FICHA 1

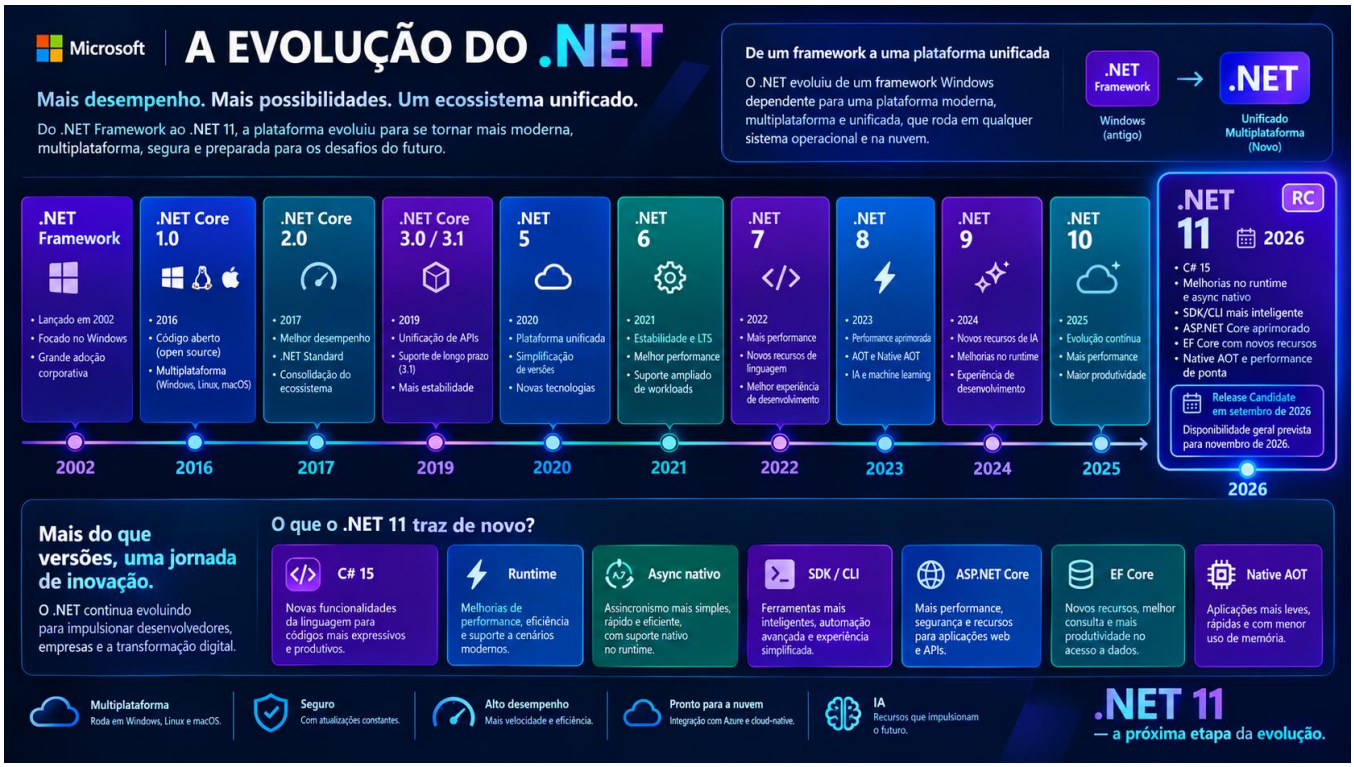
INTRODUÇÃO AO C#

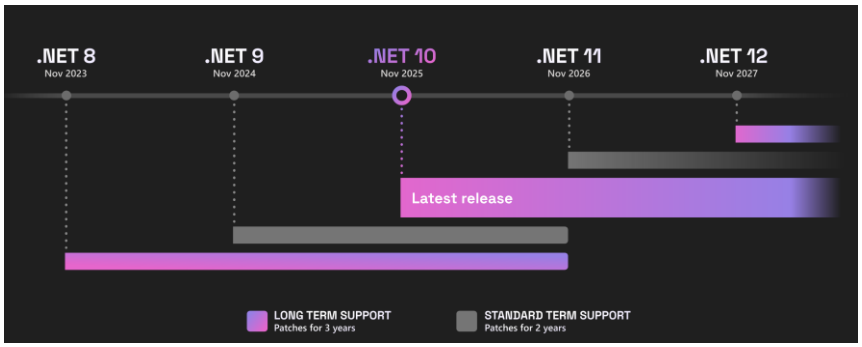
MODALIDADE:	Aprendizagem +	Escolha um item.	
CURSO:	Tecnologias de Programação de Sistemas de Informação		
UC:	Desenvolver programas complexos em linguagem de programação estruturada	CÓDIGO UC:	UC00607
FORMADOR/A:	Bruno Silva	DATA:	

OBJETIVOS

- Introdução à linguagem C# - variáveis, escrita de mensagens, leitura de dados do teclado

Evolução .NET





Version	Original release date	Latest patch version	Patch release date	Release type	Support phase	End of support
.NET 10	November 11, 2025	10.0.12	September 8, 2026	LTS	Active	November 14, 2028
.NET 9	November 12, 2024	9.0.20	September 8, 2026	STS	Maintenance	November 10, 2026
.NET 8	November 14, 2023	8.0.31	September 8, 2026	LTS	Maintenance	November 10, 2026

1 - Instruções de escrita de valores na consola

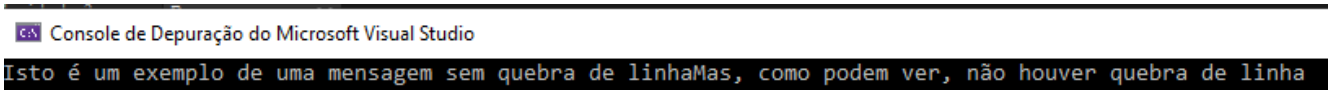
Instruções de escrita (Console.Write e Console.WriteLine)

As instruções de escrita correspondem, na linguagem C#, aos métodos da classe Console (que representa o ecrã e o teclado): Console.Write ou Console.WriteLine.

