

**PAULA PRESENTES ERP**  
**DOCUMENTAÇÃO TÉCNICA DO SISTEMA**  
**Plataforma Web Integrada de Gestão Comercial e E-commerce**

Desenvolvido por  
Caroline Vitória Paneco

ASP.NET Core MVC  
Entity Framework Core  
PostgreSQL  
Tailwind CSS

Versão 1.0

2026

CAROLINE VITÓRIA PANECO

## **DOCUMENTAÇÃO TÉCNICA DO SISTEMA**

Plataforma Web Integrada de Gestão Comercial e E-commerce

Sistema web integrado para gestão comercial e operação de e-commerce, desenvolvido utilizando ASP.NET Core MVC, PostgreSQL e arquitetura MVC escalável.

Versão 1.0

2026

## RESUMO

O presente projeto tem como objetivo o desenvolvimento de um sistema ERP Web integrado a um ecommerce para a loja Paula Presentes, utilizando a arquitetura ASP.NET MVC e banco de dados PostgreSQL. A plataforma foi criada para auxiliar no gerenciamento empresarial, permitindo o controle de produtos, estoque, vendas, movimentações financeiras e exibição de itens na loja virtual em um único ambiente integrado. O sistema conta com funcionalidades administrativas para cadastro e gerenciamento de produtos, controle de estoque, registro de vendas, controle de entradas e saídas do caixa, relatórios gerenciais e armazenamento de imagens em nuvem utilizando Supabase Storage. Além disso, a aplicação apresenta interface web responsiva, visando melhor experiência do usuário em diferentes dispositivos. O desenvolvimento do projeto permitiu a aplicação prática de conceitos de Engenharia de Software, arquitetura MVC, banco de dados relacionais, desenvolvimento web e integração com serviços em nuvem, proporcionando maior organização, automação e eficiência nos processos comerciais da empresa.

**Palavras-chave:** ERP Web; Ecommerce; ASP.NET MVC; PostgreSQL; Gestão Comercial; Engenharia de Software.

## SUMÁRIO

1	INTRODUÇÃO .....	7
2	OBJETIVOS .....	8
2.1	OBJETIVO GERAL .....	8
2.2	OBJETIVOS ESPECÍFICOS .....	8
2.3	PRINCIPAIS USUÁRIOS .....	9
2.3.1	Administrador .....	9
2.3.2	Funcionários .....	9
2.3.3	Clientes .....	9
3	TECNOLOGIAS UTILIZADAS .....	10
3.1	FRONT-END .....	10
3.1.1	HTML5 .....	10
3.1.2	CSS3 .....	10
3.1.3	Tailwind CSS .....	11
3.1.4	JavaScript .....	11
3.1.5	Photoshop .....	11
3.2	BACK-END .....	12
3.2.1	ASP.NET MVC .....	12
3.2.2	Linguagem C# .....	12
3.2.3	Entity Framework Core .....	13
3.3	BANCO DE DADOS .....	13
3.3.1	BR Modelo .....	13
3.3.2	PostgreSQL .....	14
3.4	ARMAZENAMENTO EM NUVEM .....	14
3.4.1	Supabase Storage .....	14
3.5	HOSPEDAGEM E DEPLOY .....	15
3.5.1	Railway .....	15
3.6	VERSIONAMENTO E CONTROLE DE CÓDIGO .....	15
3.6.1	GitHub .....	15
3.7	AMBIENTE DE DESENVOLVIMENTO .....	16
3.7.1	Visual Studio 2022 .....	16
4	METODOLOGIA E TRILHA DE DESENVOLVIMENTO .....	17
4.1	LEVANTAMENTO DE REQUISITOS .....	17

4.2	ANÁLISE DE REQUISITOS .....	18
4.2.1	Requisitos Funcionais.....	18
4.2.2	Requisitos Não Funcionais .....	19
4.3	MODELAGEM DE BANCO DE DADOS .....	20
4.3.1	Modelo Conceitual .....	20
4.3.2	Modelo Lógico.....	21
4.3.3	Modelo Relacional .....	24
4.3.4	Integridade e Relacionamentos .....	24
4.3.5	Integração com Entity Framework Core.....	24
4.3.6	Importância da Modelagem no Projeto .....	25
4.4	MODELAGEM UML DO SISTEMA .....	25
4.4.1	Diagrama de Casos de Uso.....	26
4.4.2	Diagrama de Classes.....	28
4.4.3	Diagrama de Sequência .....	30
5	DESIGN DIGITAL E UX/UI .....	35
5.1	Prototipação Visual .....	35
5.2	Conceito Visual da Plataforma .....	36
5.3	Psicologia das Cores .....	37
5.3.1	Cor Primária.....	37
5.3.2	Cor Secundária .....	38
5.3.3	Cores de Fundo e Superfície .....	38
5.3.4	Cores de Texto.....	39
5.4	Tipografia e Hierarquia Visual .....	39
5.5	Experiência do Usuário (UX).....	41
5.6	Responsividade e Mobile First .....	41
5.7	Importância do Design no Projeto.....	42
6	DESENVOLVIMENTO DO SISTEMA.....	43
6.1	TESTES E QA .....	43
6.1.1	Testes Funcionais .....	43
6.1.2	Testes de Integração.....	44
6.1.3	Testes de Interface e Usabilidade .....	44
6.1.4	Testes de Responsividade .....	44
6.1.5	Correções e Ajustes (Debugging) .....	44

6.1.6	Importância da Etapa de QA.....	45
6.2	DEPLOY E HOSPEDAGEM.....	45
6.2.1	Versionamento do Código-Fonte .....	45
6.2.2	Processo de Deploy.....	45
6.2.3	Banco de Dados em Ambiente Cloud .....	45
6.2.4	Armazenamento de Arquivos em Nuvem.....	46
6.2.5	Integração entre Serviços .....	46
6.2.6	Considerações sobre o Ambiente de Produção .....	46
7	FUNCIONALIDADES DO SISTEMA.....	47
7.1	ECOMMERCE .....	47
7.2	ADMINISTRAÇÃO .....	51
7.3	INTEGRAÇÃO GERAL DO SISTEMA .....	60
7.4	RESPONSIVIDADE .....	60
7.5	SEGURANÇA E INTEGRIDADE DOS DADOS .....	60
8	CONCLUSÃO .....	61
9	REFERÊNCIAS.....	62

## **1 INTRODUÇÃO**

A transformação digital vem impactando diretamente o setor comercial, tornando os sistemas de gestão cada vez mais essenciais para empresas que buscam organização, eficiência e competitividade no mercado. Nesse contexto, lojas físicas e virtuais necessitam de soluções integradas capazes de centralizar informações relacionadas a produtos, estoque, vendas, finanças e atendimento ao cliente.

Além disso, o crescimento do ecommerce ampliou a necessidade de plataformas modernas, responsivas e conectadas a bancos de dados e serviços em nuvem, permitindo maior escalabilidade e segurança das informações.

Diante desse cenário, foi desenvolvido o projeto ERP WEB para gestão empresarial, uma plataforma web de gestão comercial integrada a um ecommerce, com funcionalidades administrativas e operacionais voltadas para o gerenciamento empresarial da loja física Paula Presentes.

O sistema foi desenvolvido utilizando ASP.NET MVC, PostgreSQL e Supabase Storage, aplicando conceitos de Engenharia de Software, Modelagem de Banco de Dados, Técnicas de Programação e Design Digital.

## 2 OBJETIVOS

### 2.1 OBJETIVO GERAL

Desenvolver uma plataforma ERP Web integrada a um ecommerce, capaz de centralizar processos administrativos, comerciais e operacionais da loja Paula Presentes, proporcionando maior controle sobre produtos, estoque, vendas e movimentações financeiras. Além de criar uma nova possibilidade de venda criando uma loja virtual aumentando seu raio de clientes.

### 2.2 OBJETIVOS ESPECÍFICOS

O projeto tem como objetivo desenvolver uma plataforma web responsiva e integrada, capaz de auxiliar no gerenciamento comercial e administrativo da loja Paula Presentes. O sistema busca proporcionar maior controle sobre os produtos cadastrados, permitindo a organização das informações relacionadas a categorias, marcas, cores, preços, imagens e estoque.

Além disso, a aplicação visa automatizar processos comerciais por meio do controle de vendas e da atualização automática do estoque, contribuindo para maior praticidade e redução de erros operacionais. O sistema também possui como finalidade oferecer recursos de gerenciamento financeiro, permitindo o registro de entradas, saídas e movimentações de caixa, auxiliando na análise e organização financeira da empresa.

Outro objetivo importante do projeto é integrar o armazenamento de imagens em nuvem, garantindo maior segurança, disponibilidade e escalabilidade para os arquivos da aplicação. Durante o desenvolvimento, também foi aplicada a arquitetura MVC, promovendo melhor organização do código, separação de responsabilidades e facilidade de manutenção do sistema.

O projeto busca ainda utilizar banco de dados relacional para armazenamento estruturado das informações, assegurando integridade e confiabilidade dos dados. Por fim, a plataforma foi desenvolvida com foco em experiência do usuário, apresentando interface moderna, intuitiva e responsiva, compatível com diferentes dispositivos e tamanhos de tela.

## 2.3 PRINCIPAIS USUÁRIOS

O sistema foi projetado para atender diferentes perfis de usuários dentro do ambiente comercial da empresa.

### 2.3.1 Administrador

Responsável pelo gerenciamento completo do sistema, incluindo produtos, estoque, vendas, movimentações financeiras e relatórios.

### 2.3.2 Funcionários

Usuários responsáveis pelo cadastro de produtos, registro de vendas e atualização das informações comerciais.

### 2.3.3 Clientes

Usuários externos que acessam o ecommerce para visualizar produtos e realizar compras.

### **3 TECNOLOGIAS UTILIZADAS**

O desenvolvimento do projeto envolveu a utilização de diferentes tecnologias voltadas para desenvolvimento web, banco de dados, armazenamento em nuvem, estilização de interfaces e hospedagem da aplicação. A escolha das ferramentas foi realizada considerando fatores como desempenho, escalabilidade, organização do código, integração entre sistemas e experiência do usuário.

A seguir, são apresentadas as principais tecnologias utilizadas no projeto e suas respectivas finalidades.

#### **3.1 FRONT-END**

O Front-end corresponde à camada visual do sistema, responsável pela interação direta com o usuário. Essa etapa do projeto teve como principal objetivo desenvolver uma interface moderna, elegante, intuitiva e responsiva, proporcionando uma experiência agradável tanto em computadores quanto em dispositivos móveis.

Para isso, foram utilizadas tecnologias voltadas à estruturação das páginas, estilização visual, responsividade e interatividade da aplicação.

##### **3.1.1 HTML5**

O HTML5 foi utilizado como base estrutural de toda a aplicação web. Sua principal função foi organizar semanticamente os elementos das páginas, permitindo a construção das telas do ecommerce e do painel administrativo de forma organizada e padronizada.

Com o HTML5 foram desenvolvidos os formulários de cadastro, menus de navegação, páginas de produtos, listagens, tabelas administrativas, seções institucionais e componentes utilizados na loja virtual. A tecnologia também contribuiu para melhor acessibilidade e compatibilidade entre navegadores.

##### **3.1.2 CSS3**

O CSS3 foi responsável pela estilização visual do sistema, permitindo a definição de cores, tamanhos, alinhamentos, animações, efeitos visuais e responsividade das páginas.

Sua utilização foi essencial para transformar a estrutura criada em HTML em uma interface moderna e visualmente agradável. Além disso, o CSS3 auxiliou na

criação de transições suaves, melhor organização visual dos elementos e adaptação automática para diferentes tamanhos de tela.

### **3.1.3 Tailwind CSS**

O Tailwind CSS foi utilizado como framework principal de estilização do projeto. Sua adoção permitiu maior produtividade durante o desenvolvimento das interfaces, facilitando a criação de layouts modernos e responsivos através de classes utilitárias.

A utilização do framework também contribuiu significativamente para a padronização visual do sistema, mantendo consistência entre componentes, espaçamentos, tipografia e organização das telas. Outro fator importante foi a facilidade de adaptação para dispositivos móveis, permitindo que o sistema apresentasse boa experiência de navegação em smartphones, tablets e computadores.

### **3.1.4 JavaScript**

O JavaScript foi utilizado para implementação das funcionalidades interativas da aplicação. Sua utilização permitiu tornar a interface mais dinâmica e melhorar a experiência do usuário durante a navegação.

Entre as funcionalidades desenvolvidas com JavaScript estão recursos de pré-visualização de imagens antes do upload, contadores dinâmicos, interações visuais, manipulação de elementos da página e melhorias relacionadas à usabilidade do sistema.

A linguagem também contribuiu para reduzir recarregamentos desnecessários das páginas, proporcionando maior fluidez durante a utilização da plataforma.

### **3.1.5 Photoshop**

O Adobe Photoshop foi utilizado durante a etapa de prototipação visual e desenvolvimento da identidade visual do sistema. Antes da implementação prática das telas, foi realizado um processo de planejamento visual com objetivo de definir o conceito estético da plataforma.

Durante essa etapa foram estudados elementos como paleta de cores, tipografia, organização visual das informações, contraste, identidade da marca e experiência do usuário. Também foram aplicados conceitos relacionados à psicologia

das cores, buscando transmitir elegância, sofisticação, confiança e modernidade, características alinhadas à proposta da loja Paula Presentes.

O processo de design teve papel fundamental na construção da identidade visual do projeto, servindo como base para a implementação das interfaces no sistema web.

## 3.2 BACK-END

O Back-end corresponde à camada responsável pelo funcionamento interno do sistema, processamento das informações, regras de negócio e comunicação com o banco de dados.

Essa parte do projeto concentrou toda a lógica necessária para funcionamento das funcionalidades administrativas, gerenciamento de produtos, controle financeiro, estoque, vendas e integração com serviços externos.

### 3.2.1 ASP.NET MVC

O ASP.NET MVC foi utilizado como framework principal para desenvolvimento da aplicação web. A arquitetura MVC (Model-View-Controller) permitiu dividir o sistema em camadas organizadas, facilitando a manutenção, escalabilidade e reutilização de código.

A separação entre Models, Views e Controllers contribuiu para um desenvolvimento mais estruturado, permitindo que cada parte do sistema possuísse responsabilidades específicas. Os Models ficaram responsáveis pelas entidades e estrutura de dados, os Controllers pelo processamento das requisições e regras de negócio, enquanto as Views ficaram responsáveis pela renderização das interfaces.

A utilização do ASP.NET MVC também proporcionou maior organização do projeto, integração eficiente com banco de dados e melhor gerenciamento das rotas da aplicação.

### 3.2.2 Linguagem C#

A linguagem C# foi utilizada como principal linguagem de programação do sistema. Com ela foram desenvolvidas todas as regras de negócio e funcionalidades internas da aplicação.

Entre os recursos implementados estão cadastro e gerenciamento de produtos, controle de estoque, movimentações financeiras, controle de vendas, upload de imagens, gerenciamento administrativo e integração com armazenamento em nuvem.

O C# foi escolhido devido à sua forte integração com o ecossistema .NET, além de oferecer segurança, organização orientada a objetos, boa performance e facilidade de manutenção do código.

### **3.2.3 Entity Framework Core**

O Entity Framework Core foi utilizado como ORM (Object Relational Mapping), permitindo a integração entre a aplicação e o banco de dados de maneira mais simplificada e organizada.

Sua utilização possibilitou manipular tabelas e relacionamentos utilizando classes C#, reduzindo significativamente a necessidade de comandos SQL complexos durante o desenvolvimento.

Além disso, o Entity Framework Core auxiliou na criação das migrations, gerenciamento das entidades, controle dos relacionamentos entre tabelas e manutenção da estrutura do banco de dados ao longo da evolução do sistema.

## **3.3 BANCO DE DADOS**

O banco de dados foi responsável pelo armazenamento estruturado das informações do sistema, garantindo persistência, organização e integridade dos dados utilizados pela aplicação.

