucesso da interação com o consumidor. Sharp (2013) também aponta que sistemas com baixa usabilidade afastam usuários e geram experiências negativas.

4.2.2 Ferramentas similares

Com base em pesquisa em trabalhos acadêmicos e ferramentas profissionais, identificaram-se algumas utilizadas por floriculturas e comércios semelhantes que combinam catálogo digital e funcionalidades logísticas:

- Destefani (Supeleto, 2022): Trabalho acadêmico em que foi desenvolvido um aplicativo mobile para uma loja de móveis e eletrodomésticos no Espírito Santo, com foco na digitalização de vendas. A solução inclui catálogo de produtos, cadastro de clientes, filtros, favoritos e integração com WhatsApp para finalização dos pedidos. Utiliza Android Studio com Java e Firebase Realtime Database. A interface foi projetada no Figma, priorizando usabilidade. Apesar de funcional, o sistema não possui controle interno de entregas nem versão web, sendo voltado exclusivamente a dispositivos Android. A figura 1 apresenta o catálogo da ferramenta Destefani:

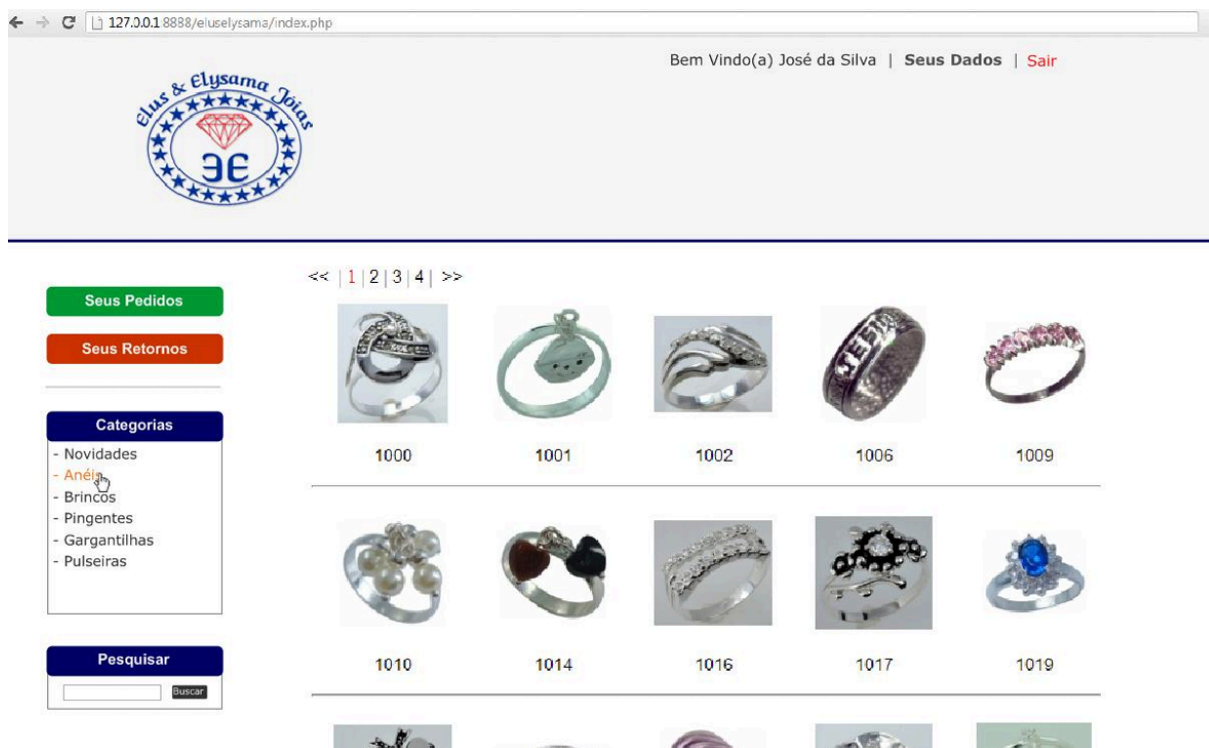
FIGURA 1 – TELA INICIAL (DESTEFANI)



FONTE: Supeleto (2022, p. 31).

- SICON (Fernandes, 2012): Sistema web desenvolvido para a empresa Elus & Elysama, especializada em semi jóias, com o objetivo de substituir catálogos físicos e desorganizados por uma plataforma digital de acesso remoto. O sistema permite que os clientes acessem o catálogo online, realizem pedidos e registrem devoluções de produtos. Também oferece funcionalidades de cadastro de usuários, categorias, produtos e geração de relatórios. O projeto foi implementado com foco em web (PHP, MySQL, HTML, CSS, jQuery), com modelagem UML e banco de dados relacional. Apesar de abrangente, a solução não possui acompanhamento em tempo real de entregas, interface agradável aos clientes, nem integração com canais externos de comunicação (como WhatsApp). A figura 2 apresenta o catálogo da ferramenta Sicon:

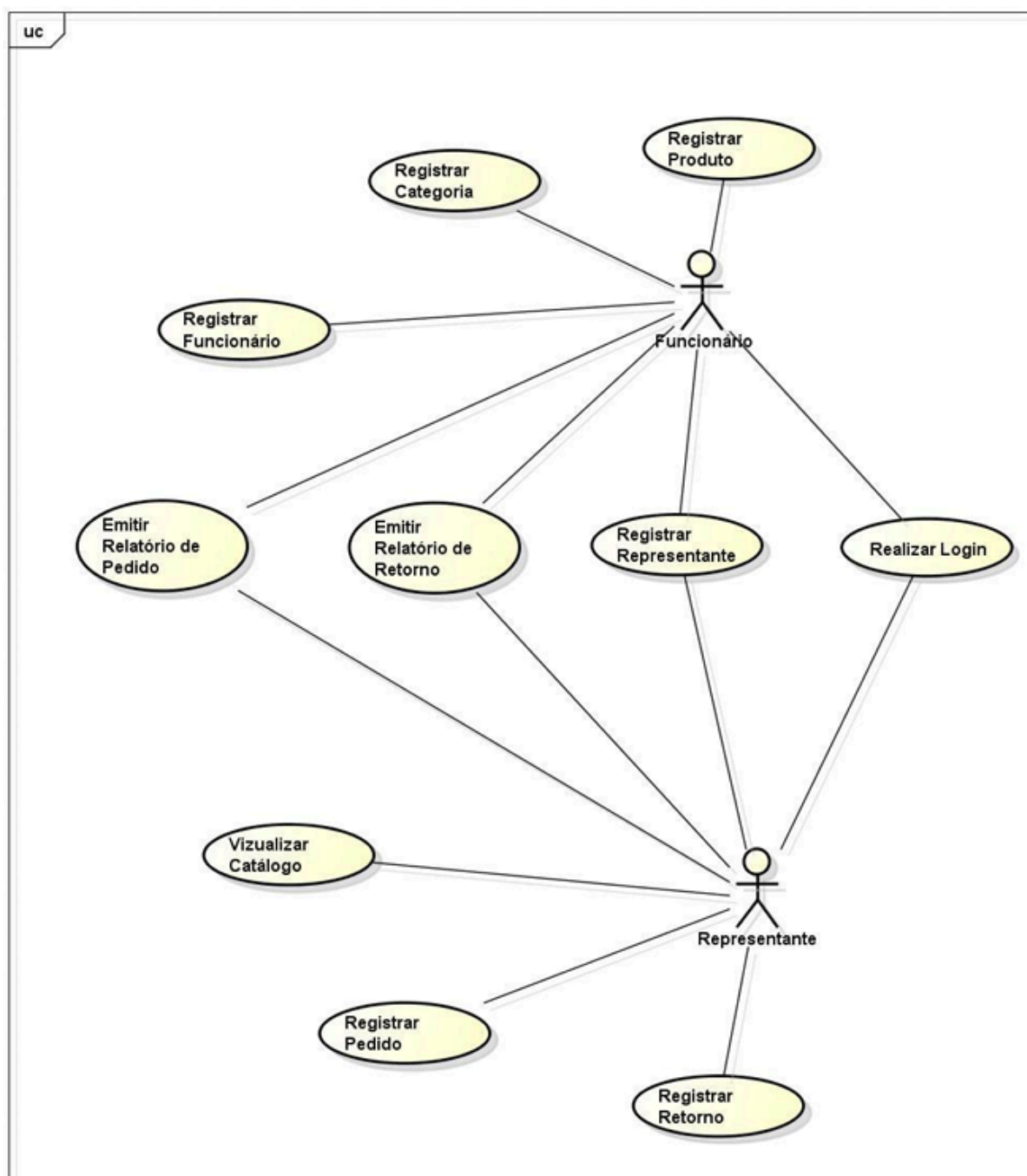
FIGURA 2 – TELA DE VISUALIZAÇÃO DE CATÁLOGO (SICON)



FONTE: Fernandes (2012, p. 79).

Na figura 3 é apresentado o diagrama de casos de uso da ferramenta SICON (Fernandes, 2012), a fim de apresentar suas funcionalidades. Se comparado ao diagrama de casos de uso da solução proposta neste trabalho, é perceptível a diferença entre as soluções.

FIGURA 3 – DIAGRAMA DE CASOS DE USO (SICON)

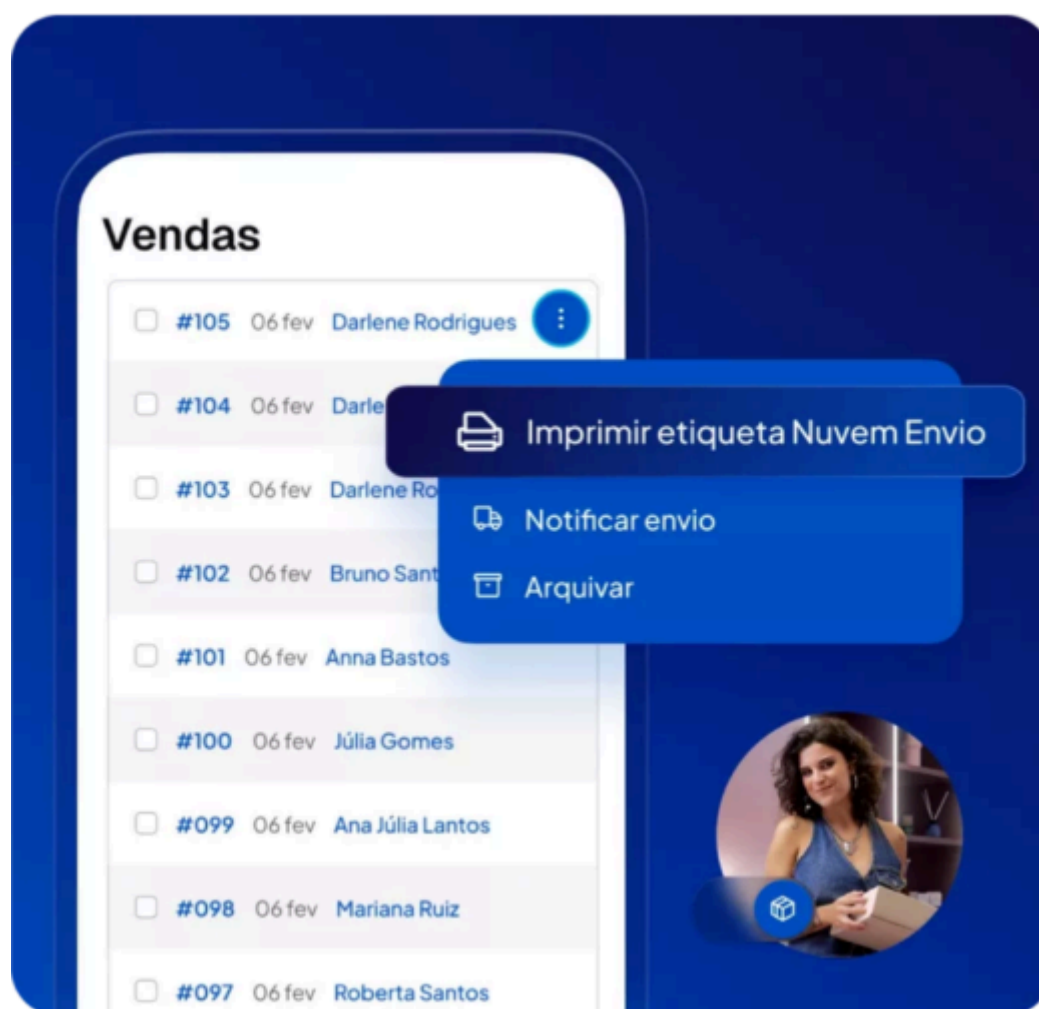


FONTE: Fernandes (2012, p. 79).

