

CONHECENDO O VISUAL STUDIO .NET

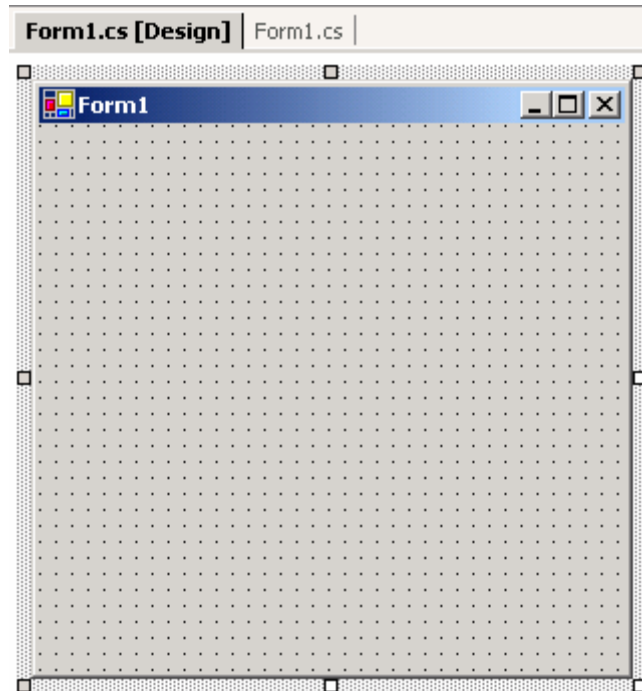
Nesta matéria estaremos apresentando a IDE (Integrated Development Environment) do Visual Studio .NET. A Microsoft incluiu nesta versão da ferramenta de desenvolvimento todos os conceitos que já vinha procurando aplicar e alguns novos conceitos visando o RAD (rapid application development).

A IDE é a mesma para todas as linguagens do Visual Studio .NET. Algumas modificações de valores para propriedades de projetos e configurações de build podem ocorrer, mas você contará com o mesmo arsenal de palhetas e janelas em qualquer linguagem que você estiver utilizando para desenvolver uma aplicação, ou seja, a mesma interface para projetos em Visual Basic ou C#.

Gerenciando o ambiente de desenvolvimento

O ambiente de desenvolvimento do Visual Studio .NET (IDE) é composto por inúmeras janelas ou caixas com recursos para auxiliá-lo no desenvolvimento rápido de aplicativos. Essas janelas se enquadram em duas categorias básicas: 'janelas de ferramentas' e 'janelas de documentos'. As janelas de ferramentas são listadas no menu View, sendo que muitas outras encontram-se no seu sub-menu Other Windows. Algumas dessas janelas de ferramentas são: Solution Explorer, Toolbox e Properties.

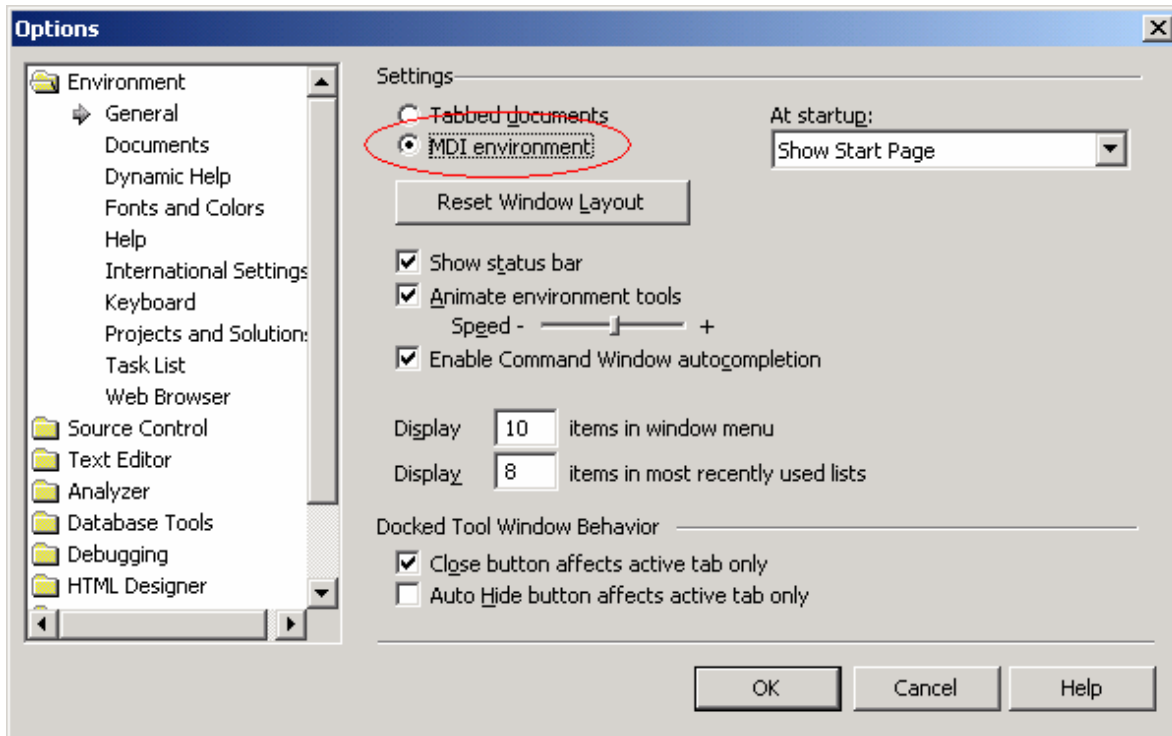
Já as janelas de documentos são criadas dinamicamente quando abrimos ou quando criamos arquivos ou outros itens. A lista de janelas de documento abertas aparece no menu Window na mesma ordem em que se encontram no ambiente de desenvolvimento, sendo a janela em primeiro plano na tela listada primeiro. Algumas janelas de documentos são o Code Editor e o Form Designer.



Na imagem anterior, a janela Form1.cs seria a janela de documento “Form Design” e a janela Form1.cs seria o “Code Editor”.


É interessante observar que você pode escolher entre dois modos de trabalhar com as janelas de documentos: o novo modo “com abas”, no qual as janelas exibem abas na parte superior para serem selecionadas (como na imagem anterior) e o modo MDI, antigo conhecido, no qual a janela em primeiro plano, quando maximizada, apresenta os botões Minimizar, Restaurar e Fechar no canto superior esquerdo do ambiente de desenvolvimento, logo abaixo da barra de títulos do mesmo.

