

## **Resenha sobre o artigo: On the Criteria To Be Used in Decomposing Systems into Modules de D.L. Parnas**

O artigo de David L. Parnas, publicado em 1972, revoluciona a forma de pensar sobre modularização de sistemas de software ao questionar os critérios convencionais de divisão de sistemas. O autor utiliza um sistema KWIC Index como exemplo, comparando duas abordagens distintas: uma convencional baseada no fluxo de processamento (dividindo por etapas sequenciais) e outra inovadora fundamentada no conceito de "information hiding" (ocultação de informação). Esta comparação demonstra que diferentes critérios produzem sistemas com características drasticamente diferentes em termos de flexibilidade, manutenção e compreensibilidade.

A principal contribuição teórica é a introdução do "information hiding" como critério fundamental para a decomposição modular. Parnas propõe que cada módulo deve ocultar uma decisão de design específica dos demais, revelando o mínimo sobre seu funcionamento interno. Esta abordagem oferece vantagens substanciais: mudanças ficam confinadas a módulos específicos, desenvolvimento independente torna-se mais eficiente através de interfaces abstratas, e a compreensibilidade melhora permitindo estudo isolado de cada módulo. O autor também introduz o conceito de estrutura hierárquica, onde módulos superiores dependem apenas dos inferiores, estabelecendo bases para arquiteturas em camadas.

As ideias de Parnas anteciparam princípios que se tornaram centrais na engenharia de software moderna, incluindo encapsulamento, abstração de dados e orientação a objetos. A separação entre representação executável e outras representações do programa revela uma visão sofisticada sobre o ciclo de vida do software. Para desenvolvedores atuais, especialmente em aplicações frontend e backend, os conceitos oferecem diretrizes valiosas para criar sistemas mais notáveis através da separação adequada de responsabilidades e ocultação de detalhes de implementação.

Este trabalho permanece como referência fundamental na arquitetura de software, demonstrando como decisões sobre estrutura de código impactam profundamente a produtividade e qualidade dos sistemas. A clareza da argumentação e exemplos práticos tornam as ideias acessíveis e aplicáveis. Embora considerações sobre eficiência computacional possam parecer menos relevantes hoje, os princípios de modularização continuam essenciais para criar software flexível, compreensível e evolutivo.

**Aluno: João Marcelo Carvalho Pereira Araújo**

**Professor: João Paulo Aramuni**

**Disciplina: Projeto de Software**