



DESENVOLVIMENTO DE SISTEMAS COM C#

Cleverson Lopes Ledur

Revisão técnica:

Jeferson Faleiro Leon

*Desenvolvimento de Sistemas
Especialista Formação Pedagógica de Professores
Professor do curso Técnico em informática*



L475d Ledur, Cleverson Lopes.

Desenvolvimento de sistemas com #C [recurso eletrônico] / Cleverson Lopes Ledur; [revisão técnica: Jeferson Faleiro Leon]. – Porto Alegre : SAGAH, 2018.

ISBN 978-85-9502-314-7

1. Ciência da computação. 2. Linguagens de programação de computador. I. Título.

CDU 004.43

Conhecer o ambiente de desenvolvimento Visual Studio

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Instalar e configurar o Visual Studio Community 2017.
- Identificar as principais funcionalidades do IDE Visual Studio.
- Explicar a arquitetura do framework .NET.

Introdução

O Visual Studio é uma ferramenta que existe há mais de 10 anos e já permitiu a criação de diversos sistemas utilizados em todo o mundo. Durante todos esses anos, ocorreram grandes evoluções na plataforma. Hoje, programadores têm a possibilidade de criar aplicações desktop, web e multiplataformas para mobile. Entender como o Visual Studio funciona é muito importante para desenvolvedores de sistemas que desejam criar aplicações em C#. Atualmente, o Visual Studio e o framework .NET oferecem um suporte muito amplo em termos de linguagens de programação. Dessa forma, eles permitem que desenvolvedores escrevam seus códigos em C, C++ e C++/CLI, VB.NET, C#, F# e TypeScript.

Neste texto, você irá explorar os recursos do Visual Studio. Também irá entender o framework .NET, que é a base de desenvolvimento que utilizará. Além disso, você vai conhecer a arquitetura desse poderoso framework.

Introdução e instalação

O Microsoft Visual Studio é um ambiente de desenvolvimento integrado (IDE – *Integrated Development Environment*) da Microsoft. Ele é usado

para desenvolver programas de computador para o Microsoft Windows, bem como sites, aplicativos da web, serviços da web e aplicativos para dispositivos móveis. O Visual Studio usa plataformas de desenvolvimento de software da Microsoft, como Windows API, Windows Forms, Windows Presentation Foundation, Windows Store e Microsoft Silverlight. Além disso, pode produzir código nativo e código gerenciado (HILLAR, 2010).

Ele também inclui um editor de código que suporta o IntelliSense (o componente de conclusão do código), bem como a refatoração de código. O depurador integrado funciona como um depurador de nível de origem e um depurador de nível de máquina. Outras ferramentas incorporadas incluem um perfilador de código (também chamado de *code profiler*), um designer de formulários para criação de aplicativos GUI (interface gráfica), um web designer, um designer de classe e um designer de esquema de banco de dados. O Visual Studio aceita plug-ins que aprimoram a funcionalidade em quase todos os níveis. Isso inclui o acréscimo de suporte para sistemas de controle de origem (como Subversion e GIT), por exemplo. Também é possível adicionar novos conjuntos de ferramentas, como editores e designers visuais, para idiomas, ou conjuntos de ferramentas específicos do domínio para outros aspectos do ciclo de vida do desenvolvimento de software, como o cliente Team Foundation Server (AMANN et al., 2016).

O Visual Studio suporta aproximadamente 36 linguagens de programação diferentes. Também permite que o editor de código e o depurador suportem (em graus variados) praticamente qualquer linguagem de programação, desde que exista um serviço específico de idioma. As linguagens internas incluem C, C++ e C++/CLI, VB.NET, C#, F# e TypeScript. O suporte para outros idiomas, como Python, Ruby, Node.js e M, entre outros, está disponível por meio de serviços de idiomas instalados separadamente. Ele também suporta XML/XSLT, HTML/XHTML, JavaScript e CSS.



Saiba mais

As versões mais recentes do Visual Studio também oferecem suporte para o desenvolvimento de soluções para dispositivos móveis multiplataforma com o Xamarin. Além disso, a plataforma Azure, combinada com o Visual Studio, oferece um ambiente de desenvolvimento em nuvem computacional. Assim, permite a criação de aplicações robustas e escaláveis.

Bem, agora que você já sabe o poder dessa ferramenta, que tal conhecê-la melhor na prática? Mas, antes, você precisa instalá-la no seu computador. Para isso, siga os seguintes passos:

- **Faça o download do Visual Studio.** Você pode baixar a versão mais atual do Visual Studio na seguinte URL: www.visualstudio.com (Figura 1). A partir deste passo a passo, você irá instalar o Visual Studio Community 2017.

