

Estrutura de um programa em Visual Studio com C#

APRESENTAR OS CONCEITOS SOBRE DESENVOLVIMENTO DE UMA APLICAÇÃO GRÁFICA. APRESENTAR O IDE (INTEGRATED DEVELOPMENT ENVIRONMENT) AMBIENTE INTEGRADO DE DESENVOLVIMENTO DO VISUAL STUDIO E IDENTIFICAR AS PRINCIPAIS FUNCIONALIDADES DA FERRAMENTA. APRESENTAR A ESTRUTURA DE UM PROGRAMA C# UTILIZANDO O VISUAL STUDIO.

AUTOR(A): PROF. APARECIDA DE FATIMA CASTELLO ROSA

Introdução aos conceitos sobre aplicação com interface gráficas (GUI)

Nesta disciplina nós vamos criar aplicações com interface gráficas (GUI), que fazem uso de Janelas, Botões, Labels, Caixas de Texto, etc. Vamos desenvolver programas com interfaces gráficas e orientados a eventos. Nesse tipo ambiente de desenvolvimento a tarefa de programação não acompanha uma sequência lógica idêntica à dos programas desenvolvidos para modo texto, como por exemplo na linguagem C.

Programador desse tipo de ambiente precisa, além de se preocupar com a lógica do programa, prever as ações do usuário, fornecendo código separado para cada ação. Essas ações podem ser um clique em um botão, digitar um texto em uma caixa de texto, fechar uma janela, etc. Esse tipo de programação é conhecido como Programação Orientada a Eventos.

O programa é composto por diversas sub-rotinas (métodos) que fornecem a lógica executada quando certa ação ocorre. Por exemplo, quando o usuário clicar em um botão, devemos implementar o código para que seja executado no evento clique desse botão.

É necessário considerar dois fatores:

- A lógica do próprio programa;
- A lógica relacionada ao evento do controle para cada ação do usuário.

Agora vamos conhecer a ferramenta para desenvolver as aplicações.

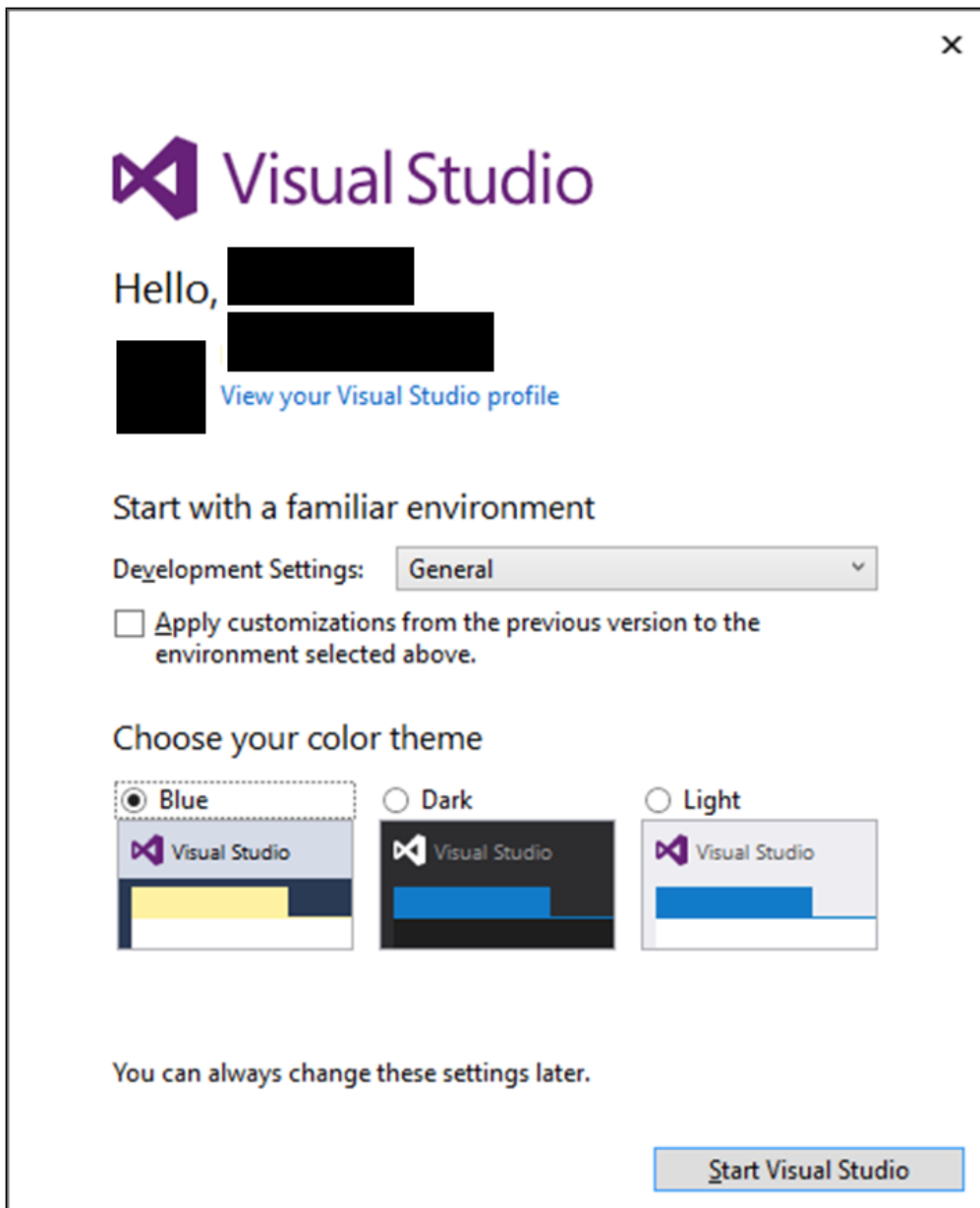
Introdução ao Visual Studio

Para começar a desenvolver os programas em C# você precisa de um IDE que dê suporte ao Microsoft .NET Framework. Neste caso estamos utilizando o Visual Studio Community 2015, mas você também poderá utilizar outras versões do Visual Studio (2013, 2012, 2010).

Configurando o Visual Studio ou Visual Studio Community 2015

Depois de fazer o download do Visual Studio do site da Microsoft, faça a sua instalação. A instalação é bem simples, basta aceitar o contrato proposto e teclar *next* até terminar a instalação.

Na instalação do Visual Studio Community 2015 você poderá definir o ambiente de linguagens que deseja desenvolver. Em *Development Settings* escolha *General*. Em *Choose your color theme* você escolhe qual a cor deseja configurar para o seu Visual Studio Community. Depois clique no botão *Start Visual Studio*.

