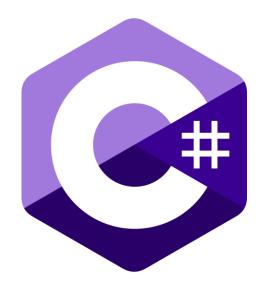
## C# Essencial



Classes e Métodos : Exercício Prático 8 - Resposta

## Resposta

```
public struct Cliente
   public string Nome;
   public string Email;
   private int idade;
   public int Idade
       get { return idade; }
        set
            if (value < 18)
                idade = 18;
                Console.WriteLine("Cliente menor que 18 anos");
            else
                Console.WriteLine("Cliente maior que 18 anos");
                idade = value;
    public Cliente(string nome, string email, int idade)
       Nome = nome;
        Email = email;
        Idade = idade;
   public static void ExibirInfo(string nome, string email, int idade = 18)
       Console.WriteLine($"{nome} - {email} - {idade}");
```

## Resposta

```
//struct
Cliente cliente = new("Maria", "maria@email.com",19);
Cliente.ExibirInfo(email: cliente.Email, nome: cliente.Nome, idade: cliente.Idade);
Console.WriteLine("\nExibindo informação sem informar a idade");
Cliente.ExibirInfo(email: cliente.Email, nome: cliente.Nome);
```

As structs são usadas para estruturar dados comuns em um mesmo contexto e são semelhantes as classes. São definidas usando a palavra-chave struct.

A principal diferença é que as structs são tipos de valor, alocados na Stack enquanto as classes são tipos de referência, alocados na heap e coletadas pelo coletor de lixo (*Garbage Collector*).

Uma struct é útil para situações nas quais precisamos de poucas variáveis, pois os objetos criados com base em structs são mais leves e por ser do tipo de valor não trabalham com referências e são desalocadas da memória Stack após sua utilização.