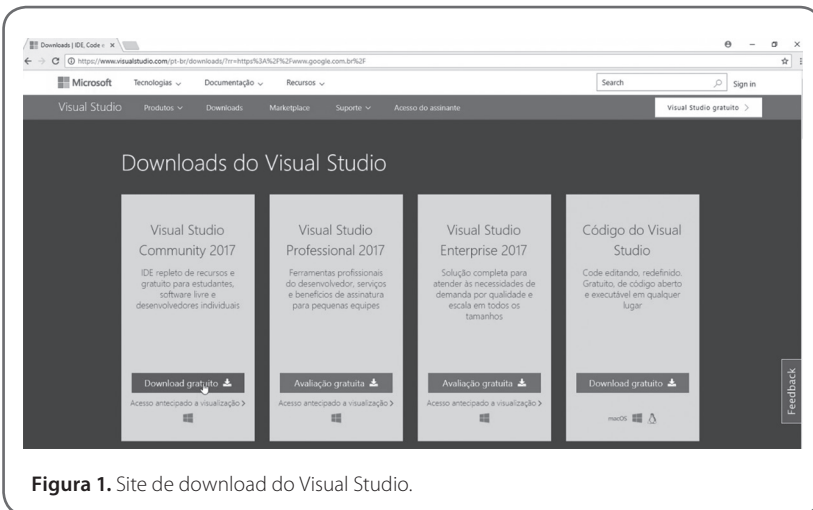


Figura 1. Site de download do Visual Studio.

1. **Execute o arquivo de instalação do Visual Studio Installer.** Esse novo instalador leve inclui tudo o que você precisa para instalar e personalizar o Visual Studio. Se você fez o download do Visual Studio Community, o arquivo de instalação deve estar com o nome **vs_community.exe**. Basta executá-lo para iniciar o processo.
2. Se você receber um aviso de Controle de Conta de Usuário, clique em **Sim**.
3. Na tela de Termos de Licença da Microsoft e Declaração de Privacidade da Microsoft, faça a leitura e, se concordar, marque que aceita as condições estabelecidas. Clique em **Continuar** (Figura 2).



Figura 2. Tela inicial de instalação e aceitação dos Termos de Licença.

Visual Studio

Nos dê um minuto, acabaremos logo...

Cancelar

Figura 3. Progresso da preparação da instalação do Visual Studio.

4. **Selecione a carga de trabalho que irá utilizar.** Nesse momento, você vai selecionar apenas *Desenvolvimento para desktop com o .NET* (Figura 4). Caso no futuro você precise instalar outra carga de trabalho, pode selecioná-la e instalá-la separadamente.

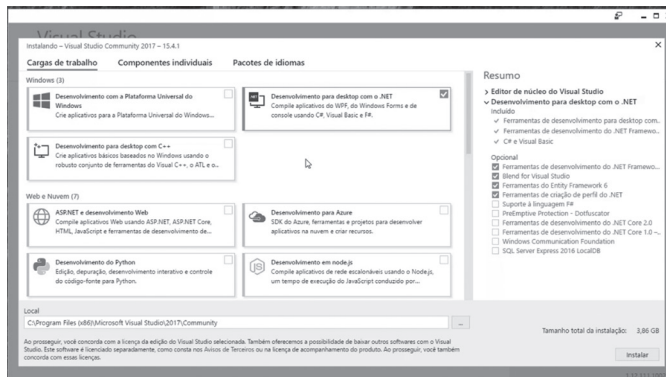


Figura 4. Tela da instalação do Visual Studio em que cargas de trabalho, componentes individuais e pacotes de idiomas são selecionados.

5. Depois de selecionar a(s) carga(s) de trabalho que deseja, clique em **Instalar**.
6. **Acompanhe o progresso da instalação.** Em seguida, as telas de status mostram o progresso da instalação do Visual Studio (Figura 5). Você pode aproveitar para ler sobre as informações da interface enquanto a instalação é realizada.

