

Introducción a la plataforma Microsoft .NET

Índice

Descripción	1
¿Qué es la plataforma Microsoft .NET?	2
¿Qué es el .NET Framework?	5
¿Cuáles son los componentes del .NET Framework?	12
¿Cuáles son las mejoras de Visual Basic .NET?	21
Descripción de los espacios de nombres	23

Descripción

Objetivo

Describir los temas y objetivos del módulo.

Presentación

En este módulo, presentamos la plataforma .NET. Hablaremos sobre el .NET Framework y sobre el soporte de lenguajes que proporciona el .NET Framework a la plataforma.

- ¿Qué es la plataforma Microsoft .NET?
- ¿Qué es el .NET Framework?
- ¿Cuáles son los componentes del .NET Framework?
- ¿Cuáles son las mejoras de Visual Basic .NET?

Sugerencia

Este módulo proporciona una descripción de los sistemas .NET para los desarrolladores de Visual Basic.

Asegúrese de destacar las ventajas de .NET respecto de versiones anteriores de las arquitecturas de desarrollo Windows y Visual Basic siempre que sea posible.

Microsoft® Visual Basic® .NET constituye una importante mejora a la línea de productos de Visual Basic. Como desarrollador de Visual Basic, le será útil entender la razón lógica y las características que forman la base de la Plataforma Microsoft .NET antes de estudiar en profundidad Visual Basic .NET.

En este módulo, aprenderá a:

- Enumerar los principales elementos de la plataforma .NET.
- Describir el .NET Framework y sus componentes.
- Enumerar las principales mejoras de Visual Basic .NET.

¿Qué es la plataforma Microsoft .NET?

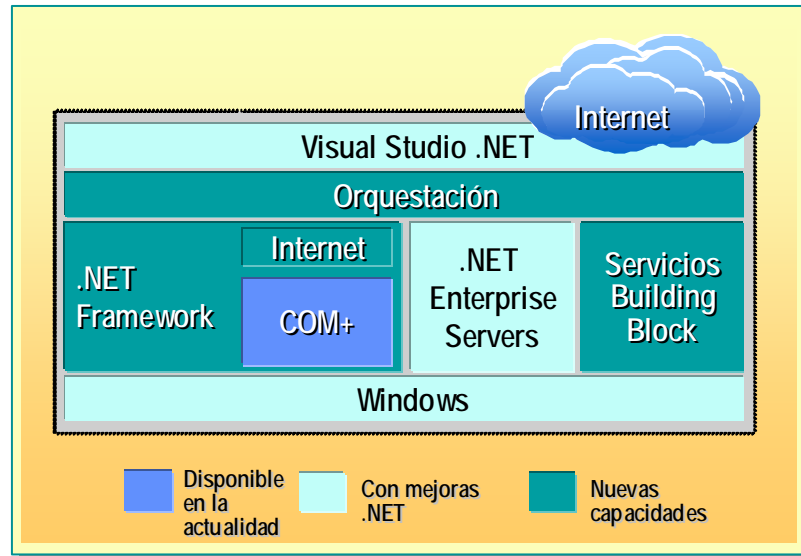
Objetivo

Presentar los principales servicios y características que proporciona la plataforma .NET.

Presentación

La plataforma .NET está formada por varios servicios y características básicos, como se muestra en esta diapositiva. Uno de los objetivos de esta nueva plataforma es simplificar el desarrollo Web. Esta sección describe la arquitectura de la plataforma .NET.

¿Qué es la plataforma Microsoft .NET?



Esta sección describe los componentes de la plataforma .NET, incluyendo el .NET Framework, los servicios .NET Building Blocks, los servidores .NET Enterprise Servers y Microsoft Visual Studio® .NET. El objetivo de la plataforma .NET es simplificar el desarrollo Web proporcionando todas las herramientas y tecnologías necesarias para desarrollar aplicaciones Web distribuidas.

La plataforma .NET proporciona:

- Un modelo de programación coherente e independiente del lenguaje para todas las capas o niveles de una aplicación.
- Una interoperabilidad transparente entre tecnologías.
- Una fácil migración desde tecnologías existentes.
- Un completo soporte de tecnologías de Internet independientes de la plataforma y basadas en estándares, incluyendo Hypertext Transfer Protocol (HTTP), Extensible Markup Language (XML) y Simple Object Access Protocol (SOAP).

La plataforma .NET

La plataforma .NET es un conjunto de tecnologías diseñadas para transformar Internet en una plataforma informática distribuida a escala completa. Proporciona nuevas formas de desarrollar aplicaciones a partir de colecciones de Servicios Web. La plataforma .NET soporta totalmente la infraestructura existente de Internet, incluyendo HTTP, XML y SOAP.

Tecnologías intrínsecas de la Plataforma .NET

Las tecnologías intrínsecas que integran la plataforma .NET incluyen:

- .NET Framework

Sugerencia

Debido al CLR, todos los lenguajes compatibles con .NET utilizarán los mismos archivos del entorno de ejecución. Esto constituye una gran ayuda para los desarrolladores de Visual Basic, puesto que ya no necesitarán librerías de entorno de ejecución específicas para Visual Basic. Estos archivos se instalarán automáticamente en futuras versiones de Windows.

El .NET Framework se basa en un nuevo entorno de ejecución de lenguaje común (*CLR - Common Language Runtime*). Este motor de ejecución proporciona un conjunto de servicios comunes para los proyectos generados en Visual Studio .NET, con independencia del lenguaje. Estos servicios proporcionan bloques de desarrollo fundamentales para aplicaciones de cualquier tipo, para todas las capas y niveles de la arquitectura de aplicaciones.

Microsoft Visual Basic, Microsoft Visual C++®, y otros lenguajes de programación Microsoft se han mejorado para beneficiarse de estos servicios. Otros lenguajes de terceros escritos para la plataforma .NET también tienen acceso a los mismos servicios.

Nota Gracias al CLR, todos los lenguajes .NET utilizarán los mismos archivos del entorno de ejecución. Esto significa que no es necesario distribuir librerías de entorno de ejecución específicas para Visual Basic ya que los archivos del entorno de ejecución de .NET se instalarán automáticamente en las futuras versiones de Microsoft Windows®.

