

# Introducción a Microsoft .NET

## Contenido

Notas generales	1
Lección: Introducción a la plataforma .NET	2
Lección: Introducción a servicios Web XML	14
Lección: Dentro de .NET Framework	20
Lección: Compilación y ejecución en .NET	25

## Notas para el instructor

Este módulo ofrece a los estudiantes una introducción a Microsoft® .NET. El módulo identifica problemas de las tecnologías actuales que se pueden resolver con la iniciativa .NET, y describe los servicios y prestaciones básicas que ofrece la plataforma .NET. Describe también las características y ventajas de los servicios Web XML y los clientes con los que es compatible la plataforma .NET. Los estudiantes aprenderán las ventajas y los componentes de .NET Framework.

Al final de este módulo, los estudiantes serán capaces de:

- Identificar los requisitos de empresa que satisface la iniciativa .NET.
- Describir las características y servicios básicos de la plataforma .NET.
- Identificar las características y ventajas de los servicios Web XML.
- Identificar clientes válidos para la plataforma .NET y los distintos tipos de dispositivos inteligentes compatibles con .NET.
- Identificar las ventajas de .NET Framework.
- Describir los componentes de .NET Framework.
- Describir el proceso de compilación y ejecución en .NET.
- Identificar las formas en que .NET simplifica la programación con respecto a otras plataformas.

## ◆ Notas generales

- Introducción a la plataforma .NET
- Introducción a servicios Web XML
- Dentro de .NET Framework
- Compilación y ejecución en .NET
- Guía básica 2002-04 para Microsoft Developer Tools

---

### Introducción

Microsoft® .NET es una plataforma que le proporciona todas las herramientas y tecnologías necesarias para crear aplicaciones Web distribuidas que utilizan protocolos Web estándar. Estas aplicaciones pueden comunicarse con una gran variedad de clientes, como teléfonos móviles y PCs de bolsillo. La plataforma .NET ofrece también un nivel de integración sin precedentes entre lenguajes de programación, además de distintos servicios en tiempo de ejecución.

En este módulo conocerá la plataforma .NET de Microsoft para servicios Web con lenguaje de marcado extensible (XML).

### Objetivos

Al final de este módulo, usted será capaz de:

- Identificar los requisitos de empresa que satisface la iniciativa .NET.
- Describir las características y servicios básicos de la plataforma .NET.
- Identificar las características y ventajas de los servicios Web XML.
- Identificar clientes válidos para la plataforma .NET y los distintos tipos de dispositivos inteligentes compatibles con .NET.
- Identificar las ventajas de .NET Framework.
- Describir los componentes de .NET Framework.
- Describir el proceso de compilación y ejecución en .NET.
- Identificar las formas en que .NET simplifica la programación con respecto a otras plataformas.

## ◆ Lección: Introducción a la plataforma .NET

- Por qué necesita .NET
- La plataforma .NET
- .NET Framework
- .NET Enterprise Servers
- Servicios de base (Building Block Services)
- Clientes para la plataforma .NET

---

### Introducción

Esta lección presenta los componentes de la plataforma Microsoft .NET, que está pensada para simplificar el desarrollo Web.

### Objetivos de la lección

Al final de esta lección, usted será capaz de:

- Identificar problemas de las tecnologías actuales que se pueden resolver con la iniciativa .NET.
- Describir los .NET Enterprise Servers.
- Definir los servicios de base (building block services) de .NET Framework.
- Identificar los clientes que se pueden utilizar en la plataforma .NET.

## Por qué necesita .NET

### ■ Microsoft .NET:

- Resuelve muchos problemas de la tecnología actual
- Proporciona aplicaciones compatibles con servicios Web XML interoperables
- Proporciona aplicaciones compatibles con clientes inteligentes

---

#### Introducción

Microsoft .NET está pensado para facilitar el trabajo con la Internet mediante la integración de estándares de Internet y la compatibilidad con servicios Web XML. La plataforma .NET también le permite acceder más fácilmente a sus datos y escribir aplicaciones Web que se pueden ejecutar en una gran variedad de exploradores y dispositivos. El tiempo de ejecución de .NET simplifica los procesos de instalación y actualización, aumentando la facilidad de uso y la fiabilidad.

#### Problemas de las tecnologías actuales

Las aplicaciones Web de hoy en día tienen interfaces de usuario interactivas creadas con HTML dinámico (DHTML) y tecnología Microsoft ActiveX®. El uso de servicios COM+ proporciona escalabilidad de los servicios de fondo. Aunque la mayor parte de estas aplicaciones se comunican de forma eficaz, muchas de ellas pueden presentar los siguientes problemas:

- Cada aplicación es un código diferenciado en un gran número de dispositivos.
- Las soluciones están formadas por grandes grupos de aplicaciones, dispositivos y servicios.
- La integración, si se consigue, resulta demasiado complicada, demasiado cara, demasiado lenta y demasiado rígida.

