

Introdução Padrões de Projeto

O que são padrões de projeto

São **soluções elegantes para problemas conhecidos** recorrentes no desenvolvimento de software que foram utilizados e testados no passado e continuam relevantes nos dias atuais.

- Foram catalogados e popularizados pelo livro "**Padrões de projeto - Soluções reutilizáveis de software orientado a objetos**" (os padrões da "GoF", de 1994/1995).

São divididos em 3 categorias:

- **creational (criação)**: que visam abstrair o processo de como objetos são criados na aplicação.
- **structural (estruturais)**: que lidam com a composição de classes e objetos.
- **behavioural (comportamentais)**: que caracterizam como as classes e objetos interagem e distribuem responsabilidades na aplicação.

SÃO APENAS SUGESTÕES DE SOFTWARE!!!

Benefícios:

- Você não precisa reinventar a roda
- Padrões universais facilitam o entendimento do seu projeto
- Evita refatoração desnecessária
- Ajuda na reutilização de código (conceito de **DRY** - *Don't repeat yourself*)
- Abstrai e nomeia partes particulares do projeto
- Ajuda na aplicação dos princípios do design orientado a objetos (**SOLID**)
- Facilitam a criação de testes unitários

Malefícios:

- Alguns padrões podem ser complexos até que você os compreenda
- Muito código para atingir um objetivo simples
- Podem trazer otimizações prematuras para seu código (**YAGNI** - *You Ain't Gonna Need It*)
- Se usados incorretamente, podem atrapalhar ao invés de ajudar

Princípios de design orientado a objetos (SOLID):

- [SRP - Single Responsibility Principle](#)

- [OCP - OpenClosed Principle](#)
- [LSP - Liskov Substitution Principle](#)
- [ISP - Interface Segregation Principle](#)
- [DIP - Dependency Inversion Principle](#)