**Padrões de projeto**

Um padrão de projeto é uma solução típica para um problema considerado comum em um projeto de software. Diferente de uma função ou um algoritmo, um padrão não é algo que podemos encontrar e copiar para utilizarmos em nosso projeto, mas sim um conceito geral, fazendo com que um mesmo padrão utilizado em dois projetos diferentes possam ter códigos completamente diferentes. Cada padrão possui três coisas em comum, que o descrevem:

- Propósito: Descreve brevemente o problema que ele soluciona e a solução.

- Motivação: Explica o fundo do problema e a solução que o torna possível.

- Estruturas de classes: Mostram cada parte do padrão e como elas se relacionam.

- Exemplos de código: Para algumas das linguagens de programação mais populares, para tornar a compreensão mais fácil.

É possível trabalhar por vários anos como programador sem conhecer padrões de projetos mas, mesmo assim, você estará aplicando alguns deles sem conhecer. Os padrões de projeto definem uma linguagem comum que você e seus colegas podem utilizar e se comunicar mais eficientemente, por exemplo: “Vamos usar singleton para resolver isso!”.

Os padrões mais básicos e de baixo nível podem ser chamados de *idiomáticos*, e normalmente se aplicam à uma única linguagem de programação. Já os padrões de mais alto nível e universais são chamados *padrões arquitetônicos*, e podem ser implementados em qualquer linguagem. Ao contrário de outros padrões, estes podem ser usados para projetar toda a arquitetura de uma aplicação.

Além dessas divisões, todos os padrões podem ser categorizados por seu *propósito*:

- Padrões Criacionais: Fornecem mecanismos de criação de objetos que aumentam a flexibilidade e a reutilização do código.

- Padrões Estruturais: Explicam como montar objetos e classes em estruturas maiores, enquanto ainda mantém as estruturas flexíveis e eficientes.

- Padrões comportamentais: Cuidam da comunicação eficiente e da atribuição de responsabilidades aos objetos.