

CONSTRUÇÃO DE ALGORITMOS

Bacharelado em Sistemas da Informação

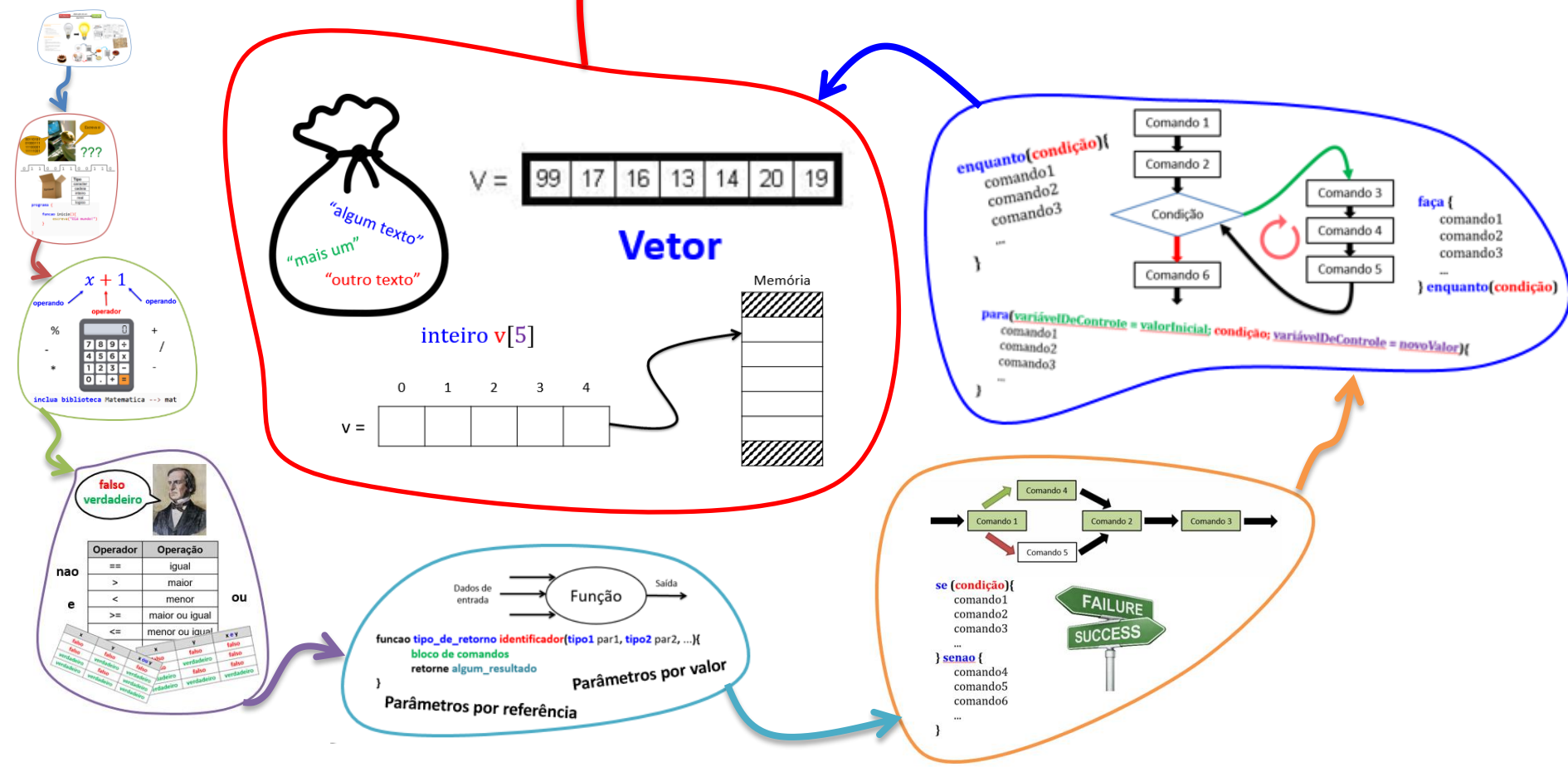
Prof. Marco André Abud Kappel

Aula 9 – Matrizes

Matrizes

• Anteriormente:

Hoje:
Matrizes



Matrizes

- **Arranjos**

- Vimos, anteriormente, que existem estruturas capazes de armazenar **vários valores**: os **arranjos**.
- Desta forma, podemos usar uma **única variável** para armazenar vários valores ao mesmo tempo.