### **3.3.1 BR Modelo**

Durante a etapa inicial do projeto, a ferramenta BR Modelo foi utilizada para auxiliar na modelagem conceitual e lógica do banco de dados. A utilização da ferramenta teve como objetivo estruturar corretamente as entidades, relacionamentos e regras da base de dados antes do início do desenvolvimento da aplicação.

Com o BR Modelo foi possível planejar a organização das tabelas, definir chaves primárias e estrangeiras, analisar relacionamentos entre entidades e estruturar o funcionamento das informações dentro do sistema. Essa etapa foi fundamental para garantir maior organização, escalabilidade e integridade dos dados ao longo do desenvolvimento do projeto.

A modelagem realizada auxiliou diretamente na construção das entidades utilizadas posteriormente no Entity Framework Core, facilitando a transição entre o planejamento teórico do banco de dados e sua implementação prática dentro da aplicação.

### **3.3.2 PostgreSQL**

O PostgreSQL foi utilizado como banco de dados relacional principal do projeto. A escolha dessa tecnologia ocorreu devido à sua estabilidade, segurança, confiabilidade e capacidade de lidar com aplicações de médio e grande porte.

No banco de dados foram armazenadas informações relacionadas aos produtos, imagens, estoque, vendas, movimentações financeiras, categorias e demais registros administrativos do sistema.

A integração entre PostgreSQL e Entity Framework Core permitiu maior facilidade na manipulação dos dados e melhor organização das entidades da aplicação, tornando o desenvolvimento mais produtivo e escalável.

## **3.4 ARMAZENAMENTO EM NUVEM**

O armazenamento em nuvem foi implementado para solucionar limitações relacionadas ao armazenamento local de arquivos em ambientes cloud.

### **3.4.1 Supabase Storage**

O Supabase Storage foi utilizado para armazenamento das imagens dos produtos em nuvem. Inicialmente, as imagens eram armazenadas localmente dentro da pasta wwwroot do projeto, porém essa abordagem apresentou limitações em ambientes de hospedagem cloud, onde arquivos locais podem ser temporários e serem perdidos após reinicializações do servidor.

Com a implementação do Supabase Storage, as imagens passaram a ser armazenadas de forma permanente na nuvem, permitindo maior segurança, disponibilidade e escalabilidade.

Além disso, a solução possibilitou melhor compatibilidade com dispositivos móveis, acesso remoto às imagens e maior estabilidade durante o deploy da aplicação. As URLs geradas pelo armazenamento em nuvem passaram a ser salvas

diretamente no banco de dados, permitindo carregamento dinâmico das imagens dentro do sistema.

### 3.5 HOSPEDAGEM E DEPLOY

A hospedagem da aplicação foi realizada em ambiente cloud, permitindo publicação online do sistema e acesso remoto à plataforma.

#### 3.5.1 Railway

A plataforma Railway foi utilizada para hospedagem e deploy da aplicação web. Sua utilização permitiu integração direta com o GitHub, automatizando processos de publicação e atualização do sistema.

Além disso, a plataforma ofereceu suporte para hospedagem do banco de dados PostgreSQL, gerenciamento de variáveis de ambiente e deploy contínuo da aplicação.

A utilização de uma infraestrutura cloud contribuiu para maior escalabilidade, estabilidade e disponibilidade do sistema em ambiente de produção.

### 3.6 VERSIONAMENTO E CONTROLE DE CÓDIGO

#### 3.6.1 GitHub

O GitHub foi utilizado como plataforma de versionamento e gerenciamento do código-fonte do projeto. Sua utilização permitiu armazenar o sistema em nuvem, acompanhar alterações realizadas durante o desenvolvimento e manter maior segurança do código.

A plataforma também auxiliou na organização do desenvolvimento, permitindo controle de versões e integração com ferramentas de deploy utilizadas na aplicação.

Além disso, o GitHub teve papel importante na manutenção do projeto, possibilitando recuperação de versões anteriores e acompanhamento da evolução do sistema ao longo do desenvolvimento.

## 3.7 AMBIENTE DE DESENVOLVIMENTO

### 3.7.1 Visual Studio 2022

O Visual Studio 2022 foi utilizado como ambiente de desenvolvimento principal do projeto. A IDE forneceu ferramentas essenciais para programação, organização do código, gerenciamento do banco de dados e depuração da aplicação.

Entre os principais recursos utilizados estão IntelliSense, debugging, gerenciamento de pacotes NuGet, integração com GitHub e ferramentas de compilação da plataforma .NET.

Sua utilização contribuiu significativamente para produtividade durante o desenvolvimento e organização geral do projeto.

## **4 METODOLOGIA E TRILHA DE DESENVOLVIMENTO**

O desenvolvimento do sistema ERP WEB foi realizado seguindo conceitos das aulas do primeiro semestre de Engenharia de Software, Desenvolvimento WEB, Modelagem de Banco de Dados, Design Digital e Técnicas de Programação, buscando estruturar uma aplicação moderna, escalável e alinhada às necessidades operacionais de um ecommerce com gestão empresarial integrada.

Durante o projeto, foi adotada uma metodologia baseada em etapas progressivas de análise, planejamento, modelagem, prototipação visual, desenvolvimento, testes e implantação. Cada fase teve como objetivo garantir maior organização do sistema, redução de falhas e melhor qualidade final da aplicação.

Além da implementação técnica, também houve preocupação com experiência do usuário, identidade visual da marca e escalabilidade futura do sistema, permitindo que a plataforma pudesse evoluir futuramente para um ERP ainda mais robusto.

### **4.1 LEVANTAMENTO DE REQUISITOS**

O desenvolvimento do projeto teve início com a etapa de levantamento de requisitos, considerada uma das fases mais importantes dentro da Engenharia de Software. Nesse momento, foi realizado um estudo sobre o funcionamento operacional da loja Paula Presentes, buscando compreender as principais dificuldades enfrentadas nos processos administrativos e comerciais.

Durante essa etapa, foram analisadas necessidades relacionadas ao controle de estoque, cadastro de produtos, organização financeira, gerenciamento de vendas e exposição dos produtos no ecommerce. Também foi observado que a ausência de centralização das informações dificultava o gerenciamento dos dados e aumentava a possibilidade de erros operacionais.

Além das necessidades técnicas, também foram analisados fatores estratégicos como custo de desenvolvimento, tempo necessário para implementação do sistema, facilidade de utilização pelos usuários e possibilidades de crescimento futuro da plataforma.

Outro ponto importante do levantamento de requisitos foi compreender o posicionamento visual da marca Paula Presentes. Desde o início do projeto, a proposta era desenvolver uma plataforma que transmitisse sofisticação, modernidade e elegância, mas sem perder a sensação de acessibilidade e proximidade com o

cliente. Essa visão influenciou diretamente tanto o design quanto as decisões técnicas do sistema.

A partir dessa análise inicial, foi possível definir os principais módulos que fariam parte do projeto, estruturando as funcionalidades do ecommerce e do painel administrativo.

## 4.2 ANÁLISE DE REQUISITOS

Após o levantamento inicial das necessidades do sistema, foi realizada a etapa de análise de requisitos, responsável por transformar as necessidades identificadas em funcionalidades práticas e regras técnicas para desenvolvimento da aplicação.

Durante essa fase, os requisitos foram separados entre funcionais e não funcionais, permitindo maior organização do projeto.

### 4.2.1 Requisitos Funcionais

Após o levantamento inicial das necessidades do sistema, foi realizada a etapa de análise de requisitos, responsável por transformar as necessidades identificadas em funcionalidades práticas e regras técnicas para desenvolvimento da aplicação.

Durante essa fase, os requisitos foram separados entre funcionais e não funcionais, permitindo maior organização do projeto.

Os requisitos funcionais envolveram todas as operações que o sistema deveria executar, incluindo cadastro de produtos, gerenciamento de imagens, controle de estoque, registro de vendas, movimentações financeiras, gerenciamento administrativo e exibição dos produtos no ecommerce.

Também foram definidos recursos relacionados à atualização automática do estoque após vendas, armazenamento das imagens em nuvem, gerenciamento de categorias, listagem dinâmica de produtos e integração entre os módulos administrativos e comerciais da plataforma.

- Ecommerce
  - Visualizar produtos com nome, preço de venda, imagem, marca, descrição e cores disponíveis
  - Controle automático de estoque na exibição dos produtos
  - Adicionar produtos ao carrinho de compras

- Finalizar compra com envio das informações do carrinho para o WhatsApp
- Cadastro e login de usuários
- Página institucional com a história da loja
- **Administração**
  - Registrar nova venda
  - Cadastrar novo produto
  - Consultar produtos
  - Consultar estoque
  - Controle de movimentações financeiras (entradas e saídas)
  - Dashboard com visão geral do sistema
  - Filtro de dados por data
  - Dashboard de categorias
  - Relatório financeiro
  - Cadastro de novos clientes
  - Consulta de clientes cadastrados
  - Exibição de aniversariantes do dia
  - Registrar nova viagem
  - Registrar gastos de viagens
  - Consultar histórico de viagens

#### **4.2.2 Requisitos Não Funcionais**

Já os requisitos não funcionais envolveram aspectos relacionados à qualidade do sistema, desempenho e experiência do usuário. Entre eles, destacam-se responsividade mobile, segurança dos dados, performance de carregamento, organização visual das páginas, integração cloud, estabilidade da aplicação e facilidade de utilização.

Outro fator importante dessa etapa foi a preocupação com escalabilidade. Mesmo sendo inicialmente um projeto acadêmico, o sistema foi planejado para suportar futuras expansões, permitindo implementação de novos módulos administrativos, melhorias de segurança e crescimento da estrutura de dados.

#### **4.2.3 REQUISITOS NÃO FUNCIONAIS**

- Sistema deve ser responsivo para dispositivos móveis, tablets e desktops

- Sistema deve garantir bom desempenho nas operações
- Sistema deve manter integridade e consistência dos dados
- Sistema deve utilizar armazenamento em nuvem para imagens
- Sistema deve seguir arquitetura MVC
- Sistema deve possuir autenticação de usuários
- Sistema deve ter interface intuitiva e de fácil uso
- Sistema deve ser compatível com navegadores modernos
- Sistema deve possuir separação entre front-end, back-end e banco de dados
- Sistema deve garantir segurança básica no controle de acesso

### 4.3 MODELAGEM DE BANCO DE DADOS

Após a etapa de levantamento e análise de requisitos do sistema, iniciou-se o processo de modelagem do banco de dados, responsável por estruturar, organizar e relacionar todas as informações utilizadas pela aplicação Paula Presentes ERP.

A modelagem de dados foi uma das etapas mais importantes do projeto, pois serviu como base para toda a construção do sistema, permitindo visualizar previamente o funcionamento das entidades, relacionamentos e regras de negócio antes do desenvolvimento prático da aplicação.

Inicialmente, foi utilizada a ferramenta BR Modelo para criação da modelagem conceitual e lógica do banco de dados. Essa etapa auxiliou na organização estrutural do projeto, permitindo compreender como os dados seriam armazenados, relacionados e manipulados dentro da aplicação.

Durante o processo de modelagem, foram analisadas as necessidades operacionais da loja, possibilitando transformar processos reais de negócio em estruturas organizadas de banco de dados. A partir disso, foram definidas entidades relacionadas ao controle de produtos, imagens, vendas, estoque, movimentações financeiras e gerenciamento administrativo do sistema.

#### 4.3.1 Modelo Conceitual

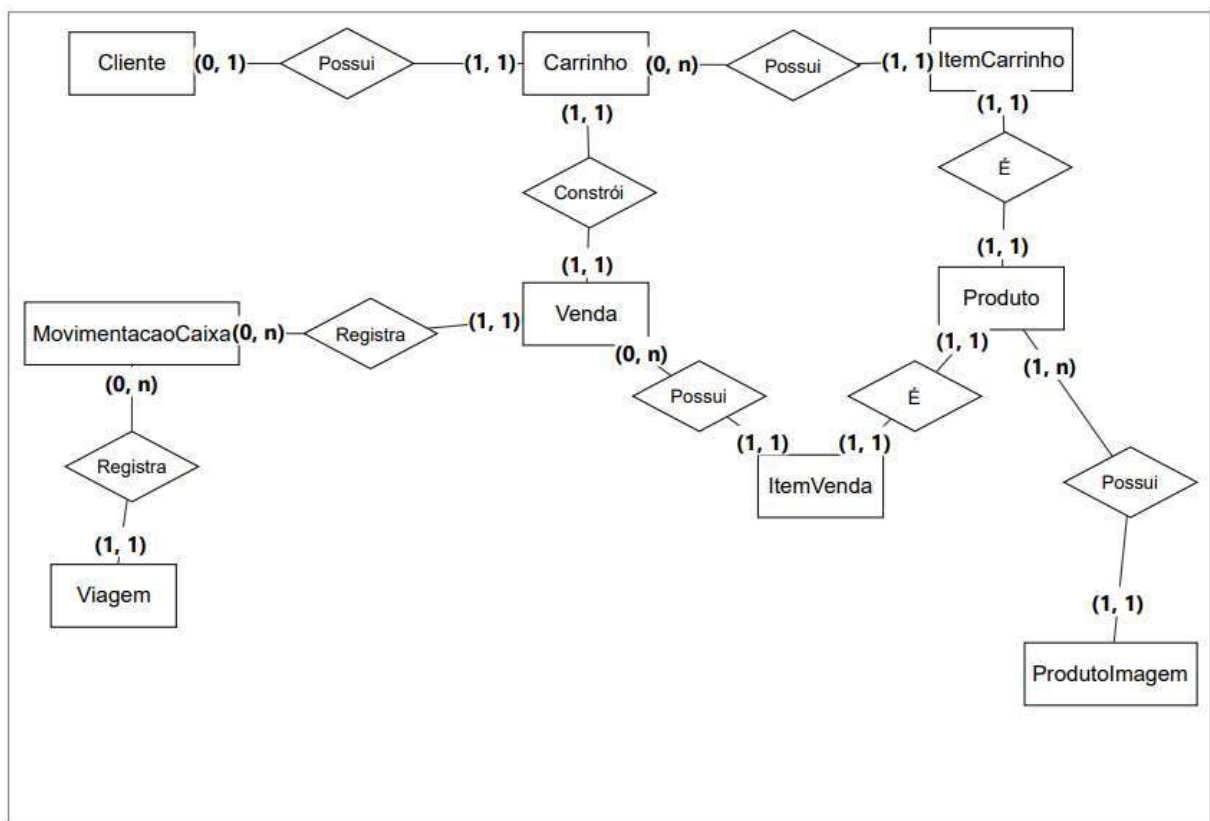
A primeira etapa da modelagem foi o desenvolvimento do modelo conceitual, responsável por representar de forma abstrata as principais entidades do sistema e seus relacionamentos.

Nessa fase, o foco principal esteve na compreensão das regras de negócio da aplicação, sem preocupação direta com implementação técnica ou linguagem de banco de dados.

Além das entidades, também foram definidos os relacionamentos entre elas, permitindo compreender como as informações interagem dentro do sistema.

Essa etapa foi essencial para validar a estrutura geral da aplicação antes da implementação técnica do banco de dados.

Figura 1 – Modelo Conceitual



Fonte: BR Modelo

#### 4.3.2 Modelo Lógico

Após a definição do modelo conceitual, foi desenvolvido o modelo lógico do banco de dados, responsável por transformar as entidades abstratas em estruturas mais próximas da implementação real.

Nessa etapa, foram definidos:

- atributos das entidades;
- tipos de relacionamento;
- cardinalidades;

- chaves primárias;
- chaves estrangeiras;
- regras de integridade.

