UNINASSAU
Campus Boa Viagem

Programação Orientada e objetos e Estrutura de dados

PROFESSOR:

Adilson da Silva

CURSO (2023.2)

Apresentação baseada em apresentação do professor Leopoldo França

MA UNG UNINORTE UNESC UNIFAEL UNI7
Grupo Ser Educacional

UNINASSAU
Campus Graças

LISTAS

Listas

UNINASSAU
Campus Graças

- Uma das formas mais comumente usadas para se manter dados agrupados é a lista
- Afinal, quem nunca organizou uma lista de compras antes de ir ao mercado ?
- As listas têm se mostrado um recurso bastante útil e eficiente no dia-a-dia das pessoas.
- Em computação a lista é uma das estruturas de dados mais empregadas no desenvolvimento de programas

Listas: Fundamentos

UNINASSAU
Campus Graças

- **Uma lista linear é uma coleção L:[a1, a2, ..., an], n≥0, cuja propriedade estrutural baseia-se na posição relativa dos elementos**
  - Se n=0, dizemos que a lista L é vazia;
  - caso contrário, são válidas as seguintes propriedades:
    - a1 é o primeiro elemento de L;
    - an é o último elemento de L;
    - ak, 1<k<n, é precedido pelo elemento ak-1 e seguido por ak+1 em L

Listas:operações

UNINASSAU
Campus Graças

- **Algumas operações realizadas sobre listas são:**
  - acessar um elemento qualquer da lista;
  - inserir um elemento numa posição específica da lista;
  - remover um elemento de uma posição específica da lista;
  - combinar duas listas em uma única;
  - particionar uma lista em duas;
  - determinar o total de elementos na lista;
  - ordenar os elementos da lista;
  - procurar um determinado elemento na lista;
  - apagar uma lista;
  - outras...

Listas:operações

UNINASSAU
Campus Graças

- **Considerando as operações de acesso, inserção e remoção, restritas aos extremos da lista, temos casos especiais que aparecem na modelagem de problemas resolvidos por computador:**
  - Pilha: inserção, remoção e acesso realizado em um único extremo. Denominadas listas LIFO (Last-In/Fist-Out)
  - Fila: inserções em um extremo e remoções e acessos no outro. Denominadas listas FIFO (First-In/First-Out)
  - Fila Dupla: inserções, remoções ou acessos são realizados em qualquer extremo. Denominadas DEQUE (Double-Ended QUEue)

Listas: Considerações sobre implementação

UNINASSAU
Campus Graças

- Ao codificar um programa que utiliza listas para armazenar dados, dificilmente iremos usar todas as operações que podem ser feitas com listas lineares.

- Na prática, apenas um subconjunto destas operações será necessário.

- As ferramentas de programação atuais, na sua maioria, já trazem implementações desse tipo de dado na forma de classes.

Listas: Considerações sobre implementação

UNINASSAU
Campus Graças

- É difícil obter uma implementação onde todas as operações sejam realizadas com a mesma eficiência.

- Ex: uma implementação que facilite o acesso a um elemento da lista, dificultará a inserção e a remoção de elementos no meio da mesma

Listas: Exemplo de Aplicação

UNINASSAU
Campus Graças

- Suponha que sejam dados os salários de todos os empregados de uma companhia e desejamos determinar quantos salários estão acima da média.
  - Uma vez que a média deve ser calculada em primeiro lugar, para que se possa comparar cada salário com a média, precisamos de um tipo de dados que possa conter todos os salários.
  - Escolhemos o tipo de dados lista.
  - Que operações serão executadas sobre a lista?

Listas: Exemplo de Aplicação

UNINASSAU
Campus Graças

- Precisamos poder executar as seguintes operações:
  - 1. Dado um salário, armazenar o salário na lista, na primeira posição desocupada.
  - 2. Acessar o salário em cada posição (para adicionar e comparar).

Listas: Implementações

UNINASSAU
Campus Graças

- Ao desenvolver uma implementação para listas lineares, o primeiro problema que surge é: como podemos armazenar os elementos da lista, dentro do computador?

- Podemos implementar uma lista através de vetor, ponteiros ou arquivo, dependendo do tamanho da lista e das operações que serão executadas sobre a lista.

Listas: Implementações

UNINASSAU
Campus Graças

- Vetor
  - Ocupa um espaço contínuo de memória.
  - Permite acesso randômico aos elementos.
  - Deve ser dimensionado com um número máximo de elementos.

4	12	45	7				

Listas: Implementações

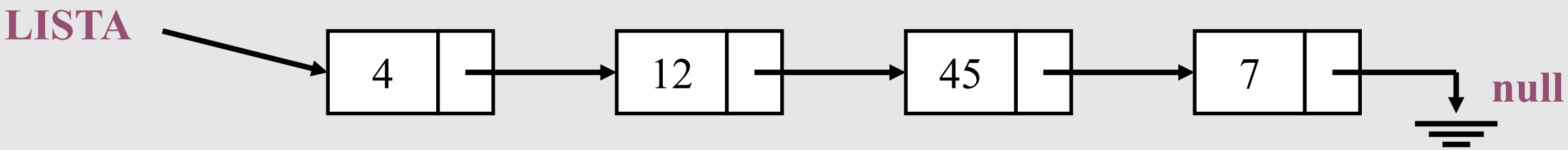
UNINASSAU
Campus Graças

- Estruturas de dados dinâmicas (ponteiros)
  - crescem (ou decrescem) à medida que elementos são inseridos (ou removidos)
  - Exemplo:
  - listas encadeadas:
    - amplamente usadas para implementar outras estruturas de dados

Listas: Implementações

UNINASSAU
Campus Graças

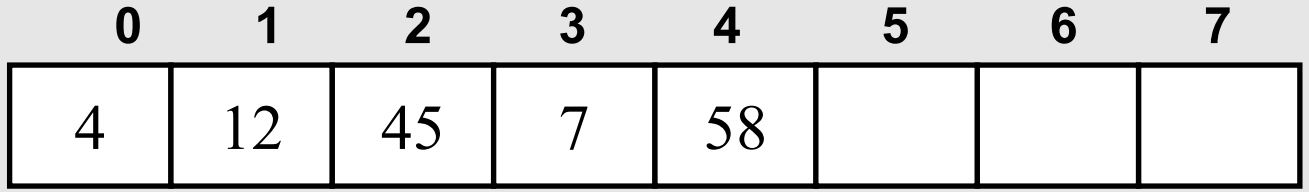
- Lista encadeada:
  - sequência encadeada de elementos, chamados de nós da lista
  - nó da lista é representado por dois campos:
    - a informação armazenada e
    - o ponteiro para o próximo elemento da lista
  - a lista é representada por um ponteiro para o primeiro nó
  - o ponteiro do último elemento é NULL



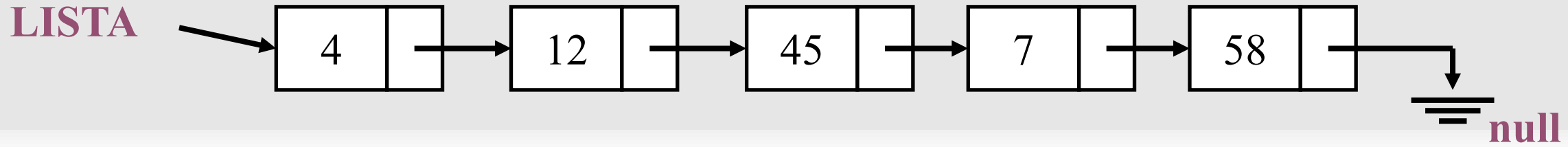
Listas: Implementações

UNINASSAU
Campus Graças

- Inserir no Fim:
  - Inserir no fim da lista o elemento “58”.
- Vetor:



