

MINICURSO



An abstract network diagram consisting of numerous grey dots (nodes) connected by thin grey lines (edges). The nodes are scattered across the frame, with a higher density on the left side, creating a web-like structure that suggests connectivity and data flow.

Linguagem de Programação C#

Introdução

O que é O .NET FRAMEWORK?

- O .NET Framework é um modelo de programação de código gerenciado da Microsoft para criar aplicativos em clientes, servidores e dispositivos móveis ou incorporados ao Windows.
- Facilidade de desenvolvimento de aplicações desde as mais simples até as mais complexas;
- Orientada a objetos;
- Interoperabilidade entre plataformas e componentes desenvolvidos em outras linguagens .NET;
- Sintonizado com as últimas tecnologias;
- Tecnologia baseada em máquina virtual;

O que posso fazer com o .net?

- O .NET permite desenvolver soluções como:
 - Aplicativos Web
 - Aplicativos para Servidores
 - Aplicativos Smart Client
 - Aplicativos de Console
 - Aplicativos de Banco de Dados
 - Serviços Windows
 - Web Services e muito mais

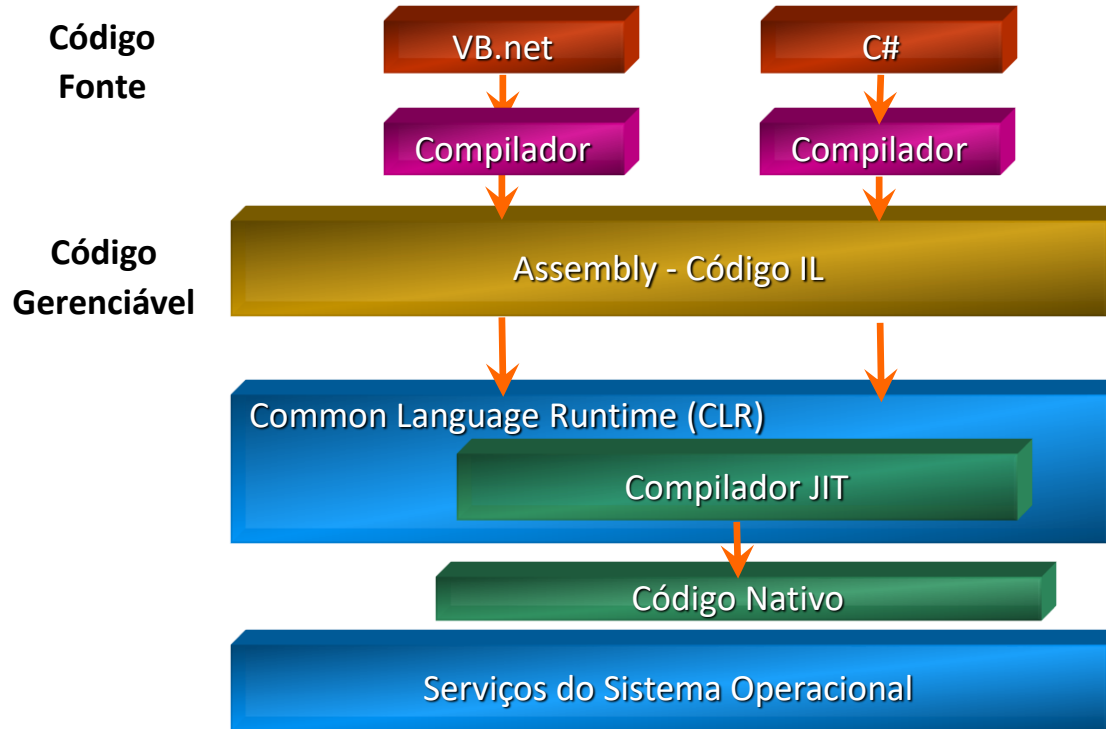
O QUE É CLR?

- Common Language Runtime (CLR) é o componente encarregado de gerenciar aplicações desenvolvidas em .NET.
- O compilador de cada linguagem segue uma série de especificações, conhecida como Common Language Infrastructure (CLI), para compilar seus códigos.

Serviços Presentes no CLR

- Serviços de gerenciamento de memória
- Serviços de tratamento de exceções
- Serviços de compilação
- Serviços de segurança, etc.

