Licenciatura en Gestión Tenológica
Programación Avanzada 2

Formación Continua

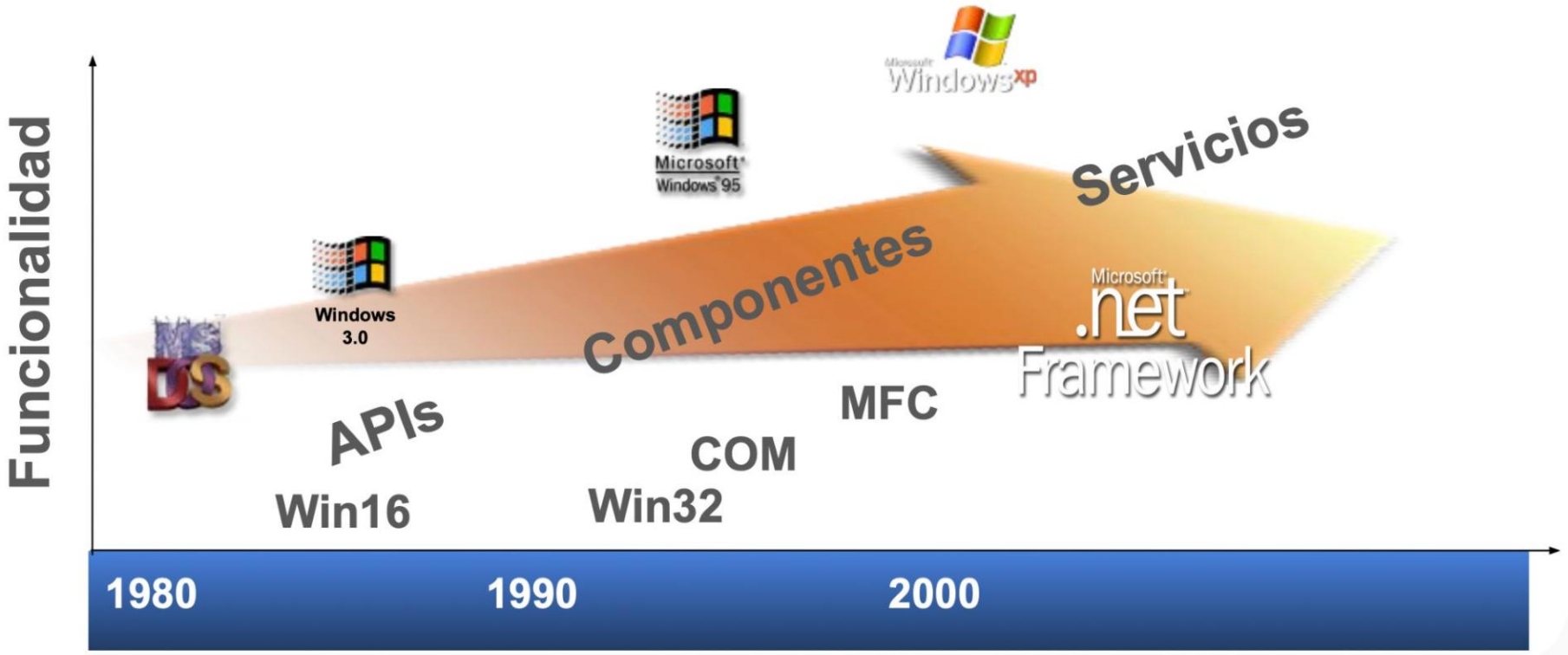
Introducción a .NET - Net Core

Ing. Mariano Juiz

Agenda

1. Introducción a Microsoft .NET
2. Componentes Fundamentales
3. .Net Framework, .Net Core, .Net “X”

Introducción a Microsoft .NET



Introducción a Microsoft .NET

- .NET no es un Sistema Operativo
- .NET no es un Lenguaje de Programación
- .NET no es un Entorno de Desarrollo
- .NET no es un Servidor de Aplicaciones
- .NET no es un producto empaquetado que se pueda comprar como tal

Introducción a Microsoft .NET

¿Qué es .NET ?

- Plataforma de Desarrollo compuesta de:
  - Entorno de Ejecución (Runtime)
  - Bibliotecas de Funcionalidad (Class Library)
  - Lenguajes de Programación
  - Compiladores
  - Herramientas de Desarrollo (IDE & Tools)
  - Guías de Arquitectura

Introducción a Microsoft .NET

.NET como evolución de COM

- Entorno de Ejecución (Runtime)
  - COM: Windows
  - .NET: Common Language Runtime
- Librerías de Funcionalidad
  - COM: Algunas (ADO, FSO, etc.)
  - .NET: Muy extensa (.NET Framework Class Library)
- Lenguajes de Programación
  - COM: VB, C++, VFP, ASP, J++
  - .NET: Common Language Specification
- Entorno de Desarrollo (IDE)
  - COM: Uno para cada lenguaje
  - .NET: Uno independiente del lenguaje (VS.NET)

Introducción a Microsoft .NET

Características de .NET

- Plataforma de ejecución intermedia.
- 100% Orientada a Objetos.
- Multilenguaje.
- Plataforma Empresarial de Misión Crítica.
- Modelo de Programación único para todo tipo de aplicaciones y dispositivos de hardware.
- Se integra fácilmente con aplicaciones existentes desarrolladas en plataformas Microsoft y en otras plataformas.
- Gestión automática de la memoria

Introducción a Microsoft .NET

Características de .NET

Entorno de ejecución Robusto y Seguro

- Manejo de Excepciones
- Fuertemente tipado
  - Solo casteos seguros
  - Inicialización de variables obligatoria
- Instalación con Cero Impacto
  - No requiere registración en la Registry
- Independencia del Lenguaje de Programación
- Múltiples Herramientas para el Desarrollador (Debug, etc.)

Introducción a Microsoft .NET

Independencia del lenguaje

- Libertad en la elección del lenguaje
  - Todas las facilidades de la plataforma .NET están disponibles a todos los lenguajes de programación .NET
  - Los componentes de una aplicación .NET pueden ser escritos en distintos lenguajes de alto nivel compatibles con la plataforma
- Herramientas compartidas
  - Debuggers, profilers, analizadores de código, y otras trabajan para todos los lenguajes

Introducción a Microsoft .NET

Interoperabilidad

Servicios Web
XML

.NET Framework

C#

C++

VB

Otros...

