



Habilitação Profissional Técnica de Nível Médio de **TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

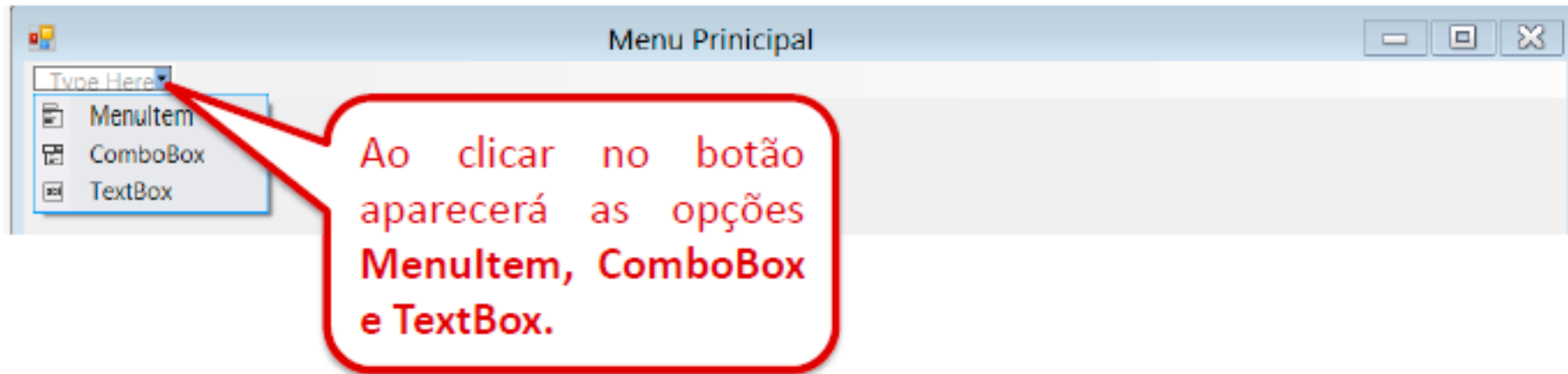
DESENVOLVIMENTO DE SISTEMAS

MenuStrip



MenuStrip

A função do **menustrip**, é disponibilizar um menu horizontal composto por vários componentes (opções). Estão disponíveis no menustrip 3 tipos de componentes:

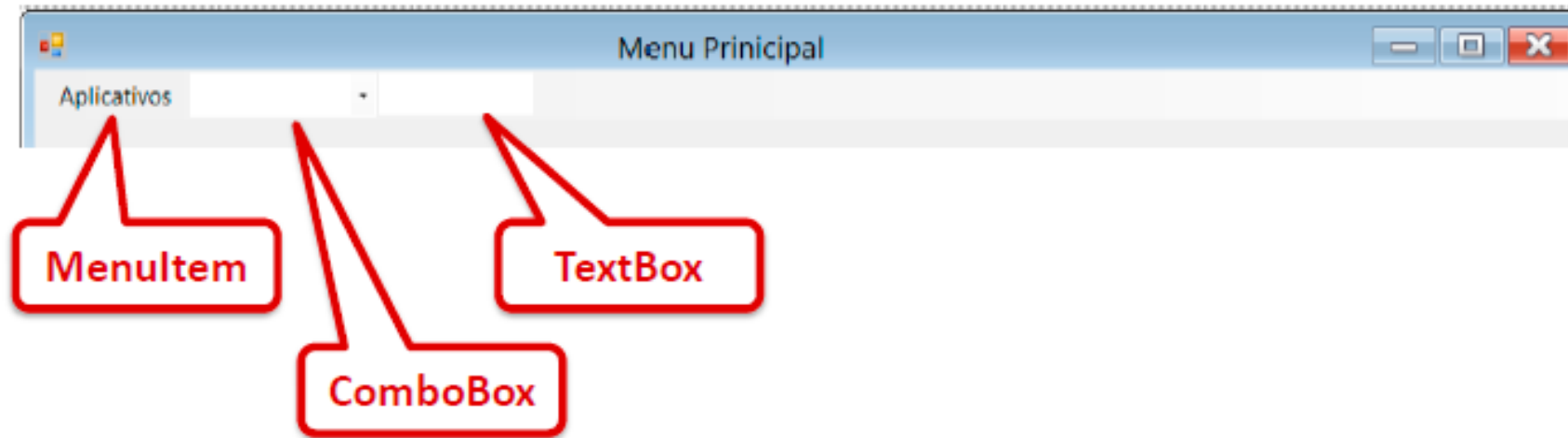


Opções do MenuStrip:

- ✓ **MenuItem** → Serve para inserir uma opção de texto.
- ✓ **ComboBox** → Serve para inserir uma opção no estilo de Caixa de Itens.
- ✓ **TextBox** → Serve para inserir uma opção de caixa de texto.

