



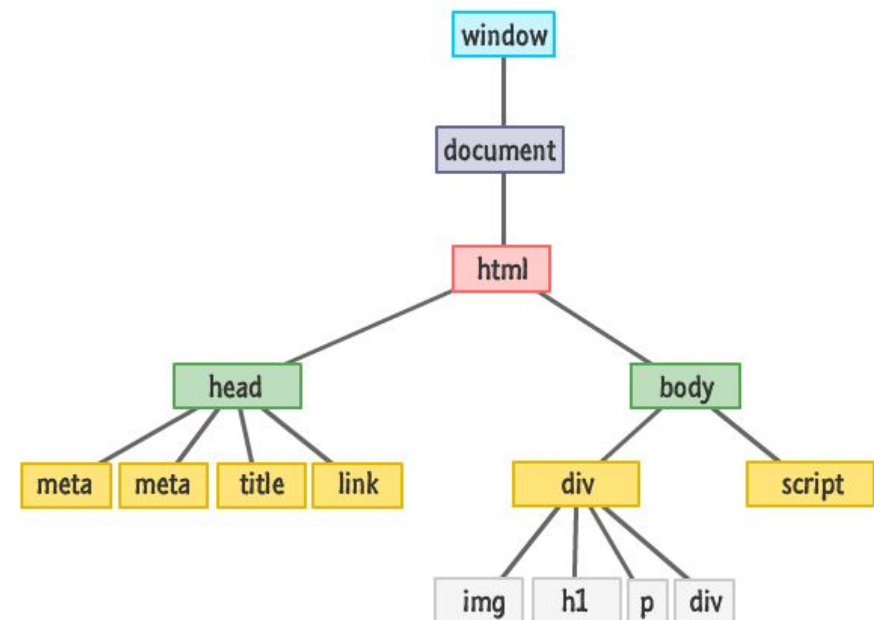
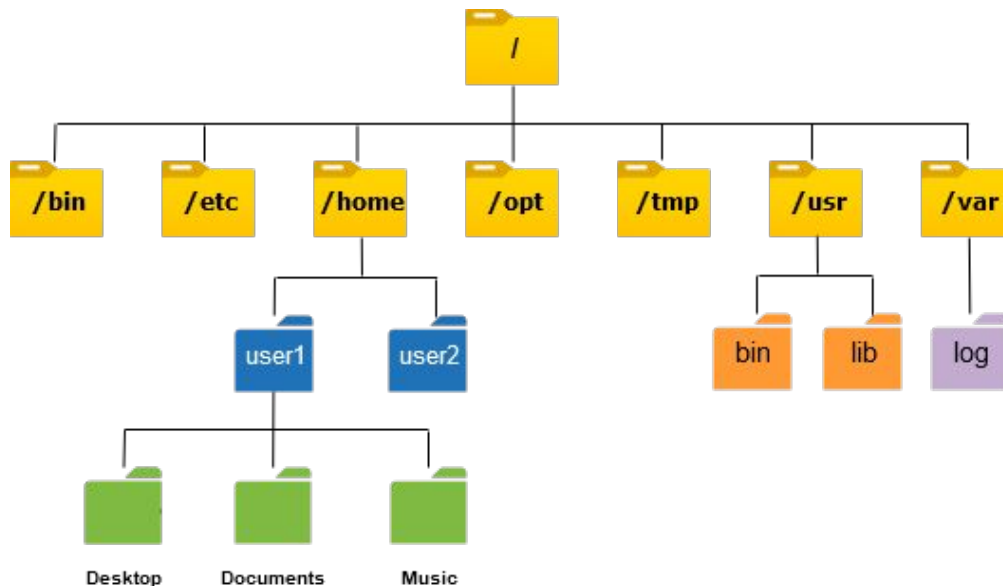
ESTRUTURA DE DADOS

