

Curso C# primeiros passos: Lógica de Programação e Algoritmos

Capítulo: Estrutura sequencial

Conteúdo

- Expressões aritméticas
- Variáveis e tipos básicos de C#
- Três operações básicas de programação
 - Entrada de dados
 - Processamento de dados
 - Saída de dados
- Funções matemáticas em C#

- Nesta seção do curso:
 - Explicações teórico-práticas
 - Excelente para aprendizado e revisão
 - PRÁTICA!
 - Exercícios resolvidos
 - Debug (execução passo a passo)
 - URI Online Judge
 - Exercícios resolvidos
 - Exercícios propostos

Expressões aritméticas

Expressões aritméticas



$$4 + 5 \xrightarrow{\text{resultado}} 9$$

Operadores aritméticos

**C, C++,
Java, C#** →

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão ("mod")

Precedência: 1º lugar: * / %
2º lugar: + -

Exemplos de expressões aritméticas

$$2 * 6 / 3$$

Resultado = 4

$$3 + 2 * 4$$

Resultado = 11

$$(3 + 2) * 4$$

Resultado = 20

$$60 / (3 + 2) * 4$$

Resultado = 48

$$60 / ((3 + 2) * 4)$$

Resultado = 3

Exemplos com o operador "mod"

$$14 \% 3$$

Resultado = 2

$$19 \% 5$$

Resultado = 4

Pois:

$$\begin{array}{r|l} 14 & 3 \\ \hline & 4 \\ \hline 2 & \end{array}$$

$$\begin{array}{r|l} 19 & 5 \\ \hline & 3 \\ \hline 4 & \end{array}$$

Variáveis e tipos básicos em C#

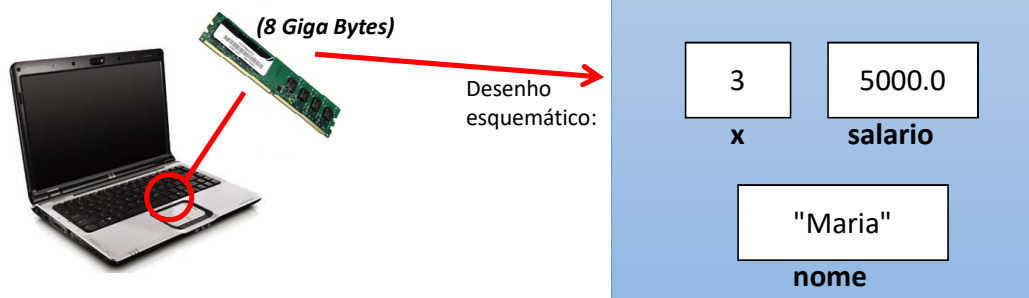
Visão geral

- Um programa de computador em execução lida com dados
- Como esses dados são armazenados?
- Em **VARIÁVEIS!**

Variáveis

Definição informal:

Em programação, uma variável é uma porção de memória (RAM) utilizada para armazenar dados durante a execução dos programas.



Declaração de variáveis

Sintaxe:

```
<tipo> <nome> = <valor inicial>;
```

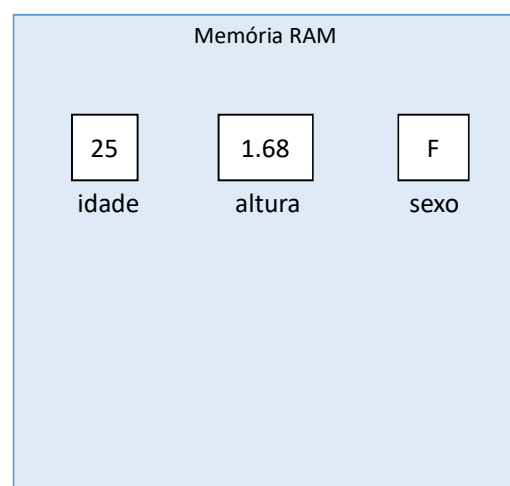
(opcional)

Exemplos:

```
int idade = 25;  
double altura = 1.68;  
char sexo = 'F';
```

Uma variável possui:

- Nome (ou identificador)
- Tipo
- Valor
- Endereço



Tipos básicos em C#

byte	0 .. 255
sbyte	-128 .. 127
short	-32,768 .. 32,767
ushort	0 .. 65,535
int	-2,147,483,648 .. 2,147,483,647
uint	0 .. 4,294,967,295
long	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807
ulong	0 .. 18,446,744,073,709,551,615

Um bit pode armazenar 2 valores possíveis (0 ou 1)

Cada bit = 2 posibilidades

8 bits:

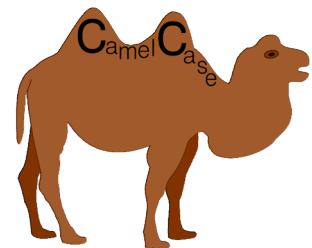
[illegible]

Tipos básicos em C#

float	-3.402823e38 .. 3.402823e38
double	-1.79769313486232e308 .. 1.79769313486232e308
decimal	-79228162514264337593543950335 .. 79228162514264337593543950335
char	A Unicode character.
string	A string of Unicode characters.
bool	True or False.
object	An object.

Nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não pode ter espaço em branco
- Não usar acentos ou til
- Sugestão: use o padrão "camel case"



Errado:

```
int 5minutos;  
int salário;  
int salário do funcionario;
```

Correto:

```
int _5minutos;  
int salario;  
int salarioDoFuncionario;
```


Variáveis e tipos básicos em C#

- Conceito informal
- Declaração de variáveis: `<tipo> <nome> = valor;`
- Tipos básicos em C#
- Nomes de variáveis / padrão camel case

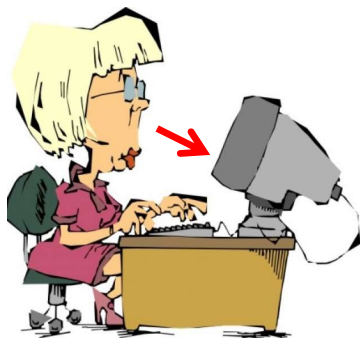
As três operações básicas de programação

Um programa de computador é capaz de realizar essencialmente três operações:



Entrada de dados

Usuário → Programa
(dentro de variáveis)



Dispositivo de ENTRADA



Também chamada de
LEITURA:

"O programa está lendo dados."

Processamento de dados

É quando o programa realiza os cálculos



O processamento de dados se dá por um comando chamado **ATRIBUIÇÃO**

```
media = (x + y) / 2.0;
```

Saída de dados

Programa → Usuário



Dispositivo de SAÍDA



Também chamada de **ESCRITA:**

"O programa está escrevendo dados."

As três operações básicas de programação

- Entrada de dados (leitura)
 - Processamento de dados
 - Saída de dados (escrita)
- VAMOS VER A SEGUIR COMO FAZER NA PRÁTICA CADA UMA DESSAS TRÊS OPERAÇÕES

Saída de dados em C#

Um programa de computador é capaz de realizar essencialmente três operações:



Saída de dados

Programa → Usuário



Dispositivo de SAÍDA



Também chamada de
ESCRITA:

"O programa está escrevendo dados."

Para escrever na tela um texto qualquer

Sem quebra de linha ao final:

```
Console.Write("Bom dia!");
```

Com quebra de linha ao final:

```
Console.WriteLine("Bom dia!");
```

Para escrever o conteúdo de uma variável com ponto flutuante

Suponha uma variável tipo **double** declarada e iniciada:

```
double x = 10.35784;
```

%n = quebra de linha
(independente de
plataforma)

```
Console.WriteLine(x);
```

```
Console.WriteLine(x.ToString("F2"));
```

```
Console.WriteLine(x.ToString("F4"));
```

```
Console.WriteLine(x.ToString("F2", CultureInfo.InvariantCulture));
```


using System.Globaliztion;

