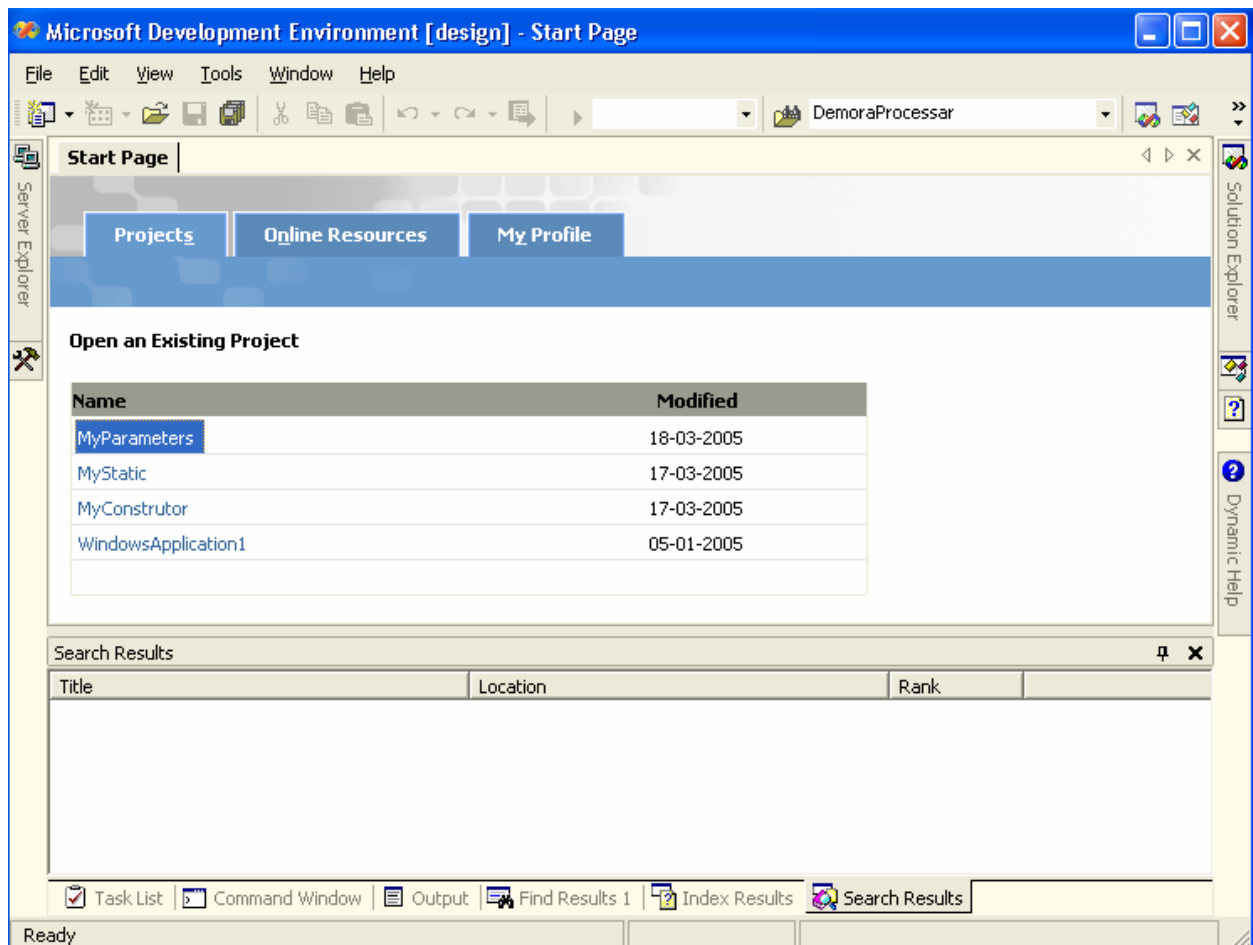




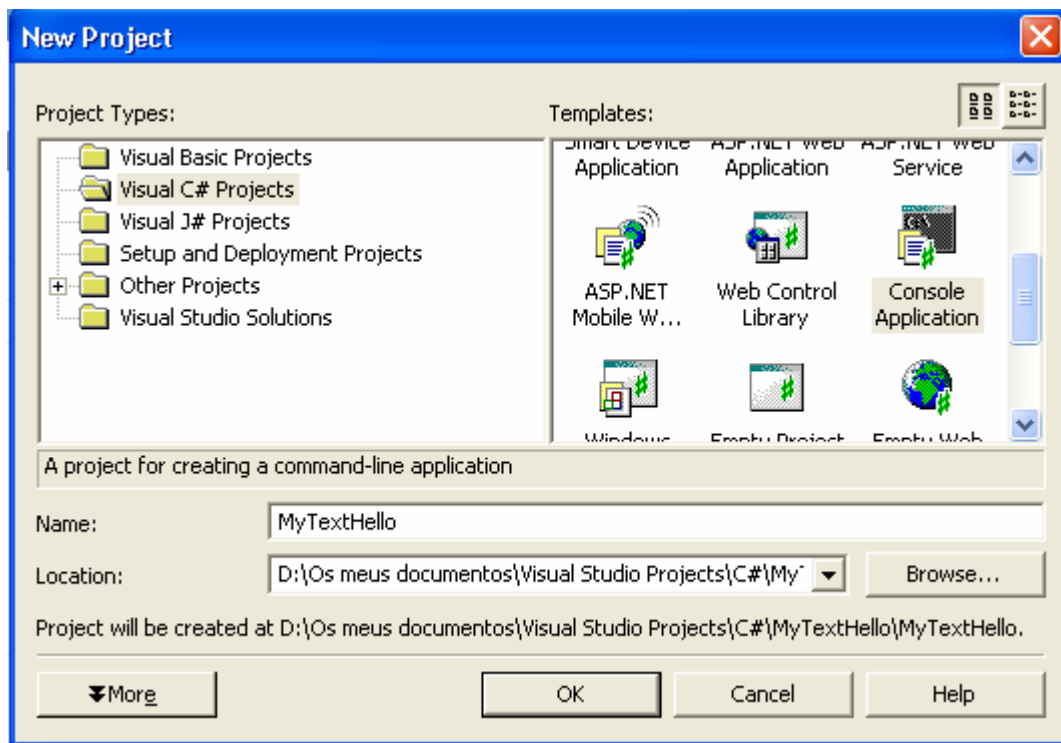
Aula 6 – C# Inicialização

O objectivo desta ficha, é familiarizar o aluno a utilização de *IDE* do *.NET*.

1. Execute o Microsoft Visual Studio .NET 2003.



