INTRODUÇÃO À PROGRAMAÇÃO COM C#

contato@rafaelpadilla.net

RAFAEL PADILLA

Doutorando em Engenharia Elétrica
- Universidade Federal do Rio de Janeiro (COPPE/UFRJ)
- Deep Learning
- Inteligência Artificial
- Processamento Digital de Imagens

Mestrado em Engenharia Elétrica pela UFAM
- Reconhecimento de Padrões
- Reconhecimento Facial
- Processamento Digital de Imagens

Graduação em Engenharia Elétrica
- Universidade Federal do Amazonas (UFAM)
- Instituto de Automação Industrial e Engenharia de Software (Universidade Stuttgart)

UFRJ

Universität Stuttgart

contato@rafaelpadilla.net

RAFAEL PADILLA

Microsoft / Instituto Nokia de Tecnologia

- Desenvolvedor C#
- Evangelista Windows Phone
- Pesquisador (Visão Computacional / Imaging)

Pace

- Engenheiro Industrial / Engenheiro de Teste

Adutech

- Desenvolvedor Sênior C#

Instituto de Automação Industrial e Engenharia de Software

- Desenvolvedor C#


TIC NOVA


Microsoft


Inatel


UEA
UNIVERSIDADE
DO ESTADO DO
AMAZONAS


UFRN
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE


SAPIENTIA
PUC-SP


WORLD ACADEMY
OF SCIENCE, ENGINEERING AND TECHNOLOGY
WASET


URI
FREDERICO
WESTPHALEN


UNOCHAPECÓ
UNIVERSIDADE COMUNITÁRIA DA REGIÃO DE CHAPECÓ


FCh
FUNDACIÓN CHILE


Microsoft
Israel R&D Center

RAFAEL PADILLA

contato@rafaelpadilla.net

www.linkedin.com/in/rafael-padilla

https://github.com/rafaelpadilla

C R O N O G R A M A

Pós-Graduação Lato Sensu: Desenvolvimento de Software para Dispositivos Móveis
Módulo: Introdução à Programação com C#

Local: Universidade Católica do Tocantins- TO
Carga horária: 30 horas
Datas:
Parte 1: 07 e 08 de outubro de 2017
Parte 2: 21 e 22 de outubro de 2017

SÁBADO (07/10 e 21/10)	DOMINGO (08/10 e 22/10)
08:00 às 12:00h	08:00 às 12:00h
14:00 às 19:00h	-

C#

EMENTA C#

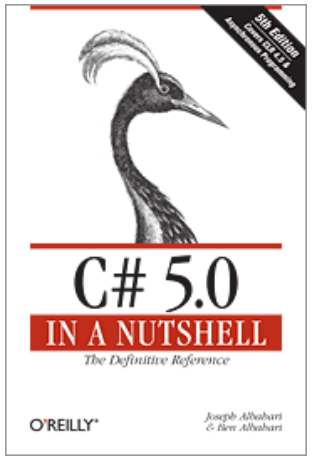
- C# e as Linguagens suportadas pelo Framework
- Visão geral sobre a linguagem C#
- Estrutura de código
- Hello World
- Compilando, Executando e Debugando
- Operações básicas de entrada e saída
- Tipos de Variáveis
- Boas práticas
- Conversão de Variáveis
- Coleções
- Comandos de Iteração for e foreach
- Exceções (try/catch/finally)
- Métodos e Parâmetros
- Sobrecarga de Métodos
- C# e a Orientação a Objetos

EMENTA C#

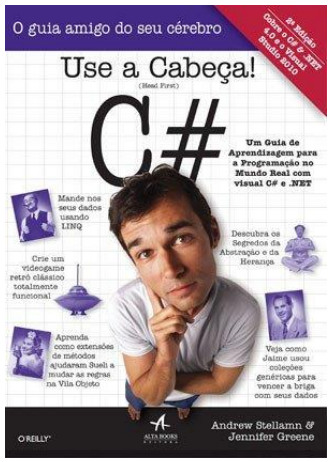
- Encapsulamento
- Get/Set
- Utilizando Variáveis (Tipo Referência)
- Referência em Memória
- Utilizando os tipos comuns
- Hierarquia dos objetos
- Herança em C#
- Utilizando Classes Seladas
- Interfaces
- Classes Abstratas
- Criando e Destruindo Objetos (instâncias)
- Utilizando os Construtores
- LINQ e Lambda
- Threads

REFERÊNCIAS BIBLIOGRÁFICAS

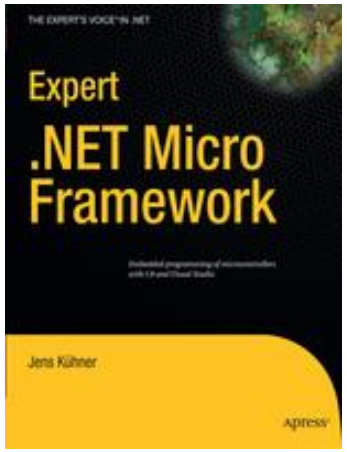
DEITEL, H. M.
C# Como Programar.
Pearson.
1ª Edição.
2003.



ALBAHARI, J**.**
C# 5.0 in a Nutshell: The Definitive Reference.
O'Reilly Media.
2012.



STELLMAN, A.
Use a Cabeça! C#.
Alta Books.
2ª Edição.
2013.



KUHNER, J.
Expert .NET Micro Framework.
Apress. 2008.

AVALIAÇÕES

A avaliação dos alunos será feita através da execução **de 6 projetos pequenos**. O projeto com menor nota será desconsiderado. Para cada projeto será dada uma nota de 0 a 10. A nota final será a média de todos os projetos.

Os pontos avaliados em cada projeto serão:

- Projeto funcional com tratamento de exceções;
- Projeto consistente com as instruções;
- Código comentado;
- Nomenclatura de variáveis, métodos e classes seguindo o padrão C#;

Desafio 1 - SUED

Desafio 2 - Limites de variáveis

Desafio 3 - Inverter caracteres

Desafio 4 - Frequência palavras

Desafio 5 - API Piadas

Desafio 6 - XML de carros

.NET FRAMEWORK

.Net Framework	
Desenvolvedor	Microsoft
Lançamento	13 de fevereiro de 2002
Versão Atual	4.7 (5 de abril de 2017)
http://www.microsoft.com/net	
https://blogs.msdn.microsoft.com/dotnet/2017/04/05/announcing-the-net-framework-4-7/	

