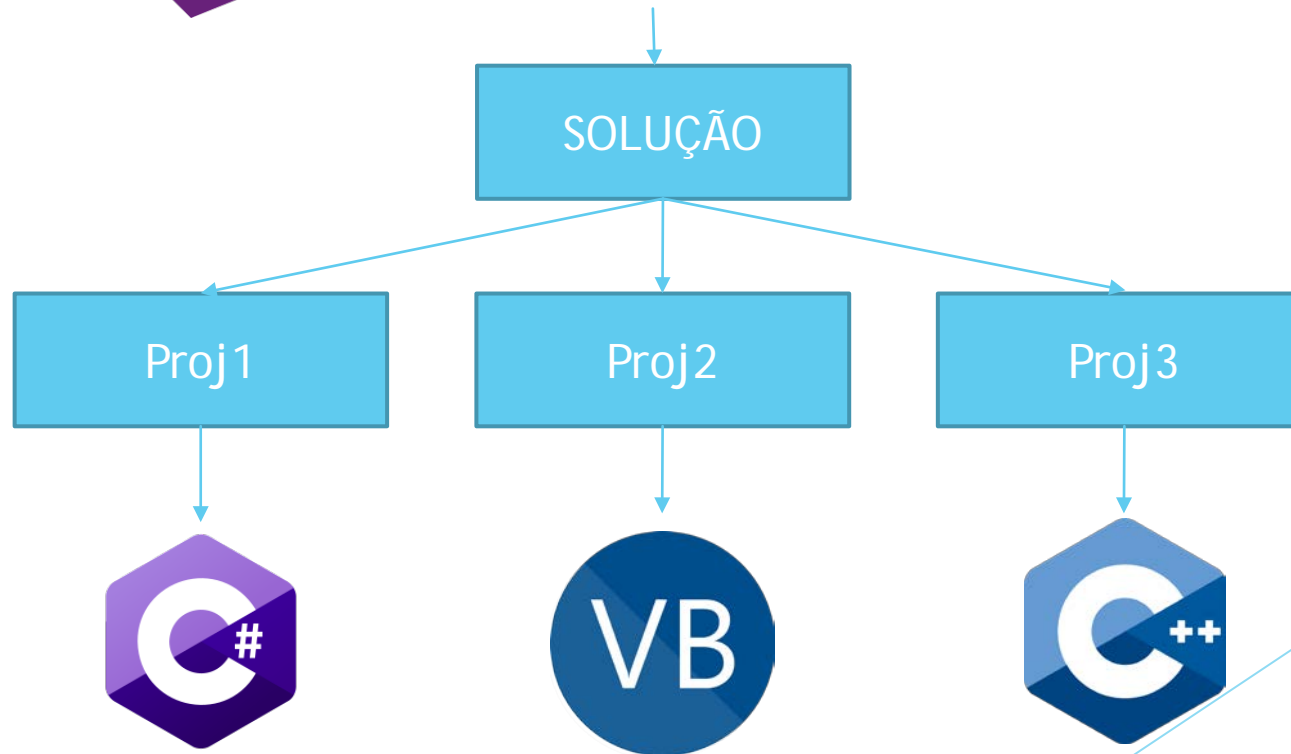


Linguagem Programação

II – Aula 02

Prof: Vinicius Drumond Gonzaga

Apresentando a IDE



Organização projeto

- ▶ Aplicação é o agrupamento de Assemblies relacionados (EXE, DLL)
- ▶ Aplicação >> Solução;
- ▶ Assembly >> Projeto

Exibindo dados (Console)

