

Linguagem de Programação Comercial

Aula 01

Prof. Cássio Huggentobler de Costa

[cassiocosta@ulbra.inf.br]

@cassiocosta_

Sumário



- Apresentação
- Plano de Ensino
- Framework dotnet
- .net Core
- Csharp(C#)
- Get started with .net Core

Contrato



- Objetivo comum: **aprendizado**
- Interação, empatia
- Apoio entre colegas
- Respeito: celular, conversa, cola, ...
- Liberdade de expressão
- Deveres: prazos, esforço, ...

Plano de Ensino (Ver EAD)

Ementa

A disciplina proporciona o desenvolvimento de sistemas de informação através do uso de uma linguagem de programação de grande utilização no mercado de trabalho.

Histórico .NET

Version	Lançamento	Visual Studio	Windows
1.0	13 de fevereiro de 2002	Visual Studio	XP, Tablets
1.1	24 de abril de 2003	Visual Studio 2003	Server 2003
2.0	07 de novembro de 2005	Visual Studio 2005	Server 2003 R2
3.0	06 de novembro de 2006	Visual Studio 2005	Vista, Server 2008
3.5	19 de novembro de 2007	Visual Studio 2008	7, Server 2008 R2
4.0	12 de abril de 2010	Visual Studio 2010	7
4.5	Lançamento oficial previsto final 2012	Visual Studio 2012	8, Server 2012
4.5.X	2015	Visual Studio 2015	7,8,10
4.6	2016	Visual Studio 2015	7,8,10
4.6.X	2016 – 2017	Visual Studio 2017/VSCode	7,8,10
1.0 Core	2017 (atualmente na versão 2.0)	Visual Studio 2017/VSCode	multiplataforma