Atenção! Todos os valores precisam ser do **mesmo tipo**!



Matrizes

- Arranjos

- Estudaremos **dois tipos** de arranjos: o arranjo **unidimensional** e o arranjo **bidimensional**.

V =

99	17	16	13	14	20	19
----	----	----	----	----	----	----

Vetor

M =

41	31	21	51	61	21	31	41
11	11	11	11	11	11	11	11
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
10	10	10	10	10	10	10	10
40	30	20	50	60	20	30	40

Matriz

Matrizes

- Arranjos

- Estudaremos **dois tipos** de arranjos: o arranjo **unidimensional** e o arranjo **bidimensional**.

V =

99	17	16	13	14	20	19
----	----	----	----	----	----	----

Vetor

Veremos agora

M =

41	31	21	51	61	21	31	41
11	11	11	11	11	11	11	11
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
10	10	10	10	10	10	10	10
40	30	20	50	60	20	30	40

Matriz

Matrizes

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

Matrizes

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

0	1	2	3	4

← Cada coluna na matriz possui um **índice**.

Matrizes

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

	0	1	2	3	4
0					
1					
2					
3					

Cada coluna na matriz possui um **índice**.

Cada linha na matriz também possui um **índice**.

Matrizes

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

Assim como nos outros casos, cada matriz terá um nome.

Cada linha na matriz também possui um **índice**.

M =

	0	1	2	3	4
0					
1					
2					
3					

Cada coluna na matriz possui um **índice**.

Matrizes



Atenção! Todos os valores precisam ser do mesmo tipo!

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

MatrizDeInteiros =

	0	1	2	3	4
0	4	2	5	45	2
1	3	63	45	44	23
2	233	2	52	7	2
3	4	6	1	45	7

Matrizes



Atenção! Todos os valores precisam ser do mesmo tipo!

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

MatrizDeInteiros =

	0	1	2	3	4
0	4	2	5	45	2
1	3	63	45	44	23
2	233	2	52	7	2
3	4	6	1	45	7



Matrizes



Atenção! Todos os valores precisam ser do mesmo tipo!

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

MatrizDeCarac =

	0	1	2	3	4
0	'a'	'b'	'c'	'd'	'e'
1	'f'	'g'	'h'	'i'	'j'
2	'k'	'l'	'm'	'n'	'o'
3	'p'	'q'	'r'	's'	't'

Matrizes



Atenção! Todos os valores precisam ser do mesmo tipo!

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

MatrizDeCarac =

	0	1	2	3	4
0	'a'	'b'	'c'	'd'	'e'
1	'f'	'g'	'h'	'i'	'j'
2	'k'	'l'	'm'	'n'	'o'
3	'p'	'q'	'r'	's'	't'

Matrizes



Atenção! Todos os valores precisam ser do mesmo tipo!

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

MatrizDeCarac =

	0	1	2	3	4
0	'a'	'b'	'c'	'd'	'e'
1	'f'	'g'	'h'	'i'	'j'
2	'k'	'l'	'm'	'n'	'o'
3	'p'	'q'	'r'	's'	't'



Matrizes



Atenção! Todos os valores precisam ser do mesmo tipo!

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

MinhaMatriz =

	0	1	2	3	4
0	'a'	'b'	'c'	'd'	'e'
1	4	2	5	45	2
2	'k'	'l'	'm'	'n'	'o'
3	1	4	22	10	97

Matrizes



Atenção! Todos os valores precisam ser do mesmo tipo!

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

MinhaMatriz =

	0	1	2	3	4
0	'a'	'b'	'c'	'd'	'e'
1	4	2	5	45	2
2	'k'	'l'	'm'	'n'	'o'
3	1	4	22	10	97

Matrizes



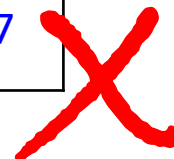
Atenção! Todos os valores precisam ser do mesmo tipo!

- **Estrutura**

- Uma matriz é constituída por uma série bidimensional de elementos.

