

SISTEMAS DISTRIBUÍDOS

Aplicações Console C#
Visual Studio

PROFESSOR HILDEBERTO MELO



Apresentação

- Técnico em Desenvolvimento de Sistemas - Ibratéc, Recife-PE
- Bacharel em Sistemas de Informação – FIR, Recife-PE
- Especialista em Docência no Ensino Superior – Faculdade Maurício de Nassau, Recife-PE
- Mestre em Ciência da Computação – UFPE/CIN, Recife-PE
- *Currículo Lattes* <http://lattes.cnpq.br/0759508594425296>)
- *Homepage* <https://sites.google.com/site/hildebertomelo/>



Disciplinas Lecionadas

- Desenvolvimento de Aplicações Desktop
- Programação Orientada a Objetos
- Estrutura de Dados
- Tecnologia da Informação & Sociedade
- Sistemas Operacionais
- Sistemas Distribuídos
- Introdução a Informática
- Lógica de Programação
- Informática Aplicada a Saúde
- Banco de Dados
- Projeto de Banco de Dados
- Análise de Projetos Orientado a Objetos
- Programação Cliente Servidor
- Linguagens de Programação: C, C#, Pascal, PHP, ASP, Delphi, Java, JavaScript
- Programação WEB

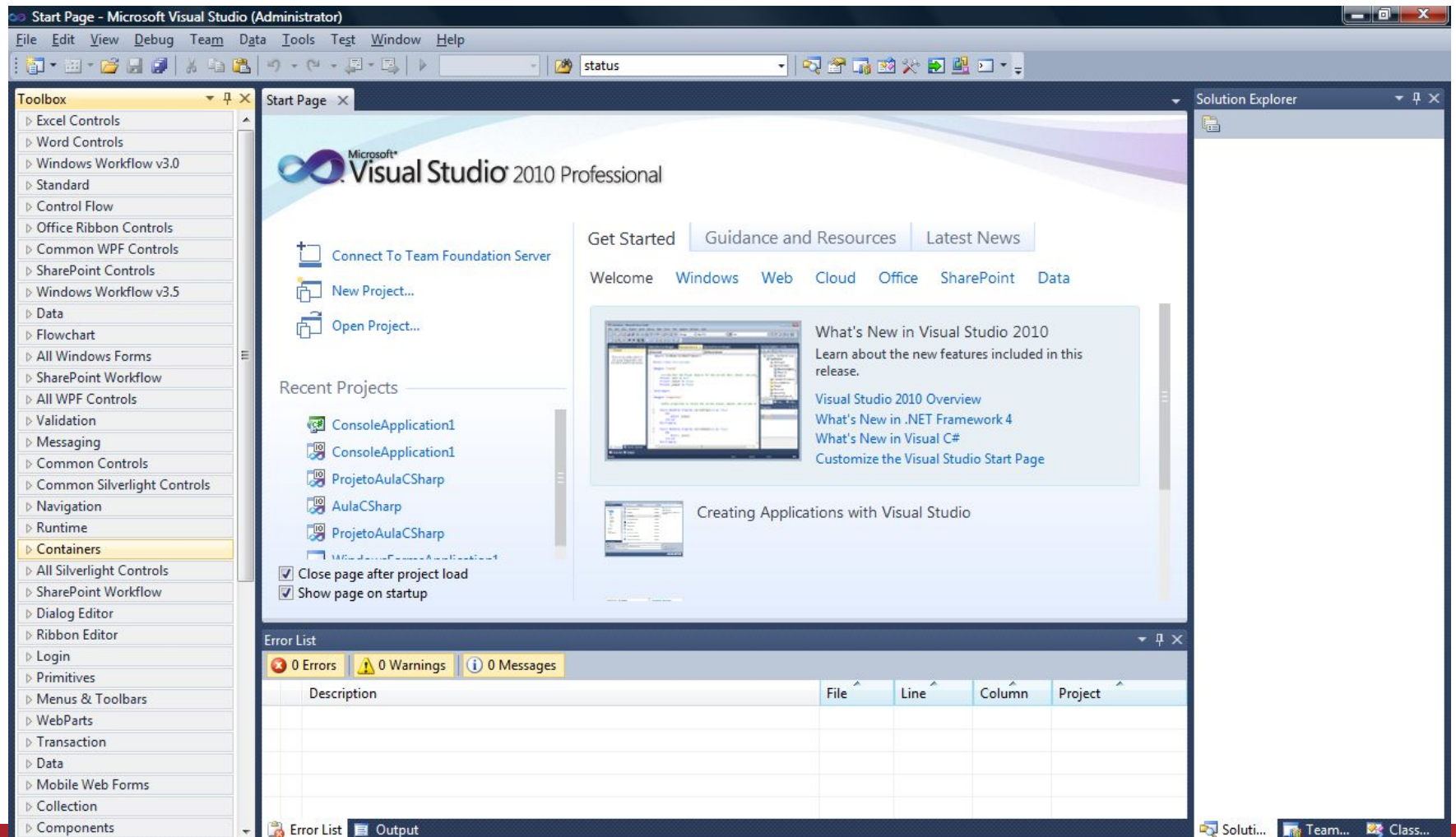


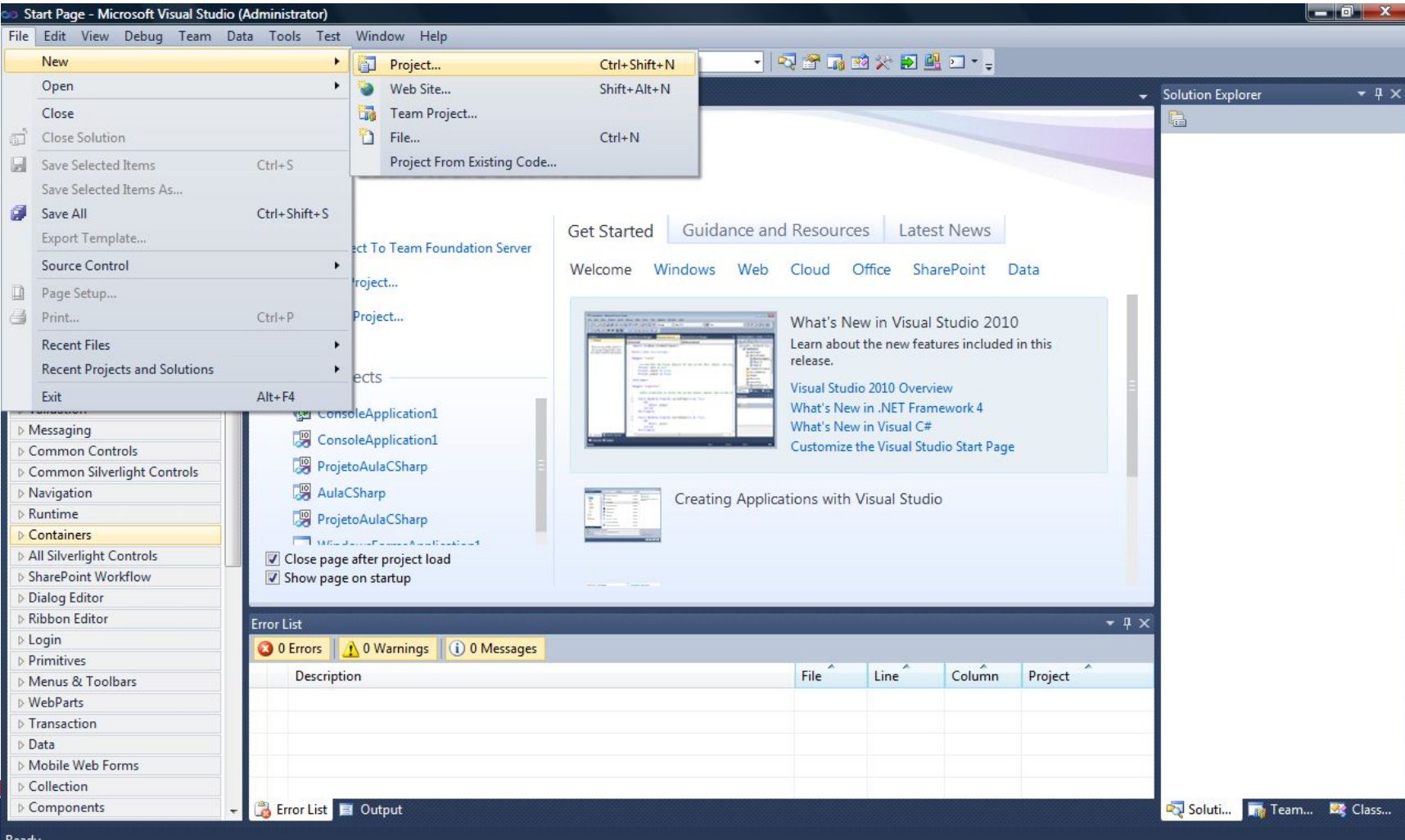
Roteiro

- Criando uma aplicação console – Visual Studio
- Principais Comandos Aplicação Console
- Conversões de Tipos
- Promoção de Argumentos
- Class Convert
- Exercícios



