

Sistema de Pedidos de Venda

Aluno: Alan Lino dos Reis

Curso: Ciéncia da Computaçao — UTFPR Câmpus Medianeira

Professor: Everton Coimbra



O Desafio: Um Ritual com Variações

Fluxo Comum

Todo pedido segue um processo fixo e previsível:

01

Validar Dados

Verificar integridade do pedido

02

Calcular Total

Aplicar regras de negócio

03

Emitir Recibo

Gerar documento fiscal

Variações por Tipo

Pedido Nacional

- Impostos internos (ICMS/IPI)
- Geração de NF-e

Pedido Internacional

- Taxas de importação
- Conversão cambial
- Commercial Invoice

O desafio é manter o fluxo fixo enquanto permitimos variações controladas em cada etapa, além de políticas flexíveis para frete e promoções.

Decisão de Design: Template Method



Herança Controlada

Classe base concreta Pedido define o ritual fixo através do método Processar(). Ganchos protected (virtual/abstract) permitem variação: Validar(), CalcularSubtotal(), EmitirRecibo().



LSP Garantido

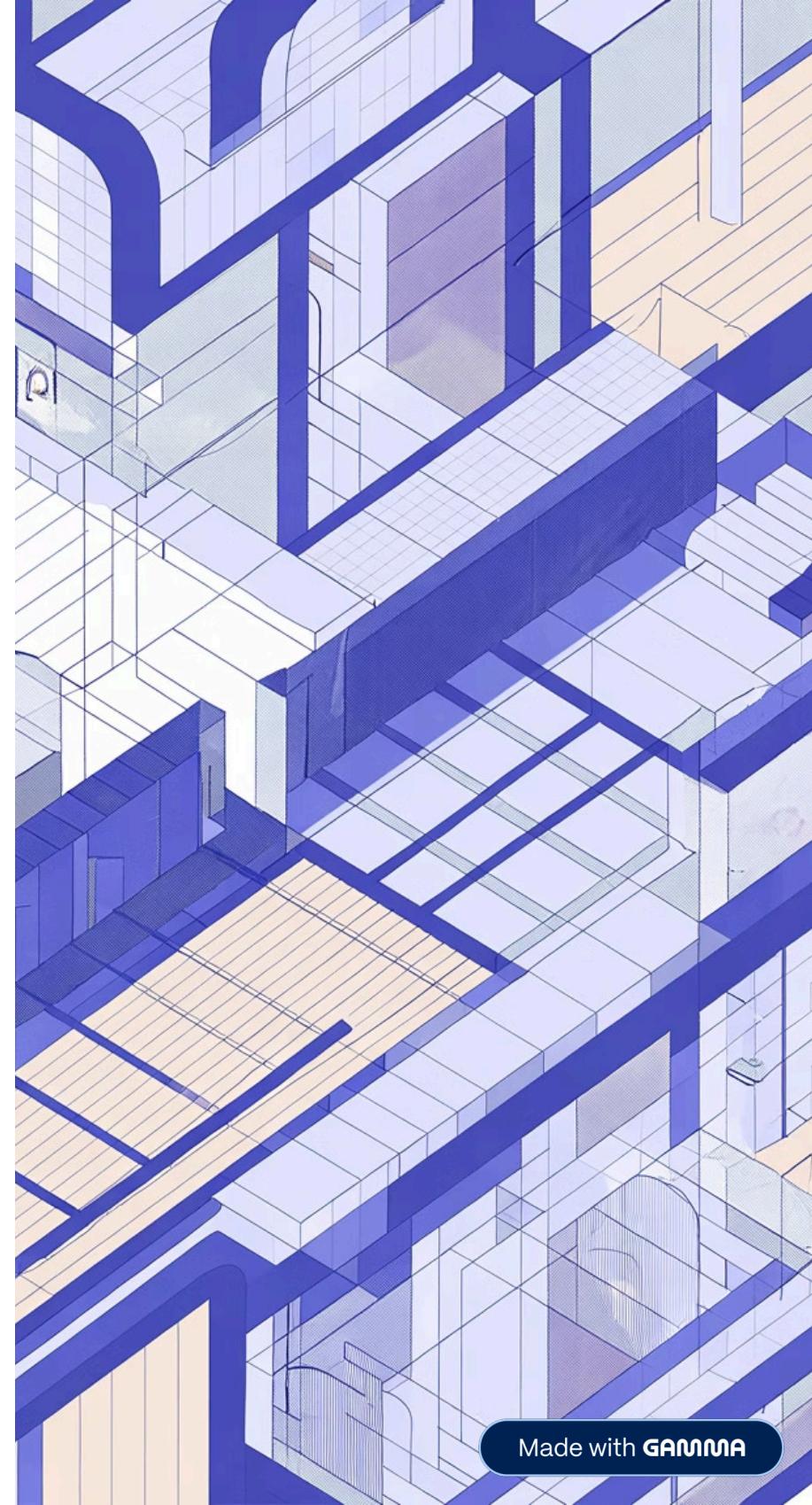
Princípio da Substituição de Liskov aplicado: cliente trabalha com Pedido sem conhecer implementações concretas.

