Vetores

O problema

- Até o momento trabalhamos com **variáveis unitárias**, ou seja, variáveis que podiam **armazenar** somente **um único valor** de determinado **tipo;**

- Imagine a necessidade de se criar **mil variáveis** para representar a nota de todos os alunos de uma escola, como faríamos?

- Tal problema é tratado através de variáveis compostas, os chamados **vetores.**

Vetores/Arrays unidimensionais

- Permitem o tratamento de um conjunto de dados com as mesmas características;
- São uma maneira de armazenar vários dados (dezenas, centenas, milhares, milhões...) em um **mesmo nome de variável** através do uso de **índices numéricos.**

Declaração

- A declaração de uma variável do tipo vetor é semelhante a uma declaração de uma variável normal;
- A diferença é que deve ser informada a **quantidade de elementos** do vetor;

Declaração

- A declaração de uma variável do tipo vetor em C# segue o formato:

- `tipo[] nome = new tipo[tamanho];`

- Onde:
  - **`tipo`**: é o tipo de dados dos elementos que serão armazenados;
  - ´
  - `nome`: é o nome da variável (nome do vetor);
  - **`[tamanho]`**: é um valor numérico que indica quantos elementos estarão armazenados no vetor, do tipo de dado especificado

Declaração

- Exemplo de declaração:

```
//vetor de 100 elementos do tipo inteiro
int[] meuVetor = new int[100];
```

```
//Vetor de 50 elementos do tipo double
double[] meuOutrovetor = new double[50];

//vetor de 25 elementos do tipo char
char[] maisUmVetor = new char[25];
```

Armazenamento em memória

- Quando declaramos variáveis comuns em nossos algoritmos, geralmente elas ficam armazenadas na memória RAM em posições não contínuas, por exemplo:

- `int a=5, b, c=7;`
- `char d='a', e`

Memória RAM		
Posição	**Variável**	**Valor**
1	a	5
2		<LIXO>
3		<LIXO>
4		<LIXO>
5	c	7
6		<LIXO>
7		<LIXO>
8		<LIXO>
9	b	<LIXO>
10		<LIXO>
11		<LIXO>
12	e	<LIXO>
13		<LIXO>
14	d	'a'
15		<LIXO>

Armazenamento em memória

- Os elementos de um Vetor, na memória do computador, são armazenados de forma contínua;
- Os elementos são referenciados por índices;
- Ex:
  - `int[] veta = new int[8];`
  - `Char[] vetB = new int[5];`

Memória RAM		
Posição	**Variável**	**Valor**
1	vetA[0]	<LIXO>
2	vetA[1]	<LIXO>
3	vetA[2]	<LIXO>
4	vetA[3]	<LIXO>
5	vetA[4]	<LIXO>
6	vetA[5]	<LIXO>
7	vetA[6]	<LIXO>
8	vetA[7]	<LIXO>
9		<LIXO>
10		<LIXO>
11	vetB[0]	<LIXO>
12	vetB[1]	<LIXO>
13	vetB[2]	<LIXO>
14	vetB[3]	<LIXO>
15	vetB[4]	<LIXO>