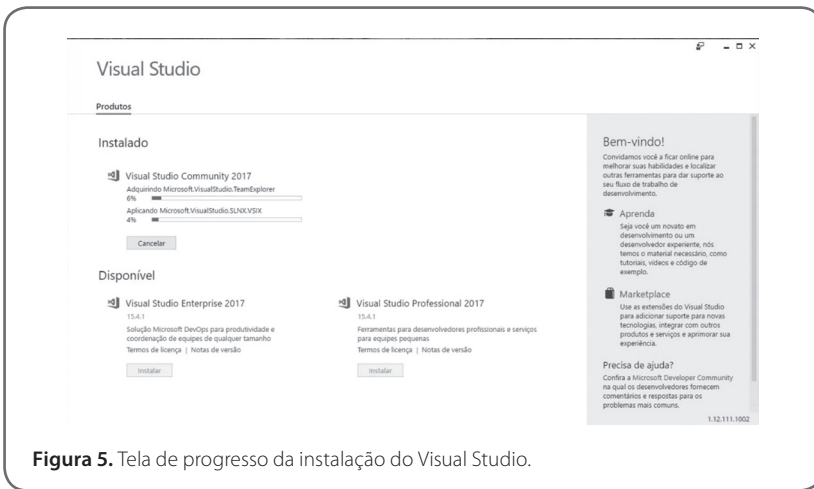


Figura 5. Tela de progresso da instalação do Visual Studio.

7. Assim que a etapa de instalação é realizada, é exibida uma tela solicitando a reinicialização do sistema. É muito importante que você reinicie o sistema nesse momento para que a instalação seja realizada de forma completa. Então, salve seus arquivos, feche os seus programas e clique em **Reiniciar** (Figura 6).

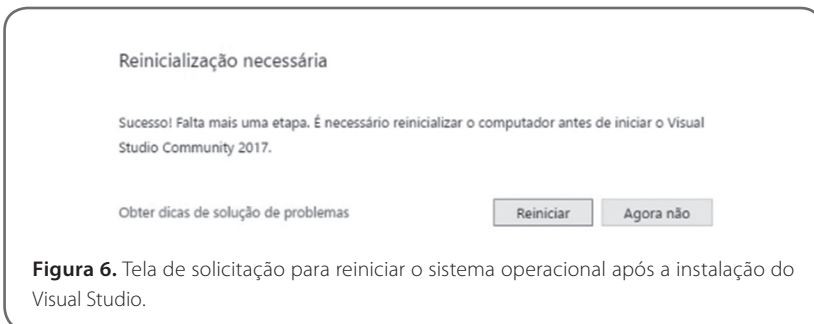


Figura 6. Tela de solicitação para reiniciar o sistema operacional após a instalação do Visual Studio.

Utilização do Visual Studio e principais elementos

Depois que o seu computador reiniciar, você já pode iniciar o Visual Studio, já que as novas cargas de trabalho e os componentes foram instalados. Para isso, clique em **Iniciar** e digite *Visual Studio* no campo de busca (Figura 7). Clique no ícone do Visual Studio 2017. Se desejar, já pode adicioná-lo à sua área de trabalho ou à sua barra de tarefas para otimizar seu tempo na próxima vez em que for utilizá-lo.