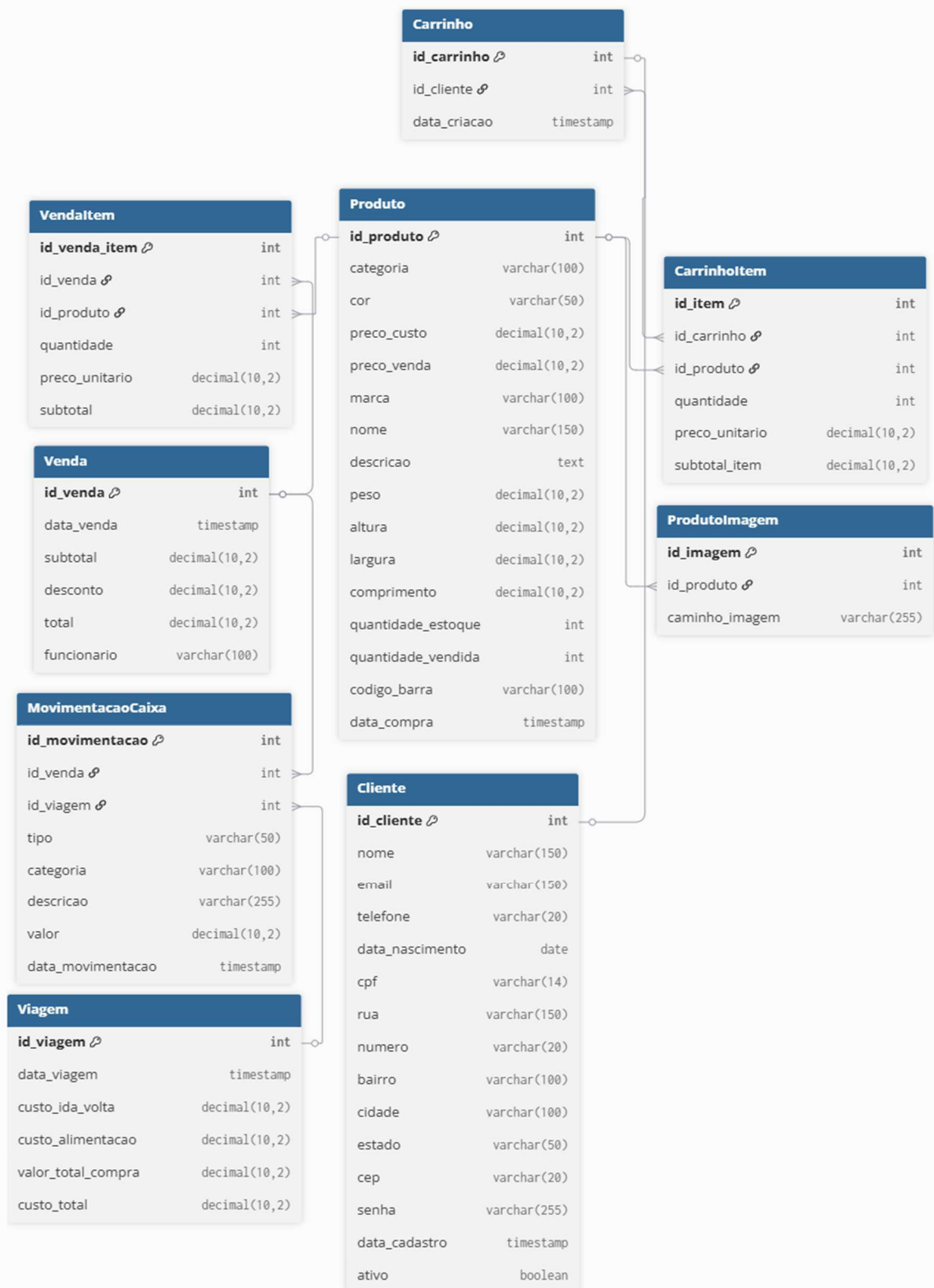
A modelagem lógica permitiu estruturar corretamente os relacionamentos entre as tabelas, garantindo organização e consistência dos dados armazenados.

Por exemplo, foi definido que um produto poderia possuir múltiplas imagens, estabelecendo um relacionamento de um para muitos entre as tabelas Produto e ProdutoImagem.

Também foram estruturados relacionamentos entre vendas e itens vendidos, permitindo que uma única venda armazenasse múltiplos produtos relacionados.

Essa organização foi fundamental para reduzir redundâncias e melhorar a integridade das informações da aplicação.

Figura 2 – Modelo Lógico



### **4.3.3 Modelo Relacional**

Após a conclusão da modelagem lógica, foi desenvolvida a estrutura relacional do banco de dados, já voltada para implementação prática utilizando PostgreSQL.

Nessa etapa, as entidades passaram a ser representadas como tabelas relacionais dentro do banco de dados, contendo colunas, tipos de dados e relacionamentos estruturados através de chaves estrangeiras.

A modelagem relacional permitiu transformar a estrutura planejada em uma base sólida para integração com o sistema ASP.NET MVC e com o Entity Framework Core.

A utilização do modelo relacional contribuiu para maior segurança, organização e desempenho no gerenciamento das informações do sistema.

### **4.3.4 Integridade e Relacionamentos**

Durante a modelagem do banco de dados, também foram implementadas regras de integridade para garantir consistência e confiabilidade das informações armazenadas.

As chaves primárias foram utilizadas para identificação única dos registros, enquanto as chaves estrangeiras garantiram o relacionamento correto entre as tabelas.

Além disso, foram aplicadas regras de cardinalidade para controlar como as entidades poderiam se relacionar dentro do sistema.

Essas definições ajudaram a evitar inconsistências, duplicidade de dados e problemas estruturais durante o funcionamento da aplicação.

### **4.3.5 Integração com Entity Framework Core**

Após a finalização da modelagem, o banco de dados foi integrado ao Entity Framework Core, ORM utilizado no projeto para realizar comunicação entre a aplicação e o PostgreSQL.

Com essa integração, as tabelas do banco passaram a ser representadas através de classes C#, permitindo manipulação dos dados de forma orientada a objetos.

O uso do Entity Framework Core facilitou significativamente o desenvolvimento do sistema, permitindo:

- criação de migrations;
- gerenciamento automático das entidades;
- manipulação simplificada dos relacionamentos;
- consultas integradas;
- maior produtividade no desenvolvimento.

Além disso, essa abordagem tornou o projeto mais organizado, escalável e alinhado às práticas modernas de desenvolvimento de software.

#### **4.3.6 Importância da Modelagem no Projeto**

A etapa de modelagem de banco de dados teve papel fundamental no desenvolvimento do Paula Presentes ERP, pois permitiu planejar toda a estrutura da aplicação antes do início da implementação prática.

A utilização do BR Modelo contribuiu diretamente para visualização dos relacionamentos, compreensão das regras de negócio e organização das entidades do sistema.

Esse processo reduziu falhas estruturais futuras, melhorou a organização da aplicação e facilitou tanto o desenvolvimento quanto a manutenção do sistema ao longo do projeto.

#### **4.4 MODELAGEM UML DO SISTEMA**

Durante o desenvolvimento do Paula Presentes ERP, também foram utilizadas modelagens UML (Unified Modeling Language), com o objetivo de representar visualmente o funcionamento, estrutura e interação dos componentes do sistema.

A utilização da UML contribuiu para melhor compreensão das regras de negócio, organização da arquitetura da aplicação e documentação técnica do projeto, auxiliando tanto no desenvolvimento quanto na manutenção futura do sistema.

Os principais diagramas desenvolvidos foram:

- Diagrama de Casos de Uso;
- Diagrama de Classes;
- Diagrama de Sequência.

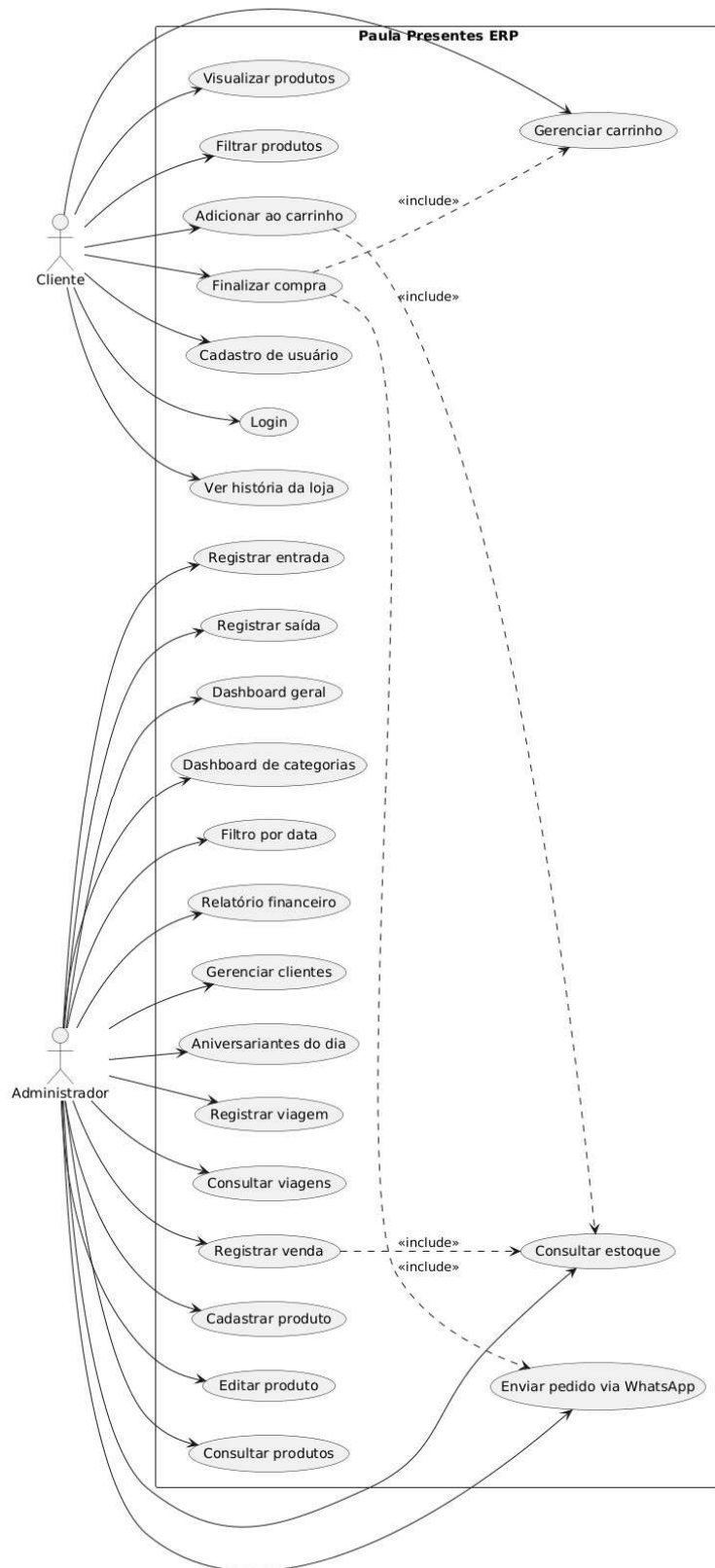
Esses diagramas permitiram representar diferentes perspectivas da aplicação, abrangendo desde a interação dos usuários até a estrutura interna das entidades e fluxo de execução das funcionalidades do sistema.

#### **4.4.1 Diagrama de Casos de Uso**

O diagrama de casos de uso foi utilizado para representar as funcionalidades disponíveis no sistema e a interação entre os usuários e a aplicação.

Através desse diagrama, foi possível identificar os principais atores do sistema, como administrador e cliente, além das operações realizadas por cada perfil dentro da plataforma.

Figura 3 – Caso de uso



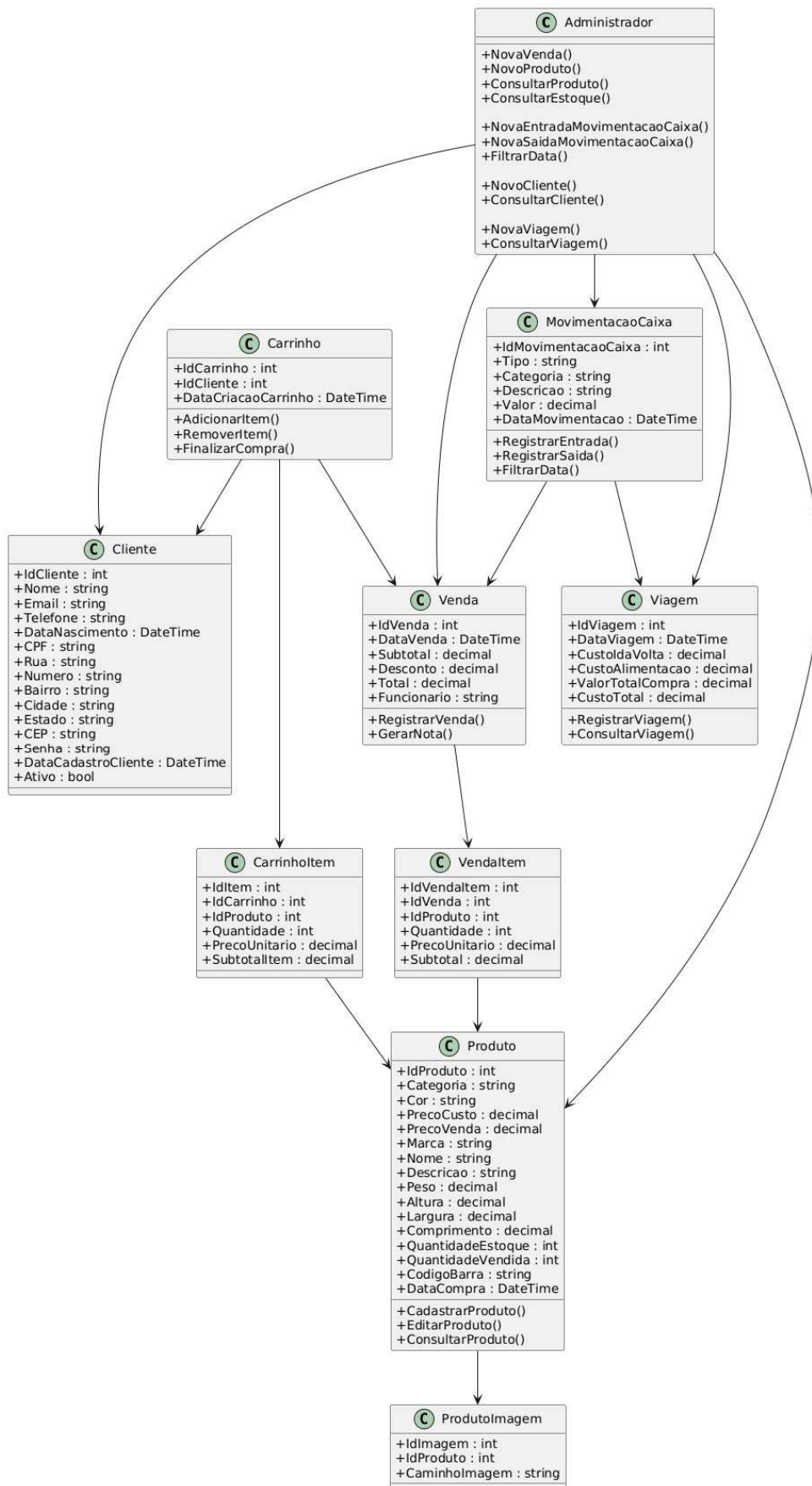
Esse diagrama contribuiu para melhor entendimento das funcionalidades do sistema e serviu como base para definição dos requisitos funcionais da aplicação.

#### **4.4.2 Diagrama de Classes**

O diagrama de classes foi desenvolvido para representar a estrutura estática do sistema, demonstrando as entidades, atributos, relacionamentos e responsabilidades das classes da aplicação.

A modelagem das classes foi baseada na arquitetura ASP.NET MVC e na estrutura do banco de dados PostgreSQL, permitindo manter alinhamento entre regras de negócio e persistência das informações.

Figura 4 – Diagrama de classe



Além disso, o diagrama possibilitou visualizar relacionamentos entre as entidades, heranças, associações e dependências existentes na aplicação.