MinhaMatriz =

	0	1	2	3	4
0	'a'	'b'	'c'	'd'	'e'
1	4	2	5	45	2
2	'k'	'l'	'm'	'n'	'o'
3	1	4	22	10	97



Matrizes

- **Sintaxe**

- Para criar uma matriz, é necessário fazer uma **declaração**.

tipo identificador[numDeLinha][numDeColuna]

Matrizes

- **Sintaxe**

- Para criar uma matriz, é necessário fazer uma **declaração**.

tipo **identificador**[numDeLinha][numDeColuna]



Um dos tipos possíveis:

caracter cadeia inteiro
real logico

Matrizes

- **Sintaxe**

- Para criar uma matriz, é necessário fazer uma **declaração**.

tipo identificador[numDeLinha][numDeColuna]

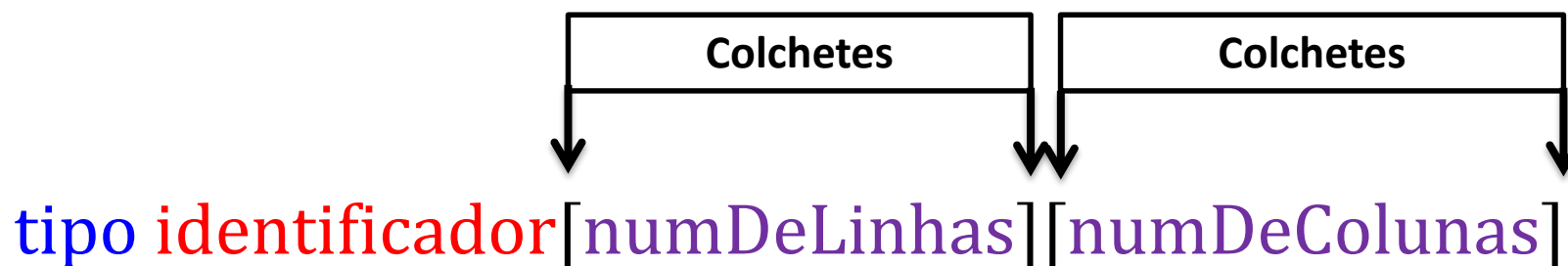
Um dos tipos possíveis:
caracter cadeia inteiro
real logico

Qualquer identificador
seguindo as **3 restrições**.

Matrizes

- **Sintaxe**

- Para criar uma matriz, é necessário fazer uma **declaração**.


tipo identificador[numDeLinha][numDeColuna]

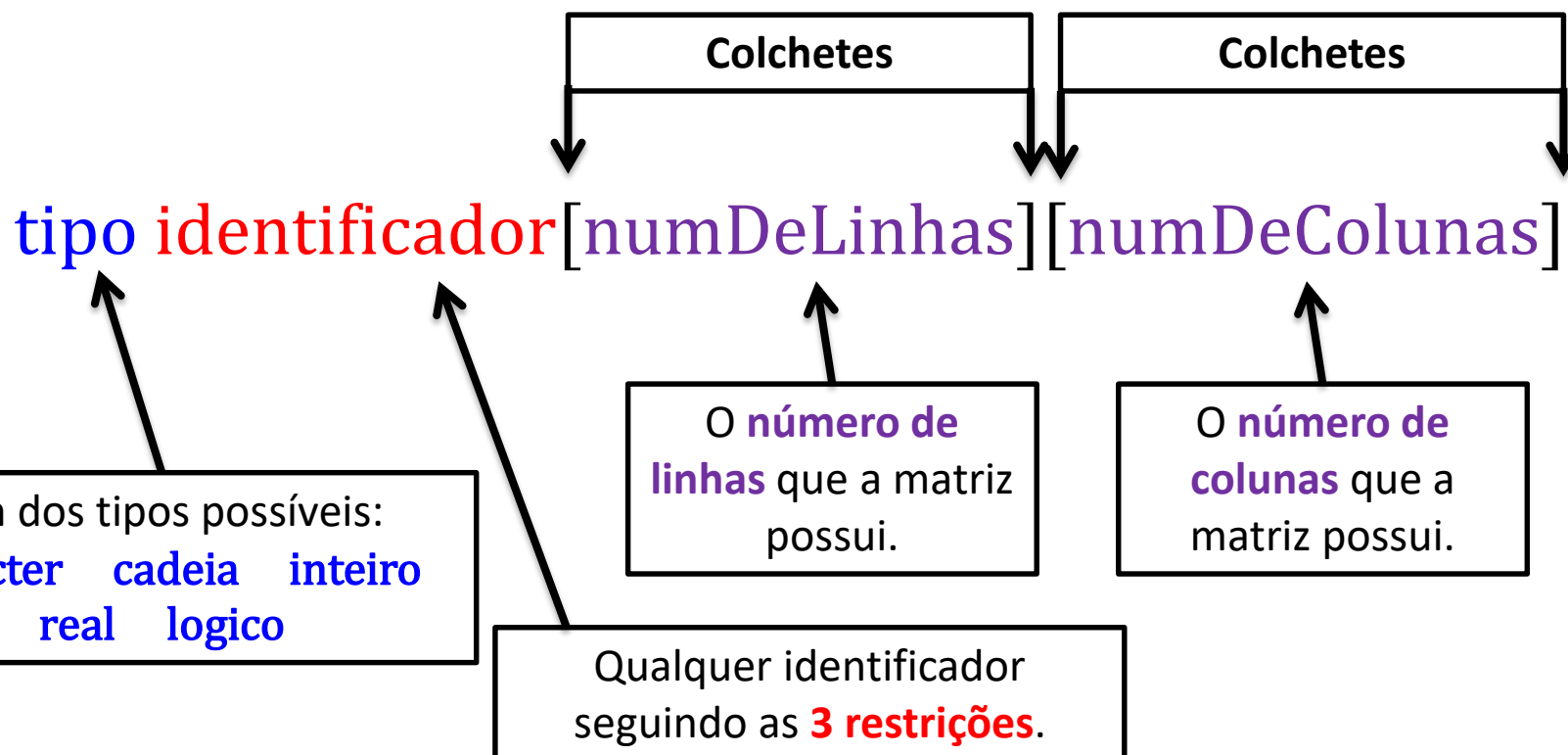
Um dos tipos possíveis:
caracter cadeia inteiro
real logico