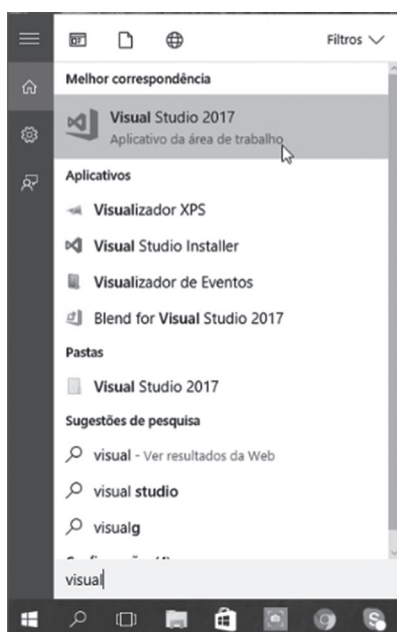
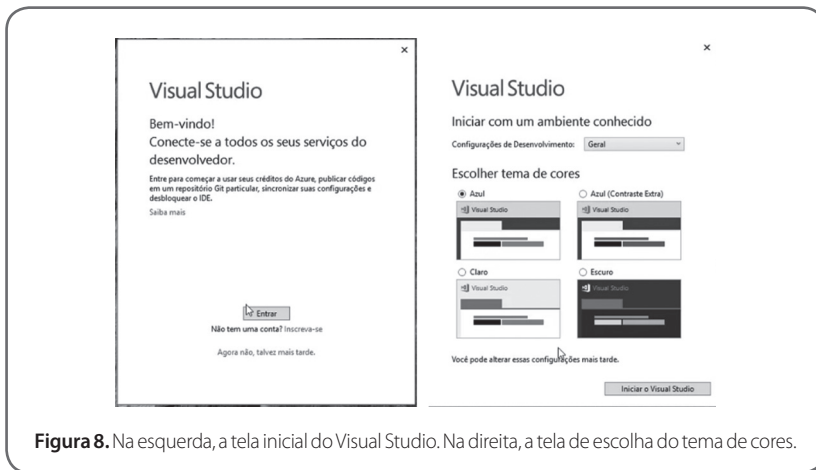
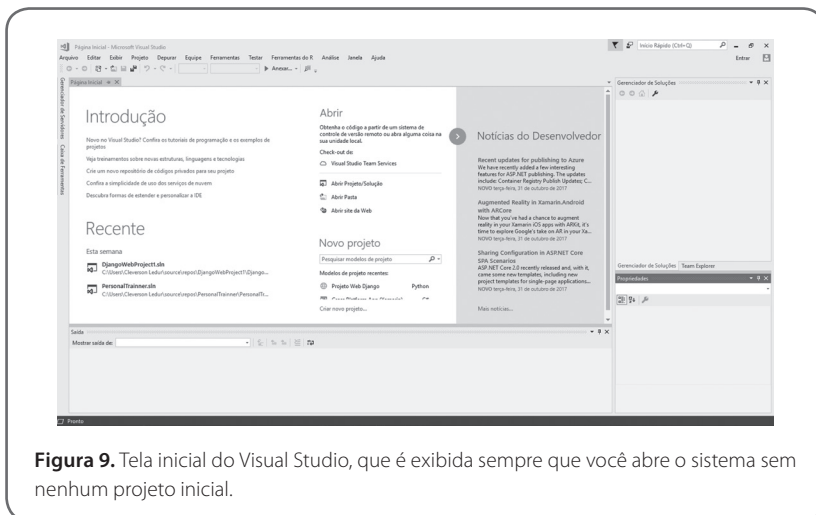


Figura 7. Tela exibindo o ícone para iniciar o Visual Studio após a instalação.

O Visual Studio apresenta inicialmente uma tela para você realizar login com sua conta Microsoft. Caso você deseje pular essa etapa, que é opcional, basta clicar em **Agora não, talvez mais tarde**. Então, você será direcionado para a tela de escolha das cores da ferramenta. Você pode escolher o conjunto de cores que melhor se adapta à sua visão (Figura 8). Se não gostar do resultado, pode alterar as cores diretamente na interface do Visual Studio, em **Opções > Ferramentas > Ambiente > Tema de cor**.



Se todas as etapas anteriores ocorreram com sucesso, você chegará na interface do Visual Studio. Agora, você iniciará seu aprendizado nessa poderosa ferramenta que permite a criação de sistemas robustos e eficientes. Está pronto para iniciar essa grande jornada pelo mundo do desenvolvimento de sistemas?



Para conhecer os principais elementos do Visual Studio, você vai primeiro iniciar um projeto. Para isso, clique em **Arquivo** e, em seguida, clique em **Novo e depois em projeto** (Figura 10).

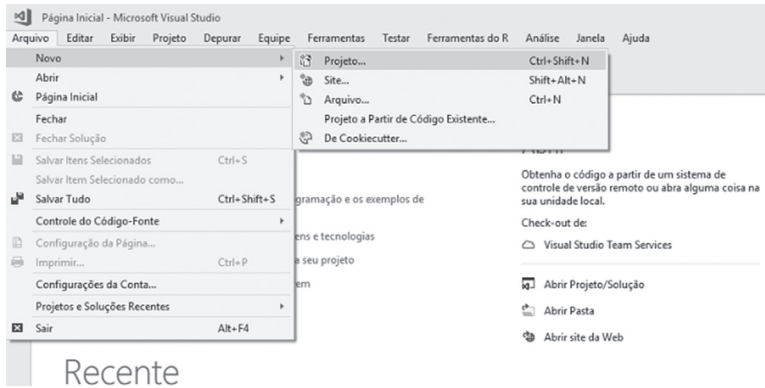


Figura 10. Criação de um projeto no Visual Studio.

