

O ambiente de desenvolvimento Visual C# enquadra-se na categoria de ferramentas do tipo **RAD - Rapid Application Development** (Desenvolvimento Rápido de Aplicações). Devido a esta característica, muitos programadores acabam por deixar de lado princípios básicos utilizados nas técnicas de programação, tais como algoritmos, diagramas de blocos, *layouts* de tela, documentação de código-fonte, documentação de variáveis, entre outros, afirmando que com o uso desse tipo de ferramenta os procedimentos de documentação não são mais necessários. Isso não é verdade, pois é exatamente agora que esses conceitos passaram a ser mais do que necessários.

## 8.1 Formulário e Controles Essenciais

A partir do conhecimento dos passos anteriores, é possível agora preocupar-se apenas com a parte gráfica.

Execute o comando **File/New Project**, selecione na caixa de diálogo **New Project** a opção **Windows Form Application**, informe no campo **Name** o nome **Cap08\_Ex01** e acione o botão **OK**.

No lado esquerdo da tela, próximo ao nome do formulário selecione a opção **Toolbox**, como mostra a Figura 8.1. Na sequência, com o cursor posicionado na barra de título denominada **Toolbox**, acione o botão direito do *mouse* e selecione no menu de contexto apresentado a opção **Dock**, para que as ferramentas **Toolbox** sejam posicionadas ao lado esquerdo do formulário, como indica a Figura 8.2.

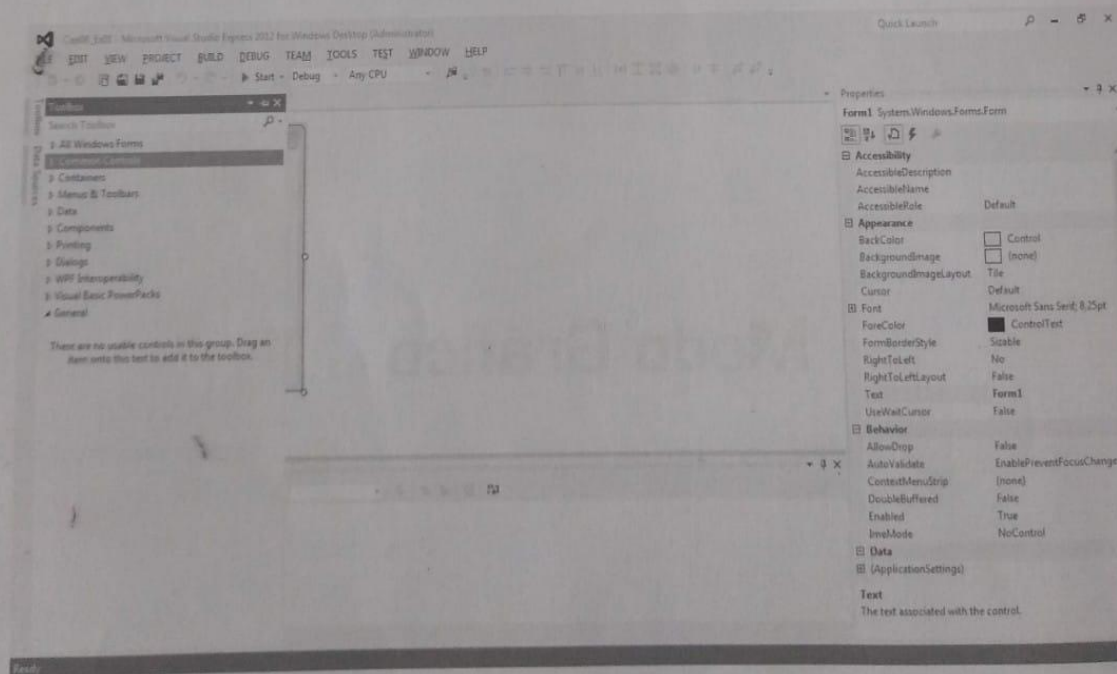


Figura 8.1 - Apresentação das ferramentas Toolbox.