.NET FRAMEWORK

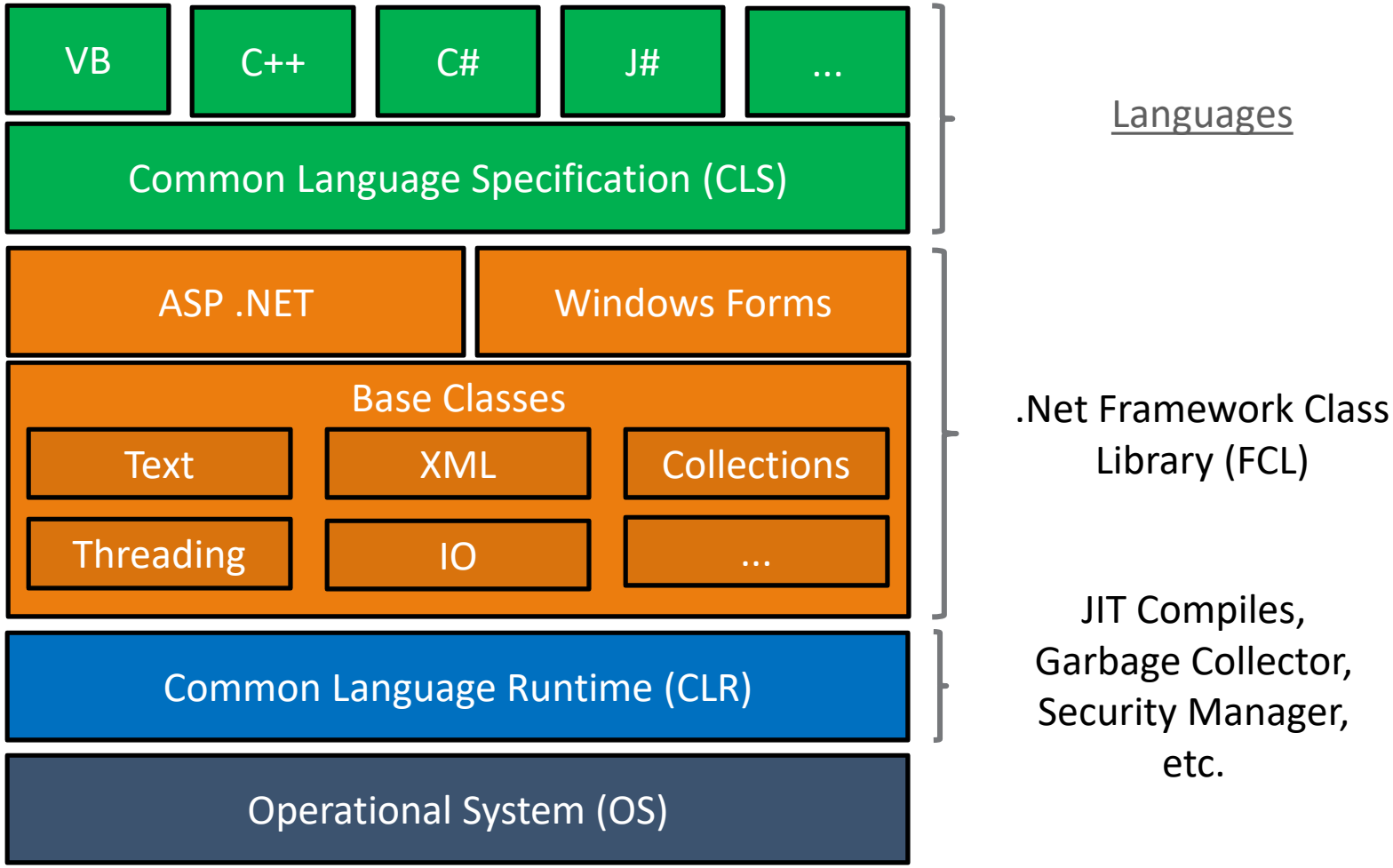
Framework:

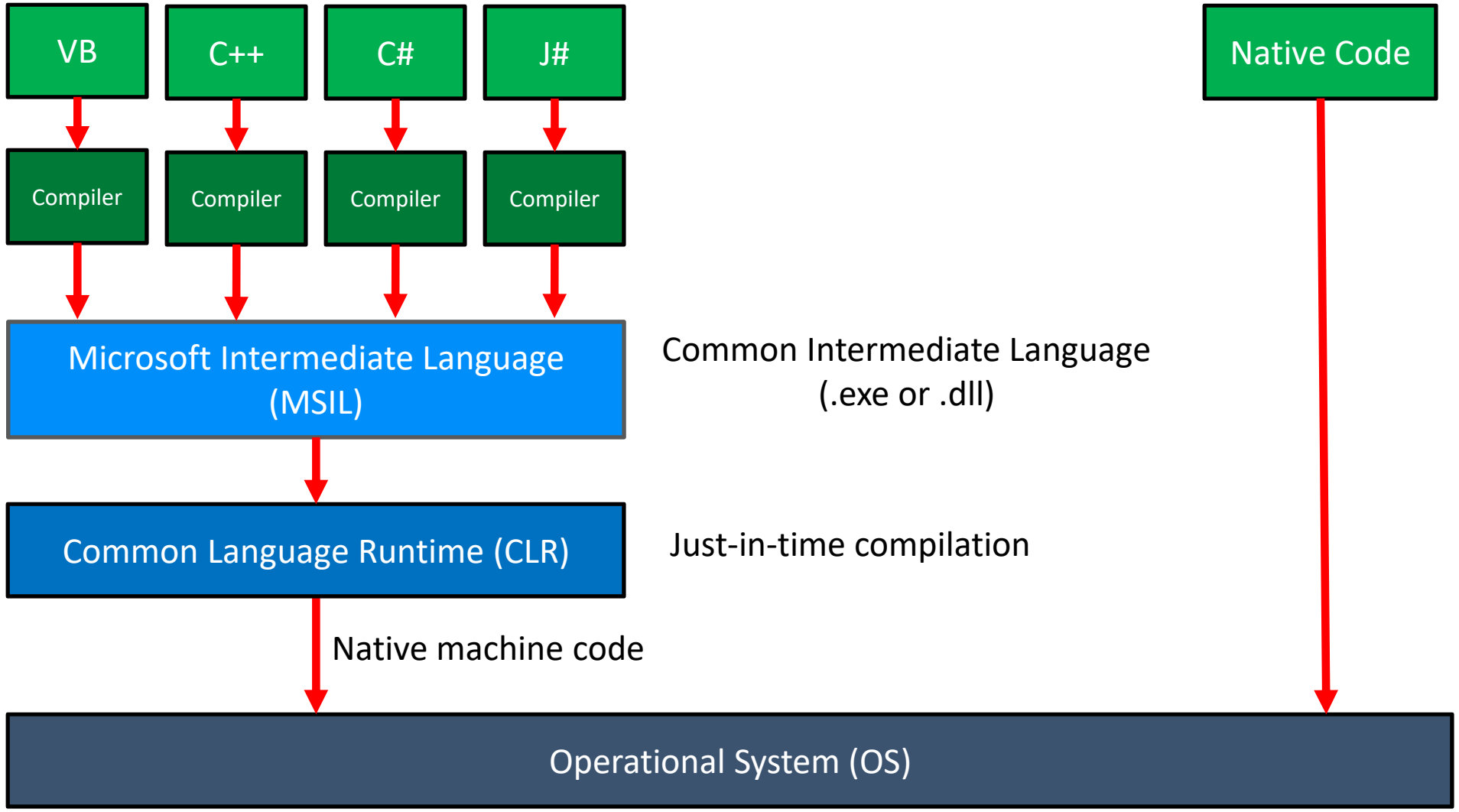
Modelo ou estrutura de dados

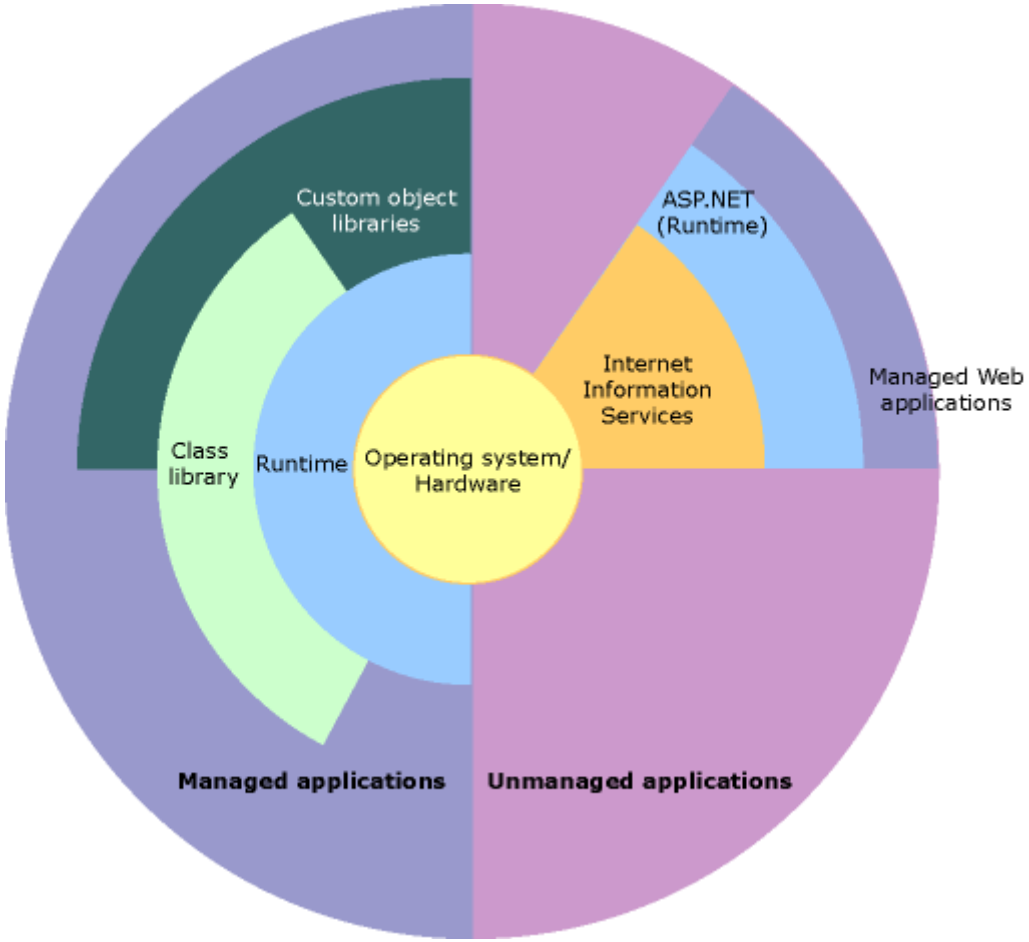
Camadas de um sistema

Estrutura de suporte que conecta diferentes componentes para a realização de uma tarefa