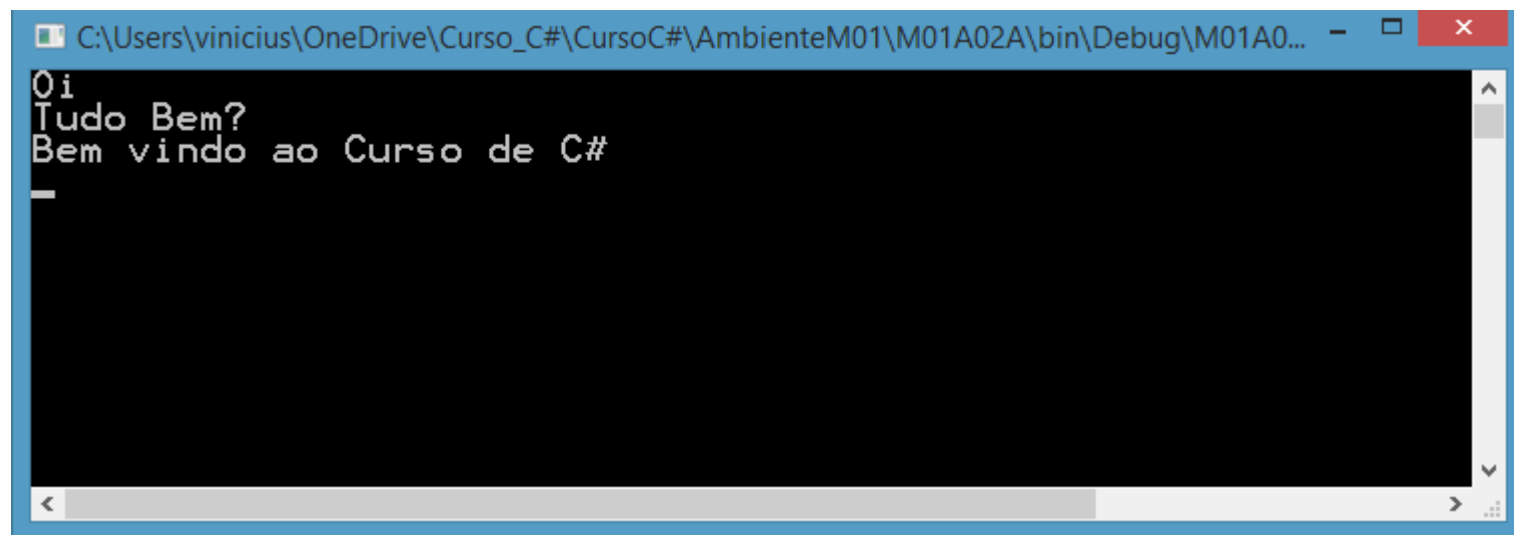
- ▶ `Console.Write();`
- ▶ `Console.WriteLine();`
- ▶ `Console.ReadKey()`



Caracteres de escape

Sequência Escape	Representação
\a	Sinal sonoro (alerta)
\b	Backspace
\n	Nova linha
\r	Carriage return
\'	Aspas simples
\"	Aspas duplas
\\	Barra invertida

