**Mercado Bornelli**

**Sistema de Gestão Integrada para Administração, Consulta de Preços e Totem de Autoatendimento**

,

**Relatório Sistêmico**

**Mercado Bornelli**

Trabalho apresentado à disciplina Desenvolvimento de Sistemas do curso Técnico em Análise e Desenvolvimento de Sistemas,  
sob orientação do professor Matheus Carvalho

**TUBARÃO  
2025**

# RESUMO

Este relatório apresenta a documentação técnica do sistema integrado de gestão desenvolvido para o Mercado Bornelli, composto por três módulos principais: administração interna, consulta de preços e totens de autoatendimento. O sistema visa otimizar processos operacionais e administrativos, proporcionando maior eficiência, controle e autonomia para clientes e funcionários. Utilizando arquitetura cliente-servidor com backend em Spring Boot e frontend em JavaScript, HTML e CSS, o sistema implementa funcionalidades como cadastro e gerenciamento de produtos, clientes, fornecedores e funcionários, além de consultas rápidas e autoatendimento para clientes. A documentação detalha a arquitetura, funcionalidades, tecnologias empregadas, implementação técnica e resultados obtidos. Destaca-se a preocupação com a usabilidade, segurança e integração dos módulos para atender às demandas do mercado varejista.

**RELATÓRIO SISTÊMICO - SISTEMA MERCADO BORNELLI**

**SUMÁRIO**

1. **INTRODUÇÃO**
   * 1.1 Objetivo do Sistema
   * 1.2 Escopo do Projeto
   * 1.3 Tecnologias Utilizadas
2. **VISÃO GERAL DO SISTEMA**
   * 2.1 Módulos Principais
   * 2.2 Fluxo de Operação
   * 2.3 Usuários e Perfis
3. **ARQUITETURA DO SISTEMA**
   * 3.1 Arquitetura Geral
   * 3.2 Arquitetura Backend
   * 3.3 Arquitetura Frontend
   * 3.4 Integração e Comunicação
4. **MÓDULO DE ADMINISTRAÇÃO**
   * 4.1 Dashboard
   * 4.2 Gestão de Produtos
   * 4.3 Gestão de Clientes
   * 4.4 Gestão de Fornecedores
   * 4.5 Gestão de Funcionários
   * 4.6 Controle de Vendas
   * 4.7 Gestão de Feedbacks
5. **MÓDULO DE CONSULTA DE PREÇOS**
   * 5.1 Interface de Consulta
   * 5.2 Fluxo de Consulta
6. **MÓDULO TOTEM DE AUTOATENDIMENTO**
   * 6.1 Interface do Totem
   * 6.2 Processo de Venda
   * 6.3 Sistema de Feedback
7. **SEGURANÇA E AUTENTICAÇÃO**
   * 7.1 Autenticação JWT
   * 7.2 Controle de Acesso
   * 7.3 Perfis de Usuário
8. **BANCO DE DADOS**
   * 8.1 Modelo de Dados
   * 8.2 Relacionamentos
   * 8.3 Estratégias de Persistência
9. **FUNCIONALIDADES TÉCNICAS**
   * 9.1 API REST
   * 9.2 Tratamento de Exceções
   * 9.3 Validações
   * 9.4 Paginação e Filtros
10. **RELATÓRIOS E ANÁLISES**
    * 10.1 Relatórios de Vendas
    * 10.2 Análise de Feedbacks
    * 10.3 Indicadores de Desempenho
11. **CONSIDERAÇÕES FINAIS**
    * 11.1 Pontos Fortes
    * 11.2 Possíveis Melhorias
    * 11.3 Conclusão

**LISTA DE FIGURAS**

* **Figura 1** - Arquitetura Geral do Sistema Mercado Bornelli
* **Figura 2** - Diagrama de Classes do Backend
* **Figura 3** - Estrutura de Pastas do Frontend
* **Figura 4** - Tela de Login do Sistema
* **Figura 5** - Tela do Dashboard Administrativo
* **Figura 6** - Tela de Gestão de Produtos
* **Figura 7** - Tela de Gestão de Clientes
* **Figura 8** - Tela de Gestão de Fornecedores
* **Figura 9** - Tela de Gestão de Funcionários
* **Figura 10** - Tela de Controle de Vendas
* **Figura 11** - Tela de Gestão de Feedbacks
* **Figura 12** - Interface de Consulta de Preços
* **Figura 13** - Interface do Totem de Autoatendimento
* **Figura 14** - Fluxo de Venda no Totem
* **Figura 15** - Tela de Feedback do Cliente
* **Figura 16** - Diagrama de Banco de Dados
* **Figura 17** - Fluxo de Autenticação JWT
* **Figura 18** - Estrutura da API REST
* **Figura 19** - Relatório de Vendas
* **Figura 20** - Análise de Feedbacks

**1. INTRODUÇÃO**

**1.1 Objetivo do Sistema**

O Sistema Mercado Bornelli foi desenvolvido para modernizar e otimizar as operações de um mercado, oferecendo soluções integradas para gestão administrativa, atendimento ao cliente e automação de vendas. O sistema visa proporcionar maior eficiência operacional, melhor experiência do cliente e controle gerencial aprimorado.

