

Tema 1

[Descargar estos apuntes](#)

Índice

1. [Índice](#)
2. [Introducción a .NET o dotnet](#)
 1. [.NET Core](#)
 2. [Common Language Runtime \(CLR\) o Runtime a secas](#)
 3. [Common Intermediate Language \(CIL\)](#)
 4. [Ensamblado o Ensamble](#)
 5. [Librerías de clase base BCL](#)
 6. [Sistema Común de Tipos \(CTS\)](#)
3. [Hoja de ruta de .NET](#)

Introducción a .NET o dotnet

- NET Son una serie de tecnologías desarrolladas por Microsoft® en 2002 para interoperabilidad de aplicaciones con la red Internet.
- En sus inicios se ha denominado **.NET Framework** hasta la versión **4.8** aún en uso.
- En la **actualidad** estas tecnologías se han **redefinido** entorno al conocido como [.NET Core](#) teniendo como base.
 - El [código abierto](#) y **Multiplataforma**.
 - Desarrollos basados en Microservicios en la nube, CI/CD y Contenedores.
- Microsoft quiere quitarle el apellido **Core** a la tecnología, a principios de 2021.

.NET Core

- En 2012 Microsoft publicó y estandarizó a través de la [ECMA-335](#) la arquitectura usada en .NET para que terceros pudieran hacer sus propias implementaciones.
- **.NET Core** es la última implementación de esta arquitectura y está basada en los siguientes componentes principales entre otros:
 - Entorno común de ejecución **CLR** (Common Language Runtime)
 - Lenguaje Intermedio Común [CIL](#) (Common Intermediate Language)
 - Biblioteca de clases de .NET Core **BCL** (Base Class Library)

- Sistema común del tipos **CTS** (Common Type System)
 - Especificación común del lenguaje **CLS** (Common Language Specification)

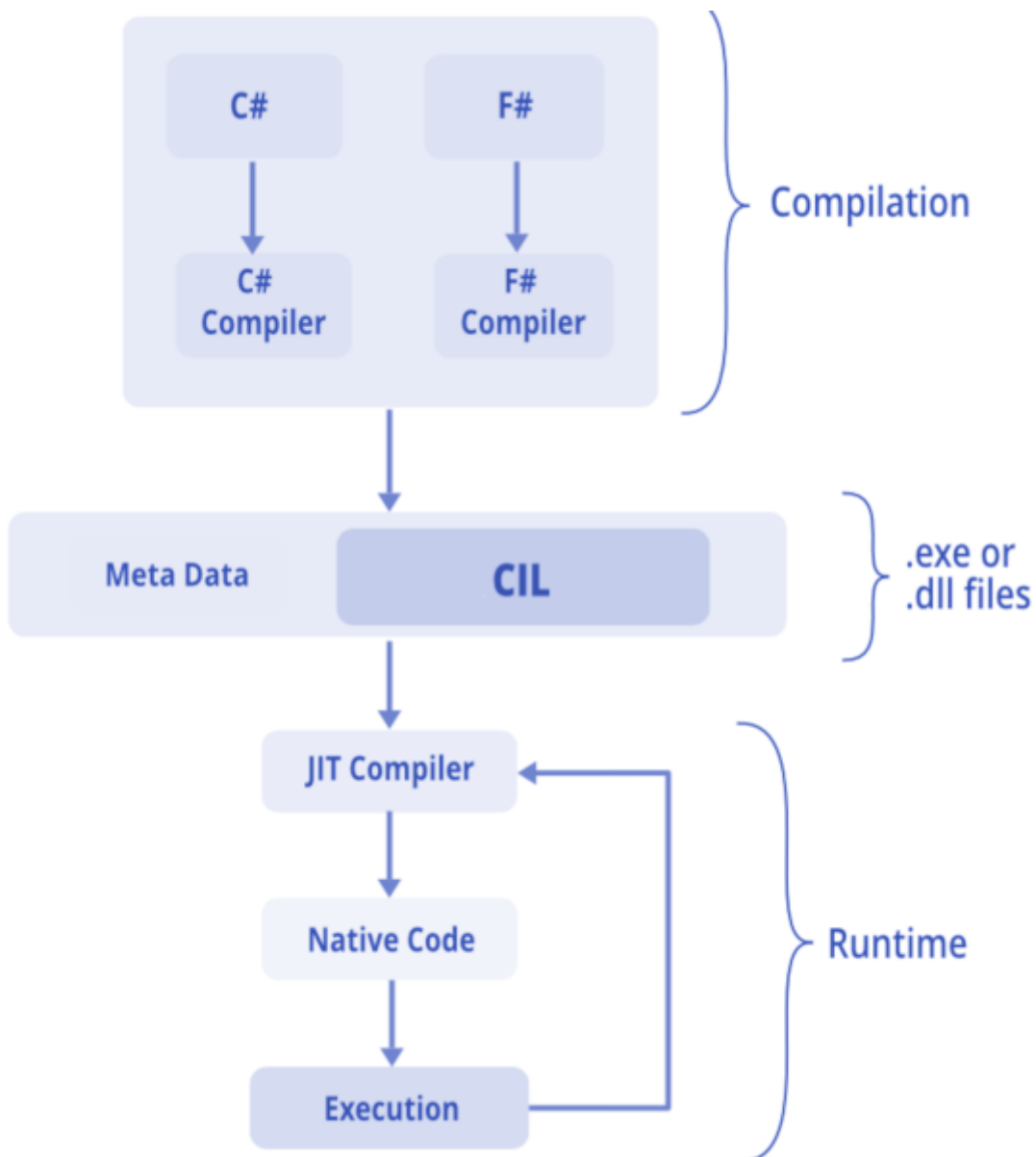
Common Language Runtime (CLR) o Runtime a secas



- Es la capa que hay justo por encima del SO y la arquitectura específica, y **se encarga de gestionar la ejecución de las aplicaciones** en ella.
- Su uso es muy común y existen otros como: JRE de Java, etc...
- Todo esto soluciona una serie de problemas de las aplicaciones tradicionales de la década de los 90, como:
 - Ejecución multiplataforma: **Si tengo un Runtime para la plataforma y arquitectura puedo ejecutar el programa.**
 - Infierno de las DLL's.
 - Integración de lenguajes.
 - Gestión de memoria.

