Árvores

Prof. Alberto Costa Neto

Motivação

- Até o momento vimos estruturas lineares
  - Listas, Filas, Pilhas

- O principal problema são os trade-offs:
  - Espaço de armazenamento x flexibilidade
  - Tempo de inserção x Tempo de busca

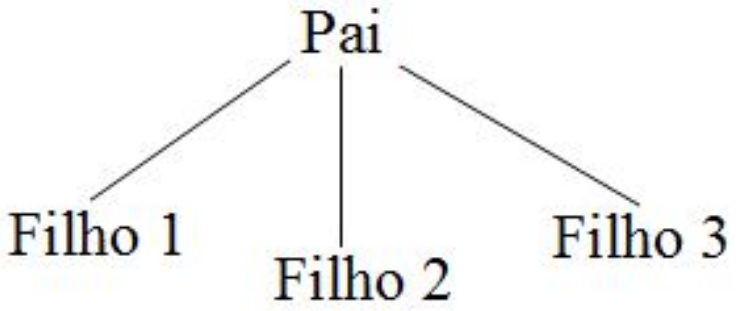
- Por isso, vamos estudar estruturas hierárquicas que lidam melhor com esses trade-offs

Árvores

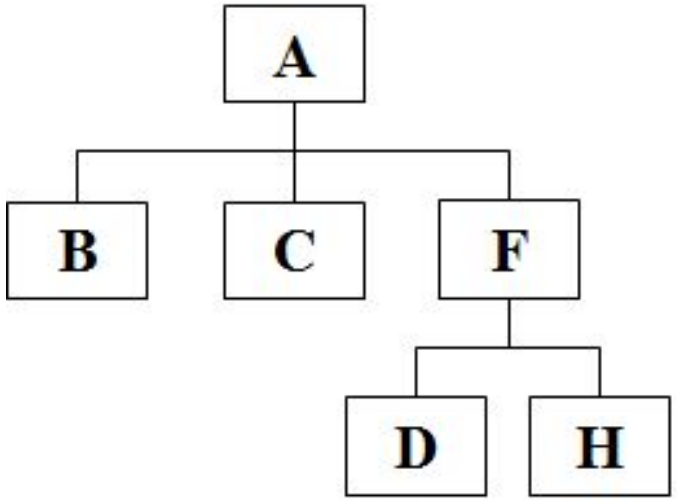
- **Definição:** É uma estrutura onde a relação entre seus elementos é de um para vários, também denominada estrutura hierárquica.

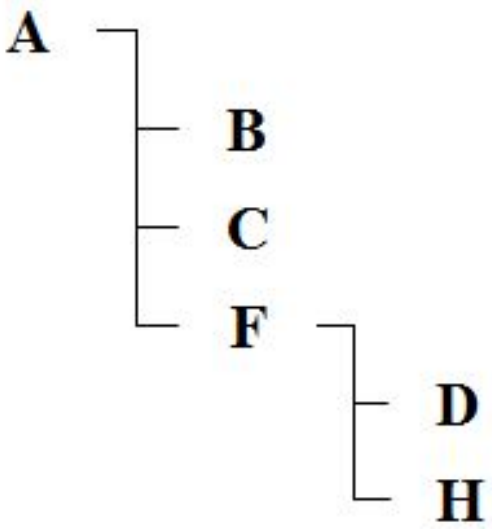
- Uma árvore consiste em um conjunto de nós, tal que:
  - Existe um nó denominado **raiz**.
  - Os **demais nós formam m (m >= 0) conjuntos** onde **cada um deles também é uma árvore.**

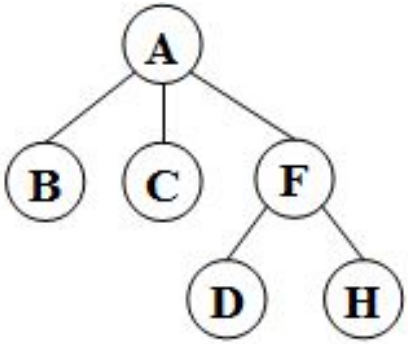
- Graficamente temos:



Notações Gráficas





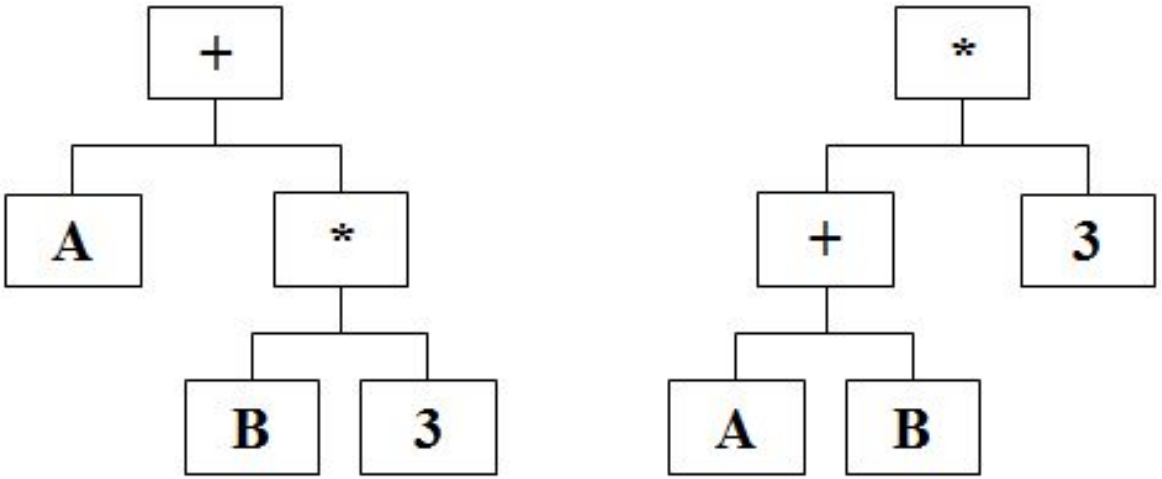


Aplicações de Árvores

Árvores podem ser usadas em diversos tipos de aplicações:

- Aplicações onde é necessário recuperar informações rapidamente (SGBD)
- Aplicações onde é necessário armazenar expressões matemáticas.
- Programas onde as informações têm que ser estruturadas de forma hierárquica (Árvore Sintática de um programa)

Para representar as expressões matemáticas A + B * 3 e (A + B) * 3 poderíamos usar uma das árvores abaixo:

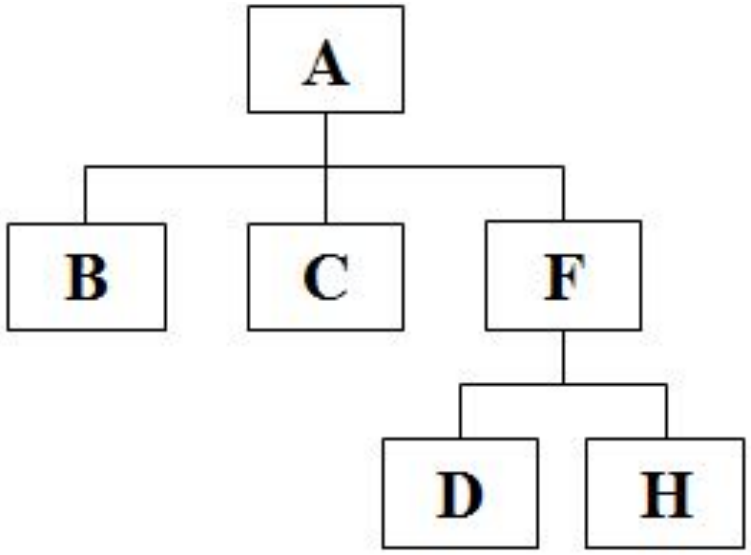


Terminologia

- Cada elemento de uma árvore pode ser classificado como:

Classificação	Descrição
Pai	quando fica imediatamente acima de outro nó.
Filho	quando fica imediatamente abaixo de outro nó.
Irmãos	se os nós têm o mesmo pai.
Raiz	se não tem nó pai (pode ter filhos).
Folha ou Terminal	quando não possui filhos (pode ter pai).

