

Algoritmos e Técnicas de Programação



Aula 4 - Matrizes
Prof. Guilherme Guerino

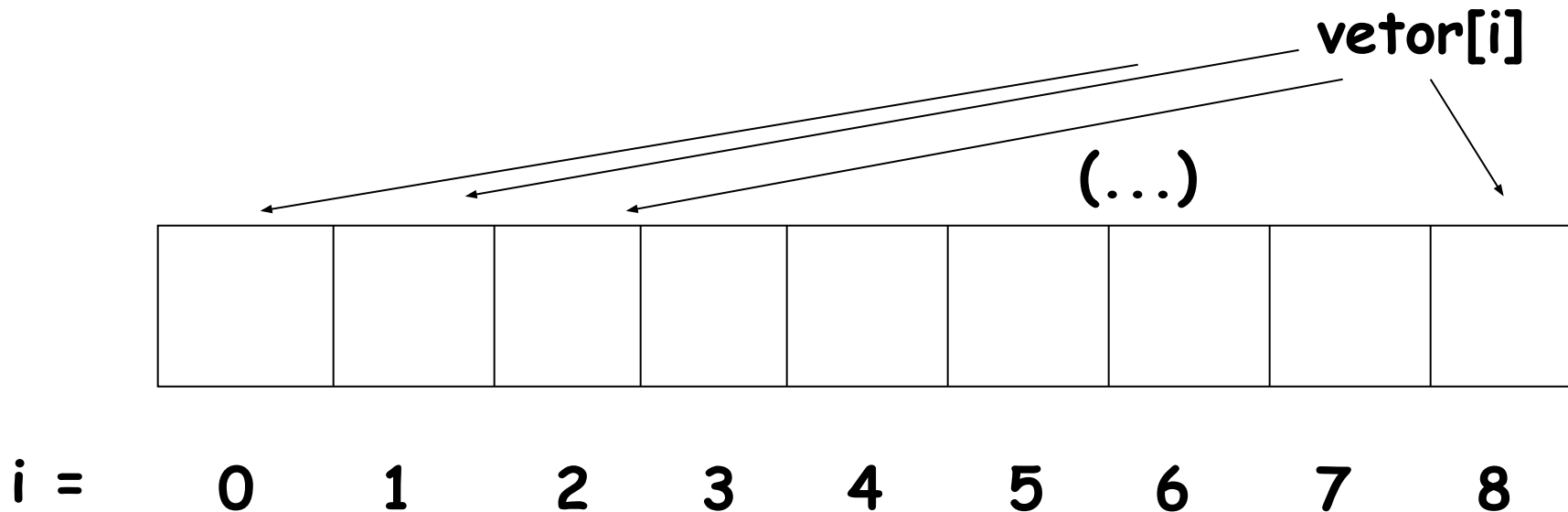


Revisão

- O tópico anterior considerou a estruturação por **vetores**.
- A utilização de vetores permite armazenar dados do mesmo tipo, em múltiplas posições de memória, utilizando somente **um identificador**.



Revisão





Observação

- **Outras estruturas** podem ser utilizadas além dos vetores.



Introdução

- Considere o seguinte programa a ser implementado
 - Implemente um programa que **receba 5 notas de 7 alunos**, que calcule e armazene a média simples de cada aluno.
 - Os alunos são identificados com números entre 0 e 6. Por fim, o programa deve **apresentar as notas e as médias**.



Introdução

- Dado o problema apresentado, quantas variáveis são necessárias para armazenar as notas e as médias dos alunos?
 - 5?, 6?
 - Mais variáveis
 - Menos variáveis
 - Nenhuma das opções citadas?



