

VISUAL STUDIO PARTE I

LINGUAGEM C#

C# é uma linguagem de programação orientada a objetos, semelhante à linguagem Java.

Principais características:

- a) Case sensitive

Ex: var soma = 30;

var Soma = 80;

link para leitura complementar:

<https://pt.stackoverflow.com/questions/47383/quando-usar-var-em-c%23-~text=Por%20fim%2C%20var%20traz%20outra,se%20fazer%20ao%20empregar%20var>

- b) Comentários

// De uma linha

/* mais de uma

linha */

- c) Blocos de instruções entre chaves

- d) O Visual Studio, junto com a linguagem C#, permite desenvolver aplicações desktop, web e para dispositivos móveis

CONVENÇÕES DA LINGUAGEM C#

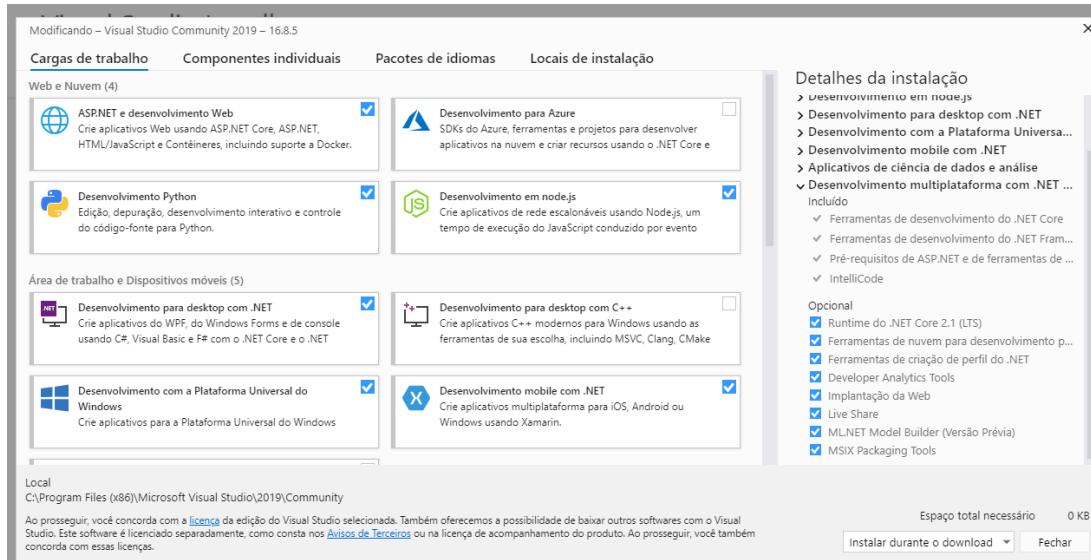
- Primeira letra de cada palavra deve ser maiúscula para nomear classes e métodos
- Use camelCasing (primeira letra minúscula e primeira letra de cada próxima palavra maiúscula) para varáveis locais e parâmetros de métodos

```
public class Calculadora
{
    public double Somar(double primeiroNumero,
segundoNumero)
    {
        double soma = primeiroNumero + segundoNumero;
        return soma;
    }
}
```

- Escreva apenas uma instrução por linha, finalizando-a com ponto e vírgula
- Declare uma variável por linha
- Utilize indentação com TAB (4 espaços)
- Mantenha as duas chaves de bloco alinhadas verticalmente à esquerda

INSTALAÇÃO DO VISUAL STUDIO

Efetue o download do Visual Studio pelo link <https://visualstudio.microsoft.com/pt-br/vs/community/> e instale-o.



Modificando – Visual Studio Community 2019 – 16.8.5

Cargas de trabalho	Componentes individuais	Pacotes de idiomas	Locais de instalação
Jogos (2)			
 Desenvolvimento de jogos com Unity Crie jogos em 2D e 3D com o Unity, um poderoso ambiente de desenvolvimento multiplataforma.	<input type="checkbox"/>	 Desenvolvimento de jogos com C++ Use todo o poder do C++ para criar jogos profissionais desenvolvidos por DirectX, Unreal ou Cocos2d.	<input type="checkbox"/>
Outros conjuntos de ferramentas (6)			
 Processamento e armazenamentos de dados Conecte-se, desenvolva e teste as soluções de dados usando SQL Server, Azure Data Lake ou Hadoop.	<input type="checkbox"/>	 Aplicativos de ciência de dados e análise Linguagens e ferramentas para criação de aplicativos de ciência de dados, incluindo Python e F#.	<input checked="" type="checkbox"/>
 Desenvolvimento de extensão do Visual Studio Crie suplementos e extensões para Visual Studio, incluindo novos comandos, analisadores de código e janelas de	<input type="checkbox"/>	 Desenvolvimento para Office/SharePoint Crie complementos do Office e do SharePoint, soluções do SharePoint e complementos do VSTO usando C#, VB e	<input type="checkbox"/>
 Desenvolvimento para Linux com C++ Crie e depure aplicativos em execução em um ambiente Linux.	<input type="checkbox"/>	 Desenvolvimento multiplataforma com .NET Core Crie aplicativos multiplataforma usando .NET Core, ASP.NET Core, HTML/JavaScript e contêineres, incluindo suporte a	<input checked="" type="checkbox"/>

Modificando – Visual Studio Community 2019 – 16.8.5

Cargas de trabalho	Componentes individuais	Pacotes de idiomas	Locais de instalação
Jogos (2)			
 Desenvolvimento de jogos com Unity Crie jogos em 2D e 3D com o Unity, um poderoso ambiente de desenvolvimento multiplataforma.	<input type="checkbox"/>	 Desenvolvimento de jogos com C++ Use todo o poder do C++ para criar jogos profissionais desenvolvidos por DirectX, Unreal ou Cocos2d.	<input type="checkbox"/>
Outros conjuntos de ferramentas (6)			
 Processamento e armazenamentos de dados Conecte-se, desenvolva e teste as soluções de dados usando SQL Server, Azure Data Lake ou Hadoop.	<input type="checkbox"/>	 Aplicativos de ciência de dados e análise Linguagens e ferramentas para criação de aplicativos de ciência de dados, incluindo Python e F#.	<input checked="" type="checkbox"/>
 Desenvolvimento de extensão do Visual Studio Crie suplementos e extensões para Visual Studio, incluindo novos comandos, analisadores de código e janelas de	<input type="checkbox"/>	 Desenvolvimento para Office/SharePoint Crie complementos do Office e do SharePoint, soluções do SharePoint e complementos do VSTO usando C#, VB e	<input type="checkbox"/>
 Desenvolvimento para Linux com C++ Crie e depure aplicativos em execução em um ambiente Linux.	<input type="checkbox"/>	 Desenvolvimento multiplataforma com .NET Core Crie aplicativos multiplataforma usando .NET Core, ASP.NET Core, HTML/JavaScript e contêineres, incluindo suporte a	<input checked="" type="checkbox"/>

Mais detalhes sobre instalação:

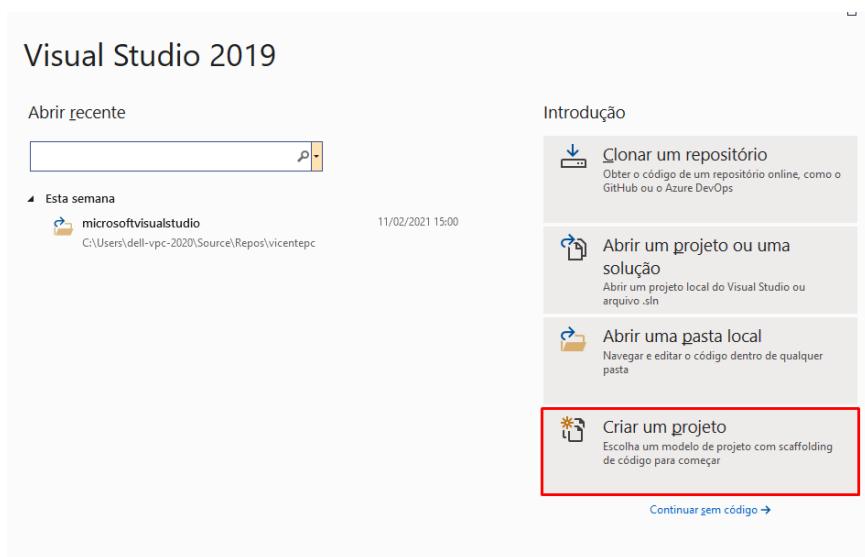
<https://docs.microsoft.com/pt-br/visualstudio/install/install-visual-studio?view=vs-2019>