Visão Geral da Compilação



Em resumo o .NET FRAMEWORK é:

- Conjunto de bibliotecas;
- Máquina Virtual (CLR);
- Conjunto de linguagens interoperáveis:
 - C#
 - VB.NET
 - Visual C++

C# x VB.net

C#

```
static void Main(string[] args) {  
    int x;  
    x = 7;  
    Console.WriteLine(x);  
}
```

VB.net

```
Sub Main()  
    Dim x As Integer  
    x = 7  
    Console.WriteLine(x)  
End Sub
```

O que é o Visual Studio (VS)?

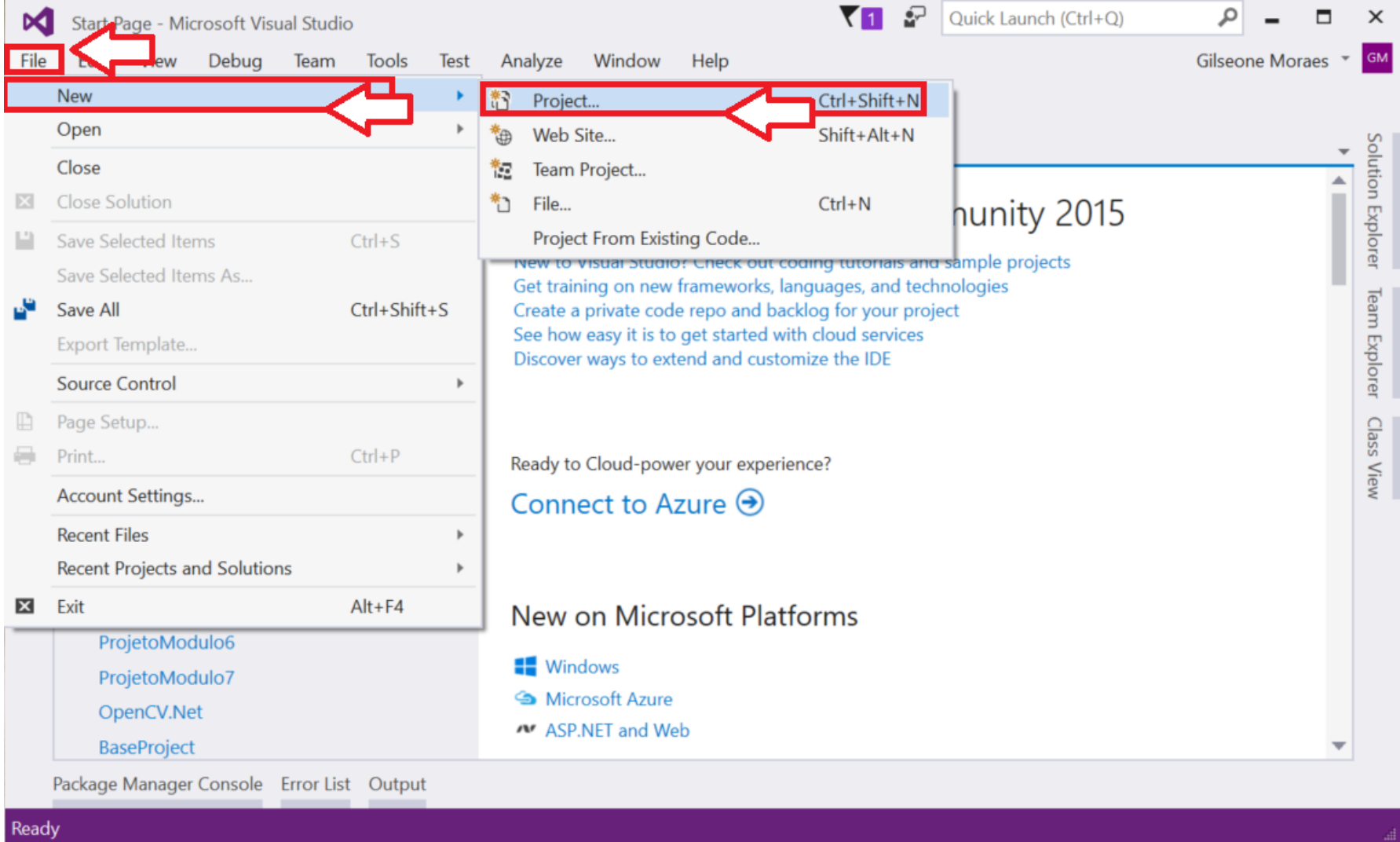
- Um conjunto de ferramentas de desenvolvimento que permite aos desenvolvedores de software resolver problemas complexos e criar soluções inovadoras.
- Baseia-se em três pilares principais:
 - Desenvolvimento rápido de aplicativos
 - Colaboração eficaz em equipe
 - Experiências de usuário inovadoras