- Ponteiro:

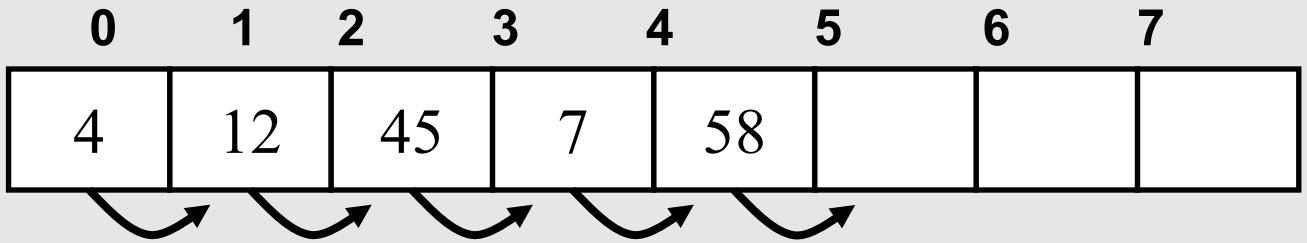


Listas: Implementações

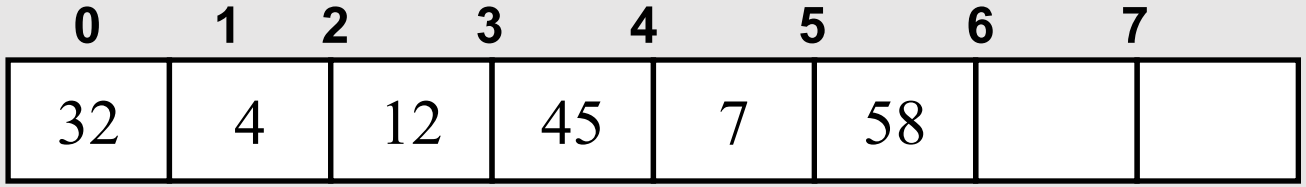
UNINASSAU
Campus Graças

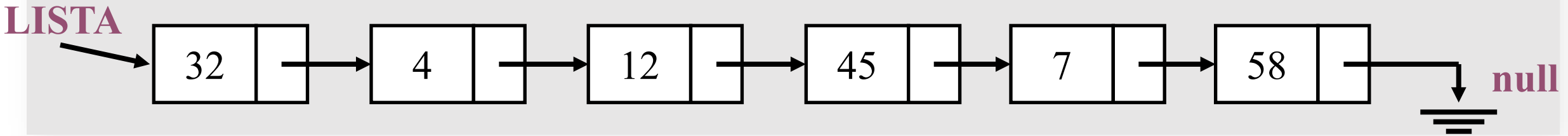
- Inserir no Início: Inserir no início da lista o elemento “32”.

Vetor:



Ponteiro:



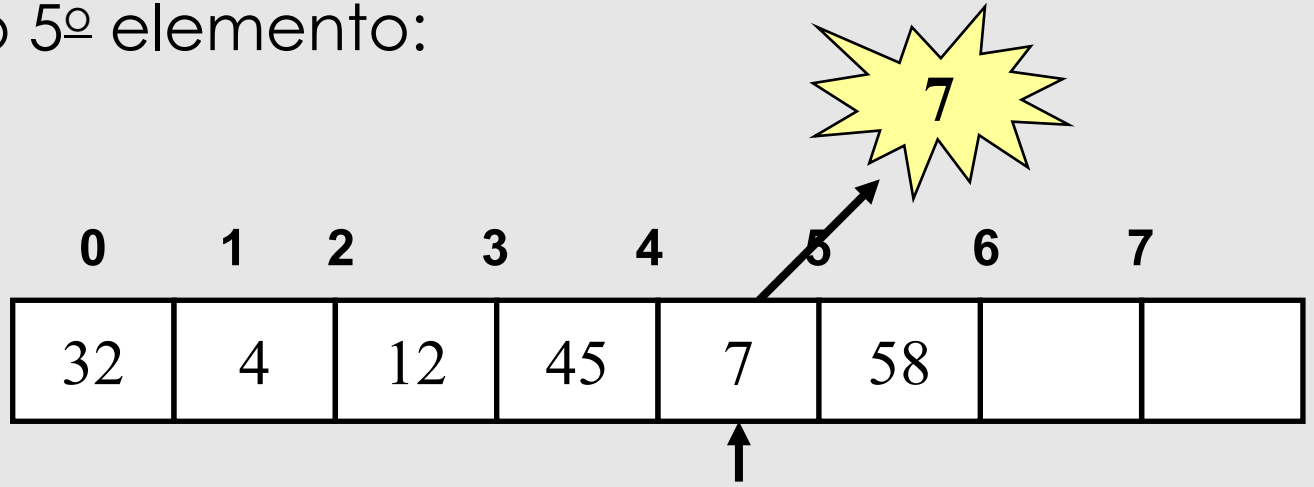


Listas: Implementações

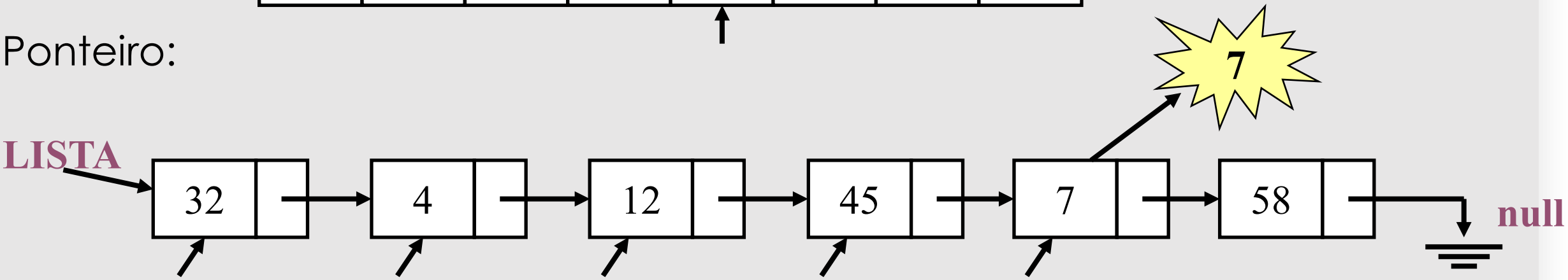
UNINASSAU
Campus Graças

- Consultar o 5º elemento:

Vetor:



Ponteiro:

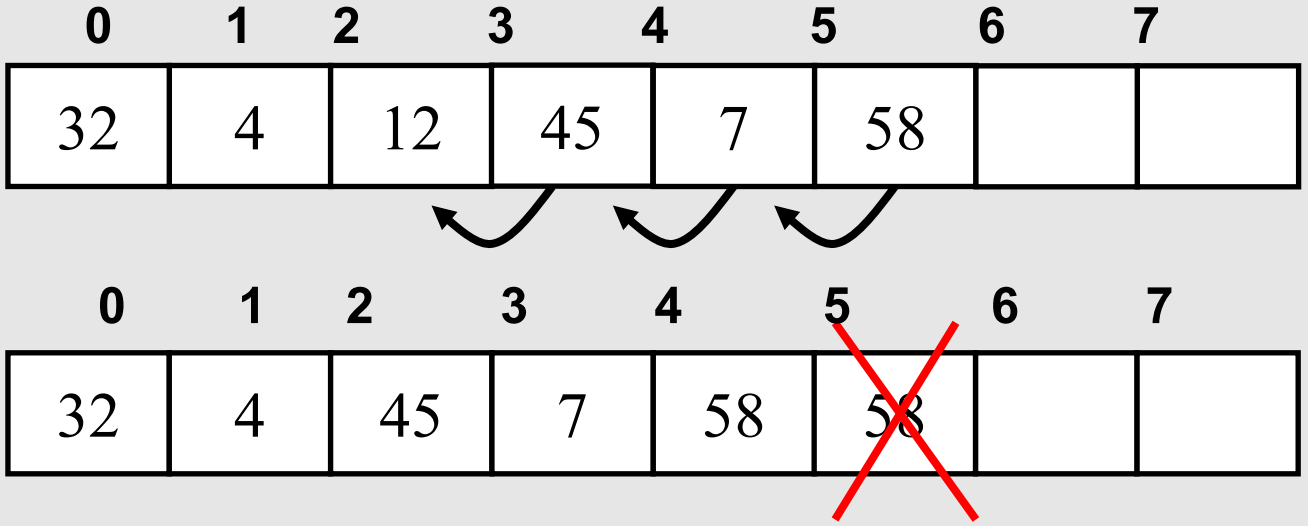


Listas: Implementações

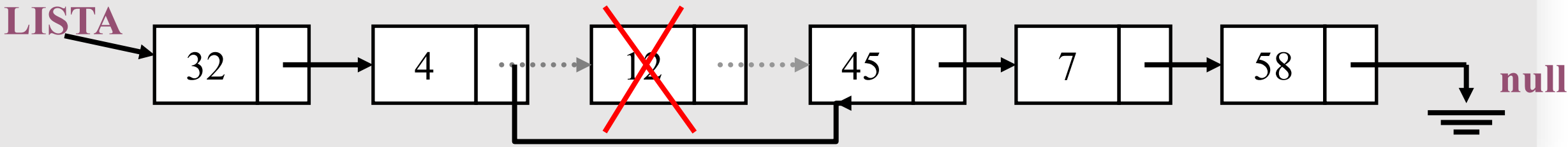
UNINASSAU
Campus Graças

- Retirar o 3º elemento:

Vetor:



Ponteiro:



UNINASSAU
Campus Graças

PILHA

Pilha

UNINASSAU
Campus Graças

- Uma pilha é uma lista linear em que a inserção ou eliminação de elementos somente pode ocorrer em uma das extremidades, que é chamada de TOPO da PILHA.

- Dada uma Pilha P = (a_1, a_2, ..., a_n), dizemos que a_1 é o elemento da base da pilha; a_n é o elemento do topo, e a $_{i+1}$ está acima de a_i na pilha.

- São também conhecidas como listas do tipo LIFO (*last in, first out*).

Pilha: Exemplo

UNINASSAU
Campus Graças

- **Ex: Pilhas de bandeja do Bandejão**

1) Bandejas inicialmente são empilhadas.

2) Pega-se (remove-se) bandeja do topo.

3) Se não há mais bandejas, a pilha está vazia e não podemos remover mais.

4) Desde que nós podemos ver a bandeja do topo, se ela estiver suja podemos não querer pegar (remover).

Pilha: Exemplo

UNINASSAU
Campus Graças

- **Este exemplo faz referência a 4 operações de pilhas:**

1) Inserir = push

2) Remover = pop

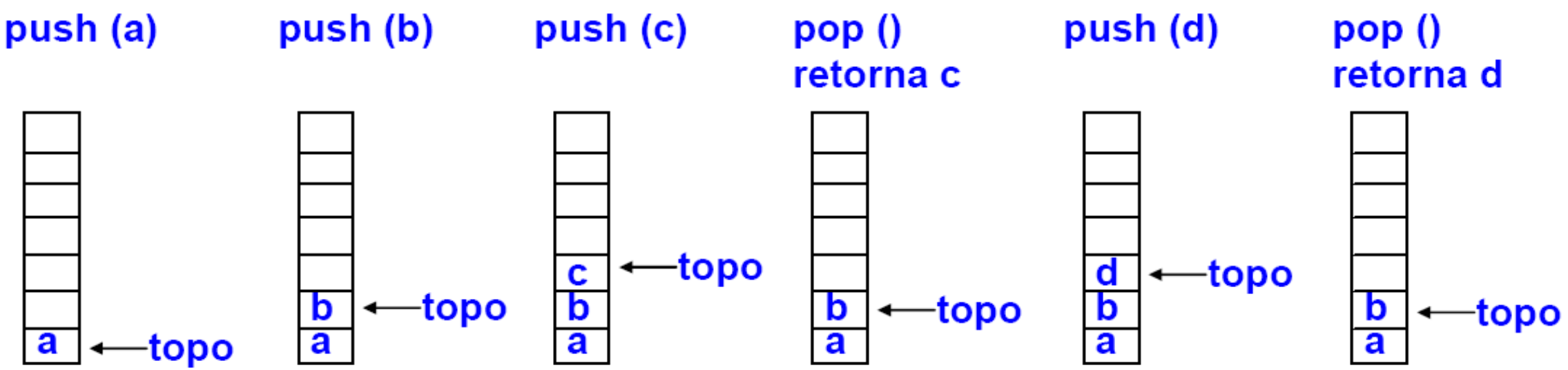
3) Se a pilha não contém elementos a verificação de vazia retorna VERDADEIRO, caso contrário FALSO.

4) Topo da pilha = top, retorna o topo da pilha (cópia) par ser examinado, sem removê-lo.

Pilha: resumo

UNINASSAU
Campus Graças

top - retorna o topo da pilha
push - insere novo elemento no topo da pilha
pop - remove o elemento do topo da pilha



Pilha: Considerações sobre implementação

UNINASSAU
Campus Graças

- O problema da implementação estática (usando vetor) é mais evidente quando existem movimentações de itens em operações de inserções/remoções.
- No caso das pilhas, essas movimentações não ocorrem!
- A alocação estática seria mais vantajosa na maioria das vezes.
- Em Java, temos a classe ***Stack*** em (java.util.Stack).

UNINASSAU
Campus Graças

FILA

Fila

UNINASSAU
Campus Graças

- Uma fila é uma lista linear em que a inserção e eliminação de elementos são feitas em extremos distintos.

- Dada uma Fila F = (a_1, a_2, ..., a_n), se dizemos que a_1 é o elemento do início da fila então a_n será o elemento do final da fila. Nesse caso, o elemento a_{i+1} estará posicionado após o elemento a_i na fila.

- São também conhecidas como listas do tipo FIFO (*first in, first out*).

Fila: Exemplo

UNINASSAU
Campus Graças

- Ex: Fila de impressão
  - 1) Os documentos são colocados em uma fila de impressão para servir a vários processos.
  - 2) Quem está no início da fila é processado primeiro, seguindo a ordem de chegada.
  - 3) Os documentos dão entrada no final da fila.
  - 4) Se não há mais documentos na fila ela está vazia e não podemos remover mais.

