

---

## Espacios de Nombres

---

La biblioteca de clases de .NET Framework proporciona los siguientes espacios de nombres.

### **Microsoft.CSharp**

Contiene clases para compilar y generar código mediante el lenguaje C#.

### **Microsoft.JScript**

Contiene clases que admiten la compilación y generación de código mediante el lenguaje JScript.

### **Microsoft.VisualBasic**

Contiene clases que admiten la compilación y generación de código mediante el lenguaje Basic .NET.

### **Microsoft.Vsa**

Contiene interfaces que permiten integrar secuencias de comandos para los motores de secuencias de comandos de .NET Framework en las aplicaciones, y compilar y ejecutar código en tiempo de ejecución.

### **Microsoft.Win32**

Proporciona dos tipos de clases: las que controlan los eventos desencadenados por el sistema operativo y las que manipulan el Registro del sistema.

## **1. System**

Contiene clases fundamentales y clases base que definen los valores y tipos de datos de referencia, eventos y controladores de eventos, interfaces, atributos y excepciones de procesamiento comúnmente utilizados.

Otras clases proporcionan servicios para convertir tipos de datos, manipular parámetros de métodos, realizar cálculos matemáticos, invocar programas remotos y locales, administrar el entorno de aplicaciones y supervisar aplicaciones administradas y no administradas.

### **1.1. System.CodeDom**

Contiene clases que se pueden utilizar para representar los elementos y la estructura de un documento de código fuente. Estos elementos pueden usarse para modelar la estructura de un documento de código fuente que se puede utilizar como código fuente en un lenguaje compatible mediante la funcionalidad que proporciona el espacio de nombres System.CodeDom.Compiler.

#### **a. System.CodeDom.Compiler**

Contiene tipos para administrar la generación y compilación de código fuente en los lenguajes de programación compatibles. Cada generador de código puede generar código fuente en un lenguaje de programación determinado basado en la estructura de los

modelos de código fuente CodeDOM (Code Document Object Model) que se componen de elementos proporcionados por el espacio de nombres System.CodeDom.

## **1.2 System.Collections**

Contiene interfaces y clases que definen diversas colecciones de objetos, tales como listas, colas, matrices de bits, tablas hash y diccionarios.

### **a. System.Collections.Specialized**

Contiene colecciones especializadas y con establecimiento inflexible de tipos; por ejemplo, un diccionario de listas vinculadas, un vector de bits o colecciones que sólo contienen cadenas.

## **1.3. System.ComponentModel**

Proporciona clases que se utilizan para implementar el comportamiento de los componentes y controles en tiempo de diseño y en tiempo de ejecución. Este espacio de nombres incluye las clases e interfaces necesarias para implementar atributos, convertidores de tipos, enlaces a orígenes de datos y componentes de licencia.

### **a. System.ComponentModel.Design**

Contiene clases que los programadores pueden utilizar para generar un comportamiento personalizado de los componentes en tiempo de diseño e interfaces de usuario para configurar los componentes en tiempo de diseño. El entorno de tiempo de diseño proporciona sistemas que permiten a los programadores organizar los componentes y configurar sus propiedades.

- **System.ComponentModel.Design.Serialization** Proporciona tipos que admiten la personalización y el control de la serialización en tiempo de diseño.

## **1.4. System.Configuration**

Proporciona clases e interfaces que permiten tener acceso mediante programación a la configuración de .NET Framework y controlar los errores de los archivos de configuración (archivos .config).

### **a. System.Configuration.Assemblies**

Contiene clases que se utilizan para configurar un ensamblado.

### **b. System.Configuration.Install**

Proporciona clases que permiten escribir instaladores personalizados para los componentes propios. La clase Installer es la clase base para todos los instaladores personalizados de .NET Framework.

## **1.5 System.Data**

Consta principalmente de las clases que constituyen la arquitectura ADO.NET. La arquitectura ADO.NET permite crear componentes que administran eficazmente los datos procedentes de múltiples orígenes. En un escenario desconectado (como Internet), ADO.NET proporciona las herramientas necesarias para solicitar, actualizar y reconciliar los datos de sistemas de varios niveles. La arquitectura ADO.NET también se implementa en aplicaciones cliente, como Windows Forms, o en las páginas HTML creadas por ASP.NET.