- Servicios .NET Building Blocks

Los servicios .NET Building Block son servicios programables distribuidos que están disponibles tanto en modo conectado como desconectado. Un servicio puede invocarse desde un ordenador aislado no conectado a Internet, ofrecido por un servidor local dentro de una empresa, o puede ser accedido a través de Internet. Los servicios .NET Building Blocks pueden utilizarse desde cualquier plataforma que soporte el protocolo SOAP. Los servicios pueden incluir identidad, notificación y mensajería, personalización, almacenamiento esquematizado, calendario, directorio, búsqueda y distribución de software.

- Visual Studio .NET

Visual Studio .NET ofrece un entorno de desarrollo de alto nivel para desarrollar aplicaciones que se ejecutan sobre el .NET Framework. Proporciona las tecnologías fundamentales para simplificar la creación, implantación y evolución constante de aplicaciones y Servicios Web seguros, escalables y de alta disponibilidad. También permite crear una nueva generación de aplicaciones basadas en Windows con numerosas y nuevas características disponibles gracias al .NET Framework.

■ Servidores .NET Enterprise Servers

Los servidores .NET Enterprise Servers proporcionan escalabilidad, fiabilidad, gestión e integración dentro de la empresa y entre organizaciones, y muchas otras características descritas en la siguiente tabla.

Servidor	Descripción
Microsoft SQL Server™ 2000	Incorpora funcionalidad basada en XML, soporte de estándares Worldwide Web Consortium (W3C), la capacidad de manipular datos XML utilizando Transact SQL (T-SQL), un análisis flexible y potente basado en Web y un acceso seguro a los datos a través de la Web utilizando HTTP.
Microsoft BizTalk™ Server 2000	Proporciona servicios de integración de aplicaciones empresariales (<i>EAI - enterprise aplicación integration</i>), integración negocio a negocio (<i>business-to-business</i>) y tecnología avanzada de orquestación <i>BizTalk Orchestration</i> para crear procesos de negocio dinámicos que abarcan aplicaciones, plataformas y organizaciones a través de Internet.
Microsoft Host Integration Server 2000	Proporciona la mejor forma de integrar Internet, intranet y las tecnologías cliente/servidor, manteniendo al mismo tiempo las inversiones realizadas en los sistemas existentes. Microsoft Host Integration Server 2000 sustituye a Microsoft SNA Server.
Microsoft Exchange 2000 Enterprise Server	Está basado en la potente tecnología de mensajería y colaboración de Exchange e incorpora nuevas funcionalidades importantes, además de mejorar la fiabilidad, la escalabilidad y el rendimiento de su arquitectura básica. Otras características mejoran la integración de Exchange 2000 con Microsoft Windows 2000, Microsoft Office 2000 e Internet.
Microsoft Application Center 2000	Ofrece una herramienta para implantar y administrar aplicaciones Web de alta disponibilidad.
Microsoft Internet Security and Acceleration Server 2000	Proporciona conectividad a Internet segura, rápida y manejable. Internet Security and Acceleration Server integra un cortafuegos empresarial extensible y multicapa, y un caché Web escalable de gran rendimiento. Se apoya en la seguridad y en el directorio de Windows 2000 para obtener seguridad basada en políticas, aceleración y gestión en redes distribuidas.
Microsoft Commerce Server 2000	Proporciona un marco de trabajo de aplicación, sofisticados mecanismos de realimentación y capacidades de análisis.

¿Qué es el .NET Framework?

Objetivo

Describir los temas que se tratan en esta sección.

Presentación

El .NET Framework proporciona todos los servicios comunes necesarios para la ejecución de aplicaciones.

- Descripción del .NET Framework
- Beneficios del .NET Framework
- Lenguajes en el .NET Framework

En esta sección, veremos cómo el .NET Framework proporciona todos los servicios comunes que sus aplicaciones necesitan para ejecutarse. Estudiaremos la disponibilidad de los servicios para cualquier lenguaje compatible con .NET a través de la especificación *Common Language Specification (CLS)*. También estudiaremos qué lenguajes están soportados en el .NET Framework.

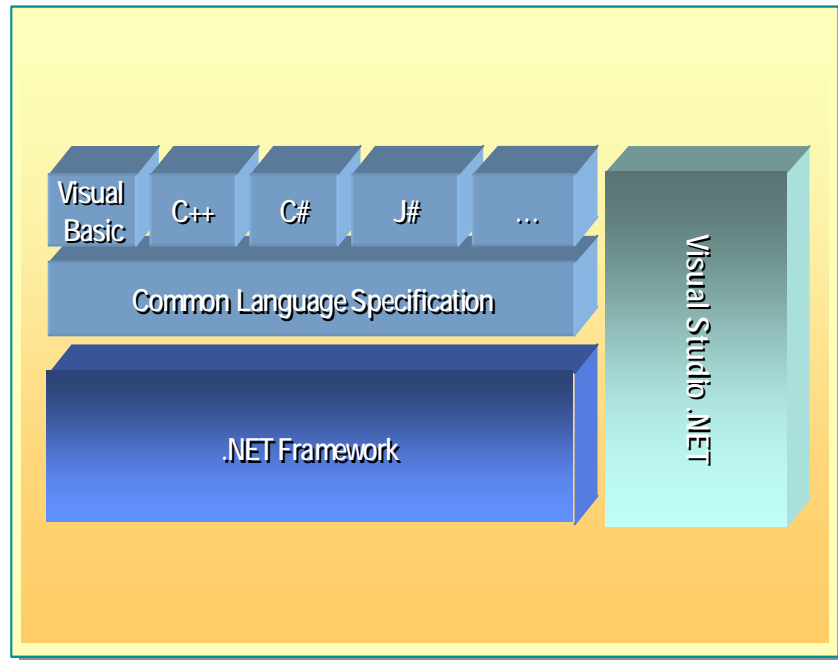
Descripción del .NET Framework

Objetivo

Explicar cómo encajan los diversos componentes del .NET Framework.

