

Argentina: \$5,90 (recargo al interior \$0,20) México: \$30

USERS

Microsoft®

Curso teórico y práctico de programación

Desarrollador .net

Con toda la potencia de **Visual Basic .NET** y **C#**

25 fascículos | 600 páginas | 1 Libro
3 CDs con las versiones express de Visual Basic .NET, C# y SQL Server

1

La mejor forma de aprender a programar desde cero



Basado en el programa Desarrollador Cinco Estrellas de Microsoft



9 789871 347438

Curso teórico y práctico

Desarrollador .net



www.reduserspremium.blogspot.com.ar

La mejor forma de aprender
a programar desde cero.

RedUSERS

COMUNIDAD DE TECNOLOGIA



EL SITIO Nº1 DE TECNOLOGIA

Noticias al instante // Entrevistas y coberturas exclusivas //
Análisis y opinión de los máximos referentes // Reviews de
productos // Trucos para mejorar la productividad //
Regístrate, participa, y comparte tus opiniones



SUSCRIBITE

SIN CARGO A CUALQUIERA
DE NUESTROS NEWSLETTERS
Y RECIBÍ EN TU CORREO
ELECTRÓNICO TODA LA
INFORMACIÓN DEL UNIVERSO
TECNOLÓGICO ACTUALIZADA
AL INSTANTE



INGRESÁ A
redusers.com/suscribirse-al-newsletter
¡Y REGÍSTRATE YA!

www.reduserspremium.blogspot.com.ar



Foros



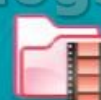
Encuestas



Tutoriales



Agenda de eventos



Videos



¡Y mucho más!



redusers.com

Seguinos en:



www.facebook.com/redusers



www.twitter.com/redusers



www.youtube.com/redusersvideos

Desarrollador .net

El más completo material visual y práctico para convertir nuestro hobby en una verdadera profesión.

El objetivo principal de esta obra es formar desde el inicio a los lectores en una de las profesiones que mayor salida laboral tienen hoy en día: la de desarrollador. Está destinada tanto a quienes recién quieren dar sus primeros pasos en este apasionante mundo, como también a quienes ya poseen algunos conocimientos, pero buscan perfeccionarse.

La plataforma

Al momento de decidirnos por esta profesión, lo primero que debemos saber es que, en la actualidad, existen muchas tecnologías, como Java (utilizada para programar aplicaciones para celulares y empresariales), PHP (para desarrollo de sitios de Internet) y .NET (para aplicaciones de escritorio, empresariales y móviles), entre otras. En este punto, surge la primera pregunta: ¿qué tecnología conviene aprender? La respuesta es más sencilla de lo que parece: para elegir una tecnología, debemos conocer cuál es el sistema operativo más utilizado por los usuarios, ya sea en su casa o en la oficina. Las estadísticas dicen que, por abrumadora mayoría, se trata de Windows, sistema que, como sabemos, pertenece a la empresa Microsoft. Y es esta empresa la que, a su vez, cuenta con numerosas herramientas para desarrollar aplicaciones. La principal ventaja es que las herramientas que ofrece Microsoft para la creación de aplicaciones son las mismas que utilizan en esa empresa para llevar adelante sus propios desarrollos. Esto nos da mayor

confianza y compatibilidad, dos aspectos fundamentales sobre los que volveremos insistentemente a lo largo de toda la obra.

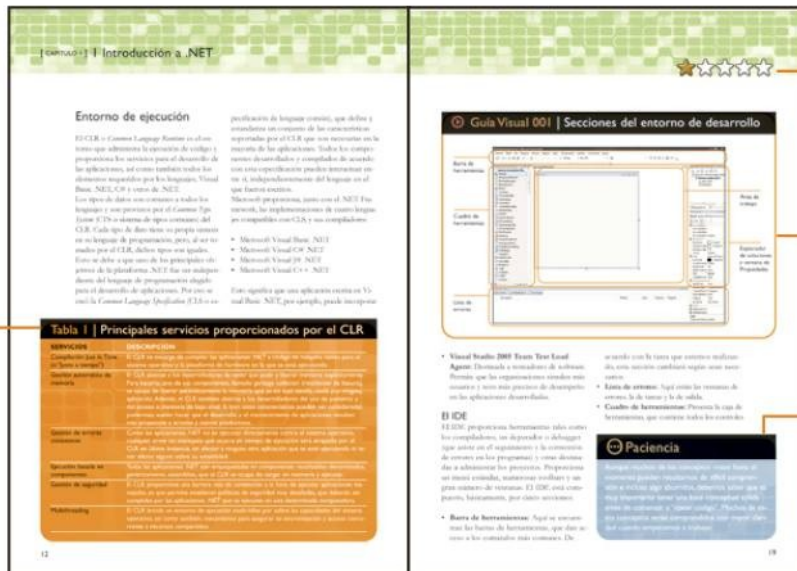
Contenido de la obra

El contenido de esta obra fue desarrollado por profesionales con conocimientos profundos acerca de cada una de las herramientas. A lo largo de las páginas de este curso, cada uno de los conceptos es tratado en forma didáctica, para que, incluso, pueda ser comprendido por quienes no poseen ningún conocimiento. Los temas serán presentados en forma simple y clara, con el apoyo de diagramas didácticos que ayudarán a llevar adelante un aprendizaje exitoso. Además, daremos detalles acerca de cómo se deben configurar los distintos programas o herramientas a medida que necesitemos de ellos para realizar la práctica.