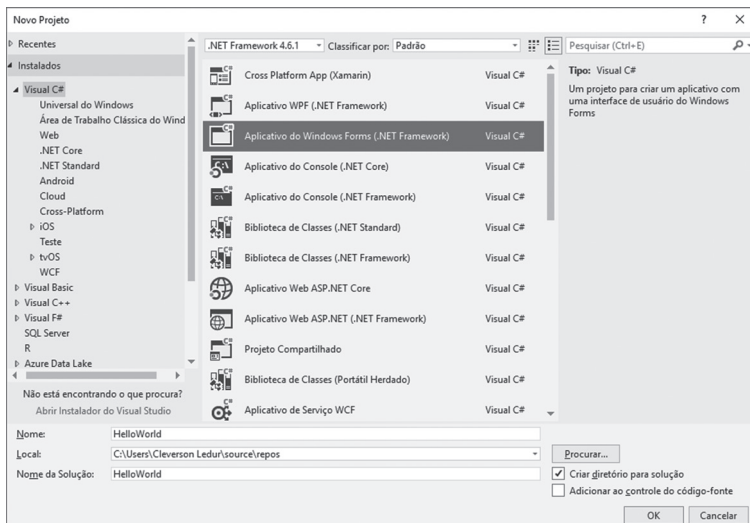


Figura 11. Tela de novo projeto do Visual Studio, em que é criado um projeto C# com Windows Forms.

Selecione o tipo de projeto como **Visual C# > Aplicativo do Windows Forms (.NET Framework)**. No campo **Nome**, você pode digitar o nome do projeto. Por convenção, você deve iniciar o nome dos projetos com letras maiúsculas em cada palavra e não deve utilizar caracteres especiais. Após, basta clicar em **OK**.

Logo, você verá uma tela inicial do projeto, como mostra a Figura 12.

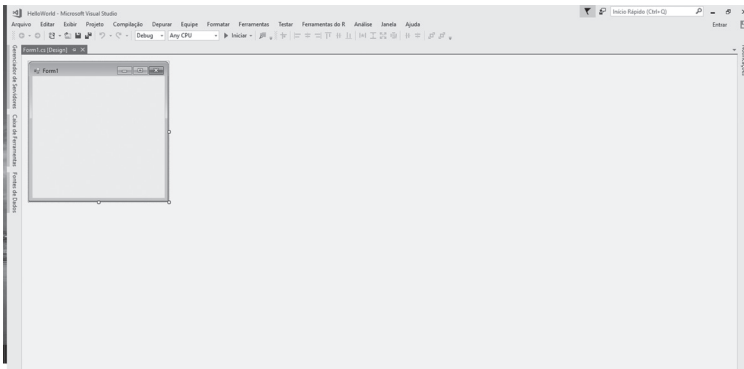


Figura 12. Tela inicial de um projeto Windows Forms.

Que tal analisar agora alguns elementos que estão presentes nessa tela? Clique nos itens dos lados esquerdo e direito para que sejam exibidos. A seguir, você vai conhecer cada um deles.

- **Gerenciador de Servidores:** permite a configuração e a seleção dos servidores em que você irá trabalhar e disponibilizar sua aplicação (Figura 13).

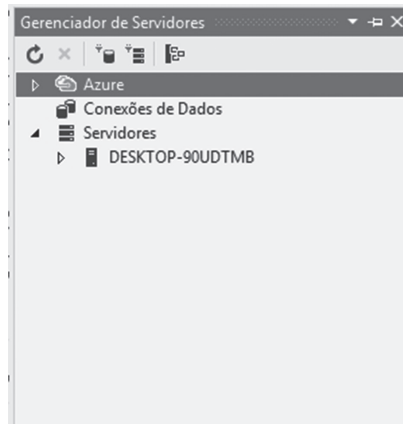


Figura 13. Gerenciador de Servidores do Visual Studio.

- **Caixa de Ferramentas:** nela você encontra todas as ferramentas para utilizar no seu projeto. Nesse caso, com Windows Forms, você pode simplesmente arrastar elementos da caixa de ferramentas para o seu projeto para inseri-los no sistema em desenvolvimento. Nesse elemento, cada ferramenta fica classificada pelo seu domínio, como menus e barras, dados, componentes e impressão. Assim, você pode facilmente encontrar o que precisa para o projeto (Figura 14).

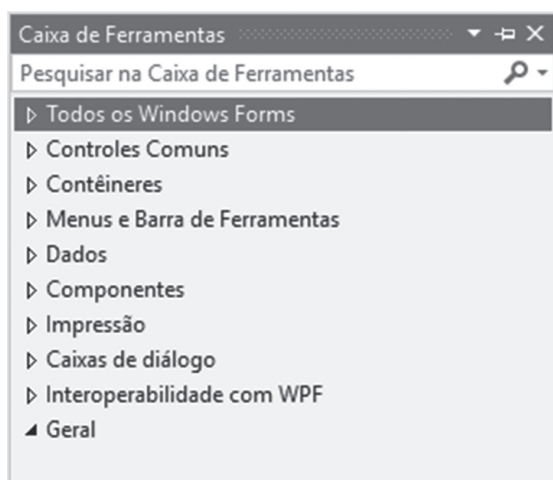


Figura 14. Caixa de Ferramentas do Visual Studio.