**1.2 Escopo do Projeto**

O projeto abrange três módulos principais:

* **Módulo Administrativo**: Gestão completa de produtos, clientes, fornecedores, funcionários, vendas e feedbacks
* **Módulo de Consulta de Preços**: Terminal para consulta rápida de preços por código de barras
* **Módulo Totem de Autoatendimento**: Sistema de vendas autônomas com coleta de feedback

**1.3 Tecnologias Utilizadas**

**Backend:**

* Java 17
* Spring Boot 3.x
* Spring Security com JWT
* Spring Data JPA
* MySQL 8.0
* Maven

**Frontend:**

* HTML5
* CSS3
* JavaScript (ES6+)
* Font Awesome (ícones)
* Design Responsivo

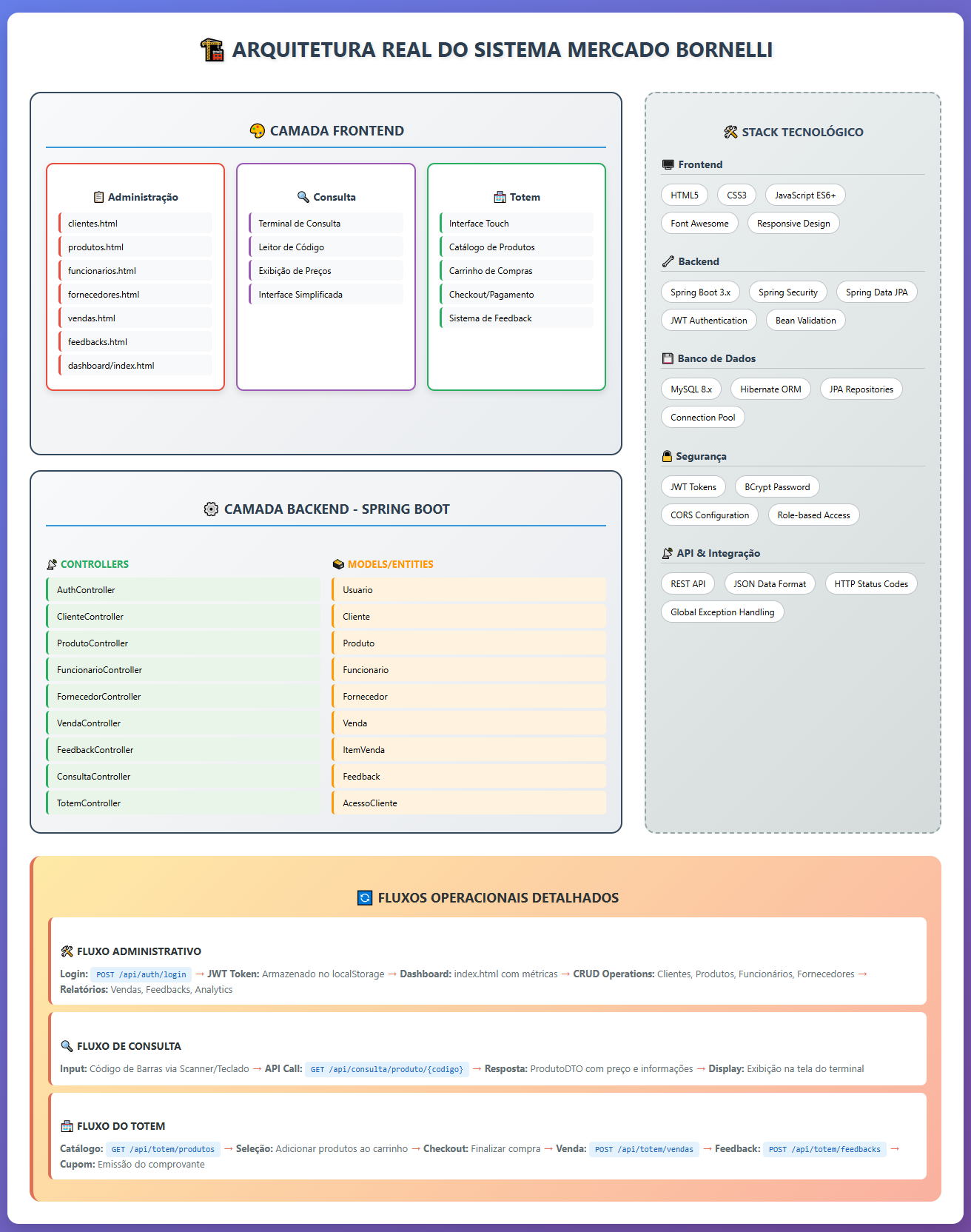
**Ferramentas:**

* MySQL Workbench
* Git para controle de versão
* IDE Spring Tool Suite/IntelliJ IDEA
* Visual Studio Code

