PROVAS DE TI
TUDO PARA VOCÊ PASSAR

.Net

Prof. Rodrigo Macedo

Escopo do Curso

- Conceitos Iniciais.
- Características.
- Arquitetura.
- Compilação.
- Utilitários.
- Questões de Concursos

Conceitos

- O .NET Framework é uma iniciativa da empresa Microsoft, que visa uma plataforma única para desenvolvimento e execução de sistemas e aplicações.
- Todo e qualquer código gerado para .NET pode ser executado em qualquer dispositivo que possua um framework de tal plataforma.
- Com ideia semelhante à plataforma Java, o programador deixa de escrever código para um sistema ou dispositivo específico, e passa a escrever para a plataforma .NET.
- Aplicações escritas para ele funcionam em um ambiente de software controlado, em oposição a um ambiente de hardware, através de uma máquina virtual de aplicação.

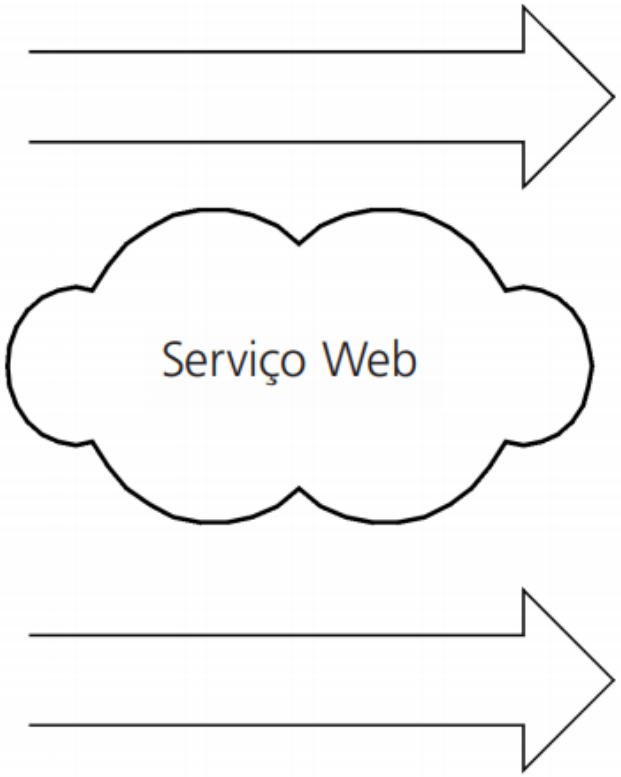
Conceitos

Essa é uma das formas pelas quais os diversos aplicativos da Microsoft possuem uma alta interoperabilidade entre si.

Laptop

IBM PC ou compatível

Estação de trabalho



Servidor

Servidor

Servidor

Conceitos

- Permite desenvolver soluções/aplicativos como:

Aplicativos Web

Aplicativos Windows

Aplicativos para Servidores

Aplicativos Smart Client

Aplicativos de Console

Aplicativos de Banco de Dados

Serviços Windows (aplicativos que rodam como serviços)

Web Services e muito mais

- Toda e qualquer aplicação gerada em .NET, pode ser executada em qualquer dispositivo ou plataforma que possua o .NET Framework.
- O .NET Framework implementa uma máquina virtual.
- Máquina virtual: Abstrair a necessidade do desenvolvedor de interagir com o sistema operacional oferecendo um rico conjunto de ferramentas e bibliotecas de objetos que permitem alta produtividade no desenvolvimento de sistemas.

Versões

2002	2003	2005	2006	2008	2010
Lançamento do Framework *1.0*	Lançamento do Framework *1.1*	Lançamento do Framework *2.0*	Lançamento do Framework *3.0*	Lançamento do Framework *3.5*	Lançamento do Framework *4.0*
Visual Studio .NET 2002	**Visual Studio .NET 2003**	**Visual Studio .NET 2005**	**Visual Studio .NET 2005**	**Visual Studio .NET 2008**	**Visual Studio .NET 2010**
WebMatrix	WebMatrix	Opções gratuitas de ferramenta de desenvolvimento da própria Microsoft (http://www.microsoft.com/express/default.aspx).			
Grande Evolução no desenvolvimento de tradicional de software. Esse primeiro release foi pouco conhecido e utilizado.	Pequenas melhorias	Consideráveis melhorias de acesso a dados (ADO.NET 2.0) Team Foundation Server (TFS: aplicação de metodologias de gerência de projeto ao desenvolvimento apoiado pelo Visual Studio.NET.)	Novos recursos para interface Windows (WPF) Novo sistema de comunicação (WCF) para aplicações distribuídas **MS-ASP.NET AJAX (biblioteca separada do Framework)** XNA Game Studio	**Linq Project** (mapeamento objeto-relacional) **Incorporação MS-ASP.NET AJAX ao Framework** **SP1: Integração total com AJAX Control ToolKit**	A IDE foi reconstruída em WPF.

A versão .NET 4.0 melhorou algumas coisas, como: Background Garbage Collector, suporte para aplicações Multitouch, A utilização do padrão MVC, A utilização de Routing no ASP.NET para Web Forms, etc.

Características

Características do .NET framework:

- **Independência de linguagem de programação:** o que permite a implementação do mecanismo de herança, controle de exceções e depuração entre linguagens de programação diferentes.
- **Reutilização de código legado:** o que implica em reaproveitamento de código escrito usando outras tecnologias como COM, COM+, ATL, DLLs e outras bibliotecas existentes.
- **Tempo de execução compartilhado:** o “runtime” de .NET é compartilhado entre as diversas linguagens que a suportam, o que quer dizer que não existe um runtime diferente para cada linguagem que implementa .NET.
- Disponibiliza um hall de dispositivos que podem ser utilizados juntos em um mesmo ambiente de desenvolvimento Web.