#### **a. System.Data.Common**

Contiene clases compartidas por los proveedores de datos de .NET Framework. Un proveedor de datos de .NET Framework describe una colección de clases que se utiliza para tener acceso a un origen de datos, como una base de datos, en el espacio administrado.

#### **b. System.Data.Odbc**

Encapsula el proveedor de datos de .NET Framework para ODBC. Un proveedor de datos de .NET Framework describe una colección de clases que se utiliza para tener acceso a un origen de datos, como una base de datos, en el espacio administrado. Mediante la clase `OdbcDataAdapter`, se puede llenar una clase `DataSet` residente en memoria, que se puede utilizar para realizar consultas y actualizar el origen de datos.

Para obtener más información sobre cómo utilizar este espacio de nombres, vea las clases `OdbcDataReader`, `OdbcCommand` y `OdbcConnection`.

Nota: Este espacio de nombres es compatible únicamente con la versión 1.1 de .NET Framework.

#### **c. System.Data.OleDb**

Encapsula el proveedor de datos de .NET Framework para OLE DB. El proveedor de datos de .NET Framework para OLE DB describe una colección de clases que se utiliza para obtener acceso a un origen de datos OLE DB en el espacio administrado.

#### **d. System.Data.OracleClient**

Encapsula el proveedor de datos de .NET Framework para Oracle. El proveedor de datos de .NET Framework para Oracle describe una colección de clases que se utiliza para obtener acceso a un origen de datos de Oracle en el espacio administrado.

Nota: Este espacio de nombres es compatible únicamente con la versión 1.1 de .NET Framework.

#### **e. System.Data.SqlClient**

Encapsula el proveedor de datos de .NET Framework para SQL Server. El proveedor de datos de .NET Framework para SQL Server describe una colección de clases que se utiliza para obtener acceso a una base de datos de SQL Server en el espacio administrado.

#### **f. System.Data.SqlServerCE**

Describe una colección de clases que se puede utilizar para obtener acceso a una base de datos de SQL Server CE de dispositivos basados en Windows CE en el entorno administrado. Mediante este espacio de nombres se pueden crear bases de datos de SQL Server CE en un dispositivo así como establecer conexiones a bases de datos de SQL Server que se encuentran en un dispositivo o un servidor remoto.

Nota: Este espacio de nombres es compatible únicamente con la versión 1.1 de .NET Framework.

#### **g. System.Data.SqlTypes**

Proporciona clases para los tipos de datos nativos de SQL Server. Estas clases proporcionan una alternativa más rápida y segura a otros tipos de datos. El uso de estas

clases en este espacio de nombres ayuda a evitar errores de conversión de tipos causados en situaciones en las que podría producirse una pérdida de precisión. Dado que otros tipos de datos se convierten en segundo plano a `SqlTypes` y viceversa, la creación y el uso explícitos de objetos en este espacio de nombres conllevan también un código más rápido.

## 1.6 System.Diagnostics

Proporciona clases que permiten interactuar con procesos del sistema, registros de eventos y contadores de rendimiento. Este espacio de nombres también proporciona clases que permiten depurar la aplicación y realizar un seguimiento de la ejecución del código. Para obtener más información, vea las clases `Trace` y `Debug`.

### a. System.Diagnostics.SymbolStore

Proporciona clases que permiten leer y escribir información de símbolos de depuración, como una línea de código fuente de las asignaciones del lenguaje intermedio de Microsoft (MSIL). Los compiladores para .NET Framework pueden almacenar la información de símbolos de depuración en los archivos de base de datos del programador (PDB). Los depuradores y las herramientas del generador de perfiles de código pueden leer la información de símbolos de depuración en tiempo de ejecución.