Aula 13 – Árvores I

Prof. Rodrigo Maciel

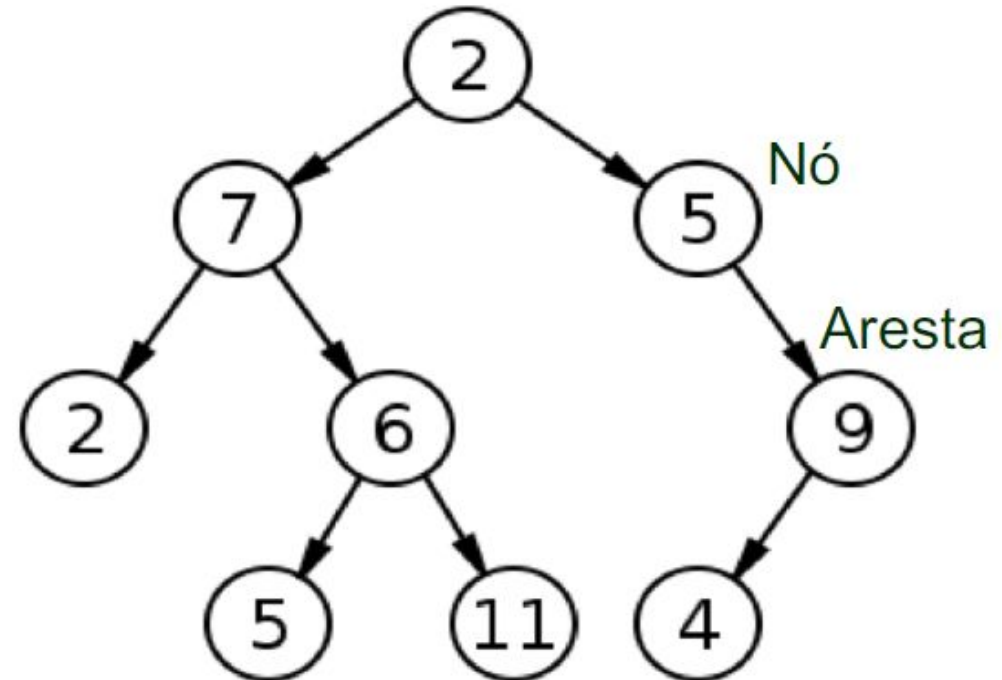
ÁRVORES

- Uma árvore combina as vantagens de duas estruturas: um array ordenado e uma lista encadeada;
- Busca rápida (como em um array ordenado);
- Inserção e eliminação rápida (como em uma lista encadeada).



ÁRVORE BINÁRIA

- Uma árvore consiste em nós (círculos) conectados por arestas (linhas);
- Máximo dois filhos;
- Filho à esquerda e filho à direita;
- Pode ter um ou nenhum filho.



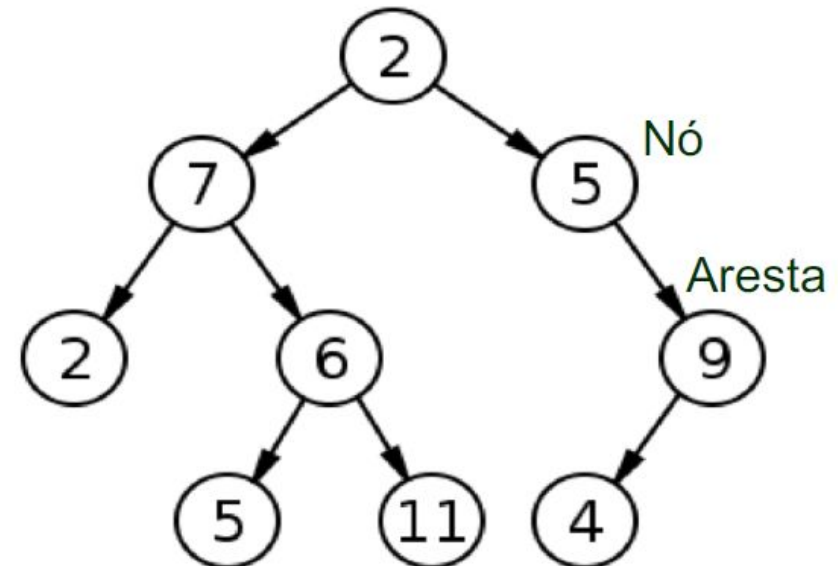
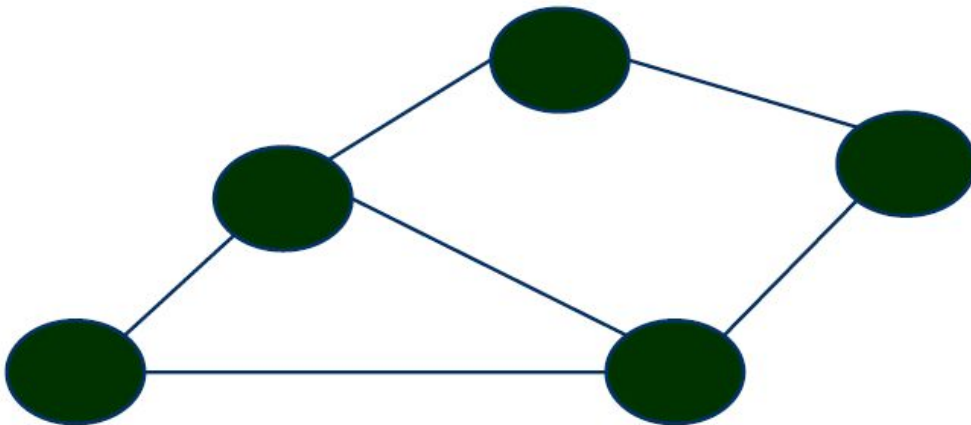
TERMINOLOGIA