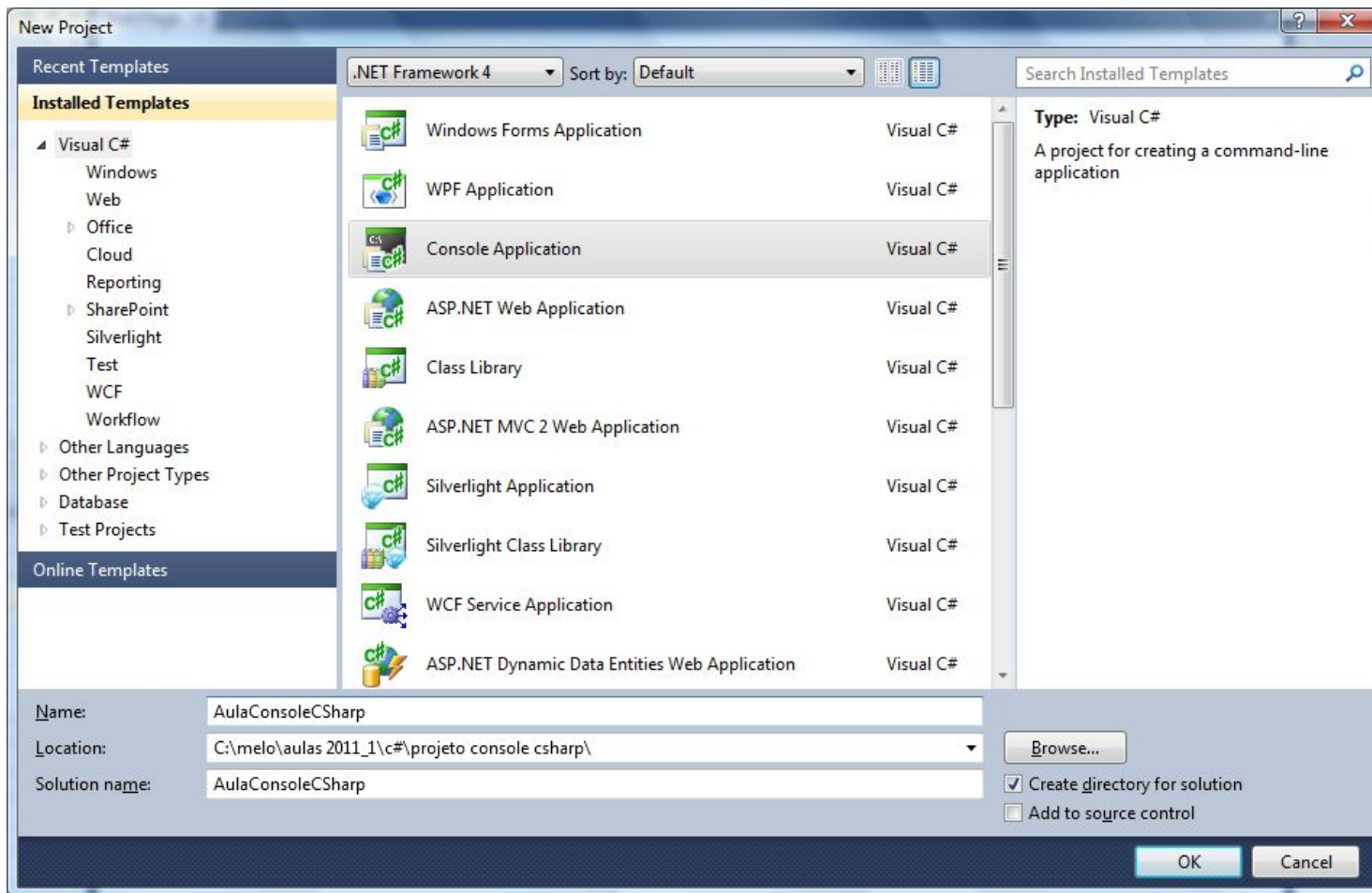
IDE Visual Studio



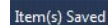