Inicialmente, o Visual Studio .NET vem configurado para exibir as janelas no modo “com abas”, mas você pode mudar para o modo MDI selecionando a opção Options no menu Tools e definindo a opção MDI Environment na opção General da pasta Environment.

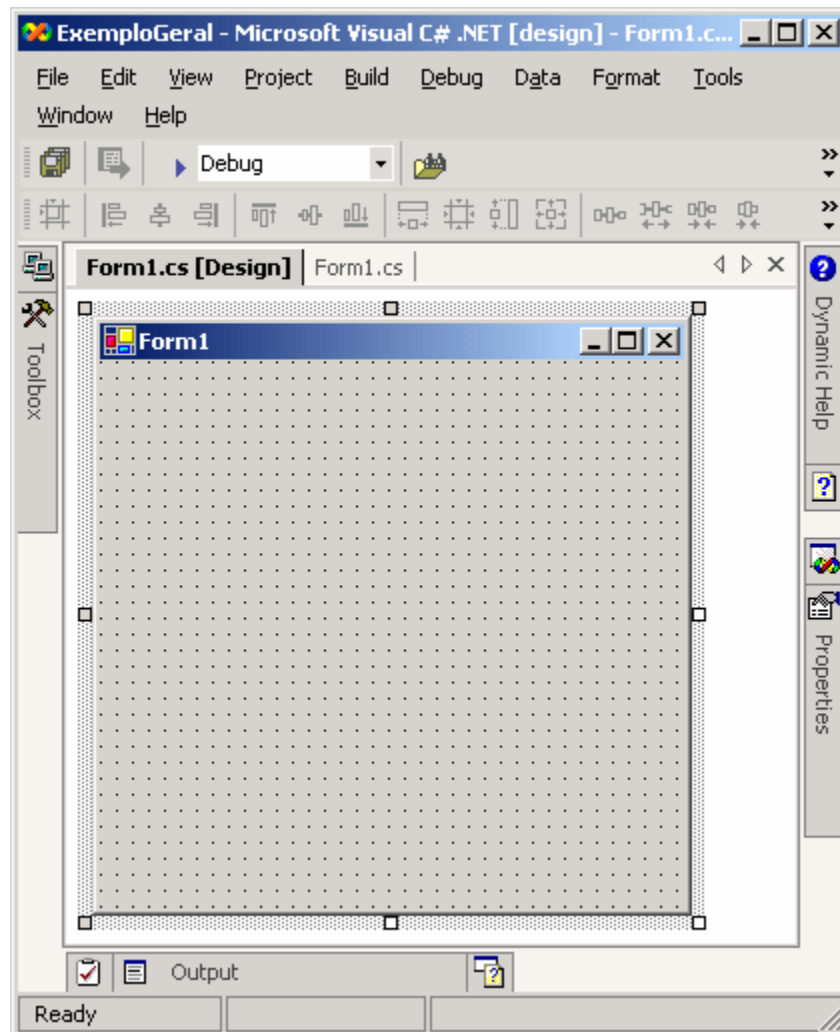


Como as janelas são muitas, foram implementados alguns excelentes recursos de gerenciamento de janelas. Por exemplo, as janelas de ferramentas podem ser arranjadas no IDE para aparecerem e desaparecerem automaticamente, ficarem ligadas a outras janelas de ferramentas com abas para seleção, “ancorarem” nas extremidades da IDE ou de outras janelas, flutuarem sobre o IDE, se comportarem como janelas-filha MDI ou até serem exibidas em outro monitor. Você pode posicionar essas janelas arrastando-as com o mouse e observando seu comportamento.

Todas as janelas de ferramentas do IDE podem assumir três tipos de comportamento: Docktable (ancorável), Floating (flutuante) e MDI (Multiple Document Interface – Interface de Documentos Múltiplos). Para definir qual deve ser o comportamento de uma determinada janela, basta selecioná-la e ativar ou desativar, no menu Window, opções Docktable e Floating. Com a opção Docktable ativa, a janela poderá ser ancorada na borda do IDE ou em outras janelas Docktable. Com a opção Floating ativa, a janela ficará flutuando sobre o ambiente, e com as opções Floating e Docktable desativadas, ela se comporta como uma janela-filha MDI, podendo ser arranjada lado a lado horizontal, ou verticalmente em relação as outras janelas MDI, minimizada ou maximizada dentro do IDE. Como se pode notar, há uma enorme flexibilidade no modo como podemos gerenciar nossas janelas. Mas isto não é tudo!

Quando a janela está definida como Docktable e ancorada no IDE, ela também se torna passível de ser Auto Hide, ou auto-ocultável. Você verá em sua barra de título um ícone de alfinete (). Este alfinete permite que você ative ou desative a ocultação automática da janela. Quando você

dá um clique no ícone do alfinete de uma janela que está em exibição, ela será escondida logo que você tirar o mouse de cima da mesma, ficando uma guia (ou paleta) na borda da tela com o nome da janela oculta. Passando o mouse sobre esta guia, a janela será exibida novamente e se você quiser deixá-la fixa na tela, basta dar um clique sobre o alfinete novamente.



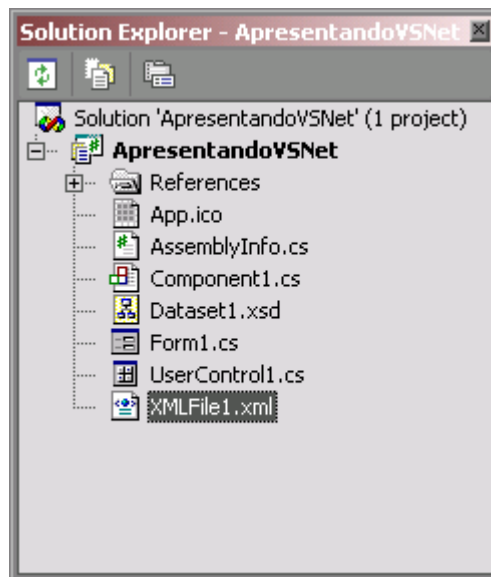
Qualquer que seja o modo de exibição de uma janela, você pode ocultá-la definitivamente dando um clique no “X” que existe no seu canto superior direito. Depois, para re-exibir a janela, você deve selecioná-la no menu View do IDE. Nos próximos tópicos conheceremos especificamente algumas dessas janelas.

Conhecendo as janelas do ambiente de desenvolvimento


Agora que você já sabe como gerenciar as janelas do IDE, vamos dar uma olhada mais de perto em algumas delas.


