

# La plataforma .NET

## Lenguajes y herramientas de .NET

**Juan Manuel Cueva Lovelle**

[cueva@lsi.uniovi.es](mailto:cueva@lsi.uniovi.es)

[www.di.uniovi.es/~cueva](http://www.di.uniovi.es/~cueva)

Departamento de Informática

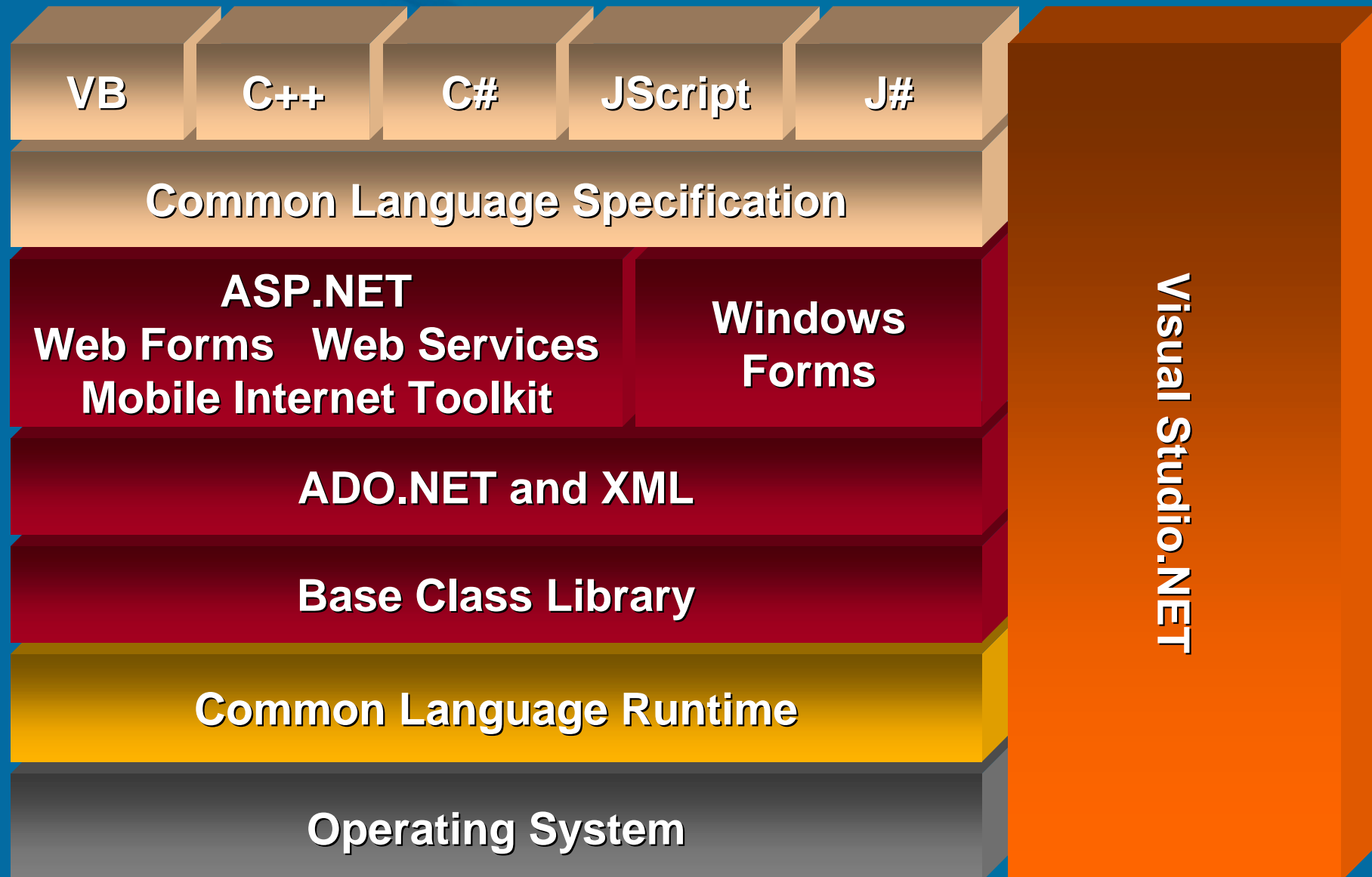
OOTLab [www.ootlab.uniovi.es](http://www.ootlab.uniovi.es)

