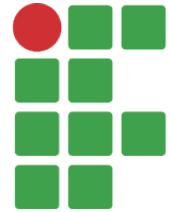


Estruturas de Dados

Eng. da Computação



**INSTITUTO
FEDERAL**

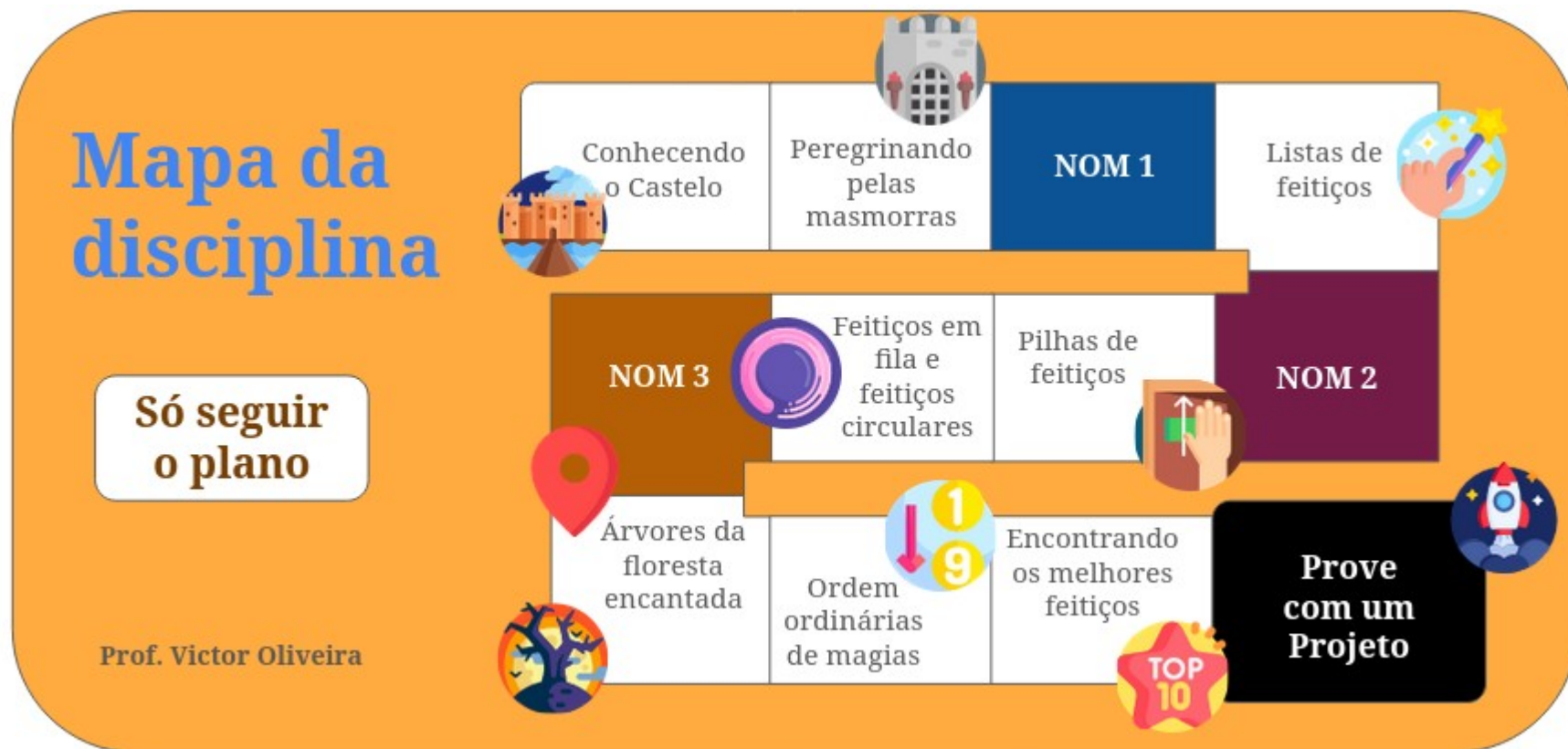
Paraíba

Campus
Campina Grande

Árvores Binárias de Busca

Prof. Victor André Pinho de Oliveira

Mapa da disciplina



Conceitos

Árvore



Conceitos

Árvore



Conceitos

Árvore



Conceitos

Árvore

Uma estrutura de dados não-linear que armazena elementos de um mesmo tipo.