Janela Solution Explorer


Todos os projetos criados dentro do Visual Studio .NET ficam dentro de uma solução. O Solution Explorer é uma interface simples entre você e sua solução. Ele lhe dá informações em tempo real sobre itens da solução e permite administrar esses itens visualmente. Por exemplo, você pode monitorar os estados atuais de arquivos-fonte gerenciados, arrastar itens de um projeto para outro e visualizar as propriedades da solução, de um projeto ou de um arquivo. Muitas das tarefas mais comuns relacionadas com a manutenção de arquivos e projetos podem ser realizadas por meio do Solution Explorer.




O Solution Explorer tem uma barra de ferramentas que pode exibir alguns ícones de acordo com o item que você seleciona sua árvore. Por exemplo, se você seleciona um Form, tal como o Form1.cs do nosso projeto, serão exibidos cinco ícones com as seguintes finalidades.

 View Code: Abre o arquivo selecionado para edição no Code Editor;

 View Designer: Abre o arquivo no Forms Designer para visualização e alteração;

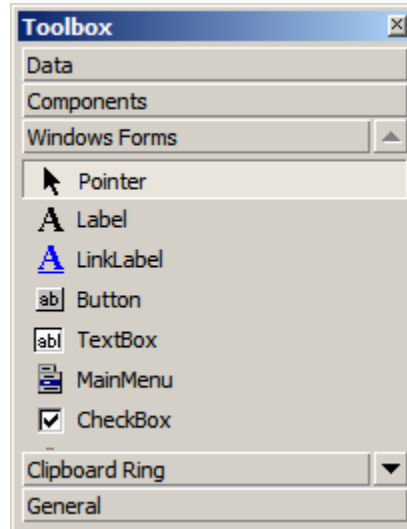
 Refresh: Atualiza a exibição do item selecionado no Solution Explorer. Este botão também pega a última versão de um arquivo que está sendo usado por outros programadores em modo compartilhado de rede.

 Show All Files: Exibe todas as pastas e arquivos da Solução, incluindo aqueles que foram excluídos do projeto atual e os que estão ocultos.

 Properties: Exibe na janela Properties as propriedades do item selecionado no Solution Explorer

Janela Toolbox

A janela Toolbox (ou “Caixa de Ferramentas”) pode ser aberta pelo menu View > Toolbox. É ela que comporta os controles (componentes) que podem ser inseridos nos projetos. Esta janela contém várias abas que podem ser estendidas com um clique do mouse. Outras abas também podem aparecer dinamicamente de acordo com o tipo de projeto em que você está trabalhando.

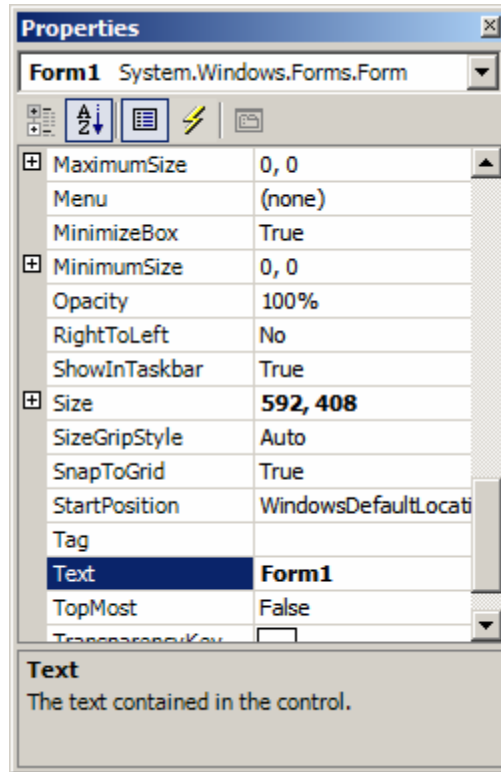


Para utilizar esta janela, basta clicar no componente desejado e clicar depois em algum lugar dentro do form. Desta forma se o componente for visual, aparecerá na região do form que acabou de clicar.

Janela Properties

A janela Properties (ou “Propriedades”) pode ser acessada por meio do menu View > Properties Window. Algumas propriedades possuem o símbolo de mais (“+”), pois estão agrupadas em categorias. Clicando sobre este sinal você pode acessar a lista de propriedades que compõem estas categorias.

Quando você escolhe um componente e coloca-o em seu form, é nesta janela que você deve configurar os valores iniciais das propriedades deste componente. Por exemplo, o próprio form que você utiliza obrigatoriamente em um projeto do tipo WinForms, se você deseja alterar o título do mesmo, basta ir na janela properties e alterar a propriedade “Text”, colocando o título desejado. Veja a imagem a seguir:

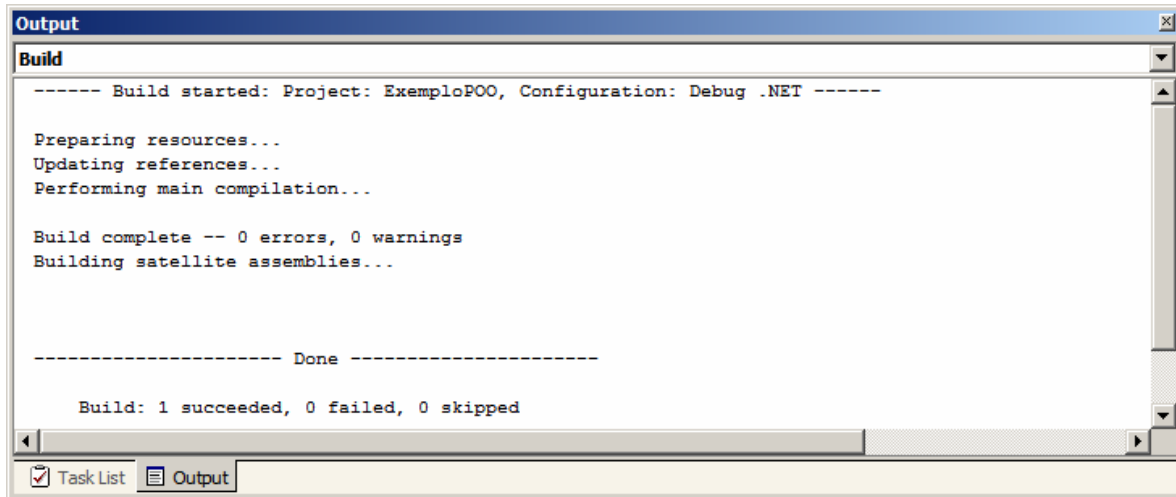


Alterando a propriedade “Text” conforme sugere a imagem, a palavra Form1 não mais irá aparecer no título do seu form.

