

EcoPneus

Site e-commerce de Pneus

Geraldo Camargo Neto
Richard Moreira

OBJETIVO GERAL DO SISTEMA

- Criar uma plataforma especializada em pneus para veículos elétricos.
- Permitir ao cliente consultar, filtrar, comprar e acompanhar pedidos.
- Oferecer uma experiência simples, segura e rápida de ponta a ponta.
- Integrar pagamento online e cálculo de frete por CEP.

REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Requisitos Funcionais:

- Visualizar catálogo
- Aplicar filtros (aro, marca, medida)
- Ver detalhes do produto
- Criar conta / Login
- Calcular frete
- Finalizar compra
- Acompanhar pedido

Requisitos Não Funcionais:

- Interface acessível
- Alta disponibilidade 24/7
- Páginas em até 2 segundos
- Segurança de dados
- Compatível com mobile

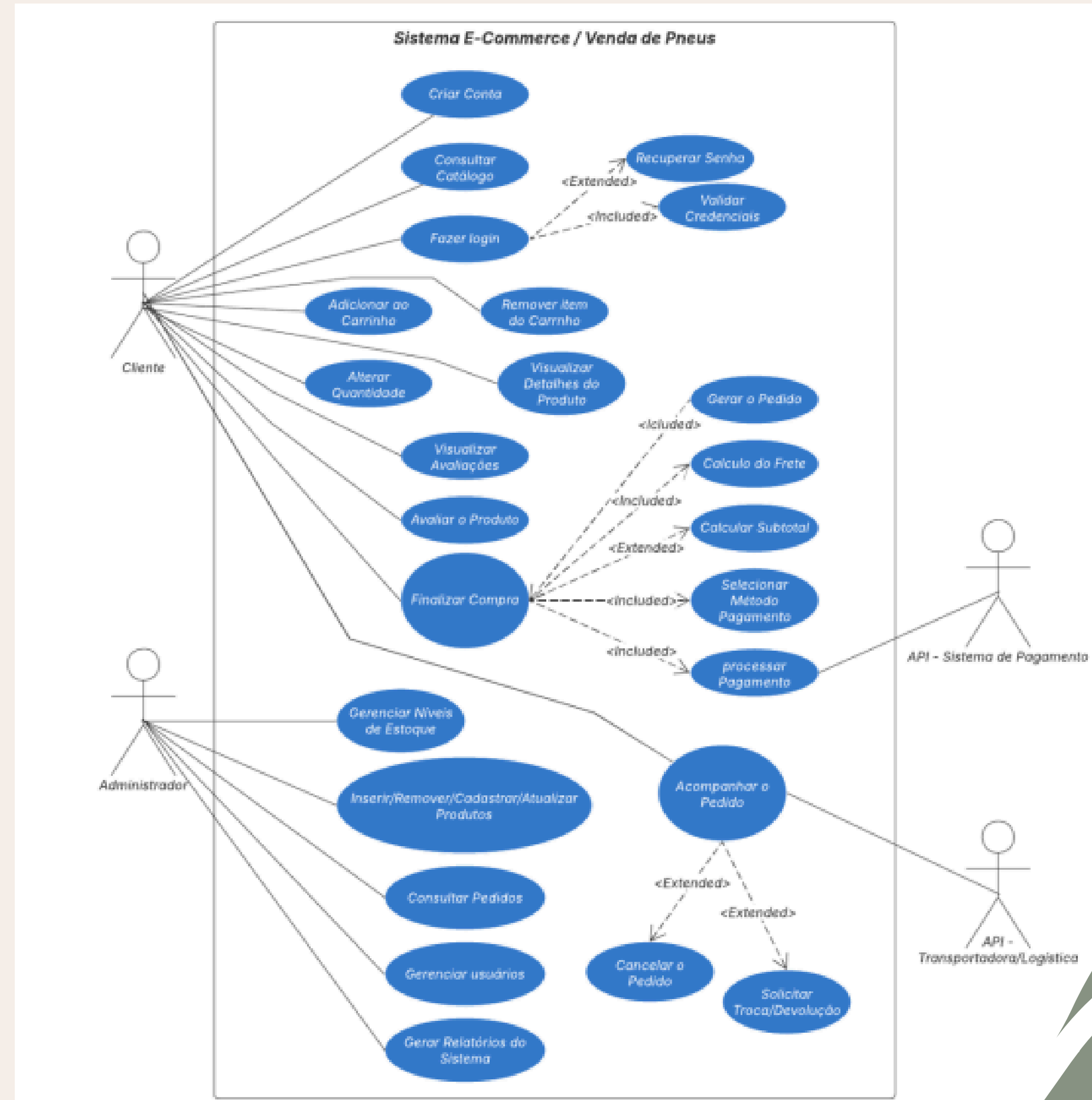
CASO DE USO

Atores:

- Cliente
- Administrador
- Sistema de Pagamento
- Transportadora

Casos principais:

- Consultar catálogo
- Filtrar
- Detalhar produto
- Carrinho
- Login
- Finalizar compra
- Pedidos
- Rastreamento



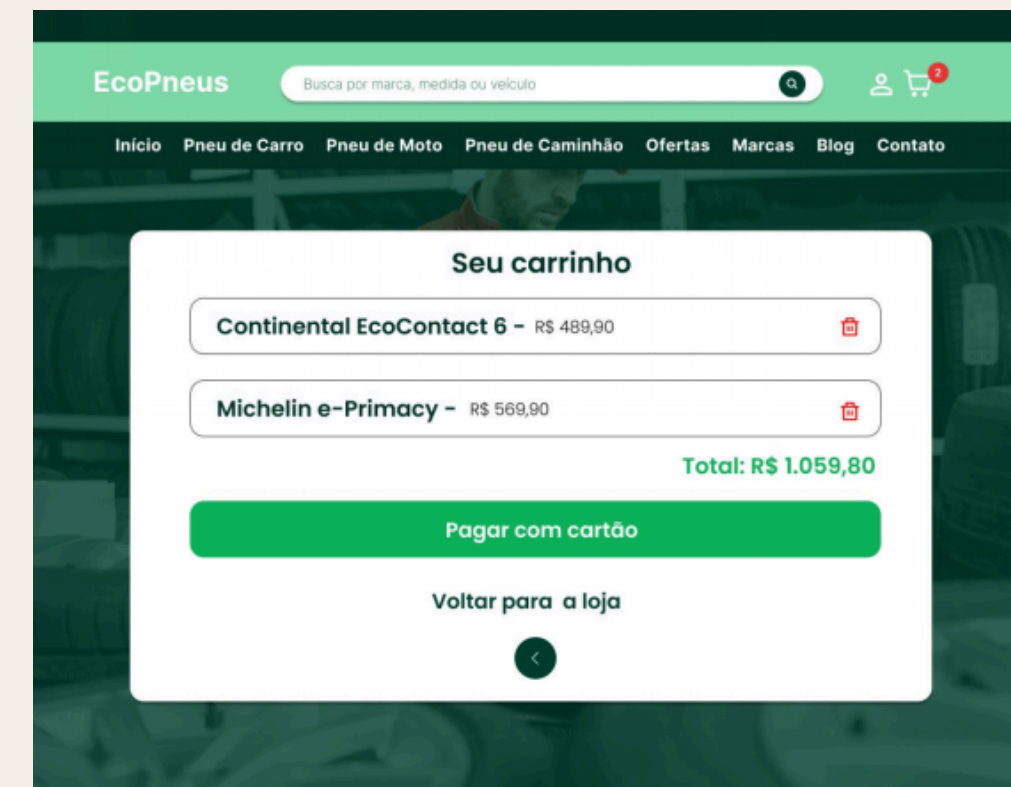
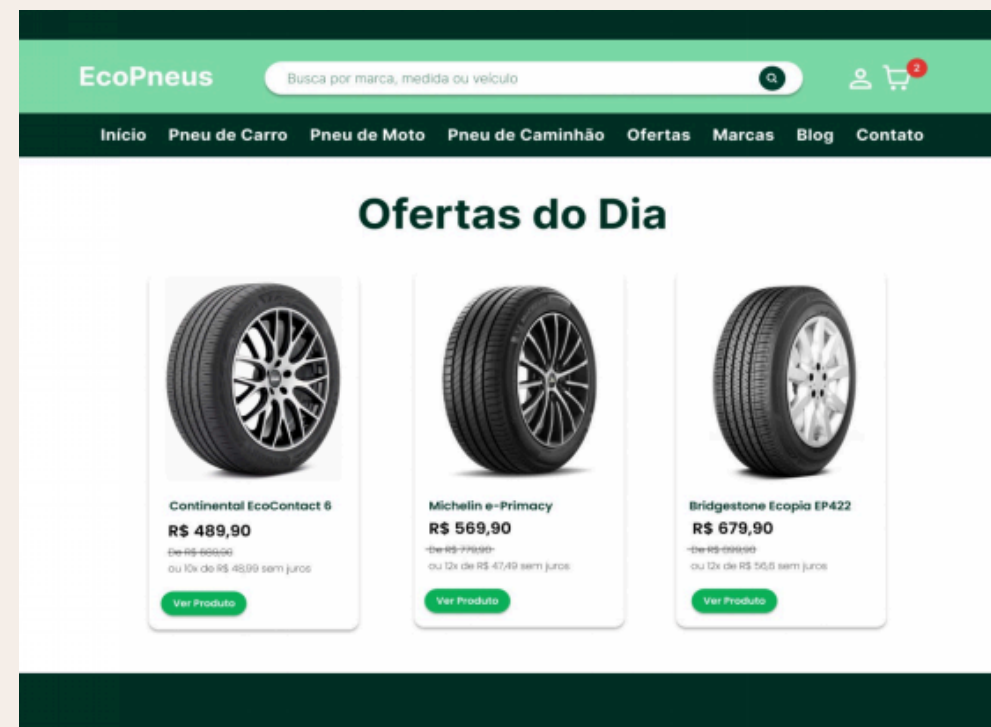
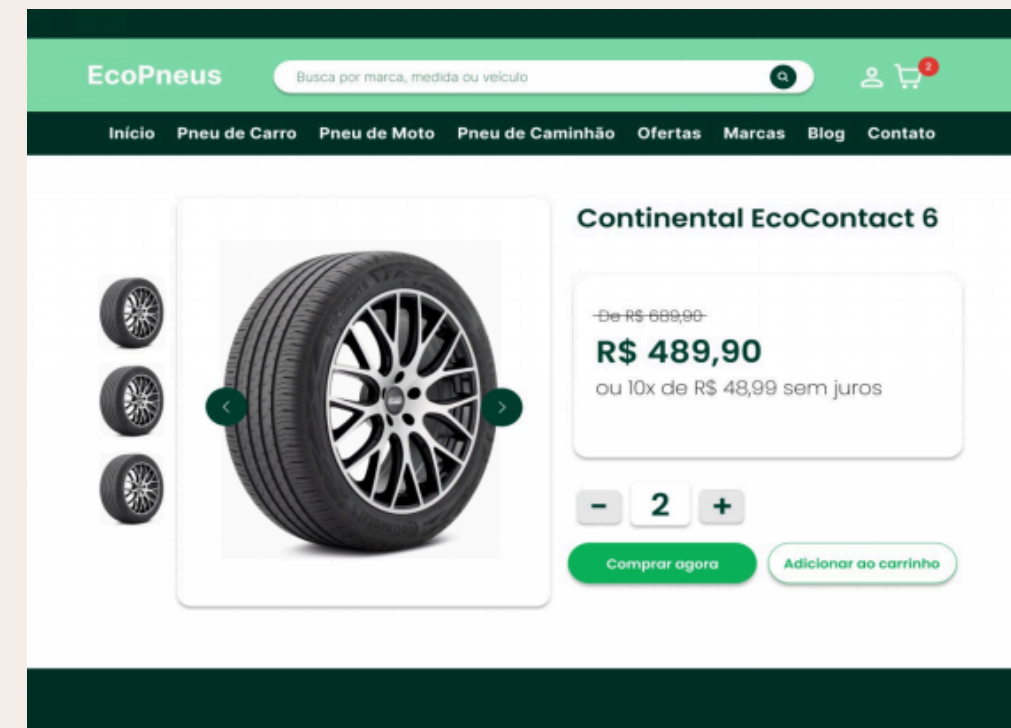
INTERFACE