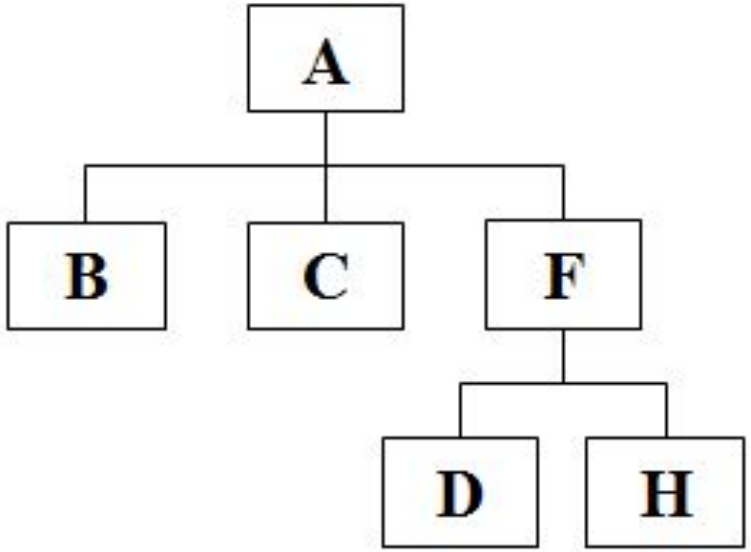
Terminologia



Dada a árvore acima, temos:

- A, B, C, D, F e H são **nós**;
- A é **pai** de B, C e F. F é **pai** de D e H;
- B, C e F são **filhos** de A. D e H são **filhos** de F;
- B, C e F são **irmãos**. D e H são **irmãos**;
- A é a **raiz** da árvores;
- B, C, D e H são **folhas**.

Terminologia



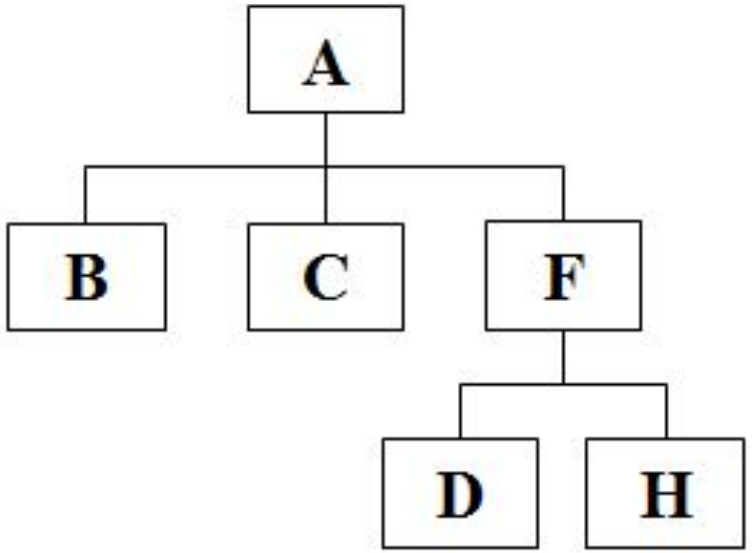
Ancestral: A é ancestral de B, C, F, D e H.

Descendente: B, C, F, D e H são descendentes de A. D e H são descendentes de F.

Arestas: são as ligações entre os nós.

Caminho: é uma seqüência de nós n1, n2, ... , ni tal que os nós sejam distintos e que exista uma aresta entre cada par de nós.

Terminologia



Comprimento de um caminho: número de arestas que um caminho contém.