Qualquer identificador
seguindo as **3 restrições**.

Matrizes

- **Sintaxe**

- Para criar uma matriz, é necessário fazer uma **declaração**.



Matrizes

- **Exemplo**

- A matriz, assim como o vetor, é uma estrutura de dados **estática**, ou seja, que não pode aumentar ou diminuir o seu número de linhas ou colunas e estes números devem ser especificados a priori.
- Por exemplo, esta matriz foi criada com o **identificador M**, 2 linhas e 4 colunas.

inteiro **M**[2][4]

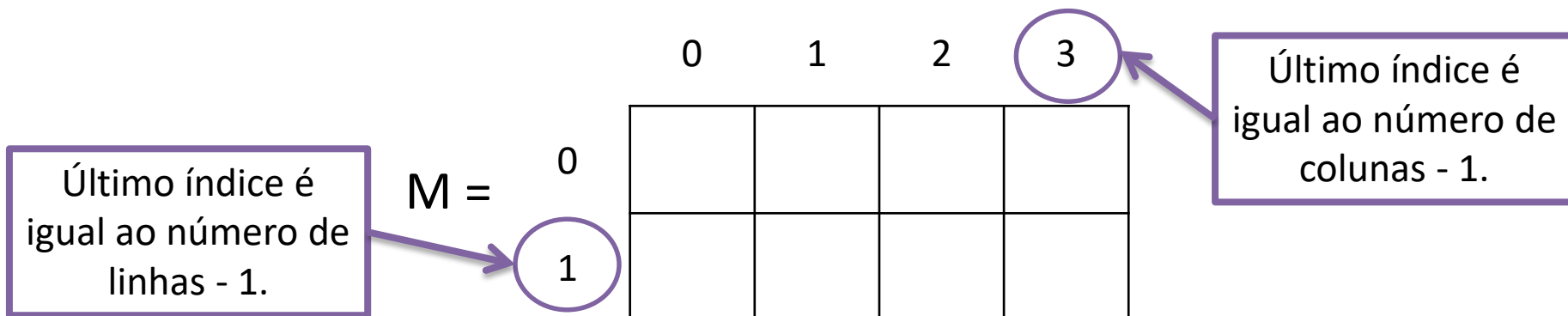
	0	1	2	3
0				
1				

Matrizes

- **Exemplo**

- Cada elemento da matriz poderá ser **acessado diretamente** a partir de um índice de **linha** e um de **coluna**.
- Os **últimos índices** da matriz serão o número de linhas (ou colunas) **decrecidos** de uma unidade.

