

SISTEMA DE GERENCIAMENTO - ELEGÂNCIA PREMIUM

Table of Contents

- Documentação Técnica Completa
- SUMÁRIO EXECUTIVO
- 1. CANVAS DE MODELAGEM DE NEGÓCIO (Business Model Canvas)
- 2. LEVANTAMENTO DE REQUISITOS
- 3. DIAGRAMAS UML
- 4. MODELO DE DADOS E NORMALIZAÇÃO
- 5. ESTRUTURA DO BANCO DE DADOS
- 6. FLUXO OPERACIONAL COMPLETO
- 7. SEGURANÇA E COMPLIANCE
- 8. ESPECIFICAÇÕES TÉCNICAS
- 9. IMPLEMENTAÇÃO
- 10. CONCLUSÃO

Documentação Técnica Completa

SUMÁRIO EXECUTIVO

Projeto: Sistema Informatizado de Cadastro e Gerenciamento de Clientes, Produtos e Vendas

Cliente: Loja de Roupas "Elegância Premium"

Localização: Centro Comercial Silva Shopping, Rua das Flores, 450

Data: Novembro de 2025

Status: Proposta de Implementação

1. CANVAS DE MODELAGEM DE NEGÓCIO (Business Model Canvas)

Segmentos de Clientes

- **Clientes varejistas:** Compradores finais de roupas (ex: Mariana Ferreira)
- **Gerência:** Maria da Silva - controle operacional e relatórios
- **Operacional:** Carlos (vendedor), Patricia (estoque), equipe
- **Fornecedores:** Integração com sistemas de reposição

Proposta de Valor

- ✓ Controle de estoque em tempo real
- ✓ Consulta rápida de disponibilidade (cor/tamanho)
- ✓ Redução de erros operacionais
- ✓ Geração automática de relatórios
- ✓ Rastreabilidade completa de transações
- ✓ Interface responsiva para PDV

Canais de Distribuição

- Interface web responsiva (desktop/tablet)
- Terminal PDV com acesso aos dados
- Painéis administrativos
- Relatórios digitais

Relacionamento com Clientes

- Suporte técnico 24h
- Treinamento de usuários
- Backup e recuperação de dados
- Atualização contínua do sistema

Fontes de Receita

- Licenciamento do software
- Suporte técnico
- Consultoria em gestão
- Customizações futuras

Recursos Principais

- Banco de dados robusto (MySQL)
- Backend escalável (Flask)
- Interface responsiva (HTML5/CSS3/JavaScript)
- Sistema de autenticação e permissões
- Gerador de relatórios

Atividades-Chave

- Cadastro de produtos/coleções
- Controle de estoque
- Processamento de vendas
- Geração de relatórios
- Gerenciamento de promoções
- Auditoria e logs

Parceiros-Chave

- Fornecedores de roupas
- Provedores de hosting
- Consultores de RH

Estrutura de Custos

- Desenvolvimento: Horas-desenvolvedor
- Infraestrutura: Servidor, backup
- Manutenção: Suporte e atualizações
- Treinamento: Equipe da loja

2. LEVANTAMENTO DE REQUISITOS

2.1 Requisitos Funcionais (RF)

ID	Descrição	Prioridade	Módulo
RF-001	Cadastro de Clientes (nome, CPF, email, telefone, endereço)	ALTA	Clientes
RF-002	Cadastro de Fornecedores	MÉDIA	Fornecedores
RF-003	Cadastro de Coleções (nome, descrição, período)	ALTA	Produtos
RF-004	Cadastro de Produtos (nome, descrição, preço, coleção)	ALTA	Produtos
RF-005	Cadastro de Cores	ALTA	Produtos
RF-006	Cadastro de Tamanhos	ALTA	Produtos
RF-007	Gerenciar Variações de Produtos (cor x tamanho x coleção)	ALTA	Produtos
RF-008	Controle de Estoque em Tempo Real	ALTA	Estoque
RF-009	Registrar Movimentação de Estoque (entrada/saída)	ALTA	Estoque
RF-010	Consulta de Disponibilidade (produto + cor + tamanho)	ALTA	Estoque
RF-011	Registrar Venda com CPF do Cliente	ALTA	Vendas