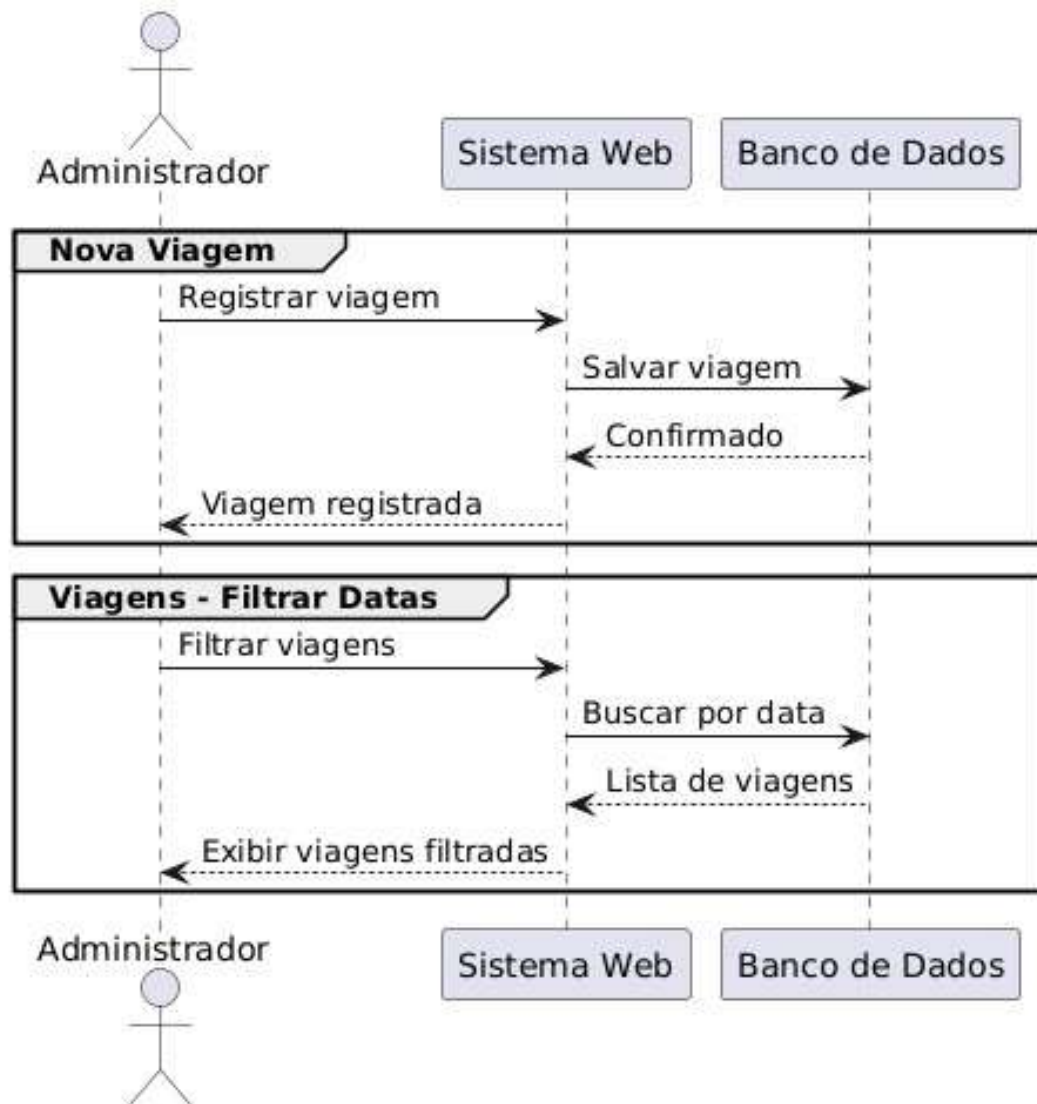
Essa etapa auxiliou diretamente na implementação utilizando C# e Entity Framework Core.

#### 4.4.3 Diagrama de Sequência

O diagrama de sequência foi utilizado para representar o fluxo de execução das principais funcionalidades do sistema ao longo do tempo.

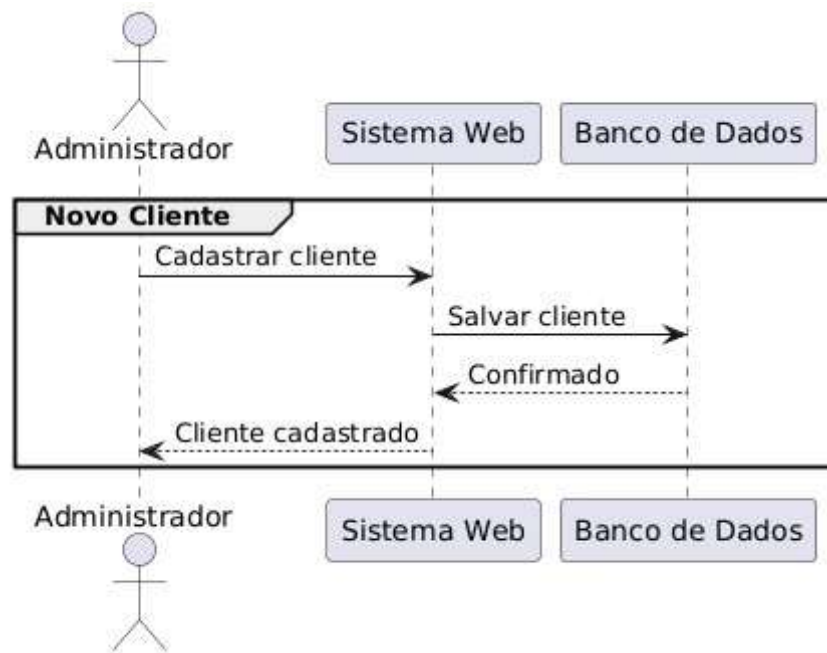
Esse tipo de modelagem permitiu visualizar a troca de mensagens entre usuário, interface, controllers, entidades e banco de dados durante a execução das operações da aplicação.

Figura 5 – Diagram de Sequência – Viagem



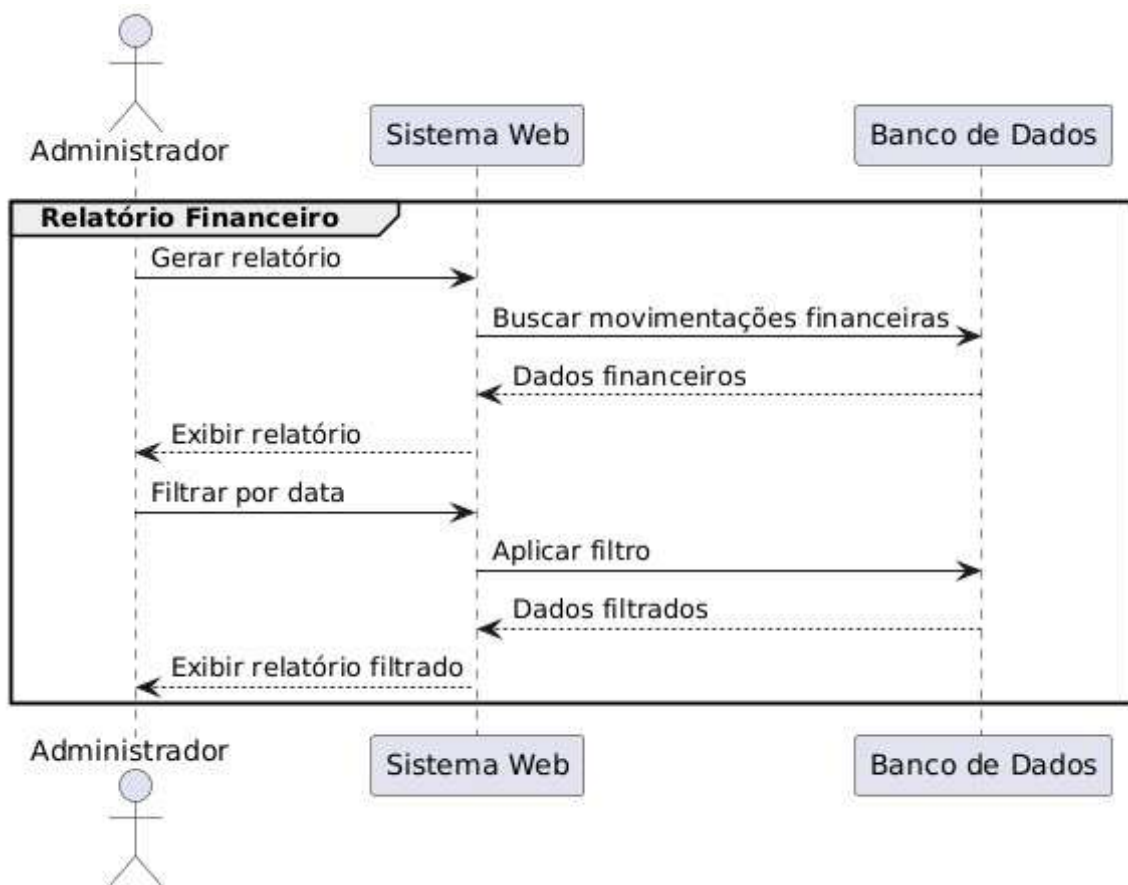
Fonte: UML

Figura 6 - Diagram de Sequência – Cliente



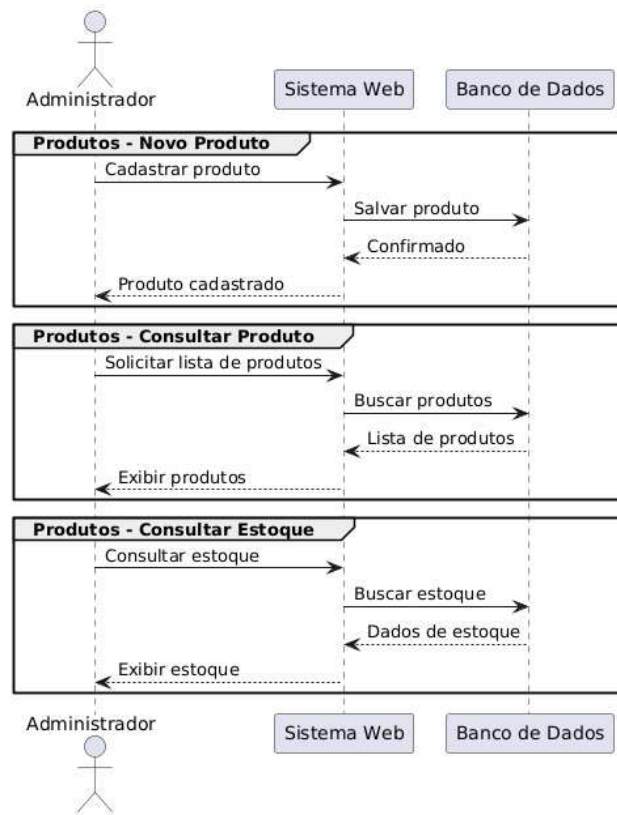
Fonte:UML

Figura 7 – Diagram de Sequência – Relatório Financeiro



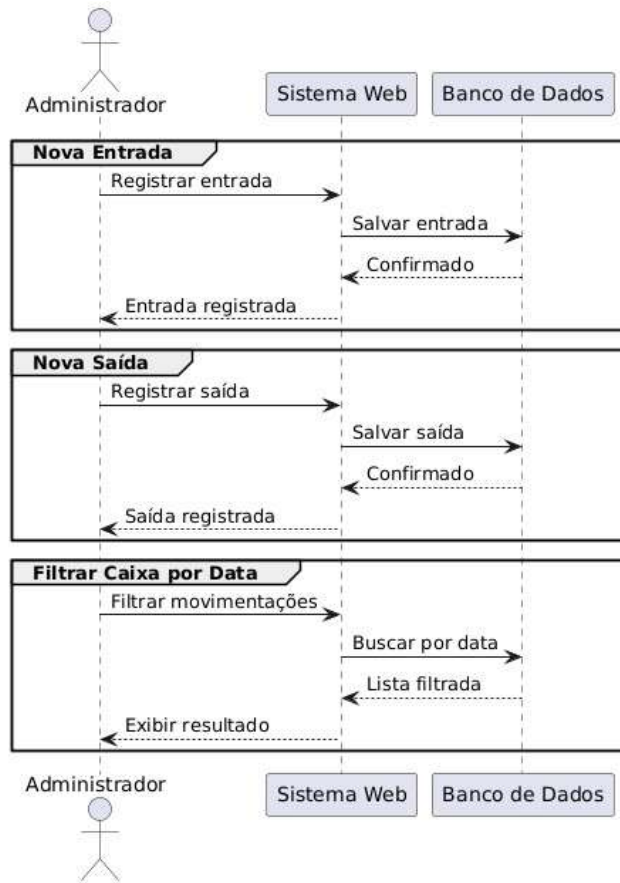
Fonte:UML

Figura 8 – Diagram de Sequência – Produtos



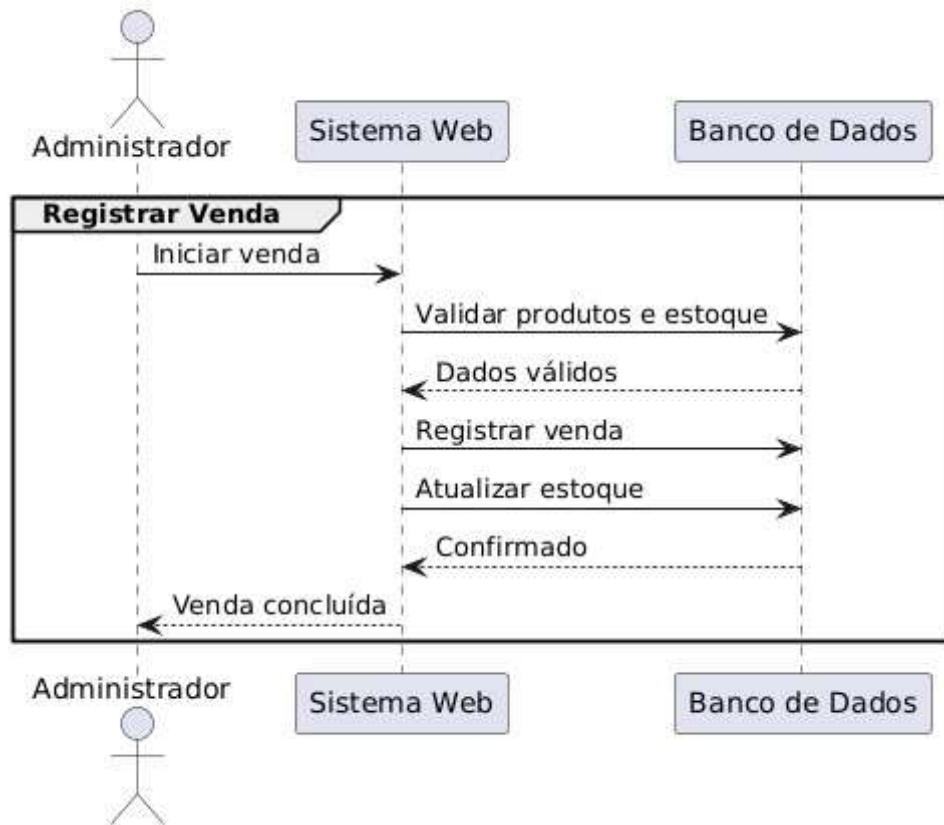
Fonte:UML

Figura 9 – Diagram de Sequência – Movimentação Financeira



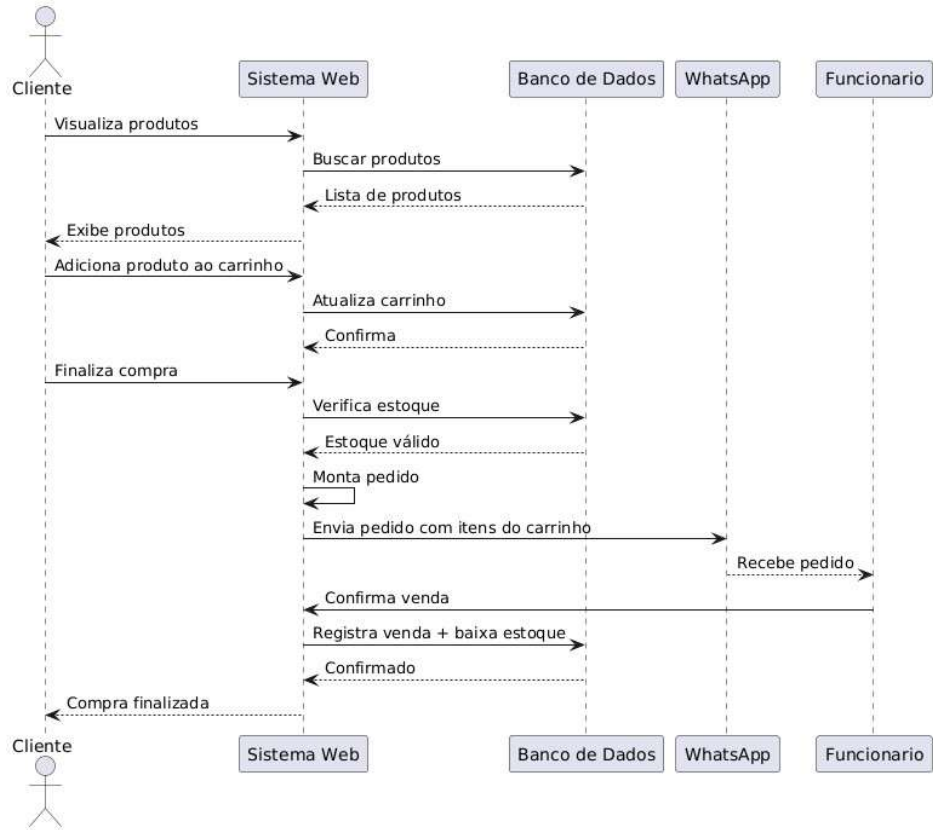
Fonte: UML

Figura 10 – Diagram de Sequência – Venda



Fonte: UML

Figura 11 – Diagram de Sequência – Cliente realizando venda



## 5 DESIGN DIGITAL E UX/UI

A etapa de Design Digital e UX/UI teve papel fundamental durante o desenvolvimento do projeto Paula Presentes ERP, pois além da construção funcional do sistema, também existia a necessidade de desenvolver uma identidade visual elegante, moderna e agradável para os usuários. Desde o início do projeto, a preocupação não esteve relacionada apenas às funcionalidades administrativas e à programação da aplicação, mas também à forma como o usuário perceberia visualmente a plataforma durante a utilização.

