Documentação Técnica — API ContaCerta v3

# Documentação Técnica — API ContaCerta v3

\*\*Autor técnico:\*\* Jefferson Paulo Barbosa

\*\*Versão:\*\* 1.0.0

\*\*Data:\*\* 2025-10-26

## 1. Introdução

A API ContaCerta v3 é um backend RESTful para gestão financeira e controle de vendas de pequenos empreendedores. Inclui autenticação JWT, CRUD de clientes/produtos/pedidos, cálculo automático de CMV (com cron, fallback e histórico) e relatórios de CMV e rendimento por produto. O projeto foi estruturado para evoluir para frontends web e mobile, mantendo camadas coesas e baixo acoplamento.

## 2. Arquitetura e camadas

- routes/, controllers/, services/, models/, middlewares/, utils/, config/, recursos/.

- Fluxo: rota → controller → service → (model) → response/erro.

- Auth: JWT (Bearer), bcrypt para senhas, middleware de proteção.

- Erros: middleware global com JSON padronizado.

## 3. Pontos fortes

- Arquitetura modular, legível e extensível.

- JWT + bcrypt, segurança básica pronta.

- Tratamento global de erros (400/401/404/500).

- CMV automático (cron 6h), fallback e histórico.

- Swagger UI em /api/docs.

## 4. Pontos de melhoria

- Migrar de memória para banco (PostgreSQL) + ORM (Prisma/Sequelize).

- Testes automatizados (Jest + Supertest), pipeline CI.

- Logs estruturados (pino/winston), métricas e tracing.

- Validação de payload (Joi/Zod) nas rotas.

- Rate limiting, CORS e Helmet.

## 5. Endpoints (resumo)

- POST /auth/login

- GET/POST/PUT/DELETE /clientes, GET /clientes/{id}

- GET/POST/PUT/DELETE /produtos, GET /produtos/{id}

- GET/POST /pedidos, GET /pedidos/{id}

- GET /relatorios/cmv

- GET /relatorios/rendimento

## 6. Erros (padrão)

{ "status": 400, "mensagem": "Campos obrigatórios ausentes" }

## 7. Relatórios

- CMV: base (valor atual com histórico) + total (somatório de custos vendidos).

- Rendimento: lucro unitário, margem percentual e rendimento total por produto.

## 8. Validação QA

- Integração/contrato/exploratórios.

- Casos: login, CRUDs, criação de pedido, relatórios, falhas de validação.

- Sugestão: Postman + Jest/Supertest.

## 9. Conclusão

Pronta para dev/demonstração; documentação OpenAPI sincronizada com o backend.