

# Processamento de Dados aplicado à Geociências

## AULA 11: Variáveis Compostas Homogêneas Vetores

UNIVERSIDADE FEDERAL DE PELOTAS  
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO  
CURSO SUPERIOR DE TECNOLOGIA EM  
GEOPROCESSAMENTO

Professor: Guilherme Tomaschewski Netto  
[guilherme.netto@inf.ufpel.edu.br](mailto:guilherme.netto@inf.ufpel.edu.br)



# Variáveis Compostas

- Até o momento trabalhamos com três tipos básicos de dados
  - Numéricos
  - Literais
  - Lógicos.
- Com base nestes tipos básicos, podemos definir os tipos compostos

# Variáveis Compostas

- Os tipos compostos dividem-se em:
  - Homogêneos
    - Vetores
    - Matrizes
  - Heterogêneos
    - Registros

# Variáveis Compostas Homogêneas

- As estruturas de dados homogêneas permitem agrupar diversas informações dentro de uma mesma variável.
- Este agrupamento ocorrerá obedecendo sempre ao mesmo tipo de dado, e é por esta razão que estas estruturas são chamadas homogêneas.
- Este tipos de dados recebem diversos nomes, como: variáveis indexadas, variáveis compostas, variáveis subscriptas, arranjos, vetores, matrizes, tabelas em memória ou arrays.

# Variáveis Compostas Homogêneas

- A declaração de variáveis, uma a uma, é insuficiente para resolver um grande número de problemas computacionais.
- Imagine, por exemplo, como faríamos para construir um algoritmo, que lesse os nome de 500 pessoas e imprimisse um relatório destes mesmos nomes.

# Variáveis Compostas Homogêneas

- Não seria uma tarefa simples, pois teríamos que definir 500 variáveis do tipo literal, como é mostrado abaixo:

**Algoritmo**

```
Declare nome1, nome2, ..., nome500 Literal
```

```
...
```

```
Leia nome1, nome2, ..., nome500
```

```
...
```

```
fim_algoritmo
```

# Variáveis Compostas Homogêneas

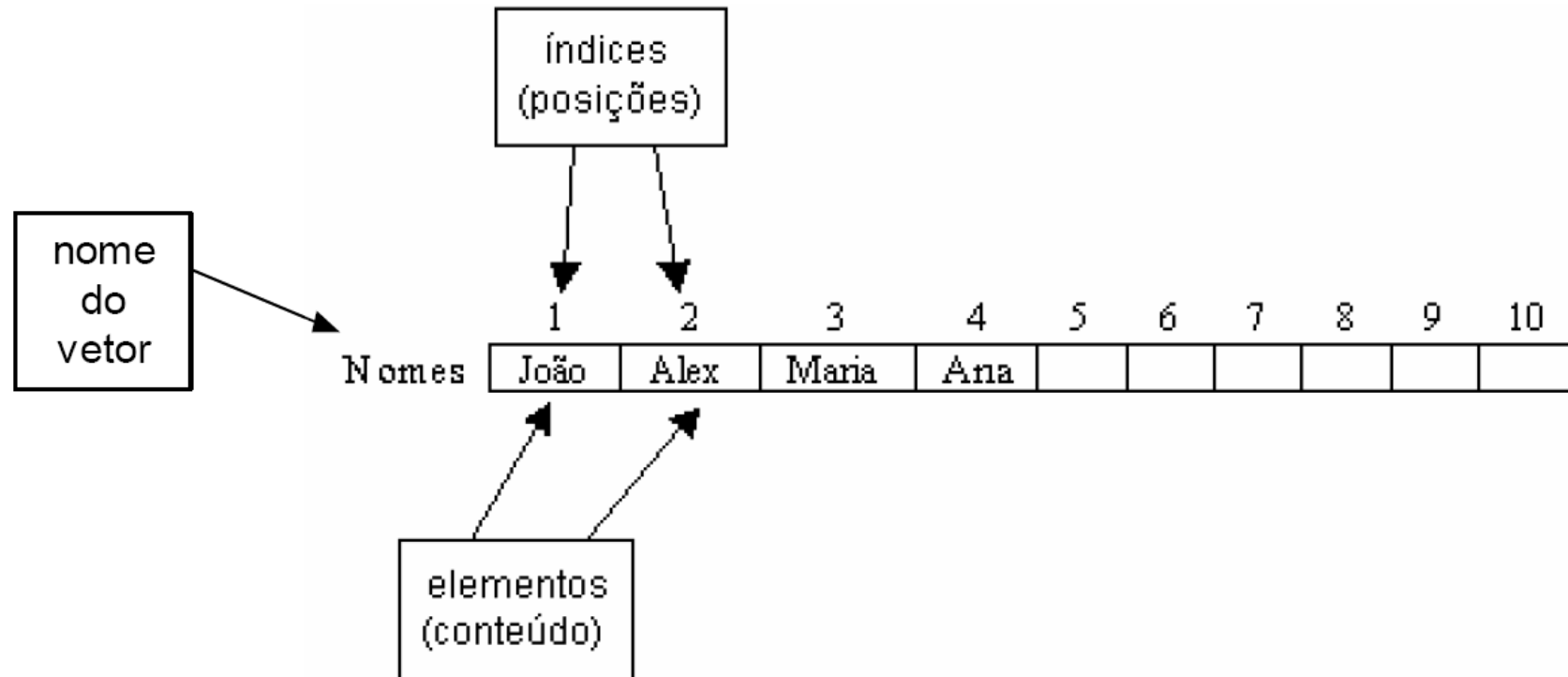
- Para resolver problemas como este, e outros, existem as variáveis indexadas.
- A declaração de uma variável indexada corresponde, na verdade, à declaração de várias variáveis cujo identificador difere apenas por um índice.
- O índice corresponde a um valor numérico começando por 1. Cada variável indexada pode receber valores no decorrer do algoritmo como se fosse uma variável comum.

