



INTRODUÇÃO AO FRAMEWORK .NET

Cairu
Fundação Visconde de Cairu - Desde 1905

AULA 2: INTRODUÇÃO AO C#

- ❖ Tipos de dados;
- ❖ If e else;
- ❖ Switch case;
- ❖ For e while;
- ❖ Console Application: entrada e saída console.

TIPOS DE DADOS

- ❖ C# é uma linguagem fortemente tipada;
- ❖ A declaração de variáveis devem ser informadas o tipo de dados;
- ❖ Pode-se fazer também conversão de tipos: implícita e explícita;

Short Name	.NET Class	Type	Width	Range (bits)
byte	Byte	Unsigned integer	8	0 to 255
sbyte	SByte	Signed integer	8	-128 to 127
int	Int32	Signed integer	32	-2,147,483,648 to 2,147,483,647
uint	UInt32	Unsigned integer	32	0 to 4294967295
short	Int16	Signed integer	16	-32,768 to 32,767
ushort	UInt16	Unsigned integer	16	0 to 65535
long	Int64	Signed integer	64	-922337203685477508 to 922337203685477507
ulong	UInt64	Unsigned integer	64	0 to 18446744073709551615
float	Single	Single-precision floating point type	32	-3.402823e38 to 3.402823e38
double	Double	Double-precision floating point type	64	-1.79769313486232e308 to 1.79769313486232e308
char	Char	A single Unicode character	16	Unicode symbols used in text
bool	Boolean	Logical Boolean type	8	True or false
object	Object	Base type of all other types		
string	String	A sequence of characters		
decimal	Decimal	Precise fractional or integral type that can represent decimal numbers with 29 significant digits	128	$\pm 1.0 \times 10e-28$ to $\pm 7.9 \times 10e28$

TIPOS DE DADOS

```
int Idade = 0;
```

```
Idade = 19;
```

```
string Nome = "Fulano";
```

```
float valor;
```

```
valor = 15.35;
```

TIPOS DE DADOS – CONVERSÃO IMPLÍCITA

Source Type	Target Type
Byte	short, ushort, int, uint, long, ulong, float, double, or decimal
Sbyte	short, int, long, float, double, or decimal
Int	long, float, double, or decimal
UInt	long, ulong, float, double, or decimal
Short	int, long, float, double, or decimal
Ushort	int, uint, long, ulong, float, double, or decimal
Long	float, double, or decimal
Ulong	float, double, or decimal
Float	double
Char	ushort, int, uint, long, ulong, float, double, or decimal

TIPOS DE DADOS — CONVERSÃO EXPLÍCITA

Source Type	Target Type
Byte	sbyte or char
Sbyte	byte, ushort, uint, ulong, or char
Int	sbyte, byte, short, ushort, uint, ulong, or char
UInt	sbyte, byte, short, ushort, int, or char
Short	sbyte, byte, ushort, uint, ulong, or char
Ushort	sbyte, byte, short, or char
Long	sbyte, byte, short, ushort, int, uint, ulong, or char
Ulong	sbyte, byte, short, ushort, int, uint, long, or char
Float	sbyte, byte, short, ushort, int, uint, long, ulong, char, or decimal
Double	sbyte, byte, short, ushort, int, uint, long, ulong, char, float, or decimal
Char	sbyte, byte, or short
Decimal	sbyte, byte, short, ushort, int, uint, long, ulong, char, float, or double

ESTRUTURAS CONDICIONAIS E REPETIÇÃO

- ❖ if/else If/else

- ❖ switch case

- ❖ if ternário

OPERADORES

Operador	Função	Exemplo
==	Comparação de igualdade	Nome == "Fulano"
!=	Comparação de diferente	Nome != "Fulano"
> e >=	Maior e maior e igual	Idade > 30 e idade >= 30
< e <=	Menor e menor e igual	Idade <30 e idade <=30
&&	Operador lógico And , as comparações devem ser todas verdadeiras	Idade == 30 && Nome = "Fulano"
	Operador lógico Or, as comparações devem ter pelo menos uma verdadeira	Idade == 30 && Nome = "Fulano"
%	Divide um número retornando o resto da divisão	Idade % 2 == 0

IF

```
If (condição) {  
    comandos  
}  
else if (condição){  
    comandos  
}  
else{  
    comandos  
}
```

```
if (Idade >= 100) {  
    Interaction.MsgBox("Idade Incorreta.");  
}  
else if (Idade <= 0) {  
    Interaction.MsgBox("Idade Incorreta.");  
}  
else {  
    Interaction.MsgBox("Idade aceita.  
Obrigado");  
}
```

IF TERNÁRIO

(condição ? retorno true : retorno false)

```
String sexo;  
sexo = (opcao == 'M' ? "Masculino" : "Feminino");
```

SWITCH CASE

```
switch Variável {
```

```
    Case 1:
```

```
        instrução
```

```
        break;
```

```
    Case 2 | 3 | 4:
```

```
        instrução
```

```
        break;
```

```
    default:
```

```
        instrução
```

```
        break;
```

```
}
```

```
int Nota = 0;
```

```
switch (Nota)
```

```
{
```

```
    case 10 | 11 | 12:
```

```
        MessageBox.Show("Acima de 10 até 12");
```

```
        break;
```

```
    case 13:
```

```
        MessageBox.Show("Valor é 13");
```

```
        break;
```

```
    default:
```

```
        MessageBox.Show("Não encontrei...");
```

```
        break;
```

```
}
```

