



## Modularidade

- ✓ conceito antigo e já visto na Arquitetura de Software  
*O software é dividido em componentes separados com características particulares, que são integrados de forma a atender os requisitos.*

Refleta o extremo: um software de um único módulo:

- número de caminhos de controle;
- número de variáveis;
- quantidade de funções e integração destas funções;
- complexidade de uma forma geral.

ESOF: PROJETO DE SOFTWARE II



## Complexidade x Esforço

Sejam:

- $C(x)$  = função de definição da complexidade de um problema;
- $E(x)$  = função de definição do esforço.

Se  $C(p1) > C(p2) \Leftrightarrow E(p1) > E(p2)$  - óbvio!

Experimentalmente, temos:  $C(p1 + p2) > C(p1) + C(p2)$  e,  
Nesta hipótese:  $E(p1 + p2) > E(p1) + E(p2)$ .

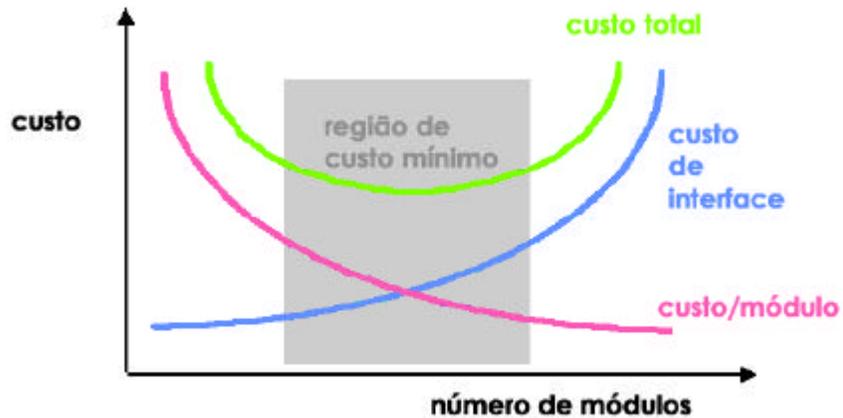
- ✓ A conclusão: se dividirmos o SW infinitamente, chegaremos a um esforço infinitamente pequeno. (?)

O que interfere nesta conclusão e que a torna incorreta?

ESOF: PROJETO DE SOFTWARE II



## Modularidade e Custo do Software



ESOF: PROJETO DE SOFTWARE II



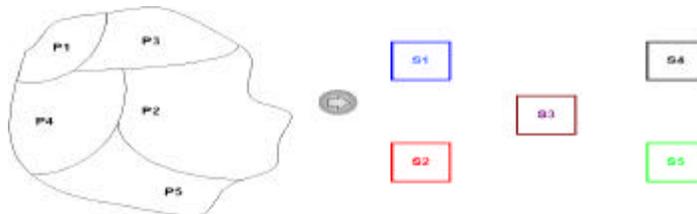
## Arquitetura do Software

### Características da Arquitetura:

- estrutura hierárquica de componentes (módulos), derivados de um processo de divisão em partições;
- estrutura de dados.

### Solução da Arquitetura:

- cada parte do problema é tratada e resolvida por um ou mais elementos de software.



ESOF: PROJETO DE SOFTWARE II



## Hierarquia de Controle

- representa a organização (geralmente hierárquica) de componentes (módulos) de programa;
- não representa aspectos procedimentais de software (sequências de processos, ocorrência e ordem de decisões etc)
- notação mais usual: diagrama de blocos na forma de árvore
- representa duas características da arquitetura: visibilidade e conectividade

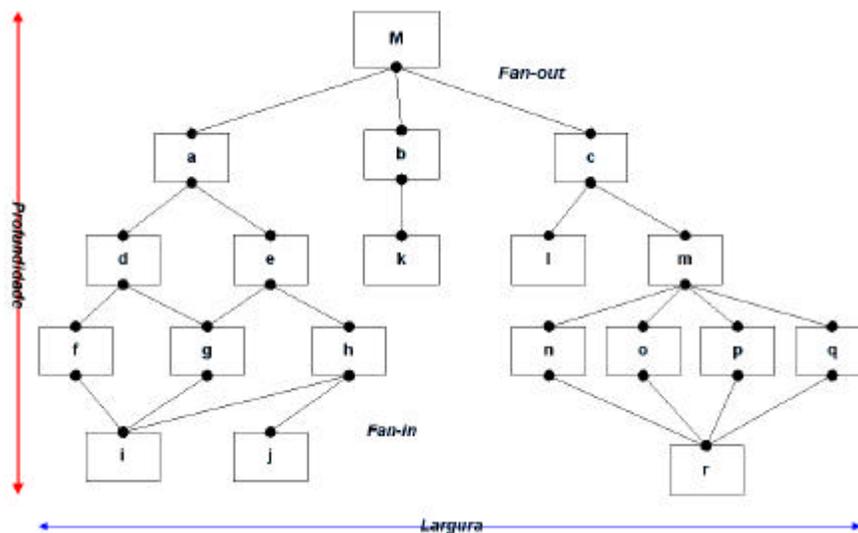
### Diagramas em árvore:

- profundidade e largura
- Fan-out
- Fan-in
- módulo que controla: superordenado ao controlado
- módulo controlado: subordinado ao controlador

ESOF: PROJETO DE SOFTWARE II



## Exemplo



ESOF: PROJETO DE SOFTWARE II



## Estrutura de Dados

---

- representação do relacionamento lógico entre os diferentes elementos de dados
- determina a organização, os métodos de acesso, o grau de associatividade e as alternativas para processar as informações
- há liberdade para o projetista compor tais estruturas
- há estruturas de dados clássicas capazes de formar blocos de solução, são eles:
  1. Item escalar
  2. Vetor sequencial
  3. Espaço n-dimensional
  4. Lista interligada
  5. Árvore hierárquica

---

ESOF: PROJETO DE SOFTWARE II



## Procedimento de Software

---

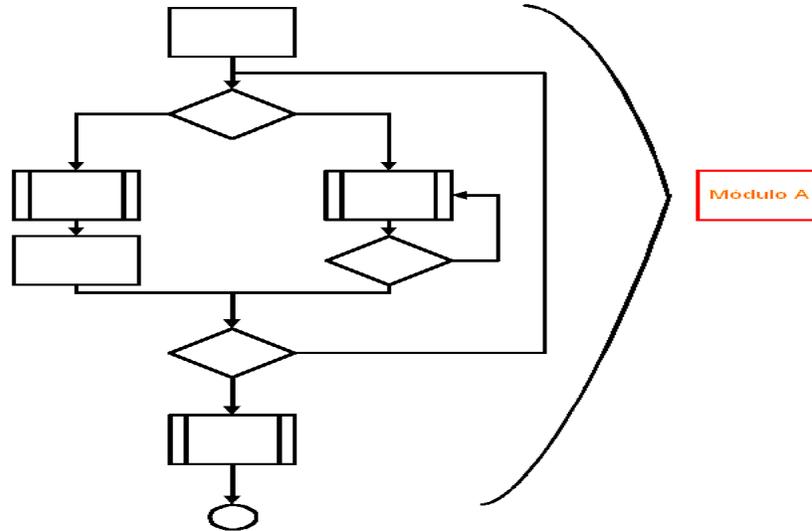
- especifica detalhes de processamento de cada módulo individualmente
- é uma especificação precisa do processamento que inclui:
  - a) sequência de eventos;
  - b) pontos de decisão;
  - c) operações repetitivas;
  - d) estrutura e organização de dados.
- o procedimento é disposto em camadas;
- há uma relação entre estrutura e procedimento;
- o procedimento deve ser refinado.

---

ESOF: PROJETO DE SOFTWARE II



## Procedimento de Software



ESOF: PROJETO DE SOFTWARE II



## Ocultação de Informações

Como obter o melhor conjunto de módulos, neste processo de decomposição???

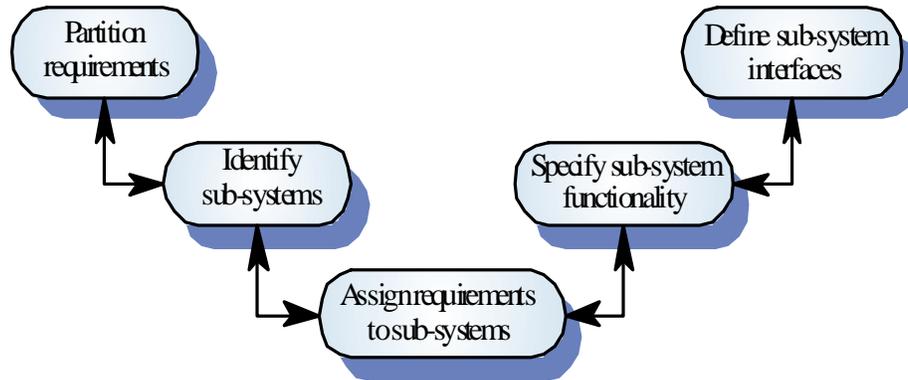
### Princípio da ocultação da informação:

- cada módulo é especificado e projetado de tal forma que todas as informações contidas nele sejam inacessíveis aos demais módulos (que não necessitem de tais informações).
- as comunicações entre módulos são restritas a informações necessárias à obtenção de uma dada tarefa do software.
- ocultação é aplicada a detalhes procedimentais e a qualquer estrutura de dados que é utilizada pelo módulo