Presentación

El .NET Framework proporciona todos los servicios comunes necesarios para ejecutar las aplicaciones Visual Basic .NET. Estos servicios están disponibles en cualquier lenguaje compatible con .NET gracias a la especificación *Common Language Specification*.



El .NET Framework proporciona todos los servicios comunes necesarios para ejecutar las aplicaciones Visual Basic .NET.

Crear componentes en el .NET Framework

Antes del modelo COM, las aplicaciones estaban formadas por entidades totalmente separadas con poca o ninguna integración. Al utilizar COM, es posible integrar componentes dentro de una aplicación y entre aplicaciones exponiendo una serie de interfaces comunes. Sin embargo, todavía es necesario que los desarrolladores escriban el código para envolver, gestionar e integrar componentes y objetos.

En el .NET Framework, los componentes se construyen sobre una base común. Ya no es necesario escribir código para permitir que los objetos interactúen directamente entre sí. En el entorno .NET, ya no es necesario escribir envoltorios de componentes, puesto que los componentes no los utilizan. El .NET Framework puede interpretar las construcciones a los que los desarrolladores están acostumbrados utilizando lenguajes orientados a objetos. El .NET Framework soporta totalmente clases, herencia, métodos, propiedades, eventos, polimorfismo, constructores y otras construcciones orientadas a objetos.

Common Language Specification

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. La especificación CLS permite a los desarrolladores de Visual Basic .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 Visual Basic .NET heredar de clases desarrolladas en lenguajes diferentes.

Visual Studio .NET

En el .NET Framework, Visual Studio .NET ofrece las herramientas que permiten el desarrollo rápido de aplicaciones.

Beneficios del .NET Framework

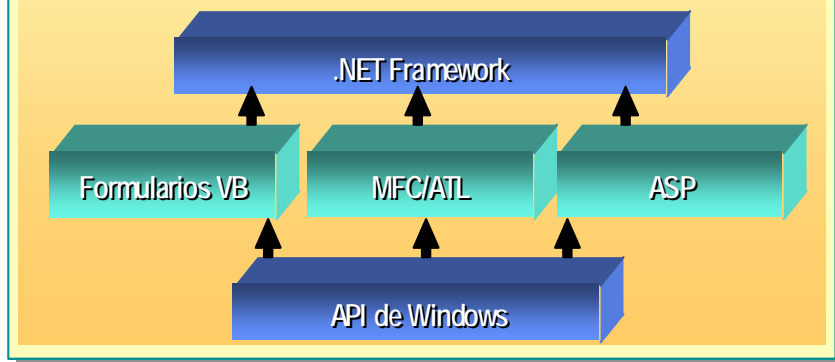
Objetivo

Explicar cómo los objetivos de diseño del marco de trabajo facilitan el desarrollo de aplicaciones.

Presentación

Antes de estudiar los diversos componentes del .NET Framework, resulta útil entender los objetivos de diseño del .NET Framework. Ello nos ayudará a ver cómo el .NET Framework simplifica el desarrollo de software.

- Basado en estándares y prácticas Web
- Extensible
- Fácil de utilizar por los desarrolladores
- Diseñado utilizando modelos de aplicaciones unificados



En esta sección, veremos algunos de los beneficios del .NET Framework.

Objetivos del .NET Framework

El .NET Framework fue diseñado para satisfacer los siguientes objetivos:

- Basado en estándares y prácticas Web

El .NET Framework soporta totalmente las tecnologías existentes de Internet, incluyendo Hypertext Markup Language (HTML), XML, SOAP, Extensible Stylesheet Language for Transformations (XSLT), Xpath y otros estándares Web. El .NET Framework promueve la creación de servicios Web débilmente conectados y sin estado.

- 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 multi-lenguaje.

- 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. No existen tipos *variant*, sólo hay un tipo de cadena (*string*), y todos los datos de la cadena son *Unicode*.

- Diseñado utilizando modelos de aplicaciones unificados

Sugerencia

Comente que las clases de .NET Framework que estén selladas no pueden ser extendidas.

La funcionalidad de una clase .NET está disponible desde cualquier modelo de programación o lenguaje compatible con .NET.

Lenguajes del .NET Framework

Objetivo

Presentar los lenguajes soportados en el .NET Framework.

Presentación

En esta sección, estudiaremos los lenguajes que Microsoft proporciona para el .NET Framework con Visual Studio .NET. También estudiaremos algunos de los lenguajes para .NET creados por terceros desarrolladores.

■ Visual Basic .NET

Nueva versión de Visual Basic, con importantes innovaciones del lenguaje

■ C# – diseñado para la plataforma .NET

Nuevo lenguaje orientado a componentes

■ Extensiones gestionadas de Visual C++

Proporciona una mayor potencia y control

■ J# .NET

Aprovecha totalmente el .NET Framework

■ Lenguajes de terceros

La plataforma .NET es independiente del lenguaje (el .NET Framework proporciona todos los servicios comunes necesarios para permitir la ejecución de los programas). El .NET Framework ofrece soporte para numerosos lenguajes de programación, incluyendo lenguajes de Microsoft y de terceros.

Lenguaje	Descripción
Microsoft Visual Basic .NET	Visual Basic .NET ofrece importantes innovaciones del lenguaje respecto a versiones anteriores de Visual Basic. Visual Basic .NET soporta herencia, constructores, polimorfismo, sobrecarga, excepciones estructuradas, estricta verificación de tipos, <i>free threading</i> , y muchas otras características. En esta versión, Visual Basic Scripting Edition proporciona funcionalidad completa Visual Basic.
Microsoft Visual 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 incluyen clases, interfaces, delegados, <i>boxing</i> y <i>unboxing</i> , espacios de nombres, propiedades, indexadores, eventos, sobrecarga de operadores, versionado, atributos, código inseguro, y la creación de documentación en formato XML. No son necesarios archivos de cabecera ni archivos IDL (<i>Interface Definition Language</i>).
Extensiones gestionadas C++	La actualización de Visual C++ es una mínima extensión gestionada del lenguaje C++. Esta extensión proporciona acceso al .NET Framework incluyendo el recolector de basura, herencia de implementación simple y herencia de interfaz múltiple. Esta actualización también elimina la necesidad de escribir código de fontanería para los componentes. Ofrece acceso a bajo nivel donde sea necesario.
Microsoft Visual J#™ .NET	J# .NET es un lenguaje para los desarrolladores que utilicen el lenguaje Java que quieran generar las aplicaciones y servicios para el .NET Framework. J# .NET está totalmente integrado con el IDE de Visual Studio .NET y está diseñado para aprovechar

al máximo el .NET Framework.