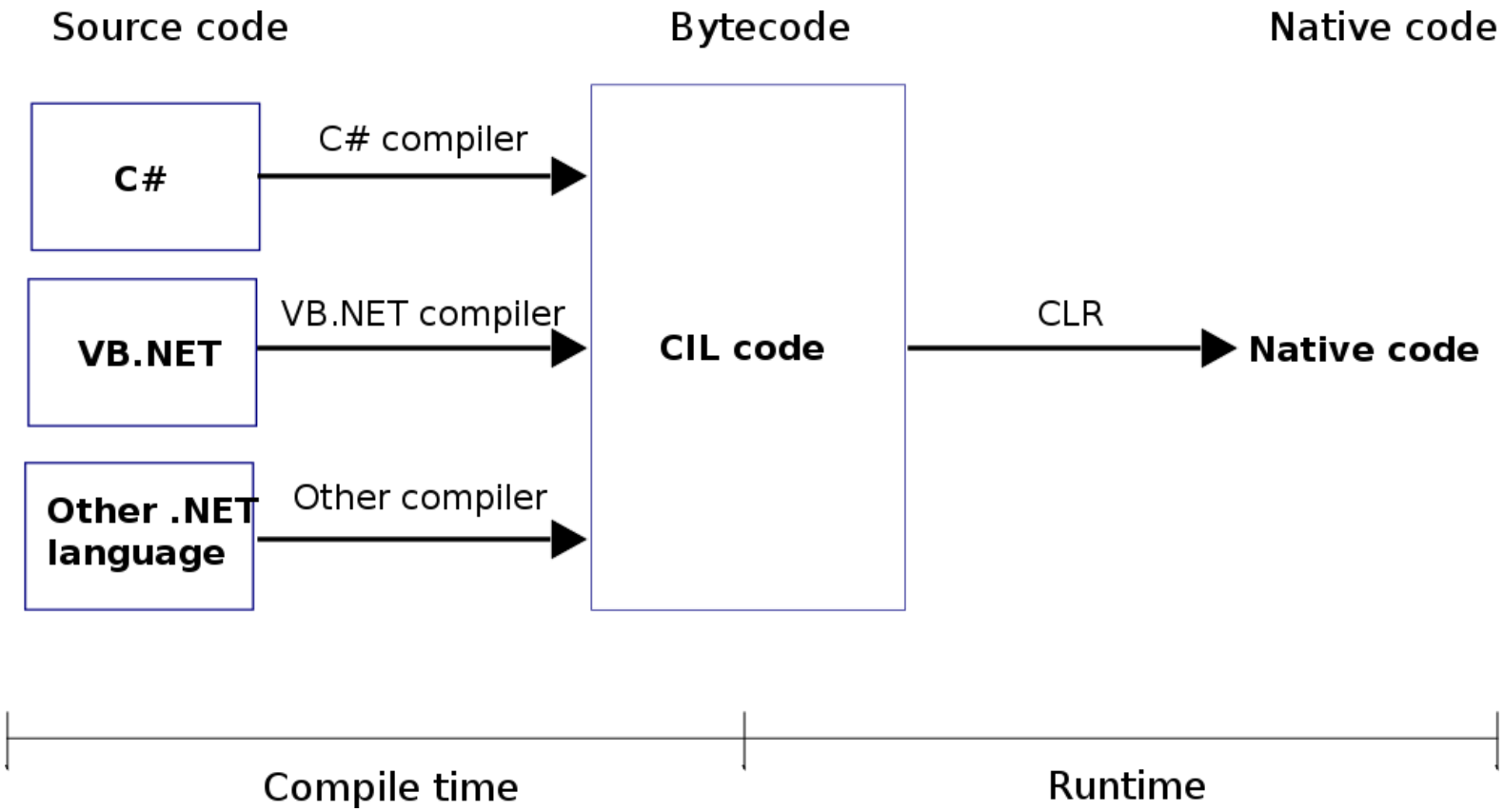
.NET FRAMEWORK



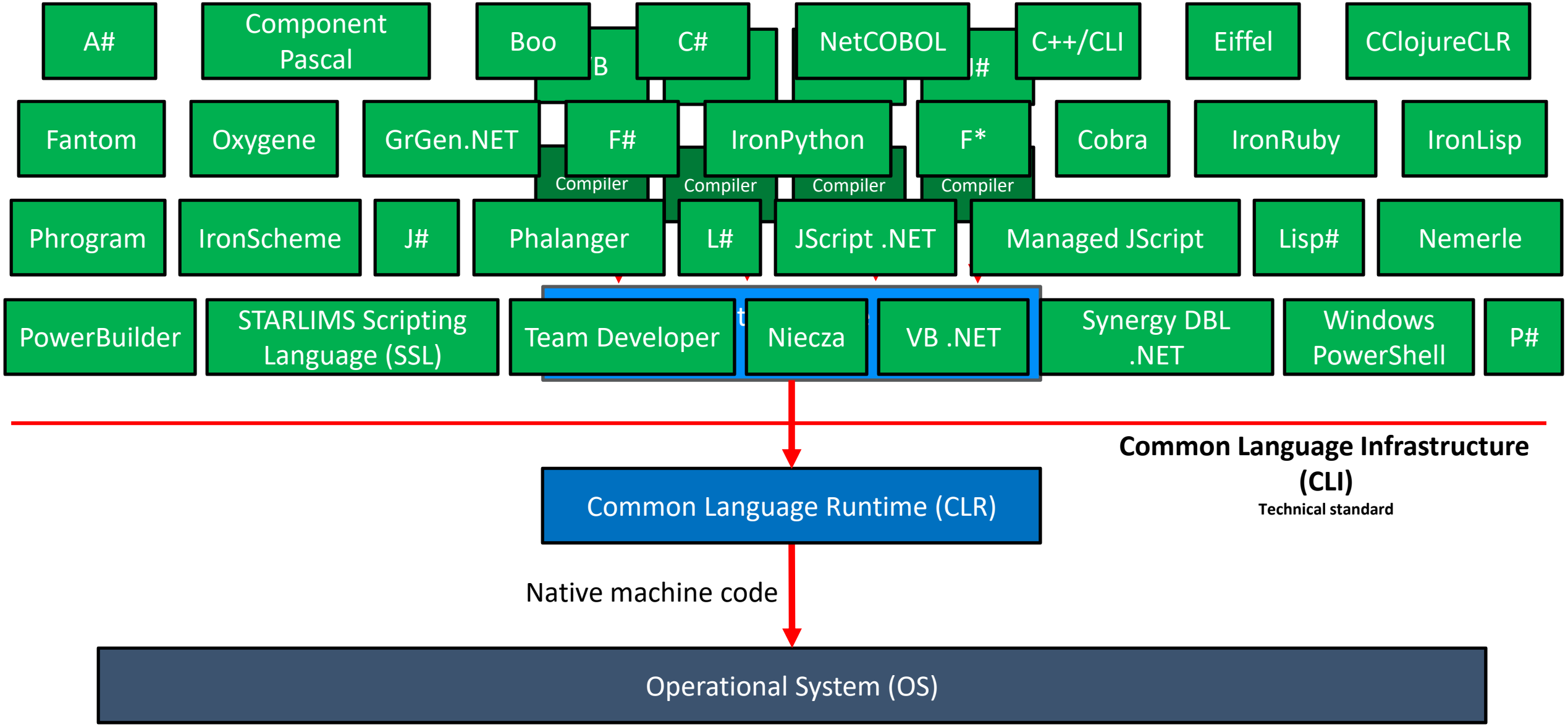

VB
C++
C#
J#
Native Code
Compiler
Compiler
Compiler
Compiler
Microsoft Intermediate Language (MSIL)
Common Intermediate Language (.exe or .dll)
Common Language Runtime (CLR)
Just-in-time compilation
Native machine code
Operational System (OS)



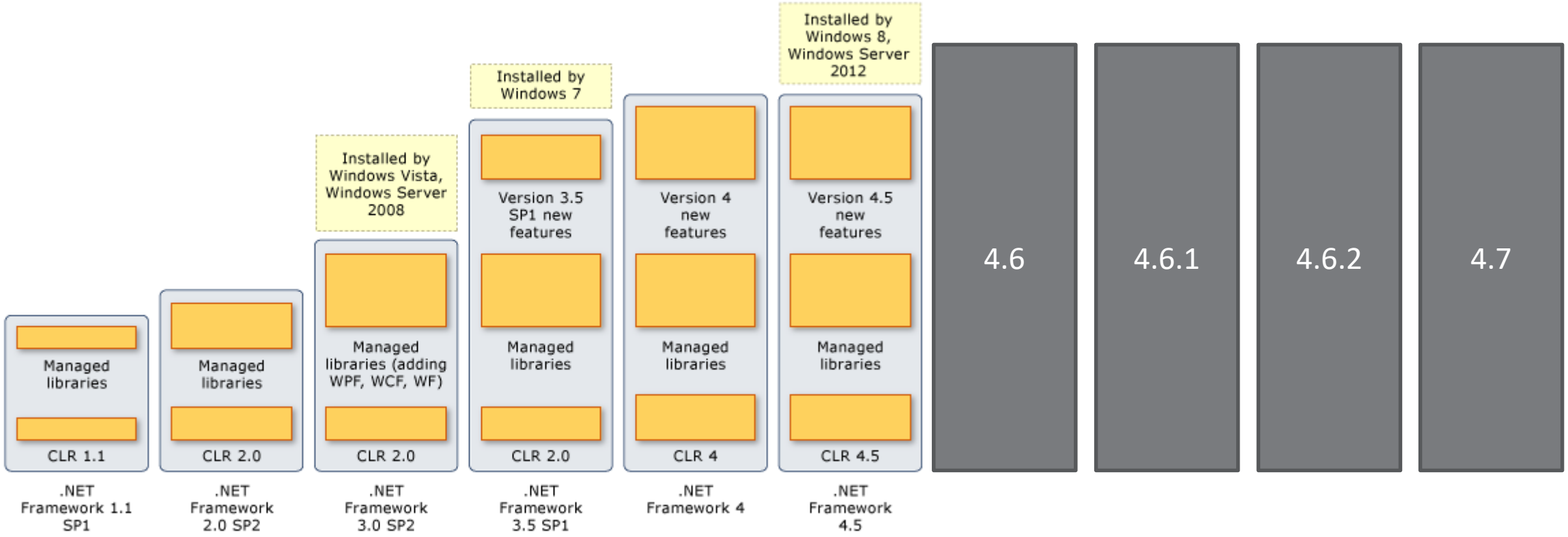
Fonte: MSDN
https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx



Fonte: Wikipedia
https://en.wikipedia.org/wiki/Common_Language_Runtime


A#
Component Pascal
Boo
C#
NetCOBOL
C++/CLI
Eiffel
CClojureCLR
Fantom
Oxygene
GrGen.NET
F#
IronPython
F*
Cobra
IronRuby
IronLisp
Compiler
Compiler
Compiler
Compiler
Phrogram
IronScheme
J#
Phalanger
L#
JScript .NET
Managed JScript
Lisp#
Nemerle
PowerBuilder
STARLIMS Scripting Language (SSL)
Team Developer
Niecza
VB .NET
Synergy DBL .NET
Windows PowerShell
P#
Common Language Infrastructure (CLI)
Technical standard
Common Language Runtime (CLR)
Native machine code
Operational System (OS)

EVOLUÇÃO DO .NET FRAMEWORK



Fonte: Adaptação de MSDN
https://msdn.microsoft.com/en-us/library/vstudio/bb822049(v=vs.100).aspx