# .NET Framework

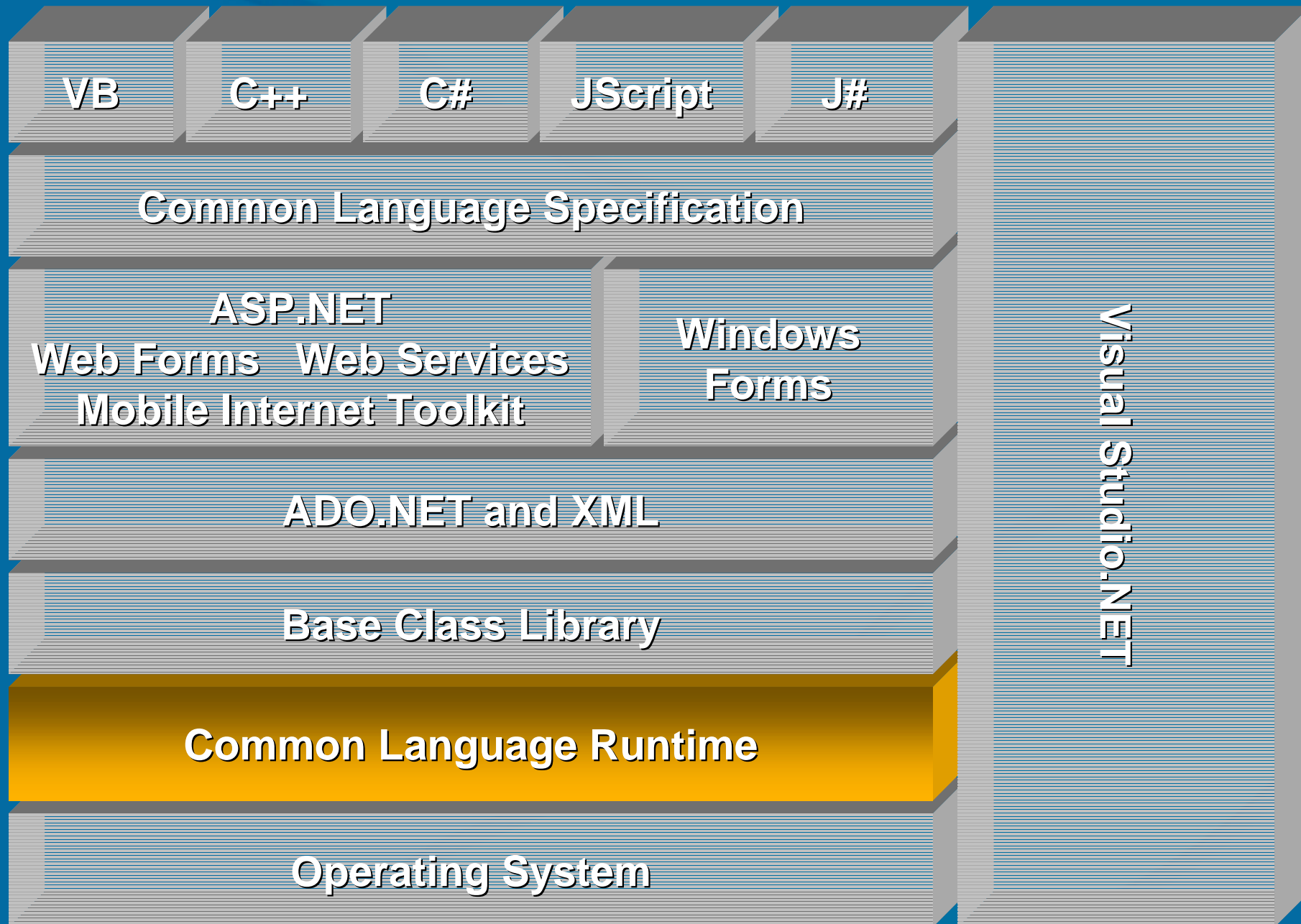
## Reglas de diseño

- Gran simplificación del desarrollo en implantación de aplicaciones
- Unificación de los modelos de programación
- Entorno de programación robusto y seguro
- Soporte para múltiples lenguajes de programación