### b. System.DirectoryServices

Proporciona fácil acceso a Active Directory desde código administrado. El espacio de nombres contiene dos clases de componentes, `DirectoryEntry` y `DirectorySearcher`, que utilizan la tecnología ADSI (Interfaces de servicio de Active Directory). ADSI es el conjunto de interfaces que Microsoft proporciona como una herramienta flexible que permite trabajar con un gran número de proveedores de red. ADSI permite al administrador localizar y administrar con relativa facilidad los recursos de una red, independientemente del tamaño de esta última.

## 1.7 System.Drawing

Proporciona acceso a la funcionalidad básica de gráficos de GDI+. Los espacios de nombres `System.Drawing.Drawing2D`, `System.Drawing.Imaging` y `System.Drawing.Text` proporcionan funcionalidades más avanzadas.

### a. System.Drawing.Design

Contiene clases que amplían la lógica y el dibujo de las interfaces de usuario (UI) en tiempo de diseño. Se puede ampliar aún más esta funcionalidad en tiempo de diseño para crear elementos de cuadro de herramientas personalizados, editores de valores específicos de tipos que pueden editar y representar gráficamente valores de los tipos admitidos, o convertidores de tipos que pueden realizar conversiones de valores entre determinados tipos. Este espacio de nombres proporciona el marco de trabajo básico para desarrollar extensiones a la interfaz de usuario en tiempo de diseño.

### b. System.Drawing.Drawing2D

Proporciona funcionalidad bidimensional y de gráficos vectoriales avanzada. Este espacio de nombres incluye los pinceles degradados, la clase `Matrix` (que se utiliza para definir transformaciones geométricas) y la clase `GraphicsPath`.

### c. System.Drawing.Imaging

Proporciona funcionalidad avanzada de procesamiento de imágenes de GDI+. La funcionalidad gráfica básica se proporciona mediante el espacio de nombres System.Drawing.

#### **d. System.Drawing.Printing**

Proporciona servicios relacionados con la impresión. Normalmente, se crea una nueva instancia de la clase PrintDocument, se establecen las propiedades que describen lo que se va a imprimir, y se llama al método Print para imprimir el documento.

#### **e. System.Drawing.Text**

Proporciona funcionalidad tipográfica avanzada de GDI+. La funcionalidad gráfica básica se proporciona mediante el espacio de nombres System.Drawing. Las clases en este espacio de nombres permiten a los usuarios crear y utilizar colecciones de fuentes.

### **1.8 System.EnterpriseServices**

Proporciona una infraestructura importante para aplicaciones empresariales. COM+ proporciona una arquitectura de servicios para los modelos de programación de componentes implementados en un entorno empresarial. Este espacio de nombres proporciona objetos de .NET Framework con acceso a servicios COM+, por lo que los objetos de .NET Framework son más prácticos para las aplicaciones empresariales.

#### **a. System.EnterpriseServices.CompensatingResourceManager**

Proporciona clases que permiten utilizar un Administrador de compensación de recursos (CRM) en código administrado. Un CRM es un servicio proporcionado por COM+ que permite incluir objetos no transaccionales en las transacciones del Coordinador de transacciones distribuidas de Microsoft (DTC). Aunque los CRM no ofrecen las funcionalidades de un administrador de recursos completo, proporcionan la atomicidad transaccional (comportamiento todo o nada) y la durabilidad por medio del registro de recuperación.

#### **b. System.EnterpriseServices.Internal**

Proporciona compatibilidad de infraestructura para los servicios COM+. Las clases e interfaces en este espacio de nombres están diseñadas específicamente para admitir llamadas a System.EnterpriseServices desde las clases COM+ no administradas.

### **1.9 System.Globalization**

Contiene clases que definen datos relativos a la referencia cultural, tales como el idioma, el país o la región, el calendario, los formatos de fecha, la moneda y los números, así como el criterio de ordenación para las cadenas. Estas clases resultan útiles para escribir aplicaciones globalizadas (de uso internacional).

### **1.10 System.IO**

Contiene tipos que permiten la lectura y escritura sincrónicas y asincrónicas en archivos y secuencias de datos.

