

# ENGENHARIA DE SOFTWARE E ARQUITETURA DE SISTEMAS

## ENTREGA 2

**Projeto:** Dashboard Interativo — PicMoney

**Equipe:** Antonio Petri, Daniel Moribe, Gabriel Paschoal, Vitor Kenzo

**Professora:** Lucy Mari

São Paulo — 2025

## 1) Design de Software — Visão Geral

Durante o desenvolvimento, a aplicação de princípios de design foi essencial para garantir que o sistema fosse robusto, escalável e de fácil manutenção. Esta versão consolida o conteúdo da entrega-base e o adapta ao contexto do projeto PicMoney (dashboard executivo com frontend React e backend Flask), mantendo a estrutura teórica e os exemplos citados e adicionando diagramas UML em alta resolução.

## 2) Princípios SOLID (com exemplos do projeto)

### **Single Responsibility Principle (SRP)**

Cada módulo possui apenas uma responsabilidade. Ex.: **PadronizacaoCSV.py** padroniza planilhas; **Metricas.py** calcula indicadores financeiros.

### **Open/Closed Principle (OCP)**

Classes abertas para extensão e fechadas para modificação. Ex.: **Detalhes.py** define o layout base, permitindo adicionar novos tipos de relatórios sem alterar código existente.

### **Liskov Substitution Principle (LSP)**

Novos módulos substituem antigos sem quebrar contratos. Ex.: novas funções em **Detalhes.py** podem substituir implementações sem alterar o funcionamento do dashboard.

### **Interface Segregation Principle (ISP)**

Interfaces especializadas. Ex.: **formatacao.py** fornece formatação; **MapaCalor.py** usa essa interface para renderizar gráficos sem dependências desnecessárias.

### **Dependency Inversion Principle (DIP)**

Módulos de alto nível dependem de abstrações. Ex.: **Relatorios.py** usa interfaces de **PadronizacaoCSV.py** e **Metricas.py**, permitindo exportar em PDF ou Excel sem alterações internas.

## 3) Modelos de Design e Arquitetura

O projeto segue uma arquitetura em três camadas: Apresentação (React/Chart.js), Negócio (Flask/Services/DTOs) e Dados (SQLite/CSV). Essa separação garante modularidade e fácil manutenção. O padrão MVC também é aplicado no frontend, com componentes React (Views) e controllers Flask (Controllers).

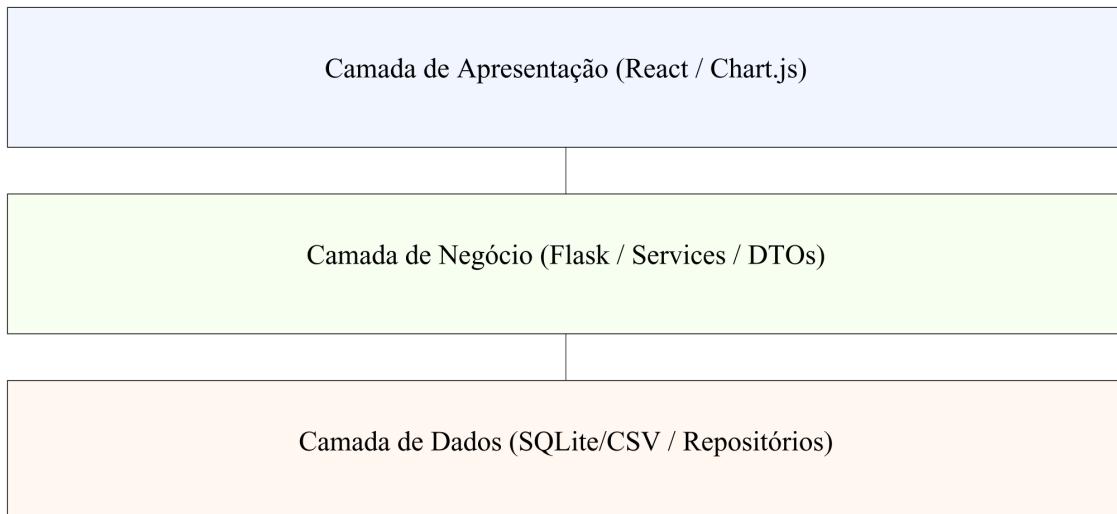


Figura 1 — Arquitetura em Camadas.