Janela OutPut

A janela OutPut (ou “Saída de Informações”) é acessível pelo menu View > Other Windows > OutPut. Como o próprio nome diz, esta janela exibe as saídas de informações do ambiente de desenvolvimento do Visual Studio .NET.

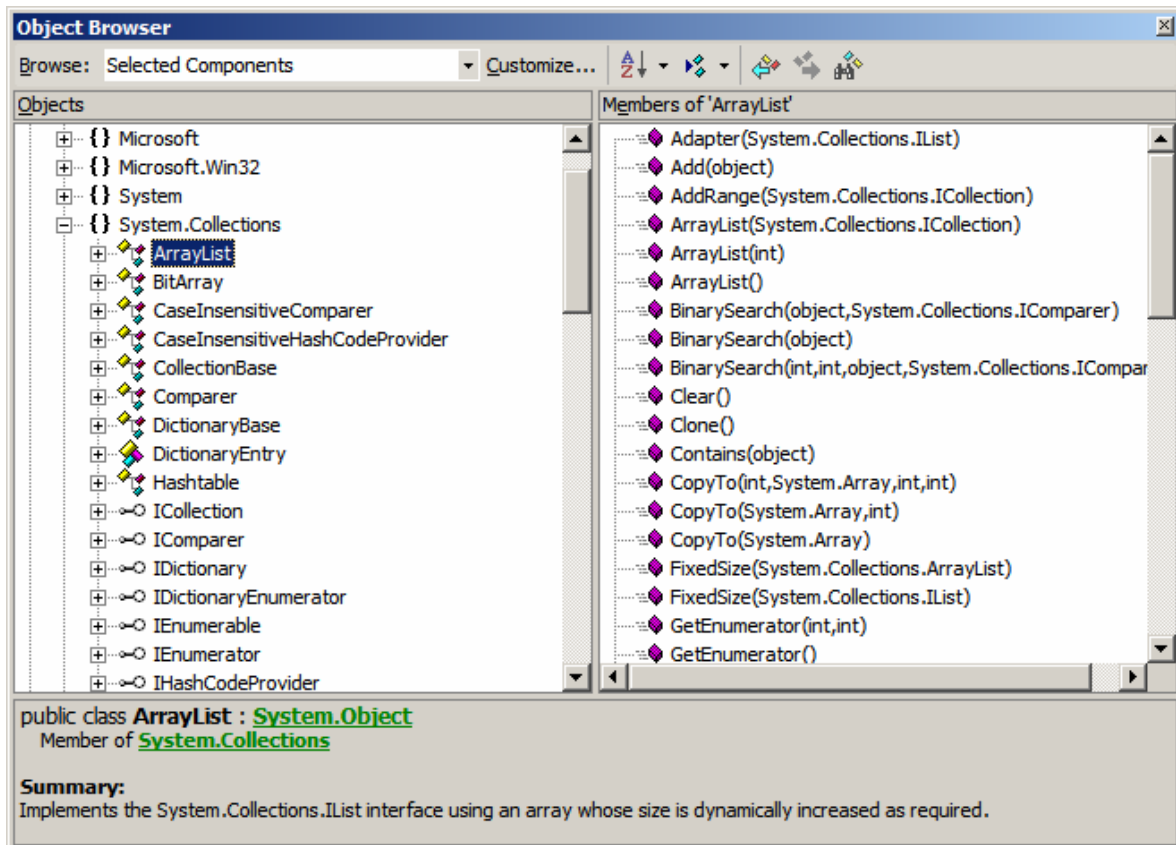
Durante o processo de compilação, por exemplo, o Visual Studio.NET coloca informações de cada etapa do procedimento que está sendo executado. Você também pode forçar uma saída de informações nesta janela utilizando o método WriteLine() do objeto Debug.



Janela Object Browser

A janela Object Browser (ou “Navegador de Objetos”) é aberta pelo menu View > Other Windows > Object Browser. Ela apresenta a estrutura hierárquica de cada objeto disponível, bem como a lista de membros pertencentes a cada objeto selecionado. Os objetos são apresentados na caixa Objects em uma estrutura de árvore, com “nós” que podem ser expandidos ou compactados por meio de um clique em seu sinal de adição (+) ou subtração (-). Isso nos dá uma visão bem ampla dos recursos contidos em cada objeto.

Devido ao suporte aprimorado da orientação a objetos, a janela Object Browser também foi melhorada para mostrar todos os detalhes dos objetos. Selecionando um objeto na caixa Objects desta janela, você poderá ver os membros desse objeto na caixa Members e uma descrição do mesmo na parte inferior da janela. É interessante também que nesta janela você pode ver o nome da assembly (DLL) onde estão contidos certos recursos, e até o caminho dessa assembly em seu computador.

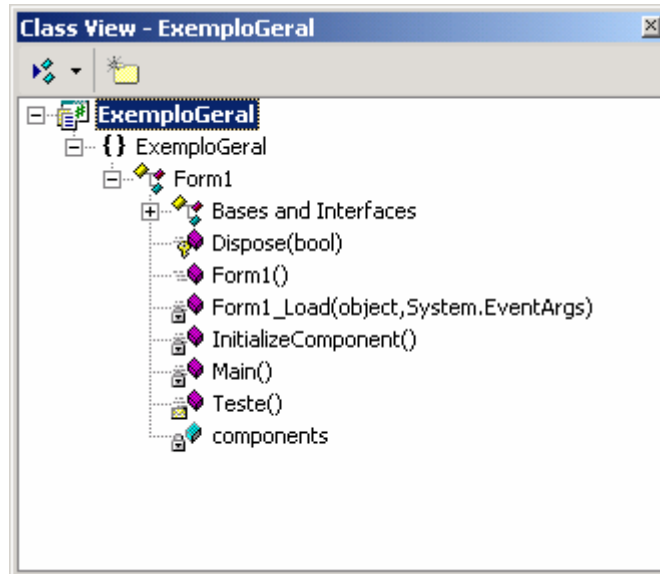


Janela Class View

A janela Class View (ou “Visualizador de Classes”) é mais uma implementação visando facilitar o trabalho com o modelo de programação orientada a objetos e pode ser acessada através do menu View > Class View.