#### **a. System.IO.IsolatedStorage**

Contiene tipos que permiten crear y utilizar almacenes aislados. Estos almacenes permiten leer y escribir datos a los que los fragmentos de código de menos confianza no pueden tener acceso y permiten evitar la exposición de información confidencial que se pueda guardar en cualquier lugar del sistema de archivos. Los datos se almacenan en

compartimentos aislados por el usuario actual y por el ensamblado en el que existe el código.

### **1.11 System.Management**

Proporciona acceso a un variado conjunto de datos y eventos de administración relativos al sistema, dispositivos y aplicaciones instrumentados para Windows Management Infrastructure (WMI).

#### **a. System.Management.Instrumentation**

Proporciona las clases necesarias para instrumentar aplicaciones de administración y exponer sus datos y eventos de administración a posibles consumidores a través de WMI. De este modo, los consumidores como Microsoft Application Center o Microsoft Operations Manager podrán administrar fácilmente la aplicación, que se podrá supervisar y configurar mediante secuencias de comandos de administrador u otras aplicaciones, tanto administradas como no administradas.

### **1.12 System.Messaging**

Proporciona clases que permiten conectar con colas de mensajes de la red, supervisarlas, administrarlas y enviar, recibir o leer mensajes.

### **1.13 System.Net**

Proporciona una interfaz de programación sencilla para muchos de los protocolos que se utilizan en las redes actuales. Las clases WebRequest y WebResponse constituyen la base de los "protocolos conectables", una implementación de servicios de red que permite desarrollar aplicaciones que usan recursos de Internet sin preocuparse por los detalles específicos de los protocolos individuales.

#### **a. System.Net.Sockets**

Proporciona una implementación administrada de la interfaz de Windows Sockets (Winsock) para programadores que tienen que controlar rigurosamente el acceso a la red.

### **1.14 System.Reflection**

Contiene clases e interfaces que proporcionan una vista administrada de los tipos cargados, métodos y campos, con la capacidad de crear e invocar dinámicamente los tipos.

#### **a. System.Reflection.Emit**

Contiene clases que permiten a un compilador o una herramienta emitir metadatos así como lenguaje intermedio de Microsoft (MSIL) y, opcionalmente, generar un archivo PE en el disco. Los principales clientes de estas clases son los motores de secuencias de comandos y los compiladores.

### **1.15 System.Resources**

Proporciona clases e interfaces que permiten a los programadores crear, almacenar y administrar varios recursos específicos de la referencia cultural utilizados en una aplicación.

#### **a. System.Runtime.CompilerServices**

Proporciona funcionalidad a los programadores de compiladores que utilicen código administrado para especificar atributos en los metadatos que afectan al comportamiento

de Common Language Runtime en tiempo de ejecución. Las clases en este espacio de nombres son sólo para los programadores de compiladores.

#### **b. System.Runtime.InteropServices**

Proporciona una amplia variedad de miembros compatibles con la interoperabilidad COM y los servicios de invocación de plataforma. Si no conoce estos servicios, vea Interoperar con código no administrado.

- **System.Runtime.InteropServices.CustomMarshalers**  
Es compatible con la infraestructura de .NET y no está diseñado para que se utilice directamente desde el código.
- **System.Runtime.InteropServices.Expando**  
Contiene la interfaz IExpando, que permiten modificar un objeto agregando o quitando sus miembros.

#### **c. System.Runtime.Remoting**

Proporciona clases e interfaces que permiten a los programadores crear y configurar aplicaciones distribuidas.

- **System.Runtime.Remoting.Activation**  
Proporciona clases y objetos que admiten la activación de objetos remotos en el servidor o el cliente.
- **System.Runtime.Remoting.Channels**  
Contiene clases que admiten y controlan canales y receptores de canales, que se utilizan como medio de transporte cuando un cliente llama a un método de un objeto remoto.

##### **System.Runtime.Remoting.Channels.Http**

Contiene canales que utilizan el protocolo HTTP para llevar mensajes y objetos a ubicaciones remotas o viceversa. De forma predeterminada, los canales HTTP codifican las llamadas a objetos y métodos en formato SOAP para la transmisión, pero se pueden especificar otros receptores de formateadores de codificación y decodificación en las propiedades de configuración de un canal.

