Portugol-webstudio.cubos.io/ide – link para testes em portugol – ensina a lógica de programação

Brunocdias.blogspot.com – professor de estrutura de dados e algoritmo

**Estrutura de dados** – Organização de dados na memória – imagine uma cidade onde existe rua, bairro, etc e para organizar e manipular essas informações, existem as estruturas de dados e podem ser guardadas em nuvem, hd, pendrive, etc

**Algoritmo** - são instruções organizadas e estruturadas e seu objetivo é realizar uma tarefa específica. E os algoritmos são usados para manipular dados, inserindo, excluindo, alterando e etc.

Imagine um armário com 5 gavetas e vc deseja localizar, inserir, ordenar, classificar o que existe dentro delas. As principais estruturas de dados são: Vetores e matrizes, registro, lista, pilha, fila, árvore, tabela hash e grafos

**Vetores** – Auxiliam quando precisamos utilizar muitas variáveis de um mesmo tipo. Imagine que vc precisa declarar 5 nomes e usando o vetor, vc não precisa declarar nome1, nome2 e etc, vc declara um vetor com 5 posições e armazena nele os 5 nomes. Possui tamanho fixo

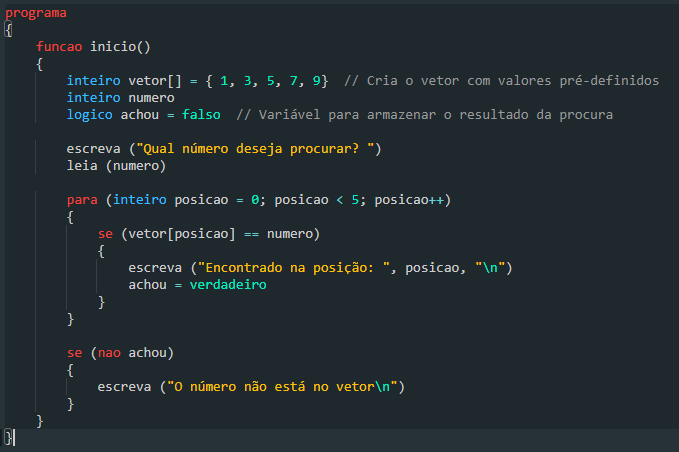
É uma estrutura indexada, possui índice, sempre iniciando de zero até o valor desejado.

Então, vamos declarar nome[5] (vetor chamado nome com 5 posições, ou nome[] = {aline, eliseu, sofia, lorena, pingo}, onde se mandar imprimir nome[3], o resultado será = lorena.



O vetor acima tem 7 posições de números inteiros, se eu tiver procurando o nro 15, ele será a posição 3 desse vetor.

Outro exemplo:





Matriz – basicamente é um vetor de vetores. Um vetor que possui mais de uma dimensão (linha x coluna). Possui tamanho fixo.

Exmplo de Diagonal principal



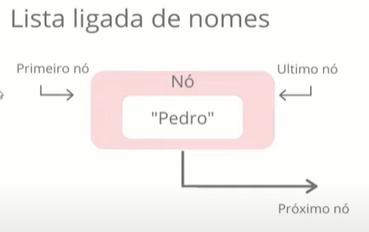
Nesse exemplo, como foi declarado tamanho [5], significa que a matriz tem 25 posições.

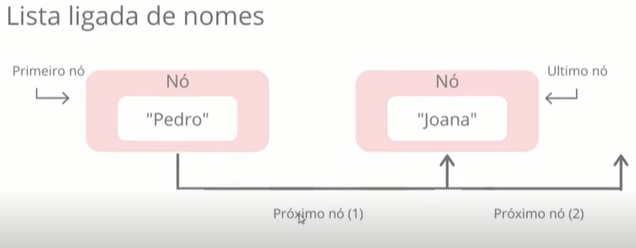
**Registro** – podem receber dados de tipos diferentes. Exemplo, um registro pode ter campos do tipo inteiro, data, string, booleano e etc

Imagine um registro Cliente = [aline, 279621, 240380, 42, casada]

Para ler o registro, usamos nomedoregistro.campo - exemplo do registro cliente, cliente.nome = aline, cliente.idade = 42

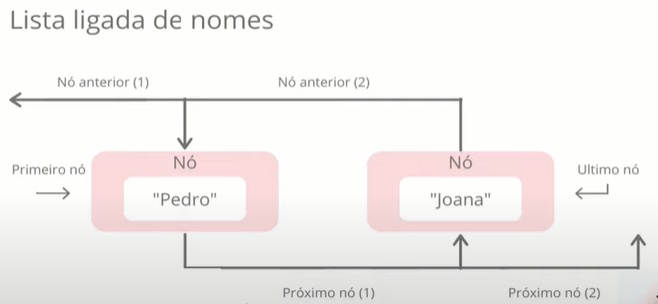
**Listas** – armazenam determinados tipos em ordem específica, e possui um tamanho ajustável, vc declara a variável do tipo lista e ela pode ir crescendo conforme uma lista de presença de uma festa. Na lista ligada, existe o nó... onde 1 elemento é ligado a outro, tendo uma indicação para o próximo nó





Se removido o dado “Joana”, o Pedro volta a ser o último nó e se inserido outro dado, joana não será mais o último nó e sim o último item inserido.

