

Upgrading Skills

Primavera
ACADEMY

EXERCÍCIOS RESOLVIDOS

Curso de Programming C# - Foundations

NET framework e IDE Microsoft Visual Studio

Índice

| | |
|--|----|
| MÓDULO 4 – LEITURA E ESCRITA DE DADOS..... | 3 |
| MÓDULO 5 – INSTRUÇÕES DE DECISÃO | 13 |
| MÓDULO 6 – INSTRUÇÕES DE REPETIÇÃO | 27 |
| MÓDULO 7 – MÉTODOS..... | 38 |
| MÓDULO 8 – VETORES E MATRIZES | 44 |
| MÓDULO 9 – <i>STRINGS</i> | 49 |
| EXERCÍCIO GLOBAL – JOGO DO GALO..... | 54 |
| MÓDULO 11 – POO | 60 |
| A) EXERCÍCIOS PPT..... | 60 |
| B) LOJA DE INFORMÁTICA | 65 |

MÓDULO 4 – Leitura e escrita de dados

1. Apresentar a área de um retângulo a partir de dois valores (altura e largura) introduzidos pelo utilizador.

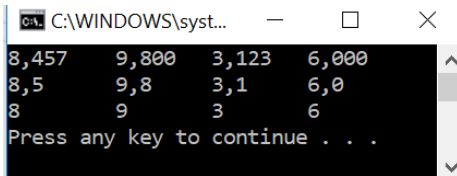
*Nota: area = altura * largura*

```
using System;
namespace M6_1_area
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, l;
            Console.WriteLine("Introduzir a altura:");
            a = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Introduzir a largura:");
            l = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("A area do retângulo é de {0}.", a * l);
        }
    }
}
```

2. Refazer o exemplo 3b com outros valores.
3. Escrever um programa que apresente no ecrã o resultado das expressões:
 - › 15+2
 - › 15/2
 - › Resto da divisão de 15 por 2.

```
using System;
namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("15 + 2 = {0} ", 15 + 2);
            Console.WriteLine("15 / 2 = {0} ", 15 / 2);
            Console.WriteLine("15 % 2 = {0} ", 15 % 2);
        }
    }
}
```

4. Escreva um programa que apresente os números (8.456796, 9.8, 3.12345, 6) arredondados em três linhas diferentes. sendo que na primeira linha aparecem todos arredondados a três casas decimais, na segunda a uma e na terceira a sem casa decimal.



```

8,457 9,800 3,123 6,000
8,5 9,8 3,1 6,0
Press any key to continue . . .

```

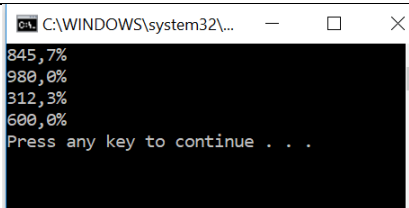
```

using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("{0:F3} \t {1:F3} \t {2:F3} \t {3:F3}", 8.456796, 9.8, 3.12345, 6); //Três casas decimais
            Console.WriteLine("{0:F1} \t {1:F1} \t {2:F1} \t {3:F1}", 8.456796, 9.8, 3.12345, 6); // Uma casa decimal
            Console.WriteLine("{0} \t {1} \t {2} \t {3}", (int)8.456796, (int)9.8, (int)3.12345, 6); // sem casas decimais... utilizou-se uma conversão de tipo (type cast)
        }
    }
}

```

5. Escrever um programa que apresente os números do exercício anterior em percentagem.



```

845,7% 980,0% 312,3% 600,0%
Press any key to continue . . .

```

```

using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("{0:P1}\n{1:P1}\n{2:P1}\n{3:P1}", 8.456796, 9.8, 3.12345, 6); // \n permite fazer mudança de linha
        }
    }
}

```

6. Escrever um programa que leia do utilizador 3 números e os apresente dois resultados diferentes:
- › Na 1ª linha, os números estão alinhados à esquerda, com duas casas decimais e cada um com 20 casas.

- Na 2ª linha, os números estão alinhados à direita, com uma casa decimal e cada um com 20 casas.

```
using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            float a, b, c;
            Console.Write("Introduza um número: ");
            a = Convert.ToSingle(Console.ReadLine());
            Console.Write("Introduza um número: ");
            b = Convert.ToSingle(Console.ReadLine());
            Console.Write("Introduza um número: ");
            c = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine("{0,-20:F1}{1,-20:F1}{2,-20:F1}", a, b, c);
            // São reservados 20 caracteres para escrever cada número.
            // O alinhamento é à esquerda faz-se com o sinal - (menos).
            Console.WriteLine("{0,20:F1}{1,20:F1}{2,20:F1}", a, b, c);
            // São reservados 20 caracteres para escrever cada número.
            // O alinhamento é à direita faz-se por omissão.
        }
    }
}
```

- Escrever um programa que apresente no ecrã a parte inteira de um número introduzido pelo utilizador.

```
using System;

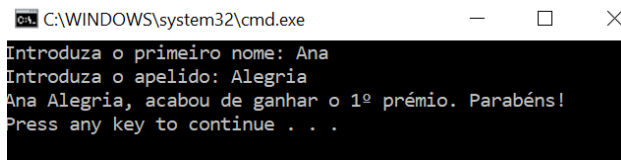
namespace Cap4
{
    class Program
    {
        static void Main(string[] args)
        {
            float n;
            Console.Write("Introduza um número real: "); // Não há validação! Só orientação
            para o utilizador escrever um número real

            n = Convert.ToSingle(Console.ReadLine()); // Atenção que se a consola for em PT,
            o separador decimal é a vírgula
        }
    }
}
```

```
        Console.WriteLine("A parte inteira de {0} é {1}", n, (int)n); // utilizou-se uma
        conversão de tipo (type cast)
    }
}
```

9. Escrever uma mensagem que peça ao utilizador o primeiro nome e o apelido e apresente o seguinte resultado:

<nome> <apelido>, acabou de ganhar o 1º prémio. Parabéns!

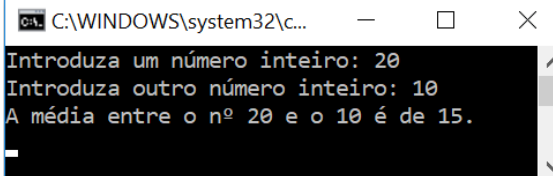


```
C:\WINDOWS\system32\cmd.exe
Introduza o primeiro nome: Ana
Introduza o apelido: Alegria
Ana Alegria, acabou de ganhar o 1º prémio. Parabéns!
Press any key to continue . . .
```

```
using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            string nome, apelido;
            Console.Write("Introduza o 1º nome: ");
            nome = Console.ReadLine();
            Console.Write("Introduza o apelido: ");
            apelido = Console.ReadLine();
            Console.WriteLine("{0} {1}, acabou de ganhar o 1º prémio. Parabéns!", nome, apelido);
        }
    }
}
```

10. Criar uma aplicação do tipo consola para receber do utilizador dois valores inteiros e devolver a média aritmética.



```
C:\WINDOWS\system32\c...
Introduza um número inteiro: 20
Introduza outro número inteiro: 10
A média entre o nº 20 e o 10 é de 15.
```

```

1  using System;
2
3  namespace m2_ex2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int a, b;
10             double media;
11
12             Console.Write("Introduza um número inteiro: ");
13             a = Convert.ToInt32(Console.ReadLine());
14             Console.Write("Introduza outro número inteiro: ");
15             b = Convert.ToInt32(Console.ReadLine());
16             media = (a + b)/2;
17             Console.WriteLine("A média entre o nº {0} e o {1} é de {2}.", a, b, media);
18             Console.ReadLine();
19             /*A última instrução foi acrescentada para o utilizador conseguir
20             visualizar o resultado antes da consola fechar.*/
21         }
22     }
23 }

```

11. Escreva um programa que converta dólares americanos para euros (a taxa de câmbio é de 1,1579 USD para 1€).

```

using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            float d, t = 1.1579F;
            // Taxa de câmbio: 1,1579 - a letra F no fim do número significa que é do tipo
            float //sem o F seria considerado do tipo double e originaria um erro na atribuição

            Console.Write("Quantos dólares pretende converter: ");
            // Não é feita validação, ou seja, o utilizador pode introduzir caracteres
            inválidos

            d = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine("${0}={1}EUR", d, d * t);
        }
    }
}

```

12. Escreva um programa que converta a temperatura lida em graus *Fahrenheit* para graus *Celsius*.

A fórmula é $C = 5/9 * (f - 32)$.

```

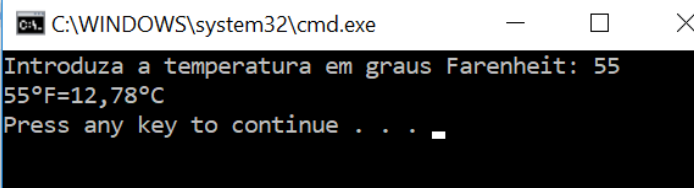
using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            float f;
            Console.Write("Introduza a temperatura em graus Fahrenheit: ");

```

```
// Não é feita validação, ou seja, o utilizador pode introduzir caracteres
inválidos
f = Convert.ToSingle(Console.ReadLine());
Console.WriteLine("{0}°F={1:F2}°C", f, 5 / 9 * (f - 32));
//A divisão 5/9F é real porque 9 é do tipo float, sem a letra F a divisão seria
inteira e conduziria a um resultado errado
// Para que a divisão 5/9 seja real o numerador ou o denominador têm que ser reais
    }
}
}
```

13. Escrever um programa que leia o preço base de um determinado produto e calcule o seu valor de venda ao público (ou seja, valor acrescido da taxa de IVA a 23%).



```
using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            float f;
            Console.Write("Introduza a temperatura em graus Farenheit: ");
            // Não é feita validação, ou seja, o utilizador pode introduzir caracteres
            inválidos
            f = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine("{0}°F={1:F2}°C", f, 5/9F * (f - 32));
            //A divisão 5/9F é real porque 9 é do tipo float, sem a letra F a divisão seria
            inteira e conduziria a um resultado errado
            // Para que a divisão 5/9 seja real o numerador ou o denominador têm que ser
            reais
        }
    }
}
```

14. Escrever um programa leia o valor base e o iva de um produto e que calcule o valor final do produto.

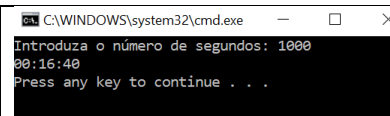
```
using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
```



```
static void Main(string[] args)
{
    float b, iva;
    Console.Write("Introduza o preço base: ");
    b = Convert.ToSingle(Console.ReadLine()); // Não é feita validação, ou seja, o
    utilizador pode introduzir caracteres inválidos
    Console.Write("Introduza a taxa de IVA 0,6 ou 0,13 ou 0,23: "); // Deve ser
    introduzido como um nº real e não em percentagem
    iva = Convert.ToSingle(Console.ReadLine()); // Não é feita validação, ou seja,
    o utilizador pode introduzir caracteres inválidos
    Console.WriteLine("PVP={0:F2}EUR", b * ( 1 + iva));
}
}
```

15. Escrever um programa que converta um valor em segundos para horas, minutos e segundos.



```
using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            int ss, s, m, h;
            Console.Write("Introduza o número de segundos: ");
            // Não é feita validação, ou seja, o utilizador pode introduzir caracteres
            inválidos
            ss = Convert.ToInt32(Console.ReadLine());
            // Todas as divisões efectuadas neste exercício são divisões inteiras
            s = ss % 60; // segundos - % : resto da divisão inteira
            m = (ss / 60) % 60; // minutos
            h = (ss / 60) / 60; // horas
            Console.WriteLine("{0:D2}:{1:D2}:{2:D2}", h, m, s);
        }
    }
}
```

16. Escrever um programa que apresente ao utilizador a sua idade daqui a vinte anos.

```
using System;

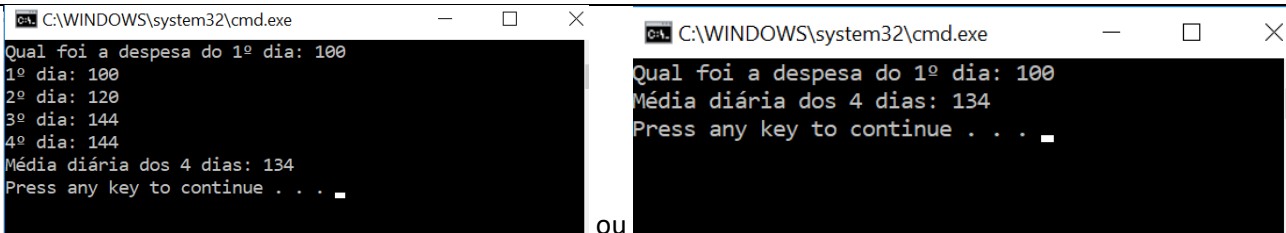
namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            int idade;
```

```

        Console.WriteLine("Introduza a sua idade: ");
        idade = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Daqui a 20 anos terá {0}.", idade + 20);
    }
}

```

17. Escrever um programa que calcule a despesa média diária que um turista despendeu numa viagem de quatro dias ao Porto, sabendo que cada dia gastou mais 20% do que no dia anterior. O valor da despesa deverá ser apresentado sem casas decimais.