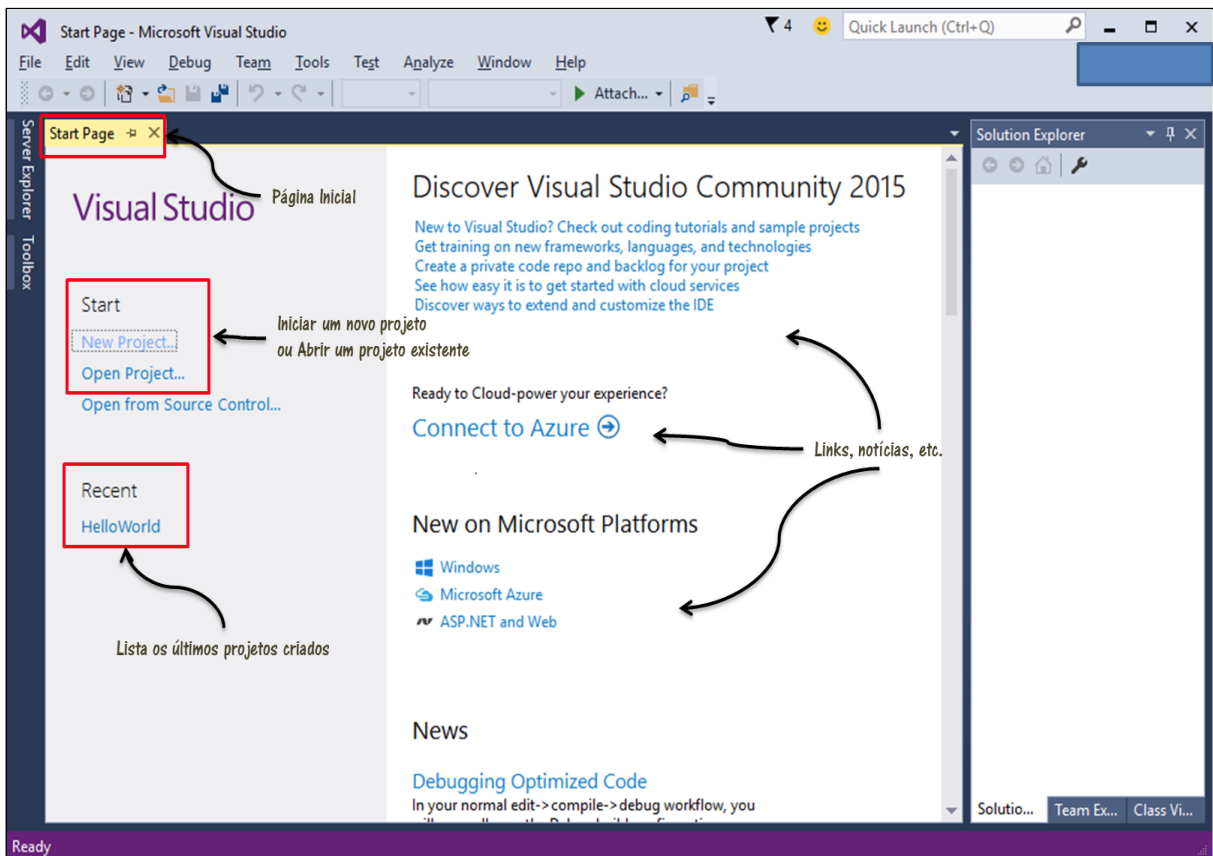


Aguarde enquanto o Visual Studio se prepara para ser iniciado pela primeira vez. Esse processo pode demorar um pouquinho.

Para as versões anteriores escolha a opção *General Development Settings* e clique no botão *Start Visual Studio*. As versões anteriores do Visual Studio podem apresentar ligeiras diferenças em suas telas.

Conhecendo o IDE do Visual Studio

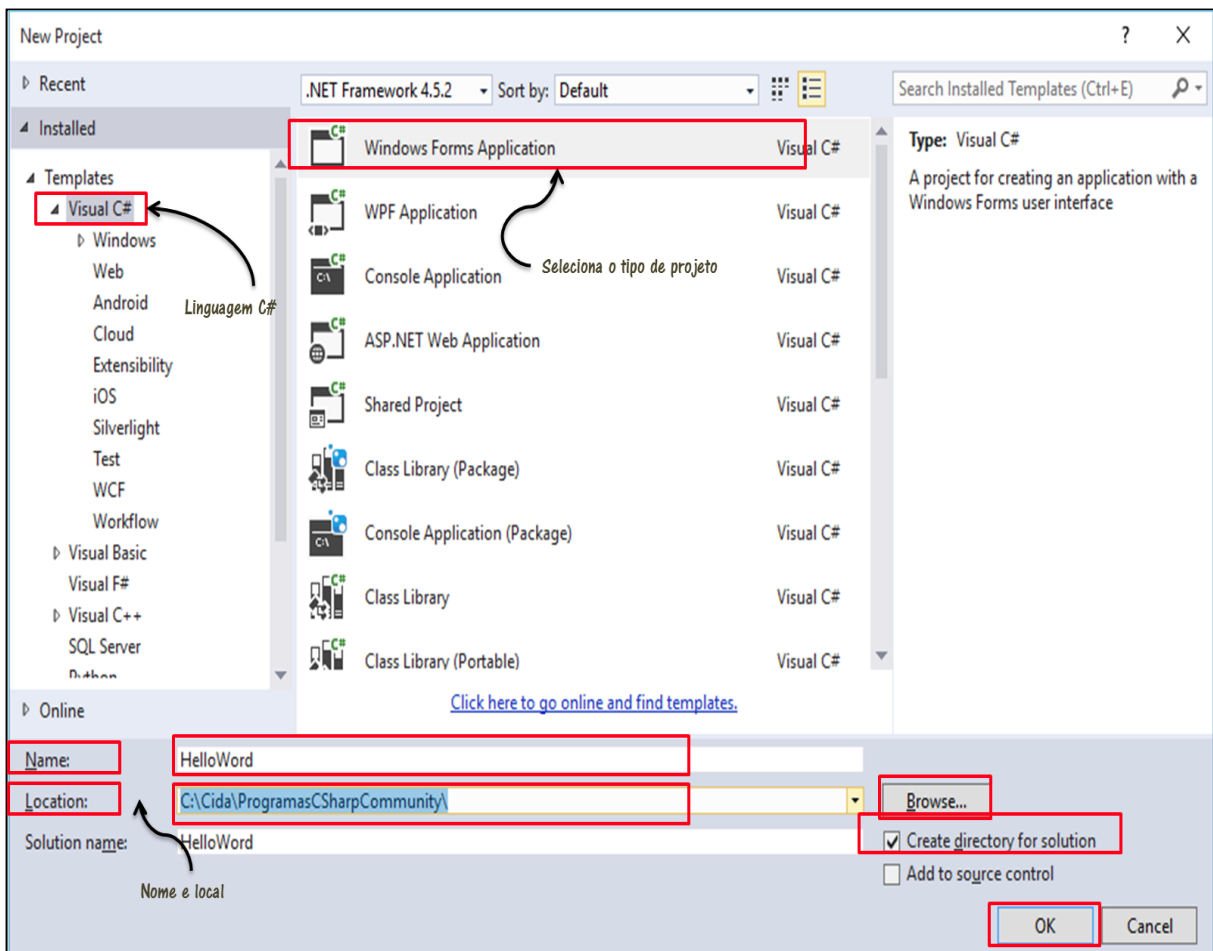
Quando o Visual Studio é executado, a *Start Page* (Página Inicial) é exibida conforme apresenta a figura abaixo.



Na parte central da Tela (*Start Page*) são apresentadas as novidades, notícias, links, vídeos que você poderá acessar clicando nos itens que desejar.

Na parte esquerda da Tela (*Start Page*) podemos criar um Novo Projeto (*New Project*), Abrir um Projeto (*Open Project*). Depois que você já tiver algum projeto criado, esses projetos aparecerão na lista *Recent* (Recentes) e você também poderá clicar no projeto que se encontra na lista para abri-lo.

Clique em *New Project* para criar um novo projeto. A tela para escolher o tipo de projeto e salvá-lo é apresentada, conforme figura a seguir.



