

Desenvolvimento de Sistemas 2

Prof.: Sidney Silva

Todos estão ouvindo bem?



Por favor desativar microfone do dispositivo



Apresentação do Professor



Fiquem tranquilos!



Apresentação da Unidade curricular

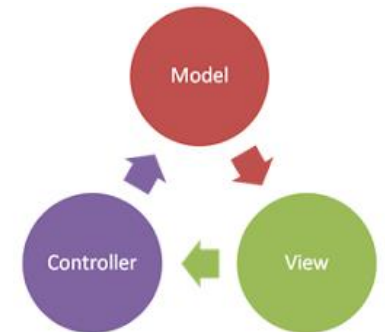
Módulo III
Perfil Profissional: Técnico em Desenvolvimento de Sistemas
Unidade Curricular: Desenvolvimento de Sistemas II
Carga Horária: 72 horas
Unidade de Competência Programar softwares, aplicando metodologias e padrões de desenvolvimento, normas técnicas, de qualidade
Objetivo Geral: Propiciar desenvolvimento de capacidades técnicas e de gestão requeridas para criação de aplicativos por meio de linguagem de programação, de acordo padrão de qualidade, robustez, integridade e segurança.

58h – online

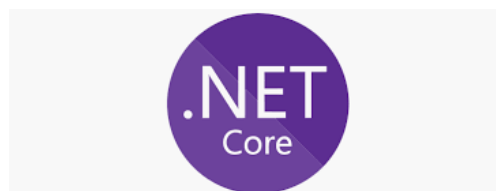
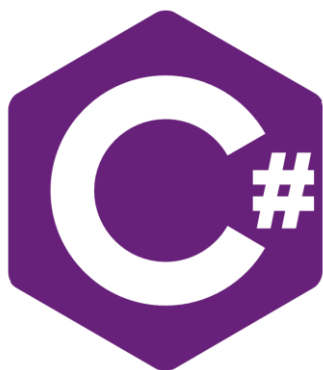
14 - EAD

DS2

SERVIÇO NACIONAL DE
APRENDIZAGEM INDUSTRIAL



DS2



<https://dotnet.microsoft.com/download>

.NET
Core

.NET Core 3.1

.NET Core is a cross-platform version of .NET for building websites, services, and console apps.

Run Apps ⓘ

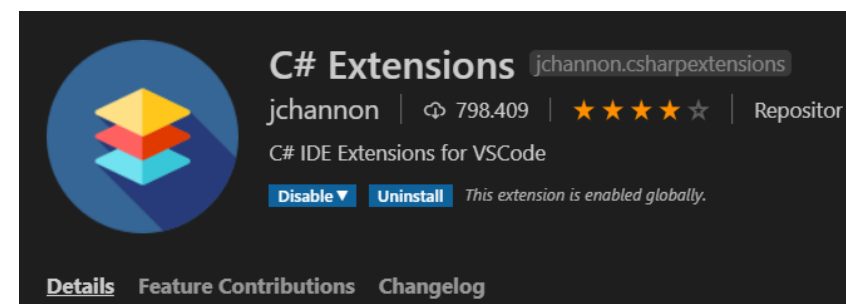
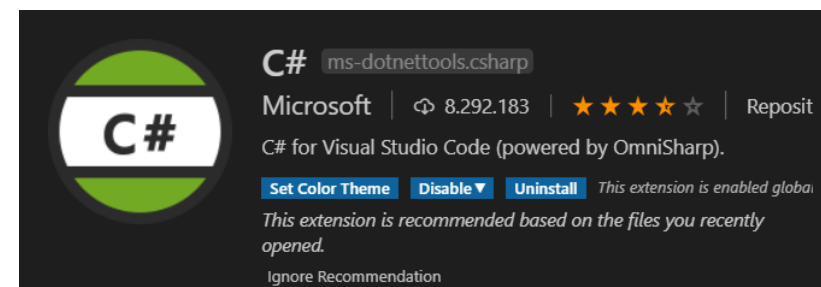
Download .NET Core Runtime

Build Apps ⓘ

Download .NET Core SDK

Advanced ⓘ

All .NET Core downloads...



Criando um novo projeto

<https://docs.microsoft.com/pt-br/dotnet/core/tutorials/with-visual-studio-code>

Crie um projeto de aplicativo de console do .NET Core chamado "HelloWorld".

1. Inicie o Visual Studio Code.
2. Selecione **arquivo** > **abrir pasta** (**arquivo** > **aberto...** no MacOS) no menu principal.
3. Na caixa de diálogo **abrir pasta**, crie uma pasta *HelloWorld* e clique em **Selecionar pasta** (**aberta** no MacOS).