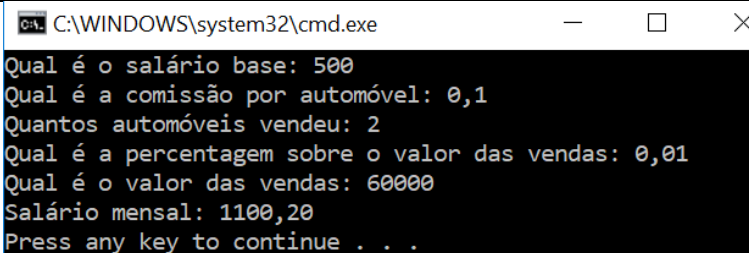
```

using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            double p,t;
            Console.Write("Qual foi a despesa do 1º dia: "); // Não é feita validação, ou
            // seja, o utilizador pode introduzir caracteres inválidos
            p = Convert.ToSingle(Console.ReadLine());
            t = p * (1 + 1.2 + 1.2 * 1.2 + 1.2 * 1.2 * 1.2) / 4;
            Console.WriteLine("1º dia: {0}", p);
            Console.WriteLine("2º dia: {0}", p * 1.2);
            Console.WriteLine("3º dia: {0}", p * 1.2 * 1.2);
            Console.WriteLine("4º dia: {0}", p * 1.2 * 1.2 * 1.2);
            Console.WriteLine("Média diária dos 4 dias: {0:F0} ",t);
        }
    }
}

```

18. Num determinado stand de automóveis, os vendedores ganham um salário mensal base X, uma comissão de Y euros por cada automóvel que vendem e uma percentagem P sobre o valor das vendas V que efetuarem. Escreva um programa que calcule e imprima o salário que um vendedor vai auferir este mês.



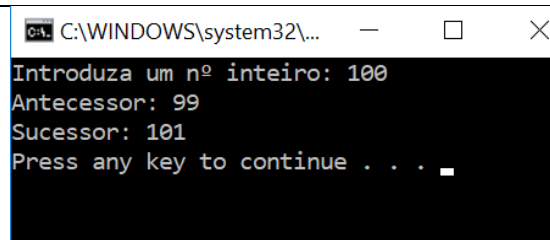
```
namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            double X, Y, P, V, q;
            Console.Write("Qual é o salário base: "); // Não é feita validação, ou seja, o
            //utilizador pode introduzir caracteres inválidos
            X = Convert.ToSingle(Console.ReadLine());
            Console.Write("Qual é a comissão por automóvel: ");
            Y = Convert.ToSingle(Console.ReadLine());
            Console.Write("Quantos automóveis vendeu: ");
            q = Convert.ToSingle(Console.ReadLine());
            Console.Write("Qual é a percentagem sobre o valor das vendas: "); //Introduza um
            //valor decimal
            P = Convert.ToSingle(Console.ReadLine());
            Console.Write("Qual é o valor das vendas: ");
            V = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine("Salário mensal: {0:F2} ", X + q * Y + P * V);
        }
    }
}
```

19. Escrever um programa que troque entre si o valor contido em duas variáveis inteiras a e b, ou seja, o valor contido em a passa a ser o valor de b e vice-versa, sem utilizar uma terceira variável.

```
using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            int a = 4, b = 5;
            Console.WriteLine("a:{0}\tb:{1}", a, b);
            //faça execução passo a passo (F10) para ver a evolução do conteúdo das variáveis
            a = a + b; // a=9 b=5
            b = a - b; // a=9 b=4
            a = a - b; // a=5 b=4
            Console.WriteLine("a:{0}\tb:{1}", a, b);
        }
    }
}
```

20. Escreva um programa que apresente no ecrã antecessor e sucessor de um número introduzido pelo utilizador.



```
C:\WINDOWS\system32\...
Introduza um nº inteiro: 100
Antecessor: 99
Sucessor: 101
Press any key to continue . . .
```

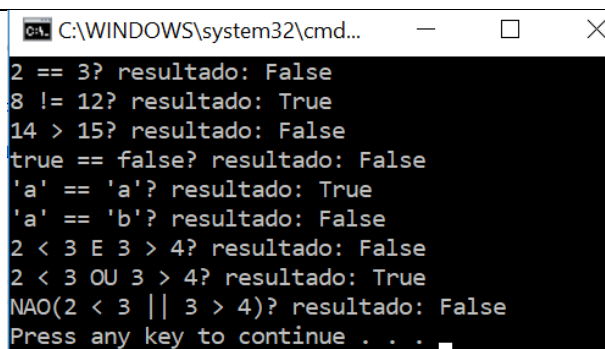
```
using System;

namespace M4_ExerciciosFinais
{
    class Program
    {
        static void Main(string[] args)
        {
            int n;
            Console.Write("Introduza um nº inteiro: "); // Não é feita validação, ou seja, o
            // utilizador pode introduzir caracteres inválidos
            n = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Antecessor: {0} \nSucessor: {1}", n - 1, n + 1);
        }
    }
}
```

MÓDULO 5 – Instruções de decisão

1. Escrever um programa que apresente o resultado das seguintes operações lógicas:

```
2 == 3
8 != 12
14 > 15
true == false
'a' == 'a'
'a' == 'b'
2 < 3 E 3 > 4
2 < 3 OU 3 > 4
NÃO (2 < 3 OU 3 > 4)
```

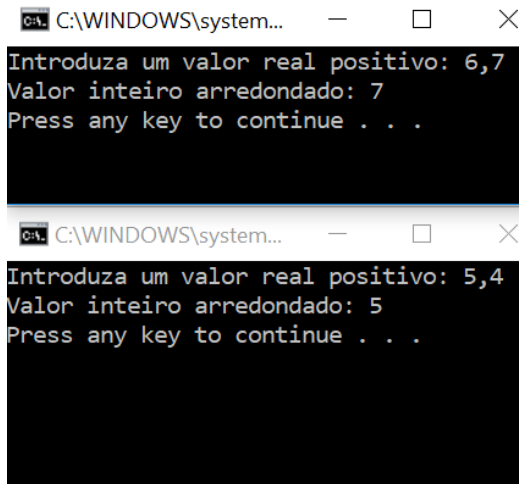


```
C:\WINDOWS\system32\cmd...
2 == 3? resultado: False
8 != 12? resultado: True
14 > 15? resultado: False
true == false? resultado: False
'a' == 'a'? resultado: True
'a' == 'b'? resultado: False
2 < 3 E 3 > 4? resultado: False
2 < 3 OU 3 > 4? resultado: True
NAO(2 < 3 || 3 > 4)? resultado: False
Press any key to continue . . .
```

```
using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("2 == 3? resultado: {0}", 2 == 3);
            Console.WriteLine("8 != 12? resultado: {0}", 8 != 12);
            Console.WriteLine("14 > 15? resultado: {0}", 14 > 15);
            Console.WriteLine("true == false? resultado: {0}", true == false);
            Console.WriteLine("'a' == 'a'? resultado: {0}", 'a' == 'a');
            Console.WriteLine("'a' == 'b'? resultado: {0}", 'a' == 'b');
            Console.WriteLine("2 < 3 E 3 > 4? resultado: {0}", 2 < 3 && 3 > 4);
            Console.WriteLine("2 < 3 OU 3 > 4? resultado: {0}", 2 < 3 || 3 > 4);
            Console.WriteLine("NAO(2 < 3 || 3 > 4)? resultado: {0}", !(2 < 3 || 3 > 4));
        }
    }
}
```

2. Escrever um programa para converter um número real positivo para um número inteiro. Deve ainda de arredondar o número.



```

C:\WINDOWS\system...
Introduza um valor real positivo: 6,7
Valor inteiro arredondado: 7
Press any key to continue . . .

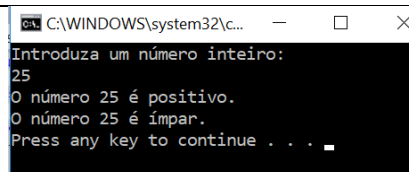
C:\WINDOWS\system...
Introduza um valor real positivo: 5,4
Valor inteiro arredondado: 5
Press any key to continue . . .
  
```

```

using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            float numero;
            int numArredondado;
            Console.Write("Introduza um valor real positivo: "); // Não é feita validação.
            numero = Convert.ToSingle(Console.ReadLine());
            if (numero - (int)numero < 0.5) // (int)numero representa a parte inteira
            {
                numArredondado = (int)numero; // arredonda por defeito
            }
            else
            {
                numArredondado = (int)numero + 1; // // ELSE arredonda por excesso
            }
            Console.WriteLine("Valor inteiro arredondado: {0}", numArredondado);
        }
    }
}
  
```

3. Escrever um programa para determinar se um número é par e positivo.



```

C:\WINDOWS\system32\c...
Introduza um número inteiro:
25
O número 25 é positivo.
O número 25 é impar.
Press any key to continue . . .
  
