

Introducción .NET Framework

Panello Fausto

Objetivo

- Presentar una introducción a la plataforma de desarrollo Microsoft .NET, describiendo sus principios básicos de funcionamiento, su arquitectura de componentes y sus principales bibliotecas reutilizables, mostrando además las novedades introducidas en la última versión de la misma.

Temas a Tratar

- Introducción a Microsoft .NET
- Componentes Fundamentales
- Funcionamiento Interno del CLR
- Bibliotecas Principales

Temas a Tratar

- Introducción a Microsoft .NET
 - ¿Qué no es .NET?
 - ¿Qué es .NET?
 - .NET Como evolución de COM

Paradigmas de Programación



¿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

¿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
- La evolución de la plataforma COM

Características de .NET (1/2)

- Plataforma de ejecución intermedia
- 100% Orientada a Objetos
- Multilenguaje
- Plataforma Empresarial de Misión Crítica

Características de .NET (2/2)

- Modelo de Programación único para todo tipo de aplicaciones y dispositivos de hardware
- Se integra fácilmente con aplicaciones existentes desarrolladas en plataformas Microsoft
- Se integra fácilmente con aplicaciones desarrolladas en otras plataformas
- Usa estándares globales (XML, HTTP, SOAP, WSDL, UDDI)

Plataforma de Ejecución Intermedia



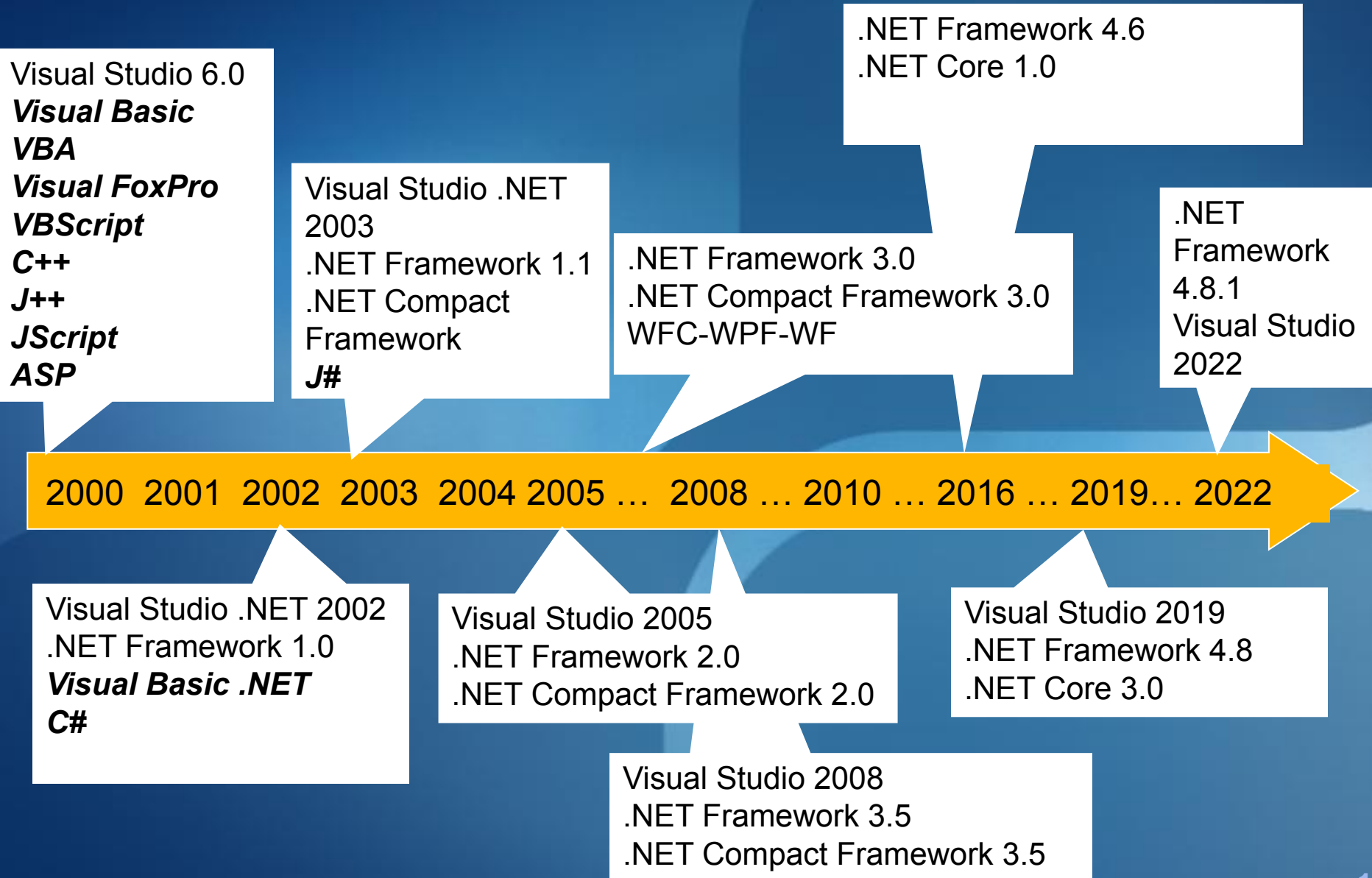
.NET como evolución de COM

- 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)