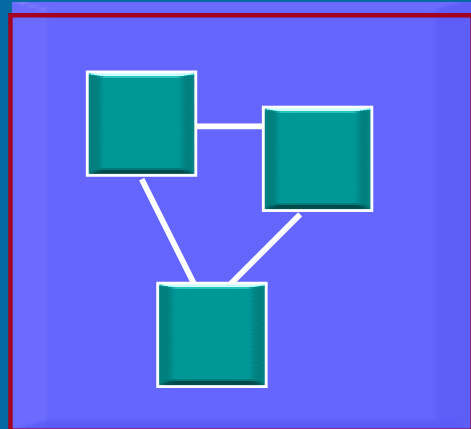
# Framework, Lenguajes y herramientas



# CLR

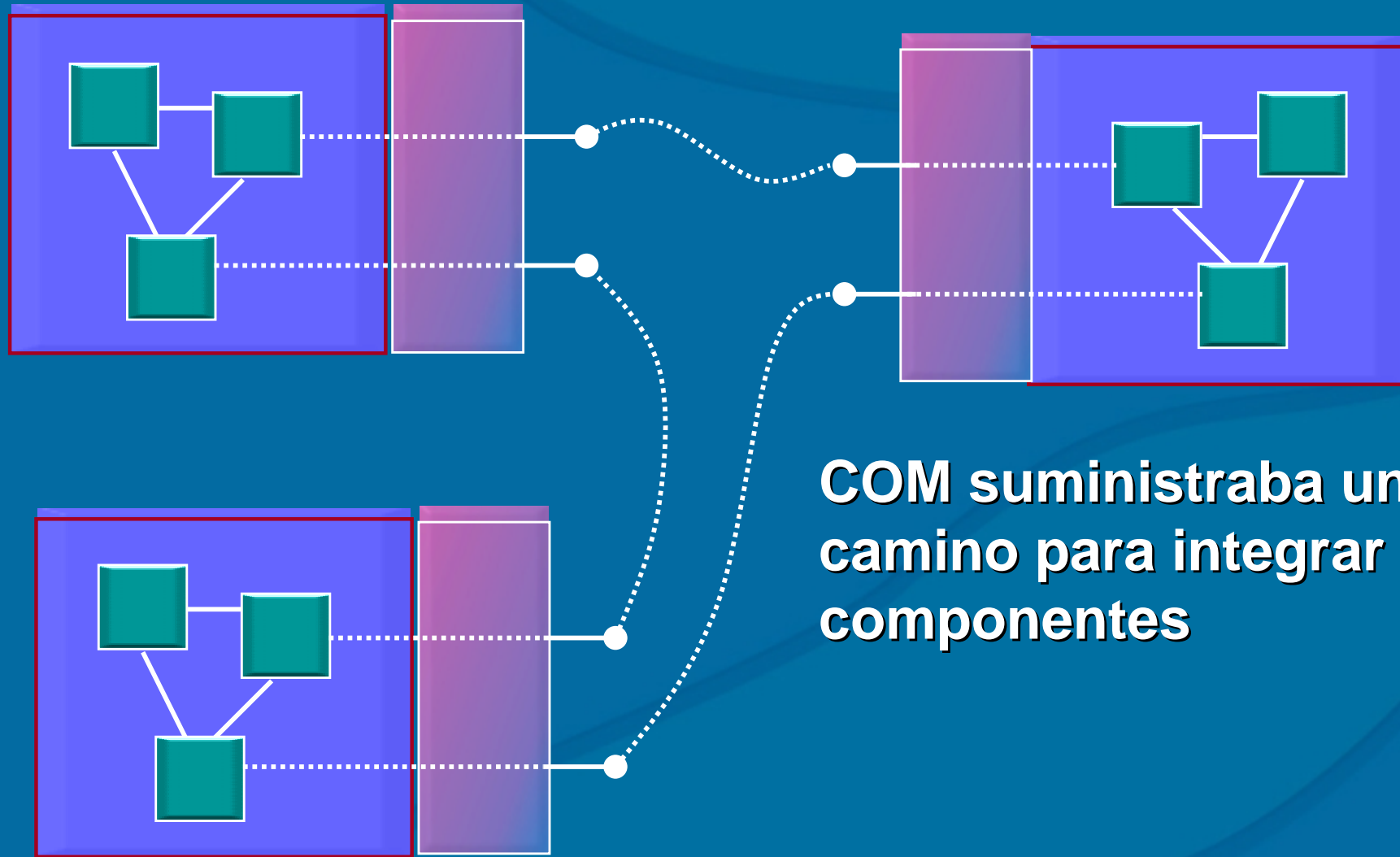


# La evolución de .NET



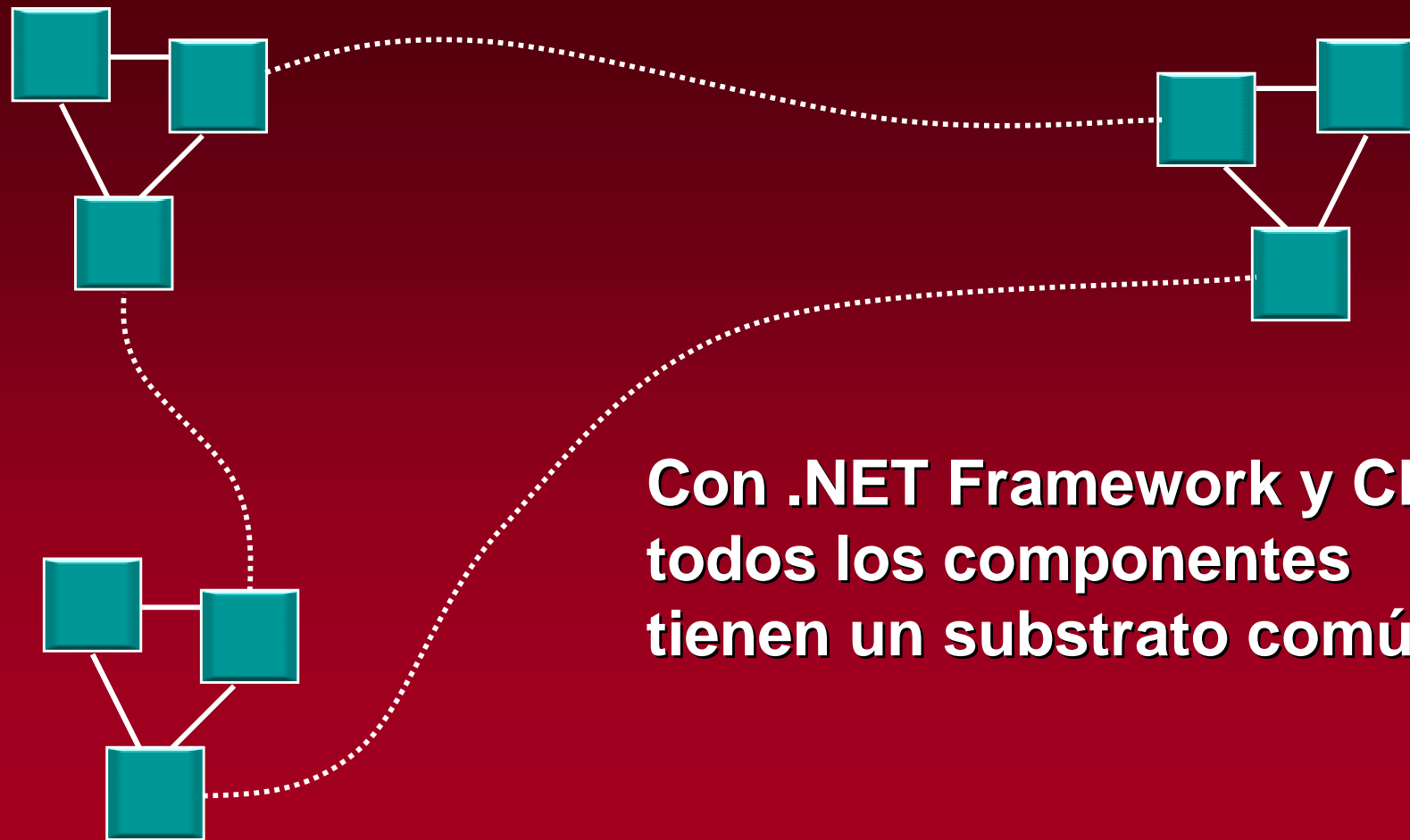
Aplicaciones desarrolladas como entidades separadas