- Nuvemshop: Diferentemente dos projetos acadêmicos anteriores, a Nuvemshop é uma plataforma comercial de e-commerce consolidada no mercado latino-americano, que serve como um exemplo de solução generalista para o varejo. A ferramenta permite que lojistas de diversos segmentos criem uma loja virtual completa, com funcionalidades robustas

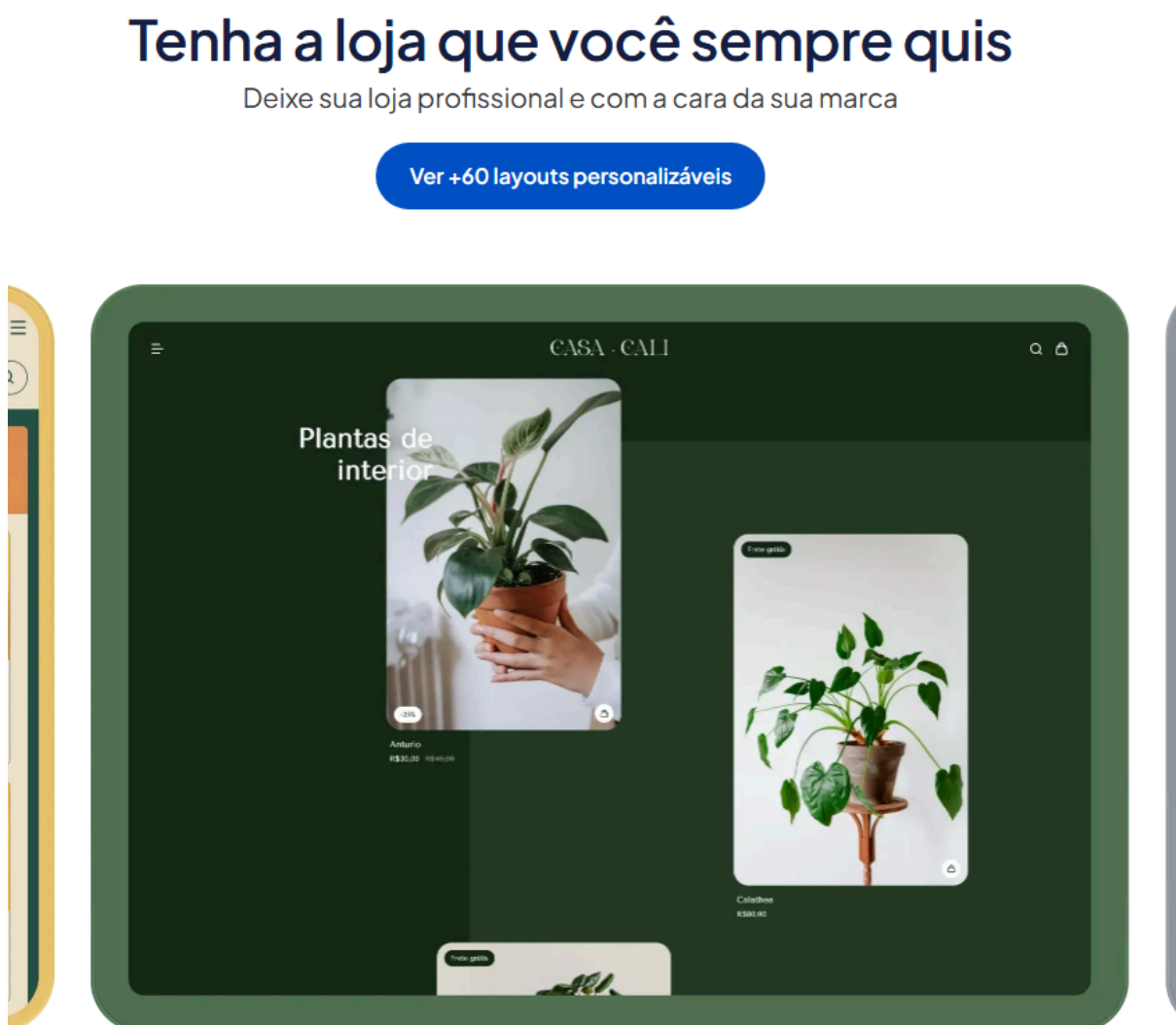
para cadastro de produtos, customização de layout, processamento de pagamentos e, de forma relevante para este trabalho, um sistema de gestão de fretes integrado a transportadoras e Correios. As figuras 4 e 5 apresentam telas importantes da NuvemShop.

FIGURA 4 – TELA DE VISUALIZAÇÃO DE VENDAS (NUVEMSHOP)



FONTE: NuvemShop (2025).

FIGURA 5 – TELA DE VISUALIZAÇÃO DE LAYOUTS (NUVEMSHOP)



FONTE: NuvemShop (2025).

Para sintetizar a análise dos trabalhos relacionados, o quadro 1 compara as principais características de cada solução com a proposta neste trabalho:

QUADRO 1 – QUADRO COMPARATIVO DE FERRAMENTAS SIMILARES

Característica	Solução Proposta	Nuvemshop	Destefani	SICON
Tipo de Solução	Sistema Web Customizado	Plataforma SaaS (Software como Serviço)	Aplicativo Mobile	Sistema Web
Foco Principal	Nicho específico (Floricultura)	E-commerce Generalista	Nicho específico (Loja de Móveis)	Vendas por Catálogo (Jóias)
Plataforma	Web (Desktop e	Web (Desktop e	Android	Web

	Mobile)	Mobile)		
Catálogo Digital	Sim, com gestão simplificada e foco em visualização	Sim, com gestão robusta e integrada	Sim, com foco em visualização	Sim, para acesso remoto de clientes
Gestão de Entregas	Painel para gestão manual e local	Integração com transportadoras/Correios	Não possui	Não possui
Modelo de Custo	Custo de Hospedagem (AWS)	Assinatura Mensal + Taxas	Gratuito (Acadêmico)	Gratuito (Acadêmico)

FONTE: Elaborado pelo autor (2025).

4.2.2.1 Considerações

Destefani e SICON validam a necessidade de digitalizar catálogos para nichos de mercado, mas frequentemente carecem de um componente robusto para a gestão logística das entregas. Por outro lado, plataformas comerciais como a Nuvemshop oferecem soluções de e-commerce completas, porém generalistas, com um modelo de custo recorrente e um foco em logística integrada a grandes transportadoras, o que não atende à realidade de um pequeno negócio que gerencia suas próprias entregas locais.

Observa-se, portanto, uma lacuna para uma solução que combine um catálogo digital com um painel de gerenciamento de encomendas projetado especificamente para o fluxo operacional da floricultura alvo.

5 METODOLOGIA

Este capítulo apresenta a metodologia adotada para o desenvolvimento da solução digital proposta. São descritas as estratégias de levantamento de requisitos, modelagem, desenvolvimento incremental e validação contínua da aplicação.

5.1 Considerações iniciais

A abordagem metodológica adotada neste projeto é o modelo incremental de desenvolvimento, conforme descrito por Ganney e Claridge (2020). Essa estratégia consiste na construção do sistema em versões sucessivas, cada uma adicionando novos recursos e melhorias, permitindo maior controle sobre o progresso do projeto e facilitando ajustes ao longo do processo.

As principais etapas do modelo incremental aplicadas neste trabalho são

descritas a seguir:

a) Levantamento de Requisitos: Nesta etapa, foram identificadas e documentadas as funcionalidades essenciais do sistema com base nas necessidades dos usuários e nos objetivos do projeto. Esse levantamento inicial serve como base para os primeiros incrementos, podendo ser refinado ao longo do desenvolvimento (Ganney; Claridge, 2020).

b) Planejamento dos Incrementos: Após a definição dos requisitos, o projeto foi dividido em versões parciais, cada uma priorizando funcionalidades de maior valor. Essa divisão permite entregas contínuas e facilita o gerenciamento de riscos e mudanças (Ganney; Claridge, 2020).

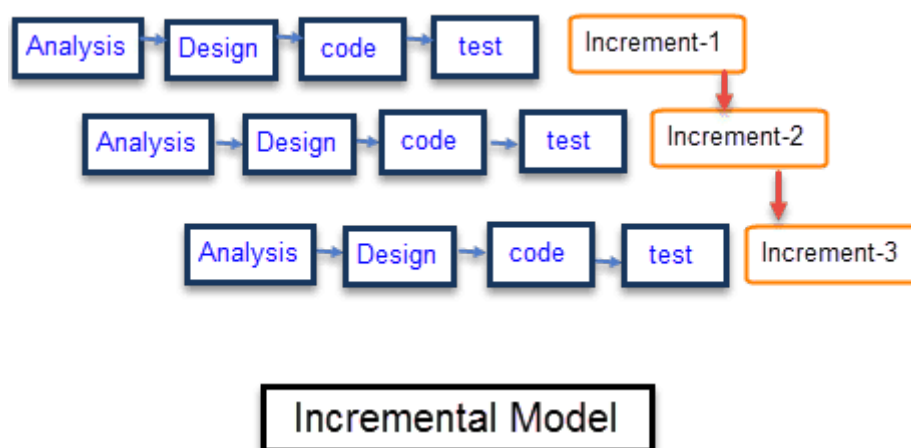
c) Implementação Incremental: Cada incremento envolverá a codificação, testes e integração das funcionalidades planejadas, mantendo o sistema sempre em estado funcional. O desenvolvimento será conduzido de forma cíclica, com feedback constante para possíveis ajustes antes do próximo ciclo.

d) Validação Contínua: Ao final de cada incremento, é previsto testes parciais para garantir a estabilidade do sistema e a conformidade das funcionalidades implementadas com os requisitos definidos. Essa validação contínua reduz a ocorrência de erros acumulados e facilita a manutenção do sistema (Ganney; Claridge, 2020).

e) Revisão e Evolução: A cada ciclo concluído, o sistema passará por revisões técnicas e funcionais, possibilitando a identificação de melhorias, ajustes nos requisitos e redefinição de prioridades para os incrementos seguintes.

A organização dessas atividades no processo de desenvolvimento é representada na Figura 6.

FIGURA 6 – MODELO INCREMENTAL DE DESENVOLVIMENTO



FONTE: TutorialtPoint (2019).

5.2 Análise de mercado

A concepção deste projeto originou-se da identificação de uma lacuna operacional em uma floricultura de pequeno porte. Através da observação prática, constatou-se que a ausência de um sistema informatizado para o controle de catálogo e para a gestão de entregas resulta em um fluxo de trabalho manual, suscetível a erros e problemas na comunicação. Este cenário validou a necessidade de uma solução tecnológica customizada para endereçar estes desafios operacionais.

5.3 Planejamento de Recursos, Custos e Prazos

O desenvolvimento da solução foi planejado com um planejamento de execução de aproximadamente seis meses, conforme detalhado no cronograma deste trabalho. Estrategicamente, optou-se pela utilização exclusiva de um conjunto de tecnologias de código aberto (open source), o que elimina custos com licenciamento de software.

Os custos operacionais recorrentes, relacionados à hospedagem da aplicação, serão gerenciados através de serviços de computação em nuvem da AWS (Amazon Web Services). Durante a fase de desenvolvimento e validação inicial do TCC, projeta-se a utilização do Nível Gratuito (Free Tier) da AWS, que permite hospedar aplicações de baixo tráfego com custo minimizado ou nulo, garantindo a

viabilidade financeira para a implementação e os testes da solução.

5.4 Análise de requisitos

A definição dos requisitos de software serve como base para todas as etapas subsequentes do desenvolvimento. De acordo com Sommerville (2019), “os requisitos para um sistema de software estabelecem o que o sistema deve fazer e definem restrições sobre sua operação e implementação”. Ou seja, a análise de requisitos permite compreender as necessidades dos usuários e transformá-las em funcionalidades específicas. No contexto da floricultura estudada, a coleta de requisitos foi direcionada à resolução de problemas operacionais identificados na rotina da empresa, como desorganização no registro de pedidos, falhas na comunicação e ausência de um catálogo digital para exposição de produtos. Para isso, foram aplicadas duas abordagens principais:

a) Observação prática: A rotina diária de atendimento e gestão no ambiente físico da floricultura foi acompanhada, permitindo identificar processos manuais, como anotações em cadernos, e dificuldades no controle de pedidos e entregas, especialmente em datas com alta demanda.

b) Análise das ferramentas existentes: Foram examinados os métodos utilizados atualmente pela empresa, como o uso do WhatsApp para contato com clientes e papéis soltos para organização das entregas. Essa análise revelou limitações na organização dos dados e na rastreabilidade das encomendas.

Segundo Pressman (2016), uma análise de requisitos deve contemplar tanto as necessidades explícitas quanto as implícitas dos usuários, além de considerar o ambiente operacional. Com base nessa premissa, o levantamento foi estruturado para garantir que o sistema proposto fosse viável, funcional e aderente à realidade da empresa.

A coleta dos requisitos também buscou alinhar as funcionalidades do sistema à experiência do usuário, conforme defendem Ferreira, Chauvel e Silveira (2006), que apontam a usabilidade como um diferencial para a qualidade de sistemas voltados ao comércio eletrônico. Portanto, esta etapa forneceu os insumos necessários para estruturar as funcionalidades da plataforma, garantindo que ela atenda não apenas aos aspectos técnicos, mas também aos objetivos da empresa: melhorar o atendimento ao cliente, otimizar a gestão de entregas e modernizar a

apresentação dos produtos por meio de uma vitrine digital interativa.

6 DESENVOLVIMENTO

6.1 Requisitos do sistema

O levantamento dos requisitos garante que o sistema desenvolvido atenda às necessidades da floricultura. Segundo Sommerville (2019), os requisitos de software são descrições das funcionalidades e restrições que o sistema deve atender, e são classificados geralmente como requisitos funcionais e não funcionais. A correta identificação e documentação desses requisitos é importante para o sucesso do projeto de software, pois servem como base para o design, implementação e validação do sistema.

6.1.1 Requisitos de Ambiente (Hardware e Software)

Para garantir o funcionamento adequado da plataforma web, são definidos os seguintes requisitos de hardware e software, tanto para o ambiente do servidor, onde a aplicação será hospedada, quanto para o ambiente do cliente, que acessa o sistema.

6.1.1.1 Requisitos para a Hospedagem

Estes são os requisitos para o ambiente do servidor, que abrigará os componentes de back-end (NestJS), front-end (Next.js) e o banco de dados da aplicação.

- **Requisitos de Hardware:**
 - CPU: 1 vCPU (ou equivalente).
 - Memória RAM: Mínimo de 512 MB, recomendado 1 GB para maior estabilidade.
 - Armazenamento: Mínimo de 10 GB de espaço em disco (SSD recomendado para melhor performance).
- **Requisitos de Software:**
 - Sistema Operacional: Linux ou qualquer outro sistema compatível com Node.js.

- Ambiente de Execução: Node.js, conforme especificado no back-end do projeto.
- Sistema de Banco de Dados: Compatível com a implementação, como um serviço de banco de dados da AWS.
- Servidor Web (se necessário): Nginx ou Apache para servir como proxy reverso (prática recomendada em produção).

6.1.1.2 Requisitos para o usuário

Estes são os requisitos para os dispositivos que acessarão o sistema, seja o cliente final visualizando o catálogo ou o funcionário gerenciando o painel administrativo.

- Requisitos de Hardware:
 - Qualquer computador de mesa, notebook, tablet ou smartphone capaz de executar um navegador de internet moderno.
 - Memória RAM: Mínimo de 2 GB.
 - Resolução de Tela: O sistema será responsivo, adaptando-se a diferentes resoluções, com um mínimo de 320px de largura para dispositivos móveis.
- Requisitos de Software:
 - Sistema Operacional: Windows, macOS, Linux, Android ou iOS.
 - Navegador de Internet (Browser): Qualquer navegador moderno com suporte completo às tecnologias web essenciais: HTML5, CSS3 e JavaScript, especificamente a versão ECMAScript 2015 (ES6) ou superior.
 - Conectividade: Acesso à internet. O tempo de carregamento da página é otimizado para ser de até 5 segundos em uma conexão 3G padrão.