**2. VISÃO GERAL DO SISTEMA**

**2.1 Módulos Principais**

FIGURA 1 - Arquitetura Geral do Sistema Mercado Bornelli



O sistema é composto por três módulos independentes, mas integrados:

1. **Administração**: Painel completo para gestão do negócio
2. **Consulta de Preços**: Terminal simplificado para clientes
3. **Totem de Autoatendimento**: Sistema de vendas self-service

**2.2 Fluxo de Operação**

O sistema opera com os seguintes fluxos principais:

* **Fluxo Administrativo**: Login → Dashboard → Operações CRUD → Relatórios
* **Fluxo de Consulta**: Leitura código → Exibição de preço
* **Fluxo do Totem**: Seleção produtos → Finalização venda → Feedback

**2.3 Usuários e Perfis**

* **ADMIN**: Acesso total ao sistema
* **USER**: Acesso às funcionalidades operacionais
* **Cliente**: Acesso aos módulos públicos (consulta e totem)

**3. ARQUITETURA DO SISTEMA**

**3.1 Arquitetura Geral**

O sistema adota arquitetura REST com separação clara entre frontend e backend:

FIGURA 2 - Diagrama de Classes do Backend

**Camadas do Backend:**

* **Controller**: Endpoints REST
* **Service**: Lógica de negócio
* **Repository**: Acesso a dados
* **Model**: Entidades JPA
* **DTO**: Objetos de transferência
* **Security**: Autenticação e autorização
* **Config**: Configurações do sistema
* **Exception**: Tratamento de erros