- **Caminho**

- Caminho que liga um nó até outro nó.

- **Raiz**

- É o nó na parte superior. Há apenas uma raiz em uma árvore e deve haver somente um caminho da raiz até qualquer outro nó.



TERMINOLOGIA

- **Pai**

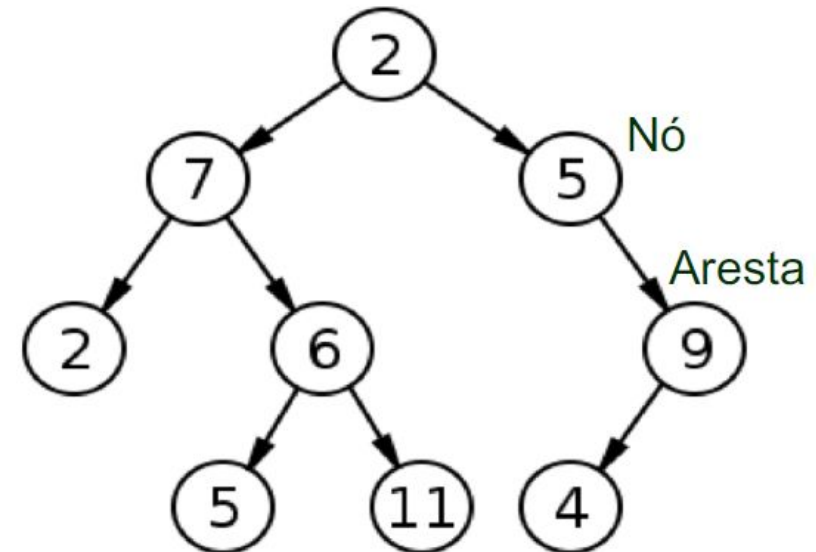
- Qualquer nó (exceto a raiz) tem exatamente uma aresta que sobe para outro nó. O nó acima dele é chamado de pai do nó.

- **Filho**

- Qualquer nó pode ter uma ou mais linhas descendo para outros nós. Esses nós abaixo de um dado nó são chamados de seus filhos.