Algumas coisas que podem ser feitas com o VS

- Desenvolver aplicativos para as mais diversas plataformas
 - Desktop, Console, Windows Phone, Web Sites, Aplicações Web (ASP.NET), Web Services, Silverlight e outras
- Extender funções do Office
- Desenvolver aplicativos para Android e iOS
- Desenvolver soluções de integração corporativa com Windows Communication Foundation e o Windows Workflow Foundation e muito mais...

Como criar um projeto no Visual Studio

Console Application



Visual Studio 2013

New Project

Recent

Installed

Templates

Visual Basic

Visual C#

Store Apps

Windows Desktop

Web

Office/SharePoint

Cloud

LightSwitch

Reporting

Silverlight

Test

WCF

Workflow

Visual C++

Visual F#

SQL Server

PowerShell

JavaScript

Online

.NET Framework 4.5

Sort by: Default



Search Installed Templates (Ctrl+E)



Windows Forms Application

Visual C#



WPF Application

Visual C#



Console Application

Visual C#



Class Library

Visual C#



Class Library (Portable)

Visual C#



WPF Browser Application

Visual C#



Empty Project

Visual C#



Windows Service

Visual C#



WPF Custom Control Library

Visual C#



WPF User Control Library

Visual C#



Windows Forms Control Library

Visual C#

Type: Visual C#

A project for creating a command-line application

[Click here to go online and find templates.](#)

Name:

MinhaAplicacaoConsole

Location:

C:\Users\Douglas Tybusch\Documents\VS Projects\

Browse...

Solution name:

MinhaAplicacaoConsole

☒ Create directory for solution☐ Add to source control

OK

Cancel

Visual Studio 2015

New Project

Recent

.NET Framework 4.5.2 by: Default

Search Installed Templates (Ctrl+E)

Installed

Templates

Visual C#

Windows

Universal

Windows 8

Classic Desktop

Web

Android

Cloud

Extensibility

iOS

Silverlight

Test

WCF








Workflow

Visual Basic

Visual F#

Visual C++

SQL Server

	Windows Forms Application	Visual C#
	WPF Application	Visual C#
	Console Application	Visual C#
	Shared Project	Visual C#
	Class Library (Portable for iOS, Android and Windows)	Visual C#
	Class Library	Visual C#
	Class Library (Portable)	Visual C#

Type: Visual C#

A project for creating a command-line application

Online

[Click here to go online and find templates.](#)

Name:

MinhaAplicacaoConsole

Location:

c:\users\gilse\documents\visual studio 2015\Projects

Solution name:

MinhaAplicacaoConsole

Browse...