**3.2 Arquitetura Backend**

**Estrutura de Pacotes:**

com.mercadobornelli/

├── config/

│ ├── CorsConfig.java

│ ├── DataInitializer.java

│ └── SecurityConfig.java

├── controller/

│ ├── AuthController.java

│ ├── ClienteController.java

│ ├── ConsultaController.java

│ ├── FeedbackController.java

│ ├── FornecedorController.java

│ ├── FuncionarioController.java

│ ├── ProdutoController.java

│ ├── TotemController.java

│ └── VendaController.java

├── dto/

│ ├── ClienteDTO.java

│ ├── FeedbackDTO.java

│ ├── FornecedorDTO.java

│ ├── FuncionarioDTO.java

│ ├── ItemVendaDTO.java

│ ├── JwtResponseDTO.java

│ ├── LoginDTO.java

│ ├── ProdutoDTO.java

│ ├── UsuarioDTO.java

│ └── VendaDTO.java

├── exception/

│ ├── GlobalExceptionHandler.java

│ └── ResourceNotFoundException.java

├── model/

│ ├── AcessoCliente.java

│ ├── Cliente.java

│ ├── Feedback.java

│ ├── Fornecedor.java

│ ├── Funcionario.java

│ ├── ItemVenda.java

│ ├── Produto.java

│ ├── Usuario.java

│ └── Venda.java

├── repository/

│ ├── AcessoClienteRepository.java

│ ├── ClienteRepository.java

│ ├── FeedbackRepository.java

│ ├── FornecedorRepository.java

│ ├── FuncionarioRepository.java

│ ├── ProdutoRepository.java

│ ├── UsuarioRepository.java

│ └── VendaRepository.java

├── security/

│ ├── AuthTokenFilter.java

│ ├── JwtUtils.java

│ └── UserDetailsServiceImpl.java

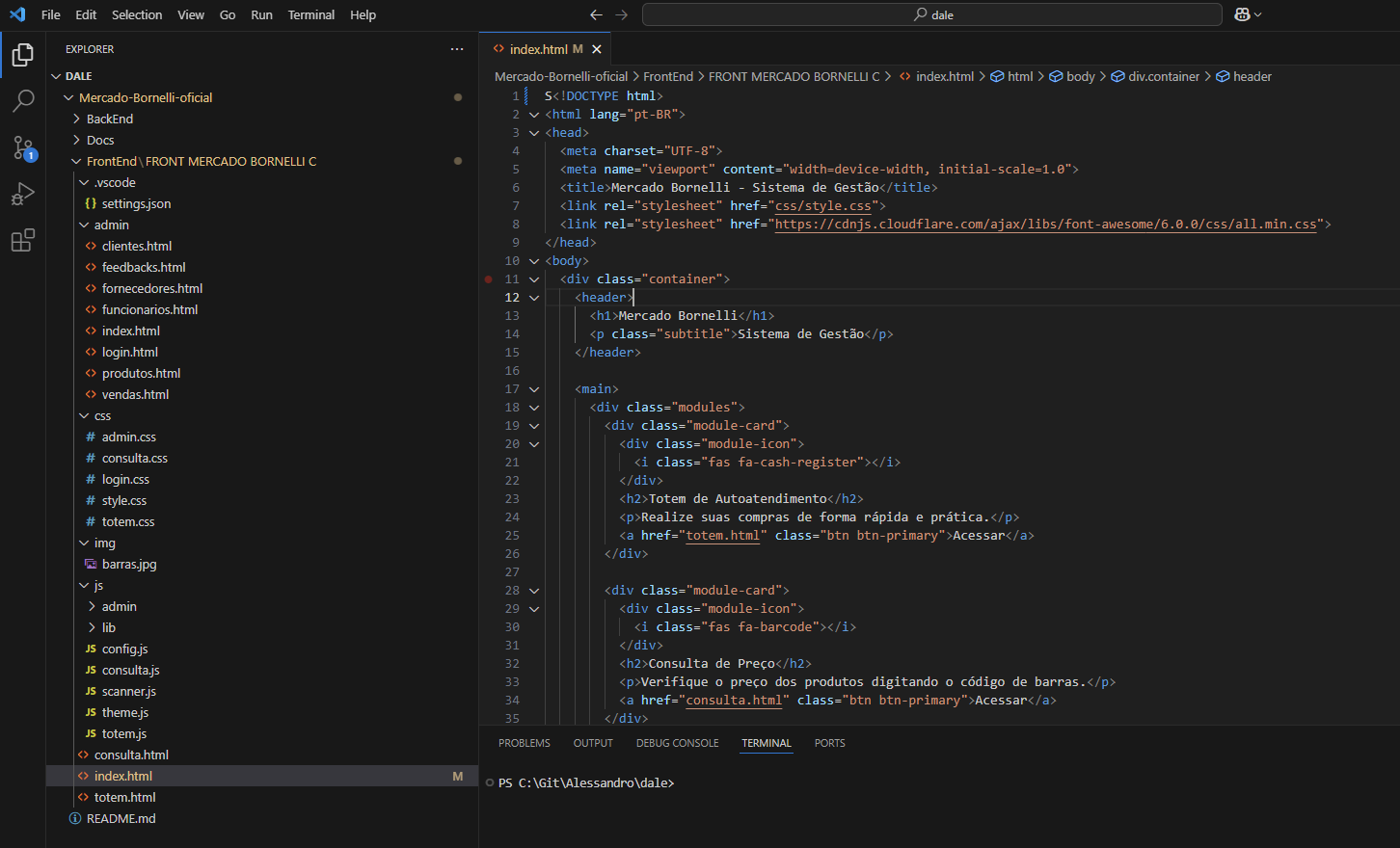
├── service/

│ └── [Implementações dos serviços]

└── MercadoBornelliApplication.java

**3.3 Arquitetura Frontend**

FIGURA 3 - Estrutura de Pastas do Frontend



**Estrutura de Diretórios:**

frontend/

├── admin/

│ ├── index.html (Dashboard)

│ ├── login.html

│ ├── produtos.html

│ ├── clientes.html

│ ├── fornecedores.html

│ ├── funcionarios.html

│ ├── vendas.html

│ └── feedbacks.html

├── consulta/

│ └── index.html

├── totem/

│ └── index.html

├── css/

│ ├── admin.css

│ ├── consulta.css

│ └── totem.css

└── js/

├── config.js

├── admin/

│ ├── auth.js

│ ├── dashboard.js

│ ├── produtos.js

│ ├── clientes.js

│ ├── fornecedores.js

│ ├── funcionarios.js

│ ├── vendas.js

│ ├── feedbacks.js

│ └── theme.js

├── consulta.js

└── totem.js

**3.4 Integração e Comunicação**

FIGURA 4 – Tela de Login do Sistema

Tela de celular

O conteúdo gerado por IA pode estar incorreto.