inteiro $M[2][4]$



- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

M =

	0	1	2
0			
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]  
M[0][0] = 1
```

M =

	0	1	2
0			
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]  
M[0][0] = 1
```

M =

	0	1	2
0	1		
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]  
M[0][0] = 1  
M[0][1] = 2
```

M =

	0	1	2
0	1		
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]  
M[0][0] = 1  
M[0][1] = 2
```

M =

	0	1	2
0	1	2	
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]  
M[0][0] = 1  
M[0][1] = 2  
M[0][2] = 3
```

M =

	0	1	2
0	1	2	
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]  
M[0][0] = 1  
M[0][1] = 2  
M[0][2] = 3
```

M =

	0	1	2
0	1	2	3
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

M =	0	1	2
0	1	2	3
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

M =

	0	1	2
0	1	2	4
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

M =

	0	1	2
0	1	2	4
1			
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

M =

	0	1	2
0	1	2	4
1	12		
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

M =

	0	1	2
0	1	2	4
1	12		
2			
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

M =

	0	1	2
0	1	2	4
1	12		
2	10		
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

```
M[3][0] = 5
```

M =

	0	1	2
0	1	2	4
1	12		
2	10		
3			

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

```
M[3][0] = 5
```

M =

	0	1	2
0	1	2	4
1	12		
2	10		
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

```
M[3][0] = 5
```

```
M[1][0] = 6
```

M =

	0	1	2
0	1	2	4
1	12		
2	10		
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

```
M[3][0] = 5
```

```
M[1][0] = 6
```

M =

	0	1	2
0	1	2	4
1	6		
2	10		
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

```
M[3][0] = 5
```

```
M[1][0] = 6
```

```
M[2][2] = 1
```

M =

	0	1	2
0	1	2	4
1	6		
2	10		
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

```
M[3][0] = 5
```

```
M[1][0] = 6
```

```
M[2][2] = 1
```

M =

	0	1	2
0	1	2	4
1	6		
2	10		1
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
```

M =

	0	1	2
0	1	2	4
1	6		
2	10		1
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
```

M =

	0	1	2
0	1	2	4
1	6		5
2	10		1
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
M[2][1] = M[0][0] + 1
```

M =

	0	1	2
0	1	2	4
1	6		5
2	10		1
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
```

```
M[0][0] = 1
```

```
M[0][1] = 2
```

```
M[0][2] = 3
```

```
M[0][2] = 4
```

```
M[1][0] = 12
```

```
M[2][0] = 10
```

```
M[3][0] = 5
```

```
M[1][0] = 6
```

```
M[2][2] = 1
```

```
M[1][2] = M[3][0]
```

```
M[2][1] = M[0][0] + 1
```

M =	0	1	2
0	1	2	4
1	6		5
2	10	2	1
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
M[2][1] = M[0][0] + 1
M[1][1] = M[2][1] + M[1][2]
```

M =

	0	1	2
0	1	2	4
1	6		5
2	10	2	1
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
M[2][1] = M[0][0] + 1
M[1][1] = M[2][1] + M[1][2]
```

M =

	0	1	2
0	1	2	4
1	6	7	5
2	10	2	1
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
M[2][1] = M[0][0] + 1
M[1][1] = M[2][1] + M[1][2]
M[3][1] = M[0][2] * M[1][1]
```

M =

	0	1	2
0	1	2	4
1	6	7	5
2	10	2	1
3	5		

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
M[2][1] = M[0][0] + 1
M[1][1] = M[2][1] + M[1][2]
M[3][1] = M[0][2] * M[1][1]
```

M =

	0	1	2
0	1	2	4
1	6	7	5
2	10	2	1
3	5	28	

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
M[2][1] = M[0][0] + 1
M[1][1] = M[2][1] + M[1][2]
M[3][1] = M[0][2] * M[1][1]
M[3][2] = M[2][1] + M[1][2] - M[1][1]
```

M =

	0	1	2
0	1	2	4
1	6	7	5
2	10	2	1
3	5	28	

- **Exemplo**

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
M[2][1] = M[0][0] + 1
M[1][1] = M[2][1] + M[1][2]
M[3][1] = M[0][2] * M[1][1]
M[3][2] = M[2][1] + M[1][2] - M[1][1]
```