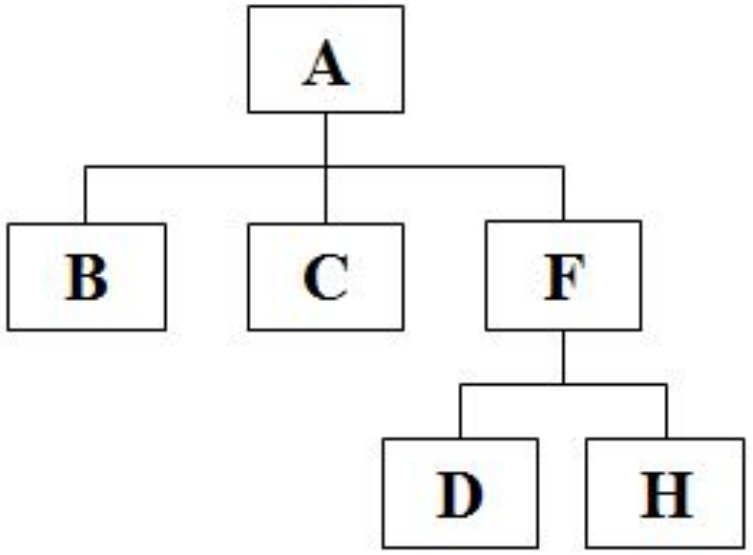
Ex: A - F - D tem comprimento 2.

Nível de um nó: comprimento de um caminho que vai da raiz até o nó, sendo o nível do nó raiz igual a 0.

Ex: A tem nível 0; B nível 1 e D nível 2.

Altura de uma árvore: é o nível mais alto de uma árvore. A árvore do exemplo tem altura 2.

Terminologia



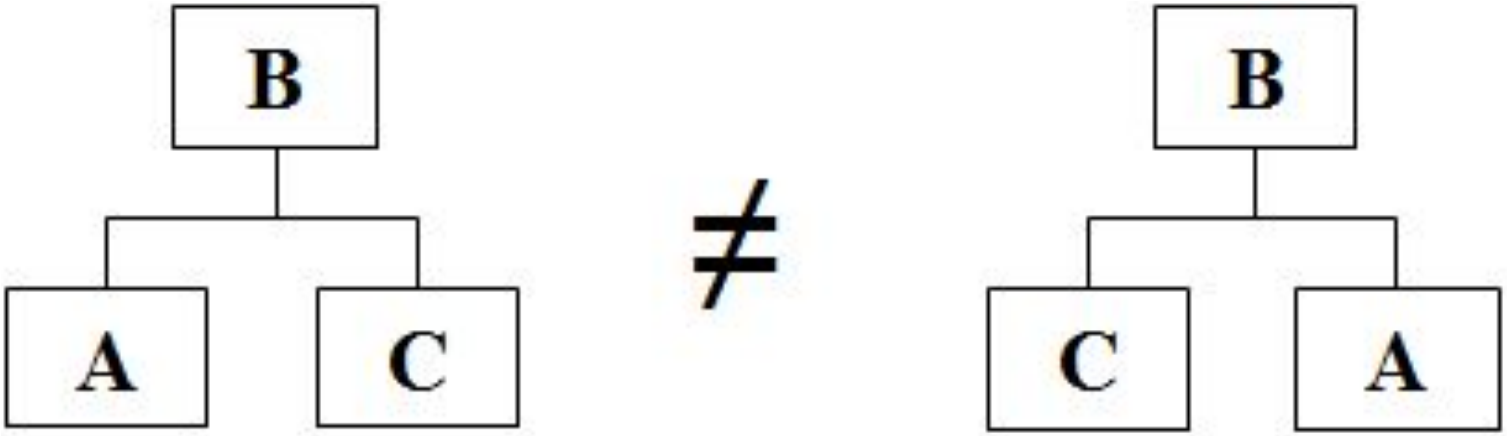
Grau de um nó: quantidade de subárvores de um nó.

Ex: A tem grau 3, F tem grau 2 e B tem grau 0.

Grau da árvore: quantidade máxima de subárvores para cada nó da árvore.

Subárvore: é uma árvore formada por um dos nós filhos e seus descendentes.