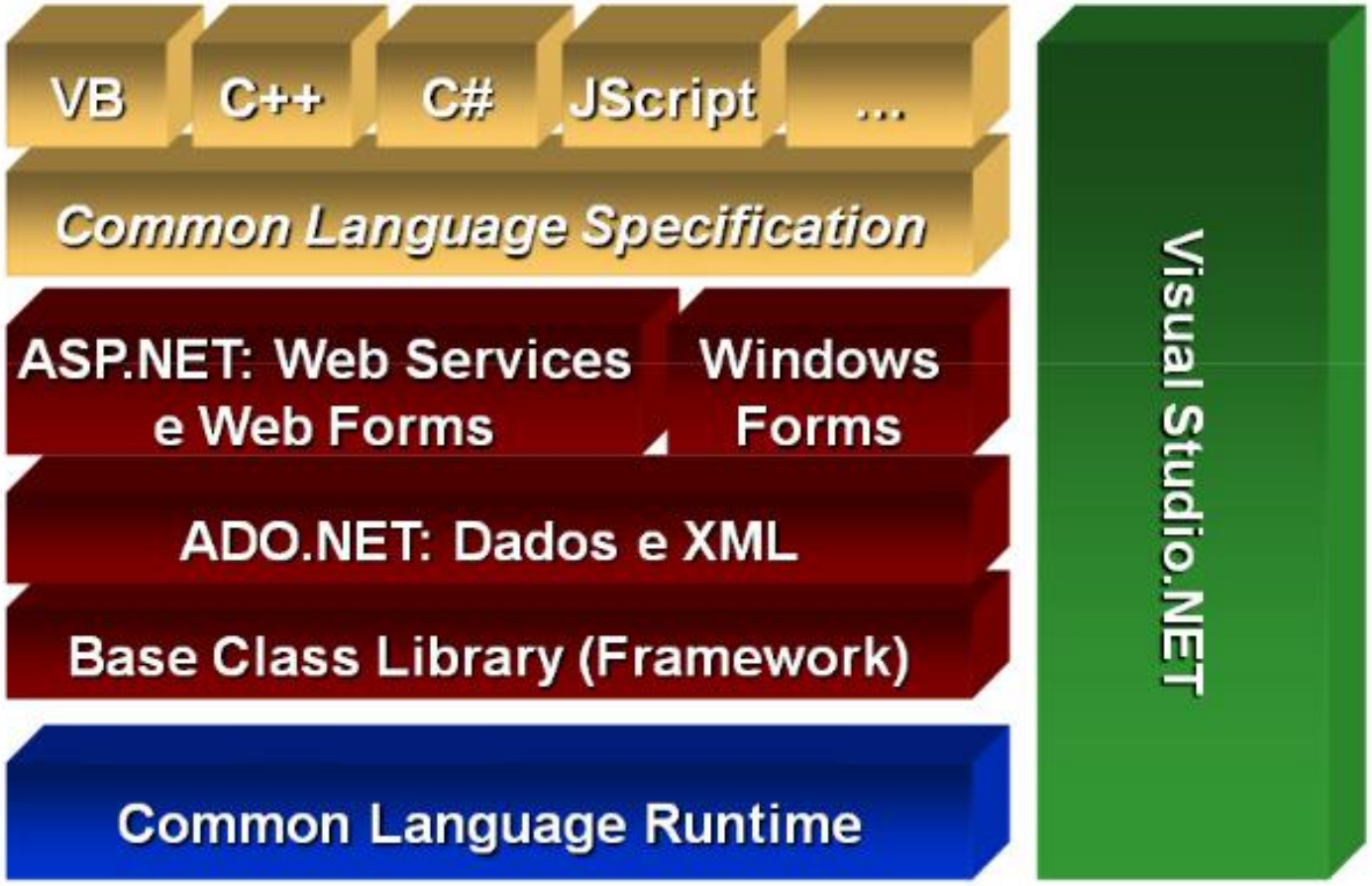
Características

Características do .NET framework:

- **Sistemas auto-explicativos e controle de versões:** cada peça de código .NET contém em si mesma a informação necessária e suficiente de formaque o runtime não precise procurar no registro do Windows mais informações sobre o programa que está sendo executado. O runtime encontraessas informações no próprio sistema em questão e sabe qual a versão a ser executada, sem acusar aqueles velhos conflitos de incompatibilidade ao registrar DLLs no Windows.
- Simplicidade na resolução de problemas complexos.
- Disponibiliza um hall de dispositivos que podem ser utilizados juntos em um mesmo ambiente de desenvolvimento Web.
- O .NET é de código aberto, usando licenças MIT e Apache 2. O .NET é um projeto do .NET Foundation.

Arquitetura

- Para compreensão do framework .NET é necessário passar pela arquitetura de .NET e os seus principais componentes.



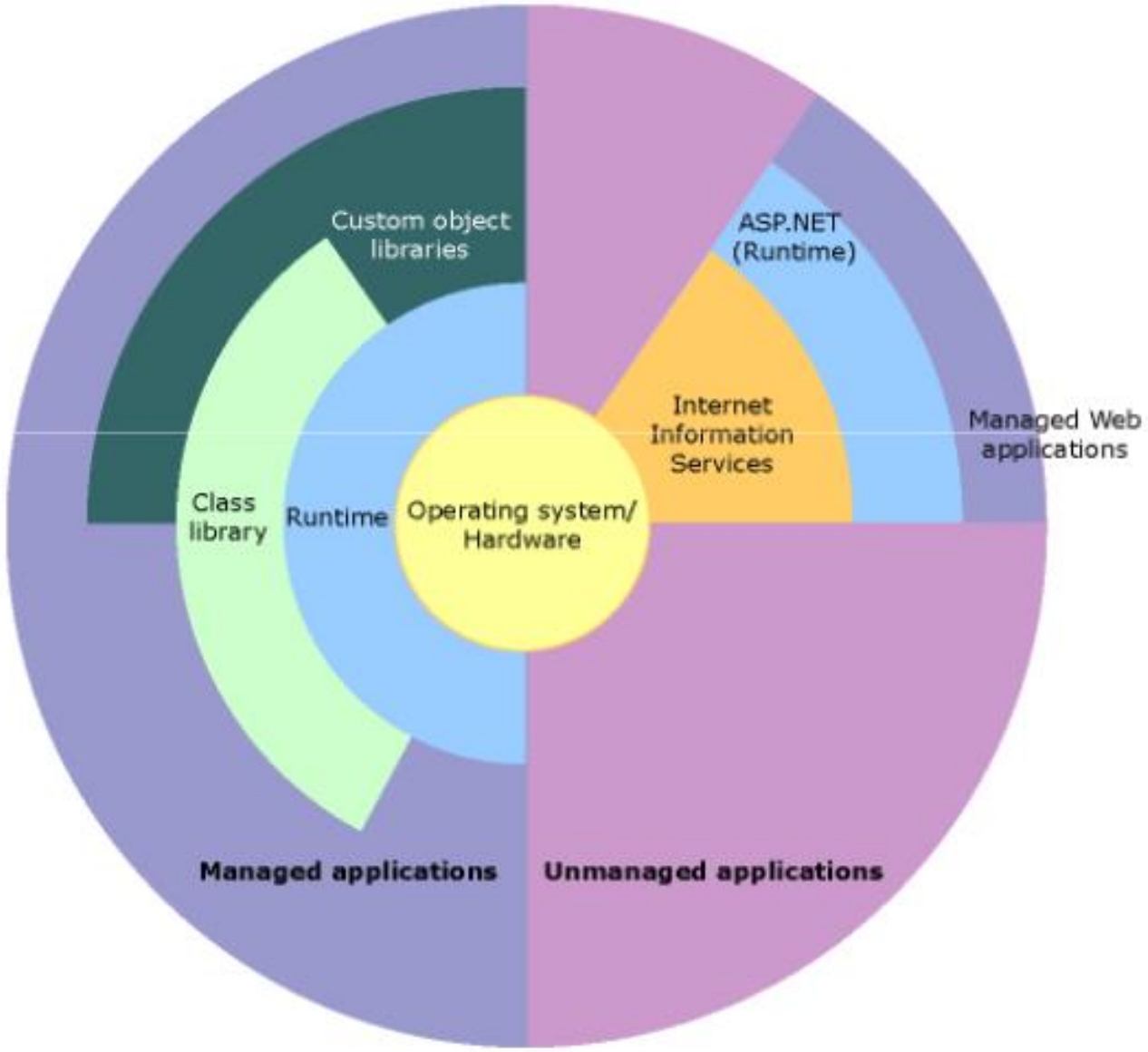
Arquitetura - CLR

- O Common Language Runtime (CLR) é o ambiente de execução das aplicações .NET.
- Realiza tarefas tais como: execução do programa, gerenciamento de memória (coleta de lixo), segurança, tratamento de erro, controle de versão e suporte de instalação.
- Realiza a interface entre a aplicação e o sistema operacional.
- O código que é executado nesse ambiente de runtime é chamado de Código Gerenciado (“Managed Code”), enquanto aquele que é executado fora é chamado de Código Não Gerenciado (“UnmanagedCode”).