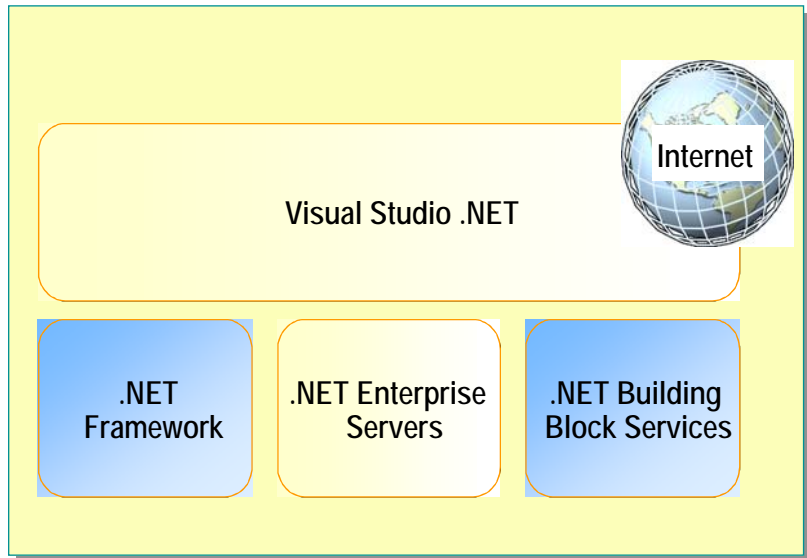
#### Servicios Web XML interoperables

La interoperabilidad es la clave para la próxima generación de aplicaciones. Las organizaciones virtuales necesitan aplicaciones Web que sean capaces de interactuar con otras aplicaciones, presentando una interfaz programable que reaccione ante otros servicios en la Web. Líneas aéreas, hoteles y empresas de alquiler de coches, por ejemplo, pueden ofrecer sus servicios como servicios Web XML, de manera que una agencia de viajes pueda emplearlos para encontrar la mejor combinación de precios y servicios para sus clientes.

**Compatibilidad con clientes inteligentes**

La tecnología también ha avanzado con gran rapidez, lo que ha provocado una reducción en el coste de hardware informático y mejores conexiones con la Internet. La necesidad de acceso instantáneo a datos por parte de las empresas ha extendido el uso de dispositivos como teléfonos móviles, asistentes digitales personales (PDAs) y PCs portátiles. Las aplicaciones desarrolladas para .NET aprovechan los últimos avances tecnológicos porque son compatibles con clientes inteligentes y se pueden ejecutar sobre distintos sistemas operativos.

## La plataforma .NET



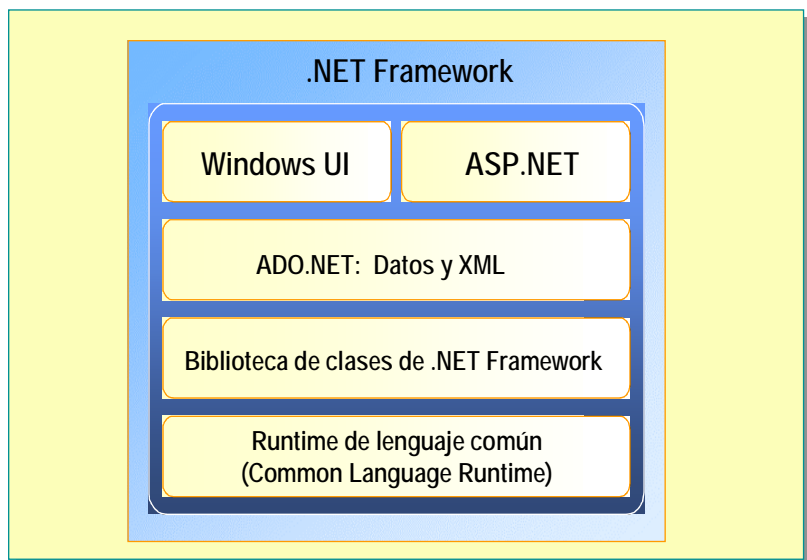
### Introducción

Microsoft .NET está formado por las siguientes tecnologías básicas:

- .NET Framework
- .NET Enterprise Servers
- Servicios de base (Building Block Services)

Microsoft Visual Studio® .NET ofrece un entorno de desarrollo de alto nivel para la creación de aplicaciones en .NET Framework. Utiliza tecnologías fundamentales para simplificar la creación, instalación y evolución continua de aplicaciones Web y servicios Web XML que son seguros, escalables y tienen una gran disponibilidad. Estas aplicaciones se pueden instalar en distintas plataformas, incluyendo .NET Enterprise Servers, y pueden hacer uso de servicios de base .NET.

## .NET Framework



### Introducción

.NET Framework es un entorno de ejecución y biblioteca de componentes independiente del lenguaje de programación utilizado. Le permite crear aplicaciones integradas y orientadas a servicios que satisfacen las necesidades actuales de las empresas de Internet reuniendo información de una gran variedad de fuentes con las que además interactúa, independientemente de las plataformas o lenguajes que se empleen.

.NET Framework está formado por los siguientes componentes:

- Entorno común de ejecución (Common Language Runtime)
- Biblioteca de clases de .NET Framework
- Microsoft ADO.NET (datos y XML)
- Microsoft ASP.NET

### Runtime de lenguaje común (Common Language Runtime)

Common Language Runtime (CLR) simplifica el desarrollo de aplicaciones, proporciona un entorno de ejecución robusto y seguro, es compatible con gran variedad de lenguajes y simplifica la instalación y gestión de aplicaciones. También se conoce como un entorno gestionado que proporciona automáticamente servicios comunes como seguridad y recolección de elementos no utilizados.

### Biblioteca de clases de .NET Framework

.NET Framework proporciona un conjunto unificado, orientado a objetos, jerárquico y extensible de bibliotecas de clases para desarrolladores. Hasta ahora, los desarrolladores de C++ empleaban MFC (Microsoft Foundation Classes) y los de Microsoft Visual Basic® usaban clases del runtime de Visual Basic, mientras que para otros lenguajes se utilizaban sus propias bibliotecas de clases y frameworks. Básicamente, .NET Framework unifica los distintos frameworks ofrecidos anteriormente por Microsoft. El resultado es que los desarrolladores ya no necesitan conocer distintos frameworks para hacer su trabajo.