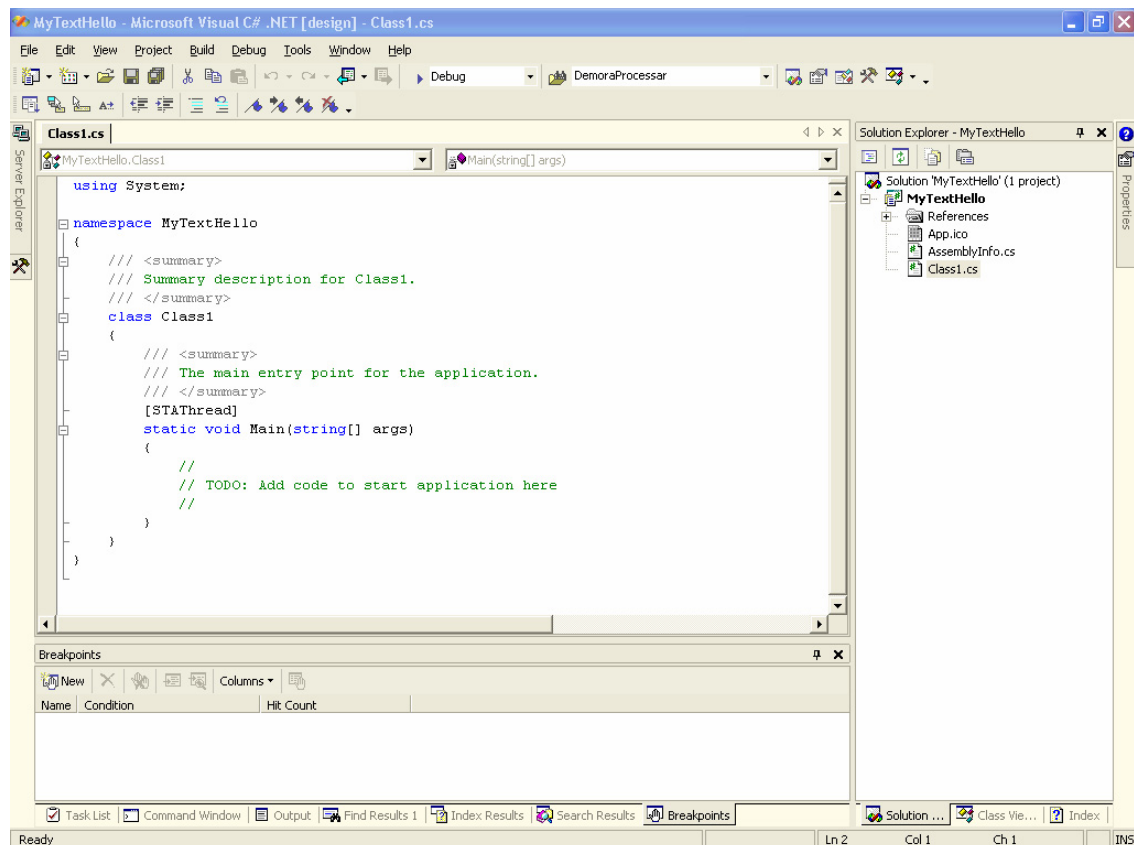
2. No menu File, seleccione a opção **New Project**. Surgirá uma caixa de diálogo, a qual permite especificar um novo projecto usando diferentes *templates*, tais como Aplicações pra Windows, Classes ou Aplicações de Consola.