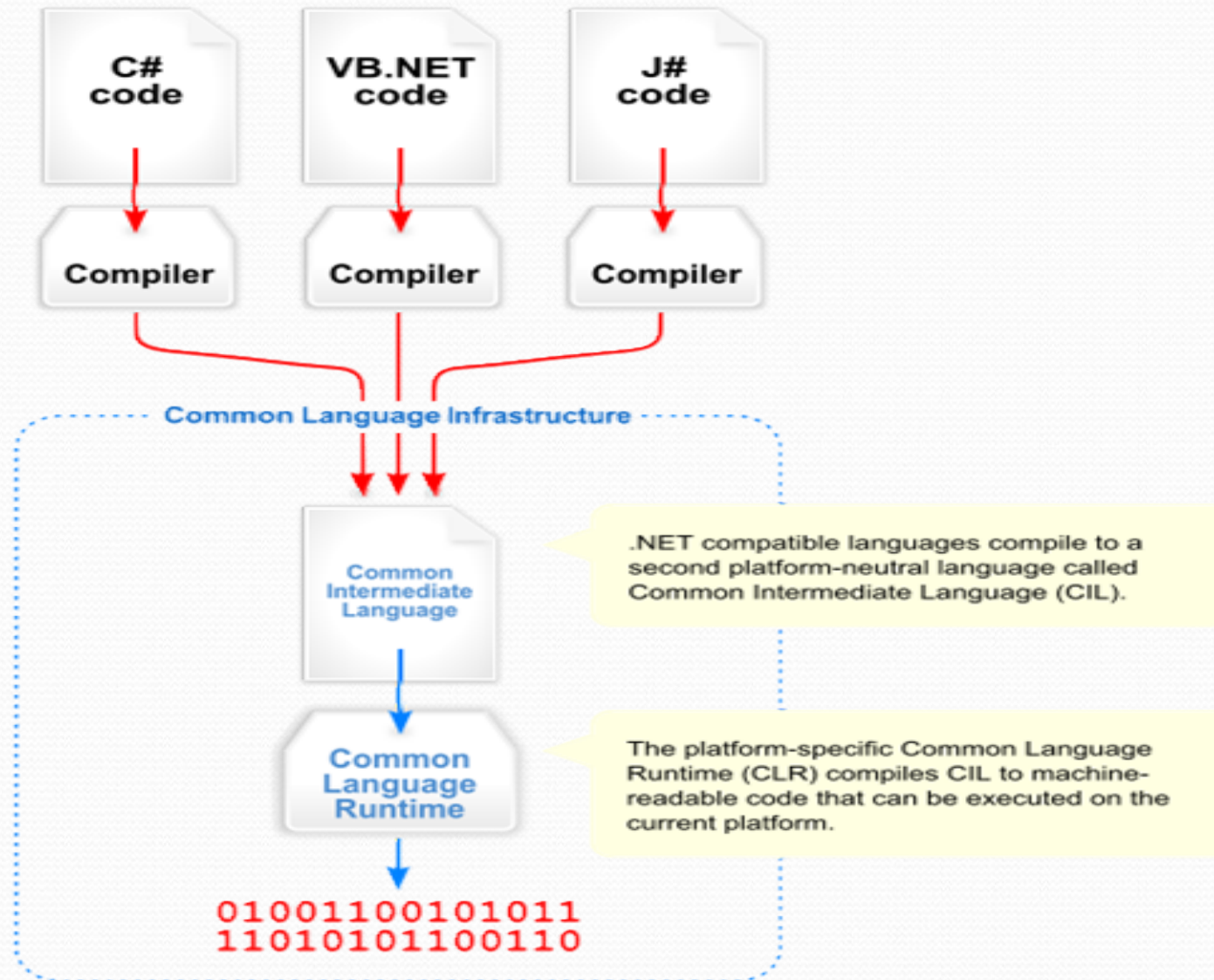
Plataforma .NET - Arquitetura

- A plataforma .Net disponibiliza uma vasta biblioteca de funcionalidades e componentes, fornecendo interoperabilidade com uma gama de linguagens de alto nível;
- O desenvolvedor não se preocupa inteiramente com o dispositivo ao qual está destinada sua aplicação e sim com a plataforma de desenvolvimento.
- A interoperabilidade de ambiente é semelhante à plataforma Java, usando uma máquina virtual independente de linguagem, chamado Common Language Runtime ou CLR. Em Java é a JVM.
- Pode ser executada em vários dispositivos atualmente, desde um smartphone, tablet, Xbox, computadores desktops até servidores dedicados ou não;