Árvore Ordenada



Implementação Sequencial

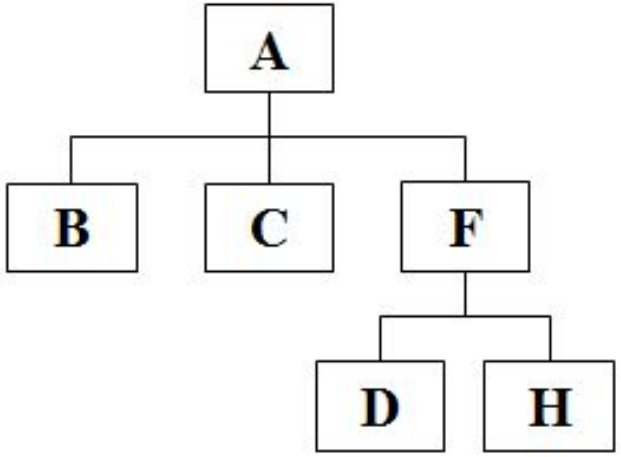
Implementação Sequencial

- Na implementação sequencial os nós de uma árvore são **colocados sequencialmente na memória**.

- Há, basicamente, duas formas de implementação sequencial de árvores.
  - Quando o **grau** da árvore é **conhecido**;
  - Quando o **grau** da árvore é **desconhecido**.

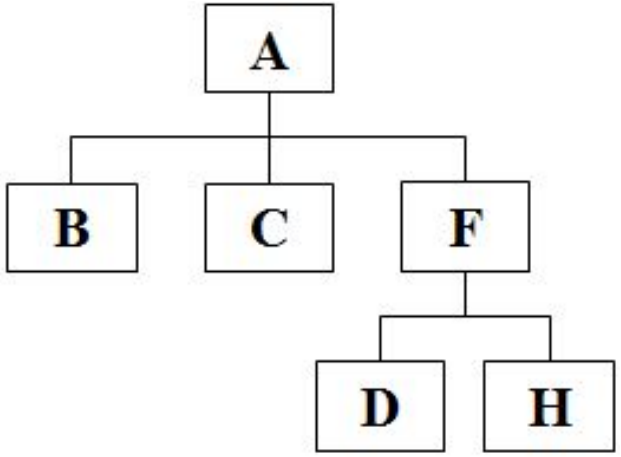
Implementação Sequencial (com grau conhecido)

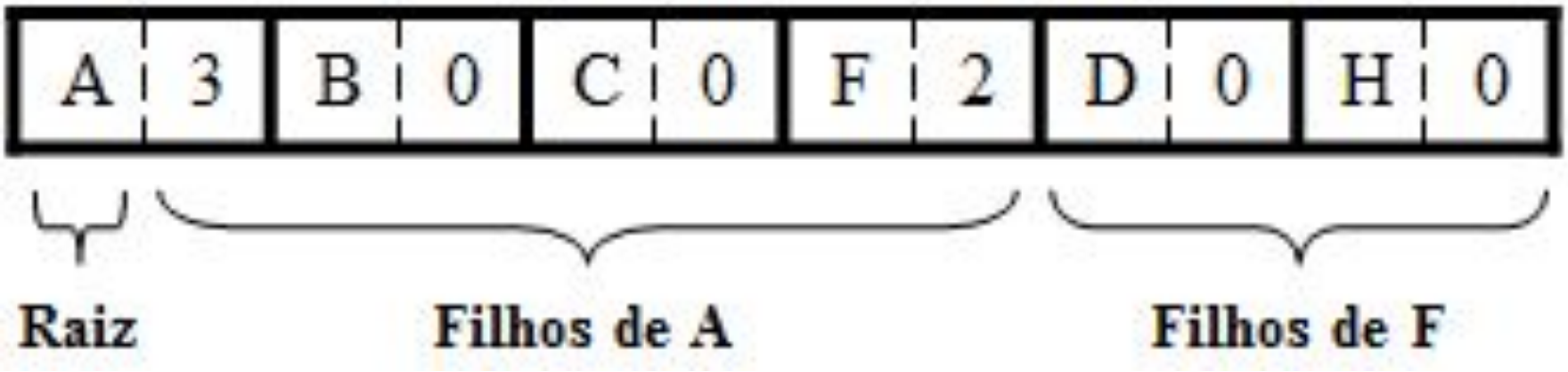
- Com grau conhecido 3:



Implementação Sequencial (com grau desconhecido)

- Com grau desconhecido:





Desvantagens de cada opção de implementação

Com **grau de árvore conhecido**:

- Existe desperdício de espaço quando há nós com menos subárvores que o grau da árvore.
- É necessário conhecer a altura da árvore antes de criar a estrutura.