¿Qué es el .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

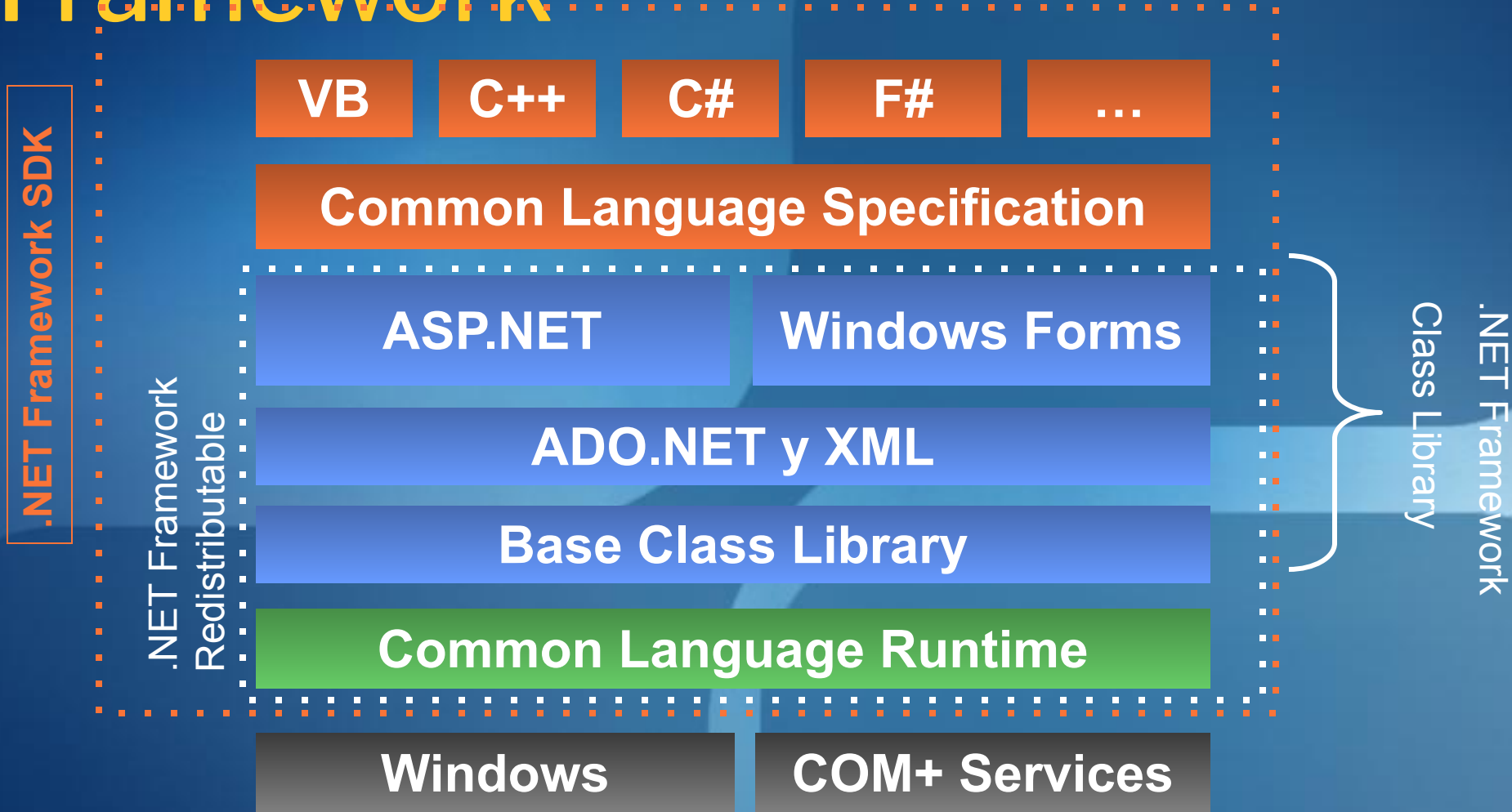
Línea del tiempo de .NET



Temas a Tratar

- Introducción a Microsoft .NET
- Componentes Fundamentales
 - Arquitectura
 - Common Language Runtime (CLR)
 - Microsoft Intermediate Language
 - Assemblies
 - .NET Class Library
 - Common Language Specification (CLS)