MenuStrip

Abaixo os modelos das opções de menu:



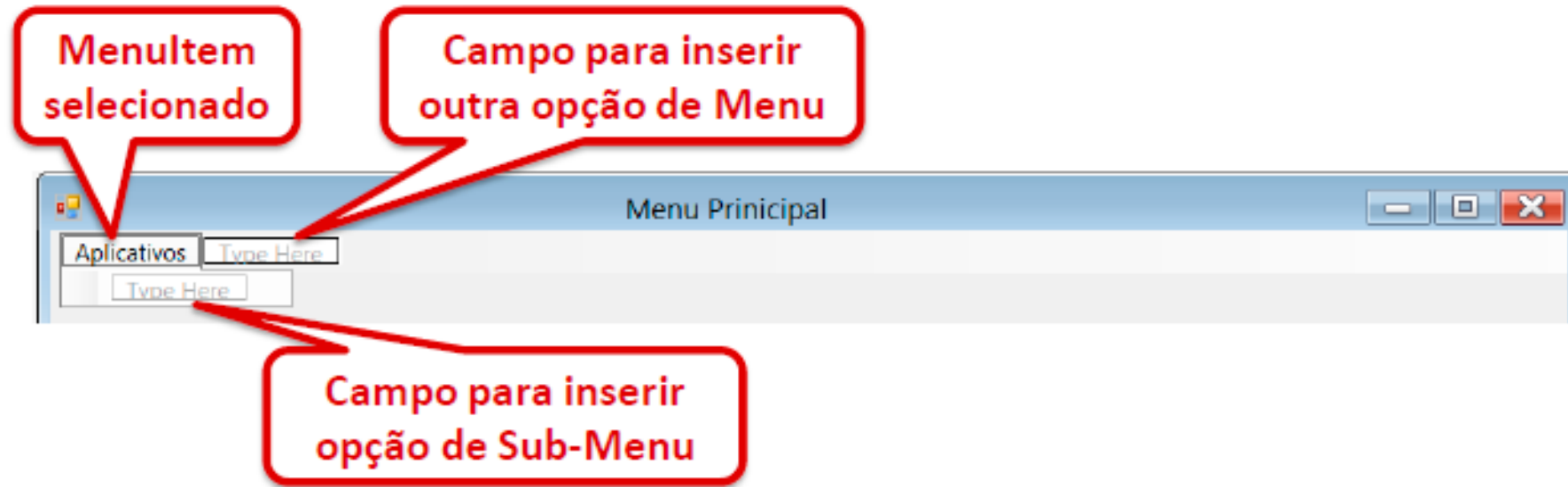
Os sub-menus só podem existir se associado a um **MenuItem**, qualquer outro tipo utilizado não será possível adicionar sub-menus.

Quando criamos um MenuItem podemos associar uma letra como tecla de atalho da opção, através do carácter **&** precedendo sempre a letra de atalho. Por exemplo, na opção **Aplicativos** iremos colocar como tecla de atalho a letra "A", então devemos digitar na propriedade Text, o seguinte texto: **&Aplicativos**.



MenuStrip

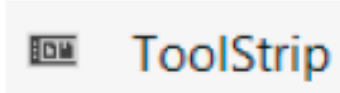
Para adicionar **sub-menus** devemos clicar sobre a opção (**MenuItem**) desejado e automaticamente aparecerá os campos para inseri-los, como exemplo abaixo:



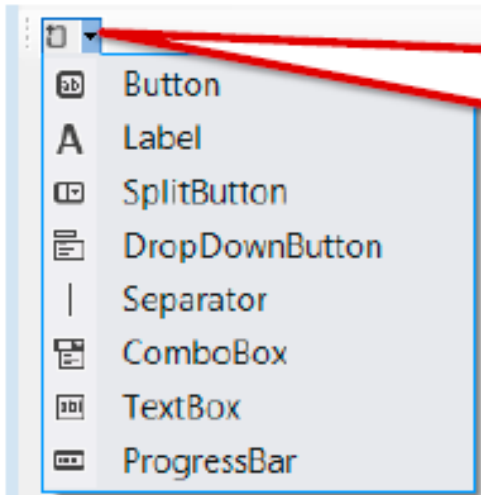
O evento default para o MenuItem é o **Click**, ou seja, quando selecionamos um item, será executado o código de programação que estiver dentro.

```
private void wordToolStripMenuItem_Click(object sender, EventArgs e)
{
}
}
```

ToolStrip

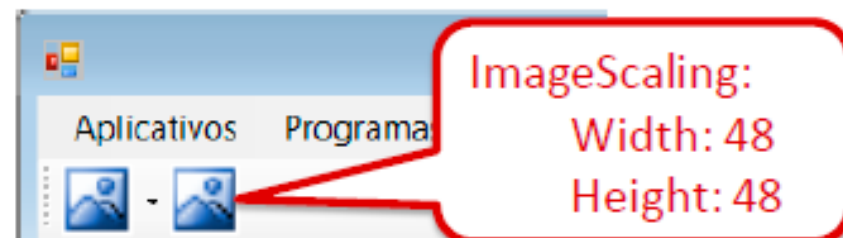


A função do **toolstrip**, é disponibilizar um menu horizontal, de **acesso rápido (Atalhos)**, composto por vários componentes (opções). Estão disponíveis no toolstrip 8 tipos de componentes:



Ao clicar no botão aparecerá as opções **Button, Label, SplitButton, DropDownButton, Separator, ComboBox, TextBox e ProgressBar.**

Para que alterar os tamanhos dos componentes na ToolStrip, devemos modificar a propriedade **ImageScalingSize**. Por default (padrão), estão atribuídos os valores, **width=16 e height=16**. Para que ocorra essa mudança deve-se alterar antes da inserção dos componentes.