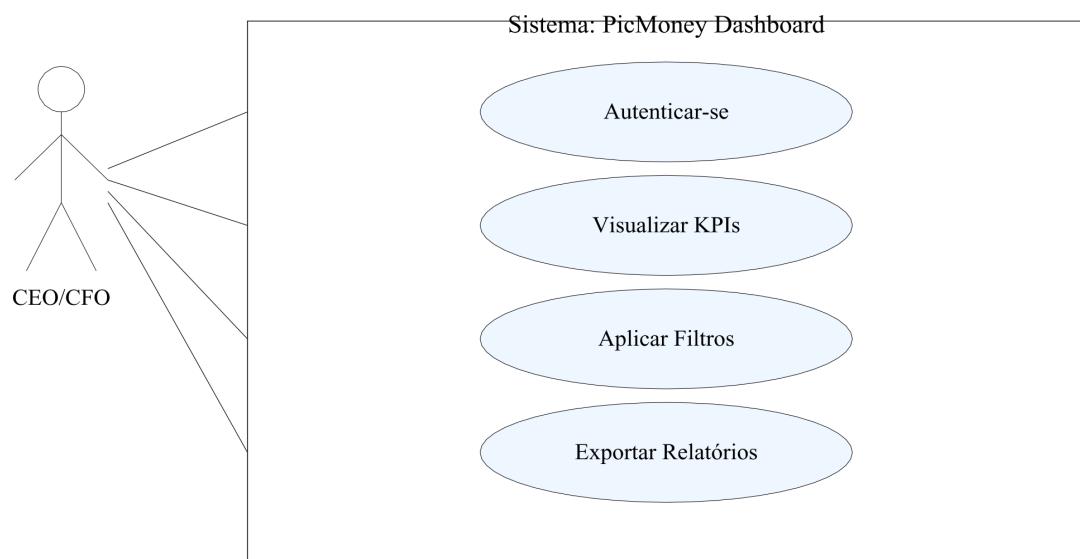


Figura 2 — Diagrama de Casos de Uso.

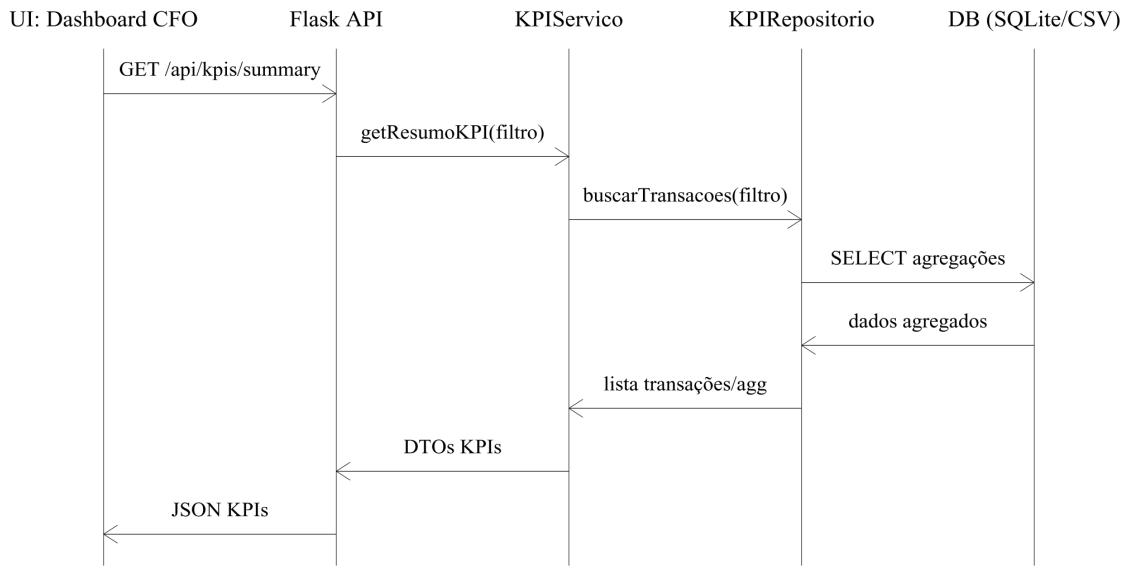


Figura 3 — Diagrama de Sequência.

#### 4) Conclusão

A aplicação dos princípios SOLID e da arquitetura em camadas garantiu que o sistema PicMoney permanecesse modular, testável e escalável. Os diagramas representam a arquitetura real do projeto, os fluxos de interação e os casos de uso práticos, consolidando o design técnico da aplicação.