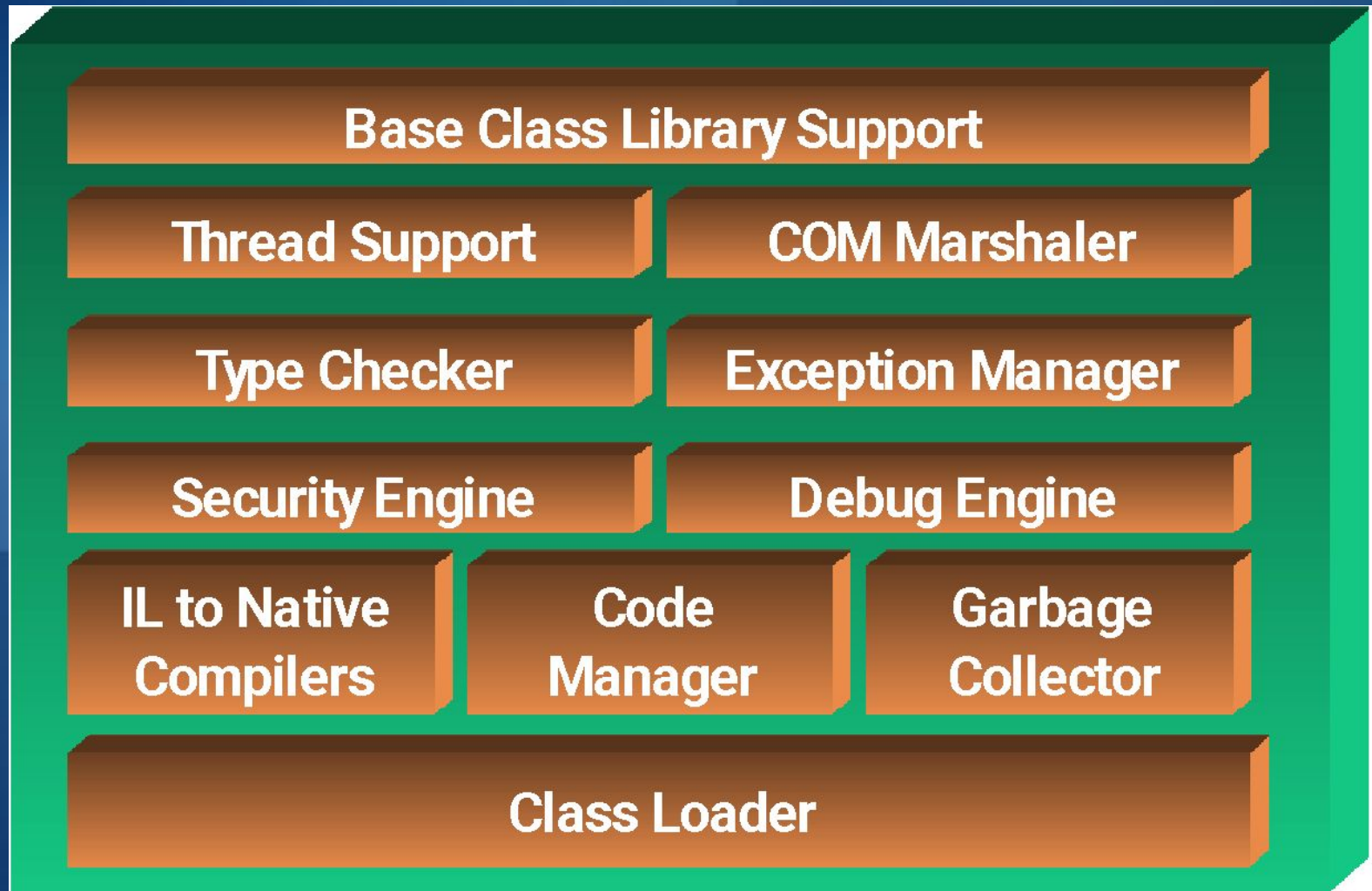
Arquitectura del .NET Framework



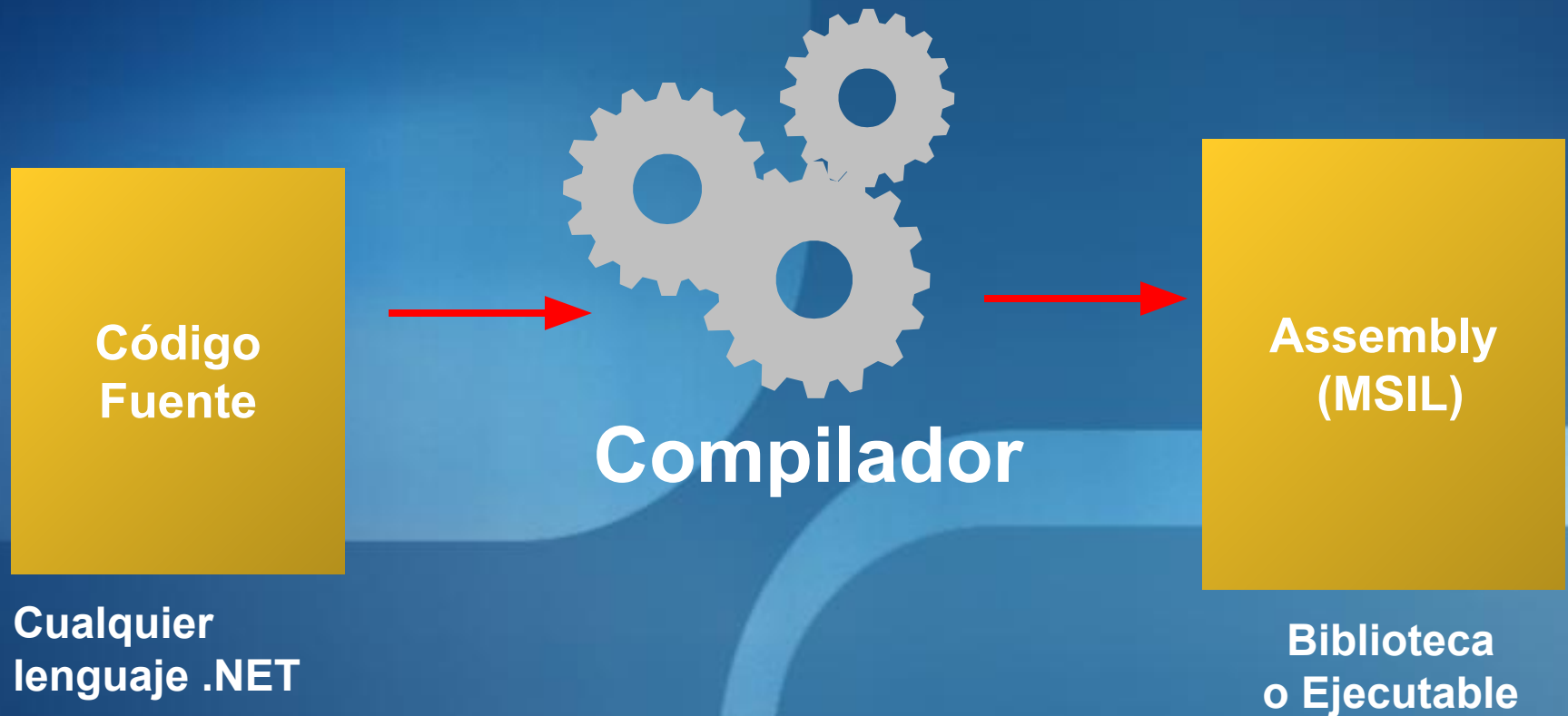
CLR – Common Language Runtime

- 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

CLR – Componentes Internos



CLR – Proceso de Compilación



CLR - MSIL

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

¿Qué es un “Assembly”?

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



Descripción de Tipos

Clases
Clases Base
Interfaces Implementadas
Atributos de las Clases
Métodos de las Clases

Manifiesto del Assembly

Nombre
Versión
Cultura

Otros Assemblies
Permisos de Seguridad
Tipos Externos

Assemblies - Aplicaciones .NET

- Uno o más Assemblies
- Al ejecutar una aplicación, ¿cómo ubico los assemblies necesarios?
 - Global Assembly Cache (GAC)
 - El Class Loader busca en el directorio local (preferido)
- Diferentes aplicaciones pueden usar diferentes versiones
 - Actualizaciones más simples
 - Desinstalación más simple

.NET Framework Class Library

- Conjunto de Tipos básicos (clases, interfaces, etc.) que vienen incluidos 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

.NET Framework Class Library

- El namespace raíz es **SYSTEM**

System.Web

Services

Description

Discovery

Protocols

UI

HtmlControls

WebControls

Caching

Configuration

Security

SessionState

System.Windows.Forms

Design

ComponentModel

System.Drawing

Drawing2D

Imaging

Printing

Text

System.Data

OleDb

Common

Odbc

SqlClient

System.Xml

XSLT

XPath

Serialization

System

Collections

Configuration

Diagnostics

Globalization

IO

Net

Reflection

Resources

Security

ServiceProcess

Text

Threading

Runtime

InteropServices

Remoting

Serialization

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 F#.NET
 - Microsoft Visual C++.NET

Common Language Specification (CLS)

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

PHP
C++.NET Visual C# F#
Basic.NET
Delphi Java Perl Python JavaScri
Pascal Haskell LISP Prolog RPG^{pt}
Oberon Mondrian Smalltalk Eiffel ML Scheme
Cobol Fortran AP Objective Caml Mercury
L

CLS - 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

Temas a Tratar

- Introducción a Microsoft .NET
- Componentes Fundamentales
- **Funcionamiento Interno del CLR**
 - Especificación CLI
 - Modelo de Ejecución
 - Application Domains
 - Common Type System

Infraestructura de Lenguaje Común (CLI)

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

En una frase...