##### **System.Runtime.Remoting.Channels.Tcp**

Contiene canales que utilizan el protocolo TCP para llevar mensajes y objetos a ubicaciones remotas o viceversa. De forma predeterminada, los canales TCP codifican las llamadas a objetos y métodos en formato binario para la transmisión, pero se pueden especificar otros receptores de formateadores de codificación y decodificación en las propiedades de configuración de un canal.

- **System.Runtime.Remoting.Contexts**  
Contiene objetos que definen los contextos en los que residen todos los objetos. Un contexto es una serie ordenada de propiedades que define un entorno para los objetos que contiene. Los contextos se crean durante el proceso de activación de los objetos configurados para requerir determinados servicios automáticos, como la sincronización, las

transacciones, la activación just-in-time (JIT), la seguridad, etc. En un contexto pueden residir varios objetos.

- **System.Runtime.Remoting.Lifetime**

Contiene clases que administran la duración de los objetos remotos. Tradicionalmente, la recolección distribuida de elementos no utilizados recurre a los recuentos de referencias y comandos ping para controlar la duración de los objetos. Esto funciona bien cuando hay pocos clientes por servicio, pero no se ajusta adecuadamente cuando hay miles de clientes por cada servicio. El servicio de duración remoto asocia una concesión a cada servicio y, cuando ésta vence, elimina el servicio. El servicio de duración puede desempeñar la función de un colector distribuido de elementos no utilizados tradicional y se adapta bien al aumentar el número de clientes por servidor.

- **System.Runtime.Remoting.Messaging**

Contiene clases que se utilizan para crear mensajes y procesarlos de forma remota. La infraestructura de interacción remota utiliza mensajes para comunicarse con objetos remotos. Los mensajes se utilizan para transmitir llamadas a métodos remotos, activar objetos remotos y comunicar información. Un objeto de mensaje lleva una serie de propiedades con nombre, entre las que se incluyen identificadores de acción, información de envío y parámetros.

- **System.Runtime.Remoting.Metadata**

Contiene clases y atributos que se pueden utilizar para personalizar la generación y el procesamiento de SOAP para objetos y campos. Las clases en este espacio de nombres se pueden utilizar para indicar el encabezado SOAPAction, el tipo de resultado, el nombre de elemento XML y el identificador URI del espacio de nombres XML del método.

**System.Runtime.Remoting.Metadata.W3cXsd2001**

Contiene la definición de esquema XML (XSD) establecida por el Consorcio World Wide Web (W3C) en 2001. La especificación de W3C titulada "XML Schema Part2: Data types" identifica el formato y el comportamiento de varios tipos de datos. Este espacio de nombres contiene clases contenedoras para los tipos de datos que se ajustan a la especificación de W3C. Todos los tipos de fecha y hora se ajustan a la especificación de estándares ISO.

- **System.Runtime.Remoting.MetadataServices**

Contiene las clases utilizadas por la herramienta de línea de comandos Soapsuds.exe y el código de usuario para convertir metadatos a esquemas XML, o viceversa, para la infraestructura de interacción remota.

- **System.Runtime.Remoting.Proxies**

Contiene clases que controlan y proporcionan funcionalidad a los servidores proxy. Un servidor proxy es un objeto local que constituye una imagen de un objeto remoto. Los servidores proxy permiten a los clientes tener acceso a objetos más allá de los límites remotos.

- **System.Runtime.Remoting.Services**

Contiene clases de servicios que proporcionan funcionalidad a .NET Framework.

#### **d. System.Runtime.Serialization**

Contiene clases que se pueden utilizar para serializar y deserializar objetos. La serialización es el proceso de convertir un objeto o un gráfico de objetos en una secuencia lineal de bytes para su almacenamiento o transmisión a otra ubicación. La deserialización es el proceso de volver a generar los objetos a partir de la información almacenada.

- **System.Runtime.Serialization.Formatters**

Proporciona enumeraciones, interfaces y clases comunes que utilizan los formateadores de serialización.