Para concatenar vários elementos em um mesmo comando de escrita

Regra geral:

elemento1 + elemento2 + elemento3 + ... + elementoN

```
Console.WriteLine("RESULTADO = " + x);
```

```
Console.WriteLine("O valor do troco é " + x + " reais");
```

```
Console.WriteLine("O valor do troco é " + x.ToString("F2") + " reais");
```

```
using System;
using System.Globalization;

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

            double x = 10.35784;
            int y = 32;
            string z = "Maria";
            char w = 'F';

            Console.Write("Olá mundo!");
            Console.WriteLine("Bom dia!");
            Console.WriteLine("Até mais!");

            Console.WriteLine();
            Console.WriteLine(x);
            Console.WriteLine(x.ToString("F2"));
            Console.WriteLine(x.ToString("F4"));
            Console.WriteLine(x.ToString("F2", CultureInfo.InvariantCulture));

            Console.WriteLine();
            Console.WriteLine("RESULTADO = " + x);
            Console.WriteLine("O valor do troco é " + x + " reais");
            Console.WriteLine("O valor do troco é " + x.ToString("F2") + " reais");

            Console.WriteLine();
            Console.WriteLine("A paciente " + z + " tem " + y + " anos e seu sexo é: " + w);

            Console.ReadLine();
        }
    }
}
```

Saída de dados em C#

- Console.Write
- Console.WriteLine
- x.ToString("F2")
- CultureInfo.InvariantCulture
- Como concatenar vários elementos em um mesmo write
- Exemplo completo


Processamento de dados em C#, Casting

Processamento de dados

Comando de atribuição.

Sintaxe:

<variável> = <expressão>;



Lê-se "recebe"

REGRA:

- 1) A expressão é calculada
- 2) O resultado da expressão é armazenado na variável

Exemplo 1

```
int x, y;
```

```
x = 5;
```

```
y = 2 * x;
```

```
Console.WriteLine(x);
```

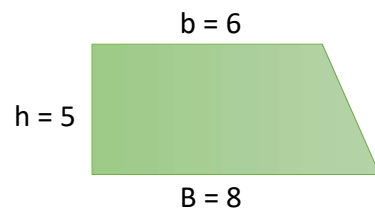
```
Console.WriteLine(y);
```

Exemplo 2

```
int x;  
double y;  
  
x = 5;  
  
y = 2 * x;  
  
Console.WriteLine(x);  
Console.WriteLine(y);
```

Exemplo 3

```
double b, B, h, area;  
  
b = 6.0;  
B = 8.0;  
h = 5.0;  
  
area = (b + B) / 2.0 * h;  
  
Console.WriteLine(area);
```



$$area = \frac{(b + B)}{2} \times h$$

No exemplo:

$$\begin{aligned} area &= \frac{(6 + 8)}{2} \times 5 \\ &= \frac{14}{2} \times 5 = 7 \times 5 = 35 \end{aligned}$$

```
double b, B, h, area;
```

```
b = 6.0;
```

```
B = 8.0;
```

```
h = 5.0;
```

```
area = (b + B) / 2.0 * h;
```

```
Console.WriteLine(area);
```

Boa prática:

Sempre indique o tipo do número, se a expressão for de ponto flutuante (não inteira).

Para **double** use:
.0

Para **float** use:
f

```
float b, B, h, area;
```

```
b = 6f;
```

```
B = 8f;
```

```
h = 5f;
```

```
area = (b + B) / 2f * h;
```

```
Console.WriteLine(area);
```

Boa prática:

Sempre indique o tipo do número, se a expressão for de ponto flutuante (não inteira).

Para **double** use:
.0

Para **float** use:
f

Exemplo 4

```
int a, b;  
double resultado;  
  
a = 5;  
b = 2;  
  
resultado = a / b;  
  
Console.WriteLine(resultado);
```

Casting

É a conversão explícita de um tipo para outro.

É necessário quando o compilador não é capaz de “adivinhar” que o resultado de uma expressão deve ser de outro tipo.