Telas desenvolvidas:

- Home com catálogo
- Filtros
- Tela do produto
- Carrinho
- Login / Cadastro
- Checkout
- Área do cliente

Características do Design:

- Responsivo (mobile-first)
- Navegação simples
- Layout limpo
- Aplicação das heurísticas de Nielsen



<https://www.figma.com/proto/eix2jiSLIFqCsVWxoCV1GX/E-commerce-pneus?node-id=0-1&t=lqtYP7uUj8hFAzkD-1>

TECNOLOGIAS E CONFIGURAÇÕES UTILIZADAS

1. Linguagens e Tecnologias Principais

- Frontend:
 - HTML5
 - CSS3 (com responsividade e mobile-first)
 - JavaScript puro + AJAX
 - Possível uso de Bootstrap para acelerar layout
- Backend:
 - PHP 8+ seguindo o padrão MVC
 - Rotas organizadas
 - Controllers separados por funcionalidade
- Banco de Dados:
 - MySQL/MariaDB
 - Conexão utilizando PDO (seguro contra SQL Injection)
 - Relacionamentos normalizados entre tabelas

BANCO DE DADOS E ORGANIZAÇÃO

Modelagem (resumo)

- Tabelas: usuários, produtos, categorias, pedidos, itens, endereços, entregas.