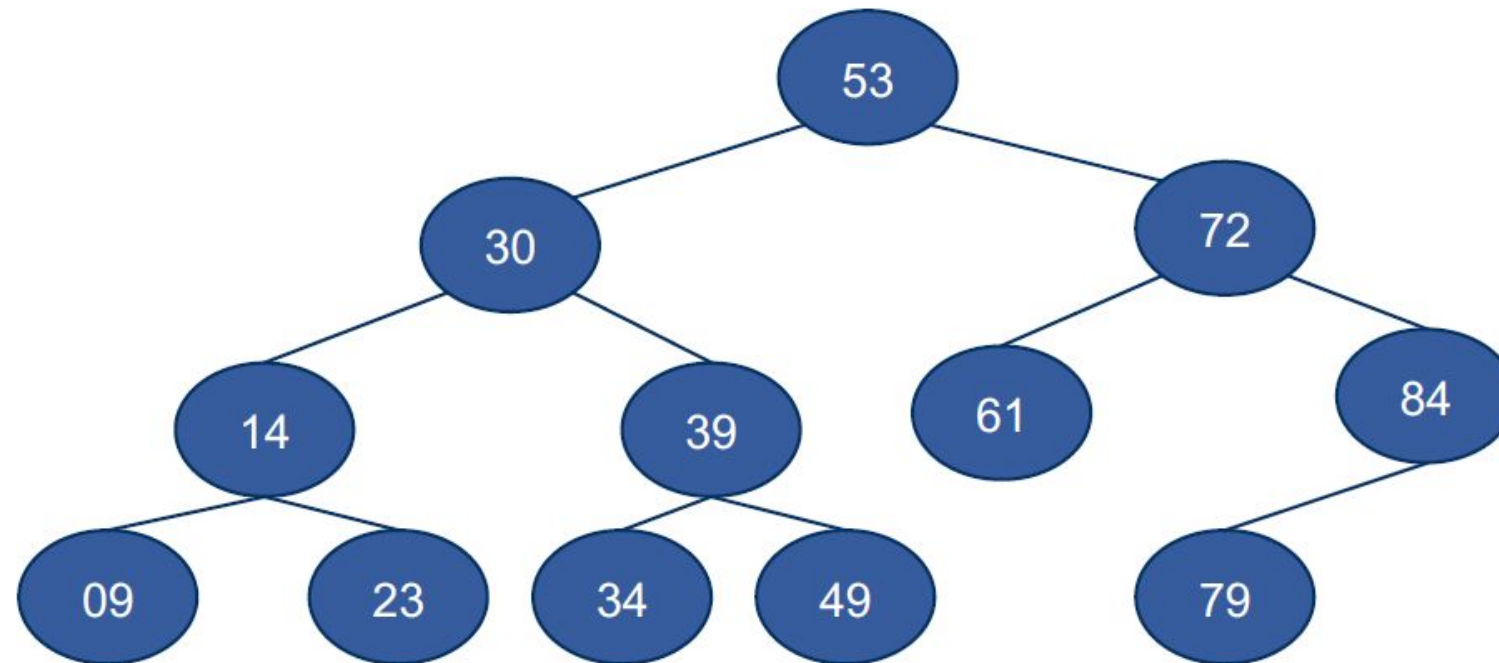
- **Folha**

- Um nó que não tem filhos.



ÁRVORE BINÁRIA DE BUSCA

- O filho à esquerda de um nó tem que ter uma chave menor que seu pai e o filho à direita de um nó tem que ter uma chave maior ou igual ao seu pai.



ÁRVORE BINÁRIA DE BUSCA - INSERÇÃO

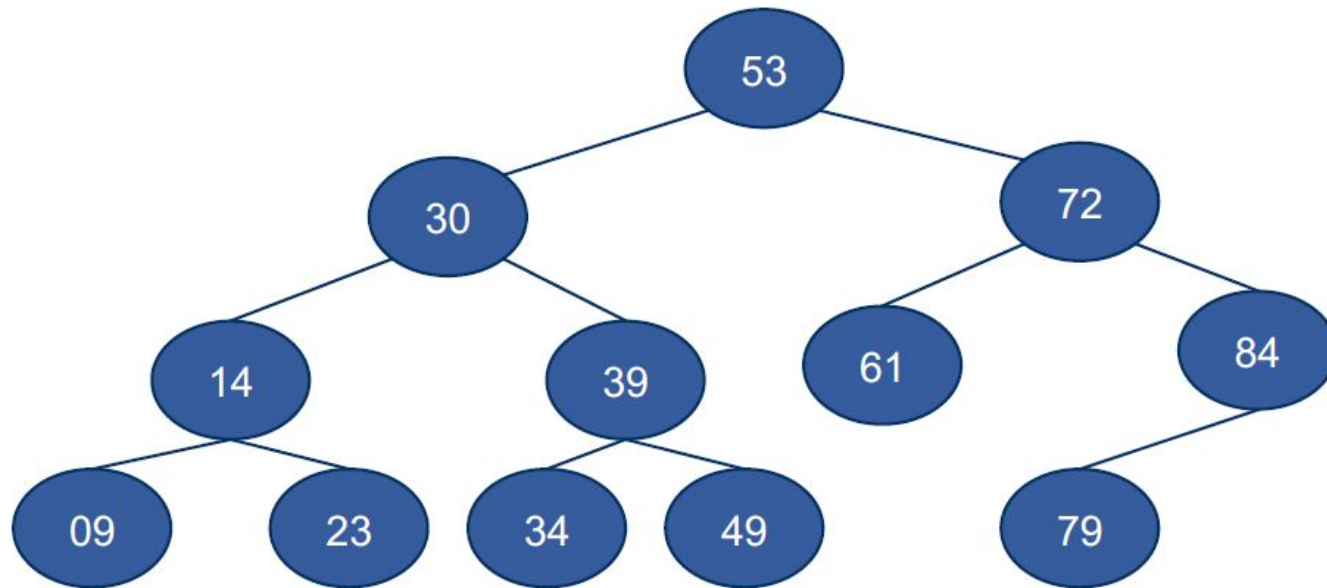
- Primeiro, o local para inserir deve ser encontrado;
- Segue-se o caminho da raiz até o devido nó, que será pai do novo nó;
- Quando esse pai for localizado, o novo nó será conectado como seu filho à esquerda ou a direita, dependendo da chave do novo nó ser menor ou maior que a do pai;
- Visualização on-line: <https://visualgo.net/en/bst>