```

```

// Versao 1

using System;

namespace M5_Exercicios
{
    class Program
  
```

```
{
    static void Main(string[] args)
    {
        int numero;
        Console.WriteLine("Introduza um número inteiro: ");
        numero = Convert.ToInt32(Console.ReadLine());
        if (numero >= 0)
        {
            Console.WriteLine("O número {0} é positivo.", numero);
        }
        else
        {
            Console.WriteLine("O número {0} é negativo.", numero);
        }
        if (numero % 2 == 0)
        {
            Console.WriteLine("O número {0} é par.", numero);
        }
        else
        {
            Console.WriteLine("O número {0} é ímpar.", numero);
        }
    }
}

// Versao 2 c/ op ternário

using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int numero;
            Console.WriteLine("Introduza um número inteiro: ");
            numero = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("{0}", numero >= 0 ? "é positivo" : "é negativo");
            Console.WriteLine("{0}", numero % 2 == 0 ? "é par" : "é ímpar");
        }
    }
}

// Versao 3
using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            Console.Write("Introduza um número inteiro: ");
            i = Convert.ToInt32(Console.ReadLine());
            if (i > 0) // Esta solução, após determinar se é positivo, distingue se é ou não par
```

```

        if (i % 2 == 0)    // Poderiam ter-se utilizado apenas dois if mas o Português não
seria tão correto
            Console.WriteLine("O número inteiro {0} é positivo e par", i);
        else
            Console.WriteLine("O número inteiro {0} é positivo mas não é par", i);
    else
        if (i % 2 == 0)
            Console.WriteLine("O número inteiro {0} não é positivo mas é par", i);
        else
            Console.WriteLine("O número inteiro {0} não é positivo nem par", i);
    }
}
}

```

4. Escrever um programa para calcular o salário semanal a pagar a um empregado, tendo em atenção que todas as horas após as 40 serão pagas a dobrar. O utilizador deve de indicar o valor do salário por hora e o valor das horas trabalhadas.

```

using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            float horas, salarioHora, salario;
            Console.Write("Introduza um número de horas trabalhadas: ");
            horas = Convert.ToSingle(Console.ReadLine());
            Console.Write("Introduza o valor do salário por hora: ");
            salarioHora = Convert.ToSingle(Console.ReadLine());
            if (horas <= 40) // Foram trabalhadas menos de 40 horas?
            {
                salario = horas * salarioHora; // Não há horas extraordinárias
            }
            else
            {
                salario = 40 * salarioHora + (horas - 40) * salarioHora * 2;
                // As primeiras 40 horas são pagas ao preço normal --> (40 * sh),
                // as horas para além das 40 --> (horas - 40), são pagas a dobrar --> (salarioHora * 2)
                // alternativa à instrução anterior --> salario = salarioHora*(2*horas-40)
                Console.WriteLine("Salário semanal: {0}", salario);
            }
        }
    }
}

```

5. Escrever um programa que permita identificar o número maior entre três números introduzidos pelo utilizador.

```

using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {

```



```
int a, b, c;
Console.WriteLine("Introduza o primeiro valor: ");
a = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Introduza o segundo valor: ");
b = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Introduza o terceiro valor: ");
c = Convert.ToInt32(Console.ReadLine());
if (a > b)
{
    if (a > c)
    {
        Console.WriteLine("Maior valor: {0}", a);
    }
    else
    {
        if (b > c)
        {
            Console.WriteLine("Maior valor: {0}", b);
        }
        else
        {
            Console.WriteLine("Maior valor: {0}", c);
        }
    }
}
else
{
    if (b > c)
    {
        Console.WriteLine("Maior valor: {0}", b);
    }
    else
    {
        if (a > c)
        {
            Console.WriteLine("Maior valor: {0}", a);
        }
        else
        {
            Console.WriteLine("Maior valor: {0}", c);
        }
    }
}
}
```

6. Escrever um programa que determine a aprovação de um formando numa ação de formação. Deve ter em atenção que só é considerado aprovado se a obtiver classificação mínima de 8 valores em dois testes e cada um tem um peso de 10 valores na nota final.

```
using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
```

```

float teste1, teste2;
Console.WriteLine("Introduza a nota 1ª teste: ");
teste1 = Convert.ToSingle(Console.ReadLine());
Console.WriteLine("Introduza a nota 2ª teste: ");
teste2 = Convert.ToSingle(Console.ReadLine());
if (teste1 >= 8 && teste2 >= 8 && (teste1 + teste2) / 2 >= 9.5)
{
    Console.WriteLine("Aprovado");
}
else
{
    Console.WriteLine("Reprovado");
}
}
}
}

```

7. Escrever um programa, utilizando o operador ternário, que apresente a mensagem de “Parabéns” se um formando obteve uma nota superior a 10. No caso contrário, apresente a mensagem “Ups, deve marcar novo exame.”

```

using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int nota;
            Console.WriteLine("Introduzir a nota final");
            nota = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine(nota >= 10 ? "Parabéns!" : "Ups, deve marcar novo exame.");
            //OU
            //Console.WriteLine("{0}", nota >= 10 ? "Parabéns!" : "Ups, deve marcar novo exame.");
        }
    }
}

```

8. Escrever um programa que determine se o ano introduzido é bissexto.

NOTA:

Anos divisíveis por 4 ou 400 são bissextos, LOGO: ano % 400 == 0 || ano % 4 == 0
 E ano não divisível por 100 são bissextos, LOGO: ano % 100 != 0

```

using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int ano;
            Console.WriteLine("Introduza o ano: ");
            ano = Convert.ToInt32(Console.ReadLine());

```

```
        if ((ano % 400 == 0 || ano % 4 == 0) && ano % 100 != 0) // Note que os parênteses na
conjunção (&&) são desnecessários já que este operador tem maior precedência que a disjunção (||)
        {
            Console.WriteLine("O ano {0} é bissexto", ano);
        }
        else
        {
            Console.WriteLine("O ano {0} não é bissexto", ano);
        }
    }
}
```

9. Escrever um programa que devolva ao utilizador, por ordem crescente, os três números introduzidos inicialmente.

```
using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int p, s, t;
            Console.Write("Introduza o primeiro valor: ");
            p = Convert.ToInt32(Console.ReadLine());
            Console.Write("Introduza o segundo valor: ");
            s = Convert.ToInt32(Console.ReadLine());
            Console.Write("Introduza o terceiro valor: ");
            t = Convert.ToInt32(Console.ReadLine());
            if (p < s)
            {
                if (p < t)
                {
                    if (s < t)
                    {
                        Console.WriteLine("{0} {1} {2}", p, s, t);
                    }
                    else
                    {
                        Console.WriteLine("{0} {1} {2}", p, t, s);
                    }
                }
                else
                {
                    Console.WriteLine("{0} {1} {2}", t, p, s);
                }
            }
            else
            {
                if (s < t)
                {
                    if (p < t)
                    {
                        Console.WriteLine("{0} {1} {2}", s, p, t);
                    }
                    else
                }
            }
        }
    }
}
```

```
{
    Console.WriteLine("{0} {1} {2}", s, t, p);
}
else
{
    Console.WriteLine("{0} {1} {2}", t, s, p);
}
}
```

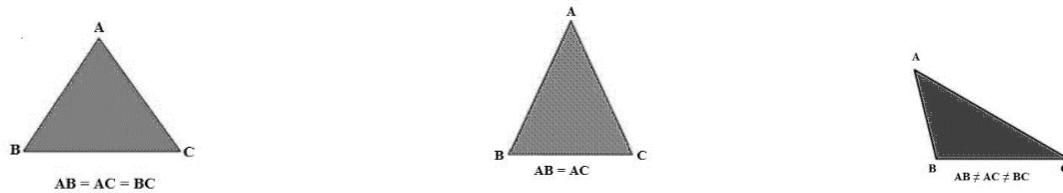
10. Escrever um programa para calcular o valor total a pagar por um determinado artigo, sabendo que o tipo de artigo e preço sem iva. Supondo que a taxa de iva é de 5% para os produtos essenciais, 30% para os bens de luxo e 20% para os restantes.

SWITCH...CASE

```
using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            float ps;
            int ta;
            Console.WriteLine("Preço sem IVA: "); // Não é feita validação, ou seja, o
            // utilizador pode introduzir
            ps = Convert.ToSingle(Console.ReadLine()); // caracteres inválidos e como tal gerar
            // exceções (ERROS)
            Console.WriteLine("** Escolha o tipo de artigo **\n1- bens essenciais\n2- produtos
            // de luxo\n3- restantes");
            ta = Convert.ToInt32(Console.ReadLine());
            switch (ta)
            {
                case 1: Console.WriteLine("Preço com IVA: {0}", ps * (1 + 0.05F)); break;
                case 2: Console.WriteLine("Preço com IVA: {0}", ps * (1 + 0.3F)); break;
                case 3: Console.WriteLine("Preço com IVA: {0}", ps * (1 + 0.2F)); break;
                default: Console.WriteLine("ERRO"); break;
            }
        }
    }
}
```

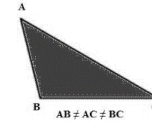
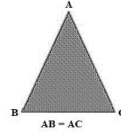
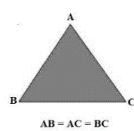
11. Escrever um programa que leia três valores inteiros e verifique se estes podem corresponder aos lados de um triângulo. No caso de se tratar de um triângulo, deve ainda de o classificar como equilátero, isósceles ou escaleno.



Triângulo Equilátero: é todo triângulo que apresenta os três lados com a mesma medida. Nesse caso dizemos que os três lados são congruentes.

Triângulo Isósceles: é todo triângulo que apresenta dois lados com a mesma medida, ou seja, dois lados de tamanhos iguais.

Triângulo Escaleno: é todo triângulo que apresenta os três lados com medidas diferentes, ou seja, três lados de tamanhos diferentes.



In <https://escolakids.uol.com.br/classificacao-dos-triangulos.htm>

```
//Versao 1
using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b, c;
            Console.WriteLine("Introduza o primeiro valor: ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Introduza o segundo valor: ");
            b = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Introduza o terceiro valor: ");
            c = Convert.ToInt32(Console.ReadLine());
            if (a < b + c && b < a + c && c < a + b) // É um triângulo?
            {
                if (a == b && a == c) // Qual é o tipo de triângulo?
                    Console.WriteLine("É um triângulo equilátero");
                else if (a == b || a == c || b == c)
                    Console.WriteLine("É um triângulo isósceles");
                else
                    Console.WriteLine("É um triângulo escaleno");
            }
            else
                Console.WriteLine("Não é um triângulo");
        }
    }
}

//Versao 2
using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b, c;
            Console.WriteLine("Introduza o primeiro valor: "); // Não é feita validação, ou seja, o utilizador pode
            a = Convert.ToInt32(Console.ReadLine()); // introduzir caracteres inválidos e gerar exceções

            Console.WriteLine("Introduza o segundo valor: ");
            b = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Introduza o terceiro valor: ");
            c = Convert.ToInt32(Console.ReadLine());
            if (a < b + c && b < a + c && c < a + b) // É um triângulo?
            {
                if (a == b) // Qual é o tipo de triângulo?
                {
                    if (a == c)
                        Console.WriteLine("É um triângulo equilátero");
                    else
                        Console.WriteLine("É um triângulo isósceles");
                }
            }
            else
                Console.WriteLine("Não é um triângulo");
        }
    }
}
```

```

        Console.WriteLine("É um triângulo isósceles");
    else
        if (a == c || b == c)
            Console.WriteLine("É um triângulo isósceles");
        else
            Console.WriteLine("É um triângulo escaleno");
    else
        Console.WriteLine("Não é um triângulo");
    }
}
}

```

12. Escrever um programa que escreva, por extenso, os números de 0 até 9.

```

using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int a;
            Console.Write("Introduza um número de 0 a 9: "); // Não é feita validação, ou seja, o
            // utilizador pode
            a = Convert.ToInt32(Console.ReadLine());           // introduzir caracteres inválidos e
            // gerar exceções (ERROS)
            switch (a)
            {
                case 0: Console.WriteLine("Zero"); break;
                case 1: Console.WriteLine("Um"); break;
                case 2: Console.WriteLine("Dois"); break;
                case 3: Console.WriteLine("Três"); break;
                case 4: Console.WriteLine("Quatro"); break;
                case 5: Console.WriteLine("Cinco"); break;
                case 6: Console.WriteLine("Seis"); break;
                case 7: Console.WriteLine("Sete"); break;
                case 8: Console.WriteLine("Oito"); break;
                case 9: Console.WriteLine("Nove"); break;
                default: Console.WriteLine("Número inválido"); break;
            }
        }
    }
}

```

13. Escrever um programa para identificar se o carácter introduzido pelo utilizador é uma vogal ou uma consoante.

```

using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {

```



```
case 5:
    if (dia < 21)
        Console.WriteLine("Touro");
    else
        Console.WriteLine("Gêmeos");
    break;
case 6:
    if (dia < 21)
        Console.WriteLine("Gêmeos");
    else
        Console.WriteLine("Caranguejo");
    break;
case 7:
    if (dia < 22)
        Console.WriteLine("Caranguejo");
    else
        Console.WriteLine("Leão");
    break;
case 8:
    if (dia < 23)
        Console.WriteLine("Leão");
    else
        Console.WriteLine("Virgem");
    break;
case 9:
    if (dia < 23)
        Console.WriteLine("Virgem");
    else
        Console.WriteLine("Balança");
    break;
case 10:
    if (dia < 23)
        Console.WriteLine("Balança");
    else
        Console.WriteLine("Escorpião");
    break;
case 11:
    if (dia < 22)
        Console.WriteLine("Escorpião");
    else
        Console.WriteLine("Sagitário");
    break;
case 12:
    if (dia < 22)
        Console.WriteLine("Sagitário");
    else
        Console.WriteLine("Capricórnio");
    break;
default:
    Console.WriteLine("Data inválida");
    break;
```

```
    }
}
}
```



```
//Versão 2
using System;

namespace M5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int dia, mes;
            Console.Write("Introduza o dia de nascimento: "); // Não é feita validação, ou seja,
// o utilizador pode
            dia = Convert.ToInt32(Console.ReadLine()); // introduzir caracteres inválidos e
// gerar exceções (ERROS)
            Console.Write("Introduza o mês de nascimento: ");
            mes = Convert.ToInt32(Console.ReadLine());
            if ((mes == 1 && dia >= 20 && dia <= 31) || (mes == 2 && dia >= 1 && dia <= 18))
                Console.WriteLine("Aquário");
            else
                if ((mes == 2 && dia >= 19 && dia <= 29) || (mes == 3 && dia >= 1 && dia <= 20))
                    Console.WriteLine("Peixes");
            else
                if ((mes == 3 && dia >= 21 && dia <= 31) || (mes == 4 && dia >= 1 && dia <=
19))
                    Console.WriteLine("Carneiro");
            else
                if ((mes == 4 && dia >= 21 && dia <= 30) || (mes == 5 && dia >= 1 && dia
<= 20))
                    Console.WriteLine("Touro");
            else
                if ((mes == 5 && dia >= 21 && dia <= 31) || (mes == 6 && dia >= 1 &&
dia <= 20))
                    Console.WriteLine("Gêmeos");
            else
                if ((mes == 6 && dia >= 21 && dia <= 30) || (mes == 7 && dia >= 1
&& dia <= 21))
                    Console.WriteLine("Caranguejo");
            else
                if ((mes == 7 && dia >= 22 && dia <= 31) || (mes == 8 && dia
>= 1 && dia <= 22))
                    Console.WriteLine("Leão");
            else
                if ((mes == 8 && dia >= 23 && dia <= 31) || (mes == 9 &&
dia >= 1 && dia <= 22))
                    Console.WriteLine("Virgem");
            else
                if ((mes == 9 && dia >= 23 && dia <= 30) || (mes == 10
&& dia >= 1 && dia <= 22))
                    Console.WriteLine("Balança");
            else
                if ((mes == 10 && dia >= 23 && dia <= 31) || (mes
== 11 && dia >= 1 && dia <= 21))
                    Console.WriteLine("Escorpião");
            else
                if ((mes == 11 && dia >= 22 && dia <= 30) ||
(mês == 12 && dia >= 1 && dia <= 21))
                    Console.WriteLine("Sagitário");
            else
                if ((mes == 12 && dia >= 22 && dia <= 31)
|| (mês == 1 && dia >= 1 && dia <= 20))
```

```
        Console.WriteLine("Capricórnio");  
    else  
        Console.WriteLine("Data inválida");  
    }  
}  
}
```

MÓDULO 6 – Instruções de repetição

1. Escrever um programa que imprima no ecrã um retângulo com o símbolo "0" sabendo a sua largura e comprimento.

```
using System;

namespace MODULO5_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, j, largura, comprimento;
            Console.WriteLine("Indicar a largura:");
            largura = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Indicar o comprimento:");
            comprimento = Convert.ToInt32(Console.ReadLine());
            for (i = 0; i < largura; i++)
            {
                for (j = 0; j < comprimento; j++)
                {
                    Console.Write("0");
                }
                Console.WriteLine();
            }
        }
    }
}
```

2. Escreva um programa que imprima no ecrã os números ímpares entre 1 e 50.

VERSÃO 1

```
using System;

namespace M6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            for(i = 1; i <=50; i++)
            {
                if (i % 2 != 0)
                {
                    Console.WriteLine(i);
                }
            }
            Console.WriteLine();
        }
    }
}
```

VERSÃO 2

```
using System;

namespace M6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            for (i = 1; i <= 50; i += 2)
            {
                Console.WriteLine(i);
            }
        }
    }
}
```

- Escreva um programa que calcule a soma, com incrementos de 3, de todos os números menores que 100, começando em 4 (ex.: 4+7+10+13+...), utilizando as três estruturas de repetição que conhece.

DESAFIO: Em que situação um ciclo WHILE e DO...WHILE não são equivalentes?

Por exemplo: se inicializar a variável (i), por qualquer motivo a 400...qual é a diferença?

No caso do WHILE o resultado é diferente do DO...WHILE;

No 1º o ciclo não executa, para logo na condição; No 2º o ciclo executa, pelo menos uma vez!

```

C:\WINDOWS\system32\c...
Soma:1616
Soma:0
Soma:400
Press any key to continue . . .

```

```

using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, soma;
            soma = 0;
            for (i = 4; i < 100; i += 3)
                soma += i;
            Console.WriteLine("Soma:{0}", soma);
            soma = 0; //Após a execução do ciclo for é necessário repor os valores das variáveis
soma e i
            i = 4;
            while (i < 100)
            {
                soma += i;
                i += 3;
            }
            Console.WriteLine("Soma:{0}", soma);
            soma = 0; //Após a execução do ciclo while é necessário repor os valores das variáveis
soma e i
            i = 4;
            do
            {
                soma += i;
                i += 3;
            } while (i < 100);
            Console.WriteLine("Soma:{0}", soma);
        }
    }
}

```

- Escreva um programa que receba dois números inteiros e gere os números inteiros que estão no intervalo compreendido por eles.

```

using System;

namespace MODULO6_Exercicios

```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            int inf, sup, i;
            Console.Write("Introduza o limite inferior do intervalo: ");
            inf = Convert.ToInt32(Console.ReadLine());
            Console.Write("Introduza o limite superior do intervalo: ");
            sup = Convert.ToInt32(Console.ReadLine());
            for (i = inf; i <= sup; i++) // ciclo com i a variar entre inf e sup
            {
                Console.Write("{0} ", i);
            }
            Console.WriteLine();
        }
    }
}
```

5. Altere o programa anterior para mostrar no final a soma dos números.

```
using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int inf, sup, i, soma = 0;
            Console.Write("Introduza o limite inferior do intervalo: ");
            inf = Convert.ToInt32(Console.ReadLine());
            Console.Write("Introduza o limite superior do intervalo: ");
            sup = Convert.ToInt32(Console.ReadLine());
            for (i = inf; i <= sup; i++) // Ciclo com i a variar entre inf e sup
            {
                Console.Write("{0}", i);
                if (i != sup) // Se não for o último número acrescenta o operador +
                    Console.Write("+");
                soma += i; //Vai adicionando cada um dos números à variável soma
            }
            Console.WriteLine("={0}\n", soma);
        }
    }
}
```

6. Escreva um programa que leia uma sequência de números inteiros a partir do teclado e apresente o máximo e o mínimo. O programa termina quando o número lido for zero.

```
C:\WINDOWS\system32\cmd.exe
Inserir nº inteiro:
23
Inserir nº inteiro (para parar inserir 0):
56
Inserir nº inteiro (para parar inserir 0):
7
Inserir nº inteiro (para parar inserir 0):
9
Inserir nº inteiro (para parar inserir 0):
8
Inserir nº inteiro (para parar inserir 0):
0
O valor máximo é 56 e o mínimo é 7
Press any key to continue . . .
```

```
using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int num, min, max;
            Console.WriteLine("Inserir nº inteiro: ");
            num = Convert.ToInt32(Console.ReadLine());
            min = num;
            max = num;
            while (num != 0)
            {
                if (num > max)
                {
                    max = num;
                }
                if (num < min)
                {
                    min = num;
                }

                Console.WriteLine("Inserir nº inteiro (para parar inserir 0): ");
                num = Convert.ToInt32(Console.ReadLine());
            }
            if (max == 0 && min == 0) // Se max e min têm o valor zero quer dizer que o primeiro nº
digitado foi zero, ou seja, a condição de saída
                Console.WriteLine("Não foi introduzido nenhum número");
            else
                Console.WriteLine("O valor máximo é {0} e o mínimo é {1}", max, min);
        }
    }
}
```

- Escreva um programa que leia uma sequência de números inteiros a partir do teclado e acumule unicamente a soma dos inteiros positivos. O programa termina quando o número lido for zero.

CICLO DO..WHILE

```
using System;

namespace MODULO6_Exercicios
{
    class Program
```

```
{
    static void Main(string[] args)
    {
        //OPÇÃO MAIS ADEQUADA!

        int n, soma = 0;
        do
        {
            Console.WriteLine("Introduza um nº inteiro (zero para sair): ");
            n = Convert.ToInt32(Console.ReadLine());
            if (n > 0)
                soma += n;
        } while (n != 0);
        Console.WriteLine("Soma dos inteiros positivos: {0}", soma);

        //OPÇÃO possível!
        int num=-1, soma = 0;
        while (num != 0)
        {
            Console.WriteLine("Inserir nº: ");
            num = Convert.ToInt32(Console.ReadLine());
            if (num > 0)
            {
                soma = soma + num;
                num++;
            }
        }
        Console.WriteLine("Soma dos inteiros positivos: {0}", soma);
    }
}
```

8. Escrever um programa que peça a altura de n funcionários de uma empresa e apresente as seguintes estatísticas:

- A altura do funcionário mais baixo;
- A altura do funcionário mais alto;
- A altura média.

```
using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int altura, alto, baixo, totalF, i, soma;
            Console.WriteLine("Quantos funcionários tem a empresa? ");
            totalF = Convert.ToInt32(Console.ReadLine());
            if (totalF > 0)
            {
                Console.WriteLine("Altura do funcionário em cm: ");
                altura = Convert.ToInt32(Console.ReadLine());
                alto = altura;
                baixo = altura;
            }
        }
    }
}
```

```

        soma = altura;
        for (i = 1; i < totalF; i++)
        {
            Console.WriteLine("Altura do funcionário em cm: ");
            altura = Convert.ToInt32(Console.ReadLine());
            if (altura > alto)
            {
                alto = altura;
            }
            if (altura < baixo)
            {
                baixo = altura;
            }
            soma = soma + altura;
        }
        Console.WriteLine("O funcionário mais alto tem {0} cm e o mais baixo tem {1} cm.",
alto, baixo);
        Console.WriteLine("A altura média dos funcionários é de {0} cm.", soma / totalF);
    }
}
}
}

```

9. Escreva um programa que leia um número inteiro e calcule a soma dos seus dígitos.

```

using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int n, soma = 0;
            Console.Write("Introduza um número inteiro: ");
            n = Convert.ToInt32(Console.ReadLine());
            while (n != 0) // Sugestão: Faça execução passo a passo (F10) e observe os valores das
variáveis na janela Locais
            {
                soma += n % 10; // O resto da divisão por 10 permite obter cada um dos dígitos do
número
                n /= 10; // Depois da obtenção esse dígito é descartado
            }
            Console.WriteLine("Soma dos dígitos:{0}", soma);
        }
    }
}

```

10. Escrever um programa que permita descobrir aleatoriamente um número gerado pelo computador.

Dica: Utilizar a classe *Random*
Random rnd = new Random();
int numeroSecreto = rnd.Next(<valorInicial>, <valorFinal>);

```

using System;

```



```
namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int n;
            Random rnd = new Random();
            int numeroSecreto = rnd.Next(1, 6);

            do
            {
                Console.WriteLine("Introduza um nº: ");
                n = Convert.ToInt32(Console.ReadLine());
            }
            while (numeroSecreto != n);
            Console.WriteLine("Acertou!");
        }
    }
}
```

11. Alterar o exercício anterior para contabilizar o nº de tentativas falhadas.