# La evolución de .NET



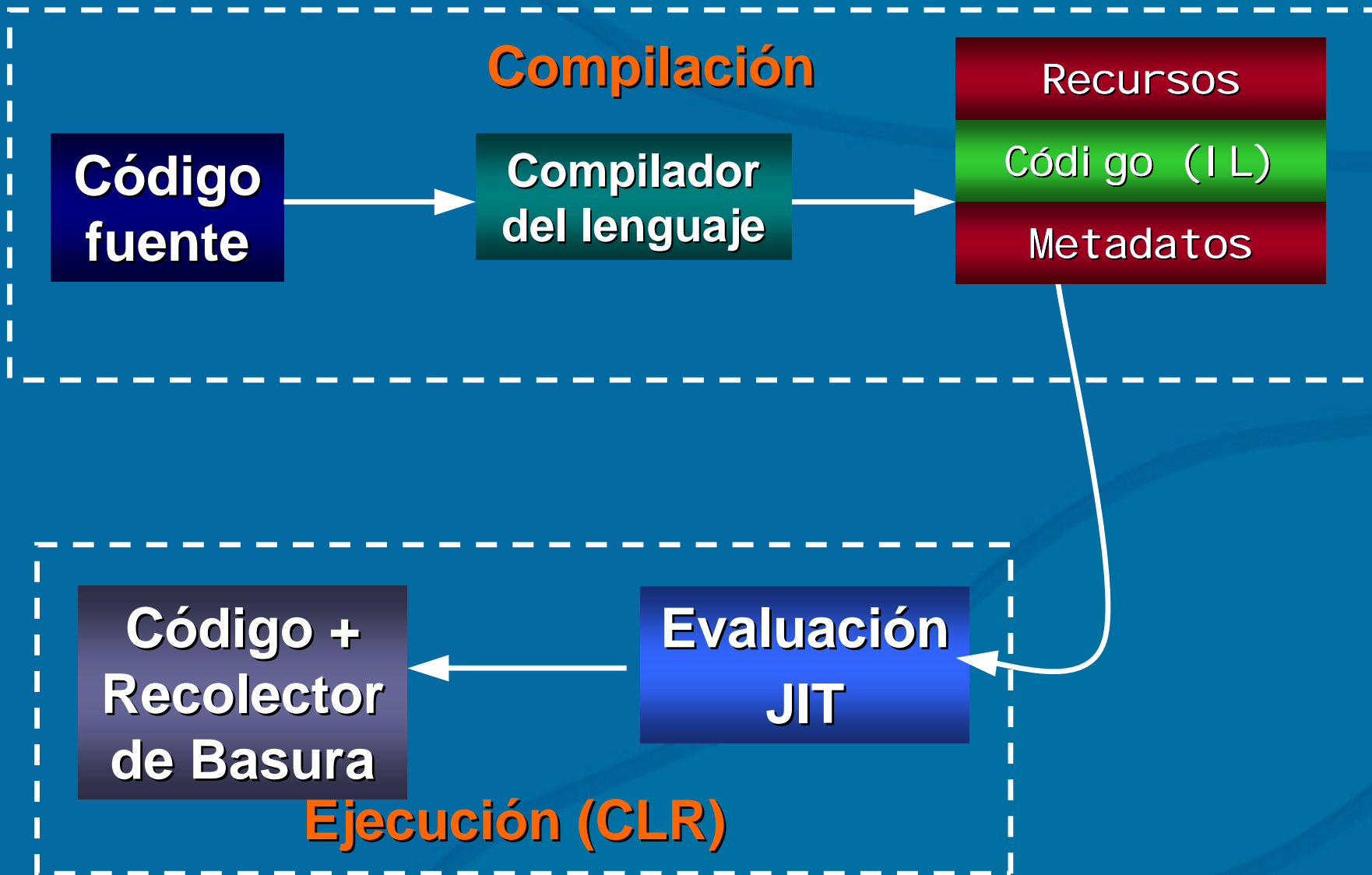
**COM suministraba un camino para integrar componentes**

# La evolución de .NET



**Con .NET Framework y CLR  
todos los componentes  
tienen un substrato común**

# Compilación y Ejecución





# Desarrollos más simplificados ...

- No es necesario conocer COM
- No más...
  - Register => aplicaciones auto descritas
  - GUIDs => namespaces jerarquicos
  - .IDL files => modelo de objetos unificado
  - HRESULTs => manejo de excepciones
  - IUnknown => Todo hereda de object
  - AddRef/Liberación => Recolector de basura
  - CoCreateInstance => operador "new"

# Desarrollo simplificado

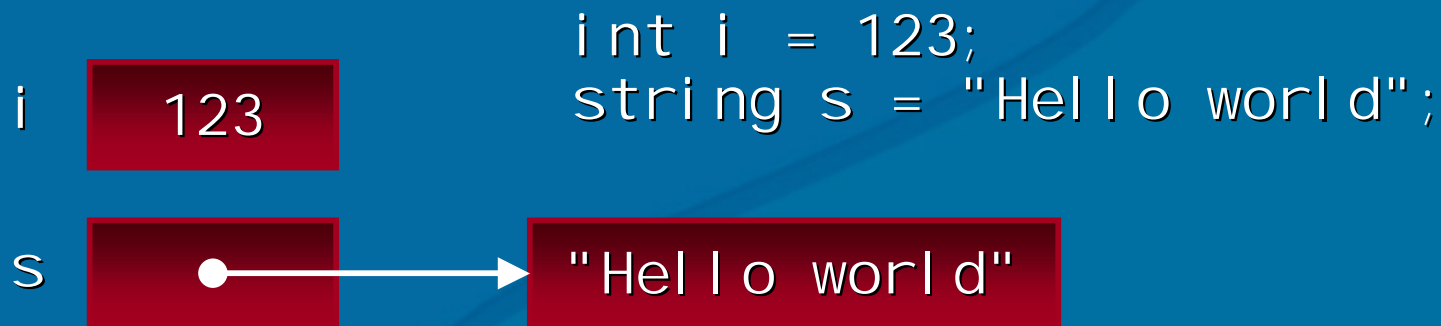
- Common Type System
  - Definición de tipos e instancias comunes
- Establece una POO más límpia
  - Clases and interfaces
  - Constructores, properties, methods, events
  - Herencia entre lenguaje
- Compatibilidad hacia atrás
  - con COM
  - con DLLs nativas (Win32<sup>®</sup>)