Exercício 01

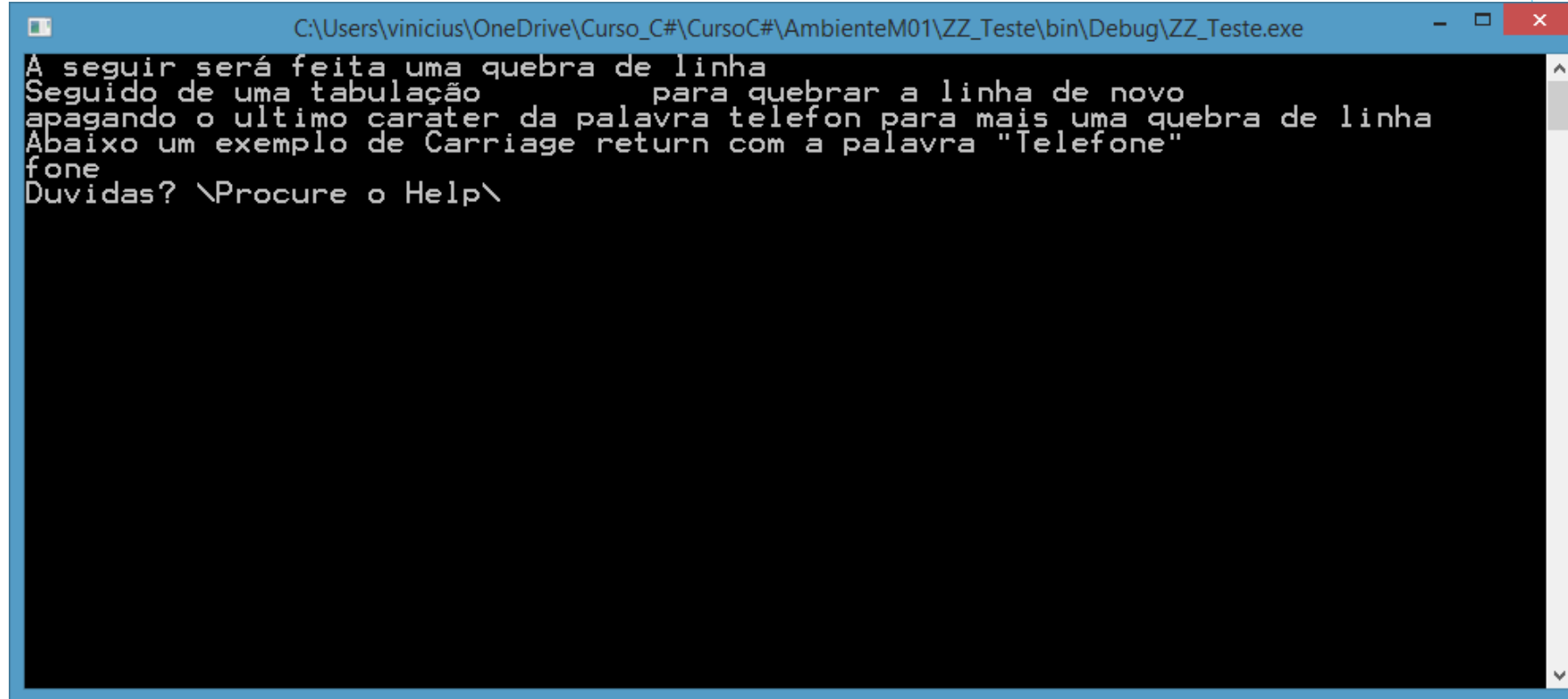


A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\vinicius\OneDrive\Curso_C#\CursoC#\AmbienteM01\M01A02A\bin\Debug\M01A0... The window contains the following text:

```
Oi  
Tudo Bem?  
Bem vindo ao Curso de C#  
_
```

The text is displayed in a monospaced font on a black background. A horizontal line is visible below the text.

Exercício 02



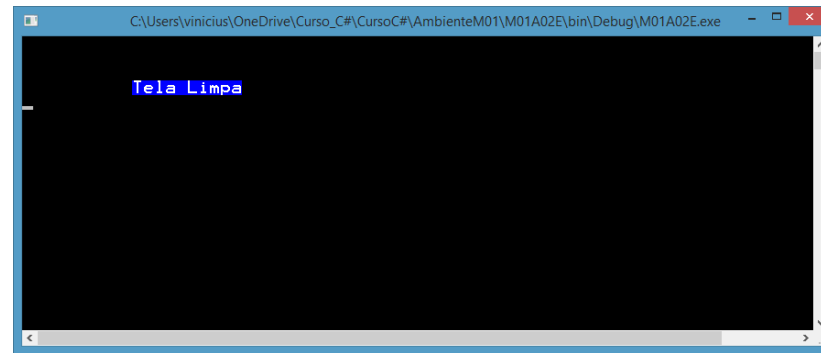
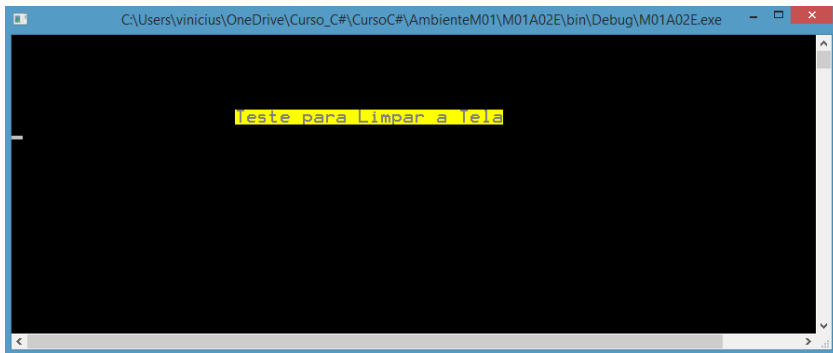
A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\vinicius\OneDrive\Curso_C#\CursoC#\AmbienteM01\ZZ_Teste\bin\Debug\ZZ_Teste.exe. The window has standard Windows window controls (minimize, maximize, close). The text inside the window is as follows:

```
A seguir será feita uma quebra de linha  
Seguido de uma tabulação      para quebrar a linha de novo  
apagando o ultimo carater da palavra telefon para mais uma quebra de linha  
Abaixo um exemplo de Carriage return com a palavra "Telefone"  
fone  
Duvidas? \Procure o Help\
```