Por otra parte, la creación de un conjunto de interfaces de programación de aplicaciones (API) común para todos los lenguajes de programación en .NET Framework permite la depuración, el tratamiento de errores y la herencia de un lenguaje a otro. De esta forma todos los lenguajes de programación, desde Microsoft JScript® hasta Visual C++®, son iguales en la práctica y los desarrolladores tienen libertad para elegir el lenguaje que quieren usar.

**ADO.NET: Datos y XML**

ADO.NET es la próxima generación de la tecnología ActiveX Data Object (ADO). El elevado nivel de integración de ADO.NET en XML permite a los desarrolladores transferir conjuntos de datos (cachés de datos en memoria) entre los distintos componentes de una solución de empresa.

**ASP.NET**

ASP.NET se basa en las clases de programación de .NET Framework para proporcionar un modelo de aplicaciones Web en la forma de un conjunto de controles e infraestructura que facilita la creación de aplicaciones Web. Los desarrolladores pueden acceder a un conjunto de controles Web de ASP.NET con funciones comunes de interfaces de usuario en lenguaje de marcado de hipertexto (HTML), como cuadros de texto y menús desplegables. Estos controles se ejecutan en el servidor Web y proyectan su interfaz de usuario a un navegador como HTML.

En el servidor, los controles muestran un modelo de programación orientada a objetos que pone su potencia a disposición de los desarrolladores de Web. ASP.NET proporciona además servicios de infraestructura, como gestión de estados de sesión y reciclaje de procesos, lo que reduce aún más la cantidad de código que debe escribir el desarrollador y aumenta la fiabilidad de las aplicaciones. ASP.NET utiliza también estos mismos conceptos para hacer que los desarrolladores puedan ofrecer el software como un servicio.

Utilizando servicios Web XML, los desarrolladores pueden simplemente escribir su lógica de empresa (business logic) y dejar que la infraestructura de ASP.NET se encargue de ofrecer el servicio mediante el protocolo simple de acceso a objetos (Simple Object Access Protocol, SOAP).

## .NET Enterprise Servers

### ■ Los .NET Enterprise Servers incluyen:

- Application Center
- BizTalk Server
- Commerce Server
- Exchange Server
- Host Integration Server
- Internet Security and Acceleration Server
- SQL Server

### Introducción

Los .NET Enterprise Servers forman la completa gama de aplicaciones de servidor de Microsoft para la creación, instalación y gestión de soluciones Web escalables e integradas. Diseñados para ofrecer las más altas prestaciones, los .NET Enterprise Servers ofrecen escalabilidad, fiabilidad y facilidad de uso para las empresas globales que trabajan en la Web. Están pensados específicamente para que sean interoperables, ya que utilizan estándares abiertos de Web como XML. Esto permite a Microsoft ofrecer la infraestructura de una nueva plataforma de desarrollo que ayudará a los desarrolladores a crear e instalar rápidamente aplicaciones distribuidas en tiempo de Internet.

### Enterprise Servers

A continuación se ofrece una descripción de los distintos .NET Enterprise Servers:

- **Microsoft Windows® 2000 Server:** Windows 2000 Server aprovecha las posibilidades de la tecnología Microsoft Windows NT® para integrar servicios basados en estándares de directorios, aplicaciones Web, comunicaciones, archivos e impresión con alta fiabilidad, gestión eficaz y compatibilidad con los últimos adelantos en hardware de red. El sistema operativo para servidores de próxima generación, Windows .NET Server, ofrecerá un nivel de integración con .NET todavía más alto.
- **Microsoft Application Center:** Application Center hace que las aplicaciones Web creadas en Microsoft Windows 2000 tengan la máxima disponibilidad (99,999 por ciento de tiempo activo) mediante adaptación de software, al tiempo que reduce la complejidad y los costes de operación.
- **Microsoft BizTalk® Server:** BizTalk Server es una plataforma basada en estándares que utiliza XML como formato común de documentos para desarrollar y gestionar la integración de aplicaciones en y entre organizaciones.
- **Microsoft Commerce Server:** Commerce Server es un sistema completo que permite crear rápidamente soluciones escalables y personalizadas de

comercio electrónico Business-to-Consumer (B2C) y Business-to-Business (B2B) en Windows 2000.

- **Exchange Server:** Esta completa aplicación de mensajería y colaboración para organizaciones de todos los tamaños ofrece los mejores niveles de interoperabilidad, facilidad de uso y gestión a bajo coste.

