

## Resenha Capítulo 5 - Princípios de Projeto

Marco Tulio Valente, escritor da obra “Engenharia de Software Moderna”, atualmente é professor associado do Departamento de Ciências da Computação da UFMG.

Em seu livro mencionado, Marco aborda um assunto fundamental para toda a Ciência da Computação. Outro grande engenheiro de software, Frederick Brooks, também realça a importância de um dos pilares de um projeto de software mencionado neste capítulo do livro, a “Integridade Conceitual”, em uma data comemorativa de 20 anos de um de seus livros.

A integridade conceitual é essencial no projeto de software. Isso significa que um sistema não deve ser um conjunto descoordenado de funcionalidades, mas sim um todo coerente. A falta de integridade conceitual pode resultar em inconsistências, como o uso de diferentes padrões de nomenclatura, frameworks distintos para a mesma finalidade e soluções variadas para problemas similares. Frederick Brooks defendeu enfaticamente esse princípio, afirmando que é melhor um sistema oferecer um conjunto coerente de funcionalidades do que um conjunto desconexo de boas ideias.

Outro conceito importante é o ocultamento de informação, promovendo modularização e desenvolvimento paralelo. Esse princípio facilita a manutenção e a compreensão do sistema, além de permitir a troca segura de componentes sem impactar o restante da implementação. Quando bem aplicado, o ocultamento de informação reduz o tempo de desenvolvimento e melhora a flexibilidade a mudanças.

A coesão e o acoplamento são também propriedades fundamentais para um bom projeto de software. A coesão estabelece que cada classe deve ter uma única responsabilidade, o que facilita a manutenção e compreensão do sistema. O acoplamento, por outro lado, refere-se à dependência entre classes, sendo desejável minimizá-lo para aumentar a estabilidade e reusabilidade do software.

Muitos dos princípios mais importantes mencionados, se comparam muito a uma Programação Orientada a Objetos, os princípios e nomenclaturas são basicamente as mesmas. Inclusive, o princípio SOLID, também presente na Programação Orientada a Objetos, porém com algumas adições de conceitos. Apesar de não serem presentes no princípio básico SOLID, são conceitos e definições que também estão presentes na Programação Orientada a Objetos, a Herança e o Demeter.

Em suma, um bom projeto de software seguindo os princípios mencionados, segue os padrões orientados a objetos priorizando a integração conceitual, modularização, coesão, baixo acoplamento e adesão a princípios básicos como o SOLID. Essas diretrizes resultam em sistemas mais fáceis de entender, manter e evoluir, garantindo maior qualidade e eficiência ao longo do ciclo de vida do software, não apenas para desenvolvedores, mas com clientes, atores de outras áreas também presentes, coordenadores, etc, e esses princípios podem ser usados ou utilizados em qualquer tipo de software.