CRIANDO PROJETO / CONHECENDO O AMBIENTE

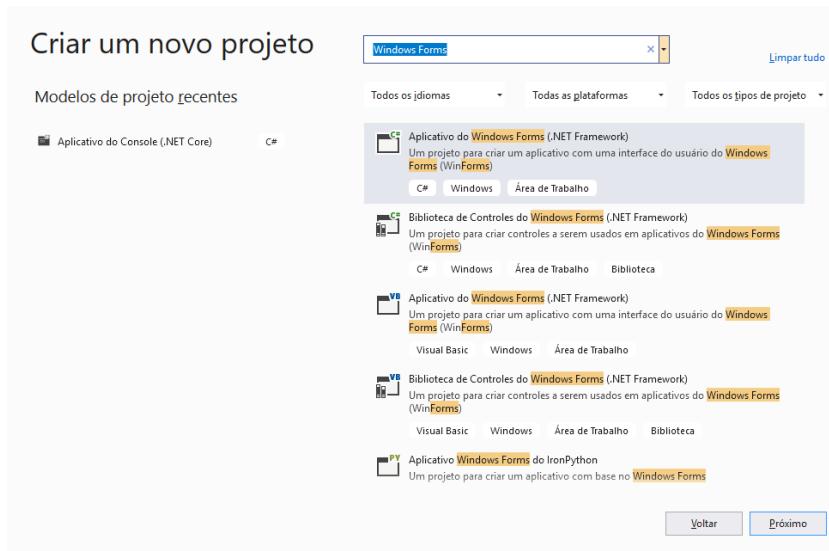
Crie uma pasta para o armazenamento dos projetos.

Abra o Visual Studio.

Será apresentada essa janela:



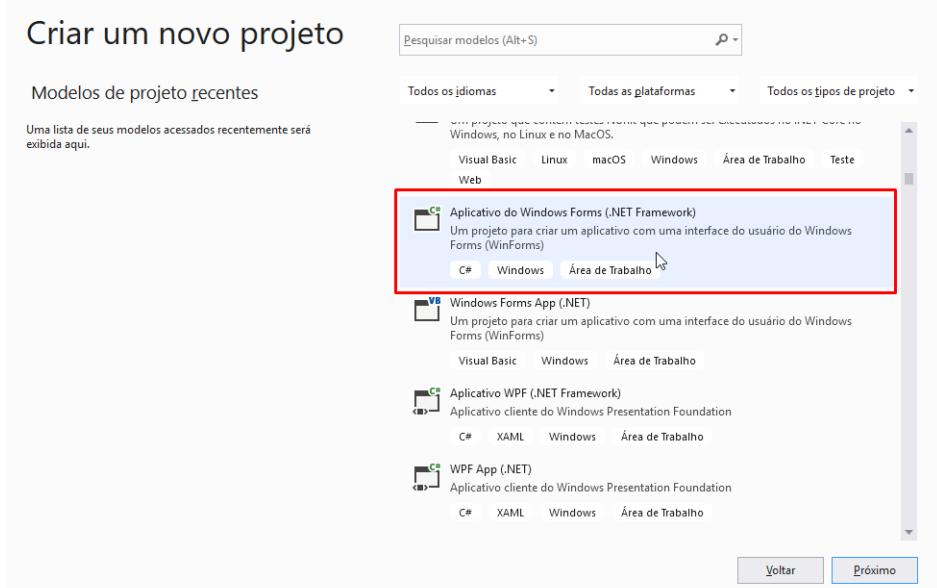
Selecione a opção destacada na figura anterior. Sugirá a seguinte janela:



Acesse a opção ilustrada na figura anterior.

Será apresentada uma nova janela:

Criar um novo projeto

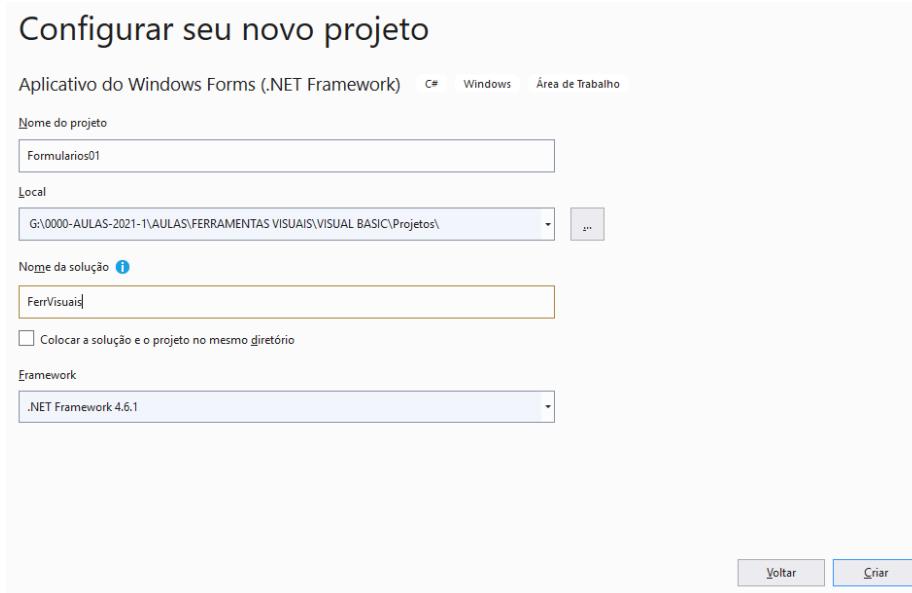


Selecione a opção em destaque na figura anterior, destacando C#.

Confirme em Próximo.

Será apresentada uma janela. Informe os dados do projeto e selecione a pasta dos seus projetos do Visual Studio, como ilustrado:

Configurar seu novo projeto

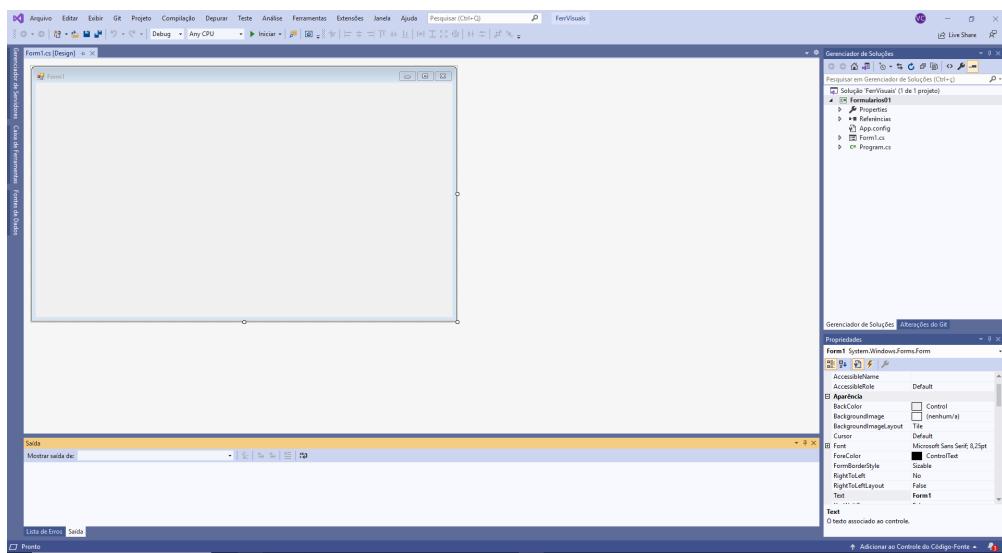


Confirme em Criar.

Será apresentado o ambiente de desenvolvimento.