O nome da pasta torna-se o nome do projeto e o nome do namespace por padrão. Você adicionará código posteriormente no tutorial que assume que o namespace do projeto é `HelloWorld`.

4. Abra o **terminal** no Visual Studio Code selecionando **Exibir** > **terminal** no menu principal.

O **terminal** é aberto com o prompt de comando na pasta *HelloWorld*.

5. No **terminal**, digite o seguinte comando:

```
dotnet new console
```

Criando um novo projeto

O modelo cria um simples aplicativo "Olá, Mundo". Ele chama o `Console.WriteLine(String)` método para exibir "Olá, mundo!" na janela do console.

O código de modelo define uma classe, `Program`, com um único método, `Main`, que usa uma `String` matriz como um argumento:

```
using System;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

`Main` é o ponto de entrada do aplicativo, o método que é chamado automaticamente pelo runtime quando ele inicia o aplicativo. Quaisquer argumentos de linha de comando fornecidos quando o aplicativo for iniciado estão disponíveis na matriz `args`.

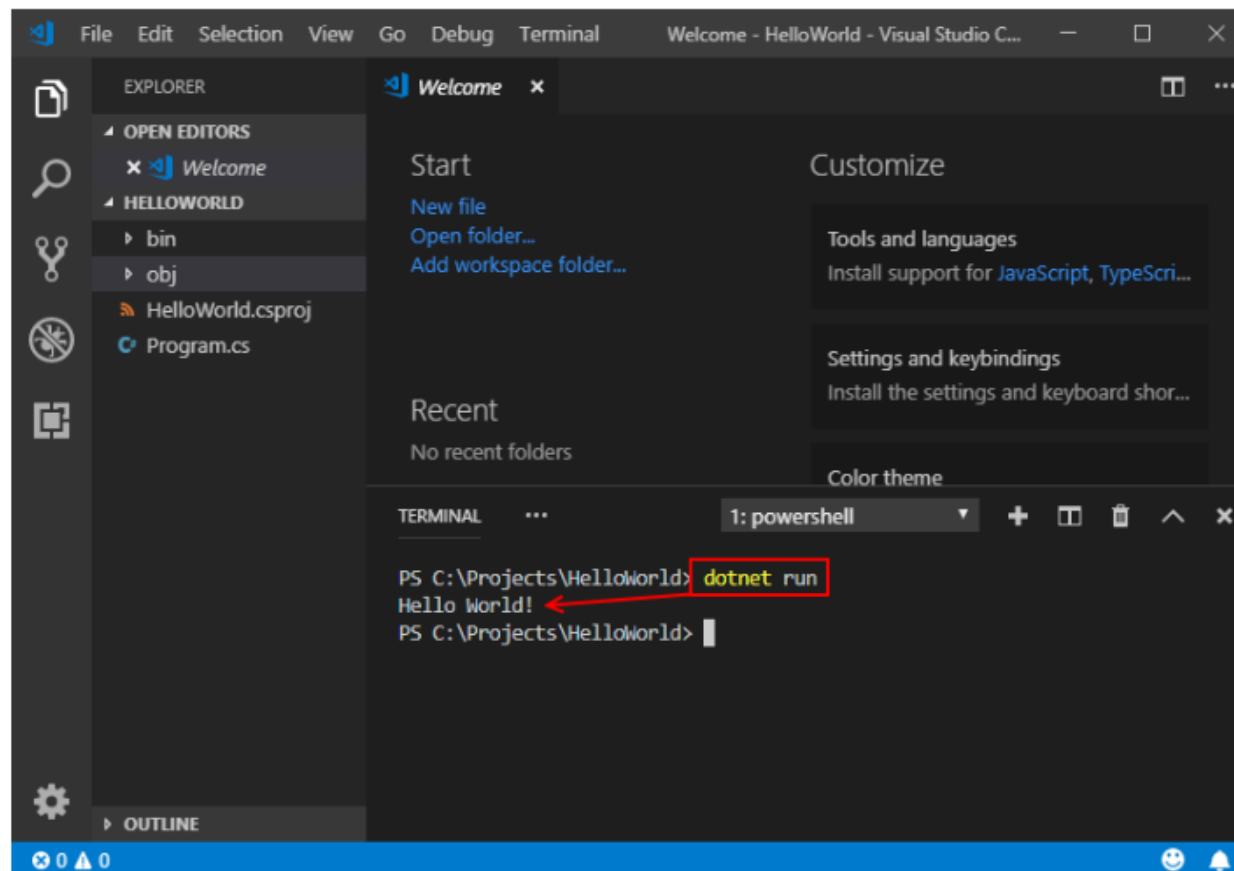
DS2

Rodando código

Execute o seguinte comando no **terminal**:

```
dotnet run
```

O programa exibe "Olá, Mundo!" e termina.



C# e .NET

- C# : uma linguagem de programação (regras sintáticas)
- .NET (2002): uma plataforma de desenvolvimento para se criar diversos tipos de aplicações, podendo usar várias linguagens de programação
 - <https://www.microsoft.com/net/learn/what-is-dotnet>

DS2

.NET

- **BCL - Base Class Library**

[https://msdn.microsoft.com/en-us/library/gg145045\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/gg145045(v=vs.110).aspx)

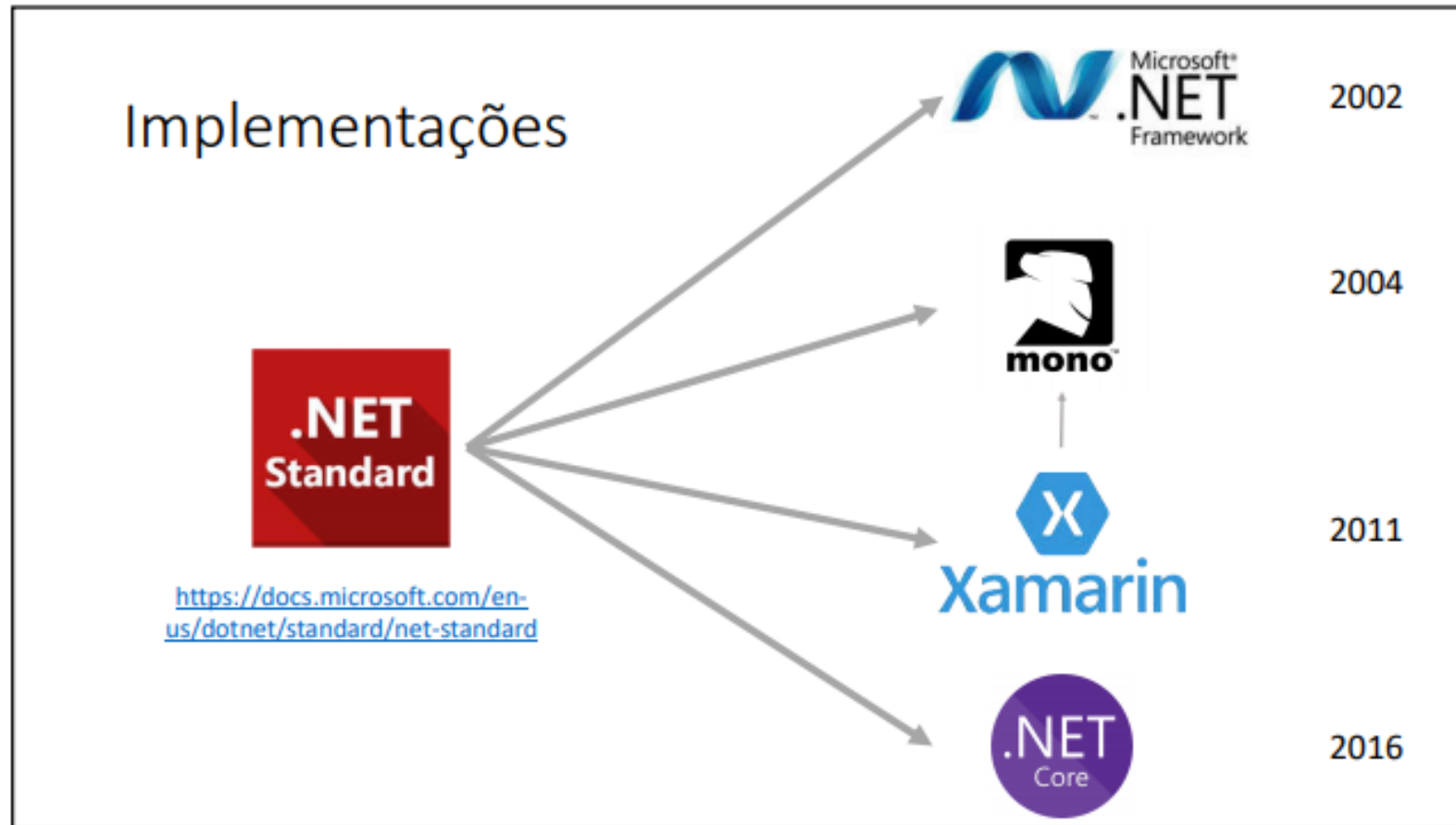
- **CLR - Common Language Runtime (Máquina Virtual)**

- Nota: possui garbage collection (objetos não utilizados são automaticamente desalocados da memória)

<https://www.microsoft.com/net/download>

<https://docs.microsoft.com/pt-br/dotnet/framework/migration-guide/how-to-determine-which-versions-are-installed>

DS2



DS2

Características

- C# é uma linguagem **estaticamente tipada**
- Tipos valor pré-definidos em C#
- Tipos referência pré-definidos em C#
- Variável não atribuída
- Overflow
- Padrão para float: sufixo "f"
- Padrão para char: aspas simples
- Padrão para string: aspas duplas
- Padrão para bool: true, false

DS2

Tipo de dados

C# Type	.Net Framework Type	Signed	Bytes	Possible Values
sbyte	System.Sbyte	Yes	1	-128 to 127
short	System.Int16	Yes	2	-32768 to 32767
int	System.Int32	Yes	4	-2^{31} to $2^{31} - 1$
long	System.Int64	Yes	8	-2^{63} to $2^{63} - 1$
byte	System.Byte	No	1	0 to 255
ushort	System.UInt16	No	2	0 to 65535
uint	System.UInt32	No	4	0 to $2^{32} - 1$
ulong	System.UInt64	No	8	0 to $2^{64} - 1$
float	System.Single	Yes	4	$\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$ with 7 significant figures
double	System.Double	Yes	8	$\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$ with 15 or 16 significant figures
decimal	System.Decimal	Yes	12	$\pm 1.0 \times 10^{-28}$ to $\pm 7.9 \times 10^{28}$ with 28 or 29 significant figures
char	System.Char	N/A	2	Any Unicode character
bool	System.Boolean	N/A	1/2	true or false

Tipo C#	Tipo .NET	Descrição
string	System.String	Uma cadeia de caracteres Unicode IMUTÁVEL (<i>segurança, simplicidade, thread safe</i>)
object	System.Object	Um objeto genérico (toda classe em C# é subclasse de object) GetType Equals GetHashCode ToString

DS2

Tipo de dados