ToolStrip

Opções do ToolStrip:

- ✓ **Button** → Serve para inserir um botão com imagem.
- ✓ **Label** → Serve para inserir um texto fixo.
- ✓ **SplitButton** → Serve para inserir um botão, mas só executará o evento Click no momento que selecionar o componente, e não o botão do mesmo.
- ✓ **DropDownButton** → Serve para inserir um botão, mas só executará o evento Click no momento que selecionar o componente ou o botão do mesmo.
- ✓ **Separator** → Serve para criar divisões de componentes. Por exemplo: Opções de Edição com as opções de Ferramentas, etc.
- ✓ **ComboBox** → Serve para inserir uma opção no estilo de Caixa de Itens.
- ✓ **TextBox** → Serve para inserir uma opção de caixa de texto.
- ✓ **ProgressBar** → Serve para inserir uma barra de progressão (termômetro).

SplitButton x DropDownButton

Diferença entre o SplitButton e DropDownButton.

Aparente os dois componentes são iguais, mas as funcionalidades são diferentes, como segue a explicação abaixo:



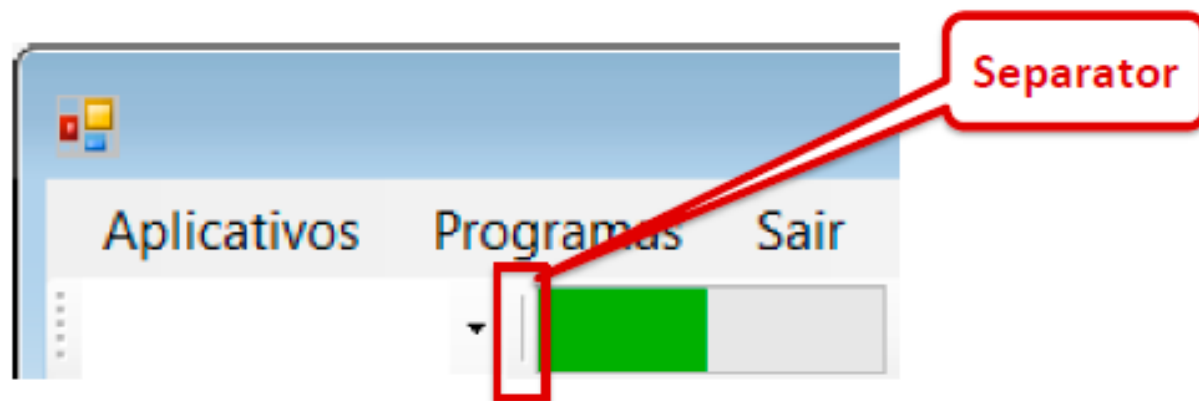
ToolStrip – ProgressBar e Separator

O componente progressbar serve para exibir a execução de algum processo em segundo plano. Por exemplo, ao carregar um arquivo texto muito grande poderíamos exibir a carga dos dados através deste componente, mostrando quantos de dados já foram carregados.

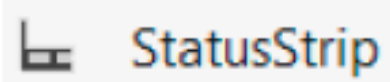


A propriedade responsável pelo nível de progresso do componente é a **Value**. Pode-se utilizar tanto o incremento como o decremento da propriedade Value.

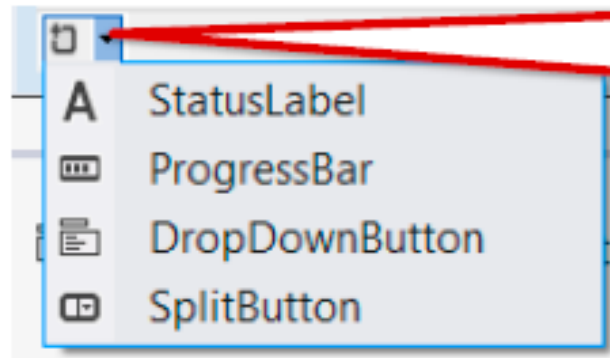
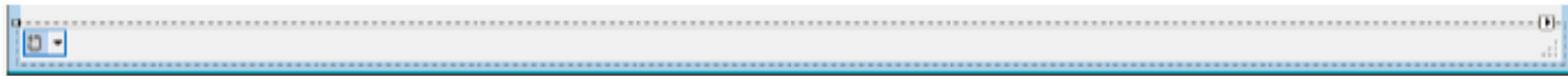
Para indicar qual será o limite inferior e o superior do componente, utilizaremos as propriedades **Minimum** e **Maximum**.



StatusStrip



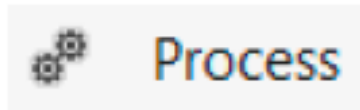
A função do **statusstrip**, é disponibilizar uma barra horizontal de informações no rodapé do formulário. É composto por vários 4 tipos de componentes:



Ao clicar no botão aparecerá as opções **StatusLabel, ProgressBar, DropDownButton e SplitButton.**

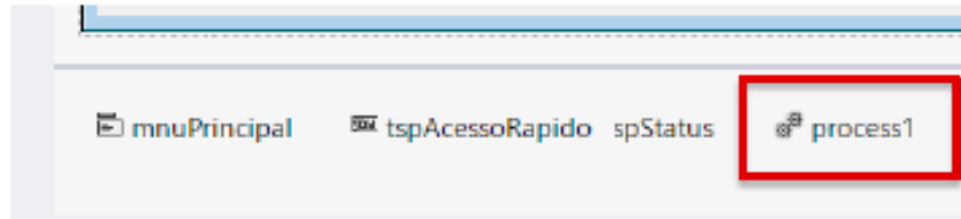
Este componente é muito utilizado para exibir informações a respeito das opções de menu, data do sistema, hora do sistema, etc.

Process



A função do **process**, é executar um processo local, ou seja, uma aplicação externa. Por exemplo, a calculadora do windows, o Word, o Excel, um sistema externo, etc.