A comunicação entre frontend e backend ocorre via API REST com:

* **Formato**: JSON
* **Autenticação**: Bearer Token (JWT)
* **CORS**: Habilitado para permitir requisições cross-origin
* **Métodos HTTP**: GET, POST, PUT, PATCH, DELETE

**4. MÓDULO DE ADMINISTRAÇÃO**

**4.1 Dashboard**

FIGURA 5 - Tela do Dashboard Administrativo

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

O Dashboard apresenta:

* **Indicadores principais**: Total de vendas, clientes cadastrados, produtos em estoque
* **Gráficos**: Vendas por período, produtos mais vendidos
* **Alertas**: Produtos com estoque baixo, feedbacks pendentes
* **Acesso rápido**: Links para principais funcionalidades

**Funcionalidades Técnicas:**

* Atualização em tempo real dos dados
* Gráficos interativos com Chart.js
* Filtros por período
* Export de dados para relatórios

**4.2 Gestão de Produtos**

FIGURA 6 - Tela de Gestão de Produtos

Tela de computador com jogo

O conteúdo gerado por IA pode estar incorreto.

**Funcionalidades:**

* Cadastro completo de produtos
* Controle de estoque
* Categorização
* Vinculação com fornecedores
* Histórico de movimentações

**Recursos Especiais:**

* Alerta de estoque baixo
* Atualização em lote
* Importação/exportação de dados
* Código de barras único

**4.3 Gestão de Clientes**

FIGURA 7 - Tela de Gestão de Clientes

Tela de computador com jogo

O conteúdo gerado por IA pode estar incorreto.

**Funcionalidades:**

* Cadastro com validação de CPF
* Histórico de compras
* Rastreamento de acessos
* Soft delete (exclusão lógica)

**Diferenciais:**

* Busca por CPF ou nome
* Registro automático de acessos
* Relatório de clientes por período

**4.4 Gestão de Fornecedores**

FIGURA 8 - Tela de Gestão de Fornecedores

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

**Funcionalidades:**

* Cadastro com validação de CNPJ
* Lista de produtos fornecidos
* Dados de contato completos
* Histórico de fornecimentos

**4.5 Gestão de Funcionários**

FIGURA 9 - Tela de Gestão de Funcionários

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

**Funcionalidades:**

* Cadastro completo com dados trabalhistas
* Controle de status (Ativo, Férias, Licença)
* Definição de cargos
* Controle de salários

**Cargos Disponíveis:**

* Gerente
* Atendente
* Caixa
* Estoquista
* Administrador

**4.6 Controle de Vendas**

FIGURA 10 - Tela de Controle de Vendas

Tela de computador com jogo

O conteúdo gerado por IA pode estar incorreto.

**Funcionalidades:**

* Registro detalhado de vendas
* Múltiplas formas de pagamento
* Vinculação com cliente (opcional)
* Cancelamento de vendas

**Relatórios:**

* Vendas por período
* Vendas por cliente
* Produtos mais vendidos
* Faturamento mensal/anual

**4.7 Gestão de Feedbacks**

FIGURA 11 - Tela de Gestão de Feedbacks

Tela de computador

O conteúdo gerado por IA pode estar incorreto.

**Funcionalidades:**

* Visualização de feedbacks dos clientes
* Classificação por tipo (Elogio, Sugestão, Reclamação)
* Análise de satisfação (notas 1-5)
* Relatórios de satisfação

**5. MÓDULO DE CONSULTA DE PREÇOS**

FIGURA 12 - Interface de Consulta de Preços

Tela de celular

O conteúdo gerado por IA pode estar incorreto.

**5.1 Interface de Consulta**

**Características:**

* Interface simplificada e intuitiva
* Foco no campo de código de barras
* Exibição clara do preço
* Informações adicionais do produto

**5.2 Fluxo de Consulta**

1. Cliente posiciona produto no leitor
2. Sistema captura código de barras
3. Consulta instantânea no banco
4. Exibição de preço e informações
5. Reset automático após 5 segundos

**6. MÓDULO TOTEM DE AUTOATENDIMENTO**

FIGURA 13 - Interface do Totem de Autoatendimento

Tela de celular

O conteúdo gerado por IA pode estar incorreto.

**6.1 Interface do Totem**

**Design:**

* Interface touch-friendly
* Botões grandes e claros
* Feedback visual em todas ações
* Carrinho de compras visível

**6.2 Processo de Venda**

FIGURA 14 - Fluxo de Venda no Totem

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Etapas:**

1. **Seleção de Produtos**: Por categoria ou busca
2. **Carrinho**: Adição, remoção, alteração de quantidades
3. **Identificação**: CPF do cliente (opcional)
4. **Pagamento**: Seleção da forma de pagamento
5. **Finalização**: Confirmação e impressão de comprovante

**6.3 Sistema de Feedback**

FIGURA 15 - Tela de Feedback do Cliente

Tela de celular

O conteúdo gerado por IA pode estar incorreto.

