

Detalhamento Estrutural Profundo do Projeto CoffeeConilon

backend/app/core/

- config.py: Carrega variáveis de ambiente (DB, chaves, debug, CORS, etc). Exemplo: Classe Settings via Pydantic/BaseSettings. Use em produção para qualquer variável mutável. Importado em main.py, database.py, security.py.
- database.py: Configura engine, sessão do SQLAlchemy, Base. Fornece SessionLocal para CRUD e testes. Usado em models, CRUD, main.py.
- security.py: Funções de hash/verificação de senha, geração/validação de JWT, autenticação OAuth2. Usado em endpoints de auth, models/user, schemas/token.

backend/app/models/

- base.py: Base declarative do SQLAlchemy (Base = declarative_base()). Herdada por todos os modelos.
- user.py: Tabela de usuários (id, nome, email, senha_hash, etc). Relações para fazenda, registros, permissões.
- farm.py, crop.py, lot.py: Tabelas de fazenda, safra/cultivo, lote/talhão. Exemplo: Farm(nome, localizacao, dono_id), Lot(nome, area, farm_id). Relacione entidades.
- financial.py: Tabela de lançamentos financeiros (tipo, valor, data, descrição, lote_id, user_id, categoria).
- stock.py: Tabela de estoque (tipo, produto, quantidade, data, lote_id, entrada/saida/perda).

backend/app/schemas/

- user.py, farm.py, crop.py, lot.py, financial.py, stock.py, report.py, token.py: Schemas Pydantic para validação de entrada/saída da API. Ex: UserCreate, UserRead, FinancialCreate, etc. Nunca exponha models diretamente na API.

backend/app/crud/

- crud_user.py, crud_farm.py, ...: Funções CRUD para cada entidade. Só operações simples de banco (Create, Read, Update, Delete). Não implemente lógica de negócio aqui.

backend/app/services/

- reporting.py: Funções e regras para geração de relatórios complexos, agregações.
- financial_calcs.py: Funções de cálculo de custos, receitas, rentabilidade.
- stock_ops.py: Funções de movimentação e alertas de estoque.

backend/app/api/v1/endpoints/

- auth.py, users.py, farms.py, ...: Endpoints FastAPI de cada domínio. Rotas REST chamam funções de service/crud, validam entrada/saída com schemas.

backend/app/main.py

- main.py: Ponto de entrada da API FastAPI. Inicializa app, inclui routers, configura CORS e middlewares.

backend/tests/

- test_auth.py, test_financial.py, ...: Testes unitários e de integração. Use pytest e SQLite in-memory para execução rápida. Teste endpoints, regras de negócio e integrações.

frontend/public/

- index.html: HTML base do app React. Contém <div id='root'>. Não precisa alterar muito.
- favicon.ico: Ícone da aplicação no navegador.

frontend/src/assets/

- .gitkeep / arquivos: Imagens, logos e arquivos estáticos da aplicação.

frontend/src/components/

- Dashboard/, Financial/, Inventory/, ...: Componentes de UI reutilizáveis, separados por domínio (cards, tabelas, formulários, gráficos).

frontend/src/contexts/

- .gitkeep / arquivos: React Context Providers para autenticação, temas, etc.

frontend/src/hooks/

- .gitkeep / arquivos: Hooks personalizados do React (useAuth, useApi, etc).

frontend/src/pages/

- Home.tsx, Dashboard.tsx, ...: Páginas de alto nível. Cada página corresponde a uma rota principal do app.

frontend/src/services/

- api.ts, auth.ts: Comunicação com a API backend. Centralize chamadas HTTP/axios/fetch aqui.

frontend/src/styles/

- theme.ts, globals.css: Definição de temas e estilos globais (cores, fontes, layouts padrão).

frontend/src/App.tsx & main.tsx

- App.tsx: Componente raiz, define rotas, providers.
- main.tsx: Ponto de entrada da aplicação, renderiza App.

frontend/package.json

- package.json: Gerenciamento de dependências e scripts do React.

Arquivos Raiz

- .env: Variáveis globais compartilhadas por backend e frontend.
- docker-compose.yml: Orquestração de serviços (backend, frontend, banco de dados).
- README.md: Documentação geral: setup, scripts, instruções de uso e deploy.

Recomendações Finais

- Implemente, teste e documente cada módulo separadamente.
- Use versionamento Git desde o início.
- Atualize este documento conforme o projeto evoluir.
- Sempre mantenha a arquitetura modular e os domínios bem separados.

Este PDF serve como guia de referência rápida durante o desenvolvimento.