Como tal, vamos verificar **2 exemplos/formas de escrita de mensagens na consola**:

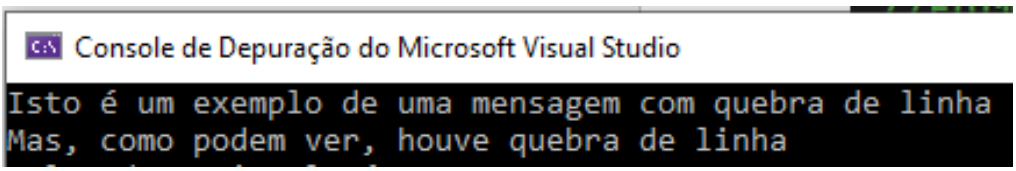
Forma 1 - Escrita de mensagens sem quebra de linha (Console.Write)

```
//Forma 1 - Escrita de mensagens (sem quebra de linha))
Console.Write("Isto é um exemplo de uma mensagem sem quebra de linha");
Console.Write("Mas, como podem ver, não houver quebra de linha");
```



Forma 2 - Escrita de mensagens com quebra de linha (Console.WriteLine)

```
//Forma 2 - Escrita de mensagens (com quebra de linha)
Console.WriteLine("Isto é um exemplo de uma mensagem com quebra de linha");
Console.WriteLine("Mas, como podem ver, houve quebra de linha");
```



2 - Tipos de Dados e Variáveis

2.1 - Tipos de Dados

Os tipos de dados são representações do valor do qual queremos trabalhar. Vamos ver os exemplos de tipos de dados mais usados:



Uma variável representa a área de memória que contém um valor dum determinado tipo de dados. A cada variável estão associados ***um tipo de dados*** e ***um nome***.