Exemplos

- Produtos: nome, aro, marca, preço, estoque.
- Pedidos: usuário, total, status.
- Entregas: pedido, status (“Separando”, “Enviado”...).

Organização do Projeto (SCRUM resumido)

- EPICs: Catálogo, Seleção, Pagamento, Entrega.
- 4 Sprints principais (Catálogo → Checkout → Entrega → Dashboard).

SCRUM

Backlog do Produto

- Catálogo, filtros, produto, carrinho.
- Checkout, pagamento.
- Pedidos e rastreamento.
- Painel administrativo.

Sprints

- Sprint 1: Catálogo + filtros.
- Sprint 2: Login + checkout.
- Sprint 3: Entrega + integração CEP.
- Sprint 4: Dashboard + melhorias.

Acompanhamento

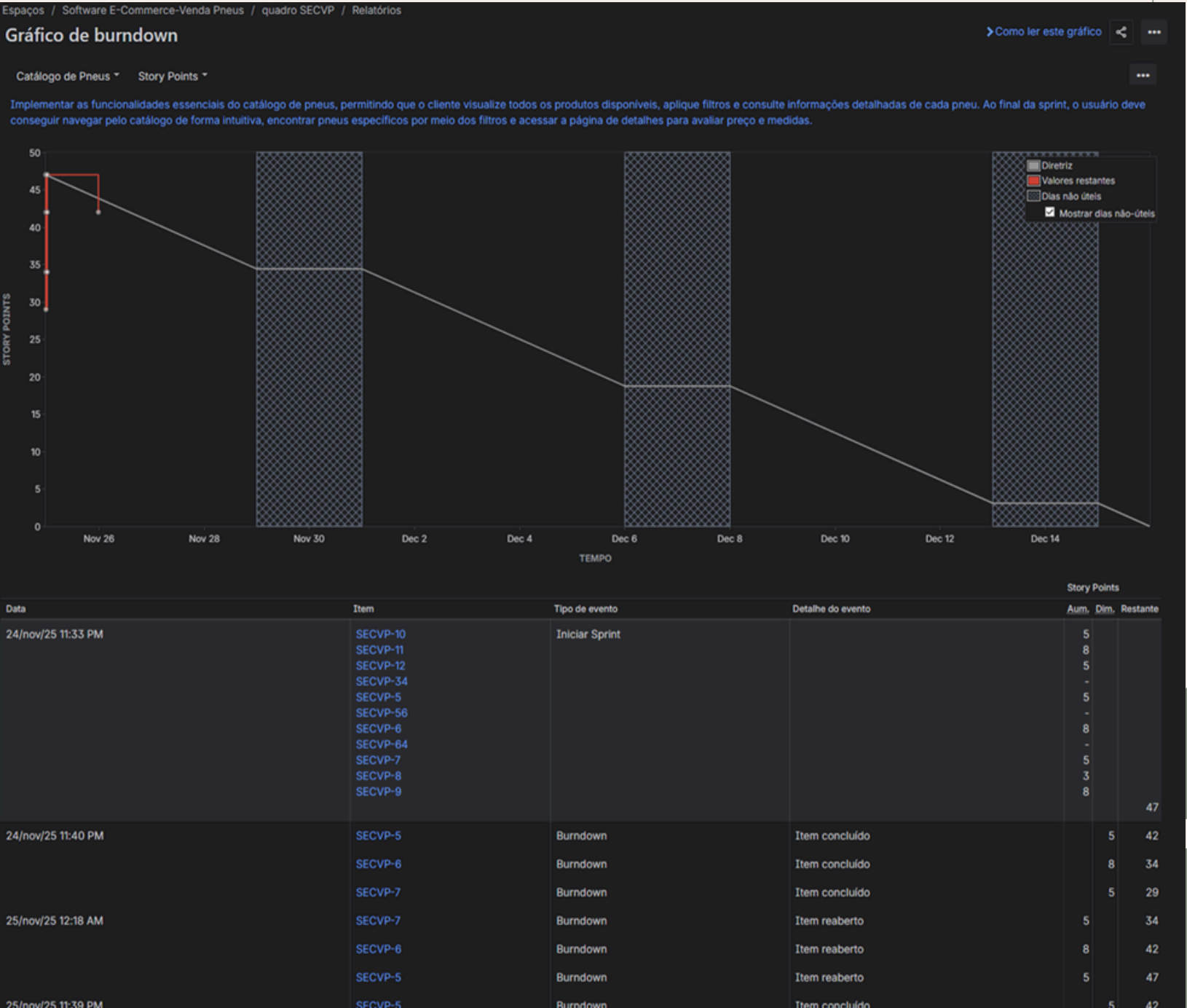
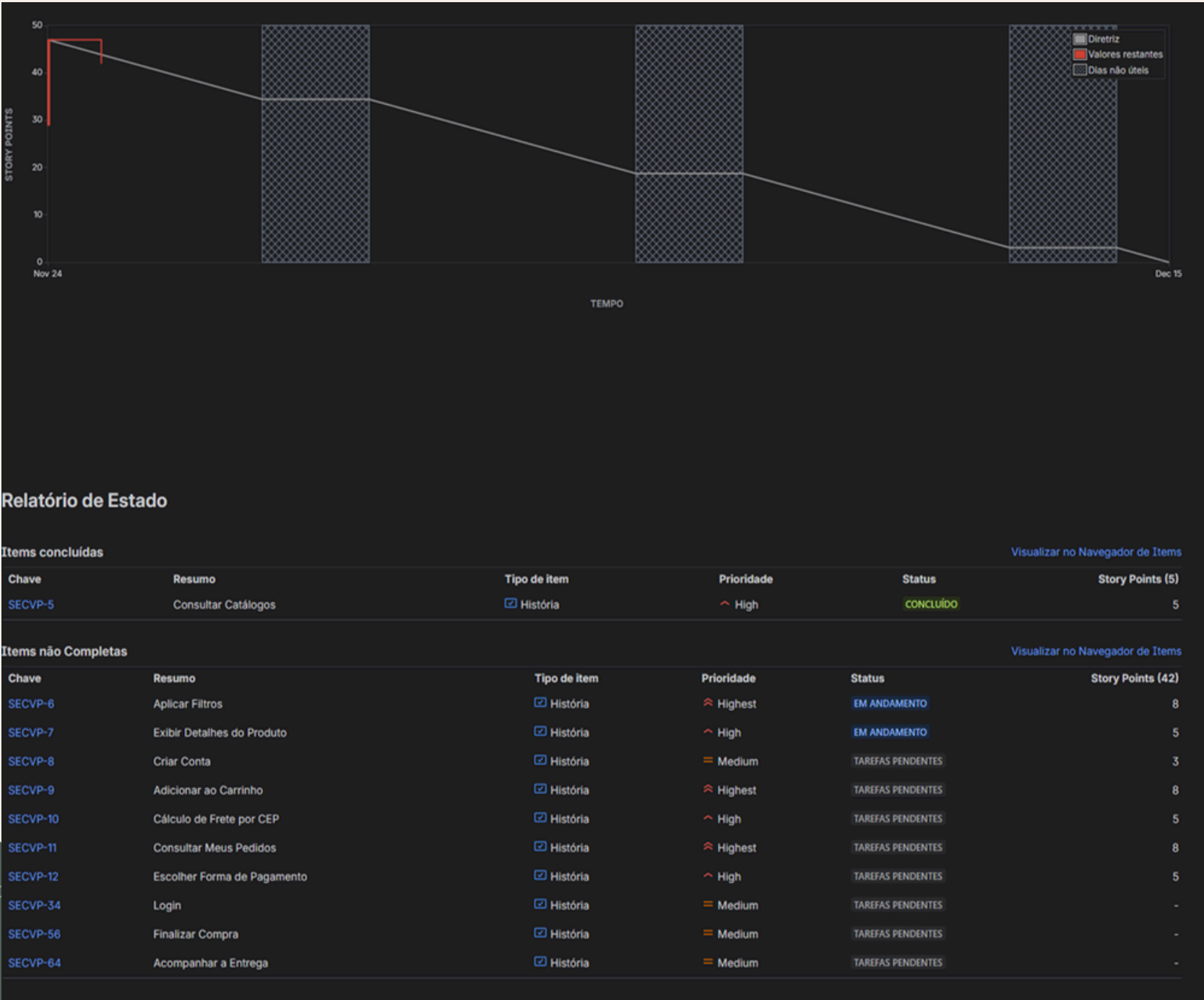
- Simulação de Sprint.
- Burndown Chart para medir progresso.

