

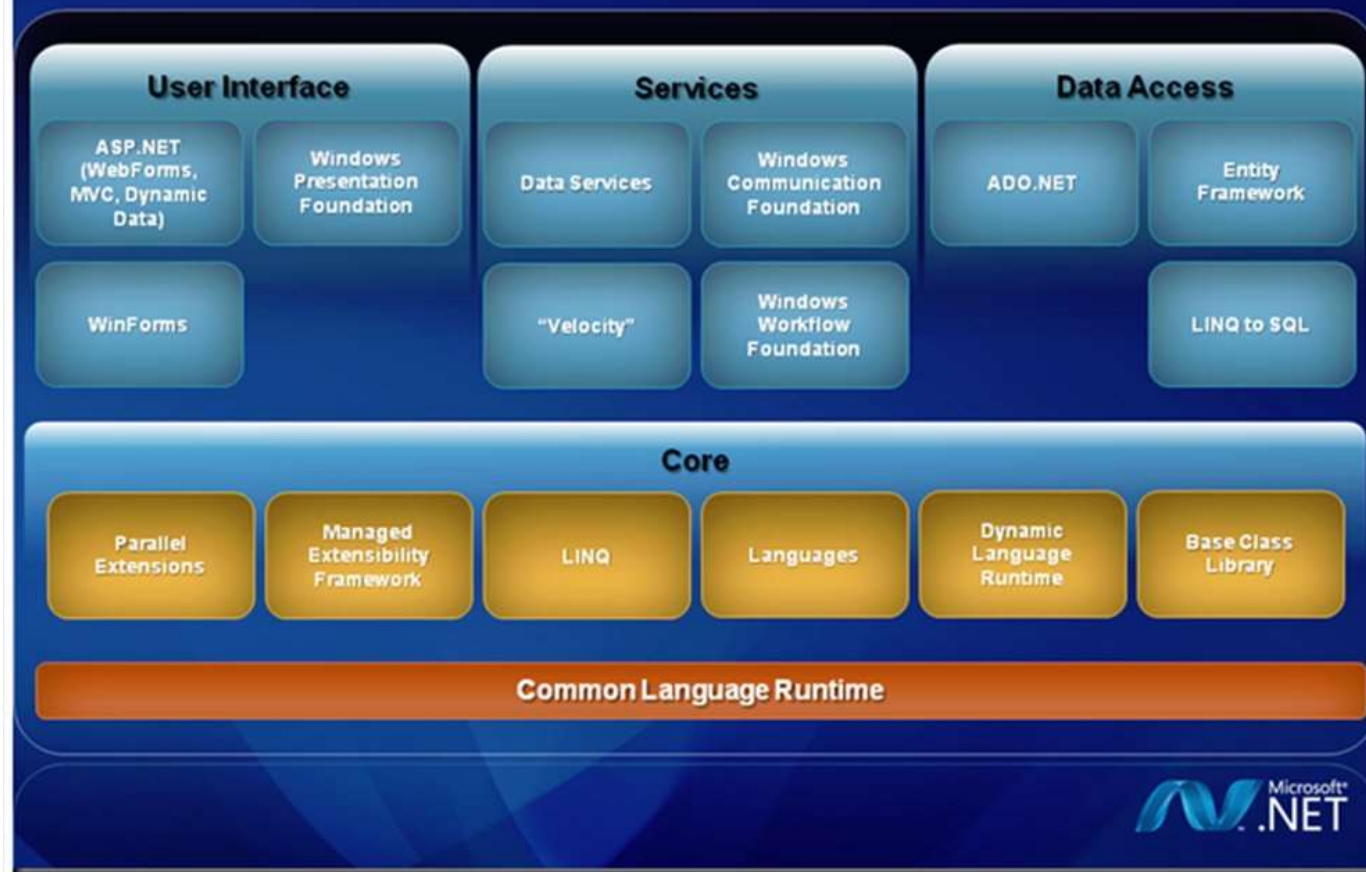
# Introdução ao .NET Framework

.NET Framework é um ambiente de execução gerenciado para Windows que fornece uma variedade de serviços para seus aplicativos em execução.

