# C#
# Estruturas de Decisão

## Estrutura IF (Se) e IF-ELSE (Se-Senão)

### IF

- if (*<condição>*)
  *<comando ou bloco>*

- Funcionamento:
  - Condição verdadeira: o comando ou o conjunto de comandos é executado
  - Condição falsa: nenhum comando é executado
    - Pode ser que não seja executado qualquer comando

### IF – ELSE

- if (*<condição>*)
  *<comando ou bloco>*
  else
  *<comando ou bloco>*

- Funcionamento:
  - Condição verdadeira: apenas o primeiro conjunto é executado
  - Condição falsa: apenas o segundo conjunto é executado
    - Algum comando sempre será executado!

## Empregando IF's de modo seqüencial

```
if (<condição1>)
   <comando ou bloco>
else
   <comando ou bloco>
if (<condição2>)
   <comando ou bloco>
else
   <comando ou bloco>
:
if (<condiçãoN>)
   <comando ou bloco>
else
   <comando ou bloco>
```

- Supondo que tenhamos um trecho de código que apresente uma sucessão de IF's, sua execução implicará obrigatoriamente na avaliação de cada uma das condições de maneira independente umas das outras.
- Note que aqui a execução de qualquer um dos blocos de comandos do segundo IF independe completamente da execução do primeiro IF

## Empregando IF's de modo encadeado

```
if (<condição1>)
   <comando ou bloco>
else if (<condição2>)
   <comando ou bloco>
else if (<condição3>)
   <comando ou bloco>
:
else if (<condiçãoN>)
   <comando ou bloco>
else
   <comando ou bloco>
```

- Observe que o agrupamento de comandos IF's de forma encadeada permite que N condições sejam avaliadas seqüencialmente mas com um resultado de execução bastante próprio.
- Neste arranjo, assim que o fluxo de execução encontrar uma condição verdadeira, todas as demais avaliações não serão avaliadas.

## Empregando IF's de modo aninhado

```
if (<condição1>) {
    if (<condição2>)
        <comando ou bloco>
    else
        <comando ou bloco>
}
else
{
    if (<condição3>)
        <comando ou bloco>
    else
        <comando ou bloco>
}
```

- Nesta situação, antes de avaliar a *condição2* já temos certeza de que *condição1* é verdadeira, bem como, antes de avaliarmos a *condição3* já sabemos de antemão que *condição1* é falsa.
- Esse processo é similar a construção:
  (*condição1*) && (*condição2*)
  !(*condição1*) && (*condição3*)
  entretanto, com clara vantagem de não ser necessário avaliar mais de que uma única vez cada condição e automaticamente eliminar a avaliação das outras condições que não nos interessam.

## Exemplo – Média de duas notas

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Condicional
{
    class Program
    {
        static void Main(string[] args)
        {
            //declaração das variáveis
            float n1, n2, media, exame;

            Console.Write("Digite a primeira nota: ");
            n1 = (float) Convert.ToDouble(Console.ReadLine());

            Console.Write("Digite a segunda nota: ");
            n2 = (float) Convert.ToDouble(Console.ReadLine());
```

```
            media = (n1 + n2) / 2;

            if (media < 3.0)
            {
               Console.WriteLine("Aluno Reprovado com media = {0}", media);
            }
            else if (media >= 7.0)
            {
               Console.WriteLine("Aluno Aprovado com media = {0}", media);
            }
            else
            {
               Console.Write("Aluno de Exame com média {0}, informe nota da
   prova: ", media);
               exame = (float) Convert.ToDouble(Console.ReadLine());
               if (exame < 5.0)
                  Console.WriteLine("Aluno Reprovado");
               else
                  Console.WriteLine("Aluno Aprovado");
            }
            Console.ReadKey();
         }
      }
   }
```

## Comando Switch