Manipulação de um vetor

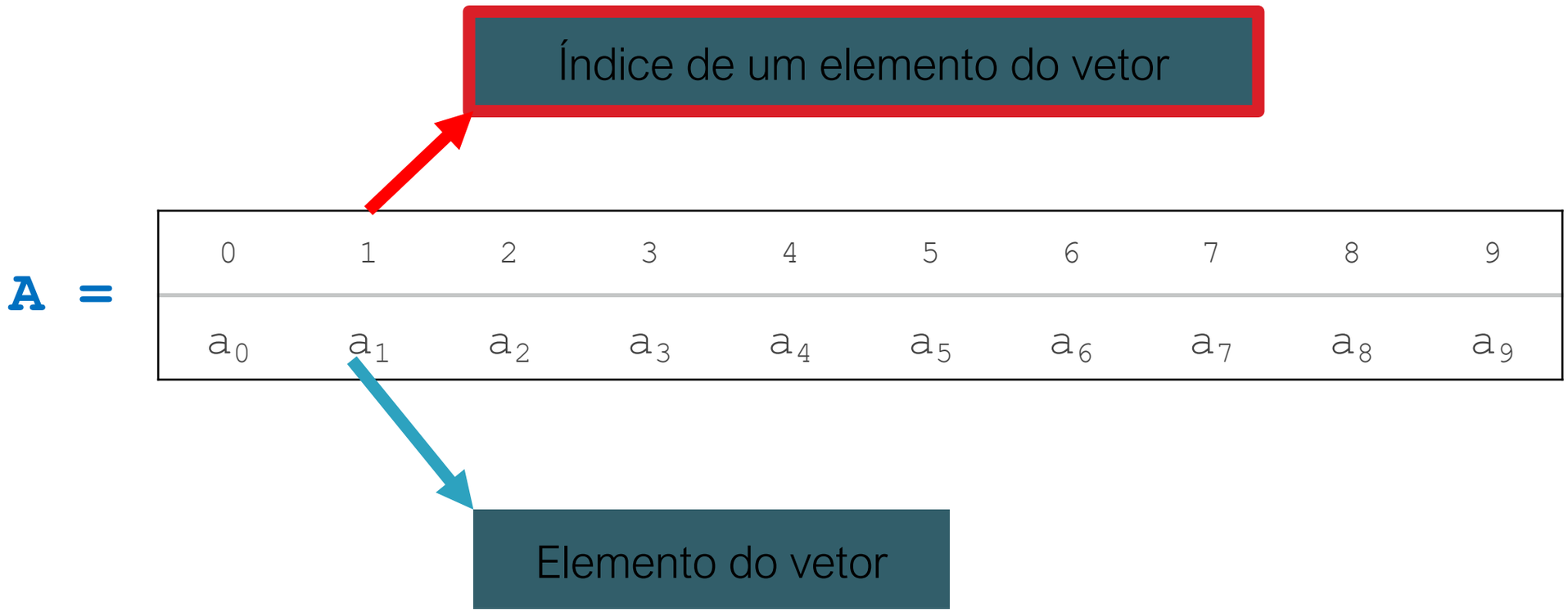
- Os elementos são **referenciados** por índices;

- O **primeiro elemento** do vetor possui sempre o índice ZERO;

- O **último elemento** do vetor tem índice igual ao TAMANHO do vetor MENOS 1 **`[tamanho-1];`**

Manipulação de um vetor

- Por exemplo, declaramos o seguinte vetor:
  - int[] A = new int[10];
- Na memória, ele seria armazenado mais ou menos assim:



Manipulação de um vetor

- Para acessar um elemento do vetor, deve-se usar o nome do vetor e o valor do índice do elemento entre colchetes [ ]

- Por exemplo, na figura abaixo, o 4º elemento do vetor é referenciado por **`A[3]`**, cujo valor é **`20`**.

```
int[] A = new int[10];
```

A =

0	1	2	3	4	5	6	7	8	9
5	10	15	20	25	30	35	40	45	50

Manipulação de um vetor

- A atribuição é feita da mesma forma, por exemplo:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2				

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- A atribuição é feita da mesma forma, por exemplo:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2		5		

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- A atribuição é feita da mesma forma, por exemplo:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2	2-5	5		

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- A atribuição é feita da mesma forma, por exemplo:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2	-3	5		

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- A atribuição é feita da mesma forma, por exemplo:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2	-3	5	5*5	

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- A atribuição é feita da mesma forma, por exemplo:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2	-3	5	25	

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- A atribuição é feita da mesma forma, por exemplo:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2	-3	5	25	25-1

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- A atribuição é feita da mesma forma, por exemplo:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2	-3	5	25	24

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- Para mostrar um elemento de um vetor em um comando `Console.WriteLine`, seguimos a mesma ideia:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;
```

b =

0	1	2	3	4
2	-3	5	25	24

