



Vinicius Santos Turma 01 .NET
06.04.2022

1. Quais são considerados loops no C#?



while



do while



for



foreach

2. Qual é a diferença entre um loop **while** e um loop **do while**?



A O loop while garante a execução ao menos uma vez



B Não possui nenhuma diferença



O loop do while garante a execução ao menos uma vez

A while executa a instrução enquanto booleana for identificada como true né!?

3. Quando declaramos um loop é importante checar uma **condição de parada**, caso não seja cumprido a condição de parada o loop permanecerá **infinitamente**!.



Verdadeiro



Falso

4.

```
for (int i = 0; i < 3; i++)  
{  
    Console.Write(i);  
}
```

Qual é a condição de parada do loop **for** da imagem?



A Quando, i = 4



B Quando, i = 3



C Quando, i = 2

Nesse caso para ele ter uma para em três ele teria que colocar ≤ 3 né!?

5.

```
int n = 0;  
do  
{  
    Console.Write(n);  
    n++;  
} while (n < 5);
```

No código da imagem, qual será a **saída**?



A 01234



B 0123



C 12345

6. O exemplo,
`int[] array = new int[5];`
é considerado uma:

☐ A

Matriz multidimensional



Matriz unidimensional

7. O exemplo,
`int[,] array;`
é considerado uma:



Matriz multidimensional



Matriz unidimensional

8.

```
int[] numbers = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };  
foreach (int number in numbers)  
{  
    if (number == 3)  
    {  
        break;  
    }  
  
    Console.Write($"{number} ");  
}  
Console.WriteLine();  
Console.WriteLine("End of the example.");
```

Qual será a saída do código?

☐ A

0 1 2
End of the example.

☐ B

0 1 2 3 4 5 6 7 8 9
End of the example.



1 2 3
End of the example.

9.

```
for (int i = 0; i < 5; i++)
{
    Console.Write($"Iteration {i}: ");

    if (i < 3)
    {
        Console.WriteLine("skip");
        continue;
    }

    Console.WriteLine("done");
}
```

Qual será a saída do código?

A

Iteration 0: done
Iteration 1: done
Iteration 2: done
Iteration 3: skip
Iteration 4: skip



Iteration 0: skip
Iteration 1: skip
Iteration 2: skip
Iteration 3: done
Iteration 4: done

C

Iteration 0: skip
Iteration 1: done
Iteration 2: done
Iteration 3: done
Iteration 4: done

10. A `foreach` instrução executa uma instrução ou um bloco de instruções para cada elemento que se encontra em um array!



Verdadeiro



Falso