Opções de exibição do console

- ▶ `Console.SetCursorPosition(Left, Top);`
- ▶ `Console.BackgroundColor = ConsoleColor.Cor;`
- ▶ `Console.ForegroundColor = ConsoleColor. Cor;`
- ▶ `Console.ResetColor();`

Exercício 03



Comentários em C#

- ▶ `//` uma linha;
- ▶ `/*` Multilinhas; `*/`;
- ▶ `///` Comentário XML.

Tipos Primitivos

- ▶ **byte** (1 Byte): 0 até 255
- ▶ **sbyte** (1 Byte): -128 até 127
- ▶ **short** (2 bytes): -32768 até 32767
- ▶ **ushort** (2 bytes): 0 até 65535
- ▶ **int** (4 bytes): -2,15E09 até 2,15E09
- ▶ **uint** (4 bytes): 0 até 4,29E09
- ▶ **long** (8 bytes): -9,22E18 até 9,22E18
- ▶ **ulong** (8 bytes): 0 até 1,84E19
- ▶ **char**: 1 caracter (´a´)
- ▶ **String**: cadeia de caracteres ("aa")
- ▶ **Float** (4 bytes) >> 7 dígitos de precisão: -3,4E38 até 3,4E38
- ▶ **Double** (8 bytes) >> 15 dígitos de precisão : -1,39E308 até 1,39E308
- ▶ **Decimal** (16 bytes) >>> 28 dígitos de precisão: -7,92282E28 até 7,92282E28
- ▶ **bool**: Booleano (1 Bit)

