

ARCHITECTURE - FINANCEFLOW

Visão geral

Finance Flow é uma aplicação de gestão financeira pessoal baseada em arquitetura desacoplada:

- Backend: Django + Django REST Framework
- Frontend: React
- Banco de dados: PostgreSQL
- Autenticação: JWT
- Deploy: Docker + VPS

O frontend consome exclusivamente a API REST.

Arquitetura geral

React -> API Gateway -> Django REST -> PostgreSQL

O frontend nunca acessa o banco de dados diretamente

Separação de camadas

Backend

- Models: regras de domínio (User, Profile, Account, Transfer, Transaction, Category, Goal, Budget, CreditCard, Invoice)
- Serializers: Validação e transformação dos dados
- Views: Endpoints HTTP
- Services: Regras de negócio (Ex fechar fatura)
- Permissions: controle de acesso

Frontend

- Pages
- Components
- Services (API)
- Hooks
- Store (estado global)

Autenticação

JWT:

- Access Token: 15 minutos
- Refresh Token: 7 dias

Fluxo:

- Login -> Recebe tokens -> Frontend armazena -> Chamadas autenticadas

Fluxo financeiro:

- Transaction registra entrada e saída
- Transfer registra movimentações entre as contas do próprio usuário:
- Transfer registra duas movimentações financeiras:
 - Transferência de saída da conta de origem
 - Transferência de recebimento na conta de destino
- Invoice agrupa transactions de cartão por mês
- Budget valida limites que podem ser gastos por mês
- Goal acompanha progresso nas metas

Padrões adotados

- UUID como primary key
- Decimal para valores monetários
- Soft delete (armazenar os itens excluídos por 1 semana para que o usuário possa recuperá-los caso queira)
- TimeStamped models (Hora da criação e hora da edição)
- Services para lógicas complexas

Convenções

- snake_case no Backend
- CamelCase no Frontend
- Pastas organizadas por domínio

Decisões técnicas

React foi escolhido por permitir:

- SPA
- App Mobile futuro
- Desacoplamento completo do backend

PostgreSQL por confiabilidade em dados financeiros

Futuras evoluções

- App Mobile
- Webhooks
- Integrações bancárias