



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SÃO PAULO**

CAMPUS GUARULHOS

BEATRIZ MAZZUCATTO - GU3046745

VCRIQUINHO

**GUARULHOS - SP
2025**

BEATRIZ MAZZUCATTO - GU3046745

VCRIQUINHO

Atividade apresentada à disciplina de Ciência, Tecnologia e Sociedade, do Instituto Federal de Educação, Ciência e Tecnologia Campus Guarulhos como nota parcial para aprovação na disciplina do curso Engenharia de Computação, sexto semestre.

Professor Mestre: Giovani
Fonseca Ravagnani Disperati

GUARULHOS - SP
2025

1. Objetivo

Este documento descreve e justifica as decisões de design adotadas no desenvolvimento do sistema VcRiquinho, considerando princípios de engenharia de software, padrões de projeto e requisitos de extensibilidade.

2. Arquitetura e Padrões Utilizados

2.1 Padrão Strategy com Herança

2.1.1 Hierarquia de Classes Cliente

Estrutura:

- Cliente (classe abstrata)
 - PessoaFisica
 - PessoaJuridica

Justificativa:

A hierarquia permite o tratamento polimórfico de diferentes tipos de clientes, preservando um núcleo comum e mantendo código reutilizável e extensível. As particularidades (CPF para pessoas físicas e CNPJ para jurídicas) são encapsuladas em subclasses específicas.

2.1.2 Hierarquia de Classes Conta

Estrutura:

- Conta (classe abstrata)
 - ContaCorrente
 - ContaCDI
 - ContaInvestAuto

Justificativa:

Cada tipo de conta possui lógica própria para cálculo de rendimento e taxas. A

separação em subclasses respeita o Princípio Aberto/Fechado (Open/Closed Principle), permitindo a adição de novos tipos de conta sem alterar código existente.

2.1.3 Hierarquia de Produtos de Investimento

Estrutura:

- ProdutoInvestimento (classe abstrata)
 - RendaFixa
 - RendaVariavel

Justificativa:

Produtos distintos apresentam características diferentes (como prazos de carência e tipos de rendimento). A estrutura atual facilita a criação de novos produtos de investimento sem impacto no restante do sistema.

2.2 Padrão Facade/Manager

2.2.1 VeRiquinhoManager

Justificativa:

Centraliza operações **CRUD** e regras de negócio, oferecendo uma interface simplificada para execução de tarefas complexas. Atua como ponto único de acesso, reduzindo acoplamento e facilitando manutenção e controle.

2.3 Composição vs Herança

2.3.1 Conta possui Cliente

Justificativa:

A relação “tem-um” reflete corretamente o vínculo entre conta e cliente, evitando herança múltipla e mantendo a clareza no modelo.

2.3.2 ContaInvestAuto contém ProdutoInvestimento

Justificativa:

Permite vincular múltiplos produtos de investimento a uma conta, possibilitando diversificação de portfólio e cálculo de rendimento agregado.

2.4 Encapsulamento e Responsabilidades

2.4.1 Separação de Responsabilidades

- **Cliente:** Armazena dados pessoais e documentos.
- **Conta:** Controla saldo e cálculos financeiros.
- **ProdutoInvestimento:** Define características do investimento.
- **VeRiquinhoManager:** Gerencia regras de negócio e fluxo de operações.

Justificativa:

O isolamento de responsabilidades segue o *Single Responsibility Principle*, facilitando manutenção, testes e evolução.

3. Extensibilidade

A arquitetura permite:

- Novos tipos de clientes (ex.: ClienteEmpresarial)
- Novos tipos de conta (ex.: ContaPoupanca)
- Novos produtos de investimento (ex.: Criptomoedas, Tesouro Direto)
- Novas regras de cálculo sem impacto no código existente

4. Uso de Polimorfismo

Vantagens:

- Tratamento uniforme de diferentes tipos de objetos.
- Código mais limpo e de fácil manutenção.

- Redução de cadeias condicionais extensas (if/else).

5. Conclusão

O design do sistema **VcRiquinho** segue princípios **SOLID**, utiliza padrões de projeto reconhecidos e foi pensado para garantir **reutilização**, **extensibilidade** e **manutenção simplificada**.

Essa abordagem assegura que o sistema possa evoluir de forma sustentável, acompanhando mudanças e novas demandas do negócio.