ID	Descrição	Prioridade	Módulo
RF-012	Registrar Devolução de Produtos	ALTA	Vendas
RF-013	Cadastro de Promoções (desconto, data início/fim)	MÉDIA	Promoções
RF-014	Ativação/Desativação Automática de Promoções	MÉDIA	Promoções
RF-015	Cadastro de Usuários com Permissões	ALTA	Segurança
RF-016	Autenticação por Login/Senha	ALTA	Segurança
RF-017	Relatório de Vendas por Período	ALTA	Relatórios
RF-018	Relatório de Vendas por Vendedor	ALTA	Relatórios
RF-019	Relatório de Vendas por Coleção	ALTA	Relatórios
RF-020	Relatório de Produtos Mais Vendidos	MÉDIA	Relatórios
RF-021	Relatório de Estoque Baixo	MÉDIA	Relatórios
RF-022	Relatório de Performance de Promoções	MÉDIA	Relatórios
RF-023	Exportar Relatórios em PDF	MÉDIA	Relatórios
RF-024	Visualização de Histórico de Operações (Logs)	ALTA	Auditoria
RF-025	Validação de Aprovação de Relatório por Gerente	MÉDIA	Relatórios

2.2 Requisitos Não Funcionais (RNF)

ID	Descrição	Métrica
RNF-001	Performance: Consulta de estoque < 500ms	< 500ms
RNF-002	Disponibilidade: Uptime do sistema	99%
RNF-003	Segurança: Criptografia de senhas	bcrypt
RNF-004	Segurança: Backup noturno automático	Diariamente 23h
RNF-005	Segurança: Logs de todas operações	100% rastreabilidade
RNF-006	Responsividade: Interface para tablet (7-10")	iOS/Android
RNF-007	Compatibilidade: Navegadores modernos	Chrome 90+, Safari, Firefox
RNF-008	Escalabilidade: Suporte até 100 transações/min	100 tx/min
RNF-009	Usabilidade: Tempo de aprendizado < 2h	2 horas
RNF-010	Manutenibilidade: Código documentado	80% cobertura