Exemplo 4

```
int a, b;  
double resultado;  
  
a = 5;  
b = 2;  
  
resultado = a / b;  
  
Console.WriteLine(resultado);
```

Exemplo 4

```
int a, b;  
double resultado;  
  
a = 5;  
b = 2;  
  
resultado = (double) a / b;  
  
Console.WriteLine (resultado);
```

Exemplo 5

```
double a;  
int b;
```

```
a = 5.0;  
b = a;
```

```
Console.WriteLine(b);
```

Processamento de dados / Casting

- Comando de atribuição
- Casting

Entrada de dados em C# - Parte 1

Comando

`Console.ReadLine();`

- Lê da entrada padrão **até a quebra de linha**.
- Retorna os dados lidos **na forma de string**.

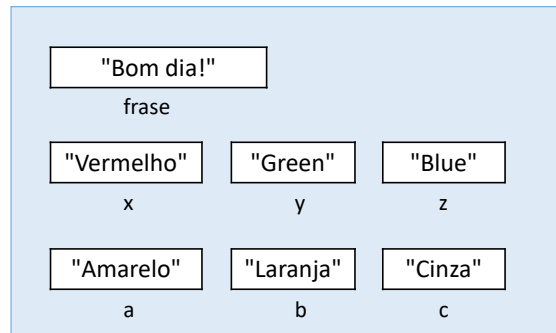
Checklist

- Ler um texto até a quebra de linha e armazenar em uma variável
- Ler três palavras, uma em cada linha, armazenando cada uma em uma variável
- Ler três palavras na mesma linha, separadas por espaço, armazenando cada uma em uma variável

Console:

```
Bom dia!  
Vermelho  
Verde  
Azul  
Amarelo Laranja Cinza
```

Memória:

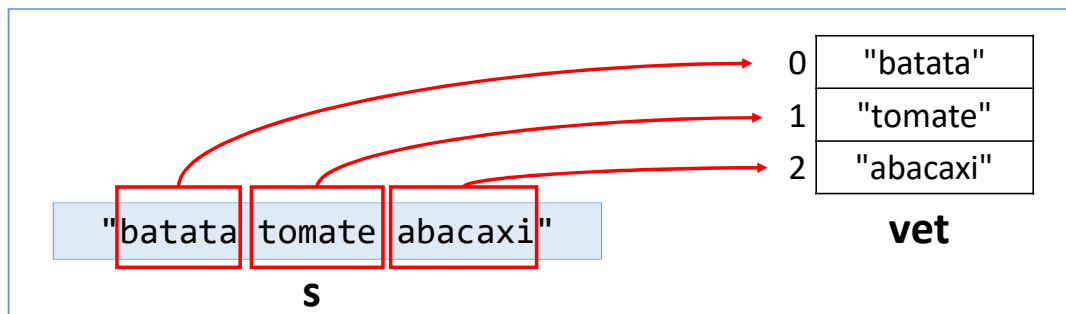
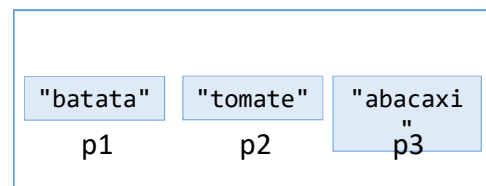


Split

```
string s = Console.ReadLine();  
  
string[] vet = s.Split(' ');  
string p1 = vet[0];  
string p2 = vet[1];  
string p3 = vet[2];
```

Console:

```
batata tomate abacaxi
```



Demo

```
string frase = Console.ReadLine();
string x = Console.ReadLine();
string y = Console.ReadLine();
string z = Console.ReadLine();

string[] v = Console.ReadLine().Split(' ');
string a = v[0];
string b = v[1];
string c = v[2];

Console.WriteLine("Você digitou: ");
Console.WriteLine(frase);
Console.WriteLine(x);
Console.WriteLine(y);
Console.WriteLine(z);
Console.WriteLine(a);
Console.WriteLine(b);
Console.WriteLine(c);
```