Criando uma Aplicação Console





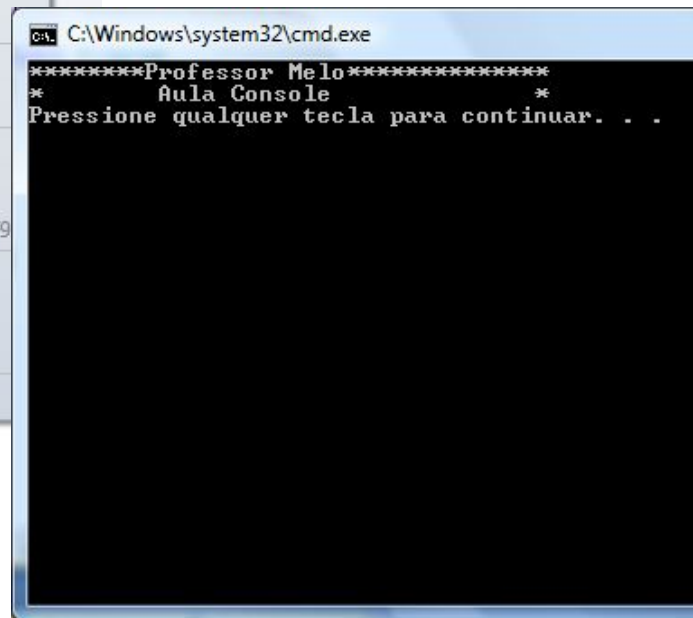
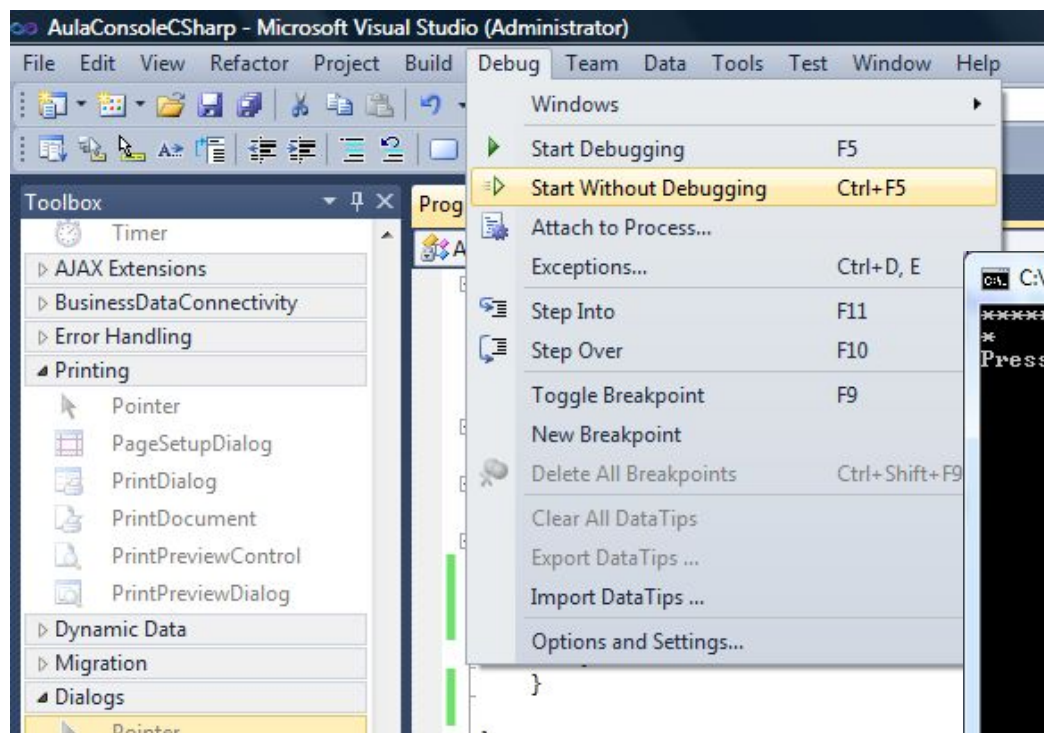