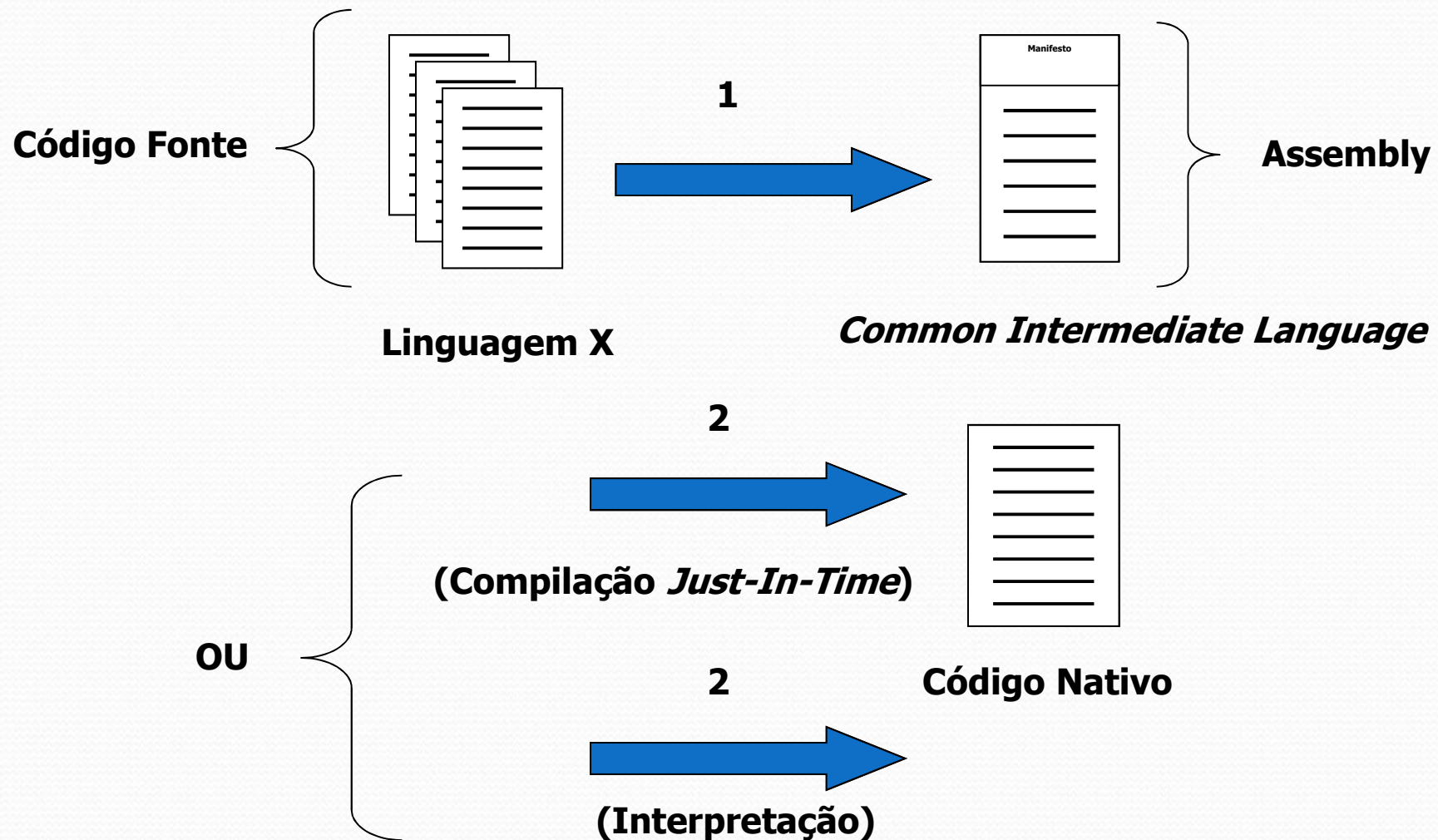
Plataforma .NET - Arquitetura

- Com isto abstrai do desenvolvedor a necessidade de usar apenas uma única linguagem, tendo hoje mais de 30 opções, sendo algumas delas
 - C#, C++, Boo, F#, COBOL, Fortran, Haskell, Lua, Object Pascal, Perl, PowerBuilder, Python, Ruby, Smalltalk, Visual Basic, entre outras.
- As aplicações são duplamente compiladas, sendo uma para uma linguagem intermediária, Common Intermediate Language (IL) ou Microsoft Intermediate Language (MSIL), e outra para linguagem de máquina.