M =

	0	1	2
0	1	2	4
1	6	7	5
2	10	2	1
3	5	28	0

Exemplo

- Para acessar os elementos, deve-se escrever o nome da matriz seguido de dois índices entre colchetes.
- Por exemplo:

```
inteiro M[4][3]
M[0][0] = 1
M[0][1] = 2
M[0][2] = 3
M[0][2] = 4
M[1][0] = 12
M[2][0] = 10
M[3][0] = 5
M[1][0] = 6
M[2][2] = 1
M[1][2] = M[3][0]
M[2][1] = M[0][0] + 1
M[1][1] = M[2][1] + M[1][2]
M[3][1] = M[0][2] * M[1][1]
M[3][2] = M[2][1] + M[1][2] - M[1][1]
```

M =

	0	1	2
0	1	2	4
1	6	7	5
2	10	2	1
3	5	28	0

Matrizes

- **Sintaxe**

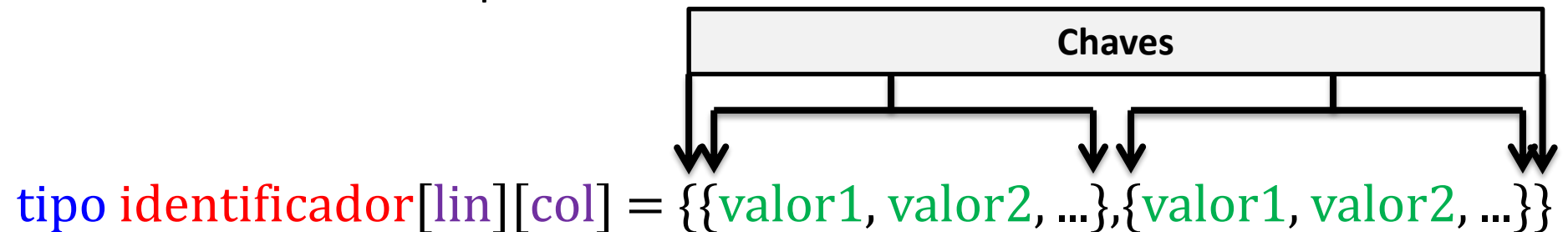
- Também é possível atribuir os valores iniciais dos elementos da matriz assim que ela for declarada.

`tipo identificador[lin][col] = {{valor1, valor2, ...},{valor1, valor2, ...}}`

Matrizes

- **Sintaxe**

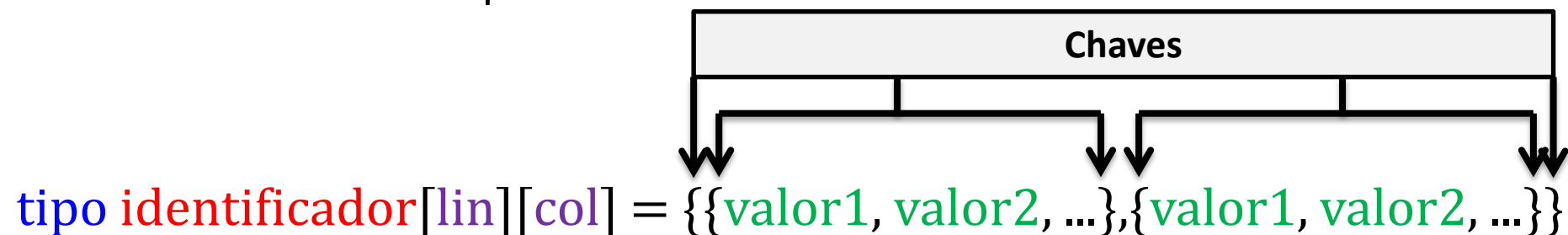
- Também é possível atribuir os valores iniciais dos elementos da matriz assim que ela for declarada.



Matrizes

- **Sintaxe**

- Também é possível atribuir os valores iniciais dos elementos da matriz assim que ela for declarada.



Atenção! Se a matriz não for inicializada, ela estará preenchida com um valor padrão para cada tipo.

Todos os valores necessários para preencher a matriz, um para cada posição.

