Hada T7: Plataforma .NET

Herramientas Avanzadas para el Desarrollo de Aplicaciones

INDICE

Introducción a .Net

¿Qué es .Net?

¿Qué es .net?

- .net es una plataforma que permite el desarrollo de **aplicaciones software y librerías.**

- .net contiene el **compilador y las herramientas necesarias** para construir, depurar y ejecutar estas aplicaciones.

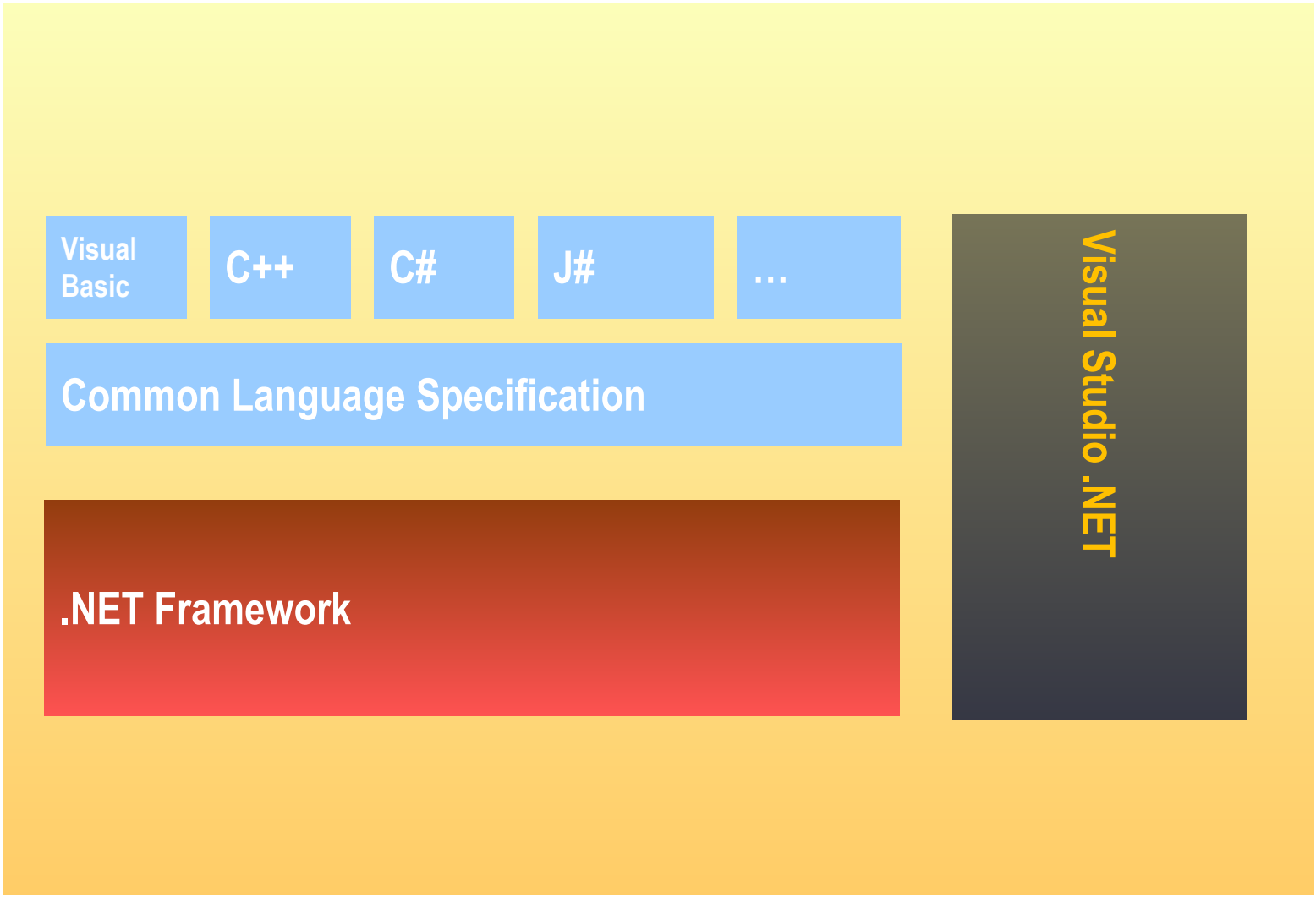
¿Qué es .net? (II)

- Es una **plataforma software**.
- **Provee** un **entorno de desarrollo** independiente del lenguaje, que permite escribir programas de forma sencilla, e incluso permite combinar código escrito en diferentes lenguajes.
- No está orientado a un Hardware/Sistema Operativo concreto, sino a cualquier plataforma para la que .net esté desarrollado.

Introducción a .Net

Arquitectura .Net

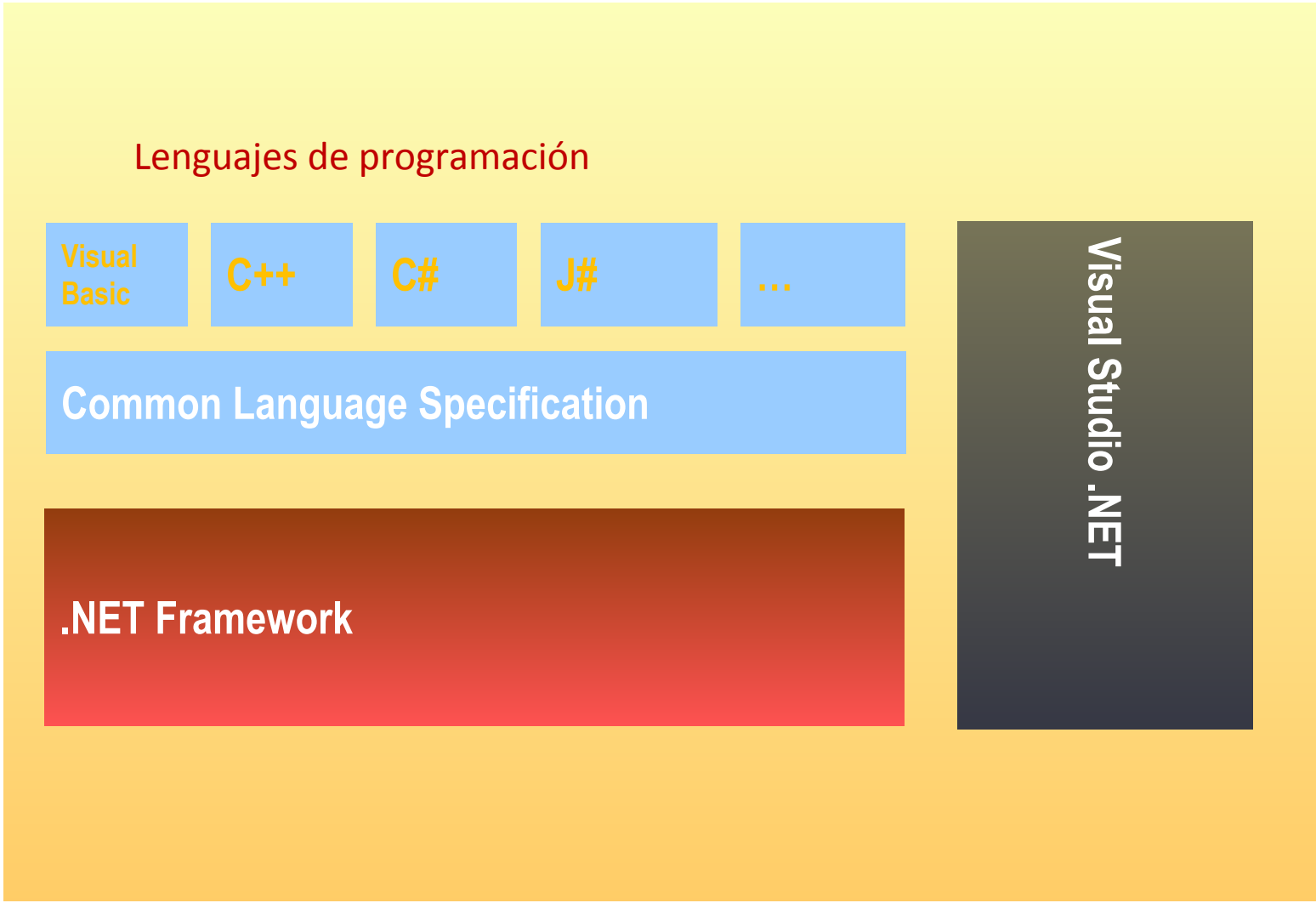
Arquitectura .net



Visual Studio .net

- **Visual Studio .NET** ofrece un **entorno de desarrollo** para desarrollar aplicaciones que se ejecutan sobre el .NET Framework.
- Proporciona las tecnologías fundamentales para simplificar la creación e implantación de
  - Aplicaciones y Servicios Web
  - Aplicaciones basadas en Windows

Arquitectura .net

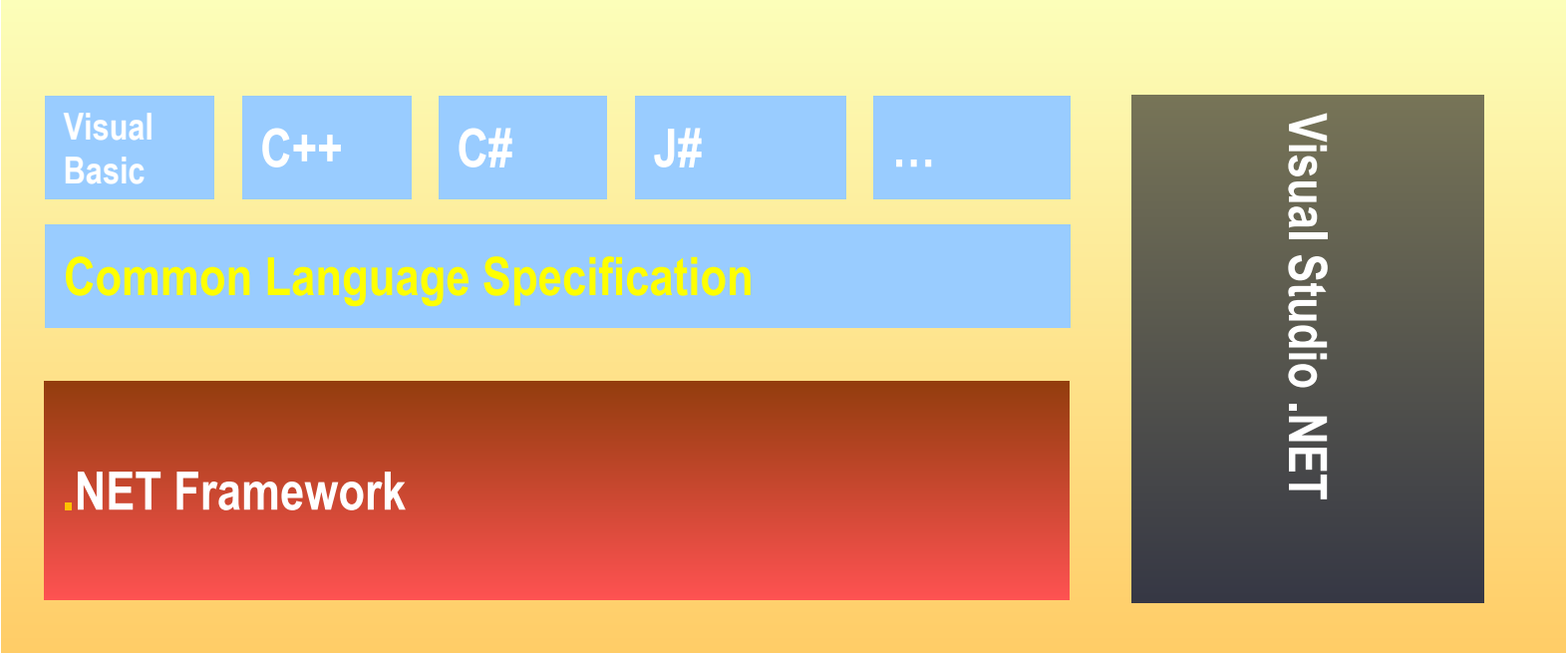


Lenguajes de programación

- **C#**
  - **C# ha sido diseñado específicamente para la plataforma .NET y es el primer lenguaje moderno orientado a componentes de la familia de C y C++.**
  - Puede incrustarse en páginas ASP.NET.
  - Algunas de las principales características de este lenguaje incluye *clases, interfaces, delegados, espacios de nombres, propiedades, indexadores, eventos, sobrecarga de operadores, atributos, creación de documentación en formato XML ….*

- **Otros lenguajes soportados:**
  - Visual Basic .net, C++, J#
  - Lenguajes de terceros

CLS (I)



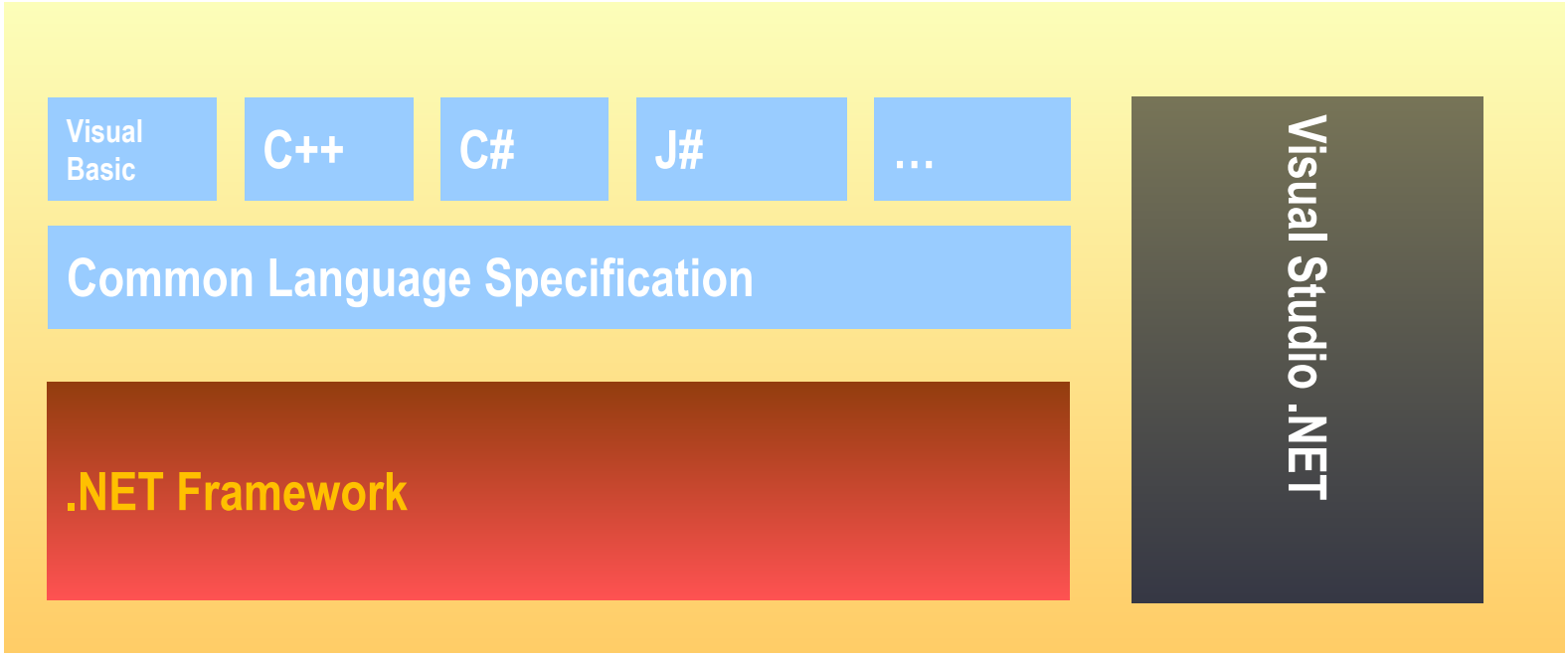
- La especificación **Common Language Specification (CLS)** define los **mínimos estándares** que deben satisfacer los lenguajes y desarrolladores si desean que sus componentes y aplicaciones sean ampliamente utilizados por otros lenguajes compatibles con .NET.

Common Language Specification (II)

- La especificación **CLS** permite a los desarrolladores de .NET crear aplicaciones como parte de un equipo que utiliza **múltiples lenguajes** con la seguridad de que no habrá problemas con la integración de los diferentes lenguajes.

- La especificación **CLS** también permite a los desarrolladores de .NET **heredar de clases desarrolladas en lenguajes diferentes.**

.net framework



- Es el **motor de ejecución**
- Proporciona un conjunto de servicios comunes para los proyectos generados en .net, con independencia del lenguaje.

.net Framework (II)

- **Extensible**
  - La jerarquía del .NET Framework no queda oculta al desarrollador. **Podemos acceder y extender clases .NET** (a menos que estén selladas) **utilizando herencia.** También podemos implementar herencia multilenguaje.
- **Fácil de usar por los desarrolladores**
  - En el .NET Framework, **el código está organizado en espacios de nombres jerárquicos y clases**. El Framework proporciona un **sistema de tipos común, denominado sistema de tipos unificado**, que utiliza cualquier lenguaje compatible con .NET.
  - **En el sistema de tipos unificado, todo es un objeto.**

Introducción a .Net

Arquitectura .net Framework

Partes del .NET Framework.

El **Common Language Runtime** (**CLR**) es el núcleo de la plataforma .NET.
Es el motor encargado de gestionar la ejecución de las aplicaciones para ella desarrolladas y a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad.

Common Language Runtime (CLR)

- **Ejecución multiplataforma:** **El CLR actúa como una máquina virtual**, encargándose de ejecutar las aplicaciones diseñadas para la plataforma .NET. Es decir, cualquier plataforma para la que exista una versión del CLR podrá ejecutar cualquier aplicación .NET.

Hasta ahora solo se habían desarrollados CLR para todas las versiones de Windows, existe la posibilidad de desarrollar una versión para sistemas como Unix o Linux debido a que la arquitectura del CLR es abierta.

Proyecto Mono: http://go-mono.com

Common Language Runtime (CLR)

- **Integración de lenguajes:**

Desde cualquier lenguaje para el que exista un compilador que genere código para la plataforma .NET **es posible utilizar código generado para la misma usando cualquier otro lenguaje** tal y como si de código escrito usando el primero se tratase.

La integración de lenguajes provee que es posible escribir una clase en C# que herede de otra escrita en Visual Basic.NET que, a su vez, herede de otra escrita en C++ *con extensiones gestionadas*.

Microsoft Intermediate Language (MSIL)

- Los compiladores que generan código para la plataforma .NET generan código escrito en el lenguaje intermedio conocido como **Microsoft Intermediate Lenguage (MSIL)**

Common Language Runtime (CLR)

- **Gestión de memoria:**

El CLR incluye un **recolector de basura** que evita que el programador tenga que tener en cuenta cuándo ha de destruir los objetos que dejen de serle útiles.

Gracias a este recolector **se evitan errores de programación muy comunes como:**

- Intentos de borrado de objetos ya borrados.
- Agotamiento de memoria por olvido de eliminación de objetos inútiles o
- Solicitud de acceso a miembros de objetos ya destruidos.

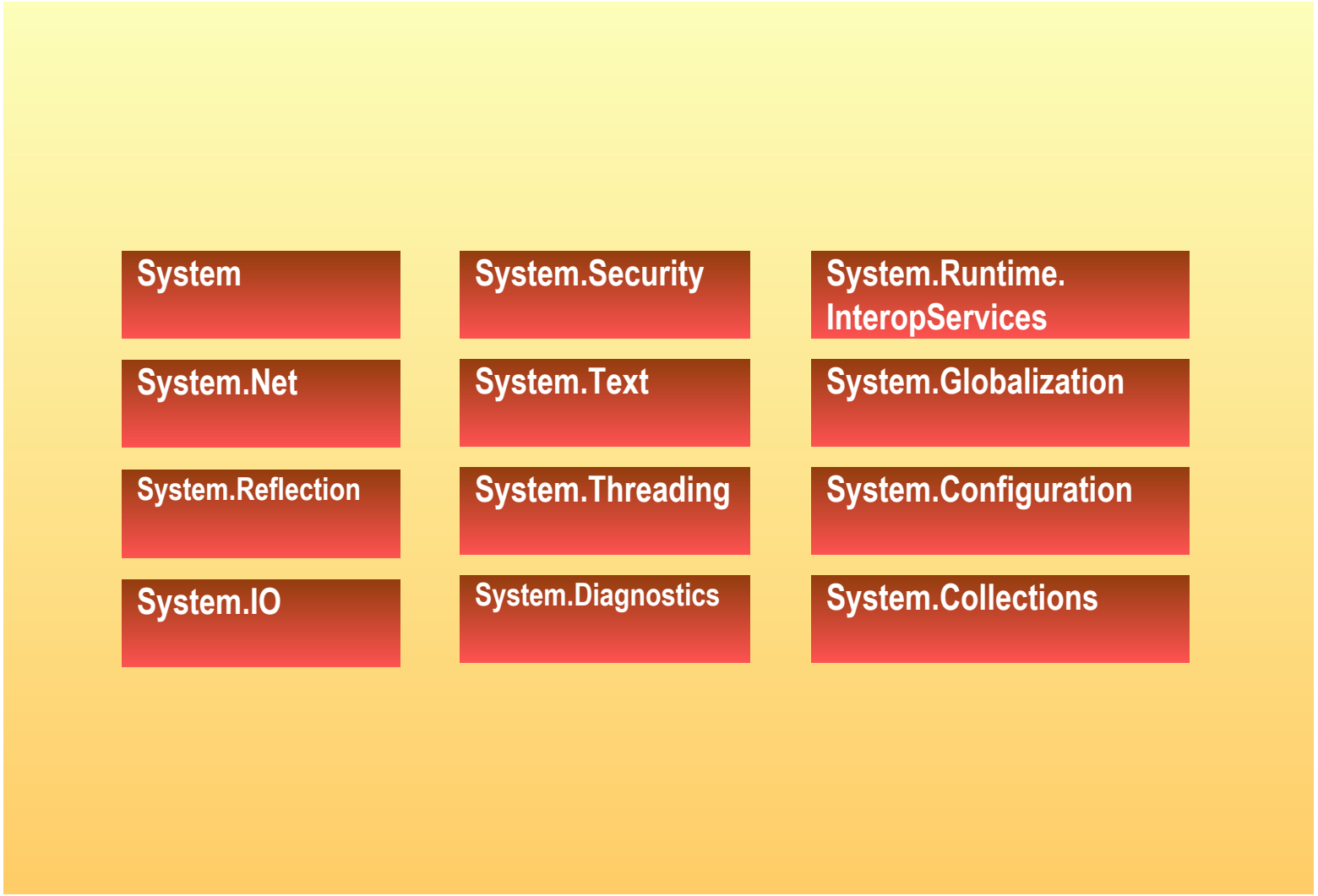
Common Language Runtime (CLR)

- **Seguridad de tipos:** El CLR facilita la detección de errores de programación difíciles de localizar **comprobando que toda conversión de tipos que se realice durante la ejecución de una aplicación .NET se haga de modo que los tipos origen y destino sean compatibles.**

Common Language Runtime (CLR)

- **Tratamiento de excepciones:** En el CLR todos los errores que se puedan producir durante la ejecución de una aplicación se propagan de igual manera: **mediante *excepciones*.**

Biblioteca de clases



Biblioteca de clases

- **La biblioteca de clases del .NET Framework** expone características del entorno de ejecución y **proporciona en una jerarquía de objetos otros servicios de alto nivel que todo programador necesita.** Esta jerarquía de objetos se denomina **espacio de nombres.**
- La biblioteca de clases del .NET Framework proporciona numerosas y potentes características nuevas para los desarrolladores:
  - Por ejemplo, **el espacio de nombres Collections** añade numerosas posibilidades nuevas, como clasificación, colas, pilas y matrices de tamaño automático.
  - **La clase de sistema Threading** también ofrece nuevas posibilidades para crear verdaderas aplicaciones multi-hilo.

Biblioteca de clases (II)

- **Espacio de nombres System**
  - El espacio de nombres System **contiene clases fundamentales y clases base** que definen tipos de datos valor y referencia comúnmente utilizados, eventos y descriptores de eventos, interfaces, atributos y procesamiento de excepciones.

Espacio de nombres	Utilidad de los tipos de datos que contiene
System	Tipos muy frecuentemente usados, como los los tipos básicos, tablas, excepciones, fechas, números aleatorios, recolector de basura, entrada/salida en consola, etc.
System.Collections	Colecciones de datos de uso común como pilas, colas, listas, diccionarios, etc.
System.Data	Manipulación de bases de datos. Forman la denominada arquitectura ADO.NET.
System.IO	Manipulación de ficheros y otros flujos de datos.
System.Net	Realización de comunicaciones en red.
System.Reflection	Acceso a los metadatos que acompañan a los módulos de código.
System.Runtime.Remoting	Acceso a objetos remotos.
System.Security	Acceso a la política de seguridad en que se basa el CLR.
System.Threading	Manipulación de hilos.
System.Web.UI.WebControls	Creación de interfaces de usuario basadas en ventanas para aplicaciones Web.
System.Winforms	Creación de interfaces de usuario basadas en ventanas para aplicaciones estándar.
System.XML	Acceso a datos en formato XML.

Introducción a .Net

El modelo de ejecución

El modelo de ejecución