- **Microsoft Host Integration Server:** Esta completa plataforma de integración es el mejor sistema para adoptar rápidamente tecnologías de Internet, intranet y cliente/servidor y rentabilizar al máximo las inversiones realizadas en sistemas heredados.
- **Microsoft Internet Security and Acceleration Server:** Internet Security and Acceleration (ISA) Server le ofrece conectividad a Internet segura y fácil de administrar y acelera la presentación de contenido Web por medio de una caché Web escalable y fiable.
- **Microsoft SQL Server™:** La conocida base de datos de Microsoft para Windows le ofrece la posibilidad de tomar decisiones más informadas a todos los niveles de su empresa con soluciones de empresa escalables, gran capacidad de almacenamiento de datos e integración en Microsoft Office.
- **Microsoft Mobile Information Server 2002:** Este servidor de aplicaciones extiende a los usuarios móviles el alcance de los Microsoft .NET Enterprise Servers, los datos de empresa y los contenidos de intranet. Lleva la intranet de empresa a la última generación de dispositivos móviles, lo que permite a los usuarios acceder de forma segura y en tiempo real a su correo electrónico, contactos, calendarios, tareas o a cualquier aplicación de intranet para unidades de negocio desde cualquier lugar.
- **Microsoft Windows .NET Server:** Windows .NET Server es la próxima generación de productos para servidores Windows. Estará incluido en .NET Framework y aprovecha las ventajas que ofrece la compatibilidad con XML y la escalabilidad de la actual generación de servidores Windows para aumentar el nivel de integración de Microsoft Active Directory® Directory Service y el sistema de autenticación Microsoft Passport. La siguiente versión de Windows .NET Server es Windows .NET Server 2003, un sistema operativo polivalente con capacidad para una gran variedad de servidores según las necesidades, tanto de forma centralizada como distribuida.
- **Supplier Accelerator:** Es una parte muy importante de la iniciativa E-Business Acceleration que resuelve el problema de la conexión a múltiples e-marketplaces. Ofrece un modelo inteligente para gestionar la publicación de catálogos y el procesamiento de pedidos en los principales e-marketplaces directamente con aplicaciones de compradores, y es ampliable para poder incluir nuevos canales de ventas en e-business. Supplier Accelerator está basado en Commerce Server 2000, BizTalk Server 2000 y SQL Server 2000 y es una excelente base para el comercio electrónico.

## Servicios de base (Building Block Services)

- Servicios de base (Building Block Services)
  - Permiten que el usuario controle los datos
  - Utilizan Microsoft Passport

---

### Introducción

Los servicios de base son un conjunto de servicios Web XML que traspasan de las aplicaciones a los usuarios el control sobre los datos de los usuarios. Estos servicios permiten la personalización y garantizan la coherencia en aplicaciones, servicios y dispositivos. Microsoft ha desarrollado en torno a la identidad, la notificación y el almacenamiento servicios privados y seguros que pueden servir de base para otros servicios Web XML y experiencias .NET.

Microsoft Passport es la iniciativa de Microsoft .NET para facilitar la integración de distintas aplicaciones.

### Microsoft Passport

Passport es un componente básico de la iniciativa Microsoft .NET. Permite a las empresas desarrollar y ofrecer servicios Web XML distribuidos en una amplia gama de aplicaciones, dispositivos y servicios complementarios, todos ellos basados en una experiencia de Internet común.

El servicio de inicio de sesión único (Single Sign-In, SSI) de Microsoft Passport permite a las empresas ofrecer a sus clientes un sistema rápido y práctico de iniciar una sesión y realizar transacciones de forma segura, ya que pueden emplear un solo nombre y contraseña en todos los sitios Web a los que accedan. Además, los nombres de inicio de sesión de Passport se asignan a personas y no a sistemas, lo que significa que los miembros pueden acceder a sitios Passport en cualquier momento y desde muchos tipos de dispositivos.

El uso de Passport en su sitio Web le permitirá:

- Aumentar el tráfico al simplificar el proceso de inicio de sesión y registro.
- Aumentar el nivel de retención de clientes al ofrecer contenidos personalizados a partir de datos de perfiles en Passport.
- Aumentar las ventas al simplificar el proceso de compra.
- Ofrecer a millones de miembros de Passport un sistema más fácil y seguro de inicio de sesión y compra.

## Clientes para la plataforma .NET

■ La plataforma .NET es compatible con los siguientes dispositivos inteligentes:

- PCs portátiles
- Estaciones de trabajo
- PCs de bolsillo
- PDAs
- Teléfonos móviles inteligentes
- Consolas de juegos
- Tablet PCs

---

### Introducción

Como apoyo para la plataforma .NET, Microsoft esta creando software de dispositivos que permitirá a los clientes utilizar distintos dispositivos complementarios en lugar de un solo dispositivo o cliente. Microsoft está desarrollando software para todo tipo de aparatos, desde teléfonos y PDAs hasta otros dispositivos y clientes, como PCs portátiles, estaciones de trabajo, PCs de bolsillo, teléfonos móviles inteligentes, consolas de juegos (Microsoft Xbox™) y Tablet PCs. Microsoft está trabajando actualmente en los siguientes productos de software para estos dispositivos: Embedded Windows XP, Windows CE 4.0, .NET Framework y .NET Compact Framework.

### Dispositivos inteligentes

Todos ellos reciben el nombre de *dispositivos inteligentes* porque pueden recordar quién es el usuario y utilizan la Web en lugar del servidor como plataforma para computación.

Los dispositivos inteligentes permiten acceder a datos, analizarlos y actuar sobre ellos en cualquier momento y en cualquier lugar. Utilizan la red de forma inteligente y reaccionan ante limitaciones del ancho de banda, permiten el uso de aplicaciones con o sin conexión, y saben qué servicios están disponibles.

Estos dispositivos son programables y personalizables, realizan actualizaciones automáticamente y no requieren ninguna administración.

Los dispositivos inteligentes presentan y reúnen información de la forma más adecuada para cada dispositivo, desde la conversión de texto en voz en un teléfono móvil hasta el reconocimiento de escritura a mano en un Tablet PC. Pueden utilizar servicios Web XML y descubrir qué servicios están disponibles, ya que emplean XML, SOAP y UDDI.

Los dispositivos inteligentes crecerán de forma explosiva en los próximos cinco años, a medida que se vayan integrando en el dispositivo inteligente por excelencia para Internet: el PC.