COM
INTEROP

COM

NET

Introducción a Microsoft .NET

.NET Framework

- Paquete de software fundamental de la plataforma .NET.
  Incluye:
  - Entorno de Ejecución (Runtime)
  - Bibliotecas de Funcionalidad (Class Library)
- Se distribuye en forma libre y gratuita
- Existen tres variantes principales:
  - .NET Framework Redistributable Package
  - .NET Framework SDK
  - .NET Compact Framework
- Está instalado por defecto en Windows 2003 Server o superior

Introducción a Microsoft .NET

¿Donde instalar el .NET Framework?

	Cliente	Servidor
Aplicación de Escritorio	✔	✔*
Aplicación Web		✔
Aplicación de Consola	✔	✔*
Aplicación Móvil	Xamarin	

* Solo si la aplicación es distribuida

Introducción a Microsoft .NET

Evolución

Año 2002: VS 2002, .NET Framework 1.0 y ASP.NET 1.0

Año 2003: VS 2003, .NET Framework 1.1 y ASP.NET 1.1

Año 2005: VS 2005, .NET Framework 2.0 y ASP.NET 2.0

Año 2008: VS 2008, .NET Framework 3.5 y ASP.NET 3.5

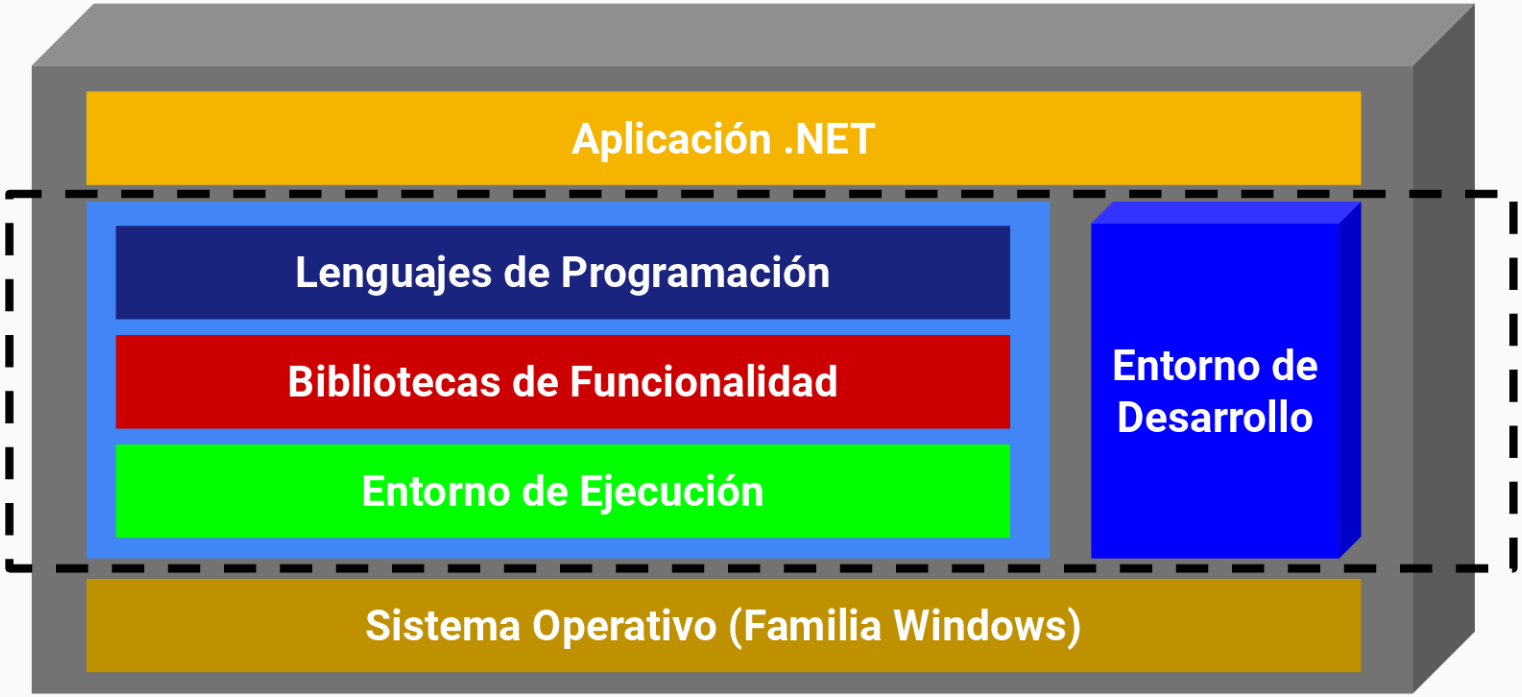
Año 2010: VS 2010, .NET Framework 4.0 y ASP.NET 4.0

Año 2012: VS 2012, .NET Framework 4.5 y ASP.NET 4.5

> Año 2015: VS 2015, VS 2017, Net Framework 4.6, 4.7, 4.8. ASP.NET 5.