Nessa tela são apresentados os Templates. Selecione a linguagem Visual C# (lado esquerdo da janela).

Na parte central da janela, escolha o tipo de aplicação. É importante que você selecione o item Windows Forms Application (*template* utilizado para criar aplicações Windows). Esse é o tipo de aplicações que vamos trabalhar com Programação Visual C#.

Depois defina o nome do seu projeto em Name: (na parte inferior da Janela). Em Location: defina o local onde você deseja salvar o seu projeto. Você poderá selecionar a pasta onde deseja salvar o projeto clicando no botão Browse.

Mantenha selecionada a caixa Create directory for solution. Dessa forma, todos os arquivos do seu projeto (*solution*) serão mantidos em um diretório.

Depois de configurar todos os itens acima, clique no botão OK.

O Visual Studio apresentará a janela na qual vamos desenvolver nossas aplicações.

Entendendo a Janela de Desenvolvimento do Visual Studio

Quando iniciamos o desenvolvimento de um aplicativo para Windows Forms Application, o Visual Studio já nos fornece um Form (Formulário) no qual iremos inserir os nossos componentes (ou controles), como por exemplo, botões, labels, caixa de texto, etc.

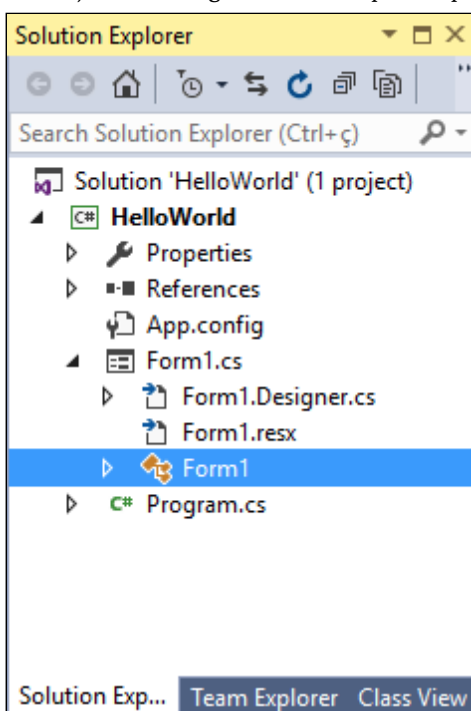
Na janela principal do Visual Studio você tem acesso a tudo o que precisa para desenvolver seus aplicativos. As janelas que são apresentadas também podem ser configuradas e podem aparecer de forma diferente no seu ambiente de desenvolvimento. Caso deseje você poderá restaurar o layout das janelas. Clique no menu Window e depois em Reset Window Layout.

Todas as janelas apresentadas a seguir são muito importantes para você trabalhar e ter maior produtividade. As janelas Solution Explorer, Toolbox e Properties são as mais utilizadas. Por isso é importante conhece-las.

Janela Solution Explorer

Quando você cria um novo projeto, o Solution Explorer exibe todo o conteúdo desse projeto, ou seja, essa janela lista todos os arquivos do seu projeto.

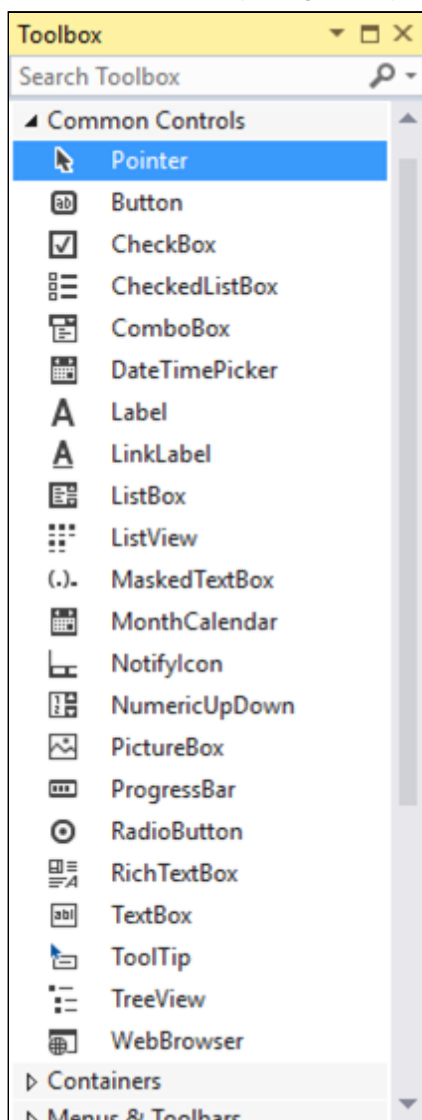
Nessa janela você gerencia os arquivos que fazem parte do projeto. Veja a figura a seguir.



Caso não esteja visualizando a janela do Solution Explorer, clique no menu View e depois em Solution Explorer, ou utilize as teclas de atalho Ctrl+Alt+L.

Janela Toolbox

A Toolbox é uma caixa de ferramentas que contém os controles (ou componentes) que podem ser utilizados para criar nossas aplicações. Utilizando *programação visual*, podemos escolher os controles que desejamos inserir no nosso Form. Basta selecionar o controle, clicar, “arrastar e soltar” no Form na posição que desejar inserir o controle. Veja a figura da janela Toolbox.



Provavelmente, os controles que você utilizará com maior frequência serão: Label, TextBox e Button.

Caso você não esteja visualizando a sua Toolbox, clique no menu View e depois em Toolbox, ou utilize as teclas de atalho Ctrl+Alt+X.

Janela Properties

A janela Properties (Janela de Propriedade) permite que você manipule as propriedades de um Form ou controle. As propriedades especificam informações sobre um controle, como por exemplo, tamanho, cor, texto, posição, name, etc. Cada controle tem seu próprio conjunto de propriedades. Quando você seleciona um controle (componente), do lado direito da janela Properties aparece o nome da propriedade e do lado esquerdo o valor para essa propriedade. Quando fazemos alguma alteração na propriedade, esta ficará grafada em negrito.

Ícones da barra de ferramentas dessa janela ajudam a organizar as propriedade em ordem alfabética (clicando no ícone *Alphabetical*) ou por ordem de categoria (clicando no ícone *Categorized*). Veja a figura a seguir da janela Properties.

Properties


Form1 System.Windows.Forms

HelpButton

False

+

Icon

 (Icon)

ImeMode

NoControl

IsMdiContain

False

KeyPreview

False

Language

(Default)

Localizable

False

+

Location

0; 0

Locked

False

MainMenuSt

(none)

MaximizeBox

True

+

MaximumSize

0; 0

MinimizeBox

True

+

MinimumSize

0; 0

Opacity

100%

+

Padding

0; 0; 0; 0

RightToLeft

No

RightToLeftLa

False

ShowIcon

True

ShowInTaskb

True

+

Size

300; 300

SizeGripStyle

Auto

StartPosition

WindowsDefa

Tag

Text

Form1

Caso não esteja visualizando a janela Properties, clique no menu View e depois em Properties Window ou utilize a telca de atalho F4.

Agora que você já conhece o IDE do Visual Studio e os conceitos de programação visual vamos conhecer a estrutura de um programa em Visual Studio com C#.