Dica:

Crie cada projecto numa directoria diferente.

3. Seccione a opção “**Console Application**”. Atribua o nome MyTextHello ao ficheiro.



- **TextHello.sln**

Este é o ficheiro que identifica a solução ao mais alto nível. Só existe um por projecto. Cada ficheiro deste género possui um ou mais ficheiros de projecto. No explorador o nome do ficheiro é identificado com a sua extensão. No *Solution Explorer*, é apresentado de forma a ser mais perceptível a sua leitura, “Solution TextHello”.

- **TextHello.csproj**

Este ficheiro representa o projecto em C#. Cada projecto possui um ou mais ficheiros de código. Os ficheiros de código do projecto têm que ser obrigatoriamente todos escritos na mesma linguagem de programação. Este ficheiro é apresentado no *Solution Explorer* somente com o nome do projecto.

- **Class1.cs**

Este é um ficheiro de código do projecto.

- **AssemblyInfo.cs**

Este é outro género de ficheiro de código de C#. Através deste ficheiro é possível adicionar atributos ao nosso programa, tais como o nome do autor, a data de criação. Existem outros atributos mais avançados, que permitem modificar a forma de execução do programa.

- **App.ico**

Este ficheiro, representa o icon associado à nossa aplicação.

O MEU PRIMEIRO PROGRAMA

1. Remova os comentários **TODO**, assim como [STAThread]. O Main terá o seguinte aspecto:

```
static void Main(string[] args)
{
}

```

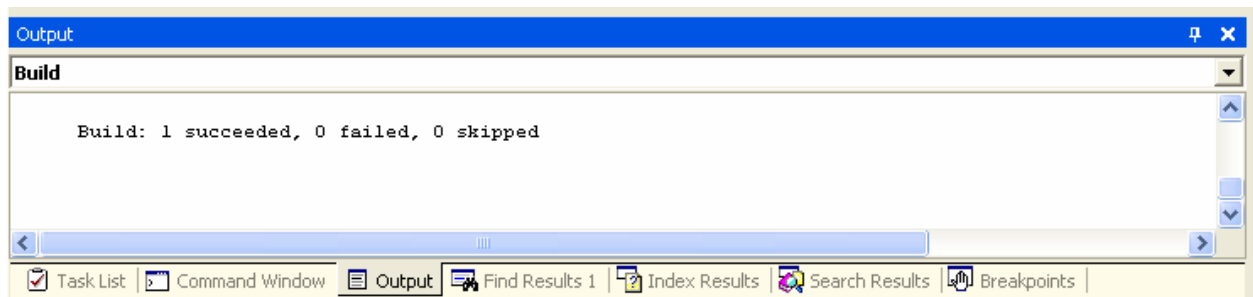
2. Adicione na linha em branco a palavra **Console**. (Console representa uma classe que contém métodos que permitem apresentar dados no ecrã e receber dados do teclado.
3. Coloque um ponto (.) a seguir à palavra. Uma lista *IntelliSense* irá aparecer. Esta lista contém todos os métodos, propriedades e atributos que a classe **Console** possui.

4. Selecciono o método **writeline** e especifique uma frase a apresentar no ecrã.

```
static void Main(string[] args)
{
    Console.WriteLine("Hello World");
}
```

COMPILAR E EXECUTAR A APLICAÇÃO

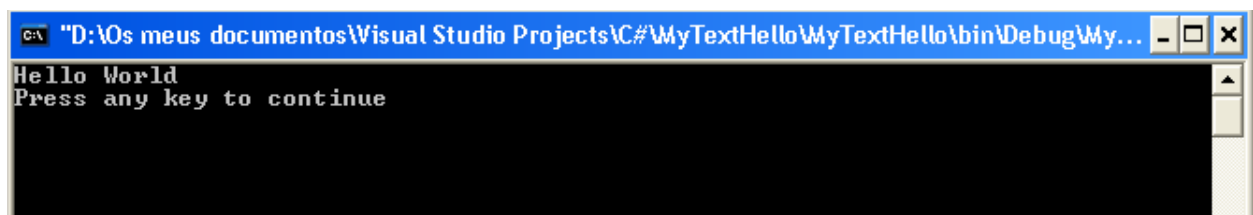
1. No menu **Build**, selecciono a opção **Build Solution**.