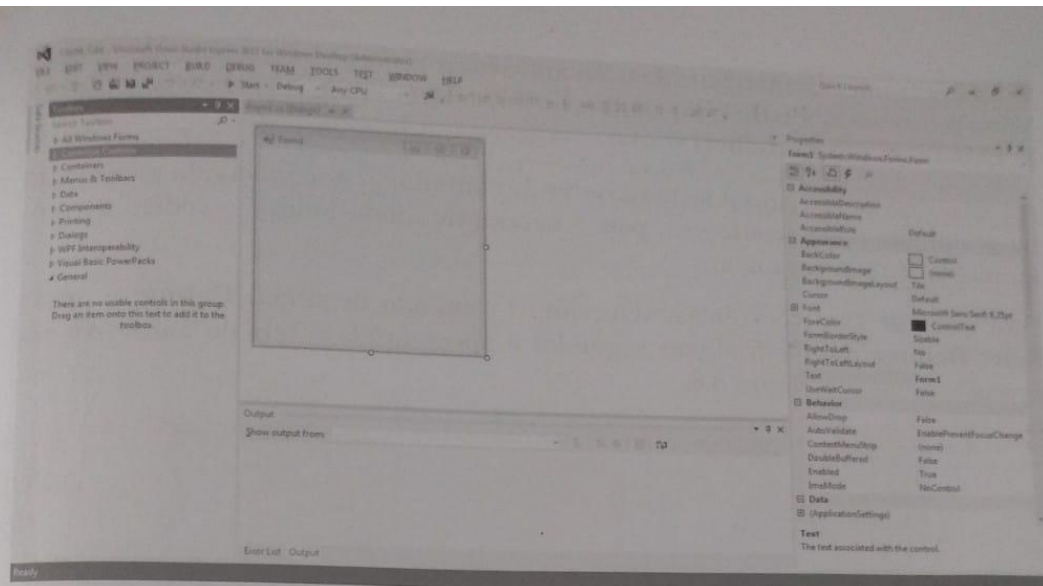


Figura 8.2 - Ferramentas Toolbox reposicionadas.

Na lista de ferramentas **Toolbox** selecione a guia **All Windows Forms** para que seja apresentado o conjunto completo dos controles visuais. Selecione na lista o controle **TextBox** que será usado para efetuar a entrada de um determinado dado. Para selecioná-lo, basta um clique no controle desejado e levá-lo para dentro do formulário **Form1**, o qual será fixado por meio de um clique dentro da área pretendida. Insira o controle, arrastando-o para o canto superior esquerdo e ajuste seu tamanho, como mostra a Figura 8.3, e coloque também um controle **Button** abaixo do controle anterior. A Figura 8.4 mostra como deve ficar a sua área de trabalho (desktop).

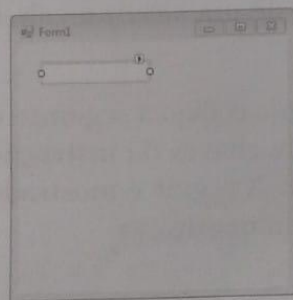


Figura 8.3 - Controle TextBox inserido no formulário Form1.

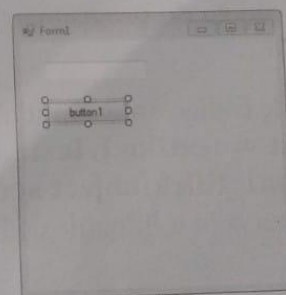


Figura 8.4 - Controle Button inserido no formulário Form1.

Foram escolhidos dois controles como mecanismo de entrada e de ação para apresentar o nome informado.

Falta ainda um terceiro controle para apresentar o nome informado. Neste caso, selecione o controle **Label** inserindo-o no formulário. A Figura 8.5 exibe o controle **Label** com o nome **Label1**.

A partir desse momento o formulário está montado. Se for solicitada a execução do programa, nada acontecerá, pois é necessário ainda indicar o código de programa de seu processamento.

Para verificar essa ocorrência, selecione o comando de menu **Debug**, a opção **Start Debugging**, e em alguns segundos é apresentada a tela do aplicativo em execução, como na Figura 8.6.

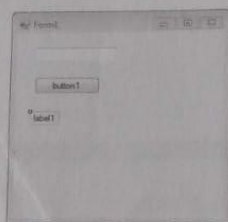


Figura 8.5 - Controle Label1 inserido no formulário Form1.

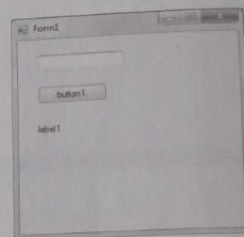


Figura 8.6 - Tela do programa.

Não se preocupe com a apresentação da tela. Para sair do formulário, dê um clique com o ponteiro do mouse no botão com o **X**.

Um programa em modo gráfico desenvolvido em Visual C# é a junção de um formulário, no caso **Form1**, com um código de programa existente que será projetado e inserido em local apropriado no ambiente de desenvolvimento.