Fila: Operações

UNINASSAU
Campus Graças

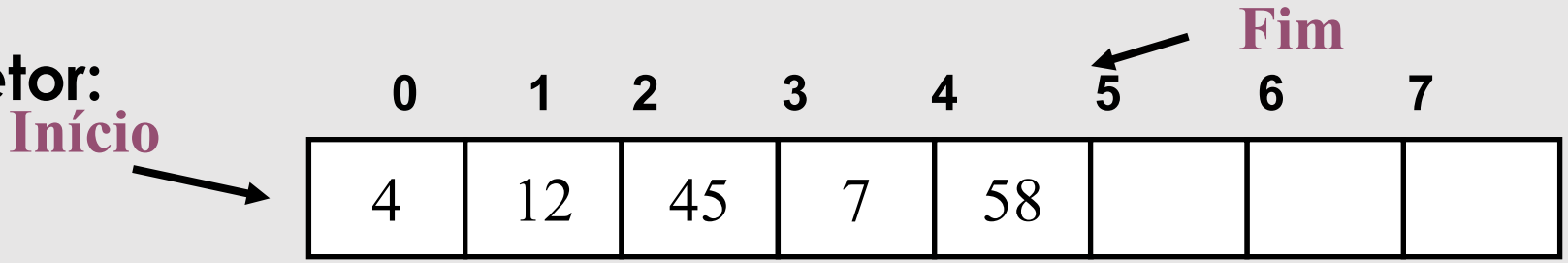
- **Operações básicas com Fila:**
  1) Entra na Fila;
  2) Sai da Fila;
  3) Consulta o elemento do início da Fila.
  4) Fila vazia (Underflow)
  5) Fila cheia (Overflow)

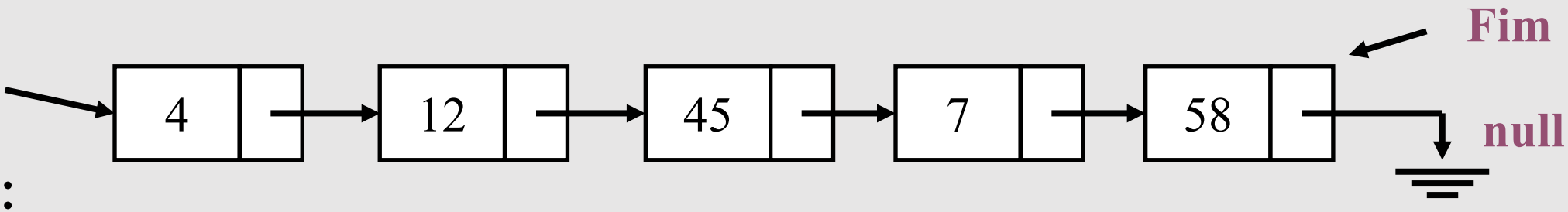
Fila: resumo

UNINASSAU
Campus Graças

- **Inserir no fim, retirar do início:**

- **Vetor:**





- **Ponteiro:**

Fila: Considerações sobre implementação

UNINASSAU
Campus Graças

- Temos problema de implementação estática (usando vetor), pois quando removemos um elemento o controle será maior.

- A implementação dinâmica tem um custo caro para a inserção no final.

- No caso de implementação dinâmica podemos usar uma referência extra para o elemento do final da fila.

UNINASSAU
Campus Graças

ÁRVORES

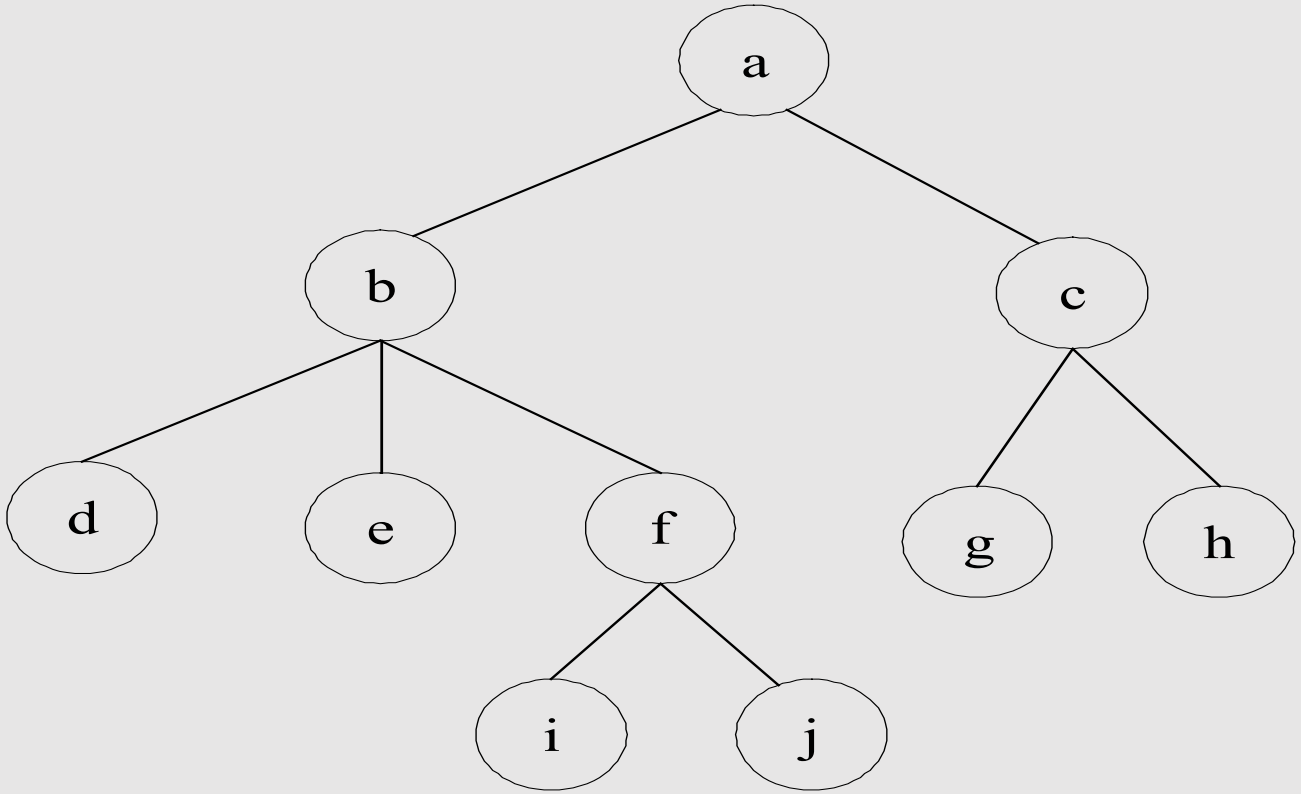
Árvore

UNINASSAU
Campus Graças

- Estrutura de dados bidimensional na qual os elementos são conectados formando uma estrutura semelhante a uma árvore;
- Diferentemente das outras estruturas vistas anteriormente, a árvore é uma estrutura de dados não linear.
- São apropriadas para representar estruturas hierárquicas de objetos.