Componentes Fundamentales

Plataforma de Ejecución Intermedia



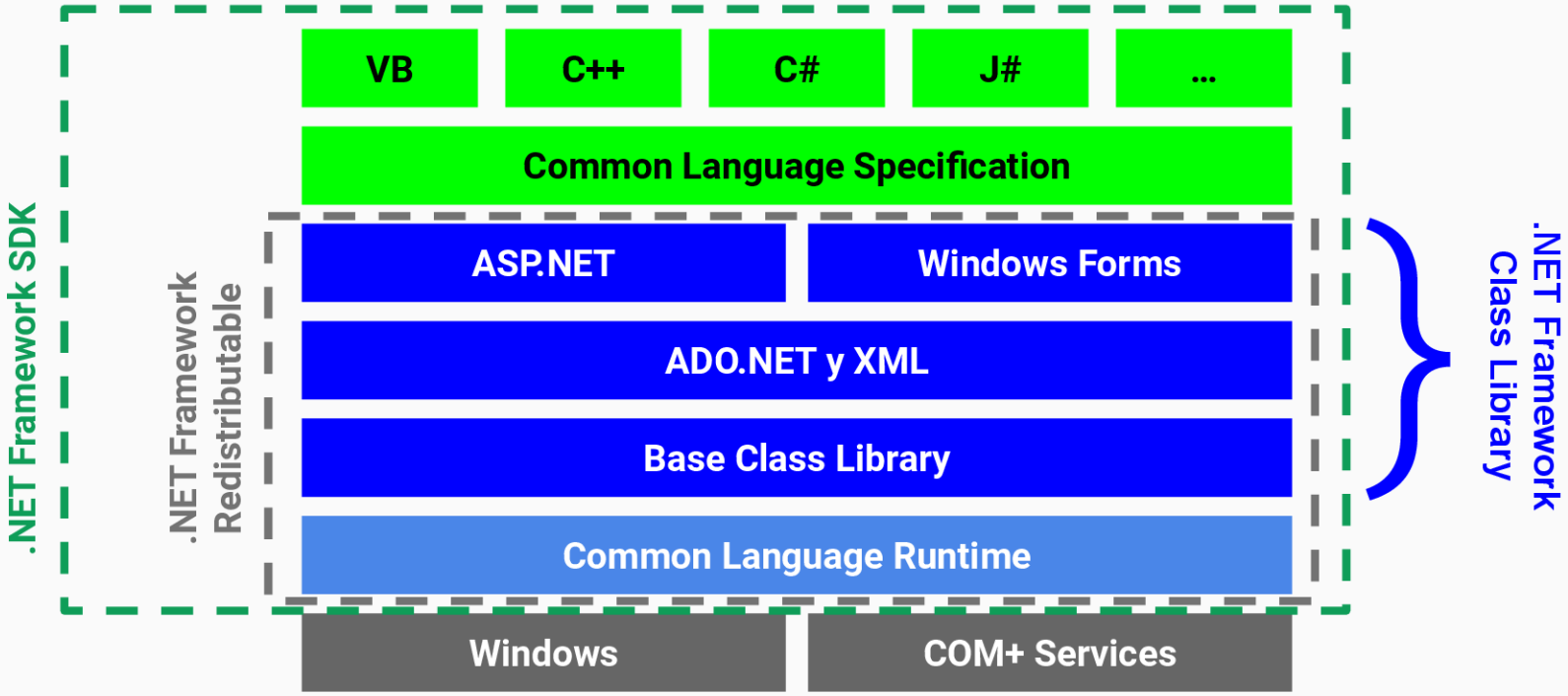
Componentes Fundamentales

Componentes Fundamentales

- Arquitectura
- Common Language Runtime (CLR)
- Common Language Specification (CLS)
- Assemblies (*.EXE, *.DLL)
- Microsoft Intermediate Language (MSIL)
- .NET Class Library

Componentes Fundamentales

Arquitectura del .NET Framework



Componentes Fundamentales

CLR - Arquitecturas de Ejecución de Aplicaciones

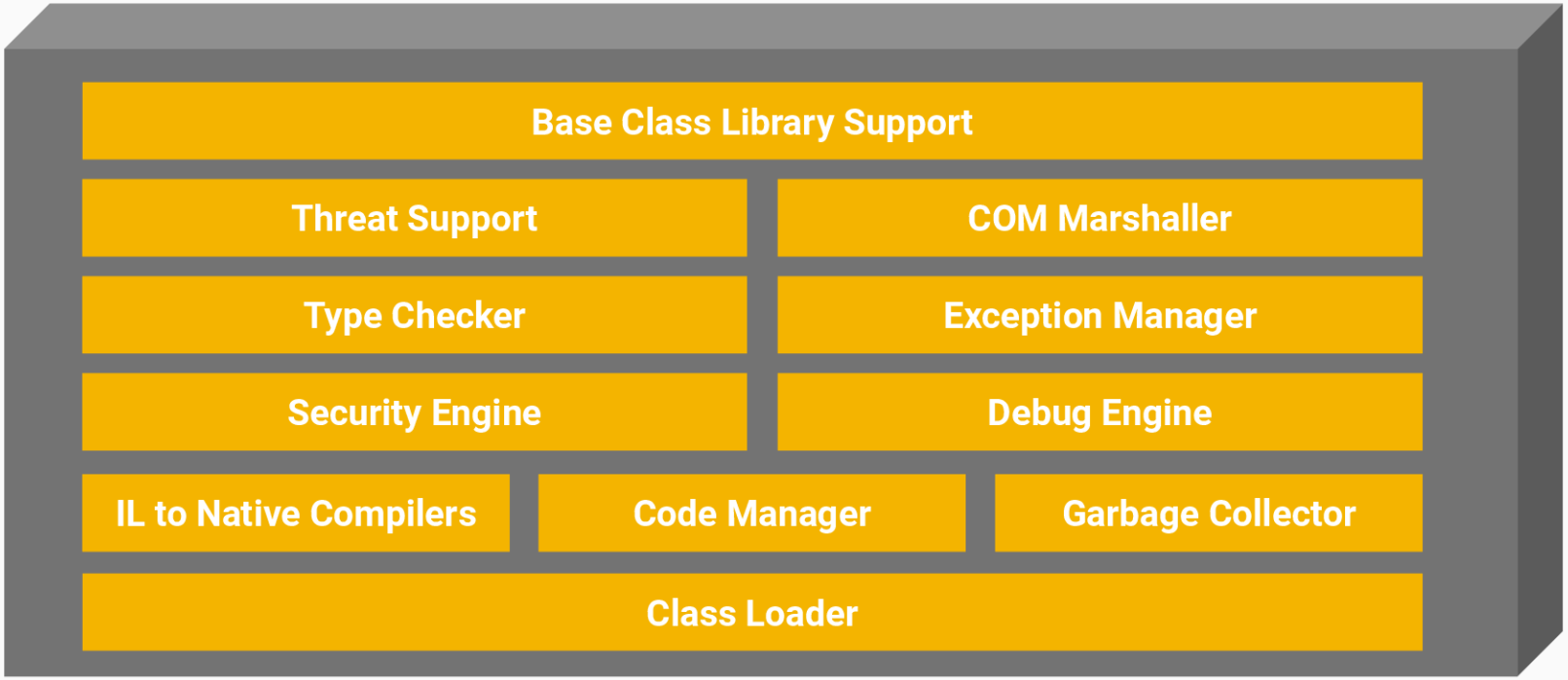
El CLR es el motor de ejecución (runtime) de .NET