Lenguajes de terceros

Varios lenguajes de terceros soportan la plataforma .NET. Estos lenguajes incluyen APL, COBOL, Pascal, Eiffel, Haskell, ML, Oberon, Perl, Python, Scheme y SmallTalk.

¿Cuáles son los componentes del .NET Framework?

Objetivo

Describir los temas que se tratan en esta sección.

Presentación

En esta sección, estudiaremos los diferentes componentes que integran el .NET Framework.

- Los componentes del .NET Framework
- Common Language Runtime - CLR
- Biblioteca de clases del .NET Framework
- ADO.NET: Datos y XML
- ASP.NET: Formularios Web y Servicios Web
- Interfaz de usuario

En esta sección, estudiaremos los componentes que integran el Microsoft .NET Framework. El .NET Framework está formado por un conjunto de tecnologías que constituyen una parte fundamental de la plataforma .NET. El .NET Framework proporciona los bloques de construcción básicos para desarrollar aplicaciones y Servicios Web utilizando ASP.NET.

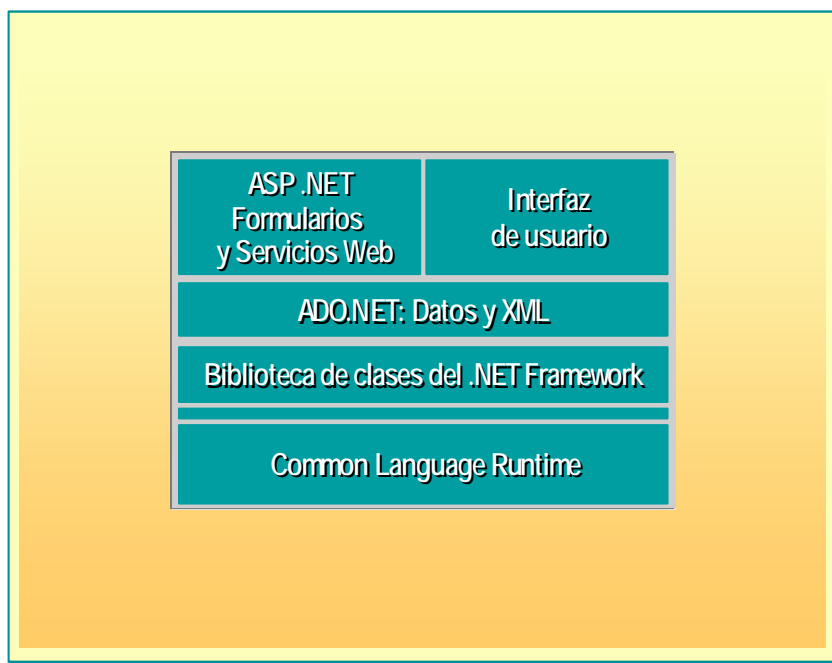
Componentes del .NET Framework

Objetivo

Ofrecer una descripción de los componentes del .NET Framework.

Presentación

Visual Basic .NET utiliza los componentes que proporciona el .NET Framework.



Los componentes principales del .NET Framework son los siguientes:

- Common Language Runtime - CLR
- Biblioteca de clases del .NET Framework
- ADO.NET: datos y XML
- ASP.NET: Formularios Web y Servicios Web
- Interfaz de usuario

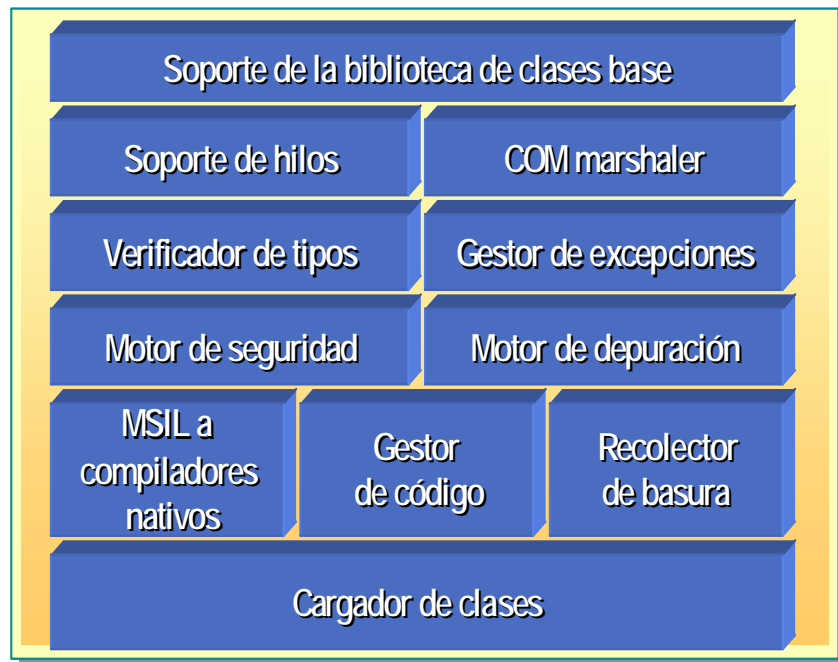
Common Language Runtime

Objetivo

Destacar algunos de los principales componentes del *Common Language Runtime*.

Presentación

Esta sección ofrece una visión general de los diversos componentes del *Common Language Runtime* y proporciona una breve descripción de cada uno. Como desarrollador de Visual Basic, nunca verá estas piezas físicamente, pero hablar sobre ellas le ayudará a comprender mejor la riqueza del entorno de ejecución.

**Sugerencia**

Muchos de estos temas son interesantes para los desarrolladores de Visual Basic .NET y se tratan en diversas partes del curso.