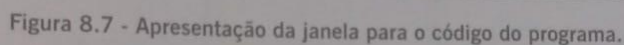
A partir do formulário ativo, dê um duplo clique no controle **button1**, e surge a janela de código do programa com a apresentação de um *namespace* aberto, como na Figura 8.7.

Na janela de código apresentada escreva como linha de código a seguinte instrução: **label1.Text = textBox1.Text;**, entre os símbolos de chaves da instrução **private void button1\_Click(object sender, EventArgs e)**. A seguir é mostrado um trecho de rotina com a linha de código a ser inserida em negrito:

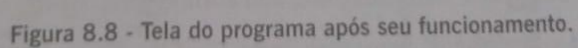
```
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = textBox1.Text;
}
```

Perceba que o conteúdo do controle **label1** vai ser implicado no conteúdo digitado e armazenado no controle **textBox1**.





Nesse momento escreva um nome qualquer no campo **text1**. Após escrever o nome desejado, dê um clique com o ponteiro do *mouse* no botão **button1**, e o nome será apresentado no controle **label1**. A Figura 8.8 mostra como deve ficar a tela após um teste.



Daqui em diante, o processo de montagem de um programa dentro de um formulário fica mais simples. Basta que sejam seguidas as instruções anteriores.

Execute o comando **File/Save All** para gravar o projeto do programa.

O próximo exemplo é um programa que pede o nome e o sobrenome de uma pessoa, apresentando em seguida o nome completo. A junção de *strings*, como será mostrada, recebe o nome de concatenação.

Execute o comando **File/New Project**, selecione na caixa de diálogo **New Project** a opção **Windows Form Application**, informe no campo **Name** o nome **Cap08\_Ex02** e acione o botão **OK**.

A partir do momento em que a tela é apresentada com um formulário, será selecionado na ferramenta **Toolbox** o componente de controle **Label**, inserindo-o no formulário, como na Figura 8.9.

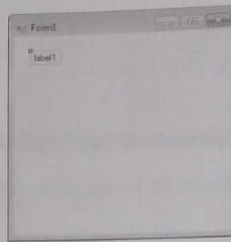


Figura 8.9 - Inserção de um controle Label.

Em seguida, será trocado o seu texto de **Label1** por **Escreva seu nome:**. Selecione na janela **Properties** a propriedade **Text** que se encontra com o rótulo **label1** e faça a alteração para **Escreva seu nome:**. Caso não esteja vendo essa janela, execute o comando de menu **View/Other Windows/Properties Windows**. A janela **Properties** fica posicionada ao lado direito inferior da tela.

As Figuras 8.10 e 8.11 apresentam a propriedade **Text** antes e depois da alteração. Preste atenção na mudança do texto dentro do formulário.

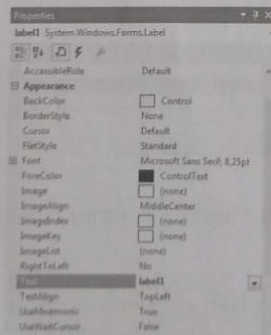


Figura 8.10 - Propriedade Text antes da alteração do texto.

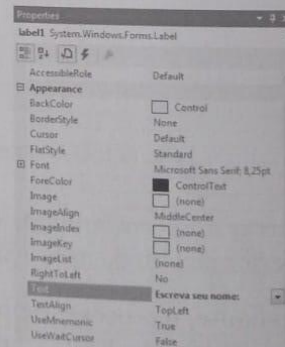


Figura 8.11 - Propriedade Text depois da alteração do texto.

Agora, procedendo da mesma forma, insira um segundo componente **Label**, logo abaixo do primeiro, cuja mensagem apresentada seja **Digite seu sobrenome:**. A Figura 8.12 indica como deve ficar o formulário.

Ao lado de cada **Label** é colocado um controle **TextBox**, para aceitar a digitação das informações, como indicado na Figura 8.13.

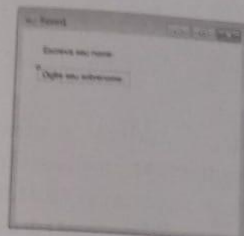


Figura 8.12 - Inserção de um segundo label.

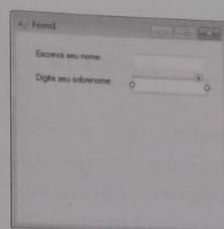


Figura 8.13 - Inserção dos controles TextBox.