- Gerenciamento de memória.
- Uma biblioteca de classes abrangente.
- Estruturas e tecnologias de desenvolvimento.
- Multiplataforma.



# .NET Framework 4.0



# Tipos de dados

Dados é a menor informação fornecida pelo usuário, ou processo computacional.

Informação é a matéria-prima que faz com que seja necessária a existência dos computadores, pois eles são capazes de manipular e armazenar um grande volume de dados

# Principais tipos de dados

- Inteiro: toda e qualquer informação numérica que pertença ao conjunto dos números inteiros relativos (negativa, nula ou positiva).
- Real: toda e qualquer informação numérica que pertença ao conjunto dos números reais (negativa, nula ou positiva).
- Caractere: toda e qualquer informação composta por um conjunto de caracteres alfanuméricos .
- Lógico: toda e qualquer informação que pode apenas assumir duas situações, verdadeiro ou falso.

# Variáveis

Variáveis são locais onde armazenamos valores na memória. Toda variável é caracterizada por um nome, que a identifica em um programa, e por um tipo, que determina o que pode ser armazenado naquela variável.

Os tipos mais comuns são, inteiro, caractere, real e lógico.

# Regras

- Não ter o nome de algo já conhecido pela linguagem (palavra reservada).
- Só pode ser iniciada com uma letra ou com o *underline* (\_).
- Não são permitidos espaços, pontos ou caracteres de acentuação.
- Não podemos ter mais de uma variável com o mesmo nome.

Tipo	Valores possíveis de se armazenar
<b>bool</b>	Verdadeiro ou Falso (Valores booleandos)
<b>byte</b>	0 a 255 (8 bits)
<b>sbyte</b>	-128 a 127 (8 bits)
<b>char</b>	Um caractere (16 bits)
<b>decimal</b>	$\pm 1.0 \times 10^{-28}$ a $\pm 7.9 \times 10^{28}$ (128 bits)
<b>double</b>	$\pm 5.0 \times 10^{-324}$ a $\pm 1.7 \times 10^{308}$ (64 bits)
<b>float</b>	$\pm 1.5 \times 10^{-45}$ a $\pm 3.4 \times 10^{38}$ (32 bits)
<b>int</b>	-2,147,483,648 a 2,147,483,647 (32 bits)
<b>uint</b>	0 a 4,294,967,295 (32 bits)
<b>long</b>	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 (64 bits)
<b>ulong</b>	0 a 18,446,744,073,709,551,615 (64 bits)
<b>object</b>	Qualquer tipo.
<b>short</b>	-32,768 a 32,767 (16 bits)
<b>ushort</b>	0 a 65,535 (16 bits)
<b>string</b>	Sequência de caracteres (16 bits por caractere)

# Declaração de Variáveis

- double num,r;
- int idade,b,c;
- string nome;

## Atribuição de valores

nome = FATEC;

Idade = 40;

num = 34,5;

