

# Laços de repetição e vetores em C#.

Jonathan Vinícius Suter

## 1 Para que servem os laços de repetição?

Servem para fazer iterações e repetir a execução de determinado trecho de código.

## 2 Utilização

### 2.1 For

O for é a estrutura de repetição mais simples de usar, conforme abaixo:

```
class Program
{
    static void Main(string[] args)
    {
        //for
        //para i igual 0, itere i enquanto este for menor 1000 e mostre ele na tela.
        for (int i = 0; i < 1000; i++)
        {
            Console.WriteLine("Valor de i: "+i);
        }
    }
}
```

### 2.2 While

Conforme o próprio nome, esta estrutura repete enquanto uma determinada condição for verdadeira:

```

class Program
{
    static void Main(string[] args)
    {
        int j = 0;
        //while
        //enquanto j for menor que 10000, mostre o valor de j na tela.
        while (j<10000)
        {
            Console.WriteLine("Valor de j: " + j);
            j++;
        }
    }
}

```

## 2.3 Do-While

Este iterador funciona de maneira semelhante ao *While*, porém, em comparação com o *while*, primeiro ele executa a operação e depois verifica se a condição será satisfeita e poderá ser feita mais uma execução.

```

class Program
{
    static void Main(string[] args)
    {
        int a = 1000;
        //do-while
        //mostre na tela o valor de "a" enquanto este for maior que zero.
        do
        {
            Console.WriteLine("Valor de a: " + a);
            a--;
        }
        while (a > 0);
    }
}

```

## 2.4 Foreach

É um iterador mais voltado para listas de objetos, como uma *string*. Com este iterador, só é possível iterar uma cadeia já preenchida, não sendo possível atribuir um valor a variável na posição atual.

```

class Program
{
    static void Main(string[] args)
    {
        string texto = "Uma cadeia de textos é percorrida assim";
        //foreach
        //para itere para cada char "b" na cadeia de caracteres
        foreach (char b in texto)
        {
            //texto.IndexOf(b) encontra a primeira ocorrência da letra em "b" e retorna
            //sua posição no array.
            Console.WriteLine($"Texto na posição {texto.IndexOf(b)}, {b}" );
        }
    }
}

```

### 3 Vetores

Os vetores são um tipo de variável que permite o uso de mais de um espaço de memória. Abaixo alguns exemplos de uso:

```
class Program
{
    static void Main(string[] args)
    {
        int[] vetorInteiro = new int[50];
        char[] vetorChar = new char[10];
        int iVetorChar = 0;
        int vetorCharLenght = vetorChar.Length;
        string[] vetorNomes = new string[10];
        int iVetorNomes = 0;

        //preenchendo um vetor usando for
        for(int i = 0; i < vetorInteiro.Length; i++)
        {
            Console.WriteLine("Insira um número");
            vetorInteiro[i] = Convert.ToInt32(Console.ReadLine());
        }

        //preenchendo um vetor usando while

        while (iVetorChar < vetorCharLenght)
        {
            Console.WriteLine("Insira um caractere");
            vetorChar[iVetorChar] = Convert.ToChar(Console.ReadLine());
            iVetorChar++;
        }

        //preenchendo um vetor usando do-while

        do
        {
            iVetorNomes++;
            Console.WriteLine("Insira um nome");
            vetorNomes[iVetorNomes] = Console.ReadLine();
        }
        while (iVetorNomes < vetorNomes.Length);
    }
}
```

## **4 Exercícios**

**4.1 Escreva um programa que receba dez nomes de pessoas e apresente-os na tela;**

**4.2 Faça um programa que receba um número indefinido de entradas e calcule a média de todas elas. Quando o usuário inserir um valor negativo, o programa encerra e mostra a média.**

**4.3 Faça um programa que receba as notas de 10 alunos e mostre a média das notas, a maior e a menor.**

**4.4 Faça um programa que receba o número de entradas (as entradas devem ser de valores inteiros), mostre se há elementos repetidos e quantas vezes estes se repetem.**

**4.5 Faça um programa que receba 10 entradas de textos e mostre todos eles intercalados do caractere “—OU—”.**

**4.6 Faça um programa que receba 5 valores em dois vetores e mostre o produto entre cada célula de cada vetor (multiplique a célula um do vetor “A” pela célula um do vetor “B” e assim sucessivamente), armazene o resultado em um terceiro vetor e mostre os resultados deste.**

**4.7 Faça um programa que tenha um menu para fazer operações entre contas bancárias. Para as contas, usar dois *arrays* de 5 posições, em um registrando a conta da pessoa e no outro, os valores iniciais de cada conta (2 vetores, cada um com 5 posições, um recebendo a conta e outro os valores respectivos das contas). Deve ser possível fazer as seguintes operações no menu, de acordo com a entrada do usuário:**

**1 - Sacar: o usuário deve indicar a conta da qual fará o saque (esta deve possuir saldo suficiente para a operação);**

**2 - Depositar: o usuário deve indicar para qual conta irá o valor;**

**3 - Transferir: o usuário deve indicar de qual conta sairá (esta deve ter saldo suficiente para a operação), o valor da transferência e para qual conta irá os valores.**

**0 - Para finalizar a execução do programa;**