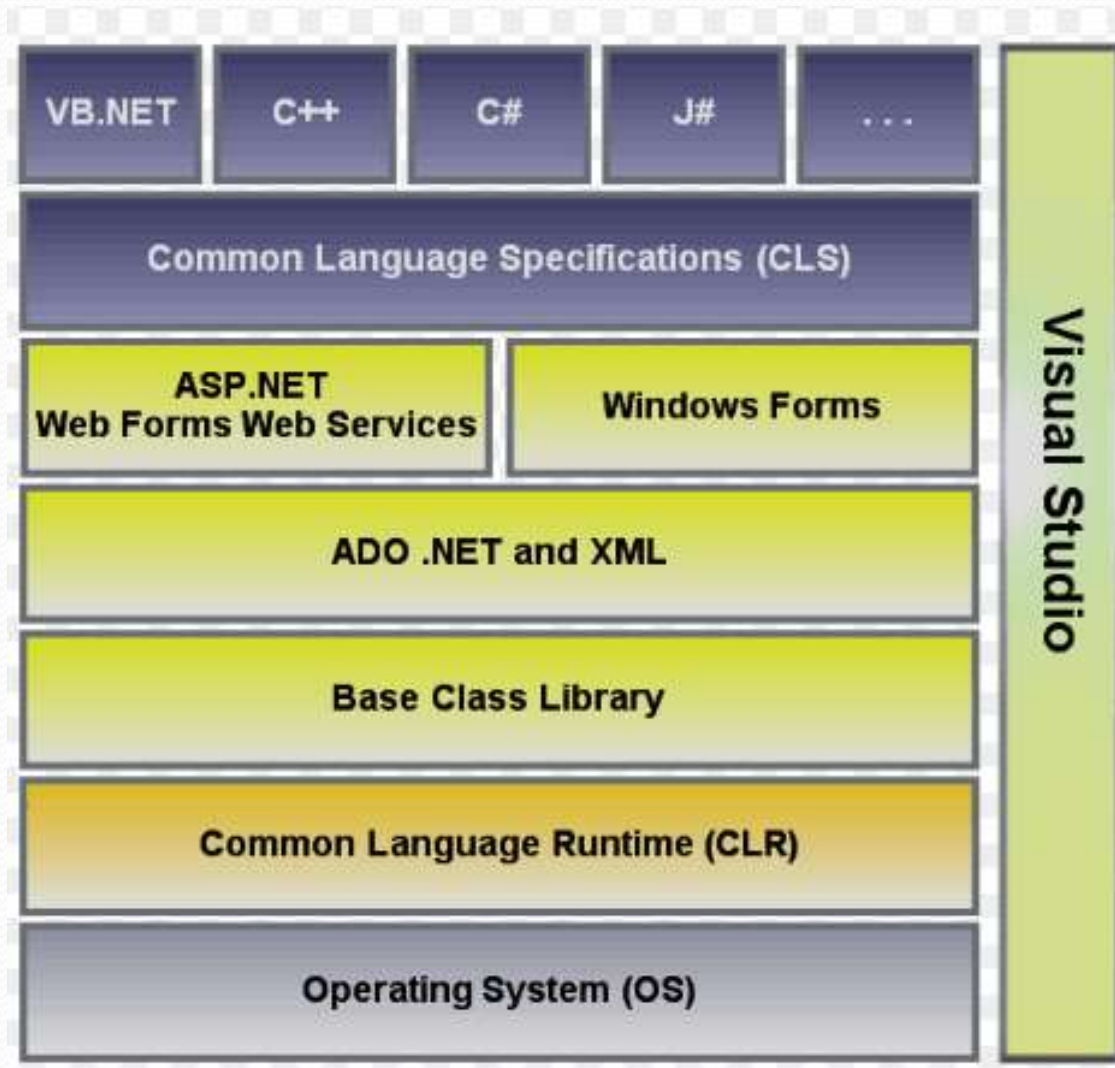
Plataforma .NET - Arquitetura



Modelo de Compilação



Plataforma .NET - Arquitetura



Plataforma .NET - Arquitetura

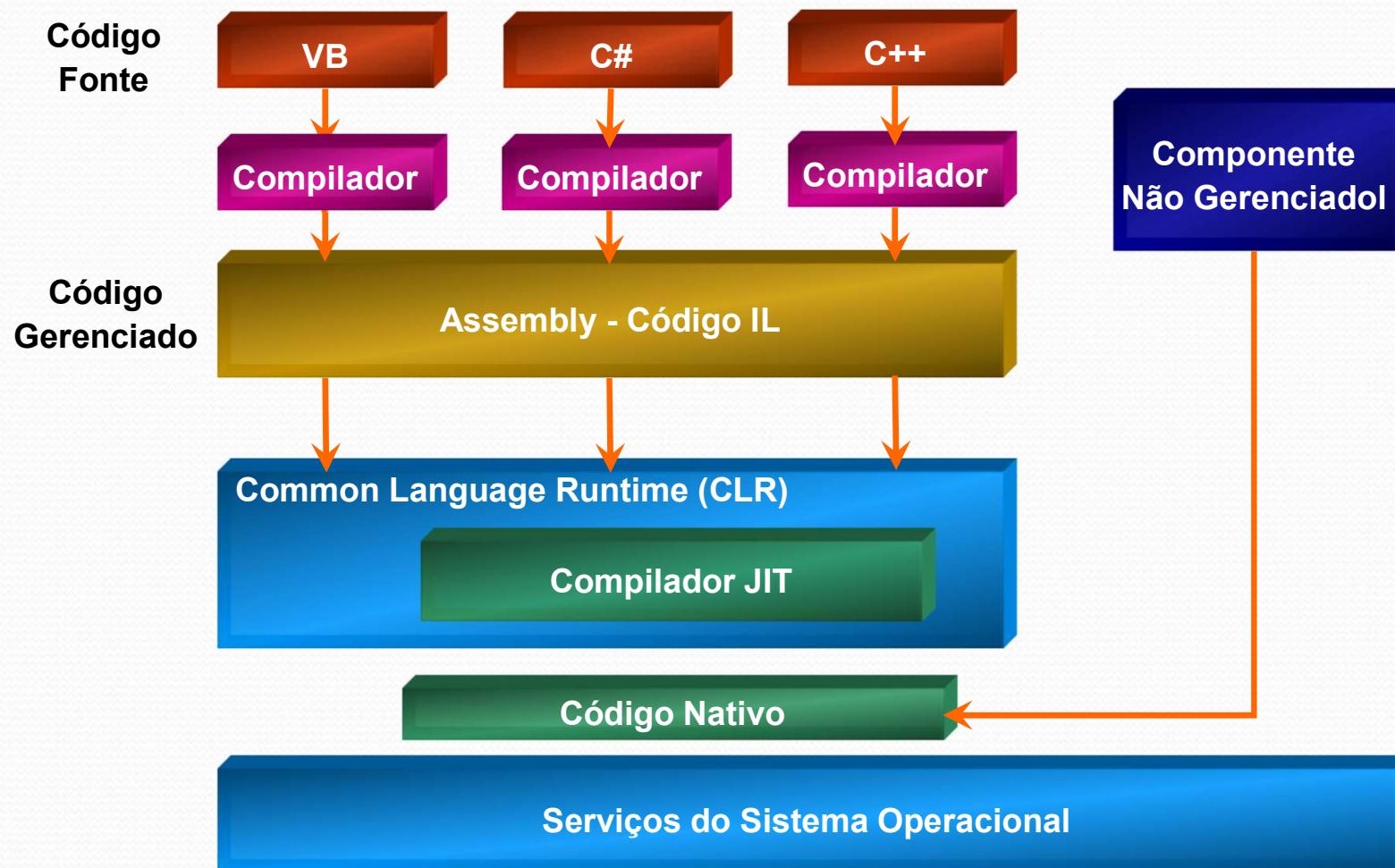
- Código gerenciado (.Net Framework)
 - Código Fonte
 - Compilador
 - Código Intermediário (IL): .DLL ou .EXE
 - Requer o ambiente CLR para executar
 - Código de Máquina (Binário)
- Código não gerenciado (Específico para o SO)
 - Código Fonte
 - Compilador
 - Código de Máquina (Binário)