Características:

- Compilación Just-In-Time (JIT)
- Gestión automática de memoria (Garbage Collector)
- Gestión de errores consistente (Excepciones)
- Ejecución basada en componentes (Assemblies)
- Gestión de Seguridad
- Multithreading

Componentes Fundamentales

CLR – Componentes Internos



Componentes Fundamentales

Funcionamiento Interno del CLR

- Especificación CLI
- Modelo de Ejecución
- Common Type System

Componentes Fundamentales

Especificación Common Language Infraestructure (CLI)

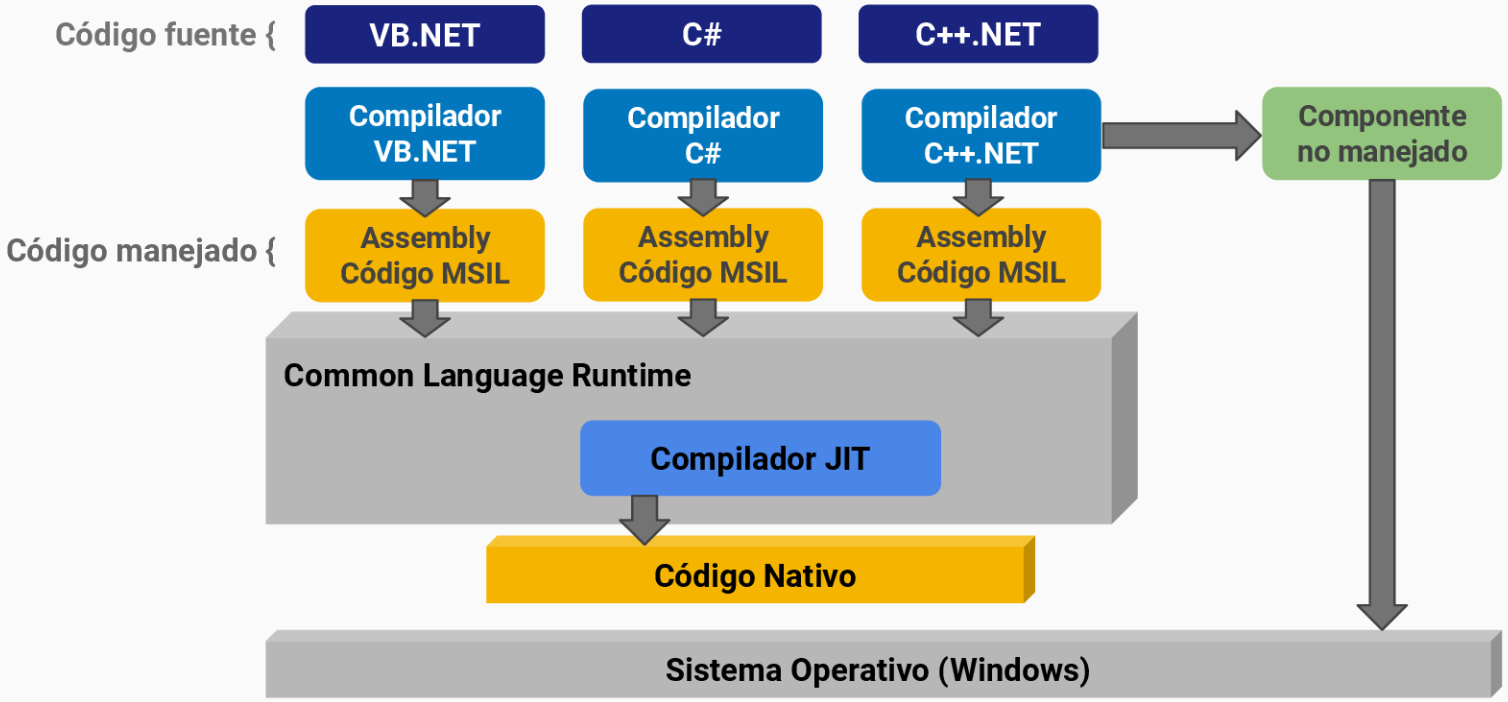
- Especificación patrocinada por Microsoft, Intel, HP y estandarizada por ECMA (2001) e ISO (2003) que describe:
  - Entorno Virtual de Ejecución de Aplicaciones
    - Permite Ejecutarse en Múltiples Arquitecturas de HW y SW
  - Conjunto de Librerías Básicas (BCL)
  - Tipos de Datos Comunes (CTS)
- El .NET Framework y el .NET Compact Framework son implementaciones de la especificación CLI

Componentes Fundamentales

CLR – Componentes Internos

Componentes Fundamentales

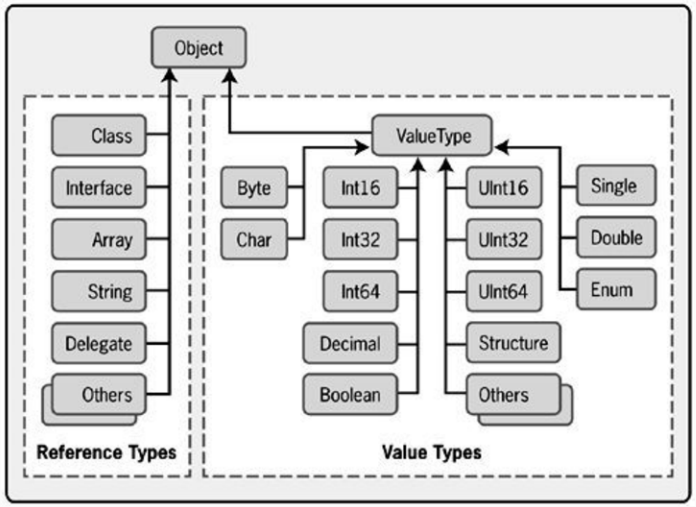
Modelo de Ejecución del CLR



Componentes Fundamentales

Common Type System (CTS)

