

# Matrizes



## **Introdução à Programação para Biologia Molecular**

Rosane Minghim

Apoio na confecção: Danilo Medeiros Eler

Rogério Eduardo Garcia

Renato Rodrigues

Carlos E. A. Zampieri

# Recordando: Vetores

- Tipos Compostos
  - Conjunto de vários elementos de mesmo tipo
- Um único nome. Ex: `V`
- Um tipo. Ex: `V:inteiro`



- Armazenamento em sequência. `V:inteiro[1..7]`
- Um índice.
  - Acesso direto: `V[i]`

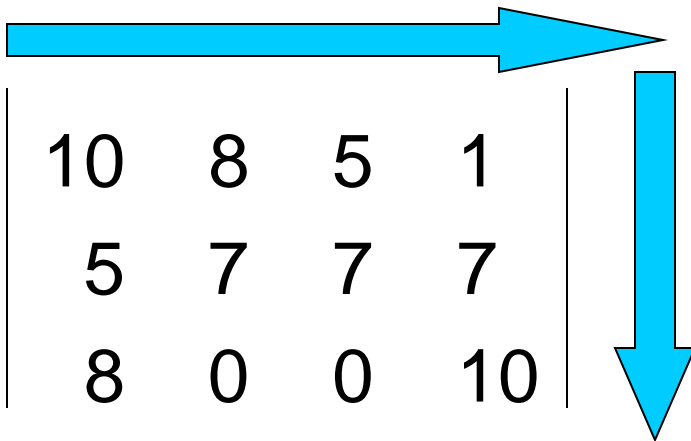
# Matrizes

- Tipos Compostos
  - Conjunto de vários elementos de mesmo tipo
- Um único nome. Ex: M
- Um tipo. Ex: M:inteiro

$$M = \begin{vmatrix} 10 & 8 & 5 & 1 \\ 5 & 7 & 7 & 7 \\ 8 & 0 & 0 & 10 \end{vmatrix}$$

# Matrizes

- A indexação é similar à dos vetores
- Armazenamento em sequência para cada dimensão.  $M:\text{inteiro}[1..3][1..4]$



M =

10	8	5	1
5	7	7	7
8	0	0	10

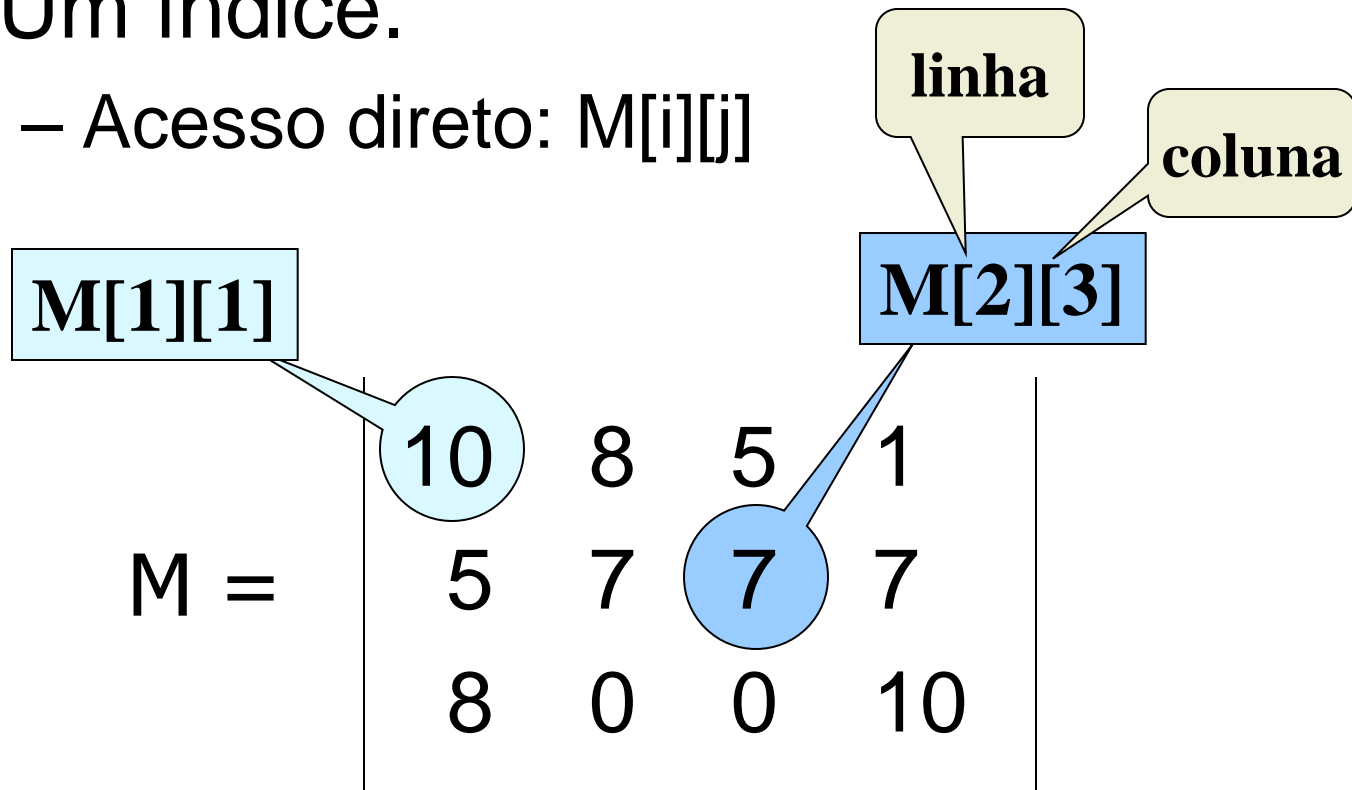
Genericamente, para  $n$  dimensões:

$M:\text{inteiro}[l_{i1},l_{s1}] [l_{i2},l_{s2}] \dots [l_{in},l_{sn}]$

# Matrizes

- Um índice.

- Acesso direto:  $M[i][j]$





# Exemplos

```
tipo  matriz = real[1..10][1..5]
```

```
variável
```

```
    notas: matriz
```

```
notas[1][3] ← 7,5
```

No exemplo a matriz possui duas dimensões,  
uma variando de 1 a 10 e a segunda de 1 a 5

# Exemplo: média aritmética

Faça um algoritmo para ler as 5 notas dos alunos de uma turma, armazená-las numa matriz, e calcular suas médias finais.

```
Algoritmo média_provas
```

```
tipo matriz =  
    real[1..50][1..5]
```

```
variável
```

```
    soma, media: real
```

```
    notas: matriz
```

```
    lin, col, nro_alunos:  
    inteiro
```

```
leia(nro_alunos)
```

```
para lin de 1 até  
nro_alunos faça
```

```
    para col de 1 até 5 faça
```

```
        leia(notas[lin][col])
```

```
    fim para
```

```
fim para
```

```
para lin de 1 até  
nro_alunos faça
```

```
    soma ← 0
```

```
    para col de 1 até 5 faça
```

```
        soma ←
```

```
        notas[lin][col] +soma
```

```
    fim para
```

```
    media ← soma / 5
```

```
    escreva('aluno', lin, '  
    = ',media)
```

```
fim para
```

```
fim
```

# Exemplo: média aritmética

```
Algoritmo média_provas
tipo matriz =
  real[1..50][1..5]
variável
  soma, media: real
  notas: matriz
  lin, col, nro_alunos:
    inteiro

  leia(nro_alunos)
  para lin de 1 até
    nro_alunos faça
      para col de 1 até 5 faça
        leia(notas[lin][col])
      fim para
    fim para
```