Obs: NÃO requer o ambiente CLR para executar

Plataforma .NET - CLR

- O Common Language Runtime (CLR) é o componente encarregado de gerenciar aplicações desenvolvidas em .NET.
- O compilador de cada linguagem segue uma série de especificações, conhecidas como Common Language Infrastructure (CLI).
- Estas especificações são abertas (ECMA-335, ISO/IEC 23271), assim permitem a interoperabilidade entre outras linguagens e plataformas.

Plataforma .NET – o que é?



.NET Core

- uma plataforma de desenvolvimento de uso geral mantida pela Microsoft e pela comunidade .NET no GitHub (<https://github.com/dotnet/core>)
- plataforma cruzada e dá suporte ao Windows, macOS e Linux.
- e podendo ser usado no dispositivo, na nuvem e em cenários inserido/IoT.

.NET Core



Inovação



Open Source



Multiplataforma

.NET Core - características

- **Implantação flexível:** pode ser incluído no seu aplicativo ou ser instalado lado a lado no usuário ou em todos os computadores.
- **Plataforma cruzada:** executa em Windows, macOS e Linux e pode ser transferida para outros sistemas operacionais. Os sistemas operacionais com suporte, CPUs e cenários de aplicação serão ampliados com o passar do tempo, fornecidos pela Microsoft, outras empresas e outros indivíduos.
- **Ferramentas de linha de comando:** todos os cenários de produto podem ser exercidos na linha de comando.

.NET Core - características

- **Compatibilidade:** o .NET Core é compatível com .NET Framework, Xamarin e Mono por meio da [.NET Standard](#).
- **Código-fonte aberto:** a plataforma do .NET Core é um software livre que usa licenças do MIT e Apache 2. A documentação é licenciada por [CC-BY](#). O .NET Core é um projeto do [.NET Foundation](#).
- **Suporte da Microsoft:** .NET Core tem suporte pela Microsoft, pelo [Suporte do .NET Core](#)

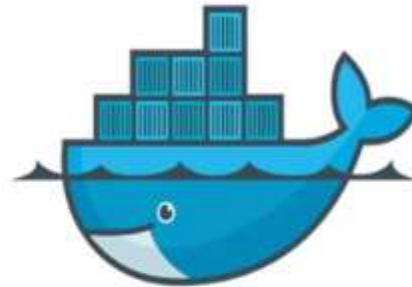
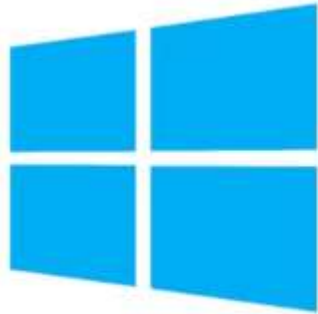
.NET Core - linguagens

