

# **Padrão de Projeto: Strategy**

**Autores:** Kéven Moraes Patrício  
Bruno Moreira Lima  
Gabriel Honorato Santos Ferraz  
Claudio Pales Costa  
Artur Maia Coqueiro

# Sumário



Sistema de Delivery - Versão 1.0

Funções disponíveis:

GERENCIAMENTO DE CARDÁPIO:

- 1 - Cadastrar Novo Item
- 2 - Listar Itens

GERENCIAMENTO DE CLIENTES:

- 3 - Cadastrar Novo Cliente
- 4 - Listar Clientes

GERENCIAMENTO DE PEDIDOS:

- 5 - Registrar Novo Pedido
- 6 - Atualizar Status Pedido
- 7 - Consultar Pedidos por Status

RELATÓRIOS:

- 8 - Relatório Simplificado
- 9 - Relatório Detalhado

ENCERRAR SISTEMA:

- 10 - Sair

Insira o número da função desejada e tecle Enter:

## 1. O que é o padrão

**Strategy:** Explicação da técnica.

## 2. Sobre o problema: ○

que e como gerar os relatórios e os seus problemas.

**3. Solução:** Solução de como aplicar de modo simples.

## 4. Como funciona:

Funcionamento do projeto.

## 5. Estrutura visual:

Estrutura em forma de diagrama.

**6. Benefícios:** Benefícios de utilizar esse método.

# 1. O que é o Padrão Strategy?



- **O Padrão Strategy (Estratégia) é uma técnica de programação que permite trocar diferentes formas de fazer a mesma coisa sem alterar o código principal. É como ter várias "receitas" diferentes para cozinhar o mesmo prato.**

## 2. Sobre o Problema

**"Como gerar diferentes tipos de relatórios sem complicar o código?"**

- **Cenário hipotético:** "A gerência quer um relatório bem detalhado. A diretoria quer apenas um resumo rápido".
- **Abordagem ingênua:** Usar if-else para cada tipo de relatório.

```
public class GeradorRelatorio {  
    public String gerarRelatorio(String tipo, List<Order> pedidos) {  
        if (tipo.equals("detalhado")) {  
            // 50 Linhas de código para relatório detalhado  
            StringBuilder relatorio = new StringBuilder();  
            relatorio.append("=== RELATÓRIO DETALHADO ===");  
            // ... muito código aqui ...  
        } else if (tipo.equals("simplificado")) {  
            // 30 Linhas de código para relatório simplificado  
            StringBuilder relatorio = new StringBuilder();  
            relatorio.append("=== RELATÓRIO SIMPLIFICADO ===");  
            // ... mais código aqui ...  
        } else if (tipo.equals("grafico")) {  
            // 40 Linhas de código para relatório gráfico  
            // ... ainda mais código ...  
        }  
        // E se quisermos adicionar mais tipos? Mais IFs!  
    }  
}
```

### Consequências:

- ❌ Código gigante e confuso.
- ❌ Difícil de manter.
- ❌ Risco de quebrar algo que já funciona.

# 3. Solução

- Cada estratégia tem a sua própria classe

```
// Uso simples e limpo:  
RelatorioStrategy estrategia = new RelatorioDetalhadoStrategy(pedidos);  
String relatorio = estrategia.gerarRelatorio();  
  
// Ou trocar por outra estratégia facilmente:  
RelatorioStrategy estrategia = new RelatorioSimplificadoStrategy(pedidos);  
String relatorio = estrategia.gerarRelatorio();
```

- ✓ Código organizado e separado
- ✓ Fácil de adicionar novos tipos (criar nova classe)
- ✓ Fácil de testar cada tipo separadamente
- ✓ Não precisa mexer no código existente

# 4. Como funciona?

- **1. RelatorioStrategy (A “Receita Base”):**
  - Um "contrato" que obriga toda estratégia a ter o método `gerarRelatorio()`.

```
public interface RelatorioStrategy {  
    String gerarRelatorio();  
}
```

# 4. Como funciona?



- 2. As "Receitas Específicas" (Classes Concretas)

- **RelatorioDetalhadoStrategy**: Gera um relatório completo, com detalhes de cada pedido, cliente e itens
- **RelatorioSimplificadoStrategy**: Gera um relatório resumido, apenas com os totais gerais (total de pedidos, valor arrecadado, etc.).

```
public class RelatorioDetalhadoStrategy implements RelatorioStrategy {  
    private final List<Order> pedidosHoje;  
  
    public String gerarRelatorio() {  
        // Gera um relatório COMPLETO com TODOS os detalhes  
    }  
}
```

=== RELATÓRIO DETALHADO - 06/10/2025 ===

Pedido #001

Cliente: João Silva (ID: 123)