# Cualquier cosa es un objeto

- Cambia la visión tradicional de los tipos primitivos
  - C++, Java™
  - Smalltalk, Lisp
- La solución de .NET: tipos valor
  - Unifica sin costes de rendimiento
  - Simplicidad a través de la jerarquía de clases
- Mejora la extensibilidad y la reusabilidad
  - Nuevos tipos primitivos: Decimal, SQL...
  - Collections, etc., trabajan con todos los tipos

# Tipos Valor y Referencia

- Tipos Valor
  - Las variables directamente contienen datos
  - No pueden ser null
- Tipos Referencia
  - Las variables contienen referencias a objetos
  - Pueden ser null



# Tipos Valor y Referencia

- Tipos Valor

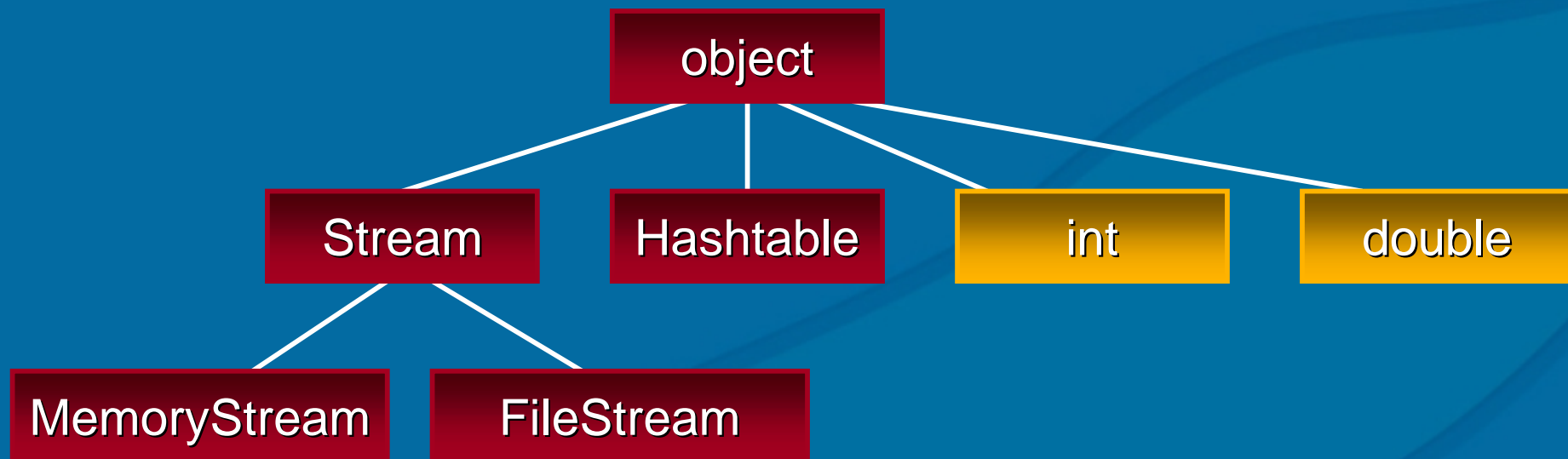
- Primitivos `int i; double x;`
- Enums `enum State { Off, On }`
- Structs `struct Point { int x, y; }`

- Tipos Referencia

- Clases `class Foo: Bar, IFoo {...}`
- Interfaces `interface IFoo: IBar {...}`
- Arrays `Foo[] a = new Foo[10];`
- Delegados `delegate void Empty();`

# Sistema de tipos unificado

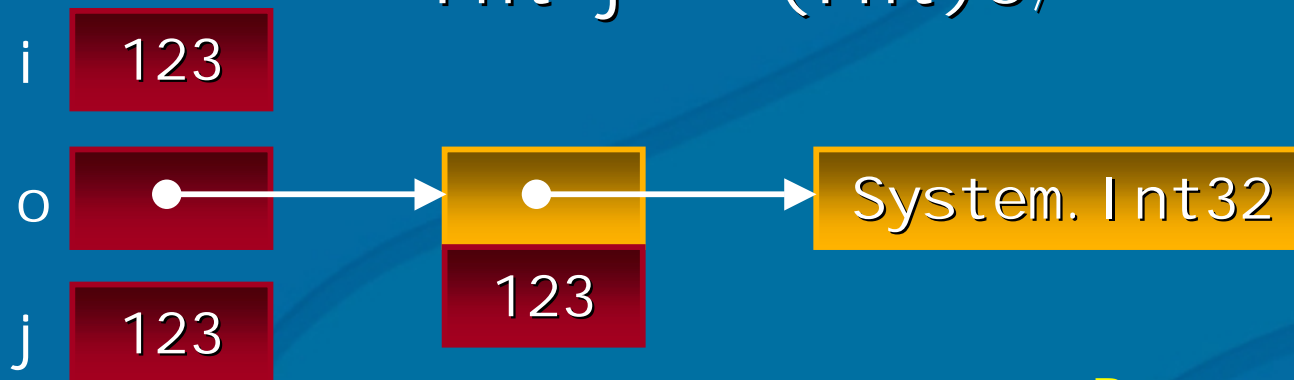
- Todos los tipos heredan de object
- Existe una conversión implícita de cualquier tipo a tipo object