Exemplo de uma árvore

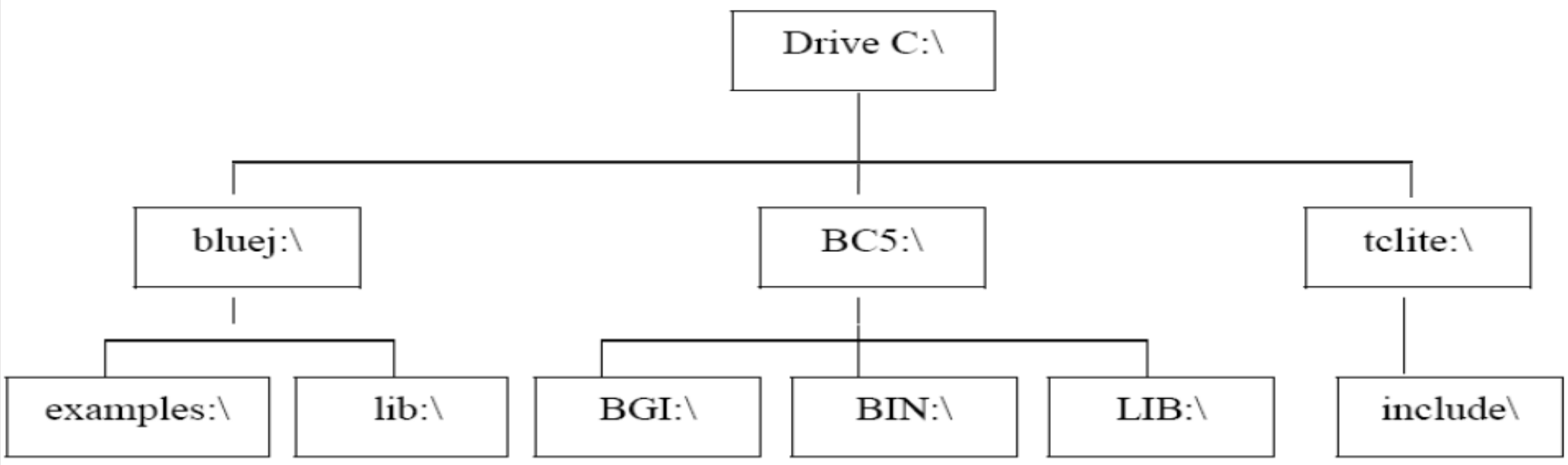
UNINASSAU
Campus Graças



Aplicações

UNINASSAU
Campus Graças

- Representar o sistema de arquivos de um computador:



Aplicações

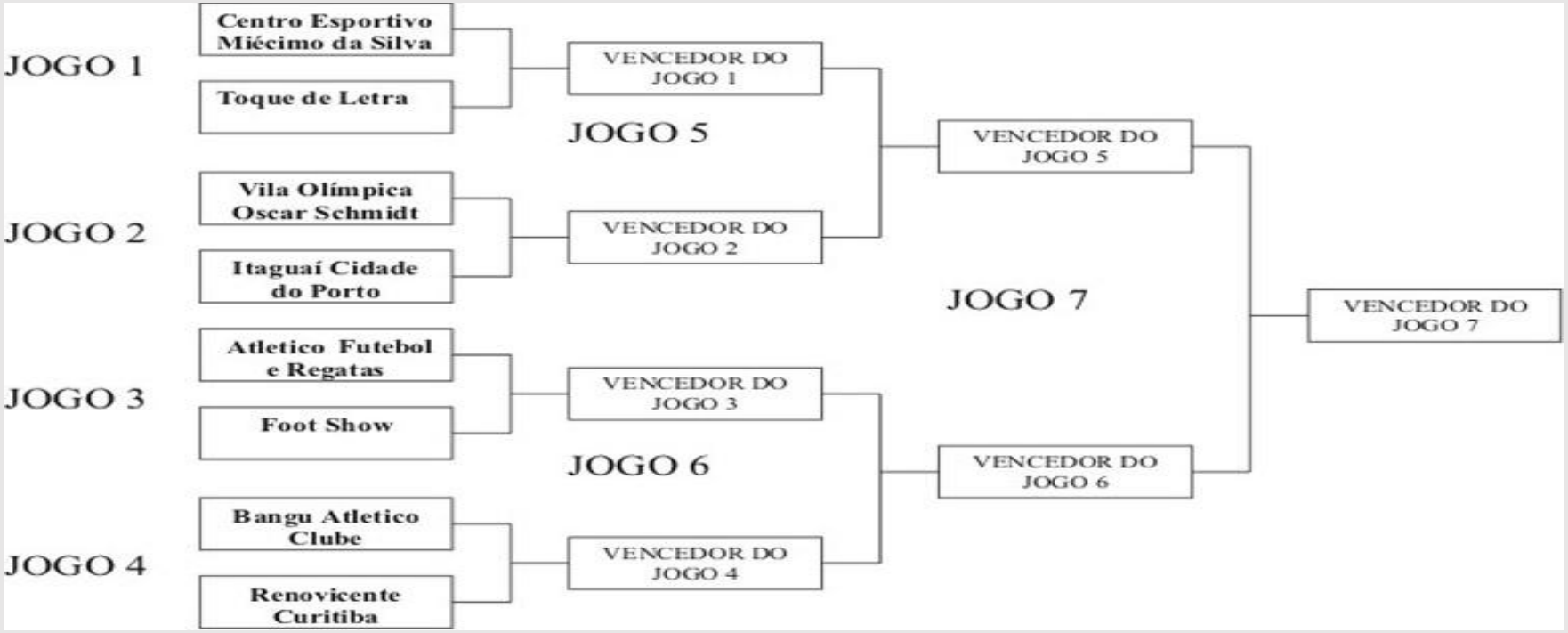
UNINASSAU
Campus Graças

- Representar a estrutura organizacional em uma empresa:

Aplicações

UNINASSAU
Campus Graças

- Definir a sequência de jogos em uma competição:



Aplicações

UNINASSAU
Campus Graças

- Conjunto de menus em um aplicativo ou site:

Árvore

UNINASSAU
Campus Graças

- Um conjunto de nós, tais que:
  - Um nó se localiza no topo da árvore, sendo chamado de nó raiz;
  - Os nós restantes estão conectados direta ou indiretamente ao nó raiz, formando novas árvores (subárvores);
  - Os nós podem significar qualquer tipo de informação, tais como números e nomes, ou objetos, como clientes, passagens aéreas, etc.

Definição:

UNINASSAU
Campus Graças

- Árvore

– um conjunto de nós tal que

  - existe um nó **r**, denominado ***raiz***, com zero ou mais sub-árvores, cujas raízes estão ligadas a **r**
  - os nós raízes destas sub-árvores são os ***filhos*** de **r**
  - os ***nós internos*** da árvore são os nós com filhos
  - as ***folhas*** ou ***nós externos*** da árvore são os nós sem filhos

Definições

UNINASSAU
Campus Graças

- Nó ancestral:
  - Refere-se a um nó que possui uma ou mais subárvores conectadas direta ou indiretamente a ele. O nó raiz é o ancestral de todos os nós da árvore;
- Nó folha:
  - Refere-se a um nó que não possui filhos;
- Nível de um nó:
  - A raiz está no nível 0, e o nível de qualquer outro nó é o nível do seu pai + 1;