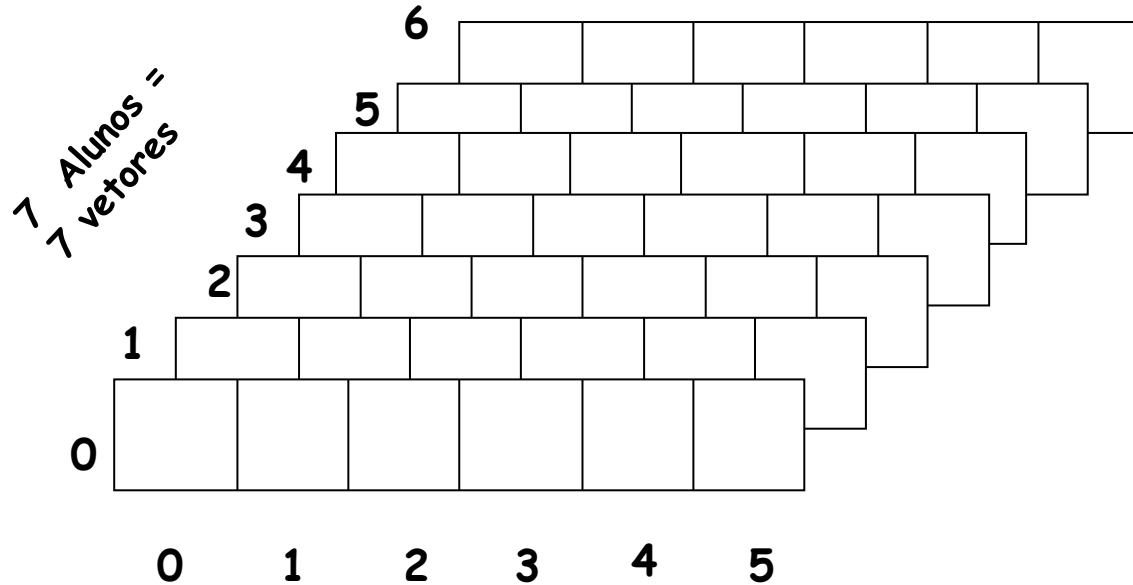
Introdução

- Dado o problema apresentado, quantas variáveis são necessárias para armazenar as notas e as médias dos alunos?

- Resposta
 - 42 variáveis simples ou
 - 7 vetores (um por aluno), cada qual com 6 elementos (5 notas + média)



Introdução



5 Notas + Média = 6 elementos por vetor



Introdução

- Para o problema apresentado, tem-se uma **possível solução** que considera 7 vetores.
- Apesar dos vetores facilitarem a resolução de problemas, ainda existem situações onde **outras estruturas se mostram mais adequadas**.
- No caso do problema apresentado, **matrizes** são mais adequadas e poderiam ser utilizadas.



Matriz

- Matrizes podem ser entendidas como “**vários vetores do mesmo tipo**”.
- Enquanto os vetores são entendidos como *agregados homogêneos unidimensionais*, as matrizes são entendidas como **agregados homogêneos multidimensionais**.



Matriz

- Um **arranjo multidimensional** é o arranjo que utiliza **múltiplos índices** para referenciar seus elementos.
- Na disciplina, matrizes bidimensionais são consideradas. Logo, para acessar as células da matriz, **dois índices** são necessários.



Matriz

- Apresentação de uma matriz bidimensional:

$$\begin{bmatrix} \text{notas}[0][0] & \text{notas}[0][1] & \dots & \text{notas}[0][5] \\ \text{notas}[1][0] & \text{notas}[1][1] & \dots & \text{notas}[1][5] \\ \text{notas}[2][0] & \text{notas}[2][1] & \dots & \text{notas}[2][5] \\ & \cdot & & \\ & \cdot & & \\ & \cdot & & \\ \text{notas}[6][0] & \text{notas}[6][1] & \dots & \text{notas}[6][5] \end{bmatrix}$$



Matriz

Exemplo: matriz bidimensional armazena 6 valores para 7 ALUNOS. Cinco NOTAS e a MÉDIA são armazenados.