```
bool completo = false;
char genero = 'F';
char letra = '\u0041';
byte n1 = 126;
int n2 = 1000;
int n3 = 2147483647;
long n4 = 2147483648L;
float n5 = 4.5f;
double n6 = 4.5;
String nome = "Maria Green";
Object obj1 = "Alex Brown";
Object obj2 = 4.5f;
```

```
Console.WriteLine(completo);
Console.WriteLine(genero);
Console.WriteLine(letra);
Console.WriteLine(n1);
Console.WriteLine(n2);
Console.WriteLine(n3);
Console.WriteLine(n4);
Console.WriteLine(n5);
Console.WriteLine(n6);
Console.WriteLine(nome);
Console.WriteLine(obj1);
Console.WriteLine(obj2);
```

DS2

Tipo de dados

Funções para valores mínimos e máximos

- `int.MinValue`
- `int.MaxValue`
- `sbyte.MaxValue`
- `long.MaxValue`
- `decimal.MaxValue`
- etc...

DS2

Tipo de dados

Restrições para nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não usar acentos ou til
- Não pode ter espaço em branco
- Sugestão: use nomes que tenham um significado

Errado:

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

Correto:

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

DS2

Tipo de dados

Convenções

- Camel Case: `lastName` (parâmetros de métodos, variáveis dentro de métodos)
- Pascal Case: `LastName` (namespaces, classe, properties e métodos)
- Padrão `_lastName` (atributos "internos" da classe)

DS2

Tipo de dados

- Conversão implícita entre tipos

<https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/keywords/implicit-numeric-conversions-table>

- Casting: conversão explícita entre tipos COMPATÍVEIS

Exemplo 1

```
double a;  
float b;  
  
a = 5.1;  
b = (float)a;  
  
Console.WriteLine(b);
```

Exemplo 2

```
double a;  
int b;  
  
a = 5.1;  
b = (int)a;  
  
Console.WriteLine(b);
```

Exemplo 3

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

DS2

Saída de dados

Imprimir na saída padrão (console)

- Comandos
 - `Console.WriteLine(valor);`
 - `Console.Write(valor);`

DS2

Saída de dados

```
using System;
using System.Globalization;

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

            char genero = 'F';
            int idade = 32;
            double saldo = 10.35784;
            String nome = "Maria";

            Console.Write("Bom dia!");
            Console.WriteLine("Boa tarde!");
            Console.WriteLine("Boa noite!");
            Console.WriteLine("-----");
            Console.WriteLine(genero);
            Console.WriteLine(idade);
            Console.WriteLine(saldo);
            Console.WriteLine(nome);
            Console.WriteLine(saldo.ToString("F2"));
            Console.WriteLine(saldo.ToString("F4"));
            Console.WriteLine(saldo.ToString("F4", CultureInfo.InvariantCulture));