El CLR (*Common Language Runtime*) simplifica el desarrollo de aplicaciones, proporciona un entorno de ejecución robusto y seguro, soporta múltiples lenguajes, simplifica la implantación y administración de aplicaciones y proporciona un entorno gestionado.

¿Qué es un entorno gestionado?

Un entorno gestionado es aquel que proporciona servicios comunes de forma automática. Algunos ejemplos de los tipos de servicios que proporciona un entorno gestionado son el recolector de basura y la seguridad.

Componentes del Common Language Runtime

La siguiente tabla describe las características del *Common Language Runtime*.

Componente	Descripción
Cargador de clases	Carga la implementación de un tipo en memoria y lo prepara para su ejecución.
<i>Microsoft Intermediate Language (MSIL)</i> a compilador nativo	Convierte lenguaje intermedio MSIL a código nativo (<i>just-in-time</i>).
Gestor de código	Gestiona la ejecución de código.
Recolector de basura	Gestiona automáticamente la vida de todos sus objetos. Es un recolector de basura multiprocesador y escalable.
Motor de seguridad	Proporciona seguridad basada en evidencias en función del origen del código y del usuario.
Motor de depuración	Le permite depurar su aplicación y trazar la ejecución de código.
Verificador de tipos	No permitirá <i>casts</i> inseguros ni variables no inicializadas. Es posible verificar IL para garantizar la seguridad de tipos.
Gestor de excepciones	Proporciona una gestión de excepciones estructurada e integrada con Windows Structured Exception Handling (SEH). Se han mejorado los informes de errores.
Soporte de hilos (<i>threads</i>)	Proporciona clases e interfaces que permiten programación multi-hilo.
<i>COM marshaler</i>	Proporciona <i>marshaling</i> hacia y desde COM.
Soporte de la biblioteca de clases del .NET Framework	Integra el código con el entorno de ejecución que soporta la biblioteca de clases del .NET Framework. La siguiente sección trata sobre la biblioteca de clases del .NET Framework.

Biblioteca de clases del .NET Framework

Objetivo

Describir la biblioteca de clases del .NET Framework y el espacio de nombres principal: **System**.

Presentación

En esta sección, aprenderemos cómo la biblioteca de clases del .NET Framework expone características del entorno de ejecución y proporciona otros servicios de alto nivel.



Sugerencia

La biblioteca de clases del .NET Framework proporciona numerosas y potentes características nuevas para los desarrolladores de Visual Basic .NET.

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 a los desarrolladores de Visual Basic .NET para crear verdaderas aplicaciones multi-hilo.

En este punto, una descripción general de los espacios de nombres puede ser útil.

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.

Espacios 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. Otras clases proporcionan servicios como la conversión de tipos de datos o la manipulación de parámetros de métodos, operaciones matemáticas, invocación local y remota de programas, administración de los entornos de aplicaciones y la supervisión de aplicaciones gestionadas y no gestionadas.

El espacio de nombre **System.Collections** proporciona listas clasificadas, tablas *hash* y otras formas de agrupar datos. El espacio de nombres **System.IO** proporciona E/S de archivos, flujos, etc. El espacio de nombres **System.NET** proporciona soporte de Transmission Control Protocol/Internet Protocol (TCP/IP) y *sockets*.

Si desea más información sobre los espacios de nombres, realice una búsqueda de “espacio de nombres” en la documentación del .NET Framework SDK.

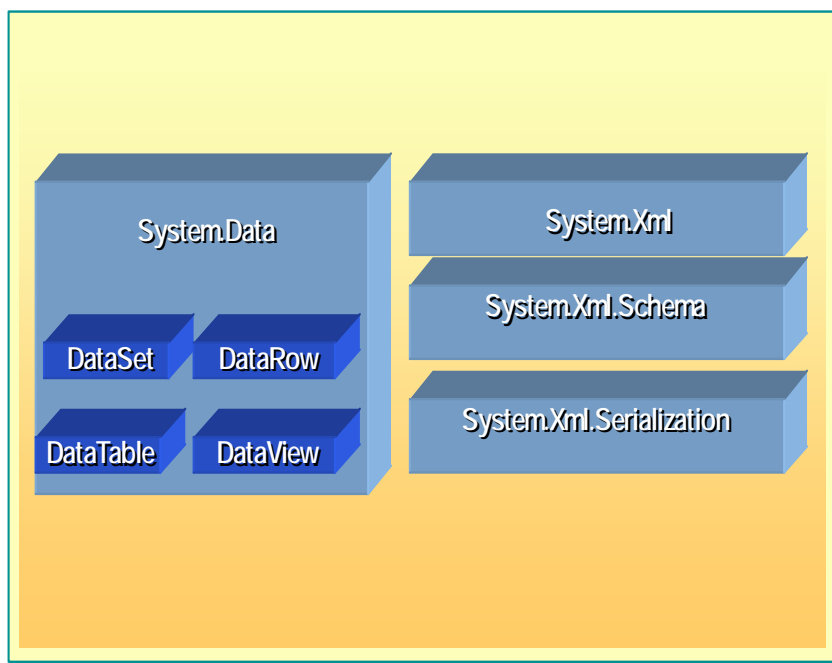
ADO.NET: Datos y XML

Objetivo

Explicar el soporte para la gestión de datos y XML en el entorno de ejecución.

Presentación

El .NET Framework proporciona un nuevo conjunto de clases ADO.NET para el manejo de datos.



Sugerencia

ADO.NET ha sido creado especialmente para ser utilizado en Internet, y utiliza XML como tecnología principal. Esto significa que los desarrolladores ya no necesitan pasar objetos **Recordset** a través del sistema, limitando potencialmente qué clientes pueden atender. También mejora el modelo de programación desconectado de versiones anteriores de ADO.

Comente que el conocimiento de ADO de los desarrolladores de Visual Basic les será de gran ayuda cuando cambien a ADO.NET.

Compruebe los conocimientos de los estudiantes sobre los conceptos básicos de XML.

ADO.NET es la siguiente generación de la tecnología ActiveX® Data Objects (ADO). ADO.NET proporciona un soporte mejorado para el modelo de programación desconectado. También ofrece un completo soporte de XML.