Arquitetura - CLR

- O runtime também impõe robustez ao código implementando uma infraestrutura rígida de verificação de tipo e código chamada CTS (Common Type System).
- Além disso, o ambiente gerenciado do runtime elimina muitos problemas de software comuns. Por exemplo, o runtime identifica automaticamente o layout do objeto e gerencia referências a eles, liberando-os quando não estão mais sendo usados.
- Embora o Common Language Runtime forneça vários serviços de tempo de execução padrão, o código gerenciado jamais é interpretado. Um recurso chamado compilação JIT (Just-In-Time) permite que todos os códigos gerenciados sejam executados na linguagem de computador nativa do sistema, na qual está em execução.
- Por fim, o runtime pode ser hospedado por aplicativos do servidor de alto desempenho, como o Microsoft SQL Server e o IIS (Serviços de Informações da Internet).

Arquitetura - CLR



Arquitetura - CTS

- O CTS, ou Sistema Comum de Tipos, que também faz parte do CLR, define os tipos suportados por .NET e as suas características.
- Cada linguagem que suporta.NET tem de, necessariamente, suportar esses tipos.
- Apesar de que a especificação não demanda que todos os tipos definidos no CTS sejam suportados pela linguagem, esses tipos podem ser um subconjunto do CTS, ou ainda um superconjunto.

Premissas:

1. Estabelecer uma estrutura para a execução em qualquer idioma.

Arquitetura - CTS

Premissas:

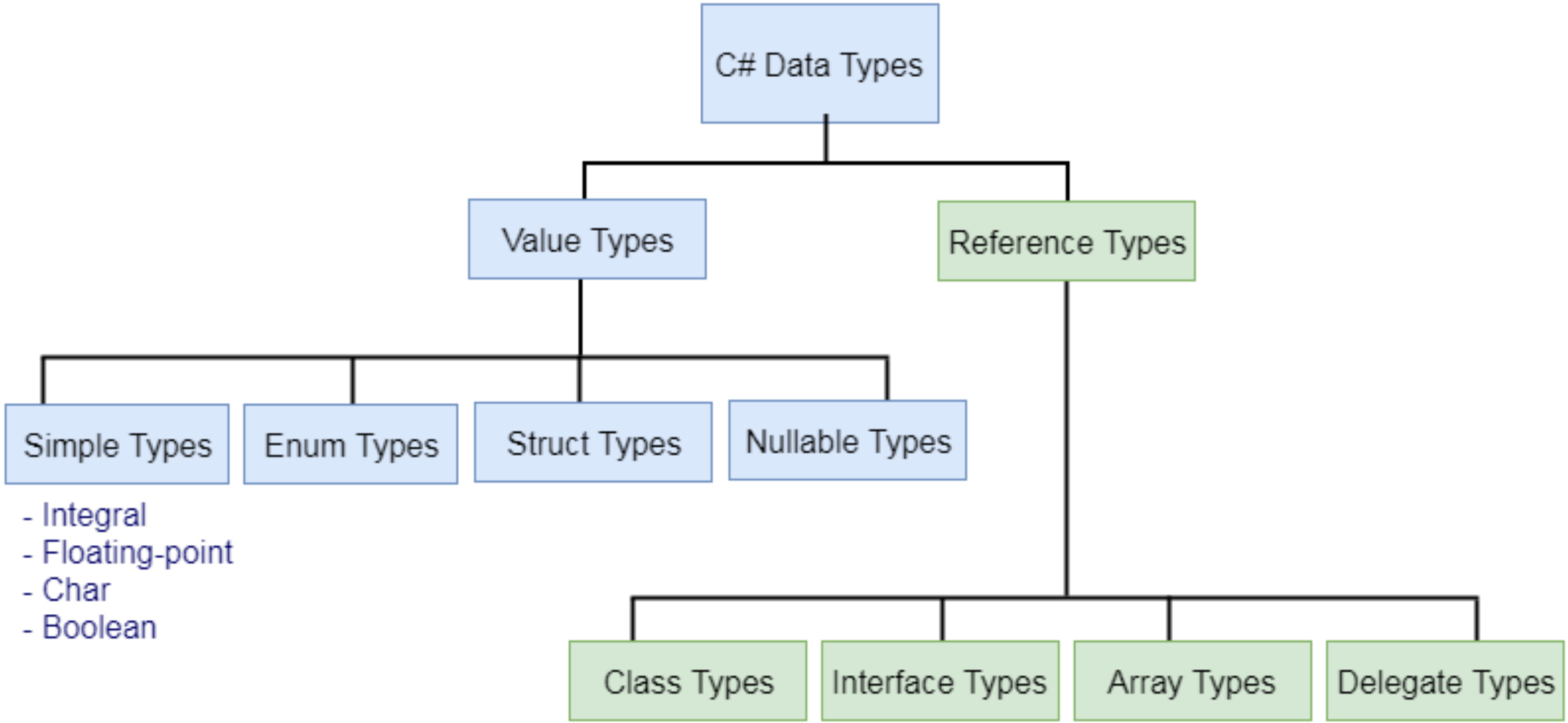
1. Fornecer um modelo orientado a objetos para dar suporte à implementação de várias linguagens em uma implementação do .NET.
2. Definir um conjunto de regras que todas as linguagens devem seguir quando se trata de trabalhar com tipos.
3. Fornecer uma biblioteca que contém os tipos primitivos básicos que são usados no desenvolvimento de aplicativos (por exemplo, Boolean, Byte, Char etc.)
4. O CTS também define todas as outras propriedades de tipos, como modificadores de acesso, o que são membros de tipo válidos, como a herança e a sobrecarga funcionam e assim por diante.

Arquitetura - CTS

- O CTS define dois tipos principais que devem ter suporte: tipos de referência e valor. Seus nomes apontam para suas definições.

1. Os objetos de **tipos de referência** são representados por uma referência ao valor real do objeto; uma referência aqui é semelhante a um ponteiro em C/C++. Ele simplesmente se refere a um local de memória onde os valores dos objetos são. Isso tem um profundo impacto sobre como esses tipos são usados. Se você atribuir um tipo de referência a uma variável e, em seguida, passar essa variável em um método, por exemplo, qualquer alteração feita no objeto será refletida no objeto principal. Não há nenhuma cópia.
2. **Tipos de valor** são o oposto, no qual os objetos são representados por seus valores. Se você atribuir um tipo de valor a uma variável, você estará, essencialmente, copiando um valor do objeto.