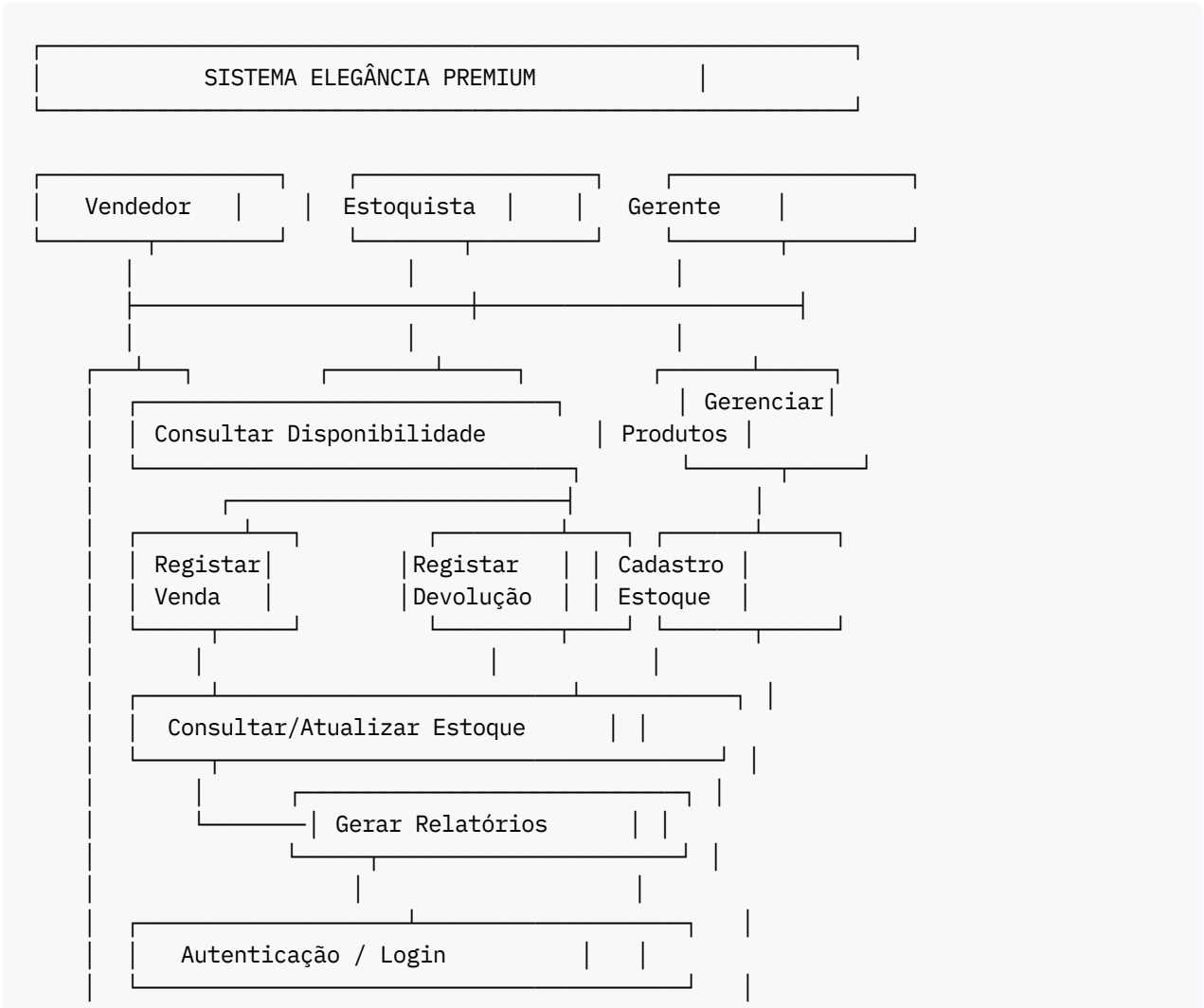
2.3 Regras de Negócio (RN)

ID	Regra	Impacto
RN-001	Produto só cadastrado mediante indicação obrigatória de coleção	Integridade de dados
RN-002	Toda variação de produto (cor/tamanho) deve ter estoque registrado	Controle de estoque