☒ Create directory for solution

☐ Add to Source Control

OK

Cancel

Visão geral de um projeto Console no Visual Studio



File

Edit

View

Project

Build

Debug

Team

Tools

Test

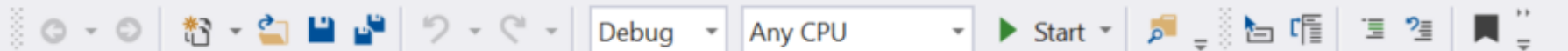
Analyze

Window

Help

Gilseone Moraes

GM



Debug

Any CPU

Start

Server Explorer
Toolbox

Program.cs



MinhaAplicacaoCon

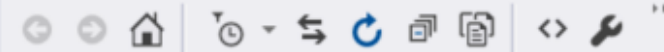
MinhaAplicacaoCon

Main(string[] args)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace MinhaAplicacaoConsole
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13         }
14     }
15 }
16
```

90 %

Solution Explorer

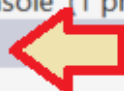


Search Solution Explorer (Ctrl+ç)



Solution 'MinhaAplicacaoConsole' (1 project)

C# MinhaAplicacaoConsole



Properties

References

App.config

Program.cs



Solution Explorer

Team Explorer

Class View

Introdução ao C#

Tipos de Dados - Inteiros

Type	Size (in bits)	Range
sbyte	8	-128 to 127
byte	8	0 to 255
short	16	-32768 to 32767
ushort	16	0 to 65535
int	32	-2147483648 to 2147483647
uint	32	0 to 4294967295
long	64	-9223372036854775808 to 9223372036854775807
ulong	64	0 to 18446744073709551615
char	16	0 to 65535

Tipos de Dados - Decimais

Type	Size (in bits)	precision	Range
float	32	7 digits	1.5×10^{-45} to 3.4×10^{38}
double	64	15-16 digits	5.0×10^{-324} to 1.7×10^{308}
decimal	128	28-29 decimal places	1.0×10^{-28} to 7.9×10^{28}

Tipos de Dados - Booleano e Caractere

Type	Size (in bits)	Range
bool	8	Apenas os valores <i>true</i> ou <i>false</i>
char	16	Caractere único UNICODE
string	variável	Até 1 GB de caracteres UNICODE

- Para maiores informações sobre o padrão UNICODE, consulte: www.unicode.org

Exemplos de declaração

//Inteiros

byte b1 = 255;

sbyte sb1 = -128;

short s = -32767;

ushort us = 65535;

int i = -2147483647;

uint ui = 4294967295;

long l = -9223372036854775808;

ulong ul = 18446744073709551615;

char c1 = (char) 123;

char c2 = 'a';

//Decimais

float f1 = (float) 3.1234567;

float f2 = 3.1234567f;



double d1 = 9812739801723123.1234567891234567;

double d2 = 9812739801723123.1234567891234567d;



decimal dc1 = (decimal) 9812739801723123.1234567891234567;

decimal dc2 = 9812739801723123.1234567891234567m;



//Booleano e Caracteres

bool verdadeiro = true;

char meuChar = 'a';

string minhaString1 = "abc...z";

string minhaString2 = "abc...z" + "\n123...9";

string minhaOutraString3 =

@ "esta string

aceita

linhas";

Escape Character Sequence

Escape Sequence	Meaning
\'	Single Quote
\"	Double Quote
\\	Backslash
\0	Null, not the same as the C# <i>null</i> value
\a	Bell
\b	Backspace
\f	form Feed
\n	Newline
\r	Carriage Return
\t	Horizontal Tab
\v	Vertical Tab

```
//Exemplo sem o caractere de Escape = não compila  
string doubleQuote2 = "\"Single Quote\"";
```

```
//Outros exemplos  
string singleQuote = "'Single Quote'";  
string doubleQuote = "\"Single Quote\"";  
string backslash = "\\Single Quote\\";
```

Uso do @ antes de strings

- O @ também é um caractere importante para ser usado com *strings*, uma vez que ele faz com que a sequência escape não seja executada

```
string s1 = "c:\vs\";  
Console.WriteLine(s1);  
Console.ReadKey();
```

Entire Solution		3 Errors	0 Warn
	Code	Description	
✗	CS1010	Newline in constant	
✗	CS1009	Unrecognized escape sequence	
✗	CS1002	; expected	

```
string s1 = "c:\\vs\\";  
Console.WriteLine(s1);  
Console.ReadKey();
```

OU

```
string s1 = @"c:\vs\";  
Console.WriteLine(s1);  
Console.ReadKey();
```



Entrada e saída de dados via teclado:

- `Console.Read();`
- `Console.ReadLine();`
- `Console.ReadKey();`

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

```
//Imprime uma string na tela
Console.Write("Minha frase");

//Imprime uma string na tela, pulando uma linha no final
Console.WriteLine("Minha nova frase");

//Efetua a leitura da linha inteira
string inputString;
Console.WriteLine("Digite seu nome:");
inputString = Console.ReadLine();

//Efetua a leitura do primeiro caractere, retornando seu valor ASCII
int inputCaractere;
Console.Write("Digite um caractere: ");
inputCaractere = Console.Read();