## ◆ Lección: Introducción a servicios Web XML

- Notas generales sobre servicios Web XML
- Características de los servicios Web XML
- Ventajas de los servicios Web XML

---

### Introducción

Uno de los principales desafíos a los que se enfrentan hoy en día los desarrolladores es la integración de aplicaciones; es decir, el proceso de tomar un grupo de aplicaciones y convertirlas en aplicaciones Web fáciles de usar, aunque se ejecuten en distintos sistemas operativos, estén escritas en distintos lenguajes de programación y hayan sido creadas con distintos modelos de objetos.

### Objetivos de la lección

Los servicios Web XML permiten crear aplicaciones Web de forma que distintos usuarios con diferentes plataformas puedan acceder a ellas y utilizarlas.

Al final de esta lección, usted será capaz de:

- Comprender los servicios Web XML.
- Describir las características de los servicios Web XML.
- Describir las ventajas de los servicios Web XML.



## Notas generales sobre servicios Web XML

- Los servicios Web XML utilizan los siguientes estándares:
  - XML
  - HTTP
  - SOAP

---

### Introducción

Los servicios Web XML proporcionan un mecanismo intrínseco para crear un servicio o sitio Web de manera que colabore perfectamente con otros servicios y sitios Web. XML Aprovechan la infraestructura y las aplicaciones existentes para ofrecer un modelo simple, flexible y basado en estándares para la vinculación de aplicaciones Web en la Internet.

Las aplicaciones Web se pueden ensamblar fácilmente empleando servicios desarrollados de forma local y servicios ya existentes, independientemente de las plataformas, lenguajes de programación o modelos de objetos que se hayan utilizado para crear cada servicio o aplicación.

Si, por ejemplo, su empresa está especializada en la creación de sistemas muy precisos de conversión de moneda para prácticamente todos los países del mundo, usted puede dedicarse a desarrollar servicios Web XML a los que acceden otros sitios Web para ofrecer conversiones de precios a sus usuarios. De esta forma combina distintos servicios y la experiencia del usuario en la Web es mucho más rica.

### Estándares

Los servicios Web XML utilizan lo siguientes estándares:

- **XML** fue diseñado para documentos Web y permite a los desarrolladores crear sus propias etiquetas personalizadas, lo que hace posible la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones.
- **HTTP** define el formato y el modo de transmisión de mensajes y las acciones que los servidores y exploradores Web deben realizar en respuesta a distintos comandos.
- **SOAP** permite la interoperabilidad de una gran variedad de programas y plataformas, y hace que las aplicaciones existentes sean accesibles a un mayor número de usuarios.

La especificación UDDI se utiliza para publicar y descubrir información sobre servicios Web XML. UDDI es la base que permite a las empresas buscar y hacer negocios entre sí de forma rápida, sencilla y dinámica utilizando las aplicaciones que prefieran.

## Características de los servicios Web XML

■ Los servicios Web XML se caracterizan por:

- Elementos programables
- Datos compartidos
- Interoperabilidad
- Independencia de lenguaje y plataforma
- Disponibilidad universal

---

### Introducción

Los servicios Web XML llevan a un nuevo nivel el desarrollo de aplicaciones distribuidas.

### Características de los servicios Web XML

Los servicios Web XML se pueden emplear para:

- Colocar elementos programables en sitios Web donde otros pueden acceder a comportamientos distribuidos.
- Invocar funciones de otras aplicaciones, además de hacer que las aplicaciones puedan compartir datos independientemente de las herramientas empleadas para crear esas aplicaciones.
- Ofrecer como un servicio Web XML las funciones y datos de cada una de las aplicaciones existentes. De esta forma se podría crear una aplicación compuesta que utilizara ese grupo de servicios Web XML para lograr la interoperabilidad de las aplicaciones que la forman. El resultado es que los servicios Web XML resuelven varios problemas básicos, como interoperabilidad, integración y extensibilidad de aplicaciones.
- Ser utilizados por aplicaciones creadas en cualquier lenguaje y para cualquier plataforma. Es posible invocar un servicio Web XML desde cualquier aplicación Web, incluyendo otro servicio Web XML. El cliente de un servicio Web XML no tiene que ser necesariamente una aplicación basada en clientes; en realidad, casi todos los clientes son aplicaciones basadas en servidores, como Web Forms y servicios Web XML.
- Promover la comunicación universal. Un sistema operativo que se desarrolle ahora o en el futuro incluirá sin duda la posibilidad de conectar a la Internet. Esto significa que la capacidad de los servicios Web XML para conectar prácticamente cualquier sistema o dispositivo a la Internet garantizará la disponibilidad universal de esos sistemas operativos para cualquier otro sistema o dispositivo que esté conectado a la Internet.

## Ventajas de los servicios Web XML

- El uso de servicios Web XML presenta las siguientes ventajas:
  - Una colaboración más sencilla para los usuarios
  - Integración de aplicaciones de empresa externas
  - Mayor productividad de los desarrolladores

---

### Introducción

Los servicios Web XML son un elemento integral del modelo de programación .NET. La integración Best-of-Breed (lo mejor de lo mejor) permite a Microsoft .NET convertir en una ventaja competitiva las infraestructuras existentes de tecnologías de la información. El objetivo de la iniciativa Microsoft .NET es conseguir que desarrolladores, empresas y usuarios puedan aprovechar la tecnología para acceder a información en cualquier momento, en cualquier lugar y en cualquier dispositivo.

