

ENSINO MÉDIO INTEGRADO - INFORMÁTICA

Disciplina de Linguagem de Programação

Aula 08: Vetor - *Inteiro*

Gil Eduardo de Andrade

Conceitos Preliminares

<https://docs.microsoft.com/pt-br/cpp/c-language/?view=msvc-170>

Introdução

Um vetor pode ser definido como é um conjunto de elementos (variáveis) que apresentam um mesmo tipo e são referenciados por um nome comum. Essas variáveis que compõem um vetor são acessadas através de um índice inteiro, onde o índice de menor valor (índice zero) corresponde ao primeiro elemento do vetor, enquanto o índice de maior valor corresponde ao último elemento.

A declaração de vetores é efetuada, de forma geral, da seguinte maneira:

tipo nome_vetor[tamanho];

- **tipo:** indica qual é o tipo das variáveis que compõem o vetor. Exemplo: *int*, *float*, *double*;
- **nome_vetor:** indica o nome pelo qual o vetor é referenciado;
- **tamanho:** indica o tamanho do vetor, o número de elementos total que ele contém;

Sintaxe e Utilização

Considerando a descrição apresentada anteriormente, podemos então declarar um vetor do **tipo *int*** de **tamanho 10**. Teríamos:

int numeros[10];

Ao declararmos o vetor “**numeros**”, garantimos que o espaço de memória necessário para armazenar todos os seus elementos seja reservado. Contudo é

importante observar que apenas declaramos o vetor, ou seja, ainda não especificamos os elementos (valores) que ele deve armazenar.

Antes desta especificação, dizemos que o vetor possui “lixo de memória” armazenado em suas posições, que são valores quaisquer gerados por outros programas que utilizaram esse mesmo espaço de memória num momento anterior.

Armazenando Valores no Vetor

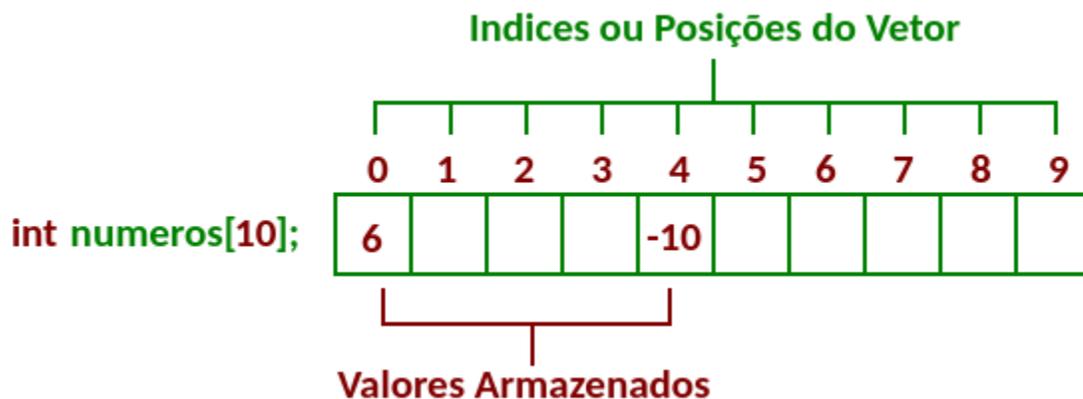
Como mencionado, o tamanho do vetor indica quantos elementos podemos armazenar nele. Para que seja possível especificar qual elemento do vetor receberá um dado que queremos armazenar, utilizamos um índice (valor inteiro) que indica a posição do vetor que receberá o dado.

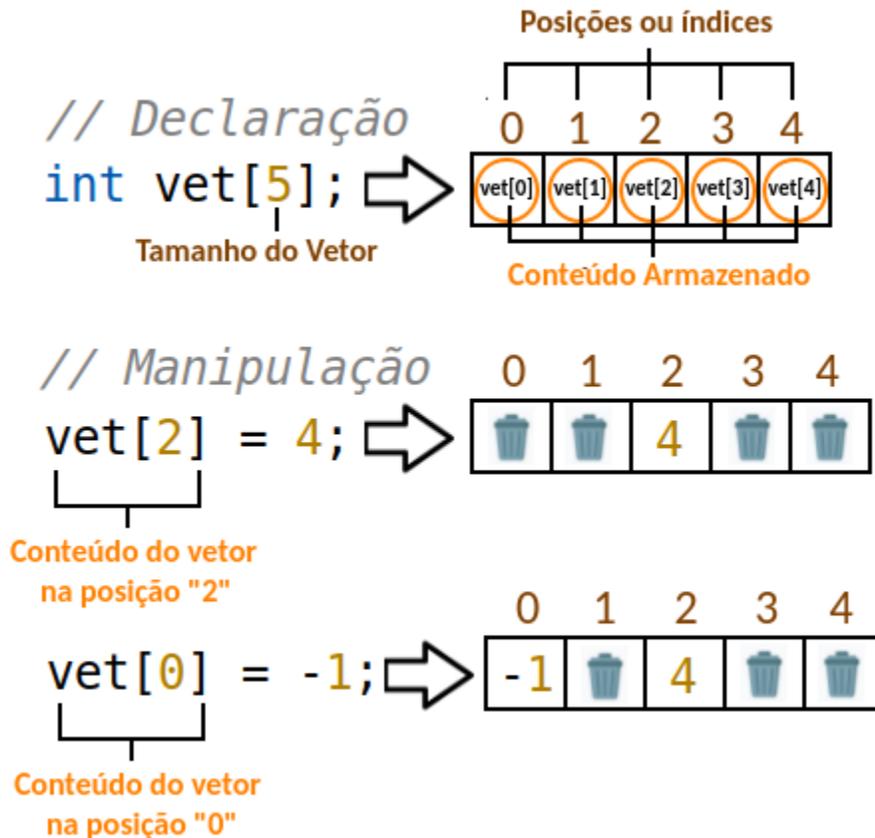
Os índices do vetor variam de **0** até seu **tamanho-1**, ou seja, o vetor **int numeros[10]** possui os **índices: 1, 2, 3, 4, 5, 6, 7, 8 e 9**. Para armazenar dados neste vetor utilizamos a seguinte sintaxe:

- **numeros[0] = 6;**
- **numeros[4] = -10;**

Representação Gráfica

Um vetor, dentro da programação, pode ser representado por um retângulo na horizontal, dividido em um número de partes exatamente igual ao seu tamanho. Onde cada uma destas partes é referenciada por um índice.





Gerando Números aleatórios

Quando trabalhamos com vetores é comum a necessidade de gerar valores aleatórios para serem armazenados em suas posições. Para tal, em linguagem C, trabalhamos com as funções *srand()* e *rand()* disponíveis na biblioteca *<time.h>*.

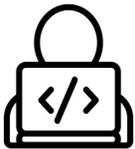
Utilizando as funções *srand()* e *rand()*

O primeiro passo que devemos efetuar é a inclusão da biblioteca no início do código fonte C, juntamente com todas as outras bibliotecas. O segundo passo consiste na geração de uma semente, necessária para que possamos gerar os números aleatórios (pseudo). A função *srand(time(NULL))* é responsável pela geração dessa sementes, e deve ser colocada logo após a declaração das variáveis. Efetuados os dois passos anteriores já podemos utilizar a função *rand()*, que permite obter valores inteiros aleatórios dentro de um limite pré-definido. A função *rand()* normalmente é utilizada juntamente com o operador matemático %

e um valor inteiro, que indica o limite superior dos valores que devem ser gerados.
Exemplo:

rand()%10

No exemplo acima serão gerados valores inteiros aleatórios entre 0 e 9. Perceba que o valor máximo gerado sempre será um valor menor que o limite inteiro especificado, nesse caso 10.



Codificação – Linguagem de Programação C

Exemplo Básico

```
#include <stdio.h>

int main() {

    // declara o vetor "numeros", do tipo "int" e tamanho "10"

    int numeros[10];

    // armazena o valor "6" na posição "0"

    numeros[0] = 6;

    // armazena o valor "-10" na posição "4"

    numeros[4] = -10;

    printf("\nnumeros[0] = %i", numeros[0]);

    printf("\nnumeros[4] = %i", numeros[4]);

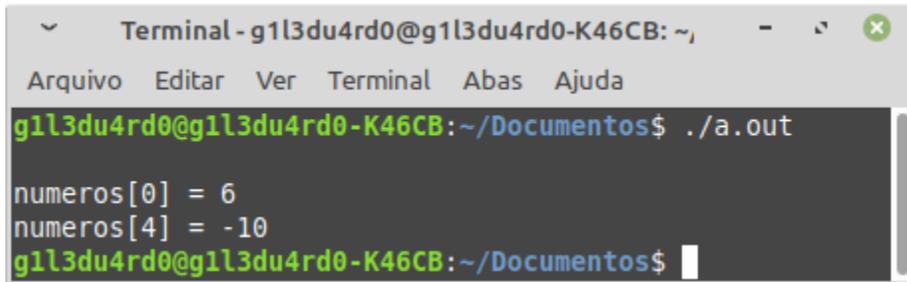
    printf("\n");

    return 0;
}
```

```
}
```

Declarando e Armazenando Valores no Vetor.

(Arquivo-fonte: 09 - Vetor inteiro/basico.c)



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~, - [x]
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$ ./a.out
numeros[0] = 6
numeros[4] = -10
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$
```

Resultado da execução do código anterior

Percorrendo e Preenchendo o Vetor

```
#include <stdio.h>

int main() {

    // declara o vetor "numeros", do tipo "int" e tamanho "10"

    int numeros[10];

    int indice;

    // percorre o vetor, da posição 0 até 9

    for(indice=0; indice<10; indice++) {

        numeros[indice] = indice * 2;

        printf("\nnumeros[%i] = %i", indice, numeros[indice]);

    }

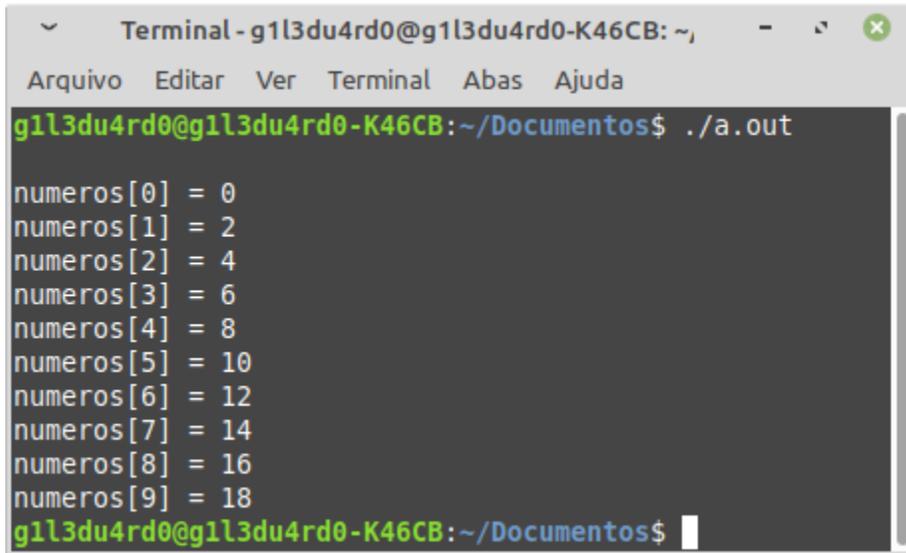
    printf("\n");

    return 0;

}
```

Utilizando um laço de repetição for(;;) para percorrer e preencher o vetor.

(Arquivo-fonte: 09 - Vetor inteiro/percorrer.c)



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~, - [x]
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$ ./a.out
numeros[0] = 0
numeros[1] = 2
numeros[2] = 4
numeros[3] = 6
numeros[4] = 8
numeros[5] = 10
numeros[6] = 12
numeros[7] = 14
numeros[8] = 16
numeros[9] = 18
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$
```

Resultado da execução do código anterior

Gerando Valores Aleatórios

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main() {

    int valor;

    // Geração da semente

    srand(time(NULL));

    // Geração de um número aleatório de 0 a 9

    valor = rand()%10;

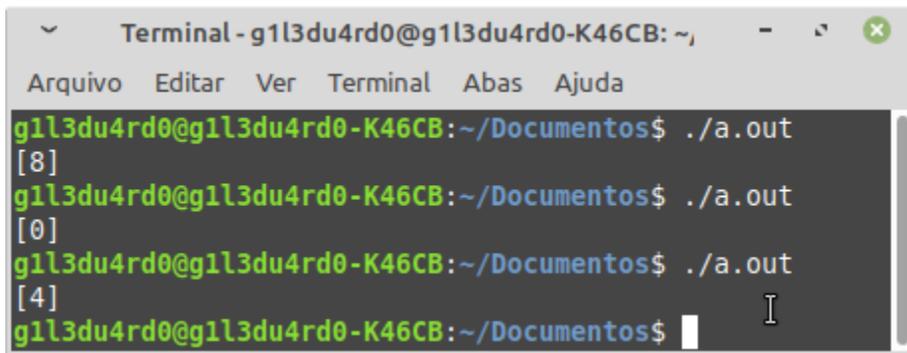
    // Apresentando o valor aleatório

    printf("[%i]", valor);
```

```
printf("\n");  
  
return 0;  
  
}
```

Gerando valores aleatórios.

(Arquivo-fonte: 09 - Vetor inteiro/aleatorio.c)



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~  
Arquivo  Editar  Ver  Terminal  Abas  Ajuda  
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$ ./a.out  
[8]  
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$ ./a.out  
[0]  
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$ ./a.out  
[4]  
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$
```

Resultado da execução do código anterior

Percorrendo e Preenchendo o Vetor com Valores Aleatórios

```
#include <stdio.h>  
  
#include <stdlib.h>  
  
#include <time.h>  
  
int main() {  
  
    // declara o vetor "numeros", do tipo "int" e tamanho "10"  
  
    int numeros[10];  
  
    int indice;  
  
    // Geração da semente  
  
    srand(time(NULL));  
  
    for(indice=0; indice<10; indice++) {
```

```
    numeros[indice] = rand()%10;

    printf("\nnumeros[%i] = %i", indice, numeros[indice]);

}

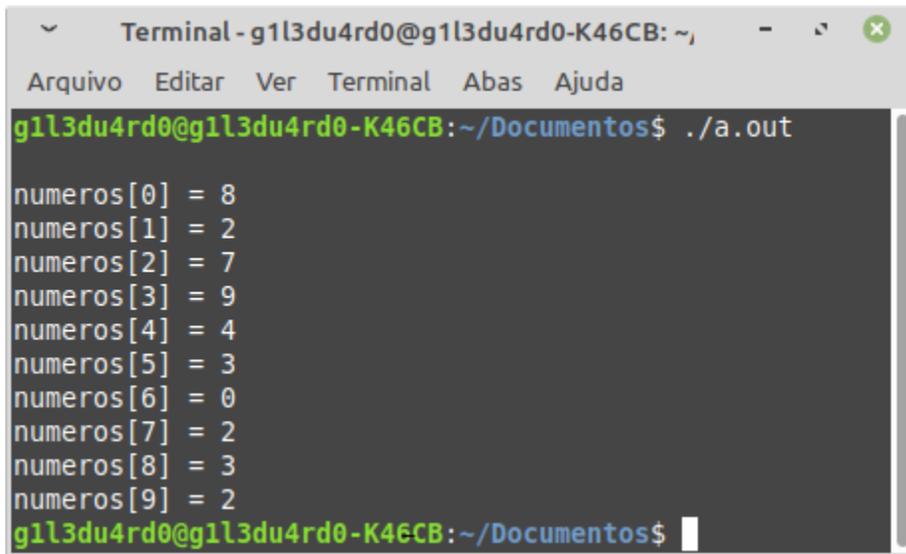
printf("\n");

return 0;

}
```

Percorrendo o vetor e preenchendo com valores aleatórios.

(Arquivo-fonte: 09 - Vetor inteiro/vetor_aleatorio.c)



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~, - [x]
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$ ./a.out
numeros[0] = 8
numeros[1] = 2
numeros[2] = 7
numeros[3] = 9
numeros[4] = 4
numeros[5] = 3
numeros[6] = 0
numeros[7] = 2
numeros[8] = 3
numeros[9] = 2
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos$
```

Resultado da execução do código anterior