ESOF: PROJETO DE SOFTWARE II



## Processo de Projeto

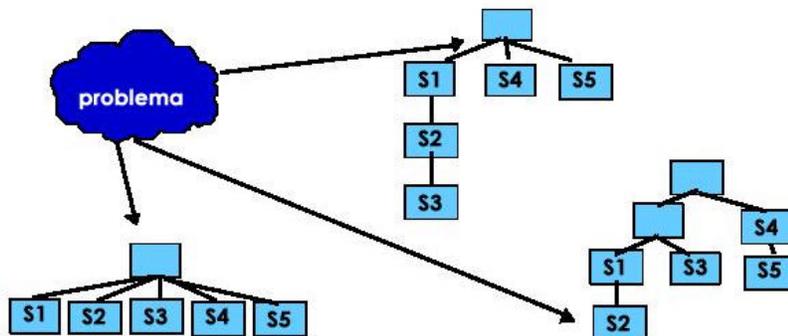


ESOF: PROJETO DE SOFTWARE II



## Diagrama Hierárquico de Funções - DHF

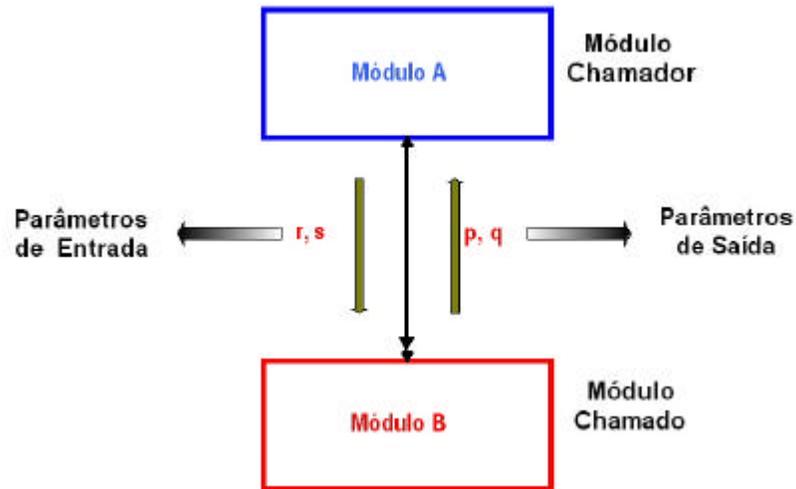
Proposição básica: decomposição do problema em partes: módulos de programas integráveis



ESOF: PROJETO DE SOFTWARE II



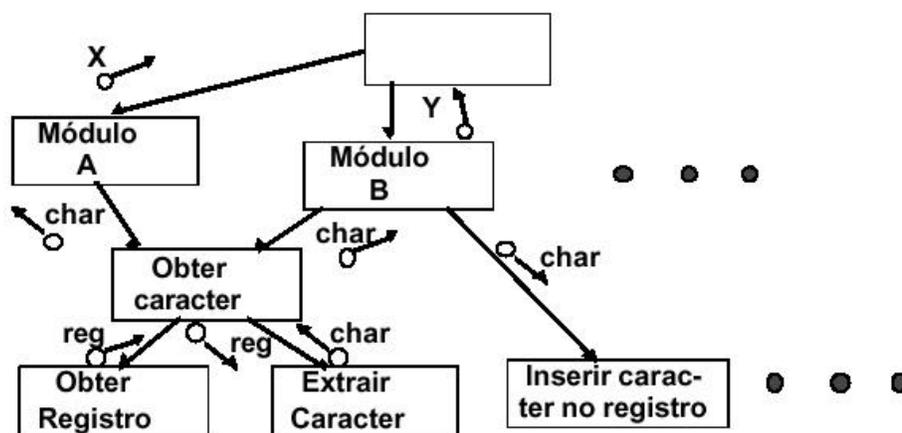
## Diagrama Hierárquico de Funções - DHF



ESOF: PROJETO DE SOFTWARE II



## Diagrama Hierárquico de Funções - DHF



ESOF: PROJETO DE SOFTWARE II



## Exercícios

---

1. Desenvolva um D.H.F. para o compilador já tratado em aulas anteriores.
2. Desenvolva um projeto procedimental para um programa que aceite um texto arbitrariamente longo como entrada e produza uma lista de palavras, além da frequência de sua ocorrência como saída.
3. Por que a ocultação da informação é um aspecto fundamental de projeto. Justifique sua opinião.
4. A partir da análise de complexidade e esforço, o que você conclui como sendo a solução mais viável para a construção dos módulos que caracterizam a solução de um projeto de software?