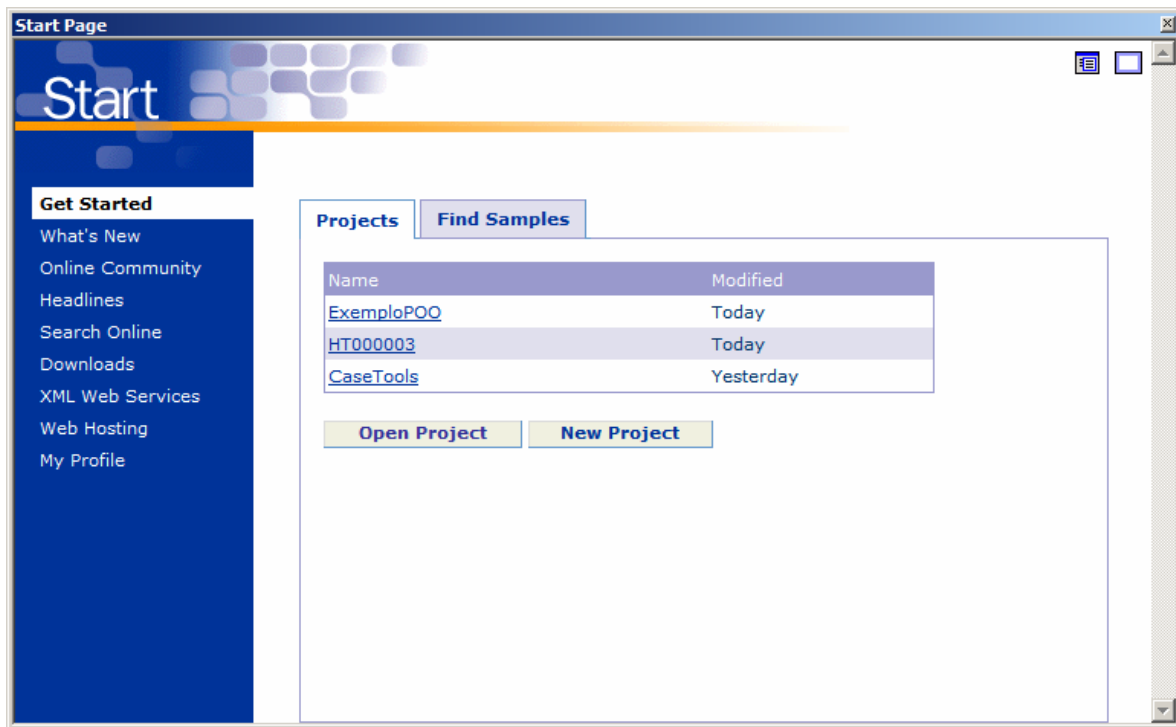
Por meio desta janela, você pode ver toda a estrutura de classes existente em seu projeto. Clicando duas vezes com o mouse em cima de método (item representado por uma caixa em rosa), o code editor é aberto automaticamente e o cursor de edição de texto é disponibilizado na primeira linha do código do método.

Conforme você vai notar na imagem seguinte, os métodos públicos são representados pela imagem de uma caixa rosa, os métodos privados são representados pela caixa rosa e um cadeado, os métodos protegidos são representados por uma caixa rosa e uma chave e finalmente os métodos internos a classe são representados por uma caixa rosa com um envelope de carta em amarelo.

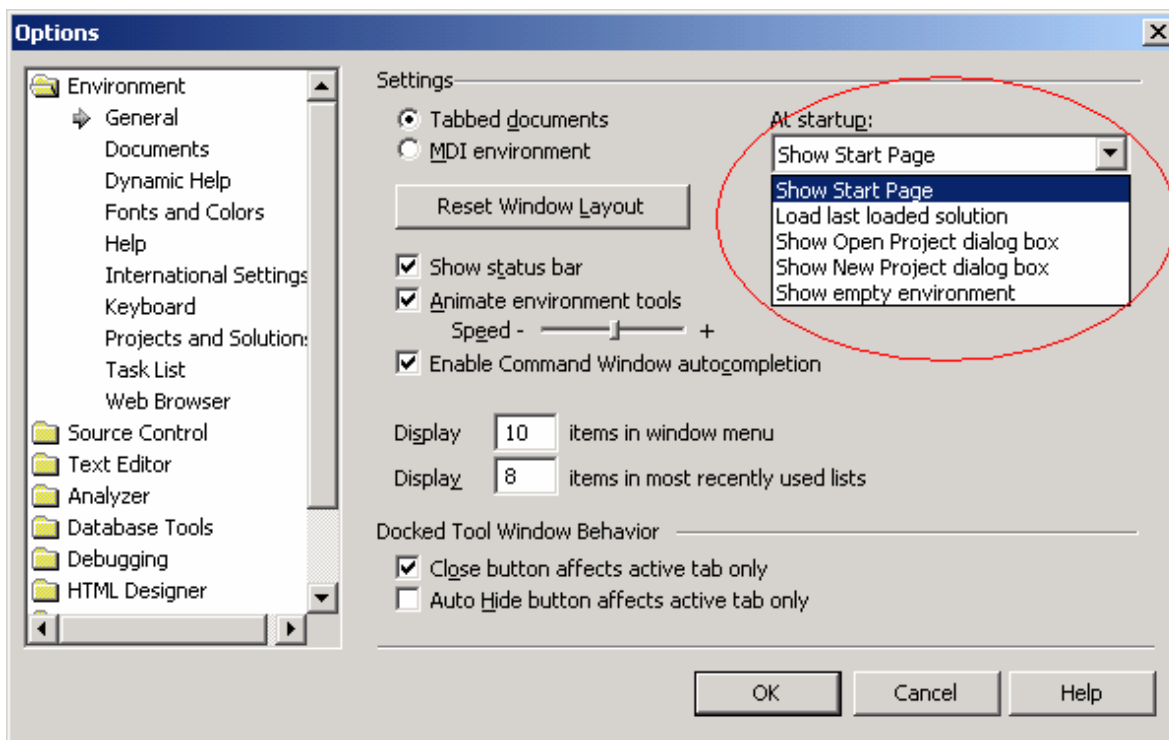


Janela Web Browser

A janela Web Browser (ou “Navegador de Internet”) é um navegador Web disponível dentro do IDE do Visual Studio .NET. Normalmente, quando você inicia o Visual Studio .NET, a janela Web Browser aparece exibindo a sua página inicial, a Visual Studio Start Page. Toda vez que uma janela Web Browser é exibida, a barra de ferramentas Web também aparece permitindo acesso rápido aos recursos de navegação pela Internet. Para acessá-la, você deve apontar o menu View > Web Browser > Show Browser.