ÁRVORE BINÁRIA DE BUSCA - PESQUISA

- Procurar nas subárvores da esquerda ou direita
- Visualização on-line: <https://visualgo.net/en/bst>

ÁRVORE BINÁRIA - TRAVESSIA PRÉ-ORDEM

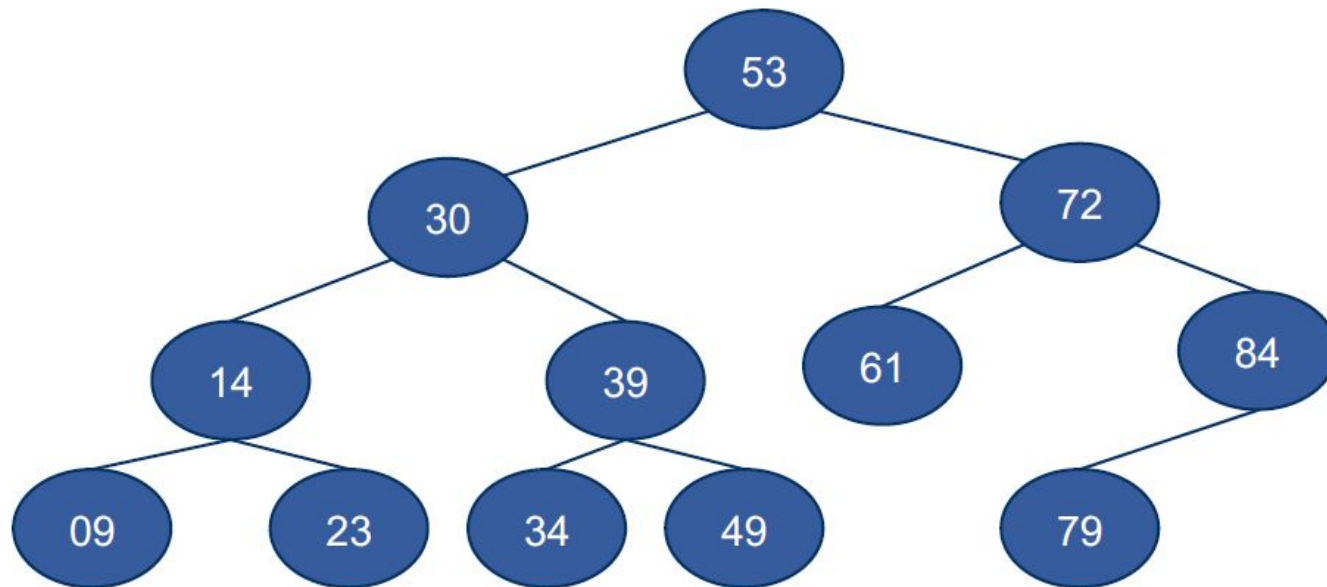
- Primeiro visita a raiz e depois **recursivamente** faz uma travessia na subárvore esquerda, seguido de uma travessia recursiva na subárvore direita;
- Raiz, esquerda, direita.



- 53
- 30
- 14
- 9
- 23
- 39
- 34
- 49
- 72
- 61
- 84
- 79

ÁRVORE BINÁRIA - TRAVESSIA EM ORDEM

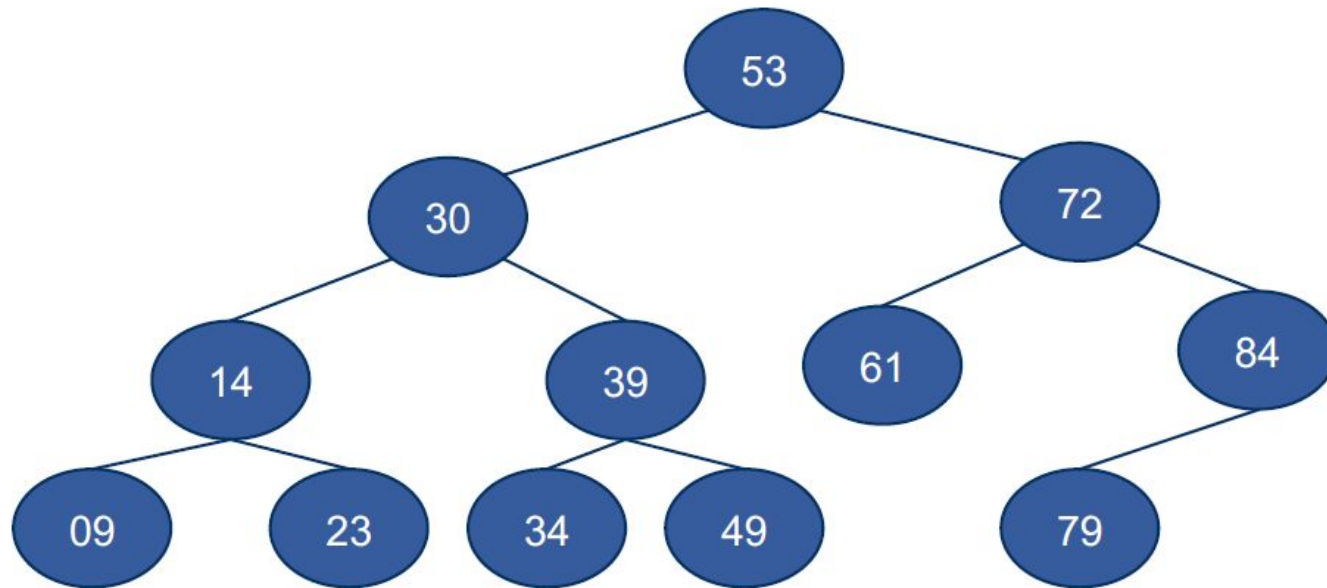
- **Recursivamente** faz a travessia na subárvore esquerda, visita a raiz e faz uma travessia recursiva na subárvore direita;
- Esquerda, raiz, direita.



- 09
- 14
- 23
- 30
- 34
- 39
- 49
- 53
- 61
- 72
- 79
- 84

ÁRVORE BINÁRIA - TRAVESSIA PÓS-ORDEM

- **Recursivamente** faz a travessia na subárvore esquerda, faz uma travessia recursiva na subárvore direita e por fim visita a raiz;
- Esquerda, direita, raiz.



- 09
- 23
- 14
- 34
- 49
- 39
- 30
- 61
- 79
- 84
- 72
- 53

ÁRVORE BINÁRIA DE BUSCA - EXCLUSÃO

- Inicia-se localizando o nó que deseja eliminar;
- Quando encontrar o nó:
 - O nó a ser apagado é uma folha;
 - O nó a ser apagado tem um filho;
 - O nó a ser apagado tem dois filhos.

ÁRVORE BINÁRIA DE BUSCA - EXCLUSÃO

- O nó a ser apagado é uma folha;
 - A mais simples operação;
 - Altera o campo apropriado para o filho no nó pai para apontar para nulo em vez do nó;
 - Visualização on-line: <https://visualgo.net/en/bst>

ÁRVORE BINÁRIA DE BUSCA - EXCLUSÃO

- O nó a ser apagado tem um filho;
 - O nó tem apenas duas conexões: com seu pai e com seu único filho;
 - Deseja-se “cortar” o nó dessa sequência conectando o pai dele diretamente ao filho dele
 - Visualização on-line: <https://visualgo.net/en/bst>

ÁRVORE BINÁRIA DE BUSCA - EXCLUSÃO

- O nó a ser apagado tem dois filhos;
 - O nó a ser apagado deve ser substituído por seu sucessor em ordem;
 - Visualização on-line: <https://visualgo.net/en/bst>