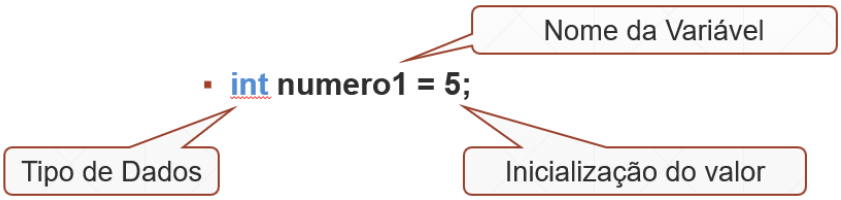
Para associar um determinado tipo de dados a uma dada variável, é necessário proceder-se à declaração da variável.

2.2 - Variáveis

Uma variável, deve ser declarada antes de ser utilizada, podendo ser inicializada imediatamente com um valor.

Escreve-se o ***tipo de dados*** da **variável**, seguido do nome para identificação e a inicialização da mesma.

Exemplo: Quero criar u**ma variável do tipo inteiro** e para **guardar números inteiros** e **iniciar** com o **número 5**.



Exemplo de criação de variáveis:

```
//declarar uma variável do tipo inteiro
int num1 = 5;

//declarar uma variável do tipo decimal
float num2 = 2.3; //permite até 8 casas decimais
double num3 = 5.11; //permite até 16 casas decimais

//declarar uma variável do tipo caracter (utilize aspa simples para colocar o caracter)
string texto = 'a'

//declarar uma variável do tipo textual (utilize aspas para colocar o texto)
string texto = "Olá Mundo"

//também pode declarar múltiplas variáveis ao mesmo tempo (agrupados pelo mesmo tipo de dados)
int numero_1 = 0, numero_2 = 5;
double nota1 = 0, nota2 =0, nota3 = 0;
string texto_1 = "Exemplo", texto_2 = "texto";
```

Mas até agora apenas declaramos variáveis. Como podemos exibir as variáveis com o método de escrita de mensagens (Console.Write() ou Consol.WriteLine()) ?

Escrita de mensagens (com a inclusão das variáveis)

Forma 1 - Escrita de mensagens (com $ e chavetas {})

Nesta última forma, como queremos indicar a combinação de texto com variáveis, temos que, escrever o **símbolo $** atrás das aspas **para indicar que vamos combinar a escrita de texto** (que fica com a cor laranja) **com variáveis** que devem ficar **dentro de chavetas curvas { }**:

```
//Forma 3 - Escrita de mensagens (com variáveis incluídas)
int numero = 5;
Console.WriteLine($"Valor da variavel número: {numero}");
```

```
Console de Depuração do Microsoft Visual Studio
Valor da variavel número: 5
```

Outro exemplo (mostrar/combinar mais do que uma variável ao mesmo tempo):

```
string primeiro_nome = "Maria";
string ultimo_nome = "Sage";
Console.WriteLine($"Primeiro nome é: {primeiro_nome} e o último é: {ultimo_nome}");
```

Forma 2 - Escrita de mensagens (por concatenações – símbolo +)

Também pode usar **outra forma de escrita** de mensagens (como feito na linguagem Java e entre outras), que passa por **colocar concatenações com o símbolo +** (mais):

```
int idade = 25;
string nome = "Rita";

//mensagem por concatenação
Console.WriteLine("O meu nome é " + nome + " e tenho " + idade + " anos");
```

Resultado:

```
Console de Depuração do Microsoft Visual Studio
O meu nome é Rita e tenho 25 anos
```

