

# Relatório princípios SOLID em Java

## Introdução

Os princípios SOLID são diretrizes para o desenvolvimento de software que visam melhorar a qualidade, manutenibilidade e escalabilidade do código. Cada princípio aborda um aspecto específico do design de software.

### 1. Single Responsibility Principle (SRP)

O que faz: Garante que uma classe tenha apenas uma razão para mudar.

Código errado: A classe Pedido tinha múltiplas responsabilidades.

Código certo: As responsabilidades foram divididas em classes separadas (Pedido, PedidoRepository, EmailService).

### 2. Open/Closed Principle (OCP)

O que faz: Classes devem ser abertas para extensão, mas fechadas para modificação.

Código errado: A classe Relatorio precisava ser modificada para adicionar novos tipos de relatório.

Código certo: Novos tipos de relatório são adicionados sem modificar o código existente, usando interfaces.

### 3. Liskov Substitution Principle (LSP)

O que faz: Subtipos devem ser substituíveis por seus tipos base.

Código errado: Quadrado alterava o comportamento de Retangulo.

Código certo: Quadrado e Retangulo implementam a interface Forma, mantendo o comportamento esperado.

### 4. Interface Segregation Principle (ISP)

O que faz: Interfaces devem ser específicas para o cliente que as usa.

Código errado: A interface Trabalhador forçava Robo a implementar métodos irrelevantes.

Código certo: Interfaces segregadas (Trabalhador, Comedor, Descansador) permitem implementações específicas.

## 5. Dependency Inversion Principle (DIP)

O que faz: Dependenda de abstrações, não de implementações concretas.

Código errado: Interruptor dependia diretamente de Lampada.

Código certo: Interruptor depende da interface Dispositivo, permitindo flexibilidade.

## Conclusão

Seguir os princípios SOLID resulta em um código mais modular, flexível e fácil de manter. Cada princípio aborda um aspecto específico do design, garantindo que o software seja robusto e adaptável a mudanças.