Los Fundamentos de la Programación y el Entorno de Desarrollo

UNIDAD N° 1

(Parte 1/2)

2023

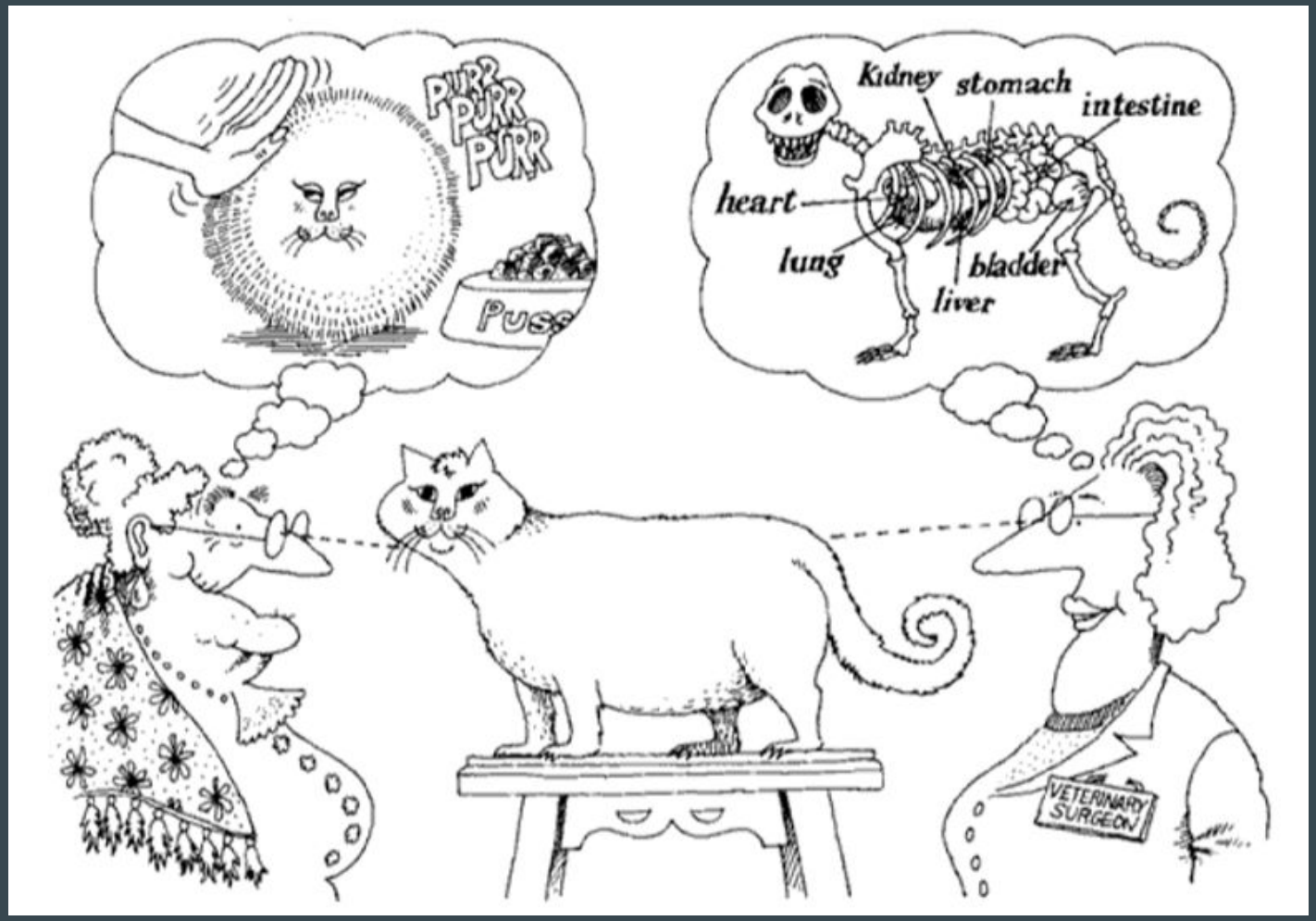
Contenido

- Revisión de conceptos de POO
- Introducción a la Plataforma
- El Entorno de Desarrollo Integrado (IDE)
- Tiempo de diseño y ejecución
- Depuración de código

POO - Elementos Fundamentales

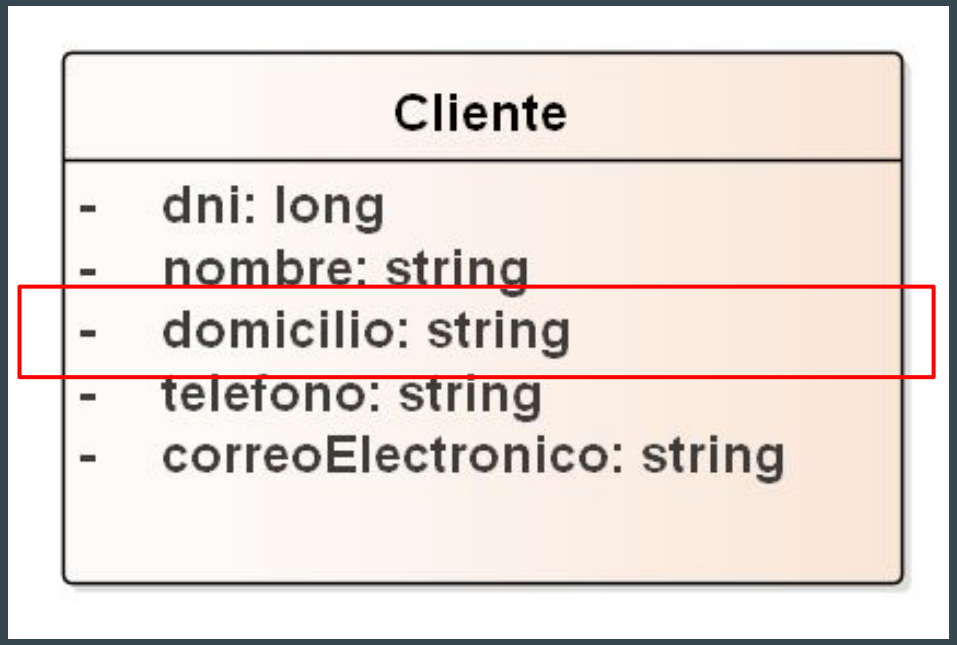
- Abstracción
- Encapsulamiento
- Modularidad
- Jerarquía