Enum

Struct

Object

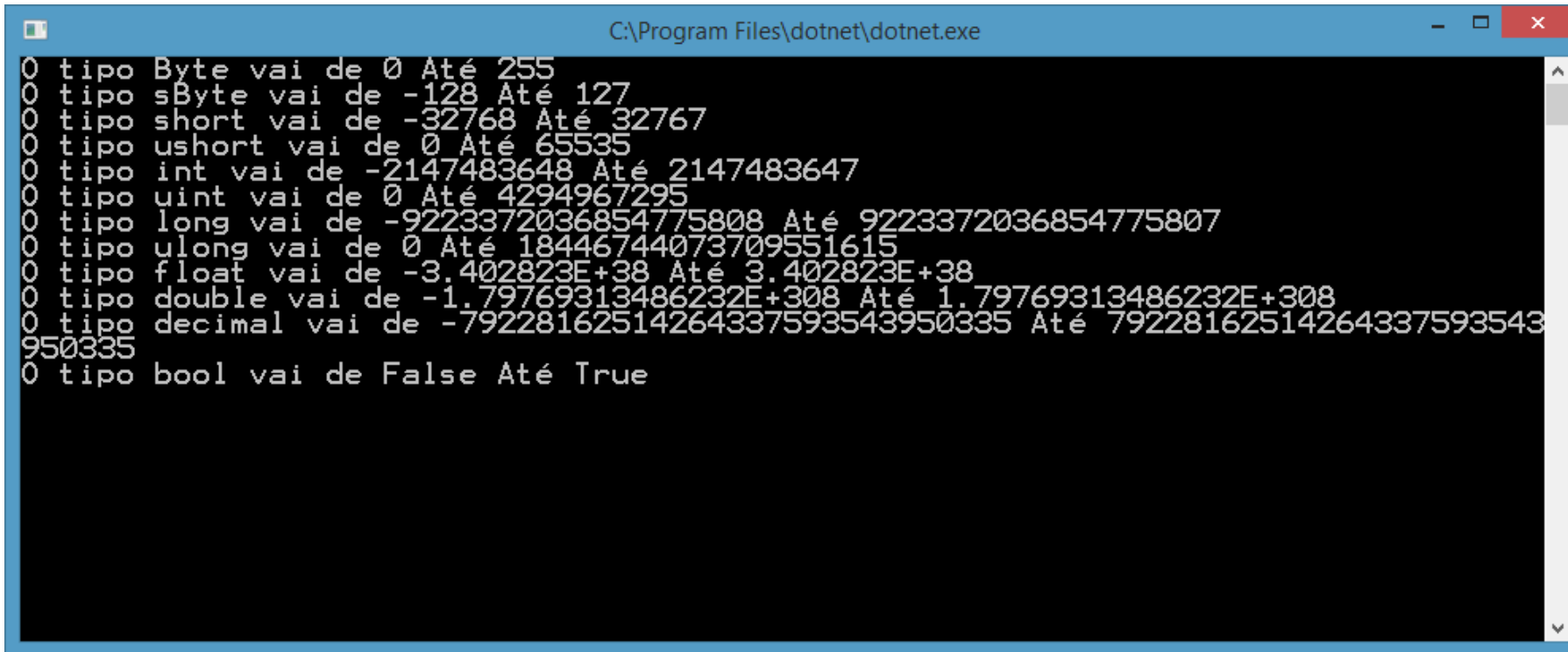
Variáveis

- ▶ Nome, Tipo, Valor;
- ▶ Declaração:
 - ▶ <tipo> NomeVariavel;
- ▶ Regras:
 - ▶ Começa com letra, sublinhado ou @;
 - ▶ Case Sensitive (Maiúsculas e minúsculas fazem diferença);
 - ▶ Só usa letras, números e sublinhados;
 - ▶ Permite acentos;
 - ▶ Não pode conter espaços;
 - ▶ Sem Símbolos;
 - ▶ Não pode palavra reservada.

Constantes

- ▶ `const <tipo> <NomeVariavel> = <Valor>`
- ▶ Letras maiúsculas (Notação);

Exercício 03



```
C:\Program Files\dotnet\dotnet.exe
0 tipo Byte vai de 0 Até 255
0 tipo sByte vai de -128 Até 127
0 tipo short vai de -32768 Até 32767
0 tipo ushort vai de 0 Até 65535
0 tipo int vai de -2147483648 Até 2147483647
0 tipo uint vai de 0 Até 4294967295
0 tipo long vai de -9223372036854775808 Até 9223372036854775807
0 tipo ulong vai de 0 Até 18446744073709551615
0 tipo float vai de -3.402823E+38 Até 3.402823E+38
0 tipo double vai de -1.79769313486232E+308 Até 1.79769313486232E+308
0 tipo decimal vai de -79228162514264337593543950335 Até 79228162514264337593543950335
0 tipo bool vai de False Até True
```

`Console.Write("O tipo Byte vai de " + byte.MinValue + " Até " + byte.MaxValue);`

Tipos de Conversão C#

- ▶ Implícita;
- ▶ Explícita (typecast);
- ▶ Classes auxiliares;

Conversão Implícita

- ▶ `int a = 10;`
- ▶ `float b = a;`

Conversão Explícita

- ▶ `float a = 4.953f;`
- ▶ `int b = a;` (ERRO!!!)
- ▶ `int b = (int)a;` (CERTO!!!)

Conversão por classes auxiliares

- ▶ `float a = 7.569f;`
- ▶ `int b = Convert.ToInt32(a);`

Entrada de dados

- ▶ Comando: `Console.ReadLine();`
 - ▶ Lê da entrada padrão até a quebra de linha;
 - ▶ Retorna os dados lidos na forma de string.