Criando uma Aplicação

Console

- Executando a aplicação





Aplicações Console

- Título da janela do programa
 - `Console.Title = "Sistema console";`
- Cor do fundo do texto
 - `Console.BackgroundColor = ConsoleColor.Green;`
- Cor do texto
 - `Console.ForegroundColor = ConsoleColor.Black;`
- Altura da janela
 - `Console.WindowHeight = 30;`
- Largura da janela
 - `Console.WindowWidth = 30;`



Aplicações Console – Cores na Aplicação

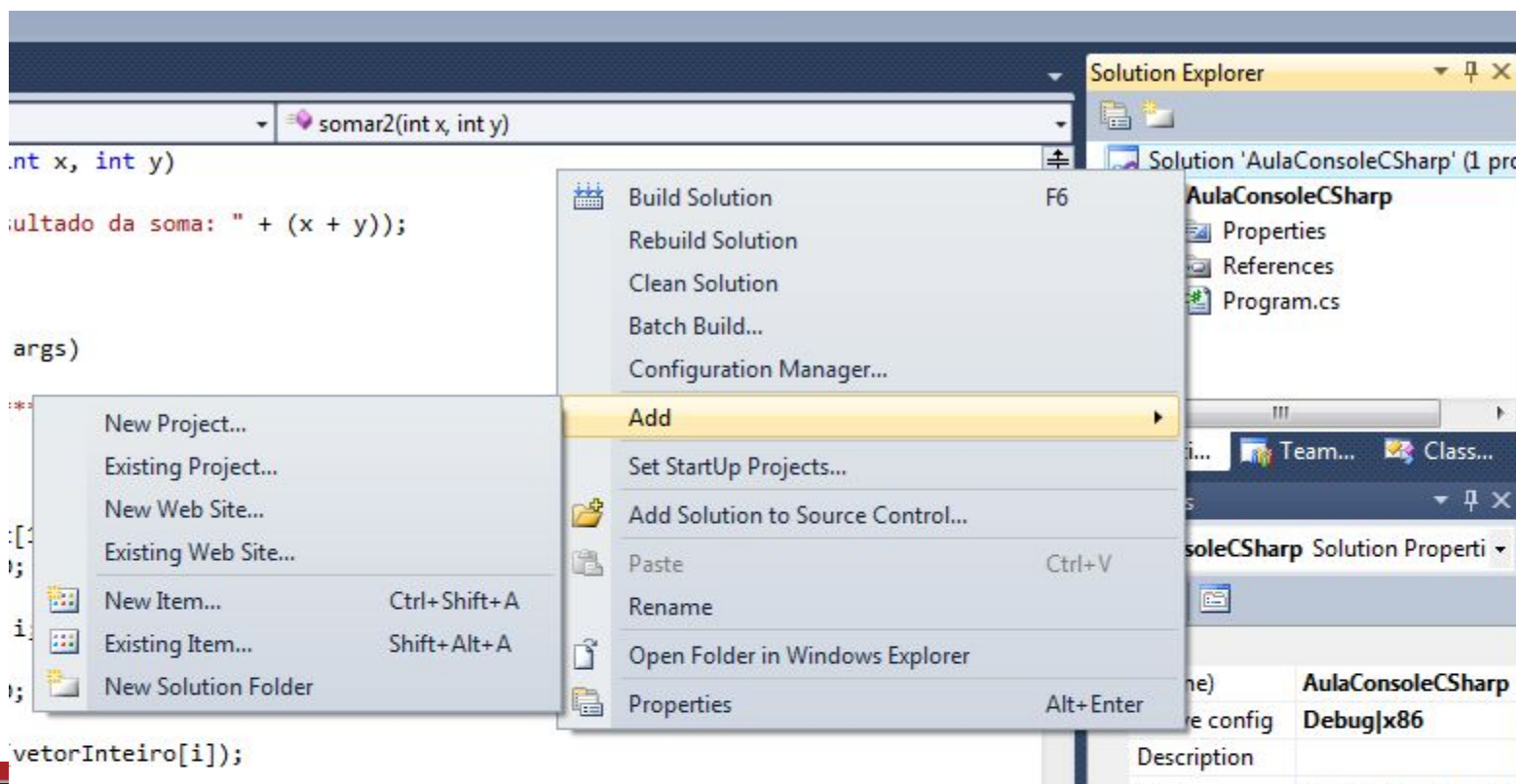
- Exemplo:
 - Console.Title = "Sistema Professor Melo";
 - Console.BackgroundColor = ConsoleColor.Green;
 - Console.ForegroundColor = ConsoleColor.Black;
 - Console.WriteLine("*****Professor Melo*****");
 - Console.BackgroundColor = ConsoleColor.Blue; Console.ForegroundColor = ConsoleColor.Black;
 - Console.WriteLine("* Aula Console *");
 - Console.BackgroundColor = ConsoleColor.Yellow;
 - Console.ForegroundColor = ConsoleColor.Black;

```
04. Sistema Professor Melo
*****Professor Melo*****
*      Aula Console      *
Pressione qualquer tecla para continuar. . .
```



Observação

- Na mesma solução poderemos ter vários projetos





Principais Comandos Aplicação Console

- `Console.Write` – comando de saída para escrever um texto no console
- `Console.WriteLine` - comando de saída para escrever um texto **com quebra de linha** no console
- `Console.ReadLine` - comando de entrada para receber uma string informada pelo usuário
- `Console.ReadKey();` - comando de entrada que espera qualquer tecla ser pressionada