- **System.Runtime.Serialization.Formatters.Binary**

Contiene la clase BinaryFormatter, que se puede utilizar para serializar y deserializar objetos en formato binario.

- **System.Runtime.Serialization.Formatters.Soap**

Contiene la clase SoapFormatter, que se puede utilizar para serializar y deserializar objetos en formato SOAP.

### **1.16 System.Security**

Proporciona la estructura subyacente del sistema de seguridad de .NET Framework, incluidas clases base relativas a permisos.

#### **a. System.Security.Cryptography**

Proporciona servicios criptográficos, incluidas la codificación y decodificación seguras de datos, así como muchas otras operaciones, como los cálculos hash, la generación de números aleatorios y la autenticación de mensajes.

- **System.Security.Cryptography.X509Certificates**

Contiene la implementación de Common Language Runtime del certificado Authenticode X.509 v.3. Este certificado se firma con una clave privada que identifica de forma única y segura al propietario del certificado.

- **System.Security.Cryptography.Xml**

Contiene clases que admiten la creación y validación de firmas digitales XML. Las clases en este espacio de nombres implementan la recomendación del Consorcio World Wide Web, "XML-Signature Syntax and Processing", descrita en <http://www.w3.org/TR/xmlsig-core/>.

#### **b. System.Security.Permissions**

Define clases que controlan el acceso a operaciones y recursos a partir de una directiva.

#### **c. System.Security.Policy**

Contiene grupos de código, condiciones de pertenencia y evidencia. Estos tres tipos de clase se utilizan para crear las reglas que aplica el sistema de directivas de seguridad de

.NET Framework. Las clases de evidencia son la entrada de la directiva de seguridad y las condiciones de pertenencia son los modificadores; juntas, crean instrucciones de directiva y determinan el conjunto de permisos concedido. Los niveles de directivas y los grupos de código constituyen la estructura de la jerarquía de directivas. Los grupos de código son la encapsulación de una regla y se organizan jerárquicamente en un nivel de directiva.

#### **e. System.Security.Principal**

Define un objeto principal que representa el contexto de seguridad en el cual se ejecuta código.

### **1.17 System.ServiceProcess**

Proporciona clases que permiten implementar, instalar y controlar las aplicaciones de servicios para Windows. Los servicios son ejecutables de ejecución prolongada que no utilizan interfaz de usuario. La implementación de un servicio implica heredar de la clase ServiceBase y definir el comportamiento específico que se debe procesar cuando se pasen comandos para iniciar, detener, pausar y continuar, así como el comportamiento personalizado y las acciones que se deben realizar cuando se cierre el sistema.

### **1.18 System.Text**

Contiene clases que representan codificaciones de caracteres ASCII, Unicode, UTF-7 y UTF-8, clases base abstractas para realizar conversiones entre bloques de caracteres y bloques de bytes, así como una clase auxiliar que manipula objetos String y les aplica formato sin crear instancias intermedias de String.

#### **a. System.Text.RegularExpressions**

Contiene clases que proporcionan acceso al motor de expresiones regulares de .NET Framework. El espacio de nombres proporciona funcionalidad de expresiones regulares que se puede utilizar desde cualquier plataforma o lenguaje que se ejecute con Microsoft .NET Framework.

### **1.19 System.Threading**

Proporciona clases e interfaces que habilitan la programación multiproceso. Además de las clases que permiten sincronizar las actividades de subproceso y obtener acceso a datos (Mutex, Monitor, Interlocked, AutoResetEvent, etc.), este espacio de nombres incluye una clase ThreadPool que permite utilizar un grupo de subprocesos proporcionados por el sistema, y una clase Timer que ejecuta los métodos de devolución de llamada en los subprocesos del grupo de subprocesos.

### **1.20 System.Timers**

Proporciona el componente Timer, que permite provocar un evento en un intervalo especificado.