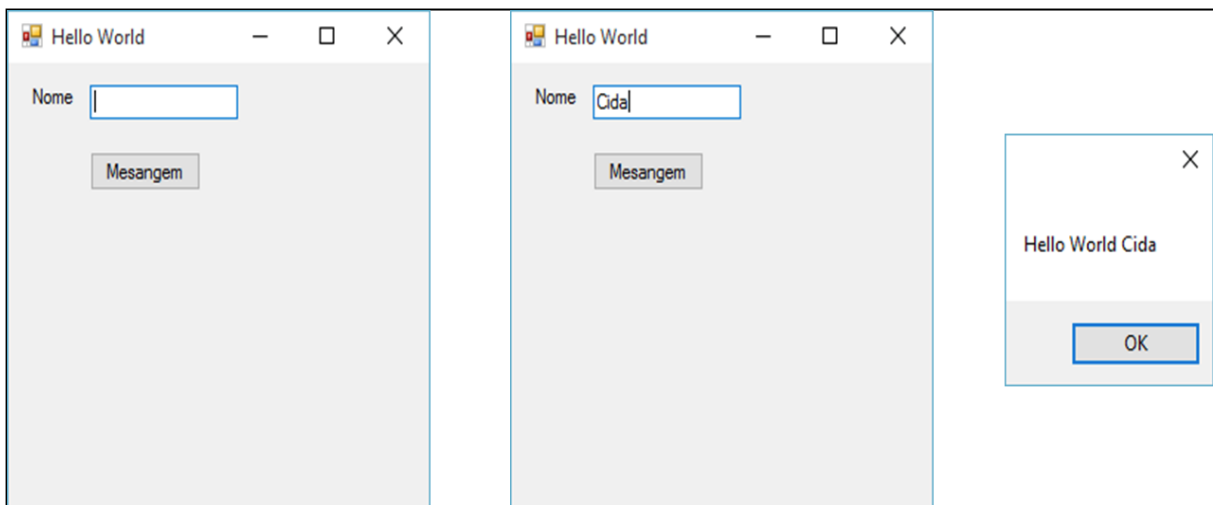
Estrutura de um programa C# utilizando o Visual Studio

Para entendermos melhor a estrutura de um programa em Visual Studio C#, vamos começar desenvolvendo um programa bem simples e também vamos conhecer três controles importantes do Visual C#. O nosso projeto tem:

- a. Label (etiqueta/rótulo): utilizada para exibir informações ao usuário;
- b. TextBox (caixa de texto): utilizado para entrada ou visualizações de dados.
- c. Button (botão): utilizado para disparar um evento (ação) por meio do clique.

Objetivo: O usuário deverá preencher o seu nome na textBox e quando pressionar o botão, aparecerá a mensagem “Hello World” e o nome digitado na textBox.

O resultado final do projeto é apresentado na figura abaixo.

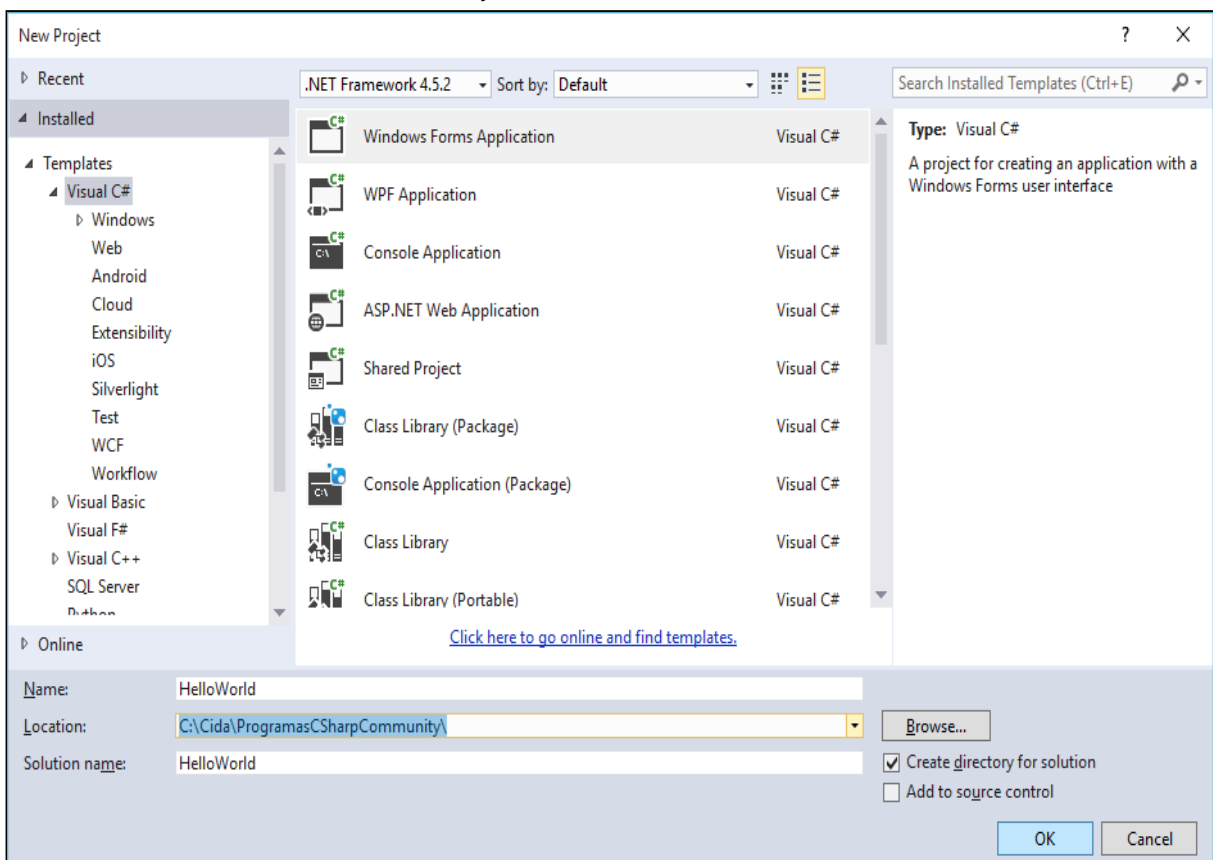


Vamos começar.

1. Inicie o Visual Studio (se ainda não o fez).
2. Na Start Page escolha New Project, ou caso a Start Page não esteja visível, na barra de menu escolha File, New Project.