Matrizes

- **Exemplo**

- Por exemplo:

```
cadeia M[2][2] = {{ "Marco", "André"},  
                  {"Abud", "Kappel"}}
```

Matrizes

- Exemplo

– Por exemplo:

```
cadeia M[2][2] = {{ "Marco", "André"},  
                  { "Abud", "Kappel"}}
```

	0	1	
M =	"Marco"	"André"	0
	"Abud"	"Kappel"	1

Matrizes

- Exemplo

– Por exemplo:

```
cadeia M[2][2] = {{ "Marco", "André"},  
                  { "Abud", "Kappel"}}
```

	0	1	
M =	"Marco"	"André"	0
	"Abud"	"Kappel"	1

```
escreva(M[0][0], " ", M[1][1])
```

Matrizes

- Exemplo

– Por exemplo:

```
cadeia M[2][2] = {{ "Marco", "André"},  
                  { "Abud", "Kappel"}}
```

	0	1	
M =	"Marco"	"André"	0
	"Abud"	"Kappel"	1

```
escreva(M[0][0], " ", M[1][1])
```



```
> _ Console Mensagens  
Marco Kappel  
Programa finalizado. Tempo de execução: 20 milissegundos
```

Matrizes

- **Definição do tamanho**

- Quando se usa matrizes, uma **boa prática** é definir no início do programa duas **constantes** com os números de linhas e colunas da matriz.
- Com estes valores **centralizados**, fica mais fácil criar **várias matrizes** com o **mesmo tamanho** e, se necessário, alterar o tamanho de **todas elas**.

```
const inteiro MAX_LINHAS = 10
const inteiro MAX_COLUNAS = 10

inteiro Matriz[MAX_LINHAS][MAX_COLUNAS]
```

Matrizes

- **Definição do tamanho**

- Quando se usa matrizes, uma **boa prática** é definir no início do programa duas **constantes** com os números de linhas e colunas da matriz.
- Com estes valores **centralizados**, fica mais fácil criar **várias matrizes** com o **mesmo tamanho** e, se necessário, alterar o tamanho de **todas elas**.

```
const inteiro MAX_LINHAS = 10  
const inteiro MAX_COLUNAS = 10  
  
inteiro Matriz[MAX_LINHAS][MAX_COLUNAS]
```


Matrizes

- **Percorrer uma matriz**

- Em muitos problemas, precisamos **percorrer** uma matriz até encontrar um elemento de interesse.
- Por exemplo, suponha que temos a seguinte matriz de números inteiros:

M =

	0	1	2	3	4
0	2	5	3	5	0
1	1	4	2	10	8

- Como faríamos para **verificar** se o número 8 está na matriz?


Matrizes

- **Percorrer uma matriz**

- Em muitos problemas, precisamos **percorrer** uma matriz até encontrar um elemento de interesse.
- Por exemplo, suponha que temos a seguinte matriz de números inteiros:

Percorreríamos a matriz, elemento por elemento!

M =



	0	1	2	3	4
0	2	5	3	5	0
1	1	4	2	10	8

- Como faríamos para **verificar** se o número 8 está na matriz?

Matrices

- **Percorrer uma matriz**
 - Ou seja, podemos usar **duas estruturas de repetição, uma dentro da outra!**
 - **Exemplo:** Complete o código do programa abaixo:

```
const inteiro MAX_LINHAS = 2  
const inteiro MAX_COLUNAS = 5  
  
inteiro M[MAX_LINHAS][MAX_COLUNAS] = {{2, 5, 3, 5, 0},  
                                         {1, 4, 2, 10, 8}}  
  
para(  
    para(  
        se(  
            escreva("Encontrei o 8 na posição: [", ", ", ", ", ", ", "]")  
        )  
    )  
}
```

- **Exemplo**

- Dadas as matrizes M e R e os comandos abaixo, qual será o conteúdo final de R?

M =

'O'	'Q'	'*'	'i'
'E'	'*'	'E'	'S'
'R'	'E'	'U'	'T'
'A'	'*'	'*'	'S'

R =



```
i = 0
enquanto(i < 4){
    j = 0
    enquanto(j < 4){
        R[j][i] = M[i][j]
        j = j + 1
    }
    i = i + 1
}
aux = R[0][0]
R[0][0] = R[3][3]
R[3][3] = aux
aux = R[1][1]
R[1][1] = R[2][2]
R[2][2] = aux
```

Matrizes

- **Passar matriz para função**

- Podemos passar uma **matriz** como **argumento** para uma **função**.
- Na chamada da função, deve-se fazer o **mesmo** que se fazia com uma variável ou um vetor, passando o **nome da matriz**.
- Na **declaração da função**, deve-se colocar **colchetes** na frente do identificador para mostrar que se trata de uma **matriz**.
- Por exemplo, a seguinte função **recebe uma matriz** e imprime seus elementos na tela:

`escreveMatriz(R)`



```
funcao escreveMatriz(inteiro M[][]){  
    para(inteiro i = 0; i < MAX_LINHAS; i++){  
        para(inteiro j = 0; j < MAX_COLUNAS; j++){  
            escreva (M[i][j], " ")  
        }  
        escreva ("\n")  
    }  
}
```

Matrizes

- **Retornar matriz de função**
 - Não é possível **retornar** uma matriz **diretamente** de uma função.
 - Mas podemos fazer isso **indiretamente**. Como?