Já na lista duplamente ligada, podemos ter ligação com o valor anterior e também posterior



Pode fazer busca reversa, inserir ou deletar antes ou depois

**Pilha –** O acesso aos itens da pilha são restritos, só pode ser lido ou removido por vez. Pode ser LIFO Last (First Out), ou UEPS – ultima que entra e primeiro que sai, onde o primeiro elemento a ser retirado é o último que vc inseriu. Imagine a inserção1

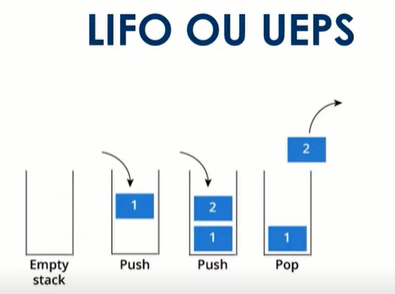
1

2

3

Nessa pilha, o primeiro elemento a sair será o 3.

Pilha FIFO (First In, First Out) ou PEPS – primeiro que entra é o primeiro que sai – nesse caso, olhando a lista de 1,2,3, o primeiro elemento a ser manipulado é o 1.





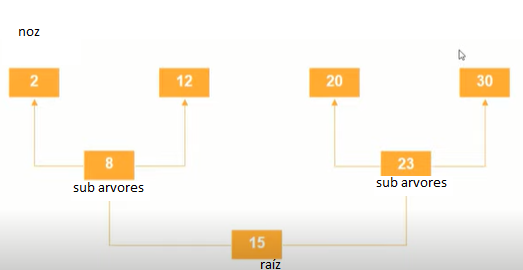
Fila – primeiro objeto a ser tratado é o que está na fila a mais tempo, segue o modelo FIFO – primeiro que entra é o primeiro que sai



**Árvore** – Estrutura organizada de dados por hierarquia, onde existe o elemento no topo da árvore, chamado de raiz (sempre o meio) e os elementos subordinados a ele, chamados de folha ou nós

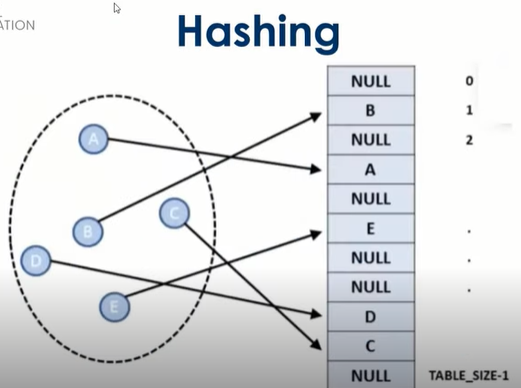


Se estiver procurando o número 15, deverá passar por cada posição, até que seja encontrado o desejado. Forma mais lenta de pesquisa

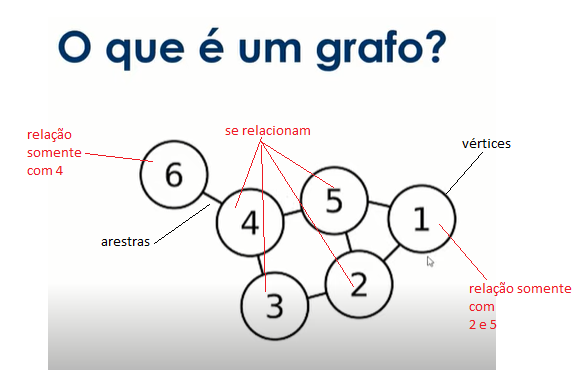


Imagine que esteja procurando o nro 30, o sistema de árvore, inicia a busca pelo meio (raiz) e quebra em partes iguais o total de dados (sub árvores), e vai eliminando as etapas até encontrar o item desejado. Busca organizada.

**Tabela Hash** – É uma generalização da ideia de vetor. Ela espalha os elementos deixando-os não ordenados dentro do vetor que define a tabela. Permite que seja associado valores a chaves, onde valor é a posição ou índice do elemento e chave é parte da informação que compõe o elemento. Facilidade de busca através da chave, mesmo com dados espalhados. Cada valor recebe uma chave



**Grafo –** Permite programar a relação entre objetos. Os objetos são vértices ou nós do grafo e os relacionamentos são arestas.



Imagine então uma rede social, onde um grupo se conhece, mas existe uma pessoa que conhece apenas 1 integrante daquele grupo – as vértices (bolinhas) são as pessoas da rede social e as arestras (setas) são os relacionamentos entre as pessoas. Esse tipo de estrutura permite a liberdade de se direcionar em qualquer posição e não somente para frente, acima, diagonal e sim todas as direções.