O projeto foi desenvolvido buscando unir sofisticação visual e praticidade operacional dentro de uma única aplicação. A intenção era criar uma experiência semelhante à de plataformas profissionais utilizadas por grandes marcas do segmento de moda, acessórios e vendas online, transmitindo uma aparência premium sem comprometer a facilidade de navegação. Durante todo o desenvolvimento, o design foi trabalhado paralelamente às etapas de Engenharia de Software, permitindo alinhar estrutura, organização visual e experiência do usuário.

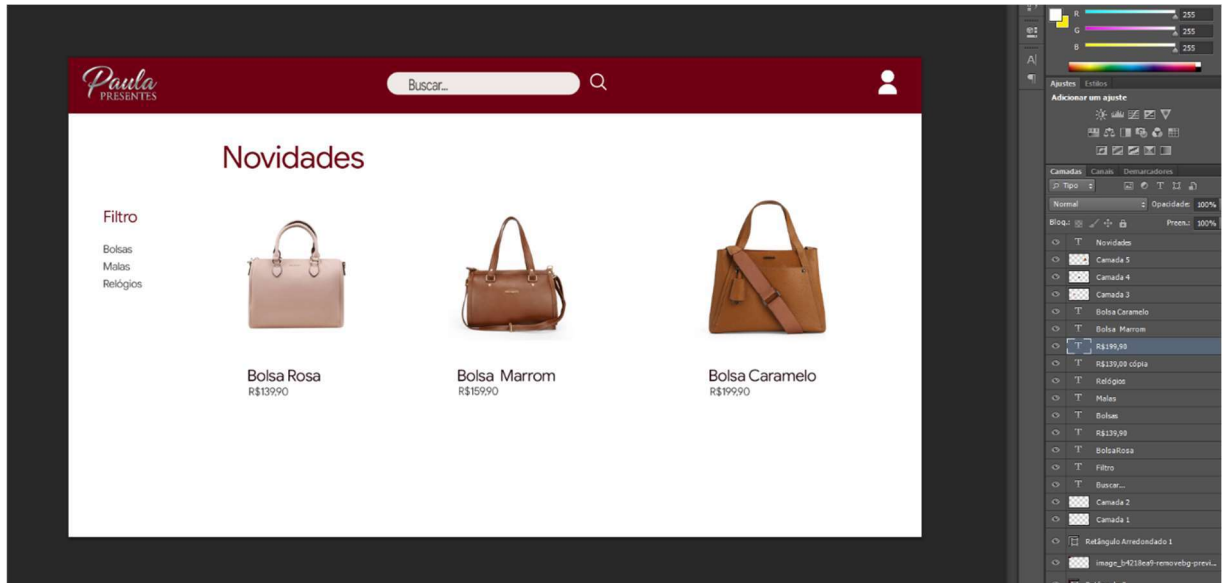
### 5.1 PROTOTIPAÇÃO VISUAL

Antes da implementação definitiva das telas, foi realizada uma etapa de prototipação visual utilizando Adobe Photoshop. Essa fase permitiu planejar previamente a aparência geral do sistema, auxiliando na definição dos elementos visuais, estrutura dos menus, organização das páginas e posicionamento dos componentes da interface.

Durante a prototipação, foram realizados diversos testes relacionados à identidade visual da plataforma, principalmente envolvendo combinações de cores, estilos de layout e distribuição das informações dentro das telas. Essa abordagem contribuiu para reduzir retrabalho durante a programação, já que grande parte da estrutura visual foi pensada antes da implementação prática no sistema.

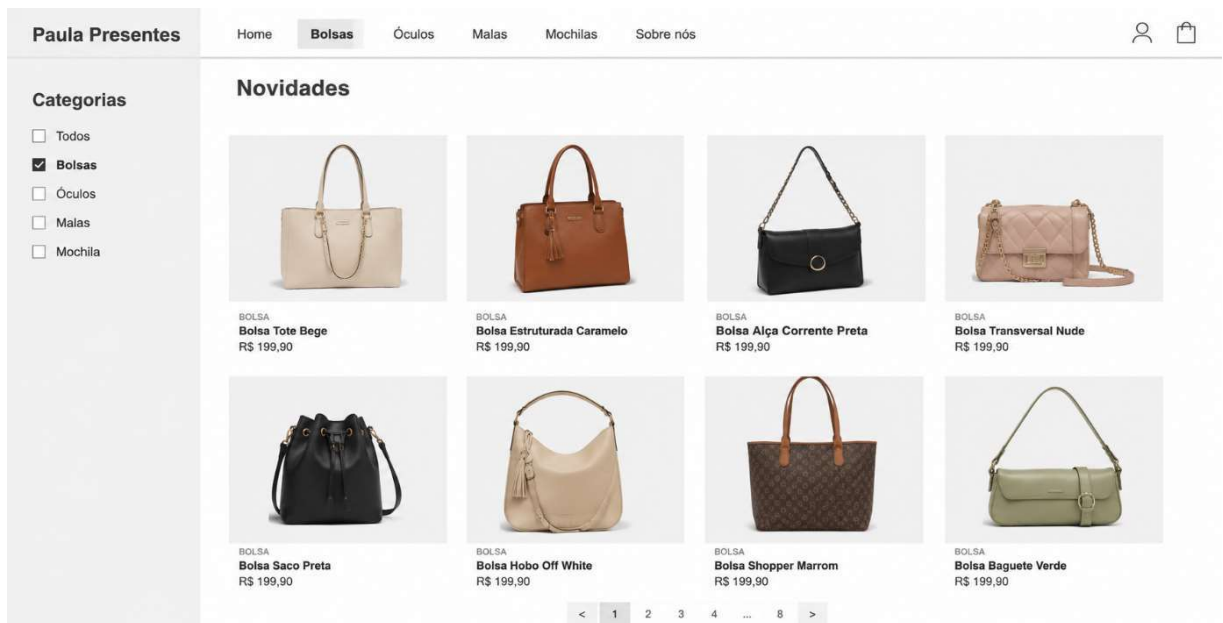
A prototipação também auxiliou no entendimento da experiência do usuário dentro da aplicação, permitindo visualizar como os usuários administrativos e clientes da loja virtual interagiriam com os componentes da interface. Dessa forma, foi possível construir uma identidade visual mais profissional e coerente com a proposta da marca Paula Presentes.

Figura 12 – Primeiro Protótipo



Fonte: Photoshop

Figura 13 – Segundo Protótipo



Fonte: Photoshop

## 5.2 CONCEITO VISUAL DA PLATAFORMA

A identidade visual do projeto foi inspirada em um conceito de luxo casual, buscando transmitir sofisticação, modernidade e elegância sem tornar a interface pesada ou excessivamente formal. O objetivo principal era criar um sistema visualmente refinado, mas ao mesmo tempo confortável, intuitivo e agradável para utilização diária.

O conceito visual foi fortemente influenciado pelo segmento da loja Paula Presentes, voltado para bolsas, acessórios, malas, mochilas e produtos femininos relacionados à moda e sofisticação. Por esse motivo, o sistema buscou se diferenciar dos tradicionais sistemas administrativos, que normalmente possuem aparência extremamente técnica e pouco atrativa visualmente.

Durante o desenvolvimento das interfaces, houve grande preocupação com detalhes estéticos como espaçamento, alinhamento dos elementos, proporção das imagens, suavidade das animações e harmonia visual entre os componentes. A proposta foi criar uma plataforma administrativa que mantivesse aparência premium mesmo sendo um sistema ERP de gestão empresarial.

### 5.3 PSICOLOGIA DAS CORES

A escolha das cores da aplicação foi baseada em conceitos de psicologia das cores, buscando transmitir sensações alinhadas ao posicionamento visual da marca. A paleta principal da plataforma utiliza tons de vinho escuro, vermelho profundo, dourado suave, branco e tons neutros claros.

O vinho escuro foi utilizado como principal cor da identidade visual por transmitir elegância, exclusividade, sofisticação e modernidade. Diferente de um vermelho tradicional mais intenso, o vinho proporciona uma aparência mais refinada e madura, reforçando o conceito premium da aplicação. Essa tonalidade aparece principalmente na navbar, títulos, ícones, botões e elementos de destaque da interface.

A aplicação também utiliza tons dourados suaves em alguns detalhes visuais e efeitos de interação, contribuindo para reforçar a sensação de luxo e exclusividade da plataforma. Já os tons claros e neutros foram utilizados para equilibrar visualmente as telas, evitando excesso de peso visual e proporcionando maior conforto durante a navegação.

A combinação dessas cores permitiu criar uma identidade visual sofisticada e moderna, alinhada diretamente ao segmento da loja Paula Presentes e à proposta estética do sistema.

#### 5.3.1 Cor Primária

A cor principal da aplicação é um tom de vinho escuro profundo:

- Primary: #570013

Aplicações:

- Navbar
- Ícones principais
- Destaques
- Links ativos
- Elementos de identidade visual
- Essa tonalidade foi escolhida por transmitir:
- Luxo
- Sofisticação
- Elegância
- Exclusividade
- Feminilidade moderna

### **5.3.2 Cor Secundária**

A cor secundária utilizada no sistema é um tom dourado suave:

- Secondary: #735C00

Aplicações:

- Destaques secundários
- Hover effects
- Elementos complementares
- Contrastes refinados

Essa tonalidade ajuda a reforçar o aspecto premium da aplicação, remetendo visualmente a:

- Ouro
- Exclusividade
- Sofisticação
- Valor agregado

### **5.3.3 Cores de Fundo e Superfície**

As telas utilizam tons claros e neutros para proporcionar conforto visual e equilíbrio entre os elementos da interface.

Fundo Principal:

- Background: #F9F9F9

Superfícies:

- Surface: #F9F9F9
- Surface Container: #EEEEEE
- Surface High: #E8E8E8
- Surface Low: #F3F3F3

Essas tonalidades foram utilizadas para:

- Reduzir fadiga visual
- Melhorar leitura
- Criar sensação de limpeza e organização
- Destacar os elementos principais da interface

#### 5.3.4 Cores de Texto

Texto Principal

- On Background: #1B1B1B

Texto Secundário

- On Surface Variant: #584141

Os textos utilizam tons escuros neutros para garantir:

- Alta legibilidade
- Contraste adequado
- Conforto visual
- Aparência moderna

#### 5.4 TIPOGRAFIA E HIERARQUIA VISUAL

Outro aspecto importante durante o desenvolvimento foi a definição da tipografia utilizada no sistema. Foram escolhidas fontes modernas e elegantes, buscando equilíbrio entre sofisticação visual e boa legibilidade.

A principal fonte utilizada foi a Inter, aplicada nos textos gerais, menus, botões e conteúdos administrativos devido à sua aparência moderna e excelente leitura em telas digitais. Já a fonte Noto Serif foi utilizada em títulos, logotipo e elementos de destaque visual, trazendo uma aparência mais refinada e sofisticada para a identidade da plataforma.

Fontes Utilizadas:

- Inter

Utilização:

- Textos do sistema
- Menus
- Botões
- Informações administrativas
- Conteúdo geral
- Características:
- Moderna
- Minimalista
- Alta legibilidade
- Excelente leitura em telas

- Noto Serif

Utilização:

- Logo
- Títulos
- Cabeçalhos
- Destaques visuais
- Características:
- Elegante
- Sofisticada
- Estilo premium
- Aparência refinada

A combinação entre uma fonte serifada sofisticada e uma fonte sans-serif moderna permitiu criar equilíbrio visual entre luxo e praticidade, alinhando-se diretamente à proposta estética da marca.

Além da escolha das fontes, também houve preocupação com hierarquia visual das informações. Diferentes tamanhos, pesos tipográficos e espaçamentos foram utilizados para destacar títulos, subtítulos, menus e conteúdos importantes dentro da interface. Essa organização contribuiu diretamente para criação de páginas mais limpas, intuitivas e visualmente agradáveis.

O cuidado com alinhamento, distribuição dos componentes e espaçamento entre elementos também auxiliou na melhoria da experiência do usuário, tornando a navegação mais organizada e confortável.

## 5.5 EXPERIÊNCIA DO USUÁRIO (UX)

A experiência do usuário foi considerada durante todas as etapas de construção das interfaces do sistema. O projeto foi desenvolvido buscando reduzir dificuldades de navegação e tornar os processos administrativos mais intuitivos e organizados.

Os menus, formulários e páginas foram estruturados para facilitar localização das informações e reduzir complexidade operacional durante utilização da plataforma. Houve preocupação constante com clareza visual, simplicidade de navegação e facilidade de aprendizado da interface.

Diversos detalhes foram desenvolvidos pensando diretamente na experiência do usuário, como posicionamento estratégico dos menus, padronização visual dos botões, feedback visual das ações realizadas, utilização de animações suaves e organização das informações dentro das telas.

O sistema também utiliza efeitos modernos como transparência parcial, backdrop blur e sombras suaves na barra de navegação superior, proporcionando maior sensação de profundidade visual e modernidade na interface. Esses elementos ajudaram a construir uma experiência mais sofisticada e profissional dentro da aplicação.

## 5.6 RESPONSABILIDADE E MOBILE FIRST

Outro ponto extremamente importante durante o desenvolvimento foi a responsividade das interfaces. Considerando o crescimento do uso de dispositivos móveis para navegação e compras online, o sistema foi desenvolvido pensando na adaptação para diferentes tamanhos de tela.

As páginas foram construídas utilizando Tailwind CSS, permitindo criação de layouts flexíveis e responsivos de forma mais organizada. Durante o desenvolvimento, diversos testes foram realizados em diferentes resoluções para garantir funcionamento adequado tanto em smartphones quanto em tablets, notebooks e monitores widescreen.

A estrutura responsiva permitiu que a aplicação mantivesse organização visual, alinhamento dos componentes e boa experiência de navegação independentemente do dispositivo utilizado pelo usuário.

## 5.7 IMPORTÂNCIA DO DESIGN NO PROJETO

O Design Digital e UX/UI tiveram papel essencial na construção do Paula Presentes ERP, contribuindo não apenas para estética da aplicação, mas também para organização, usabilidade e experiência geral do usuário.