Exemplo da combinação de várias variáveis com o método Console.WriteLine()

```
//Variáveis: são os contentores para guardar informação

//Temos de indicar qual o tipo do valor e dar um nome a variável
//Exemplos:

//variável para guradar números inteiros (int)
int numero = 5;
Console.WriteLine($"Valor da variavel número: {numero}");

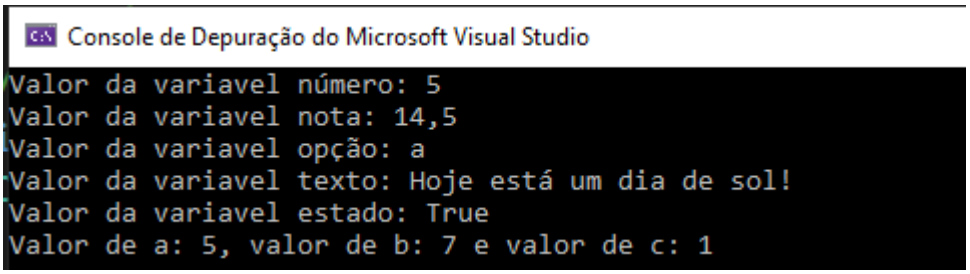
//variável para guardar números decimais (double)
double nota = 14.5;
Console.WriteLine($"Valor da variavel nota: {nota}");

//variável para guardar apenas um caracter (char)
char opcao = 'a';
Console.WriteLine($"Valor da variavel opção: {opcao}");

//variável para guardar uma cadeia de caracteres (string)
string texto = "Hoje está um dia de sol!";
Console.WriteLine($"Valor da variavel texto: {texto}");

//variável do tipo booleano (apenas 1 de dois estados: verdadeiro ou falso)
bool estado = true;
Console.WriteLine($"Valor da variavel estado: {estado}");

//exemplo de declaração de múltiplas variaveis com o mesmo tipo de dados
int a = 5, b = 7, c = 1; //cada variavel o respetivo valor
Console.WriteLine($"Valor de a: {a}, valor de b: {b} e valor de c: {c}");
```




.NET 10
C# .NET 10 – Variáveis, Tipos de Variáveis e Exibição na Consola (sem Main)
Neste exemplo, usamos o C# 10 com a abordagem de instruções de nível superior (top-level statements), ou seja, não há método Main.
.NET 10
Program.cs
// Exemplo de C# .NET 10 - Variáveis, Tipos e Exibição na Consola
// (sem método Main - usando instruções de nível superior)
using System;
// Declaração de variáveis com diferentes tipos
string nome = "João Silva"; // string (texto)
int idade = 28; // int (número inteiro)
double salario = 3500.75; // double (número decimal)
bool ativo = true; // bool (verdadeiro/falso)
char inicial = 'J'; // char (caractere único)
decimal altura = 1.75m; // decimal (alta precisão)
DateTime dataNascimento = new DateTime(1997, 5, 14); // DateTime (data e hora)
object qualquerCoisa = "Exemplo de object"; // object (tipo genérico)
// Exibição dos valores no console
Console.WriteLine($"Nome: {nome}");
Console.WriteLine($"Idade: {idade}");
Console.WriteLine($"Salário: {salario:C}");
Console.WriteLine($"Ativo: {ativo}");
Console.WriteLine($"Inicial: {inicial}");
Console.WriteLine($"Altura: {altura} m");
Console.WriteLine($"Data de Nascimento: {dataNascimento:dd/MM/yyyy}");
Console.WriteLine($"Objeto: {qualquerCoisa}");
Console.WriteLine("\nPressione qualquer tecla para sair...");
Console.ReadKey();
Console (saída)
Nome: João Silva
Idade: 28
Salário: R$ 3.500,75
Ativo: True
Inicial: J
Altura: 1,75 m
Data de Nascimento: 14/05/1997
Objeto: Exemplo de object
Pressione qualquer tecla para sair...
Por que não há Main?
No C# 10, é possível usar instruções de nível superior (top-level statements). Assim, o compilador gera o método Main automaticamente, tornando o código mais simples e direto.

3 - Instruções de leitura de valores (a partir do teclado)

Instruções leitura de valores (Console.ReadLine())

As instruções de escrita correspondem, na linguagem C#, aos métodos da classe **Console.ReadLine()** (ler valores através do teclado).

Exemplo 1 – Ler um valor do teclado com a função Console.ReadLine()

```
//Receber um valor Textual
Console.WriteLine("Insira um valor textual: ");
valor_texto = Console.ReadLine();
```

Mas, **quando lemos um valor do teclado, este vem sempre no formato textual**.

Quando é pretendido **ler um dado do tipo numérico** (inteiro ou decimal) em C#, devemos sempre **converter o que recebemos do teclado** de texto para numérico.

