

# Value Types

- A memória do computador tem duas partes principais → HEAP e STACK
- Todos os tipos no C# ficam em um ou nos dois

Reference Types



→ ARMAZENA OS DADOS EM SI

→ Value Types

→ ARMAZENA A REFERÊNCIA AOS DADOS SENDO REFERÊNCIA O ENDEREÇO A

esses valores.

- Value Type → ARMAZENA OS DADOS E FICA NA STACK. ESSES VALORES SÃO ACESSADOS DIRETAMENTE, CASO OUTRA VARIÁVEL RECEBA O VALOR DESSA VARIÁVEL ELE SERÁ COPIADO, SENDO INDEPENDENTES DE SI.

- São os tipos → Built-in, Structs e Enums

- e o Garbage collector ele NÃO ACESSA A STACK

↳ remove itens não usados

- Tomar cuidado com objetos grandes, pois pode impactar performance.

// Value Types

```
int primeiroValor = 10;
int segundoValor = primeiroValor; // 10

primeiroValor = 25;

Console.WriteLine("Primeiro Valor: " + primeiroValor);
Console.WriteLine("Segundo Valor: " + segundoValor);
```

← recebeu a cópia

- Reference Types → Armazena somente o endereço do objeto, onde contém os dados

- ficando no HEAP, apontando p/ mesma info (não sendo independentes)

- Garbage collect passa limpando

↳ classes, objects e arrays.

// Reference types

```
var arr = new string[2]; // Criação de um array
arr[0] = "Item 1"; // atribuição ao primeiro index de um array
var arr2 = arr; // Não cria uma cópia, tudo que for alterado no primeiro array, será modificado nesse também

Console.WriteLine(arr[0] + " " + arr2[0]);

arr[0] = "Novo item";

Console.WriteLine(arr[0] + " " + arr2[0]);
```

→ CRIAÇÃO 1º ARRAY

→ 2º ARRAY, o dependente do primeiro

→ 2º "Novo Item"