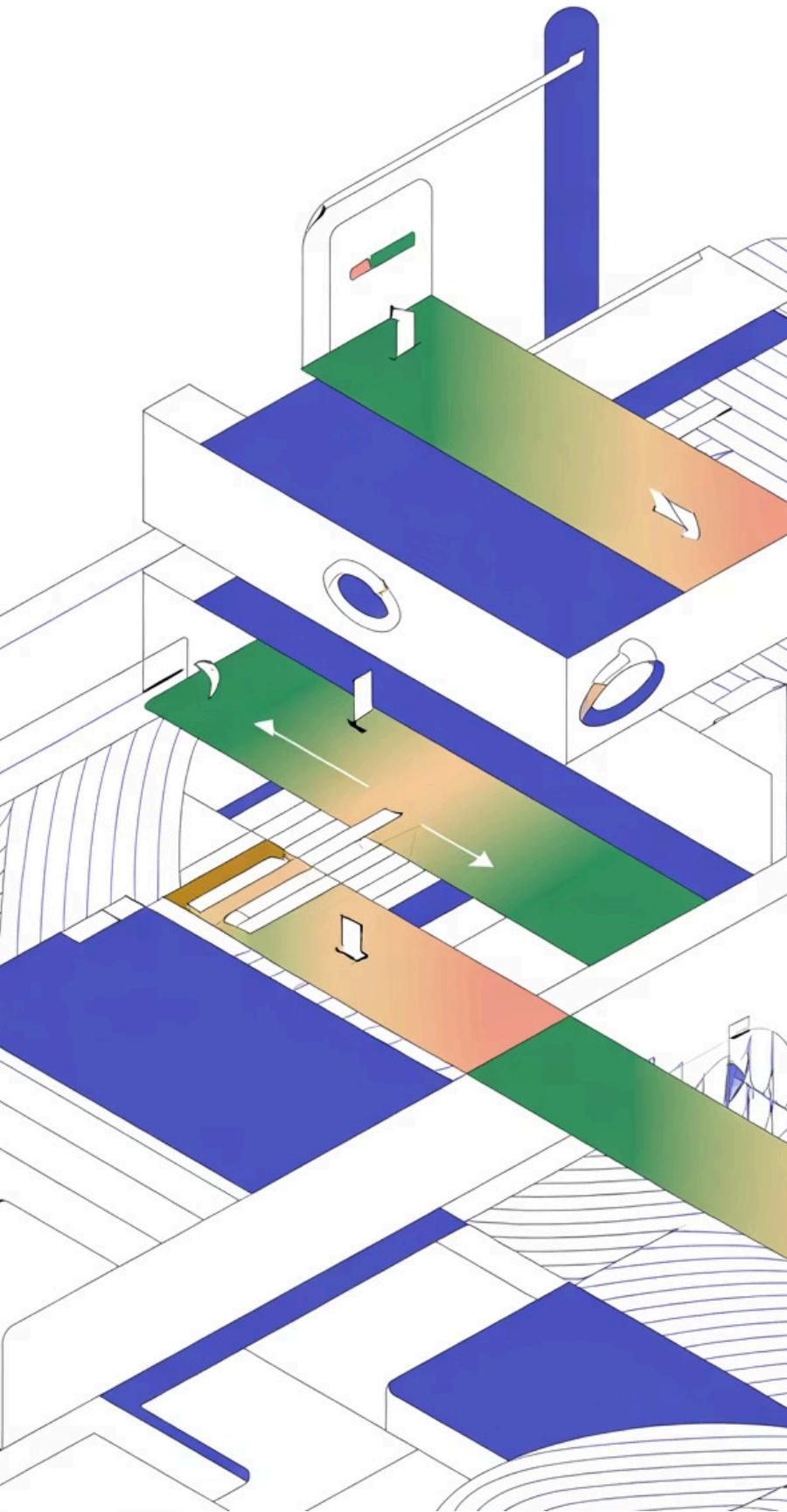
Exemplo: Pedido p = new PedidoNacional(...);
p.Processar();



Composição via Delegates

Estratégias plugáveis de Frete e Promoção usando Func<decimal, decimal>. Permite trocar regras de negócio sem criar novas subclasses, mantendo flexibilidade máxima.