- .NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

Sub-Especificaciones de 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)

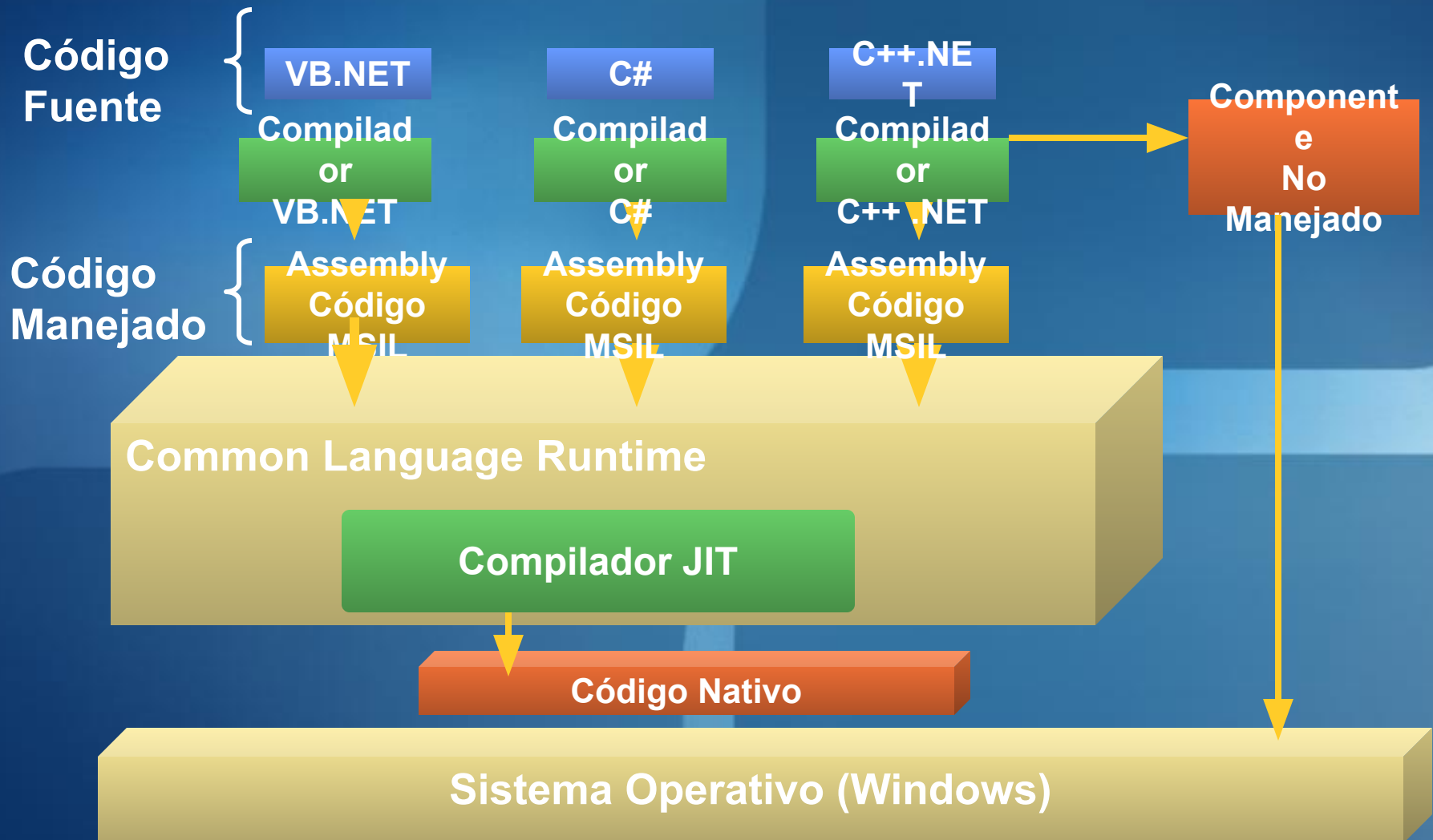
y se ejecutan bajo el control de y usan los servicios del...

CLR (Common Language Runtime)

que está acoplado al y utiliza los servicios del ...

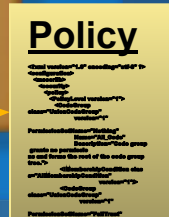
Sistema Operativo

Modelo de Ejecución del CLR



CLIR

Instalación



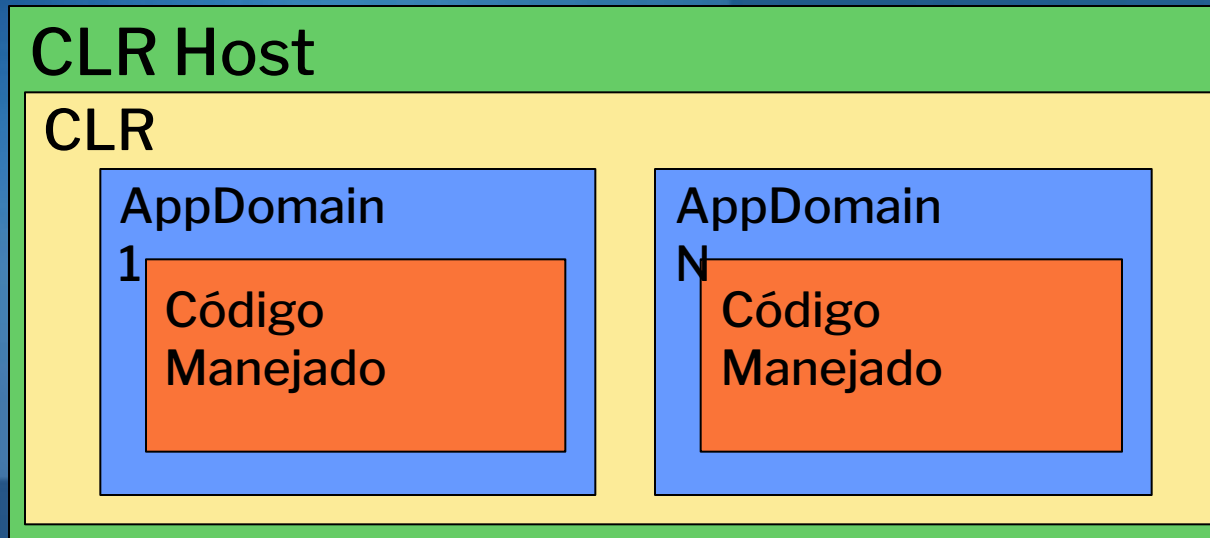
A diagram showing a yellow arrow pointing from a blue oval labeled "Source File" to a purple oval labeled "Assembly Loader".