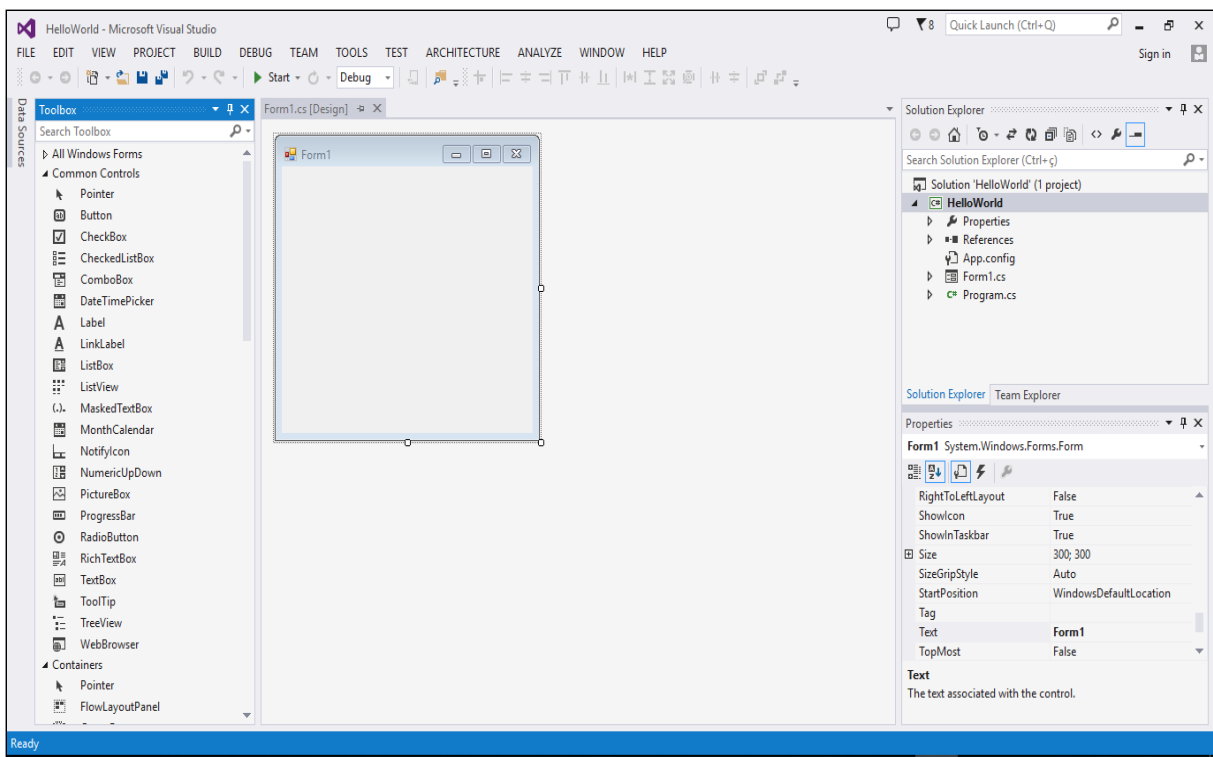
A caixa de diálogo é exibida.

1. Selecione no painel à esquerda o template Visual C#, no painel central escolha Windows Forms Application.
2. Preencha na caixa de texto Name: HelloWorld (tudo junto mesmo).
3. Location: escolha o local onde deseja salvar o seu projeto.
4. Mantenha selecionado: Create directory for solution.



Clique em OK.

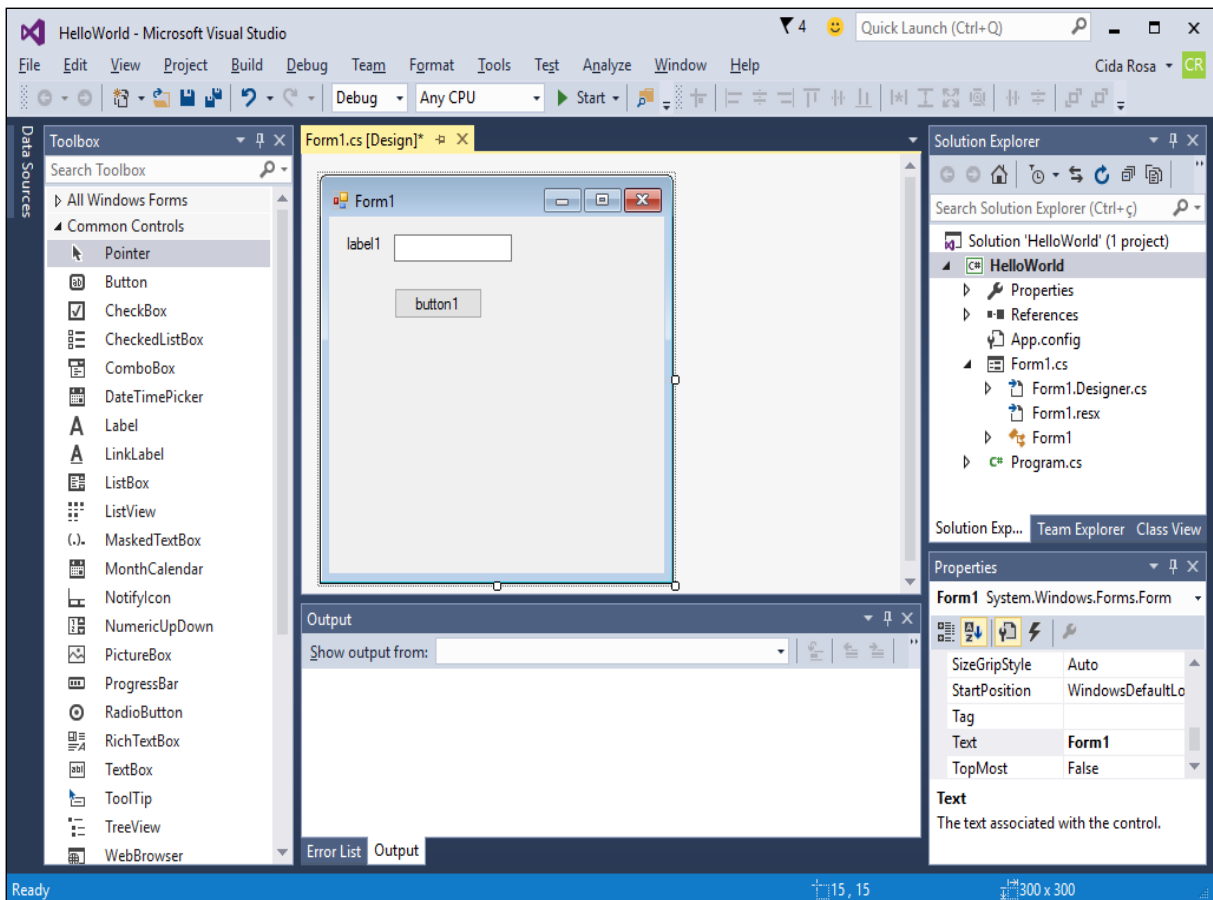
O IDE do Visual Studio cria o projeto para você começar a trabalhar.



Criando o design do Form

1. Após fixar (Auto Hide) a Toolbox, abra o nó Common Controls, selecione o controle Label, clique, “segure e arraste” a Label no canto superior esquerdo do Form.
2. Selecione o controle TextBox e coloque-o ao lado da Label no Form.
3. Selecione o controle Button e coloque abaixo da textBox no Form.

Você deverá ter um layout parecido com a figura a seguir.



Propriedades dos controles

Cada controle possui um conjunto de propriedades. Vamos ver agora as propriedades dos controles que vamos alterar nesse exemplo.

Name: a propriedade name define um nome único para o seu controle, ou seja, estamos dando uma identificação para o controle e que deve ser única.

Text: a propriedade text de uma TextBox é utilizada para inserir ou receber dados. Também é utilizada para apresentar o texto que vai aparecer em uma Label, o título do Form ou o rótulo de um Button.

Agora vamos trabalhar nas propriedades desses controles.