Com **grau de árvore desconhecido**:

- Há desperdício de espaço devido ao indicador de grau do nó.
- É necessário mais processamento para manipular os nós. Inserções e remoções provocam movimentações dos nós.

Implementação Encadeada

Implementação Encadeada

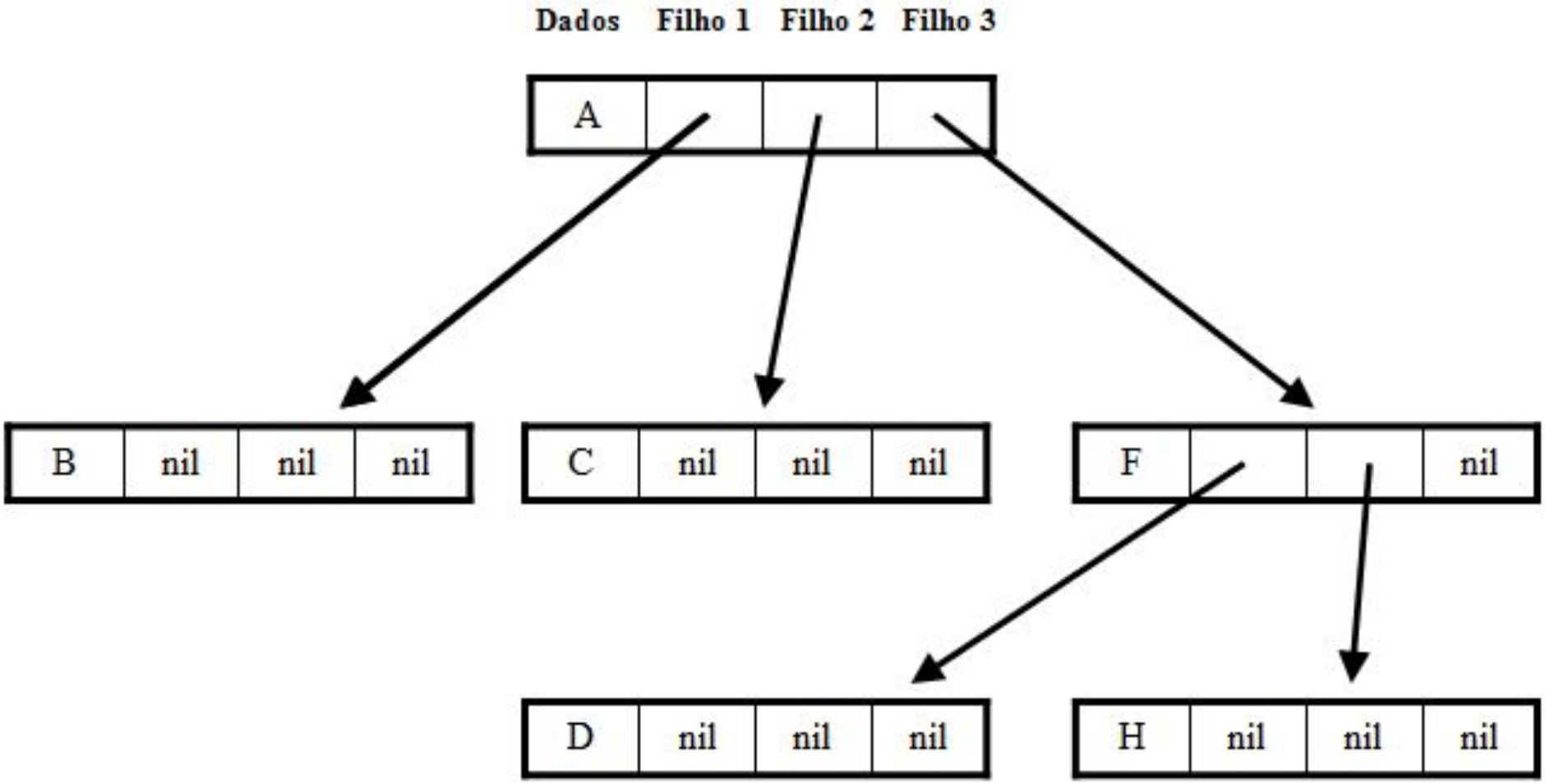
A implementação encadeada oferece:

- Melhor aproveitamento de memória.
- Chance de alterar dinamicamente o tamanho da árvore.