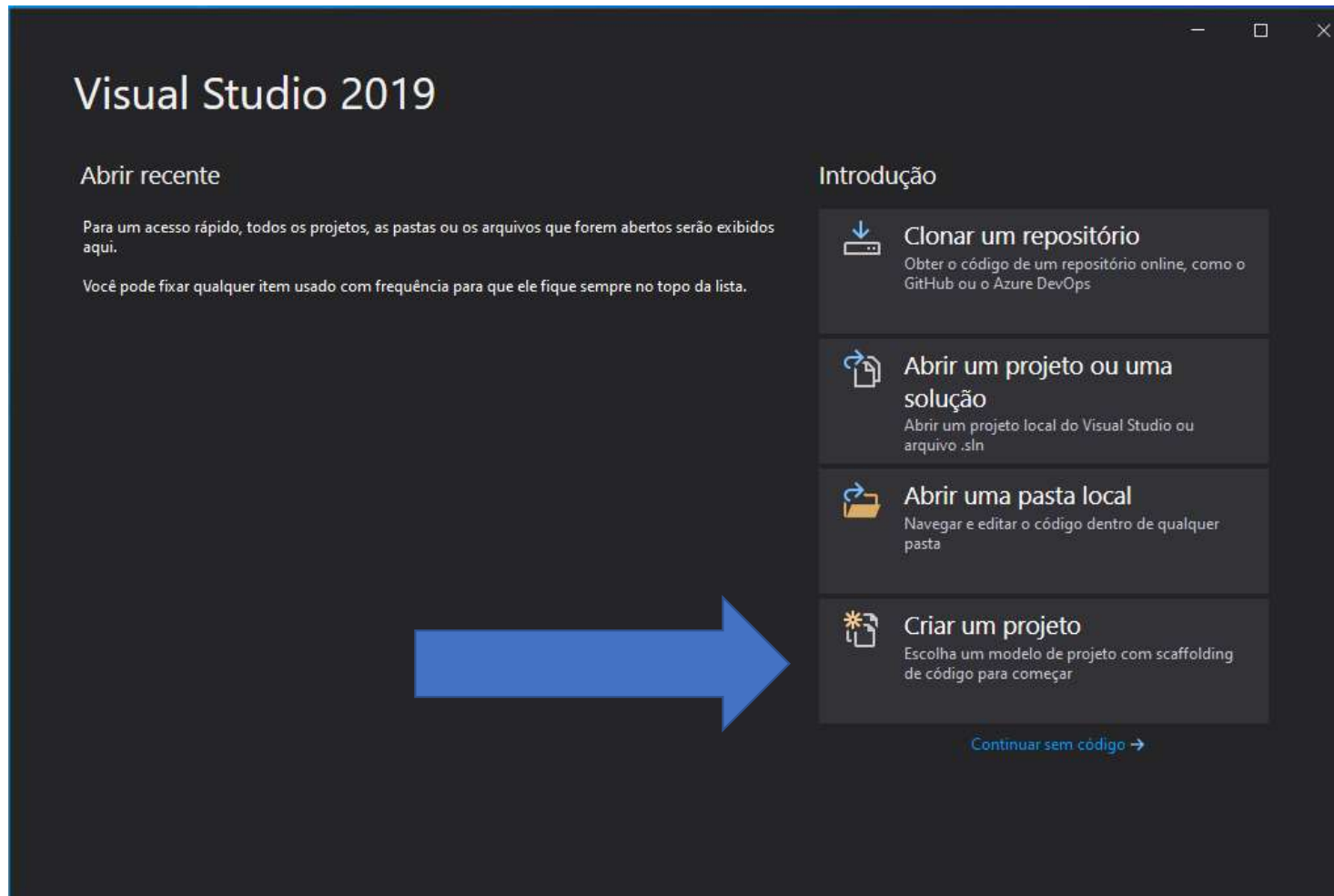


# Exemplo

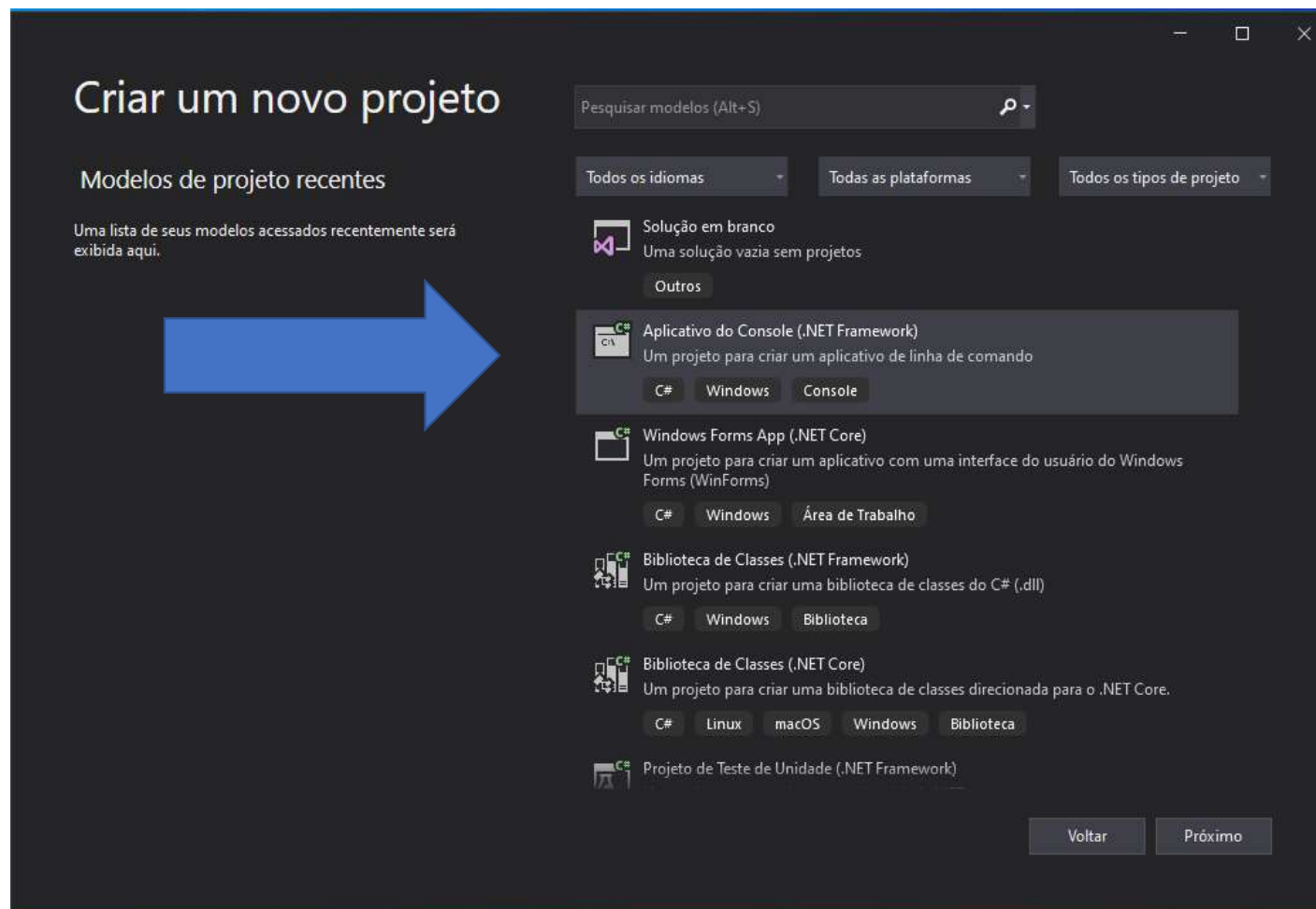
```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         double a,b,c;
8         a = 10;
9         b = 20;
10        c = a + b;
11        Console.WriteLine(c);
12    }
13 }
```

- Console.WriteLine("AQUI"); Tem a função de imprimir uma mensagem na tela de seu programa.