ECEC – ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CMP1554 – FERRAMENTAS VISUAIS DE DESENVOLVIMENTO DE SOFTWARE
PROFESSOR VICENTE PAULO DE CAMARGO



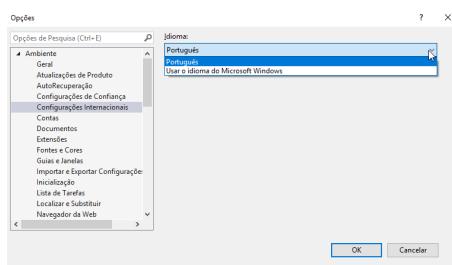
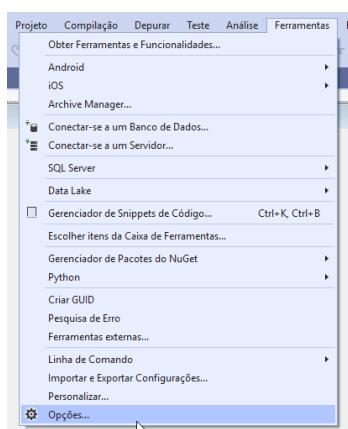
No lado direito tem-se o Gerenciador de soluções e a área de propriedades e eventos:



Caso a janela de propriedades não stá visualizada, basta pressionar F4.

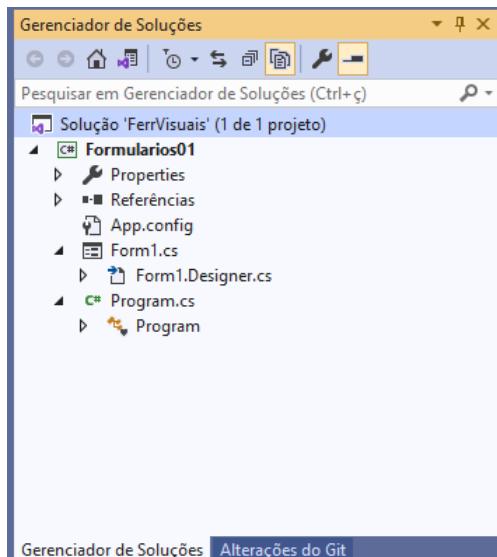
Muitas das janelas de apoio podem ser acessadas pela opção view do menu principal.

Para alterar o idioma:



Ajuste de acordo com sua conveniência.

O lado direito da tela principal tem o gerenciador de soluções:

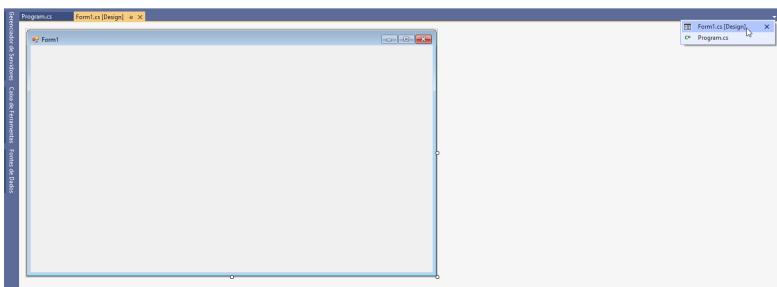


A solução (FerrVisualais) e o projeto Formularios01 com sua estrutura com formulários (só o formulário Form1.Designer.cs) e programas em C# (Program.cs).

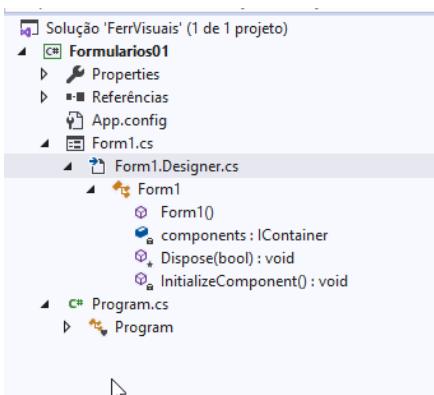
A solução pode possuir vários projetos.

Para abrir um arquivo basta efetuar um duplo clique sobre o seu nome.

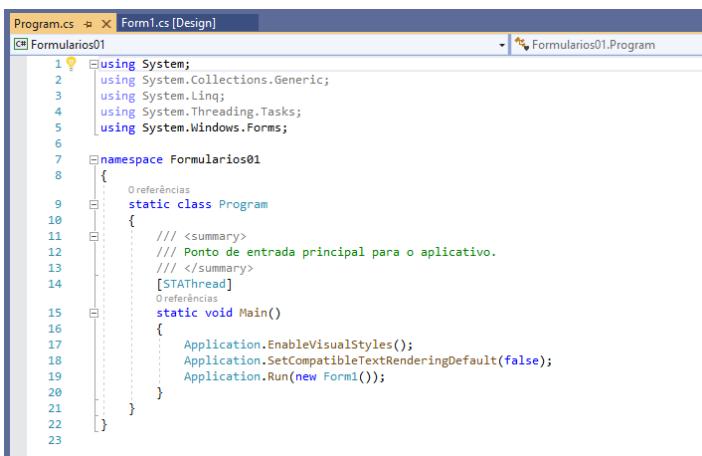
A próxima figura destaca o recurso para se selecionar um determinado arquivo quando se tem mais de um arquivo disponível no ambiente de design e de edição.



Observe no Gerenciador de soluções:



Ao clicar na opção em destaque na figura anterior, ela abre o código correspondente ao Form1.cs.

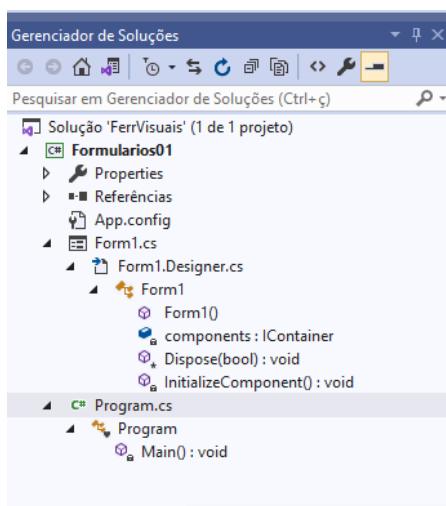


```

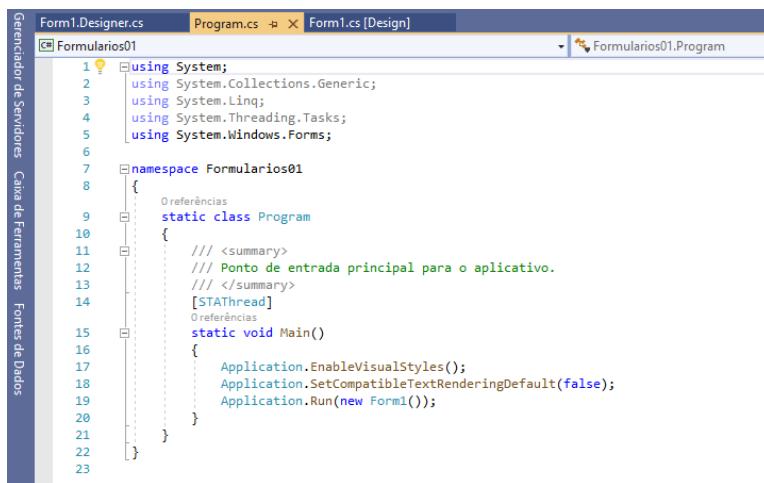
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.Windows.Forms;
6
7 namespace Formularios01
8 {
9     static class Program
10    {
11        /// <summary>
12        /// Ponto de entrada principal para o aplicativo.
13        /// </summary>
14        [STAThread]
15        static void Main()
16        {
17            Application.EnableVisualStyles();
18            Application.SetCompatibleTextRenderingDefault(false);
19            Application.Run(new Form1());
20        }
21    }
22}

```

Observe a opção destacada na próxima figura:



Elle permet de visualiser le code principal du programme, c'est à dire le code qui est exécuté lorsque l'application est lancée.