LAÇOS

❖ While

❖ Do/While

❖ For

❖ **For Each**

WHILE

```
while (condição){  
    instruções  
}
```

```
int Contador = 10;  
while (Contador >= 10 && Contador < 15)  
{  
  
    Console.WriteLine("Rodei...while");  
    Contador += 1;  
}
```

DO/WHILE

```
do{  
    instruções  
} While (condição);
```

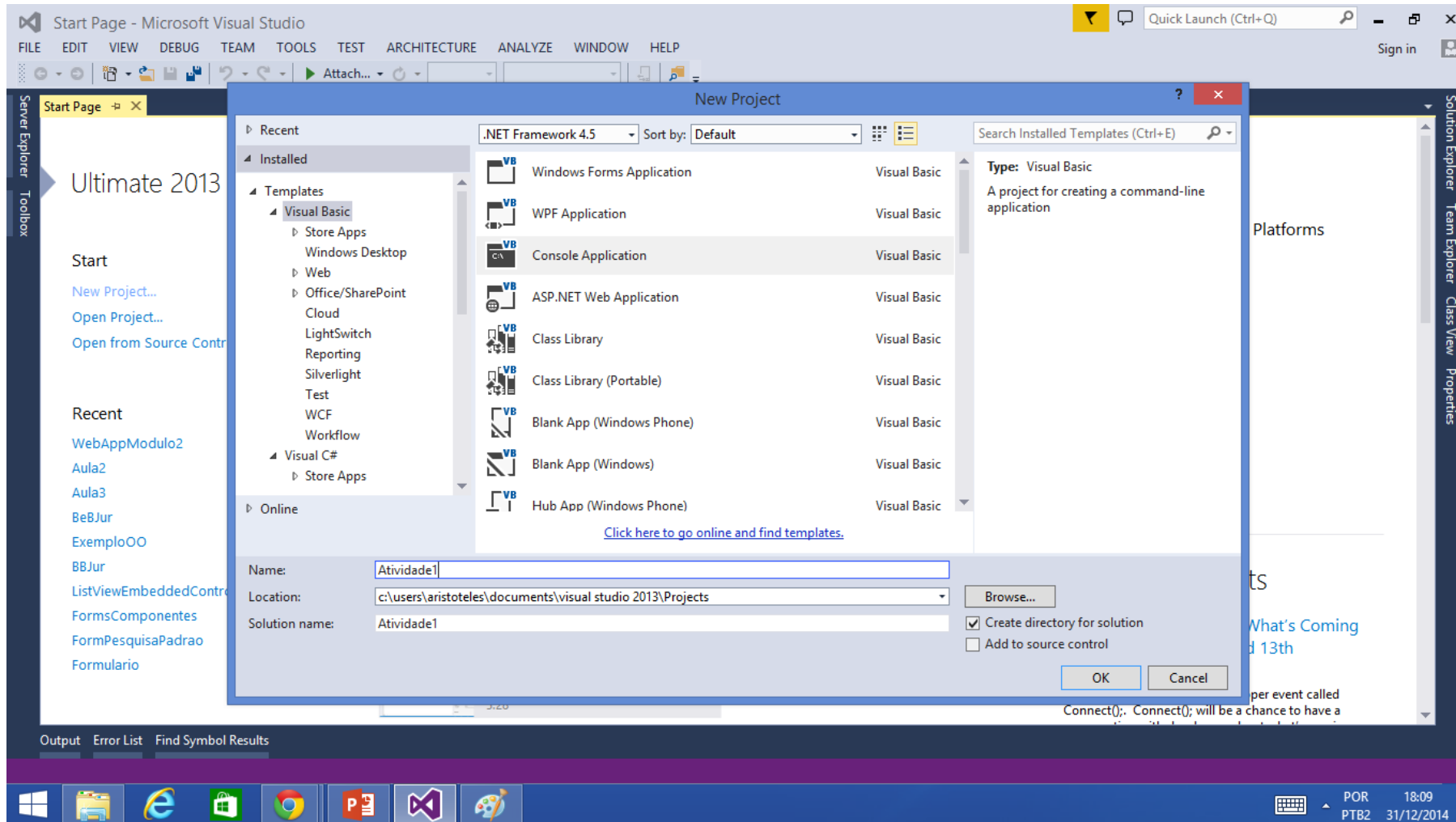
```
int Contador = 1;  
do  
{  
    Console.WriteLine("Rodei...do");  
    Contador += 1;  
} while (Contador < 10);
```

FOR

```
for (inicialização; teste;  
incremento){  
    instruções  
}
```

```
int Contador = 0;  
for (Contador = 1; Contador <= 10; Contador++)  
{  
    Console.WriteLine("Rodei..{0}", Contador);  
}
```


CRIANDO CONSOLE APPLICATION



CONSOLE APPLICATION

❖ Entrada de Dados

```
string retorno = null;  
retorno = Console.ReadLine();
```

❖ Saída de Dados

```
Console.WriteLine("Retorno: {0}", retorno);
```



PRÁTICA 2