//Imprime na tela, e aguarda que alguma tecla seja pressionada para prosseguir
Console.WriteLine("Pressione alguma tecla para finalizar.");
Console.ReadKey();

return;
```

Operadores

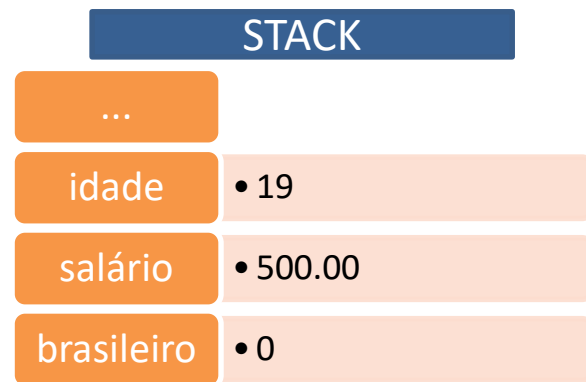
Category	Operator(s)	Associativity
Primary	x.y f(x) a[x] x++ x-- new typeof default checked unchecked delegate	left
Unary	+ - ! ~ ++x --x (T)x	right
Multiplicative	* / %	left
Additive	+ -	left
Shift	<< >>	left
Relational	< > <= >= is as	left
Equality	== !=	right

Category	Operator(s)	Associativity
Logical AND	&	left
Logical XOR	^	left
Logical OR		left
Conditional AND	&&	left
Conditional OR		left
Null Coalescing	??	left
Ternary	?:	right
Assignment	= *= /= %= += - = <<= >>= &= ^= = =>	right

Exemplos mais utilizados

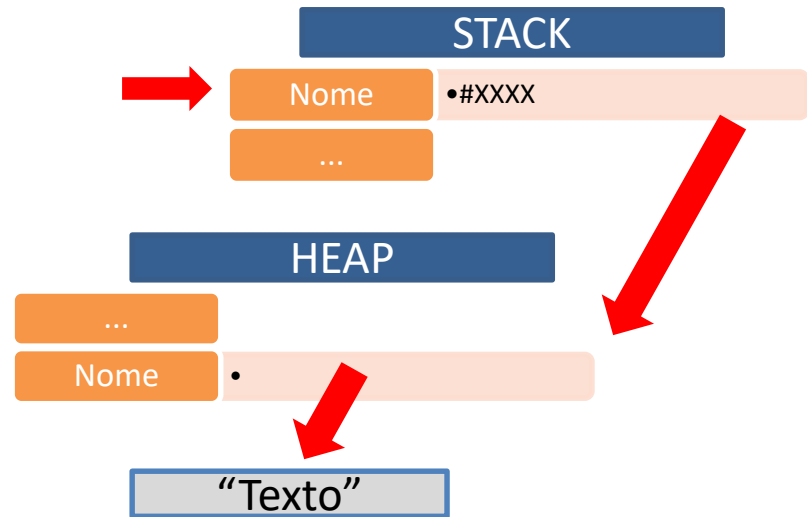
Variáveis Por Valor (Value Types)

- Armazenado na memória Stack
- Trabalha com dados diretamente
- Não pode ser nulo
- Exemplo:
 - Inteiros
 - Decimais
 - Booleanos
 - Estruturas
 - Enumeradores



Por Referência (reference types)

- Contém uma referência a um ponteiro na memória Heap
- Pode ser nulo
- Exemplo:
 - Vetores
 - Textos
 - Instâncias de Classes
 - Classes



Comentários

```
// Comentário de uma linha  
  
/*  
    Comentário com  
    mais de uma linha  
*/
```

```
/// <summary>  
/// Documentação XML  
/// </summary>  
private int Atributo;
```

```
#region Região  
  
private int Atributo1;  
private int Atributo2;  
  
    references  
public void MeuMetodo() { }  
  
#endregion
```

Estruturas de controle de fluxo

- `if ... else if ... else`
 - A cláusula `if` ocorre apenas uma vez
 - A cláusula `else if` pode ocorrer nenhuma ou várias vezes
 - A cláusula `else` é opcional.

Exemplo - IF

```
int idade = 18;

if (idade >= 18) {
    Console.WriteLine("Autorizado.");
    Console.WriteLine("Sua idade é: " + idade);
}
else if(idade > 15 && idade < 18) {
    Console.WriteLine("Somente com os pais.");
    Console.WriteLine("Menor de 18 anos.");
}
else {
    Console.WriteLine("Não autorizado.");
    Console.WriteLine("Menor de 15 anos.");
}
```

Operador Ternário

Sintaxe:

condição ? expressao_eh_true : expressão_eh_falsa;