### Ventajas

El uso de servicios Web XML presenta las siguientes ventajas:

- Da más posibilidades a los usuarios
- Permite la integración de aplicaciones de otras empresas
- Ofrece nuevas oportunidades de desarrollo

### Una colaboración más sencilla para los usuarios

La plataforma .NET utiliza XML para situar en primer plano la experiencia del usuario y dejar la tecnología en un segundo plano. Los servicios Web XML:

- Permiten a los usuarios interactuar con sus datos a través de tecnologías de visión, voz y escritura a mano.
- Datos seguros de los usuarios en la Internet para que puedan acceder a ellos desde el trabajo con sus PCs y desde casa con sus teléfonos móviles, localizadores o PDAs. Esto da a los usuarios un mayor control, capacidad para tomar mejores decisiones y una colaboración más sencilla.
- Permite ejecutar aplicaciones en una gran variedad de dispositivos. Esto hace que los usuarios puedan realizar transacciones fácilmente desde teléfonos inteligentes, PDAs y muchos otros dispositivos inteligentes.

### Ejemplo

Por ejemplo, un paciente podría usar una aplicación basada en .NET para buscar clínicas cercanas que utilicen servicios Web XML para ofrecer sus

calendarios de citas. El paciente puede encontrar de forma rápida y sencilla un servicio sanitario que satisfaga sus necesidades, mientras que las empresas que ofrecen sus principales procesos como servicios Web XML crean experiencias de usuario más personales e inteligentes y amplían las posibles interacciones de los clientes con la empresa.

#### Integración de aplicaciones de empresa

Los servicios Web XML ofrecen importantes ventajas a organizaciones y empresas, ya que aumentan sus posibilidades de alcance y exposición y crean nuevas oportunidades de negocio. Esto puede producir un ahorro considerable en los costes de desarrollo, además de generar nuevos ingresos.

Las organizaciones pueden integrar más fácilmente sus aplicaciones internas, así como acceder a servicios ofrecidos por otras empresas. La combinación de servicios Web XML ofrecidos en la Internet permite a las empresas crear para la Web una gran variedad de aplicaciones con valor añadido.

En el ejemplo anterior, una clínica que intenta ofrecer un mejor servicio convierte su calendario de citas en un servicio Web XML y lo ofrece en la Web. Otras empresas, como organizaciones de gestión sanitaria, directorios de empresas y clínicas, podrían crear sistemas que utilizaran o “consumieran” el servicio, lo que permitiría a los posibles clientes reservar una cita más eficazmente.

Por otra parte, esa misma clínica puede usar servicios Web XML creados por empresas de seguros y organizaciones de gestión sanitaria para enviar facturas directamente. También podría contratar un sistema de contabilidad con una tercera empresa para integrarlo en sus propios sistemas, ahorrando tiempo y recursos.

#### Mayor productividad de los desarrolladores

Desde el punto de vista técnico, Microsoft .NET cambiará la forma en que los desarrolladores crean aplicaciones.

- Visual Studio .NET y .NET Framework permite a los desarrolladores aprovechar su experiencia y conocimientos para crear de forma rápida y sencilla avanzados servicios Web XML y aplicaciones. Estas herramientas ayudarán a los desarrolladores a transformar la Web de un sistema de presentación estática de información en un mundo rico en servicios Web XML interactivos.
- La aplicación de técnicas de desarrollo rápido de aplicaciones (Rapid Application Development, RAD) a servicios y aplicaciones Web aumentan la productividad de los desarrolladores y ahorra tiempo y dinero. Al ser compatibles con cualquier lenguaje de programación, estas herramientas aprovechan los conocimientos de los desarrolladores y les permiten utilizar la herramienta más adecuada para cada tarea.
- A los desarrolladores independientes de software, .NET les ofrece la oportunidad de crear nuevos servicios avanzados para la era de la Internet. Los desarrolladores pueden crear servicios que accedan automáticamente a información y hagan uso de ella de forma local o remota, trabajando con cualquier dispositivo o lenguaje sin tener que reescribir el código para cada entorno.

## ◆ Lección: Dentro de .NET

- Runtime de lenguaje común (Common Language Runtime)
- La biblioteca de clases de .NET Framework
- Espacios de nombres de ADO.NET
- Espacios de nombres de ASP.NET

---

### Introducción

.NET Framework consta de los siguientes componentes:

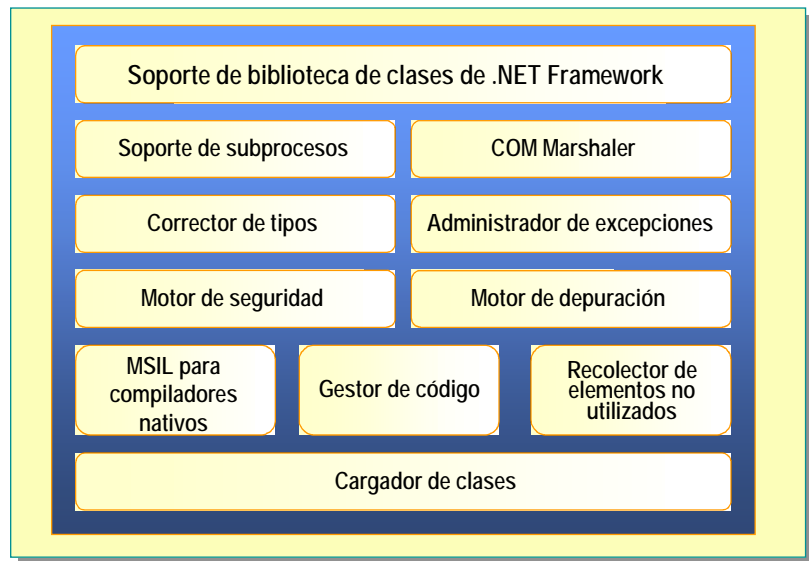
- Runtime de lenguaje común (Common Language Runtime, CLR)
- Biblioteca de clases de .NET Framework

### Objetivos de la lección

Al final de esta lección, usted será capaz de:

- Comprender la función del CLR en el desarrollo y la simplificación de aplicaciones Web mediante .NET.
- Identificar los espacios de nombres en la biblioteca de clases de .NET Framework..