Este componente não é visual, mas ficará disponível para todo o formulário.



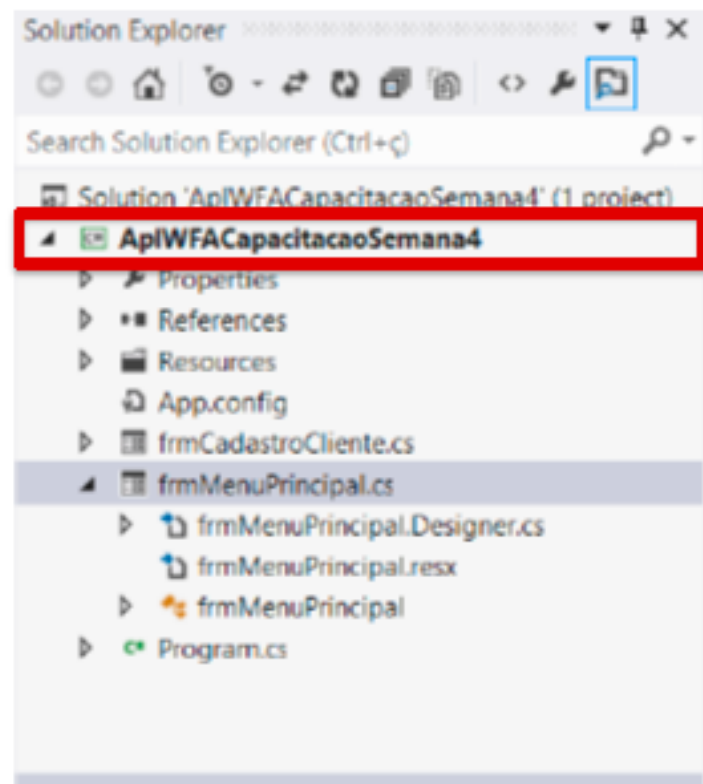
Antes de executar uma aplicação externa devemos configurar o componente para a determinada aplicação. Para isto, utilizaremos as seguintes propriedade e método:

- ✓ **StartInfo.FileName** → Especificar o nome da aplicação externa, junto com o **Path** da mesma.
- ✓ **Start()** → Executar a aplicação configurada na propriedade **FileName**.

Criar Formulário em um Projeto

Como estamos trabalhando com menus, a partir de agora podemos criar um ou mais formulários dentro de um mesmo projeto, desse modo começaremos a desenvolver um sistema integrado.

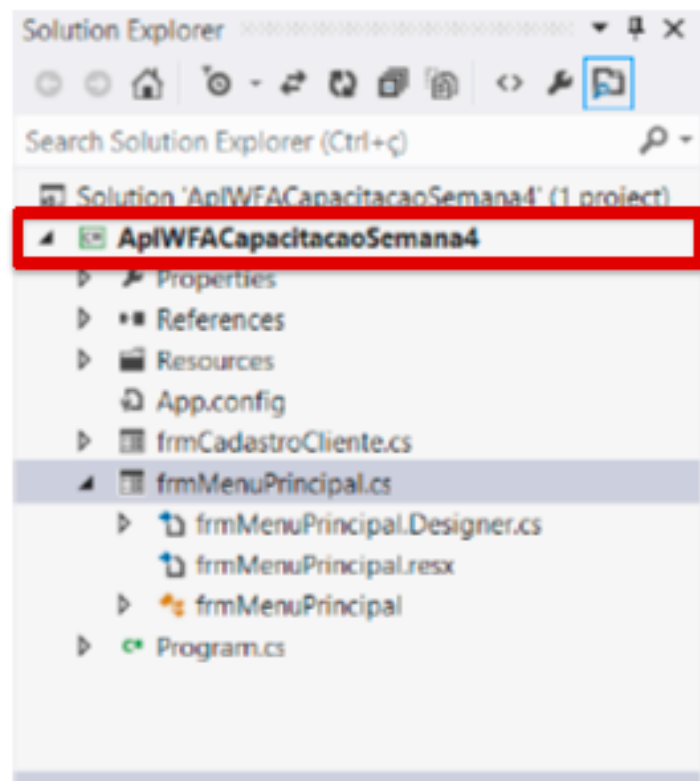
Para criar um formulário novo, devemos clicar com o botão direito em cima do nome do projeto (negrito) que se encontra na janela **Solution Explorer** ao lado direito da tela.



Criar Formulário em um Projeto

Como estamos trabalhando com menus, a partir de agora podemos criar um ou mais formulários dentro de um mesmo projeto, desse modo começaremos a desenvolver um sistema integrado.