```
using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int n, numeroTentativas = 0;
            Random rnd = new Random();
            int numeroSecreto = rnd.Next(1, 3);

            do
            {
                Console.WriteLine("Introduza um nº: ");
                n = Convert.ToInt32(Console.ReadLine());
                numeroTentativas++;
            }
            while (numeroSecreto != n);
            Console.WriteLine("Acertou após {0} tentativas.", numeroTentativas);
        }
    }
}
```

12. Escrever um programa que peça a n pessoas de uma empresa a sua idade. No final o programa deverá verificar se a média de idades varia entre 0 e 25, 26 e 60 e maior que 60; e dizer se os funcionários são jovens, adultos ou idosos, conforme a média calculada.

```
using System;

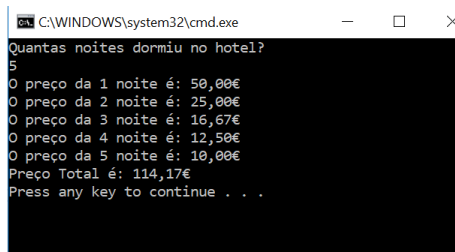
namespace MODULO6_Exercicios
{
    class Program
    {
```

```
static void Main(string[] args)
{
    int totalFuncionarios, idade, somaIdades=0;
    float media;
    Console.WriteLine("Quantas pessoas tem a empresa?");
    totalFuncionarios = Convert.ToInt32(Console.ReadLine());
    for (int i = 1; i <= totalFuncionarios; i++)
    {
        Console.WriteLine("Qual é a idade do {0} funcionário?", i);
        idade = Convert.ToInt32(Console.ReadLine());
        somaIdades += idade;
    }
    media = somaIdades / totalFuncionarios;
    if (media > 60)
    {
        Console.WriteLine("Funcionários idosos.");
    }
    else
    {
        if (media > 25)
        {
            Console.WriteLine("Funcionários adultos.");
        }
        else
        {
            Console.WriteLine("Funcionários jovens.");
        }
    }
}
}
```

13. O dono de um hotel concebeu uma forma original de cobrar aos seus clientes. A primeira noite custa 50 euros. A segunda custa 25 euros (ou seja, 50\2 euros), a terceira 50\3 euros e a n-ésima noite custa 50\ n euros. Escreva um programa que calcule a cobrança a efetuar a um cliente que fique X noites no hotel. O programa deve indicar o preço a pagar por cada noite e também o total.

Dica: Para conseguir visualizar o carácter € pode introduzir, no *Main*, a seguinte instrução:

```
Console.OutputEncoding = System.Text.Encoding.UTF8;
```



```
C:\WINDOWS\system32\cmd.exe
Quantas noites dormiu no hotel?
5
O preço da 1 noite é: 50,00€
O preço da 2 noite é: 25,00€
O preço da 3 noite é: 16,67€
O preço da 4 noite é: 12,50€
O preço da 5 noite é: 10,00€
Preço Total é: 114,17€
Press any key to continue . . .
```

```
using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
```

```

        Console.OutputEncoding = System.Text.Encoding.UTF8;
        int noitesDormidas, i;
        float precoNoite = 0, precoTotal = 0;
        Console.WriteLine("Quantas noites dormiu no hotel?");
        noitesDormidas = Convert.ToInt32(Console.ReadLine());
        for (i = 1; i <= noitesDormidas; i++)
        {
            precoNoite = 50F / (i);
            precoTotal += precoNoite;
            Console.WriteLine("O preço da {0} noite é: {1:0.00}€", i, precoNoite);
        }
        Console.WriteLine("Preço Total é: {0:0.00}€", precoTotal);
    }
}

```

14. Supondo que a população do país A é de 80 000 habitantes com uma taxa anual de crescimento de 3% e que a população do país B é 200 000 habitantes com uma taxa de crescimento de 1.5%. Escreva um programa que calcule e escreva o número de anos necessários para que a população do país A ultrapasse ou iguale a população do país B, mantidas as taxas de crescimento.

// Ciclo WHILE ou DO..WHILE; compliquei um pouco nos tipos de dados, na 1ª resolução.

```

using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            long populacaoA = 80000, populacaoB = 200000;
            int nAnos = 0;
            double txA = 0.03, txB = 0.015;
            do
            {
                nAnos++; //cada ano
                populacaoA = (long)(populacaoA * (1 + txA)); //forçar um float a ser um
                long...typecast.
                populacaoB = (long)(populacaoB * (1 + txB)); //forçar um float a ser um
                long...typecast.
                Console.WriteLine("Ano {0}: A: {1} B: {2}", nAnos, populacaoA, populacaoB);
            } while (populacaoA < populacaoB);
            Console.WriteLine(nAnos);
        }
    }
}

```

OU

```

using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)

```

```

    {
        {
            double A = 80000, tcA = 0.03, B = 200000, tcB = 0.015, a = 0;
            while (A < B)
            {
                A *= (1 + tcA);
                B *= (1 + tcB);
                a++;
            }
            Console.WriteLine("A população do país A igualará ou ultrapassará a população do país
B em {0} anos", a);
        }
    }
}

```

15. Escreva um programa que calcule o total das entradas a pagar num parque de diversões, sabendo que os bilhetes para crianças até aos 4 anos são grátis, dos 6 aos 12 são 6 euros, dos 12 aos 17 são 12 euros e para os adultos são 18 euros. O programa deverá pedir ao utilizador o número e tipo de entradas.

```

using System;

namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.OutputEncoding = System.Text.Encoding.UTF8;
            int nPessoasA, nPessoasB, nPessoasC, nPessoasD, soma = 0;

            Console.WriteLine("Inserir o nº total de pessoas com idade < 4: ");
            nPessoasA = Convert.ToInt32(Console.ReadLine());
            soma += 0 * nPessoasA;

            Console.WriteLine("Inserir o nº total de pessoas com idade >= 18: ");
            nPessoasB = Convert.ToInt32(Console.ReadLine());
            soma += 18 * nPessoasB;

            Console.WriteLine("Inserir o nº total de pessoas com idade entre 12 e 17 anos: ");
            nPessoasC = Convert.ToInt32(Console.ReadLine());
            soma += 12 * nPessoasC;

            Console.WriteLine("Inserir o nº total de pessoas com idade entre 6 e 12 anos: ");
            nPessoasD = Convert.ToInt32(Console.ReadLine());
            soma += 6 * nPessoasD;

            Console.WriteLine("O total a pagar é {0}€ ", soma);
        }
    }
}

```

16. Escreva um programa que leia 20 números entre 10 e 30 (os números lidos devem ser validados) e apresente o produto dos que pertencerem ao intervalo [10,20].

```

using System;

```

```
namespace MODULO6_Exercicios
{
    class Program
    {
        static void Main(string[] args)
        {
            int n, i, prod = 1;
            for (i = 0; i < 20; i++)
            {
                do
                {
                    Console.Write("Introduza um nº inteiro entre 10 e 30: ");
                    n = Convert.ToInt32(Console.ReadLine());
                } while (n < 10 || n > 30);
                if (n >= 10 && n <= 20)
                    prod *= n;
            }
            Console.WriteLine("Produto: {0}", prod);
        }
    }
}
```

MÓDULO 7 – Métodos

1. Escreva e teste um método que escreva “Olá, <nome>”.

```
using System;

namespace Modulo7
{
    class Program
    {
        static void Sauda(string nome)
        {
            Console.WriteLine("Olá {0} ", nome);
        }
        static void Main(string[] args)
        {
            Console.Write("Qual o nome? ");
            string nome = Console.ReadLine();
            Sauda(nome);
        }
    }
}
```

2. Escreva e teste um método que devolva a soma de dois números inteiros.

```
using System;

namespace Modulo7
{
    class Program
    {
        static int Soma(int a, int b)
        {
            return a + b;
        }

        static void Main(string[] args)
        {
            int x, y;
            Console.Write("Introduza o 1º valor: ");
            x = Convert.ToInt32(Console.ReadLine());
            Console.Write("Introduza o 2º valor: ");
            y = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Soma: {0}", Soma(x, y));
        }
    }
}
```

3. Escrever e testar um método que imprima no ecrã uma linha com n asteriscos, em que n é um número introduzido pelo utilizador.

```
using System;

namespace Modulo7
```

```
{
    class Program
    {
        static void linhaAster(int num)
        {
            for (int i = 0; i < num; i++)
            {
                Console.Write("*");
            }
            Console.WriteLine();
        }

        static void Main(string[] args)
        {
            int num;
            Console.Write("Quantos asteriscos? ");
            num = Convert.ToInt32(Console.ReadLine());
            linhaAster(num);
        }
    }
}
```

4. Escrever um programa que simule uma máquina de calcular, de números inteiros, em que cada uma das operações soma, subtração, multiplicação e divisão é implementada através de um método.

```
using System;

namespace Modulo7
{
    class Program
    {
        static int soma(int a, int b)
        {
            return a + b;
        }

        static int sub(int a, int b)
        {
            return a - b;
        }

        static int mult(int a, int b)
        {
            return a * b;
        }

        static double div(int a, int b)
        {
            return (double)a / b;
        }

        static void Main(string[] args)
        {
            int op1, op2;
            string op;
            Console.Write("Introduza o 1º operando: ");
            op1 = Convert.ToInt32(Console.ReadLine());
            Console.Write("Introduza o 2º operando: ");
            op2 = Convert.ToInt32(Console.ReadLine());
        }
    }
}
```

```

    Console.Write("Introduza o operador: ");
    op = Console.ReadLine();
    switch (op)
    {
        case "+":
            Console.WriteLine("{0}+{1}={2}", op1, op2, soma(op1, op2));
            break;
        case "-":
            Console.WriteLine("{0}-{1}={2}", op1, op2, sub(op1, op2));
            break;
        case "*":
            Console.WriteLine("{0}*{1}={2}", op1, op2, mult(op1, op2));
            break;
        case "/":
            Console.WriteLine("{0}/{1}={2}", op1, op2, div(op1, op2));
            break;
        default:
            Console.WriteLine("Operador inválido");
            break;
    }
  }
}

```

5. Escrever e testar um método que devolva o máximo de dois valores.

```

using System;

namespace Modulo7
{
    class Program
    {
        static double Maior(double a, double b)
        {
            if (a > b)
            {
                return a;
            }
            else
            {
                return b;
            }
        }

        static void Main(string[] args)
        {
            double x, y;
            Console.Write("Introduza o 1º valor: ");
            x = Convert.ToDouble(Console.ReadLine());
            Console.Write("Introduza o 2º valor: ");
            y = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("O maior valor é: {0}", Maior(x, y));
        }
    }
}

```


6. Escrever e testar um programa que devolva o valor da hipotenusa.

Usar o método *Math.Sqrt()*

[https://msdn.microsoft.com/pt-br/library/system.math.sqrt\(v=vs.100\).aspx](https://msdn.microsoft.com/pt-br/library/system.math.sqrt(v=vs.100).aspx)

```
using System;

namespace Modulo7
{
    class Program
    {
        static double hipotenusa(double a, double b)
        {
            return Math.Sqrt(a * a + b * b);
        }

        static void Main(string[] args)
        {
            double c1, c2;
            Console.Write("Introduza o valor do 1º cateto: ");
            c1 = Convert.ToDouble(Console.ReadLine());
            Console.Write("Introduza o valor do 2º cateto: ");
            c2 = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("Hipotenusa {0}", hipotenusa(c1, c2));
        }
    }
}
```

7. Escrever e testar um método que receba, como argumento, um valor inteiro positivo e devolva o nº de dígitos do valor recebido.