## Runtime de lenguaje común (Common Language Runtime)



### Introducción

Los runtimes no son nada nuevo en programación. Muchos otros lenguajes de programación han empleado runtimes, incluyendo Microsoft Visual Basic (VBRUN hasta la versión 4.0 y MSVBVM hasta la versión 6.0), Visual C++ (MSVCRT), Visual FoxPro® y JScript, además de lenguajes de otras empresas como SmallTalk, Perl y Java.

La función más importante de .NET Framework, y lo que realmente lo diferencia de otros runtimes, es que proporciona un entorno unificado para todos los lenguajes de programación. Este entorno se conoce también como entorno gestionado.

### CLR simplifica el proceso de desarrollo

A pesar de su nombre, el CLR no sólo actúa en tiempo de ejecución, sino también durante el desarrollo de un componente. Mientras se ejecuta el componente, el CLR es responsable de administrar la asignación de memoria, iniciar y terminar subprocessos y procesos, aplicar las directivas de seguridad y satisfacer las dependencias del componente respecto a otros componentes.

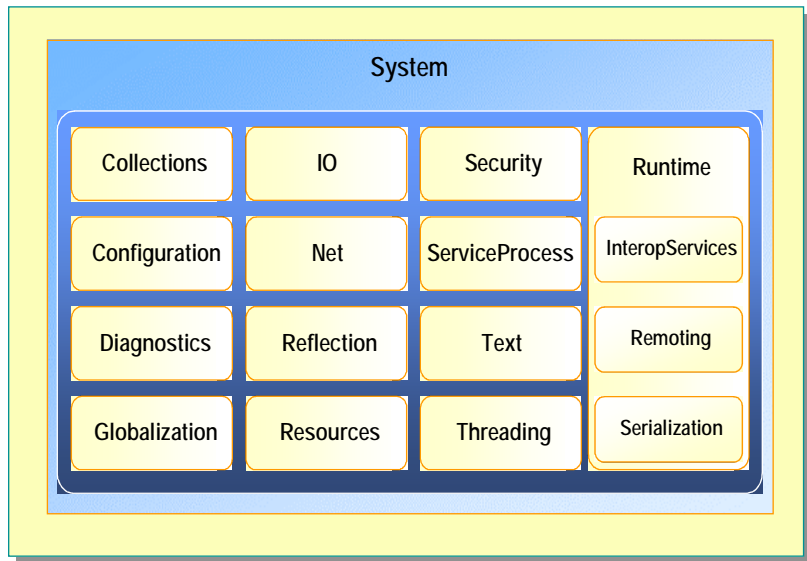
La función del runtime cambia ligeramente durante el desarrollo. Debido a que automatiza muchas funciones (como la administración de memoria), el CLR facilita el trabajo de los desarrolladores. En particular, el CLR garantiza la corrección del código y la seguridad de tipos. El CLR también reduce drásticamente la cantidad de código que tiene que escribir un desarrollador para transformar lógica de empresa en un componente reutilizable.

A continuación se indican algunas de las funciones de programación que hasta ahora necesitaban los sistemas operativos pero que el CLR hace innecesarias:

- No se necesita registrar el sistema, ya que las aplicaciones se describen por sí mismas.
- No se necesitan identificadores únicos globales (GUID), ya que las clases se organizan utilizando un sistema jerárquico con nombres que pueden leer las personas.
- No se necesitan archivos de lenguaje de definición de interfaz (IDL), ya que el sistema y los lenguajes utilizan las mismas representaciones.
- No se utilizan HRESULTs, ya que el CLR utiliza control estructurado de excepciones.
- No se necesita IUnknown, ya que sus funciones las realiza el System.Object raíz.
- No se necesita añadir y liberar referencias a objetos, ya que el CLR sabe qué objetos se están usando y si es necesario pueden limpiar los objetos no utilizados.
- El método COM **CoCreateInstance** se convierte en un operador más.



## La biblioteca de clases de .NET Framework



### Introducción

La biblioteca de clases de .NET Framework es una colección de clases reutilizables, o tipos, con un alto grado de integración en el CLR. La biblioteca contiene varios espacios de nombres. A modo de ejemplo examinaremos el espacio de nombres System.

### Espacio de nombres System

El espacio de nombres System.IO se encuentra bajo el espacio de nombres System y contiene servicios de entrada/salida (I/O). El espacio de nombres System.Collections proporciona listas ordenadas, tablas hash y otros grupos de datos. El espacio de nombres System.Net da soporte para TCP/IP y sockets.

La biblioteca de clases de .NET Framework se diseñó para:

- Habilitar la factorización y extensibilidad.
- Activar como base prácticas y estándares Web.
- Unificar modelos de aplicaciones.
- Aumentar la productividad de los desarrolladores ofreciendo un solo modelo de programación jerárquico e intuitivo.
- Permitir la herencia y depuración entre lenguajes.
- Facilitar la adición y modificación de funciones de .NET Framework.
- Permitir la creación de aplicaciones seguras.

