

# Tipos de Dados Estruturados

Prof<sup>a</sup>.Dr<sup>a</sup>.Thatyana de Faria Piola Seraphim (ECO)  
Prof.Dr.Enzo Seraphim (ECO)

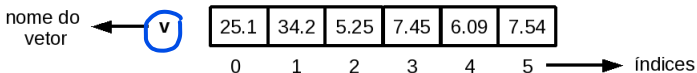
Universidade Federal de Itajubá

thatyana@unifei.edu.br  
seraphim@unifei.edu.br

# Estruturas Homogêneas

## Vetores

- ▶ Um **vetor** é uma sequência de objetos do mesmo tipo.
- ▶ Os objetos são chamados **elementos do vetor** e são numerados consecutivamente (0, 1, 2, 3, ...).
- ▶ Os números dos elementos do vetor são denominados **valores índices**. Os números localizam a posição do elemento dentro do vetor, possibilitando *acesso direto* ao vetor.
- ▶ Os tipos de elementos armazenados no vetor podem ser qualquer tipo de dado em C, incluindo estruturas definidas pelo usuário.
- ▶ Se o nome do vetor é **v**, então **v[0]** é o nome do elemento que está na posição 0, **v[1]** é o nome do elemento que está na posição 1.



# Estruturas Homogêneas

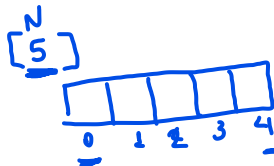
## Vetores

- ▶ Assim como qualquer tipo de variável, deve-se declarar um vetor antes de utilizá-lo.
- ▶ Um vetor é declarado de forma similar a outros tipos de dados, exceto que se deve indicar ao compilador o **tamanho** ou o **comprimento** do vetor.
- ▶ Para indicar o tamanho ou o comprimento do vetor, deve-se colocar após o nome, o tamanho entre colchetes.

### Sintaxe para declarar um vetor

tipo nomeVetor[numeroDeElementos];

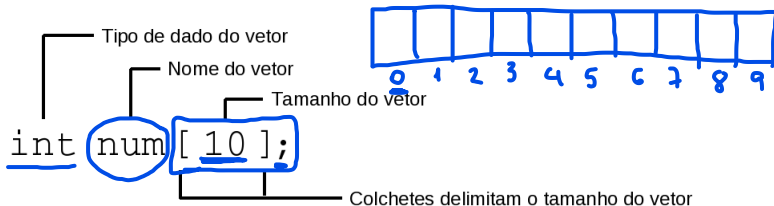
- ▶ **tipo**: tipo de dado do vetor.
- ▶ **nomeVetor**: nome dado ao vetor.
- ▶ **numeroDeElementos**: tamanho do vetor. Deve ser uma constante inteira, assim como 10.



# Estruturas Homogêneas

## Vetores

### Criando um vetor de 10 elementos inteiros

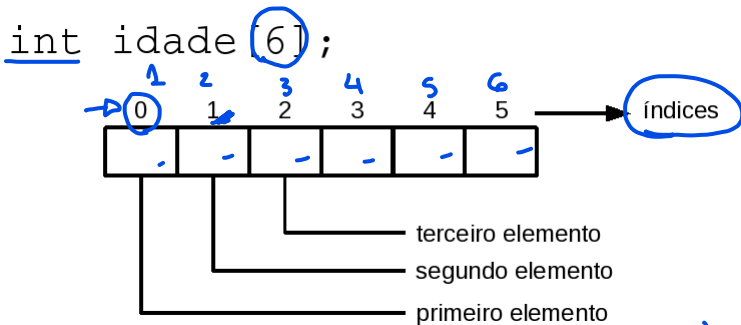


- Essa declaração permite que o compilador reserve espaço suficiente para conter 10 valores inteiros.

# Estruturas Homogêneas

## Vetores

- ▶ Grande parte da utilidade de um vetor provém do fato de se poder acessar os elementos do vetor de forma individual.
- ▶ Um método para acessar um elemento é utilizar um **índice**.



- ▶ Os **vetores sempre começam em zero**.

→ índice do vetor!

# Estruturas Homogêneas

## Vetores

- Os elementos do vetor são armazenados em blocos contínuos.