Arquitetura - CTS



Arquitetura - CLS

- O CLS, ou Especificação Comum da Linguagem, é um subconjunto do CTS, e define um conjunto de regras que qualquer linguagem que implemente a .NET deve seguir a fim de que o código gerado resultante da compilação de qualquer peça de software escrita na referida linguagem seja perfeitamente entendido pelo runtime .NET.
- Seguir essas regras é um imperativo porque, caso contrário, um dos grandes ganhos do .NET, que é a independência da linguagem de programação e a sua interoperabilidade, fica comprometido.
- A CLS define um conjunto de recursos que são necessários para muitos aplicativos comuns. Ela também fornece uma espécie de receita para qualquer linguagem que é implementada no .NET, no que ela precisar dar suporte.

Arquitetura - CLS

- A CLS é um subconjunto do CTS. Isso significa que todas as regras no CTS também se aplicam à CLS, a menos que as regras da CLS sejam mais estritas. Se um componente é criado usando apenas as regras na CLS, ou seja, ele expõe apenas os recursos da CLS em sua API, ele é chamado de compatível com CLS.
- Por exemplo, as <framework-librares> estão em conformidade com CLS precisamente porque precisam funcionar em todas as linguagens com suporte no .NET.
- As regras que se deve obedecer, são como as definições de membros(isto é, nomenclatura, parâmetros e tipo de retorno).
- Por exemplo, no VB.NET, as letras maiúsculas não diferem das minúsculas, mas no c# elas são diferenciadas.
- Outro exemplo é no caso do c# precisar do “;” no final da linha de comando, enquanto o VB.NET não exige isto.

Arquitetura - BCL

- Uma plataforma que promete facilitar o desenvolvimento de sistemas precisa ter uma biblioteca de classes básica que alavanque a simplicidade e a rapidez no desenvolvimento de sistemas.
- Seu objetivo é oferecer ao desenvolvedor uma biblioteca consistentede componentes de software reutilizáveis que não apenas facilitem, mas também que acelerem o desenvolvimento de sistemas.
- Na BCL encontramos classes que contemplam desde um novo sistema de janelas a bibliotecas de entrada/saída, gráficos, sockets, gerenciamento da memória etc.
- Esta biblioteca de classes é organizada hierarquicamente em uma estrutura conhecida como namespace. Ao desenvolver um componente de software reusável, este precisa ser estruturado em um namespace para que possa ser usado a partir de um outro programa externo.

Arquitetura - BCL

Alguns namespaces .NET	
System	Contém algumas classes de baixo nível usadas para trabalhar com tipos primitivos, operações matemáticas, gerenciamento de memória etc.
System.Collections	Pensando em implementar suas próprias pilhas, filhas, listas encadeadas? Elas já foram implementadas e se encontram aqui.
System.Data, **System.Data.Commom,** **System.Data.OleDb,** **System.Data.SqlClient**	Aqui você vai encontrar tudo o que é necessário para lidar com bases de dados e, como é de se esperar, você encontrará ADO.NET aqui.

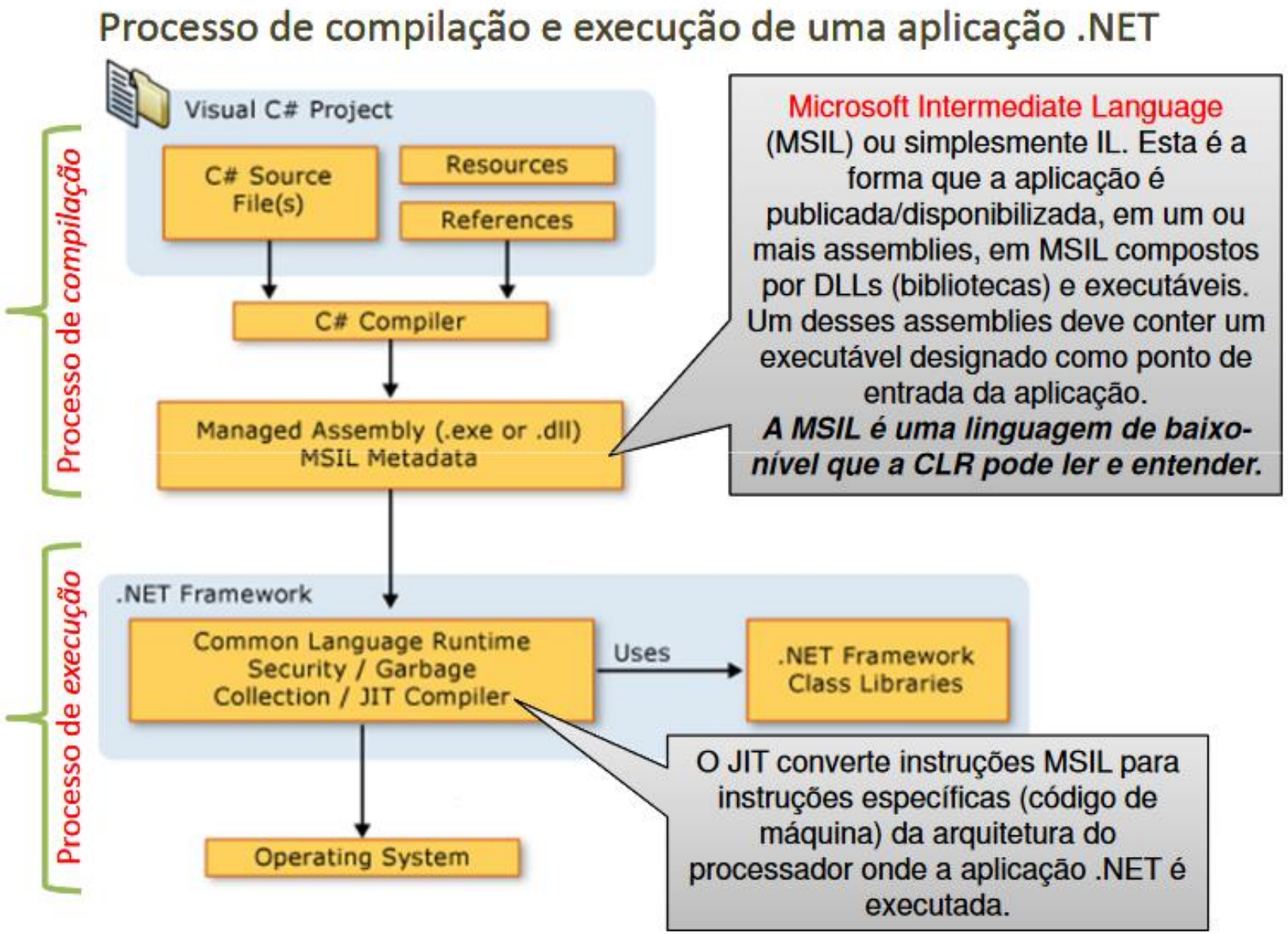
Arquitetura - BCL

System.Diagnostics	Log de Event, medição de performance, classes para gerenciamento de processos, depuração e mais você poderá encontrar neste namespace.
System.Drawing e namespaces derivados	A .NET oferece uma biblioteca de componentes para trabalhar com gráficos, chamada GDI+, que se encontra neste namespace.
System.IO	Biblioteca para lidar com entrada e saída, gerenciamento de arquivos etc.
System.NET	Aqui você encontra bibliotecas para programação de redes, sockets etc.
System.Reflection	Em .NET você pode gerar código em tempo de execução, descobrir tipos de variáveis etc. As bibliotecas necessárias para isso encontram-se neste namespace.