Application Domains

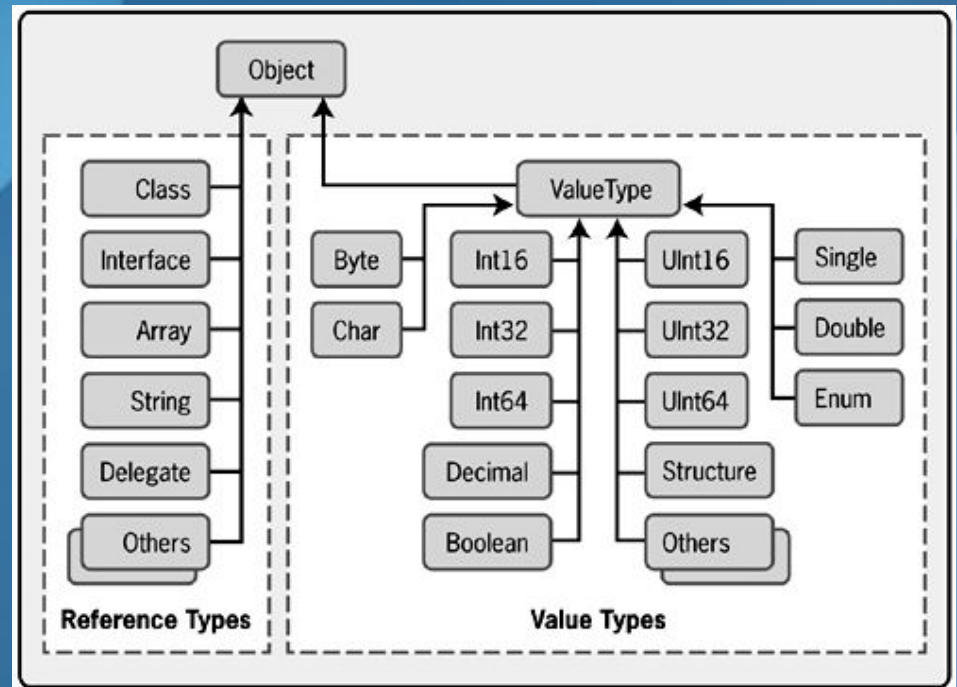
- Procesos virtuales dentro del CLR
 - Se ejecutan dentro de un proceso del Sistema Operativo
 - Un proceso del sistema operativo puede contener varios AppDomains
 - Más eficiente que múltiples procesos del sistema operativo
 - Más eficiente en el intercambio de contexto de ejecución
- Un Assembly y sus tipos son siempre cargados dentro de un AppDomain
- Provee una frontera para: Fallos, Tipos, Seguridad

Application Domains - CLR Host



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



Temas a Tratar

- Introducción a Microsoft .NET
- Componentes Fundamentales
- Funcionamiento Interno del CLR
- **Bibliotecas Principales**
 - Base Class Library (BCL)
 - ADO.NET
 - Windows Forms
 - ASP.NET

Base Class Library

System

Collections

Configuration

Diagnostics

Globalization

IO

Net

Reflection

Resources

Security

ServiceProcess

Text

Threading

InteropServices

Remoting

Serialization

Acceso a Datos: ADO.NET

System.Data

Common

SqlClient

OracleClient

OleDb

Odbc

SqlTypes

System.Xml

XSLT

Serialization

XPath

Schema

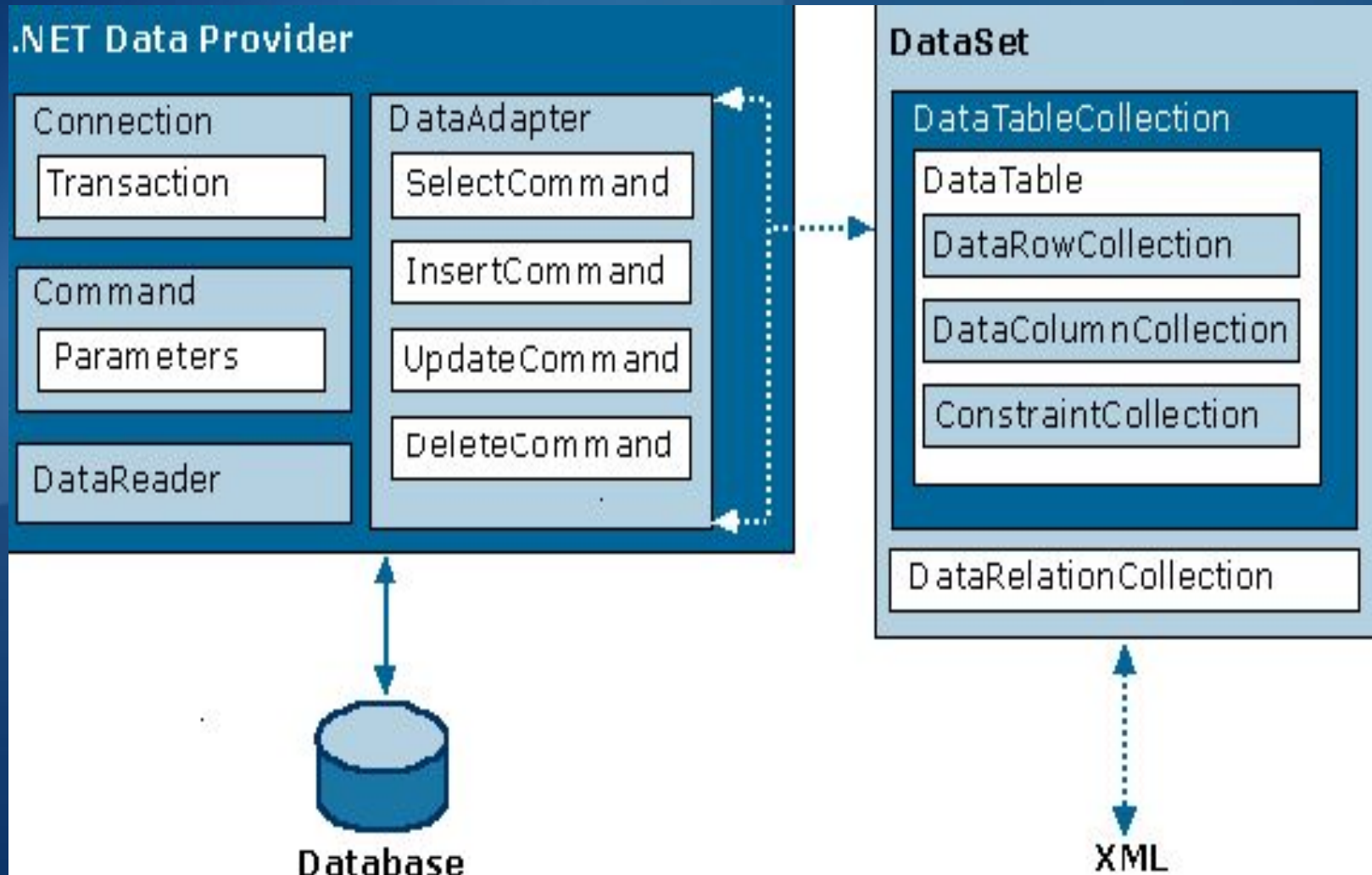
Acceso a Bases de Datos Relacionales Escenario Conectado

