

Programación 3

Orientación a objetos con C#

1. Introducción, uso del IDE

maurogullino@gmail.com

Temario

Lenguaje C#

Teoría de objetos

Patrones de Diseño

Diagramas UML

Windows Forms

X

SQL / ADO.net

LINQ

ASP.net

Documentación

docs.microsoft.com/en-us/dotnet/csharp/programming-guide

docs.microsoft.com/es-es/dotnet/csharp/programming-guide

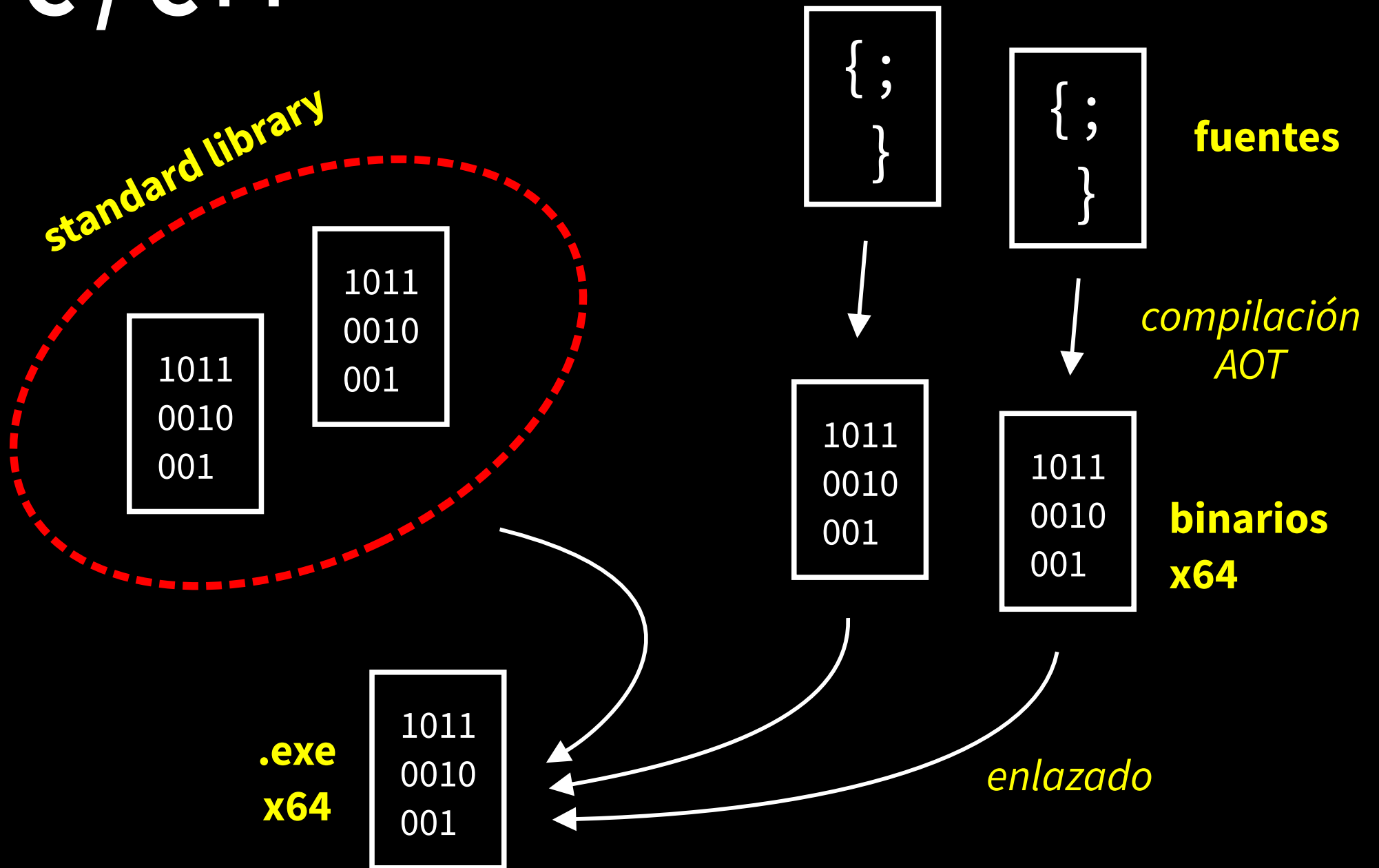
- Deitel & Deitel (castellano)
- Schildt (inglés)

Cronograma

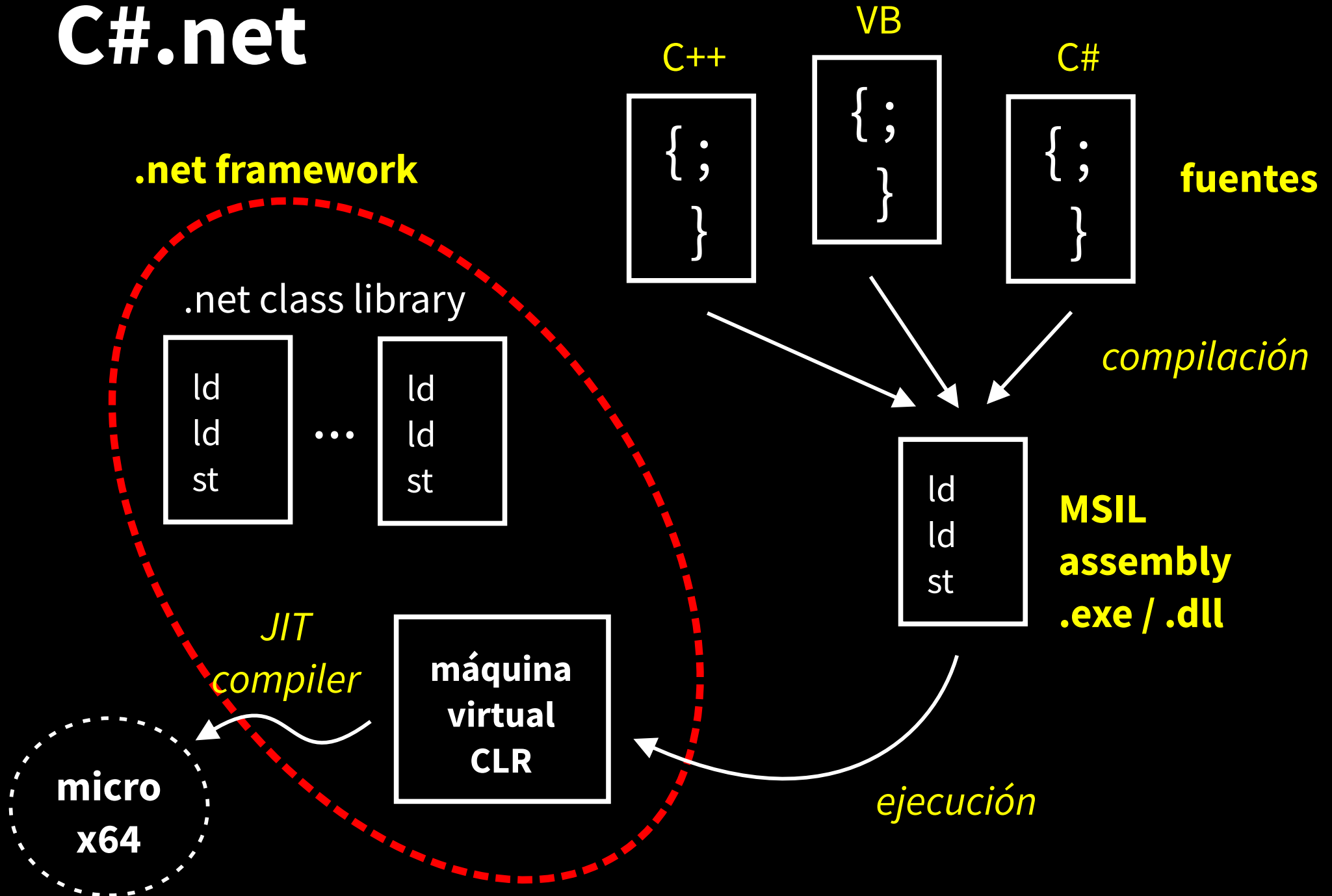
- teóricas
- parcial
- recuperatorio
- proyecto

C# y el framework .net

C / C++



C#.net



Características de un assembly

- Contiene código MSIL a ejecutar por el CLR
- Tiene un solo punto de entrada (Main)
- Es un contexto de seguridad (*security boundary*)
- Contiene tipos (*type boundary*)
- Es la menor cosa versionable (*version boundary*)

Ejemplo MSIL

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        int a = 4;
```

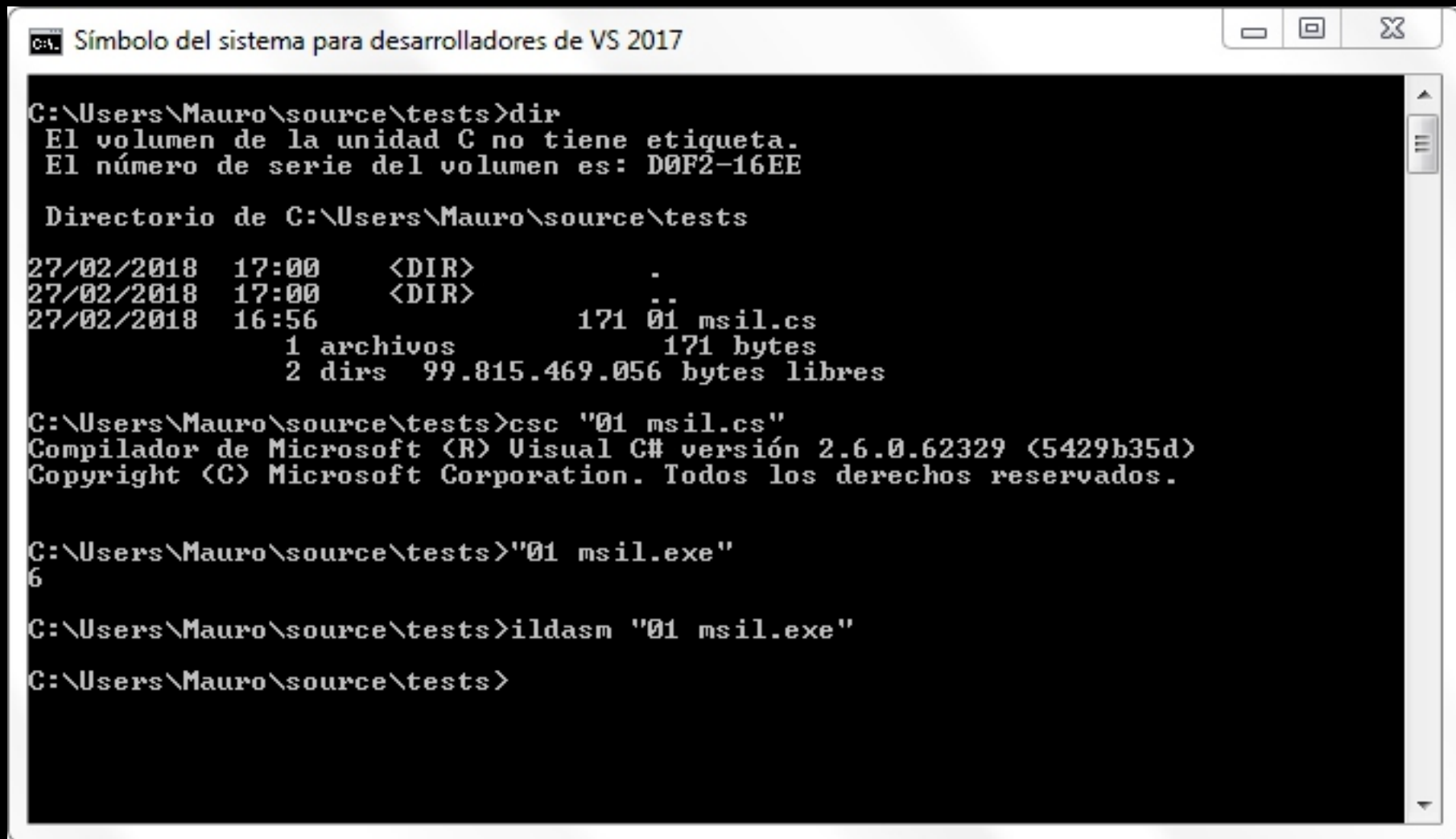
```
        int b = 2;
```

```
        int c = a + b;
```

```
        Console.WriteLine(c);
```

```
    }
```

```
}
```



```
C:\Users\Mauro\source\tests>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: D0F2-16EE

Directorio de C:\Users\Mauro\source\tests

27/02/2018  17:00    <DIR>          .
27/02/2018  17:00    <DIR>          ..
27/02/2018  16:56                171 01 msil.cs
                1 archivos                171 bytes
                2 dirs  99.815.469.056 bytes libres

C:\Users\Mauro\source\tests>csc "01 msil.cs"
Compilador de Microsoft (R) Visual C# versión 2.6.0.62329 (5429b35d)
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Mauro\source\tests>"01 msil.exe"
6

C:\Users\Mauro\source\tests>ildasm "01 msil.exe"

C:\Users\Mauro\source\tests>
```

Ubicación del compilador:

C:\Windows\Microsoft.NET\Framework64\v4.0

- 01 msil.exe
 - MANIFEST
 - Program
 - .class private auto ansi beforefieldinit
 - .ctor : void()
 - Main : void()

Program::Main : void()

Buscar Buscar siguiente

```
.method private hidebysig static void Main() cil managed
{
    .entrypoint
    // Code size      17 (0x11)
    .maxstack 2
    .locals init (int32 V_0,
                  int32 V_1,
                  int32 V_2)
    IL_0000: nop
    IL_0001: ldc.i4.4
    IL_0002: stloc.0
    IL_0003: ldc.i4.2
    IL_0004: stloc.1
    IL_0005: ldloc.0
    IL_0006: ldloc.1
    IL_0007: add
    IL_0008: stloc.2
    IL_0009: ldloc.2
    IL_000a: call        void [mscorlib]System.Console::WriteLine(int32)
    IL_000f: nop
    IL_0010: ret
} // end of method Program::Main
```

```
.assembly '01
{
    .ver 0:0:0:0
```

Otro ejemplo MSIL

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        for(int i=0; i<5; i++)
```

```
            Console.WriteLine(i);
```

```
    }
```

```
}
```

IL_0000:	nop		
IL_0001:	ldc.i4.0		
IL_0002:	stloc.0		
IL_0003:	br.s	IL_0010	branch
IL_0005:	ldloc.0		
IL_0006:	call	WriteLine(int32)	
IL_000b:	nop		
IL_000c:	ldloc.0		
IL_000d:	ldc.i4.1		
IL_000e:	add		
IL_000f:	stloc.0		
IL_0010:	ldloc.0		
IL_0011:	ldc.i4.5		
IL_0012:	clt		compare lesser than
IL_0014:	stloc.1		
IL_0015:	ldloc.1		
IL_0016:	brtrue.s	IL_0005	branch if true
IL_0018:	ret		

Otros lenguajes con compilación intermedia

- Java
- Smalltalk
- Action Script

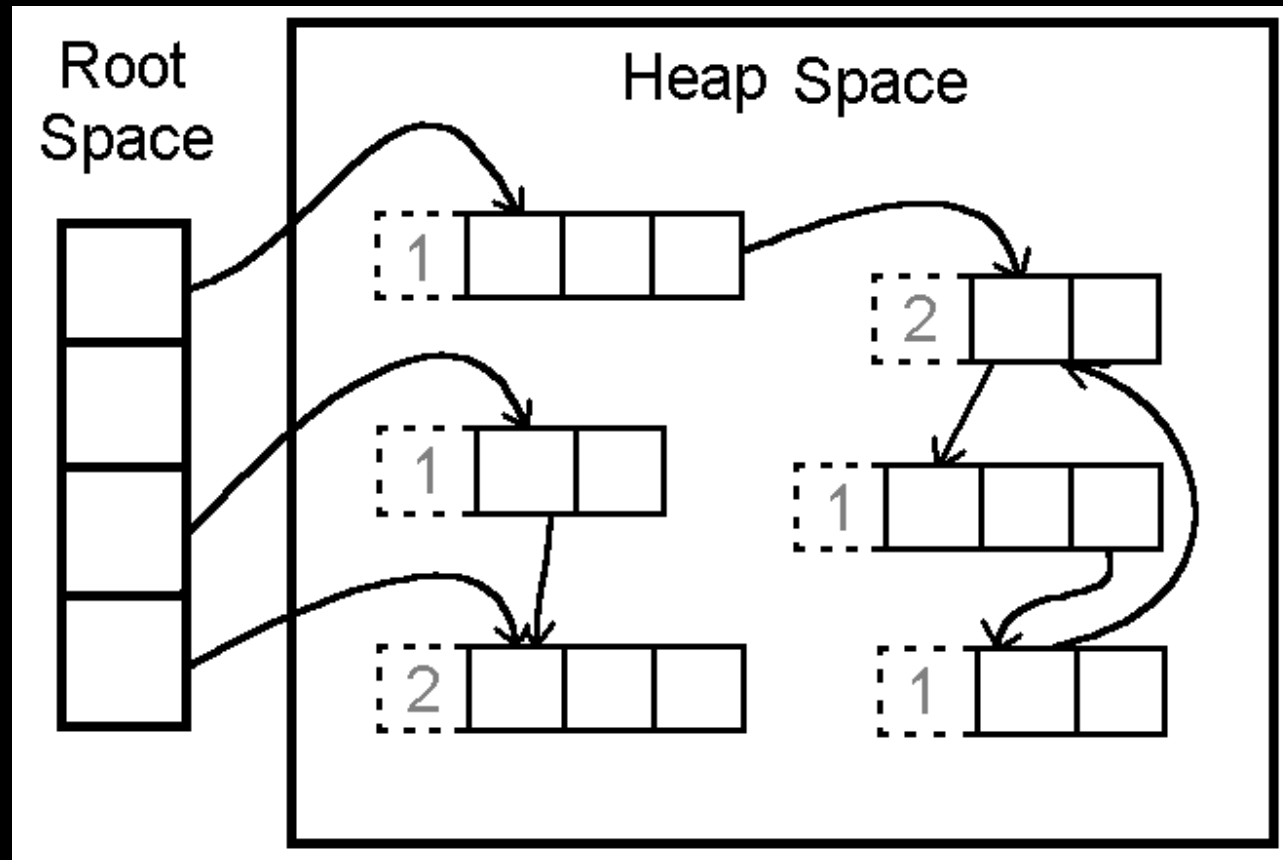
Instrucciones MSIL

https://en.wikipedia.org/wiki/List_of_CIL_instructions

Características del CLR

- “Managed code”
- Sistema de tipos común
- Distintos lenguajes
- Garbage collection
- Facilita el threading

Garbage Collector



Reference counting / Mark - sweep

Versiones

C# 4.0 / .NET 4 / VS2010 Abr 2010

C# 5.0 / .NET 4.5 / VS2012 Ago 2012

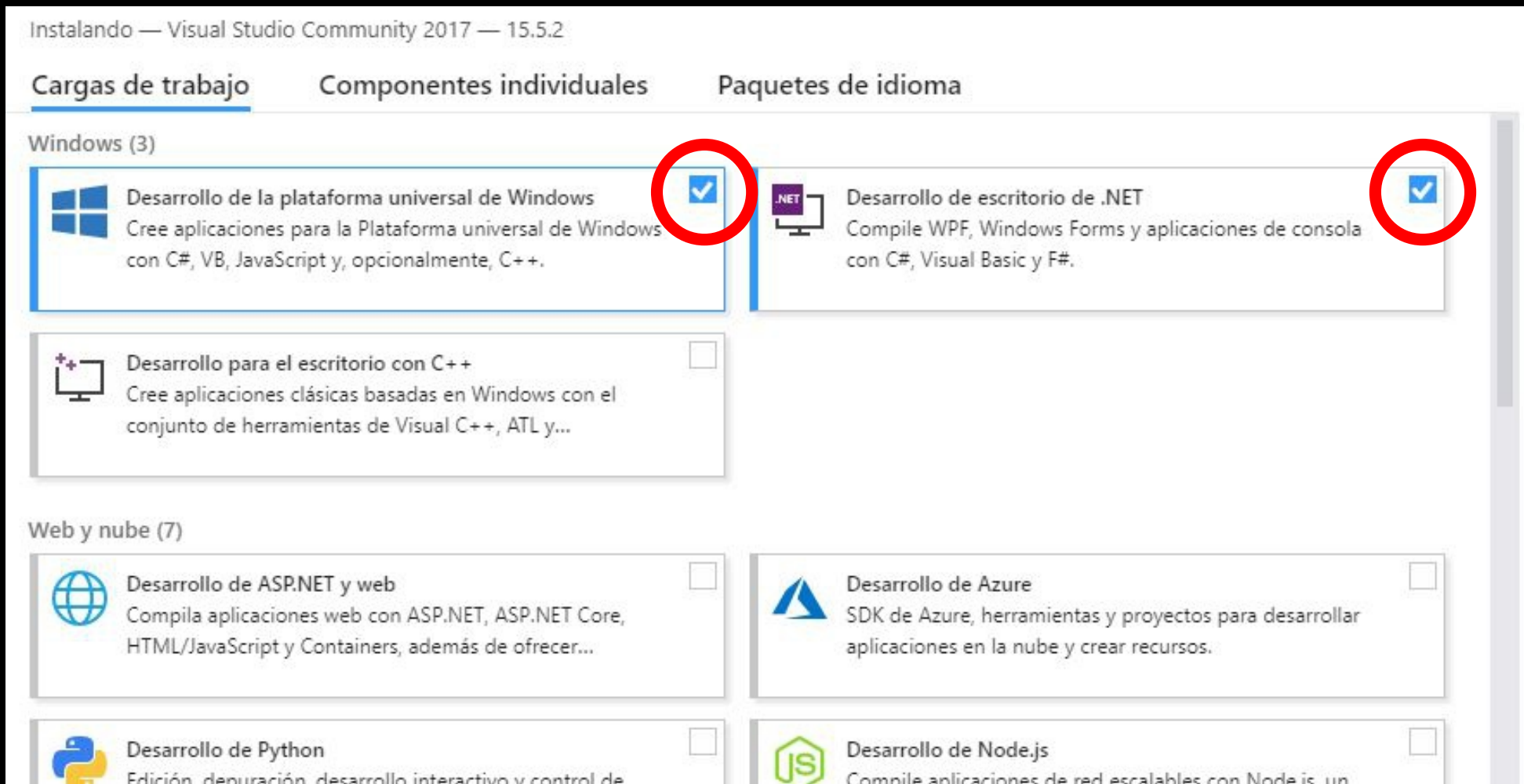
C# 6.0 / .NET 4.6 / VS2015 Jul 2015

C# 7.0 / .NET 4.7 / VS2017 Mar 2017

Visual Studio

Instalar **Visual Studio Community 2019**

visualstudio.microsoft.com/es/downloads



Nuevo proyecto

► Recientes

.NET Framework 4.6.1

Ordenar por: Predeterminado

▲ Instalado

▲ Visual C#

Windows Universal

Escritorio clásico de Windows

.NET Core

.NET Standard

Prueba

► Otros lenguajes

► Otros tipos de proyectos

► En línea

¿No encuentra lo que busca?

[Abrir el instalador de Visual Studio](#)



Aplicación vacía (Windows universal)

Visual C#



Aplicación de WPF (.NET Framework)

Visual C#



Aplicación de Windows Forms (.NET Framework)

Visual C#



Aplicación de consola (.NET Core)

Visual C#



Aplicación de consola (.NET Framework)

Visual C#



Biblioteca de clases (.NET Standard)

Visual C#



Biblioteca de clases (.NET Framework)

Visual C#



Proyecto compartido

Visual C#



Biblioteca de clases (portátil heredada)

Visual C#



Biblioteca de clases (Windows universal)

Visual C#

Nombre:

App1

```
using System;

namespace Prueba
{
    class Program
    {
        static void Main()
        {
            string ingreso;
            ingreso = Console.ReadLine();
            foreach (char letra in ingreso)
                Console.WriteLine(letra);

            Console.ReadKey(false);
        }
    }
}
```

F5: Ejecutar

F9: Setear punto de interrupción

F11: Avanzar instrucción

F10: Avanzar sin entrar en llamadas

Shift+F11: Avanzar hasta el retorno

Shift+F5: Detener

Ctrl+F5: Ejecutar sin depuración

Comparando con C++

Similitudes

- Sintaxis (heredada de C)
- Concepto de clases y objetos
- Modelo de herencia, *abstract*, *virtual*
- Constructores y destructores

Comparando con C++

Diferencias

- Vocabulario (variables miembro = Fields)
- Resultado de la compilación (IL)
- Administración de memoria
- Relación con las bibliotecas estándar
- Punto de entrada Main
- Clases y structs son diferentes en C#
- *Threading* nativo en C#
- *Reflection* en C#