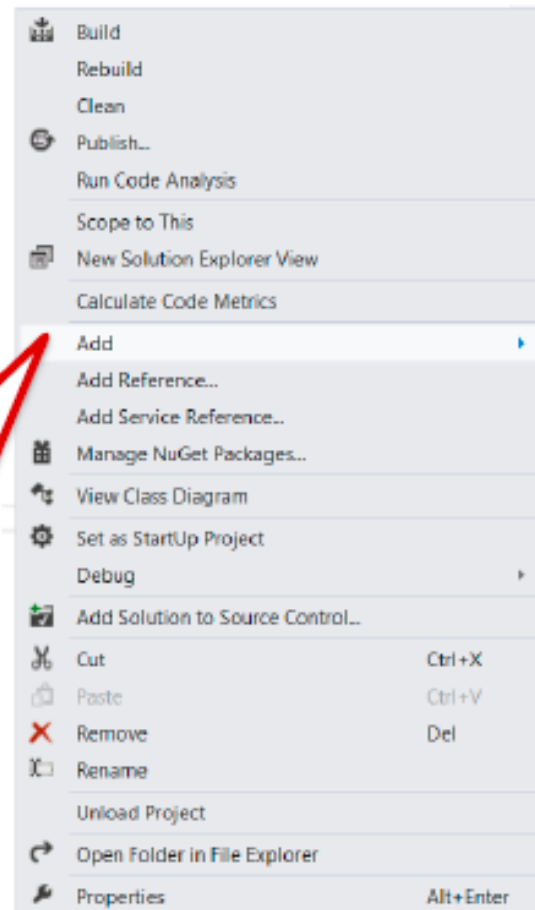
Para criar um formulário novo, devemos clicar com o botão direito em cima do nome do projeto (negrito) que se encontra na janela **Solution Explorer** ao lado direito da tela.



Criar Formulário em um Projeto

Em seguida vai aparecer o menu
Suspense, como mostra a figura:

Opção para
adicionar um
componente
qualquer ao
projeto.

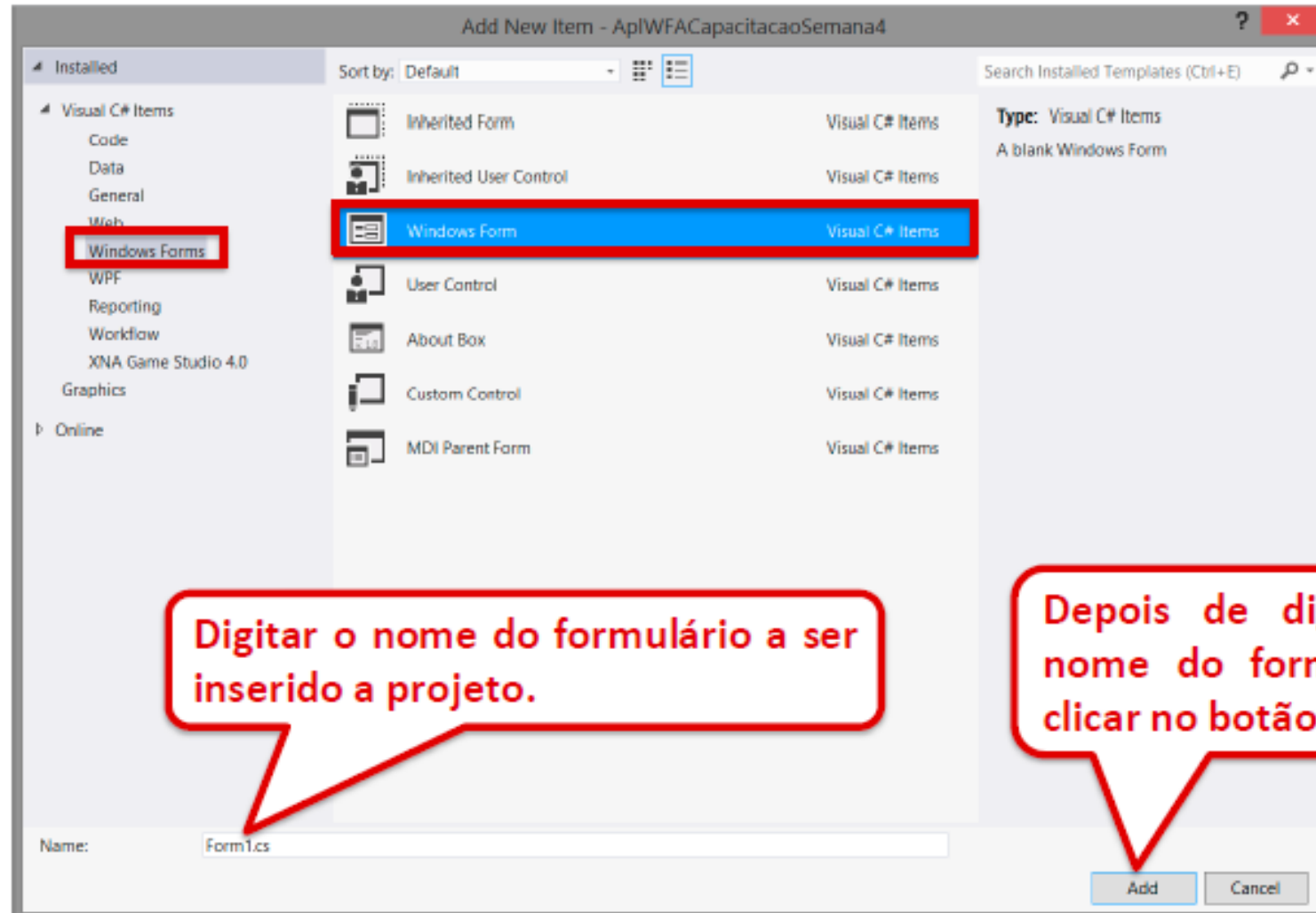


Opção para inserir
um componente
novo ao projeto.

Opção para inserir
um componente
já existente ao
projeto. Neste
caso, após a
inclusão do
componente,
deverá alterar o
namespace antigo
para o novo.