- As linguagens C# e F# (o Visual Basic virá em breve) podem ser usadas para criar aplicativos e bibliotecas para .NET Core.
- Os compiladores são executados no .NET Core, permitindo que você desenvolva soluções para .NET Core em qualquer lugar ele é executado.
- Em geral, você não usará os compiladores diretamente, mas sim indiretamente, usando as ferramentas do SDK.

.NET Core – linguagens

- Os compiladores C# e F# e as ferramentas do .NET Core são ou podem ser integradas em vários editores de texto e IDEs, incluindo o Visual Studio, [Visual Studio Code](#), Sublime Text e Vim;
- tornando o desenvolvimento no .NET Core uma boa opção nos seus ambientes de codificação e sistemas operacionais favoritos.
- Essa integração é fornecida, em parte, pelo pessoal do [projeto OmniSharp](#).

.NET Core



docker



C#.net

C#

- Lançada em conjunto com a plataforma .NET
- Completamente nova, sem carregar o “peso” de compatibilidade com versões anteriores
- Moderna, robusta e Orientada a Objetos

C# - curiosidades

História da linguagem C#

- Foi criado junto com a arquitetura .NET. Embora existam várias outras linguagens que suportam essa tecnologia (como VB.NET, C++, J#), C# é considerada a linguagem símbolo do .NET pelas seguintes razões:
 - Foi criada praticamente do zero para funcionar na nova plataforma, sem preocupações de compatibilidade com código de legado.
 - O compilador C# foi o primeiro a ser desenvolvido.
 - A maior parte das classes do .NET Framework foram desenvolvidas em C#.
- A criação da linguagem, embora tenha sido feita por vários desenvolvedores, é atribuída principalmente a Anders Hejlsberg. Anders era desenvolvedor de compiladores na Borland, e entre suas criações mais conhecidas estão o Turbo Pascal e o Delphi.

O nome C#

- Muitos pensam que o nome C# viria de uma sobreposição de 4 símbolos "+" dando a impressão de "++++". Na verdade o "#" de C# refere-se ao sinal musical (sustenido), que aumenta em 1/2 tom uma nota musical.
- O símbolo real seria o \sharp e não o #, porém, devido a limitação de telas, fontes e alguns browsers, no momento da normalização junto a ECMA, fora especificado apenas que o nome da linguagem seria uma letra C maiúscula (U+0043) e o sinal "#" (U+0023), facilitando assim, publicações e artigos com um caracter encontrado facilmente dos layouts de teclado padrões.
- Desta forma, caso o nome fosse usado em português, seria "C-Sustenido" (ou "Dó-Sustenido"), e não "C-cerquilha".