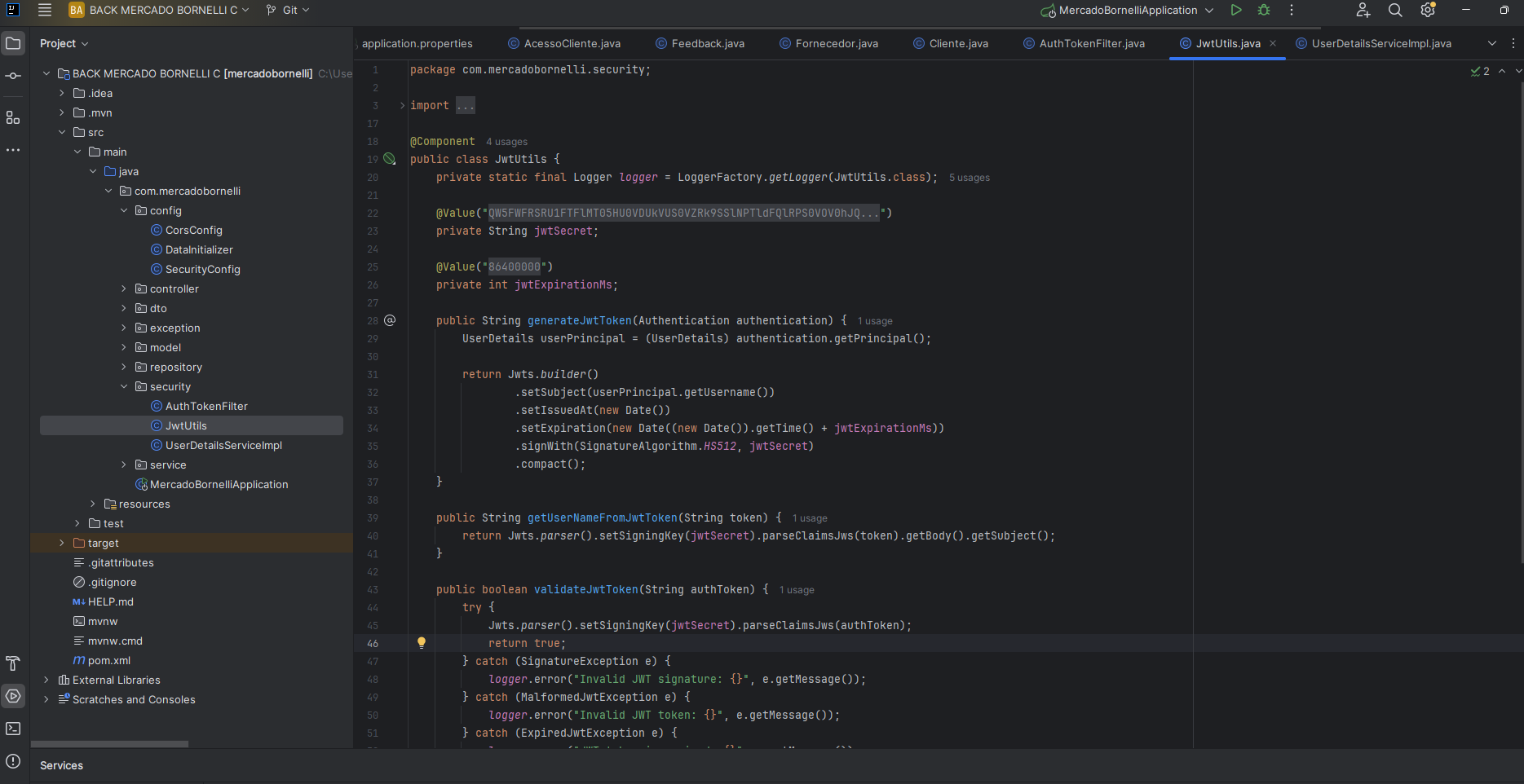
**Coleta de Feedback:**

* Avaliação por estrelas (1-5)
* Tipo de feedback
* Comentário opcional
* Vinculação automática à venda

**7. SEGURANÇA E AUTENTICAÇÃO**

**7.1 Autenticação JWT**

FIGURA 17 - Fluxo de Autenticação JWT



**Processo:**

1. Login com credenciais
2. Geração de token JWT
3. Armazenamento no cliente
4. Envio em todas requisições
5. Validação no servidor

**Configurações:**

* Expiração: 24 horas
* Algoritmo: HS512
* Secret key: Base64 encoded

**7.2 Controle de Acesso**

**Níveis de Acesso:**

* **Público**: Consulta e Totem
* **USER**: Operações básicas
* **ADMIN**: Acesso total

**Proteção de Rotas:**

java

@PreAuthorize("hasRole('ADMIN')")

@PreAuthorize("hasRole('USER') or hasRole('ADMIN')")

**7.3 Perfis de Usuário**

**Gerenciamento:**

* Criação pelo administrador
* Atribuição de perfis
* Ativação/desativação
* Histórico de acessos

**8. BANCO DE DADOS**

**8.1 Modelo de Dados**

FIGURA 16 - Diagrama de Banco de DadosDiagrama

O conteúdo gerado por IA pode estar incorreto.