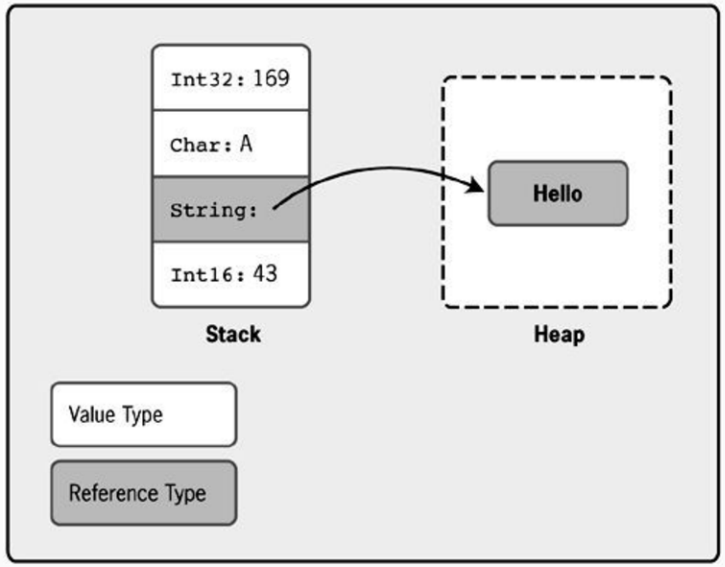
- Define un conjunto común de “tipos” de datos orientados a objetos
- Todo lenguaje de programación .NET debe implementar los tipos definidos por el CTS
- Todo tipo hereda directa o indirectamente del tipo System.Object
- Define Tipos de VALOR y de REFERENCIA



Componentes Fundamentales

La Memoria y los Tipos de Datos

- El CLR administra dos segmentos de memoria: Stack (Pila) y Heap (Montón)
- El Stack es liberado automáticamente y el Heap es administrado por el GC (Garbage Collector). No se sabe cuando liberará la memoria.
- Los tipos VALOR (enteros, decimales, etc.) se almacenan en el Stack
- Los tipos REFERENCIA (String, objetos, etc.) se almacenan en el Heap



Componentes Fundamentales

Common Language Specification (CLS)

- Especificación que estandariza una serie de características soportadas por el CLR
- Contrato entre diseñadores de lenguajes de programación y autores de bibliotecas
- Permite la interoperabilidad entre lenguajes
- Microsoft provee implementaciones de 4 lenguajes, todos compatibles con CLS
  - Microsoft Visual Basic .NET
  - Microsoft Visual C# .NET
  - Microsoft Visual J#.NET
  - Microsoft Visual C++.NET

Componentes Fundamentales

Common Language Specification (CLS)

El resto de la industria y el sector académico han desarrollado más de 20 lenguajes compatibles con la especificación CLS

C++.NET C# J#
Visual Basic.NET
Delphi Java PHP Perl Python JavaScript
Pascal Haskell LISP Prolog RPG
Oberon Mondrian Smalltalk Eiffel ML Scheme
Cobol Fortran APL Objective Calm Mercury

Componentes Fundamentales

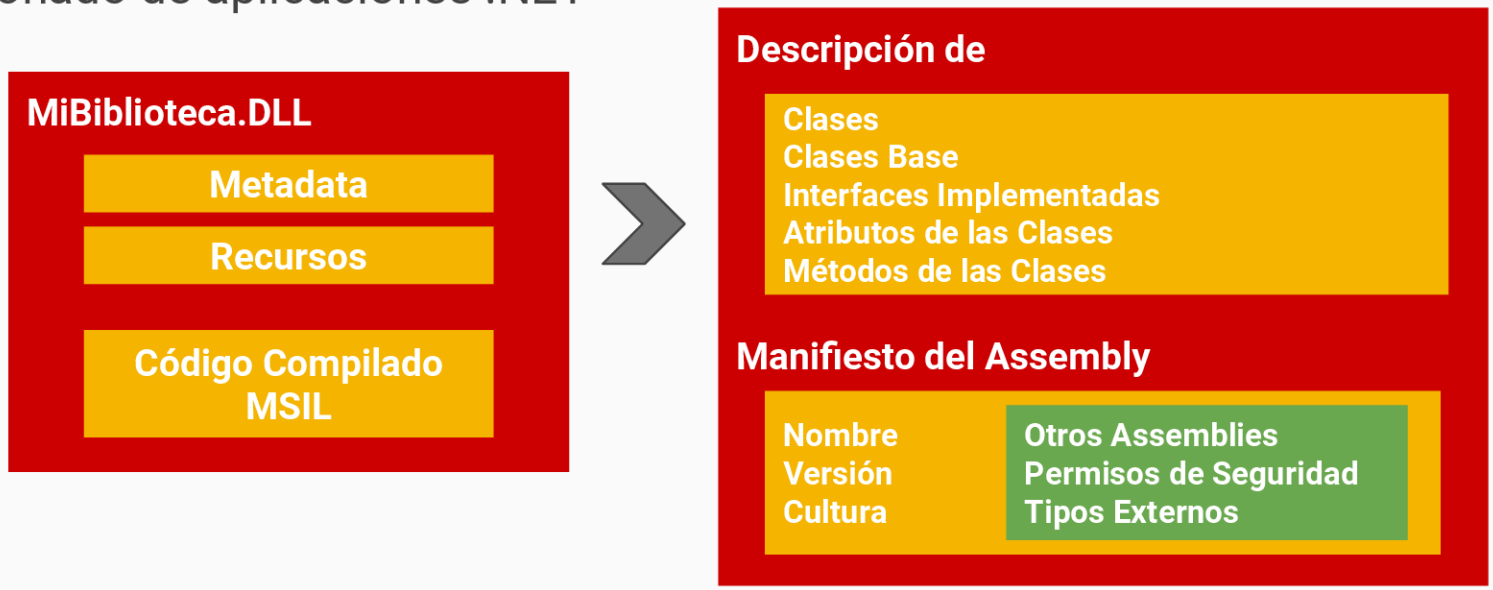
Elección del lenguaje

- .NET posee un único runtime (el CLR) y un único conjunto de bibliotecas para todos los lenguajes
- No hay diferencias notorias de performance entre los lenguajes provistos por Microsoft
- El lenguaje a utilizar, en gral., dependerá de su experiencia previa con otros lenguajes o de gustos personales
  - Si conoce Java, Delphi, C++, etc. □ C#
  - Si conoce Visual Basic o VBScript □ VB.NET
- Los tipos de aplicaciones .NET son INDEPENDIENTES del lenguaje que elija

Componentes Fundamentales

¿Qué es un Assembly?

