

# Resenha: *On the Criteria to be Used in Decomposing Systems into Modules*

Diogo Brunoro

September 2025

O artigo de David Parnas, publicado em 1972, aborda a importância da **modularização** no desenvolvimento de sistemas de software. O autor argumenta que dividir um sistema em módulos permite reduzir o tempo de desenvolvimento, já que diferentes equipes podem trabalhar em paralelo, além de aumentar a **flexibilidade** e a **compreensibilidade** do sistema. O texto apresenta a discussão de critérios que devem ser utilizados na decomposição de sistemas em módulos. Parnas utiliza como exemplo um sistema de produção de índice KWIC (*Key Word In Context*) e mostra duas abordagens de modularização:

1. **Modularização tradicional:** baseada no fluxo de processamento, com módulos de entrada, circular shift, ordenação, saída e controle.
2. **Modularização alternativa:** baseada no princípio de *information hiding*, em que cada módulo esconde uma decisão de projeto dos demais, como a forma de armazenamento ou o formato de dados.

A análise comparativa mostra que a abordagem tradicional cria dependências entre módulos, o que dificulta futuras alterações. Já a abordagem de *information hiding* isola as decisões de projeto, facilitando mudanças, aumentando a clareza e permitindo maior independência no desenvolvimento.

Embora essa segunda abordagem possa parecer menos eficiente em termos de execução devido ao maior número de chamadas entre módulos, Parnas sugere técnicas de implementação que permitem manter a eficiência sem abrir mão das vantagens conceituais. Ele também demonstra que essa forma de decomposição facilita o reaproveitamento de código e garante maior adaptabilidade do sistema a mudanças. O ponto principal defendido por Parnas é que não se deve modularizar um sistema apenas com base em fluxogramas ou na sequência de execução. Em vez disso, a decomposição deve ser feita escondendo as decisões de projeto mais propensas a mudanças dentro de módulos específicos. Essa estratégia torna o software mais organizado, sustentável e preparado para evoluções futuras, estabelecendo as bases de um princípio que até hoje influencia o design de software moderno.