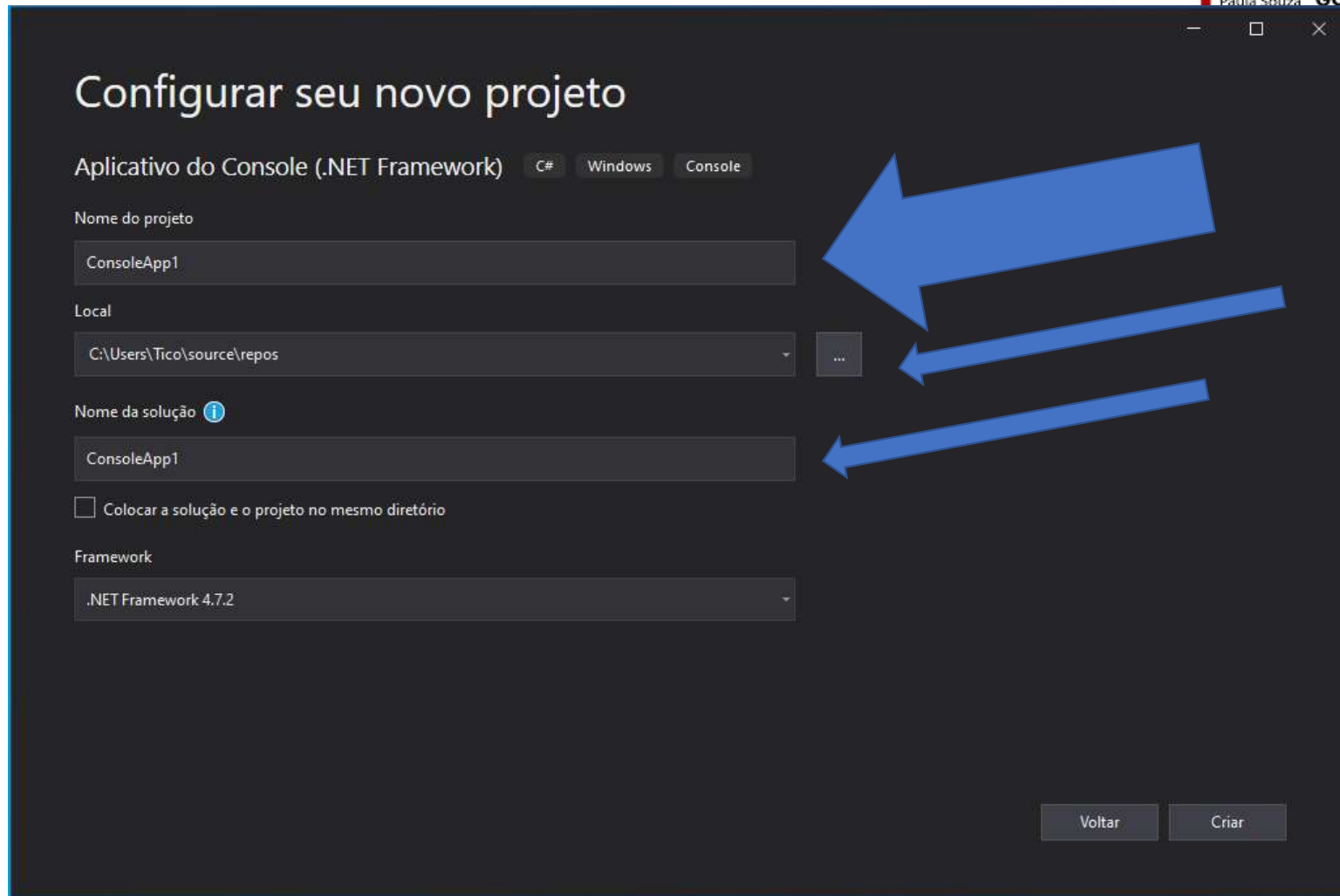
# IDE



# IDE



# IDE



Configurar seu novo projeto

Aplicativo do Console (.NET Framework) C# Windows Console

Nome do projeto

ConsoleApp1

Local

C:\Users\Tico\source\repos

Nome da solução ⓘ

ConsoleApp1

☐ Colocar a solução e o projeto no mesmo diretório

Framework

.NET Framework 4.7.2

Voltar Criar

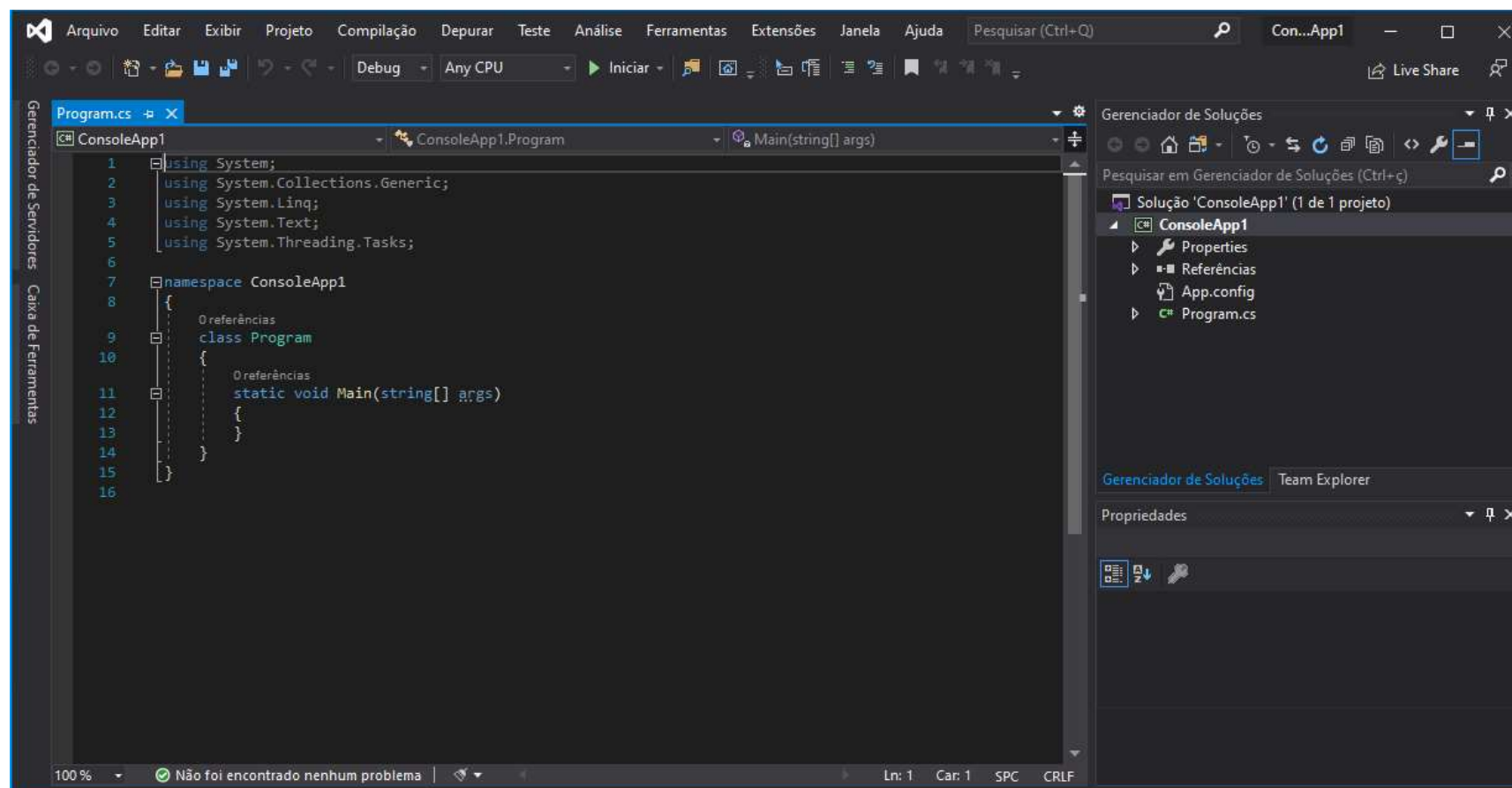
# IDE

Fatec  
São Paulo

CPs  
Centro  
Paula Souza

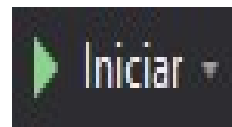


GOVERNO DO ESTADO  
DE SÃO PAULO



# IDE

- São as opções de salvar e abrir novos arquivos.
- Diz respeito ao teste, neste botão rodamos o programa que programamos para testar e operar seus resultados.



# Primeiro Programa

```
namespace primeiroprojeto
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("BEM VINDO AO PRIMEIRO PROGRAMA");
            Console.ReadKey();
        }
    }
}
```

# O que estes comandos fazem?

- *Console.WriteLine("AQUI");* Tem a função de imprimir uma mensagem na tela de seu programa.
- *Console.ReadKey();* Trava a tela, espera o usuário digitar alguma coisa antes de fechar o programa.



# Resultado

```
BEM VINDO AO PRIMEIRO PROGRAMA
```

Crie um programa que imprima na tela uma mensagem de vindo e seu nome.