# Sistema de tipos unificado

- Boxing
  - Asigna una posición de memoria y copia el valor en ella
- Unboxing
  - Comprueba el tipo de la posición de memoria y copia el valor en otra posición.

```
int i = 123;  
object o = i;  
int j = (int)o;
```

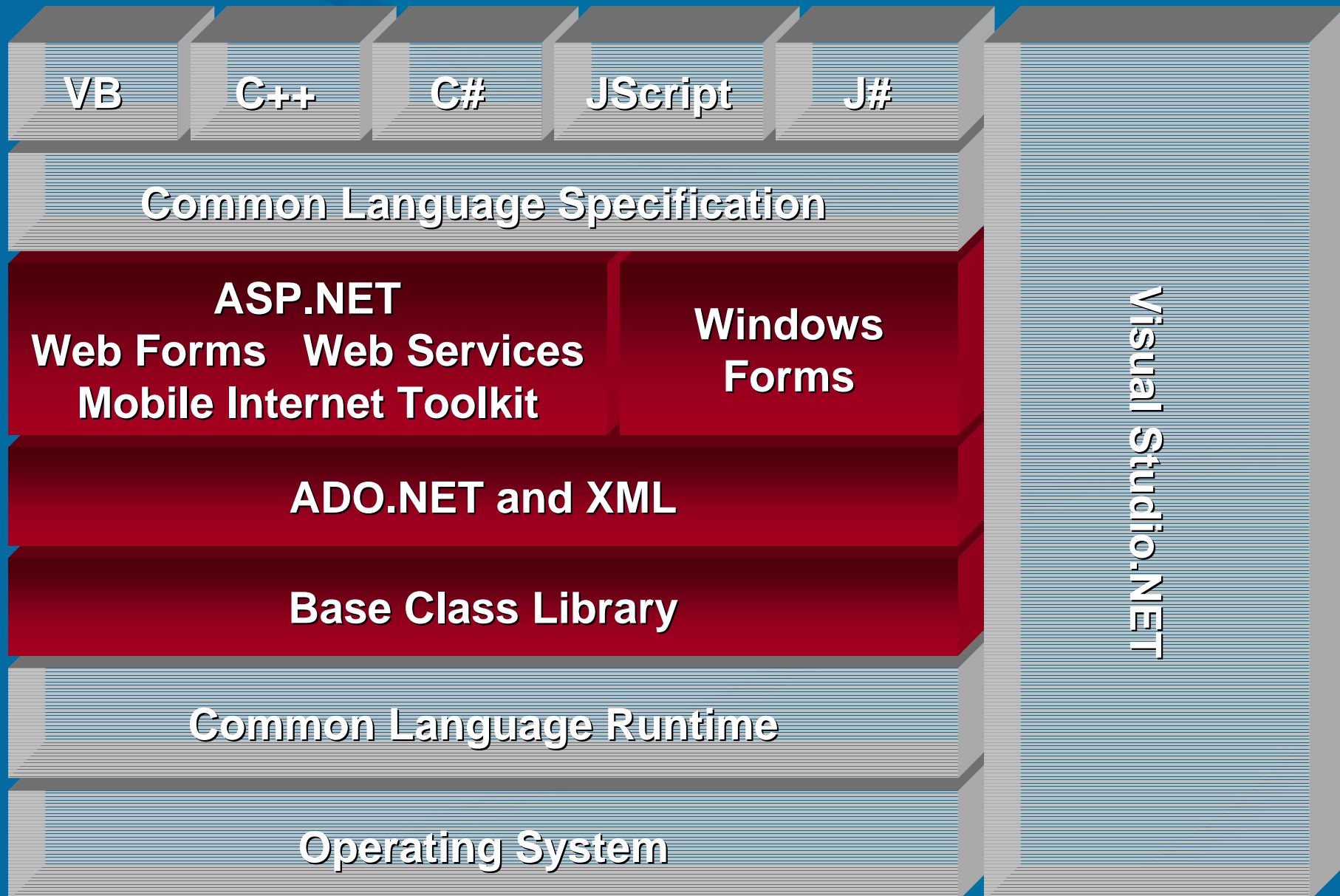


# Entorno robusto

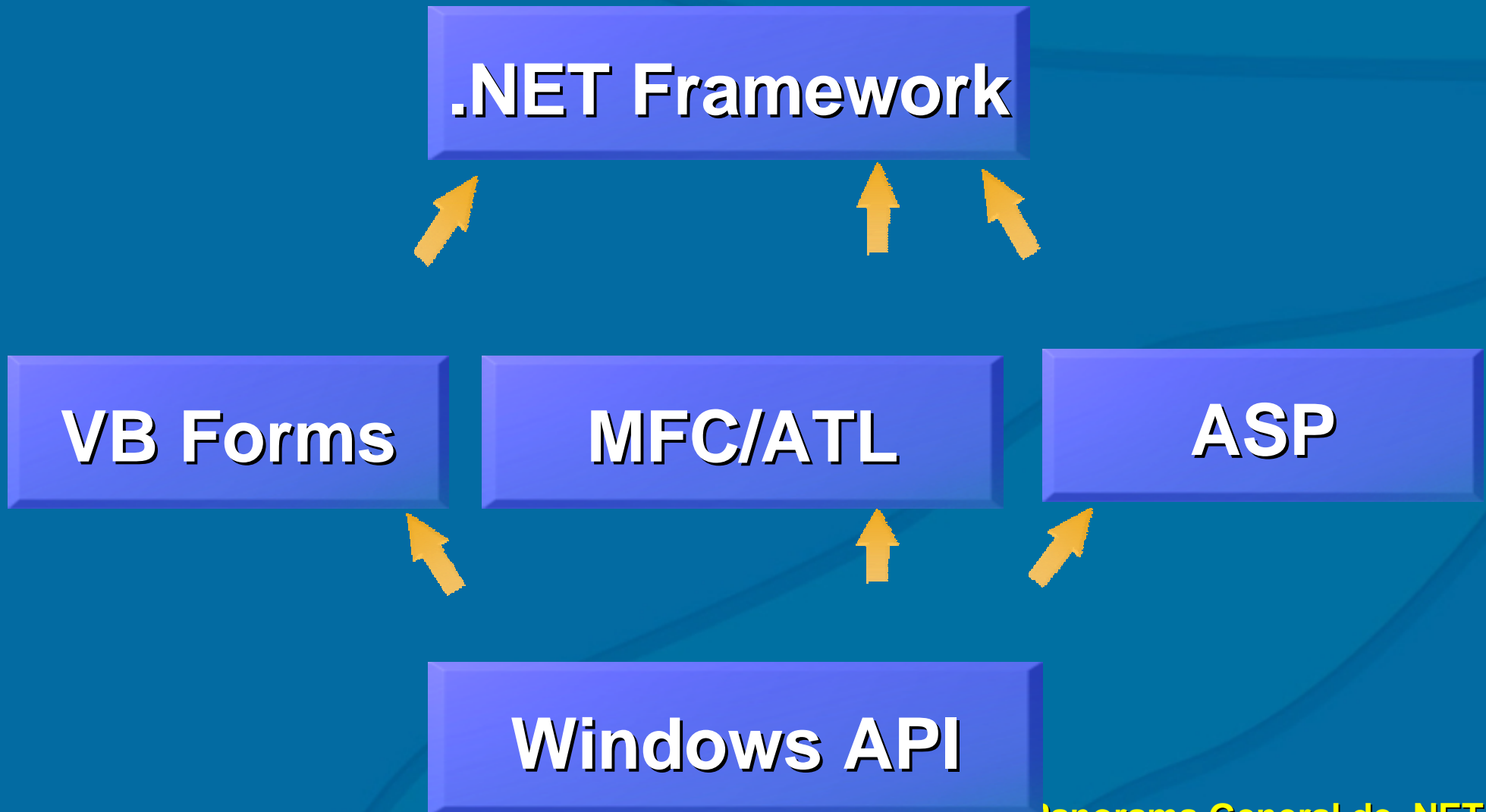
- Gestión de memoria automática
  - Todos los objetos están manejados por el recolector de basura
- Manejo de excepciones
- Seguridad en el manejo de tipos
  - No hay cast inseguros



# Framework, Lenguajes y herramientas



# Modelo de programación unificado



# Simplicidad

## Windows API

```
HWND hwndMain = CreateWindowEx(  
    0, "MainWClass", "Main Window",  
    WS_OVERLAPPEDWINDOW | WS_HSCROLL | WS_VSCROLL,  
    CW_USEDEFAULT, CW_USEDEFAULT,  
    CW_USEDEFAULT, CW_USEDEFAULT,  
    (HWND)NULL, (HMENU)NULL, hInstance, NULL);  
ShowWindow(hwndMain, SW_SHOWDEFAULT);  
UpdateWindow(hwndMain);
```