### **1.21 System.Web**

Proporciona clases e interfaces que permiten la comunicación entre explorador y servidor. Este espacio de nombres incluye la clase HttpRequest, que facilita amplia información sobre la solicitud HTTP actual, la clase HttpResponse, que administra el resultado HTTP dirigido al cliente, y el objeto HttpServerUtility, que proporciona acceso a las utilidades y los procesos que residen en el servidor. System.Web también incluye clases para la manipulación de

cookies, la transferencia de archivos, la información de excepciones y el control de la caché de resultados.

**a. System.Web.Caching**

Proporciona clases para almacenar en caché los datos que se utilizan con frecuencia en el servidor. Incluye la clase Cache, un diccionario que permite almacenar los objetos de datos arbitrarios, como tablas hash y conjuntos de datos. Proporciona asimismo la funcionalidad de vencimiento para dichos objetos así como métodos que permiten agregar y quitar los objetos. Se pueden agregar también los objetos con dependencia de otros archivos o entradas en la caché, además de realizar devoluciones de llamada para notificar a la aplicación cuando se quita un objeto de la memoria caché.

**b. System.Web.Configuration**

Contiene clases que se utilizan para configurar ASP.NET.

- **System.Web.Hosting**

Proporciona la funcionalidad necesaria para alojar aplicaciones ASP.NET de aplicaciones administradas fuera de los servicios de Microsoft Internet Information Server (IIS).

**c. System.Web.Mail**

Contiene clases que permiten crear y enviar mensajes mediante el componente de mensaje CDOSYS. El mensaje de correo se entrega a través del servicio de correo SMTP integrado en Microsoft Windows 2000 o a través de un servidor SMTP arbitrario. Las clases en este espacio de nombres se pueden utilizar desde ASP.NET o cualquier aplicación administrada.

**d. System.Web.Mobile**

Contiene las principales funciones, incluidos la autenticación y el control de errores, necesarias para generar las aplicaciones Web móviles de ASP.NET.

**e. System.Web.Security**

Contiene clases que se utilizan para implementar la seguridad de ASP.NET en aplicaciones de servidor Web.

**f. System.Web.Services**

Se compone de las clases que permiten crear servicios Web XML mediante clientes de servicios Web XML y ASP.NET. Los servicios Web XML son aplicaciones que permiten intercambiar mensajes en un entorno de correspondencia imprecisa mediante protocolos estándar como HTTP, XML, XSD, SOAP y WSDL. Los servicios Web XML permiten generar aplicaciones modulares ubicadas dentro y fuera de empresas de entornos heterogéneos, por lo que pueden interoperar con una amplia variedad de implementaciones, plataformas y dispositivos. Los mensajes XML basados en SOAP de estas aplicaciones pueden tener partes debidamente definidas (estructuradas y con establecimiento de tipos) o pueden tener partes definidas de manera imprecisa (mediante XML arbitrario). La capacidad de los mensajes de evolucionar con el tiempo sin romper el protocolo es fundamental para la flexibilidad y la solidez de los servicios Web XML como componente básico para el futuro del Web.

- **System.Web.Services.Configuration**

Consta de las clases que configuran la forma en que se ejecutan los servicios Web XML creados con ASP.NET.

- **System.Web.Services.Description**

Consta de las clases que permiten describir públicamente un servicio Web XML mediante el Lenguaje de descripción de servicios Web (WSDL). Cada clase en este espacio de nombres se corresponde con un elemento específico de la especificación WSDL, y la jerarquía de clases se corresponde con la estructura XML de un documento WSDL válido.

- **System.Web.Services.Discovery**

Consta de las clases que permiten a los clientes de servicios Web XML encontrar los servicios Web XML disponibles en un servidor Web mediante un proceso denominado Descubrimiento de servicios Web XML.

- **System.Web.Services.Protocols**

Consta de las clases que definen los protocolos utilizados para transmitir datos a través de la red durante la comunicación entre los clientes del servicio Web XML y los servicios Web XML creados con ASP.NET.

#### **g. System.Web.SessionState**

Proporciona las clases e interfaces que habilitan el almacenamiento de los datos específicos de un cliente individual en una aplicación Web del servidor. Los datos relativos al estado de la sesión se utilizan para ofrecer al cliente la apariencia de una conexión permanente con la aplicación. La información de estado se puede almacenar en la memoria del proceso local o, en el caso de las configuraciones de baterías de servidores Web, fuera de proceso mediante el servicio de estado de ASP.NET o una base de datos de SQL Server.

