

POR: BRUNO FAGANELLO NETO

---

# CONCEITOS DE SOLID

# O QUE É O SOLID?

- ▶ **SOLID** é um acrônimo dos cinco primeiros princípios da programação orientada a objetos e design de código identificados por **Robert C. Martin** (ou Uncle Bob) por volta do ano 2000. O acrônimo SOLID foi introduzido por **Michael Feathers**, após observar que os cinco princípios poderiam se encaixar nesta palavra.

OS 5 PRINCÍPIOS

Letra	Sigla	Nome	Descrição
S	SRP	Single Responsiblity Principle (Princípio da responsabilidade única)	Uma classe deve ter um, e somente um, motivo para mudar.
O	OCP	Open-Closed Principle (Princípio Aberto-Fechado)	Você deve ser capaz de estender um comportamento de uma classe, sem modificá-lo.
L	LSP	Liskov Substitution Principle (Princípio da substituição de Liskov)	As classes base devem ser substituíveis por suas classes derivadas.
I	ISP	Interface Segregation Principle (Princípio da Segregação da Interface)	Muitas interfaces específicas são melhores do que uma interface única.
D	DIP	Dependency Inversion Principle (Princípio da inversão da dependência)	Dependa de uma abstração e não de uma implementação.

## **VANTAGENS DE SUA APLICAÇÃO:**

- ▶ **Seja fácil de se manter, adaptar e se ajustar às alterações de escopo**
- ▶ **Seja testável e de fácil entendimento**
- ▶ **Seja extensível para alterações com o menor esforço necessário**
- ▶ **Que forneça o máximo de reaproveitamento**
- ▶ **Que permaneça o máximo de tempo possível em utilização**

### O QUE VOCÊ EVITA:

- ▶ **Dificuldade na testabilidade / criação de testes de unidade**
- ▶ **Código macarrônico, sem estrutura ou padrão**
- ▶ **Dificuldades de isolar funcionalidades**
- ▶ **Duplicação de código, uma alteração precisa ser feita em N pontos**
- ▶ **Fragilidade, o código quebra facilmente em vários pontos após alguma mudança**

**OBRIGADO**