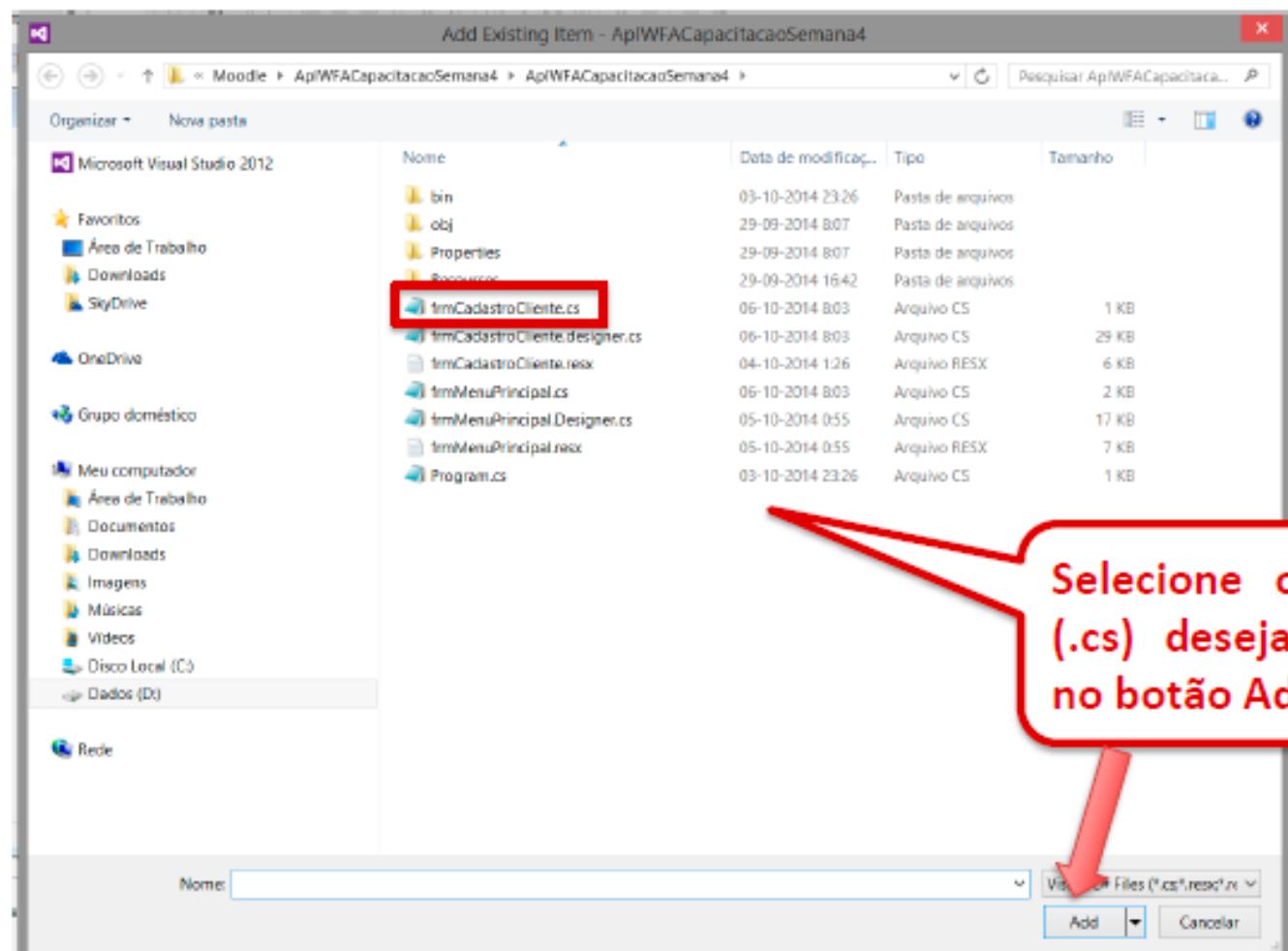
Adicionando um Formulário Novo

Com já vimos anteriormente, quando selecionamos a opção **New Item**, aparecerá a tela abaixo:




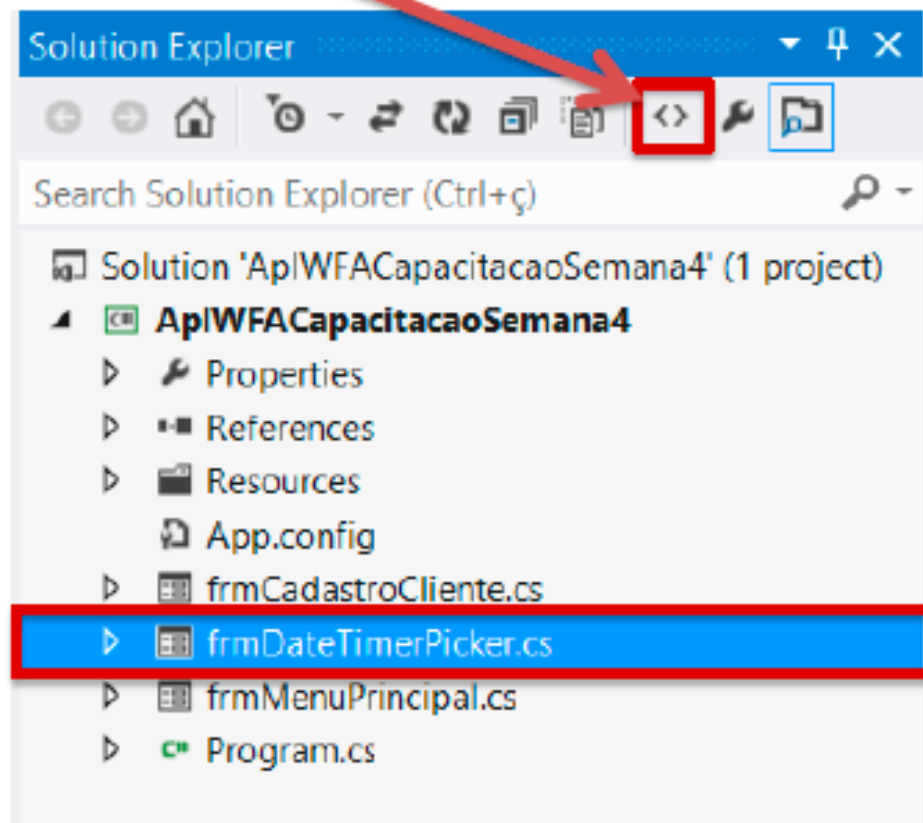
Adicionando um Formulário Existente

Com já vimos anteriormente, quando selecionamos a opção **Existing Item**, aparecerá a tela abaixo:



Adicionando um Formulário Existente

A clicar no botão **Add**, o formulário será visto na janela da **Solution Explorer**. Selecione o formulário e clique no botão  (**View Code**), para poder alterar o namespace do formulário.



Adicionando um Formulário Existente

Na área de código, devemos alterar o namespace para que seu projeto o reconheça, como mostra a figura abaixo:

Namespace Antigo

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AplWFADateTimePicker
{
    public partial class frmDateTimePicker : Form
    {
        public frmDateTimePicker()
        {
            InitializeComponent();
        }
    }
}
```



Namespace Novo

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AplWFACapacidadeSemana4
{
    public partial class frmDateTimePicker : Form
    {
        public frmDateTimePicker()
        {
            InitializeComponent();
        }
    }
}
```

Adicionando um Formulário Existente


Depois de alterar o nome do namespace, aparecerá um sublinhado na cor vermelha abaixo do novo nome, isto significa que o novo nome só foi alterado no ambiente local.


Para o projeto atualizar as referências internas com o novo componente, devemos posicionar o mouse sobre o sublinhado e clicar no botão , na opção  Rename 'AplWFADateTimerPicker' to 'AplWFACapacidadeSemana4'

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

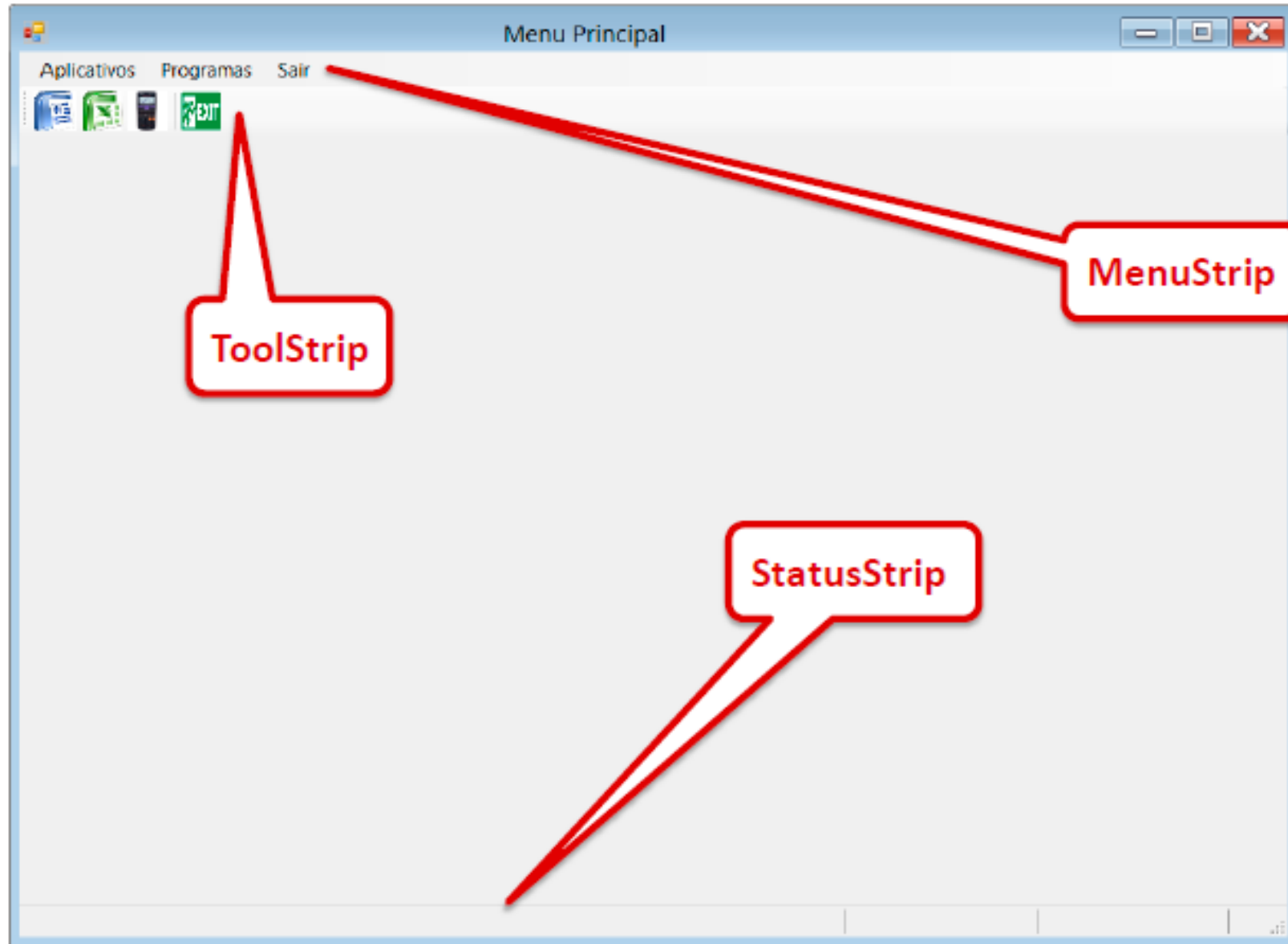
```
namespace AplWFACapacidadeSemana4
{
    public partial class frmDateT
    {
        public frmDateTimerPicker
        {
            InitializeComponent();
        }
    }
}
```