Entrada de dados em C# - Parte 2

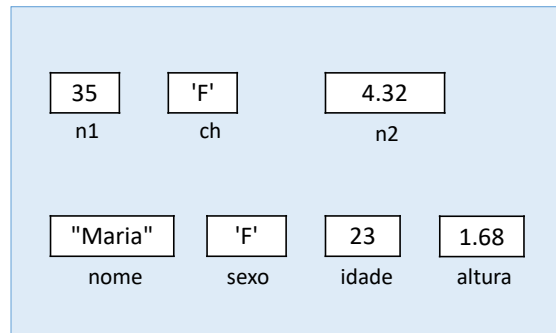
Checklist

- Ler um número inteiro
- Ler um caractere
- Ler um número double
- Ler um nome (única palavra), sexo (caractere F ou M), idade (inteiro) e altura (double) na mesma linha, armazenando-os em quatro variáveis com os devidos tipos

Console:

```
35
F
4.32
Maria F 23 1.68
```

Memória:



Demo

```
int n1 = int.Parse(Console.ReadLine());
char ch = char.Parse(Console.ReadLine());
double n2 = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

string[] vet = Console.ReadLine().Split(' ');
string nome = vet[0];
char sexo = char.Parse(vet[1]);
int idade = int.Parse(vet[2]);
double altura = double.Parse(vet[3], CultureInfo.InvariantCulture);

Console.WriteLine("Você digitou:");
Console.WriteLine(n1);
Console.WriteLine(ch);
Console.WriteLine(n2.ToString("F2", CultureInfo.InvariantCulture));
Console.WriteLine(nome);
Console.WriteLine(sexo);
Console.WriteLine(idade);
Console.WriteLine(altura.ToString("F2", CultureInfo.InvariantCulture));
```

Exercício de fixação

Fazer um programa para executar a seguinte interação com o usuário, lendo os valores destacados em vermelho, e depois mostrar os dados na tela:

Entre com seu nome completo:
Alex Green
Quantos quartos tem na sua casa?
3
Entre com o preço de um produto:
500.50
Entre seu último nome, idade e altura (mesma linha):
Green 21 1.73

SAÍDA ESPERADA (NÚMEROS
REAIS COM DUAS CASAS
DECIMAIS):

Alex Green
3
500.50
Green
21
1.73

(correção na próxima página)

```
using System;
using System.Globalization;
namespace Course {
    class Program {
        static void Main(string[] args) {

            Console.WriteLine("Entre com seu nome completo:");
            string fullName = Console.ReadLine();
            Console.WriteLine("Quantos quartos tem na sua casa?");
            int bedrooms = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter product price:");
            double price = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            Console.WriteLine("Entre seu último nome, idade e altura (mesma linha):");
            string[] vect = Console.ReadLine().Split(' ');
            string lastName = vect[0];
            int age = int.Parse(vect[1]);
            double height = double.Parse(vect[2], CultureInfo.InvariantCulture);

            Console.WriteLine(fullName);
            Console.WriteLine(bedrooms);
            Console.WriteLine(price.ToString("F2", CultureInfo.InvariantCulture));
            Console.WriteLine(lastName);
            Console.WriteLine(age);
            Console.WriteLine(height.ToString("F2", CultureInfo.InvariantCulture));

        }
    }
}
```

Funções matemáticas

Algumas funções matemáticas em C#

Exemplo	Significado
<code>A = Math.Sqrt(x);</code>	Variável A recebe a raiz quadrada de x
<code>A = Math.Pow(x, y);</code>	Variável A recebe o resultado de x elevado a y
<code>A = Math.Abs(x);</code>	Variável A recebe o valor absoluto de x

```

using System;
using System.Globalization;

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

            double x = 3.0;
            double y = 4.0;
            double z = -5.0;
            double A, B, C;

            A = Math.Sqrt(x);
            B = Math.Sqrt(y);
            C = Math.Sqrt(25.0);
            Console.WriteLine("Raiz quadrada de " + x + " = " + A);
            Console.WriteLine("Raiz quadrada de " + y + " = " + B);
            Console.WriteLine("Raiz quadrada de 25 = " + C);

            A = Math.Pow(x, y);
            B = Math.Pow(x, 2.0);
            C = Math.Pow(5.0, 2.0);
            Console.WriteLine(x + " elevado a " + y + " = " + A);
            Console.WriteLine(x + " elevado ao quadrado = " + B);
            Console.WriteLine("5 elevado ao quadrado = " + C);

            A = Math.Abs(y);
            B = Math.Abs(z);
            Console.WriteLine("Valor absoluto de " + y + " = " + A);
            Console.WriteLine("Valor absoluto de " + z + " = " + B);

            Console.ReadLine();

        }
    }
}