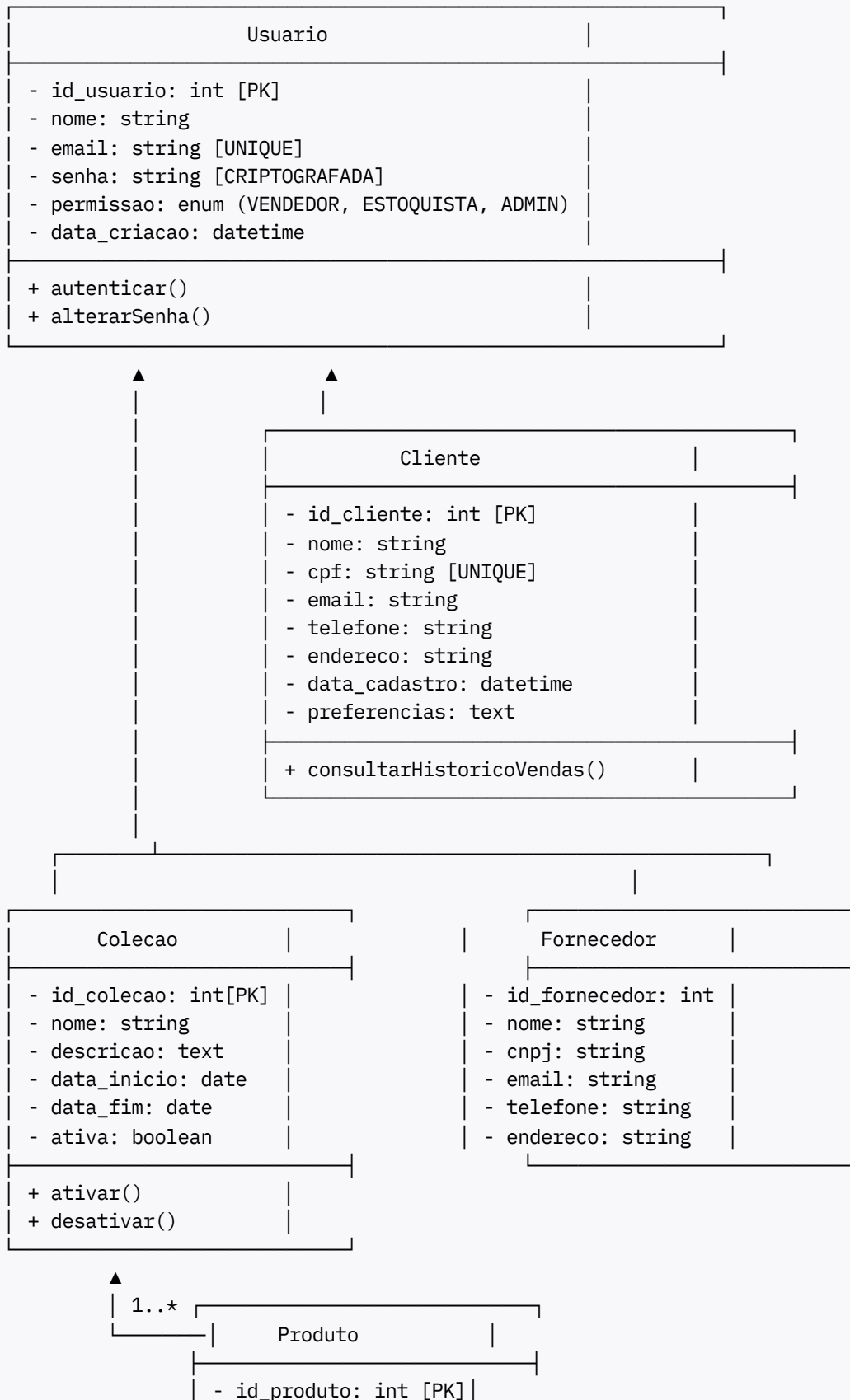
ID	Regra	Impacto
RN-003	Venda não é permitida se estoque insuficiente	Evita sobrevenda
RN-004	Promoção só ativa dentro das datas programadas	Controle promocional
RN-005	Cada movimento de estoque gera log com usuário/data/hora	Auditoria
RN-006	Estoquista: consulta estoque, atualiza quantidade (permissão restrita)	Segurança
RN-007	Vendedor: consulta disponibilidade, registra vendas	Segurança
RN-008	Gerente: acesso total + aprovação de relatórios	Segurança
RN-009	Preço da venda aplica promoção automaticamente se ativa	Automação
RN-010	Relatório semanal validado por gerente com registro digital	Compliance
RN-011	Devolução de produto repõe automaticamente estoque	Automação
RN-012	Cliente duplicado: validação por CPF	Integridade