```
using System;

namespace Modulo7
{
    class Program
    {
        static uint numDigitos(uint n)
        {
            uint contador = 0;
            do
            {
                n = n / 10;
                contador++;
            } while (n != 0);
            return contador;
        }

        static void Main(string[] args)
        {
            uint N;
            Console.Write("Introduza um nº inteiro positivo: ");
            N = Convert.ToUInt32(Console.ReadLine());
            Console.WriteLine("O número {0} tem {1} dígitos", N, numDigitos(N));
        }
    }
}
```

```
}  
}
```

8. Analise o programa e descreva o seu comportamento.

Método que simula o lançamento de um dado e utiliza este método num programa que simula n lançamentos e verifica quantas vezes saiu o 6.

```
using System;  
  
namespace Modulo7  
{  
    class Program  
    {  
        static int Exemplo()  
        {  
            return rnd.Next(1, 7);  
        }  
        static Random rnd = new Random();  
  
        static void Main(string[] args)  
        {  
            int N, v, contador = 0;  
            Console.Write("Quantas vezes quer repetir: ");  
            N = Convert.ToInt32(Console.ReadLine());  
            for (int i = 0; i < N; i++)  
            {  
                if ((v = Exemplo()) == 6)  
                    contador++;  
                Console.WriteLine(v);  
            }  
            Console.WriteLine("O número 6 ocorreu {0} vezes", contador);  
        }  
    }  
}
```

9. Analise o programa e descreva o seu comportamento.

Escrever um método para calcular por recursividade potências de 2.

```
using System;  
  
namespace Modulo7  
{  
    class Program  
    {  
        static int pot2(int e)  
        {  
            if (e == 0)  
                return 1;  
            else  
                return 2 * pot2(e - 1);  
        }  
  
        static void Main(string[] args)  
        {  
            for (int i = 0; i < 10; i++)
```

```
        Console.WriteLine("2^{0}={1}", i, pot2(i));  
    }  
}
```

MÓDULO 8 – Vetores e matrizes

1. Escrever um programa que permita criar um vetor com comprimento 10. Depois deve de alterar o valor dos índices 3 e 9. No final deve de mostrar a evolução desta troca.

```
using System;
namespace Modulo8
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] a = new int[] { -1, 0, 1, 2, 3, 4, 5, 6, 8, 9};
            Console.WriteLine("O valor inicial do índice 3 é: {0} \n O valor inicial do índice 9 é o {0}", a[3], a[9]);
            Console.WriteLine("Inserir novo nº para índice 3:");
            a[3] = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Inserir novo nº para índice 9:");
            a[9] = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Agora o novo valor do índice 3 é: {0}\n Agora o novo valor do índice 9 é: {1}", a[3], a[9]);
        }
    }
}
```

2. Escrever um programa para calcular a multiplicação, soma e média de todos os elementos de um vetor com comprimento de 7.

```
using System;
namespace VetorExemplo5
{
    class Program
    {
        static void Main(string[] args)
        {
            //Subtrair e somar todos os elementos do vetor.
            int[] a = new int[] { -2, 9, 15, 34, -5, 11, 6, 7, 0 };
            int sub = 0, soma = 0;
            float media = 0;
            for (int i = 0; i < a.Length; i++)
            {
                sub -= a[i];
                soma += a[i];
            }
            media = soma / a.Length;
            Console.WriteLine("A subtração do vetor é: {0}", sub);
            Console.WriteLine("A soma do vetor é: {0}", soma);
            Console.WriteLine("A média do vetor é: {0:0.0}", media);
        }
    }
}
```

3. Escrever um programa para encontrar o índice e o valor de maior valor.

```
using System;

namespace VetorExemplo6
{
    class Program
    {
        static void Main(string[] args)
        {
            //Encontrar índice e valor de maior valor.
            int[] a = new int[] { -2, 9, 15, 34, -5, 11, 6, 7, 0 };
            int maior = 0, posMaior = 0;
            for (int i = 0; i < a.Length; i++)
            {
                if (a[i] > maior)
                {
                    maior = a[i]; //Guarda o maior valor do vetor.
                    posMaior = i; //Guarda o índice.
                }
            }
            Console.WriteLine("O valor é {0}.", maior);
            Console.WriteLine("O índice é {0}.", posMaior);
        }
    }
}
```

4. Escrever um programa que permita criar um vetor com comprimento 8 e que recorrendo ao ciclo *foreach* passe o valor de cada índice para uma variável inteira.

```
using System;

namespace VetorExemplo6
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] a = new int[8] {100, 200, 300, 400, 500, 600, 700, 800 };
            a[0] = 100;
            a[1] = 200;
            a[2] = 300;
            a[3] = 400;
            a[4] = 500;
            a[5] = 600;
            a[6] = 700;
            a[7] = 800;

            foreach (int n in a)
            {
                Console.WriteLine(n);
            }
        }
    }
}
```

5. Escrever um programa que leia N números reais para um vetor com o máximo de 100 números e apresente no ecrã a soma dos números.

```
using System;

namespace Cap8
{
    class Program
    {
        static void Main(string[] args)
        {
            float[] v = new float[100];
            float soma = 0;
            Console.Write("Quantos números vai introduzir? ");
            int N = Convert.ToInt32(Console.ReadLine());
            for (int i = 0; i < N; i++)
            {
                Console.Write("Introduza um número real: ");
                v[i] = Convert.ToSingle(Console.ReadLine());
            }
            //Tenho todos os números dentro do array e agora...
            for (int i = 0; i < N; i++)
            {
                soma += v[i];
                //Os valores foram guardados no array e
                //agora posso utilizá-los para efetuar uma soma.
            }
            Console.WriteLine("Soma: {0} ", soma);
        }
    }
}
```

6. Escrever e testar um método que pesquise num vetor de inteiros V um determinado número inteiro N e devolva o número de vezes que encontrou esse número. Se não encontrar deverá devolver -1.

```
using System;

namespace Modulo8_Exercicios
{
    class Program
    {
        static int Pesquisa(int[] vetor, int numProcurar)
        {
            int cont = 0;
            foreach (int elemento in vetor)
            // os elementos da variável vetor vão ser colocados, 1 a 1, na variável "elemento".
            {
                if (elemento == numProcurar)
                {
                    cont++;
                }
            }
            if (cont == 0)
            {
                return -1;
            }
            else
            {
                return cont;
            }
        }
    }
}
```

```

    }
}

static void Main(string[] args)
{
    int[] arr = { 2, 6, 8, 9, 3, 4, 6, 4, 4, 6 };
    Console.Write("Qual número pretende pesquisar? ");
    int num = Convert.ToInt32(Console.ReadLine());
    int contagem = Pesquisa(arr, num);
    if (contagem == -1)
        Console.WriteLine("O número {0} não foi encontrado", num);
    else
        Console.WriteLine("O número {0} foi encontrado {1} vezes", num, contagem);
}
}
}

```

7. Escrever e testar um método para verificar se dois vetores de inteiros são iguais. Dois vetores são iguais se na mesma posição tiverem elementos com o mesmo valor.

```

using System;

namespace Modulo8_Exercicios
{
    class Program
    {
        // percorre array de uma ponta a outra e mostra cada elemento, um a um
        static void ApresentaVetor(int[] v)
        {
            foreach (int n in v)
            {
                Console.Write(" {0}", n);
            }
            Console.WriteLine();
        }

        //Compara 2 vetores e devolve true se forem iguais.
        static bool Iguais(int[] a, int[] b)
        {
            if (a.Length != b.Length)
            {
                return false;
            }
            else
            {
                for (int i = 0; i < a.Length; i++)
                {
                    if (a[i] != b[i])
                    {
                        return false;
                    }
                }
                return true;
            }
        }
        static void Main(string[] args)
        {
            int[] O = { 2, 6, 8, 7, 3, 9, 1, 5 }, R = { 2, 6, 8, 7, 3, 9, 1, 5 };// Experimentar
            alterar os valores
        }
    }
}

```

```
        ApresentaVetor(0);
        ApresentaVetor(R);
        if (Iguais(0, R))
        {
            Console.WriteLine("Os vetores são iguais");
        }
        else
        {
            Console.WriteLine("Os vetores não são iguais");
        }
    }
}
```

8. Escreva um programa que leia e apresente um conjunto de número inteiros para uma matriz bidimensional 4 x 3

```
using System;

namespace Modulo8_Exercicios
{
    class Program
    {
        static void ApresentaMatriz(int[,] v)
        {
            for (int i = 0; i < v.GetLength(0); i++) // Linha
            {
                for (int j = 0; j < v.GetLength(1); j++) // Coluna
                {
                    Console.Write(" {0}", v[i, j]);
                }
                Console.WriteLine();
            }
        }

        static void Main(string[] args)
        {
            int[,] V = new int[4, 3];
            for (int i = 0; i < V.GetLength(0); i++)
                for (int j = 0; j < V.GetLength(1); j++)
                {
                    Console.Write("Introduza um número inteiro: ");
                    V[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            ApresentaMatriz(V);
        }
    }
}
```


MÓDULO 9 – *Strings*

1. Escrever e testar um método que leia o seu nome e o apresente no ecrã.

```
using System;

namespace Modulo9
{
    class Program
    {
        static void Main(string[] args)
        {
            string nome;
            Console.Write("Introduza o seu nome: ");
            nome = Console.ReadLine();
            Console.WriteLine("O seu nome é {0} ", nome);
        }
    }
}
```

2. Escrever e testar um programa que peça um nome completo e mostre só o primeiro e o último nome.

```
using System;

namespace Modulo9
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Introduza o seu nome completo: ");
            string nome = Console.ReadLine();
            string[] nomesSeparados = nome.Split(' '); //Separa a string em palavras (por
            espaço).
            //Assim sei que a 1ª palavra está no 1º índice e a última está no último índice.
            Console.WriteLine("1º nome é: {0}", nomesSeparados[0]);
            Console.WriteLine("último nome é:{0}", nomesSeparados[nomesSeparados.Length-1]);
        }
    }
}
```

3. Escrever e testar um programa que peça o nome e apelidos, um de cada vez, e que junte tudo numa única *string*.

```
using System;

namespace Modulo9
{
    class Program
    {
        static void Main(string[] args)
        {
            int numNomes;
            Console.WriteLine("Quantos nomes tem o seu nome: ");
```

```

    numNomes = Convert.ToInt32(Console.ReadLine());
    string nomeCompleto = " ";
    for (int i = 1; i <= numNomes; i++)
    {
        Console.WriteLine("Escreva o {0} nome: ", i);
        string nome = Console.ReadLine();
        nomeCompleto += " " + nome;
    }
    Console.WriteLine("O seu nome completo é: {0}", nomeCompleto);
}
}

```

4. Escrever e testar um programa que peça um nome completo e converta para maiúsculas o 1º, 3º, 5º nomes.

```

using System;

namespace Modulo9
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Introduza o seu nome completo: ");
            string nome = Console.ReadLine();
            string[] nomesSeparados = nome.Split(' '); //Separa a string em palavras (por
            espaço).

            for(int i = 0; i < nomesSeparados.Length; i++)
            {
                if (i % 2 == 0) // se i é par...
                {
                    nomesSeparados[i] = nomesSeparados[i].ToUpper();
                }
            }

            Console.WriteLine("O seu nome completo alterado é: ");
            foreach(string nomeIndividual in nomesSeparados)
            {
                Console.Write(" {0}", nomeIndividual);
            }

            Console.WriteLine();
        }
    }
}

```

5. Escrever e testar um método que devolva o 1º índice, numa *string*, de um carácter introduzido pelo utilizador.