- **Fontes de Dados:** nesse item, você irá adicionar suas conexões com bancos de dados ou outras formas de persistência de dados. Nessa área do Visual Studio, há diversas opções de configuração das bases, que você pode escolher de forma bastante fácil e rápida (Figura 15).

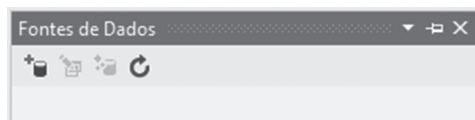


Figura 15. Fontes de Dados do Visual Studio.

No menu superior, você vai encontrar algumas opções. A seguir, você pode conhecer cada uma delas.

- **Iniciar:** esse botão permite que você execute o sistema em desenvolvimento para testar ou debugar. Ao clicar nele, você poderá interagir com o sistema como o usuário faria. Além disso, a ferramenta de debug permite adicionar breakpoints para depurar comportamentos do sistema (Figura 16).



Figura 16. Botão **Iniciar** (Run) do Visual Studio.

- **Busca:** você pode perder muito tempo tentando encontrar o local de funções dentro de um IDE. Assim como em diversas outras ferramentas, você pode utilizar o campo de busca do Visual Studio a fim de encontrar funções e controles para utilizar no sistema (Figura 17). Dessa forma, evita a necessidade de saber exatamente onde está cada controle. Faça um experimento: digite na barra de buscas a palavra *class* e veja as opções que são apresentadas. Você pode, por exemplo, criar uma nova classe apenas clicando em uma das opções apresentadas. Tente utilizar esse campo para aumentar sua produtividade!

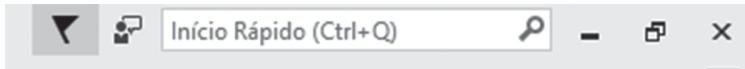


Figura 17. Barra de buscas do Visual Studio, que permite o fácil acesso às funcionalidades da ferramenta.

- **Saída:** ao clicar, por exemplo, em **Iniciar**, você receberá um elemento inferior de saída. Nesse elemento, você tem toda a saída do seu programa, que pode ser um log, um erro ou informações internas do sistema. Você pode utilizar essa janela para depurar e entender o comportamento do sistema (Figura 18).



Figura 18. Elemento de saída (output) do Visual Studio.

Agora, clique com o botão direito sobre o **Form** da aplicação. Você verá algumas opções, conforme a Figura 19.

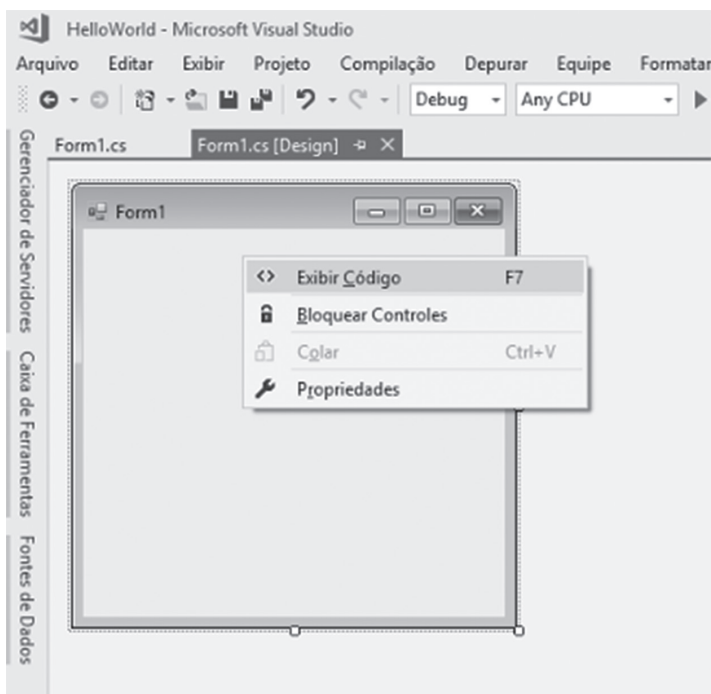


Figura 19. Opções exibidas ao clicar com o botão direito do mouse sobre um elemento do Windows Forms.