```
public void ExemploTernario()
{
    int entrada = Convert.ToInt32(Console.ReadLine());
    string classificacao;

    // Utilizando Operador ternário
    classificacao = (entrada > 0) ? "positivo" : "negativo";

    Console.WriteLine(classificacao);
}
```

Comparação: IF e Operador Ternário

```
public void ExemploIf()
{
    int entrada = Convert.ToInt32(Console.ReadLine());
    string classificacao;

    // Utilizando IF
    if (entrada > 0)
        classificacao = "positivo";
    else
        classificacao = "negativo";

    Console.WriteLine(classificacao);
}
```

```
public void ExemploTernario()
{
    int entrada = Convert.ToInt32(Console.ReadLine());
    string classificacao;

    // Utilizando Operador ternário
    classificacao = (entrada > 0) ? "positivo" : "negativo";

    Console.WriteLine(classificacao);
}
```

Estruturas de controle de fluxo

- switch ... case
 - Estruturas de decisão caracterizadas pela possibilidade de uma variável possuir vários valores.
 - A cláusula **switch** ocorre uma vez
 - A cláusula **case** pode ocorrer uma ou várias vezes
 - A cláusula **default** é opcional.

Exemplo: switch

```
public void ExemploSwitch()
{
    string sexo = "Homem";

    switch (sexo)
    {
        case "masculino":
            Console.Write("Homem");
            break;

        case "feminino":
            Console.Write("Mulher");
            break;

        default:
            Console.Write("Não informado");
            break;
    }
}
```

Estruturas de Repetição: for

- Estrutura de repetição caracterizada pela existência de três parâmetros:
 - Valor inicial
 - Condição para parada das iterações
 - Quantidade de incrementos/decrementos a cada iteração

```
for (int i = 0; i < 10; i++)  
{  
    Console.WriteLine(i);  
}
```


Estruturas de Repetição: while

Estrutura de repetição que realiza as operações desejadas enquanto a condição especificada for verdadeira.

```
int j = 0;
while (j != 10)
{
    Console.WriteLine(j);
    j++;
}
```

Estruturas de Repetição: do ... while

- Semelhante à anterior, com o diferencial de que as condições são verificadas no final da execução
- Permite que as operações especificadas sejam executadas pelo menos uma vez.

```
int k = 0;  
do  
{  
    Console.WriteLine(k);  
    k++;  
} while (k != 10);
```

Estruturas de Repetição: foreach

- É uma variação do for
- Sua diferença está no fato de ser necessário apenas especificar uma variável e a coleção ou array cujos registros serão percorridos.

```
int[] vetor = { 1, 3, 5, 7, 9 };  
foreach (int l in vetor)  
{  
    Console.WriteLine(l);  
}
```

Comparação: foreach e for

```
List<Pessoa> lstPessoas2 = new List<Pessoa>();  
for (int m = 0; m < lstPessoas2.Count; m++)  
{  
    Console.WriteLine(lstPessoas2[m].Nome);  
    Console.WriteLine(lstPessoas2[m].Endereco);  
    Console.WriteLine(lstPessoas2[m].CPF);  
}
```

```
List<Pessoa> lstPessoas1 = new List<Pessoa>();  
foreach (Pessoa p in lstPessoas1)  
{  
    Console.WriteLine(p.Nome);  
    Console.WriteLine(p.Endereco);  
    Console.WriteLine(p.CPF);  
}
```

Convertendo

```
// Exemplo I
string S = "123";
int i = S;
Console.WriteLine(i);
```

```
// Exemplo II
string S = "123";
int i = Convert.ToInt32(S);
Console.WriteLine(i);
```

```
// Exemplo III
string S = "123";
int i = int.Parse(S);
Console.WriteLine(i);
```

```
// Exemplo IV
string S = "valor";
int i = Convert.ToInt32(S);
Console.WriteLine(i);
```