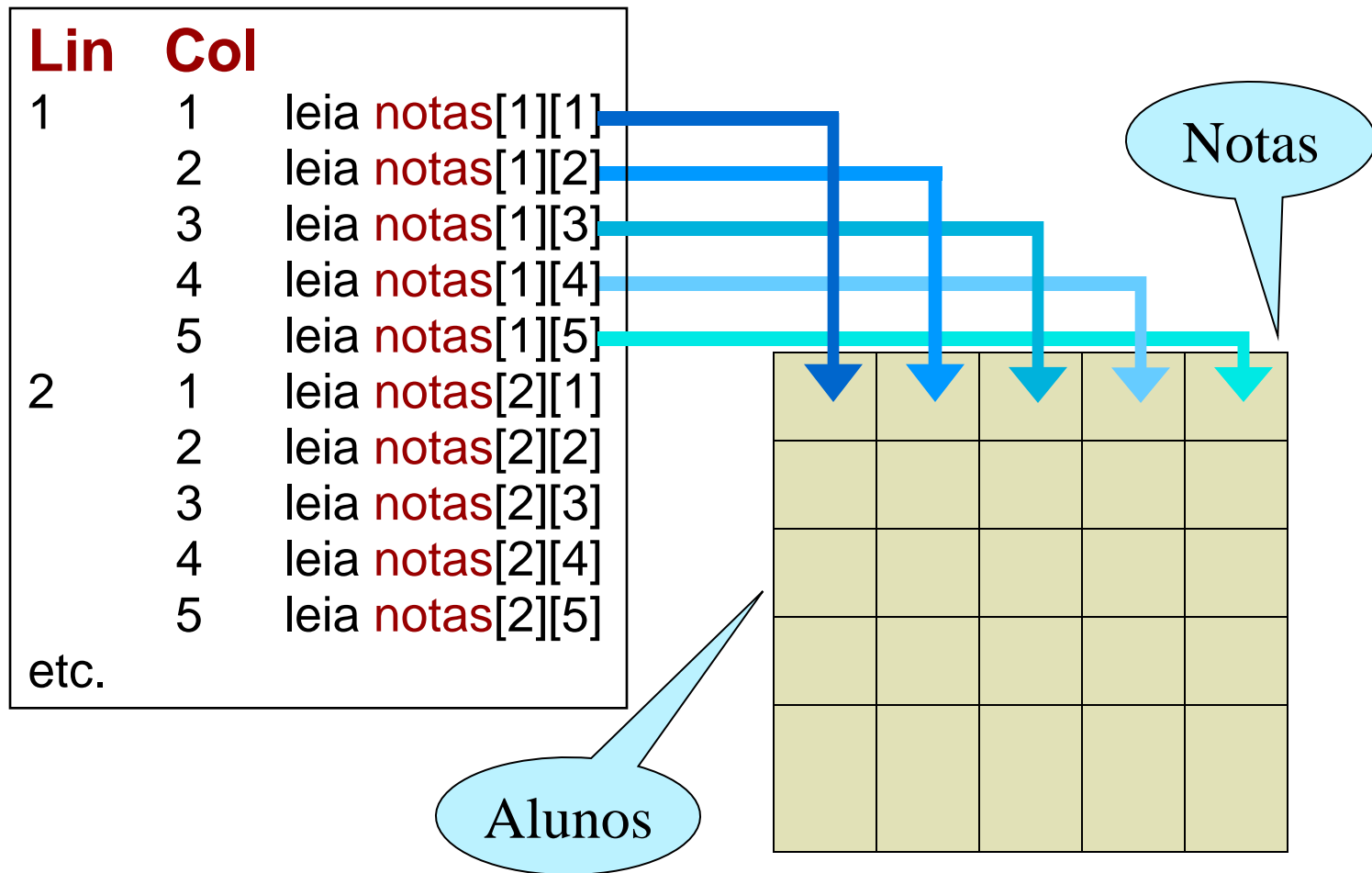
Alunos

Notas

Lin	Col	
1	1	leia notas[1][1]
	2	leia notas[1][2]
	3	leia notas[1][3]
	4	leia notas[1][4]
	5	leia notas[1][5]
2	1	leia notas[2][1]
	2	leia notas[2][2]
	3	leia notas[2][3]
	4	leia notas[2][4]
	5	leia notas[2][5]
etc.		



# Exemplo: média aritmética





# Matrizes em Subprogramas

- Similar ao uso de vetores
- É importante para o uso de matrizes em subprogramas que o seu tipo seja definido previamente, por uma questão de consistência, compatibilidade de tipos e organização do código



# Declaração de Tipos

```
mat_notas = real[1..50][1..5]  
objeto_logico = lógico[-10..10][-10..10][-10..10]
```

- Desta forma, subprogramas e algoritmos principais podem trabalhar sobre matrizes de mesmo tipo pré-definido
- Utilizando as definições acima, as declarações ficariam:

Variável

```
nota: mat_notas  
objeto: objeto_logico
```

# Matrizes em subprogramas - Exemplo

O subprograma recebe a matriz e armazena os valores digitados.

```
Subprograma leia_matriz(mat, linhas, colunas)
```

```
e: linhas, colunas: inteiro {dimensões da matriz}
```

```
s: mat: matriz {valores lidos}
```

```
variável
```

```
    lin, col: inteiro
```

```
inicio
```

```
    para lin de 1 até linhas faça
```

```
        para col de 1 até colunas faça
```

```
            leia(mat[lin][col])
```

```
        fim para
```

```
    fim para
```

```
fim
```



# Matrizes em subprogramas - Exemplo

O subprograma recebe a matriz apresenta o conteúdo.

```
Subprograma escreva_matriz(mat, linhas,colunas)
```

```
e: linhas, colunas: inteiro {dimensões da matriz}  
   mat: matriz {notas}
```

```
variável
```

```
   lin, col: inteiro
```

```
inicio
```

```
   para lin de 1 até linhas faça
```

```
     para col de 1 até colunas faça
```

```
       escreva(mat[lin][col])
```

```
     fim para
```

```
   fim para
```

```
fim
```