```

using System;

namespace Modulo9
{
    class Program
    {
        static int Pesquisa(string s, char c)

```

```

    {
        for (int i = 0; i < s.Length; i++)
            if (s[i] == c)
                return i; //sai do método devolvendo o i
        return -1; //Se o carácter não existir devolve -1. Qualquer valor abaixo de zero
        para n correr o risco de coincidir com o índice do carácter existente.
    }
    static void Main(string[] args)
    {
        string s;
        char c;
        Console.Write("Introduza uma frase: ");
        s = Console.ReadLine();
        Console.Write("Introduza o carácter a pesquisar: ");
        c = Convert.ToChar(Console.ReadLine());
        if (Pesquisa(s, c) == -1)
            Console.WriteLine("O carácter {0} não existe na string", c);
        else
            Console.WriteLine("O carácter {0} aparece no índice {1}", c, Pesquisa(s, c));
    }
}

```

6. Escrever e testar um método que devolva quantas vezes um determinado carácter aparece numa *string*.

```

using System;

namespace Modulo9
{
    class Program
    {
        static int Pesquisa(string s, char c)
        {
            int contador = 0;
            for (int i = 0; i < s.Length; i++)
                if (s[i] == c)
                    contador++;
            return contador;
        }
        static void Main(string[] args)
        {
            string s;
            char c;
            Console.Write("Introduza uma frase: ");
            s = Console.ReadLine();
            Console.Write("Introduza o carácter a pesquisar: ");
            c = Convert.ToChar(Console.ReadLine());
            Console.WriteLine("O carácter '{0}' aparece {1} vez(es)", c, Pesquisa(s, c));
        }
    }
}

```

7. Escrever e testar um método que concatene duas *strings* introduzidas pelo utilizador.

```

using System;

namespace Modulo9

```

```
{
    class Program
    {
        static string Concatena(string a, string b)
        {
            return a + b;
        }
        static void Main(string[] args)
        {
            string s1, s2;
            Console.Write("Introduza uma frase: ");
            s1 = Console.ReadLine();
            Console.Write("Introduza outra frase: ");
            s2 = Console.ReadLine();
            Console.WriteLine(Concatena(s1, s2));
        }
    }
}
```

8. Analise o programa e descreva o seu comportamento.

```
C:\WINDOWS\system32\cmd.exe
Introduza uma frase: que lindo dia
Introduza o carácter a inserir: *
Introduza o índice da primeira posição: 4
Introduza o índice da última posição: 7
que ****o dia
Press any key to continue . . .
```

```
using System;

namespace Modulo9
{
    class Program
    {
        static string Substitui(string s, char c, int de, int ate)
        {
            char[] v = s.ToCharArray();//Converte a string para um array de caracteres.

            for (int k = de; k <= ate; k++) //No intervalo entre "de" e "até"
            {
                v[k] = c;        //substituo o que lá estiver pelo carácter "c"
            }
            return new string(v); // cria uma string equivalente ao vetor de caracteres "v".
        }
        static void Main(string[] args)
        {
            string s;
            char c;
            int i, j;
            Console.Write("Introduza uma frase: ");
            s = Console.ReadLine();
            Console.Write("Introduza o carácter a inserir: ");
            c = Convert.ToChar(Console.ReadLine());
            Console.Write("Introduza o índice da primeira posição: ");
            i = Convert.ToInt32(Console.ReadLine());
```

```
        Console.Write("Introduza o índice da última posição: ");
        j = Convert.ToInt32(Console.ReadLine());
        s = Substitui(s, c, i, j); // Sem a utilização da passagem por referência as
alterações efetuadas no método substitui não se refletiriam na função main
Transformo a string num array para conseguir alterar o conteúdo. No final, tenho de transformar
um array numa string.

        Console.WriteLine(s);
    }
}
```

Exercício global – Jogo do galo

```
using System;
```

```
namespace Primavera.Games.TicTacToe
```

```
{
```

```
    public class Program
```

```
    {
```

```
        // faz uma jogada no tabuleiro, com o jogador, na linha e coluna indicadas. Devolve true se conseguiu fazer a jogada (se ainda não há jogada lá)
```

```
        public static bool FazerJogada(char[,] tabuleiro, char jogador, int linha, int coluna)
```

```
        {
```

```
            // se a linha e coluna está dentro dos limites do tabuleiro
```

```
            if(linha >= 0 && linha < tabuleiro.GetLength(0) && coluna >= 0 && coluna < tabuleiro.GetLength(1))
```

```
            {
```

```
                // se a posição da tabela tem o carater com o código 0 (o valor por omissão de "caixinhas" de um array de caracteres) quer dizer que ainda não há lá uma jogada
```

```
                if(tabuleiro[linha, coluna] == '\0')
```

```
                {
```

```
                    // faz jogada
```

```
                    tabuleiro[linha, coluna] = jogador;
```

```
                    // "consegui fazer a jogada"
```

```
                    return true;
```

```
                }else
```

```
                {
```

```
                    // "não consegui fazer a jogada"
```

```
                    return false;
```

```
                }
```

```
            }
```

```
            // a execução chega aqui quando a linha ou a coluna está fora dos limites. Por isso também não consegue fazer a jogada.
```

```
            return false;
```

```
        }
```

```
        public static bool ExisteEmpate(char[,] tabuleiro)
```

```
        {
```

```
            // vou percorrer todas as linhas
```

```
            for(int i = 0; i < tabuleiro.GetLength(0); i++)
```

```
            {
```

```
                // e todas as colunas
```

```

    for(int j = 0; j < tabuleiro.GetLength(1); j++)
    {
        // para ver se a célula atual é vazia (se for, então não há um empate porque ainda é
        possível fazer uma próxima jogada)
        if(tabuleiro[i, j] == '\0')
        {
            // posso sair já da função porque já vi que ainda NÃO há empate
            return false;
        }
    }
}

// chega aqui quando não encontrou nenhuma célula vazia. Portanto há empate se não houver
um vencedor
return ObterVencedor(tabuleiro) == '\0';
}

// verifica se neste momento existe alguma jogada vencedora no tabuleiro. Devolve a letra
do vencedor, ou então \0 se não houver um vencedor
public static char ObterVencedor(char[,] tabuleiro)
{
    // vai guardar o jogador vencedor
    char jogadorVencedor = '\0';

    // vai varrer linha a linha à procura de uma jogada vencedora na horizontal
    for(int i = 0; i < tabuleiro.GetLength(0); i++)
    {
        // para já vamos assumir que quem jogou em [i, 0] é o vencedor.
        jogadorVencedor = tabuleiro[i, 0];

        // varre todas as colunas até à penúltima. Não é até à última porque estamos sempre a
        comparar a atual com a próxima e quando chegarmos à penúltima vamos estar a compará-la com a
        última
        for(int j = 0; j < tabuleiro.GetLength(1) - 1; j++)
        {
            // se a célula atual é jogada vazia ou se não for igual à célula na coluna seguinte,
            então não há aqui uma jogada vencedora na horizontal
            if(tabuleiro[i, j] == '\0' || tabuleiro[i, j] != tabuleiro[i, j + 1])
            {
                // para já não há vencedor
                jogadorVencedor = '\0';

                // não vale a pena continuar a procurar nas colunas desta linha, posso sair do
                ciclo de j
                break;
            }
        }
    }
}

```

```

    // se já encontrei um vencedor, nem vale a pena procurar no resto da tabela, salto já
    fora do ciclo de i
    if(jogadorVencedor != '\0')
    {
        break;
    }

}

// se ainda não encontrei um vencedor, então continuo à procura, mas agora na vertical
if(jogadorVencedor == '\0')
{

    // vai varrer coluna a coluna a procura de uma jogada vencedora na vertical
    for(int j = 0; j < tabuleiro.GetLength(1); j++)
    {
        // para já vamos assumir que quem jogou em [0, j] é o vencedor.
        jogadorVencedor = tabuleiro[0, j];

        // varre todas as linhas até à penúltima. Não é até à última porque estamos sempre a
        comparar a atual com a próxima e quando chegarmos à penúltima vamos estar a compará-la com a
        última
        for(int i = 0; i < tabuleiro.GetLength(0) - 1; i++)
        {
            // se a célula atual é jogada vazia ou se não for igual à célula na linha seguinte,
            então não há aqui uma jogada vencedora na vertical
            if(tabuleiro[i, j] == '\0' || tabuleiro[i, j] != tabuleiro[i + 1, j])
            {
                // para já não há vencedor
                jogadorVencedor = '\0';

                // não vale a pena continuar a procurar nas linhas desta coluna, posso sair do
                ciclo de i
                break;
            }
        }

        // se já encontrei um vencedor, nem vale a pena procurar no resto da tabela, salto
        já fora do ciclo de j
        if(jogadorVencedor != '\0')
        {
            break;
        }
    }
}

// se ainda não encontrei um vencedor, então continuo à procura, mas agora na primeira
diagonal

```



```
if(jogadorVencedor == '\0')
{
    // para já vamos assumir que quem jogou em [0, 0] é o vencedor.
    jogadorVencedor = tabuleiro[0, 0];

    // vai varrer a diagonal \ canto sup. esquerdo -> canto inf. direito (até à penúltima
    célula) à procura de um vencedor
    for(int i = 0; i < tabuleiro.GetLength(0) - 1; i++)
    {
        // se for uma jogada vazia ou se a célula atual for diferente da próxima
        if(tabuleiro[i,i] == '\0' || tabuleiro[i,i] != tabuleiro[i + 1, i + 1])
        {
            // para já não há vencedor
            jogadorVencedor = '\0';

            // não vale a pena continuar a procurar na diagonal , posso sair do ciclo
            break;
        }
    }
}

// se ainda não encontrei um vencedor, então continuo à procura
if(jogadorVencedor == '\0')
{
    // para já vamos assumir que quem jogou em [0, tabuleiro.GetLength(1) - 1] é o vencedor.
    jogadorVencedor = tabuleiro[0, tabuleiro.GetLength(1) - 1];

    // vai varrer a diagonal / canto sup. direito -> canto inf. esquerdo (até à penúltima
    célula) à procura de um vencedor
    for(int i = 0; i < tabuleiro.GetLength(0) - 1; i++)
    {
        // para não ter de calcular isto duas vezes e ser mais fácil de ler
        int colunaCalculada = tabuleiro.GetLength(1) - 1 - i;

        // se for uma jogada vazia ou se a célula atual for diferente da próxima
        if(tabuleiro[i,colunaCalculada] == '\0' || tabuleiro[i,colunaCalculada] !=
        tabuleiro[i + 1, colunaCalculada - 1])
        {
            // para já não há vencedor
            jogadorVencedor = '\0';

            // não vale a pena continuar a procurar na diagonal, posso sair do ciclo
            break;
        }
    }
}
```

```
// quando chegar aqui ou tem o jogadorVencedor, ou então tem a jogada vazia (carater
'\0')
return jogadorVencedor;
}

public static void MostrarTabuleiro(char[,] tabuleiro)
{
    for(int i = 0; i < tabuleiro.GetLength(0); i++)
    {
        for(int j = 0; j < tabuleiro.GetLength(1); j++)
        {
            // imprime um underscore no ecrã se célula atual for uma jogada vazia, caso contrário
            escreve a letra do jogador que lá jogou
            Console.Write(tabuleiro[i,j] == '\0' ? " _ " : " " + tabuleiro[i,j] + " ");
        }

        Console.WriteLine();
    }
}

public static void Main(string[] args)
{
    // tabuleiro de 3 x 3
    char [,] tabuleiro = new char[3,3];
    char vencedor;
    int linhaJogada, colunaJogada; // podíamos declarar estas variáveis dentro do ciclo
do...while, mas assim evitamos operações sucessivas na stack de criar e destruir variáveis
    char jogadorAtual = 'O'; // controla o jogador que está atualmente a fazer uma jogada. O
primeiro será sempre o 'O'

    do
    {
        // a cada jogada mostra o estado atual do tabuleiro
        MostrarTabuleiro(tabuleiro);

        // e mostra também quem vai jogar agora
        Console.WriteLine("O jogador atual é: {0}", jogadorAtual);