### Funcionalidad

La biblioteca de clases de .NET Framework proporciona una gran cantidad de funciones, incluyendo las siguientes:

- Gestión de colecciones de objetos
- Acceso a bases de datos
- Salida a pantalla

- Seguridad y cifrado

## ◆ Lección: Compilación y ejecución en .NET

- El modelo de ejecución CLR
- Demostración: Uso de MSIL Disassembler
- Examen de código: Programación en .NET

---

### Introducción

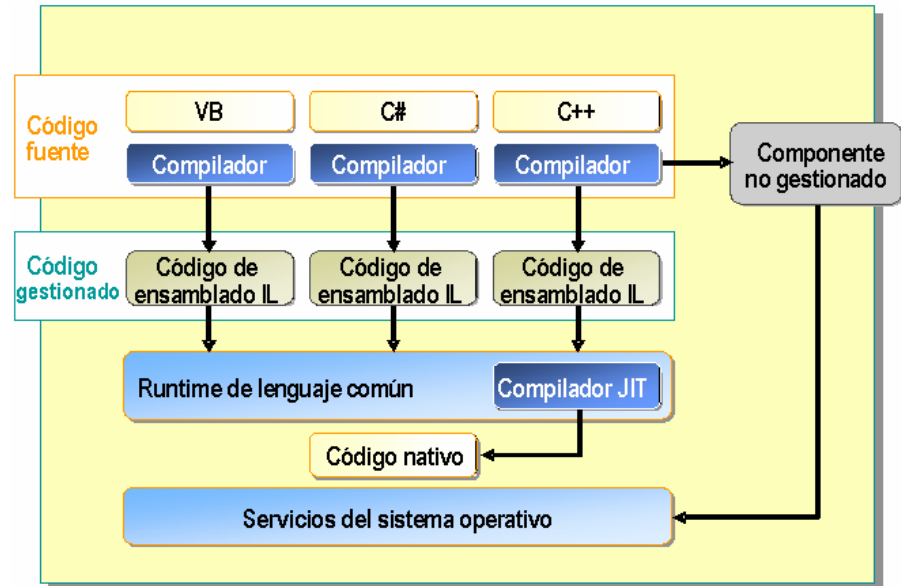
Esta lección describe los procesos de ejecución y compilación empleados en .NET.

### Objetivos de la lección

Al final de esta lección, usted será capaz de:

- Comprender el modelo de ejecución CLR.
- Definir un ensamblado y cómo se utiliza en .NET.
- Comprender la compilación y ejecución de aplicaciones .NET.
- Usar el MSIL Disassembler para ver un archivo portátil ejecutable.

## El modelo de ejecución CLR



### Introducción

Esta lección los procesos de ejecución y compilación empleados en .NET.

### Ensamblados y metadatos

La unidad básica de una aplicación basada en .NET recibe el nombre de *ensamblado*. Un ensamblado es la unidad de versión e instalación. En general, el código fuente se puede compilar en un archivo .exe o un DLL (también es posible crear módulos para vincularlos posteriormente a un archivo .exe o un DLL), que son las formas más sencillas de un ensamblado.

Los compiladores generan ensamblados que contienen lenguaje intermedio de Microsoft (Microsoft Intermediate Language, MSIL). Pero además del código MSIL, el compilador también inserta metadatos en el ensamblado. Un *metadato* es una colección de información que describe todos los tipos, clases, métodos, campos y eventos contenidos en el ensamblado, de forma similar a una biblioteca de tipos. Pero al contrario de un servidor COM, que puede o no incluir una biblioteca de tipos, un ensamblado y sus metadatos son inseparables. Esto significa que el ensamblado se describe por sí mismo.

### Ensamblados de un archivo y multiarchivo

En muchos casos es posible considerar un ensamblado como un solo archivo .exe o DLL. En algunas situaciones, un solo ensamblado puede hacer que la instalación sea mucho más simple, ya que todos los componentes necesarios están juntos. Sin embargo, algunos ensamblados (por ejemplo, DLLs) se pueden vincular en un ensamblado que recibe el nombre de ensamblado multiarchivo.

Para ello se utiliza el vinculador de ensamblados, AL.exe. En algunos casos, como en aplicaciones basadas en Web, el hecho de que los ensamblados estén contenidos en archivos separados puede resultar muy ventajoso, porque es posible descargar únicamente los módulos que se necesitan.

**El modelo de ejecución CLR**

Antes de ejecutar el código MSIL, es preciso convertirlo en instrucciones binarias nativas. La compilación se realiza normalmente con un compilador Just-in-Time (JIT).

El código fuente escrito en C#, Visual Basic .NET u otro lenguaje apropiado para el CLR se transforma primero en MSIL usando el compilador del lenguaje correspondiente. Antes de la ejecución, este código MSIL se compila en código nativo con un compilador JIT para el procesador en que se vaya a ejecutar el código. El compilador JIT no compila todo el código a la vez. La opción por defecto es hacer una compilación JIT de cada método la primera vez que se invoca, pero también es posible hacer una “compilación JIT previa” del código IL empleando Native Image Generator (NGEN.exe). Con esta opción todos los métodos se compilan antes de que se cargue la aplicación, lo que evita la compilación JIT cada vez que se invoca un método por primera vez.

Hay que tener en cuenta que todos los lenguajes para el CLR deben tener unas prestaciones similares. Aunque algunos compiladores pueden crear mejor código MSIL que otros, es poco probable que haya grandes diferencias en la velocidad de ejecución.