Telefone: (77) 99999-9999

Status: Entregue

Itens:

Pizza Margherita	Qtd: 2	Preço Unit: R\$ 25,00	Subtotal: R\$ 50,00
Coca-Cola 2L	Qtd: 1	Preço Unit: R\$ 8,00	Subtotal: R\$ 8,00

TOTAL DO PEDIDO: R\$ 58,00

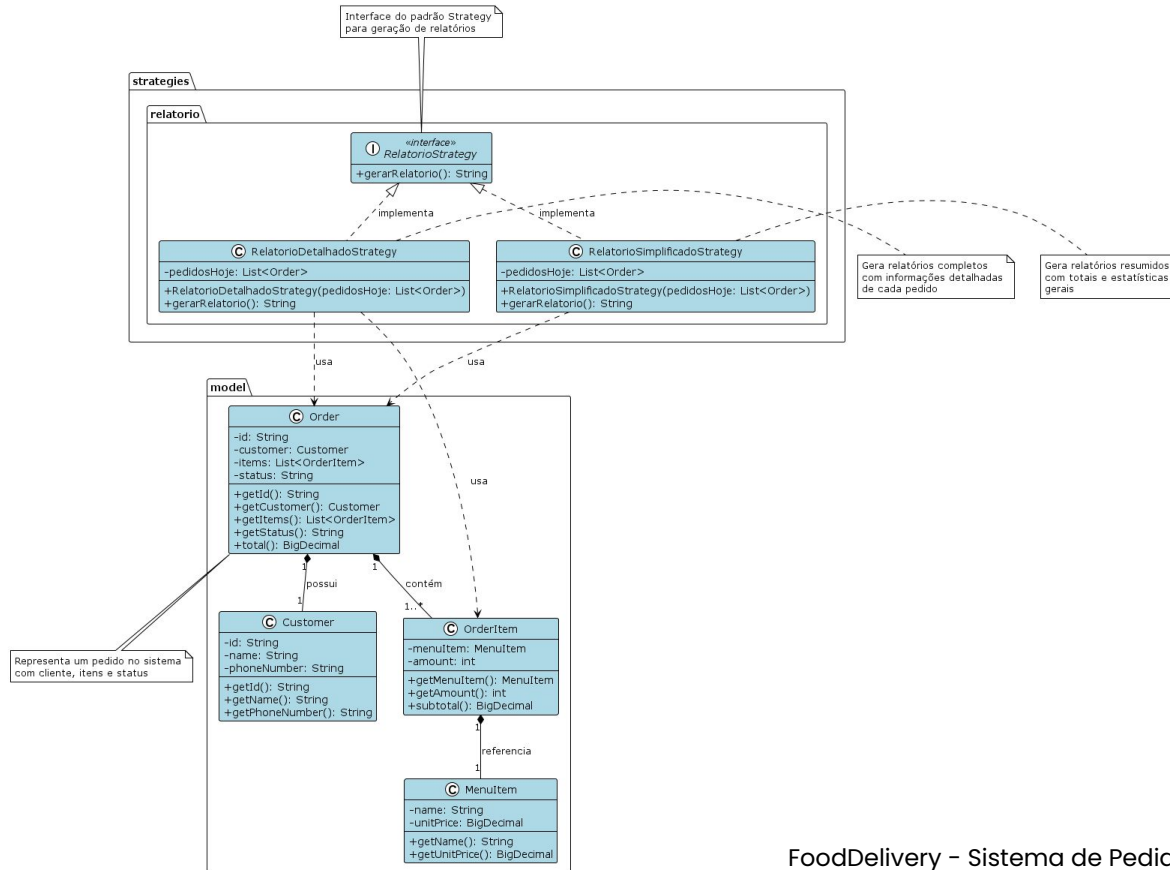
RESUMO GERAL:

Total de Pedidos: 15

Valor Total Arrecadado: R\$ 450,00

# 5. Estrutura visual

Sistema TiaLu Delivery - Diagrama de Classes





# 6. Benefícios

## 1. Flexibilidade

- Pode trocar entre diferentes tipos de relatório facilmente
- Cada cliente pode ter seu tipo preferido

## 2. Facilidade para Crescer

- Quer adicionar "Relatório em PDF"? → Crie nova classe
- Quer "Relatório com Gráficos"? → Crie nova classe
- **Não precisa mexer no código que já funciona!**

## 3. Código Limpo

- Cada estratégia é independente
- Fácil de entender e manter
- Fácil de testar

## 4. Reutilização

- Mesmas estratégias podem ser usadas em partes diferentes do sistema
- Código não fica duplicado