A preocupação com identidade visual, psicologia das cores, responsividade, hierarquia visual e experiência de navegação permitiu desenvolver uma plataforma mais moderna, elegante e alinhada às tendências atuais do mercado digital.

Dessa forma, o projeto conseguiu unir gestão empresarial e experiência visual sofisticada dentro de uma única aplicação, fortalecendo a proposta premium da marca Paula Presentes e proporcionando uma utilização mais agradável tanto para clientes quanto para administradores do sistema.

## **6 DESENVOLVIMENTO DO SISTEMA**

Após a conclusão das etapas de planejamento e modelagem, iniciou-se o desenvolvimento prático do sistema.

O desenvolvimento foi realizado utilizando a arquitetura MVC, organizando a aplicação em Models, Controllers e Views. Os Models ficaram responsáveis pela estrutura das entidades e integração com banco de dados, os Controllers pela lógica de negócio e processamento das informações, enquanto as Views foram responsáveis pela construção das interfaces visuais.

Durante essa etapa foram implementadas funcionalidades relacionadas ao cadastro de produtos, gerenciamento de imagens, controle de estoque, movimentações financeiras, registro de vendas, relatórios administrativos e exibição dinâmica dos produtos no ecommerce.

Também foi desenvolvida integração com Supabase Storage para upload e armazenamento permanente das imagens dos produtos em nuvem. Essa funcionalidade solucionou problemas relacionados ao armazenamento temporário de arquivos em servidores cloud.

Ao longo do desenvolvimento, diversos desafios técnicos precisaram ser solucionados, incluindo integração entre banco de dados e aplicação, manipulação de imagens, deploy da aplicação, correções de responsividade e ajustes de carregamento das páginas.

Grande parte do processo envolveu testes contínuos, refatoração de código e melhorias estruturais para garantir maior estabilidade do sistema.

### **6.1 TESTES E QA**

Após a implementação das funcionalidades do sistema, foi realizada a etapa de testes com o objetivo de garantir a qualidade da aplicação, identificar possíveis falhas e validar o comportamento geral do sistema em diferentes cenários de uso. Essa fase foi fundamental para assegurar a estabilidade, confiabilidade e consistência das funcionalidades desenvolvidas antes da disponibilização em ambiente de produção.

#### **6.1.1 Testes Funcionais**

Os testes funcionais foram realizados com o objetivo de validar o correto funcionamento das principais funcionalidades do sistema. Nesta etapa, foram

verificadas operações como cadastro, edição e exclusão de produtos, autenticação de usuários, controle de estoque, movimentações financeiras e integração entre as camadas da aplicação. Cada funcionalidade foi testada individualmente para garantir que os resultados obtidos estivessem de acordo com os requisitos previamente definidos.

### **6.1.2 Testes de Integração**

Os testes de integração tiveram como foco a validação da comunicação entre os diferentes módulos do sistema, incluindo a integração entre o backend desenvolvido em ASP.NET MVC, o banco de dados PostgreSQL e o serviço de armazenamento em nuvem Supabase Storage. Essa etapa permitiu identificar inconsistências na troca de dados entre as camadas, garantindo que as operações realizadas no sistema fossem corretamente persistidas e recuperadas.

### **6.1.3 Testes de Interface e Usabilidade**

Foram realizados testes de interface com o objetivo de avaliar a experiência do usuário (UX) e a usabilidade do sistema. Nessa etapa, foram analisados aspectos como organização dos elementos na tela, clareza das informações apresentadas e facilidade de navegação entre as páginas. O objetivo foi garantir que a aplicação apresentasse uma interface intuitiva, funcional e adequada ao público-alvo.

### **6.1.4 Testes de Responsividade**

Os testes de responsividade foram executados para verificar o comportamento da interface em diferentes tamanhos de tela, incluindo dispositivos móveis, tablets e desktops. Durante essa etapa, foram avaliados ajustes de layout, adaptação de componentes e legibilidade das informações. Esses testes foram essenciais para garantir que a aplicação mantivesse boa usabilidade em diferentes dispositivos.

### **6.1.5 Correções e Ajustes (Debugging)**

Durante a fase de testes, foram identificadas e corrigidas diversas inconsistências relacionadas principalmente à renderização de imagens, rotas da aplicação, integração com o Supabase Storage e ajustes de responsividade. O

processo de depuração (debugging) permitiu estabilizar o sistema, corrigindo erros e melhorando a performance geral da aplicação.

#### **6.1.6 Importância da Etapa de QA**

A etapa de testes e garantia de qualidade (QA) foi essencial para assegurar a confiabilidade do sistema antes da publicação em ambiente de produção. Além de validar o funcionamento das funcionalidades, essa fase contribuiu para a identificação de melhorias estruturais e refinamento da experiência do usuário, resultando em uma aplicação mais estável e consistente.

## **6.2 DEPLOY E HOSPEDAGEM**

Após a conclusão das etapas de desenvolvimento e validação por meio de testes, foi realizada a publicação da aplicação em ambiente de produção. Essa etapa teve como objetivo disponibilizar o sistema em uma infraestrutura cloud, permitindo acesso remoto e simulando um ambiente real de operação.

### **6.2.1 Versionamento do Código-Fonte**

O código-fonte do projeto foi versionado e hospedado no GitHub, permitindo o controle de alterações, histórico de desenvolvimento e colaboração entre diferentes versões do sistema. O uso de versionamento contribuiu para maior organização do projeto, além de garantir segurança no gerenciamento do código e facilidade de manutenção futura.

### **6.2.2 Processo de Deploy**

O deploy da aplicação foi realizado utilizando a plataforma Railway, que possibilita a publicação de aplicações web de forma simplificada e integrada com bancos de dados em nuvem. Essa plataforma foi responsável por hospedar a aplicação ASP.NET MVC, permitindo sua execução em ambiente online com alta disponibilidade.

### **6.2.3 Banco de Dados em Ambiente Cloud**

O banco de dados PostgreSQL foi configurado em ambiente cloud, garantindo acesso remoto seguro e persistência dos dados da aplicação. Essa abordagem

eliminou a necessidade de infraestrutura local, além de facilitar o escalonamento e manutenção do sistema em ambiente de produção.

#### **6.2.4 Armazenamento de Arquivos em Nuvem**

O Supabase Storage foi utilizado como solução para armazenamento de arquivos em nuvem, especialmente imagens de produtos. Essa estratégia permitiu maior estabilidade, evitando perda de dados e reduzindo a dependência do servidor da aplicação para armazenamento local.

#### **6.2.5 Integração entre Serviços**

A arquitetura em ambiente cloud foi composta pela integração entre GitHub, Railway, PostgreSQL e Supabase Storage. Essa integração permitiu que a aplicação funcionasse de forma distribuída, separando responsabilidades entre código-fonte, banco de dados e armazenamento de arquivos, o que contribui para maior escalabilidade e robustez do sistema.

#### **6.2.6 Considerações sobre o Ambiente de Produção**

A utilização de serviços em nuvem permitiu que o sistema fosse disponibilizado em um ambiente semelhante ao utilizado em aplicações comerciais reais. Essa abordagem garantiu maior confiabilidade, escalabilidade e acessibilidade, além de representar uma arquitetura moderna alinhada com práticas atuais de desenvolvimento e deploy de aplicações web.

## 7 FUNCIONALIDADES DO SISTEMA

O sistema Paula Presentes ERP foi desenvolvido com o objetivo de integrar funcionalidades comerciais, administrativas e gerenciais em uma única plataforma web. A aplicação foi estruturada para atender tanto às necessidades do ecommerce quanto às operações internas da empresa, proporcionando maior organização, controle e praticidade nos processos administrativos.

As funcionalidades do sistema foram divididas em módulos, permitindo melhor organização da aplicação e facilitando futuras expansões da plataforma.

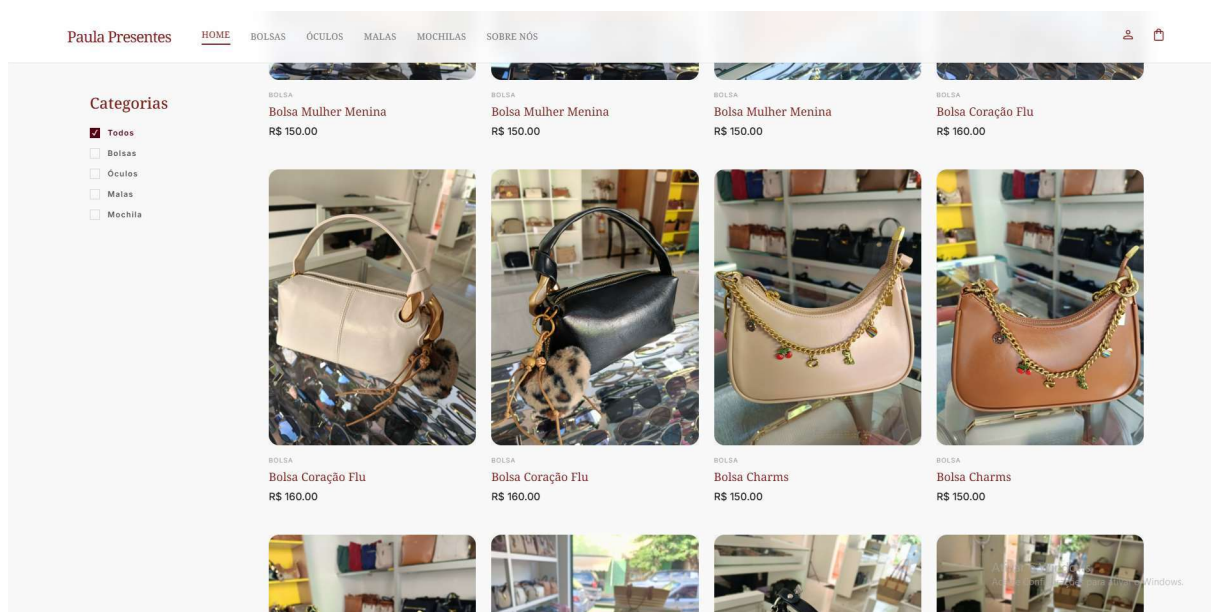
### 7.1 ECOMMERCE

O módulo de ecommerce foi desenvolvido para disponibilizar a vitrine digital dos produtos e permitir a navegação e interação dos usuários com a loja virtual.

Neste módulo, estão presentes as seguintes funcionalidades:

- Apresentação de produtos

Figura 14 – Tela Home

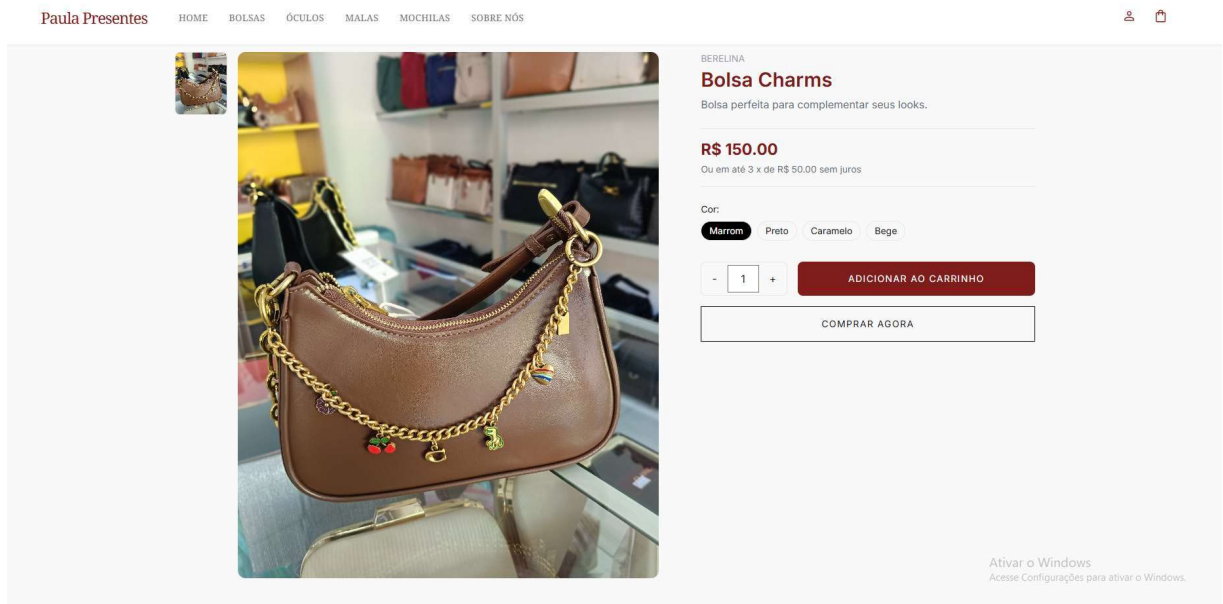


Fonte: <https://paulapresentes.com/>

Visualização de produtos com informações completas, incluindo nome, preço de venda, imagem, marca, descrição e cores disponíveis

- Detalhes do produto

Figura 15 – Tela Detalhes

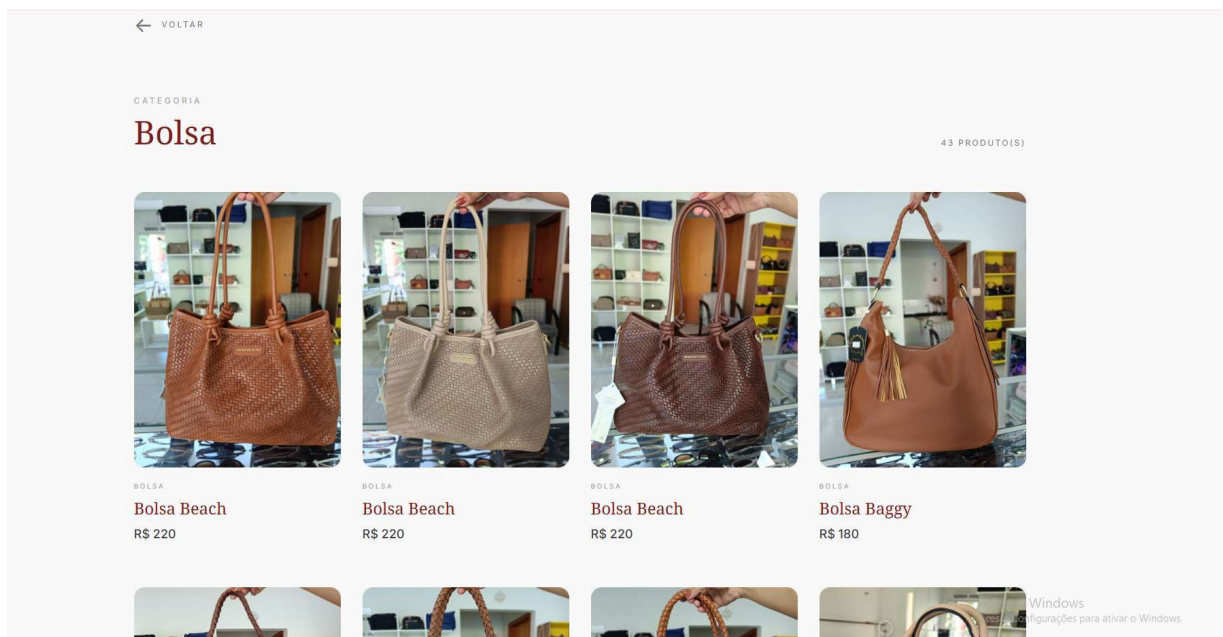


Fonte: <https://paulapresentes.com/>

Visualização de produtos com informações completas, incluindo nome, preço de venda, imagem, marca, descrição e cores disponíveis

- Categoria

Figura 16 – Tela Categoria Bolsas

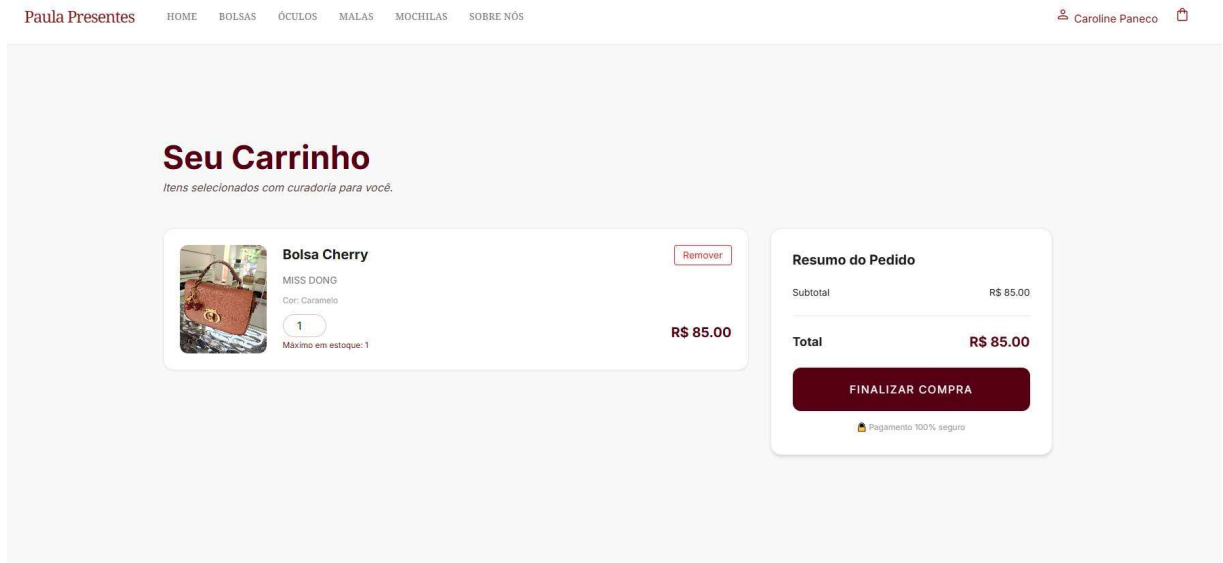


Fonte: <https://paulapresentes.com/>

Exibição de produtos específicos separados por categoria.