Como tal, vão verificar **2 formas de conversão** de texto para numérico:

- **Conversão para inteiros:** Convert.ToInt32()

```
//Ler dados do teclado
int valor_inteiro = 0;

//Receber valor inteiro
Console.WriteLine("Insira um valor inteiro: ");
valor_inteiro = Convert.ToInt32(Console.ReadLine());
```

- **Conversão para decimais:** Convert.ToDouble()

```
//Ler dados do teclado
double valor_decimal = 0;

//Receber valor decimal
Console.WriteLine("Insira um valor decimal: ");
valor_decimal = Convert.ToDouble(Console.ReadLine());
```

4 - Variáveis VS Constantes

Como verificamos anteriormente, uma variável permite receber valores e, posteriormente, podemos alterar à medida que trabalhamos no programa, como podemos ver no seguinte exemplo:

```
//declarar uma variável inteira
int valor1 = 10;

Console.WriteLine($"Valor: {valor1}");

//alterar o valor da variável
valor1 = 20;

Console.WriteLine($"Valor: {valor1}");
```

```
Console de Depuração do Mic
Valor: 10
Valor: 20

C:\Users\twsk7\source\repos\C
ódigo 0 (0x0).
Para fechar o console automat
console automaticamente quand
Pressione qualquer tecla para
```

Até agora, está a fazer o que era pretendido.

No entanto, **pode haver necessidade** de fazer com que a variável inicie o valor na declaração da mesma**, mas não deixar alterar o valor no decorrer do programa**.

É aqui que entram as constantes. Uma **constante** recebe um valor no início da declaração e não deixa alterar esse valor posteriormente, ou seja, apenas podem ler o valor declarado no início.

Para declarar uma variável do tipo constante, basta colocar a palavra reservada **const** **antes** da declaração da variável. Exemplo:

```
//declarar uma variável constante inteira
const int valor1 = 10;

Console.WriteLine($"Valor: {valor1}");

//alterar o valor da variável
valor1 = 20;

Console.WriteLine($"Valor: {valor1}");
```

Como podem ver, quando tentam alterar o valor1 para outro valor, este não vai permitir e dar um erro vermelho, pois **uma variável do tipo constante só permite ler o valor que foi estabelecido no início**.

Exercícios

Exercício 1 – Construa um programa para escrever um menu de opções, mais especificamente:

1 – Inserir Formando
2 – Alterar Formando
3 – Eliminar Formando
4 – Sair

Exercício 2 – Construa um programa para declarar 3 tipos de variáveis: um inteiro, um decimal e um texto. De seguida, use a função de escrita de mensagens para mostrar os dados das variáveis (combinando texto com o valor depositado nas variáveis).

Exercício 3 – Construa um programa, do qual, crie duas variáveis inteiras (com valores à sua descrição). De seguida, faça cálculos aritméticos com os dois valores, mais especificamente, a soma (+), subtração (-), multiplicação (*) e a divisão (/).

Exercício 4 – Construa um programa para pedir 3 números inteiros do teclado, e de seguida, mostre o resultado da soma;

Exercício 5 – Construa um programa com 3 variáveis decimais (com valores à sua descrição). De seguida, mostre a média dos números;

Exercício 6 – Construa um programa que peça o nome do utilizador e no final devolva a mensagem "Bem-vindo ...", do qual, as reticências deve ter o nome digitado do teclado com a função para ler dados;

Exercício 7 – Construa um programa para calcular a área de um triângulo, do qual, deve pedir os valores da base e da altura ao utilizador (com números decimais) e mostre o resultado da área;

Exercício 8 – Construa um programa para calcular o preço de um terreno, do qual, terá de perguntar ao utilizador qual o comprimento, a largura e o preço por metro quadrado (m2). No final mostre o resultado.

Exercício 9 – Construa um programa para calcular os descontos de um trabalhador, do qual deve pedir ao utilizador qual o ordenado bruto (sem descontos). Considere que o desconto da segurança social é 11% e finanças é 13%.

Como tal, deve criar duas variáveis constantes para declarar e posteriormente ler os valores dos descontos: uma para a Segurança Social com o nome ss com o valor 0.11 e uma para a Finanças com o nome irs com o valor 0.13.

De seguida, deve mostrar quanto desconta de SS e IRS e mostrar o salário líquido (com todos os descontos).