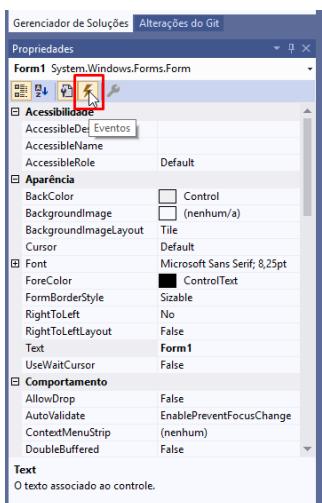
```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using System.Windows.Forms;
6
7  namespace Formularios01
8  {
9      static class Program
10     {
11         /// <summary>
12         /// Ponto de entrada principal para o aplicativo.
13         /// </summary>
14         [STAThread]
15         static void Main()
16         {
17             Application.EnableVisualStyles();
18             Application.SetCompatibleTextRenderingDefault(false);
19             Application.Run(new Form1());
20         }
21     }
22 }
23

```

A ligne 19 souligne le formulaire (Form1) qui sera activé lorsque l'application sera exécutée. À partir de ce point, le formulaire aura ses événements pour diriger le flux d'actions de l'application.

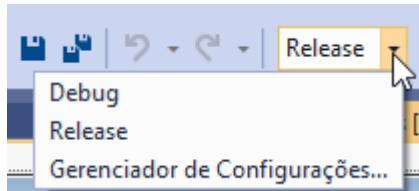
A janela de propriedades, prochaine figure, permet au développeur de modifier les propriétés d'un formulaire ou d'un composant dans un formulaire.



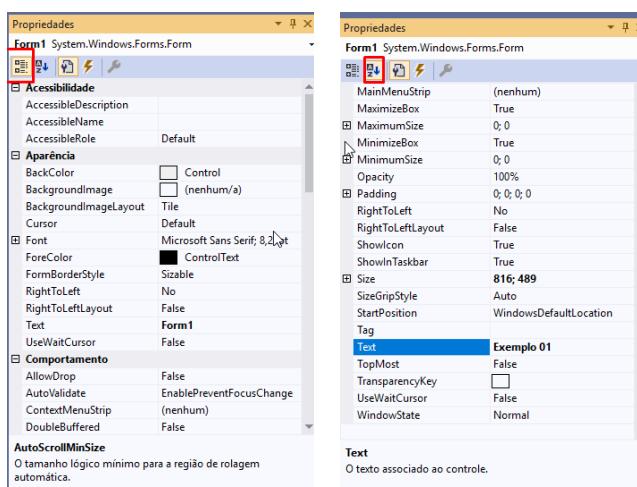
Note que a figura anterior destaca um botão, o qual permite acessar opções de eventos de um determinado componente.

***Um evento corresponde a um bloco de código que permite efetuar uma determinada ação. Por exemplo: quando um botão é clicado ativa o seu evento click do botão esquerdo, acionando o bloco de código desse evento.**

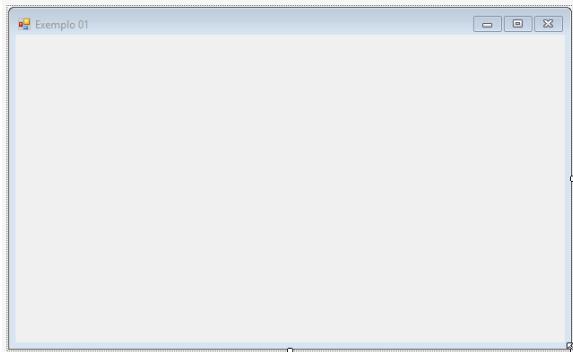
Para executar a aplicação utiliza-se CTRL+F5 (modo release) ou F5 (modo debug). Essas duas opções podem ser acessadas no menu como ilustrado:



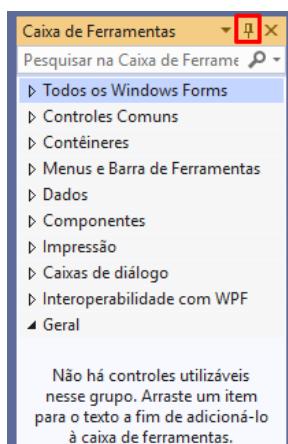
Com o formulário que está aberto, observe suas propriedades agrupadas ou em ordem alfabética, conforme mostra as duas figuras a seguir (com os seus respectivos botões de ativação):



A propriedade **Text** apresenta o título da barra principal do formulário:

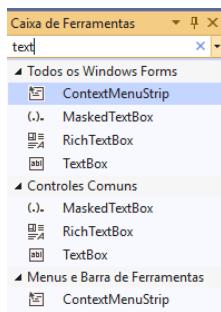


Ao selecionar um formulário é possível acessar a caixa de ferramentas que oferece uma grande quantidade de componentes que podem ser incluídos em um formulário:



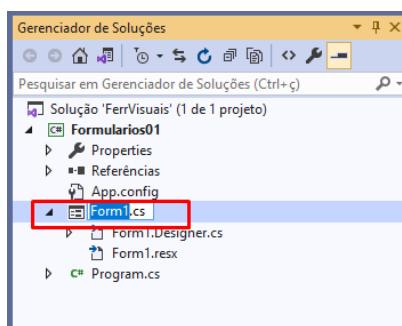
Se esta não estiver disponível, basta exibi-la pela opção **Exibir** do menu principal. Caso deseje, pode deixá-la fixa no ambiente clicando no botão em destaque na figura anterior. Note na figura anterior que os componentes estão distribuídos por categorias. Caso deseje acessar um componente específico, basta informar na caixa de texto de pesquisa

uma parte do nome do componente desejado. Por exemplo, a próxima figura ilustra a pesquisa pela palavra text:



Após selecionar um componente, basta clicar dentro do formulário para inseri-lo no local onde o mouse for clicado.

Altere o nome do formulário atual, seguindo os passos a seguir:



Selecione o nome do formulário. Pressione F2 e altere o nome para **FormMain**. Será informado que o código correspondente será alterado. Altere sempre o nome de um arquivo pelo seu nós raiz.

Para salvar um determinado arquivo utilize CTRL+S ou SHIFT+CTRL+S para salvar todos os arquivos.

Ressalta-se que Form1.Designer.cs é criado pela ferramenta e não deve ser editado. O código fonte correspondente às ações relacionadas ao formulário é acessado pela tecla F7. Esse atalho permite ao desenvolvedor codificar os eventos relacionados ao formulário.

Algumas propriedades de um formulário:

BackColor = Cor de fundo

ControlBox = Quando False remove os botões da parte superior direita

MaximumSize / MinimumSize = Ativa/Desativa os respectivos botões (True/False)

Opacite = Permite deixar o formulário com mais ou menos intensidade de transparência

ShowIcon = Habilita ou desabilita o ícone do formulário

ShowInTaskBar = Habilita / desabilita barra de tarefas

StartPosition = Ajusta a posição do formulário na tela do computador

Para codificar um evento basta efetuar um duplo clique sobre o evento desejado.

Alguns eventos:

Shown = É ativado quando o formulário está se preparando para ser mostrado

```
private void FormMain_Shown(object sender, EventArgs e)
{
    MessageBox.Show("Seja Bem vindo!!");
}
```

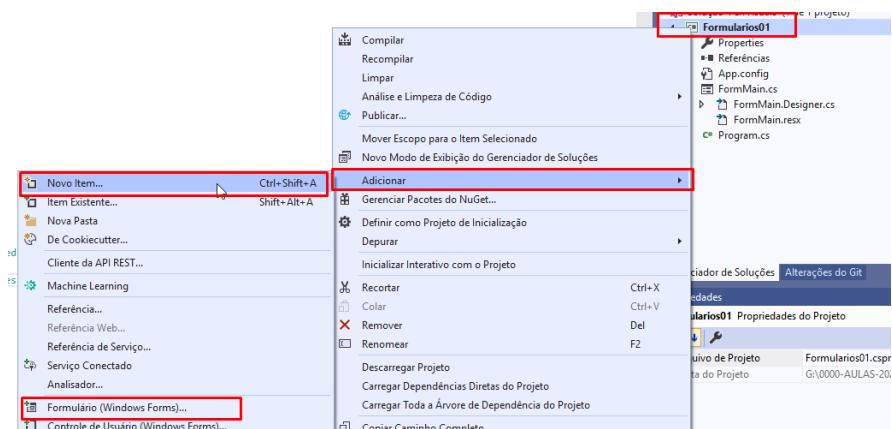
Neste exemplo, o evento mostrará uma mensagem com MessageBox.Show.