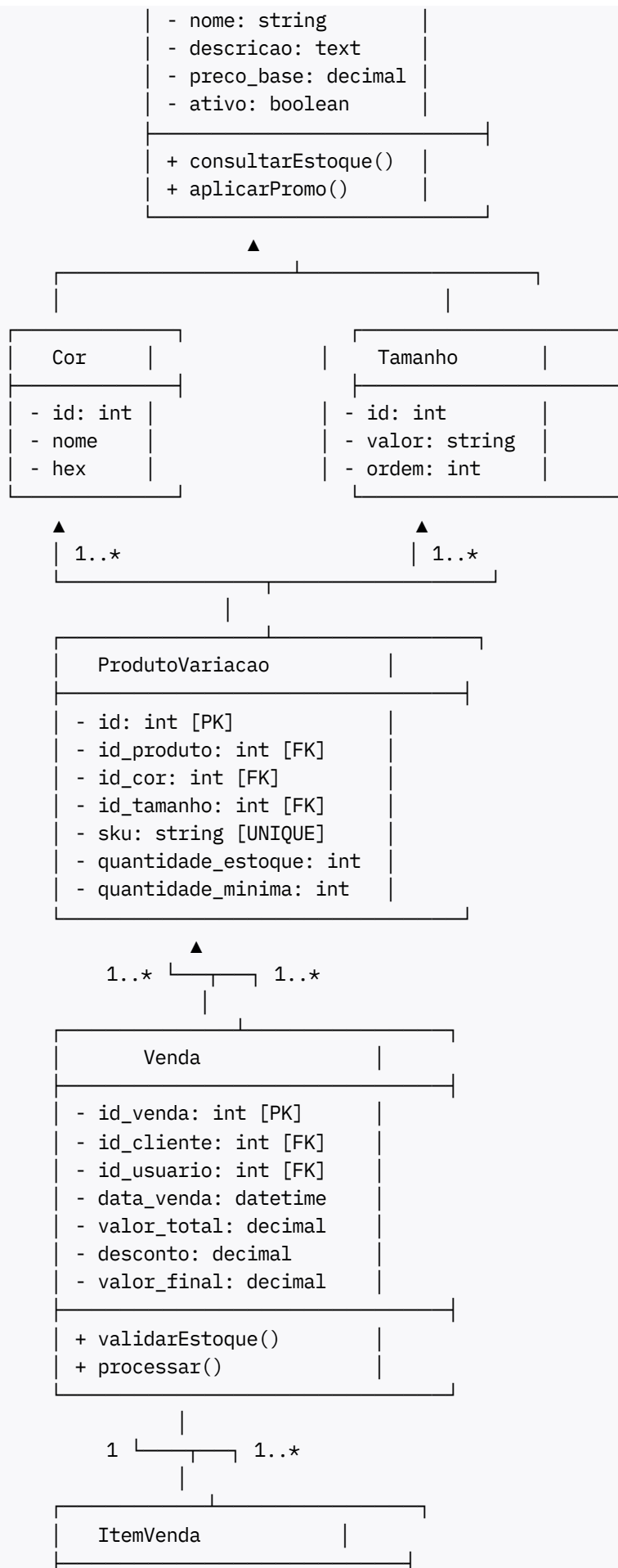
3. DIAGRAMAS UML

3.1 Diagrama de Casos de Uso

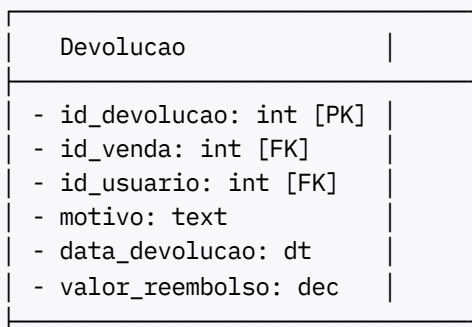
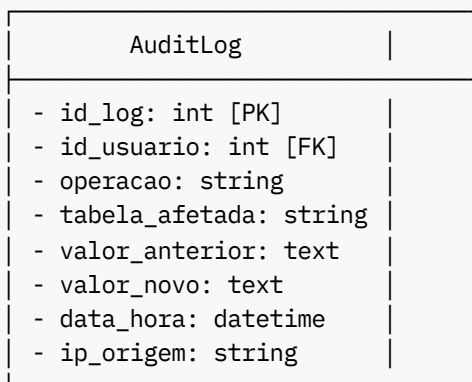
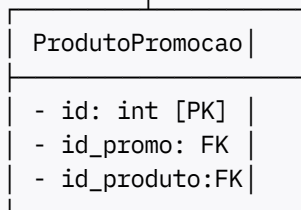
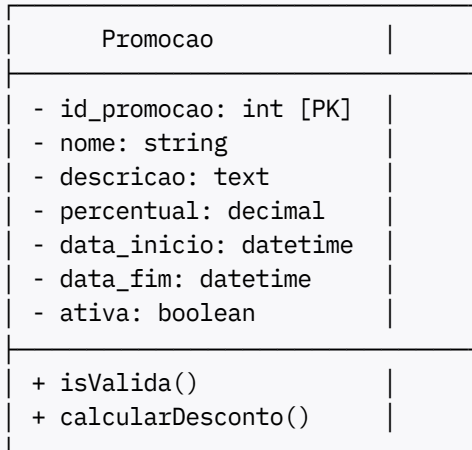


3.2 Diagrama de Classes





- id: int [PK]
- id_venda: int [FK]
- id_variacao: int [FK]
- quantidade: int
- preco_unitario: dec
- desconto: decimal




```
+ processar()
+ reporEstoque()
```

4. MODELO DE DADOS E NORMALIZAÇÃO

4.1 Processo de Normalização até 3FN

Primeira Forma Normal (1FN)

Objetivo: Eliminar atributos multivalorados

Passo 1: Separar cores e tamanhos em tabelas normalizadas

- ANTES: Produto (cores[], tamanhos[])
- DEPOIS: Cor, Tamanho, ProdutoVariacao

Passo 2: Cada célula contém apenas um valor atômico

- Endereço de cliente em campo único (não separado)
- CPF normalizado (remover caracteres especiais)

Segunda Forma Normal (2FN)

Objetivo: Eliminar dependências parciais (atributo não chave depende apenas de parte da chave)

Passo 1: ItemVenda depende completamente de (id_venda, id_variacao)

- id_variacao → id_produto (dependência removida por FK)
- Apenas dados específicos do item: quantidade, preço

Passo 2: ProdutoVariacao depende completamente de (id_produto, id_cor, id_tamanho)

- Estoque, SKU relacionados à variação específica

Terceira Forma Normal (3FN)

Objetivo: Eliminar dependências transitivas ($X \rightarrow Y \rightarrow Z$, onde X não é chave)

Passo 1: Cliente → Preferencias (não dependência transitiva)

- Preferências armazenadas como JSON em campo separado
- Endereço completo em um campo de texto

Passo 2: Produto → Colecao (sem dependência de outros atributos)

- Apenas relação 1:N, sem outros atributos transitivos

Passo 3: Venda → ItemVenda → ProdutoVariacao

- Cada entidade tem sua responsabilidade clara
- Sem campos redundantes

5. ESTRUTURA DO BANCO DE DADOS

[Ver arquivo SQL separado: elegancia_premium.sql]

Principais Tabelas:

1. **usuarios** - Controle de acesso
2. **clientes** - Dados dos compradores
3. **colecoes** - Agrupamento de produtos
4. **fornecedores** - Cadastro de fornecedores
5. **produtos** - Catálogo de produtos
6. **cores** - Paleta de cores
7. **tamanhos** - Grade de tamanhos
8. **produto_variacao** - SKU com estoque
9. **vendas** - Transações de venda
10. **item_venda** - Itens por venda
11. **promocoes** - Campanhas promocionais
12. **produto_promocao** - Associação produto-promo
13. **devolucoes** - Registro de devoluções
14. **audit_log** - Auditoria completa

6. FLUXO OPERACIONAL COMPLETO

Cenário: Venda com Reserva e Devolução Posterior

Dia 1 - Venda Inicial:

1. **Mariana chega na loja** procurando "Camiseta Floral Primavera 2025"
2. **Carlos (Vendedor)** acessa o tablet:
 - Login no sistema
 - Consulta "Produtos" → Filtra por "Primavera 2025"
 - Encontra: Floral Rosa, M, 3 unidades disponíveis
3. **Carlos registra a venda:**
 - CPF da Mariana (123.456.789-00)
 - Produto: Floral Primavera 2025, Tamanho M, Cor Rosa

- Quantidade: 1 unidade
- Sistema valida estoque (✓ tem 3)
- Aplica promoção ativa "VERÃO-10%" (se válida)
- Gera Venda #1001
- Estoque reduz de 3 para 2 automaticamente
- Log gerado: "Venda #1001 - Usuário: Carlos - 14/11/2025 15:30"

4. Patricia (Estoque) - Final do dia:

- Acessa "Relatórios" → "Estoque Baixo"
- Identifica produtos com estoque \leq quantidade_minima
- Gera requisição de compra para fornecedor

Dia 7 - Devolução:

1. **Mariana retorna** com a camiseta (não gostou)

2. Carlos registra devolução:

- Busca Venda #1001 (por CPF ou número)
- Motivo: "Não gostou do tamanho"
- Sistema cria Devolucao #001
- Reembolso: R\$ 89,90 (com desconto aplicado)
- Estoque repostado automaticamente: $2 + 1 = 3$ unidades
- Log: "Devolução #001 - Venda #1001 - Usuário: Carlos"

Semana 2 - Relatório:

1. Maria (Gerente) acessa painel:

- Relatório Semanal: "14/11 - 20/11"
- Vendas por Coleção: Primavera 2025 = R\$ 4.500 (12 unidades)
- Vendas por Vendedor: Carlos = 8 vendas, Patricia = 4 vendas
- Devoluções: 1 (taxa 4,2%)
- Promoção "VERÃO-10%": 6 produtos vendidos, economia do cliente = R\$ 54
- **Aprova relatório digitalmente**
- Log: "Relatório aprovado por Maria em 20/11/2025 10:15"

7. SEGURANÇA E COMPLIANCE

Autenticação

- Login obrigatório
- Senha criptografada com bcrypt (sha256+salt)
- Sessão com timeout de 30 minutos

Autorização (Permissões)

- **VENDEDOR:** Consulta estoque, registra vendas, consulta dados do cliente
- **ESTOQUISTA:** Consulta estoque, atualiza quantidade, consulta movimentações
- **GERENTE (ADMIN):** Acesso total, aprova relatórios, gera backups

Proteção de Dados

- Criptografia de campos sensíveis (CPF, senha)
- Validação de entrada (SQL injection, XSS)
- HTTPS obrigatório em produção
- Backups noturnos automatizados (23:00)
- Retenção de logs: 2 anos

Auditoria (AuditLog)

Cada operação registra:

- Usuário que executou
- Data/hora exata
- Operação (INSERT, UPDATE, DELETE)
- Tabela afetada
- Valor anterior (se UPDATE/DELETE)
- Valor novo (se INSERT/UPDATE)
- IP de origem

8. ESPECIFICAÇÕES TÉCNICAS

Stack Tecnológico

- **Backend:** Python 3.9+ com Flask
- **Banco de Dados:** MySQL 5.7+
- **Frontend:** HTML5, CSS3, JavaScript ES6+
- **Servidor:** Gunicorn + Nginx
- **Versionamento:** Git

Requisitos de Hardware (Mínimo)

- CPU: 2 cores
- RAM: 4GB
- Armazenamento: 50GB (banco + backups)
- Conexão: 10Mbps

Performance

- Consultas < 500ms (índices em campos principais)
- Cache de sessão em memória
- Lazy loading de relatórios grandes
- Backup incremental diário

9. IMPLEMENTAÇÃO

Fase 1: Infraestrutura (Semana 1-2)

- ☒ Criar banco de dados
- ☒ Configurar servidor
- ☒ Estrutura do projeto Flask

Fase 2: Backend (Semana 3-4)

- ☐ Modelos de dados
- ☐ API de autenticação
- ☐ Lógica de negócio

Fase 3: Frontend (Semana 5-6)

- ☐ Telas responsivas
- ☐ Integração com API
- ☐ Validações client-side

Fase 4: Testes e Deploy (Semana 7)

- ☐ Testes unitários
- ☐ Testes de integração
- ☐ Deploy em produção

10. CONCLUSÃO

Este sistema fornecerá à "Elegância Premium" uma solução robusta, segura e escalável para modernizar suas operações, eliminar erros manuais e fornecer inteligência de dados para decisões estratégicas.

Investimento: Altamente recomendado para crescimento operacional.