Abstracción



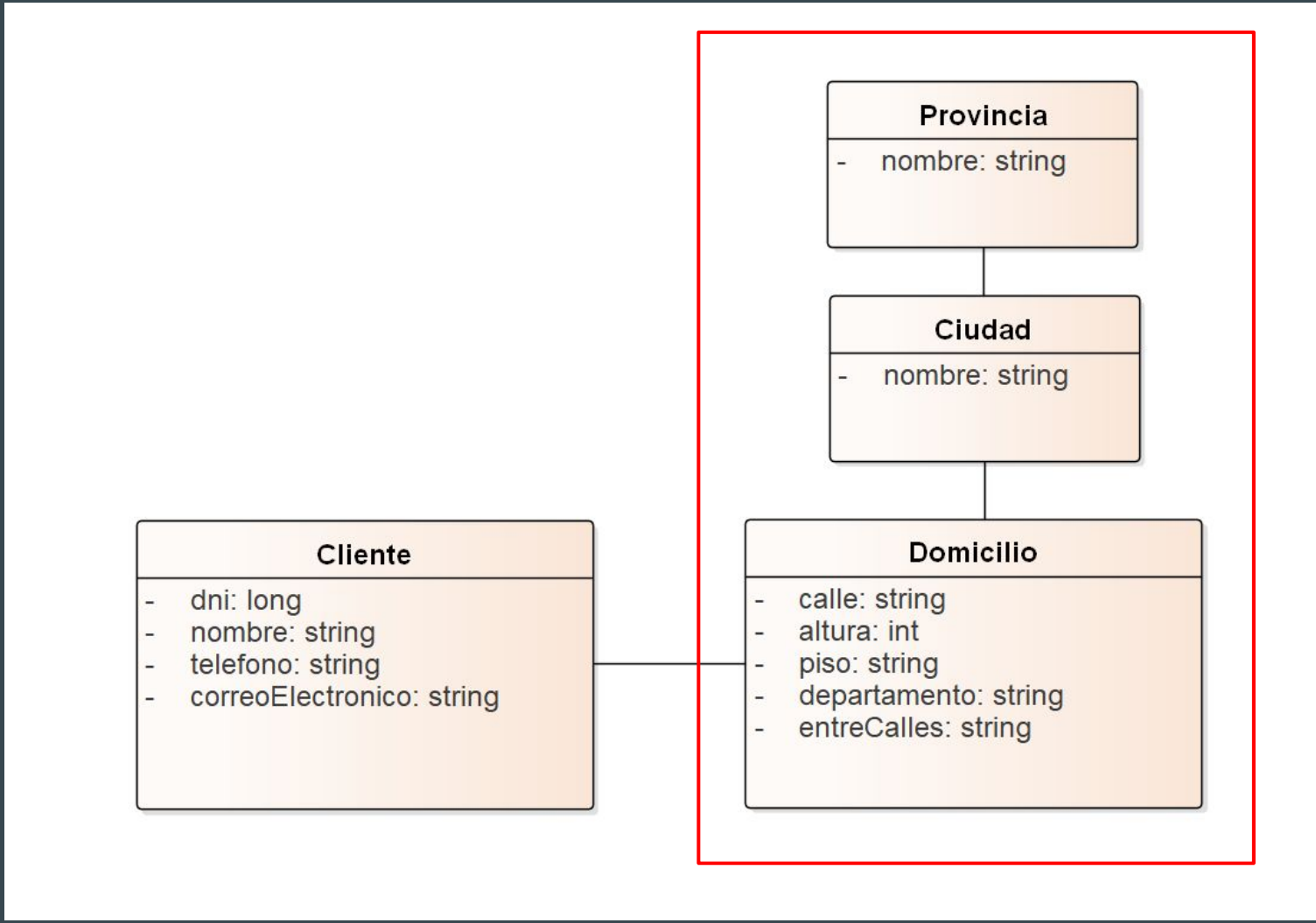
Abstracción - Ejemplo

Sistema para Bares



Abstracción - Ejemplo

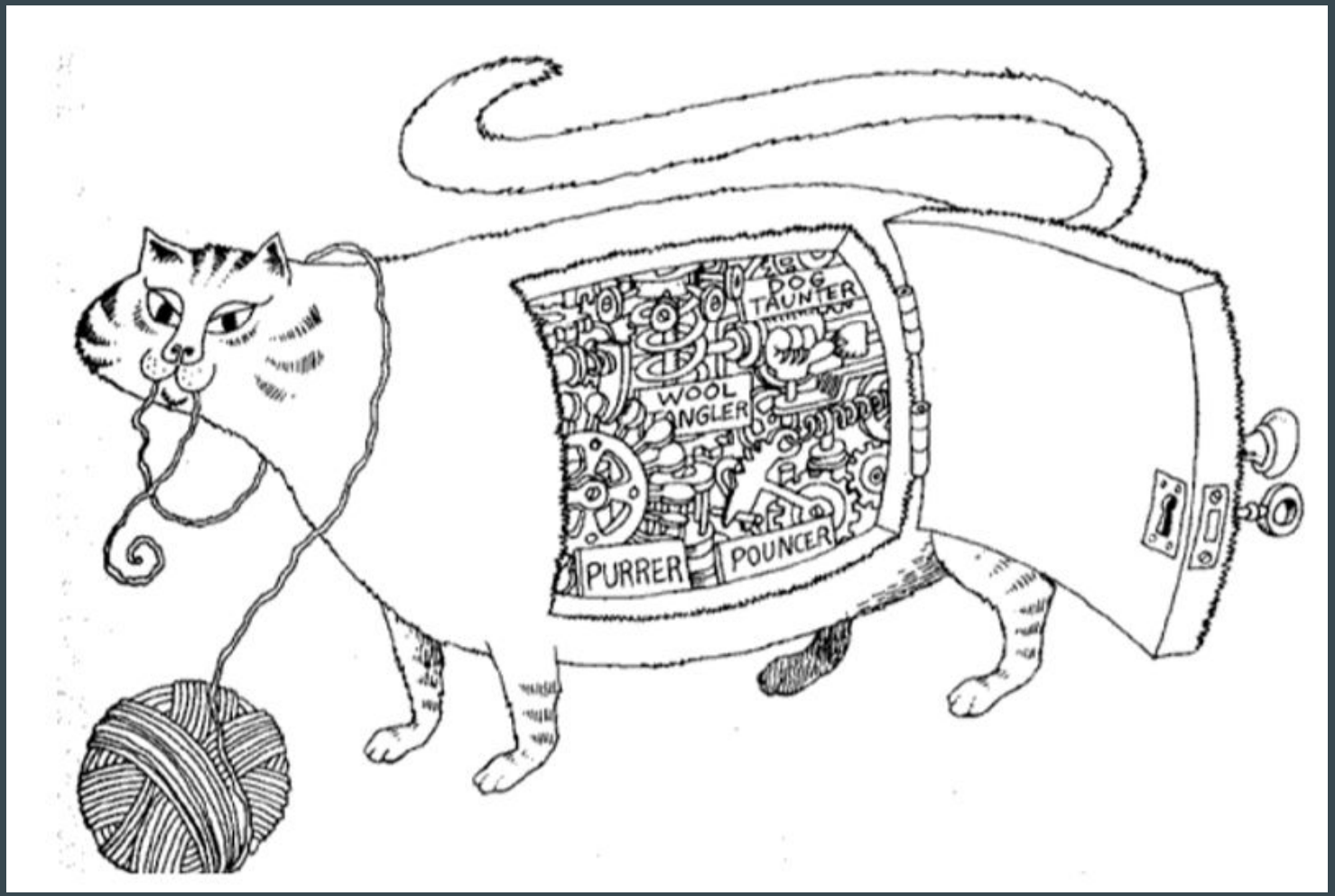
Sistema para Transportes



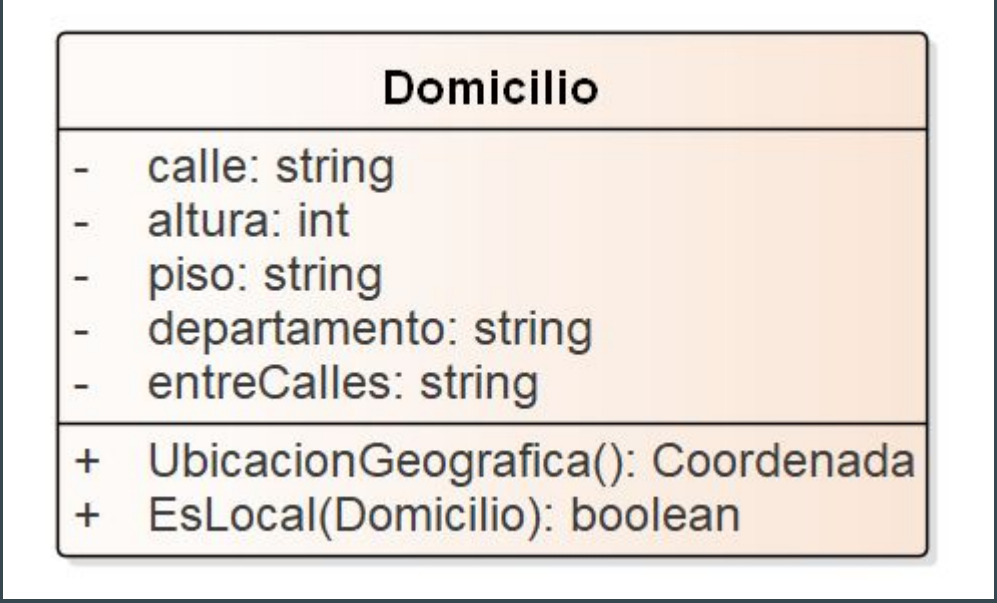
Abstracción - Conceptos asociados

- Principio de mínima sorpresa
- Modelo contractual de programación
- Responsabilidad
- Invariancia (pre y poscondiciones)
- Excepciones