#### **h. System.Web.UI**

Proporciona las clases e interfaces que permiten crear los controles y páginas que aparecerán en las aplicaciones Web como interfaz de usuario de una página Web. Este espacio de nombres incluye la clase Control, que proporciona a todos los controles, ya sean controles HTML, controles Web o controles de usuario, un conjunto común de funcionalidades. También incluye el control Page, que se genera automáticamente siempre que se realiza una solicitud de una página de la aplicación Web. También se proporcionan clases que ofrecen la funcionalidad de enlace de datos a los controles de servidor de formularios Web Forms, la capacidad de guardar el estado de la vista de un control o una página determinada, así como la funcionalidad de análisis para los controles programables y literales.

- **System.Web.UI.Design**

Contiene clases que se pueden utilizar para ampliar la compatibilidad en tiempo de diseño de Web Forms.

##### **System.Web.UI.Design.WebControls**

Contiene clases que se pueden utilizar para ampliar la compatibilidad en tiempo de diseño de los controles de servidor Web.

- **System.Web.UI.HtmlControls**

Se compone de una colección de clases que permiten crear controles de servidor HTML en una página de formularios Web Forms. Los controles de servidor HTML se ejecutan en el servidor y se asignan directamente a las etiquetas HTML estándar compatibles con la mayoría de los exploradores. Esto permite controlar mediante programación los elementos HTML de una página de formularios Web Forms.

- **System.Web.UI.MobileControls**

Contiene un conjunto de controles de servidor ASP.NET que pueden procesar de manera inteligente la aplicación para diferentes dispositivos móviles.

**System.Web.UI.MobileControls.Adapters**

Contiene las principales clases de adaptadores de dispositivo que utilizan los controles móviles de ASP.NET para la personalización de dispositivos y la compatibilidad extendida de dispositivos.

- **System.Web.UI.WebControls**

Contiene clases que permiten crear controles de servidor Web en una página Web. Los controles de servidor Web se ejecutan en el servidor e incluyen controles de formulario como botones y cuadros de texto. También incluyen controles para fines especiales como un calendario. Dado que los controles de servidor Web se ejecutan en el servidor, se pueden controlar estos elementos mediante programación. Los controles de servidor Web son más abstractos que los controles de servidor HTML. Su modelo de objetos no refleja necesariamente la sintaxis HTML.

## **I . System.Windows.Forms**

Contiene clases para crear aplicaciones basadas en Windows y que aprovechan plenamente las características avanzadas de interfaz de usuario disponibles en este sistema operativo.

- **System.Windows.Forms.Design**

Contiene clases que son compatibles con la configuración y el comportamiento en tiempo de diseño de los componentes de formularios Windows Forms. Estas clases se componen de: clases de diseñador que proporcionan compatibilidad con los componentes de formularios Windows Forms, un conjunto de servicios en tiempo de diseño, clases UITypeEditor para configurar determinados tipos de propiedades, así como clases para importar controles ActiveX.

### **1.21 System.Xml**

Proporciona compatibilidad basada en estándares para procesar XML.

#### **a. System.Xml.Schema**

Contiene las clases XML que proporcionan compatibilidad basada en estándares para los esquemas XSD.

**b. System.Xml.Serialization**

Contiene clases que se utilizan para serializar objetos en documentos o secuencias de formato XML.

**c. System.Xml.XPath**

Contiene el analizador y el motor de evaluación de XPath. Es compatible con la recomendación de W3C titulada "XML Path Language (XPath) Version 1.0" ([www.w3.org/TR/xpath](http://www.w3.org/TR/xpath)).

**d. System.Xml.Xsl**

Proporciona compatibilidad con las transformaciones XSLT (Extensible Stylesheet Transformation). Es compatible con la recomendación de W3C titulada "XSL Transformations (XSLT) Version 1.0" ([www.w3.org/TR/xslt](http://www.w3.org/TR/xslt)).