        }
    }
}
```


DS2

Saída de dados

Placeholders, concatenação e interpolação

```
int idade = 32;
double saldo = 10.35784;
String nome = "Maria";

Console.WriteLine("{0} tem {1} anos e tem saldo igual a {2:F2} reais", nome, idade, saldo);

Console.WriteLine($"{nome} tem {idade} anos e tem saldo igual a {saldo:F2} reais");

Console.WriteLine(nome + " tem " + idade + " anos e tem saldo igual a "
    + saldo.ToString("F2", CultureInfo.InvariantCulture) + " reais");
```

DS2

Saída de dados

Exercício de fixação

Em um novo programa, inicie as seguintes variáveis:

```
string produto1 = "Computador";  
string produto2 = "Mesa de escritório";  
  
byte idade = 30;  
int codigo = 5290;  
char genero = 'M';  
  
double preco1 = 2100.0;  
double preco2 = 650.50;  
double medida = 53.234567;
```

Em seguida, usando os valores das variáveis, produza a seguinte saída na tela do console:

```
Produtos:  
Computador, cujo preço é $ 2100,00  
Mesa de escritório, cujo preço é $ 650,50  
  
Registro: 30 anos de idade, código 5290 e gênero: M  
  
Medida com oito casas decimais: 53,23456700  
Arredondado (três casas decimais): 53,235  
Separador decimal invariant culture: 53.235
```

DS2

Entrada de dados

`Console.ReadLine();`

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

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

DS2

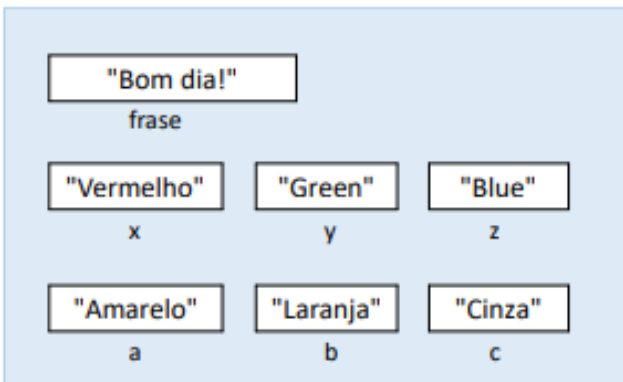
Entrada de dados

- 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:

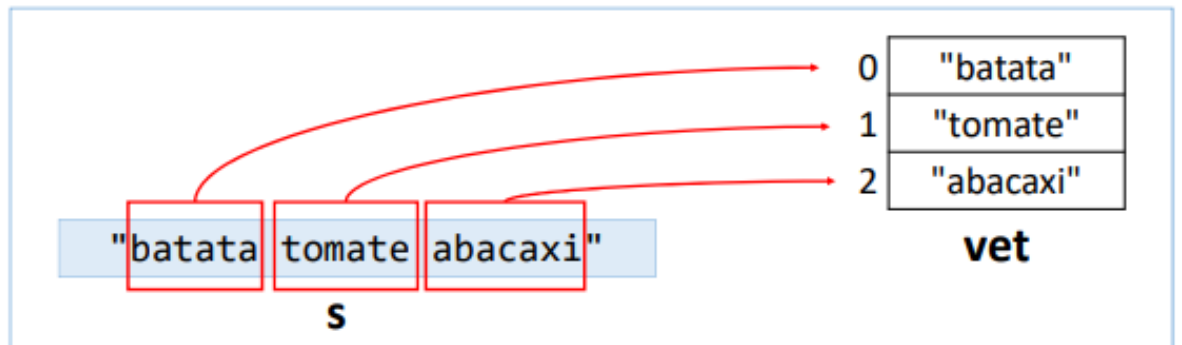
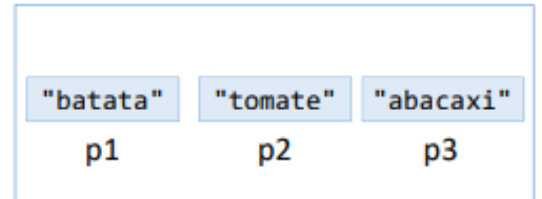


Split

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

Console:

```
batata tomate abacaxi
```



DS2

Entrada de dados

```
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);
```


DS2

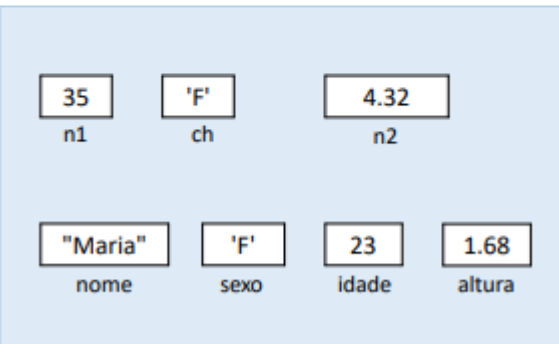
Entrada de dados

- 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:



```
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));
```

DS2

Entrada de dados

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

DS2

Entrada de dados

Operadores aritméticos

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão

NOTAS:

1) * / % tem precedência maior que + -

2) Exemplos:

$3 + 4 * 2 \longrightarrow$ Resultado: 11

$(3 + 4) * 2 \longrightarrow$ Resultado: 14

3) Pode-se usar parêntesis à vontade

4) Exemplo com mod:

$17 \% 3 \longrightarrow$ Resultado: 2

$$\begin{array}{r} 17 \overline{) 3} \\ 2 \quad 5 \end{array}$$

DS2

Exercícios

Faça um programa para ler dois valores inteiros, e depois mostrar na tela a soma desses números com uma mensagem explicativa, conforme exemplos.

Exemplos:

Entrada:	Saída:
10 30	SOMA = 40
Entrada:	Saída:
-30 10	SOMA = -20
Entrada:	Saída:
0 0	SOMA = 0

DS2

Exercícios

Faça um programa para ler o valor do raio de um círculo, e depois mostrar o valor da área deste círculo com quatro casas decimais conforme exemplos.

Fórmula da área: $area = \pi \cdot raio^2$

Considere o valor de $\pi = 3.14159$

Exemplos:

Entrada:	Saída:
2.00	A=12.5664
Entrada:	Saída:
100.64	A=31819.3103
Entrada:	Saída:
150.00	A=70685.7750

DS2

Exercícios

Fazer um programa para ler quatro valores inteiros A, B, C e D. A seguir, calcule e mostre a diferença do produto de A e B pelo produto de C e D segundo a fórmula: $DIFERENCA = (A * B - C * D)$.

Exemplos:

Entrada:	Saída:
5 6 7 8	DIFERENCA = -26

Entrada:	Saída:
5 6 -7 8	DIFERENCA = 86

DS2

Exercícios

Fazer um programa que leia o número de um funcionário, seu número de horas trabalhadas, o valor que recebe por hora e calcula o salário desse funcionário. A seguir, mostre o número e o salário do funcionário, com duas casas decimais.

Exemplos:

Entrada:	Saída:
25 100 5.50	NUMBER = 25 SALARY = U\$ 550.00
Entrada:	Saída:
1 200 20.50	NUMBER = 1 SALARY = U\$ 4100.00
Entrada:	Saída:
6 145 15.55	NUMBER = 6 SALARY = U\$ 2254.75

DS2

Exercícios

Fazer um programa para ler o código de uma peça 1, o número de peças 1, o valor unitário de cada peça 1, o código de uma peça 2, o número de peças 2 e o valor unitário de cada peça 2. Calcule e mostre o valor a ser pago.

Exemplos:

Entrada:	Saída:
12 1 5.30 16 2 5.10	VALOR A PAGAR: R\$ 15.50
Entrada:	Saída:
13 2 15.30 161 4 5.20	VALOR A PAGAR: R\$ 51.40
Entrada:	Saída:
1 1 15.10 2 1 15.10	VALOR A PAGAR: R\$ 30.20

DS2

Exercícios

Fazer um programa que leia três valores com ponto flutuante de dupla precisão: A, B e C. Em seguida, calcule e mostre:

- a) a área do triângulo retângulo que tem A por base e C por altura.
- b) a área do círculo de raio C. ($\pi = 3.14159$)
- c) a área do trapézio que tem A e B por bases e C por altura.
- d) a área do quadrado que tem lado B.
- e) a área do retângulo que tem lados A e B.

Exemplos:

Entrada:	Saída:
3.0 4.0 5.2	TRIANGULO: 7.800 CIRCULO: 84.949 TRAPEZIO: 18.200 QUADRADO: 16.000 RETANGULO: 12.000

Entrada:	Saída:
12.7 10.4 15.2	TRIANGULO: 96.520 CIRCULO: 725.833 TRAPEZIO: 175.560 QUADRADO: 108.160 RETANGULO: 132.080

DS2

Operadores

Operadores comparativos

Operador	Significado
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
==	igual
!=	diferente

```
int a = 10;  
bool c1 = a < 10;  
bool c2 = a < 20;  
bool c3 = a > 10;  
bool c4 = a > 5;  
Console.WriteLine(c1);  
Console.WriteLine(c2);  
Console.WriteLine(c3);  
Console.WriteLine(c4);  
Console.WriteLine("-----");
```