Conceitos

Árvore

Uma estrutura de dados **não-linear** que armazena elementos de um **mesmo tipo**.

Os dados são armazenados em entidades denominadas **nó**.

Cada **nó** pode **ramificar** para outros **nós** – **subárvores**.

Conceitos

Árvore

Cada **nó** pode **ramificar** para outros **nós** – **subárvores**:

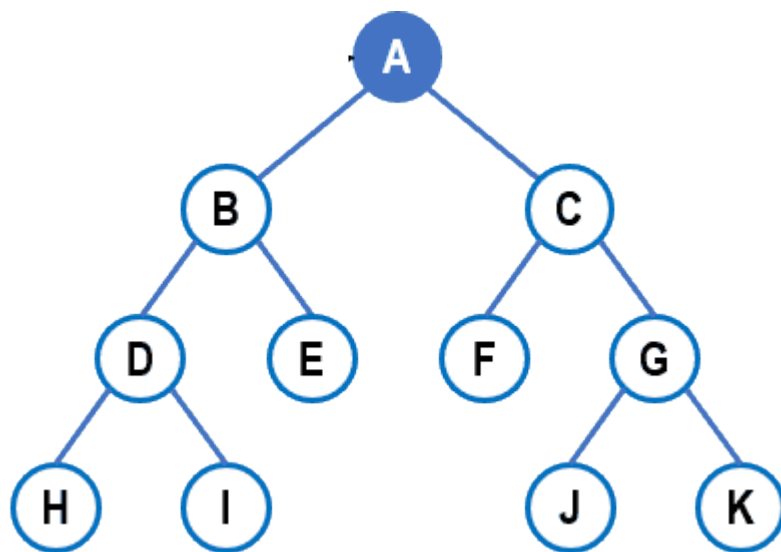
O primeiro **nó** da árvore é chamado **raiz**.

Se um **nó** ramifica, dizemos que ele tem **filhos**.

Se um **nó** não tem filhos, ele é chamado de **folha**.