# Vetores

- Os vetores são também conhecidos como variáveis compostas homogêneas unidimensionais
- Se trata de um conjunto de variáveis de mesmo tipo, que possuem o mesmo identificador (nome).
- Como as variáveis têm o mesmo nome, o que as distingue é um índice que referencia sua localização dentro da estrutura.



# Vetores



# Vetores

- A sintaxe para declaração é:

```
Declare <identificador> (<tamanho>) <tipo>
```

- Exemplos:

```
Declare idade (5) Numérico
```

```
Declare nomes (5) Literal
```

- As declarações acima correspondem à declaração de 10 variáveis:  
idades(1), idades(2), idades(3), idades(4) e idades(5),  
nomes(1), nomes(2), nomes(3), nomes(4), nomes(5).

# Vetores

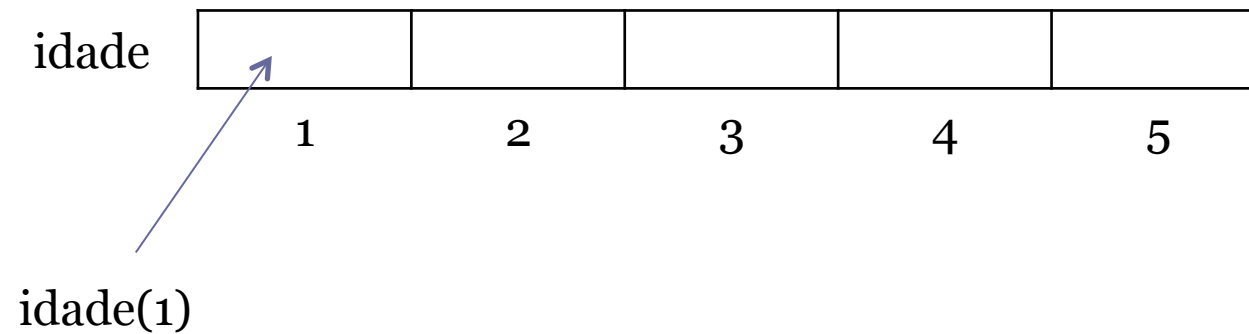
- A sintaxe para atribuição é:

`<identificador> (<posição>) := <expressão>`

- Exemplos:
  - `nomes(1) := “João da Silva”`
  - `idades(1) := 35`
  - `nomes(3) := “Maria Aparecida”`
  - `idades(3) := idades(1)`
  - `i := 5`
  - `idades(i) := 45`

# Vetores

- **Declare** `idade(5)` **Numérico**



# Preenchendo um vetor

- Preencher um vetor significa atribuir valores a todas as suas posições. Assim, deve-se implementar um mecanismo que controle o valor do índice.
- Pode-se, por exemplo, utilizar a estrutura de repetição Para-até-faça para garantir que a variável que controla o índice assuma todos os valores possíveis para o índice do vetor. Assim, para cada execução da repetição, será utilizada uma posição diferente do vetor.

# Preenchendo um vetor

## **Algoritmo**

**Declare** X(5), i **Numerico**

**Para** i **de** 1 **até** 5 **faça**

**Escreva** “Digite o”, i, “º número”, \n

**Leia** X(i)

**fim\_para**

**fim\_algoritmo**

# Preenchendo um vetor - Simulação

i=1	X	95					Digite o 1º número
		1	2	3	4	5	95
i=2	X	95	13				Digite o 2º número
		1	2	3	4	5	13
i=3	X	95	13	-25			Digite o 3º número
		1	2	3	4	5	-25
i=4	X	95	13	-25	47		Digite o 4º número
		1	2	3	4	5	47
i=5	X	95	13	-25	47	0	Digite o 5º número
		1	2	3	4	5	0

# Mostrando os elementos do vetor

- Mostrar os valores contidos em um vetor também implica na utilização do índice.
- Pode-se, por exemplo, também utilizar a estrutura de repetição Para-até-faça para garantir que a variável que controla o índice assuma todos os valores possíveis para o índice do vetor. Assim, para cada execução da repetição, será utilizada uma posição diferente do vetor e, dessa forma, todos os valores do vetor serão mostrados.



# Mostrando os elementos do vetor

## Algoritmo

**Declare** X(5), i **Numerico**

**Para** i **de** 1 **até** 5 **faça** /\*preenchendo o vetor\*/

**Escreva** “Digite o”, i, “º número”, \n

**Leia** X(i)

**fim\_para**

**Para** i **de** 1 **até** 5 **faça** /\*mostrando o vetor\*/

**Escreva** “Este é o”, i, “º número do vetor”, \n

**Escreva** X(i)

**fim\_para**

**fim\_algoritmo**

# Exercício 1

- Faça um algoritmo que, para um vetor de tamanho 100, preencha suas posições com o valor 30.

# Exercício 1 - Solução

**Algoritmo**

**Declare** X(100), i **Numerico**

**Para** i **de** 1 **até** 100 **faça**

X(i):=30

**fim\_para**

**fim\_algoritmo**

## Exercício 2

- Faça um algoritmo que, para um vetor de tamanho 100, preencha suas posições com os números inteiros de 1 a 100

# Exercício 2 - Solução

**Algoritmo**

**Declare** X(100), i **Numerico**

**Para** i **de** 1 **até** 100 **faça**

X(i):=i

**fim\_para**

**fim\_algoritmo**

# Problema

- Fazer um algoritmo que leia uma nota para três alunos e calcule a média destas notas lidas, exibir a média calculada e as notas que estão abaixo desta média.

# 1° Solução

- Ler as três notas uma primeira vez para calcular a média
- Ler novamente cada nota para comparar com a média e verificar quais notas estão abaixo da média

# 1° Solução

## Algoritmo

**Declare** Nota, Media, S, i **Numerico**

S:=0

**Para** i **de** 1 **até** 3 **faça**

**Escreva** “Digite a nota do aluno”, i, \n

**Leia** Nota

    S:=S+Nota

**fim\_para**

Media:=S/3

**Escreva** “Média das notas”, Media, \n

**Para** i **de** 1 **até** 3 **faça**

**Escreva** “Digite a nota do aluno”, i, \n

**Leia** Nota

**Se** Nota<Media

**então Escreva** Nota, “esta abaixo da média”, \n

**fim\_se**

**fim\_para**

**fim\_algoritmo**



# 1º Solução

- INEFICIENTE, pois:
  - deve-se ler as notas uma primeira vez para calcular a média
  - e ler novamente cada nota para comparar com a média e verificar quais notas estão abaixo da média
- Além disso a informação de cada nota é perdida a cada nova leitura, pois utiliza-se uma única variável

## 2° Solução

- Utilizar uma variável para cada nota

## 2° Solução

### Algoritmo

```
Declare Nota1, Nota2, Nota3, Media Numerico  
Escreva “Digite a nota dos 3 alunos”, i, \n  
Leia Nota1, Nota 2, Nota 3  
Media:=(Nota1+Nota 2+Nota 3)/3  
Escreva “Média das notas”, Media, \n  
Se Nota1<Media  
    então Escreva Nota1, “esta abaixo da média”, \n  
fim_se  
Se Nota2<Media  
    então Escreva Nota2, “esta abaixo da média”, \n  
fim_se  
Se Nota3<Media  
    então Escreva Nota3, “esta abaixo da média”, \n  
fim_se  
fim_algoritmo
```

## 2° Solução

- PROBLEMA, pois
  - deve-se manter uma variável para cada nota
- O algoritmo pode se tornar impraticável a medida que aumentamos o número de variáveis
  - Exemplo, calcular a média para 100 alunos

## 3° Solução

- Associar o nome Nota ao conjunto ordenado de notas
  - $\text{Nota} = (N_1, N_2, \dots, N_{100})$
- Uma abordagem mais realística!

## 3° Solução

### Algoritmo

```
Declare Nota(100), Media, S, i Numerico  
S:=0  
Para i de 1 até 100 faça  
    Escreva “Digite a nota do aluno”, i, \n  
    Leia Nota(i)  
    S:=S+Nota(i)  
fim_para  
Media:=S/100  
Escreva “Média das notas”, Media, \n  
Para i de 1 até 100 faça  
    Se Nota(i)<Media  
        então Escreva Nota(i), “esta abaixo da média”, \n  
    fim_se  
fim_para  
fim_algoritmo
```

## Exercício 3

- Faça um algoritmo que leia o nome e a nota de 10 alunos, calcule a média das notas e escreva os nomes dos alunos com notas maiores que a média.

## Exercício 3 - Solução

### Algoritmo

```
Declare Nota(10), Media, S, i Numerico  
Declare Nome(10) Literal  
S:=0  
Para i de 1 até 10 faça  
    Escreva “Digite o nome do aluno”, i, \n  
    Leia Nome(i)  
    Escreva “Digite a nota do aluno”, i, \n  
    Leia Nota(i)  
    S:=S+Nota(i)  
fim_para  
Media:=S/10  
Escreva “Média das notas”, Media, \n  
Para i de 1 até 10 faça  
    Se Nota(i)>Media  
        então Escreva Nome(i), “esta acima da média”, \n  
    fim_se  
fim_para  
fim_algoritmo
```



## Exercício 4

- Faça um algoritmo que leia a nota de 10 alunos, calcule a média das notas e escreva a quantidade de alunos com notas maiores que a média.

# Exercício 4 - Solução

## Algoritmo

**Declare** Nota(10), Media, S, i, Quant **Numerico**

S:=0

Quant:=0

**Para** i **de** 1 **até** 10 **faça**

**Escreva** “Digite a nota do aluno”, i, \n

**Leia** Nota(i)

    S:=S+Nota(i)

**fim\_para**

Media:=S/10

**Escreva** “Média das notas”, Media, \n

**Para** i **de** 1 **até** 10 **faça**

**Se** Nota(i)>Media

**então** Quant:=Quant+1

**fim\_se**

**fim\_para**

**Escreva** “Quantidade de alunos com nota acima da média”, Quant, \n

**fim\_algoritmo**