6.2 Requisitos funcionais

Os requisitos funcionais representam as funcionalidades essenciais que o sistema deve oferecer, refletindo as ações esperadas pelos usuários no contexto da floricultura. Como afirma Pressman (2016), requisitos funcionais descrevem "os serviços que o sistema deve fornecer, as reações do sistema a entradas específicas

e seu comportamento em determinadas situações". No Quadro 2, é detalhado cada requisito funcional do sistema.

QUADRO 2 – REQUISITOS FUNCIONAIS

ID	Requisito Funcional	Descrição
RF 01	Exibição dos produtos em vitrine digital	A vitrine digital deve apresentar os produtos com foto, nome, valor e descrição, permitindo que o cliente visualize as opções de forma clara e atrativa
RF 02	Gerenciamento de Produtos	O administrador deve poder cadastrar, editar e excluir produtos do catálogo. Ao cadastrar ou editar, ele deverá associar o produto a uma categoria previamente definida.
RF 03	Gerenciamento de Categorias	O administrador deve poder criar, editar e excluir as categorias de produtos, garantindo a consistência dos dados para a organização e filtragem na vitrine.
RF 04	Filtro de produtos por categoria	Na vitrine, o cliente deve poder filtrar os produtos por categoria, tornando a experiência de navegação mais rápida e intuitiva.
RF 05	Autenticação e Controle de Acesso	O acesso à área administrativa deve ser restrito a usuários autenticados com login e senha e com as devidas permissões, garantindo a segurança das operações. As senhas dos usuários devem ser armazenadas no banco de dados de forma segura, utilizando um algoritmo de hashing moderno com salt (ex: Bcrypt).
RF 06	Registro e Gerenciamento de Entregas	O administrador deve poder registrar novas entregas (pedidos), inserindo informações detalhadas do cliente, do destinatário, do produto escolhido e do agendamento.
RF 07	Acompanhamento do Status das Entregas	Cada pedido deve ter um status (ex: "Recebido", "Entregue") que o administrador possa atualizar para organizar o fluxo logístico e controlar as encomendas.
RF 08	Geração de Relatórios Simples	O sistema deve permitir a geração de relatórios básicos, como o de produtos mais vendidos, para fornecer informações que auxiliem na tomada de decisões estratégicas.

FONTE: Elaborado pelo autor (2025).

6.3 Requisitos não funcionais

Já os requisitos não funcionais tratam de propriedades do sistema, como desempenho, usabilidade, confiabilidade e portabilidade. De acordo com Glinz (2007), requisitos não funcionais influenciam diretamente a experiência do usuário e a aceitabilidade do sistema, mesmo que não definam comportamentos diretamente observáveis. O quadro 3 apresenta todos os requisitos não funcionais do sistema, cada um com sua descrição e objetivo.

QUADRO 3 – REQUISITOS NÃO FUNCIONAIS

ID	Requisito Não Funcional	Descrição
RNF 01	Interface Amigável e Intuitiva	A interface será construída com foco na simplicidade e organização lógica , utilizando ícones e botões claros para reduzir a curva de aprendizado e permitir o uso eficiente por usuários com pouca familiaridade com tecnologia.
RNF 02	Responsividade e Compatibilidade	A plataforma será responsiva e compatível com diferentes resoluções de tela , adaptando-se a smartphones, tablets e desktops para garantir o funcionamento correto em qualquer dispositivo do usuário.
RNF 03	Desempenho de Carregamento	As páginas principais do sistema, como a vitrine e o painel administrativo, devem ter um tempo de carregamento máximo de 5 segundos em uma conexão 3G padrão , visando minimizar a frustração do usuário e melhorar a experiência de navegação.
RNF 04	Alta Disponibilidade	O sistema deverá manter uma disponibilidade de 95% do tempo , com alta tolerância a falhas, para garantir o acesso contínuo à plataforma, inclusive em períodos de alto volume de acessos.
RNF 05	Arquitetura Escalável	A arquitetura do sistema será estruturada de forma modular e escalável , com separação de componentes, a fim de facilitar a manutenção e a inclusão de novas funcionalidades no futuro sem a necessidade de uma reestruturação completa.
RNF 06	Performance de Transação	A operação de salvar um novo registro (ex: um produto ou uma entrega) deve ser concluída e retornar uma confirmação ao usuário em no máximo 1.5 segundos.
RNF 07	Feedback de Interface	Todas as operações de criação, edição ou exclusão devem fornecer um feedback visual imediato ao usuário (ex: uma notificação de "sucesso" ou "erro").

FONTE: Elaborado pelo autor (2025).

6.3.1 Métodos de validação

- RNF01: Realização de testes práticos com o usuário-alvo, que deverá executar tarefas predefinidas (ex: cadastrar um produto) sem treinamento. O sucesso será medido pela capacidade de completar a tarefa de forma autônoma.
- RNF02: Verificação da interface utilizando as ferramentas de desenvolvedor dos navegadores para simular diferentes dispositivos. Validação final através de testes de navegação em um celular e um desktop físicos.
- RNF03: Utilização de ferramentas como o Google Lighthouse⁷ para

⁷ ferramenta automatizada do Google que avalia a performance, acessibilidade e boas

auditar as páginas principais, com a simulação de rede configurada para "3G Lenta". Os relatórios devem apontar um tempo de interação inferior a 5s.

- RNF04: Configuração de um serviço de monitoramento de uptime (ex: UptimeRobot, AWS CloudWatch) para verificar a disponibilidade do sistema em intervalos regulares (ex: a cada 5 minutos) durante um período de observação.
- RNF05: Inspeção da arquitetura do código-fonte para verificar a correta separação de responsabilidades entre as camadas (Controller⁸, Service⁹, Repository¹⁰ e Entity¹¹), garantindo baixo acoplamento e alta coesão.
- RNF06: Execução da ação de salvar 10 vezes em ambiente de teste e cálculo do tempo médio de resposta do servidor para a requisição.
- RNF07: Execução de todas as operações de CRUD¹², verificando se uma notificação visual clara e apropriada é exibida em cada caso.

6.4 Modelagem do Sistema

A modelagem do sistema foi necessária para garantir organização e clareza na estrutura da aplicação. Foram utilizados diferentes diagramas para representar as funcionalidades, entidades e interações do sistema.

6.4.1 Diagrama de Casos de Uso

Este diagrama representa as funcionalidades principais disponíveis no sistema e os atores envolvidos em cada ação (LUCIDCHART, s.d.). A figura 6 os

práticas de páginas web, gerando relatórios com sugestões de melhoria.

⁸ componente responsável por receber as requisições do usuário, processar a lógica de controle e encaminhar as respostas, atuando como intermediário entre a interface e a lógica de negócio.

⁹ camada que contém a lógica de negócio, responsável por processar dados e regras específicas da aplicação.

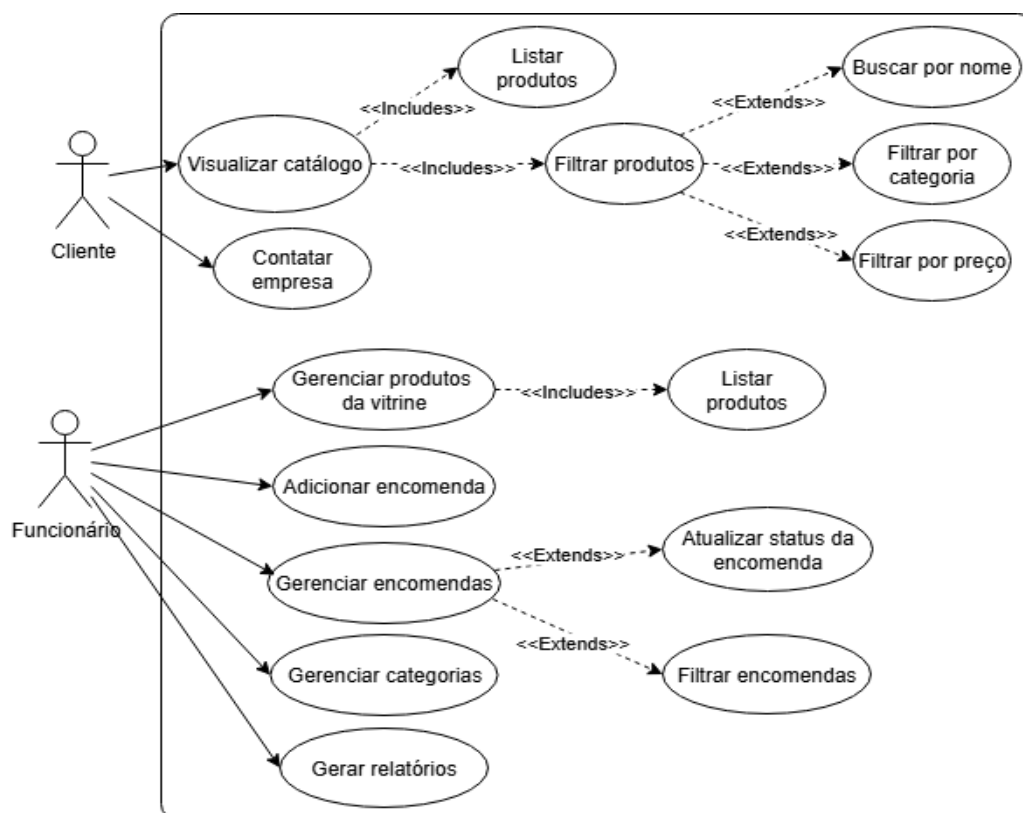
¹⁰ camada que gerencia o acesso aos dados, abstraindo operações de armazenamento e recuperação do banco de dados.

¹¹ representação dos objetos do domínio, geralmente mapeados para tabelas do banco de dados, contendo atributos e relacionamentos.

¹² conjunto básico de operações para manipulação de dados: Create (criar), Read (ler), Update (atualizar) e Delete (excluir).

detalha a seguir na figura 7:

FIGURA 7 – DIAGRAMA DE CASOS DE USO



FONTE: Elaborado pelo autor (2025).

6.4.1.1 Detalhamento dos casos de uso

1. Caso de Uso: Visualizar Catálogo

- Ator Principal: Cliente
- Resumo: Permite que um visitante ou cliente visualize os produtos oferecidos pela floricultura através de uma vitrine digital interativa.
- Pré-condições: Nenhuma. O catálogo é de acesso público.
- Fluxo Principal (Cenário de Sucesso):
 1. O Cliente acessa a página principal da aplicação.
 2. O sistema busca e exibe a lista de produtos disponíveis, mostrando para cada um a imagem, nome e preço.
 3. O Cliente pode opcionalmente utilizar os filtros de categoria e preço para refinar a busca.

4. O sistema atualiza a lista de produtos em tempo real conforme os filtros são aplicados.
 5. O Cliente clica em um produto para ver mais detalhes.
- Pós-condições: O Cliente visualizou os produtos e pode decidir qual comprar, por meio de contato no whatsapp da floricultura.
 - Fluxo Alternativo: Caso não haja produtos cadastrados no sistema, uma mensagem informativa como "Nenhum produto encontrado" é exibida.

2. Caso de Uso: Gerenciar Produtos

- Ator Principal: Funcionário
- Resumo: Representa o conjunto de funcionalidades que permitem ao Funcionário manter o catálogo de produtos da floricultura atualizado. Conforme o requisito funcional RF02, este caso de uso engloba o ciclo de vida completo de um produto.
- Pré-condições: O funcionário deve estar autenticado no sistema com permissões de administrador.
- Fluxo Principal (Cenário de Sucesso):
 1. O funcionário acessa a área de "Gerenciar Catálogo".
 2. O sistema exibe a lista de todos os produtos cadastrados.
 3. A partir desta tela, o Funcionário pode iniciar uma das seguintes ações:
 - Cadastrar: Clica em "Adicionar Produto" , preenche o formulário com nome, descrição, preço, imagem e seleciona uma categoria de uma lista pré-definida, e então salva o novo produto.
 - Editar: Seleciona um produto existente, o sistema exibe um formulário com os dados atuais, o funcionário altera as informações desejadas e salva.
 - Excluir: Seleciona um produto, confirma a ação em uma janela de aviso e o sistema remove o produto do banco de dados.
- Pós-condições: O catálogo de produtos é modificado (um item foi criado, atualizado ou removido).

3. Caso de Uso: Adicionar Encomenda

- Ator Principal: Funcionário
- Resumo: Permite que o Funcionário registre no sistema uma nova venda ou pedido de entrega, centralizando as informações logísticas.
- Pré-condições: O funcionário deve estar autenticado no sistema com permissões de administrador.
- Fluxo Principal (Cenário de Sucesso):
 1. O Funcionário acessa a área de "Entregas" e seleciona a opção para adicionar uma nova encomenda.
 2. O sistema apresenta o formulário de "Adicionar Entrega".
 3. O Funcionário preenche todos os campos obrigatórios: dados do produto, data e hora da entrega, nome e telefone do cliente, nome do destinatário, endereço completo e informações de pagamento.
 4. Ao submeter o formulário, o sistema valida os dados inseridos.
 5. O sistema salva as informações no banco de dados, criando um novo registro de Pedido com o status inicial "Recebido".
 6. O sistema exibe uma mensagem de sucesso e atualiza a lista de entregas.
- Pós-condições: Uma nova encomenda é registrada e fica visível no painel de gerenciamento de entregas.

4. Caso de Uso: Gerenciar Categorias

- Ator Principal: Funcionário
- Resumo: Permite que o Funcionário administre a lista de categorias de produtos, garantindo a organização e a consistência dos dados do catálogo.
- Pré-condições: O funcionário deve estar autenticado no sistema com permissões de administrador.
- Fluxo Principal (Cenário de Sucesso):
 1. O funcionário acessa a funcionalidade de gerenciamento de categorias no painel administrativo.
 2. O sistema exibe a lista de todas as categorias já criadas.
 3. O Funcionário pode realizar as seguintes ações:
 - Criar: Adiciona uma nova categoria informando seu nome.
 - Editar: Altera o nome de uma categoria existente.