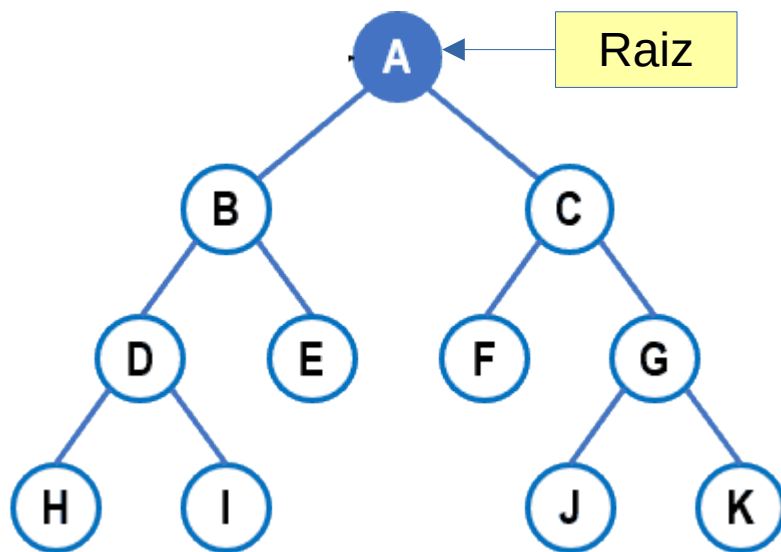
Conceitos

Árvore



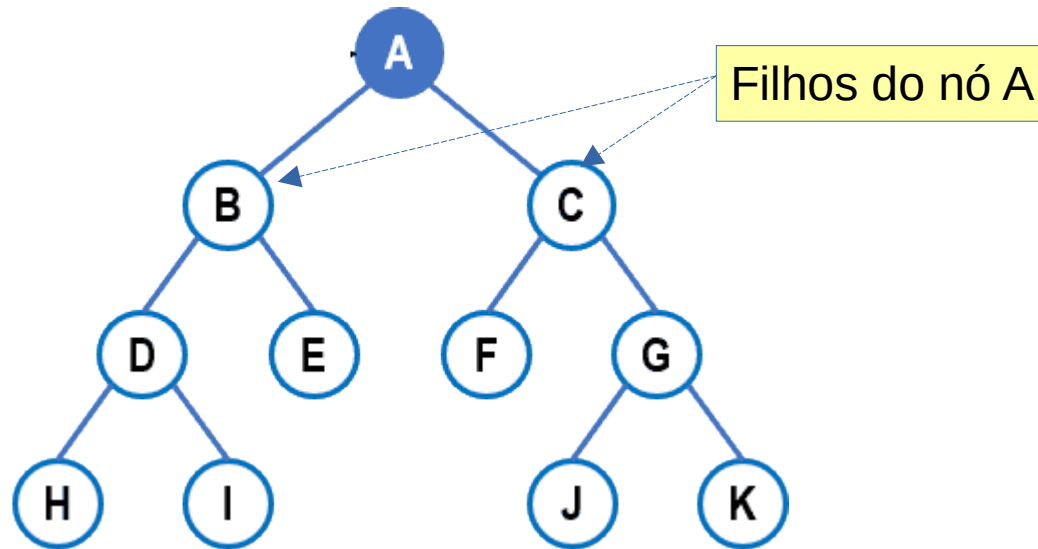
Conceitos

Árvore



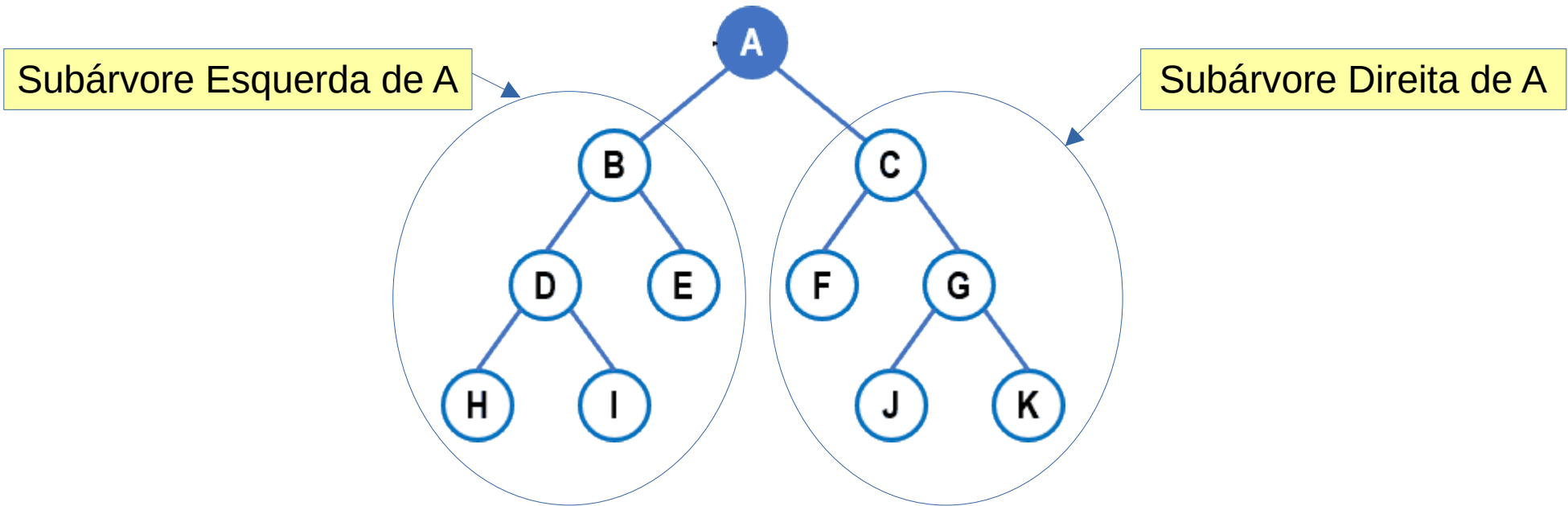
Conceitos

Árvore



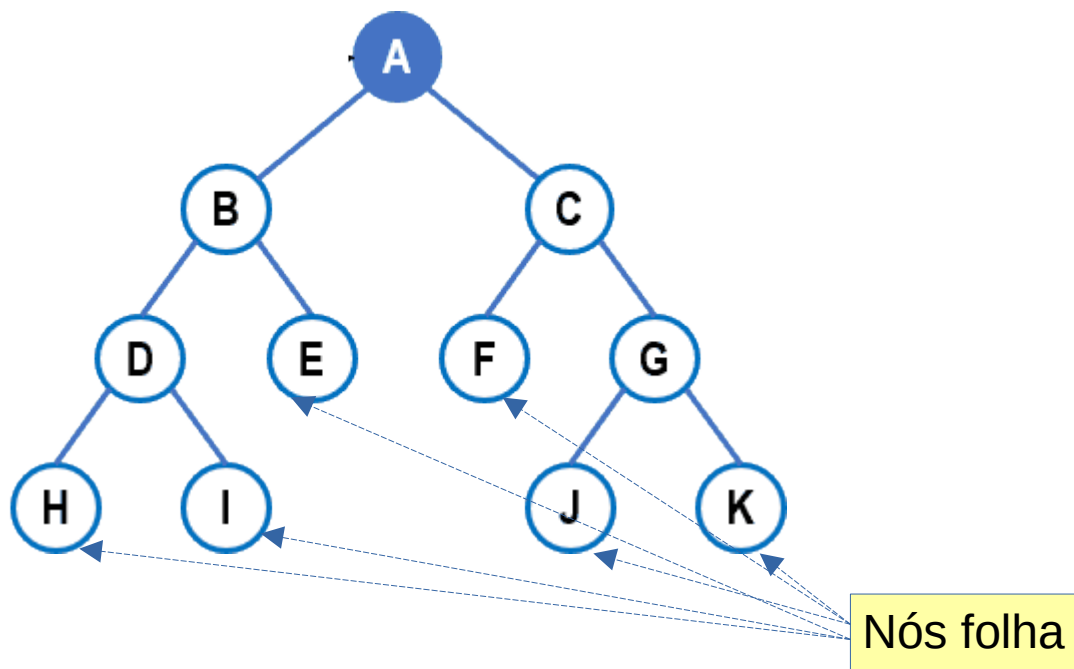
Conceitos

Árvore



Conceitos

Árvore



Conceitos

Árvore Binária

Uma árvore é dita **binária** quando cada nó possui, no máximo, dois filhos.

Conceitos

Árvore Binária de Busca (ABB)

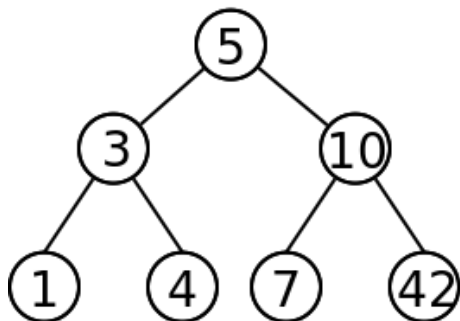
Uma árvore binária é dita **ordenada** (ou **de busca** ou **de pesquisa**) quando, para cada nó:

- os filhos da subárvore esquerda são todos menores que ele;
- os filhos da subárvore direita são todos maiores que ele.

Conceitos

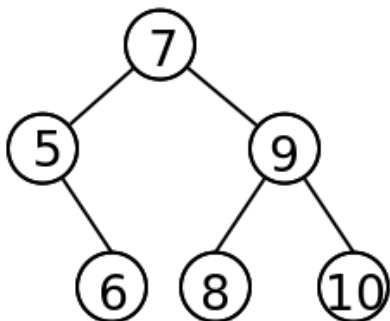
Árvore Binária de Busca (ABB)