Espacio de nombres System.Data

El espacio de nombres **System.Data** está compuesto por clases que constituyen el modelo de objetos ADO.NET. A alto nivel, el modelo de objetos ADO.NET se divide en dos capas: la capa conectada y la capa desconectada.

El espacio de nombres **System.Data** incluye la clase **DataSet**, que representa múltiples tablas y sus relaciones. Estos conjuntos de datos son estructuras de datos completamente autocontenidas que pueden poblarse desde diversas fuentes de datos. Una posible fuente de datos podría ser XML, otra podría ser OLE DB, y una tercera fuente de datos podría ser el adaptador directo para SQL Server.

Espacio de nombres System.Xml

El espacio de nombres **System.Xml** proporciona soporte para XML. Incluye un parseador y un escritor XML, siendo compatibles con el W3C. Las transformaciones las proporciona el espacio de nombres **System.Xml.Xsl** y la implementación de XPath que permite navegación de grafos de datos en XML. El espacio de nombres **System.Xml.Serialization** proporciona toda la infraestructura básica para los Servicios Web, incluyendo características como el movimiento atrás y adelante entre objetos y una representación XML.

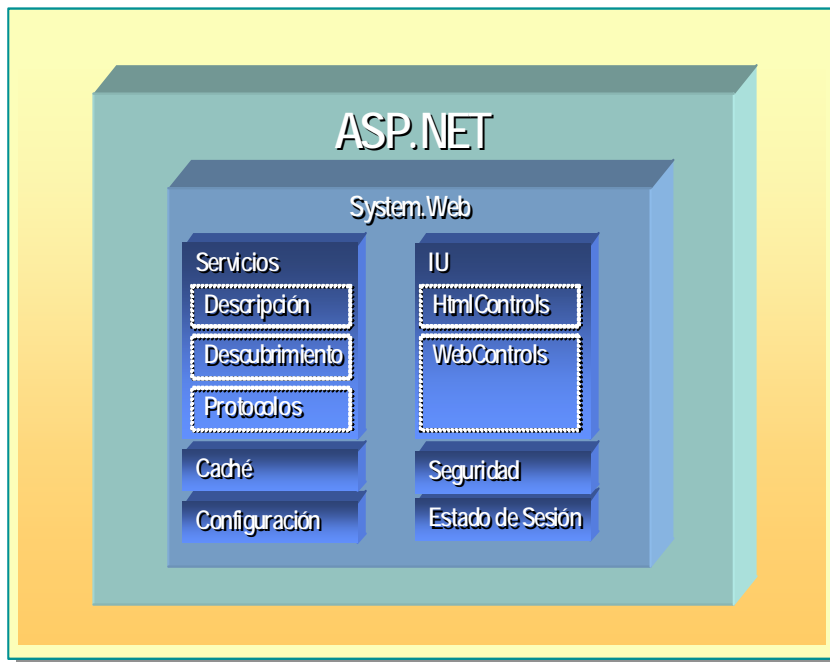
ASP.NET: Formularios Web y Servicios Web

Objetivo

Presentar los Formularios y Servicios Web.

Presentación

Internet evoluciona rápidamente desde los actuales sitios Web que únicamente sirven páginas de interfaz de usuario a navegadores, hasta una próxima generación de sitios Web programables que integran directamente organizaciones, aplicaciones, servicios y dispositivos. Estos sitios Web programables son mucho más que sitios a los que se accede de modo pasivo; son Servicios Web reutilizables e inteligentes.



ASP.NET es un marco de trabajo de programación basado en el Common Language Runtime y que puede ser utilizado en un servidor para crear potentes aplicaciones Web. Los Formularios Web de ASP.NET proporcionan una forma fácil y potente de generar interfaces de usuario (IUs) Web dinámicos. Los Servicios Web de ASP.NET proporcionan las piezas para construir aplicaciones distribuidas basadas en la Web. Los Servicios Web están basados en estándares abiertos de Internet, como HTTP y XML. Podemos pensar en un Servicio Web como un componente reutilizable accesible a través de Internet, en lugar de estar limitado a clientes Windows en una red de área local.

Sugerencia

Con Visual Basic .NET, el desarrollo Web es casi tan sencillo como el desarrollo de aplicaciones basadas en Windows.

La creación de aplicaciones Web puede llevarse a cabo en un mismo IDE, tanto si son aplicaciones basadas en páginas Web o en Servicios Web.

El Common Language Runtime proporciona un soporte nativo para crear y exponer Servicios Web utilizando una abstracción de programación coherente y familiar para desarrolladores tanto de Active Server Pages (ASP) como de Visual Basic. El modelo resultante es escalable y extensible. Este modelo se basa en estándares abiertos de Internet (HTTP, XML, SOAP y SDL) y, por tanto, puede ser accedido e interpretado desde cualquier cliente o dispositivo habilitado para Internet. Algunas de las clases ASP.NET más habituales se describen en los siguientes párrafos.

Espacio de nombres System.Web

En el espacio de nombres **System.Web**, existen servicios como la gestión de caché, seguridad, configuración y otros que son compartidos por los Servicios Web y por el interfaz de usuario (IU) Web.

Espacio de nombres System.Web.Services

El espacio de nombres **System.Web.Services** gestiona los requerimientos de los Servicios Web como los protocolos de transporte y el descubrimiento de servicios.

Espacio de nombres System.Web.UI

El espacio de nombres **System.Web.UI** proporciona dos clases de controles: los controles HTML y los controles Web. Los controles **HTMLControl** proporcionan un mapeado directo de las etiquetas HTML, como <INPUT>. Existen también otros controles **WebControl** más ricos que permiten estructurar controles mediante plantillas (por ejemplo, un control de rejilla).