NOVIDADES NO .NET FRAMEWORK 4.7

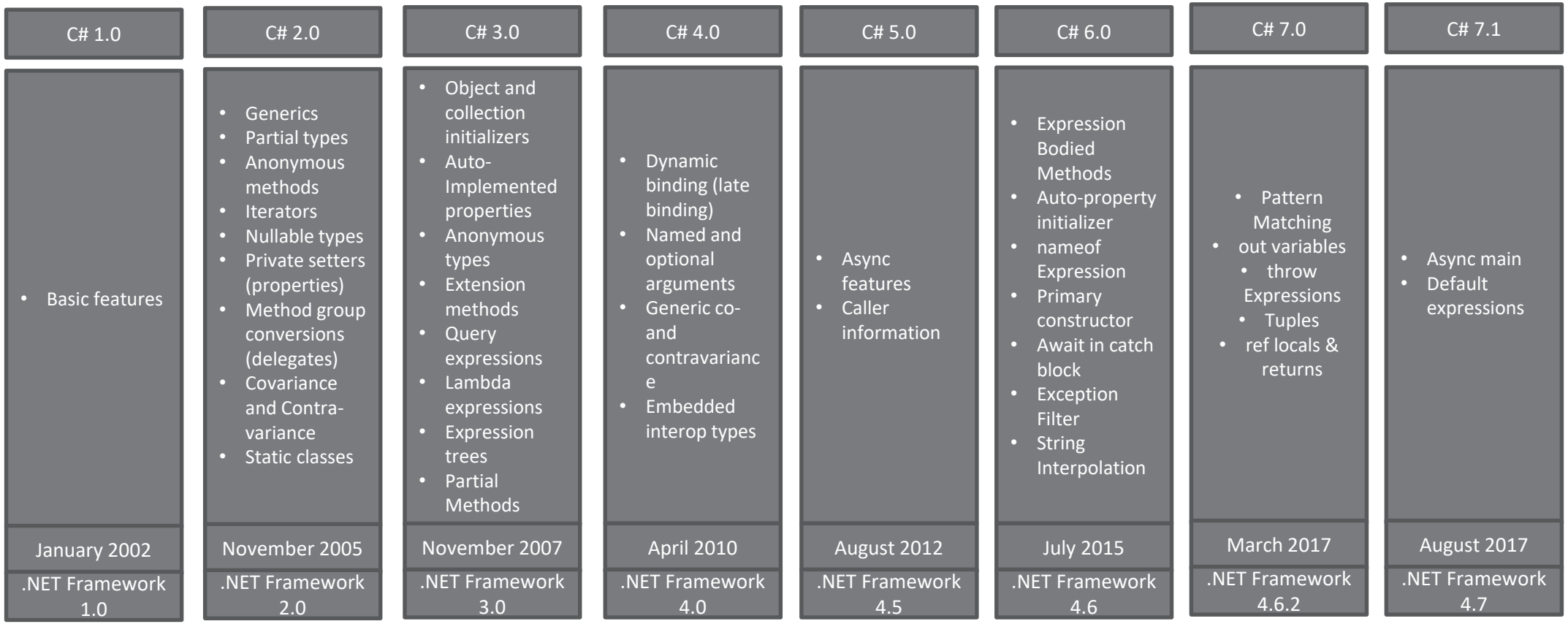
Melhorias nas seguintes áreas:

- Core
- Networking
- ASP.NET
- Windows Communication Foundation (WCF)
- Windows Forms
- Windows Presentation Foundation (WPF)

Principais melhorias:

- Suporte para aplicações Windows Forms para novos hardwares com alta resolução (*high DPI*) – Channel 9
- Suporte de *touch* para Windows 10 Support
- Melhoria no modulo de cryptografia
- Melhorias no desempenho