Ao clicar nesta opção o nome (namespace) será alterado em o todo o ambiente.

 Rename 'AplWFADateTimerPicker' to 'AplWFACapacidadeSemana4'

 Rename with preview...

Exemplo – Menu e Process



Exemplo – Código Fonte

Dê um duplo clique na opção Word do menu Aplicativos, e digite o código abaixo:

```
private void tspmnuAplicativosWord_Click(object sender, EventArgs e)
{
    prcExecutor.StartInfo.FileName = "C:\\Program Files\\Microsoft Office\\Office15\\WinWord.Exe";
    prcExecutor.Start();
}
```

A “\” é um caractere especial na linguagem C. Para que a string (Path) tenha a “\” como separador de diretório devemos duplicá-las.

Dê um duplo clique na opção Excel do menu Aplicativos, e digite o código abaixo:

```
private void tspmnuAplicativosExcel_Click(object sender, EventArgs e)
{
    prcExecutor.StartInfo.FileName = @"C:\Program Files\Microsoft Office\Office15\Excel.Exe";
    prcExecutor.Start();
}
```

Podemos utilizar o caractere “@” no início da string (Path) para não precisar usar a instrução “\\”.

Exemplo – Código Fonte

Dê um duplo clique na opção Calculadora do menu Aplicativos, e digite o código abaixo:

```
private void tspmnuAplicativosCalculadora_Click(object sender, EventArgs e)
{
    prcExecutor.StartInfo.FileName = "Calc.Exe";
    prcExecutor.Start();
}
```

Todos os aplicativos que fazem parte do Path do windows, não precisará informar o caminho dos aplicativos.

Exemplo – Código Fonte

Dê um duplo clique na opção Cadastro de Clientes do menu Programas, e digite o código abaixo:

```
private void tspmnuProgramasCadastroClientes_Click(object sender, EventArgs e)
{
    frmCadastroCliente objTela = new frmCadastroCliente();
    objTela.ShowDialog();
}
```

Quando trabalhamos com Menu temos que carregar os outros formulários através das opções de menu. Para isso ocorrer devemos declarar uma variável do tipo do formulário que vamos carregar e em seguida instanciá-los.



Para exibir o formulário instanciado, podemos utilizar dois métodos: Show() e ShowDialog().

Show() → Exibir um formulário, sendo que para voltar ao menu principal é necessário clicar sobre a janela de menu e assim as telas se sobrepõe.

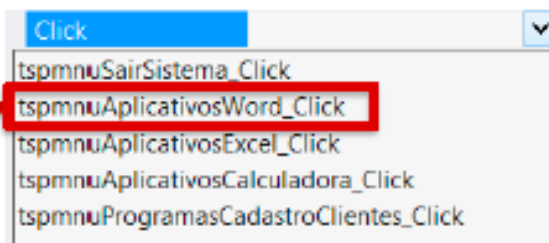
ShowDialog() → Exibir um formulário, sendo que para voltar ao menu principal é necessário fechar o formulário carregado e assim voltar ao anterior.

Exemplo – Código Fonte

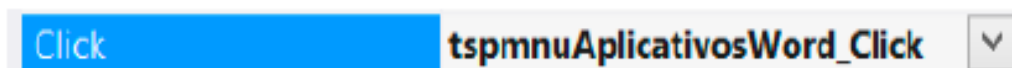
Como foram programados alguns eventos das opções do menu, vamos aprender a reutilizar os eventos já programados, associando a outros componentes. Por exemplo, podemos executar o Word através do menu Aplicativos, além disso podemos executar o Word através do botão na barra de atalho (ToolStrip). Como já está programado a opção Word do menu Aplicativos, iremos associar o mesmo evento da opção para o botão Word da barra de atalho, como segue abaixo:

1. Selecionamos o botão Word da barra de atalho (ToolStrip) 
2. Na caixa de propriedades clicamos no botão  , para selecionar o evento que será programado.
3. No evento click do botão, selecione o evento desejado.

Evento programado para executar o Word através da opção Word do menu Aplicativos.



4. Ao clicar sobre o botão do Word executará a mesma programação da opção do menu.



***** Repetir este processo para todos os botões da barra de atalho *****

Exemplo – Código Fonte



Na barra de status foram inseridos uma label para mensagem, data e hora. Abaixo a programação de cada uma delas.

```
private void tmrTempo_Tick(object sender, EventArgs e)
{
    tspData.Text = DateTime.Now.ToString("dd/MMM/yyyy");
    tspHora.Text = DateTime.Now.ToString("HH:mm:ss");
}

private void frmMenuPrincipal_Load(object sender, EventArgs e)
{
    tspMensagem.Text = "Bem Vindo ao Menu Principal";
}

private void tspmnuSairSistema_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```