```
// Exemplo V
string S = "valor";
int i = 0;
if (int.TryParse(S, out i)) {
    Console.WriteLine(i);
}
else {
    Console.WriteLine("inválido");
}
```

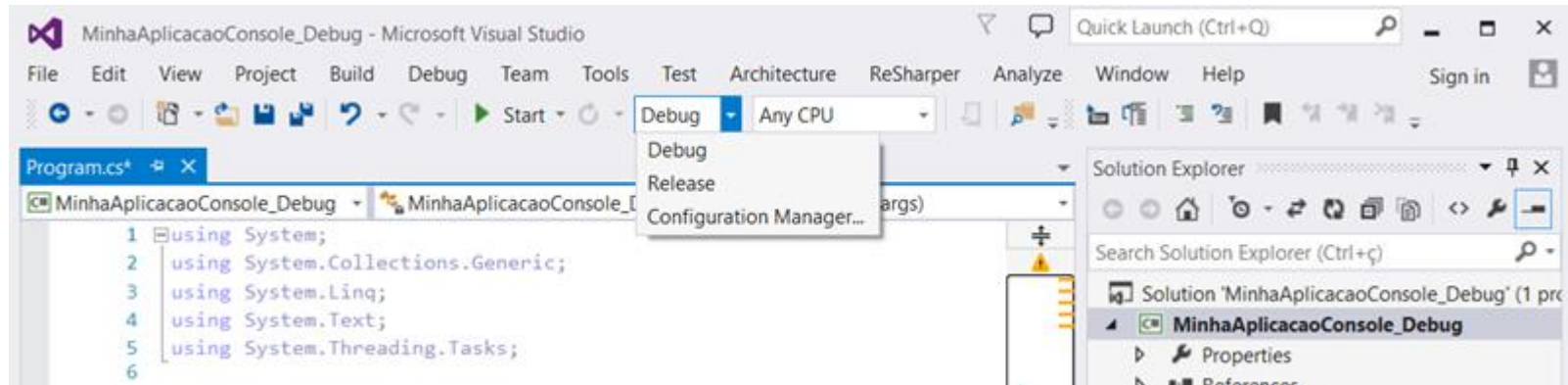
Convertendo: String para short, int e long

- `Convert.ToInt16();` → short → `Int16`
- `Convert.ToInt32();` → int → `Int32`
- `Convert.ToInt64();` → long → `Int64`

Convertendo: cast

- `double vlr = 25.5525;`
- `Int valor = (int) vlr;`

Debug x Release



MinhaAplicacaoConsole_Debug - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Architecture ReSharper Analyze Window Help

Quick Launch (Ctrl+Q)

Start

Debug Any CPU

Program.cs

MinhaAplicacaoConsole_Debug MinhaAplicacaoConsole_Debug.Program Main(string[] args)

```
6
7 namespace MinhaAplicacaoConsole_Debug
8 {
9     0 references
10    class Program
11    {
12        0 references
13        static void Main(string[] args)
14        {
15            int[] vetor;
16            decimal d;
17
18            vetor = new int[5]{1, 2, 3, 4, 5};
19
20            d = 123.123m;
21
22        }
23    }
24 }
```

100 %

Output

Show output from: Debug

10:32.896 'MinhaAplicacaoConsole_Debug.vshost.exe' (CLR v4.0.30319: MinhaAplicacaoConsole_Debug.vshost.exe): Loaded 'C:\Users\Douglas Tybusch\Documents\VS Projects\Minha'▲
11:01.778 The program '[2084] MinhaAplicacaoConsole_Debug.vshost.exe' has exited with code -1 (0xffffffff).

Error List Output

Ready Ln 22 Col 6 Ch 6 INS

```
Program.cs
MinhaAplicacaoConsole_Debug
MinhaAplicacaoConsole_Debug.Program
Main(string[] args)


6
7 namespace MinhaAplicacaoConsole_Debug
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             int[] vetor;
14             decimal d;
15
16             vetor = new int[5]{1, 2, 3, 4, 5};
17
18             d = 123.123m;
19
20         }
21     }
22 }
23
24
```

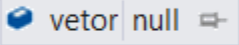
100 %

Output

Show output from: Debug

10:32.896 MinhaAplicacaoConsole_Debug.vshost.exe (CLR v4.0.30319: MinhaAplicacaoConsole_Debug.vshost.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft...'
10:32.896 'MinhaAplicacaoConsole_Debug.vshost.exe' (CLR v4.0.30319: MinhaAplicacaoConsole_Debug.vshost.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_32\System.Data'
10:32.896 'MinhaAplicacaoConsole_Debug.vshost.exe' (CLR v4.0.30319: MinhaAplicacaoConsole_Debug.vshost.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Xml'
10:32.896 The thread 0xf6c has exited with code 259 (0x103).
10:32.896 The thread 0x1488 has exited with code 259 (0x103).
10:32.896 'MinhaAplicacaoConsole_Debug.vshost.exe' (CLR v4.0.30319: MinhaAplicacaoConsole_Debug.vshost.exe): Loaded 'C:\Users\Douglas Tybusch\Documents\VS Projects\Minha'