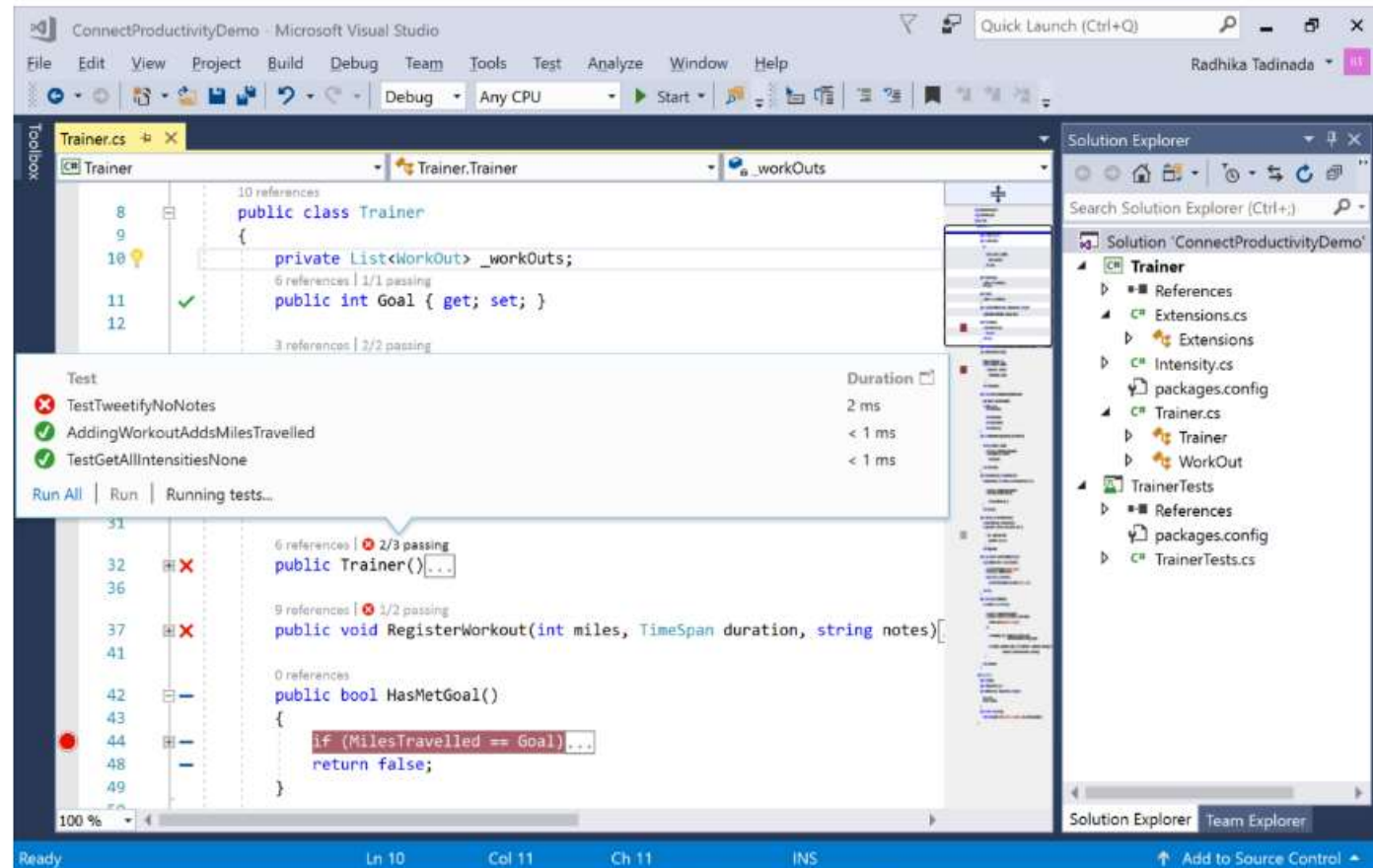
IDE, Editores, <https://www.visualstudio.com>