```
Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

Manipulação de um vetor

- Para mostrar um elemento de um vetor em um comando `Console.WriteLine`, seguimos a mesma ideia:

```
int[] b = new int[5];
b[0] = 2;
b[2] = 5;
b[1] = b[0] - b[2];
b[3] = b[2] * b[2];
b[4] = b[3] - 1;

Console.WriteLine(b[0] + " " + b[1] + " " + b[2] + " " + b[3] + " " + b[4]);
```

b =

0	1	2	3	4
2	-3	5	25	24

Saída: 2 -3 5 25 24

Exemplo 1 – apresentação de valores

- **Geralmente utilizamos um comando `for`:**

```
static void Main(string[] args)
{
    int i;
    double[] b = new double[5];
    b[0] = 2;
    b[2] = 5;
    b[1] = b[0] - b[2];
    b[3] = Math.Pow(b[2], 2);
    b[4] = b[3] - 1;

    for (i = 0; i < 5; i++)
    {
        Console.WriteLine(b[i]);
    }
}
```

quando i=0, mostra b[0];
quando i=1, mostra b[1];
quando i=2, mostra b[2];
quando i=3, mostra b[3];
quando i=4, mostra b[4];
quando i=5, termina o laço.

Exemplo 2 – leitura de valores

```
static void Main(string[] args)
{
    int i;
    int[] b = new int[5];
    Console.WriteLine("Digite o valor 0: ");
    b[0] = int.Parse(Console.ReadLine());
    Console.WriteLine("Digite o valor 1: ");
    b[1] = int.Parse(Console.ReadLine());
    Console.WriteLine("Digite o valor 2: ");
    b[2] = int.Parse(Console.ReadLine());
    Console.WriteLine("Digite o valor 3: ");
    b[3] = int.Parse(Console.ReadLine());
    Console.WriteLine("Digite o valor 4: ");
    b[4] = int.Parse(Console.ReadLine());

    for (i = 0; i < 5; i++)
    {
        Console.WriteLine("Valor índice "+i+" é igual a "+b[i]);
    }
}
```

Exemplo 3 – leitura de valores v2

```
static void Main(string[] args)
{
    int i;
    int[] b = new int[5];
    Console.WriteLine("------ - Leitura dos valores:-------");
    for (i = 0; i < 5; i++)
    {
        Console.WriteLine("Digite o valor " + i + ":");
        b[i] = int.Parse(Console.ReadLine());
    }


    for (i = 0; i < 5; i++)
    {
        Console.WriteLine("Valor índice " + i + " é igual a " + b[i]);
    }
}
```

Exemplo 4 – porque usar vetores?

```
static void Main(string[] args)
{
    double media;
    double nota1, nota2, nota3, nota4, nota5, nota6, nota7, nota8, nota9, nota10;
    double nota11, nota12, nota13, nota14, nota15, nota16, nota17, nota18, nota19, nota20;
    double nota21, nota22, nota23, nota24, nota25, nota26, nota27, nota28, nota29, nota30;
    double nota31, nota32, nota33, nota34, nota35, nota36, nota37, nota38, nota39, nota40;
    double nota41, nota42, nota43, nota44, nota45, nota46, nota47, nota48, nota49, nota50;
    Console.WriteLine("Digite a nota do aluno 1:");
    nota1 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 2:");
    nota2 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 3:");
    nota3 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 4:");
    nota4 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 5:");
    nota5 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 6:");
    nota6 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 7:");
    nota7 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 8:");
    nota8 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 9:");
    nota9 = double.Parse(Console.ReadLine());
    Console.WriteLine("Digite a nota do aluno 10:");
    nota10 = double.Parse(Console.ReadLine());

    media = nota1 + nota2 + nota3 + nota4 + nota5 + nota6 + nota7 + nota8 + nota9 + nota10 +
            nota11 + nota12 + nota13 + nota14 + nota15 + nota16 + nota17 + nota18 + nota19 + nota20 +
            nota21 + nota22 + nota23 + nota24 + nota25 + nota26 + nota27 + nota28 + nota29 + nota30 +
            nota31 + nota32 + nota33 + nota34 + nota35 + nota36 + nota37 + nota38 + nota39 + nota40 +
            nota41 + nota42 + nota43 + nota44 + nota45 + nota46 + nota47 + nota48 + nota49 + nota50;


    Console.WriteLine("-------Mostra a media:-------");
    Console.WriteLine("A media de notas é igual " + media);

}
```

