

O que é uma  
**ESTRUTURA DE DADOS?**

**ESTRUTURA DE DADOS**  
**=**  
**CONJUNTO DE DADOS**

O que é uma  
estrutura de dados  
HOMOGÊNEA?

**ESTRUTURA DE DADOS  
HOMOGÊNEA**

**=**

**CONJUNTO DE DADOS  
DO MESMO TIPO**

# Estrutura de dados



Alunos

Rosa

Ana

Pedro

Joana

José

Ana

João

**Conjunto de Alunos**

**=**

**Estrutura de dados**

**homogênea**

**=**

**Estrutura de dados do**

**tipo CHARACTER**

**Conjunto de Idades**

**=**

**Estrutura de dados  
homogênea**

**=**

**Estrutura de dados do  
tipo INTEIRO**

Idades

23

38

47

19

26

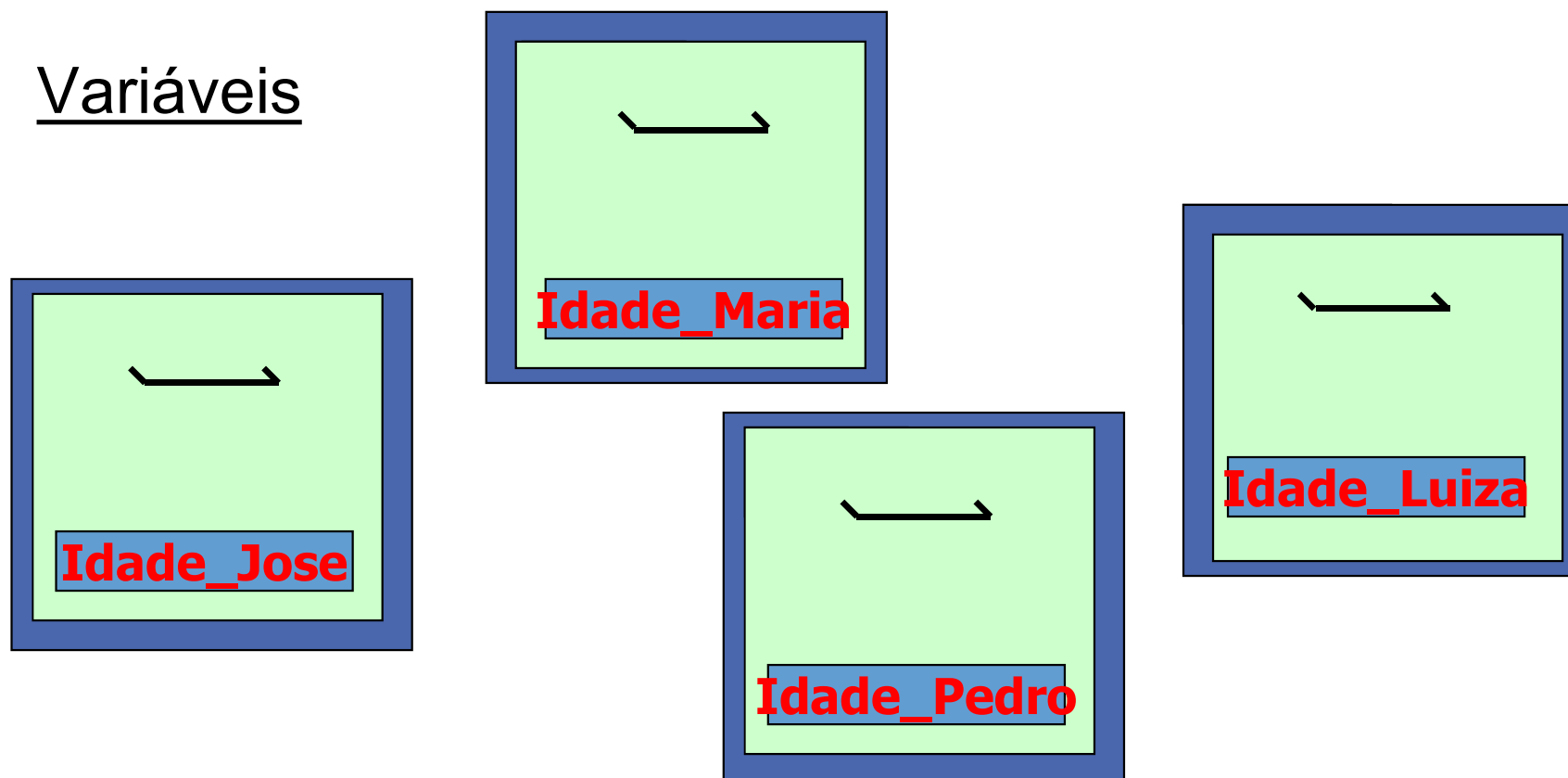
52

29

# Problema

- Preciso de um software que armazene a idade, em anos, de quatro pessoas:
  - Maria
  - José
  - Pedro
  - Luiza;
- Neste caso, precisamos então criar quatro lugares (caixinhas) na memória para guardar, em cada uma, a idade de uma das pessoas.