Un Assembly es la unidad mínima de ejecución, distribución, instalación y versionado de aplicaciones .NET



Componentes Fundamentales

¿Qué es un Assembly?

- Una aplicación puede generar uno o más Assemblies
- Al ejecutar una aplicación:
  - El Class Loader busca en el directorio local (preferido)
  - Global Assembly Cache (GAC)
- Diferentes aplicaciones pueden usar diferentes versiones
  - Actualizaciones más simples
  - Desinstalación más simples

Componentes Fundamentales

Proceso de Compilación

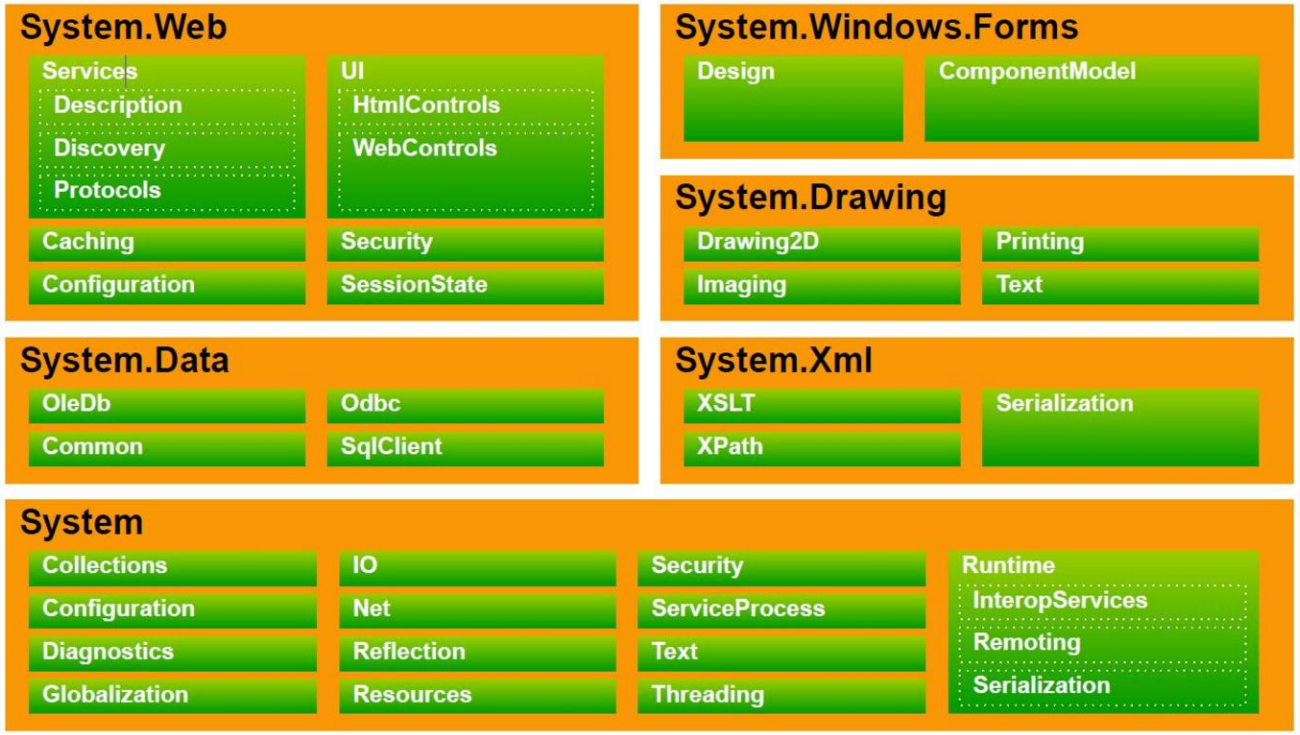
Componentes Fundamentales

MSIL Generado

```
.method private hidebysig static void Main(string[] args) cil
managed {
.entrypoint
maxstack 8
L_0000: ldstr "Hola Mundo"
L_0005: call void [mscorlib]System.Console::WriteLine(string)
L_000a: ret
}
```