Dica:

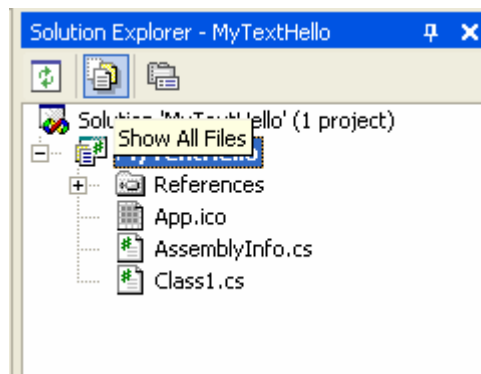
Um asterisco após o nome do ficheiro, significa que este foi modificado desde a última gravação. Não é necessário efectuar um *Save* antes de compilar, pois a opção **Build Solution** grava automaticamente.

2. No menu Debug, selecciono a opção **Start Without Debugging**. A aplicação será executada numa janela do DOS, apresentando a frase especificada por si.



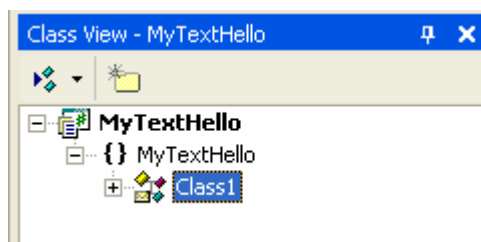
3. Após ter executado a aplicação. No *Solution Explorer* selecciono a opção **Show All Files**, tal como surge no exemplo seguinte.

Esta opção permite visualizar as directorias bin e obj do projecto. Estas directorias foram criadas aquando da compilação da aplicação, e contêm a versão executável do programa e outros ficheiros usados para efectuar *debug* ao programa.

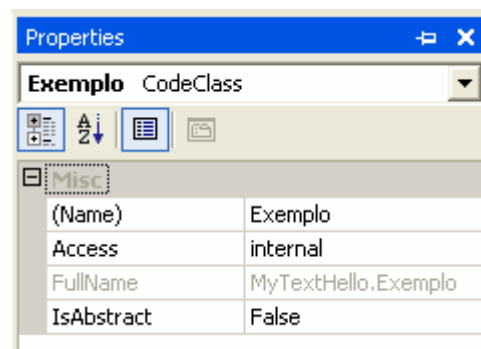


GERAR DOCUMENTAÇÃO XML ATRAVÉS DE COMENTÁRIOS

1. No menu **View**, selecione a opção **Class View**.



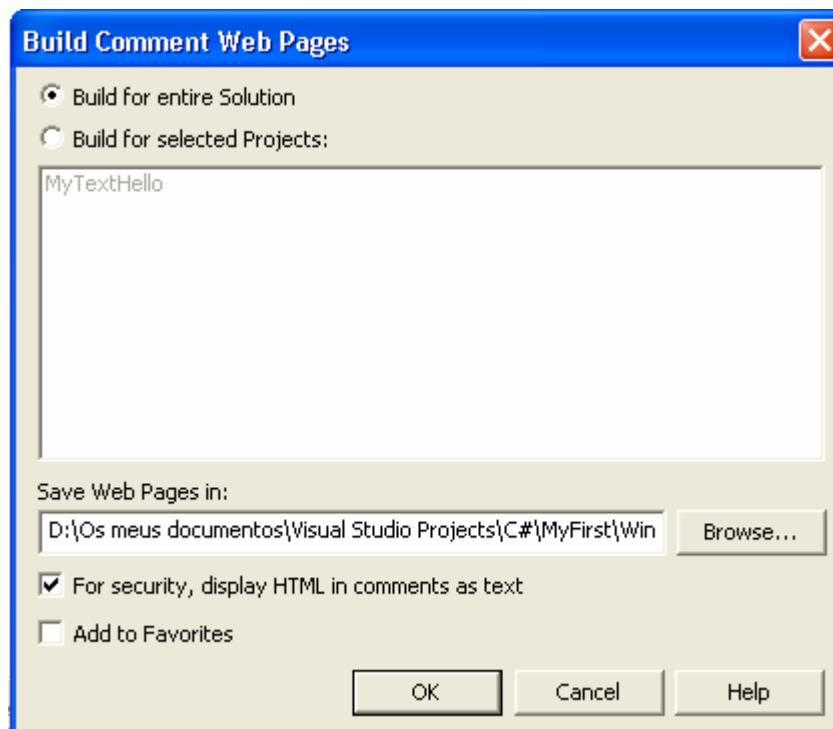
2. Ao pressionar o botão do direito do rato, surgirá um menu de **pop-up**, com diversas opções. Selecione a opção **Properties**.