## ■ Variáveis



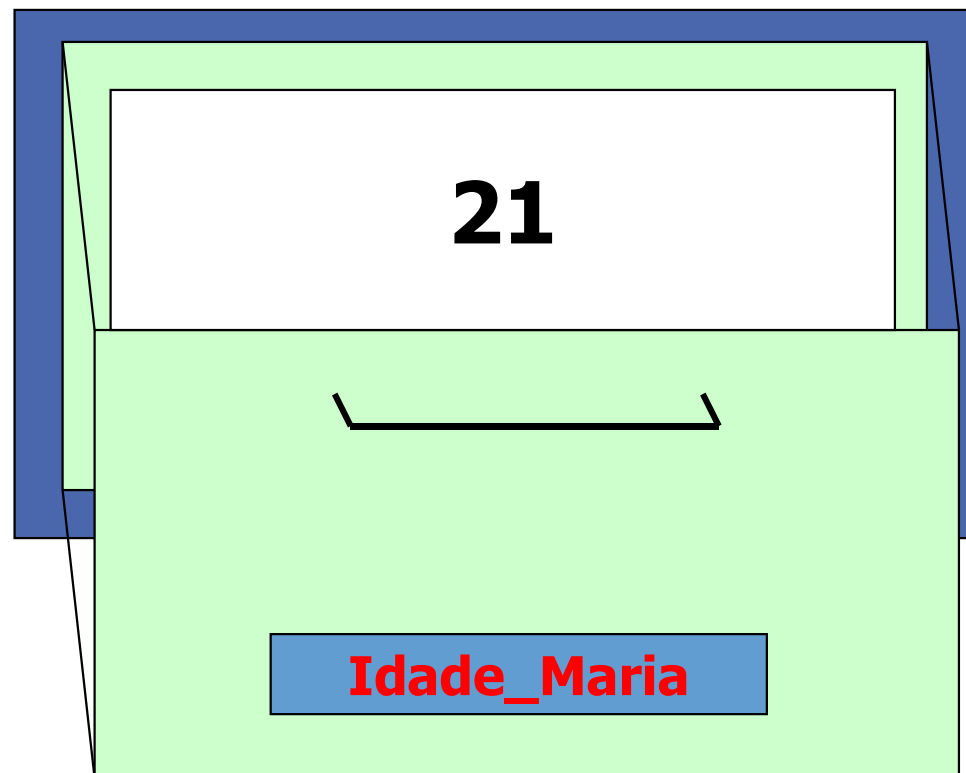
**Todas as variáveis guardam um conteúdo de mesmo significado e são do mesmo tipo de dados.**



## Vamos atribuir valores às variáveis criadas:

- Variáveis:
  - Idade\_Maria ← 21
  - Idade\_Pedro ← 55
  - Idade\_Jose ← 42
  - Idade\_Luiza ← 10

- Variável Idade\_Maria



# Algoritmo

Início

Inteiro: Idade\_Maria, Idade\_Jose, Idade\_Pedro, Idade\_Luiza ro

Idade\_Maria      21

Idade\_Jose   ← 42

Idade\_Pedro ← 55

Idade\_Luiza   ← 10

←  
escreva "Idade de Maria: ", Idade\_Maria

escreva "Idade de José: ", Idade\_Jose

escreva "Idade de Pedro: ", Idade\_Pedro

escreva "Idade de Luiza: ", Idade\_Luiza

fim

## Pergunta

- Para guardar 100 idades de 100 pessoas distintas, o que precisamos fazer?
  - Até então, criar 100 variáveis. Uma para cada pessoa.
- E se tivermos que guardar as idades de 1000 pessoas?
- Será que não existe nada mais prático?

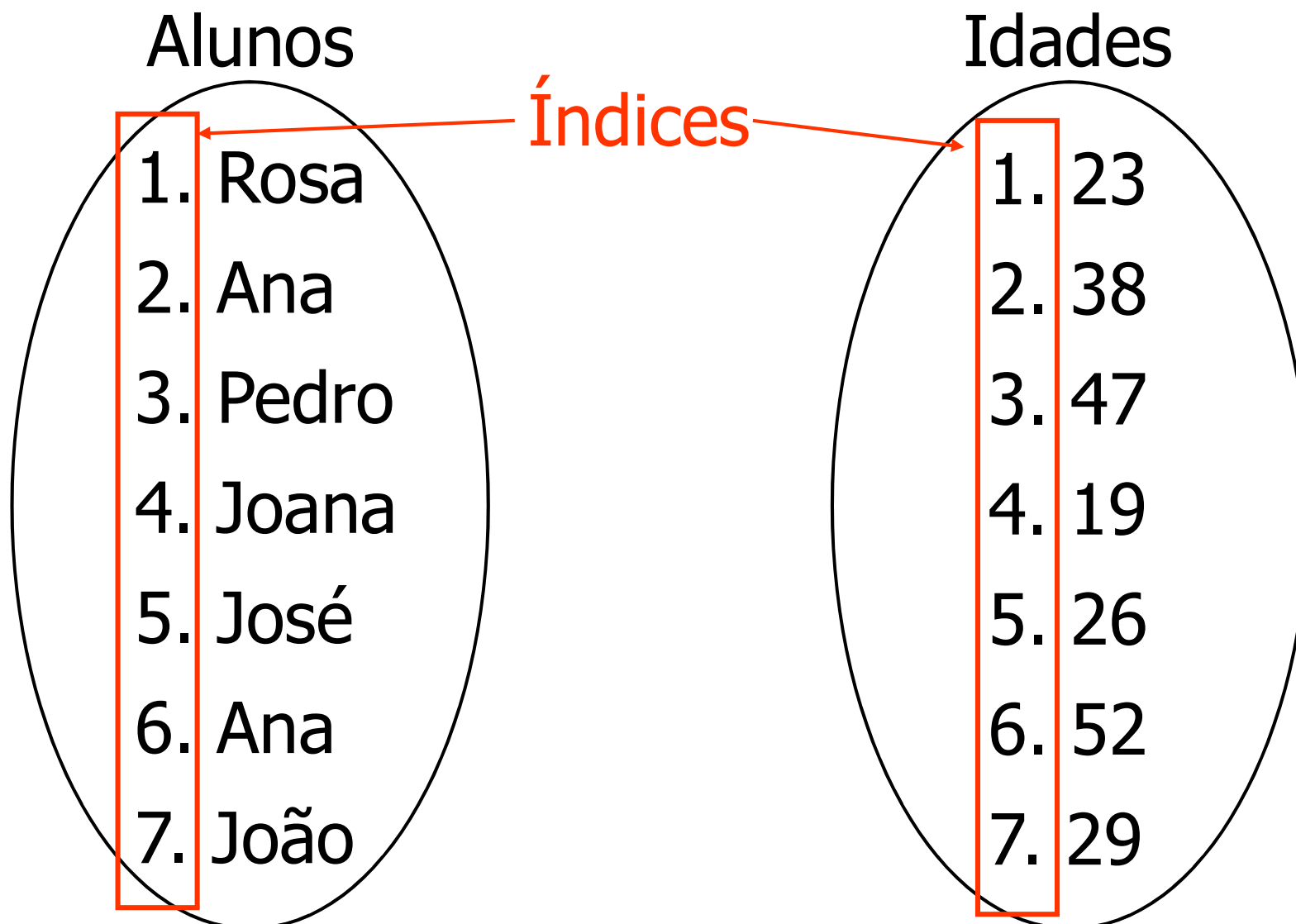
O que é estrutura de dados  
**UNIDIMENSIONAL?**