Representação
espacial

Cada elemento desta
matriz necessita de
dois índices para ser
especificado--> notas
é portanto uma
matriz bidimensional

	0	1	2	3	4	5
0						
1						
2						
3		5.4				
4						
5						
6						

`float notas[3] [1]`



Matriz

- Em uma matriz **bidimensional**, tem-se que os índices representam linhas e colunas.
- **Linhas (vertical)**
 - primeiro índice
- **Colunas (horizontal)**
 - segundo índice



Matriz

- Independente do número de dimensões de uma matriz, os elementos de um arranjo **são todos do mesmo tipo**.
- O **número de dimensões** corresponde ao **número de índices** necessários para acessar um elemento da matriz.
-



Matriz

- Independente do número de dimensões de uma matriz, os elementos de um arranjo **são todos do mesmo tipo**.
- O **número de dimensões** corresponde ao **número de índices** necessários para acessar um elemento da matriz.
-



Matriz

Simulação: preenchimento de uma matriz



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 0

j --> 0
m[0][0] = 9

	0	1	2
0	9		
1			
2			



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 0

j --> 1
m[0][1] = 5

	0	1	2
0	9	5	
1			
2			



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 0

j --> 2
m[0][2] = 7

	0	1	2
0	9	5	7
1			
2			



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 0

j --> 3

	0	1	2
0	9	5	7
1			
2			



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 1

j --> 0
m[1][0] = 11

	0	1	2
0	9	5	7
1	11		
2			



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 1

j --> 1
m[1][1] = 4

	0	1	2
0	9	5	7
1	11	4	
2			



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 1

j --> 2
m[1][2] = 1

	0	1	2
0	9	5	7
1	11	4	1
2			



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 1

j --> 3

	0	1	2
0	9	5	7
1	11	4	1
2			



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 2

j --> 0
m[2][0] = 6

	0	1	2
0	9	5	7
1	11	4	1
2	6		



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 2

j --> 1
m[2][1] = 3

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 2

j --> 2
m[2][2] = 2

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 2

j --> 3

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        scanf("%d", &m[i][j]);
```

Preenchimento da matriz m

i --> 3

j --> 3

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

Simulação: apresentação de elementos de uma matriz



Matriz

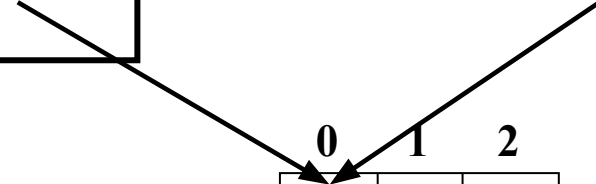
```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 0

j --> 0

9



	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

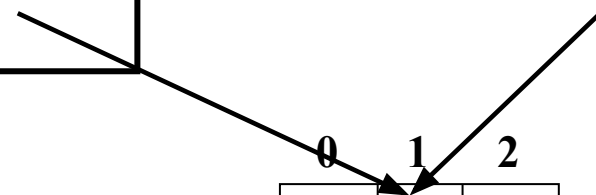
```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 0

j --> 1

9 5



	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 0

j --> 2

9 5 7

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 0

j --> 3

9 5 7

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 1

j --> 0

9 5 7
11

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 1

j --> 1

9 5 7
11 4

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 1

j --> 2

9 5 7
11 4 1

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 1

j --> 3

```
9 5 7  
11 4 1
```

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 2

j --> 0

9 5 7
11 4 1
6

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 2

j --> 1

9 5 7
11 4 1
6 3

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 2

j --> 2

9	5	7
11	4	1
6	3	2

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 2

j --> 3

```
9 5 7  
11 4 1  
6 3 2
```

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2



Matriz

```
int m[3][3];  
  
for (i = 0; i < 3; i++)  
    for (j = 0; j < 3; j++)  
        printf("%d", m[i][j]);
```

Preenchimento da matriz m

i --> 3

j --> 3

```
9 5 7  
11 4 1  
6 3 2
```

	0	1	2
0	9	5	7
1	11	4	1
2	6	3	2

Dúvidas?