```
bool c5 = a <= 10;  
bool c6 = a >= 10;  
bool c7 = a == 10;  
bool c8 = a != 10;  
Console.WriteLine(c5);  
Console.WriteLine(c6);  
Console.WriteLine(c7);  
Console.WriteLine(c8);
```

DS2

Operadores

Operadores lógicos

Operador	Significado
&&	E
 	OU
!	NÃO

C1	C2	C1 E C2
F	F	F
F	V	F
V	F	F
V	V	V

C1	C2	C1 OU C2
F	F	F
F	V	V
V	F	V
V	V	V

NOTAS:

1) Precedência: ! > && > ||

2) Pode-se usar parêntesis à vontade

3) Exemplos:

2 > 3 || 4 != 5 → Resultado: true

!(2>3) && 4 != 5 → Resultado: true

```
bool c1 = 2 > 3 || 4 != 5; // true
bool c2 = !(2 > 3) && 4 != 5; // true
Console.WriteLine(c1);
Console.WriteLine(c2);
```

```
Console.WriteLine("-----");
```

```
bool c3 = 10 < 5; // false
bool c4 = c1 || c2 && c3; // true
Console.WriteLine(c3);
Console.WriteLine(c4);
```

DS2

Estruturas Condicionais

Estrutura condicional

Simple

```
if ( condição ) {  
    comando 1  
    comando 2  
}
```

Composta

```
if ( condição ) {  
    comando 1  
    comando 2  
}  
else {  
    comando 3  
    comando 4  
}
```

Encadeamentos

```
if ( condição 1 ) {  
    comando 1  
    comando 2  
}  
else if ( condição 2 ) {  
    comando 3  
    comando 4  
}  
else if ( condição 3 ) {  
    comando 5  
    comando 6  
}  
else {  
    comando 7  
    comando 8  
}
```

Nota: se o bloco de comandos possuir apenas um comando, as chaves são opcionais.

DS2

Estruturas Condicionais

Entre com um número inteiro:

10

Par!

Entre com um número ímpar?

15

Ímpar!

```
using System;

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

            Console.WriteLine("Entre com um número inteiro:");
            int x = int.Parse(Console.ReadLine());

            if (x % 2 == 0) {
                Console.WriteLine("Par!");
            }
            else {
                Console.WriteLine("Ímpar");
            }
        }
    }
}
```

DS2

Estruturas Condicionais

Qual a hora atual?
10
Bom dia!

hora < 12

Qual a hora atual?
14
Boa tarde!

12 <= hora < 18

Qual a hora atual?
19
Boa noite!

hora >= 18

```
using System;

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

            Console.WriteLine("Qual a hora atual?");
            int hora = int.Parse(Console.ReadLine());

            if (hora < 12) {
                Console.WriteLine("Bom dia!");
            }
            else if (hora < 18) {
                Console.WriteLine("Boa tarde!");
            }
            else {
                Console.WriteLine("Boa noite!");
            }
        }
    }
}
```

DS2

Estruturas Condicionais

switch-case

Estrutura opcional a vários if-else encadeados, **quando a condição envolve o teste do valor de uma variável.**

Sintaxe:

```
var minhaVariavel = (...);  
  
switch (minhaVariavel) {  
    case 1:  
        Console.WriteLine("Caso 1");  
        break;  
    case 2:  
        Console.WriteLine("Caso 2");  
        break;  
    default:  
        Console.WriteLine("Caso padrão");  
        break;  
}
```

Estruturas Condicionais

```
int x = int.Parse(Console.ReadLine());
string day;

if (x == 1) {
    day = "Sunday";
}
else if (x == 2) {
    day = "Monday";
}
else if (x == 3) {
    day = "Tuesday";
}
else if (x == 4) {
    day = "Wednesday";
}
else if (x == 5) {
    day = "Thursday";
}
else if (x == 6) {
    day = "Friday";
}
else if (x == 7) {
    day = "Saturday";
}
else {
    day = "Invalid value";
}

Console.WriteLine("Day: " + day);
```

```
int x = int.Parse(Console.ReadLine());
string day;

switch (x) {
    case 1:
        day = "Sunday";
        break;
    case 2:
        day = "Monday";
        break;
    case 3:
        day = "Tuesday";
        break;
    case 4:
        day = "Wednesday";
        break;
    case 5:
        day = "Thursday";
        break;
    case 6:
        day = "Friday";
        break;
    case 7:
        day = "Saturday";
        break;
    default:
        day = "Invalid value";
        break;
}

Console.WriteLine("Day: " + day);
```


DS2

Estruturas Condicionais

Expressão condicional ternária

Estrutura opcional ao if-else quando se deseja decidir um **VALOR** com base em uma condição.

Sintaxe:

```
( condição ) ? valor_se_verdadeiro : valor_se_falso
```

Exemplos:

```
( 2 > 4 ) ? 50 : 80       80
```

```
( 10 != 3 ) ? "Maria" : "Alex"       "Maria"
```

Estruturas Condicionais

```
double preco = 34.5;
double desconto;
if (preco < 20.0) {
    desconto = preco * 0.1;
}
else {
    desconto = preco * 0.05;
}
```

```
double preco = 34.5;
double desconto = (preco < 20.0) ? preco * 0.1 : preco * 0.05;
```

DS2

Exercícios

Fazer um programa para ler um número inteiro, e depois dizer se este número é negativo ou não.

Exemplos:

Entrada:	Saída:
-10	NEGATIVO
Entrada:	Saída:
8	NAO NEGATIVO
Entrada:	Saída:
0	NAO NEGATIVO

DS2

Exercícios

Fazer um programa para ler um número inteiro e dizer se este número é par ou ímpar.

Exemplos:

Entrada:	Saída:
12	PAR

Entrada:	Saída:
-27	IMPAR

Entrada:	Saída:
0	PAR

DS2

Exercícios

Leia 2 valores inteiros (A e B). Após, o programa deve mostrar uma mensagem "Sao Multiplos" ou "Nao sao Multiplos", indicando se os valores lidos são múltiplos entre si. Atenção: os números devem poder ser digitados em ordem crescente ou decrescente.

Exemplos:

Entrada:	Saída:
6 24	Sao Multiplos
Entrada:	Saída:
6 25	Nao sao Multiplos
Entrada:	Saída:
24 6	Sao Multiplos

DS2

Exercícios

Com base na tabela abaixo, escreva um programa que leia o código de um item e a quantidade deste item. A seguir, calcule e mostre o valor da conta a pagar.

CODIGO	ESPECIFICAÇÃO	PREÇO
1	Cachorro Quente	R\$ 4.00
2	X-Salada	R\$ 4.50
3	X-Bacon	R\$ 5.00
4	Torrada simples	R\$ 2.00
5	Refrigerante	R\$ 1.50

Exemplos:

Entrada:	Saída:
3 2	Total: R\$ 10.00
Entrada:	Saída:
2 3	Total: R\$ 13.50

DS2

Exercícios

Em um país imaginário denominado Lisarb, todos os habitantes ficam felizes em pagar seus impostos, pois sabem que nele não existem políticos corruptos e os recursos arrecadados são utilizados em benefício da população, sem qualquer desvio. A moeda deste país é o Rombus, cujo símbolo é o R\$.

Leia um valor com duas casas decimais, equivalente ao salário de uma pessoa de Lisarb. Em seguida, calcule e mostre o valor que esta pessoa deve pagar de Imposto de Renda, segundo a tabela abaixo.

Renda	Imposto de Renda
de 0.00 a R\$ 2000.00	Isento
de R\$ 2000.01 até R\$ 3000.00	8 %
de R\$ 3000.01 até R\$ 4500.00	18 %
acima de R\$ 4500.00	28 %

Lembre que, se o salário for R\$ 3002.00, a taxa que incide é de 8% apenas sobre R\$ 1000.00, pois a faixa de salário que fica de R\$ 0.00 até R\$ 2000.00 é isenta de Imposto de Renda. No exemplo fornecido (abaixo), a taxa é de 8% sobre R\$ 1000.00 + 18% sobre R\$ 2.00, o que resulta em R\$ 80.36 no total. O valor deve ser impresso com duas casas decimais.

Exemplos:

Entrada:	Saída:
3002.00	R\$ 80.36
Entrada:	Saída:
1701.12	Isento
Entrada:	Saída:
4520.00	R\$ 355.60

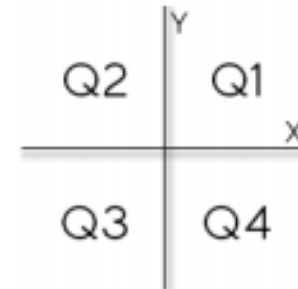
DS2

Exercícios

Leia 2 valores com uma casa decimal (x e y), que devem representar as coordenadas de um ponto em um plano. A seguir, determine qual o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos ou na origem ($x = y = 0$).

Se o ponto estiver na origem, escreva a mensagem "Origem".

Se o ponto estiver sobre um dos eixos escreva "Eixo X" ou "Eixo Y", conforme for a situação.



Exemplos:

Entrada:	Saída:
4.5 -2.2	Q4
Entrada:	Saída:
0.1 0.1	Q1
Entrada:	Saída:
0.0 0.0	Origem