EVOLUÇÃO DO C#



Fontes:
Wikipedia
Microsoft

APLICAÇÕES COM C#

AMBIENTES

- Windows
- Linux
- Mac
- Docker

DESKTOP/TABLET

- Windows Forms
- Console
- Windows 10

MOBILE

- Windows Phone
- Android
- iPhone

WEB

- Browsers

EMBARCADOS

- Raspberry Pi
- .NET Microframework

CLOUD

- Azure

APLICAÇÕES COM C#
.NET Microframework

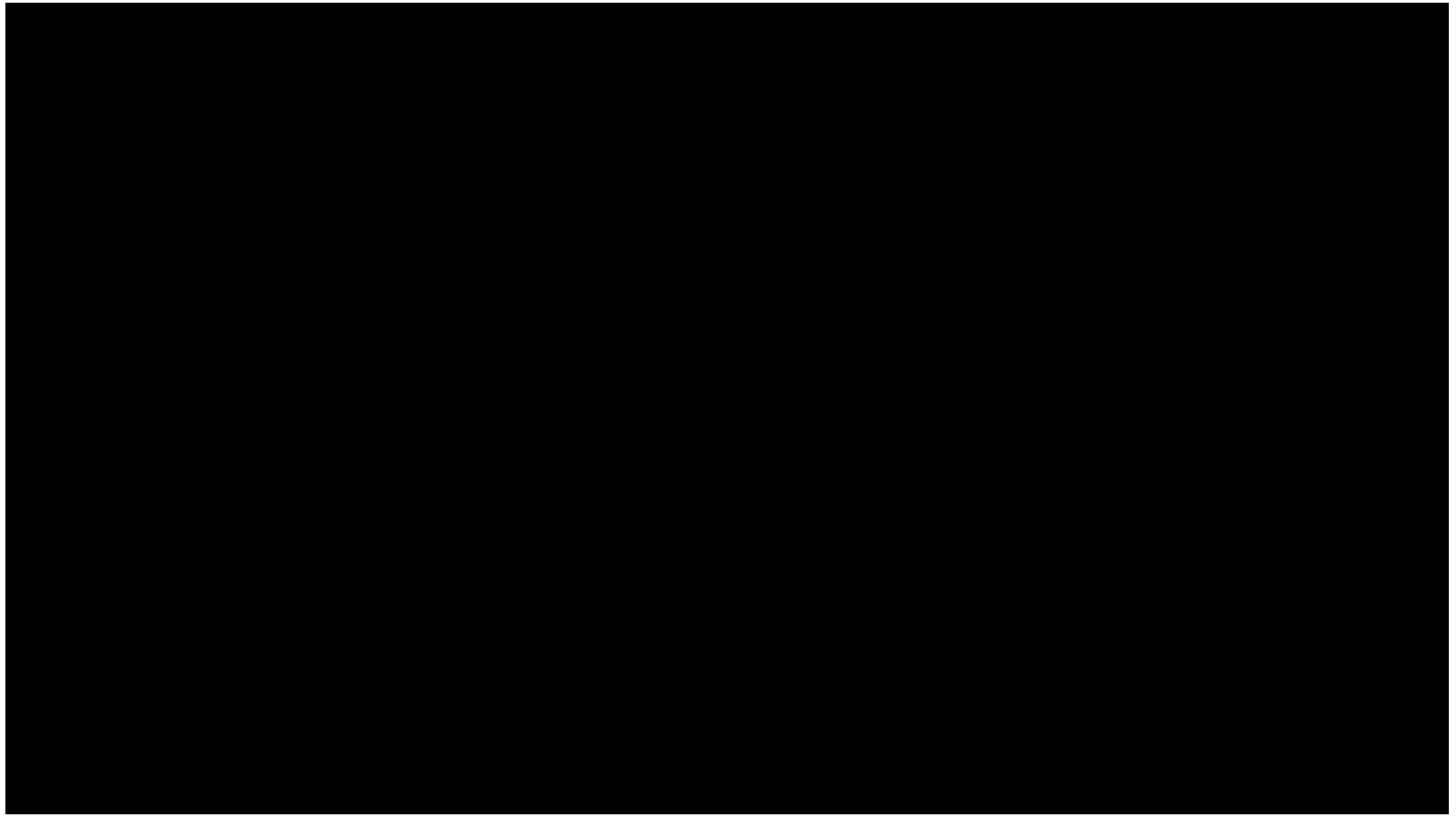
Fonte: Youtube

.NET Microframework

Fonte: Youtube

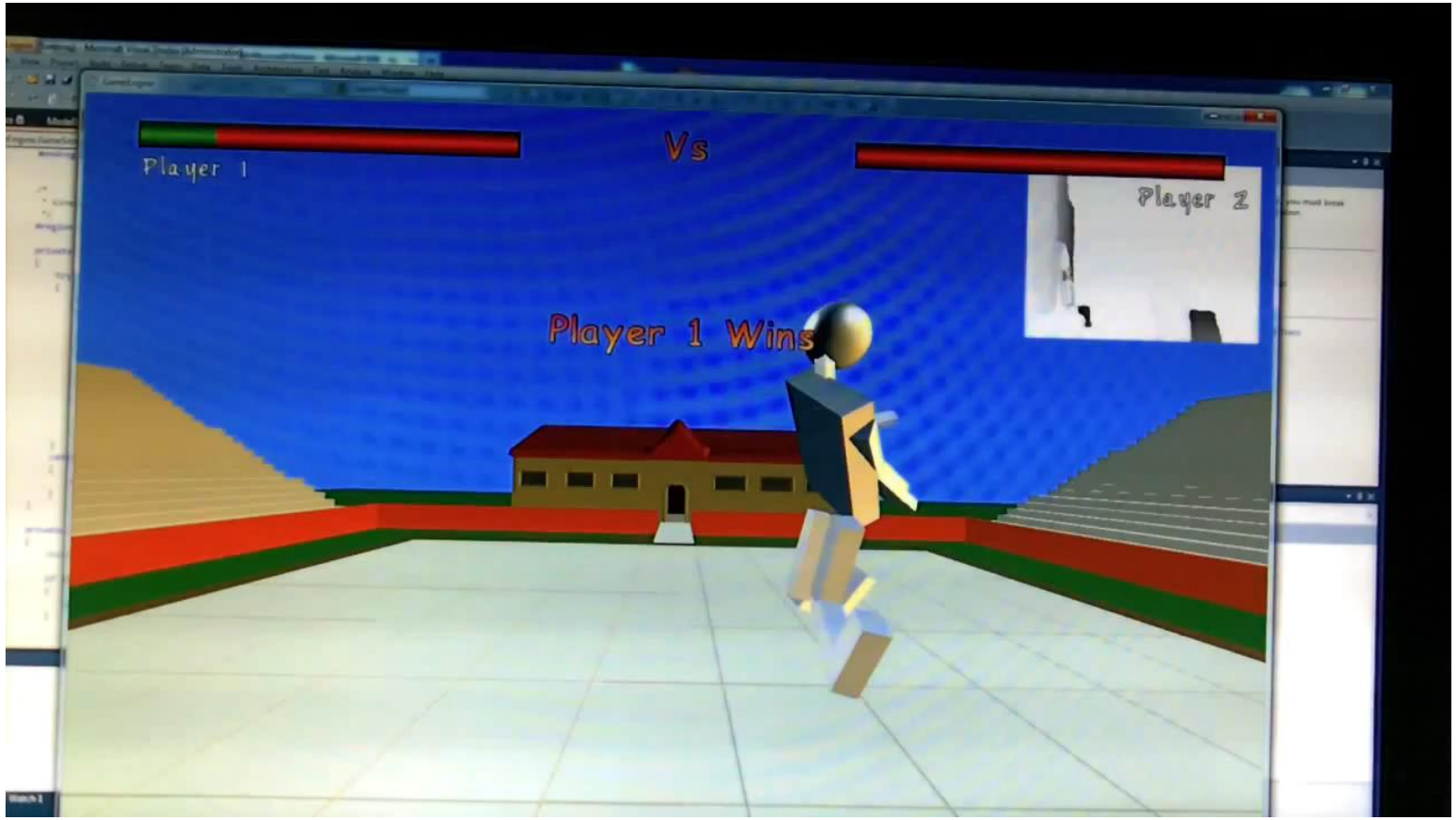
APLICAÇÕES COM C#

Raspberry Pi



Fonte: Youtube