- Un entorno conectado es uno en el cual los usuarios están constantemente conectados a la fuente de datos
- Ventajas:
 - Mayor seguridad
 - Mejor control de concurrencia
 - Los datos se mantienen actualizados
- Desventajas:
 - Se requiere una conexión constante (consume recursos del servidor)
 - Escalabilidad

Acceso a Bases de Datos Relacionales Escenario Desconectado

- En un entorno desconectado, una parte de los datos del repositorio central se copia y modifica en forma local, para luego sincronizarse con éste.
- Ventajas
 - Se puede trabajar en forma independiente
 - Mayor escalabilidad y performance
- Desventajas
 - Los datos no están sincronizados
 - Resolución manual de conflictos

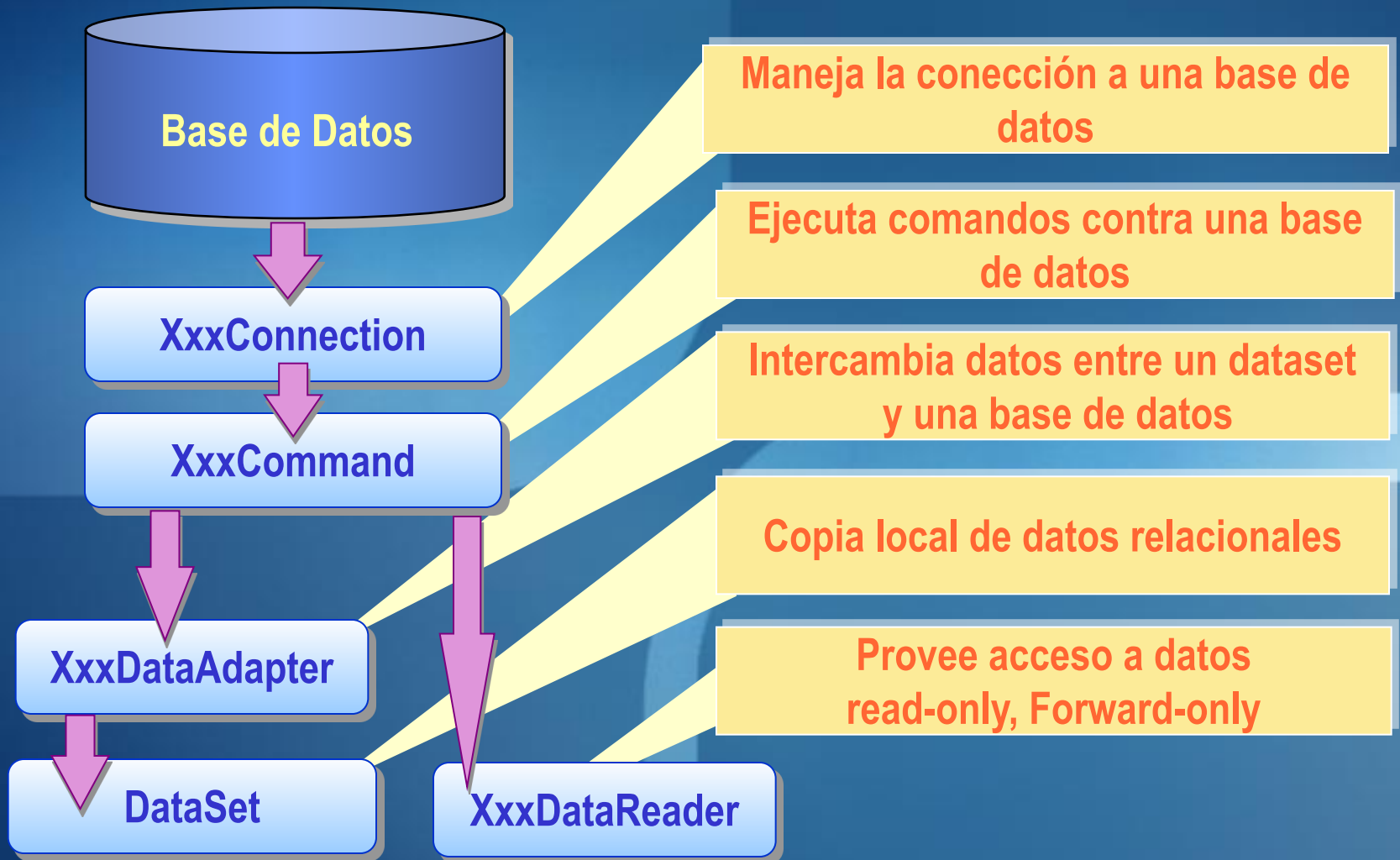
ADO.NET - Arquitectura



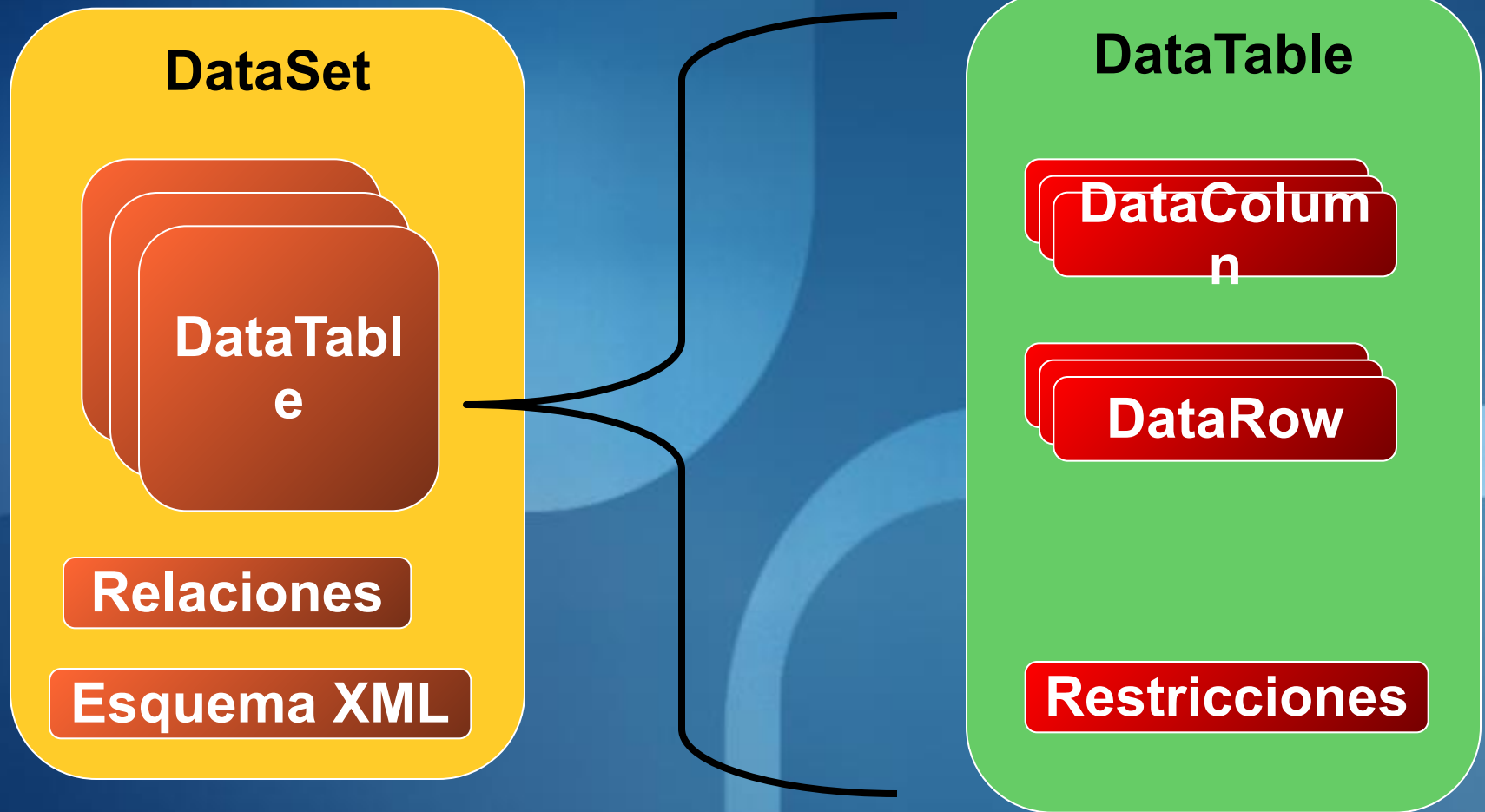
ADO.NET- Proveedores de Acceso a Datos

- SQL Server/Access (System.Data.SqlClient)
- OLE DB (System.Data.OleDb)
- ODBC (System.Data.Odbc)
- Oracle (System.Data.OracleClient)
- Otros provistos por terceros (MySQL, PostgreSQL, DB2, etc..)