Componentes Fundamentales

.Net Framework Class Library

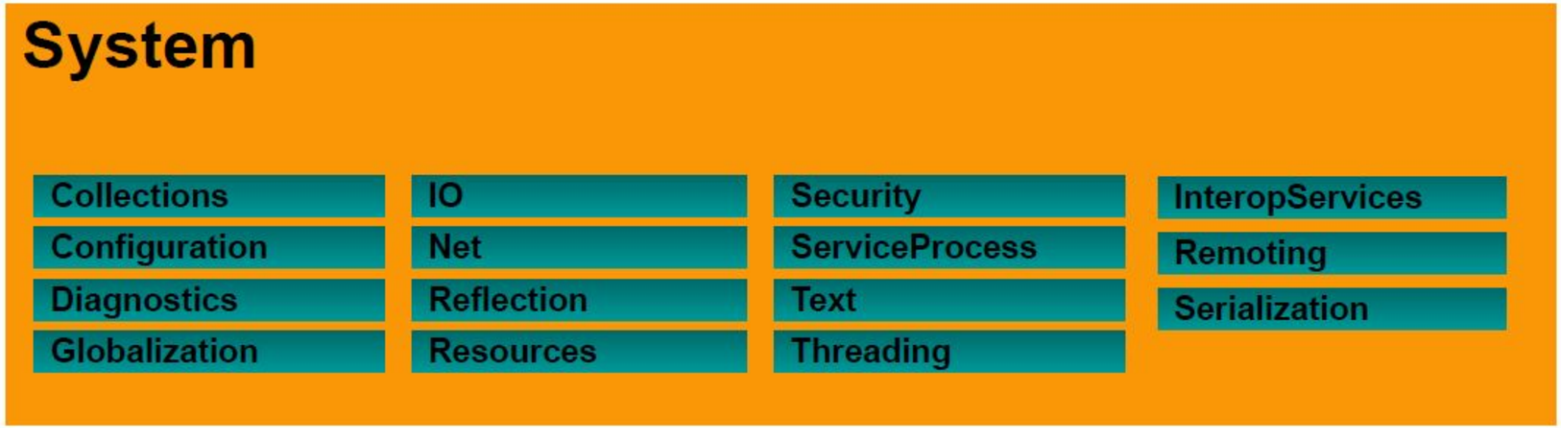


Componentes Fundamentales

Biblioteca Principal

Base Class Library (BCL)

Provee la mayor parte de las funcionalidades elementales que pueden necesitarse para construir una Aplicación o Servicio



.Net Core

.Net Core es un framework Open Source, para construir aplicaciones multiplataforma y de alto rendimiento.

Versiones:

- Net Core 1.0 (2016)
- Net Core 2.0 (2017)
- Net Core 3.0 (2019)
- Net Core 3.1 (2019)

.NET Core es multiplataforma, es decir, este se ejecuta sobre entornos Windows, OS X y varias distribuciones de Linux. También soporta diferentes arquitecturas de CPU.

Si bien tienen similitudes, no es una nueva versión escrita sobre la base de .Net Framework Tradicional; por lo contrario fue escrito desde Cero.

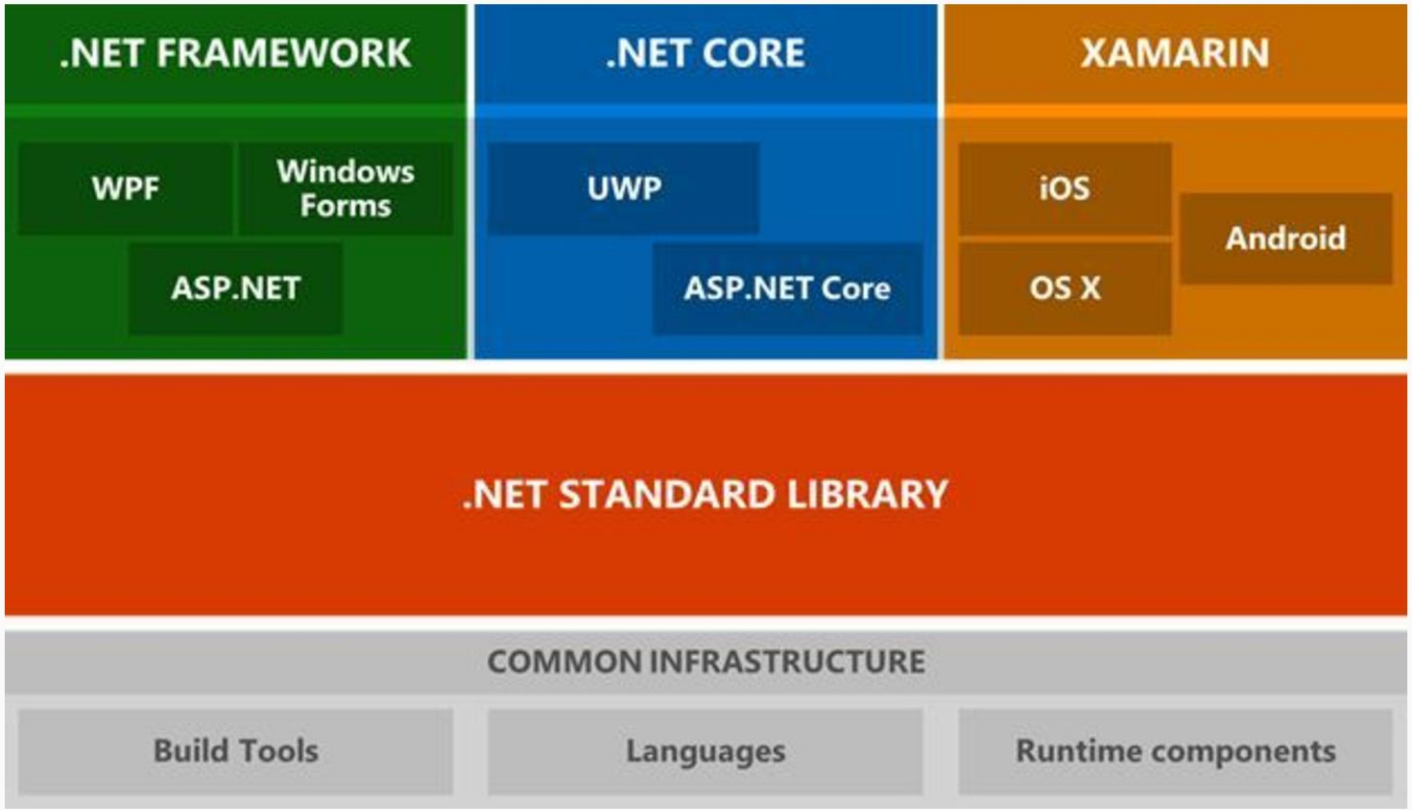
.Net Standard

Es una especificación formal de las API de .NET que están disponibles en varias implementaciones de .NET. La motivación detrás de .NET Standard fue establecer una mayor uniformidad en el ecosistema .NET

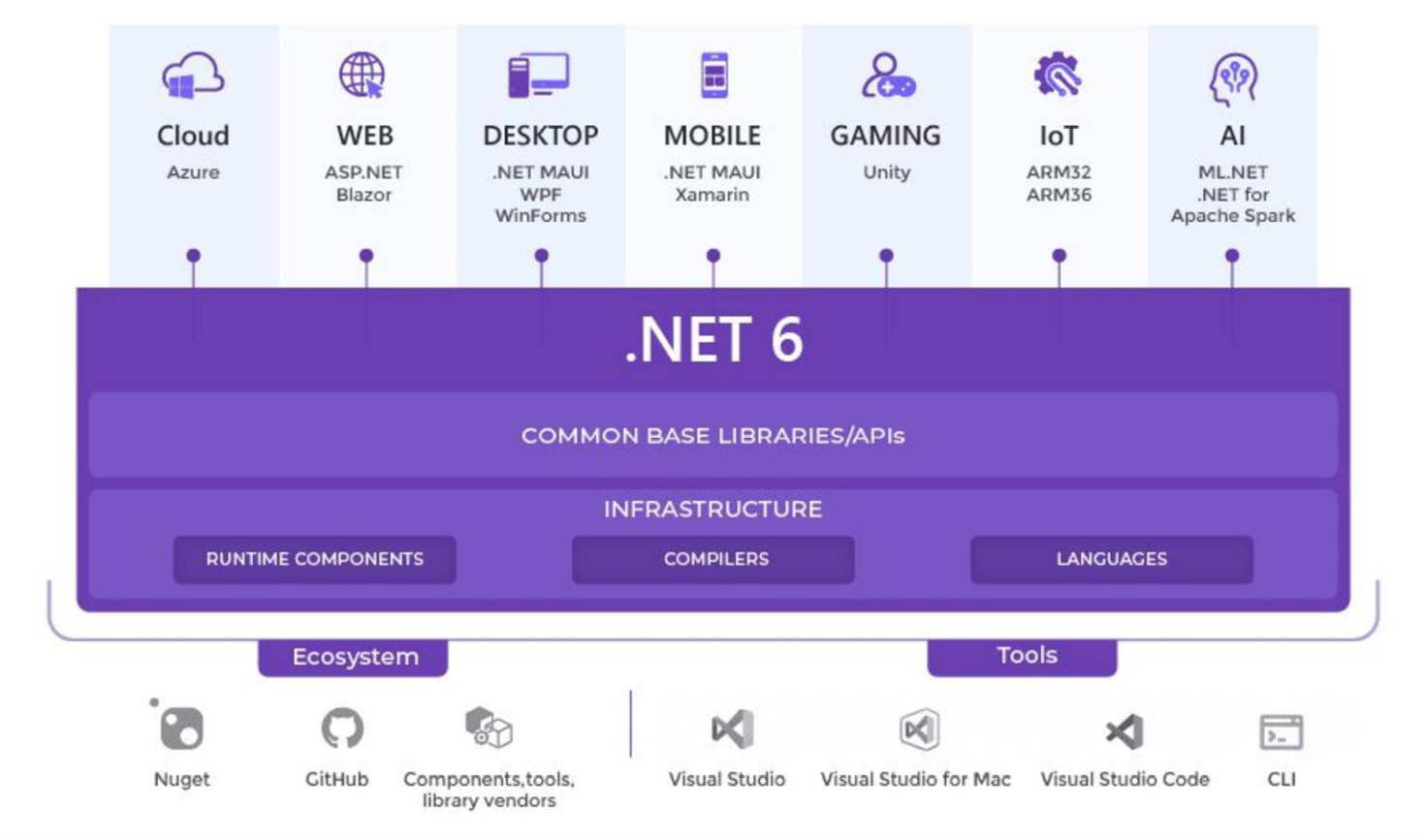
Una biblioteca .NET Standard es una especificación de APIs, es decir es súper fácil saber si un API es compatible con .NET Standard y en qué versión (https://apisof.net).

Con Net Standard. Todas las implementaciones de .NET (.NET Framework, Xamarin, .NET Core) pueden tener una misma base sobre la cual trabajar.

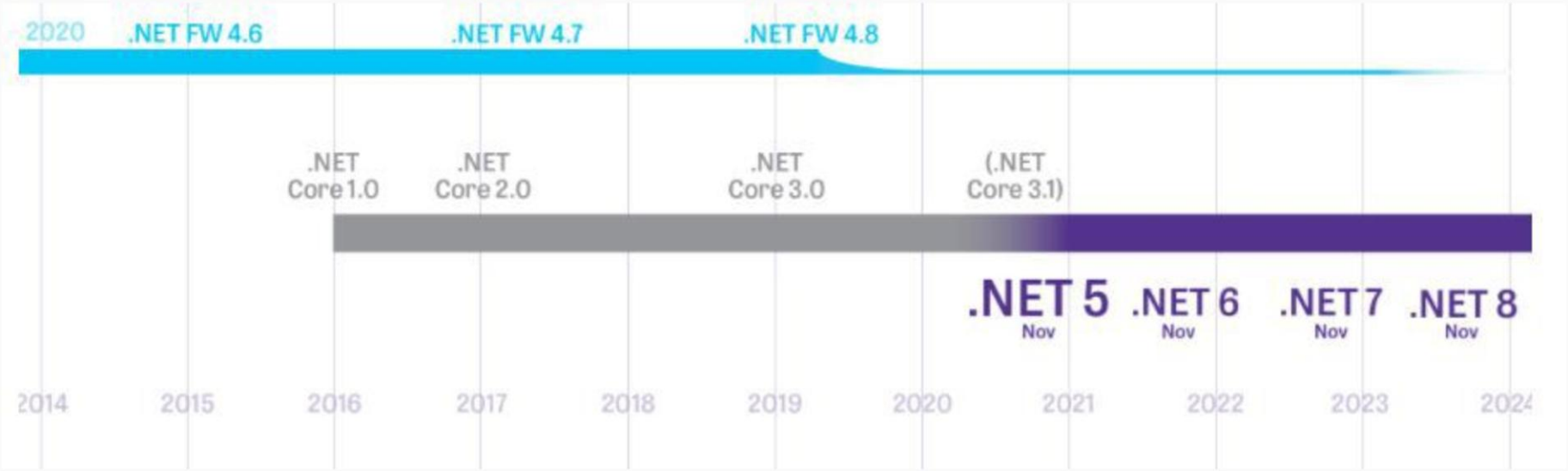
.Net Core vs .Net Framework



.Net Framework - .Net Core - .Net "X"



.Net Framework - .Net Core - .Net “X”



MUCHAS GRACIAS