Definições

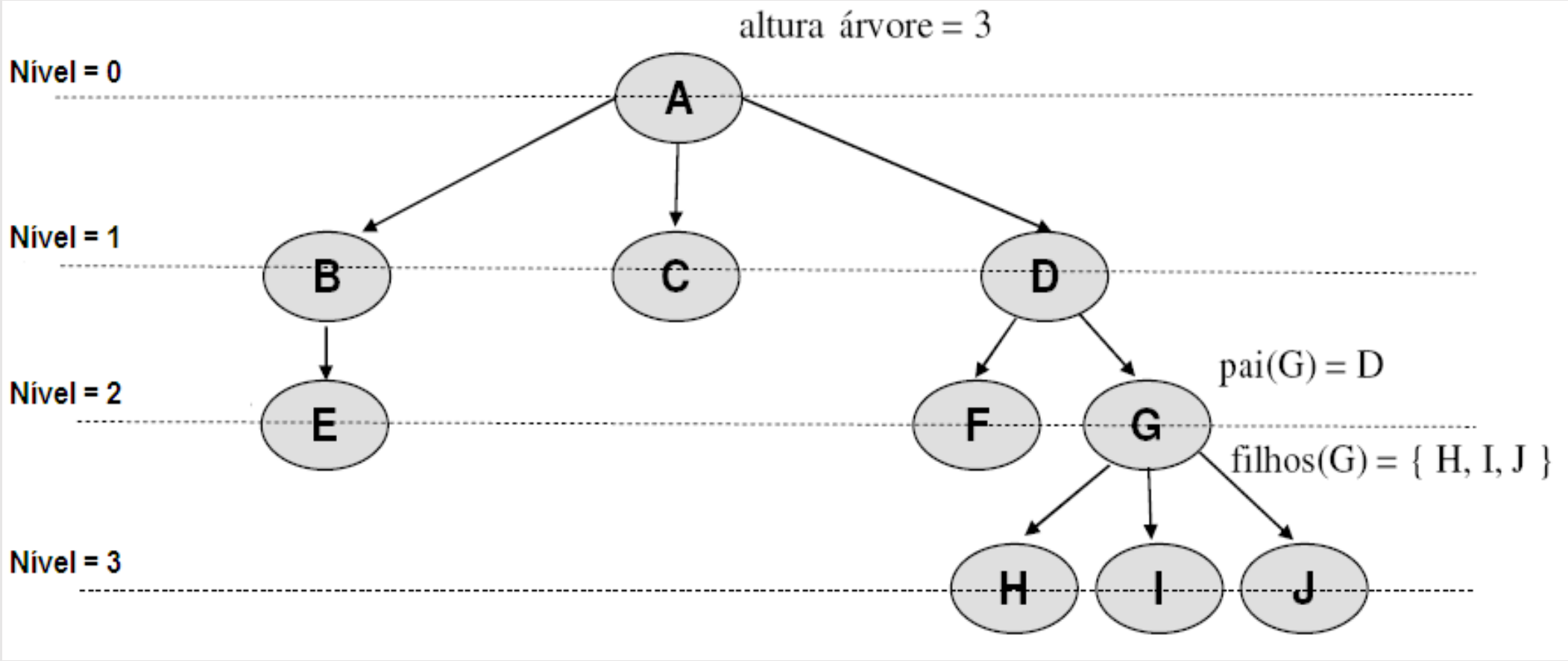
UNINASSAU
Campus Graças

- Profundidade (altura) de uma árvore:
  - Corresponde ao nível máximo de qualquer folha da árvore, ou seja, é o tamanho do percurso mais distante da raiz até uma folha;

- Árvore não balanceada:
  - Refere-se a uma árvore que tem mais nós de um lado do nó raiz quando comparado com o outro lado;