### Exemplo

```
1 int idade[5];  
2 char codigo[5];
```

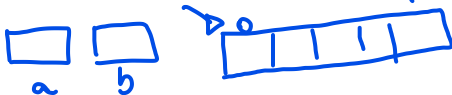
idade [ro]

idade

[0]	18
[1]	21
[2]	19
[3]	20
[4]	21

codigo

[0]	
[1]	
[2]	
[3]	
[4]	



var idade: vetor[1..5] de inteiro  
int idade [5]\*

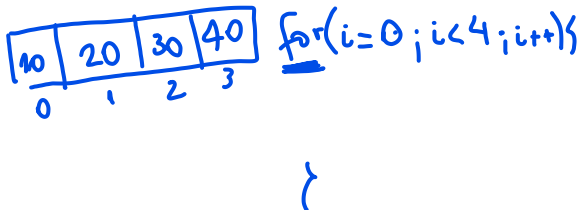
# Estruturas Homogêneas

## Vetores

- ▶ Deve-se atribuir valores aos elementos do vetor antes de utilizá-los, assim como se atribuem valores a variáveis.

### Exemplos

```
1 idade[0] = 10;  
2 idade[1] = 20;  
3 idade[2] = 30;  
4 idade[3] = 40;  
5 ...
```

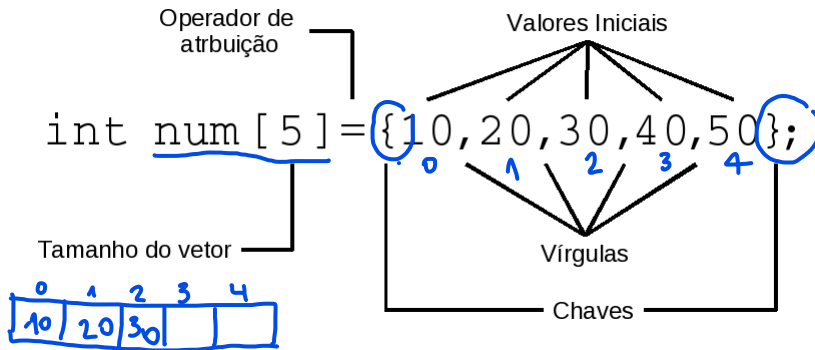


- ▶ A primeira sentença atribui o valor 10 à variável `idade[0]` e o valor 20 à variável `idade[1]`.
- ▶ Este método não é eficaz quando o vetor contém muitos elementos.

# Estruturas Homogêneas

## Vetores

- Pode-se dar valores aos elementos de um vetor, quando este é definido.
- O sistema consiste em armazenar os valores correspondentes que estão entre chaves e separados por vírgula.





# Estruturas Homogêneas

## Vetores

- ▶ Não é necessário utilizar o tamanho do vetor quando são inicializados os elementos do vetor, uma vez que o compilador contará o número de variáveis que inicializa.
- ▶ Quando se inicializa um vetor, o seu tamanho pode ser determinado automaticamente pelas constantes de inicialização.
- ▶ As constantes de inicialização são separadas por vírgulas e estão entre chaves.

## Exemplos

```
1  int num[6] = {10, 20, 30, 40, 50, 60};  
2  int n[ ] = {3, 4, 5}; //declara um vetor de 3 elementos  
3  float k[ ] = {2.15, 32.1, 17.5, 65.6}; //declara um vetor  
4  //de 4 elementos
```

# Estruturas Homogêneas

## Vetores

- ▶ Os vetores de caracteres podem ser inicializados com uma constante de cadeia.

### Exemplo

```
1 //declara um vetor de 6 elementos
2 char c[ ] = {'U', 'n', 'i', 'f', 'e', 'i'};
3 //declara um vetor de 7 elementos
4 char c[ ] = "Unifei";
```



- ▶ O método de inicializar vetores por meio de valores constantes depois de sua definição.
  - ▶ é adequado quando o número de elementos do vetor é pequeno.

# Estruturas Homogêneas

## Vetores

### Exemplo 01

Inicializar o vetor com 10 elementos e imprimir os elementos em ordem inversa.

FOR  
WHILE

# Estruturas Homogêneas

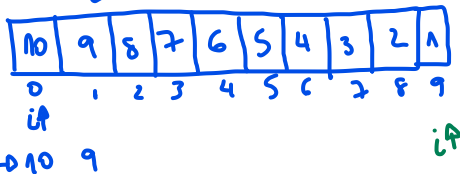
## Vetores

### Exemplo 01

```
1  #include<stdio.h>
2  int main(int argc, char *argv[]){
3      int conta[10]= {10,9,8,7,6,5,4,3,2,1}; //inicializacao do vetor
4      int i;
5      //impressao do vetor original
6      printf("Vetor Original = ");
7      for(i = 0; i < 10; i++){*
8          printf("%d ", conta[i]);
9      }//end for
10     printf("\n");
11     //impressao do vetor em ordem inversa
12     printf("Ordem inversa = ");
13     for(i = 9; i >= 0; i--){
14         printf("%d ", conta[i]);
15     }//end for
16     return 0;
17 }
```

$conta[i] = i * 3;$

\* printf("Digite o valor = ");  
scanf("%d", &conta[i]);



1 2 3 4 .... 10

# Estruturas Homogêneas

## Vetores

### Exercício 01

Dado um vetor de números inteiros, faça a sua inicialização e some os valores do vetor.

# Estruturas Homogêneas

## Vetores

### Resposta do exercício 01

```
1  #include<stdio.h>
2  int main(int argc, char *argv[]){
3      int i, soma, vet[10];
4      soma = 0;
5      //inicializacao do vetor
6      for(i = 0; i < 10; i++){
7          vet[i] = i * 45; → scanf( 1;
8      }//end for
9      //soma dos elementos do vetor
10     for(i = 0; i < 10; i++){ printf("%d ", vet[i]);
11         soma = soma + vet[i];
12     }//end for
13     return 0;
14 }
```

# Estruturas Homogêneas

## Vetores

### Exercício 02

Dado dois vetores:

```
1  int vetA[5] = {1, 2, 3, 4, 5};  
2  int vetB[5] = {2, 3, 4, 5, 6};
```

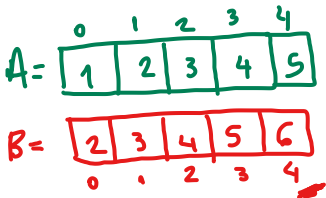
Encontre a diferença entre os dois vetores, ou seja, quais elementos estão no vetA mas não estão no vetB.

# Estruturas Homogêneas

## Vetores

### Resposta do exercício 02

```
1  #include<stdio.h>
2  int main(int argc, char *argv[]){
3      int vetA[5] = {1, 2, 3, 4, 5};
4      int vetB[5] = {2, 3, 4, 5, 6};
5      int i, j;
6      for(i = 0; i < 5; i++){
7          for(j = 0; j < 5; j++){
8              if(vetA[i] == vetB[j]){
9                  break;
10             }//end if
11         }//end for j
12         if(j == 5){
13             printf("%d", vetA[i]);
14         }//end if
15     }//end for i
16     return 0;
17 }
```



vetA[i];  $vetC[i] = vetA[i];$



# Estruturas Homogêneas

## Matrizes

- ▶ Uma **matriz** é uma variável composta homogênea bidimensional.
  - ▶ formada por uma sequência de variáveis, todas do mesmo tipo;
  - ▶ possuem o mesmo identificador (mesmo nome).
- ▶ Uma vez que as variáveis tem o mesmo nome, o que as diferencia são os índices que referenciam sua localização dentro da estrutura.
- ▶ Uma variável do **tipo matriz** de duas dimensões pode ser referenciada como tendo **linhas** (dimensao1) e **colunas** (dimensao2).

**Definição da matriz** *linha* *coluna*  
tipoDado nomeMatriz[dimensao1][dimensao2];

### Definição da matriz

**tipoDado** nomeMatriz[dimensao1][dimensao2];

onde:

- ▶ **tipoDado**: é o tipo de dados que poderá ser armazenado na matriz.
- ▶ **nomeMatriz**: é o nome dado à variável do tipo da matriz.
- ▶ **[dimensao1]**: indica o tamanho da dimensão 1 (número de linhas).
- ▶ **[dimensao2]**: indica o tamanho da dimensão 2 (número de colunas).
- ▶ Da mesma maneira como ocorre com os vetores, os **índices** começam sempre em 0 (zero).

# Estruturas Homogêneas

## Matrizes

- ▶ Por exemplo:

```
int a[2][6];
```

- ▶ Criou-se uma matriz chamada **a** contendo duas linhas (0 e 1) e 6 colunas (0 à 5), capazes de armazenar números inteiros.

	0	1	2	3	4	5
0	1	5	2	3		
1						

- ▶ Por exemplo:

```
float mat[3][3] //contem 3 linhas e 3 colunas
```

	0	1	2
0	1.5	2.3	
1			
2			

# Estruturas Homogêneas

## Matrizes

### Atribuindo valores à matriz

a[1][4] = 10;

	0	1	2	3	4	5
0						
1					10	

mat[2][1] = 3.45;

	0	1	2
0			
1			
2		3.45	

# Estruturas Homogêneas

## Matrizes

- ▶ As matrizes podem ser inicializadas quando são declaradas.
- ▶ A inicialização consta de uma lista de constantes separadas por vírgulas e estão entre chaves.

### Exemplos de inicialização de matriz

```
1 int mat[2][3] = {51, 52, 53, 54, 55, 56};  
2 int mat[2][3] = ({51, 52, 53}, {54, 55, 56});  
3 int mat[2][3] = { {51, 52, 53},  
4   {54, 55, 56} };
```

		colunas		
		0	1	2
linhas	0	51	52	53
	1	54	55	56

mat[2][3]

$mat[i][j][k];$

		colunas			
		0	1	2	3
linhas	0	51	52	53	53
	1	54	55	56	56
	2	51	52	53	53

mat[3][4]

# Estruturas Homogêneas

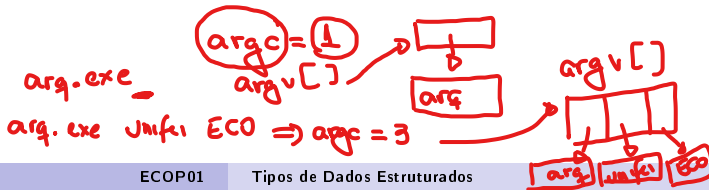
## Matrizes

### Inicialização da matriz

```
1  #include<stdio.h>
2  int main(int argc, char *argv[]) {
3      int i, j, mat[3][4];
4      //inicializacao da matriz
5      for(i = 0; i < 3; i++){
6          for(j = 0; j < 4; j++){
7              scanf("%d", &mat[i][j]);
8              //end for de j
9          }
10         //end for de i
11     }
12     return 0;
13 }
```

*Handwritten notes:*

- $/* m * n */$  (next to line 3)
- $mat[i][j] = i \times j + 1;$  (next to line 7)
- $rand();$  (next to line 7)



# Estruturas Homogêneas

## Matrizes

### Mostra os elementos da matriz

```
1  #include<stdio.h>
2  int main(int argc, char *argv[]){
3      int i, j, mat[3][4];
4      for(i = 0; i < 3; i++){ ← Linha
5          for(j = 0; i < 4; j++){ ← Coluna
6              printf("elem[%d][%d] = %d \n", i, j, mat[i][j]);
7          } //end for de j
8      } //end for de i
9      return 0;
10 } //end main
```

# Estruturas Homogêneas

## Matrizes

### Exercício 01

Ler duas matrizes de inteiros A e B, cada uma de duas dimensões com 5 linhas e 3 colunas. Construir uma matriz C de mesma dimensão, onde C é formada pela soma dos elementos da matriz A com os elementos da matriz B.



# Estruturas Homogêneas

## Matrizes

### Resposta do exercício 01

```
1  #include<stdio.h>
2  int main(int argc, char *argv[]){
3      int i, j, matA[5][3], matB[5][3], matC[5][3];
4      //inicializacao das matrizes A e B
5      for(i = 0; i < 5; i++){ -linha
6          for(j = 0; j < 3; j++){ -coluna
7              scanf("%d", &matA[i][j]);
8              scanf("%d", &matB[i][j]);
9              matC[i][j] = matA[i][j] + matB[i][j];
10         }//end for j
11     }//end for i
12     return 0;
13 }//end main
```

### Exercício 02

Faça um programa que inicializa uma matriz identidade de ordem 1000.

# Estruturas Homogêneas

## Matrizes

### Resposta do exercício 02

```
1  #include<stdio.h>
2  int main(int argc, char *argv[]){
3      int i, j, matI[1000][1000];
4      //inicializacao da matriz identidade
5      for(i = 0; i < 1000; i++){-linha
6          for(j = 0; j < 1000; j++){-coluna
7              if(i == j){
8                  matI[i][j] = 1;
9              }//end if
10             else{
11                 matI[i][j] = 0;
12             }//end else
13         }//end for j
14     }//end for i
15     return 0;
16 }
```