```

Incluindo funções em expressões maiores

$$x = \frac{-b \pm \sqrt{\Delta}}{2.a}$$

$$\Delta = b^2 - 4ac$$

```
delta = Math.Pow(b, 2.0) - 4*a*c;
```

```
x1 = (-b + Math.Sqrt(delta)) / (2.0 * a);
```

```
x2 = (-b - Math.Sqrt(delta)) / (2.0 * a);
```

Funções matemáticas

- sqrt – raiz quadrada
- pow – potenciação
- abs – valor absoluto
- Exemplos

Como utilizar o Debug no Visual Studio (execução passo a passo)

Exercício resolvido 01

Fazer um programa para ler as medidas da largura e comprimento de um terreno retangular com uma casa decimal, bem como o valor do metro quadrado do terreno com duas casas decimais. Em seguida, o programa deve mostrar o valor da área do terreno, bem como o valor do preço do terreno, ambos com duas casas decimais, conforme exemplo.

Exemplo:

Entrada:	Saída:
10.0 30.0 200.00	AREA = 300.00 PRECO = 60000.00

```

using System;
using System.Globalization;

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

            double largura, comprimento, precoMetroQuadrado, area, preco;

            largura = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            comprimento = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            precoMetroQuadrado = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);


            area = largura * comprimento;
            preco = area * precoMetroQuadrado;

            Console.WriteLine("AREA = " + area.ToString("F2", CultureInfo.InvariantCulture));
            Console.WriteLine("PRECO = " + preco.ToString("F2", CultureInfo.InvariantCulture));

            Console.ReadLine();
        }
    }
}

```

Como executar o debug do Visual Studio

- Para marcar uma linha de breakpoint:
 - Depurar -> Alternar ponto de interrupção (F9)
- Para iniciar o debug:
 - Depurar -> Iniciar depuração (F5)
- Para executar uma linha:
 - F10
- Para interromper o debug:
 - SHIFT + F5 

Nosso primeiro exercício no URI Online Judge

Nosso primeiro exercício no URI Online Judge

- Cadastrar no URI Online Judge
- Selecionar um problema para resolver
- Enviar a solução do problema
- Prevenção de erros:
 - Wrong answer
 - Presentation error

Exercício URI 1001

```
using System;

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

            int A, B, X;

            A = int.Parse(Console.ReadLine());
            B = int.Parse(Console.ReadLine());

            X = A + B;

            Console.WriteLine("X = " + X);
        }
    }
}
```

Exercícios propostos - PARTE 1

- **Exercícios resolvidos:**

- Resolvido 01
- URI 1001

- **Exercícios propostos do URI:**

- 1002, 1003, 1004, 1007, 1008, 1010, 1014, 1016, 1017
- 1005, 1006 - média ponderada

- **Soluções:**

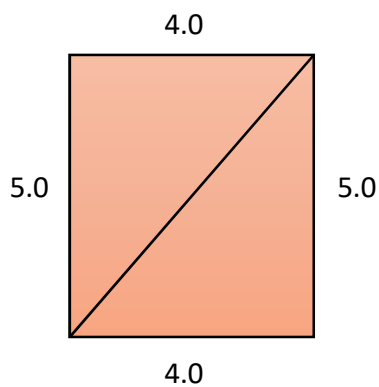
<https://github.com/acenelio/curso-logica-de-programacao-csharp>

Exercício resolvido 02

Fazer um programa para ler as medidas da base e altura de um retângulo. Em seguida, mostrar o valor da área, perímetro e diagonal deste retângulo, com quatro casas decimais, conforme exemplo.