Exemplo 4v2 – porque usar vetores?

```
static void Main(string[] args)
{
    double[] notas = new double[50];
    double media;
    int i;

    Console.WriteLine("-------Leitura dos valores:-------");
    for (i = 0; i < 50; i++)
    {
        Console.WriteLine("Digite a nota do aluno "+i+":");
        notas[i] = double.Parse(Console.ReadLine());
    }

    media = 0;
    Console.WriteLine("-------Calculo da media sendo feito:-------");
    for (i = 0; i < 50; i++)
    {
        media += notas[i];
    }
    media = media / 50;

    Console.WriteLine("-------Mostra a media:-------");
    Console.WriteLine("A media de notas é igual "+media);

}
```

Exemplo 4v3 – aprimorado

```
static void Main(string[] args)
{
    double[] notas = new double[50];
    double media = 0;
    int i;

    Console.WriteLine("-------Leitura dos valores:-------");
    for (i = 0; i < 50; i++)
    {
        Console.WriteLine("Digite a nota do aluno " + i + ":");
        notas[i] = double.Parse(Console.ReadLine());
        media += notas[i];
    }
    media = media / 50;

    Console.WriteLine("-------Mostra a media:-------");
    Console.WriteLine("A media de notas é igual " + media);

}
```

Exemplo 5 – maiores de 18

```
static void Main(string[] args)
{
    int[] idade = new int[40];
    int i, conta = 0;
    Console.WriteLine("Informe as idades dos alunos da turma:");
    for (i = 0; i < 40; i++)
    {
        idade[i] = int.Parse(Console.ReadLine());
        if (idade[i] >= 18)
        {
            conta++;
        }
    }
    Console.WriteLine("Existem "+conta+"alunos com idade maior ou igual a 18 anos");
    for (i = 0; i < 40; i++)
    {
        if (idade[i] >= 18)
        {
            Console.WriteLine("aluno "+i+" tem "+idade[i]+" anos");
        }
    }
}
```

Exemplo 6 – maior valor do vetor

```
static void Main(string[] args)
{
    int[] vetor = new int[10];
    int i, maior = -1000000;
    for (i = 0; i < 10; i++)
    {
        Console.WriteLine("Digite o valor "+i);
        vetor[i] = int.Parse(Console.ReadLine());
        if (vetor[i] > maior)
        {
            maior = vetor[i];
        }
    }
    Console.WriteLine("O maior valor do vetor é = "+maior);

}
```

Exemplo 6v2 – maior valor do vetor

```
static void Main(string[] args)
{
    int[] vetor = new int[10];
    int i, maior = -1000000, maiorIndice = 0;
    for (i = 0; i < 10; i++)
    {
        Console.WriteLine("Digite o valor " + i);
        vetor[i] = int.Parse(Console.ReadLine());
        if (vetor[i] > maior)
        {
            maior = vetor[i];
            maiorIndice = i;
        }
    }
    Console.WriteLine("O maior valor do vetor é = " + vetor[maiorIndice]);

}
```

Exemplo 7 – vetor A e vetor B

```
static void Main(string[] args)
{
    int[] vetorA = new int[15], vetorB = new int[15];
    int i;
    Console.WriteLine("-------Leitura dos elementos de A-------");
    for (i = 0; i < 15; i++)
    {
        Console.WriteLine("Digite o valor " + i);
        vetorA[i] = int.Parse(Console.ReadLine());
        if (vetorA[i] % 2 == 0)
        {
            vetorB[i] = vetorA[i] * 5;
        }
        else
        {
            vetorB[i] = vetorA[i] + 5;
        }

    }
    for (i = 0; i < 15; i++)
    {
        Console.WriteLine("VetorB["+i+"] ="+vetorB[i]);
    }
}
```

Exemplo 8 – vetor A e vetor B

```
static void Main(string[] args)
{
    int[] vetorA = new int[15], vetorB = new int[15];
    int i, prod = 0;
    Console.WriteLine("-------Leitura dos elementos de A-------");
    for (i = 0; i < 15; i++)
    {
        Console.WriteLine("Digite o valor "+i);
        vetorA[i] = int.Parse(Console.ReadLine());
    }
    Console.WriteLine("-------Leitura dos elementos de B-------");
    for (i = 0; i < 15; i++)
    {
        Console.WriteLine("Digite o valor "+i);
        vetorB[i] = int.Parse(Console.ReadLine());
    }
    for (i = 0; i < 15; i++)
    {
        prod += vetorA[i] * vetorB[i];
    }
    Console.WriteLine("O produto ="+prod);

}
```