Você pode configurar qual página deseja que apareça inicialmente quando o Visual Studio.NET é ativado no menu Tools > Options > Environment > General, conforme a imagem a seguir.



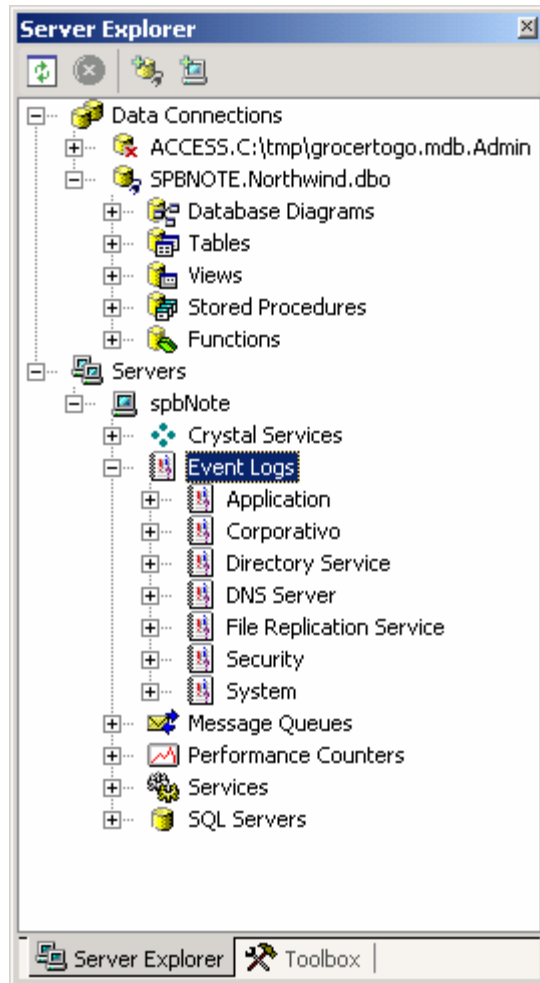
Janela Favorites

A janela Favorites (ou “Favoritos”) exibe a lista dos sites apontados nos favoritos do seu navegador Internet Explorer, permitindo que você adicione novos endereços a essa lista, ou até a organize. Essas opções são disponibilizadas por meio de View > Other Windows > Favorites.

Janela Server Explorer

O Server Explorer (ou “Explorador de Servidores”) é um gerenciador de servidores do Visual Studio .NET. Ele apresenta uma lista com as conexões de dados e todos os servidores disponíveis em sua rede. Com ele você pode examinar e manipular os servidores de bancos de dados, bem como os dados que estes contêm. Você pode, por exemplo, usá-lo para se conectar a uma base de dados para um Form de sua aplicação, para um diagrama de banco de dados ou para uma consulta SQL.

A janela Server Explorer pode ser acessada pelo menu View > Server Explorer.



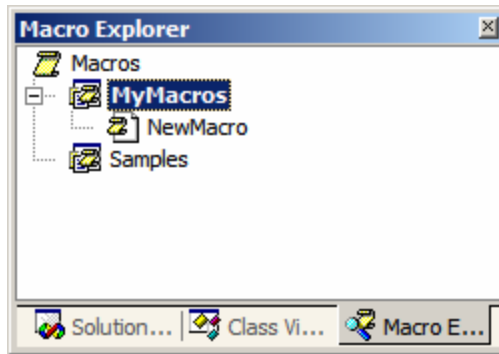
Além de permitir facilmente a visualização de tabelas, views e stored procedures, você também tem acesso ao event logs (do Event Viewer), a filas MSMQ, Services e ao Performance Monitor entre outros.

Nota do Autor: Estes itens só poderam ser visualizados se você estiver desenvolvendo em uma máquina instalada com o sistema operacional Windows 2000 Server ou Advanced Server.

Macro Explorer

O Visual Studio .NET possui agora suporte a macros, para agilizar a execução de tarefas repetitivas, como ocorre com aplicativos como o Microsoft Word e Excel.

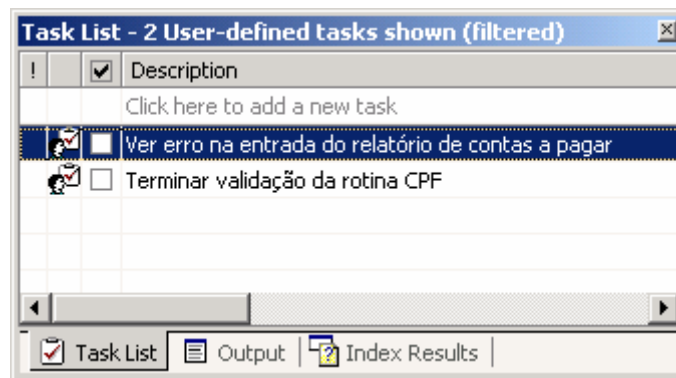
A janela Macro Explorer (ou “Explorador de Macros”) lista todas as macros disponíveis no ambiente e é sua ferramenta principal para ver e executar macros. Para acessar esta janela você deve apontar para View > Other Windows > MacroExplorer.



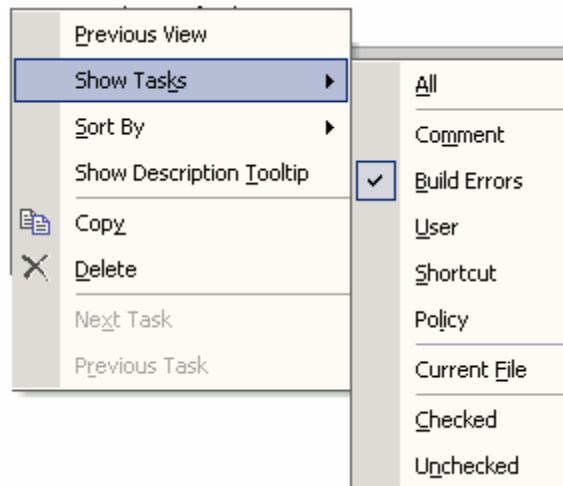
Task List

A janela Task List (ou “Lista de Tarefas”) ajuda você a organizar e administrar o trabalho de construir sua solução. Excelente hábito para desenvolvimentos muito extensos, aonde você deixa algumas tarefas para serem ajustadas depois com o objetivo de não atrasar o cronograma.

Para exibi-la selecione a opção Show Tasks no menu View e então escolha a categoria da mensagem que deseja ver, ou selecione Other Windows no menu View e escolha Task List.