FormClosing = Ocorre quando o formulário estiver fechando.

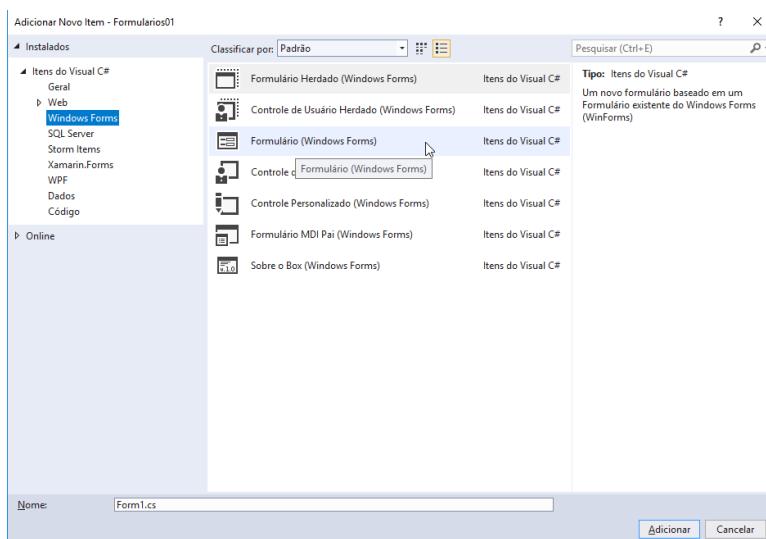
```
1 referência
private void FormMain_FormClosing(object sender, FormClosingEventArgs e)
{
    if (MessageBox.Show("Deseja Sair?", "Confirmação", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)
    {
        e.Cancel = true;
    }
}
```

FormClosed = Quando o formulário estiver fechado.

Para criar um novo formulário basta clicar no nome projeto com o botão direito:

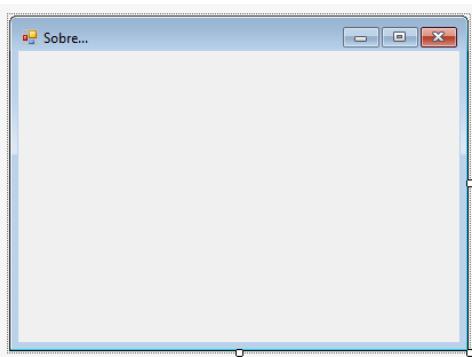


Selecionar Adicionar e, em seguida, selecionar ou Novo Item ou Formulário Windows Forms, como ilustrado na figura anterior. Nos dois casos, será apresentada a seguinte janela:



Selecione as opções destacadas, como ilustrado na figura anterior, e informe o nome do novo formulário na caixa de texto **Nome**.

Informe o nome **FormSobre**. Na propriedade **Text** desse formulário informe **Sobre...** como ilustrado:



Altere a propriedade **StartPosition** desse formulário para **CenterParent**. Assim, quando abrir, esse formulário será aberto no centro do outro formulário.

Arraste um componente **Button** para o formulário **FormMain** e altere sua propriedade **Text** para **Sobre**. Redimensione o tamanho desse formulário.

Dê um duplo click no evento **Click** desse botão, digite o código a seguir:

```
1 referência
private void button1_Click(object sender, EventArgs e)
{
    var form = new FormSobre();
    form.ShowDialog();
}
```

Ajuste o código de FormMain, como ilustrado:

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace Formularios01
12 {
13     public partial class FormMain : Form
14     {
15         public FormMain()
16         {
17             InitializeComponent();
18         }
19
20         private void FormMain_Load(object sender, EventArgs e)
21         {
22             // MessageBox.Show("Seja bem vindo!!!");
23         }
24
25
26         private void button1_Click(object sender, EventArgs e)
27         {
28             var form = new FormSobre();
29             form.ShowDialog();
30         }
31
32         private void FormMain_FormClosing(object sender, FormClosingEventArgs e)
33         {
34             if (MessageBox.Show("Deseja Sair?", "Confirmação", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)
35             {
36                 e.Cancel = true;
37             }
38         }
39     }
40 }
41
42

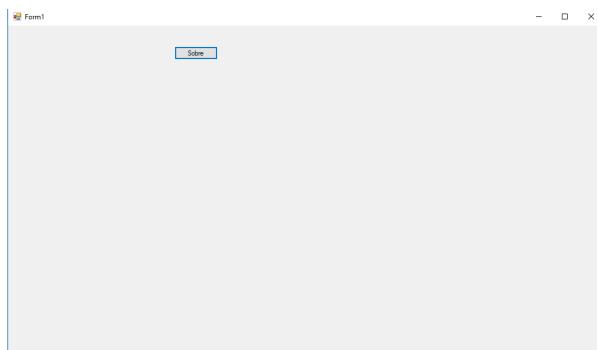
```

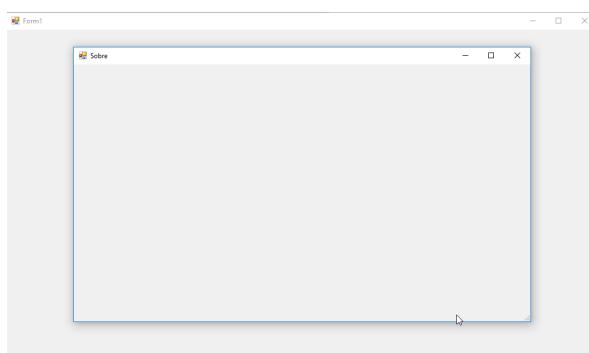
A palavra-chave **namespace** é usada para declarar um escopo que contém um conjunto de objetos relacionados.

O seu código talvez esteja um pouco diferente, mas insira os comentários indicados nos locais específicos do seu código.

O comando da linha 29 permite abrir a segunda janela como uma janela filha, sendo que só se pode acessar a primeira janela quando a segunda for fechada.

Salve a aplicação e execute-a. Após o FormMain ser aberto efetue um clique sobre o botão.



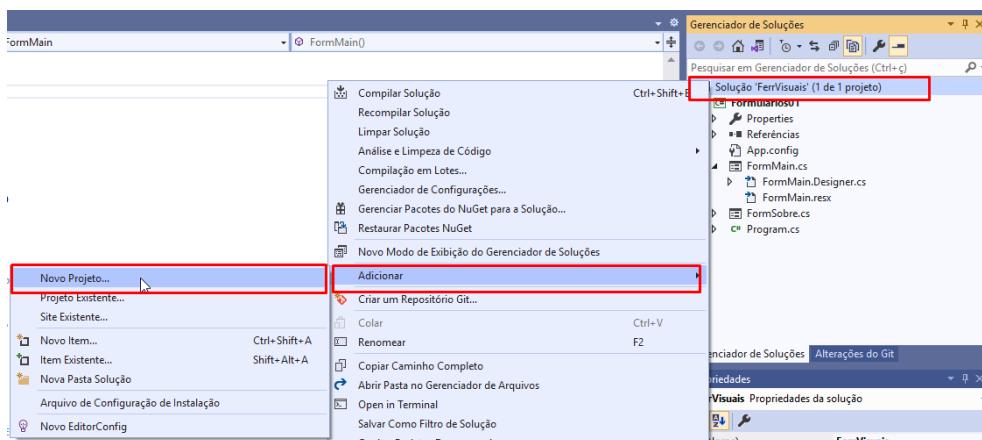


Fica como atividade a criação de um evento que possa abrir a segunda janela com o pressionar da tecla **a** (de abrir), por exemplo.

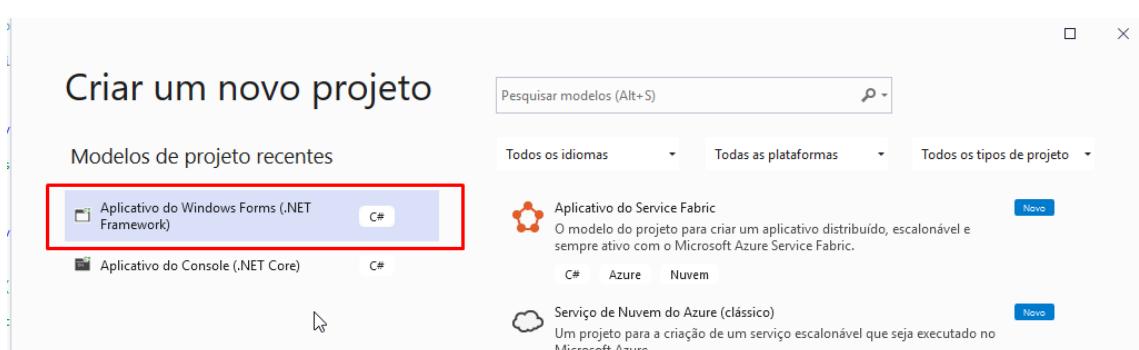
Feche esse projeto.

COMPONENTES VISUAIS / CRIANDO MENUS

Crie um novo projeto de nome **Menus** efetuando um duplo clique sobre o nome da solução, como ilustrado:

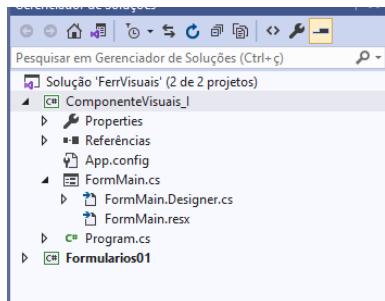


Em seguida, selecione como ilustrado e confirme em Próximo:

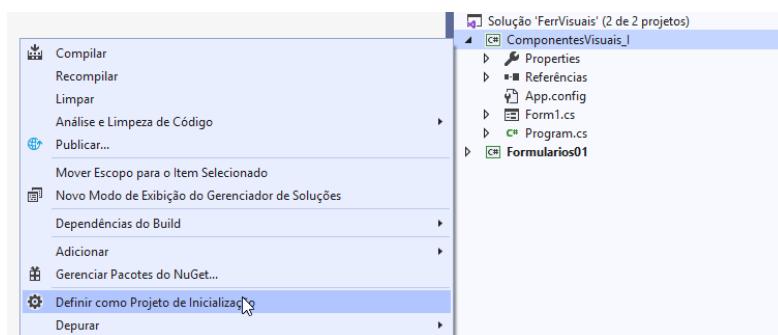


Informe o nome do projeto:

Depois de criado o novo projeto, a solução tem dois projetos:



Mude o status desse novo projeto para projeto de inicialização. Para isso, selecione-o com o botão direito e acompanhe a figura:



Em seguida, altere o nome de Form1 do projeto ComponentesVisuais_I para FormMain.

Há várias ajudas em tempo de codificação como:

F7 = para abrir a área de edição do código de um formulário

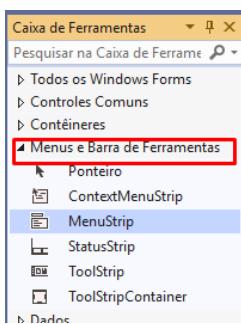
SHIFT+CTRL+B = para compilar.

SHIFT+CTRL+L = remover a linha onde o está o cursor de edição

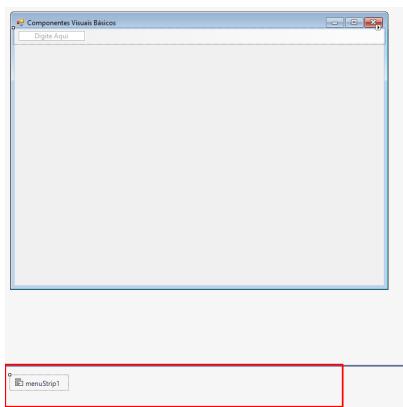
CTRL+F5 = executar o projeto

Altere a propriedade Text do formulário para **Componentes Visuais Básicos**.

Acesse a Caixa de ferramentas e selecione o componente como indicado:

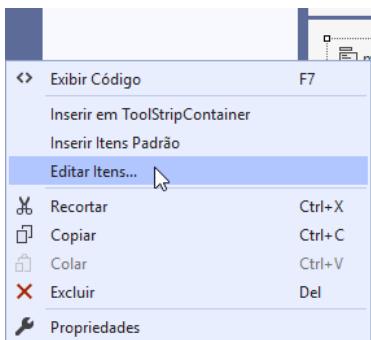


Clique dentro do formulário, deixando o ambiente de desenvolvimento como ilustrado:

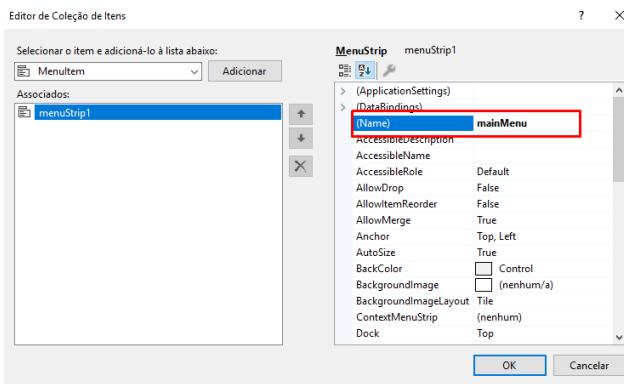


A área em destaque mostra os componentes que não são visuais. No caso, o elemento existente faz parte da barra de menu.

Clique com o botão direito sobre esse elemento:



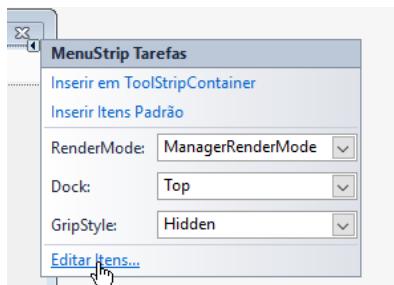
Selecione Editar Itens... e altere a propriedade Name para **mainMenu**, como ilustrado:



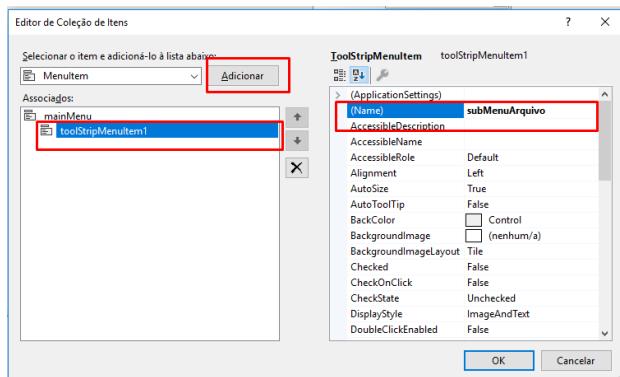
Em seguida, edit os itens de menu. Para isso acesse a seta ilustrada a seguir:



Irá apresentar a janela de diálogo a seguir:

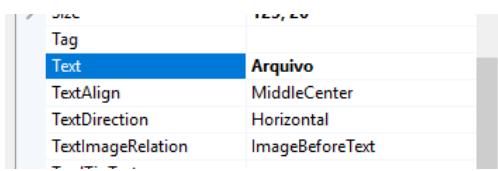


Selecione a opção ilustrada na figura anterior. Será apresentada a mesma janela de diálogo quando foi alterado o nome de mainMenu:



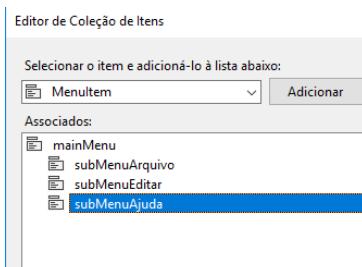
Clique em adicionar. Será inserido um novo item de menu abaixo de mainMenu. Altere a propriedade Name para subMenuArquivo, como ilustrado anteriormente.

Altere a propriedade Text para Arquivo:

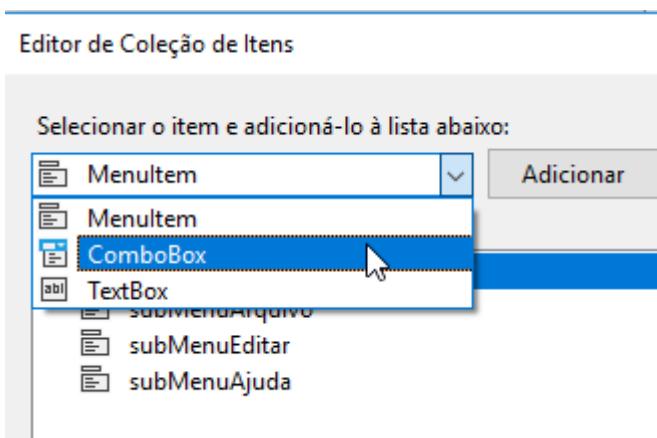


Insira mais dois itens: **subMenuItem**, com o título **Editar** e **subMenuItem**, com o título **Ajuda**.

Como ilustrado:

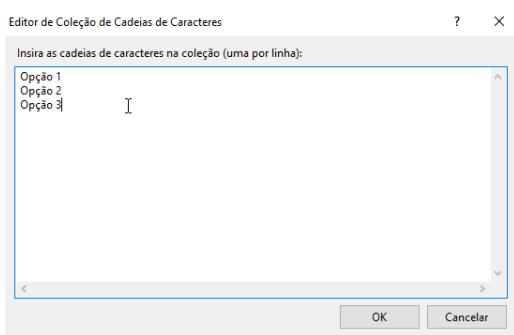