**ESTRUTURA DE DADOS  
UNIDIMENSIONAL**

**=**

**CONJUNTO DE DADOS  
onde cada elemento é  
identificado por um  
único índice**

# Estrutura de dados Unidimensional



# ESTRUTURA DE DADOS UNIDIMENSIONAL

=

# VETOR em Computação



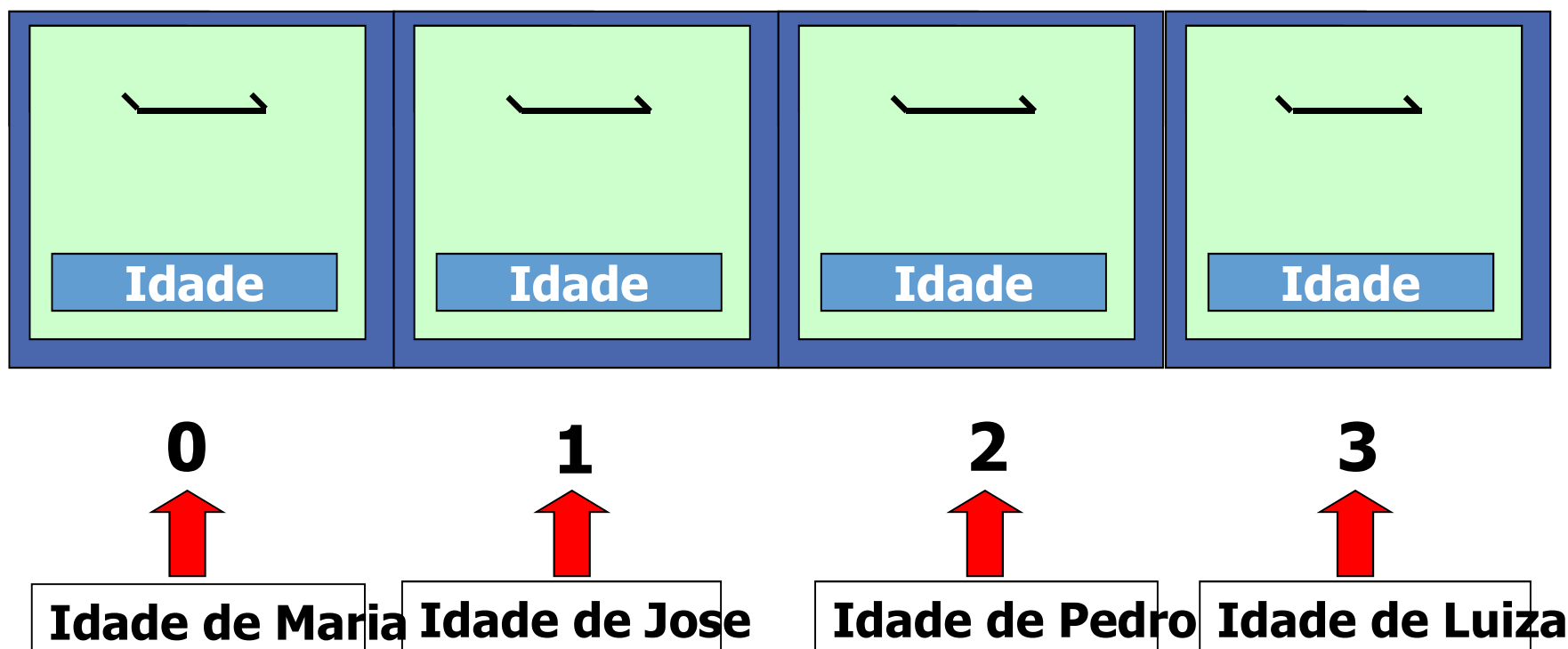
Qual o nome e a idade  
do Aluno 5?

José, 26!

# Vetor

- Para situações como esta, apresentada anteriormente, foi criada uma estrutura conhecida como **VETOR**;
- Um vetor nada mais é do que **UMA** variável com diversas posições (caixinhas) numeradas. Onde pode-se guardar diversos valores (um em cada caixinha) do mesmo tipo.