Árvore 1



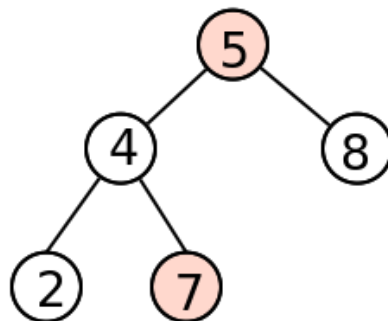
Árvore Binária
de Pesquisa

Árvore 2



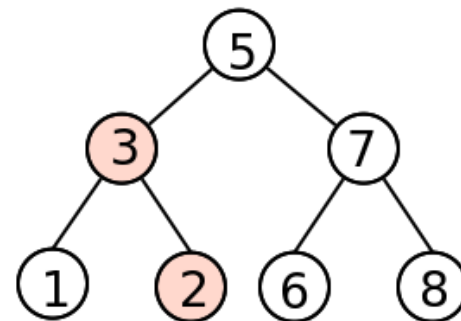
Árvore Binária
de Pesquisa

Árvore 3



Não é Árvore
Binária de Pesquisa

Árvore 4



Não é Árvore
Binária de Pesquisa

Conceitos

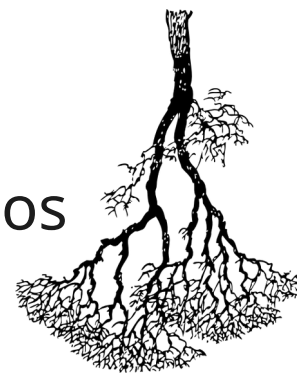
Operações em ABB:

- **Base**
 - Inserir
 - Remover
- **Consulta**
 - EmOrdem
 - PreOrdem
 - PosOrdem
- **Auxiliares**
 - Buscar
 - Obter
 - Apagar

Conceitos

Aplicações de Árvores :

- Estruturas de pastas;
- HTML → Web scraping;
- Interfaces gráficas → componentes
- Estrutura hierárquica no geral;
- **ABB** → Armazenamento e rápida recuperação de dados
- Etc.



Implementando Árvores Binárias de Busca

... de inteiros

Base:

```
struct sNODE{  
    int dado;  
    struct sNODE *esq, *dir;  
};  
  
struct sNODE *raiz = NULL;  
  
struct sNODE *inserir(struct sNODE *no, int dado);  
struct sNODE *remover(struct sNODE *no, int dado);  
  
void emOrdem(struct sNODE *no);  
void preOrdem(struct sNODE *no);  
void posOrdem(struct sNODE *no);  
  
struct sNODE *buscar(struct sNODE *no, int dado);  
int obter(struct sNODE *no);  
  
struct sNODE *apagar(struct sNODE *no);
```

Implementando Árvores Binárias de Busca

... de inteiros

Explicando:

```
struct sNODE { int dado; struct sNODE *esq, *dir; };
```

//struct sNODE → registro para o nó que guarda dado

//dado → armazena o dado

//esq, dir → ponteiro para o filho esquerdo e direito,
respectivamente

```
struct sNODE *raiz = NULL; // ponteiro para a raiz da árvore
```

Implementando Árvores Binárias de Busca

... de inteiros

Mãos à obra:

```
struct sNODE *inserir(struct sNODE *no, int dado);
```

»»»»» Função deve inserir o dado preservando atributos da ABB.

Implementando Árvores Binárias de Busca

... de inteiros

Mãos à obra:

```
struct sNODE *remover(struct sNODE *no, int dado);
```

»»»»» Função deve remover o dado preservando atributos da ABB.

Implementando Árvores Binárias de Busca

... de inteiros

Mãos à obra:

```
void emOrdem(struct sNODE *no);
```

»»»»» Função deve visitar (printar) toda a subárvore esquerda, seguida do nó. Depois, deve visitar toda a subárvore direita.

Implementando Árvores Binárias de Busca

... de inteiros

Mãos à obra:

```
void preOrdem(struct sNODE *no);
```

»»»»» Função deve visitar (printar) primeiro o nó, depois toda a subárvore esquerda, depois toda a subárvore direita.

Implementando Árvores Binárias de Busca

... de inteiros

Mãos à obra:

```
void posOrdem(struct sNODE *no);
```

»»»»» Função deve visitar (printar) a subárvore esquerda, depois a subárvore direita, e só depois o nó.

Implementando Árvores Binárias de Busca

... de inteiros

Mãos à obra:

```
struct sNODE *buscar(struct sNODE *no, int dado);
```

»»»»» Função buscar e retornar o nó que contém o dado.

Implementando Árvores Binárias de Busca

... de inteiros

Mãos à obra:

```
struct sNODE *apagar(struct sNODE *no);
```

»»»»» Função deve apagar a árvore.