Jira:

<https://richardmoreira7.atlassian.net/jira/software/c/projects/SECVP/summary?issueParent=10008&atlOrigin=eyJpIjoiMDU3NGM2ZGVhMGFjNGE0M2IzZmNmODE2ZTYwZjRjNmliLCJwIjoiajI9>

SIMULAÇÃO DO SPRINT

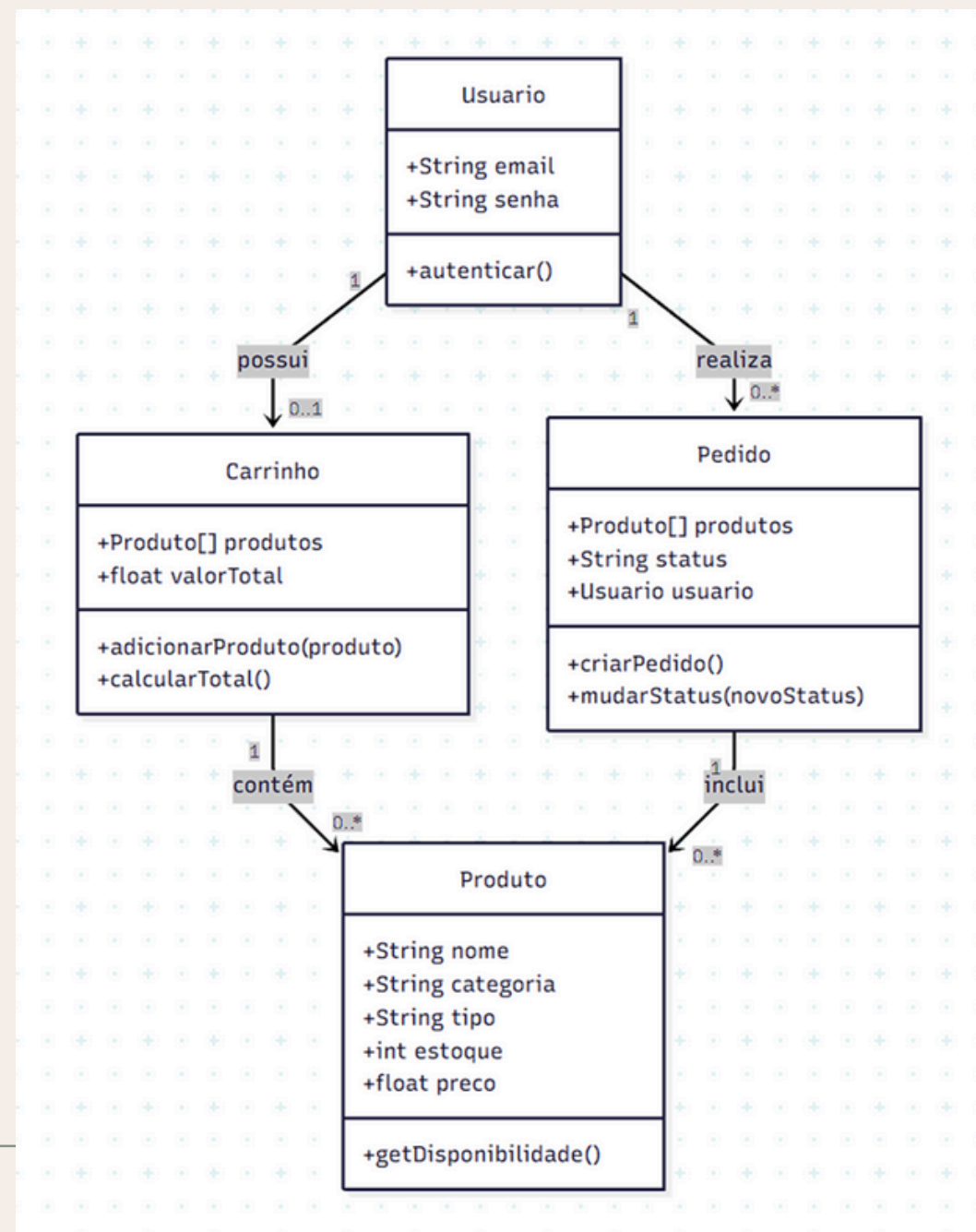
BURNDOWN CHART



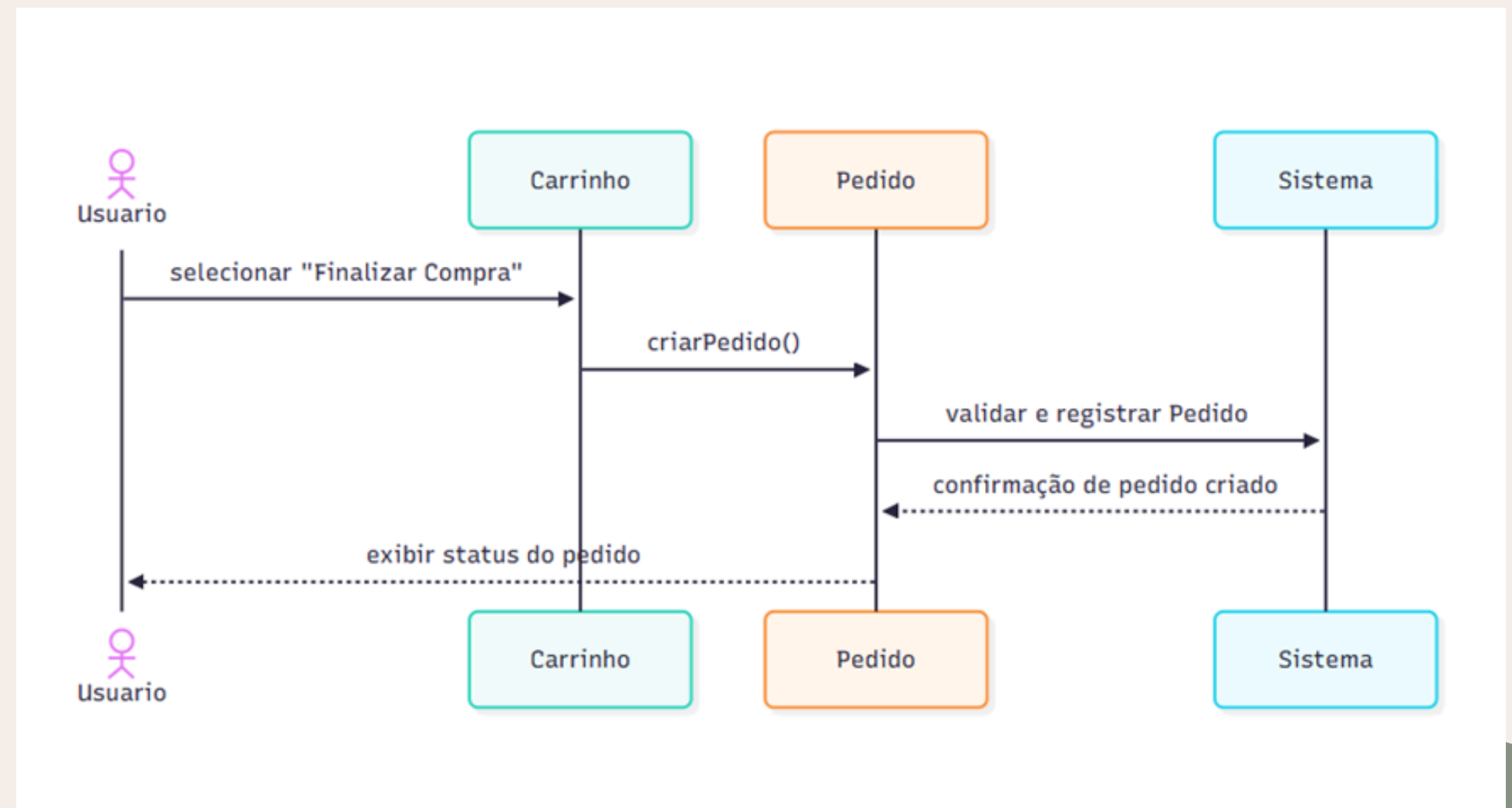
PROJETO - MODELAGEM

Diagramas Essenciais:

- Diagrama de Classes: estrutura das entidades do sistema

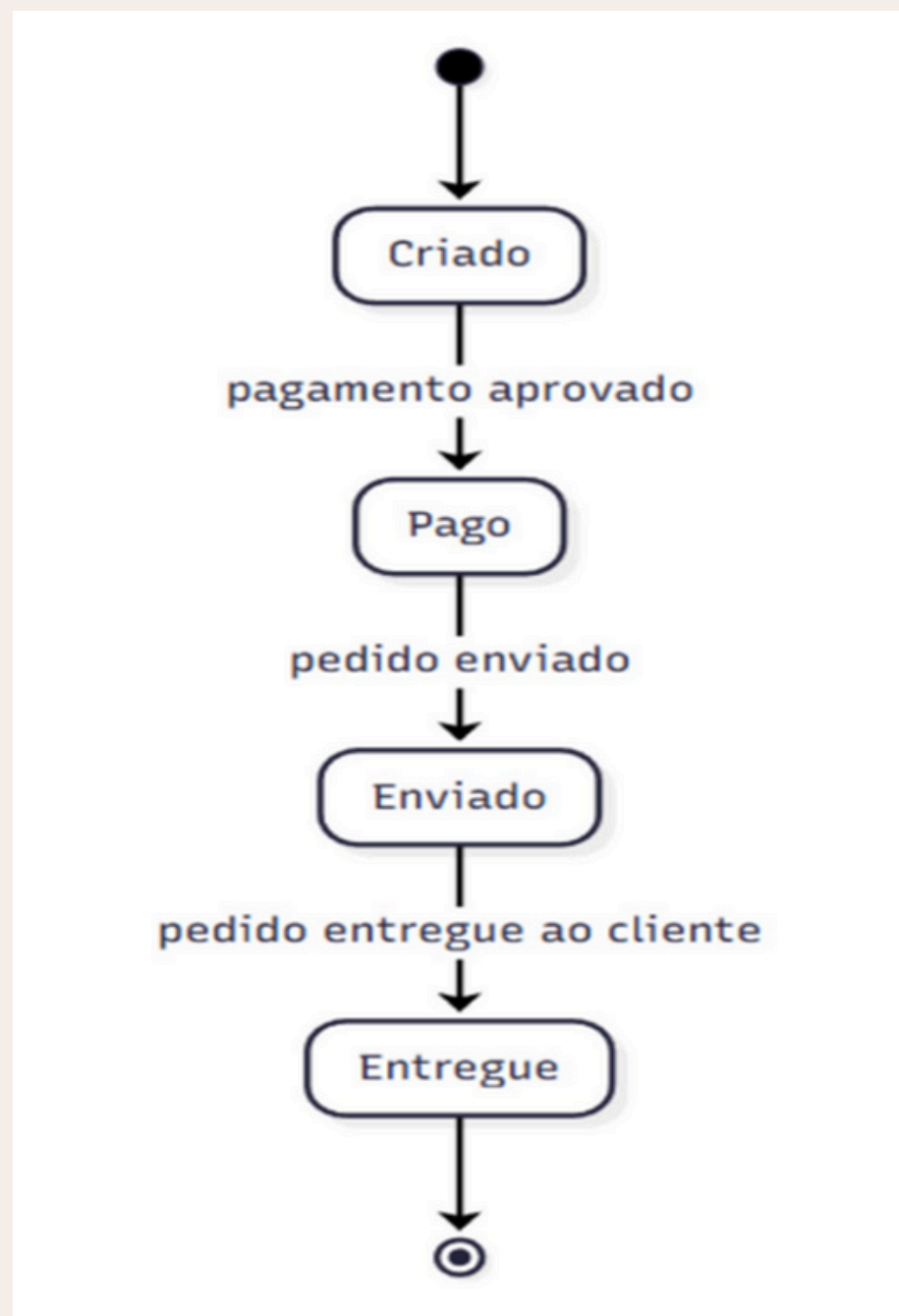


- Diagrama de Sequências: fluxo entre usuário, frontend e backend.



PROJETO - MODELAGEM

Máquina de Estados: ciclo do pedido (Criado → Pago → Enviado → Entregue).



Objetivo

- Garantir clareza na arquitetura.
- Facilitar implementação e testes.

TESTES

Objetivo Geral

- Garantir qualidade, estabilidade e funcionamento dos requisitos.

Escopo

- Catálogo, filtros, login, carrinho, pagamentos, pedidos, entrega.

Critérios de Aceitação

- 95% dos testes funcionando.
- Nenhum erro crítico.
- Páginas < 2s de carregamento.
- Pagamento 100% funcional no ambiente de teste.

Riscos

- Gateway de pagamento instável.
- Falha na transportadora.
- Alta carga inesperada.

Exemplos de CTs

- CT01: Cadastro de Usuário.
- CT02: Login.
- CT03: Consulta ao catálogo.
- CT04: Aplicar filtros.
- CT05: Exibir detalhes do produto.
- CT06: Adicionar ao carrinho.
- CT07: Finalizar compra.
- CT08: Pagamento com cartão.
- CT09: Acompanhar entrega.
- CT10: Cadastro/edição de produtos (admin).



OBRIGADO