Há basicamente 2 formas de implementação encadeada:

- Com grau conhecido
- Com grau desconhecido

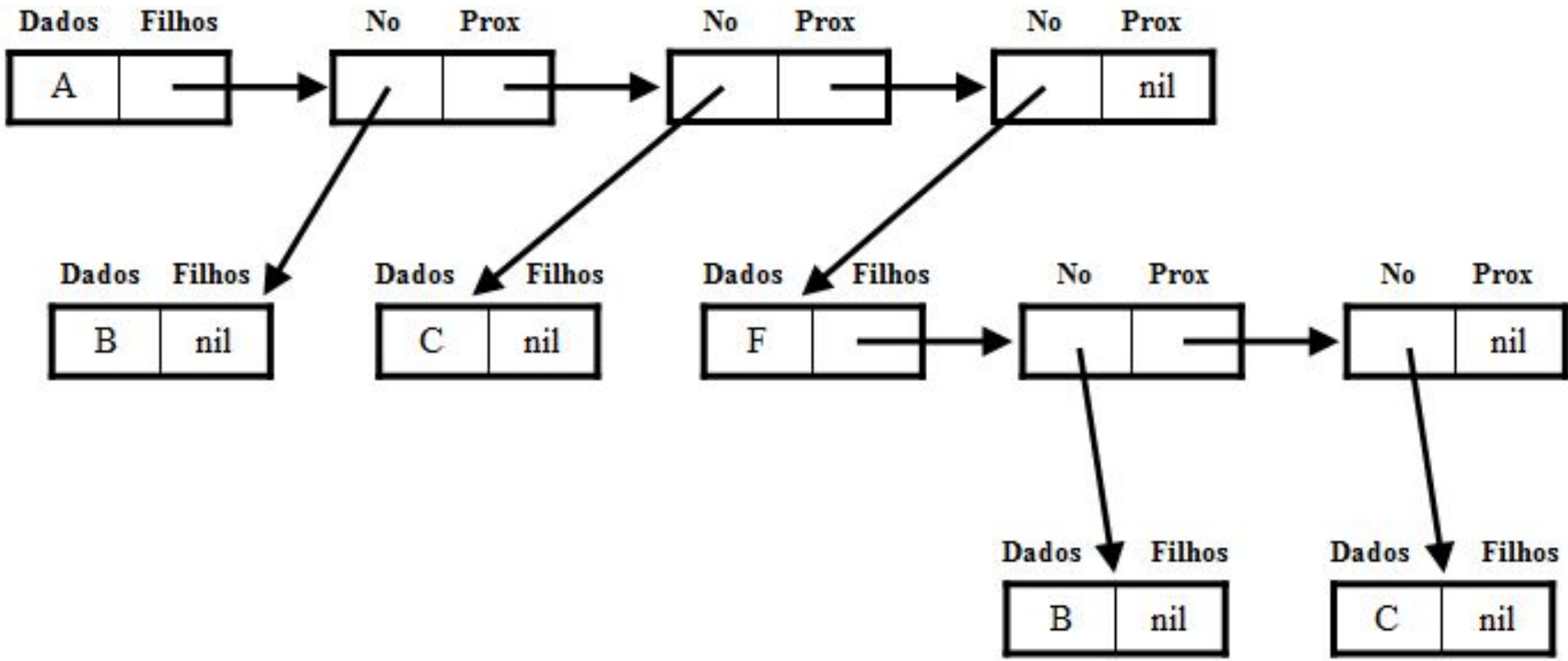
Implementação encadeada com grau conhecido



Implementação encadeada com grau desconhecido

Cada nó da árvore contém:

- Os dados
- Lista encadeada com apontadores para n filhos (n >= 0)



Sugestões de Estudo

Estruturas de Dados (Nina Edelweiss)

- Capítulo 5

Estruturas de dados (Paulo Veloso)

- Seções 7.1 a 7.3