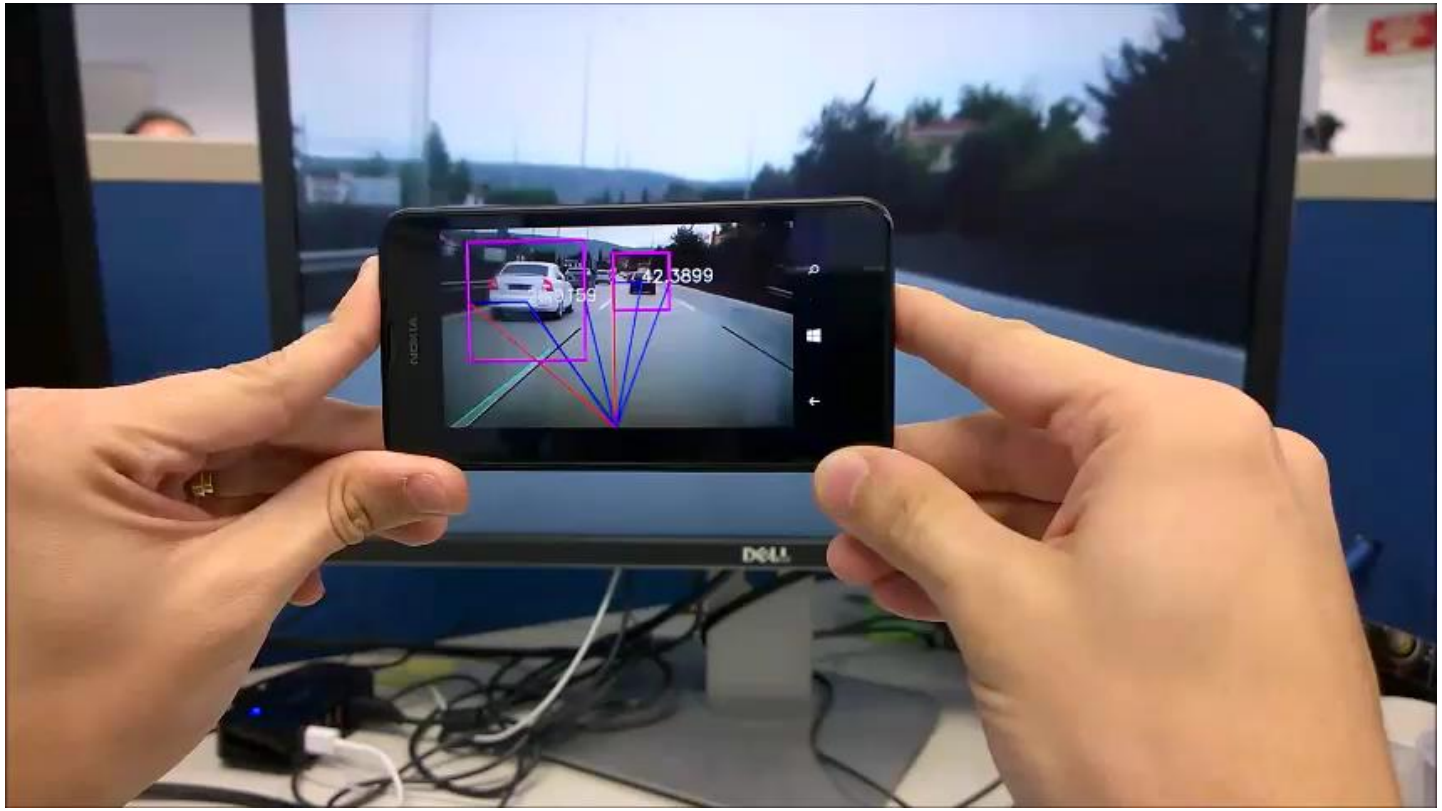
XNA com Kinect



Fonte: Youtube

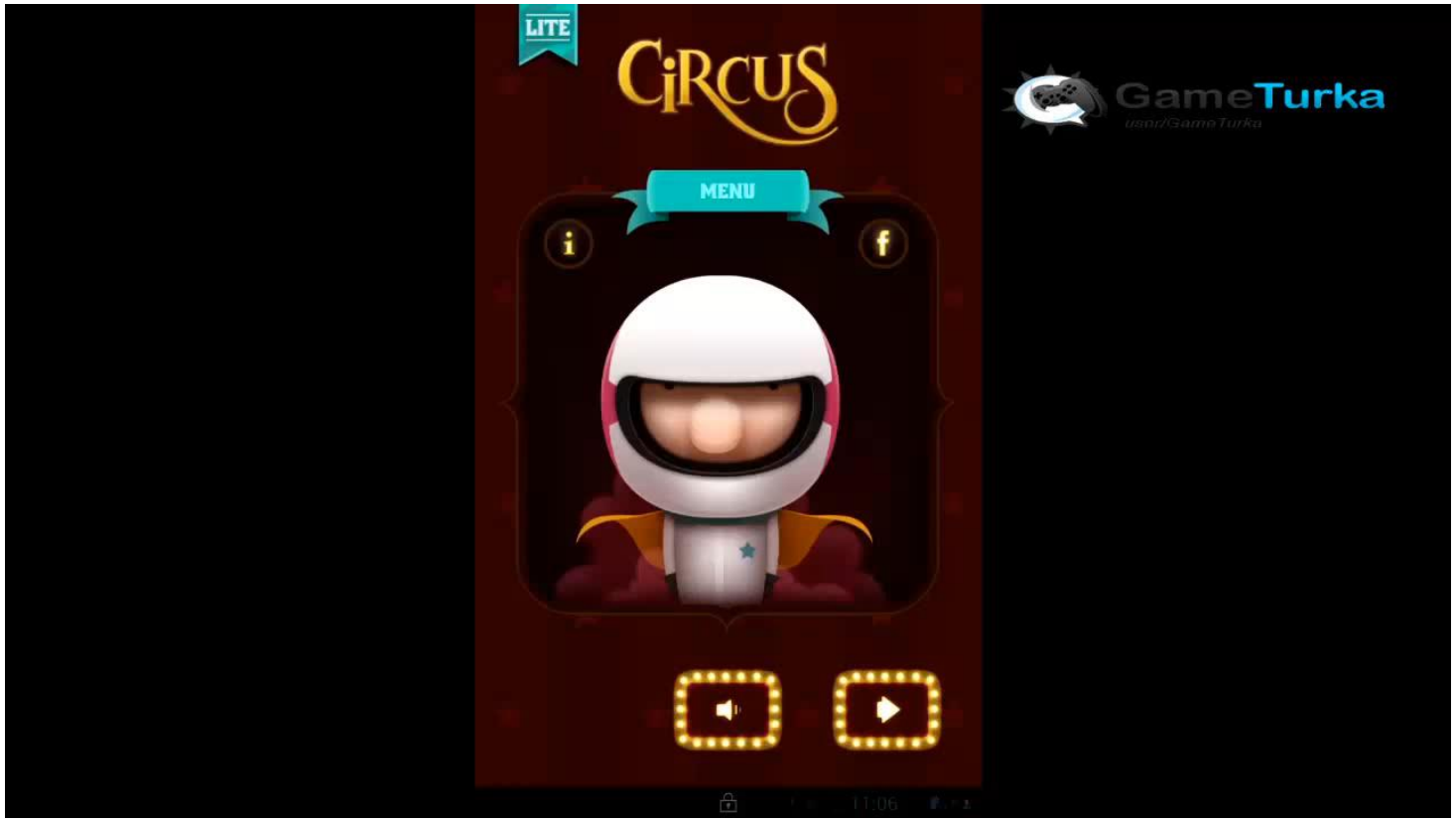
APLICAÇÕES COM C#

Windows Phone

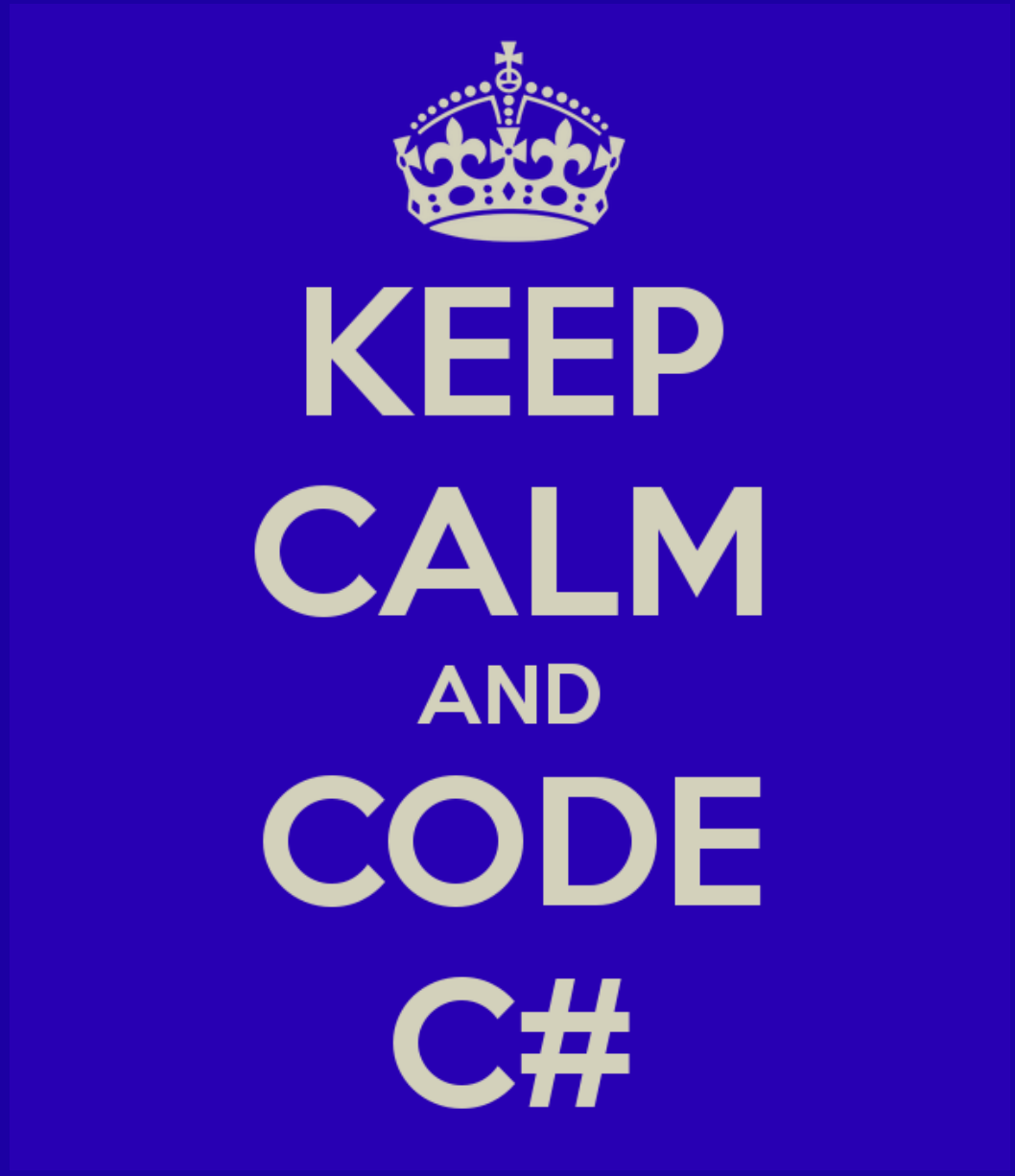


APLICAÇÕES COM C#

Windows Phone com XNA



Fonte: Youtube


KEEP
CALM
AND
CODE
C#

ESTRUTURA DE UM CÓDIGO C#

```
using System;
namespace HelloWorld
{
    class Hello
    {
        static void Main()
        {
            Console.WriteLine("Hello World!");
            Console.ReadKey();
        }
    }
}
```