## ■ Variável Idade (Vetor)



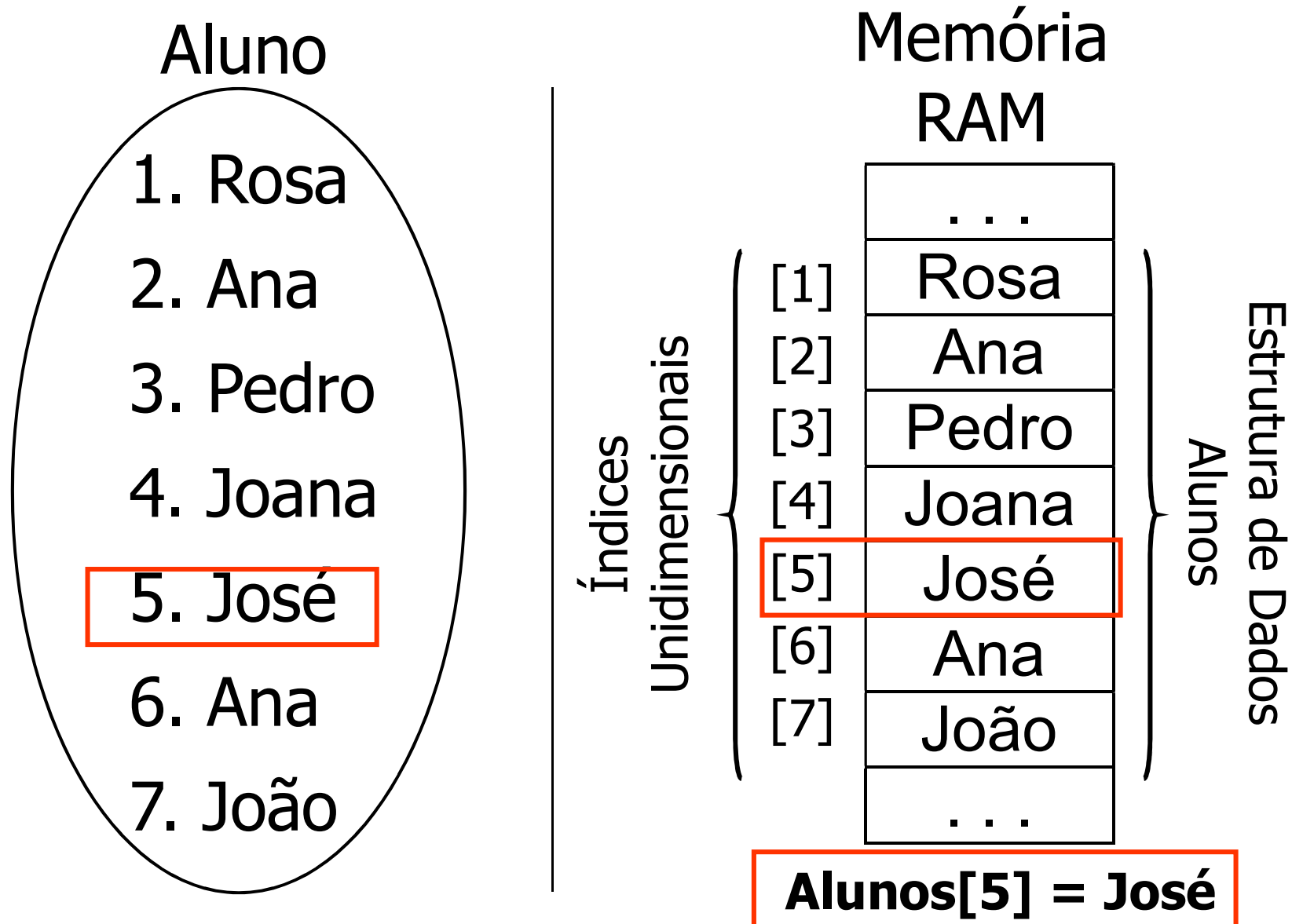
## Associação

- Deve-se fazer, internamente, uma associação das posições de memória à cada pessoa;

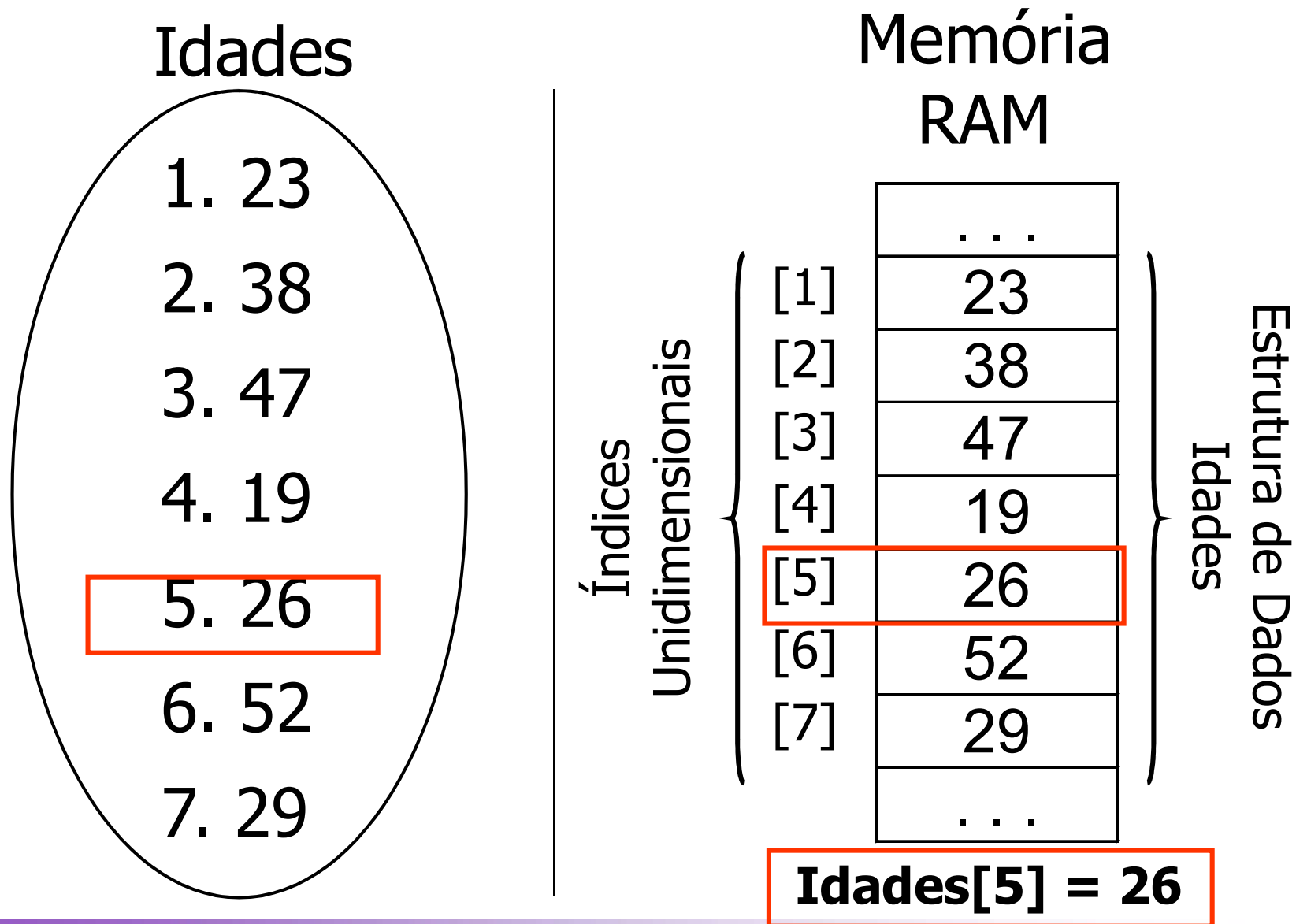
Pessoa	Posição de Armazenagem
Maria	0
Jose	1
Pedro	2
Luiza	3

Como isso funciona  
no Computador?

# Estrutura de dados Unidimensional



# Estrutura de dados Unidimensional





Como se faz isso  
no Algoritmo?

# Estrutura de dados Unidimensional



- Faça um algoritmo para ler o nome e a idade de 7 Alunos.