- `Console.Clear();` - limpa a tela do console



Principais Comandos Aplicação Console

- **#region**
- //divide o bloco de código em partes
- **#endregion**

```
class Program
{
    static void Main(string[] args)
    {

        #region te amo melo

        Console.Write("Professor \n" + "Melo");

        Console.ReadKey();

        #endregion
        posicionando o cursor
        ordenando vetor
        capturando a tecla digitada
    }
}
```



Principais Comandos Aplicação Console

```
//recebendo a tecla digitada pelo usuário
ConsoleKeyInfo tecla = Console.ReadKey();
//verificando uma determinada tecla
if (tecla.Key == ConsoleKey.Escape)
{
    Console.WriteLine("Clicou ESC");
}
else if (tecla.Key == ConsoleKey.Enter)
{
    Console.WriteLine("Clicou Enter");
}
```



Principais Comandos Aplicação Console

//posicionando o cursor de digitação

`Console.SetCursorPosition(coluna, linha);`

```
Program.cs
Program.cs X
ConsoleApplication2.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {

            Console.SetCursorPosition(0, 15);
            Console.Write("Digitando na 15ª linha ");
            Console.ReadKey();

            #region

            #region

        }
    }
}
```

```
file:///C:/Users/Professor/Desktop/ConsoleApplication1Figuras/ConsoleApplication2/bin/Debug/C...
Digitando na 15ª linha
```



Identificação de Código Fonte

- Ctrl+E, D --> Formata documento
- Ctrl+E. F --> Formata seleção.