1. Selecione o Form e altere as propriedades conforme definido abaixo:

Propriedade	Valor
Text	Hello World

2. Selecione o controle Label e altere as propriedades conforme definido abaixo:

Propriedade	Valor
(name)	lblNome
Text	Nome

3. Selecione o controle TextBox e altere as propriedades conforme definido abaixo:

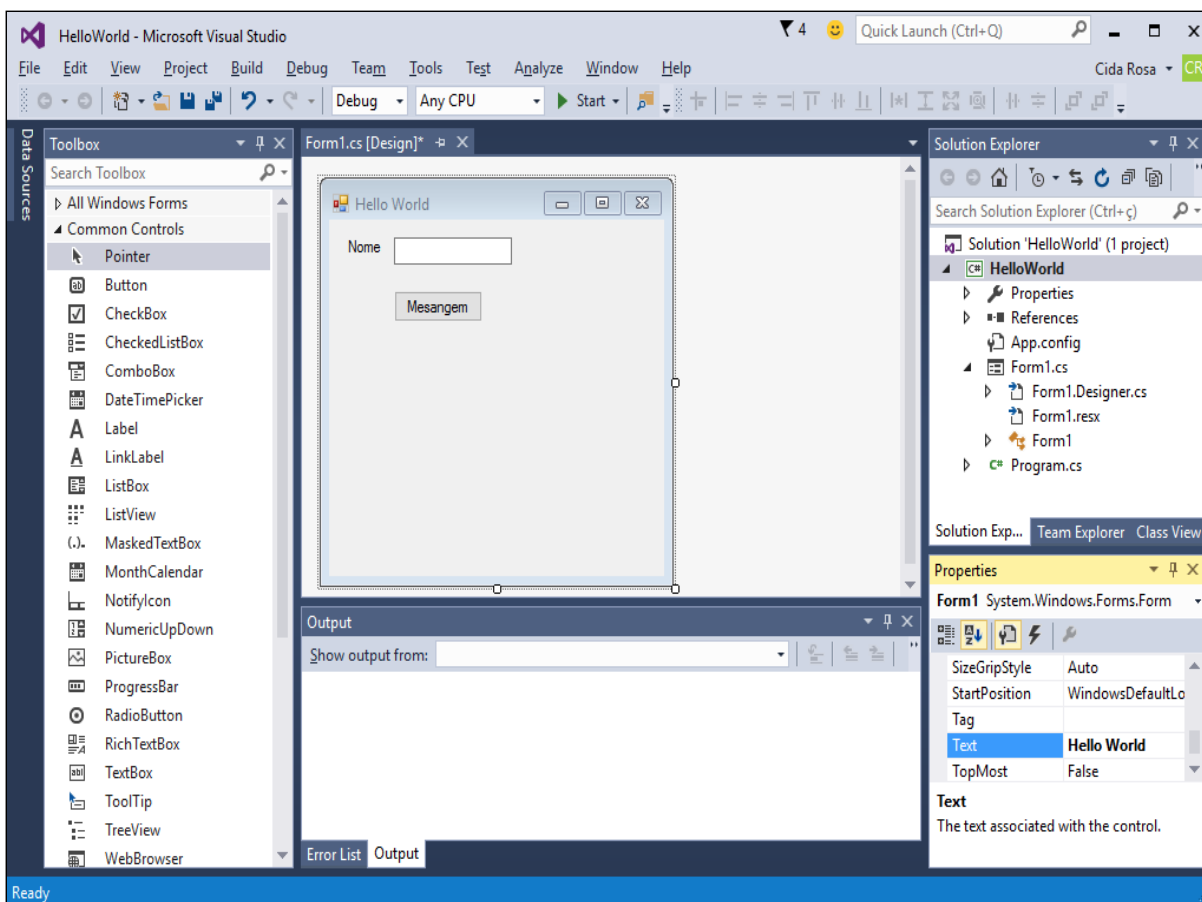
Propriedade	Valor
(name)	txtNome

4. Selecione o controle Button e altere as propriedades conforme definido abaixo:

Propriedade	Valor
(name)	btnMensagem
Text	Mensagem

Obs.: a propriedade name não é visível para o usuário. A propriedade name deve ser única para cada controle/objeto. Não podemos ter mais de um controle, componente, objeto, variável, etc com o mesmo name.

Nesse momento você deverá ter um layout parecido com a figura a seguir.

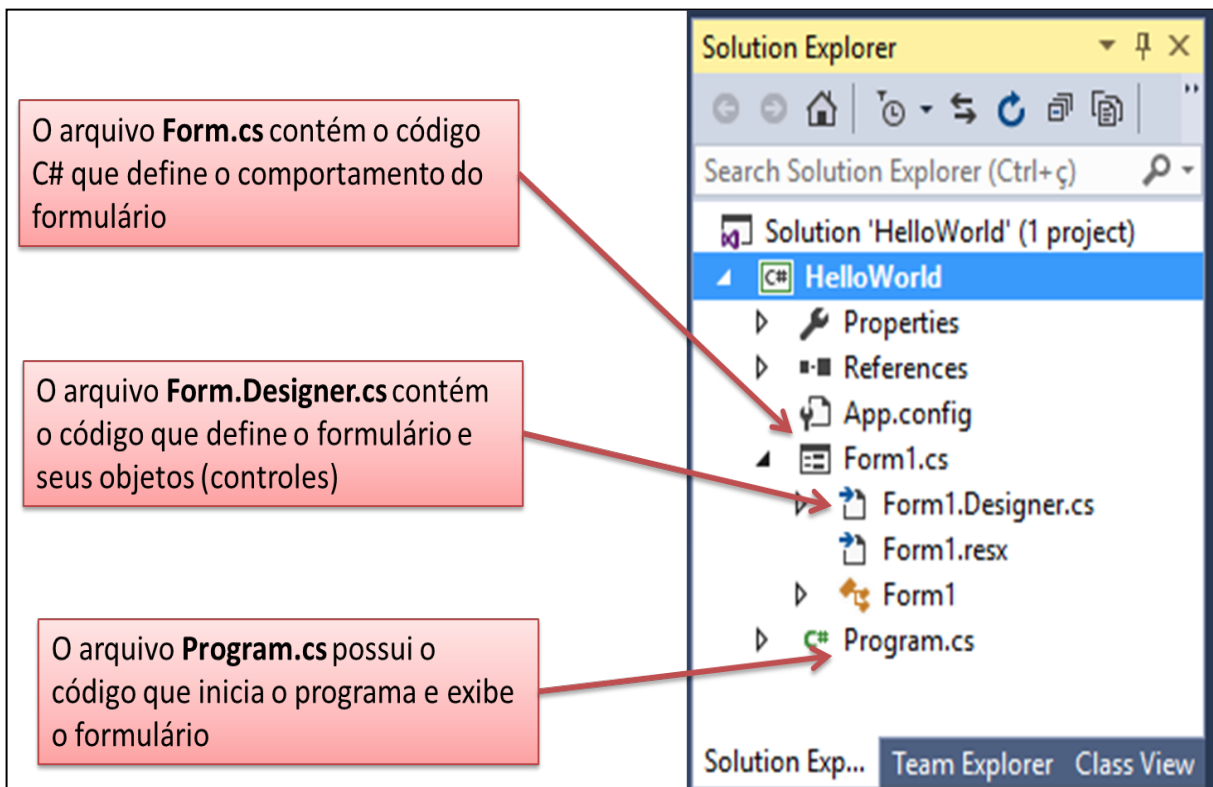


Entendendo os arquivos gerados pelo Visual Studio

Antes de começarmos a escrever o código para o nosso projeto, vamos entender o que o IDE fez.

Assim que você salvou o seu projeto, o Visual Studio criou esses três arquivos automaticamente: Form1.cs, Form1.Designer.cs e Program.cs. Esses arquivos estão na janela Solution Explorer.