- Excluir: Remove uma categoria que não está mais em uso.
- Pós-condições: A lista de categorias disponíveis para associação com produtos é atualizada.
- Fluxo Alternativo: O sistema deve impedir a exclusão de uma categoria se houver produtos associados a ela, informando o usuário sobre a restrição.

5. Caso de Uso: Gerar Relatórios

- Ator Principal: Funcionário
- Resumo: Permite que o funcionário gere relatórios simples para análise do desempenho do negócio, conforme o requisito funcional RF08.
- Pré-condições: O funcionário deve estar autenticado no sistema com permissões de administrador.
- Fluxo Principal (Cenário de Sucesso):
 1. O Funcionário acessa a seção de "Relatórios" do sistema.
 2. O sistema apresenta os tipos de relatórios disponíveis (ex: "Produtos Mais Vendidos").
 3. O Funcionário seleciona o tipo de relatório e, opcionalmente, um período de tempo.
 4. O sistema processa os dados de pedidos e produtos para compilar as informações.
 5. O sistema exibe o relatório formatado na tela.
- Pós-condições: O funcionário obtém insights sobre o desempenho das vendas para auxiliar na tomada de decisões.

6. Caso de Uso: Contatar Empresa

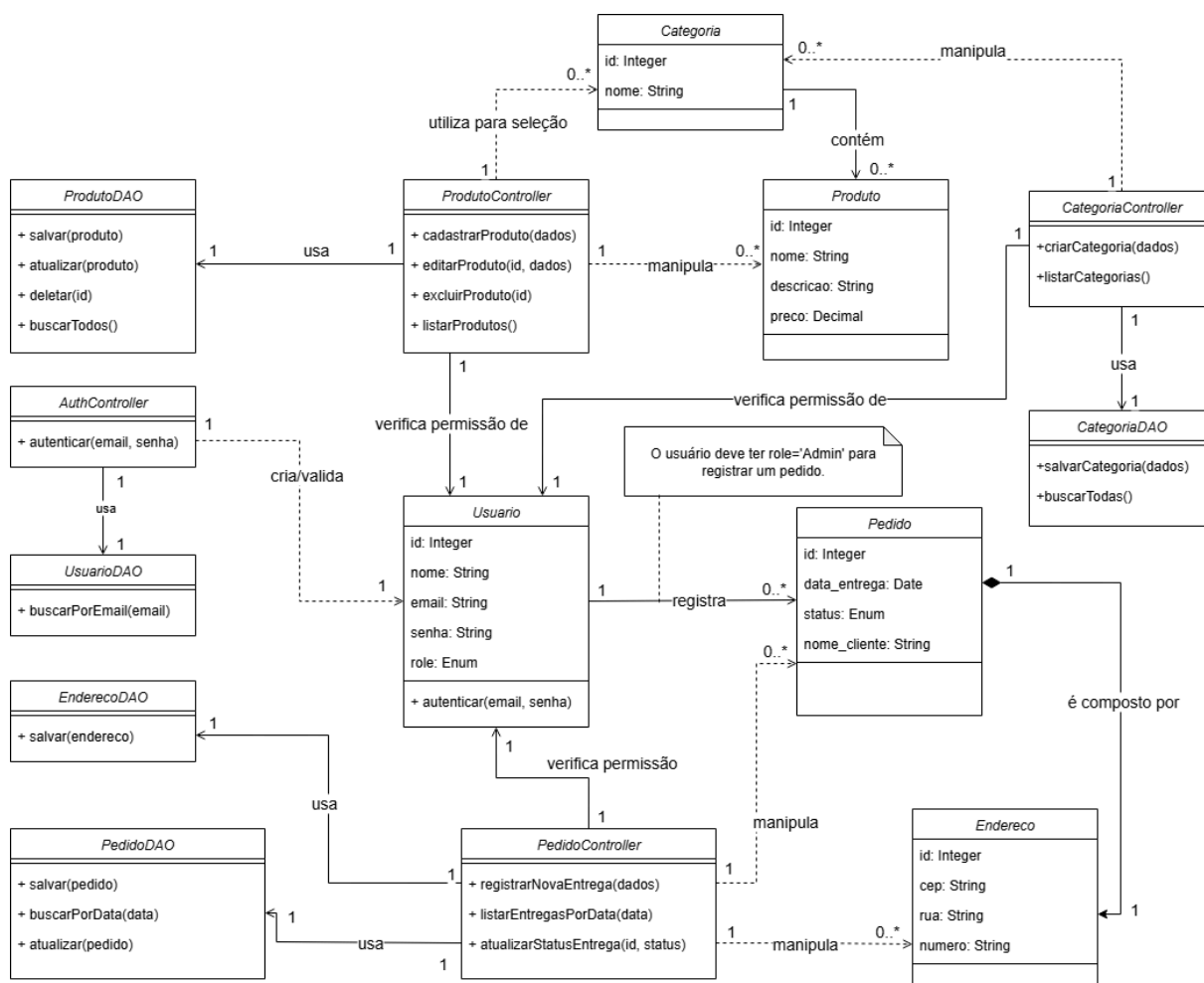
- Ator Principal: Cliente
- Resumo: Este caso de uso permite que o Cliente inicie uma conversa direta com a floricultura através do aplicativo WhatsApp. O objetivo é fornecer um canal de comunicação de baixa fricção para tirar dúvidas, solicitar orçamentos ou fazer encomendas que não se encaixam diretamente nos produtos do catálogo.
- Pré-condições: O cliente precisa ter acesso à internet. A presença do aplicativo WhatsApp no dispositivo do cliente facilita a interação.

- Fluxo Principal (Cenário de Sucesso):
 1. O Cliente visualiza um botão ou ícone de contato na página da vitrine (ex: "Faça seu pedido!").
 2. O Cliente clica no botão.
 3. O sistema redireciona o navegador do Cliente para uma URL específica da API do WhatsApp.
 4. Esta URL abre o aplicativo WhatsApp (se instalado no dispositivo) ou a interface do WhatsApp Web, já com uma conversa iniciada com o número de telefone da floricultura.
- Pós-condições: O Cliente está fora da aplicação e dentro do ambiente do WhatsApp, pronto para se comunicar com a floricultura.
- Fluxo Alternativo: Caso o Cliente não tenha o WhatsApp instalado em seu dispositivo (especialmente em um desktop), o navegador abrirá o WhatsApp Web e solicitará que ele se conecte à sua conta escaneando um QR Code com o celular.

6.4.2 Diagrama de Classes

O diagrama de classes apresenta a estrutura lógica do sistema, com os objetos principais, atributos e métodos envolvidos no processo (GEEKSFORGEES, 2025). A figura 8 apresenta o diagrama de classes do sistema:

FIGURA 8 – DIAGRAMA DE CLASSES



FONTE: Elaborado pelo autor (2025).

6.4.3 Diagrama Entidade-Relacionamento (DER)

O DER foi elaborado para representar graficamente o banco de dados do sistema, destacando as tabelas, atributos e relacionamentos (LUCIDCHART, s.d.).

Para ilustrar as cardinalidades, o diagrama utiliza a notação pé de galinha (*crow's foot*), um padrão adotado na modelagem de dados por sua expressividade e clareza visual (SOMMERVILLE, 2019). A notação "pé de galinha" é utilizada em DERs para representar os relacionamentos entre entidades, especificando com clareza a cardinalidade e a obrigatoriedade das participações. Essa notação faz uso de símbolos gráficos nas extremidades das linhas de relacionamento:

- Um pequeno círculo indica participação opcional (zero ocorrências possíveis);
- Duas barras verticais (||) representam participação obrigatória e única (um e somente um);

decisão de incluir o atributo role com o tipo ENUM ('Admin', 'Funcionario') diretamente no banco de dados é estratégica por dois motivos:

- **Integridade de Dados:** O banco de dados só permitirá a inserção dos valores predefinidos, eliminando o risco de papéis inválidos serem cadastrados no sistema.
- **Implementação da Regra de Negócio:** Conforme definido no requisito funcional de autenticação, nem todos os usuários podem ter acesso a todas as funcionalidades. A lógica de negócio na aplicação tratará esse atributo da seguinte forma: antes de executar uma operação crítica (como cadastrar um novo produto ou gerar relatórios), a camada de controle (Controller) verificará o role do usuário logado. Se o valor não for 'Admin', a ação será bloqueada, retornando uma mensagem de acesso não autorizado. Isso garante que apenas administradores possam realizar ações sensíveis.

b) Entidades Categoria e Produto

A separação de Categoria em uma entidade própria, em vez de ser um simples campo de texto no Produto, é uma decisão de normalização de dados. Isso resolve um problema comum em sistemas de catálogo: a inconsistência de dados. Sem uma entidade Categoria, os usuários poderiam cadastrar "Buquês", "buques" e "Buque", tratando-os como três categorias distintas. Com a entidade Categoria, o administrador cria uma lista única e controlada, e ao cadastrar um Produto, ele seleciona uma das categorias existentes. Isso garante a consistência para os filtros da vitrine e para a geração de relatórios.

c) Entidade Pedido

A entidade Pedido centraliza todas as informações de uma transação. As decisões de design mais importantes aqui são:

- **Composição com Endereco:** Um endereço de entrega não existe no sistema sem um Pedido associado. A relação de composição garante que cada pedido tenha seu próprio registro de endereço, evitando que a alteração de um endereço em um pedido afete outro, e garante que, se um pedido for excluído, seu endereço correspondente também seja.

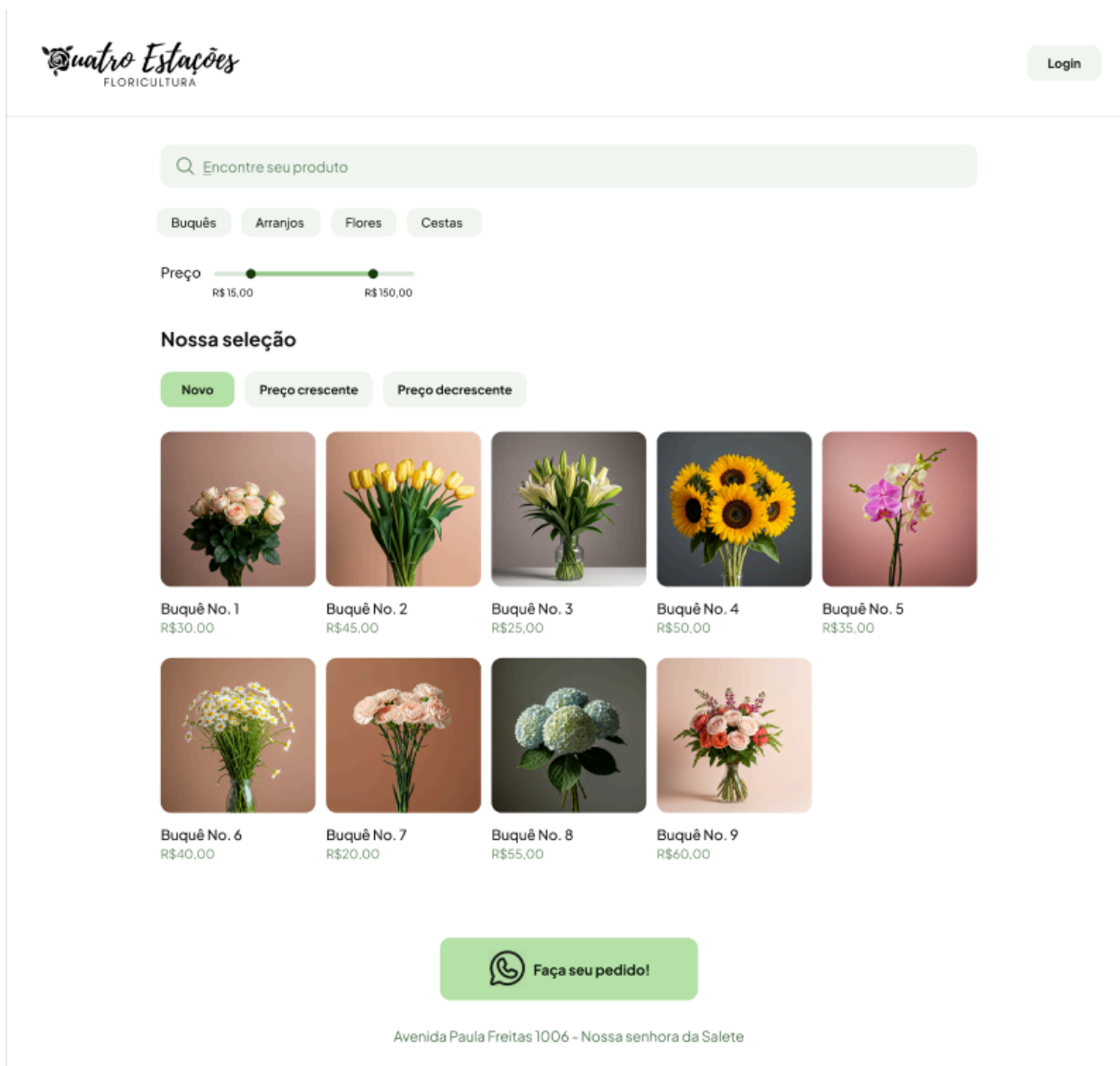
- Armazenamento de Dados do Cliente: Os dados essenciais do comprador (nome_cliente, telefone_cliente) são armazenados diretamente na tabela Pedido. Isso simplifica o modelo, alinhando-o à necessidade do negócio, que é registrar a informação da venda de forma direta, conforme o processo descrito no protótipo.
- Rastreabilidade com Usuario: A chave estrangeira usuario_id cria um vínculo direto do Pedido com o Usuario que o registrou. Isso é fundamental para a rastreabilidade e auditoria, permitindo saber qual funcionário registrou cada venda.

6.5 Prototipagem

A prototipagem da interface foi realizada utilizando a ferramenta Figma, uma plataforma online voltada ao design de interfaces e prototipação de aplicações digitais. As telas iniciais foram pensadas com base nos requisitos do sistema e, posteriormente, refinadas com ajustes visuais, hierarquia de informação e responsividade. Durante o processo, foram aplicadas as heurísticas de usabilidade propostas por Jakob Nielsen, um conjunto de dez princípios amplamente aceitos para orientar o design de interfaces eficientes e intuitivas. Essas heurísticas incluem diretrizes como visibilidade do status do sistema, correspondência entre o sistema e o mundo real, controle e liberdade do usuário, consistência, prevenção de erros, entre outras, que contribuem diretamente para uma boa experiência do usuário (NIELSEN, 1995). O uso do Figma favoreceu a criação de wireframes, mockups e protótipos navegáveis de forma intuitiva, priorizando usabilidade e apelo visual.