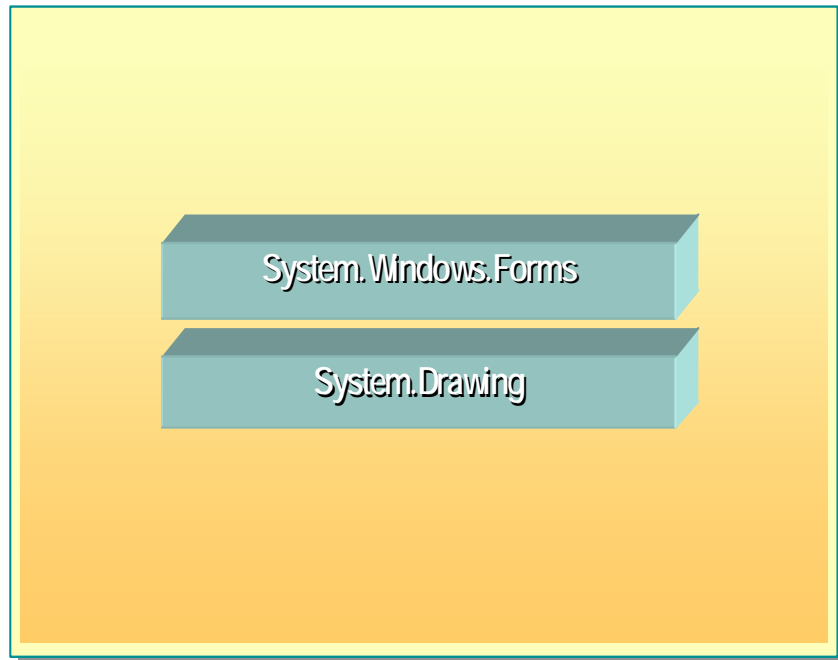
Interfaz de usuario

Objetivo

Explicar cómo el .NET Framework gestiona el interfaz de usuario para las aplicaciones Windows tradicionales.

Presentación

En esta sección, estudiaremos cómo .NET gestiona el interfaz de usuario. La clase **System.Windows.Forms** proporciona el IU de lado cliente y **System.Drawing** proporciona la siguiente generación de servicios GDI.



Las aplicaciones Windows pueden ahora proporcionar interfaces de usuario más potentes que nunca utilizando los espacios de nombres

System.Windows.Forms y **System.Drawing** del .NET Framework. El aspecto de los nuevos formularios Windows Forms de .NET resultará muy familiar para los desarrolladores de Visual Basic.

Espacio de nombres System.Windows.Forms

Podemos utilizar las clases del espacio de nombres **System.Windows.Forms** para crear el IU cliente. Esta clase nos permite implementar el IU de Windows estándar en nuestras aplicaciones .NET. Muchas funciones a las que anteriormente sólo podían accederse mediante llamadas a la API (*Application Programming Interface*) están ahora disponibles como parte de los propios formularios, lo cual hace el desarrollo mucho más fácil y potente.

Espacio de nombres System.Drawing

El espacio de nombres **System.Drawing** proporciona acceso a la funcionalidad de gráficos básica de GDI+. Los espacios de nombres **System.Drawing.Drawing2D**, **System.Drawing.Imaging** y **System.Drawing.Text** ofrecen funcionalidades más avanzadas.

¿Cuáles son las mejoras de Visual Basic .NET?

Objetivo

Presentar algunas de las principales mejoras de Visual Basic .NET.

Presentación

Visual Basic .NET proporciona importantes mejoras del lenguaje, un acceso total al .NET Framework, y un desarrollo Web mejorado.

¿Cuáles son las mejoras de Visual Basic .NET?

- **Principales mejoras del lenguaje**
 - Soporte mejorado orientado a objetos
 - Gestión de excepciones estructurada
- **Acceso total al .NET Framework**
 - Nuevas opciones de manejo de hilos
 - Recolector de basura
- **Desarrollo Web mejorado**
 - Creación de Formularios Web tan fácilmente como formularios Windows
 - Crear servicios Web rápidamente

Visual Basic .NET proporciona importantes mejoras del lenguaje, un acceso total al .NET Framework y un desarrollo Web mejorado.

Sugerencia

Explique cada uno de los puntos de la diapositiva y sus ventajas para los desarrolladores.

Importantes mejoras de lenguaje

En Visual Basic .NET, Visual Basic ha experimentado algunos cambios importantes para permitir a los desarrolladores de Visual Basic crear aplicaciones empresariales potentes, robustas y escalables.

- **Soporte orientado a objetos mejorado**

Permite a los desarrolladores de Visual Basic .NET utilizar herencia de clases, constructores, destructores, sobrecarga, interfaces y polimorfismo. Esto otorga a Visual Basic .NET tanto poder orientado a objetos como cualquier otro lenguaje .NET, como C# o Visual C++ con extensiones gestionadas.
- **Gestión de excepciones estructurada**

Simplifica el control de excepciones y permite utilizar potentes características como las excepciones anidadas. Esto mejora enormemente el anterior gestor de errores de Visual Basic.

Acceso total al .NET Framework

Los desarrolladores de Visual Basic .NET tienen acceso total al .NET Framework, incluyendo toda la biblioteca de clases de .NET Framework.

- **Nuevas opciones de manejo de hilos**
Permite crear aplicaciones que utilizan capacidades con múltiples hilos de ejecución. Por primera vez, los desarrolladores de Visual Basic podrán crear aplicaciones capaces de competir con aplicaciones basadas en Visual C++ en esta área.
- **Recolector de basura**
Garantiza que las aplicaciones creadas en Visual Basic .NET no acceden a posiciones de memoria sin referencia. Este proceso en segundo plano aborda aspectos relacionados con la gestión de memoria, como posiciones de memoria vacías y referencias circulares que existían en versiones anteriores de Visual Basic y en otros lenguajes.

Desarrollo Web mejorado

Visual Basic .NET permite a los desarrolladores crear rápidamente potentes aplicaciones Web.