- Carrinho

Figura 17 – Tela Carrinho

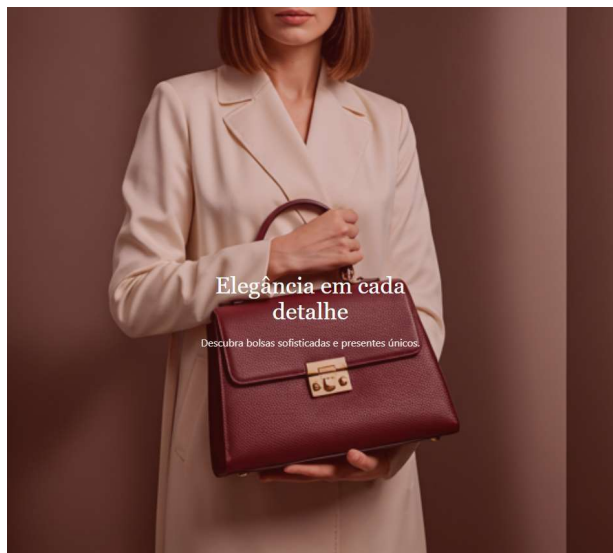


Fonte: <https://paulapresentes.com/>

Carrinho de compras, permitindo que o usuário adicione e gerencie produtos selecionados

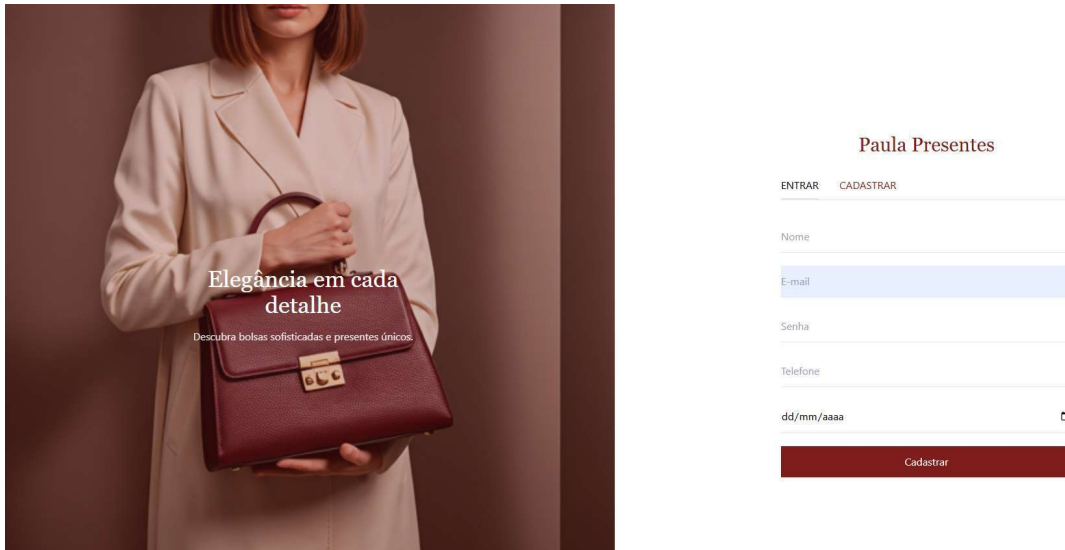
- Login e Cadastro

Figura 18 – Tela Login



Fonte: <https://paulapresentes.com/>

Figura 19 – Tela Cadastro

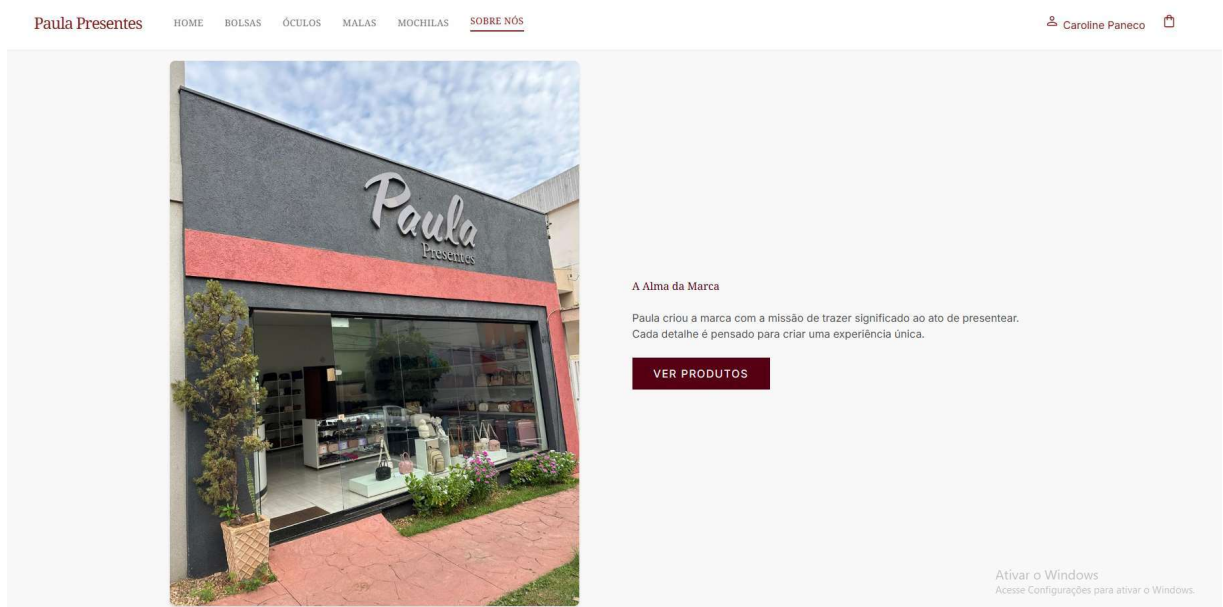


Fonte: <https://paulapresentes.com/>

Cadastro e login de usuários, permitindo autenticação e personalização da experiência

- Sobre nós

Figura 20 – Tela Sobre nós



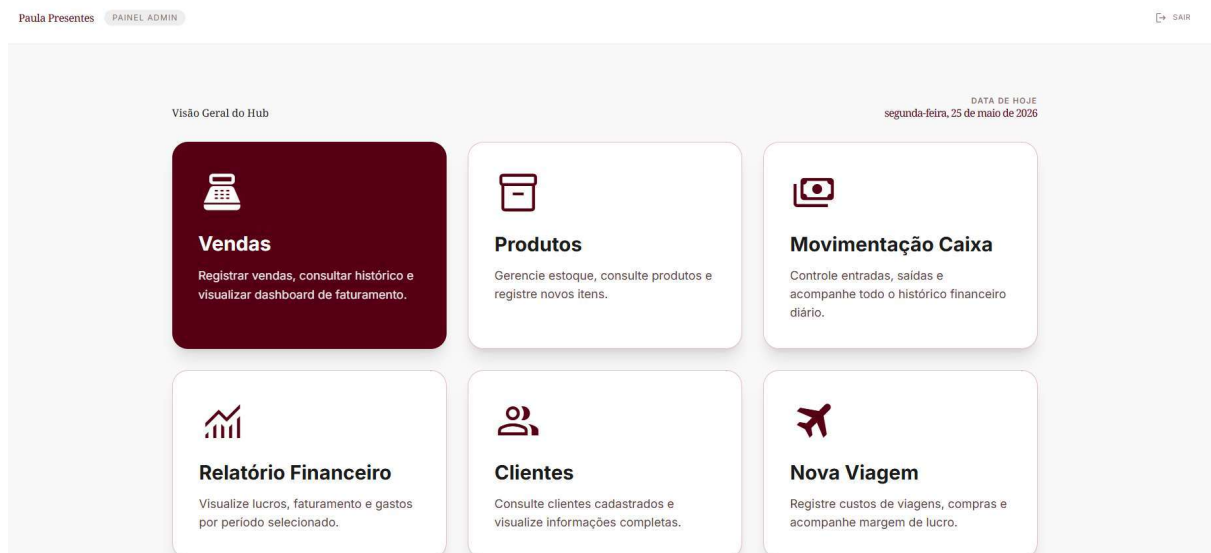
Fonte: <https://paulapresentes.com/>

Página institucional “Conheça a nossa história”, apresentando informações sobre a loja e sua identidade

## 7.2 ADMINISTRAÇÃO

O módulo administrativo foi desenvolvido para centralizar o gerenciamento completo da plataforma, permitindo o controle das operações internas do sistema.

Figura 21 – Tela Administrativo

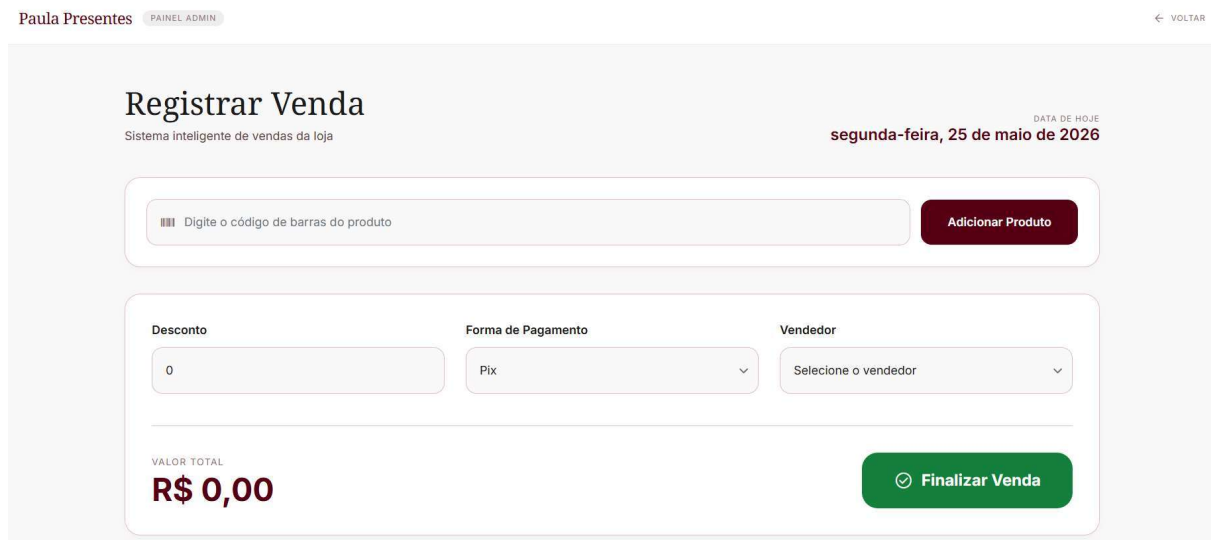


Fonte: <https://paulapresentes.com/>

As funcionalidades deste módulo incluem:

- Registro de nova venda

Figura 22 – Tela Venda




Fonte: <https://paulapresentes.com/>

Figura 23 – Tela Venda em process

Paula Presentes PAINEL ADMIN ← VOLTAR

☐ Digite o código de barras do produto Adicionar Produto

 **Bolsa Strong**  
BELLA PAULA • Marrom  
Estoque: 1 **R\$ 220** Quantidade: 1 Remover

Desconto: 0 Forma de Pagamento: Dinheiro Vendedor: Paula

VALOR TOTAL **R\$ 220,00** Finalizar Venda

Ativar o Windows  
Acesse Configurações para ativar o Windows.

Fonte: <https://paulapresentes.com/>


- Gestão de produtos


Figura 24 – Tela Produto


Paula Presentes PAINEL ADMIN ← VOLTAR

## Produtos

Controle de estoque e gerenciamento de produtos

 **Novo Produto**  
Registre novos produtos, valores, estoque e informações completas.

 **Consultar Estoque**  
Consulte quantidade disponível e acompanhe produtos com estoque baixo.

 **Consultar Produtos**  
Visualize todos os produtos cadastrados com imagens e detalhes.

Fonte: <https://paulapresentes.com/>

- Cadastro de novo produto

Figura 25 – Tela Novo Produto

Paula Presentes PAINEL ADMIN ← VOLTAR

## Novo Produto

Cadastre um novo produto no sistema

Nome	Marca
<input type="text"/>	<input type="text"/>
Categoria	Cor
<input type="text" value="Selecione"/>	<input type="text" value="Selecione"/>
Código de Barras	Preço Venda
<input type="text"/>	<input type="text"/>
Preço Custo	Quantidade Estoque
<input type="text"/>	<input type="text"/>
Peso (kg)	Altura (cm)
<input type="text"/>	<input type="text"/>

Ativar o Windows  
Acesse Configurações para ativar o Windows.

