

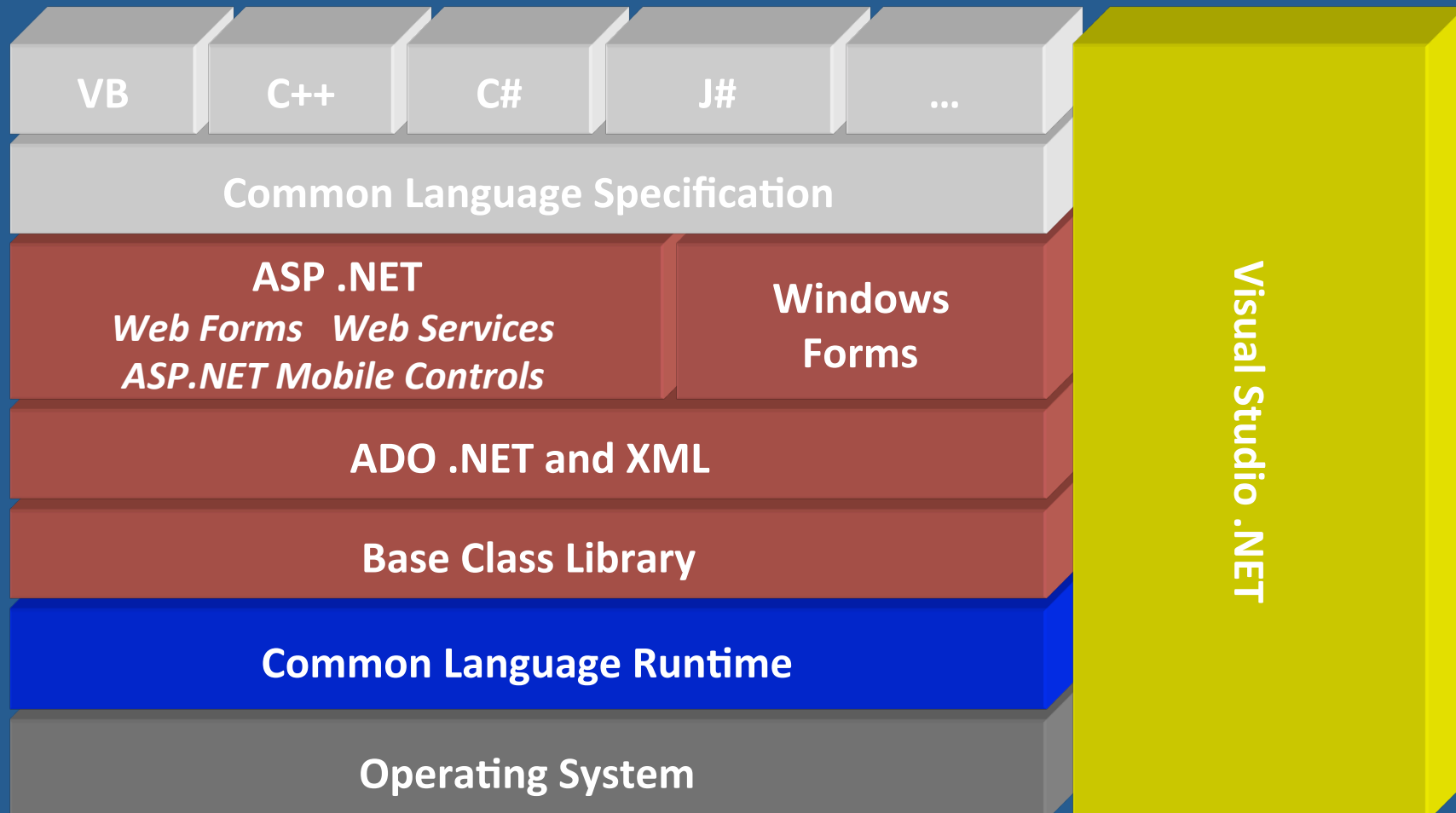
# PROGRAMACIÓN ORIENTADA A OBJETOS



## Framework .NET

Titular: Dario Guillermo Cardacci

## .NET Framework



## ¿Qué problemas resuelve .NET?

- Desde Internet, muchas aplicaciones y dispositivos están fuertemente comunicados entre sí.
- Los programadores escribían arquitectura en lugar de aplicaciones.
- Los programadores tenían conocimientos limitados o debían aprender nuevos lenguajes.

## **.NET Framework**

- El .NET Framework constituye las bases sobre las que, tanto aplicaciones como servicios, son ejecutadas y construidas.
- La naturaleza unificada del .NET Framework permite que cualquier tipo de aplicación sea desarrollada mediante herramientas comunes haciendo la integración mucho más simple.
- El .NET Framework esta compuesto de:

El CLR (Common Language Runtime)

La BCL (Base Class Library)

## CLR – Common Language Runtime

- El CLR es el motor de ejecución (runtime) del .NET Framework.
- Ofrece servicios automáticos tales como:
  - Administración de la memoria
  - Seguridad del código, asegurando:
  - Conversión de tipos
  - Inicialización de variables
  - Indexación de arreglos fuera de sus límites
  - Versionamiento

## Componentes de CLR

**Entorno Común de Ejecución para Lenguajes (CLR)**

**Soporte a las Librerías de Clases Base**

**Manejo de hilos**

**Interacción con COM**

**Verificador de Tipos**

**Controlador de Excepciones**

**Motor de Seguridad**

**Motor de depuración**

**Compilador de MSIL  
a código nativo (JIT)**

**Controlador  
de código**

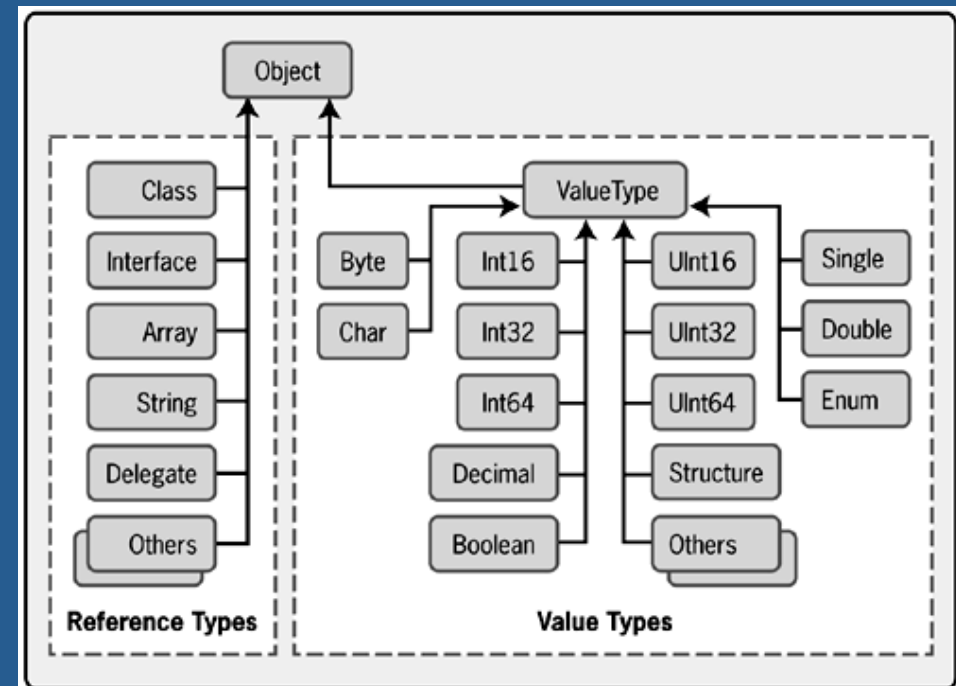
**Recolector de  
Basura (GC)**

**Cargador de Clases**

**Sistema Operativo**

## CTS (Common Type System)

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



## Microsoft Intermediate Language (MSIL)

- El compilador convierte al código soportado por .NET en lenguaje intermedio.
- Es un conjunto de instrucciones independientes de la CPU, que se convierten en código nativo al ejecutarse.
- Para convertir MSIL a código nativo, se utilizan compiladores llamados “Just In Time” JIT
- Los archivos ejecutables estan conformados por
  - MSIL
  - Datos Adicionales (Metadata)
- El MSIL es independiente del lenguaje en el que se desarrolla



## Conclusiones

- Costo inicial de aprendizaje para el uso.
- Facilita el desarrollo.
- Facilita el mantenimiento.
- Arduo trabajo de elección.
- Adecuados para productos cuyos requisitos cambian rápidamente.
- Documentación



FIN