Exemplo:

Entrada:	Saída:
5.0 4.0	AREA = 20.0000 PERIMETRO = 18.0000 DIAGONAL = 6.4031



$$\text{área} = \text{base} \times \text{altura}$$

$$\text{perímetro} = 2 \times \text{base} + 2 \times \text{altura}$$

$$\text{diagonal} = \sqrt{\text{base}^2 + \text{altura}^2}$$

Exercício resolvido: estrutura sequencial

- Funções matemáticas
 - sqrt
 - pow
- Precedência de operadores e uso de parêntesis
- Código fonte na próxima página

```
using System;
using System.Globalization;

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

            double b, a, area, perimetro, diagonal;

            b = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            a = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            area = b * a;
            perimetro = 2 * (b + a);
            diagonal = Math.Sqrt(Math.Pow(b, 2.0) + Math.Pow(a, 2.0));

            Console.WriteLine("AREA = " + area.ToString("F4", CultureInfo.InvariantCulture));
            Console.WriteLine("PERIMETRO = " + perimetro.ToString("F4", CultureInfo.InvariantCulture));
            Console.WriteLine("DIAGONAL = " + diagonal.ToString("F4", CultureInfo.InvariantCulture));

            Console.ReadLine();

        }
    }
}
```

Exercício resolvido 03

Fazer um programa para ler o nome (apenas uma palavra) e idade de duas pessoas. Ao final mostrar uma mensagem com os nomes e a idade média entre essas pessoas, com uma casa decimal, conforme exemplo.

Exemplo:

Entrada:	Saída:
Maria 19 Joaquim 20	A idade média de Maria e Joaquim é de 19.5 anos

Exercício resolvido: estrutura sequencial

- Aplicamos:
 - String
 - Casting
- Código fonte na próxima página

```
using System;
using System.Globalization;

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

            string nome1, nome2;
            int idade1, idade2;
            double media;
            string[] vet;

            vet = Console.ReadLine().Split(' ');
            nome1 = vet[0];
            idade1 = int.Parse(vet[1]);

            vet = Console.ReadLine().Split(' ');
            nome2 = vet[0];
            idade2 = int.Parse(vet[1]);

            media = (double)(idade1 + idade2) / 2.0;

            Console.WriteLine("A idade média de " + nome1 + " e " + nome2 + " é de "
                + media.ToString("F1", CultureInfo.InvariantCulture) + " anos");

            Console.ReadLine();
        }
    }
}
```

Exercícios propostos - PARTE 2

- **Exercícios resolvidos:**

- Resolvidos 02 e 03

- **Exercícios propostos do URI:**

- 1009 - string
- 1011 - casting
- 1012 - geometria
- 1013 - abs
- 1015 - sqrt, pow

- **Soluções:**

<https://github.com/acenelio/curso-logica-de-programacao-csharp>

Exercício resolvido 04 - URI 1019

ANÁLISE:

1 minuto = 60 segundos

1 hora = 60 minutos = 3600 segundos

140153 segundos contém quantas horas?

$140153 / 3600 = \mathbf{38,931388...}$

$38 \times 3600 = 136800$

SOBRA: 3353

$3353 / 60 = \mathbf{55,8833...}$

$55 \times 60 = 3300$

SOBRA: **53**

Exercício resolvido: estrutura sequencial

- URI 1019
- Aplicamos:
 - Operador % ("mod")
- Código fonte na próxima página

```
using System;

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

            int N, resto, horas, minutos, segundos;

            N = int.Parse(Console.ReadLine());

            horas = N / 3600;
            resto = N % 3600;

            minutos = resto / 60;
            segundos = resto % 60;

            Console.WriteLine(horas + ":" + minutos + ":" + segundos);

            Console.ReadLine();

        }
    }
}
```

Exercícios propostos - PARTE 3

- **Exercício resolvido:**

- URI 1019

- **Exercícios propostos do URI:**

- 1018 - mod
- 1020 - mod
- 1021 - mod / (desafio - casting)
- 1061 - mod / string (desafio - interpretação)

- **Soluções:**

<https://github.com/acenelio/curso-logica-de-programacao-csharp>