Clique em **Exibir Código**. Você será direcionado para a tela de edição de códigos do Visual Studio. Nessa tela, você poderá alterar o código, além de inserir comportamentos e funcionalidades no seu sistema (Figura 20).



Figura 20. Tela de edição de código do Visual Studio.

Agora, retorne para o Form e clique novamente com o botão direito. Nas opções, selecione **Propriedades** (Figura 21). Do lado direito, você irá notar a presença de uma nova barra de propriedades. Nessa tela, é possível modificar as características do elemento. Nesse caso, você está trabalhando no Form1, então tem as propriedades e os eventos desse elemento.

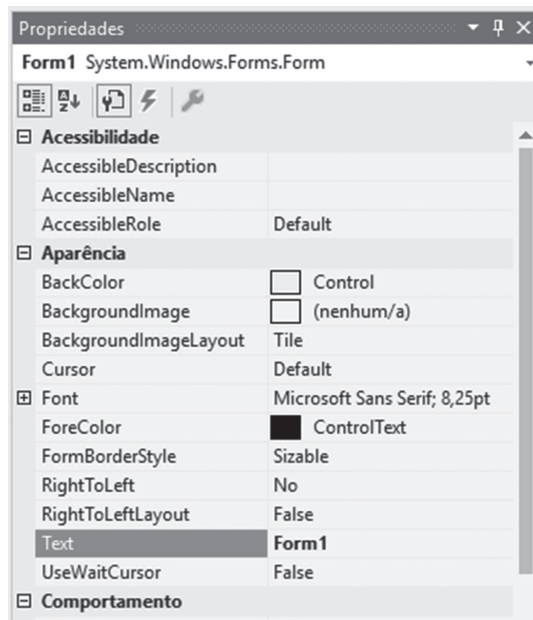
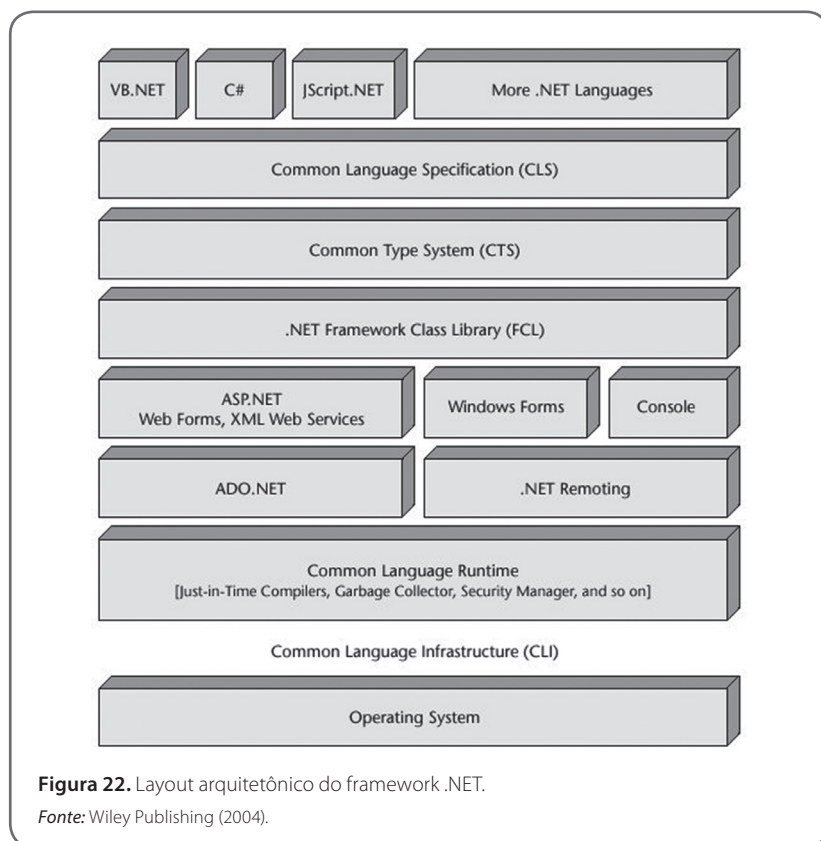


Figura 21. Propriedades de um elemento do Windows Forms.