Abaixo dos controles **Label** e **TextBox** são colocados dois botões obtidos a partir do componente **Button**. O primeiro deve ter sua propriedade **Text** alterada para **Concatenar**. O segundo botão tem sua propriedade **Text** alterada para **Limpar**.

A Figura 8.14 mostra os botões inseridos no formulário.

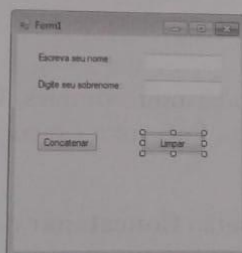


Figura 8.14 - Botões inseridos no formulário.

O primeiro botão tem a função de apresentar o nome completo, enquanto o segundo remove os *strings* informados nos componentes **TextBox**.

Falta ainda definir um terceiro **Label** que será usado pelo botão **Concatenar** para apresentar o nome completo. Coloque esse novo **Label** na parte de baixo do formulário. A Figura 8.15 apresenta uma ideia desta sugestão.

A partir do momento que o formulário está preparado, será definido o código de ação para a execução do botão **Concatenar**. Dê um duplo clique no botão **Concatenar** e escreva na rotina o código seguinte grafado em negrito:

```
label3.Text = textBox1.Text + textBox2.Text;
```



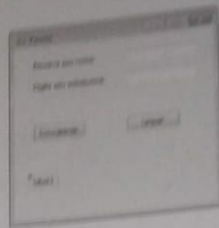


Figura 8.15 - Label para apresentação do nome completo.

Nesse momento, já é possível executar o programa e visualizar o efeito da concatenação dos dois *strings*. Após utilizar o botão **Concatenar**, nota-se que a saída do nome completo não ficou muito boa, pois é necessária a apresentação de um espaço em branco entre o nome e o sobrenome. A Figura 8.16 mostra este efeito.

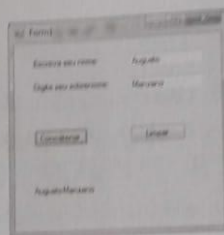


Figura 8.16 - Concatenação de dois strings.

A correção deste detalhe é algo bastante simples. Basta concatenar no meio dos dois *strings* um espaço em branco. Encerre a execução do programa acionando na janela Form1 o botão **X**.

Em seguida altere o código do botão **Concatenar** definido como:

```
label3.Text = textBox1.Text + textBox2.Text;
```

Para o código:

```
label3.Text = textBox1.Text + " " + textBox2.Text;
```

Após proceder à alteração proposta, rode novamente o programa e certifique-se de que está funcionando corretamente.

A Figura 8.17 apresenta a saída após a alteração anterior.

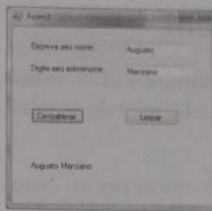


Figura 8.17 - Concatenação dos strings apresentados corretamente.

Em seguida será codificado o botão **Limpar**. Feche a execução do programa **Form1**, selecione a guia **Form1.cs (Design)** e dê um duplo clique no referido botão. Digite na rotina apresentada o seguinte código grafado em negrito:

```
textBox1.Clear();
textBox2.Clear();
label3.Text = "";
```

Observe que nos componentes **TextBox1** e **TextBox2** é utilizado o método **Clear()** para proceder à limpeza dos dados existentes nos componentes. Já o **Label3** possui um *string* sem conteúdo, em branco, para efetivar a limpeza do componente.

Execute o programa, entre nome e sobrenome acione o botão **Concatenar** e em seguida acione o botão **Limpar**. Os campos foram limpos, porém o cursor não voltou para o primeiro campo. Esse tipo de ocorrência pode causar alguns transtornos para quem for usar o programa.

Encerre o programa, abra a janela **Form1.cs (Design)**, dê um duplo clique no botão **Limpar** e acrescente a linha de código indicada a seguir em negrito:

```
private void button2_Click(object sender, EventArgs e)
{
    textBox1.Clear();
    textBox2.Clear();
    label3.Text = "";
    textBox1.Focus();
}
```

A instrução **textBox1.Focus();** faz com que o cursor seja posicionado no campo **TextBox1** após a ação do botão **Limpar**. Desta forma, é possível manipular o foco para qualquer objeto dentro do formulário. Execute o programa e veja como ficou bastante adequado o seu funcionamento.

Execute o comando **File/Save All** para gravar o projeto do programa.