Fonte: <https://paulapresentes.com/>

- Consulta de estoque

Figura 26 – Tela Consultar Estoque

Paula Presentes PAINEL ADMIN ← VOLTAR

## Consultar Estoque

Produtos disponíveis no estoque

Pesquisar por código, nome, categoria ou cor

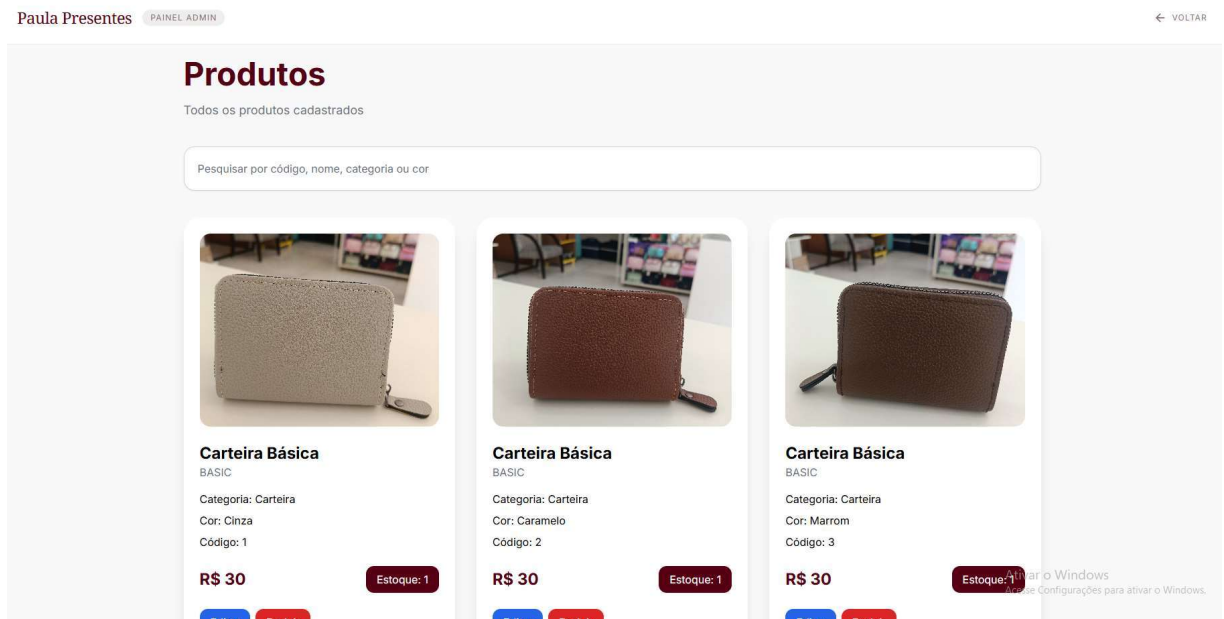
Produto	Categoria	Cor	Estoque
Carteira Básica	Carteira	Marron	1
Carteira Básica	Carteira	Azul	1
Carteira Franja	Carteira	Off White	1
Carteira Esmeralda	Carteira	Preto	3
Carteira Básica	Carteira	Cinza	1
Carteira Básica	Carteira	Caramelo	1
Carteira Básica	Carteira	Dourado	1

Ativar o Windows  
Acesse Configurações para ativar o Windows.

Fonte: <https://paulapresentes.com/>

- Consulta de produtos cadastrados

Figura 27 – Tela Consultar Produtos



Fonte: <https://paulapresentes.com/>

Figura 28 – PopUp Editar Produto

Fonte: <https://paulapresentes.com/>

Figura 29 – PopUp Excluir Produto



Fonte: <https://paulapresentes.com/>

- Movimentação financeira

Figura 30 – Tela Movimentacao financeira

Paula Presentes PAINEL ADMIN ← VOLTAR

## Movimentação Caixa

Controle financeiro completo da loja

Nova Entrada Nova Saída

Entradas **R\$ 327.70**

Saídas **R\$ 4,350.00**

Saldo Atual **R\$ -4,022.30**

Data Inicial   Data Final   Filtrar Período

Tipo	Categoria	Descrição	Valor	Data
Entrada	Venda	Venda #3 - Sueli - Cartão Débito	R\$ 127.90	24/05/2026 22:48
Saída	Viagem	Viagem #0	R\$ 4,350.00	23/05/2026 00:22
Entrada	Venda	Venda #2 - Sueli - Dinheiro	R\$ 199.80	23/05/2026 00:08

Ativar o Windows  
Vá em Configurações para ativar o Windows.

Fonte: <https://paulapresentes.com/>

- Registro de entradas

Figura 31 – PopUp Nova Entrada

**Nova Entrada** ✕

Categoria  
Recebimento ▾

Descrição

Valor  
R\$ 0,00

**Salvar Entrada**

Fonte: <https://paulapresentes.com/>

- Registro de saídas

Figura 32 – PopUp Nova Saída

**Nova Saída** ✕

Categoria  
Selecione ▾

Descrição

Valor

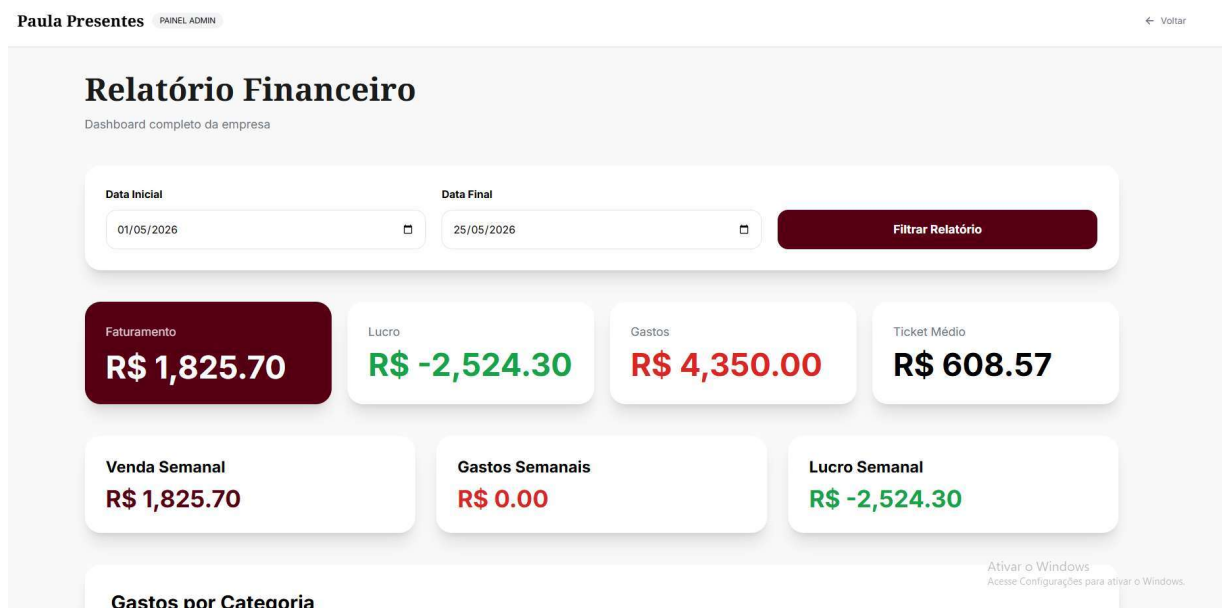
**Salvar Saída**

Fonte: <https://paulapresentes.com/>

- Dashboard gerencial com visualização geral das informações do sistema
- Filtro de data para análise de informações em períodos específicos
- Dashboard de categorias, permitindo análise organizada dos produtos por classificação

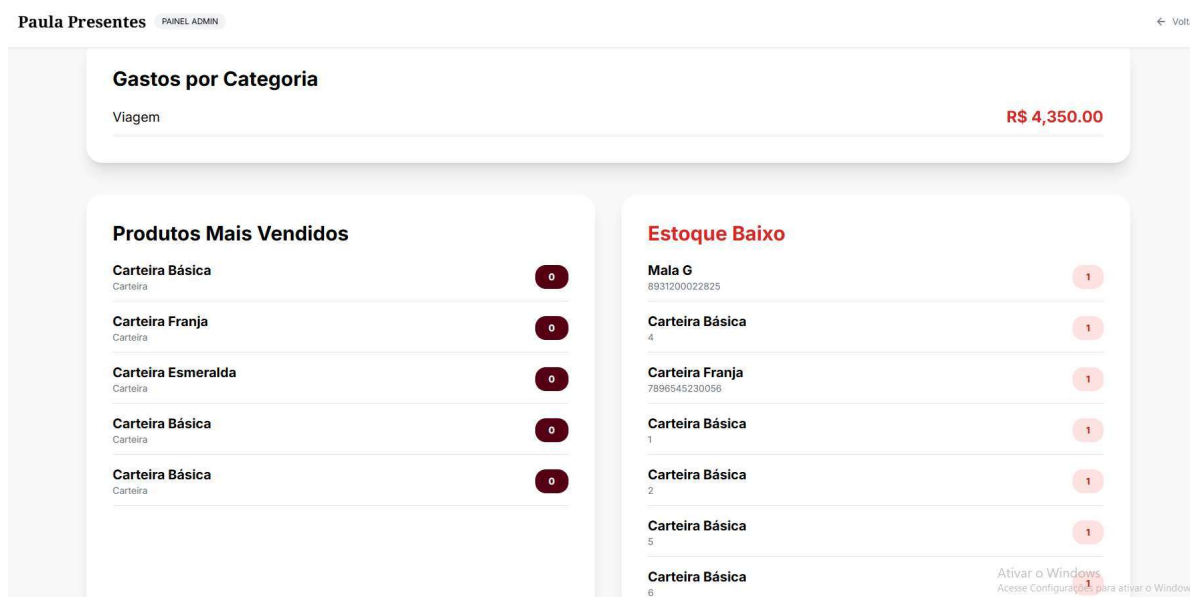
- Relatório financeiro

Figura 33 – Tela Relatório Financeir



Fonte: <https://paulapresentes.com/>

Figura 34 – Tela Gastos Financeiro



Fonte: <https://paulapresentes.com/>

- Clientes

O módulo de clientes tem como objetivo o gerenciamento das informações dos consumidores cadastrados no sistema.

- Suas funcionalidades incluem:
- Cadastro de novos clientes

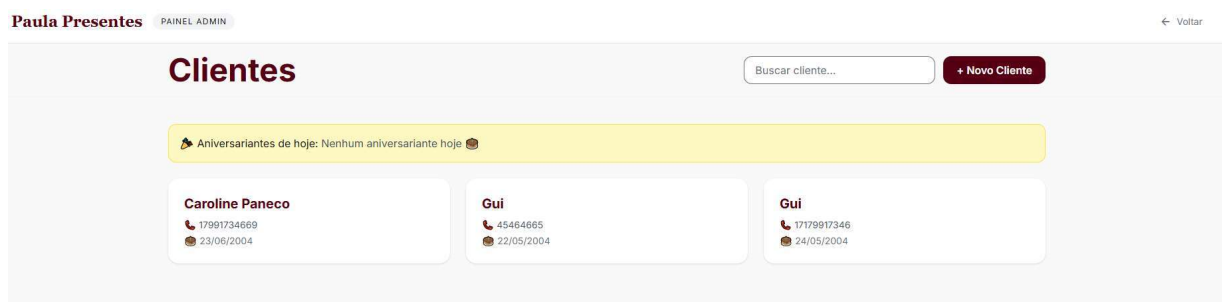
Figura 35 – PopUp Novo Cliente



Fonte: <https://paulapresentes.com/>

- Visualização e controle de clientes cadastrados

Figura 36 – Tela Clientes



Fonte: <https://paulapresentes.com/>

Listagem de aniversariantes do dia, permitindo ações de relacionamento e fidelização

- Viagens

O módulo de viagens foi desenvolvido para controle de deslocamentos e seus respectivos custos, permitindo melhor organização das despesas operacionais.

As funcionalidades deste módulo incluem:

- Histórico de viagens registradas no sistema
- Registro de novas viagens

Figura 37 – Tela Viagens

The image shows a web interface for managing travel. It features a form titled 'Nova Viagem' with four input fields: 'Data da Viagem' (dd/mm/aaaa), 'Custo Ida e Volta', 'Custo Alimentação', and 'Valor Total Compra'. A dark red 'Salvar Viagem' button is positioned below the form. Below the form is a table titled 'Histórico de Viagens' with the following data:

Data	Ida e Volta	Alimentação	Compras	Total
17/05/2026	R\$ 300.00	R\$ 50.00	R\$ 4,000.00	R\$ 4,350.00

Fonte: <https://paulapresentes.com/>

- Controle de viagens realizadas com detalhamento de gastos

Figura 38 – Tela Nova Viagem

The image shows a dashboard titled 'Controle de Viagens' with the subtitle 'Gestão de viagens, custos e compras'. The current date is 'segunda-feira, 25 de maio de 2026'. The dashboard includes a filter section for 'Filtrar Viagens' with two date pickers, a dropdown menu set to 'Todos', and a 'Filtrar' button. Below the filter are four summary cards: 'Total Viagens' (1), 'Total Gastos' (R\$ 4,350.00), 'Compras' (R\$ 4,000.00), and 'Média Viagem' (R\$ 4,350.00). At the bottom, there is a 'Nova Viagem' form with the same fields as in Figure 37.

Fonte: <https://paulapresentes.com/>

### 7.3 INTEGRAÇÃO GERAL DO SISTEMA

Todas as funcionalidades descritas estão integradas entre si dentro da mesma plataforma, permitindo que informações cadastradas em um módulo impactem diretamente outros setores do sistema. Essa integração garante maior consistência dos dados, automação de processos e maior eficiência operacional, caracterizando o sistema como uma solução unificada de gestão e ecommerce.

### 7.4 RESPONSABILIDADE

A responsividade foi uma das preocupações centrais durante o desenvolvimento do projeto.

As interfaces foram desenvolvidas utilizando Tailwind CSS, permitindo adaptação automática das telas para diferentes tamanhos de dispositivos, incluindo smartphones, tablets e computadores.

Durante os testes, foram realizados diversos ajustes relacionados ao comportamento dos componentes em dispositivos móveis, buscando melhorar usabilidade, organização visual e experiência do usuário.

O objetivo foi garantir que o sistema apresentasse navegação confortável e aparência moderna independentemente do dispositivo utilizado pelo usuário.

### 7.5 SEGURANÇA E INTEGRIDADE DOS DADOS

O sistema também contou com preocupações relacionadas à segurança e integridade das informações armazenadas.

Foram implementadas validações de dados durante os processos de cadastro, reduzindo riscos de informações inconsistentes dentro da aplicação.

Além disso, o armazenamento das imagens em nuvem contribuiu para maior segurança dos arquivos da plataforma, evitando perda de dados em ambientes cloud temporários.

A utilização do PostgreSQL também auxiliou na integridade das informações através de relacionamentos estruturados, chaves estrangeiras e organização relacional dos dados.

Outro fator importante foi o uso de arquitetura MVC, que contribuiu para melhor separação das responsabilidades do sistema e organização geral da aplicação.

## 8 CONCLUSÃO

O desenvolvimento do projeto Paula Presentes ERP proporcionou uma experiência extremamente importante para aplicação prática dos conhecimentos adquiridos ao longo do curso, permitindo integrar conceitos de Engenharia de Software, Banco de Dados, Design Digital e Técnicas de Programação em um único sistema completo.

Durante o projeto, foi possível compreender na prática a importância do planejamento e da organização no desenvolvimento de software, desde o levantamento de requisitos até a etapa final de deploy da aplicação em ambiente cloud.

Além da parte técnica, o desenvolvimento também trouxe aprendizados relacionados à experiência do usuário, identidade visual, organização de projetos, integração entre tecnologias e resolução de problemas reais encontrados durante a implementação do sistema.

Diversos desafios precisaram ser solucionados ao longo do desenvolvimento, principalmente relacionados à integração com banco de dados, armazenamento de imagens em nuvem, deploy da aplicação, responsividade mobile e estruturação da arquitetura do sistema. Esses desafios contribuíram significativamente para evolução técnica e profissional durante a construção do projeto.

A utilização da arquitetura MVC, integração com PostgreSQL, implementação de armazenamento cloud e desenvolvimento de um ecommerce responsivo permitiram criar uma aplicação moderna, organizada e escalável, aproximando o projeto de sistemas empresariais utilizados no mercado profissional.

O projeto também demonstrou a importância da Engenharia de Software na organização do desenvolvimento, definição de requisitos, modelagem dos dados, planejamento das funcionalidades e controle da qualidade da aplicação.

Como possibilidades futuras, o sistema poderá receber novas funcionalidades, como autenticação completa de usuários, dashboards analíticos, relatórios avançados, integração com pagamentos online, controle de pedidos, emissão de notas fiscais e expansão dos módulos administrativos, transformando a plataforma em um ERP ainda mais robusto e completo.

## 9 REFERÊNCIAS

MICROSOFT. ASP.NET Core Documentation. Disponível em:  
<https://learn.microsoft.com/aspnet/core>.

MICROSOFT. Entity Framework Core Documentation. Disponível em:  
<https://learn.microsoft.com/ef/core>.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL Documentation.  
Disponível em: <https://www.postgresql.org/docs>.

SUPABASE. Supabase Documentation. Disponível em: <https://supabase.com/docs>.

TAILWIND LABS. Tailwind CSS Documentation. Disponível em:  
<https://tailwindcss.com/docs>.

GITHUB. GitHub Documentation. Disponível em: <https://docs.github.com>.

RAILWAY. Railway Documentation. Disponível em: <https://docs.railway.app>.

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