Exemplo

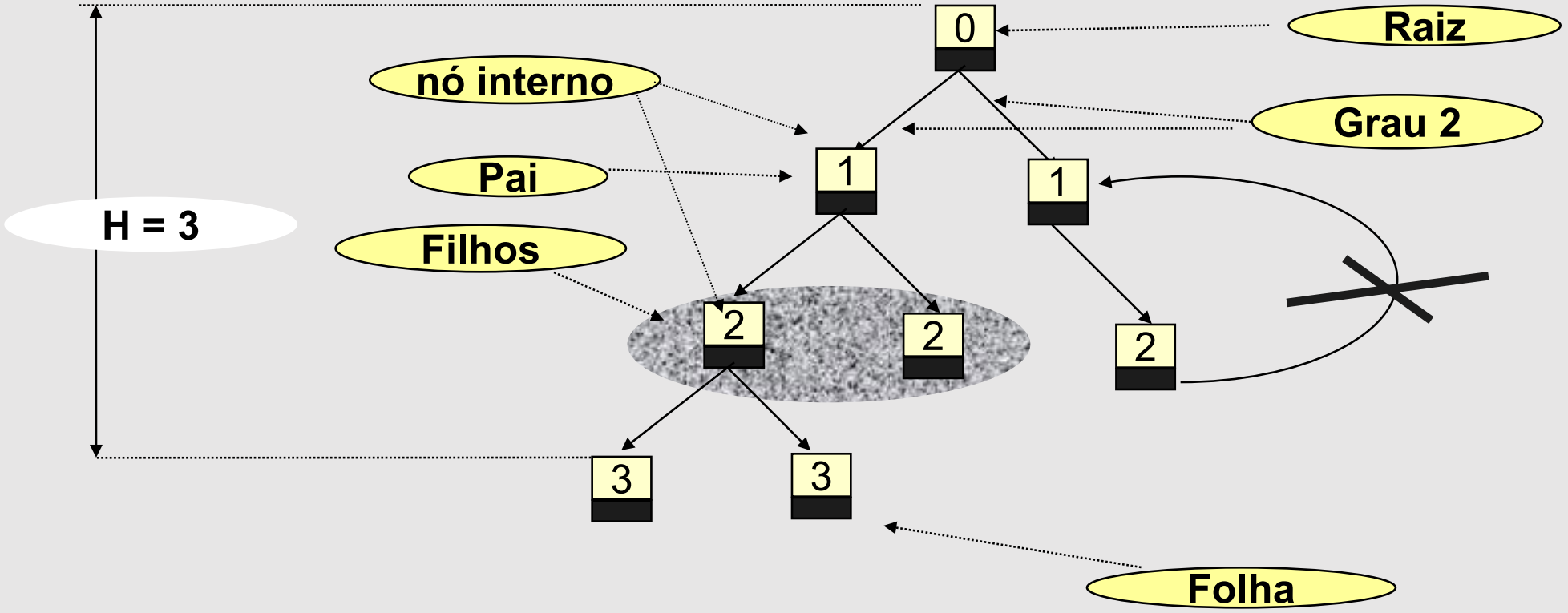
UNINASSAU
Campus Graças



Árvores

UNINASSAU
Campus Graças

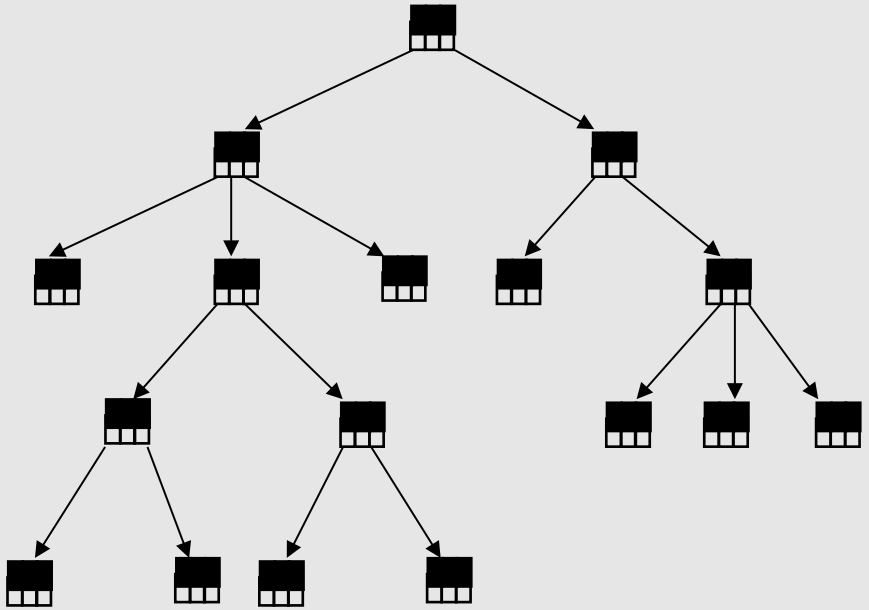
Estruturas Hierárquicas

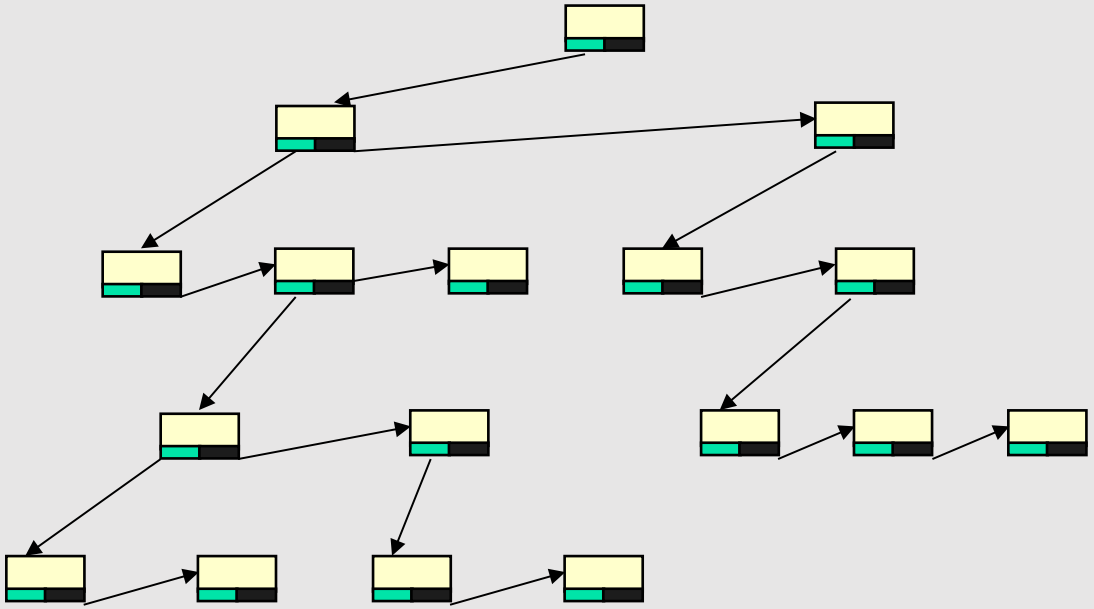


Árvores

UNINASSAU
Campus Graças

Representação de Árvores: Explícita





Classificação

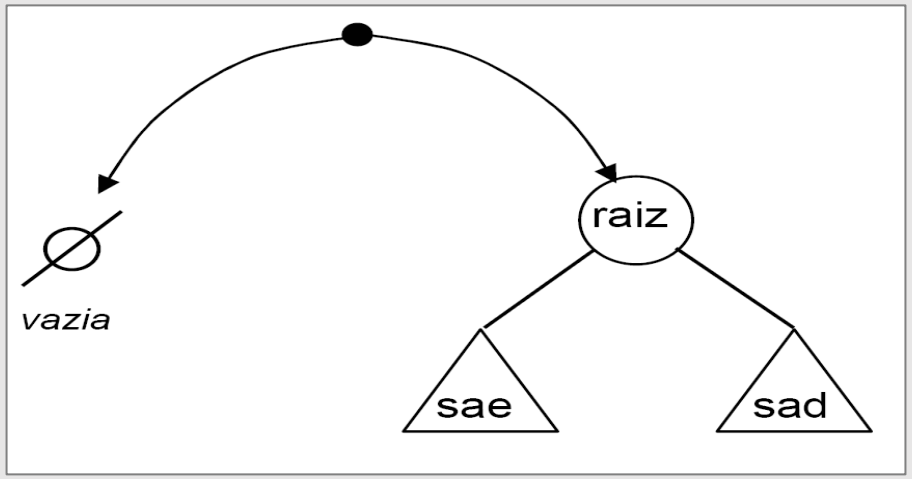
UNINASSAU
Campus Graças

- Podemos classificar as árvores de acordo com o número de filhos que cada nó pode ter:
  - Árvores binárias: Cada nó pode ter no máximo 2 filhos;
  - Arvores genéricas: Cada nó pode conter um número indeterminado de filhos.

Árvores Binárias

UNINASSAU
Campus Graças

- Uma árvore em que cada nó tem zero, um ou dois filhos
- Uma árvore binária é:
  - uma árvore vazia; ou
  - um nó raiz com duas sub-árvores:
    - a sub-árvore da direita (sad)
    - a sub-árvore da esquerda (sae)



Árvores binárias de busca

UNINASSAU
Campus Graças

- Ao inserir nós em uma árvore binária, devemos obedecer as seguintes regras:
  - O nó filho da esquerda tem valor menor que o nó pai;
  - O nó filho da direita tem valor igual ou superior ao nó pai.

- Se a árvore binária é completa (todos os nós, exceto os nós folhas, possuem dois filhos), dizemos que a mesma é estritamente binária e a quantidade de nós para uma árvore com n folhas é dada por:
  - Quantidade de nós = 2*n - 1.