Insira um item de ComboBox, como ilustrado:

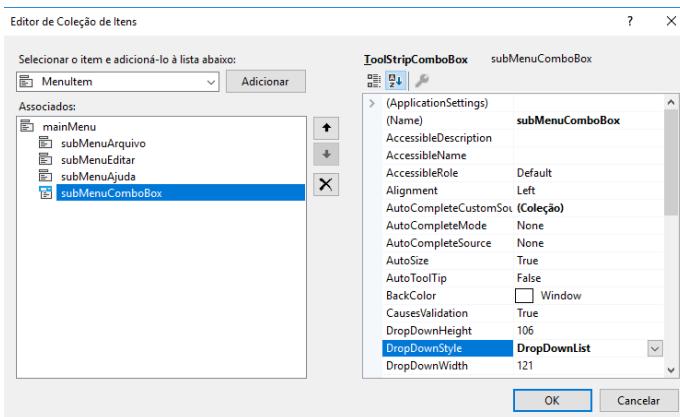


Adicionar:

Altere Name para **subMenuItemComboBox**. Na propriedade Itens, selecione o botão de reticências e insira os seguintes:



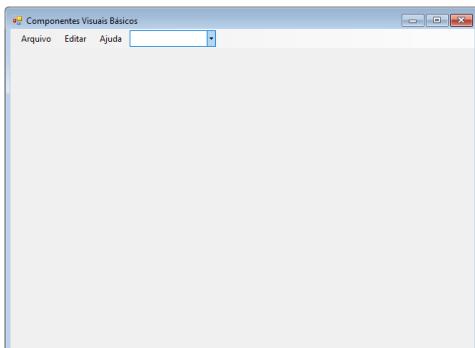
Acesse a propriedade DropDownStyle, como ilustrado:



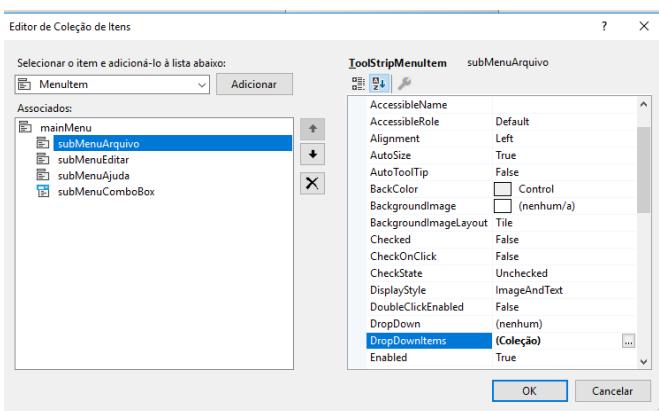
Altere-a para o indicado na figura anterior. Essa opção faz com que os itens não sejam alterados em tempo de execução.

Confirme e OK.

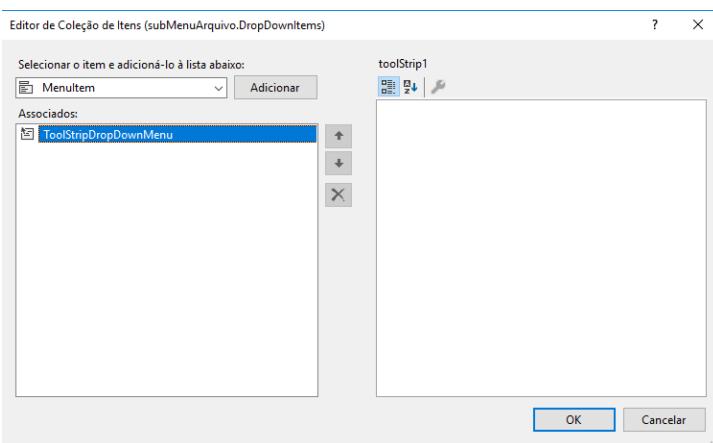
O formulário atualizado:



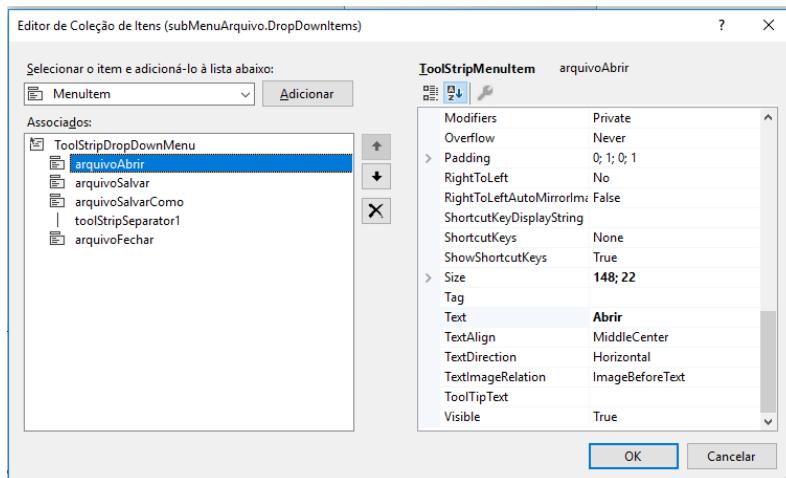
Vamos inserir alguns itens na opção Arquivo. Para isso, acesse a barra de menu e a seta auxiliar. Em seguida, selecione o item Arquivo e a propriedade indicada a seguir:



Clique no botão de reticências. A janela será alterada como ilustrado:



Selecione a opção **MenuItem** e depois o botão **Adicionar**:



Selecione três **MenuItem**, um **Separador** e mais um **MenuItem**.

Em seguida, informe, respectivamente, as propriedades para os três **MenuItem**:

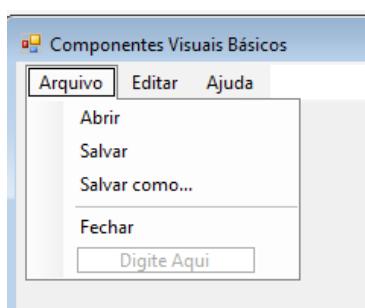
Name: **arquivoAbrir** e Text: Abrir

Name: **arquivoSalvar** e Text: Salvar

Name: **arquivoSalvarComo** e Text: Salvar como...

Name: **arquivoAjuda** e Text: Fechar

O menu Arquivo ficará assim:



Salve o projeto.

Faça o mesmo processo para a opção Editar, inserindo itens para essa opção:

Name: editarCopiar e Text: Copiar

Name: editarColar e Text: Colar

Em seguida, faça o mesmo para a opção Ajuda:

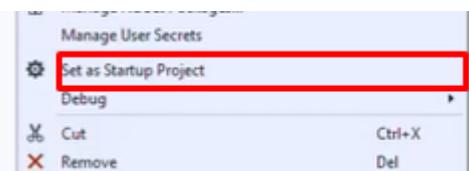
Name: ajudaConteudoAjuda e Text: Conteúdo de ajuda

Separador

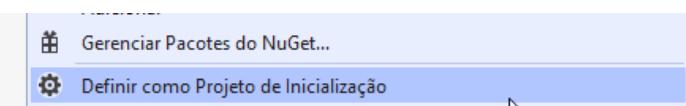
Name: ajudaSobre e Text: Sobre

Salve o projeto

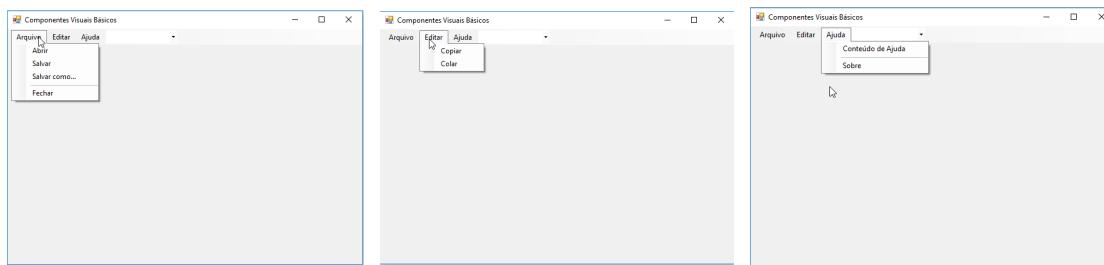
Para executar esse projeto, deve-se observar o gerenciador de soluções. Ele possui dois projetos, porém o projeto que está em negrito é sempre o que será executado. No caso, deseja-se executar o projeto ComponentesVisuais_I. Ele não está em negrito e não será executado quando se tentar executá-lo. Clique com o botão direito no nome desse projeto. Se o idioma for inglês, selecione a opção indicada:



Se o idioma for português, selecione a opção indicada:



Salve o projeto e execute-o.



Pode-se também inserir itens de menu como caixa de texto, que é a opção Text. Mas, não é o foco nesse momento.

Insira **tecla aceleradora** para as opções principais do menu. Basta acessar cada opção e ir na Propriedade Text de cada opção e inserir um e comercial (&) na frente de cada letra que deseja ser a tecla de atalho, desde que sejam distintas para essas opções de menu principal, podendo-se repetir as letras para os sub menus.

Arquivo = &Arquivo; Editar = &Editar; Ajuda = Aj&uda.

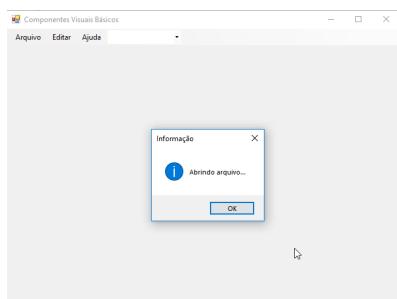
Ao executar, basta pressionar a tecla Alt seguida da tecla de atalho desejada.

Para criar um evento para as opções de um menu, basta selecionar a opção desejada e clicar duas vezes no evento desejada na área de propriedades. Faça isso para a opção Abrir de Arquivo. Clique duas vezes no evento Click dessa opção. Será criado um método na área de código. Informe o que está ilustrado:

```
private void arquivoAbrir_Click(object sender, EventArgs e)
{
    MessageBox.Show("Abrindo arquivo...","Informação",MessageBoxButtons.OK,MessageBoxIcon.Information);
}
```

Para melhorar a disposição do formulário principal, altere sua propriedade StartPosition para CenterScreen.

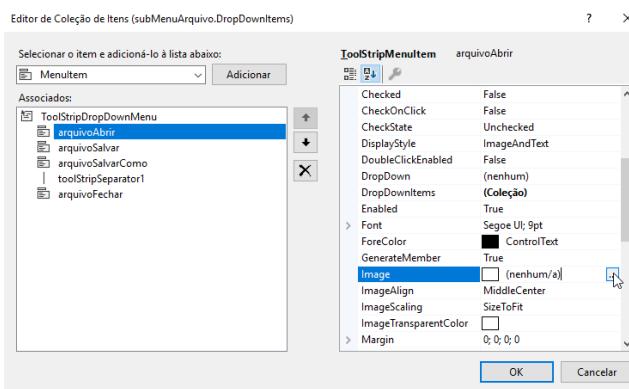
Salve e execute.



Caso se deseje inserir ícones nas opções, deve-se utilizar imagens de 16x16, jpeg ou png, preferencialmente. Assim, selecione algumas imagens em uma pasta para que se possa utilizá-las.

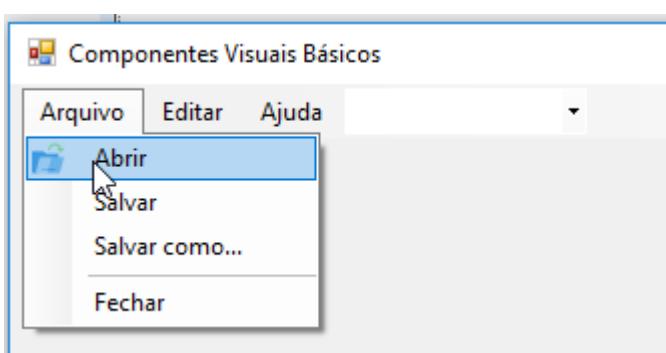
Um software interessante que oferece diversas imagens e permite editá-las é o Metro Studio (<https://www.syncfusion.com/downloads/metrostudio>). Efetue o download desse software e instale-o. É free. Há também sites que oferecem download de ícones como, por exemplo, flaticon.com.

Acesse o menu desejado pela seta de configurações de menu. Acesse a propriedade Image:

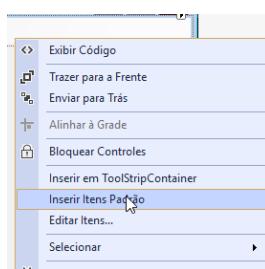


Selecione o arquivo de ícone desejado.

Salve o projeto e execute-o.

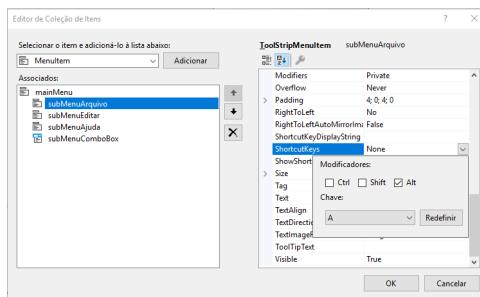


Há também a possibilidade de inserir itens de menu de uma maneira mais rápida. Basta clicar com o botão direito sobre a barra de menu e selecionar como ilustrado:

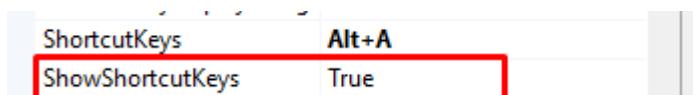


Serão inseridos itens de menu com ícones. Basta fazer os ajustes necessários de acordo com suas necessidades.

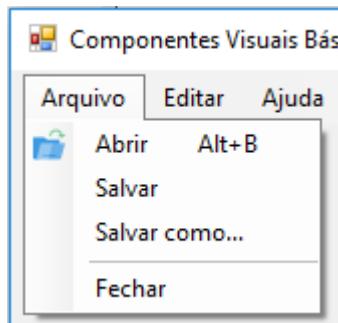
Para finalizar, crie **tecla de atalho** para um determinado item de menu. **Uma tecla de atalho permite acessar diretamente um item de menu sem acessar o menor diretamente.** A tecla aceleradora, vista anteriormente, necessita acessar o menu. Para criar uma tecla de atalho basta acessar um item de menu e selecionar a opção **ShortcutKeys** e selecionar uma combinação de tecla para o item de menu:



Um detalhe interessante é a possibilidade de mostrar ou não a tecla de atalho junto ao item. Para isso, basta deixar a propriedade **ShowShortcutKeys** como True, como ilustrado a seguir:



Observe a tecla de atalho sendo visualizada para o menu Abrir:



Salve o projeto e execute-o. Verifique o uso da tecla de atalho criada.