Conversões de Tipos

- **Conversões implícitas** : Nenhuma sintaxe especial é necessária porque a conversão é segura tipo e nenhum dado será perdido. Exemplos incluem conversões de menor para o maior tipos integral, e conversões de classes derivadas de classes base.

```
int num = 2147483647;  
long bigNum = num;
```



Conversões de Tipos

- **Conversões explícitas (cast)** : Conversões explícitas exigem um operador de conversão. Função é necessária quando as informações podem ser perdidos na conversão, ou quando a conversão não poderia ter sucesso por outras razões. Exemplos típicos incluem conversão numérica para um tipo que tem menos precisão ou um intervalo menor, e conversão de uma instância da classe base para uma classe derivada.

```
double x = 1234.7;  
int a;  
// Cast double to int.  
a = (int)x;  
System.Console.WriteLine(a);
```




Conversões de Tipos

- `Int32.parse`("texto a ser convertido") – comando para converter uma string em um inteiro
 - Ex1:
 - `int idade;`
 - `idade = Int32.Parse("18");`
 - Ex2:
 - `Console.WriteLine("Digite a idade");`
 - `idade = Int32.Parse(Console.ReadLine);`
- `float.parse`("texto a ser convertido") – comando para converter uma string em um float
 - Ex1:
 - `float salario;`
 - `salario = float.Parse("1000.45");`
 - Ex2:
 - `Console.WriteLine("Digite o salario");`
 - `salario = float.Parse(Console.readline);`



Promoção de Argumentos

Tipo	Pode ser convertido para
bool	object
byte	decimal, double, float, int, uint, long, ulong, object, short ou ushort
sbyte	decimal, double, float, int, long, object ou short
char	decimal, double, float, int, uint, long, ulong, object ou ushort
decimal	object
double	object
float	double ou object
int	decimal, double, float, long ou object
uint	decimal, double, float, long, ulong ou object
long	decimal, double, float ou object
ulong	decimal, double, float ou object
short	decimal, double, float, int, long ou object
ushort	decimal, double, float, int, long, ulong ou object



Class Convert

- Converte um tipo de dados base para outro tipo de dados base.

```
int x = Convert.ToInt32("10");  
double y = Convert.ToDouble(45);  
double z = Convert.ToDouble("45");
```



ToBoolean	Converte um valor especificado para um valor booleano equivalente.
ToByte	Converte um valor especificado para um inteiro de 8 bits sem sinal.
ToChar	Converte um valor especificado para um caractere Unicode.
ToDateTime	Converte um valor especificado para um DateTime .
ToDecimal	Converte um valor especificado para um Decimal número.
ToDouble	Converte um valor especificado para um número de ponto flutuante de precisão dupla.
ToInt16	Converte um valor especificado para um inteiro assinado de 16 bits.
ToInt32	Converte um valor especificado para um inteiro assinado de 32 bits.
ToInt64	Converte um valor especificado para um inteiro assinado de 64 bits.
ToSByte	Converte um valor especificado para um inteiro assinado de 8 bits.
ToSingle	Converte um valor especificado para um número de ponto flutuante de precisão simples.
ToString	Converte o valor especificado para seu equivalente Cordas representação.
ToUInt16	Converte um valor especificado para um inteiro de 16 bits sem sinal.
ToUInt32	Converte um valor especificado para um inteiro de 32 bits sem sinal.
ToUInt64	Converte um valor especificado para um inteiro de 64 bits sem sinal.



Exercícios

- Criar uma solução no VisualStudio
- Utilizar os principais comandos vistos nesta aula
- Na solução criada, adicionar um projeto no modo Console para cada exercício proposto.