A Figura 10 apresenta a tela principal da vitrine digital, que é a interface pública do sistema. Seu objetivo é exibir os produtos disponíveis de forma visualmente atrativa, com imagem, nome e preço, além de oferecer ferramentas de filtro e ordenação para facilitar a navegação e a decisão de compra do cliente. Além disso, a tela possui o botão de ação principal ("Faça seu pedido!"), que direciona o cliente para o contato via WhatsApp.

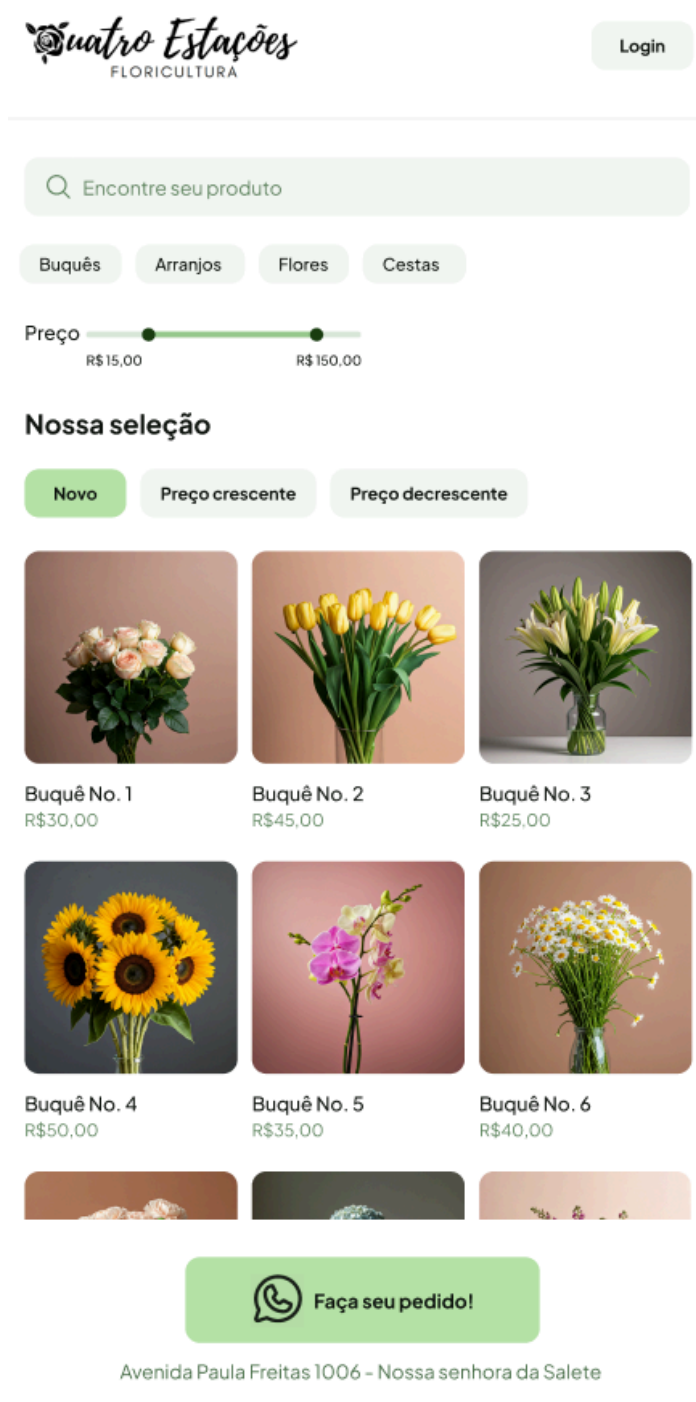
FIGURA 10 – VITRINE DIGITAL



FONTE: Elaborado pelo autor (2025).

A Figura 11 demonstra a aplicação do design responsivo no protótipo, apresentando a versão da vitrine digital para dispositivos móveis. Conforme o requisito não funcional RNF02.

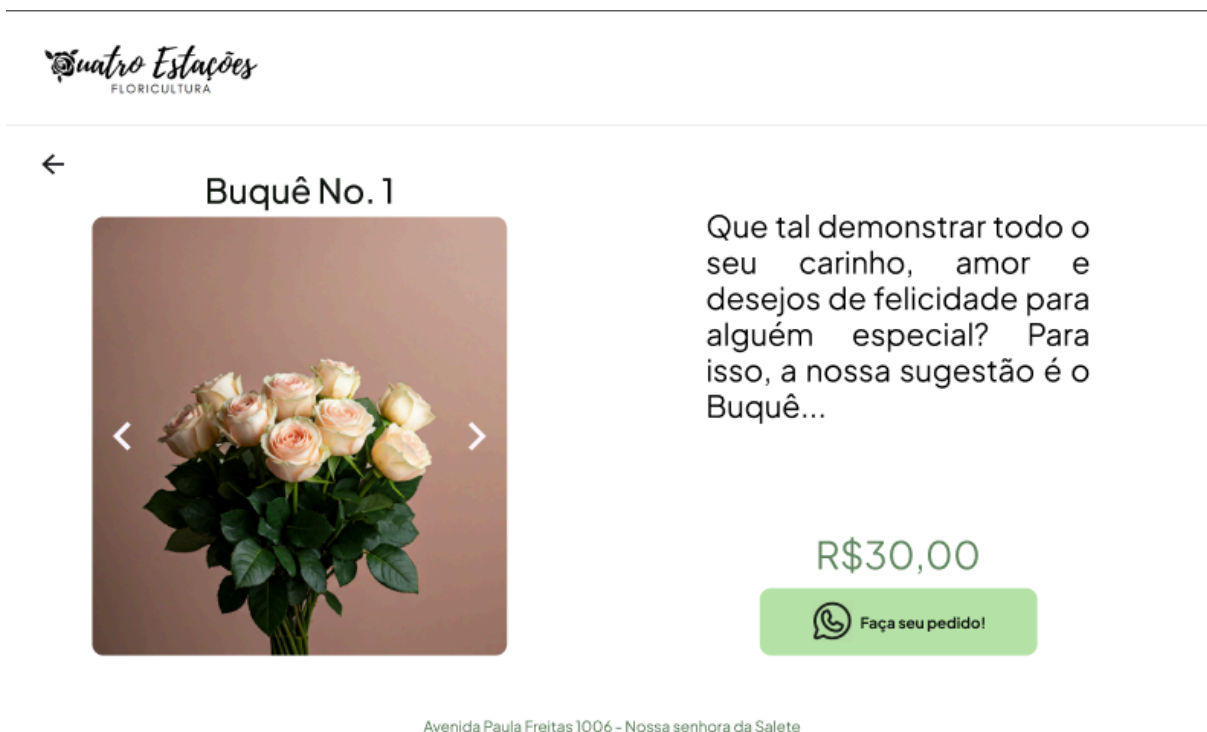
FIGURA 11 – VITRINE DIGITAL (VERSÃO MOBILE)



FONTE: Elaborado pelo autor (2025).

A Figura 12 detalha a página de um produto específico. Nela, são exibidos a imagem em destaque, uma descrição completa, o preço e o botão de ação principal ("Faça seu pedido!"), que direciona o cliente para o contato via WhatsApp.

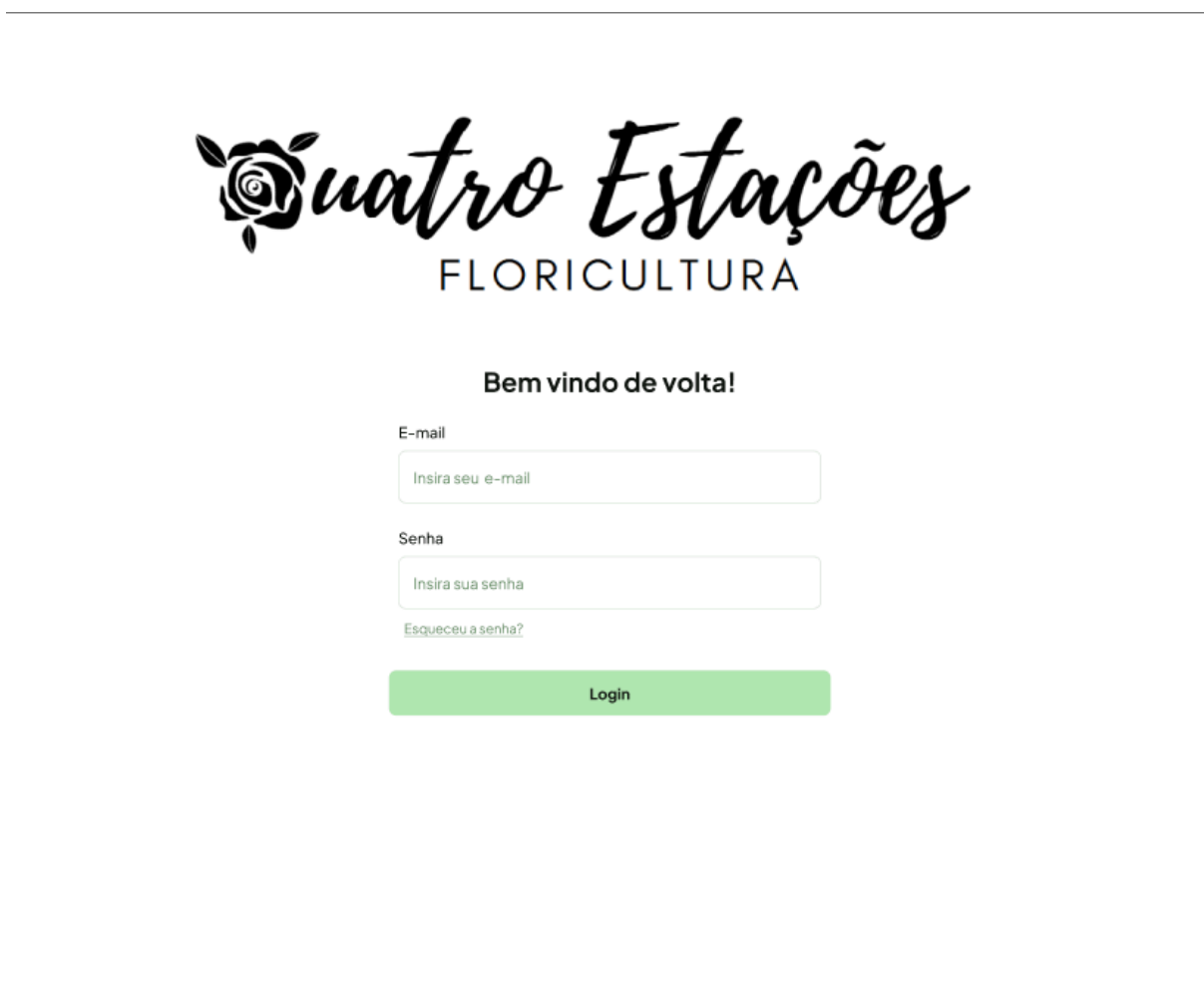
FIGURA 12 – DETALHES DO PRODUTO



FONTE: Elaborado pelo autor (2025).

A Figura 13 ilustra a tela de autenticação, que serve como portal de entrada para a área restrita do sistema. Nela, o funcionário deve inserir suas credenciais (e-mail e senha) para acessar as funcionalidades de gerenciamento, conforme o requisito funcional RF05.

FIGURA 13 – TELA DE LOGIN

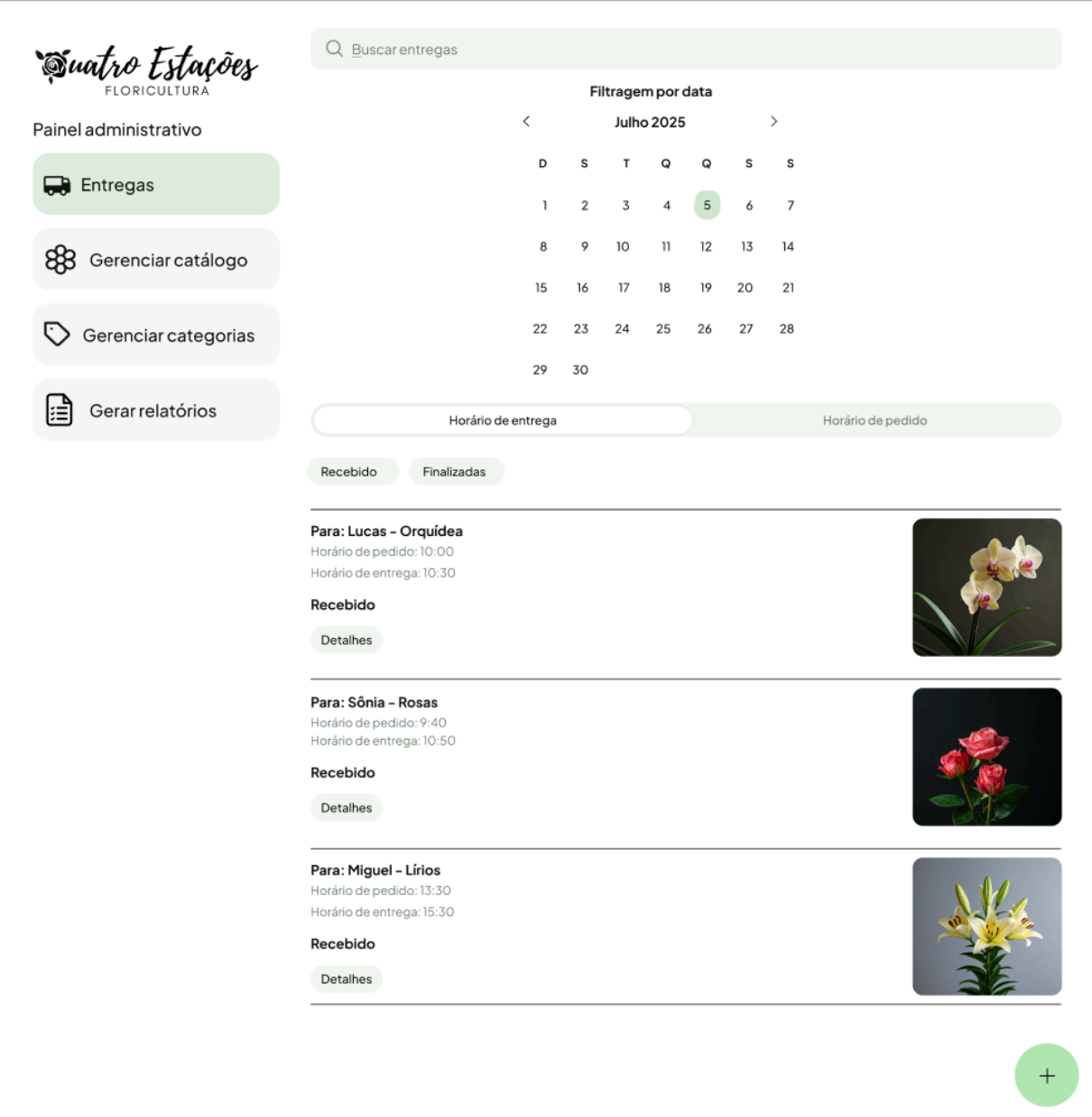


The login screen features the company logo at the top, which consists of a stylized rose icon followed by the text "Quatro Estações" in a large, elegant script font, and "FLORICULTURA" in a smaller, clean sans-serif font below it. Centered below the logo is the greeting "Bem vindo de volta!". The form includes two input fields: "E-mail" with a placeholder "Insira seu e-mail" and "Senha" with a placeholder "Insira sua senha". Below the password field is a link "Esqueceu a senha?". At the bottom of the form is a green "Login" button.