Árvore binária de busca

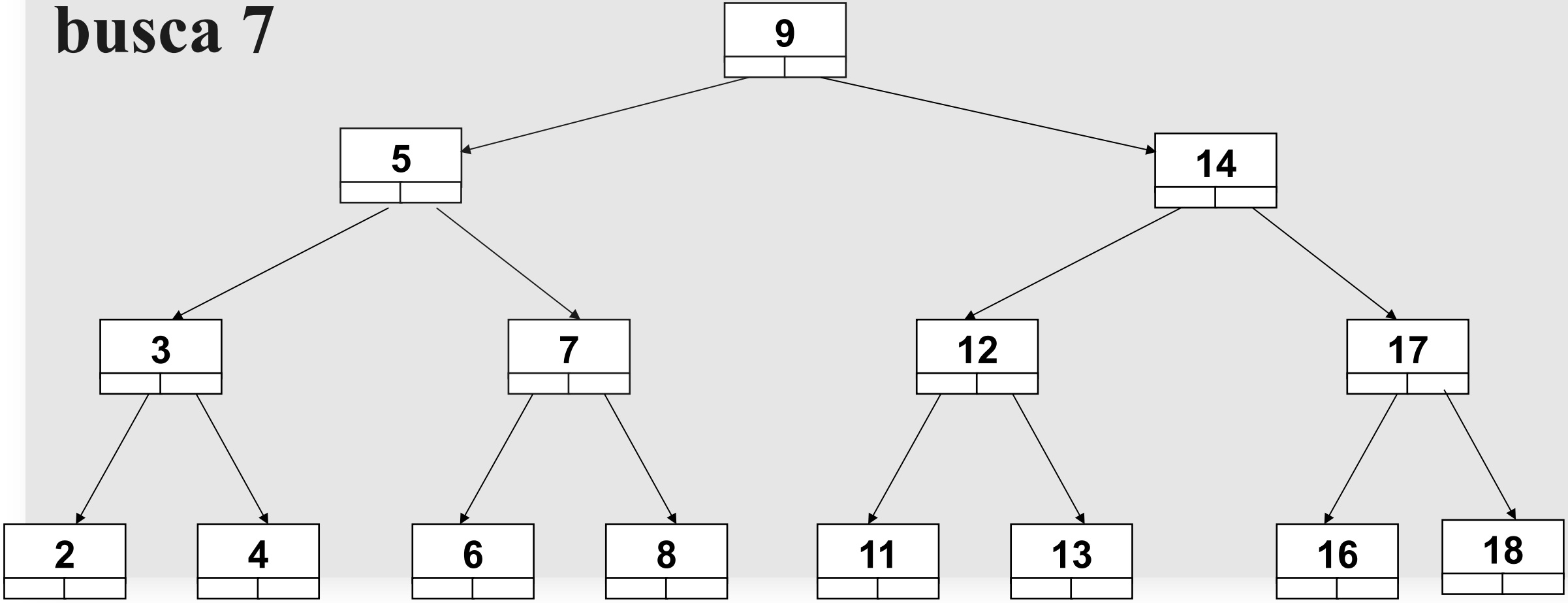
UNINASSAU
Campus Graças

- TAD que manipulam operações chamados **dicionários**.
- Árvores binárias de busca implementam dicionários eficientemente.
- Cada nó da árvore é um registro contendo:
  - **chave**
  - **esquerda** e **direita** são **ponteiros** para outros nós (ou *null*)
  - chave dos filhos **esquerdos** são **menores** que a chave do nó pai
  - chave dos filhos **direitos** são **maiores** que a chave do nó pai

Árvore binária de busca

UNINASSAU
Campus Graças

busca 7

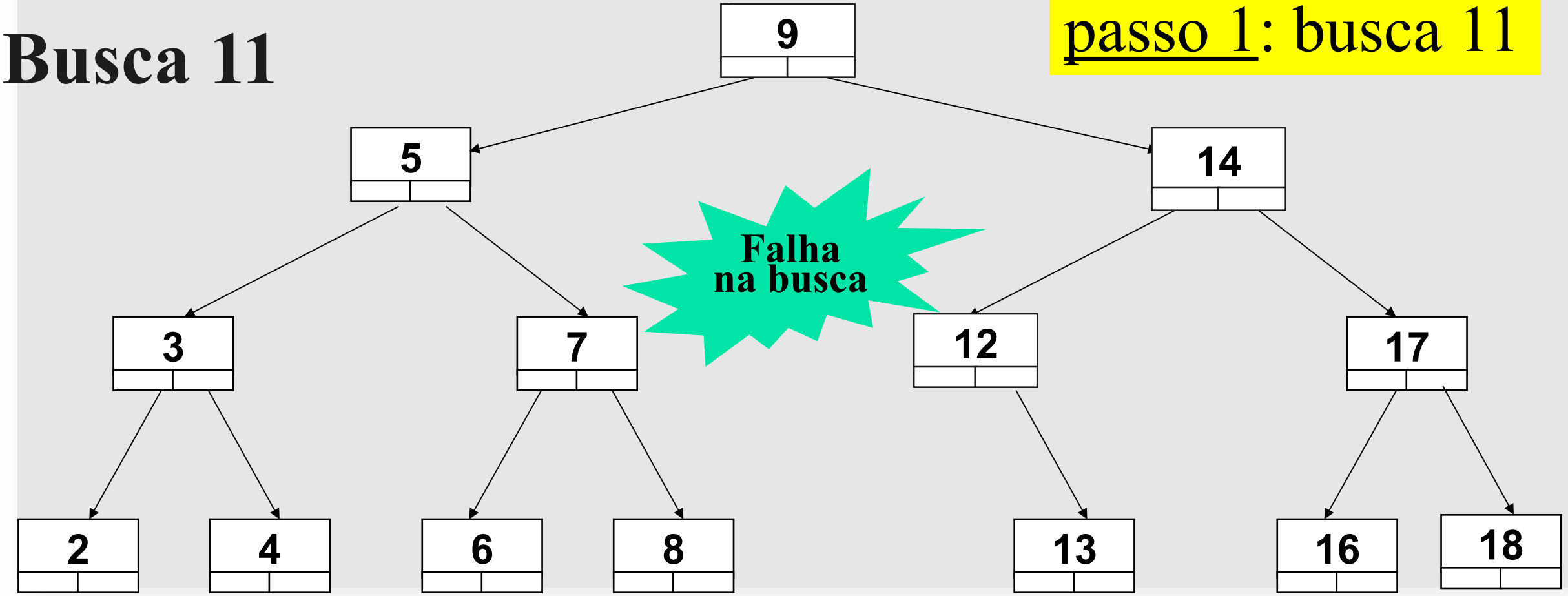


Árvore binária de busca

UNINASSAU
Campus Gracas

Busca 11

passo 1: busca 11



Percurso (recursivo)

UNINASSAU
Campus Graças

Árvore Binária, Percurso em Pré-Ordem

```
< 1>  ALGORITMO PercursoPréOrdem-Recursivo (PtrTrb)
< 2>
< 3>  PtrTrb {Ponteiro de trabalho para o percurso}
< 4>
< 5>  INÍCIO
< 6>     SE PtrTrb <> #
< 7>     ENTÃO
< 8>        INÍCIO
< 9>           UTILIZA (PtrTrb)
<10>           PercursoPréOrdem-Recursivo (LinkEsq (PtrTrb))
<11>           PercursoPréOrdem-Recursivo (LinkDir (PtrTrb))
<12>        FIM
<13>  FIM
```

Percurso (recursivo)

UNINASSAU
Campus Graças

Árvore Binária, Percurso em Ordem Simétrica

```
< 1>  ALGORITMO PercursoOrdemSimétrica-Recursivo (PtrTrb)
< 2>
< 3>  PtrTrb {Ponteiro de trabalho para o percurso}
< 4>
< 5>  INÍCIO
< 6>     SE PtrTrb <> #
< 7>     ENTÃO
< 8>        INÍCIO
< 9>           PercursoOrdemSimétrica-Recursivo (LinkEsq (PtrTrb))
<10>           UTILIZA (PtrTrb)
<11>           PercursoOrdemSimétrica-Recursivo (LinkDir (PtrTrb))
<12>        FIM
<13>  FIM
```

Percurso (recursivo)

UNINASSAU
Campus Graças

Árvore Binária, Percurso em Ordem Final

```
< 1>  ALGORITMO PercursoOrdemFinal-Recursivo (PtrTrb)
< 2>
< 3>  PtrTrb (Ponteiro de trabalho para o percurso)
< 4>
< 5>  INÍCIO
< 6>     SE PtrTrb <> #
< 7>     ENTÃO
< 8>        INÍCIO
< 9>           PercursoOrdemFinal-Recursivo (LinkEsq (PtrTrb))
<10>           PercursoOrdemFinal-Recursivo (LinkDir (PtrTrb))
<11>           UTILIZA (PtrTrb)
<12>        FIM
<13>  FIM
```

Obrigado

E- mail:adilson.silva@sereducacional.com

@prof.Adilson.silva