Entrada de Dados

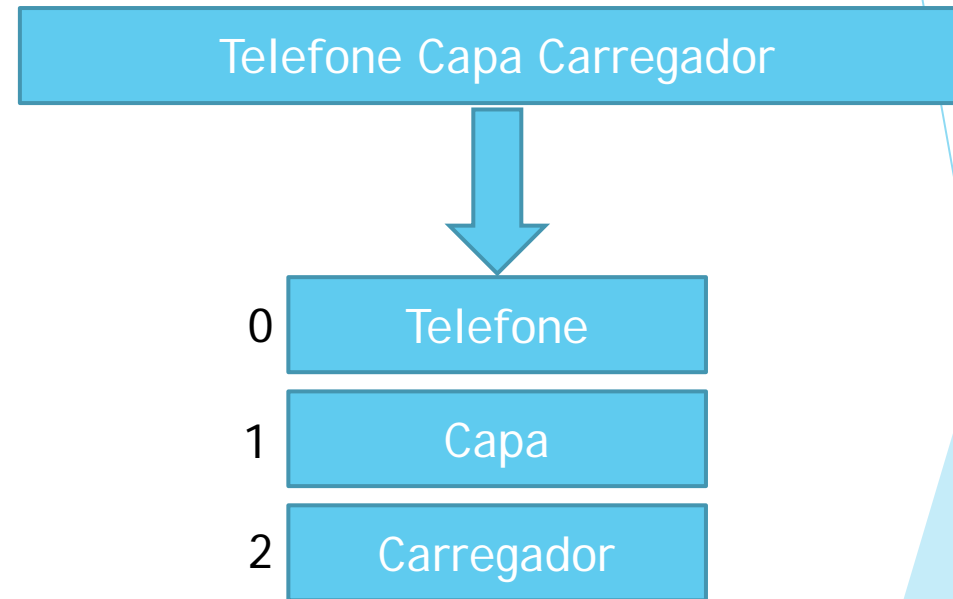
- ▶ String:
 - ▶ Nome = Console.ReadLine()
 - ▶ Ex: Alfredo
- ▶ Console.WriteLine(Nome);

Alfredo

Nome

Entrada de Dados (Split)

- ▶ `string x = Console.ReadLine();`
- ▶ `string[] x1 = x.Split(' ');`
- ▶ `"Telefone Capa Carregador";`
- ▶ `Console.WriteLine(x1[0]);`
- ▶ `Console.WriteLine(x1[1]);`
- ▶ `Console.WriteLine(x1[2]);`



Entrada de Dados

- ▶ Números

- ▶ `int num;`
- ▶ `num = int.Parse(Console.ReadLine());`
- ▶ `int num;`
- ▶ `num = Convert.ToInt32(Console.ReadLine());`

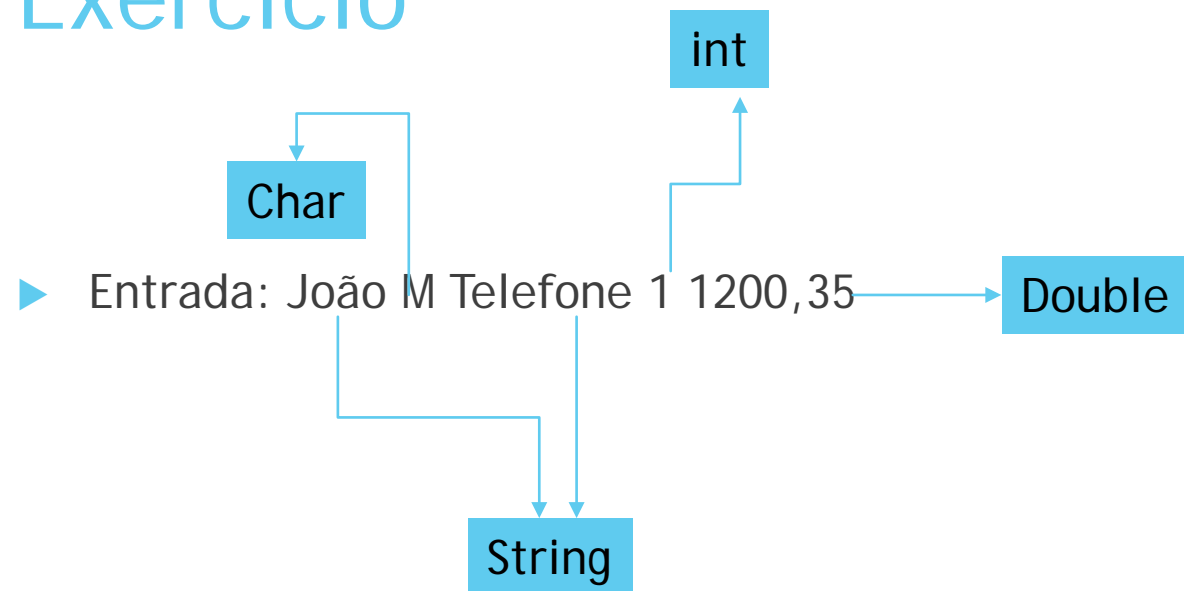
- ▶ Tentando Conversão:

- ▶ `int num = 0;`
- ▶ `Int.TryParse(Console.ReadLine(), out num)`

Entrada de Dados (Sem Configuração Regional)

- ▶ Namespace: `System.Globalization`
- ▶ `CultureInfo.InvariantCulture`
- ▶ Aplicação: `double.Parse(var, CultureInfo.InvariantCulture)`

Exercício



Nome = João

Sexo = M

Produto = Telefone

qtde = 1

Valor = 1200,25

Números Aleatórios

- ▶ Radom (Classe);
- ▶ Radom <variavel> = new Radom();
- ▶ Exemplo:
 - ▶ Radom gerador = new Radom();
 - ▶ Int n1 = gerador.next();
 - ▶ Int n2 = gerador.next(10);
 - ▶ Int n3 = gerador.next(10,50);
- ▶

Sáida de Dados e Formatação

- ▶ Placeholders;
- ▶ Concatenação;
- ▶ interpolação

Placeholders

- ▶ `int idade = 33;`
- ▶ `double saldo = 12.35784;`
- ▶ `String nome = "Henrique";`
- ▶ `Console.WriteLine("{0} tem {1} anos e tem saldo igual a {2} reais", nome, idade, saldo);`
- ▶ `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.ToString("F2", CultureInfo.InvariantCulture) + " reais");`

Concatenação

- ▶ `string nome;`
- ▶ `nome = Console.ReadLine();`
- ▶ `Console.Write("Bem vindo " + nome);`

Strings Interpoladas

- ▶ `string nome;`
- ▶ `nome = Console.ReadLine();`
- ▶ `Console.Write("Bem vindo " + nome);`
- ▶ `$` representa a interpolação de strings;
- ▶ `Console.Write($"Bem vindo {nome}");`

Formatação de Strings Interpoladas (String)

- ▶ `string nome = "Vinicius";`
- ▶ `Console.WriteLine($"Olá {nome,20}");`
 - ▶ Positivo >> Direita
 - ▶ Negativo >> Esquerda

Formatação de Strings Interpoladas (Números)

- ▶ :C >> Monetário;
 - ▶ :D >> Decimal (inteiro);
 - ▶ :N >> Número (real);
 - ▶ :E >> Científico
 - ▶ :X >> Hexadecimal
-
- ▶ `Console.WriteLine($"O valor foi de {n1:N2}")`
 - ▶ `Int n1=1;`
 - ▶ `Console.WriteLine($"O valor foi de {n1:C2}")`

Exercício

- ▶ `string produto1 = "Telefone";`
`string produto2 = "capa";`
- ▶ `byte idade = 30;`
- ▶ `int codigo = 5290;`
- ▶ `char genero = 'M';`
- ▶ `double preco1 = 4100.0;`
- ▶ `double preco2 = 150.50;`
- ▶ `double medida = 54.234567;`

```
D:\OneDrive\Cesgranrio\2021_1\LPII\Aulas\Rascunho\Rascunho4\bin\Debug\Ras
Produtos:
Telefone, cujo preço é $ 4100,00
Capa, cujo preco é $ 150,50

Registro: 44 anos de idade, código 5193 e gênero: M

Medida com oito casas decimais: 54,23456700
Arredondado (três casas decimais): 54,235
Separador decimal invariant culture: 54.235
_
```


Data e Hora

- ▶ `DateTime` (Classe);
- ▶ `int dia = DateTime.Now.Day;`
- ▶ `int mes = DateTime.Now.Month;`
- ▶ `int ano = DateTime.Now.Year;`

Temporização

- ▶ `System.Threading;`
- ▶ `Thread.Sleep(Tempo em Milissegundos);`
 - ▶ `Thread.Sleep(2000);`