Você também pode configurar qual o tipo de informação que deseja apresentar nesta janela. Para fazer isto, clique com o botão direito do mouse e selecione no menu que irá surgir o tipo de informação desejada. Na imagem anterior de exemplo, estamos visualizando “User defined tasks”. A opção default desta janela é “Build Errors”.



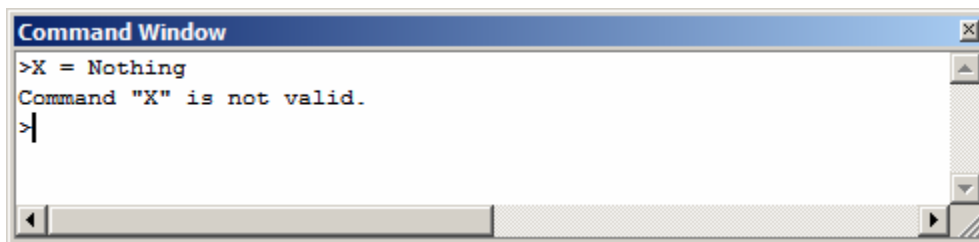
Command Window

A janela Command Window (ou “Janela de Comando”) tem dois modos diferentes: Comando e Imediato.

O modo Comando é usado para executar rapidamente comandos diretamente no ambiente Visual Studio .NET, evitando o sistema de menus, ou para executar comandos que não aparecem em qualquer menu.

Já o Modo Imediato é usado para propósitos de depuração e é igual ao das versões anteriores do Visual Studio.

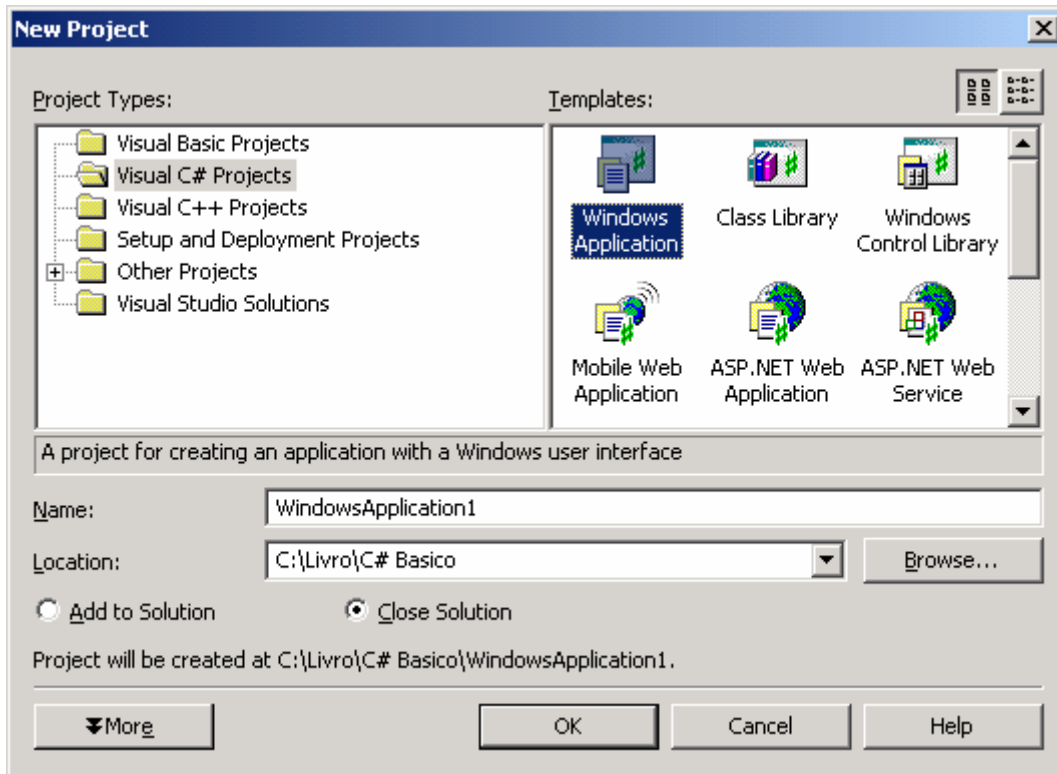
Para acessar esta janela, aponte o menu View > Other Windows > Command Window.



Criando Aplicações baseados em Templates

O Visual Studio .NET possui dezenas de templates que facilitam muito no desenvolvimento de aplicativos. Para acessá-los, clique sobre o menu File > New > Project.

Por exemplo, se você deseja construir uma aplicação do tipo WindowsForm, o Visual Studio.NET parte do pressuposto que pelo menos um form deverá existir em sua solução e ele traz isto definido para você. Você não precisa se preocupar em adicionar o Main form ou escrever a estrutura de seu código (como a namespace e a classe principal que vai conter o método estático Main());).



Como pode observar na imagem anterior, existem vários templates disponíveis nas linguagens que você possui em seu Visual Studio.NET. Imagine a facilidade de criar um Web Services ou Mobile Web App através destes templates. Basta selecioná-lo e toda a tarefa “pesada” como declarações e diretivas de compilação específicas já estão prontas.

Crie um projeto qualquer e note que existe uma grande quantidade de código já escrito bem como indicações (em comentários) de como o código pode ser colocado.

Esta é uma funcionalidade que promove um grande ganho de produtividade para seu desenvolvimento.

Ocultamento do Texto

Este novo recurso nos permite ter uma visão hierárquica e resumida de todo nosso código. Consiste numa funcionalidade do code editor em colocar uma linha vertical do lado esquerdo do código e, em pontos específicos, como no início de funções, classes, etc., inserir um sinal de subtração (ou adição), indicando que aquela linha pode ser expandida ou ocultada.

É possível também ocultar e re-exibir qualquer bloco de código por meio das opções disponíveis no sub-menu Outlining do menu Edit.

```

namespace ApresentandoVSNet
{
    public class MainClass
    {
        public MainClass()
        {
            //TODO: Implementar o código do construtor
        }

        Região de Declaração de variáveis fechada

        #region Região de Declaração de variáveis aberta

        private string nome;
        private int idade;
        private decimal salario;

        #endregion
    }
}

```

Com este recurso novo você pode muito bem minimizar regiões do texto que não estão sendo trabalhadas em um determinado momento e se preocupar com o que realmente é relevante para o momento. Para os desenvolvedores que conhecem XML, é inevitável a comparação.