- **Crear Formularios Web fácilmente**
Es tan fácil como crear formularios Windows utilizando la técnica de escribir código tras el evento de un control. Los desarrolladores ya no necesitan utilizar un entorno de desarrollo integrado (*IDE, Integrated Development Environment*) para el desarrollo de páginas Web (como Microsoft Visual InterDev®) y otro entorno para el desarrollo de componentes. Ahora, todos los aspectos de una aplicación Web se crean simplemente en un único entorno para los desarrolladores de Visual Basic y de un modo fácil de entender.
- **Crear Servicios Web rápidamente**
Es posible crear Servicios Web rápidamente de modo similar a cómo se creaban componentes en las versiones anteriores de Visual Basic. Los Servicios Web son una potente tecnología que le permite acceder a sus componentes (o a los componentes de otras personas) a través de Internet mediante HTTP. Cualquier desarrollador de Visual Basic que haya creado componentes podrá realizar fácilmente la transición a los Servicios Web.

Descripción de los espacios de nombres

Objetivo

Describir los espacios de nombres en el .NET Framework.

Presentación

En esta sección, estudiaremos los espacios de nombres del .NET Framework.

Descripción de los espacios de nombres

- Espacios de nombres
- Espacios de nombres utilizados en módulos obligatorios
- Espacios de nombres utilizados en módulos opcionales

En esta sección, estudiaremos los espacios de nombres del .NET Framework. También veremos qué espacios de nombres se enseñan en este curso.

Esta sección incluye los siguientes temas:

- Espacios de nombres
- Espacios de nombres utilizados en módulos obligatorios
- Espacios de nombres utilizados en módulos opcionales

Espacios de nombres

Objetivo

Entender el modo en que los espacios de nombres proporcionan una jerarquía de tipos y funcionalidades fáciles de utilizar.

Presentación

El .NET Framework incluye un gran conjunto de ensamblados de bibliotecas de clases, que contienen cientos de tipos. Estos ensamblados proporcionan acceso a funcionalidades del sistema en nuestro proceso de desarrollo.

Espacios de nombres



El .NET Framework incluye un gran conjunto de ensamblados que forman las bibliotecas de clases, que conjuntamente contienen cientos de tipos. Estos ensamblados proporcionan acceso a funcionalidades del sistema en nuestro proceso de desarrollo.

Objetivo de los espacios de nombres

Debido a que la biblioteca de clases del .NET Framework incluye definiciones para numerosos tipos, la biblioteca se organiza en un espacio de nombres de estructura jerárquica.

Los espacios de nombres utilizan un esquema nombres con una sintaxis basada en un punto para agrupar lógicamente clases relacionadas de modo que puedan buscarse y referenciarse fácilmente. Por ejemplo, el espacio de nombres **System.Data** contiene clases que constituyen la arquitectura de ADO.NET. El espacio de nombres **System.Xml** es el espacio de nombres global para todas las clases XML que proporcionan soporte basado en estándares para procesar XML.

El espacio de nombres System

El espacio de nombres **System** es el espacio de nombres raíz para los tipos en el .NET Framework. El espacio de nombres **System** contiene el tipo base **Object**, desde el que se derivan otros tipos.

El espacio de nombres **System** también contiene tipos para la gestión de excepciones, recolector de basura, E/S de consola, varios tipos de herramientas, tipos de datos de formato, generadores de números aleatorios y funciones matemáticas.

Espacios de nombres utilizados en módulos obligatorios

Objetivo

Explicar qué espacios de nombres se enseñan en este curso, y qué espacios de nombres no.

Presentación

Este curso cubre muchos de los espacios de nombres **System**. Dos espacios de nombres que no se cubren son los espacios de nombres **System.Data** y **System.Security**.

Espacios de nombres utilizados en módulos obligatorios

Módulo 2 <ul style="list-style-type: none">• System.Console	Módulo 10 <ul style="list-style-type: none">• System.IO
Módulo 3 <ul style="list-style-type: none">• System.Windows.Forms• System.Drawing	Módulo 11 <ul style="list-style-type: none">• System.Net• System.Net.Sockets
Módulo 4 <ul style="list-style-type: none">• System.Reflection	Módulo 12 <ul style="list-style-type: none">• System.Runtime.Serialization
Módulo 7 <ul style="list-style-type: none">• System.Text• System.Collections	Módulo 13 <ul style="list-style-type: none">• System.Runtime.Remoting.Channels• System.Web.Services

Este curso cubre muchos de los espacios de nombres del Microsoft .NET Framework.

El Módulo 2 explica el espacio de nombres **System.Console** para enviar salida de impresión a la consola. El Módulo 3 explica los espacios de nombres **System.Windows.Forms** y **System.Drawing** para generar un formulario que contenga botones para interactuar con el usuario.

El Módulo 4 explica el espacio de nombres **System.Reflection** para almacenar información relativa a versiones y archivos de claves en un ensamblado. El Módulo 7 explica el espacio de nombres **System.Text** para la gestión avanzada de cadenas y el **System.Collections** para mantener colecciones de datos.

El Módulo 10 explica el espacio de nombres **System.IO** para leer y escribir archivos. El Módulo 11 explica los espacios de nombres **System.Net** y **System.Net.Sockets** para la transmisión de datos a través de la red.

El Módulo 12 explica el espacio de nombres **System.Runtime.Serialization** para la persistencia de objetos en el almacenamiento. El Módulo 13 explica los espacios de nombres **System.Runtime.Remoting.Channels** y **System.Web.Services** para la invocación de objetos remotos, y para la creación de servicios Web XML.

Espacios de nombres utilizados en módulos opcionales

Objetivo

Completar la información sobre los espacios de nombres presentados en la sección anterior.

Presentación

Estos son algunos espacios de nombres que se cubren en módulos opcionales de este curso.

Espacios de nombres utilizados en módulos opcionales

Módulo 14

- System.Threading

Módulo 16

- System.Data

Módulo 17

- System.Reflection

Los Módulos 14 a 17 son opcionales.

El Módulo 14 explica el espacio de nombres **System.Threading** para permitir la programación multi-hilo. El Módulo 16 explica el espacio de nombres **System.Data**, que proporciona los objetos y tipos básicos para el modelo de programación ADO.NET. ADO.NET también proporciona soporte de XML en el espacio de nombres **System.Xml**. Finalmente, el Módulo 17 explica el espacio de nombres **System.Reflection**, que contiene clases que pueden utilizarse para examinar metadatos.

Espacios de nombres no cubiertos

Este curso no cubre el espacio de nombres **System.Security**.