Exercícios

- Exercício 1: criar um programa que receba quatro valores inteiros e ao final mostre os valores em ordem crescente
- Exercício 2: criar um programa que receba o nome, e duas notas de 50(cinquenta alunos) e ao final informar o aluno que possua a maior média, o aluno que possua a menor média



Exercícios

- Exercício 3: criar um programa que receba 15 valores numéricos e ao final apresente os valores na ordem crescente
- Exercício 4: criar um programa que receba o nome, e o salário de 20(vinte) funcionários e ao final informar o funcionário que possua o maior salário, e a relação do nome e salário de todos os funcionários, esta relação deverá estar em ordem decrescente de salário



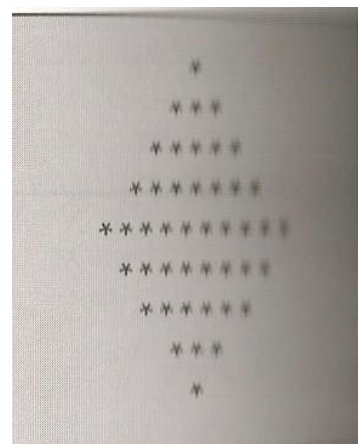
Exercícios

- Exercício 5: criar um programa que utiliza métodos para calcular
 - A área das seguintes figuras geométricas
 - Quadrado = $\text{lado} * \text{lado}$
 - Retângulo = $\text{altura} * \text{base}$
 - Triângulo = $(\text{base} * \text{altura}) / 2$
 - Losango = $(\text{diagonalMaior} * \text{diagonalMenor}) / 2$
 - Trapézio = $((\text{baseMaior} + \text{baseMenor}) / 2) * \text{altura}$
 - Circulo = **Math.PI** * $(\text{raio} * \text{raio})$
 - Raiz quadrada de um número
 - Perímetro das seguintes figuras geométricas
 - Quadrado
 - Retângulo
 - Triângulo equilátero

•



- [illegible]



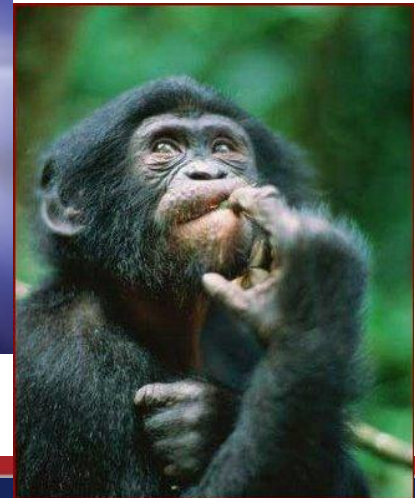


Exercícios

- Exercício 7: Ver descrição da atividade na documentação de avaliação da primeira competência. (disponibilizado pelo professor melo)



Perguntas





Referências

Bibliografia Básica (títulos, periódicos, etc.)						
Título/Periódico	Autor	Edição	Local	Editora	Ano	LT/BV¹
Microsoft Visual C# 2008: passo a passo	SHARP, John	1ª	Porto Alegre	Bookman	2008	LT
Desenvolvimento em camadas com C# .Net	CAMACHO JÚNIOR, Carlos Olavo de Azevedo	1ª	Florianópolis	Visual Books	2008	LT
C#: como programar	DEITEL, H. M.	1ª	São Paulo	Pearson	2003	LT
Bibliografia Complementar (títulos, periódicos, etc.)						
Título/Periódico	Autor	Edição	Local	Editora	Ano	LT/BV¹
Use a cabeça C#	STELLMAN, Andrew; GREENE, Jeniffer	2ª	Rio de Janeiro	Alta Books	2011	LT
Treinamento avançado em XML	FARIA, Rogério Amorim De	1ª	São Paulo	<u>Digerati</u>	2005	LT
Projeto de Banco de Dados com XML	GRAVES, Mark	1ª	São Paulo	Pearson	2003	BV