**Algoritmo ExemploEstruturaUnidimensional**

**N = 7; {Declaração da constante N com o valor 7}**

**Nomes: vetor [1..N] de Caracter; {Declaração do tipo TNomes}**

**Idades: vetor [1..N] de Inteiro; {Declaração do tipo TIdade}**

**i: Inteiro; {Declaração da variável contadora do laço para}**

**Início**

**Para  $i \leftarrow 1$  até N faça {Laço para ler os 7 nomes e idades dos Alunos}**

**Escreva (“Informe nome e idade do Aluno ” , i);**

**Leia (Nomes[i], Idades[i]);**

**Fim Para;**

**Escreva (Nomes[5], Idades[5]); {escreve o nome e idade do funcionário 5}**

**FIM.**

# Vetores em Java

---



- Declaração
  - `int c[ ];`
- Vetor em Java é um objeto, então deve ser instanciado
  - `c = new int[10];`
- Declarando e criando
  - `int c[] = new int [10];`

# Vetores em Java

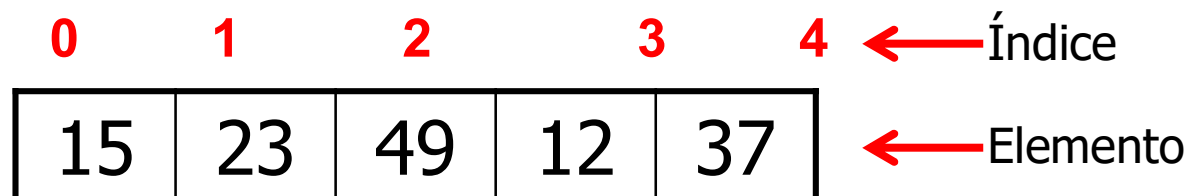


- Outros exemplos:

- `String nomes[] = new String [ 100 ];`
- `double notas[] = new double [ 150 ];`

- Iniciando vetores com valores

- `int dados[] = {15,23,49,12,37};`



- `String meses[] = { "Janeiro", "Fevereiro", "Dezembro" };`

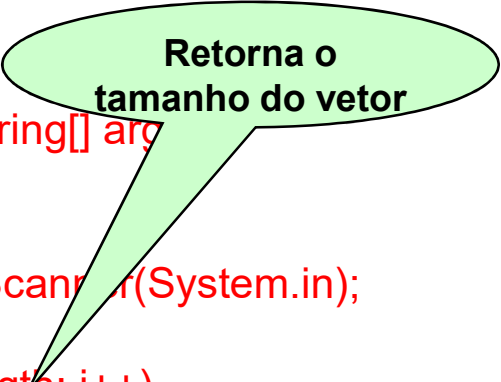


# Vetores em Java



Lendo e imprimindo um vetor de tamanho 5:

```
public class Vetor {  
    public static void main(String[] args) {  
        int vet[] = new int[5];  
  
        Scanner input = new Scanner(System.in);  
  
        for (int i = 0; i < vet.length; i++)  
        {  
            System.out.format("Digite o elemento %d do vetor: ", i);  
            vet[i] = input.nextInt();  
        }  
  
        System.out.println("Imprimindo o vetor...");  
        for (int i = 0; i < vet.length; i++)  
        {  
            System.out.println(vet[i]);  
        }  
    }  
}
```



Retorna o tamanho do vetor

# Vetores



- Ler um vetor de 5 posições, e em seguida, a soma de seus elementos:

```
public class Vetor {  
    public static void main(String[] args) {  
        int vet[] = new int[5];  
  
        int soma = 0;  
  
        Scanner input = new Scanner(System.in);  
  
        for (int i = 0; i < vet.length; i++)  
        {  
            System.out.format("Digite o elemento %d do vetor: ", i);  
            vet[i] = input.nextInt();  
            soma += vet[i];  
        }  
  
        System.out.format("\nSoma dos elementos do vetor: %d\n", soma);  
    }  
}
```

# Exemplo



Ler um vetor de inteiros de 5 posições e imprimir este vetor em ordem inversa

```
public class Vetor {  
    public static void main(String[] args) {  
        int vet[] = new int[5];  
  
        Scanner input = new Scanner(System.in);  
  
        for (int i = 0; i < vet.length; i++)  
        {  
            System.out.format("Digite o elemento %d do vetor: ", i);  
            vet[i] = input.nextInt();  
        }  
  
        System.out.println("Imprimindo o vetor em ordem inversa...");  
        for (int i = vet.length-1; i >= 0; i--)  
            System.out.format("%d \t", vet[i]);  
    }  
}
```

- Declaração de vetores
- Semântica:
  - São estruturas de dados homogêneas unidimensionais que permitem agrupar diversas informações dentro de uma variável.
  - Estas correspondem a um grupo de posições contínuas na memória que possuem o mesmo nome e o mesmo tipo de dado e são acessadas por um ÚNICO índice.
  - Seu tamanho é definido por constantes inteiras e positivas e a definição do seu nome segue as mesmas regras aplicadas para identificadores.



## Note I

Os vetores Nomes e Idades têm 7 posições cada. Isto é, cada um equivale a 7 variáveis de mesmo nome, só distinguíveis pelos seus índices. Ou seja, os dois vetores juntos equivalem a criar 14 variáveis, só que é menos trabalhoso!

E se aumentar de 7 para  
700 Alunos?

# Estrutura de dados Unidimensional



## Algoritmo ExemploEstruturaUnidimensional

**N = 700;** {é só alterar o valor da constante, que N muda automaticamente!}

**Nomes :** vetor [N] de Caracter;

**Idades :** vetor [N] de Inteiro;

**i:**Inteiro;

**Início**

**Para**  $i \leftarrow 1$  até N **faça**

**Escreva** (“Informe nome e idade do funcionário ”, i);

**Leia** (Nomes[i], Idades[i]);

**Fim Para;**

**Escreva** (Nomes[5], Idades[5]);

**FIM.**

## Note II

Resolver o problema anterior sem vetores implicaria o árduo trabalho de ter que declarar e manipular 700 variáveis!

Por isso, a solução de certos problemas só é viável usando estruturas de dados!

## Note III

Não é possível operar com todos os elementos do vetor de uma só vez. Por isso, o correto é acessar cada um de seus elementos isoladamente.