Encapsulamiento

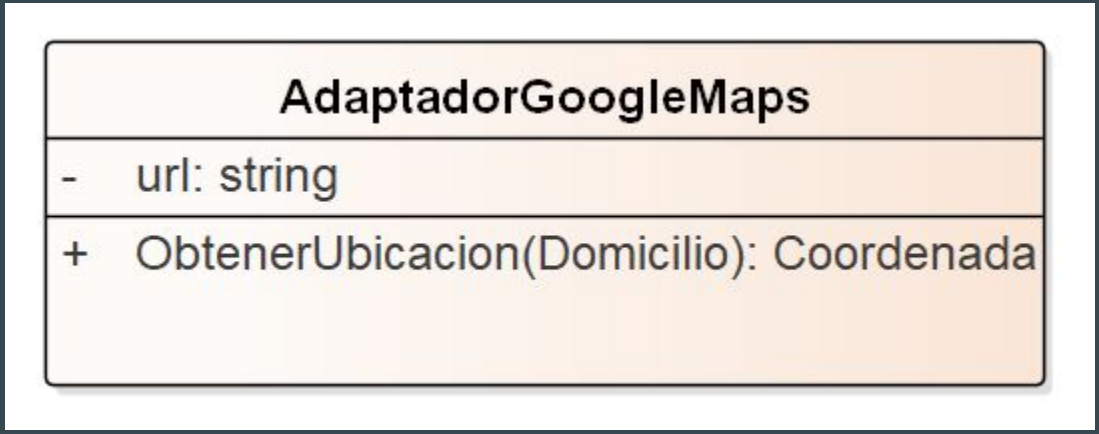


Encapsulamiento - Ejemplo



La abstracción y el encapsulamiento son complementarios: la abstracción se centra en el comportamiento observable de un objeto, mientras el encapsulamiento se centra en la implementación que da lugar a ese comportamiento.

Encapsulamiento - Ejemplo

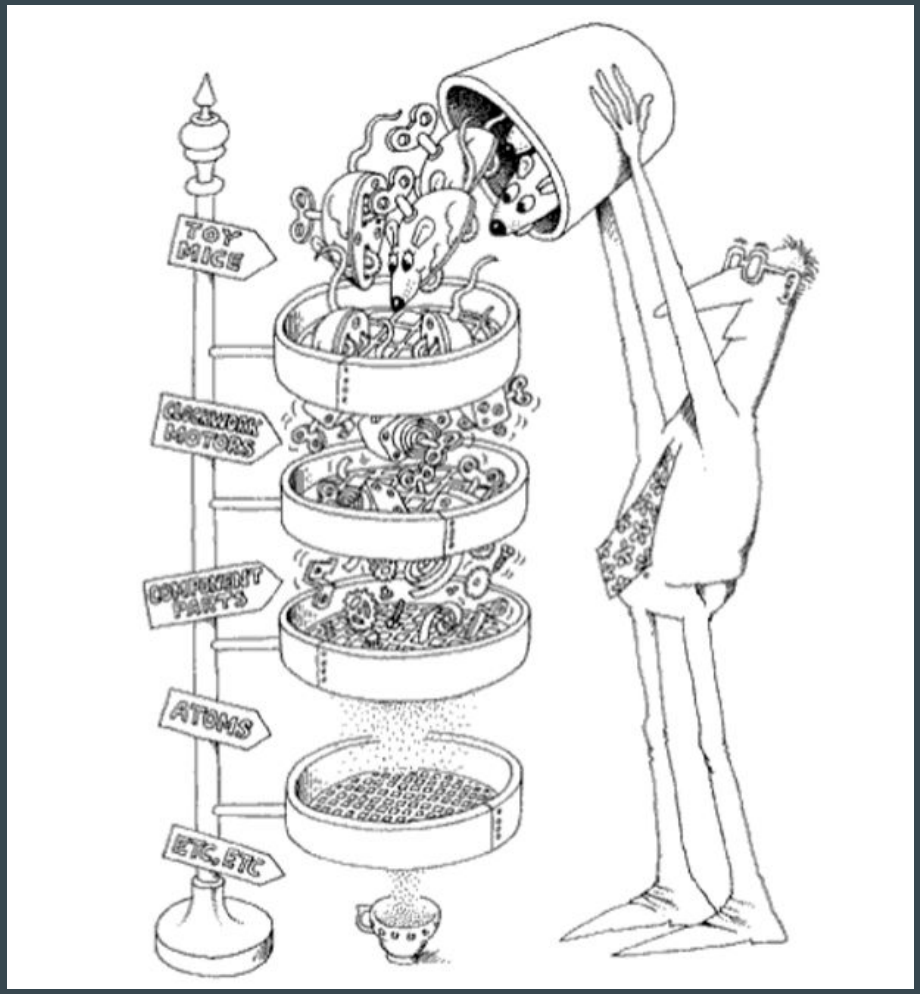



```
public class AdaptadorGoogleMaps{

    public Coordenada ObtenerUbicacion(Domicilio domicilio){
        var cliente = new HttpClient("https://maps.google.com");
        cliente.Conectar();
        var ubicacion = cliente.Geocode(domicilio.Calle, domicilio.Altura,
                                        domicilio.Ciudad, domicilio.Provincia);
        return new Coordenada(ubicacion.Latitud, ubicacion.Longitud);
    }
}
```

Modularidad

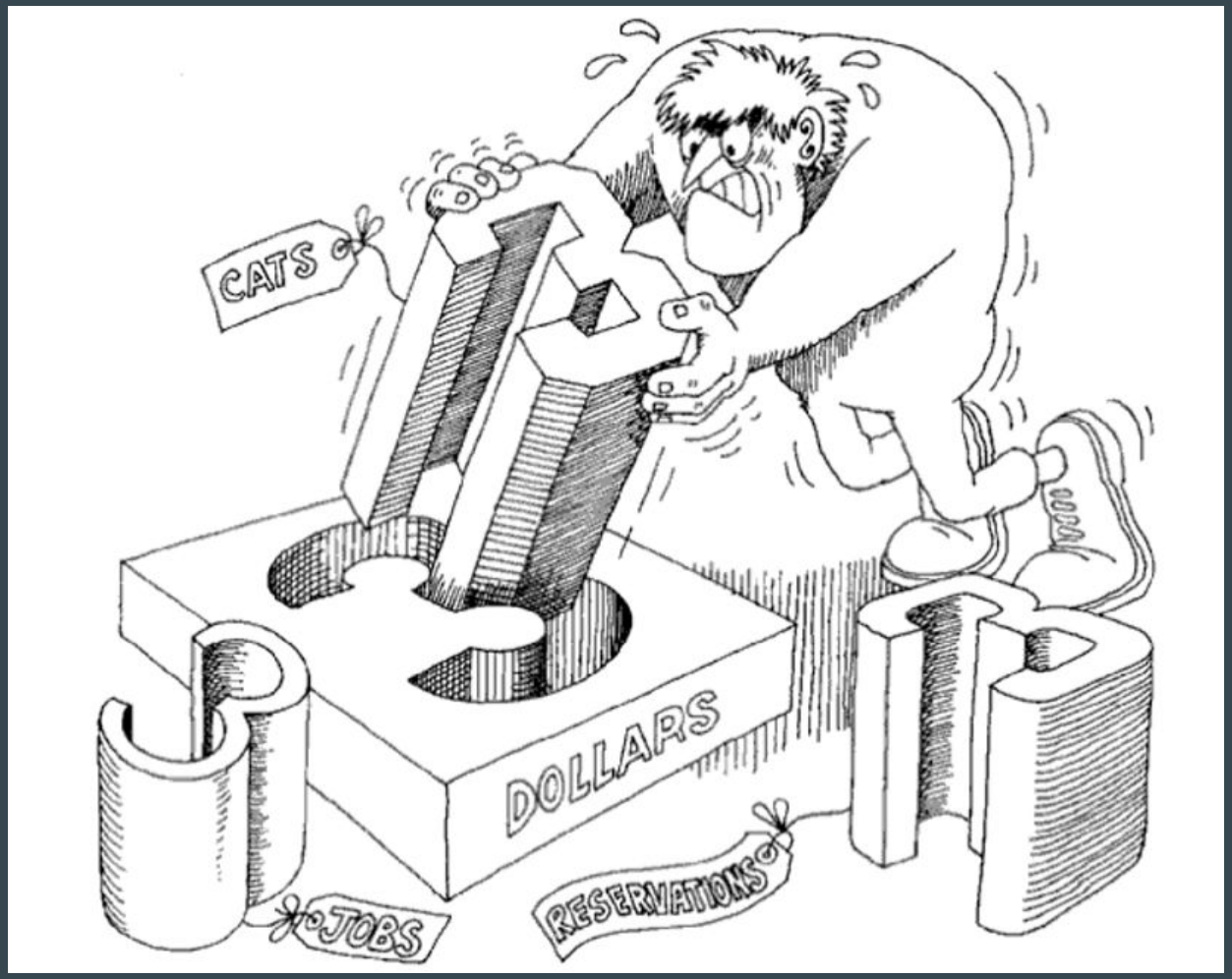
Jerarquía



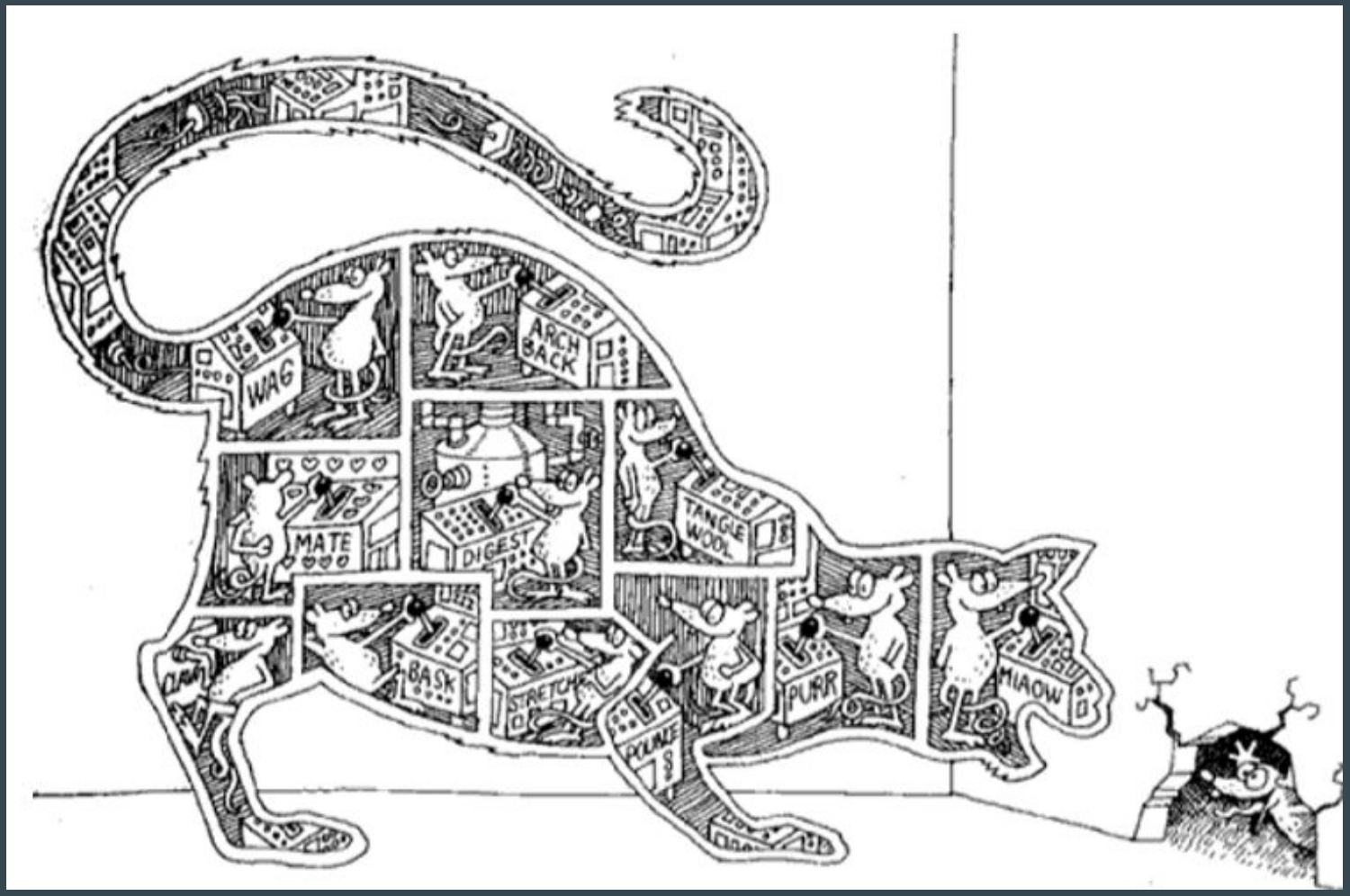
POO - Elementos Secundarios

- Tipos (Tipificación)
- Concurrencia
- Persistencia

Tipos (Tipificación)



Concurrencia



Persistencia

.NET - Definición

.NET es una plataforma para desarrolladores de código abierto multiplataforma* y de uso general. Tiene varias características clave, como la compatibilidad con varios lenguajes de programación, modelos de programación asincrónica y simultánea e interoperabilidad nativa, que permiten una amplia variedad de escenarios en diversas plataformas.

https://docs.microsoft.com/es-es/dotnet/standard/tour

.NET - Definición

.NET es una plataforma de desarrollo gratuita de código abierto para compilar muchos tipos de aplicaciones, como las siguientes:

- Aplicaciones web, API web y microservicios
- Funciones sin servidor en la nube
- Aplicaciones nativas de la nube
- Aplicaciones móviles
- Aplicaciones de escritorio
- Windows WPF
- **Windows Forms**
- Plataforma universal de Windows (UWP)
- Juegos
- Internet de las cosas (IoT)
- Aprendizaje automático
- Aplicaciones de consola
- Servicios de Windows

¿Qué NO ES .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

Código Administrado

Al trabajar con .NET, a menudo se encontrará el término "código administrado". Se trata de código cuya ejecución está administrada mediante un tiempo de ejecución (*runtime*).

Ejecución de Código Nativo

Ejecución de Código Administrado

Plataforma de Ejecución Intermedia

.NET - Componentes de la arquitectura - .NET Standard

Una aplicación de .NET se desarrolla y se ejecuta en una o varias implementaciones de .NET.

Hay una especificación de API común a todas las implementaciones de .NET que se denomina .NET Standard.

- Es una especificación uniforme de contratos contra los que se compila el código
- Es también una plataforma de destino que se puede ejecutar en cualquier implementación compatible

https://dotnet.microsoft.com/en-us/platform/dotnet-standard

.NET - Componentes de la arquitectura - .NET 5 - .NET 6 - .NET 7

.NET 7 es ahora la implementación principal, la que recibe desarrollo continuo. Se basa en una única base de código que admite varias plataformas y muchas cargas de trabajo, como aplicaciones de escritorio de Windows y aplicaciones de consola multiplataforma, servicios en la nube y sitios web.

¿Qué es .NET?

.NET - Componentes de la arquitectura - Implementaciones

- .NET Framework (4.8)
- .NET (Core 3.1 - 5.0 - 6.0 - 7.0)
- Mono
- Plataforma Universal de Windows (UWP)

https://dotnet.microsoft.com/download

.NET - Componentes de la arquitectura - Implementaciones

Cada implementación de .NET incluye los siguientes componentes:

- Uno o varios entornos de ejecución. Ejemplos: CLR para .NET Framework, CoreCLR y CoreRT para .NET Core.
- Una biblioteca de clases que implementa .NET Standard y puede implementar API adicionales. Ejemplos: biblioteca de clases base de .NET Framework, biblioteca de clases base de .NET Core.
- Opcionalmente, uno o varios marcos de trabajo de la aplicación. Ejemplos: ASP.NET, Windows Forms y Windows Presentation Foundation (WPF) se incluyen en .NET Framework y .NET Core.
- Opcionalmente, herramientas de desarrollo. Algunas herramientas de desarrollo se comparten entre varias implementaciones. Ejemplo: C#.

.NET - Componentes de la arquitectura - Entornos de Ejecución

Un entorno de ejecución es el entorno de ejecución de un programa administrado.

- Common Language Runtime (CLR) para .NET Framework
- Core Common Language Runtime (CoreCLR) para .NET Core
- .NET Native para la Plataforma universal de Windows
- El entorno de ejecución Mono para Xamarin.iOS, Xamarin.Android, Xamarin.Mac y el marco de escritorio de Mono

.NET - Componentes de la arquitectura - Herramientas

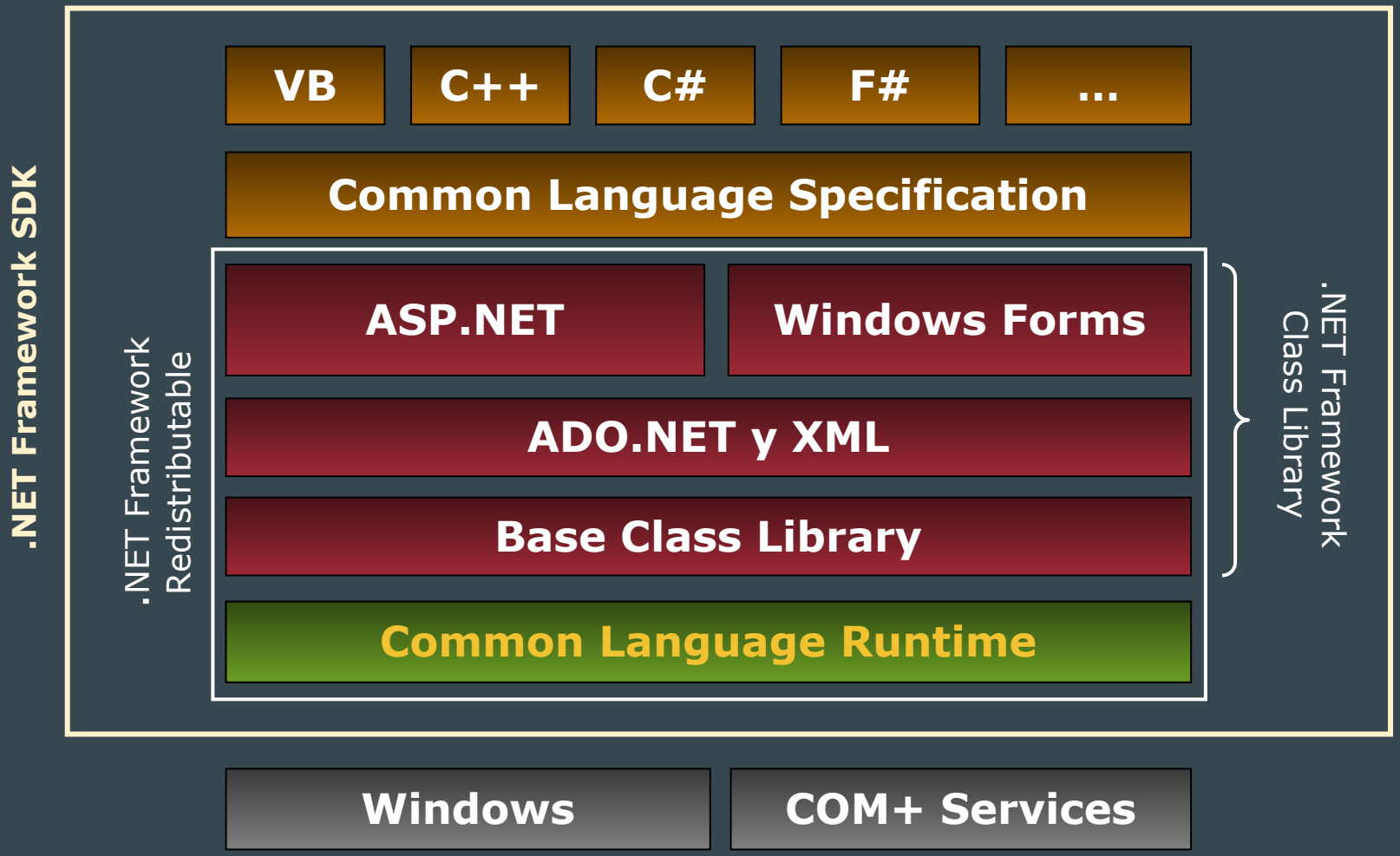
Acceso a un amplio conjunto de herramientas y componentes de infraestructura que funcionan con todas las implementaciones de .NET. Estas herramientas y componentes incluyen:

- Los lenguajes .NET y sus compiladores
- El sistema de proyectos de .NET (basado en archivos .csproj, .vbproj y .fsproj)
- MSBuild, el motor de compilación usado para compilar proyectos
- NuGet, administrador de paquetes de Microsoft para .NET
- Herramientas de organización de compilación de código abierto, como CAKE y FAKE

.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 (obsoleto)

Arquitectura del .NET Framework

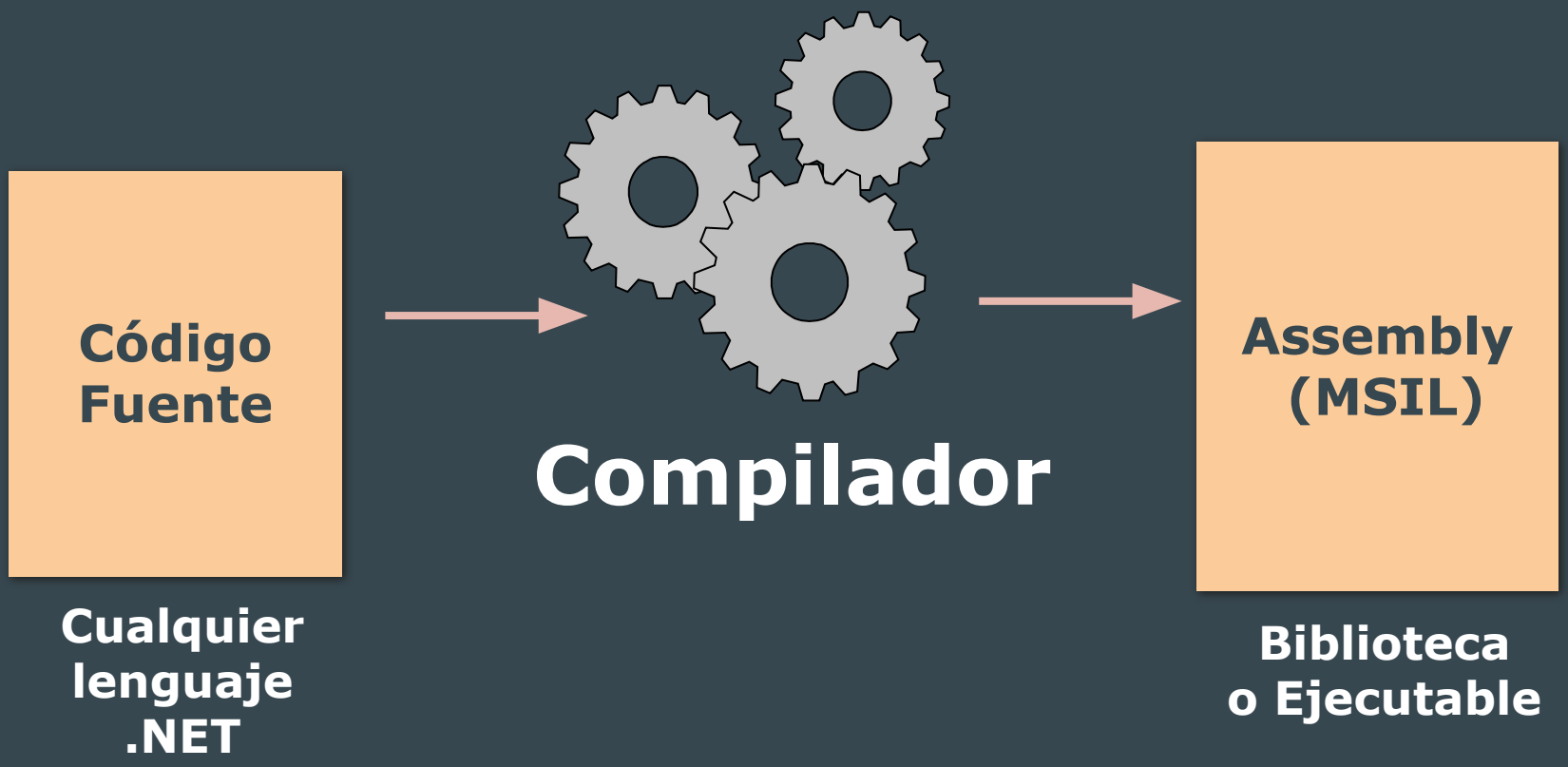


CLR - Common Language Runtime

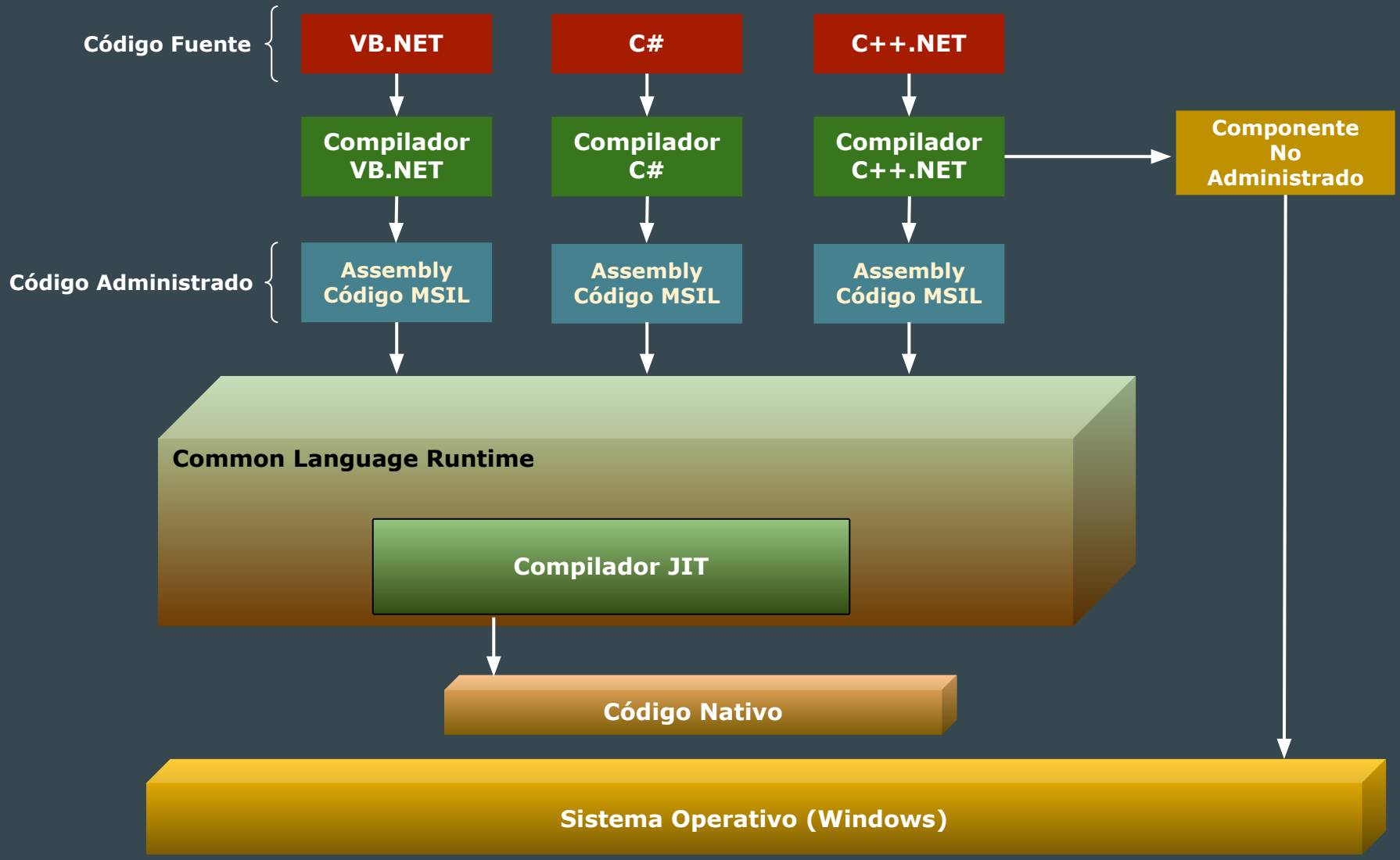
- El CLR es el motor de ejecución (*runtime*) de .NET
- Características
  - Compilación Just-In-Time (JIT)
  - Gestión automática de memoria (*Garbage Collector*)
  - Gestión de errores consistente (Excepciones)
  - Ejecución basada en componentes (*Assemblies*)
  - Gestión de Seguridad
  - *Multithreading*

https://docs.microsoft.com/es-es/dotnet/standard/clr

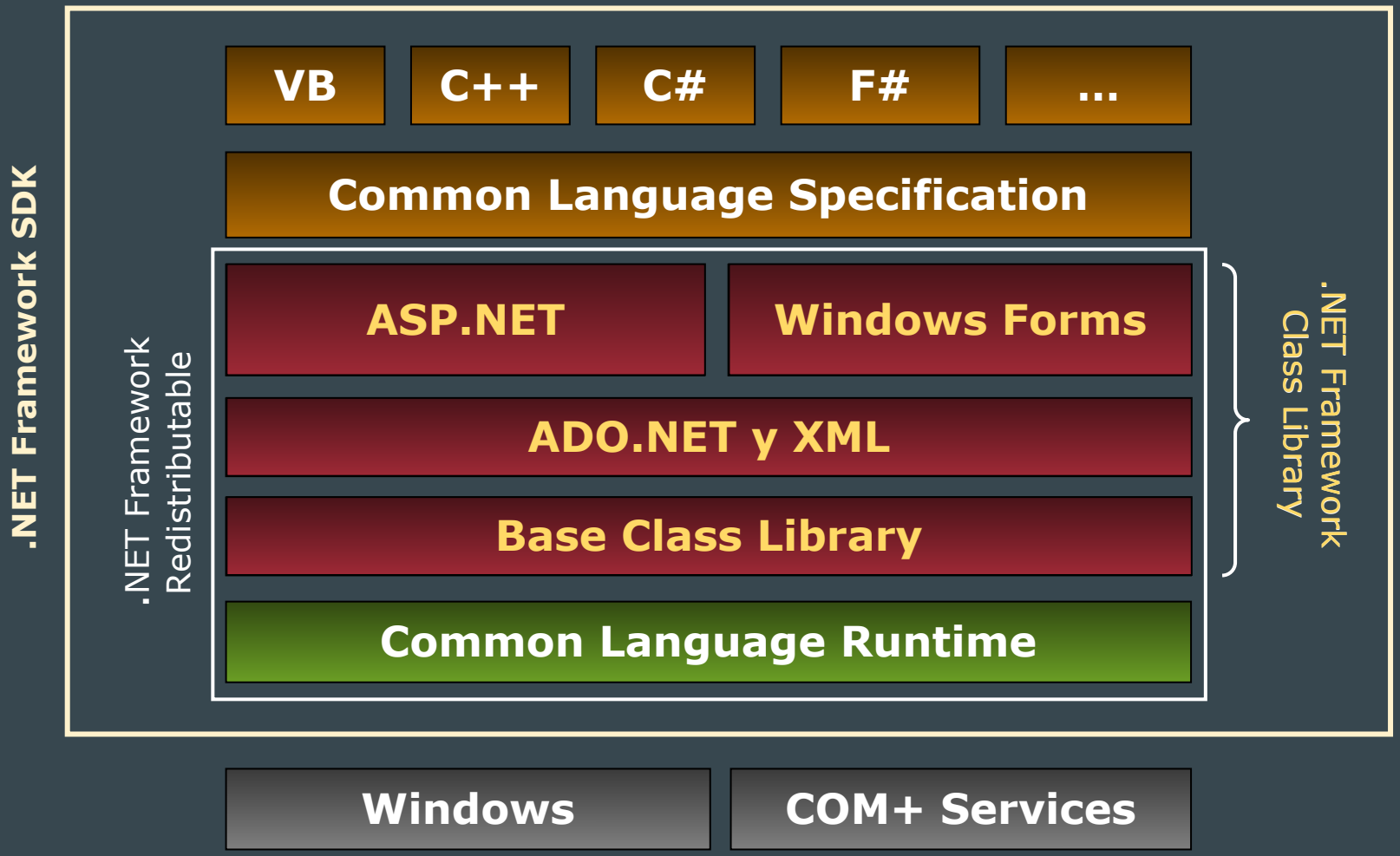
CLR - Proceso de Compilación



CLR - Modelo de Ejecución

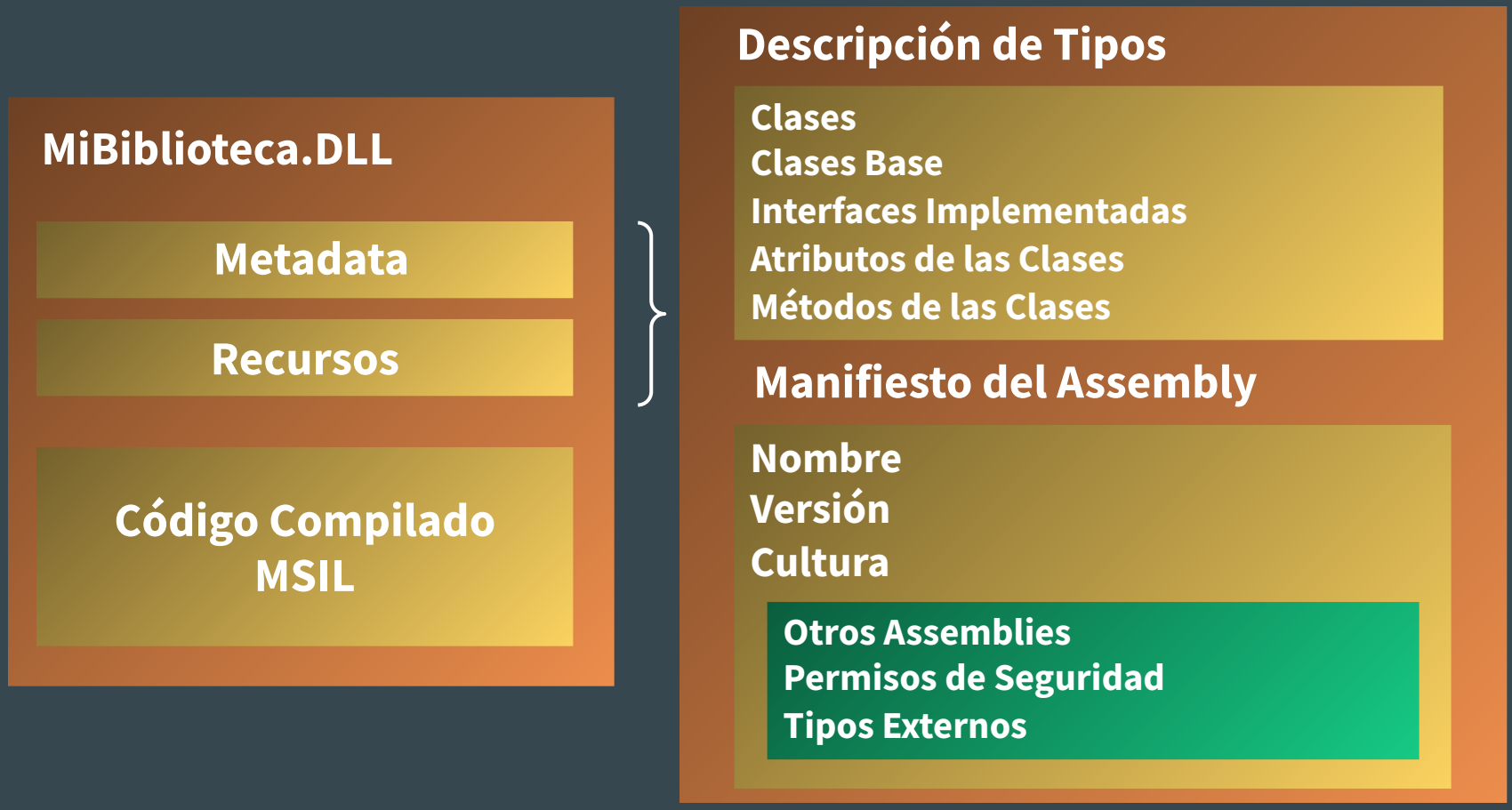


Arquitectura del .NET Framework



Ensamblados de .NET (Assembly)

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



MSIL - Ejemplo

```
.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
}
```

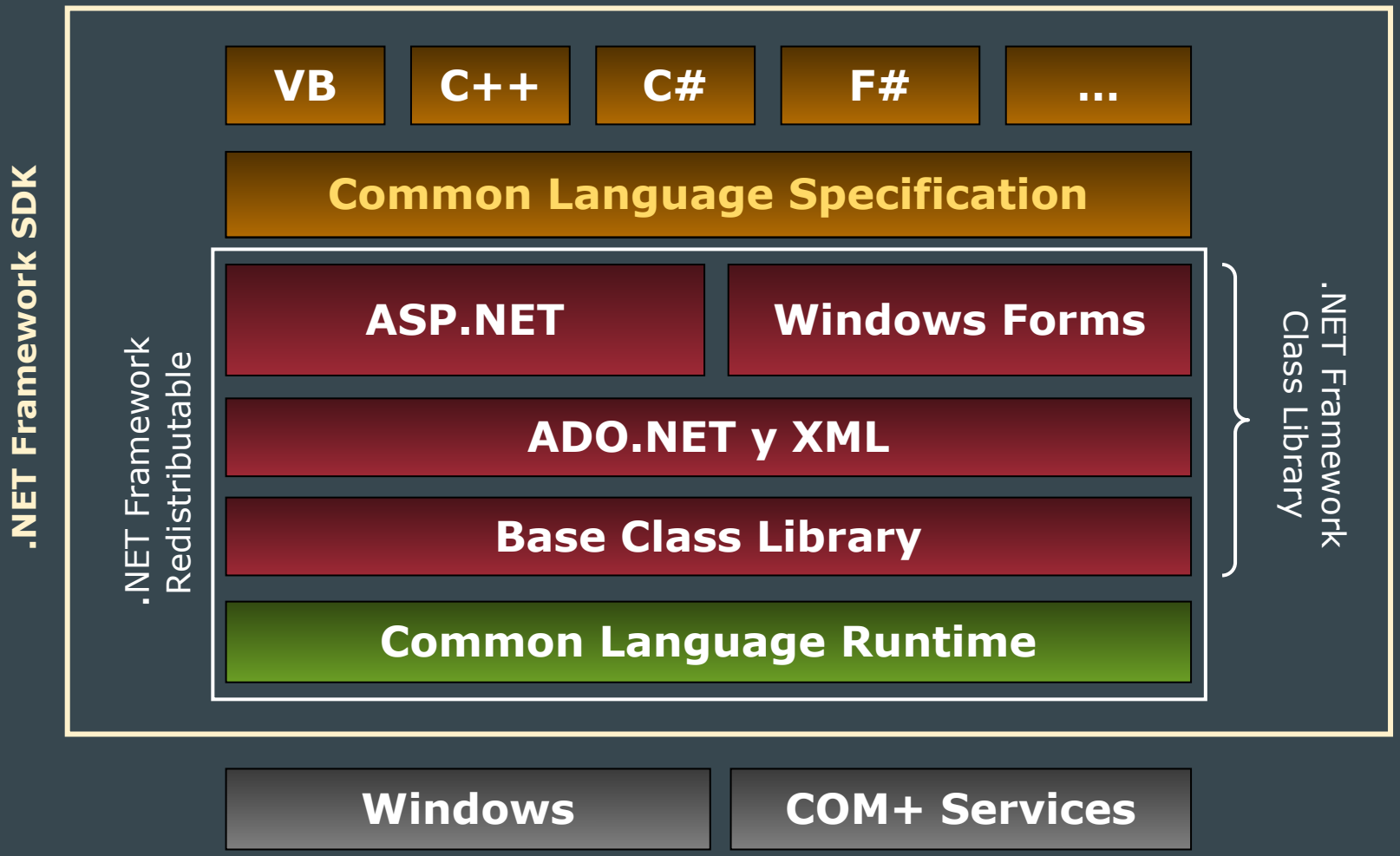
Ensamblados (assemblies) en las Aplicaciones .NET

- Uno o más *assemblies*
- Adoptan la forma de un archivo ejecutable (.exe) o de biblioteca de vínculos dinámicos (.dll)
- Ubicación
  - El Class Loader busca en el directorio local (preferido)
  - *Global Assembly Cache* (GAC)
- Diferentes aplicaciones pueden usar diferentes versiones
- Solo se cargan en memoria si son necesarios
- Pueden ser estáticos o dinámicos
- Se agregan a las aplicaciones a través de referencias

.NET Framework Class Library

- Conjunto de Tipos básicos (clases, interfaces, etc.) que vienen incluídos en el .NET Framework
- Los tipos están organizados en jerarquías lógicas de nombres, denominados NAMESPACES
- Los tipos son INDEPENDIENTES del lenguaje de desarrollo
- Es extensible y totalmente orientada a objetos

Arquitectura del .NET Framework



CLS - Common Language Specification

- 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 varios lenguajes, todos compatibles con CLS:
  - C#
  - C++/CLI
  - Eiffel
  - F#
  - IronPython
  - IronRuby
  - PowerBuilder
  - Visual Basic
  - Visual COBOL
  - Windows PowerShell

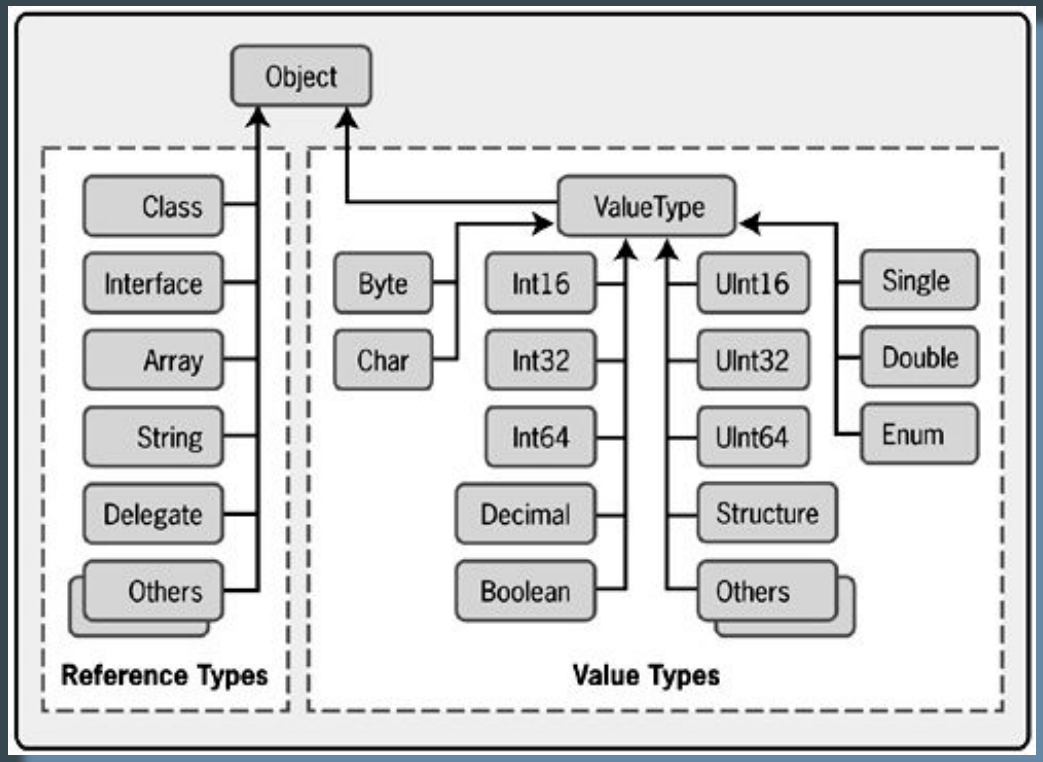
https://docs.microsoft.com/es-es/dotnet/standard/language-independence

CTS - Common Type System

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



CLI - Common Language Infrastructure

Lenguajes de Alto Nivel

se ajustan a las reglas de la...

CLS (Common Language Specification)

y utilizan las clases de la...

BCL (Base Class Library)

cuyos tipos básicos forman el...

CTS (Common Type System)

y se ejecutan bajo el control y usan los servicios del...

CLR (Common Language Runtime)

que está acoplado y utiliza los servicios del...

Sistema Operativo