```
9  class Program
10  {
    0 references
11  static void Main(string[] args)
12  {
13      int[] vetor;
14      decimal d;
15
16
17      vetor = new int[5]{1, 2, 3, 4, 5};
18      
19      d = 123.123m;
20
21  }
22  }
```

```
9  class Program
10  {
    0 references
11  static void Main(string[] args)
12  {
13      int[] vetor;
14      decimal d;
15
16
17      vetor = new int[5]{1, 2, 3, 4, 5};
18
19      d = 1;
20
21  }
22  }
23
24
```

vetor {int[5]}

[0]	1
[1]	2
[2]	3
[3]	4
[4]	5








Autos			▼	📌	✕
Name		Value	Type		
 d		123.123	decimal		

AutosLocalsWatch 1

Locals				▼	🔍	✕
Name		Value		Type		
args		{string[0]}		string[]		
vetor	▲	{int[5]}		int[]		
	args [0]	1		int		
	args [1]	2		int		
	args [2]	3		int		
	args [3]	4		int		
	args [4]	5		int		
d		123.123		decimal		

AutosLocalsWatch 1

Watch 1

Name	Value	Type
▼  vetor	{int[5]}	int[]
 [0]	1	int
 [1]	2	int
 [2]	3	int
 [3]	4	int
 [4]	5	int
▶  vetor	{int[5]}	int[]

Autos Locals Watch 1

Breakpoints

New ▾



Columns ▾

Search:



In Column:

All visible

Name

Labels

Condition

Hit Count



Program.cs, line 12 character 9

(no condition)

break always (currently 1)



Program.cs, line 17 character 13

(no condition)

break always (currently 1)



Program.cs, line 21 character 9

(no condition)

break always (currently 0)

[Breakpoints](#)

Command Window

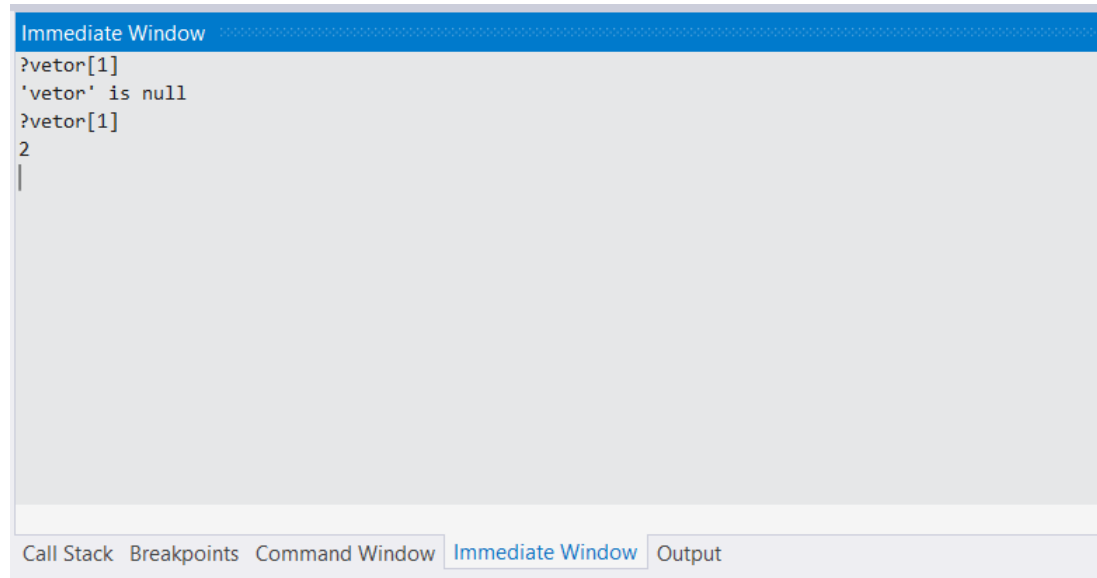
Immediate Window

Output

Error List

Debug de código no Visual Studio

- Immediate Window



MINICURSO