Common Intermediate Language (CIL)

- .NET No genera código máquina para ninguna plataforma concreta.
- Se genera un código especial denominado **IL** que es el que interpreta el CLR.
- En otros 'RunTimes' como JRE se le conoce como **ByteCode**.
- Se encarga de transformarlo a código nativo el **JIT** (Just in time compiler) o jitter que forma parte del CLR.
- Esta compilación, como el nombre indica, se hace dinámicamente o en el momento.



Ensamblado o Ensamble

- Al compilar C# o F# generaremos normalmente **grupo de ficheros con extensión DLL y/o EXE** que denominaremos **ensamblados**.
- Un ensamblado es portable a cualquier SO con el CLR de .NET Core instalado.
- Estará formado por:
 - **Metadatos** (Datos sobre los datos):
 - **Manifiesto** con la descripción del ensamblado y la **firma del mismo**.
 - Información sobre **tipos** incluidos y definidos en el ensamblado.
 - **Programa** con el código ejecutable en **CIL**.
 - **Recursos** agrupados lógicamente como imágenes, vídeo, Tipos de letra, sonido, etc...

Librerías de clase base BCL

- También definidas en la ECMA-332
- Librerías incluidas en el .NET Core SDK para el desarrollo, por defecto, en todos los lenguajes soportador por la plataforma.
- A través de las clases suministradas en ella, es posible desarrollar cualquier tipo de aplicación:
 - Aplicaciones de consola, escritorio, web y móvil devices.
 - Microservicios en la nube, como por ejemplo una API Rest.
 - Comunicaciones en red.
 - etc.
- Además incluyen todo tipo de TAD's.
- Se organizan en **namespaces** o espacios de nombres.
 - Un namespace es la forma que tiene C# de organizar los tipos de datos.
 - Esta organización es jerárquica y en forma de árbol.

Ejemplo de namespaces en las BCL

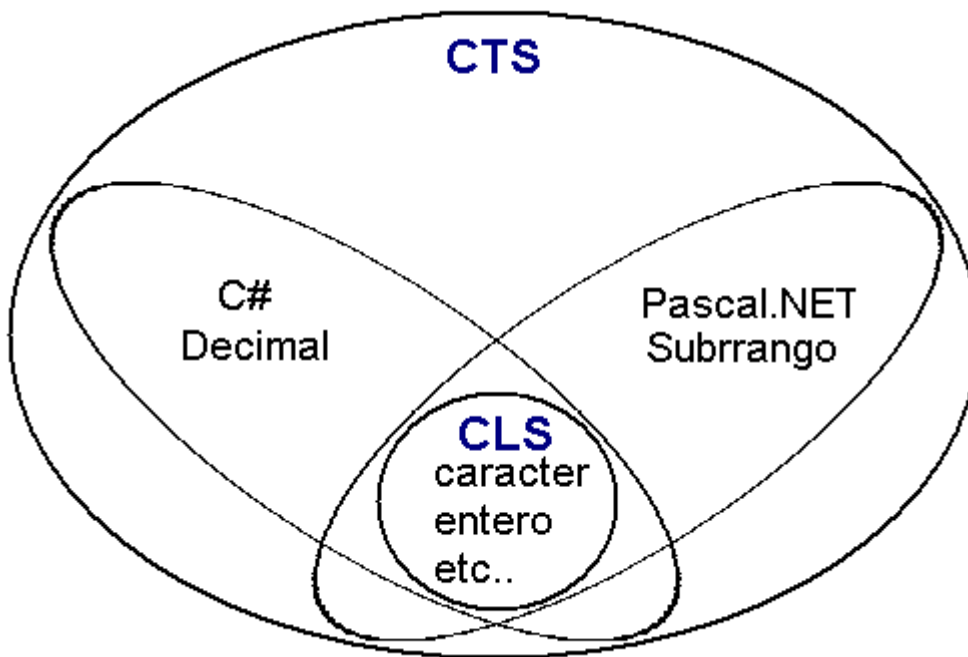
Espacio de nombres	Uso definiciones que contiene
System	Tipos muy frecuentemente usados, como los tipos básicos, tablas, excepciones, fechas, números aleatorios, recolector de basura, entrada/salida en consola, etc.
System.IO	Manipulación de ficheros y otros flujos de datos.
System.Collections	Colecciones de datos de uso común como pilas, colas, listas, diccionarios, etc.

Sistema Común de Tipos (CTS)

- Cada lenguaje gestionado puede tener y definir sus propios tipos de datos que pueden ser diferentes.
- Pero todos ellos deben cumplir unas reglas para que el **CLR** los acepte a través del **CIL** generado.

Especificación Común del Lenguaje (CLS)

- Reglas que tienen que seguir las definiciones de los tipos de un lenguaje gestionado para poder ser accedidos por otro lenguaje gestionado.



Hoja de ruta de .NET

.NET – A unified platform



.NET Schedule

