

# Design Orientado a Objetos: Sistema de Pagamentos

**Aluno:** Alan Lino dos Reis

**Curso:** Ciência da Computação 4 UTFPR Câmpus Medianeira

**Professor:** Everton Coimbra

# Arquitetura Base: O Ritual de Processamento

O sistema é estruturado em torno de uma **classe base concreta** **Pagamento** que orquestra um ritual fixo e previsível de processamento.

## O Fluxo Consistente

Todas as formas de pagamento seguem rigorosamente três etapas sequenciais:



### Validar

Verificação dos dados de entrada



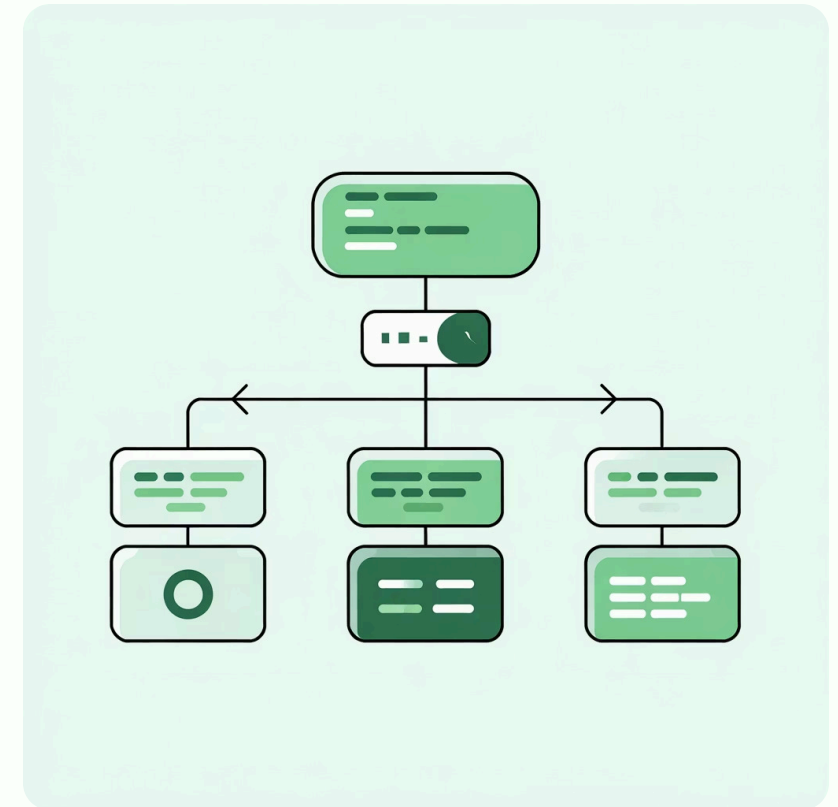
### Autorizar/Capturar

Processamento específico do tipo



### Confirmar

Geração do comprovante final



A consistência do fluxo é garantida pela classe base, que define pontos de variação controlada através de **métodos protegidos virtuais** (ganchos).



# Especialização por Herança: As Três Variações

As subclasses concretas representam variações legítimas do mesmo processo, refinando apenas o necessário mantendo o ritual intacto.



## PagamentoCartao

Realiza autorização e captura junto à operadora financeira



## PagamentoPix

Gera e valida QR Code para transferência instantânea



## PagamentoBoleto

Emite linha digitável e confirma compensação bancária

- ☐ Todas as classes são **seladas (sealed)** e não adicionam novos métodos públicos, garantindo uso seguro da herança por especialização.

# Princípio da Substituição de Liskov (LSP)

O design segue três regras fundamentais para garantir que subclasses substituam a classe base sem quebrar o comportamento do sistema:

01

## Substituibilidade Total

Qualquer componente que utilize **Pagamento** funciona com **PagamentoCartao**, **PagamentoPix** ou **PagamentoBoleto** sem verificações de tipo