# Matrizes em subprogramas - Exemplo

**O subprograma recebe a matriz e calcula as médias das linhas, armazenando o resultado num vetor.**

```
Subprograma calcule_médias (mat, linhas, colunas, vet)
```

```
e: linhas, colunas: inteiro {dimensões da matriz}
```

```
e: mat: matriz {notas}
```

```
s: vet: vetor {medias}
```

```
variável
```

```
  lin, col: inteiro
```

```
inicio
```

```
  para lin de 1 até linhas faça
```

```
    vet[lin] ← 0
```

```
    para col de 1 até colunas faça
```

```
      vet[lin] ← vet[lin] + mat[lin][col]
```

```
    fim para
```

```
    vet[lin] ← vet[lin] / colunas
```

```
  fim para
```

```
fim
```



# Matrizes em subprogramas - Exemplo

Algoritmo Principal:

Algoritmo média\_provas

tipo

```
matriz = real[1..50][1..5]
```

```
vetor = real[1..50]
```

variável

```
notas: matriz
```

```
medias: vetor
```

```
nro_alunos: inteiro
```

```
leia(nro_alunos)
```

```
leia_matriz(notas,nro_alunos,5)
```

```
calcule_media(notas,nro_alunos,5, medias)
```

```
escreva_notas(notas,nro_alunos,5, medias)
```

```
fim para
```

**Exercício:** desenvolver o subprograma escreva\_notas



# Matrizes em subprogramas - Exemplo

O subprograma recebe a matriz apresenta o conteúdo.

```
Subprograma escreva_notas(mat, linhas,colunas,vet)
```

```
e: linhas, colunas: inteiro {dimensões da matriz}
   mat: matriz {notas}
   vet: vetor {medias}
```

```
variável
```

```
   lin, col: inteiro
```

```
inicio
```

```
   para lin de 1 até linhas faça
```

```
     para col de 1 até colunas faça
```

```
       escreva(mat[lin][col])
```

```
     fim para
```

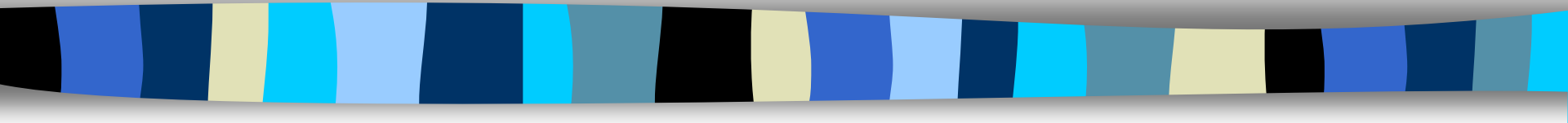
```
   escreva (vet[lin])
```

```
   fim para
```

```
fim
```



# Matrizes em PYTHON





# Matrizes em Python

- Em Python não existe uma estrutura específica para matrizes
- Para simular uma matriz usamos listas encadeadas, ou seja, listas de listas
- Python admite conjuntos de dados heterogêneos com indexação em várias dimensões
- Os formatos para declaração de variável multidimensional são apresentados a seguir

```
matriz = [ ]
```

```
matriz = [ [ ], [ ], [ ] ]
```

ou

```
matriz = [ [1,2,3], [4,5,6], [7,8,9] ]
```

# Exemplo – Média Aritmética

```
#Algoritmo media_provas

nro_notas = 5
notas = []
nro_alunos = input('Digite o numero de alunos: ')

for lin in range(nro_alunos):
    provas = []
    for col in range(nro_notas):
        valor = input('Digite a nota: ')
        provas.append(valor)
    notas = notas + [provas]

for lin in range(nro_alunos):
    soma = 0
    for col in range(5):
        soma = soma + notas[lin][col]
    media = soma / nro_notas
    print 'aluno ',lin, ' = ',media
```

# Python – Matriz em Subprogramas

- Em Python não existe passagem por referência, assim para que uma matriz seja modificada em um subprograma ela deve ser retornada como resultado da função
- Exemplo:

```
#programa EXEMPLO
```

```
def
```

```
alt_value(mat, linha, coluna, valor) :  
    mat[linha][coluna] = valor  
    return mat
```

```
matriz = alt_value (matriz,1,1,10)
```



# MATRIZ - Exercício em Classe

- Escrever um algoritmo que leia duas matrizes e calcule a soma delas armazenando em uma terceira matriz. **Apresente a matriz resultante.**



# MATRIZ - Exercício em Classe

- Escrever um algoritmo para o seguinte problema:

Ler uma matriz  $A$  de dimensão  $N \times N$  ( $N \leq 50$ ) e verificar se a matriz é simétrica. Escrever a matriz e a mensagem de resposta

Matrize

# FAMM

Introdução à Biologia Molecular

Rosane Minghim

apoio na confecção: Danilo Medeiros Eler

Rogério Eduardo Garcia

Renato Rodrigues

Carlos E. A. Zampieri

Baseado na Apostila: Curso Introductório de Computação por R. Minghim e G. P. Telles