# Entrada, Saída e processamento

- Todo tipo de software que construímos, ou programamos, tem em seu interior, uma divisão simples que, mesmo que não percebida, sempre esta lá. Esta divisão é:

**Entrada, processamento e saída.**

# Entrada, Saída e processamento

```
double n1, n2, r;  
Console.WriteLine("Digite o primeiro numero");  
n1 = Convert.ToDouble(Console.ReadLine());  
Console.WriteLine("Digite o segundo numero");  
n2 = Convert.ToDouble(Console.ReadLine());  
r = n1 + n2;  
Console.WriteLine("A Soma: "+r);  
Console.ReadKey();
```

# Entrada, Saída e processamento

- **Double n1,n2,r;** Nesta linha criamos as **variáveis**. Explicaremos melhor sobre elas adiante mas em um pequeno resumo são como as variáveis matemáticas, uma letra que representa seu conteúdo.
- **Console.WriteLine("Digite o primeiro numero");** Este comando já visto anteriormente simboliza a escrita na tela, escrevendo a mensagem entre as aspas duplas na tela.
- **n1 = Convert.ToDouble(Console.ReadLine());** este comando faz com que o usuário digite uma informação e esta informação será armazenada na variável 'n1'. Este comando pode ser dividido em duas partes:
  - **Convert.ToDouble()** Converte qualquer coisa entre parênteses para o tipo de variável Double(veremos mais sobre isso no próximo capítulo).
  - **Console.ReadLine()** Faz com que o usuário do programa possa digitar alguma informação através do teclado.
- As linhas 4 e 5 são iguais as linhas 2 e 3, com a diferença que o que o usuário digitar será armazenado na variável 'n2'.
- **r = n1 + n2;** Aqui é executada a operação matemática, no caso a variável 'r' ira conter a soma das variáveis 'n1' e 'n2'.
- No caso das linhas 7 e 8 temos a execução de comandos que já conhecemos, um para exibir a resposta e outro para travar a tela ate o usuário digitar uma tecla qualquer.



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace projeto2
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             double n1, n2, r;
14             Console.WriteLine("Digite o primeiro numero");
15             n1 = Convert.ToDouble(Console.ReadLine());
16             Console.WriteLine("Digite o segundo numero");
17             n2 = Convert.ToDouble(Console.ReadLine());
18             r = n1 + n2;
19             Console.WriteLine("A Soma: "+r);
20             Console.ReadKey();
21         }
22     }
23 }
24
```

# Sites para desenvolver diagramas

- <https://www.gliffy.com/>
- <http://www.draw.io>

# Site para desenvolver script

- <https://dotnetfiddle.net/>
- <https://repl.it/languages/csharp>

# Prática

- 1 - Desenvolva um programa que permita a entrada do nome do usuário e exiba na tela.
- 2 – Crie um programa e fluxograma que permita ao usuário entrar com dois números. Executar as quatro operações com esses dois números (Soma, Subtração, multiplicação, divisão). Não se esquecer de mostrar as respostas na tela.
- 3 – Desenvolva um programa e fluxograma para calcular a idade de uma pessoa, entrando com o ano de nascimento e ano atual.
- 4- Elaborar um programa e fluxograma para calcular a média aritmética de três números digitados pelo usuário.



# Prática

- 5) Escreva um código para ler um valor (do teclado) e escrever (na tela) o seu antecessor.
- 6) Escreva um programa para ler as dimensões de um retângulo (base e altura), calcular e escrever a área do retângulo.
- 7) Faça um código em C# que leia a idade de uma pessoa expressa em anos, meses e dias e escreva a idade dessa pessoa expressa apenas em dias. Considerar ano com 365 dias e mês com 30 dias.
- 8) Faça um código em C# que leia três notas de um aluno, calcule e escreva a média final deste aluno. Considerar que a média é ponderada e que o peso das notas é 2, 3 e 5. Fórmula para o cálculo da média final é:

$$\text{mediafinal} = \frac{n1 * 2 + n2 * 3 + n3 * 5}{10}$$