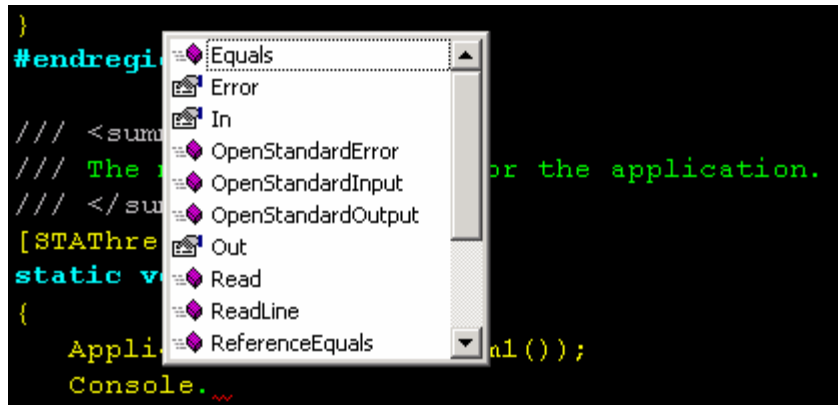
É, sem dúvida, um diferencial impressionante que fará você optar pelo code editor do Visual Studio.NET em comparação com qualquer outro editor de textos.

Intellisense

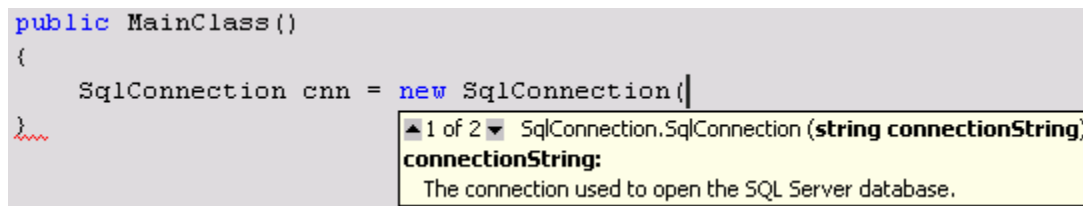
O Intellisense é uma engenhosidade que lhe fornece uma lista de facilidades enquanto digita o código fonte no code editor do Visual Studio.NET.

O Intellisense fornece as seguintes funcionalidades:

List Members (Lista de membros): Quando você digita o nome de um objeto seguido de um ponto, o Intellisense abre uma caixa de listagem com todos os membros deste objeto (propriedades, métodos, eventos, variáveis, etc) e você poderá escolher aquele que deseja usar. Auxilia na fixação dos nomes dos membros;



Parameter Info (Informações de parâmetros): Quando você digita o nome de uma função e abre parenteses, o Intellisense exibe rapidamente a lista de sobrecargas do método e a lista de parâmetros que cada sobrecarga necessita;



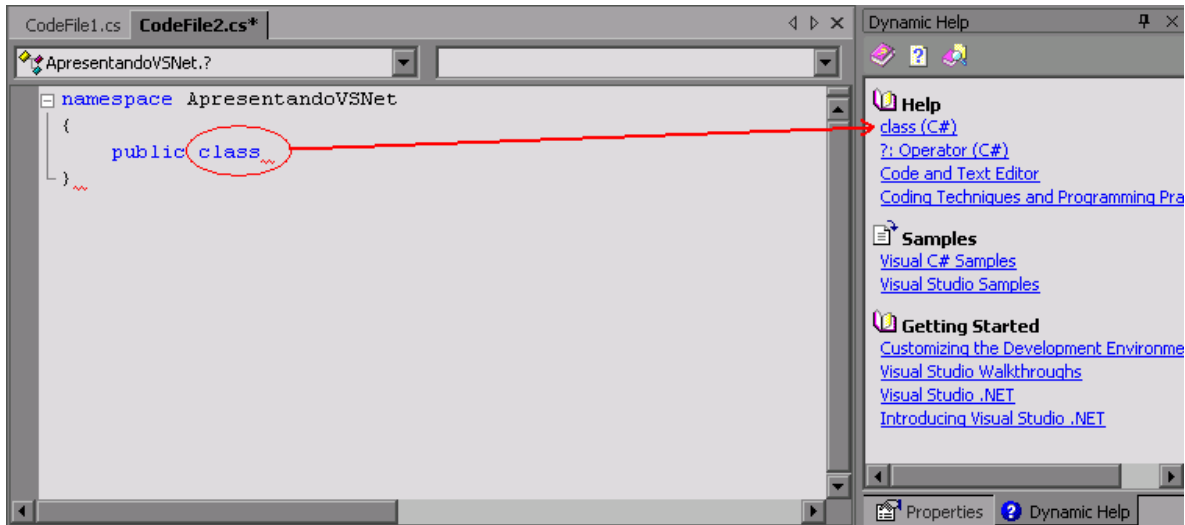
Quick Info (Informações Rápidas): Quando você passa o mouse sobre um especificador o Intellisense exibe a declaração completa deste especificador;

Complete Word (Completar palavra): Quando você está digitando o nome de um membro e pressiona CTRL + ESPAÇO, o Intellisense completa a palavra (nome do membro) ou exibe uma lista de membros com nomes que iniciam com as letras já digitadas.

Obtendo Ajuda

O sistema de ajuda do Visual Studio.NET é a referência mais completa sobre a ferramenta, seguida pelos manuais do usuário cedidos juntos com o sistema.

Uma das funcionalidades mais interessantes que o Visual Studio .NET possui é o *Dynamic Help*, que serve como uma espécie de ajudante do programador. Sua tarefa é buscar informações e referências sobre o objeto que você está utilizando dinamicamente enquanto esta digitando o código.



O Help tradicional ativado a partir da tecla F1 também está presente na ferramenta de desenvolvimento.

Quando você pressiona a tecla F1, o Visual Studio.NET não procura uma combinação exata da palavra embaixo do cursor na lista Help Search. Em vez disso, ele tenta adivinhar o que você está precisando. Por exemplo, se você pressionar F1 quando o cursor de texto estiver sobre a palavra `button1` no código fonte, o sistema Help do Visual Studio.NET abrirá automaticamente a descrição da classe `button`, uma vez que é isto o que provavelmente você está procurando.

É interessante notar que não existe apenas um único arquivo Help no Visual Studio.NET. Na maioria das vezes, você invocará o Help do Visual Studio.NET, mas este arquivo é complementado pelo Help do framework.

Estes arquivos de Help e outros relacionados com ferramentas externas do Visual Studio.NET possuem um visual e um mecanismo de procura que você pode ativar pressionando o botão Help Topics enquanto estiver no sistema de Help.