        // pede a localização da célula onde quer jogar
        Console.Write("Indique a linha onde quer jogar (1 a {0}): ", tabuleiro.GetLength(0));
        linhaJogada = Convert.ToInt32(Console.ReadLine()) - 1; // subtraímos 1 para "converter"
a posição para índice

        Console.Write("Indique a coluna onde quer jogar (1 a {0}): ", tabuleiro.GetLength(1));
        colunaJogada = Convert.ToInt32(Console.ReadLine()) - 1; // subtraímos 1 para
"converter" a posição para índice
    }
}
```

```

    // tenta fazer a jogada e avaliar o resultado dessa jogada (que é true se conseguiu
fazer, ou false caso contrário)
    if(!FazerJogada(tabuleiro, jogadorAtual, linhaJogada, colunaJogada))
    {
        Console.WriteLine("Jogada inválida, por favor tente de novo");
    }
    else
    {
        // muda para o outro jogador
        jogadorAtual = (jogadorAtual == 'O' ? 'X' : 'O');
    }

    // obtém o potencial vencedor (se não houver para já vencedor a variável fica com o
carater da jogada vazia)
    vencedor = ObterVencedor(tabuleiro);
} while(vencedor == '\0' && !ExisteEmpate(tabuleiro)); // enquanto não houver vencedor e
não houver empate

// se o vencedor for diferente de \0, então é um dos jogadores (é X ou é O)
if(vencedor != '\0')
{
    Console.WriteLine("Parabéns jogador {0}, venceu o jogo!", vencedor);
}
else
{
    Console.WriteLine("Jogo terminou com empate");
}
MostrarTabuleiro(tabuleiro);
}
}
}

```

MÓDULO 11 – POO

a) Exercícios PPT

```
using System;
```

```
namespace Primavera.OOP
```

```
{
```

```
class MainClass {  
    public static void Main (string[] args)  
    {  
        CorrerExercicio4();  
    }  
}
```

```
// Exercício 1 abordagem procedimental
```

```
public static void CorrerExercicio1()
```

```
{
```

```
    int divisor, dividendo;
```

```
    float resultado;
```

```
    Console.WriteLine ("Introduza o dividendo");
```

```
    dividendo = Convert.ToInt32(Console.ReadLine());
```

```
    Console.WriteLine ("Introduza o divisor");
```

```
    divisor = Convert.ToInt32(Console.ReadLine());
```

```
    resultado = (float) dividendo / divisor;
```

```
    Console.WriteLine("O resultado é: {0}", resultado);
```

```
}
```

```
public static void CorrerExercicio2()
```

```
{
```

```
    int divisor, dividendo;
```

```
    Console.WriteLine ("Introduza o dividendo");
```

```
    dividendo = Convert.ToInt32(Console.ReadLine());
```

```
    Console.WriteLine ("Introduza o divisor");
```

```
    divisor = Convert.ToInt32(Console.ReadLine());
```

```
// cria uma nova instância da classe que representa uma divisão
```

```
Divisao div = new Divisao(dividendo, divisor);
```

```
// invoca o método responsável por fazer o cálculo
```

```
div.RealizarOperacao();
```

```
// mostra o resultado que o método GetResultado da classe Divisao devolve
Console.WriteLine("O resultado é: {0}", div.GetResultado());
}
```

```
public static void CorrerExercicio3()
{
    int divisor, dividendo;
```

```
    Console.WriteLine ("Introduza o dividendo");
    dividendo = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine ("Introduza o divisor");
    divisor = Convert.ToInt32(Console.ReadLine());
```

// Cria uma instância de DivisaoConcreta. Como a classe abstrata tem tudo o que precisamos de usar a seguir, a variável é desse tipo (é boa prática declarar variáveis de tipos o mais abstrato possível quando lidamos com classes em hierarquia)

```
AbstractOperacaoAritmetica div = new DivisaoConcreta(dividendo, divisor);
```

```
// invoca o método responsável por fazer o cálculo
div.RealizarOperacao();
```

```
// mostra o resultado que o método GetResultado da classe Divisao devolve
Console.WriteLine("O resultado é: {0}", div.GetResultado());
}
```

```
public static void CorrerExercicio4()
{
    int divisor, dividendo;
```

```
    Console.WriteLine ("Introduza o dividendo");
    dividendo = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine ("Introduza o divisor");
    divisor = Convert.ToInt32(Console.ReadLine());
```

// Cria uma instância da classe DivisaoConcretaComInterface. Desta vez é a interface que tem tudo o que precisamos, pelo que podemos declarar a variável

```
IExecutorDeOperacao div = new DivisaoComInterface(dividendo, divisor);
```

```
// invoca o método responsável por fazer o cálculo
div.RealizarOperacao();
```

```
// mostra o resultado que o método GetResultado da classe Divisao devolve
Console.WriteLine("O resultado é: {0}", div.GetResultado());
}
```

```

}

//-----
/**
  Para exercício 2
 */
public class Divisao
{

  // atributos da classe. São privados porque não é preciso chegar a eles fora da classe
  private int divisor, dividendo;
  private float resultado;

  // construtor da classe. Para instanciar tenho de fazer new Divisao(umValor, outroValor)
  public Divisao(int dividendo, int divisor)
  {
    // inicializo os valores dos atributos da classe com o que entrou no construtor
    // NOTA: usa-se o this. para não haver dúvida que se está a aceder a um atributo da
    classe
    this.divisor = divisor;
    this.dividendo = dividendo;

    // aproveito para por o resultado a zero
    this.resultado = 0;
  }

  // Método responsável por realizar a operação e guardar o valor em resultado
  public void RealizarOperacao()
  {
    // guarda o resultado da operação
    this.resultado = (float) this.dividendo / this.divisor;
  }

  // método responsável por devolver o resultado. Como resultado é privado (está encapsulado),
  usamos um método para chegar ao valor desse atributo
  public float GetResultado()
  {
    return this.resultado;
  }
}

//-----
/**
  Para exercício 3
 */

```

```
// Uma classe abstrata que representa qualquer operação aritmética com dois operandos
public abstract class AbstractOperacaoAritmetica
{
    // os atributos aqui são protected porque preciso de chegar a eles nas classes derivadas
    protected int operandoA, operandoB;
    protected float resultado;

    public AbstractOperacaoAritmetica(int operandoA, int operandoB)
    {
        this.operandoA = operandoA;
        this.operandoB = operandoB;
        this.resultado = 0;
    }

    // a este nível só sei que terei mais tarde de realizar a operação, mas não sei como é que
    // ela se comporta. Por isso é abstrata.
    // A partir do momento em que há pelo menos um método abstrato numa classe, a classe também
    // tem de ser abstrata.
    public abstract void RealizarOperacao();

    // método responsável por devolver o resultado. Como resultado é privado (está encapsulado),
    // usamos um método para chegar ao valor desse atributo
    public float GetResultado(){
        return this.resultado;
    }
}

// Esta classe é uma derivada da AbstractOperacaoAritmetica
public class DivisaoConcreta: AbstractOperacaoAritmetica
{
    // :base(dividendo, divisor) significa: executa o construtor da classe mãe passando os
    // argumentos dividendo e divisor
    public DivisaoConcreta(int dividendo, int divisor)
        :base(dividendo, divisor)
    {
    }

    // como estou a "substituir" um método da classe mãe, tenho de usar a palavra chave override.
    // Neste caso a substituição define o comportamento da divisão
    override public void RealizarOperacao(){
        this.resultado = (float) this.operandoA / this.operandoB;
    }
}
```

```
//-----
/**
 * Para exercício 4
 */
```

// Representa qualquer coisa que seja uma operação e que tenha um resultado, mas não sei sequer quantos operandos tem, só sei que posso realizar a operação e que posso obter o resultado. Um interface é perfeito para este tipo de coisas.

```
public interface IExecutorDeOperacao
{
    void RealizarOperacao();
    float GetResultado();
}
```

// Esta classe compromete-se a ter tudo o que a interface IExecutorDeOperacao tem

```
public class DivisaoComInterface: IExecutorDeOperacao
{
    // atributos são privados porque não há necessidade de mais "alguém" mexer neles
    private int divisor, dividendo;
    private float resultado;

    // construtor da classe
    public DivisaoComInterface(int dividendo, int divisor)
    {
        this.dividendo = dividendo;
        this.divisor = divisor;

        this.resultado = 0;
    }

    // implementação do método definido na interface
    public void RealizarOperacao()
    {
        this.resultado = (float) this.dividendo / this.divisor;
    }

    // implementação do método definido na interface
    public float GetResultado(){
        return this.resultado;
    }
}
```


b) Loja de Informática

```
using System;
```

```
namespace Primavera.Managers  
{
```

```
class ArranquePrograma {  
    public static void Main (string[] args) {  
  
        Pessoa cliente = new Pessoa("João", "211111111");  
  
        Vendavel disco = new DiscoHdd("Western Digital Blue 2TB", 123.30F, 5400, 2000);  
        Vendavel instalarWindows = new Servico("Instalar Windows", 30, 2);  
  
        CestoDeCompras cesto = new CestoDeCompras(cliente);  
  
        cesto.adicionarItem(disco);  
        cesto.adicionarItem(instalarWindows);  
  
        Console.WriteLine("O total do seu cesto de compras é: {0}", cesto.getPrecoTotal());  
  
    }  
}
```

```
public class Vendavel  
{  
    protected string designacao;  
    private float preco;  
  
    public Vendavel(string designacao, float preco)  
    {  
        this.designacao = designacao;  
        this.preco = preco;  
    }  
  
    public string getDesignacao()  
    {  
        return this.designacao;  
    }  
  
    public virtual float getPreco()
```

```
{  
    return this.preco;  
}  
}
```

```
public class Servico : Vendavel  
{  
    protected int tempo;  
  
    public Servico(string designacao, float precoPorUnidadeTempo, int tempo)  
        :base(designacao, precoPorUnidadeTempo)  
    {  
        this.tempo = tempo;  
    }  
  
    override public float getPreco()  
    {  
        return base.getPreco() * this.tempo;  
    }  
}
```

```
public class Produto : Vendavel  
{  
    public Produto(string designacao, float preco)  
        :base(designacao, preco)  
    {}  
}
```

```
public class DiscoHdd : Produto  
{  
    private int rotacoes;  
    private long capacidade;  
  
    public DiscoHdd(string designacao, float preco, int rotacoes, long capacidade)  
        :base(designacao, preco)  
    {  
        this.rotacoes = rotacoes;  
        this.capacidade = capacidade;  
    }  
}
```

```
public class PlacaGrafica : Produto  
{  
    private int memoria;  
    private int frequencia;
```

```
public PlacaGrafica(string designacao, float preco, int memoria, int frequencia)
    :base(designacao, preco)
{
    this.memoria = memoria;
    this.frequencia = frequencia;
}
}
```

```
public class Pessoa
{
    protected string nome;
    protected string nif;

    public Pessoa(string nome, string nif)
    {
        this.nome = nome;
        this.nif = nif;
    }
}
```

```
public class Funcionario : Pessoa
{
    protected int numFuncionario;

    public Funcionario(string nome, string nif, int numFuncionario)
        :base(nome, nif)
    {
        this.numFuncionario = numFuncionario;
    }
}
```

```
public class CestoDeCompras
{
    protected Pessoa cliente;
    private Vendavel[] listaItems;
    private int quantProdutos;

    public CestoDeCompras(Pessoa cliente)
    {
        this.cliente = cliente;

        this.listaItems = new Vendavel[100];
        this.quantProdutos = 0;
    }
}
```

```
public void adicionarItem(Vendavel item){
    this.listaItems[this.quantProdutos] = item;
}
```

```
        ++this.quantProdutos;
    }

    public float getPrecoTotal(){
        float precoTotal = 0;

        for(int i = 0; i < this.quantProdutos; i++){
            precoTotal += this.listaItems[i].getPreco();
        }

        return precoTotal;
    }
}
```