**Tabelas Principais:**

* tb\_usuario
* tb\_cliente
* tb\_produto
* tb\_fornecedor
* tb\_funcionario
* tb\_venda
* tb\_item\_venda
* tb\_feedback
* tb\_acesso\_cliente

**8.2 Relacionamentos**

**Principais Relacionamentos:**

* Produto N:1 Fornecedor
* Venda N:M Produto (via ItemVenda)
* Venda N:1 Cliente
* Feedback 1:1 Venda
* Cliente 1:N AcessoCliente

**8.3 Estratégias de Persistência**

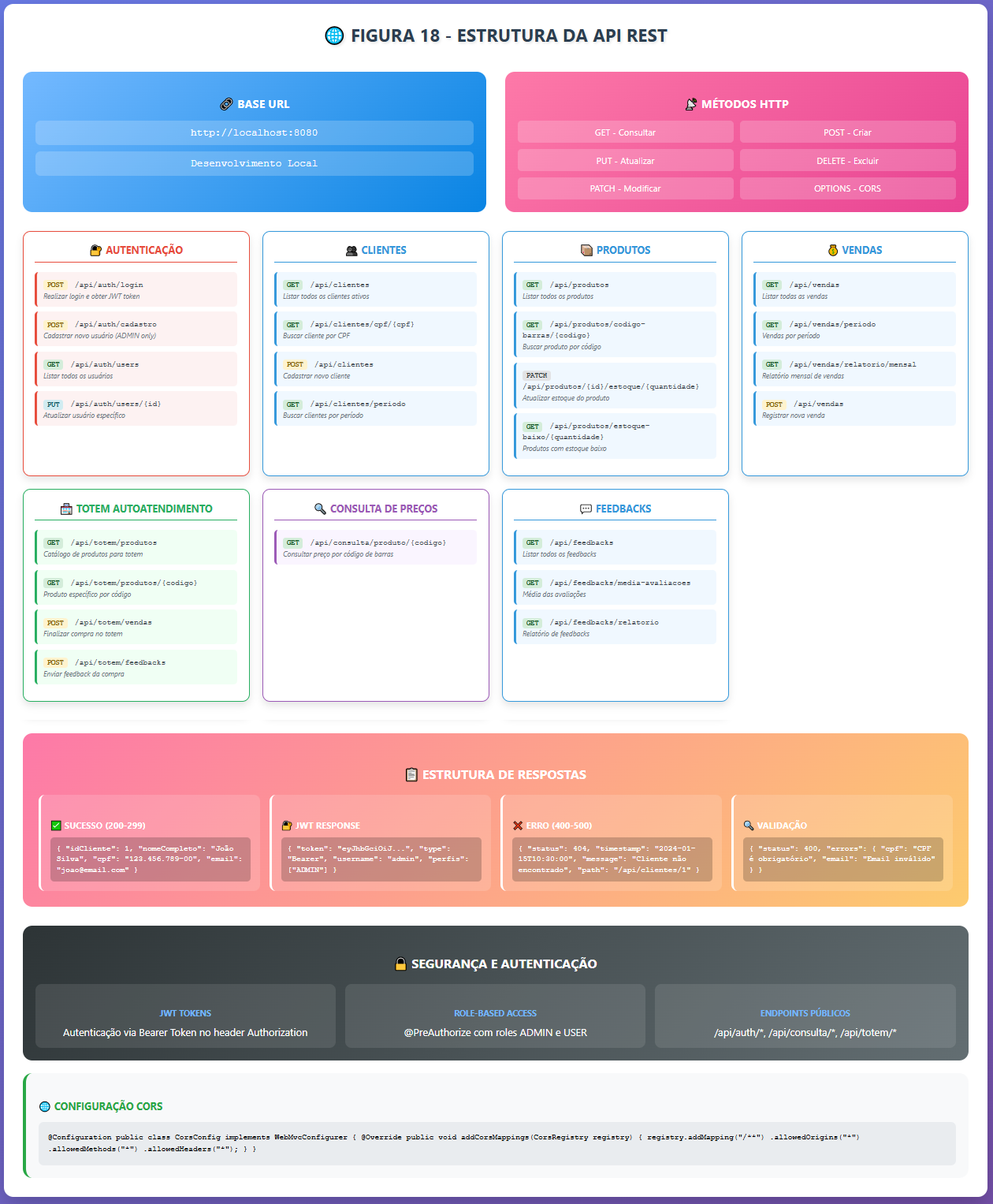
**Técnicas Implementadas:**

* Soft Delete para dados críticos
* Timestamps automáticos
* Validações em nível de banco
* Índices otimizados
* Constraints de integridade

