

# Resenha do Artigo

O artigo de David L. Parnas é considerado um marco na história da engenharia de software, pois introduz a noção fundamental de ocultamento de informações (information hiding) como critério principal para a modularização de sistemas. Parnas argumenta que a forma como um sistema é decomposto em módulos impacta diretamente sua flexibilidade, compreensibilidade e capacidade de manutenção .

## Contexto e Motivação

Na época, a modularização era entendida principalmente como uma divisão de programas em sub-rotinas baseadas no fluxo de processamento (por exemplo, etapas de entrada, processamento e saída). Parnas critica essa abordagem por não oferecer suporte adequado à evolução e manutenção de sistemas complexos. Em contrapartida, ele propõe que a decomposição deve se basear nas decisões de projeto suscetíveis a mudanças, de modo que cada módulo esconda detalhes que possam variar ao longo do tempo .

## Estudo de Caso: KWIC Index

Para ilustrar seu ponto, Parnas apresenta o problema do KWIC (Key Word In Context), um sistema que gera índices de palavras em contexto a partir de linhas de texto circularmente deslocadas. Ele compara duas formas de modularização:

- Modularização 1 (convencional): baseada em fases de processamento (entrada, deslocamento, ordenação, saída).
- Modularização 2 (alternativa): baseada no ocultamento de informações, onde cada módulo encapsula detalhes de implementação, como armazenamento de linhas ou forma de calcular os deslocamentos.

A comparação mostra que a segunda abordagem permite isolar mudanças (como formato de entrada, estrutura de armazenamento ou método de ordenação), afetando apenas um módulo em vez de todo o sistema .

## CrITÉrios Propostos

O autor defende que cada módulo deve ocultar uma decisão de projeto que provavelmente mudará. Alguns exemplos incluem:

- Estrutura de dados e procedimentos de acesso;
- Formatos de controle em sistemas operacionais;
- Codificação de caracteres e ordenação alfabética;
- Sequência de processamento de itens.

Essa forma de modularização favorece a independência entre equipes, reduz a necessidade de coordenação constante e aumenta a compreensão localizada do sistema .

## **Implicações e Impacto**

Parnas também discute preocupações com a eficiência: modularizações baseadas em ocultamento podem parecer menos eficientes devido ao aumento de chamadas entre módulos. Contudo, ele sugere que estratégias de implementação, como montadores e otimizações de compiladores, podem mitigar esse custo.

A principal conclusão do artigo é que não se deve modularizar sistemas a partir de fluxogramas ou etapas de execução, mas sim a partir de decisões de projeto críticas e sujeitas a mudança. Esse princípio inaugurou o paradigma moderno de design orientado a abstrações e influenciou diretamente a evolução de conceitos como orientação a objetos e design de APIs .