## .NET Framework (VB.NET)

```
Dim form As New Form()  
form.Text = "Main Window"  
form.Show()
```

# La biblioteca .NET Framework

## ASP.NET

Web Forms Web Services  
Mobile Internet Toolkit

## Windows Forms

## ADO.NET and XML

## Base Class Library

# La biblioteca .NET Framework

## System.Web

Services

Description

Discovery

Protocols

Caching

Configuration

UI

HtmlControls

WebControls

Security

SessionState

## System.Windows.Forms

Design

ComponentModel

## System.Drawing

Drawing2D

Imaging

Printing

Text

## System.Data

OleDb

Common

SqlClient

SQLTypes

## System.Xml

XSLT

XPath

Serialization

## System

Collections

Configuration

Diagnostics

Globalization

IO

Net

Reflection

Resources

Security

ServiceProcess

Text

Threading

Runtime

InteropServices

Remoting

Serialization

# Framework Base

## System

Collections

Configuration

Diagnostics

Globalization

IO

Net

Reflection

Resources

Security

ServiceProcess

Text

Threading

Runtime

InteropServices

Remoting

Serialization

# Acceso a Datos y XML

## System.Data

OleDb

SQLClient

Common

SQLTypes

## System.Xml

XSLT

Serialization

XPath

# ASP.NET

## System.Web

### Services

Description

Discovery

Protocols

Caching

Configuration

### UI

HtmlControls

WebControls

Security

SessionState



# Windows® Forms

## System.Windows.Forms

Design

ComponentModel

## System.Drawing

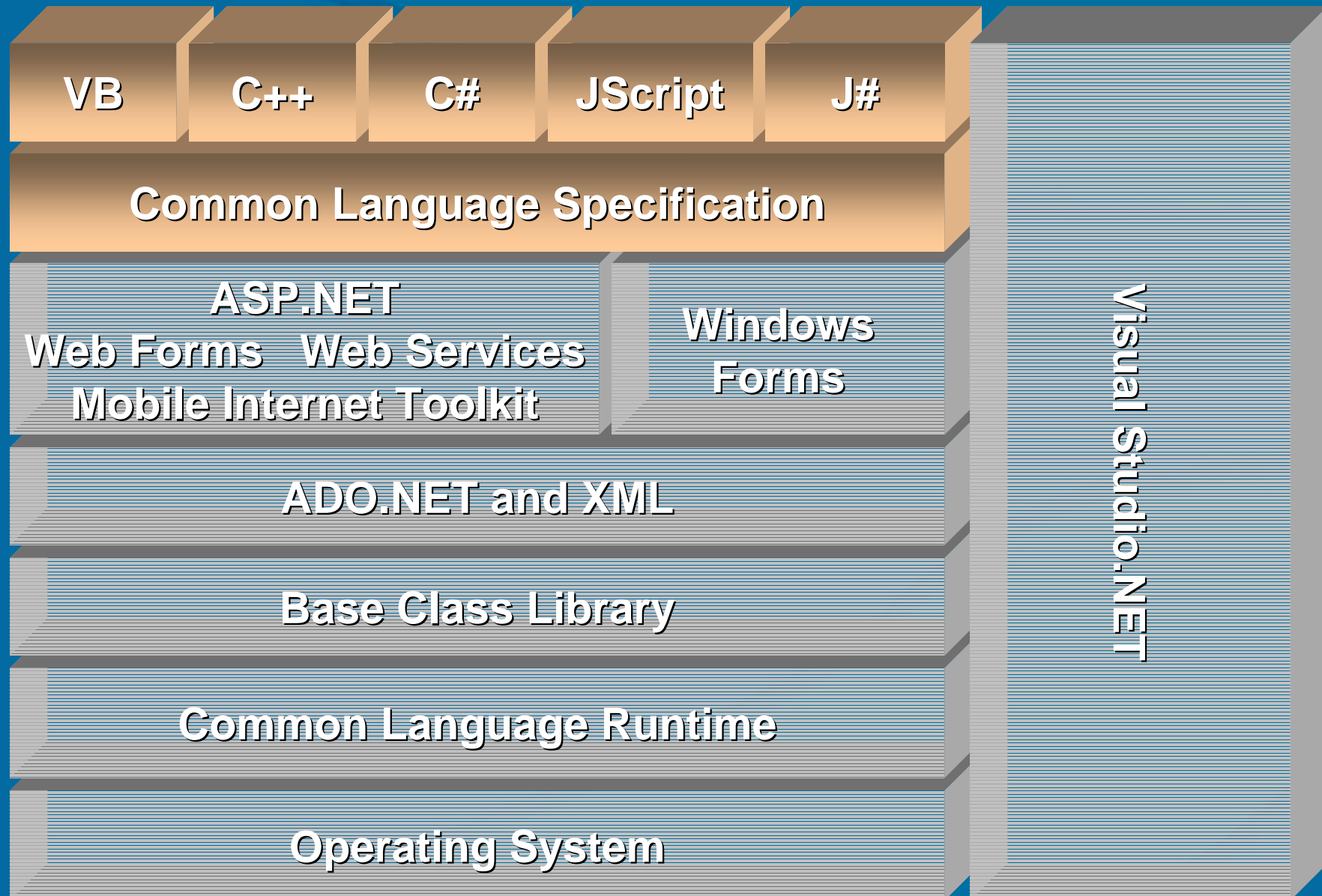
Drawing2D

Printing

Imaging

Text

# Framework, Lenguajes y herramientas



# Lenguajes

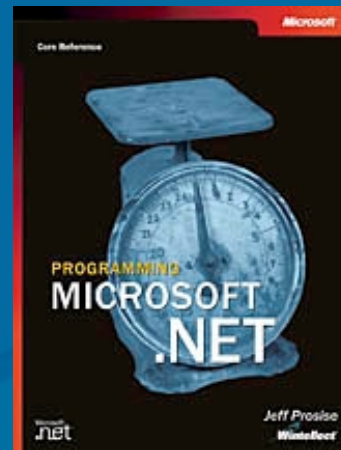
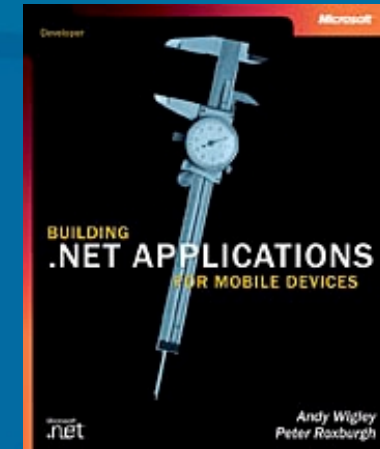
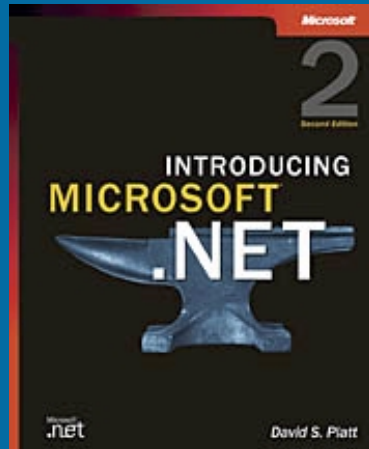
- La Plataforma .NET Platform es neutral respecto a los lenguajes
  - Todos los lenguajes de .NET tienen los mismos privilegios
- Common language specification
  - Hay un conjunto de características que se garantizan en todos los lenguajes
- Microsoft ha desarrollado
  - Visual Basic<sup>®</sup>, C++, C#, J#, JScript<sup>®</sup>
- Otros fabricantes
  - APL, COBOL, Delphi, Pascal, Eiffel, Haskell, ML, Oberon, Perl, Python, Scheme, Smalltalk...

# Estandarización

- Un subconjunto de CLR, .NET Framework and C# estandarizado por ECMA
  - También estandarizado por ISO
  - Copatrocinado por Intel, Hewlett-Packard
- El subconjunto del CLR se denomina Common language infrastructure (CLI)

# Recursos de Microsoft Press

## MICROSOFT .NET FRAMEWORK



# Discusión y preguntas