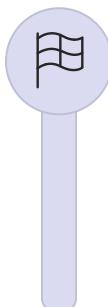
Arquitetura e Implementação

Hierarquia de Classes



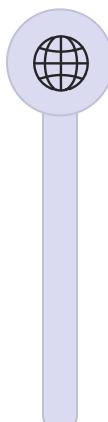
Pedido (Base)

Classe concreta
com Processar()



PedidoNaci onal

Sealed,
impostos BR



PedidoIn ternacio nal

Sealed,
taxas
import

Pontos-Chave do Código

1 Orquestração Central

Processar() coordena o fluxo fixo:
validação, cálculo e emissão —
invariável entre tipos de pedido.

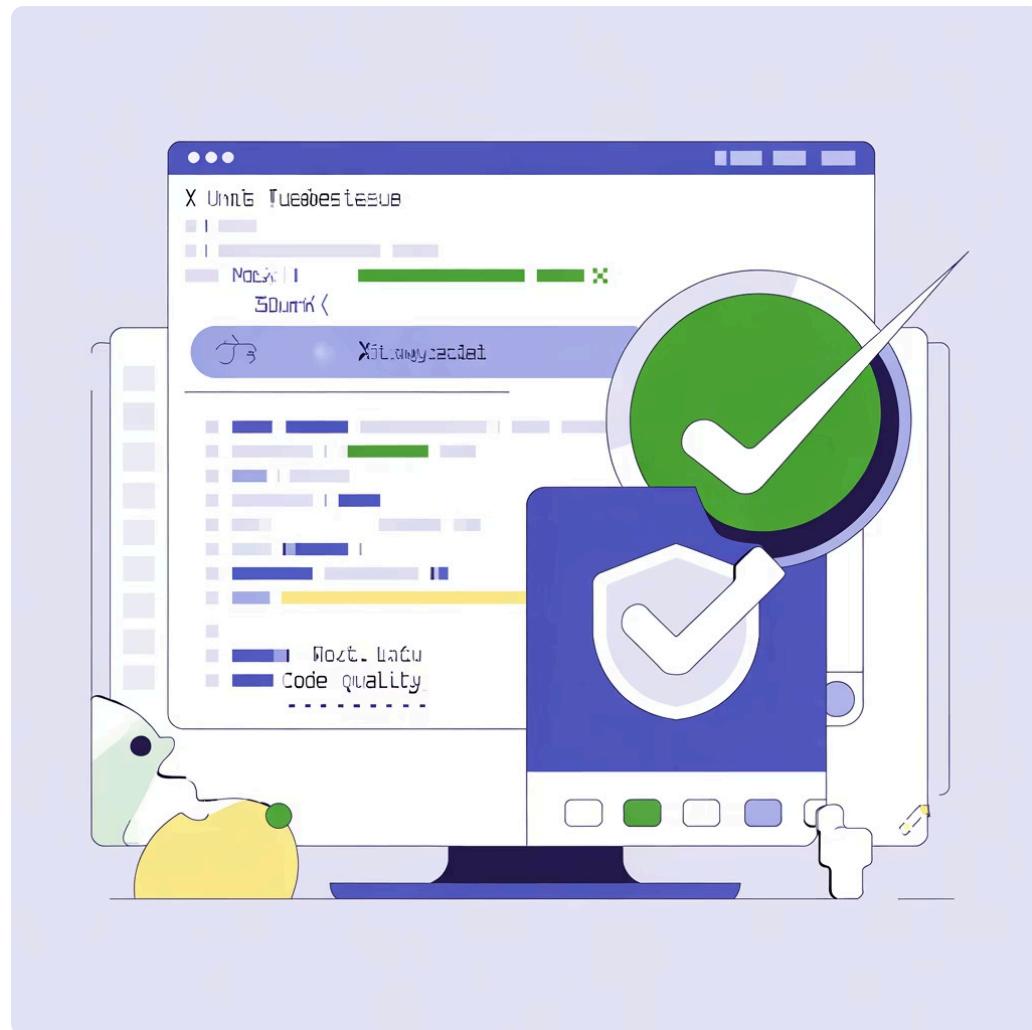
2 Ganchos Especializados

CalcularSubtotal() e EmitirRecibo()
são sobrescritos nas classes folha
(sealed) para lógica específica.

3 Injeção de Estratégias

Delegates de frete e promoção
injetados via construtor —
composição pura sem explosão de
subclasses.