- Form1.cs: contém o código C# que define o comportamento do formulário.
- Form.Designer.cs: contém o código que define o formulário e seus objetos.
- Program.cs: contém o código que inicia o programa e exibe o formulário.

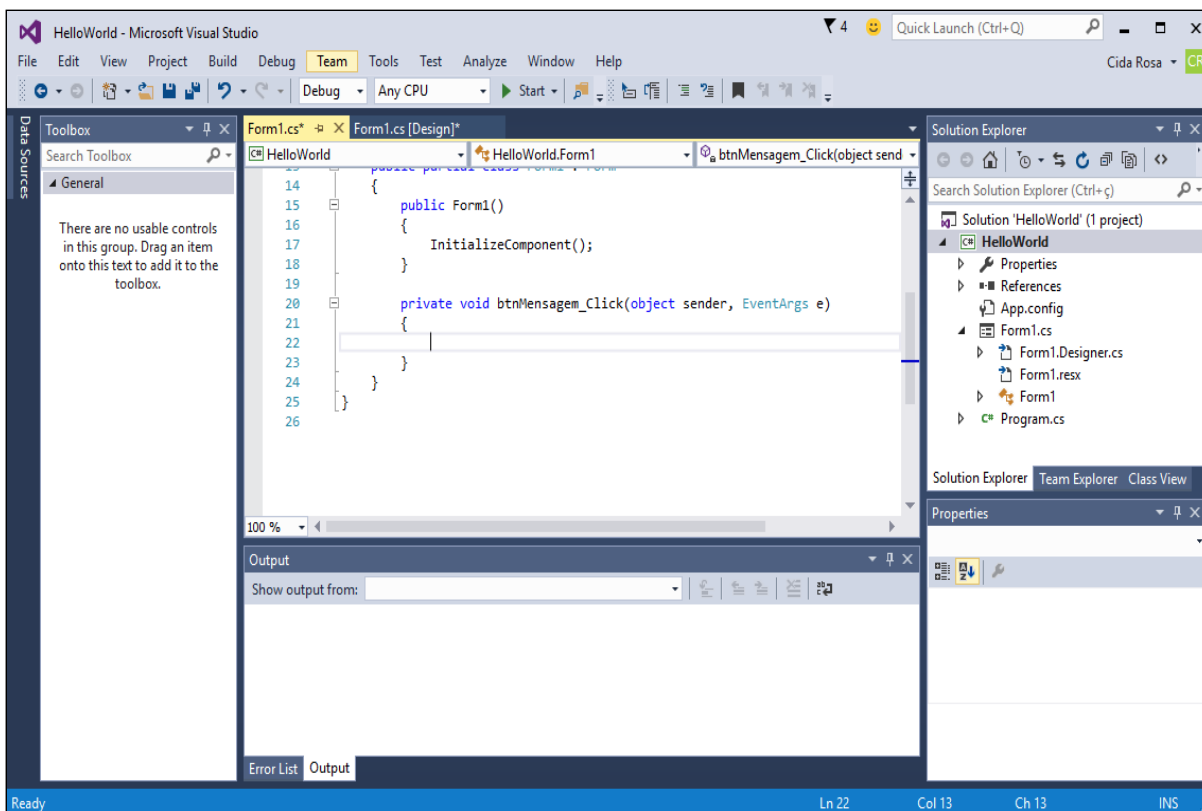


Escrevendo o código do programa Hello World

Agora que você já criou a interface e entendeu como as coisas funcionam no Visual C#, deve estar pronto para começar a codificar.

Uma forma mais rápida e simples para começar é clicando duas vezes no botão que criamos. Você tem que estar em modo de desenvolvimento e não executando o seu programa.

Quando damos duplo clique no controle que queremos codificar, podemos observar que o IDE já criou muito código, mas ainda temos que acrescentar o nosso código. Você deve ver algum código semelhante ao apresentado a seguir.



Veja que o IDE criou o código para o nosso botão btnMensagem para o evento clique. É dentro desse par de chaves ({ }) que iremos escrever o nosso código. Veja o código a seguir.

```
1. MessageBox.Show("Hello World " + txtNome.text);
```

Entendendo linha de código que foi digitada:

```
namespace HelloWorld
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnMensagem_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Hello World " + txtNome.Text);
        }
    }
}
```

Quando você clicou duas vezes no botão, o IDE criou esse método. Ele será executado sempre que o usuário clicar no botão, com a aplicação em execução.

Esse nome do método dá uma boa ideia sobre quando ele será executado. No evento click do botão.

Digite essa linha de código e não esqueça de colocar os parênteses e o ponto-e-vírgula no final

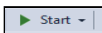
Quando você clicou duas vezes no botão, o IDE abriu esse código e deixou o cursor piscando aqui para você digitar o seu código.

O `MessageBox` é um objeto que apresenta uma caixa de mensagem na tela.

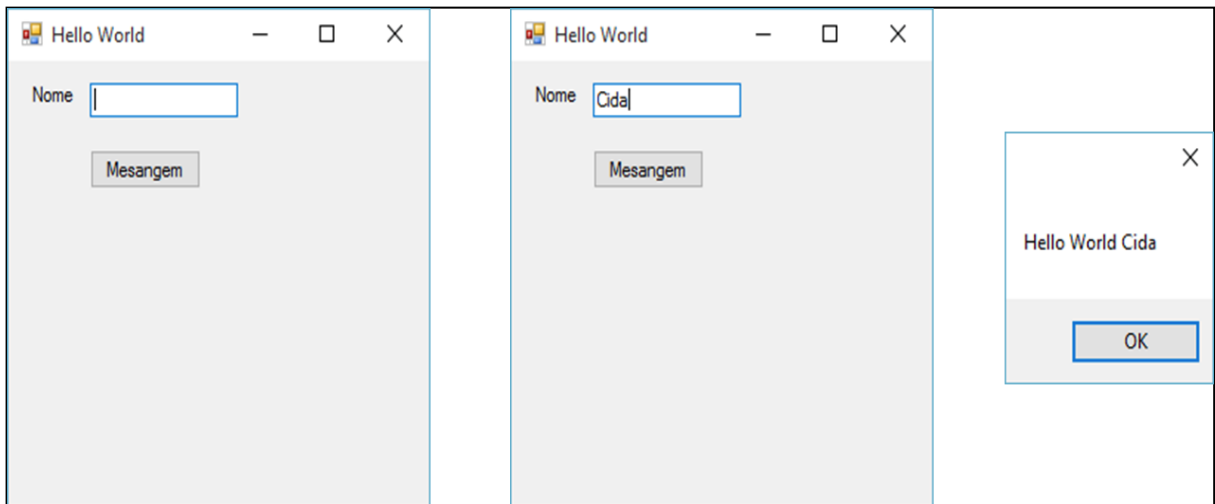
Dentro do método `Show` codificamos o texto "Hello World ". O sinal de mais (+), significa concatenar, ou seja, unir textos.

O nome que o usuário digitar na nossa `txtNome` (TextBox) está armazenado na propriedade `Text`. Dessa forma, para recuperar o texto que foi digitado colocamos o nome do controle seguido de um ponto (.) e a propriedade `text` (`txtNome.text`).

Sempre que quisermos recuperar o texto armazenado em um controle, colocamos o nome do controle, ponto, a propriedade `text`.