COMPILANDO, EXECUTANDO E *DEBUGANDO*

Compilando

- Manualmente (Developer Command Prompt for VSXXXX csc MeuArquivo.cs)
- Visual Studio

Executando

Debug x Release

OPERAÇÕES BÁSICAS DE ENTRADA E SAÍDA

Saída
- System.Console.WriteLine()
- System.Console.Write()

Entrada:
- System.Console.Read()
- System.Console.ReadKey()
- System.Console.ReadLine()

Componentes Visuais

Break Points, Auto, Locals, Watch, Immediate Window, etc

DESAFIO #1 (SUED)

DESAFIO #1 (SUED)

No final dos anos 90, um programa de grande sucesso surgiu com o nome de **Einstein**. Sua funcionalidade também foi copiada em soluções similares, a mais famosa foi apresentada com o nome de **SUED**.
Este programa possuía uma interface bastante primitiva: a famosa tela preta do MS-DOS. Apesar da interface pobre, os usuários divertiam-se mostrando para seus amigos que o Einstein era capaz de responder quaisquer perguntas digitadas pelo dono do computador. A parte engraçada era que para o Einstein responder corretamente, era preciso fazer elogios ao programa.
Neste exercício você deverá criar este programa (utilizando C#) para responder as perguntas (e respostas) digitadas pelo usuário. Siga as instruções a seguir:

i) Você deverá definir uma 'frase de elogio' para seu programa.
(Ex: "Máquina dotada de inteligência, capaz de responder todas as minhas perguntas, "
"Programa de grande inteligência, desenvolvido por mentes brilhantes, "
"Minha máquina linda, inteligente, graciosa, superdotada, “)

DESAFIO #1 (SUED)

ii) Ao fazer a pergunta, se o primeiro caractere inserido for um ";", cada próximo caractere digitado deverá mostrar os caracteres da 'frase de elogio' em ordem. Os caracteres que o usuário digitar após o ";" formarão a resposta da pergunta. Neste momento a 'frase de elogio‘ estará sendo exibida conforme as teclas forem sendo pressionadas. Após digitada a resposta, o usuário deverá apertar o "Enter" e, a partir deste momento, todos caracteres inseridos serão mostrados na tela conforme são digitados.
Caso o primeiro caractere inserido não for um ";", todos os caracteres inseridos serão mostrados na tela conforme são digitados e a resposta não estará sendo capturada

iii) Após digitado o "Enter", você deverá mostrar uma 'animação' simulando que o programa está 'pensando' na resposta. (Ex: "Pensando @&%!@#!... ")

iv) Depois de 'pensar', a reposta deverá ser mostrada para o usuário. Caso a resposta não tenha sido informada, uma desculpa deverá ser mostrada (Ex: "Esta eu só respondo para meu mestre" ou "Estou cansado agora, não posso te responder")

TIPOS DE VARIÁVEIS

Data Type	Range
byte	0 .. 255
sbyte	-128 .. 127
short	-32,768 .. 32,767
ushort	0 .. 65,535
int	-2,147,483,648 .. 2,147,483,647
uint	0 .. 4,294,967,295
long	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807

Data Type	Range
ulong	0 .. 18,446,744,073,709,551,615
float	-3.402823e38 .. 3.402823e38
double	-1.79769313486232e308 .. 1.79769313486232e308
decimal	-7922816251 .. 7922816251
char	A Unicode character.
string	A string of Unicode characters.
bool	True or False.
object	An object.

DESAFIO #2 (LIMITES DE VARIÁVEIS)

DESAFIO #2 (LIMITES DE VARIÁVEIS)

Crie um programa em C# mostrando em uma Windows Form os limites máximos e mínimos dos seguintes tipos de variáveis:
byte, sbyte, short, ushort, int, uint, long, ulong, float, double e decimal

Obs: Para obter os limites máximos e mínimos, use as constantes de cada classe.

OBJECT X VAR X DYNAMIC

Object

- Todos objetos em C# são diretamente ou indiretamente herdados de System.Object
- Uma variável System.Object representa a referência de uma instância

Compilador não permite chamar propriedades da classe padrão. Somente com um cast

Var

- Criada para suportar os tipos anônimos (anonymous types), usada no LINQ
- É necessário inicializar a variável
- Seu tipo é definido pelo compilador durante a compilação

Usar var quando não souber o tipo da variável

OBJECT X VAR X DYNAMIC

Dynamic

- Compilador transforma a variável em System.Object
- Todas as validações serão feitas em runtime

Performance pode ser prejudicada

Recomenda-se usar *dynamic* somente com COM APIs

CONVERSÃO DE VARIÁVEIS

Parse / TryParse

- byte
- sbyte
- short
- ushort
- int
- uint
- long
- ulong
- ulong
- float
- double
- decimal
- char
- bool

CONVERSÃO DE VARIÁVEIS

Operador as

- Caso não seja possível fazer a conversão, objeto retornado é nulo.
- Deve ser feito com tipo que pode ser nulo.

CONVERSÃO DE VARIÁVEIS

Conversão explícita (cast)

De	Para
sbyte	**byte** , **ushort**, **uint**, **ulong** ou **char**
byte	**Sbyte** ou **char**
short	**sbyte** , **byte**, **ushort**, **uint**, **ulong** ou **char**
ushort	**sbyte** , **byte**, **short** ou **char**
int	**sbyte** , **byte**, **short**, **ushort**, **uint**, **ulong** ou **char**
uint	**sbyte** , **byte**, **short**, **ushort**, **int**, ou **char**
long	**sbyte** , **byte**, **short**, **ushort**, **int**, **uint**, **ulong** ou **char**

CONVERSÃO DE VARIÁVEIS

Conversão explícita (cast)

De	Para
ulong	**sbyte** , **byte**, **short**, **ushort**, **int**, **uint**, **long** ou **char**
char	**sbyte** , **byte** ou **short**
float	**sbyte** , **byte**, **short**, **ushort**, **int**, **uint**, **long**, **ulong**,**char** ou **decimal**
double	**sbyte** , **byte**, **short**, **ushort**, **int**, **uint**, **long**, **ulong**, **char**,**float** ou **decimal**
decimal	**sbyte** , **byte**, **short**, **ushort**, **int**, **uint**, **long**, **ulong**, **char**,**float** ou **double**

CONVERSÃO DE VARIÁVEIS

Convert

- ToBase64CharArray
- ToBase64String
- ToBoolean
- ToByte
- ToChar
- ToDateTime
- ToDecimal
- ToDouble
- ToInt16
- ToInt32
- ToInt64
- ToSByte
- ToSingle
- ToString
- ToUInt16
- ToUInt32
- ToUInt64

contato@rafaelpadilla.net

www.linkedin.com/in/rafael-padilla

https://github.com/rafaelpadilla