Matrizes

- **Retornar matriz de função**

- Não é possível **retornar** uma matriz **diretamente** de uma função.
- Mas podemos fazer isso **indiretamente**. Como?
- Passagem de valor por **referência**!
- Basta colocar o **&** indicando que a matriz está sendo passada desta forma.
- Por exemplo, a função abaixo **preenche uma matriz** com valores digitados pelo usuário:

```
funcao leMatriz(inteiro &M[][]){  
    para(inteiro i = 0; i < MAX_LINHAS; i++){  
        para(inteiro j = 0; j < MAX_COLUNAS; j++){  
            escreva("Digite o elemento de indice [",i,""][",j,"]: ")  
            leia(M[i][j])  
        }  
    }  
}
```

- **Exemplo:**

- Implemente o seguinte programa, que cria uma matriz, lê e imprime os seus valores:

```
programa{

    const inteiro MAX_LINHAS = 3
    const inteiro MAX_COLUNAS = 3

    funcao inicio(){

        inteiro M[MAX_LINHAS][MAX_COLUNAS]

        leMatriz(M)
        escreveMatriz(M)
    }

    funcao escreveMatriz(inteiro M[][]){
        para(inteiro i = 0; i < MAX_LINHAS; i++){
            para(inteiro j = 0; j < MAX_COLUNAS; j++){
                escreva (M[i][j], " ")
            }
            escreva ("\n")
        }
    }

    funcao leMatriz(inteiro &M[][]){
        para(inteiro i = 0; i < MAX_LINHAS; i++){
            para(inteiro j = 0; j < MAX_COLUNAS; j++){
                escreva("Digite o elemento de indice [",i,""][",j,"]: ")
                leia(M[i][j])
            }
        }
    }
}
```


• Exercícios

1. Crie uma função que recebe duas matrizes com 3 x 3 elementos inteiros cada, e retorne a soma das duas matrizes. Para retornar a matriz com a soma, utilize um terceiro argumento de entrada, passado por referência.

2 . Faça um programa que leia uma matriz de ordem 3 x 5, de elementos inteiros, calcule e exiba:

- a) O maior elemento da matriz.
- b) A soma dos elementos da matriz.
- c) A média dos elementos da matriz.

3. Faça uma função que recebe uma matriz com 10 x 10 elementos inteiros cada, e retorna um valor lógico indicando se ela é perfeita. Uma matriz é perfeita se a soma dos elementos de sua diagonal principal é igual à soma dos elementos da diagonal secundária.

- **Exercícios**

4. A tabela a seguir demonstra a quantidade de vendas dos fabricantes de veículos durante o período entre 2000 e 2005, em mil unidades.

Fabricante / Ano	2000	2001	2002	2003	2004	2005
Fiat	204	223	230	257	290	322
Ford	195	192	198	203	208	228
GM	220	222	217	231	245	280
Volkswagen	254	262	270	284	296	330

Faça um programa que:

- a) Leia dos dados da tabela, digitados pelo usuário e guarde da seguinte forma: um vetor de fabricantes, um vetor de anos e uma matriz de vendas.
- b) Determine e exiba o fabricante que mais vendeu em 2003.
- c) Determine e exiba o ano de maior volume geral de vendas.
- d) Determine e exiba a média de vendas de cada fabricante durante o período.

Matrizes

- **Exercícios**

5. Faça uma função que recebe uma matriz e dois valores inteiros x e y . A função deve permutar as linhas x e y da matriz.
6. Faça uma função que recebe uma matriz quadrada M e constrói sua transposta M^T .
7. Faça uma função que recebe uma matriz quadrada, um número inteiro c e um vetor de dimensão igual ao número de linhas da matriz. A função deve substituir a coluna c da matriz pelo vetor dado.

Matrizes

FIM