Validação através de Testes



Testes com xUnit (.NET 9)

Suite completa de testes automatizados garante a solidez do design:

LSP em Ação

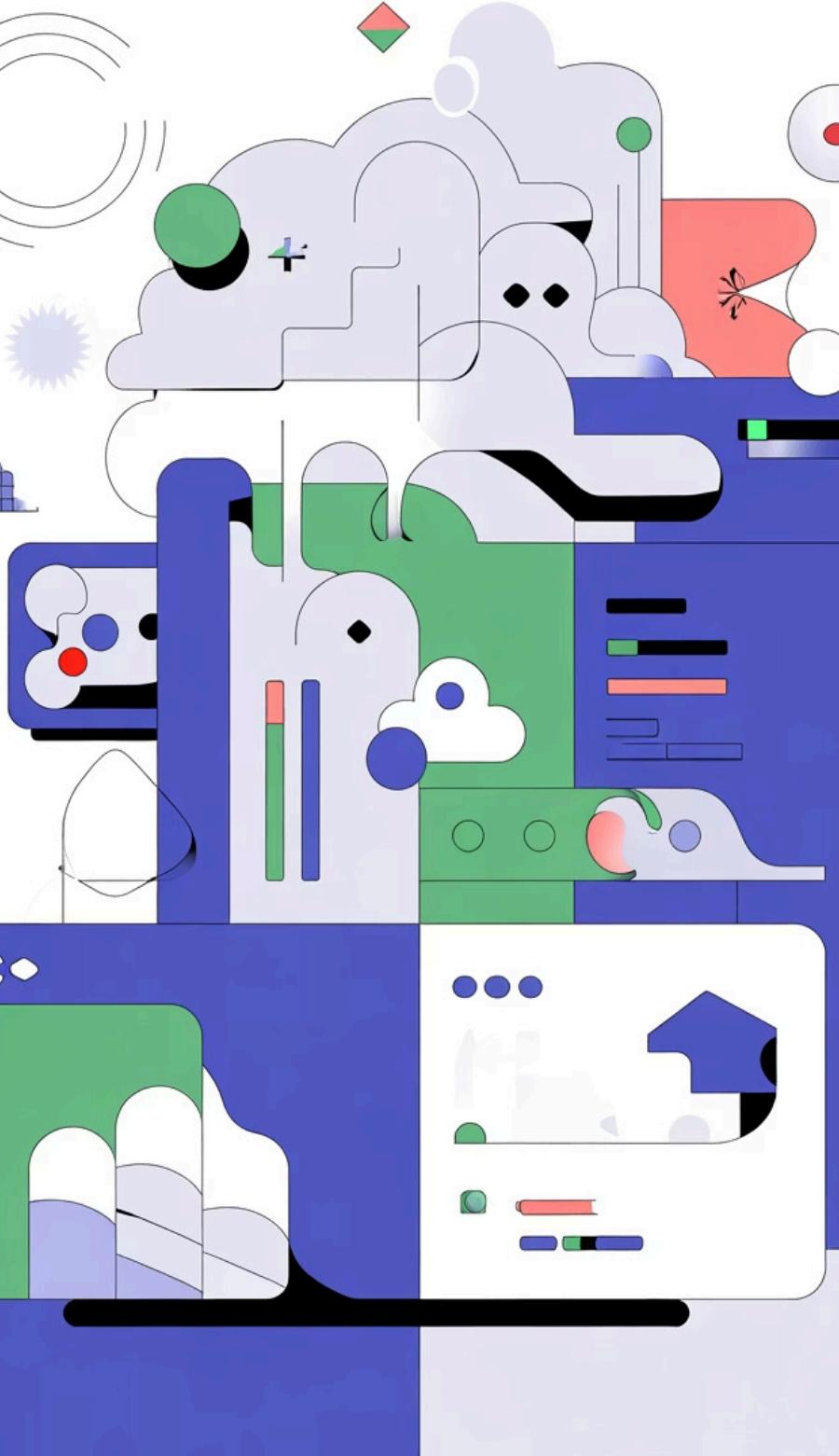
Cliente trata PedidoNacional e PedidoInternacional como Pedido sem downcast ou verificação de tipo.

Composição Plugável

Troca dinâmica entre Frete fixo ↔ Frete percentual e Promoção nula ↔ Cupom de desconto, sem modificar hierarquia.

Manutenibilidade

Código extensível e estável — novos comportamentos adicionados sem quebrar implementações existentes.



Resultados e Aprendizados

✓ Pontos Fortes

- Fluxo previsível e consistente via Template Method
- Delegates garantem flexibilidade com baixo acoplamento
- Testes comprovam LSP e composição plugável
- Código pronto para produção

→ Próximos Passos

- Migrar de delegates para interfaces (IFrete, IPromocao)
- Aumentar testabilidade e injeção de dependência
- Explorar padrão Strategy formalmente

Conclusão

Design equilibrado entre herança com propósito e composição estratégica — um exemplo sólido de **Clean Architecture** aplicada a domínio de negócios com padrões GoF.