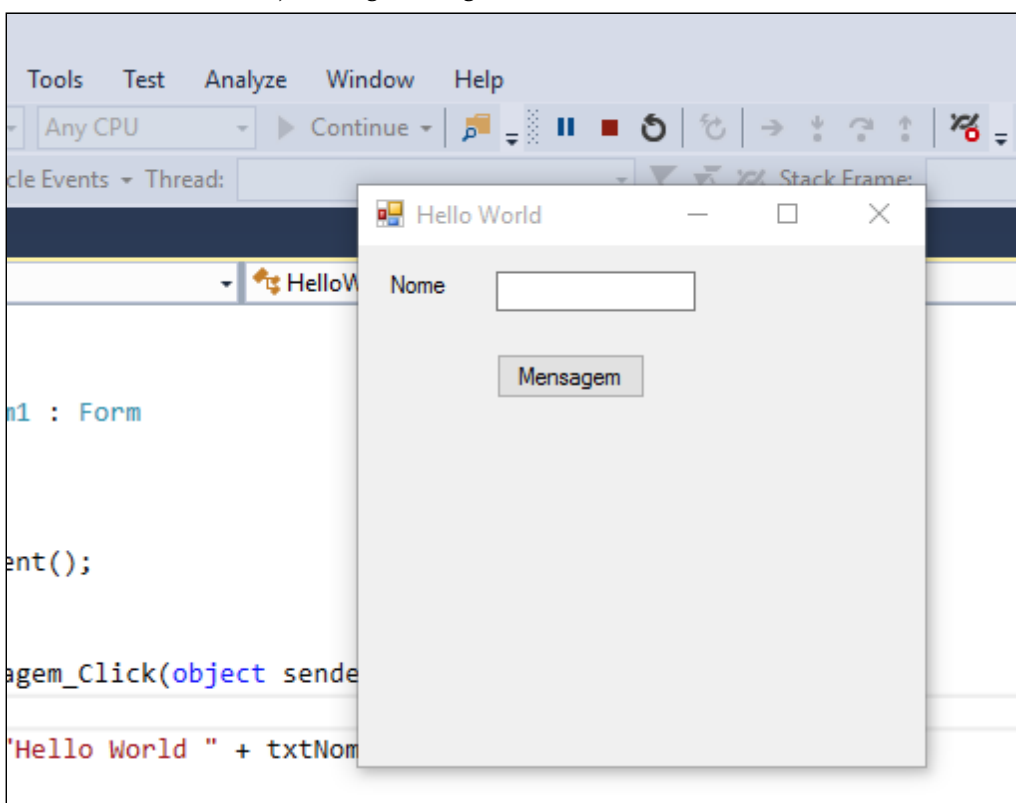
Para executar o programa, pressione a tecla F5 ou clique no ícone Start () na barra de menu.

Veja na figura abaixo o resultado final do programa HelloWorld sendo executado.



Para finalizar, feche o programa Hello World e volte ao Visual Studio.

Você também poderá encerrar o programa Hello World pelo Visual Studio de várias formas: acesse o menu Debug, Stop Debugging; ou pressionando as teclas Shift+F5; ou clicar no botão stop localizado na barra de ferramentas. Veja a imagem a seguir.



Para fixar todos esses conceitos pratique. Faça o exemplo apresentado.

Não deixe de assistir ao vídeo do passo-a-passo para desenvolver o programa Hello World.



Legenda: VÍDEO PASSO-A-PASSO PARA O DESENVOLVIMENTO DO PROGRAMA HELLO WORLD EM LINGUAGEM C#.

ATIVIDADE

Quando criamos um projeto em Visual C# do tipo Windows Forms Application, no IDE do Visual Studio, ele gera automaticamente três arquivos: Form1.cs, Form.Designer.cs e Program.cs.

Selecione a alternativa correta que define o que cada um desses arquivos contém.

- A. Form.Designer.cs: contém o código C# que define o comportamento do formulário.
Form1.cs: contém o código que define o formulário e seus objetos.
Program.cs: contém o código que inicia o programa e exibe o formulário.
- B. Program.cs: contém o código C# que define o comportamento do formulário.
Form1.cs: contém o código que define o formulário e seus objetos.
Form.Designer.cs: contém o código que inicia o programa e exibe o formulário.
- C. Program.cs: contém o código C# que define o comportamento do formulário.
Form.Designer.cs: contém o código que define o formulário e seus objetos.
Form1.cs: contém o código que inicia o programa e exibe o formulário.
- D. Form1.cs: contém o código C# que define o comportamento do formulário.
Form.Designer.cs: contém o código que define o formulário e seus objetos.
Program.cs: contém o código que inicia o programa e exibe o formulário.

ATIVIDADE

Em aplicações com interface gráficas (GUI) que fazem uso de controles como Janelas, Botões, Labels, Caixas de Texto, etc., a tarefa de programação não acompanha uma sequência lógica idêntica à dos programas desenvolvidos para modo texto, como por exemplo na linguagem C.

Considere as afirmações a seguir:

- I. Programador desse tipo de ambiente precisa, além de se preocupar com a lógica do programa, tem que prever as ações do usuário.
- II. Esse tipo de programação é chamada de Programação Orientada a Evento.
- III. O programa é composto por uma única sub-rotina que fornece a lógica do programa e dos eventos.
- IV. Os eventos são programados nos controles que compõem o form da aplicação.

É correto apenas o que se afirma em

- A. I, II e IV
- B. I, II e III
- C. I, III e IV
- D. II, III, e IV

ATIVIDADE

No desenvolvimento de aplicações do tipo Windows trabalha-se com controles como Label, TextBox, Button, Form, etc.

Considere as afirmações a seguir.

- I. Um Form é um controle (componente) do tipo container, ou seja, todos os outros controles serão inseridos no Form.
- II. Um controle do tipo Label é utilizado para o usuário inserir dados.
- III. Para recuperar o texto digitado em uma TextBox, utiliza-se a propriedade Name.
- IV. O controle Button é utilizado para disparar um evento quando o usuário clicar nele.

É correto apenas o que se afirma em

- A. I e II
- B. I e IV
- C. I e III
- D. II, III, e IV

DICAS

Sempre consulta o site da Microsoft para tirar suas dúvidas ou ficar atualizado.

O site do Macoratt (<http://www.macoratti.net>) também traz vários exemplos explicativos sobre as ferramentas e linguagens da Microsoft.

REFERÊNCIA

DEITEL, H. M. *et al.* C# Como Programar. São Paulo: Pearson Makron Books, 2003.

STELLMAN, Andrew; GREENE, Jennifer. *Head First C#*. 3rd. ed. O'Reilly Media, Inc., 2013.

SITES

Instruções passo a passo: criar um aplicativo simples com o Visual C# ou o Visual Basic. Disponível em <https://msdn.microsoft.com/pt-br/library/jj153219.aspx#bkmk_configureide (https://msdn.microsoft.com/pt-br/library/jj153219.aspx#bkmk_configureide)>. Acesso em: 07/09/2015.

Chapter 2: Writing a C# Program. Disponível em <[https://msdn.microsoft.com/pt-BR/Library/hh145615\(v=vs.88\).aspx](https://msdn.microsoft.com/pt-BR/Library/hh145615(v=vs.88).aspx) ([https://msdn.microsoft.com/pt-BR/Library/hh145615\(v=vs.88\).aspx](https://msdn.microsoft.com/pt-BR/Library/hh145615(v=vs.88).aspx))>. Acesso em 07/09/2015