## Note IV

O acesso a cada elemento de um vetor é feito pela manipulação do seu índice entre [colchetes]!

# Estrutura de dados Unidimensional



EX: Algoritmo ExemploUsoDeVetor

**{Calcular a média de 10 idades e exibir aquelas acima da média}**

N = 10;

Idade : vetor [N] de Inteiro;

Inteiro: i, soma ; Real : media ;

Início

soma  $\leftarrow$  0;

Para i  $\leftarrow$  1 até N faça

    Escreva (“informe a idade”, i);

    Leia (idade [ i ]);

    soma  $\leftarrow$  soma + idade[ i ];

Fim Para;

media  $\leftarrow$  soma / n;

Para i  $\leftarrow$  1 até n faça

    Se (idade [ i ] > media) Então

        Escreva (idade[ i ]);

Fim Para

Fim.

{Laço para ler e  
somar as N notas}

{Laço para exibir as notas que  
são maiores que a média}

O que é estrutura de dados  
**MULTIDIMENSIONAL?**



**ESTRUTURA DE DADOS  
MULTIDIMENSIONAL**

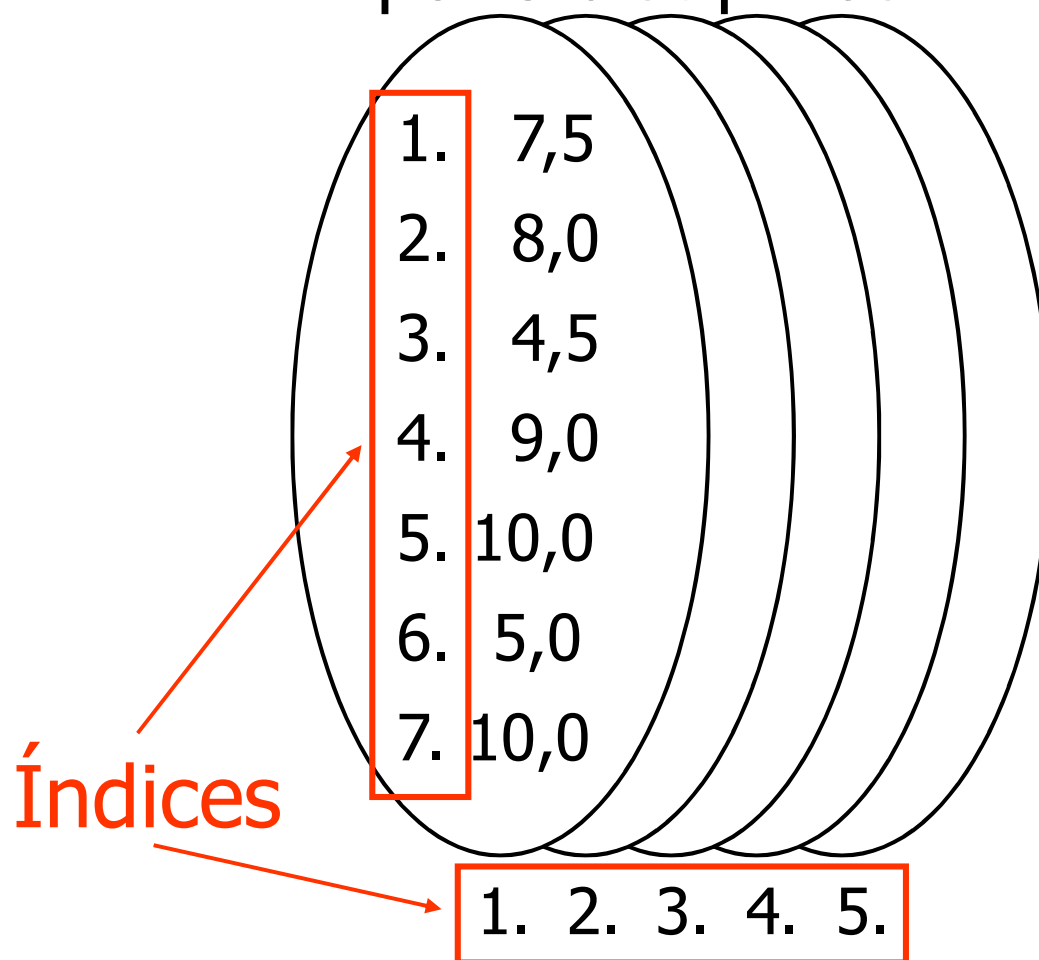
**=**

**CONJUNTO DE DADOS  
onde cada elemento é  
identificado por MAIS  
de um índice**

# Estrutura de dados Multidimensional



Notas de 7 alunos  
por 5 disciplinas



# ESTRUTURA DE DADOS MULTIDIMENSIONAL

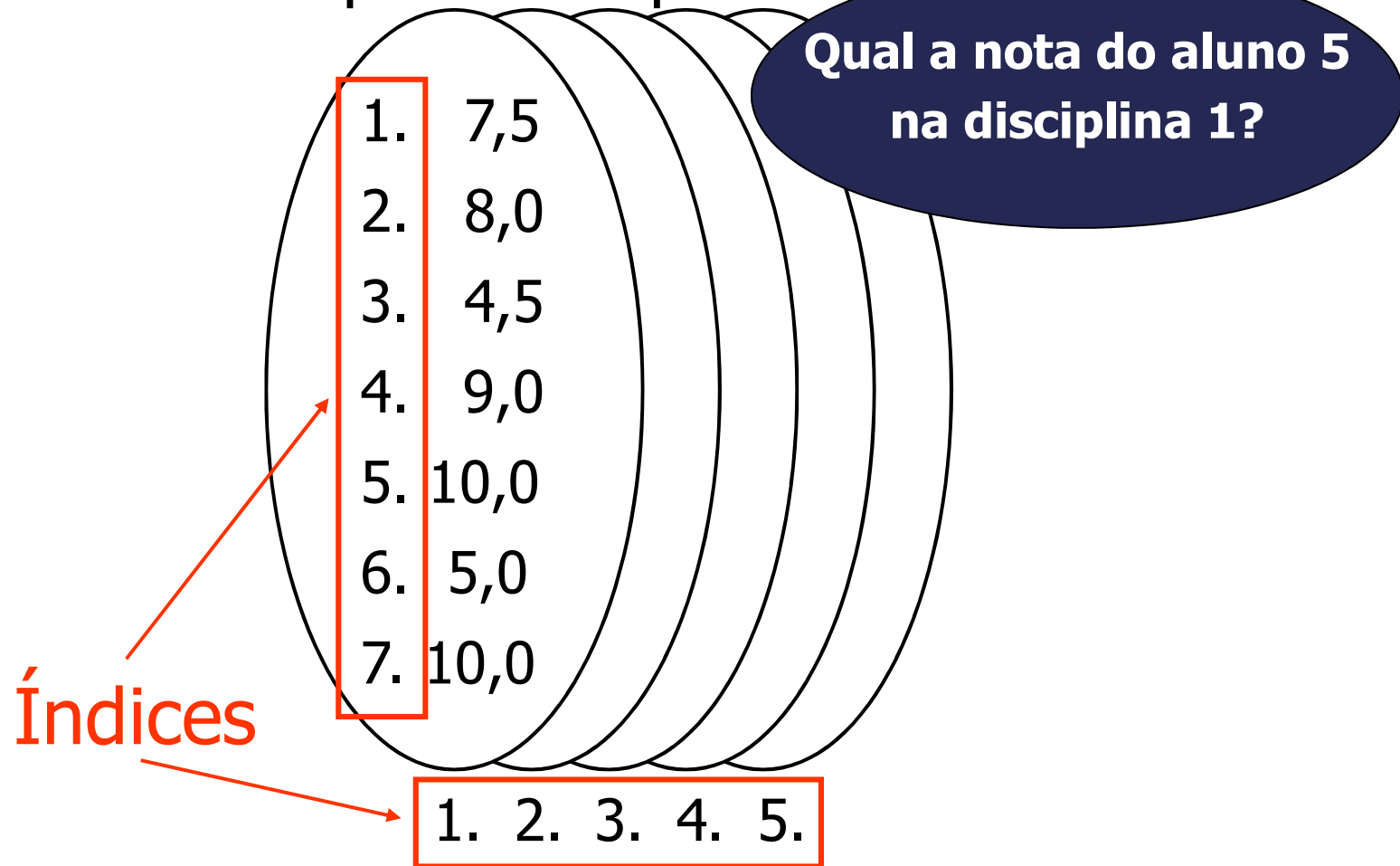
=

# MATRIZ em Computação

# Estrutura de dados Multidimensional



Notas de 7 alunos  
por 5 disciplinas

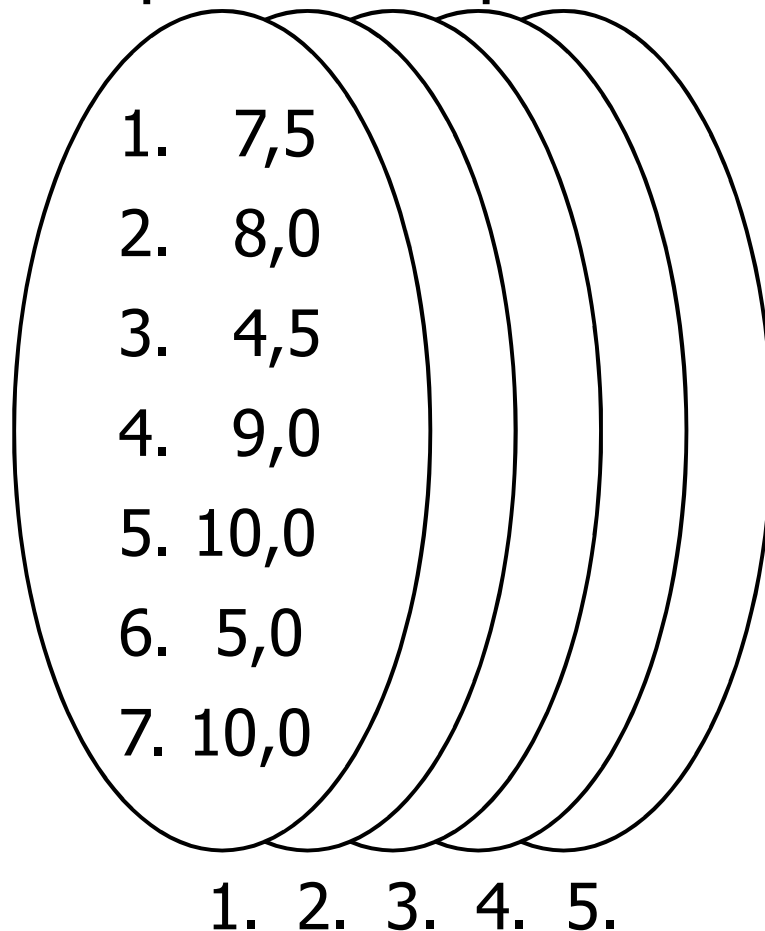


Como isso funciona  
no Computador?

# Estrutura de dados Multidimensional



Notas de 7 alunos  
por 5 disciplinas



Memória  
RAM

## Estrutura de Dados Notas

Índices Multidimensionais

	[1]	[2]	[3]	[4]	[5]
[1]	7,5	8,0	5,0	8,5	9,0
[2]	8,0	7,0	6,5	8,5	10,0
[3]	4,5	8,0	7,0	7,0	8,5
[4]	9,0	6,5	8,5	9,0	9,0
[5]	10,0	9,0	8,5	9,0	10,0
[6]	5,0	7,5	6,5	8,0	9,0
[7]	10,0	9,0	8,5	9,5	8,0

**notas[5,1] = 10,0**

Como se faz isso  
no Algoritmo?