Compilar aplicativos mais inteligentes, de forma rápida

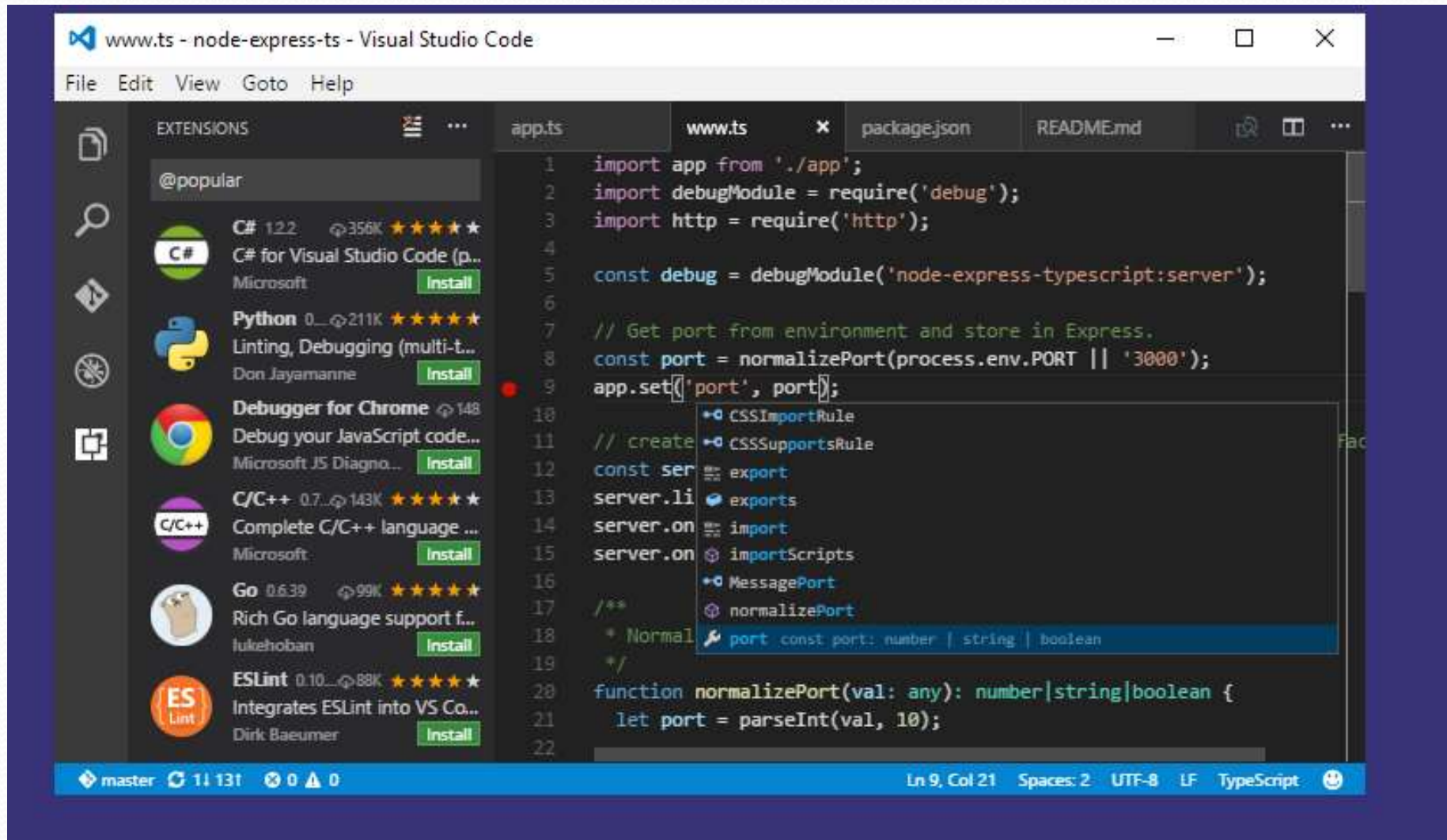
Os novos recursos, como a validação de dependência dinâmica, ajudam a orientar o DevOps mais cedo no processo de desenvolvimento. E mais, os aprimoramentos nos recursos populares, como a navegação do código, IntelliSense, refatoração, correções do código, economizam tempo e esforço, independentemente da linguagem ou da plataforma.

Localizar e corrigir os bugs mais cedo

A experiência de depuração e teste inteira foi melhorada para ajudar a localizar e endereçar os problemas o mais cedo possível. Recursos como o Live Unit Testing, Auxiliares de Exceção e Executar para Clicar restringem o loop do DevOps reduzindo os riscos de regressão e expondo imediatamente a principal causa de novos bugs.



VSCode - Multiplataforma



Get Started with .NET Core

- <https://www.microsoft.com/net/core>
- Selecione a plataforma e o ambiente de command line.
- Clique em [Download .net core SDK](#)
- Siga o exemplo get started

Certificações Developers.net

- **MCP** - A credencial Microsoft Certified Professional é destinada a indivíduos com habilidades de implementar com sucesso produtos ou tecnologias Microsoft.
- **MCDST** - Microsoft Certified Desktop Support Technician prova que você tem a qualificação necessária para dar suporte a usuários finais e solucionar problemas em ambientes de desktop que executam o sistema operacional Microsoft Windows
- **MCAD** - Microsoft Certified Application Developer se destina a desenvolvedores que utilizam ferramentas e tecnologia Microsoft para desenvolver e manter aplicações, componentes, clientes da Web ou de desktop. Destina-se também a profissionais que trabalham em grupos de desenvolvimento de aplicações corporativas.
- **MCSD** - Microsoft® Certified Solution Developer for Microsoft .NET é a principal certificação para profissionais que projetam e desenvolvem soluções de tecnologia usando ferramentas e tecnologias de desenvolvimento Microsoft e o Microsoft .NET Framework.
- **MCDBA** - Microsoft® Certified Database Administrator é a mais importante para profissionais que implementam e administram bancos de dados Microsoft SQL