```
switch (<expressão>) {
   case valor1:
         <comandos>
         break;
   case valor2:
         <comandos>
         break;
   :
   case valorN:
         <comandos>
         break;
   default:
         <comandos>
}
```

- Este comando possibilita que se compare o conteúdo de uma variável para diversos casos de uma vez.
- Os comandos relacionados ao primeiro caso somente serão executados somente se a expressão informada assumir valor igual a *valor1*;
- Os comandos relacionados ao segundo caso, somente se for igual ao *valor2*; e assim por diante.
- Os comandos do bloco *default* somente serão executados se a expressão não combinar com qualquer um dos valores definidos nas clausulas case.
  - O bloco *default* é opcional e portanto pode ser omitido quando desnecessário

# Exemplo

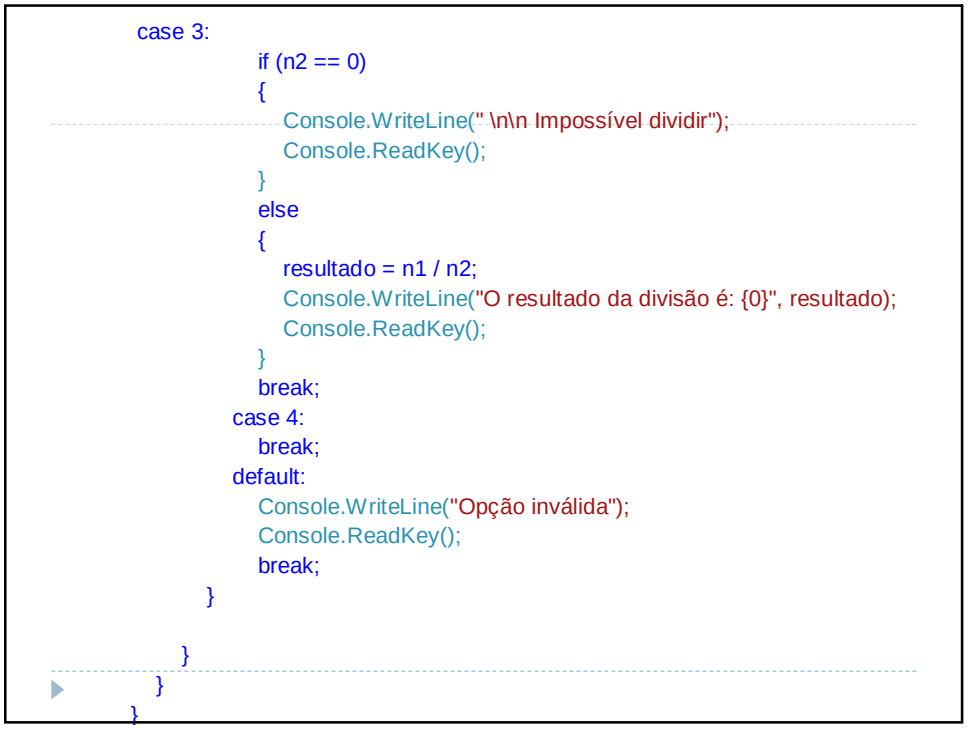
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Switch
{
    class Program
    {
        static void Main(string[] args)
        {
            int opcao;
            float n1, n2, resultado;

            Console.Write("\n\nDigite o primeiro número: ");
            n1 = (float) Convert.ToDouble(Console.ReadLine());
            Console.Write("Digite o segundo número: ");
            n2 = (float) Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("\n\n[1] - Somar os números");
            Console.WriteLine("[2] - Multiplicar os números");
            Console.WriteLine("[3] - Dividir os números");
            Console.WriteLine("[4] - Sair do Programa");
            Console.Write("Selecione uma opção -> ");
            opcao = Convert.ToInt32(Console.ReadLine());
```

```
switch (opcao)
        {
            case 1:
                resultado = n1 + n2;
                Console.WriteLine("O resultado da soma é: {0}",
resultado);
                Console.ReadKey();
                break;
            case 2:
                resultado = n1 * n2;
                Console.WriteLine("O resultado da multiplicação é: {0}",
resultado);
                Console.ReadKey();
                break;
```



```
            case 3:
                if (n2 == 0)
                {
                    Console.WriteLine(" \n\n Impossível dividir");
                    Console.ReadKey();
                }
                else
                {
                    resultado = n1 / n2;
                    Console.WriteLine("O resultado da divisão é: {0}", resultado);
                    Console.ReadKey();
                }
                break;
            case 4:
                break;
            default:
                Console.WriteLine("Opção inválida");
                Console.ReadKey();
                break;
        }

    }
}
}
```