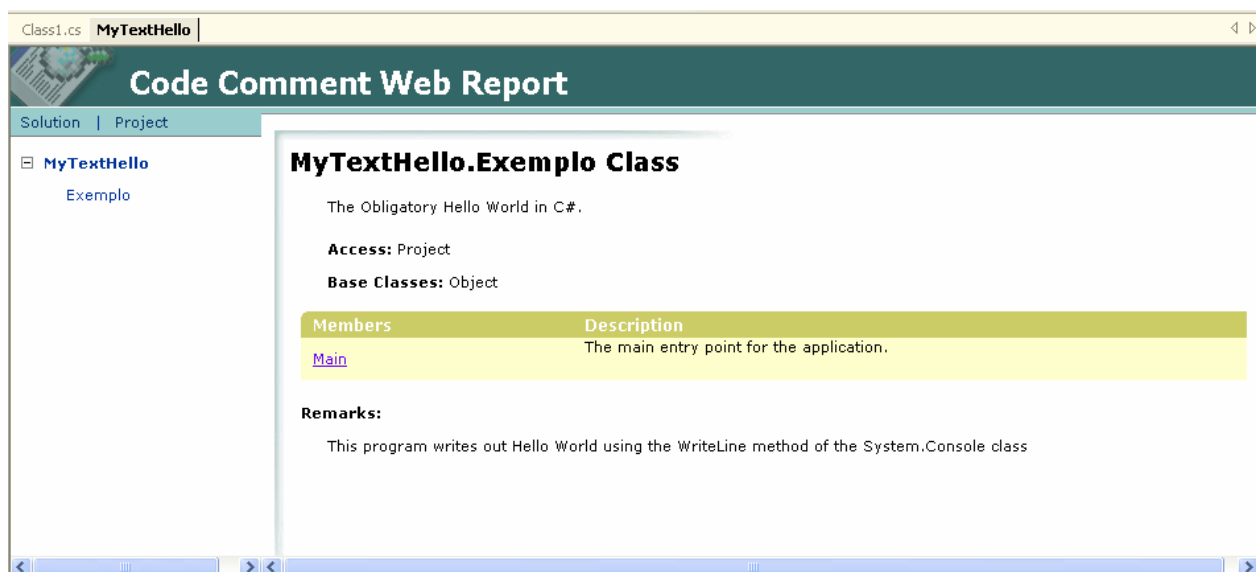
3. Modifique a propriedade **Name**. Como pode constatar, no código o nome da classe foi automaticamente modificada.
4. No código fonte, adicione os seguintes comentários, antes da definição da classe.

```
/// <summary>
/// The Obligatory Hello World in C#.
/// </summary>
/// <remarks>
/// This program writes out Hello World
/// using the WriteLine method of the
/// System.Console class
/// </remarks>
```

5. No menu **Tools**, selecione a opção **Build Comment Web Pages**.



6. Pressione o botão OK.



EXERCÍCIOS

1. Desenvolva uma aplicação que permita somar dois números. Ambos os números estão armazenados em variáveis previamente definidas por si.
2. Desenvolva uma aplicação que apresente no ecrã o resultado da tabuada do 9.
3. Desenvolva uma aplicação que permita ao utilizador introduzir dois números e especificar qual a operação aritmética que pretende efectuar.
Garanta que o programa não termina abruptamente em determinadas situações (como por exemplo divisão por zero). Implemente também código que permita ao utilizador ter conhecimento do erro que efectuou.
4. Desenvolva uma classe pessoa especificando os seus atributos mais importantes, como IDADE, MORADA, NOME, BI. Implemente os métodos necessários para manipular essa informação.
5. Desenvolva uma classe Empregado que derive da classe pessoa. Implemente os métodos necessários para manipular os atributos desta classe.
6. Desenvolva uma classe AVE que represente genericamente todas as aves. (NOME e capacidade de voar). Desenvolva outras duas classes, Avestruz e Pardal, que representam especificamente dois tipos de aves. **Atenção que a avestruz não voa.**
7. Desenvolva uma classe que ajude a testar as classes implementadas na alínea 6.