O framework .NET

O framework .NET é multicamadas, modular e hierárquico. Cada camada do framework .NET é uma forma de abstração. As linguagens .NET são o nível superior e o nível mais abstraído. O runtime da linguagem comum (*common language*) é o nível inferior, o menos abstraído e o mais próximo do ambiente nativo. Isso é importante, uma vez que o runtime da linguagem comum funciona em estreita colaboração com o ambiente operacional para o gerenciador de aplicativos .NET. O framework .NET é dividido em módulos, cada um com sua própria responsabilidade. Logo, os níveis mais altos solicitam serviços para os níveis mais baixos, já que o .NET é hierárquico (MICROSOFT, 2017). Você pode ver o layout arquitetônico do framework .NET na Figura 22.



Camada de componente de tempo de execução

A camada de componente de execução (RCL – *Runtime Component Layer*) inclui três componentes: o .NET Framework Common Language Runtime (CLR), uma camada de abstração de hardware (HAL – *Hardware Abstraction Layer*) e uma camada de abstração de plataforma (PAL – *Platform Abstraction Layer*). A seguir, você vai conhecer cada uma delas.

CLR

O .NET Micro Framework CLR é um subconjunto do .NET Framework CLR, que é o ambiente de execução fornecido pelo .NET Framework. O .NET Micro Framework CLR fornece suporte robusto a aplicativos. Ele administra memória, execução de threads, execução de código e outros serviços de sistema. O CLR é capaz de fornecer todos esses recursos e serviços a partir de um conjunto de memória muito pequeno (MICROSOFT, 2017).

Você pode programar e depurar o CLR usando o Visual Studio .NET. Atualmente, o CLR suporta apenas C# como sua linguagem de programação.

Os recursos do .NET Micro Framework CLR incluem:

- Tipos numéricos, tipos de classe, tipos de valor, matrizes (apenas uma dimensão), delegates, eventos e referências;
- Sincronização, threads e temporizadores;
- Reflexão;
- Serialização;
- Coleta de lixo;
- Manipulação de exceção;
- Classes de tempo, incluindo DateTime e TimeSpan (usando a aritmética INT64);
- Gerenciamento de thread.

Exceções e extensões do CLR incluem:

- Restrições de execução que limitam a duração das chamadas e evitam falhas.

HAL e PAL

O HAL e o PAL controlam o hardware do sistema subjacente. Tanto o HAL como o PAL são grupos de funções C++ chamadas pelo CLR. As funções

PAL são independentes do hardware e não precisam ser portadas. No entanto, você deve escrever sua própria versão do HAL para sua plataforma de hardware.

O código de bootstrap é um código adicional que está associado ao HAL. O código do bootstrap inicializa o hardware de baixo nível quando o dispositivo está ligado. Em seguida, inicia o CLR, que executa as inicializações de nível superior. O código bootstrap executa suas tarefas por meio de chamadas para as rotinas HAL e linguagem assembly. Além de iniciar o CLR, não tem interação com o código que o precede na arquitetura do software (MICROSOFT, 2017).

Camada de biblioteca de classes

A biblioteca de classes incluída no .NET Micro Framework é uma coleção orientada a objetos de tipos reutilizáveis que você pode usar para desenvolver aplicativos incorporados. Ela inclui bibliotecas C# que fornecem suporte para o seguinte:

- Criptografia;
- Debug, gráficos e DLL de shell;
- Biblioteca de classes API CLR e CLOREL CLR.



Saiba mais

Camada de aplicação

A camada superior do .NET Micro Framework contém aplicativos gerenciados que você pode criar para seus dispositivos. Os tipos de aplicativos que você desenvolve dependem inteiramente do hardware em seu dispositivo (MICROSOFT, 2017).



Referências

AMANN, S. et al. A study of visual studio usage in practice. In: INTERNATIONAL CONFERENCE ON SOFTWARE ANALYSIS, EVOLUTION, AND REENGINEERING, 23., 2016, Suita. *Anais...* Suita: SANER, 2016. p. 124-134.

HILLAR, G. *Professional parallel programming with C#: master parallel extensions with .NET 4*. Birmingham: Wrox, 2010.

MICROSOFT. *Visual studio reference*. Redmond, 2017. Disponível em: <<https://msdn.microsoft.com/en-us/library/scesz732.aspx>>. Acesso em: 02 nov. 2017.

WILEY PUBLISHING. *The .NET Architecture*. [S.l.]: CodeGuru, 2004. Disponível em: <https://www.codeguru.com/csharp/sample_chapter/article.php/c8245/The-NET-Architecture.htm>. Acesso em: 14 nov. 2017.

Leitura recomendada

SHARP, J. *Microsoft Visual C# 2013: Passo a passo*. Porto Alegre: Bookman, 2014.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS