

# INTRODUCCION A LA PROGRAMACION C#

## 1 - INTRODUCCIÓN A MICROSOFT .NET

## UNIDAD: 1

**PRESENTACIÓN:** En esta unidad se explicarán los fundamentos y conceptos sobre los cuales está construido el framework de desarrollo .NET.

### OBJETIVOS

**Que los participantes logren:** Comprender los fundamentos de Microsoft .NET y el contexto de trabajo para un desarrollador que se inicia en C#.

## TEMARIO

Microsoft .Net.....	4
Contexto histórico .....	4
Evolución.....	4
Disponibilidad: .....	5
Estructura de .NET .....	6
Lenguajes Soportados.....	7
Common Language Runtime (CLR) .....	7
Lenguaje Común Intermedio (CIL).....	8
Instrucciones CIL .....	9
Metadatos .....	9
Ensamblados.....	9
Librería de clase base (BCL) .....	10
Operaciones a cargo de la BCL: .....	11
Solo Windows o Plataforma abierta ? .....	12
Sistema de Tipo Común (CTS) .....	12
Que es un tipo ? .....	13
Especificación Común del Lenguaje (CLS) .....	13

## Microsoft .Net

.NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones.

Basada en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

### Contexto histórico

Lanzamiento Inicial: 13 de febrero de 2002

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Oracle Corporation y a los diversos framework de desarrollo web basados en PHP.

Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

### Evolución

Año	Versión
2002	1.0
2005	2.0
2006	3.0
2007	3.5
2010	4.0
2012	4.5

Disponibilidad:

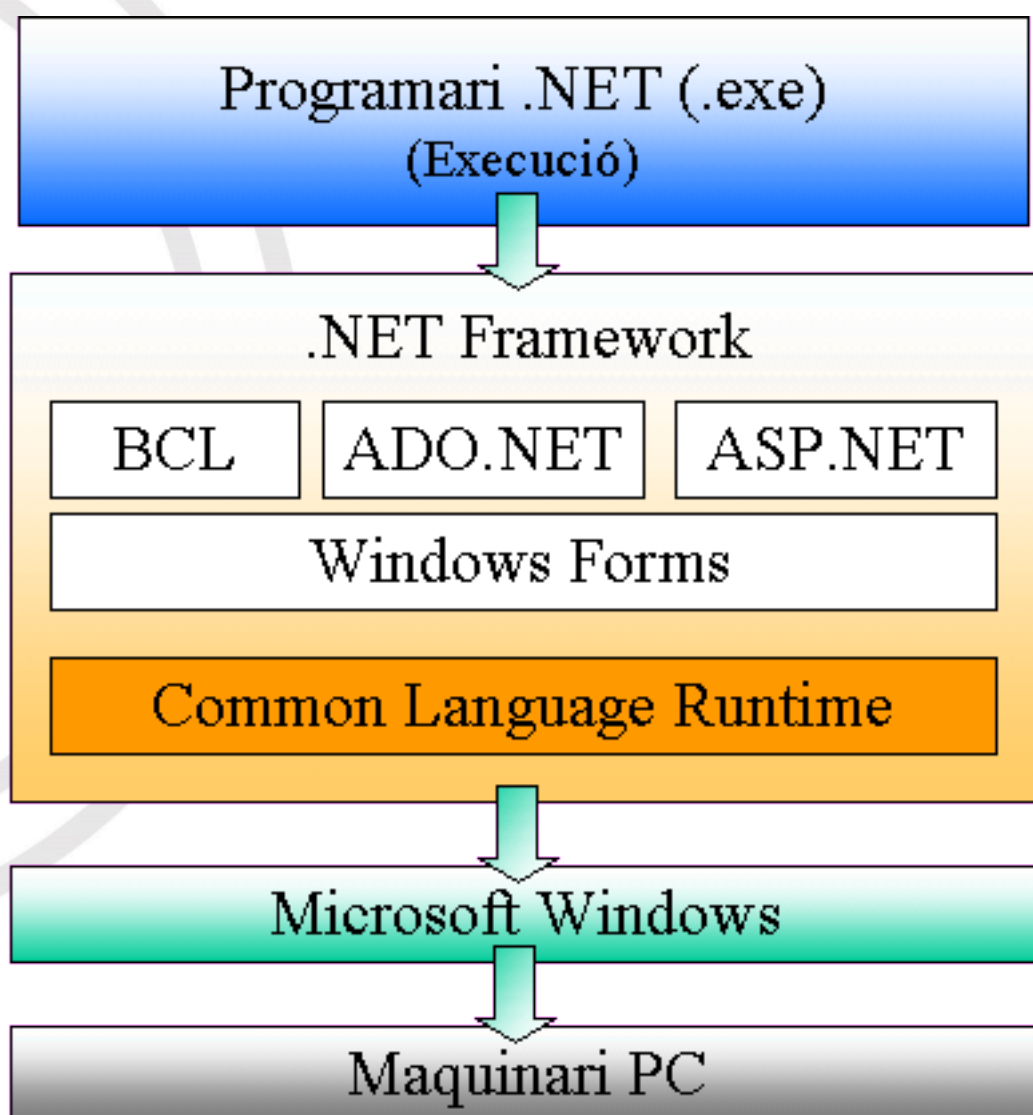
NET Framework se incluye en Windows Server 2008, Windows Vista y Windows 7. De igual manera, la versión actual de dicho componente puede ser instalada en Windows XP, y en la familia de sistemas operativos Windows Server 2003. Una versión "reducida" de .NET Framework está disponible para la plataforma Windows Mobile,(deprecated).

Con .NET se puede desarrollar para mobile con el entorno Xamarin.

Con .NET se puede desarrollar en conjunto con el entorno Unity, juegos, realidad aumentada, etc.

## Estructura de .NET

.NET Framework es un entorno de ejecución que administra aplicaciones cuyo destino es .NET Framework. Incorpora Common Language Runtime, que proporciona la administración de la memoria y otros servicios del sistema, y una biblioteca de clases completa, que permite a los programadores aprovechar el código estable y fiable de todas las áreas principales del desarrollo de aplicaciones.



## Lenguajes Soportados

Debido a la publicación de la norma para la infraestructura común de lenguajes (CLI por sus siglas en inglés), el desarrollo de lenguajes se facilita, por lo que el marco de trabajo .NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje.

Algunos de los lenguajes desarrollados para el marco de trabajo .NET son:

- C#
- Visual Basic .NET
- Delphi (Object Pascal)
- C++
- F#
- J#
- Perl
- Python
- Fortran
- Prolog (existen al menos dos implementaciones, el P#[1] y el Prolog.NET[2])
- Cobol
- PowerBuilder.

## Common Language Runtime (CLR)

El Common Language Runtime o CLR ("entorno en tiempo de ejecución de lenguaje común") es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. El CLR es el **encargado de compilar** una forma de código intermedio llamada Common Intermediate Language (CIL, anteriormente conocido como MSIL, por Microsoft Intermediate Language), al código de máquina nativo, mediante un compilador en tiempo de ejecución

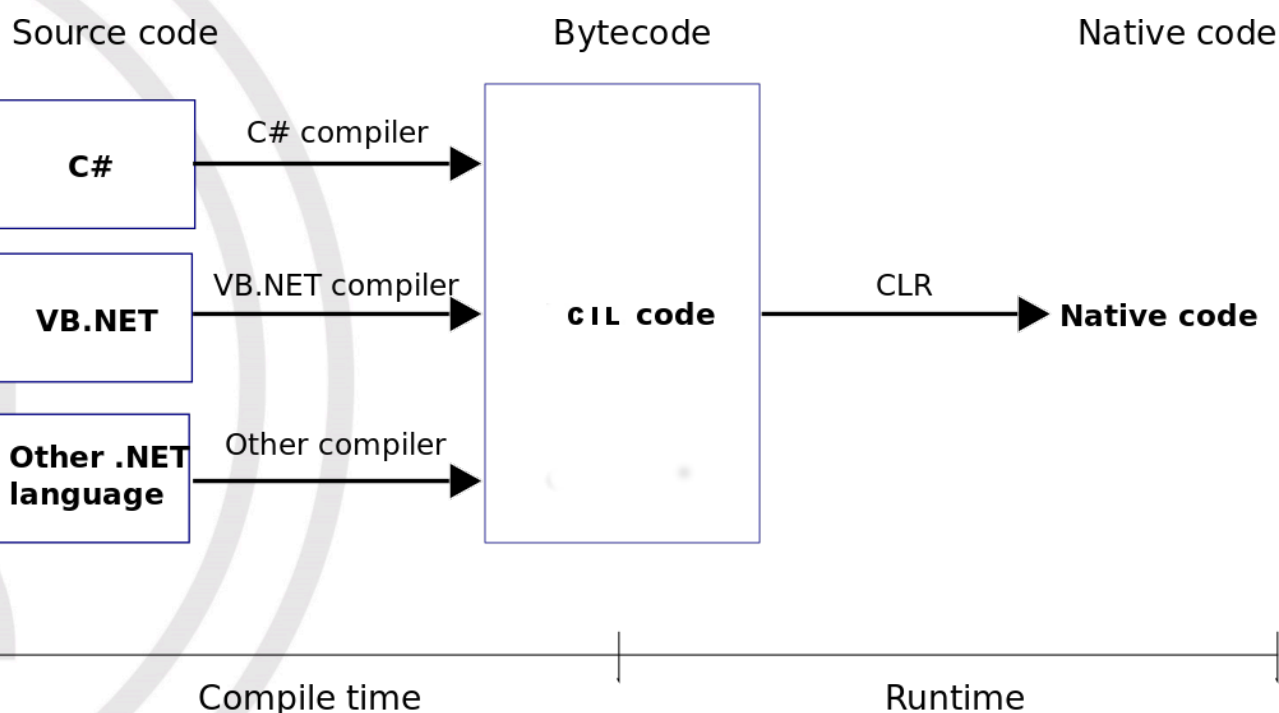
No debe confundirse el CLR con una máquina virtual, ya que una vez que el código está compilado, corre nativamente sin intervención de una capa de abstracción sobre el hardware subyacente. Es una implementación del estándar Common Language Infrastructure (CLI).

- Los desarrolladores que usan CLR escriben el código fuente en un lenguaje compatible con .NET, como C# o Visual Basic .NET.
- En tiempo de compilación, un compilador .NET convierte el código a CIL.
- En tiempo de ejecución, el compilador del CLR convierte el código CIL en código nativo para el sistema operativo.

Alternativamente, el código CIL es compilado a código nativo en un proceso separado anterior a la ejecución. Esto acelera las posteriores ejecuciones del software debido a que la compilación de MSIL a nativo ya no es necesaria.

La manera en que la máquina virtual se relaciona con el CLR permite a los programadores ignorar muchos detalles específicos del microprocesador que estará ejecutando el programa. El CLR también permite otros servicios importantes, incluyendo los siguientes:

- Administración de la memoria
- Administración de hilos
- Manejo de excepciones
- Recolección de basura
- Seguridad



### Lenguaje Común Intermedio (CIL)

Common Intermediate Language (CIL, pronunciado "sil" o "kil") (anteriormente llamado Microsoft Intermediate Language o MSIL) es el lenguaje de programación legible por humanos de más bajo nivel en el Common Language Infrastructure y en el .NET Framework. Los lenguajes del .NET Framework compilan a CIL. CIL es un lenguaje ensamblador orientado a objetos, y está basado en pilas. Es ejecutado por una máquina virtual. Los lenguajes .NET principales son C#, Visual Basic .NET, C++/CLI, y J#.

CIL fue conocido originalmente como Microsoft Intermediate Language (MSIL) durante las versiones de prueba de los lenguajes .NET. Debido a esto, CIL es frecuentemente llamado MSIL. Es posible



ejecutar este lenguaje en plataformas GNU/Linux gracias al Proyecto Mono, que implementa una máquina virtual similar a la de .NET pero de software libre.

Durante la compilación de los lenguajes de programación .NET, el código fuente es convertido a código CIL en lugar de código objeto específico del procesador o de la plataforma. CIL es un conjunto de instrucciones independientes de la plataforma y que puede ser ejecutado en cualquier CPU.

### *Instrucciones CIL*

El código intermedio de CIL incluye un conjunto de instrucciones para las siguientes grupos de tareas:

- Carga y almacenamiento
- Aritméticas
- Conversión de tipos
- Creación y manipulación de objetos
- Operadores de pila (push / pop)
- Transferencia de control (saltos)
- Invocación y retorno de métodos
- Manejo de excepciones
- Concurrencia

### Metadatos

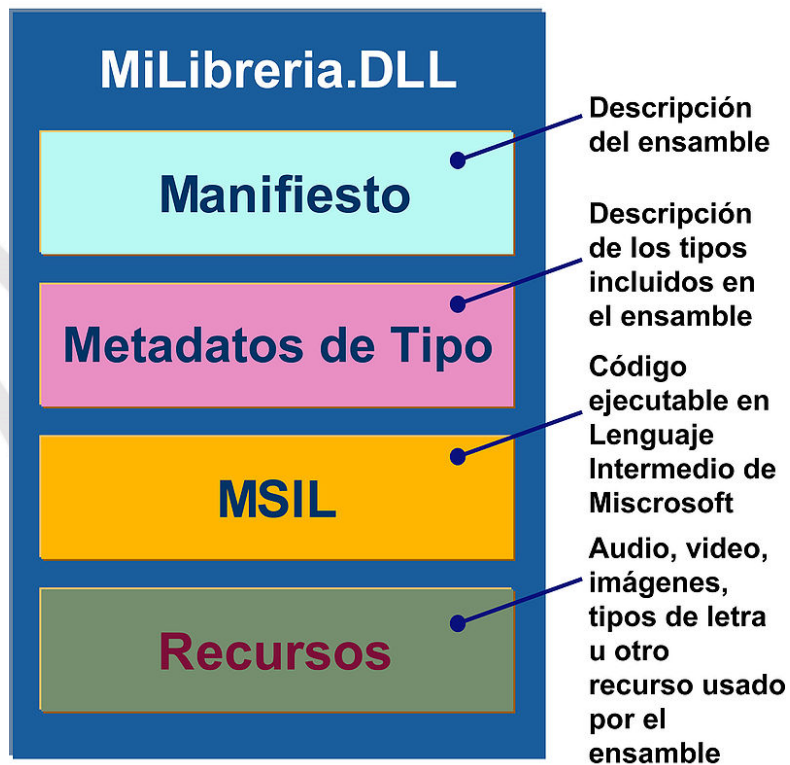
Los metadatos .NET, en la plataforma Microsoft .NET, se refieren a ciertas estructuras de datos embebidas en el código CIL que describen la estructura de alto nivel del código. Los metadatos describen todas las clases y los miembros de clase que se definen en el ensamblado, así como las clases y métodos de clase de otros ensamblados que son invocados por el ensamblado actual. Los metadatos de un método contienen su descripción completa, incluyendo la clase, el ensamblado que contiene la clase, el tipo de retorno y todos los parámetros.

Un compilador de un lenguaje .Net generará los metadatos y los almacenará en el ensamblado que contiene el CIL. Cuando el CLR ejecute el CIL comprobará que los metadatos del método llamado coinciden con los metadatos almacenados en el método llamante. Esto asegura que sólo se pueda invocar el método con el número y tipo de parámetros correctos.

### Ensamblados

Un ensamblado es un compilado EXE o DLL que contiene código CIL que se genera desde los diferentes lenguajes .NET, y que es ejecutado por el CLR.

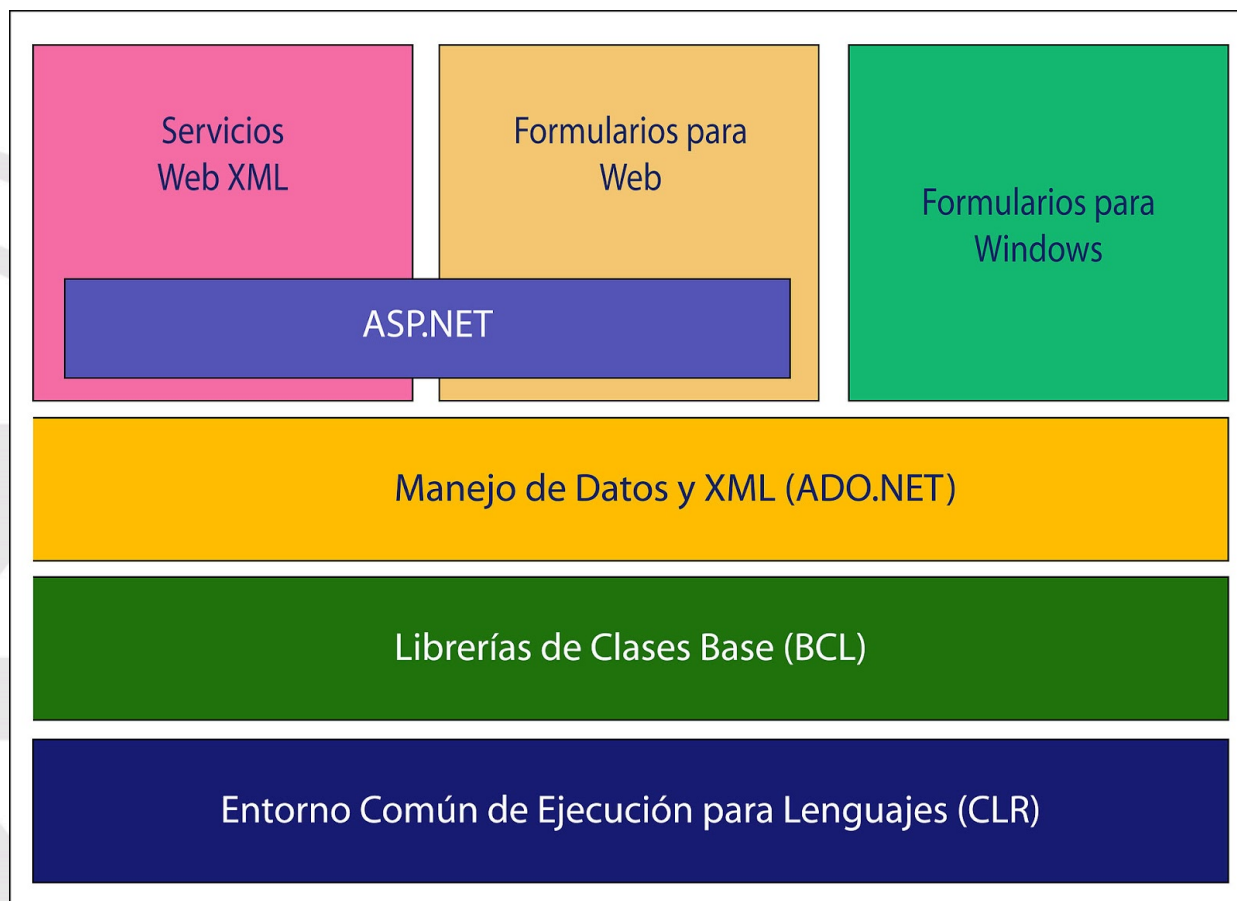
## Ensamble .NET



Un ensamble es una unidad lógica, no física. Puede consistir de uno o más módulos (.dll, .jpg, etc)

### Librería de clase base (BCL)

La biblioteca de clases base de .NET o BCL maneja la mayoría de las operaciones básicas que se encuentran involucradas en el desarrollo de aplicaciones.



*Operaciones a cargo de la BCL:*

- Interacción con los dispositivos periféricos
- Manejo de datos (ADO.NET)
- Administración de memoria
- Cifrado de datos
- Transmisión y recepción de datos por distintos medios (XML, TCP/IP)
- Administración de componentes Web que corren tanto en el servidor como en el cliente (ASP.NET)
- Manejo y administración de excepciones
- Manejo del sistema de ventanas
- Herramientas de despliegue de gráficos (GDI+)
- Herramientas de seguridad e integración con la seguridad del sistema operativo
- Manejo de tipos de datos unificado
- Interacción con otras aplicaciones
- Manejo de cadenas de caracteres y expresiones regulares

- Operaciones aritméticas
- Manipulación de fechas, zonas horarias y periodos de tiempo
- Manejo de arreglos de datos y colecciones
- Manipulación de archivos de imágenes
- Aleatoriedad
- Generación de código
- Manejo de idiomas
- Auto descripción de código
- Interacción con el API Win32 o Windows API.
- Compilación de código

Esta funcionalidad se encuentra organizada por medio de espacios de nombres jerárquicos.

La Biblioteca de Clases Base se clasifica, en cuatro grupos clave:

- ASP.NET y Servicios Web XML
- Windows Forms
- ADO.NET
- .NET

*Solo Windows o Plataforma abierta ?*

La norma (incluido en ECMA-335, ISO/IEC 23271) que define el conjunto de funciones que debe implementar la biblioteca de clases base (BCL por sus siglas en inglés, tal vez el más importante de los componentes de la plataforma), define un conjunto funcional mínimo que debe implementarse para que el marco de trabajo sea soportado por un sistema operativo. Aunque Microsoft implementó esta norma para su sistema operativo Windows, la publicación de la norma abre la posibilidad de que sea implementada para cualquier otro sistema operativo existente o futuro, permitiendo que las aplicaciones corran sobre la plataforma independientemente del sistema operativo para el cual haya sido implementada. El Proyecto Mono emprendido por Ximian pretende realizar la implementación de la norma para varios sistemas operativos adicionales bajo el marco del código abierto.