Compilação

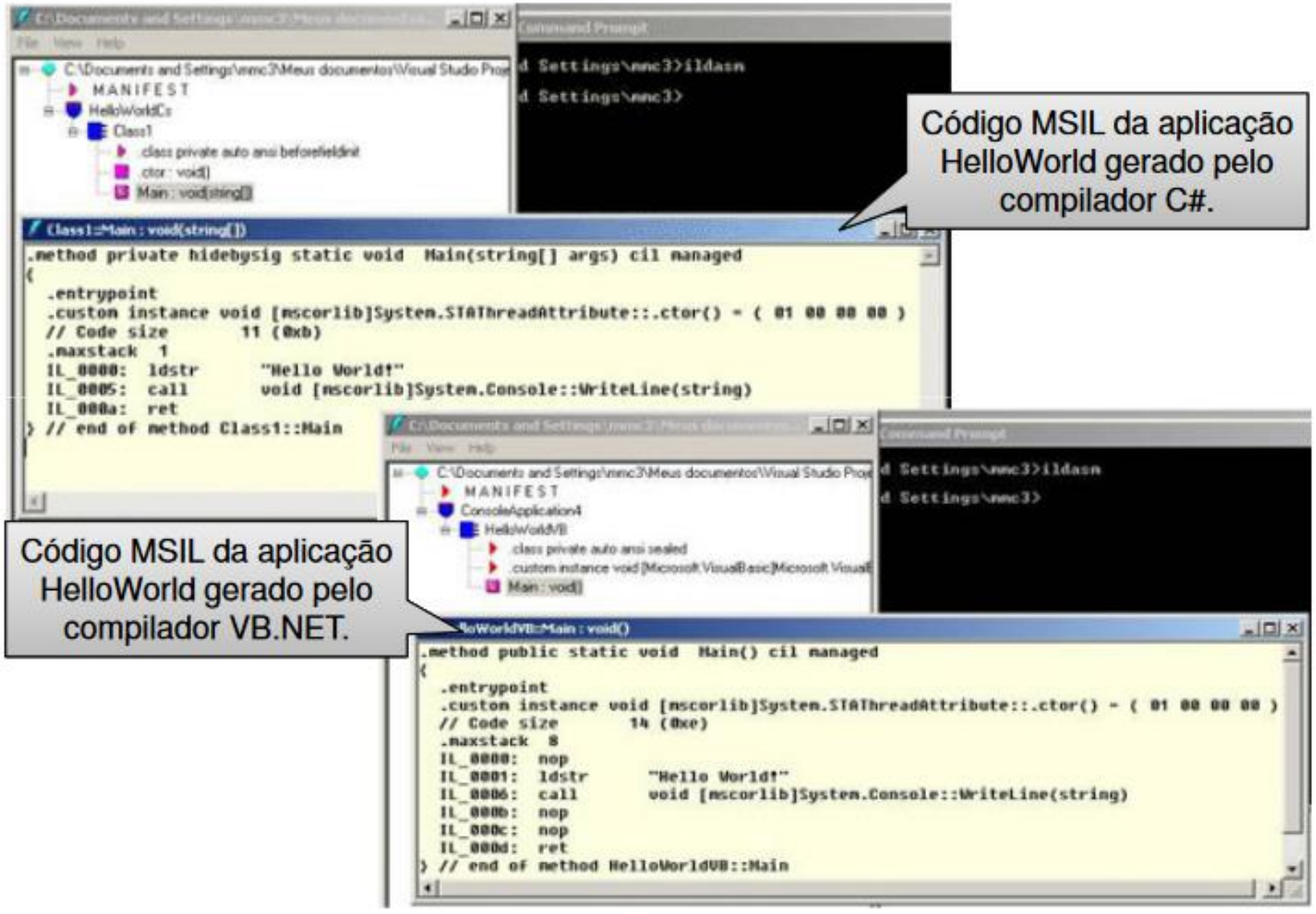
- A MSIL (Microsoft Intermediate Language) – ou simplesmente IL– é a linguagem intermediária para qual é interpre_x0002_tado qualquer programa .NET, independente da linguagem em que este for escrito.
- Essa tradução é feita para código intermediário (como em JAVA com os byte codes) sintaticamente expresso na IL.
- Por sua vez, qualquer linguagem .NET compatível, na hora da compilação, gerará código IL e não código assembly específico da arquitetura do processador onde a compilação do programa é efetua-da, conforme aconteceria em C++ ou Delphi, por exemplo. E por que isso? Isso acontece para garantir duas coisas: a independência da linguagem e a independência da plataforma (arquitetura do processador).
- A MSIL é a linguagem intermediária para qual é interpretado qualquer programa .NET na hora da compilação, independente da linguagem em que este for escrito.

