# Aula 01 (5/8/24)

algoritmo: conjunto finito de passos em ordem lógica para solucionar ou fazer uma tarefa.

todo algoritmo processa/manipula dados para gerar uma saída.

1980 - 1995: programadores não se preocupavam em como guardar os dados no banco. Resultado: bug do milênio.

## Dados (a importância e o refinamento)

Base da pirâmide: Dados brutos (coletando) [maior volume de dados]

2° andar: Informação (Processado)

3° andar: Conhecimento (Analisada)

4° andar: Inteligência (Comunicado)

5° andar: Decisão (Aplicada)

6° andar: ação (Formulada)

No topo: Valor (Implantada)

TI trabalha até 3° andar

## O que são tipos de dados?

Definição do conjunto de valores (domínio [o que eu posso colocar e seus elementos]) que a variável pode assumir na aplicação e do conjunto de operações que podem ser aplicadas sobre ele.

Byte menor unidade de informação formado por 256 bits.

## Quais são os dados básicos (primitivos):

- Numéricos (inteiro e real)

- Alfanuméricos (char e string\*)

- Lógicos (True e False)

## O que são dados estruturados?

Permitem agregar mais de um valor em uma variável.

Existe relação estrutural entre os seus elementos.

Ex: vetor

## Quais são os dados estruturados?

-Arranjo (vetor, matriz, agregado e array): utilizados para agregar componentes do mesmo tipo (homogêneos).

Ex: int a = [1,4,2,4,6,2,3]

- Registros: Utilizados para agregar componentes diferentes (heterogêneos)

Ex: struct reg{

string nome;

int idade;

float altura;

char sexo;

};

- Dados: Constituídos hierarquicamente através de componentes. Conjunto de componentes que podem ser de tipos diferentes, agrupados sob uma única variável. Relação semântica. Manipulação organizada dos dados.

## O que são estruturas de dados?

Especificam conceitualmente os dados, de forma a refletir um relacionamento lógico entre os dados e o domínio do problema considerado.

Utilizar um sistema de organização com uma regra.

Ex: queue (fila)

Incluem operações para manipulação de seus dados.

Esse nível conceitual não é fornecido diretamente pelas linguagens.

## Tipos abstratos de dados (TAD)

Haver a OO;

Capazes de representar os tipos de dados que não foram previstos no núcleo das linguagens e que, normalmente, são necessários para aplicação específicas.

Ex: classe Cliente

Duas partes: DADOS e OPERAÇÕES (que devemos nos preocupar)

Vantagens: mais fácil de programar, sem se preocupar com detalhes de implantação; mais fácil preservar a integridade dos dados; maior independência e portabilidade de código; e maior reutilização.

## Tipos de representação:

Forma utilizada para representar fisicamente os relacionamentos lógicos entre os dados.

Exisem 2 representações: Contiguidade física (sequencial) e encadeamento.

Contiguidade física: a pesquisar

Encadeamento: os dados são armazenados em posições não contíguas na memória. São alocados aleatoriamente.

## Tipos de alocação de memória:

Processo de solicitar/utilizar memória durante processo de execução

-Estática: tamanho fixo; organizados sequencialmente na memória do computador.

-Dinâmica: Tamanho pode ser modificado em tempo de execução; Não precisam estar precisamente organizados de maneira sequencial, podendo estar distribuídos de forma dispersa (não ordenada) na memória do computador.

### Atividade: Estruturas de Dados

Questão 1: O que é um algoritmo e qual a sua função principal?

Questão 2: Quais são os seis andares na hierarquia de dados e o que cada um representa?

Questão 3: Qual a importância de armazenar dados corretamente, conforme o exemplo do bug do milênio?

Questão 4: Defina o que são dados brutos e forneça um exemplo.

Questão 5: Diferencie dados numéricos, alfanuméricos e lógicos com exemplos para cada tipo.

Questão 6: O que são dados estruturados? Dê um exemplo e explique sua estrutura.

Questão 7: Explique o conceito de vetor e forneça um exemplo em código de um vetor de inteiros.

Questão 8: Qual é a diferença entre um vetor e uma matriz?

Questão 9: Descreva a estrutura de um registro em C++ e forneça um exemplo com três diferentes tipos de dados.

Questão 10: O que é uma estrutura de dados e qual a sua importância na programação?

Questão 11: Explique a diferença entre tipos abstratos de dados (TAD) e tipos de dados primitivos.

Questão 12: O que é uma fila (queue) e como ela é organizada?

Questão 13: Qual é a diferença entre contiguidade física e encadeamento na representação de dados?

Questão 14: Defina alocação de memória estática e forneça um exemplo onde ela seria utilizada.

Questão 15: Explique o conceito de alocação dinâmica de memória e quando seria mais vantajoso usá-la.

Questão 16: Descreva a estrutura de dados "lista encadeada" e como ela é diferente de um vetor.

Questão 17: Como a orientação a objetos (OO) contribui para a criação de tipos abstratos de dados?

Questão 18: Explique o conceito de encapsulamento em tipos abstratos de dados e sua importância.

Questão 19: Dê um exemplo de como a reutilização de código é facilitada por tipos abstratos de dados.

Questão 20: Descreva um cenário onde a alocação dinâmica de memória é preferível em comparação à alocação estática e explique por quê.