### Sistema de Tipo Común (CTS)

El sistema de tipos comunes o Common Type System (CTS) es un estándar que especifica cómo las definiciones de los distintos Tipos y valores específicos se representan en la memoria de la computadora.

Este estándar permite compartir un marco de trabajo común a los distintos lenguajes de programación que son soportados por el framework .NET.

*Que es un tipo ?*

Un tipo puede establecerse como un conjunto de valores y las operaciones posibles entre estos valores.

Ejemplo: Todos los enteros entre 0 y 10

Operaciones: Suma y Resta.

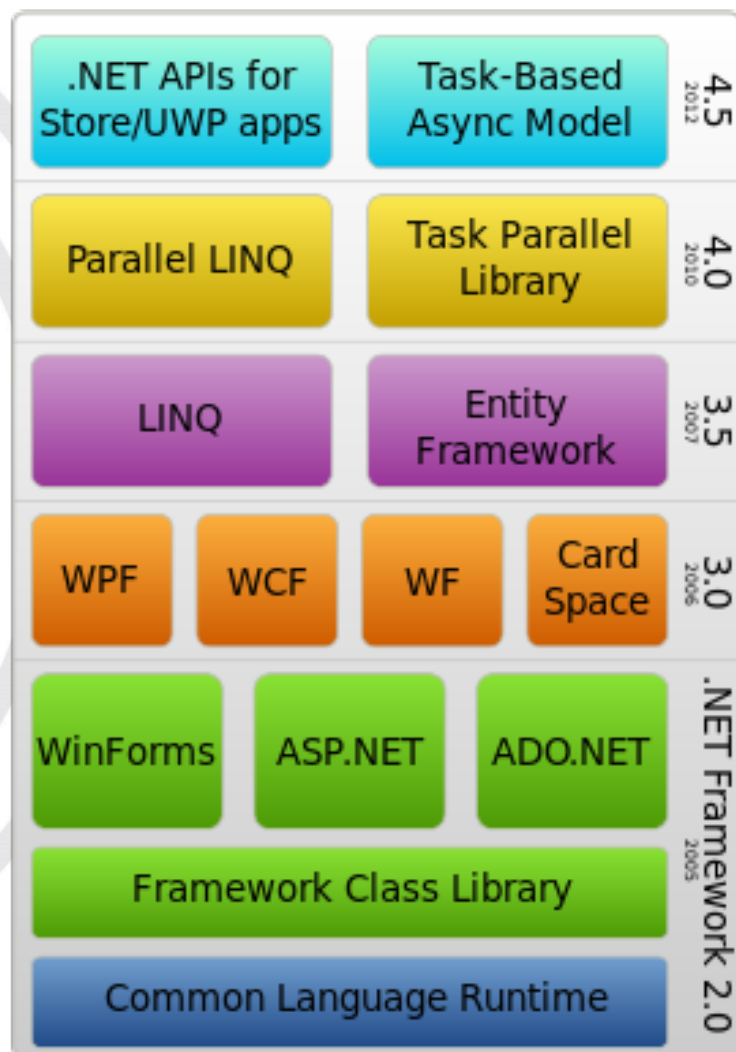
### Especificación Común del Lenguaje (CLS)

Es el encargado de definir los mínimos estándares que deben cumplir los lenguajes y desarrolladores, para que sus componentes y aplicaciones puedan ser utilizados con otros lenguajes compatibles con .NET.

Esto es útil para escribir componentes comunes que van ser utilizados desde diferentes lenguajes disponibles en .NET.



## Versiones y evolución del framework .NET



## BIBLIOGRAFÍA RECOMENDADA:

### OBLIGATORIA

<https://docs.microsoft.com/es-es/dotnet/framework/get-started/>

### COMPLEMENTARIA

<https://www.loginworks.com/blogs clr-work-dot-net-framework/>

[https://es.wikipedia.org/wiki/Microsoft\\_.NET](https://es.wikipedia.org/wiki/Microsoft_.NET)

[https://en.wikipedia.org/wiki/.NET\\_Framework\\_version\\_history](https://en.wikipedia.org/wiki/.NET_Framework_version_history)