# Estrutura de dados Multidimensional



- Faça um algoritmo para ler as notas de 7 alunos em 5 disciplinas

**Algoritmo ExemploEstruturaMultidimensional**

**Alunos = 7; Disciplinas = 5; {Constantes alunos e disciplina}**

**notas = matriz [1..Alunos, 1..Disciplinas] de Real; {Tipo TNotas}**

**Inteiro: a, d; {Variáveis contadoras dos laços paras}**

**Início**

**Para a ← 1 até Alunos faça {Laço para percorrer o índice de alunos}**

**Para d ← 1 até Disciplinas faça {Laço para percorrer o índice de disciplinas}**

**Escreva (“Informe a nota do aluno ”, a, “ na disciplina ”, d);**

**Leia (notas[a,d]);**

**Fim Para;**

**Fim Para;**

**Escreva (notas[5,1]); {escreve a nota do aluno 5 na disciplina 1}**

**FIM.**



# Estrutura de dados Multidimensional



- Declaração de Matrizes
- Sintaxe:

**{declaração do tipo matriz}**

**<Tidentificador> = matriz [início1 .. fim1, . . . , inícioN .. fimN,] de <tipo primitivo>;**

**:**  
**:**  
**:**

**{declaração da variável do tipo matriz}**

**<Tidentificador> : <identificador1>, ..., <identificador1>;**

**ONDE:**

**Tidentificador = nome da variavel a ser criada.**

**matriz = palavra reservada para indicar que o tipo criado é uma matriz.**

**início/fim = índices iniciais e finais das dimensões da matriz.**

# Algoritmo Matriz



Exemplo:

Caractere: `m[3][5];`

Estrutura que se forma :

	0	1	2	3	4
0					
1					
2					

Para acessar cada elemento da matriz:

```
Leia (m[1][1]);  
m[1][2] ← 'X';  
escreva(m[1][3]);
```

- Declaração de Matrizes
- Semântica:
  - São estruturas de dados homogêneas multidimensionais que permitem agrupar diversas informações dentro de uma variável.
  - Estas correspondem a um grupo de posições contínuas na memória que possuem o mesmo nome e o mesmo tipo de dado e são acessadas por MAIS DE UM índice.
  - Seu tamanho é definido por constantes inteiras e positivas e a definição do seu nome segue as mesmas regras aplicadas para identificadores.

## Note V

A matriz notas ocupa 35 posições de memória (notas de 7 alunos X 5 disciplinas = criar 35 variáveis).

Ressalta-se que a matriz nota não armazena nome de aluno ou disciplina.

## Somente Notas!

E se aumentar de 7  
para 700 alunos e de  
5 para 50 disciplinas?

# Estrutura de dados Multidimensional



## Algoritmo ExemploEstruturaMultidimensional

Alunos = **700**; Disciplinas = **50**; {é só alterar o valor de alunos e disciplinas!}

TNotas = matriz [1..Alunos, 1..Disciplinas] de Real;

Inteiro: a, d;

Início

Para  $a \leftarrow 1$  até Alunos faça

Para  $d \leftarrow 1$  até Disciplinas faça

Escreva (“Informe a nota do aluno ”, a, “ na disciplina ”, d);

Leia (notas[a,d]);

Fim Para;

Fim Para;

Escreva (notas[5,1]);

FIM.

E se acrescentar uma  
nova dimensão?

Notas por  
7 Alunos,  
5 Disciplinas  
e 10 Cursos?



# Estrutura de dados Multidimensional



Algoritmo ExemploEstruturaMultidimensional

Alunos = 7; Disciplinas = 5; **Cursos = 10;**

TNotas = matriz [1..Alunos, 1..Disciplinas, **1.. Cursos**] de Real;

Inteiro: a, d, **c**;

Início

Para a ← 1 até Alunos faça

Para d ← 1 até Disciplinas faça

**Para c ← 1 até Cursos faça**

Escreva (“Informe a nota do aluno”, a, “na disciplina”, d, “no curso”, c);

Leia (notas[a,d,**c**]);

**Fim Para;**

Fim Para;

Fim Para;

FIM.

## Note VI

Cada dimensão da matriz é manipulada por um laço exclusivo. Por isso, ao se adicionar uma nova dimensão, deve-se criar uma nova variável e um novo laço para manipulá-la.

## Note VII

Análogo aos vetores, não é possível operar com todos os elementos da matriz de uma só vez.

Por isso, o correto é operar com cada um de seus elementos isoladamente.

**Lembre-se: um laço para cada índice!**

## Note VIII

Por isso, semelhante aos vetores, o acesso a cada elemento de uma matriz é feito pela manipulação dos seus índices entre [colchetes]!

## Exemplo

nomes[5,1,1]  $\leftarrow$  10,0;

...

Escreva(notas[a,d,c]);

...

Para a  $\leftarrow$  1 até Alunos faça

    Para d  $\leftarrow$  1 até Disciplinas faça

        Para c  $\leftarrow$  1 até Cursos faça

            Leia (notas[a,d,c]);

        Fim Para;

    Fim Para;

Fim Para;

# Estrutura de dados Multidimensional



## EX: Algoritmo ExemploUsoDeMatriz

**{Calcula a média de 10 notas (2 disciplinas) e exibe aquelas acima da média}**

N = 10; D = 2; {número de elementos da matriz}

notaXdis : Matriz [1..N,1..D] de Real;

Inteiro: i, j, soma; Real: média;

Início

soma  $\leftarrow$  0;

Para i  $\leftarrow$  1 até n faça

Para j  $\leftarrow$  1 até d faça

leia(notaXdis [ i, j ]);

soma  $\leftarrow$  soma + notaXdis [ i, j ];

Fim Para;

Fim Para;

media  $\leftarrow$  soma / (N\*D) ;

Para i  $\leftarrow$  1 até n faça

Para j  $\leftarrow$  1 até d faça

Se (notaXdis [ i, j ] > média) Então

Escreva(notaXdis [ i, j ]);

Fim Para;

Fim Para;

Fim.

{Laço para ler e somar as N  
notas X D disciplinas}

{Laço para exibir as N notas X  
D disciplinas que são maiores  
que a média}

## Atividade 2



1) Dadas as matrizes A e B abaixo:

Matriz A

X	U	O
X	D	A
E	A	T
M	N	R
P	S	I
L	O	Z
M	X	E

Matriz B


qual será o conteúdo de B depois de executados os seguintes comandos:

```
Para i ← 1 até 4 faça
  Para j ← 1 até 4 faça
    B[j, i] ← A[i, j];
  Fim Para
Fim Para
aux ← B[1,1];
B[1,1] ← B[3,7];
B[3,7] ← aux;
aux ← B[3,1];
B[3,1] ← B[1,7];
B[1,7] ← aux;
aux ← B[2,2];
B[2,2] ← B[2,5];
B[2,5] ← aux;
```

FIM