FONTE: Elaborado pelo autor (2025).

A Figura 14 exibe o painel de gerenciamento de entregas, o núcleo do controle logístico do sistema. A tela apresenta uma lista de encomendas com seu respectivo status (ex: "Recebido"), ela permite ver os detalhes da entrega e oferece opções de filtragem por uma data específica através do calendário.

FIGURA 14 – GERENCIAMENTO DE ENTREGAS



FONTE: Elaborado pelo autor (2025).

A Figura 15 detalha o formulário para adicionar uma nova encomenda. Esta tela foi projetada para centralizar o registro de todas as informações cruciais para a entrega, como dados do cliente, do destinatário, endereço completo, produto e horário.

FIGURA 15 – ADIÇÃO DE ENCOMENDA

Quatro Estações
FLORICULTURA

Painel administrativo

- Entregas
- Gerenciar catálogo
- Gerenciar categorias
- Gerar relatórios

Adicionar entrega

Adicionar imagem do produto
Toque para selecionar um arquivo ou tirar uma foto

Produto *
Insira o nome do produto

CEP
Insira o CEP do destino

Rua *
Insira a rua do destino

Número *

Bairro *
Insira o bairro do destino

Observação (opcional)
Instruções de entrega, aonde cobrar, etc...

Horário da entrega *
00 : 00

Dia da entrega *
XX / XX

Nome de quem vai receber *
Insira o nome de quem vai receber

Nome do cliente *
Insira o nome de quem comprou o produto

Telefone do cliente *
Insira o telefone de quem comprou o produto

☒ Cliente pagou?

☒ Cobrar na entrega?

Cancelar **Enviar**

FONTE: Elaborado pelo autor (2025).

A Figura 16 exemplifica uma das heurísticas de usabilidade aplicadas no projeto, a de prevenção de erros. O sistema apresenta uma janela modal¹³ de confirmação antes de executar uma ação destrutiva, como o cancelamento de uma edição ou uma exclusão de imagem, garantindo que o usuário tenha controle total sobre suas ações.

¹³ elemento de interface que aparece sobre a tela principal, bloqueando a interação com o conteúdo de fundo até ser fechada, usada para destacar informações ou ações específicas.

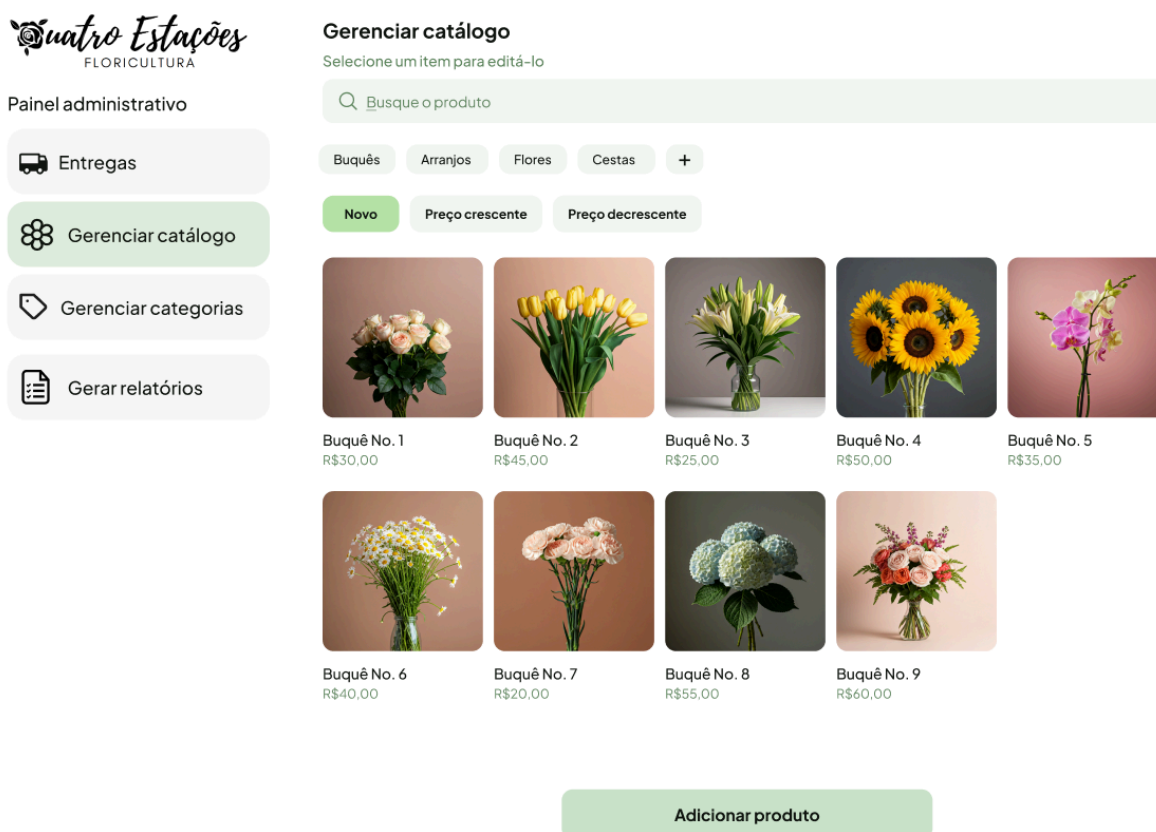
FIGURA 16 – CONFIRMAÇÕES DE AÇÃO



FONTE: Elaborado pelo autor (2025).

A Figura 17 apresenta o painel de gerenciamento de catálogo, que serve como a visão geral administrativa dos produtos. Esta tela permite ao funcionário visualizar todos os itens, utilizar filtros de categoria e ordenação, buscar por um produto específico e iniciar o processo de adição ou edição de um item no catálogo.

FIGURA 17 – TELA DE GERENCIAMENTO DE CATÁLOGO



FONTE: Elaborado pelo autor (2025).

A Figura 18 ilustra o formulário para a criação de um novo produto. Nele, o administrador insere todas as informações essenciais, como nome, descrição, preço,

imagens e associa o item a uma categoria pré-existente, garantindo a consistência dos dados do catálogo.

FIGURA 18 – TELA DE CADASTRO DE NOVO PRODUTO

Quatro Estações
FLORICULTURA

Painel administrativo

- Entregas
- Gerenciar catálogo
- Gerenciar categorias
- Gerar relatórios

Adicione um produto

Imagens do produto
Toque para selecionar um arquivo ou tirar uma foto

Nome do produto

Descrição

Preço (R\$)

Categoria

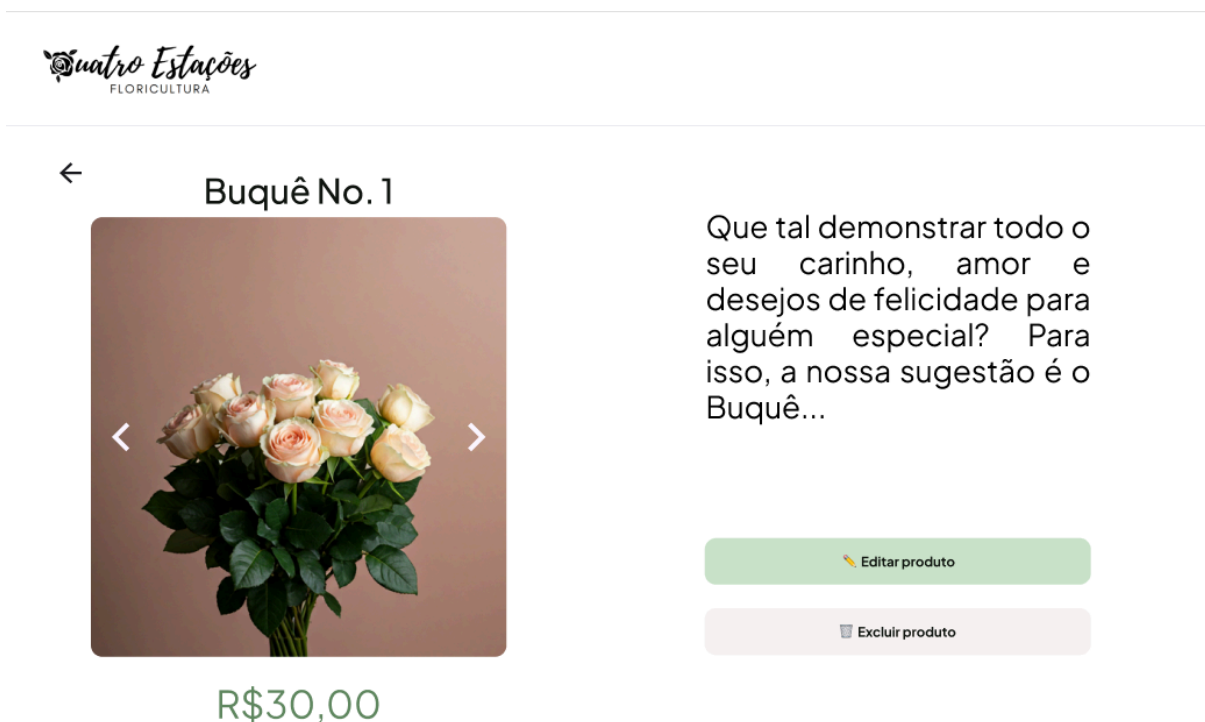
Selecione a categoria

CancelarAdicionar

FONTE: Elaborado pelo autor (2025).

A Figura 19 exibe a tela de detalhe de um produto a partir da perspectiva do administrador. Diferente da visão do cliente, esta interface fornece acesso direto às ações de "Editar produto" e "Excluir produto", centralizando as operações de gerenciamento para um item específico.

FIGURA 19 – TELA DE DETALHE E AÇÕES DO PRODUTO



FONTE: Elaborado pelo autor (2025).

A Figura 20 detalha o formulário de edição de um produto. Os campos são pré-preenchidos com as informações existentes para facilitar as alterações, e esta tela serve como portal para o gerenciamento das imagens do produto.


FIGURA 20 – TELA DE EDIÇÃO DE PRODUTO

Quatro Estações
FLORICULTURA

Painel administrativo

- Entregas
- Gerenciar catálogo
- Gerenciar categorias
- Gerar relatórios

Editar produto



Imagens do produto
Toque para editar as imagens do produto

Nome do produto

Buquê No.1

Descrição

Que tal demonstrar todo o seu carinho, amor e desejos de felicidade para alguém especial? Para isso, a nossa sugestão é o Buquê...

Preço (R\$)

30,00

Categoria

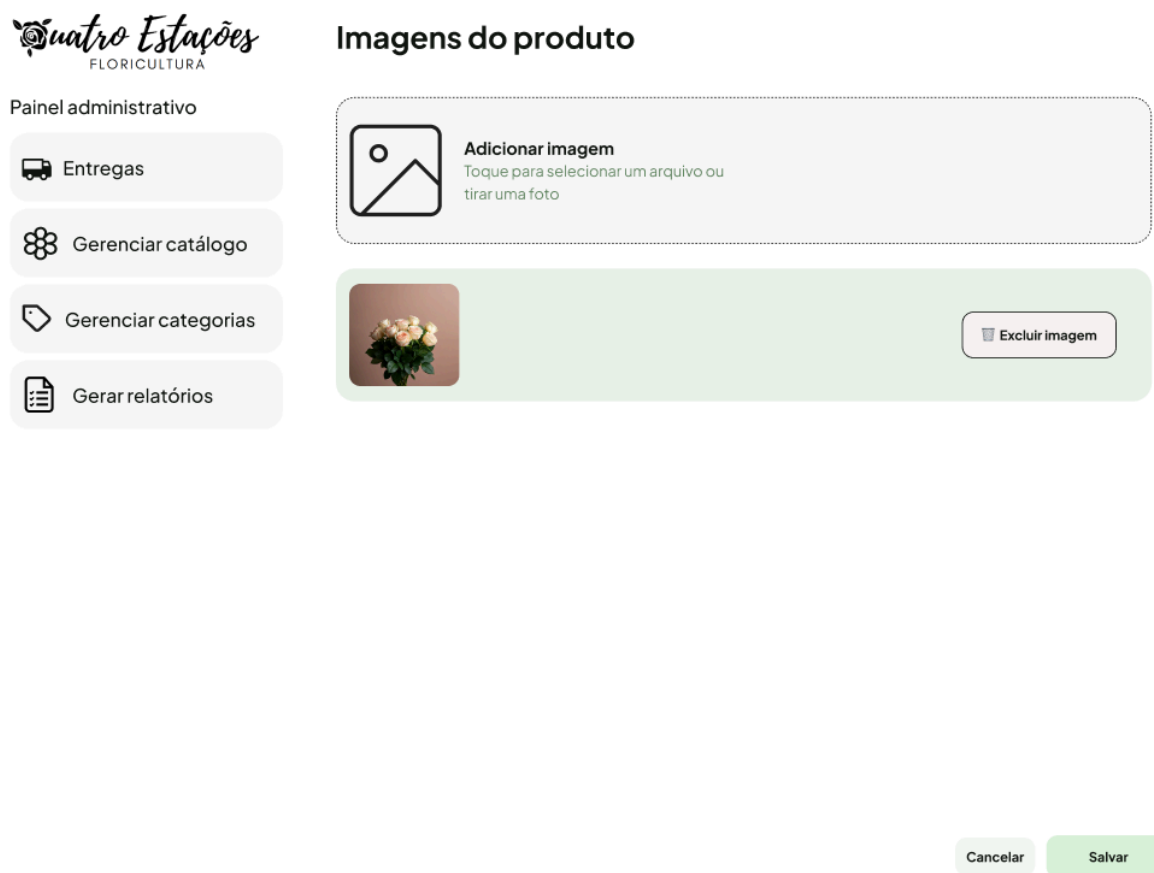
Buquês

Cancelar Salvar

FONTE: Elaborado pelo autor (2025).