Compilação

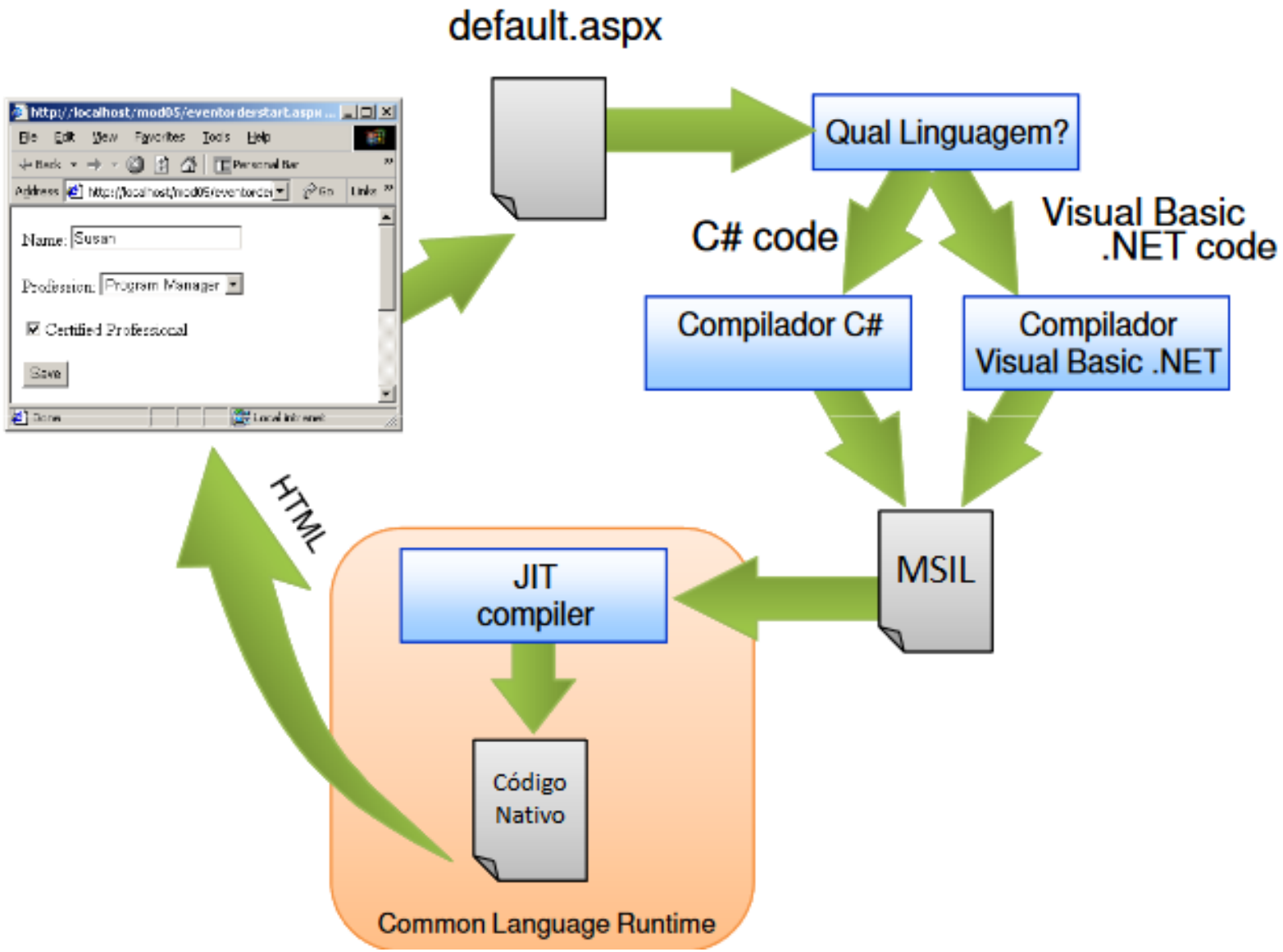

Processo de compilação e execução de uma aplicação .NET
Visual C# Project
C# Source File(s)
Resources
References
C# Compiler
Managed Assembly (.exe or .dll)
MSIL Metadata
Processo de compilação
Microsoft Intermediate Language (MSIL) ou simplesmente IL. Esta é a forma que a aplicação é publicada/disponibilizada, em um ou mais assemblies, em MSIL compostos por DLLs (bibliotecas) e executáveis. Um desses assemblies deve conter um executável designado como ponto de entrada da aplicação.
A MSIL é uma linguagem de baixo-nível que a CLR pode ler e entender.
.NET Framework
Common Language Runtime Security / Garbage Collection / JIT Compiler
Uses
.NET Framework Class Libraries
Operating System
Processo de execução
O JIT converte instruções MSIL para instruções específicas (código de máquina) da arquitetura do processador onde a aplicação .NET é executada.

Compilação

Processo de compilação e execução de uma aplicação .NET



Compilação


default.aspx
Qual Linguagem?
C# code
Visual Basic
.NET code
Compilador C#
Compilador
Visual Basic .NET
MSIL
JIT
compiler
Código
Nativo
Common Language Runtime
HTML
Name: Susan
Profession: Program Manager
Certified Professional
Save

Implementações .NET

- O .NET vem em diferentes tipos, mais formalmente conhecidos como implementações. O .NET 5+ (incluindo o .NET Core) é a implementação mais recente e é executado em qualquer plataforma. .NET Framework é a implementação original do .NET e é executado somente no Windows. Mono é usado quando um runtime pequeno é necessário. A UWP (Plataforma Windows Universal) é usada para criar aplicativos Windows modernos.
- Cada implementação inclui um runtime e uma biblioteca de classes. Ele também pode incluir estruturas de aplicativos e ferramentas de desenvolvimento.
- .NET Standard é uma implementação do .NET, mas uma especificação de API que permite desenvolver bibliotecas de classes para várias implementações do .NET.

Linguagens Programação .NET

- **C#:** é uma linguagem de programação moderna, orientada a objeto e fortemente digitada. O C# tem suas raízes na família de linguagens C e os programadores em C, C++, Java e JavaScript a reconhecerão imediatamente.
- **F#:** F# é uma linguagem de programação interoperável, de plataforma cruzada e de código de código sucinta, robusta e de desempenho. Seu foco permanece no domínio do problema, em vez dos detalhes da programação. A programação em F# é orientada a dados, em que o código envolve a transformação de dados com funções.
- **Visual Basic:** Entre as linguagens .NET, a sintaxe Visual Basic é a mais próxima da linguagem humana comum, o que pode facilitar o aprendizado. Ao contrário de C# e F#, para os quais a Microsoft está desenvolvendo ativamente novos recursos, Visual Basic linguagem é estável. Visual Basic há suporte para aplicativos Web, mas há suporte para APIs Web.

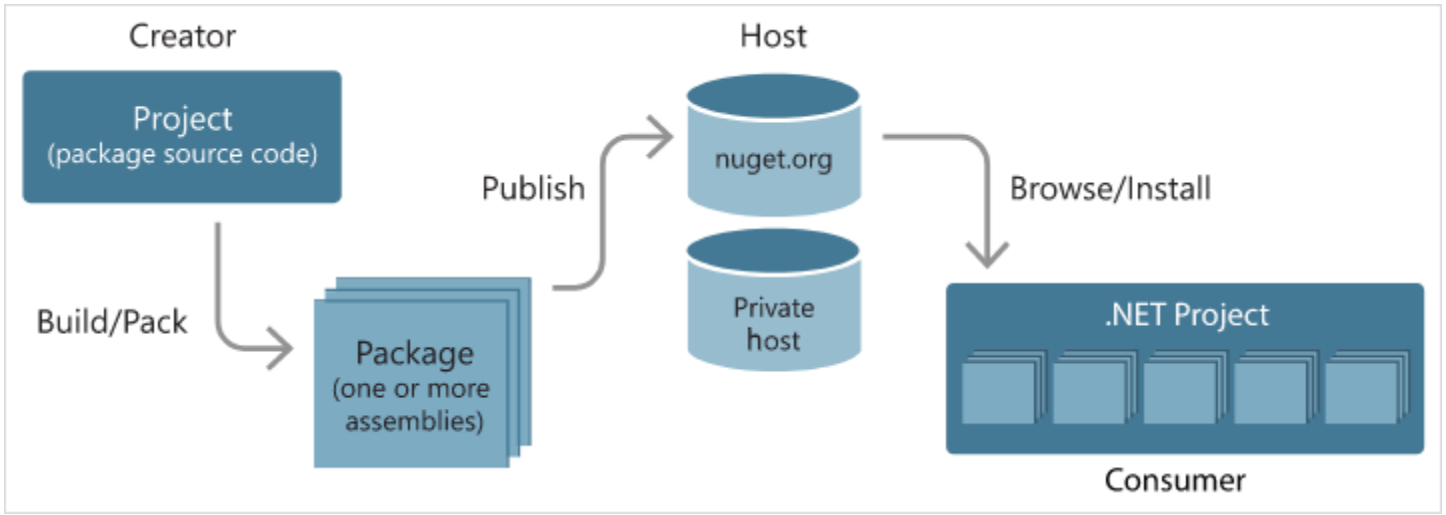
IDEs

Os ambientes de desenvolvimento integrados para .NET incluem:

- **Visual Studio:** É executado Windows somente. Tem uma funcionalidade interna extensiva projetada para funcionar com o .NET. A Community edição é gratuita para alunos, colaboradores de código-fonte aberto e indivíduos.
- **Visual Studio Code:** É executado Windows, macOS e Linux. Gratuito e software livre. As extensões estão disponíveis para trabalhar com linguagens .NET.
- **Visual Studio para Mac:** É executado somente no macOS. Para desenvolver aplicativos e jogos .NET para iOS, Android e Web.
- **Codespaces do Github:** Um ambiente Visual Studio Code online, atualmente em beta.

NuGet

- NuGet é um gerenciador de pacotes de código aberto projetado para .NET. Um pacote NuGet é um arquivo.zipcom a extensão que contém DLLs (código compilado), outros arquivos relacionados a esse código e um manifesto descritivo que inclui informações como o número de versão do .nupkg pacote.
- Os desenvolvedores com código para compartilhar criam pacotes e publicam-nos nuget.org ou em um host privado. Os desenvolvedores que querem usar o código compartilhado adicionam um pacote ao projeto e, em seguida, podem chamar a API exposta pelo pacote em seu código de projeto.



Q1) [CESPE SEED-PR 2021] A iniciativa .NET, da Microsoft, apresenta um conjunto totalmente novo de conceitos dessa plataforma. Esse framework possui diversos elementos que a compõem: um deles é o conjunto de tipos de dados comuns, na forma de objetos, os quais podem ser utilizados por todas as linguagens habilitadas ao .NET, como o C#. O elemento do framework .NET citado no texto é

a) o metadata.

b) o web services.

c) a class library.

d) o common type system.

e) o common language runtime.

Q1) [CESPE SEED-PR 2021] A iniciativa .NET, da Microsoft, apresenta um conjunto totalmente novo de conceitos dessa plataforma. Esse framework possui diversos elementos que a compõem: um deles é o conjunto de tipos de dados comuns, na forma de objetos, os quais podem ser utilizados por todas as linguagens habilitadas ao .NET, como o C#. O elemento do framework .NET citado no texto é

a) o metadata.

b) o web services.

c) a class library.

d) o common type system.

e) o common language runtime.

Q2) [FCC TJ-SC 2021] ADO.NET é um conjunto de classes do .NET Framework desenvolvidas para facilitar o acesso das aplicações às bases de dados. Um desenvolvedor que deseja utilizar classes da ADO.NET que representam tabelas, colunas, linhas e que deseja utilizar a classe DataSet deve fazer por meio do namespace

a) System.Command.

b) System.Data.

c) System.Odbc.

d) System.Sql.

e) System.DataBase.

Q2) [FCC TJ-SC 2021] ADO.NET é um conjunto de classes do .NET Framework desenvolvidas para facilitar o acesso das aplicações às bases de dados. Um desenvolvedor que deseja utilizar classes da ADO.NET que representam tabelas, colunas, linhas e que deseja utilizar a classe DataSet deve fazer por meio do namespace

a) System.Command.

b) System.Data.

c) System.Odbc.

d) System.Sql.

e) System.DataBase.

Q3) [VUNESP 2019 Câmara Piracicaba] Na plataforma .NET, o coletor de lixo (garbage collector) é um componente

a) da Biblioteca de Classes do Framework (FCL – Framework Class Library).

b) da Especificação Comum da Linguagem (CLS – Common Language Specification).

c) da Linguagem Intermediária da Microsoft (MSIL – Microsoft Intermediate Language).

d) do Sistema Comum de Tipos (CTS – Common Type System).

e) do Tempo de Execução de Linguagem Comum (CLR – Common Language Runtime).

Q3) [VUNESP 2019 Câmara Piracicaba] Na plataforma .NET, o coletor de lixo (garbage collector) é um componente

a) da Biblioteca de Classes do Framework (FCL – Framework Class Library).

b) da Especificação Comum da Linguagem (CLS – Common Language Specification).

c) da Linguagem Intermediária da Microsoft (MSIL – Microsoft Intermediate Language).

d) do Sistema Comum de Tipos (CTS – Common Type System).

e) do Tempo de Execução de Linguagem Comum (CLR – Common Language Runtime).

Q4) [INSTITUTO AOCP PRODEB 2018] Ferramentas para o Gerenciamento de Dependências já estão disponíveis para as principais linguagens de desenvolvimento utilizadas no mercado. Com base nisso, dentre as linguagens escolhidas, relacione as colunas com a linguagem de programação e a sua respectiva ferramenta para o gerenciamento de dependências e assinale a alternativa com a sequência correta.

1. PHP

2. JAVA

3. .NET

4. RUBY

5. JAVASCRIPT ( ) YARN ( ) RUBYGEMS ( ) NUGET ( ) COMPOSER ( ) MAVEN

a) 1 – 2 – 5 – 4 – 3.

b) 3 – 4 – 1 – 2 – 5.

c) 1 – 3 – 4 – 2 – 5.

d) 5 – 4 – 3 – 1 – 2.

e) 5 – 2 – 3 – 4 – 1.

Q4) [INSTITUTO AOCP PRODEB 2018] Ferramentas para o Gerenciamento de Dependências já estão disponíveis para as principais linguagens de desenvolvimento utilizadas no mercado. Com base nisso, dentre as linguagens escolhidas, relacione as colunas com a linguagem de programação e a sua respectiva ferramenta para o gerenciamento de dependências e assinale a alternativa com a sequência correta.

1. PHP

2. JAVA

3. .NET

4. RUBY

5. JAVASCRIPT ( ) YARN ( ) RUBYGEMS ( ) NUGET ( ) COMPOSER ( ) MAVEN

a) 1 – 2 – 5 – 4 – 3.

b) 3 – 4 – 1 – 2 – 5.

c) 1 – 3 – 4 – 2 – 5.

d) 5 – 4 – 3 – 1 – 2.

e) 5 – 2 – 3 – 4 – 1.

Q5) [VUNESP Saae Barretos 2018] No .NET Framework, fazem parte do namespace System.Collections as seguintes classes:

a) Cookie, WebClient, Authorization e HttpListener.

b) DataSet, DataTable, DataView e DataColumn.

c) Hashtable, Queue, Stack e Comparer.

d) BinaryReader, File, Stream e StringWriter.

e) StringBuilder, Decoder, Encoding e UnicodeEncoding.

Q5) [VUNESP Saae Barretos 2018] No .NET Framework, fazem parte do namespace System.Collections as seguintes classes:

a) Cookie, WebClient, Authorization e HttpListener.

b) DataSet, DataTable, DataView e DataColumn.

c) Hashtable, Queue, Stack e Comparer.

d) BinaryReader, File, Stream e StringWriter.

e) StringBuilder, Decoder, Encoding e UnicodeEncoding.