**9. FUNCIONALIDADES TÉCNICAS**

**9.1 API REST**

FIGURA 18 - Estrutura da API REST



**Padrões REST:**

* URLs semânticas
* Métodos HTTP apropriados
* Status codes padronizados
* Respostas em JSON

**Endpoints Principais:**

GET /api/produtos

POST /api/produtos

PUT /api/produtos/{id}

DELETE /api/produtos/{id}

PATCH /api/produtos/{id}/estoque/{quantidade}

**9.2 Tratamento de Exceções**

**GlobalExceptionHandler:**

* ResourceNotFoundException
* ValidationException
* BadCredentialsException
* AccessDeniedException
* Generic Exception

**Resposta Padronizada:**

json

{

"status": 404,

"timestamp": "2024-01-15T10:30:00",

"message": "Recurso não encontrado",

"path": "/api/produtos/999"

}

**9.3 Validações**

**Níveis de Validação:**

1. **Frontend**: Validação básica de formulários
2. **DTO**: Bean Validation annotations
3. **Service**: Regras de negócio
4. **Repository**: Constraints de banco

**Exemplos:**

* CPF/CNPJ válidos
* Email no formato correto
* Campos obrigatórios
* Valores dentro de faixas

**9.4 Paginação e Filtros**

**Implementação:**

* Paginação no frontend
* Filtros por categoria, status, período
* Ordenação customizável
* Busca textual

**10. RELATÓRIOS E ANÁLISES**

**10.1 Relatórios de Vendas**

FIGURA 19 - Relatório de Vendas

Tela de jogo de vídeo game

O conteúdo gerado por IA pode estar incorreto.

**Tipos de Relatórios:**

* Vendas diárias/mensais/anuais
* Produtos mais vendidos
* Faturamento por período
* Análise de formas de pagamento

**Formatos:**

* Visualização em tela
* Gráficos interativos
* Export para PDF/Excel

**10.2 Análise de Feedbacks**

FIGURA 20 - Análise de Feedbacks

Tela de jogo de vídeo game

O conteúdo gerado por IA pode estar incorreto.

**Métricas:**

* Média de satisfação
* Distribuição por tipo
* Evolução temporal
* Comentários mais frequentes

**10.3 Indicadores de Desempenho**

**KPIs Monitorados:**

* Ticket médio
* Produtos com giro baixo
* Taxa de satisfação
* Crescimento de vendas
* Novos clientes

**11. CONSIDERAÇÕES FINAIS**

**11.1 Pontos Fortes**

1. **Arquitetura Robusta**: Separação clara de responsabilidades
2. **Segurança**: Autenticação JWT e controle de acesso
3. **Usabilidade**: Interfaces intuitivas e responsivas
4. **Escalabilidade**: Estrutura preparada para crescimento
5. **Manutenibilidade**: Código organizado e documentado

**11.2 Possíveis Melhorias**

1. **Performance**:
   * Implementar cache Redis
   * Otimizar queries com índices
   * Implementar lazy loading
2. **Funcionalidades**:
   * Integração com sistemas de pagamento
   * App mobile para administração
   * Sistema de notificações em tempo real
   * Integração com contabilidade
3. **Infraestrutura**:
   * Containerização com Docker
   * CI/CD pipeline
   * Monitoramento com ferramentas APM
   * Backup automatizado
4. **UX/UI**:
   * Modo escuro nativo
   * Acessibilidade aprimorada
   * PWA para funcionar offline
   * Personalização de dashboard

**11.3 Conclusão**

O Sistema Mercado Bornelli representa uma solução completa e moderna para gestão de mercados, combinando tecnologias atuais com boas práticas de desenvolvimento. A arquitetura modular permite expansão e manutenção facilitadas, enquanto a interface intuitiva garante adoção rápida pelos usuários.

O sistema atende plenamente aos requisitos estabelecidos, oferecendo:

* Gestão administrativa completa
* Experiência de compra modernizada
* Controles de segurança robustos
* Análises e relatórios detalhados

Com as melhorias sugeridas, o sistema pode evoluir para uma plataforma ainda mais completa, mantendo-se alinhado com as tendências tecnológicas e necessidades do mercado varejista.