A Figura 21 apresenta a interface dedicada para o gerenciamento das imagens de um produto que aparecerão na vitrine. A tela permite ao administrador adicionar novas imagens, seja por upload ou captura, e excluir fotos existentes de forma individualizada.

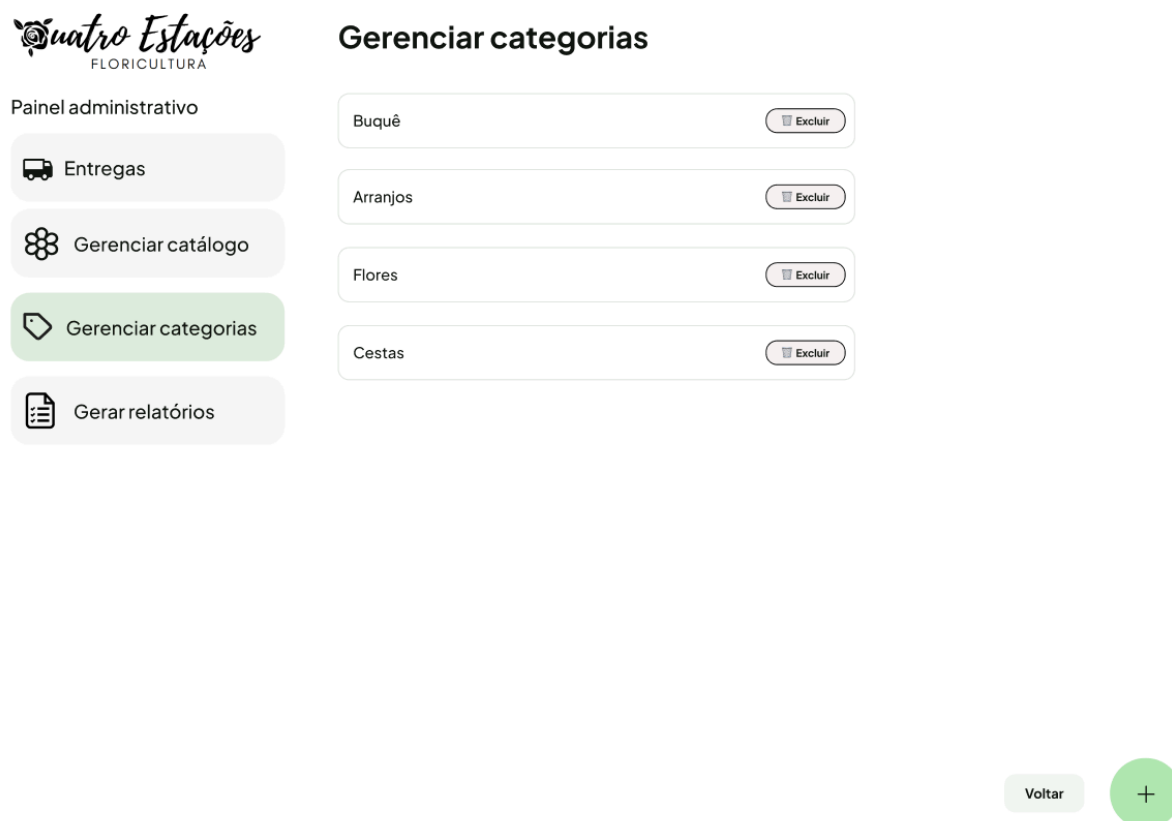
FIGURA 21 – TELA DE GERENCIAMENTO DE IMAGENS DO PRODUTO



FONTE: Elaborado pelo autor (2025).

A Figura 22 mostra o painel para a gestão de categorias. Esta interface permite ao administrador visualizar a lista de todas as categorias existentes, excluí-las e iniciar, através de um modal, a criação de novas categorias para a organização dos produtos.

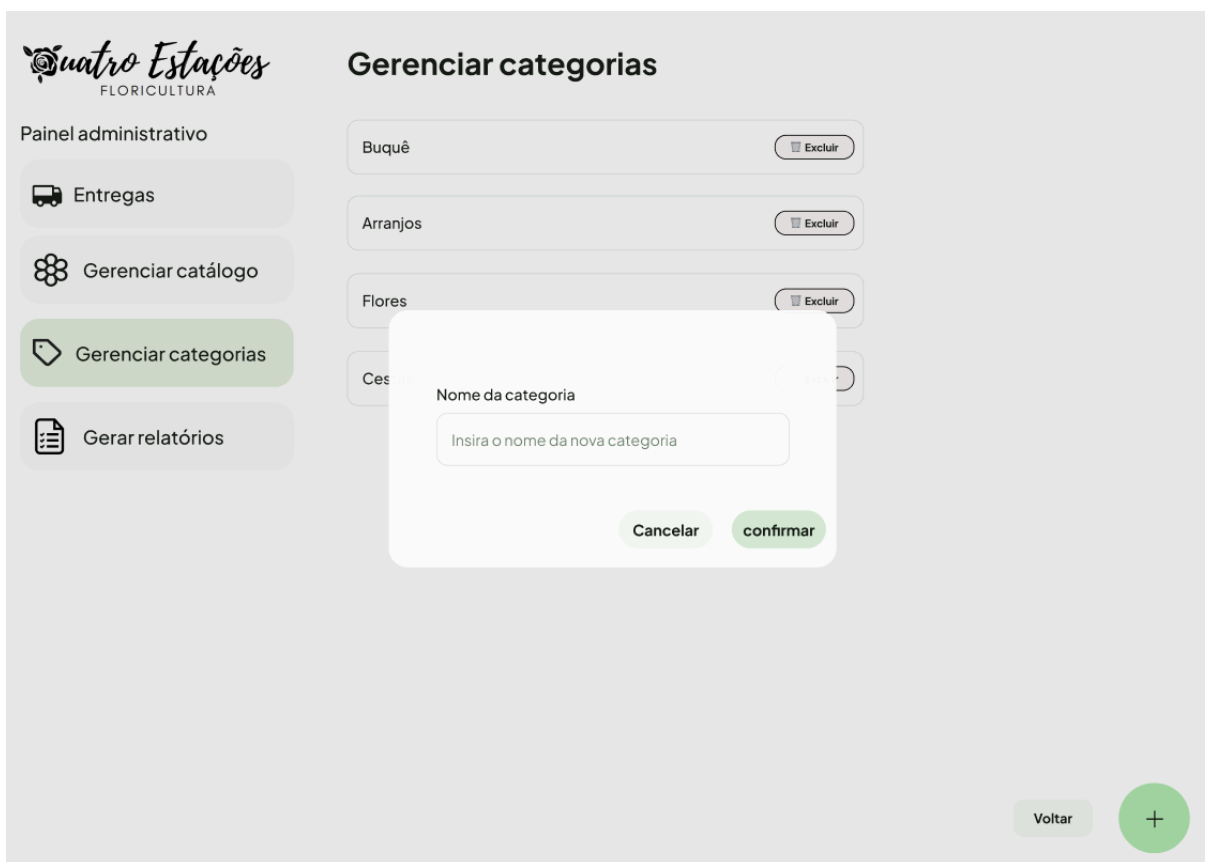
FIGURA 22 – TELA DE GERENCIAMENTO DE CATEGORIAS



FONTE: Elaborado pelo autor (2025).

A Figura 23 apresenta o modal utilizado para adicionar uma categoria.

FIGURA 23 – TELA DE ADICIONAR CATEGORIA



FONTE: Elaborado pelo autor (2025).

A Figura 24 exibe a tela de relatórios do sistema. Projetada para inteligência de negócio, ela permite ao administrador selecionar um tipo de relatório (ex: "Produtos mais vendidos"), filtrar por um período de tempo e baixar os relatórios gerados.

FIGURA 24 – TELA DE GERAÇÃO DE RELATÓRIOS

Quatro Estações
FLORICULTURA

Painel administrativo

- Entregas
- Gerenciar catálogo
- Gerenciar categorias
- Gerar relatórios**

Gerar relatórios

Período

De: XX / XX / XXXX Até: XX / XX / XXXX

Produtos mais vendidos

Entregas por bairro

Entregas por rua

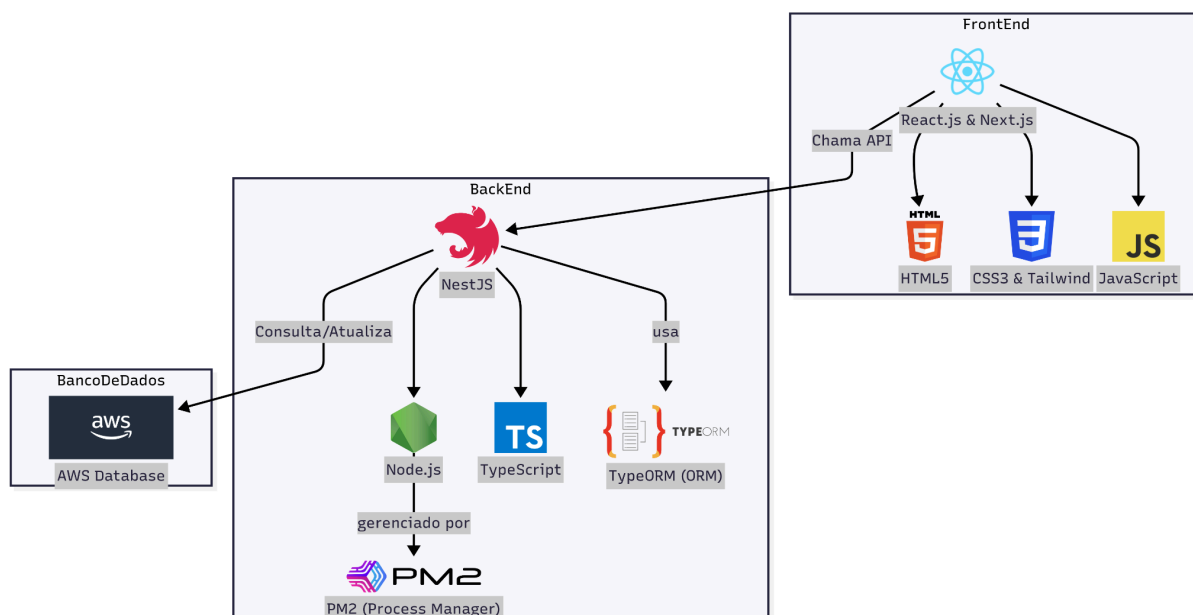
[Voltar](#)

FONTE: Elaborado pelo autor (2025).

6.6 Tecnologias Utilizadas

A seleção das tecnologias para este projeto foi pautada por critérios de performance, escalabilidade, manutenibilidade e pela robustez de cada ferramenta. A seguir, detalha-se o conjunto de tecnologias (stack) adotado para o desenvolvimento da solução. A figura 25 apresenta a arquitetura das tecnologias utilizadas:

FIGURA 25 – TECNOLOGIAS UTILIZADAS



FONTE: Elaborado pelo autor (2025).

6.6.1 Front-end

A interface do usuário foi construída com React.js, uma biblioteca JavaScript mantida pelo Facebook, reconhecida por sua abordagem baseada em componentes. Essa arquitetura permite a criação de UIs complexas a partir de peças isoladas e reutilizáveis, o que simplifica o desenvolvimento e a manutenção (REACT, 2025).

Para estruturar a aplicação React, foi utilizado o framework Next.js. A escolha pelo Next.js foi estratégica devido às suas funcionalidades que otimizam a performance e a experiência do usuário, como a renderização no lado do servidor (Server-Side Rendering - SSR) e a geração de sites estáticos (Static Site Generation - SSG). Esses recursos melhoram o tempo de carregamento inicial da página e são benéficos para a indexação por motores de busca (SEO) (NEXT.JS, 2025).

Para a estilização, optou-se pelo Tailwind CSS, um framework utility-first que permite construir interfaces customizadas de forma rápida, aplicando classes diretamente no HTML, sem a necessidade de escrever CSS customizado, o que acelera o desenvolvimento e mantém a consistência visual (TAILWIND CSS, 2025).

6.6.2 Back-end

Para o desenvolvimento do servidor e da API RESTful, a escolha foi o NestJS, um framework progressivo para Node.js. O NestJS utiliza TypeScript por padrão, o que adiciona tipagem estática ao JavaScript, aumentando a robustez do código e reduzindo a ocorrência de erros em tempo de execução. Sua arquitetura é fortemente inspirada em padrões de desenvolvimento consolidados e faz uso extensivo de Injeção de Dependência (Dependency Injection), o que promove um código modular, desacoplado, escalável e altamente testável (NESTJS, 2025). O ambiente de execução para o back-end é o Node.js, escolhido por sua arquitetura orientada a eventos e não-bloqueante, ideal para construir aplicações de rede de alta performance. Para a comunicação com o banco de dados, foi utilizado o TypeORM. Além do mapeamento objeto-relacional, a ferramenta foi empregada para gerenciar a evolução do esquema do banco de dados através de Migrations, garantindo que as alterações na estrutura das tabelas sejam versionadas e aplicadas de forma consistente (TypeORM, 2024).