Q6) [FGV DPE-RO 2015] Programas escritos em C# operam na presença do .NET framework, que é uma tecnologia cujos elementos fundamentais são:

a) um interpretador universal e um conjunto de WEB services;

b) um conjunto unificado de tipos de dados e um conjunto de WEB services;

c) um interpretador universal e um conjunto unificado de tipos de dados;

d) um ambiente comum de runtime e um ambiente comum de desenvolvimento;

e) um ambiente comum de runtime e uma biblioteca de classes.

Q6) [FGV DPE-RO 2015] Programas escritos em C# operam na presença do .NET framework, que é uma tecnologia cujos elementos fundamentais são:

a) um interpretador universal e um conjunto de WEB services;

b) um conjunto unificado de tipos de dados e um conjunto de WEB services;

c) um interpretador universal e um conjunto unificado de tipos de dados;

d) um ambiente comum de runtime e um ambiente comum de desenvolvimento;

e) um ambiente comum de runtime e uma biblioteca de classes.

Q7) [VUNESP TCE SP 2015] Na plataforma .NET, o componente responsável pela execução do código é chamado de

a) Common Execution Architecture – CEA

b) Common Intermediate Language – CIL.

c) Common Language Runtime – CLR

d) Common Type System – CTS.

e) Common Virtual Machine – CVM.

Q7) [VUNESP TCE SP 2015] Na plataforma .NET, o componente responsável pela execução do código é chamado de

a) Common Execution Architecture – CEA

b) Common Intermediate Language – CIL.

c) Common Language Runtime – CLR

d) Common Type System – CTS.

e) Common Virtual Machine – CVM.

Q8) [CETAP MPC-PA 2015] A plataforma .Net possui um conceito semelhante ao conceito de pacote em Java. Este conceito se conhece como:

a) Package.

b) Plattform.

c) System.Net.

d) Namespace.

e) Assembly.

Q8) [CETAP MPC-PA 2015] A plataforma .Net possui um conceito semelhante ao conceito de pacote em Java. Este conceito se conhece como:

a) Package.

b) Plattform.

c) System.Net.

d) Namespace.

e) Assembly.

Q9) [FCC DPE-SP 2013] O .NET Framework é um ambiente de execução gerenciado que consiste de dois componentes principais: o Common Language Runtime (CLR) e a .NET Framework Class Library. Sobre o .NET Framework, analise:

I. Em muitas linguagens de programação, os programadores são responsáveis por alocar e liberar memória e por manipular o tempo de vida do objeto. Em aplicativos do .NET Framework, o CLR fornece esses serviços.

II. Em muitas linguagens de programação tradicionais, os tipos básicos são definidos pelo compilador, o que complica a interoperabilidade entre linguagens. No .NET Framework, os tipos básicos são definidos pelo .NET Framework Type System e são comuns a todas as linguagens que o utilizam.

III. O .NET Framework inclui bibliotecas para áreas específicas de desenvolvimento de aplicativos, como o ASP.NET para aplicativos da web, o ADO.NET para acesso a dados e o Windows Communication Foundation para aplicativos orientados a serviços.

IV. Compiladores de linguagens direcionadas ao .NET Framework geram um código intermediário chamado de Common Intermediate Language (CIL), que, por sua vez, é compilado em tempo de execução pelo CLR. Com esse recurso, as ro- tinas escritas em uma linguagem tornam-se acessíveis a outras linguagens da plataforma .NET.

Está correto o que se afirma em

a) I, II, III e IV.

b) III e IV, apenas.

c) I e II, apenas.

d) II e IV, apenas.

e) III, apenas.

Q9) [FCC DPE-SP 2013] O .NET Framework é um ambiente de execução gerenciado que consiste de dois componentes principais: o Common Language Runtime (CLR) e a .NET Framework Class Library. Sobre o .NET Framework, analise:

I. Em muitas linguagens de programação, os programadores são responsáveis por alocar e liberar memória e por manipular o tempo de vida do objeto. Em aplicativos do .NET Framework, o CLR fornece esses serviços.

II. Em muitas linguagens de programação tradicionais, os tipos básicos são definidos pelo compilador, o que complica a interoperabilidade entre linguagens. No .NET Framework, os tipos básicos são definidos pelo .NET Framework Type System e são comuns a todas as linguagens que o utilizam.

III. O .NET Framework inclui bibliotecas para áreas específicas de desenvolvimento de aplicativos, como o ASP.NET para aplicativos da web, o ADO.NET para acesso a dados e o Windows Communication Foundation para aplicativos orientados a serviços.

IV. Compiladores de linguagens direcionadas ao .NET Framework geram um código intermediário chamado de Common Intermediate Language (CIL), que, por sua vez, é compilado em tempo de execução pelo CLR. Com esse recurso, as ro- tinas escritas em uma linguagem tornam-se acessíveis a outras linguagens da plataforma .NET.

Está correto o que se afirma em

a) I, II, III e IV.

b) III e IV, apenas.

c) I e II, apenas.

d) II e IV, apenas.

e) III, apenas.

Q10) [VUNESP MPE-ES 2013] Na plataforma .NET, a classe base de todas as outras classes é a

a) Base.System.Class

b) Base.Object

c) Net.System.Object

d) System.Base

e) System.Object

Q10) [VUNESP MPE-ES 2013] Na plataforma .NET, a classe base de todas as outras classes é a

a) Base.System.Class

b) Base.Object

c) Net.System.Object

d) System.Base

e) System.Object

Q11) [FCC MPE-SE 2010] NÃO se trata de uma linguagem de programação normalmente usada no desenvolvimento de aplicativos com o .NET Framework:

a) Visual C++

b) Visual C#

c) Visual J#

d) Visual Basic

e) Visual Ruby

Q11) [FCC MPE-SE 2010] NÃO se trata de uma linguagem de programação normalmente usada no desenvolvimento de aplicativos com o .NET Framework:

a) Visual C++

b) Visual C#

c) Visual J#

d) Visual Basic

e) Visual Ruby

Q12) [CETRO Fundação Casa 2014] Quanto ao .NET, assinale a alternativa correta.

a) Não utiliza o registro do Windows para localizar fisicamente os componentes.

b) Não acessa componentes no diretório local da aplicação.

c) Não possui gerenciamento automático da utilização dos componentes.

d) Não possui um gerenciador para replicação e balanceamento de carga entre diferentes servidores.

e) Não realiza liberação de memória sem intervenção do desenvolvedor.

Q12) [CETRO Fundação Casa 2014] Quanto ao .NET, assinale a alternativa correta.

a) Não utiliza o registro do Windows para localizar fisicamente os componentes.

b) Não acessa componentes no diretório local da aplicação.

c) Não possui gerenciamento automático da utilização dos componentes.

d) Não possui um gerenciador para replicação e balanceamento de carga entre diferentes servidores.

e) Não realiza liberação de memória sem intervenção do desenvolvedor.