ADO.NET- Clases más comunes

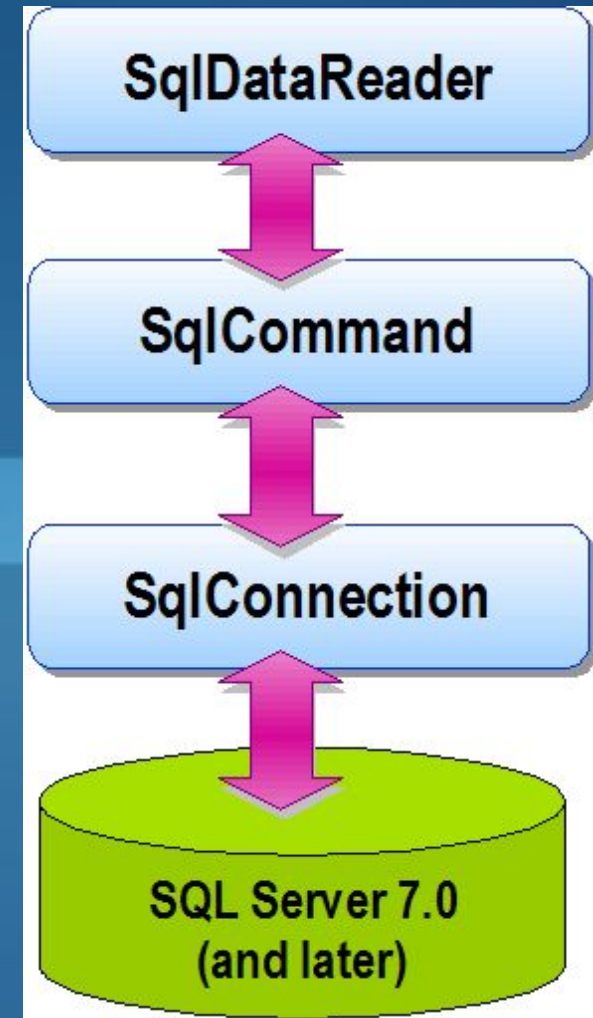


ADO.NET- DataSet



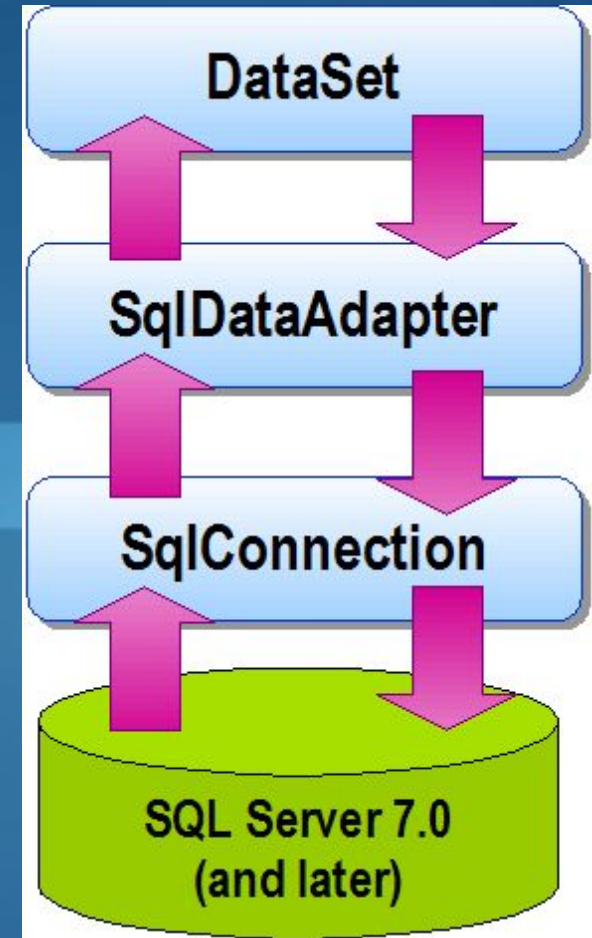
ADO.NET – Accediendo a datos Conectado

- En un escenario conectado, los recursos se mantienen en el servidor hasta que la conexión se cierra
- 1) Abrir Conexión
- 2) Ejecutar Comando
- 3) Procesar Filas en DataReader
- 4) Cerrar Reader
- 5) Cerrar Conexión



ADO.NET – Accediendo a datos Desconectado

- En un escenario desconectado, los recursos no se mantienen en el servidor mientras los datos se procesan
- 1) Abrir Conexión
- 2) Llenar DataSet mediante DataAdapter
- 3) Cerrar Conexión
- 4) Procesar DataSet
- 5) Abrir Conexión
- 6) Actualizar fuente de datos mediante DataAdapter
- 7) Cerrar Conexión



.NET Framework Class Library

XML

Conceptos de
Aplicación

Datos

Esquemas

Servicios

Invocaciones

.NET Framework facilita
Un mapeo bi-direccional

Web

Framework

XML

XSD

WSDL

SOAP

Objetos

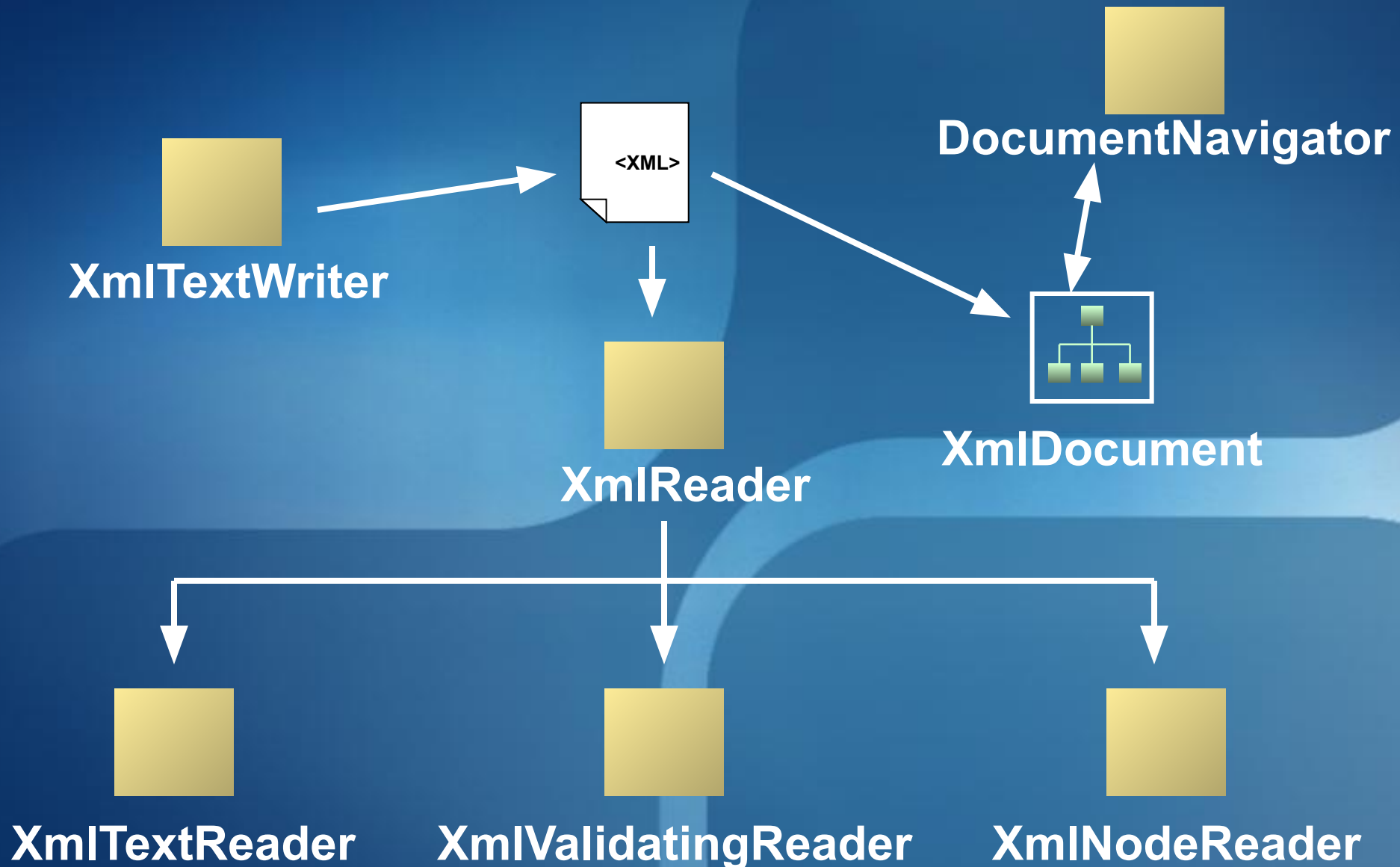
Clases

Métodos

Llamadas



ADO.NET - Soporte a XML



Aplicaciones Web: ASP.NET

System.Web

Services

Description

Discovery

Protocols

Caching

Configuration

UI

HtmlControls

WebControls

Security

SessionState

Proceso ASP.NET

Proceso de peticiones