02

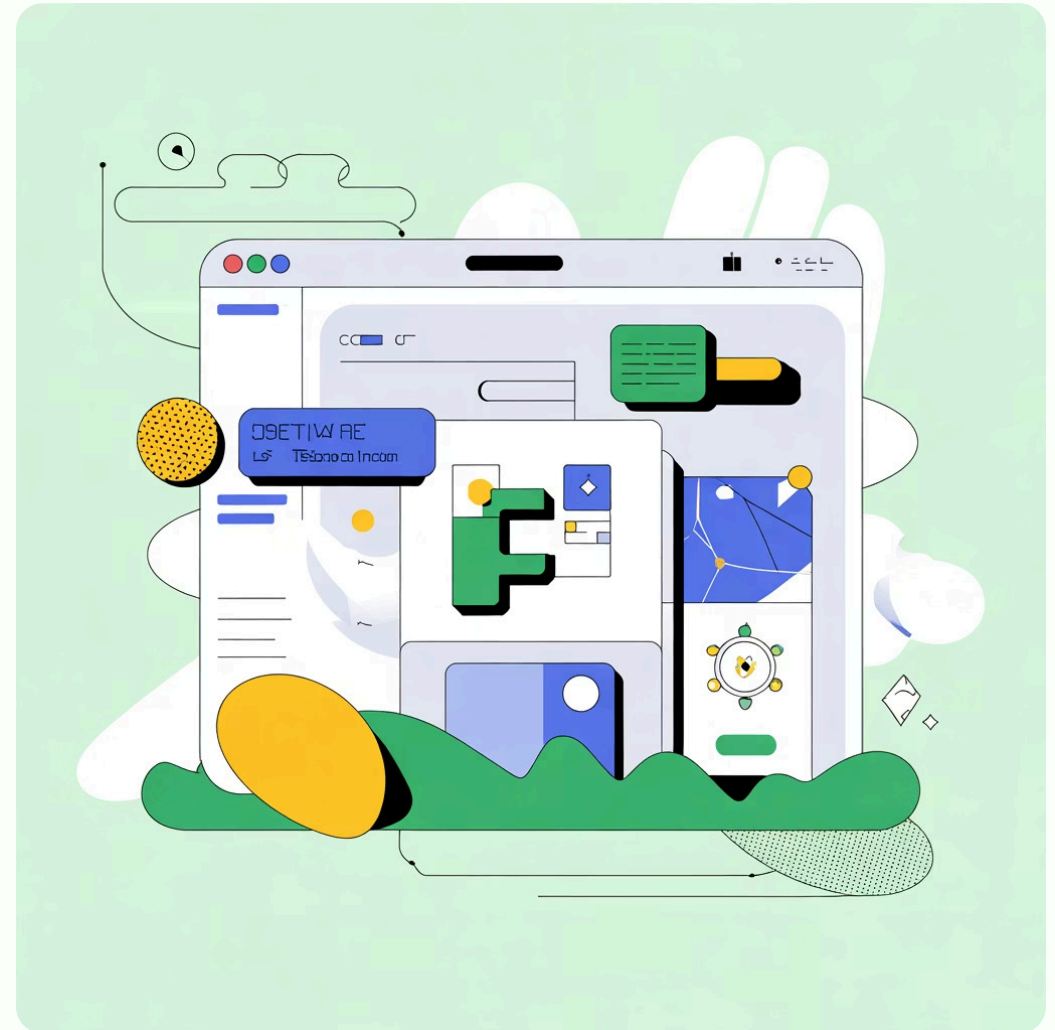
## Invariantes Preservados

Validações mínimas da classe base são mantidas ou fortalecidas, nunca enfraquecidas

03

## Contratos de Saída Equivalentes

Resultado final mantém coerência, entregando comprovante válido sem exceções inesperadas



"O LSP assegura que a herança é usada de forma segura e previsível, garantindo estabilidade em todas as formas de pagamento."

# Composição: Eixos Plugáveis e Intercambiáveis

Além da herança, o modelo incorpora **composição por delegates** para tratar políticas independentes e combináveis, preservando baixo acoplamento.

## Antifraude

Decide se o pagamento é aprovado ou rejeitado antes da autorização

- Função injetável: recebe valor, retorna decisão booleana
- Estratégias intercambiáveis sem modificar classes

## Câmbio

Converte valores quando há pagamento em moeda estrangeira

- Função injetável: recebe valor, devolve valor convertido
- Políticas de conversão plugáveis e independentes

Esses eixos permitem variar comportamentos **sem alterar a estrutura das classes**, reforçando o uso adequado da composição para políticas de negócio.





# Síntese: Design Coeso e Extensível

O modelo final combina o melhor de dois mundos, criando um sistema robusto e preparado para evolução:

## Herança Controlada

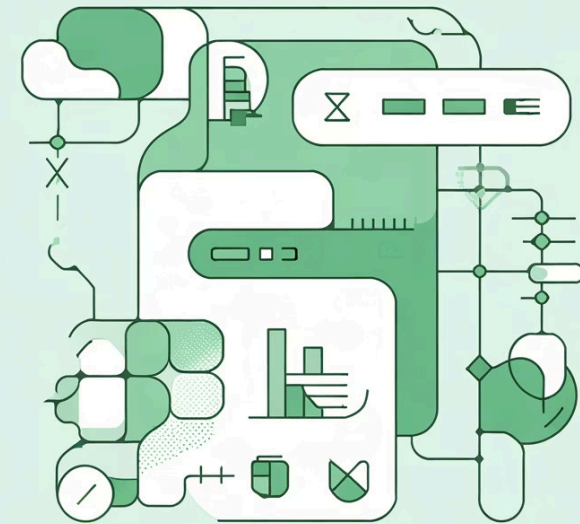
Ritual fixo e consistente entre tipos de pagamento

- Fluxo previsível e estável
- Especialização segura por ganchos
- Aderência total ao LSP

## Composição Flexível

Políticas de negócio independentes e intercambiáveis

- Delegates plugáveis
- Baixo acoplamento
- Alta extensibilidade



**Resultado:** Sistema que assegura estabilidade, reduz acoplamento e permite futuras extensões sem comprometer a arquitetura fundamental.