Para garantir a disponibilidade da aplicação back-end em um ambiente de produção, será utilizado o PM2, um gerenciador de processos avançado para Node.js. A ferramenta atua como um supervisor do processo da aplicação, assegurando que ela permaneça em execução contínua (24/7). Suas principais responsabilidades no projeto são o reinício automático da aplicação em caso de falhas inesperadas e o gerenciamento do modo *cluster*¹⁴, que permite à aplicação utilizar todos os núcleos de CPU disponíveis no servidor, otimizando a performance e a capacidade de resposta sob carga (KEYMETRICS, 2025).

6.6.3 Banco de Dados

Para a persistência dos dados, foi selecionada uma solução de banco de dados relacional hospedada na nuvem da Amazon Web Services (AWS), como o Amazon RDS. A escolha pela AWS se justifica por sua alta disponibilidade, escalabilidade sob demanda e segurança, permitindo que a infraestrutura de dados cresça conforme a necessidade do negócio, ao mesmo tempo que se beneficia do

¹⁴ técnica de gerenciamento que permite executar múltiplas instâncias de uma aplicação distribuídas entre todos os núcleos de CPU do servidor, aumentando a performance e a disponibilidade.

Nível Gratuito (Free Tier) para viabilizar os custos iniciais do projeto.

6.6.4 Ferramentas de Apoio e Versionamento

O desenvolvimento será realizado no ambiente de desenvolvimento integrado (IDE) Visual Studio Code. Para a modelagem dos diagramas UML e do DER, foram utilizadas as ferramentas Mermaid.js e Draw.io. A prototipagem da interface foi feita no Figma. Por fim, todo o código-fonte do projeto será versionado com o Git e gerenciado na plataforma GitHub, garantindo um controle de versão seguro.

7 CRONOGRAMA DE EXECUÇÃO

O desenvolvimento da solução digital proposta será realizado em etapas, conforme o modelo incremental descrito na metodologia. A organização do cronograma tem como objetivo garantir entregas parciais e funcionais, possibilitando ajustes com base em validações constantes. O planejamento considera um período total de aproximadamente 20 semanas, com dedicação semanal do autor.

A seguir, apresenta-se quadro 4, que contém o cronograma dividido em fases:

QUADRO 4 – CRONOGRAMA DE EXECUÇÃO

Etapa	Descrição	Jul.	Ago.	Set.	Out.	Nov.	Dez.
Implementação do catálogo de produtos	Desenvolvimento das funcionalidades de exibição, cadastro, edição e filtro de produtos.	X	X				
Testes do catálogo e ajustes de interface	Realização de testes funcionais e correções baseadas no feedback obtido.		X				
Implementação do módulo de pedidos e entregas	Criação das funcionalidades de anotação de pedidos, controle de status e painel administrativo.			X	X		
Integração e testes dos módulos desenvolvidos	Integração entre os módulos de catálogo e pedidos. Validação da consistência e funcionalidade do sistema.				X		
Implementação da geração de relatórios e refino da aplicação	Desenvolvimento das funcionalidades de relatórios e aplicação de melhorias identificadas.					X	

Documentação e Preparação para Entrega	Documentação do sistema, backup do código-fonte, revisão geral e apresentação final.						X
--	--	--	--	--	--	--	---

FONTE: Elaborado pelo autor (2025).

8 REFERÊNCIAS

CAPTERRA. Pesquisa em e-commerce de flores: rapidez na entrega supera preço e atendimento, segundo consumidores. Plataforma online, 2025.

CNA BRASIL. Mão de obra pode limitar crescimento do setor de flores e plantas ornamentais em 2025. Comunicado do Sistema CNA/Senar-SP, 23 abr. 2025.

Disponível em:

<https://www.cnabrasil.org.br/noticias/mao-de-obra-pode-limitar-crescimento-do-setor-de-flores-e-plantas-ornamentais-em-2025>. Acesso em: 24 abr. 2025.

CRUZ, José Jovenildo da; BICUDO, Dhiego Augusto dos Santos. A arquitetura de informação aplicada no desenvolvimento de um catálogo digital em CD-ROM. In: **CONGRESSO NACIONAL DE EXTENSÃO UNIVERSITÁRIA**, 4.º; **ENCONTRO DE ATIVIDADES CIENTÍFICAS DA UNOPAR**, 13., 2010, Anais... Londrina: Unopar, 2010.

FERNANDES, Adriano de Oliveira. SICON: sistema de controle de vendas para produtos de catálogo. 2012. 88 f. Trabalho de Conclusão de Curso (Graduação em Bacharelado em Sistemas de Informação) – Universidade Estadual de Goiás, Anápolis, 2012. Disponível em:

<https://repositorio.ueg.br/jspui/bitstream/riueg/2474/2/MG%20024%200063-2012.pdf>. Acesso em: 15 mai. 2025.

FERREIRA, Simone Bacellar Leal; CHAUVEL, Marie Agnes; SILVEIRA, Denis Silva da. Usability of virtual organization's sites: um estudo de usabilidade de sites de empresas virtuais. **Produção**, v. 16, n. 2, p. 300-313, maio/ago. 2006. Acesso em: 29 mai. 2025.

FIELDING, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. 2000. 172 f. Tese (Doutorado em Ciência da Computação) – University of California, Irvine, 2000. Disponível em:

<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>. Acesso em: 24 abr. 2025.

FOREACH PARTNERS. Nest.js: A scalable and modular framework for building modern Node.js applications. 2025. Disponível em:

<https://foreachpartners.com/technology/nestjs>. Acesso em: 13 jun. 2025.

FOWLER, Martin. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2002.

GANNEY, Paul S.; CLARIDGE, Edwin. Incremental model/prototyping model. In: CLINICAL ENGINEERING. 2. ed. [S. l.]: Elsevier, 2020. Disponível em:

<https://www.sciencedirect.com/topics/computer-science/incremental-model>.

GEEKSFORGEES. Unified Modeling Language (UML) Class Diagrams – what is a class diagram?. GeeksforGeeks, 3 jan. 2025. Disponível em:

<https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams>.

Acesso em: 13 jun. 2025.

GLINZ, Martin. On Non-Functional Requirements. In: **15TH IEEE INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE (RE 2007)**. Anais... IEEE, 2007. p. 2–7. DOI: 10.1109/RE.2007.45.

HEUSER, Carlos Alberto. Projeto de banco de dados. 6. ed. Porto Alegre: Bookman, 2009.

IBM NS1 Connect. O que é hospedagem web (web hosting)? Think, IBM Brasil, 2023. Disponível em: <https://www.ibm.com/br-pt/topics/web-hosting>. Acesso em: 10 jul. 2025.

ILIS, Vitor. Mercado de flores no Brasil atingiu R\$ 10,9 bilhões em 2021. CNA Brasil, 19 set. 2022. Disponível em: <https://cnabrasil.org.br/noticias/mercado-de-flores-no-brasil-atingiu-r-10-9-bilhoes-em-2021>. Acesso em: 16 abr. 2025.

INSTITUTO BRASILEIRO DE FLORICULTURA – IBRAFLO. Perspectivas do mercado de flores no Brasil. 2025. Informação divulgada em artigo da Graciosa Digital, 2 jun. 2025.

KEYMETRICS. PM2: Advanced, production process manager for Node.js. 2025. Disponível em: <https://pm2.keymetrics.io/>. Acesso em: 26 jun. 2025.

KIM, J.; LEE, S. *Content management systems and navigation in digital catalogs*. **Information Systems Journal**, v. 31, n. 3, p. 479–498, 2021.

KOTLER, Philip; KARTAJAYA, Hermawan; SETIAWAN, Iwan. *Marketing 4.0: do tradicional ao digital*. Rio de Janeiro: Sextante, 2017.

LEONARDI, Marco. Responsabilidade civil dos provedores de conteúdo de internet. Jus Navigandi, 2023. p. 2.

LIMA, Samia Mendes de; QUEIROZ, Cyntia Tavares Marques de. MARKETING 4.0 E A MODA: PLATAFORMAS DE COMÉRCIO DIGITAL NO MERCADO DE MODA BRASILEIRO. In: **ENCONTRO DE INICIAÇÃO ACADÊMICA DA UNIFOR**, 24., 2018, Fortaleza. Anais... Fortaleza: Unifor, 2018. Acesso em: 6 jun. 2025.

LIU, Wei et al. Digital catalog design and its impact on user decision-making. **Journal of Retailing and Consumer Services**, v. 65, p. 102–118, 2022.

LOUREIRO, Ana C. R.; BARBOSA, Simony A. *A experiência do usuário e a usabilidade em interfaces digitais: fatores críticos para o engajamento online*. **Revista de Sistemas e Computação**, v. 12, n. 3, p. 56–70, 2022.

LUCIDCHART. Diagrama de caso de uso UML: o que é, como fazer e exemplos. Lucidchart. Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>. Acesso em: 13 jun. 2025.

LUCIDCHART. What is an ER diagram?. Lucidchart. Disponível em: <https://www.lucidchart.com/pages/er-diagrams>. Acesso em: 13 jun. 2025.

MARTINS, R.; ALMEIDA, P. A importância da arquitetura da informação em sistemas digitais. *Revista Brasileira de Ciência da Informação*, v. 18, n. 2, p. 112–124, 2022.

MELLO, L. T. C. de; et al. *Análise do lead time nos processos logísticos de uma rede varejista de flores*. 16 mai. 2025. *Revista Produção Online*, Florianópolis, v. 16, n. 4, p. 1237–1261, 2016. Acesso em: 29 mai. 2025.

MENIN, Rennan M. Tchuvaiff. Comercialização e logística da floricultura no Sudoeste do Paraná. 2019. 38 p. Monografia (Trabalho de Conclusão de Curso) – Universidade Tecnológica Federal do Paraná, Pato Branco, 2019. Disponível em: https://riut.utfpr.edu.br/jspui/bitstream/1/24593/1/PB_COAGR_2019_2_19.pdf. Acesso em: 6 jun. 2025.

MÉTRICAS DE ATENDIMENTO. Quais são os principais indicadores de sucesso no atendimento ao cliente? Zendesk, 22 mar. 2019. Disponível em: <https://www.zendesk.com.br/blog/indicadores-atendimento-cliente/>. Acesso em: 29 mai. 2025.

MUNDO CORPORATIVO. E-commerce de flores busca entregas rápidas com clientes mais exigentes. Portal UAI, Belo Horizonte, 02 abr. 2025. Disponível em: <https://www.em.com.br/mundo-corporativo/2025/04/7100144-e-commerce-de-flores-busca-entregas-rapidas-com-clientes-mais-exigentes.html>. Acesso em: 7 jun. 2025.

NESTJS. A progressive Node.js framework. 2025. Disponível em: <https://nestjs.com/>. Acesso em: 6 jul. 2025.

NEXT.JS. The React Framework for the Web. Vercel, 2025. Disponível em: <https://nextjs.org/>. Acesso em: 6 jul. 2025.

NIELSEN, Jakob; BUDIU, Raluca. *UX Strategy for Lean Startups: Faster User Research and Smarter UX Design*. San Francisco: New Riders, 2021.

NUVEMSHOP. Crie sua loja online e venda pela internet. São Paulo: Nuvemshop, 2025. Disponível em: <https://www.nuvemshop.com.br/>. Acesso em: 26 jun. 2025.

PRESSMAN, Roger S. Engenharia de software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

REACT. A JavaScript library for building user interfaces. Meta, 2025. Disponível em: <https://react.dev/>. Acesso em: 6 jul. 2025.

RIBEIRO, Danielly dos Santos. Sistema de gerenciamento de conteúdo: proposta de um catálogo bibliográfico 2.0 no WordPress. 2015. 68 p. Monografia (Graduação em Biblioteconomia) – Universidade de Brasília, Brasília, 2015. Disponível em: https://bdm.unb.br/bitstream/10483/25480/1/2015_DaniellyDosSantosRibeiro_tcc.pdf. Acesso em: 29 mai. 2025.

RICHARDSON, L.; RUBY, S. *RESTful Web APIs*. 2. ed. O'Reilly Media, 2020.

SHARP, Helen; ROGERS, Yvonne; PREECE, Jenny. Design de interação: além da interação homem-computador. 3. ed. Porto Alegre: Bookman, 2013.

SILVA, J.; ALMEIDA, F. Arquitetura cliente-servidor e seu papel em aplicações web modernas. **Revista de Sistemas e Tecnologia**, v. 14, n. 1, p. 25–38, 2022.

SILVA, Jeferson Luan da. Análise da Usabilidade em Sites de Comércio Eletrônico. 2023. 38 p. Monografia (Graduação em Tecnologia da Informação) – Universidade Federal Rural do Semi-Árido, Pau dos Ferros, 2023. 29 mai. 2025.

SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo: Pearson, 2019.

SOUZA, Clóvis. Depoimento sobre logística na Giuliana Flores. 2025. Em: E-commerce de flores busca entregas rápidas com clientes mais exigentes. *Graciosa Digital*, 2 jun. 2025.

SUPELETO, Adriano Fernando Bezerra. Destefani: aplicativo para catálogo e venda de produtos. 2022. Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal do Espírito Santo, Campus Alegre, Alegre, 2022. Disponível em: https://repositorio.ifes.edu.br/bitstream/handle/123456789/2596/TCC_Destefani_Adrino.pdf?sequence=1&isAllowed=y. Acesso em: 15 mai. 2025.

TAILWIND CSS. A utility-first CSS framework. Tailwind Labs, 2025. Disponível em: <https://tailwindcss.com/>. Acesso em: 9 jul. 2025.

TUTORIALPOINT. Incremental Model in Software Engineering. 2019. Disponível em: <https://www.tutorialpoint.net/2019/07/incremental-model-in-software-engineering.html>. Acesso em: 8 jul. 2025.

TYPEORM. TypeORM documentation. 2024. Disponível em: <https://typeorm.io/>. Acesso em: 8 jul. 2025.

UPLARN. Benefits of React JS for Front-end Development. Uplarn, 2025. Disponível em: <https://www.uplarn.com/benefits-of-react-js-for-front-end-development/>. Acesso em: 13 jun. 2025.