

# Especificação de Projeto

---

## 1) Visão Geral

**Objetivo:** gerenciar clientes e contas bancárias e permitir operações financeiras básicas com rastreabilidade.

**Stack:** Angular 20 (standalone components, routing moderno), TypeScript.

**Arquitetura:** SPA modularizada por domínios, lazy loading, Services + State, Interceptors (auth, error, loading).

## 2) Escopo Funcional

**Home:** cards com KPIs (clientes, contas, saldos), atalhos e feed de operações.

**Módulo de Clientes (CRUD):** listar, detalhar, criar/editar, excluir clientes (campos: id, nome, cpf, email, observacoes, ativo).

**Módulo de Contas (CRUD):** listar, detalhar, criar/editar, excluir contas (campos: id, numero, agencia, saldo, cliente, nomeCliente).

**Operações (Saque, Depósito, Transferência):** regras de negócio, limites, comprovantes e histórico. Validações de saldo, conta ativa, valores > 0.

## 3) UX e Telas

**Home:** métricas e lista de operações.

**Clientes:** lista detalhe formulário.

**Contas:** lista detalhe (saldo e histórico) formulário.

**Operações:** telas para saque, depósito e transferência com modal de confirmação.

**Navbar:** logo, links, avatar.

**Footer:** versão do app, links úteis, status.

## 4) Navegação & Rotas

/ (Home)

/clientes (lista, novo, detalhe, editar)

/contas (lista, novo, detalhe, editar)

/operacoes (saque, deposito, transferencia)

## 5) Modelos de Dados

Interfaces TypeScript para Cliente, Conta e Operação, com campos de identificação, dados pessoais/financeiros e histórico.

## 6) Contratos de API

Clientes: GET/POST/PUT/DELETE.

Contas: GET/POST/PUT/DELETE.

Operações: POST saque/deposito/transferencia.

Erros padronizados com código, mensagem e detalhes.

## 7) Arquitetura Angular

Standalone Components + Feature Modules.

Lazy routes.

Services por domínio.

NgRx para estado global e Signals para estado local.

Interceptors: Auth, Error, Loading.

Observabilidade e Acessibilidade.

## 8) Estrutura de Pastas

src/app/core (interceptors, guards, services, utils, layout)

src/app/features (home, clientes, contas, operacoes)

src/app/shared (components, directives, pipes)

environments

## 9) Casos de Teste

**Clientes:** validações de CPF/email, exclusão bloqueada com conta ativa.

**Contas:** criação com cliente inexistente, encerramento bloqueado se saldo > 0.

**Operações:** saque sem saldo, depósito negativo, transferência inválida, idempotência.