¡A programar!

Para lograr una mejor comprensión de los temas, en todo momento haremos referencia a ejemplos prácticos que nos permitirán aplicar los conceptos asimilados. Un buen programador es aquel que practica y adquiere experiencia a partir de los errores cometidos, y no quien sólo conoce los conceptos teóricos. Es por eso que también encontraremos claves y secretos que únicamente pueden obtenerse de la práctica profesional y que nos ayudarán a insertarnos finalmente en el mercado laboral.

▶ La obra | Visual y práctica



En cada tema encontraremos tablas que complementan los conceptos desarrollados.

En cada sección figura claramente identificada la estrella en la que nos encontramos.

Las guías visuales nos permiten conocer y aprender a utilizar las herramientas del entorno.

En todo el curso hallaremos recuadros que nos permitirán entender los conceptos más complejos.

▶ Los recuadros | Cómo identificarlos

➕ TRUCOS



? PROBLEMAS



* FAQ



☰ CONSEJOS

! ATENCIÓN



💰 GLOSARIO



Desarrollador Cinco Estrellas

La organización general de este curso estará apoyada en el programa DCE (Desarrollador Cinco Estrellas) de Microsoft. La selección de éste como base de estudio no es casual, ya que es uno de los programas de capacitación más reconocidos en Latinoamérica. El material proporcionado a lo largo de esta obra servirá de apoyo para quienes desean rendir los exámenes disponibles en el sitio de DCE, www.dce2005.com, cuya registración es gratuita. Este programa tiene la particularidad de estar dividido en estrellas, cada una de las cuales abarca una cierta cantidad de temas o conceptos. Cada estrella posee una certificación que sirve como manera de calificar el nivel de conocimiento que poseemos. La obra, a su vez, está estructurada en capítulos, que, como veremos, pertenecerán a diferentes estrellas del programa DCE (claramente identificadas en la parte superior de cada página). A medida que avancemos en la obra, indicaremos cuáles son los conceptos más importantes que debemos manejar para rendir los distintos exámenes, y al finalizar cada una de las etapas, haremos un resumen de lo visto, para poder repasar los conceptos más importantes y pasar con éxito las evaluaciones.

Material complementario

Además de los veinticinco fascículos, se incluyen CDs interactivos y un libro con los concepto teóricos.

CD1: Visual Basic edición Express

En este CD, encontraremos la versión Express del entorno de programación Visual Studio para el lenguaje Visual Basic (VB), además de varias aplicaciones complementarias que nos serán útiles a lo largo de la obra.



CD2: SQL Server 2005 edición Express

En este CD, se ofrece la versión Express del servidor de bases de datos SQL Server 2005.



Libro complementario

Este libro presenta todos los conceptos básicos que deben tenerse en cuenta antes de comenzar a programar. Por otra parte, contiene todos los temas que nos permitirán rendir el examen de la estrella inicial DCE.

CD2: C# edición Express

En este CD, se incluye la versión Express del entorno de programación Visual Studio para el lenguaje C# (C Sharp).

! Información

Si los CDs no se abren de manera automática, debemos dirigirnos a Mi PC y, desde allí, hacer doble clic en la unidad de CD-ROM (en la mayoría de los casos, la D). A continuación, hacemos doble clic en el archivo que tiene extensión .exe.



El programador

Todos oímos hablar de programadores o desarrolladores de software, pero poco sabemos de su historia.

Un programador es, básicamente, una persona que ejerce la programación; es decir, que se dedica a escribir programas para PC. Los programadores también reciben el nombre de desarrolladores de software; técnicamente, podemos usar uno u otro término.

Funciones del programador

El programador se encarga de implementar algoritmos mediante un lenguaje de programación que pueda entender la computadora. Inicialmente, su función era resolver en forma particular los problemas de sistemas que surgían en las empresas, así como un mecánico cambia o repara partes defectuosas. Con el tiempo, se fueron incorporando a su misión nuevas necesidades y tareas, hasta llegar al día de hoy, en que un programador o grupo de ellos se ocupa de realizar aplicaciones completas y generar soluciones empresariales.

La programación

Es la creación de un programa de computadora (código máquina compilado e interpretado) directamente desde el núcleo del sistema o desde un script (código fuente interpretado), esto es, un conjunto concreto de instrucciones que una computadora puede ejecutar. El programa se escribe en un lenguaje de programación (a continuación veremos de qué se trata esto), aunque también se puede escribir directamente en lenguaje de máquina, con cierta dificultad. El lenguaje de programación es un lenguaje que puede utilizarse para controlar el



La primera programadora

El primer contacto de Ada Lovelace con las máquinas fue en 1833, gracias a Charles Babbage, conocido matemático inglés y científico protoinformático, a quien se le atribuye la primera idea acerca de la concepción de una computadora, ya que su máquina analítica funcionaba con el mismo principio que las actuales. Ada conoció el trabajo de Babbage al acudir a una conferencia de Dionysus Lardner, quien disertaba en el Instituto de Mecánica sobre la máquina de Babbage. A partir de ese momento, comenzó a colaborar con él. Es considerada la primera programadora de la historia, y su contribución consistió en un método para calcular los números de Bernoulli en la máquina analítica de Babbage.

comportamiento de una máquina, particularmente, una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. Aunque muchas veces lenguaje de programación y lenguaje informático se usan como si fuesen sinónimos, no tiene por qué ser así, ya que los informáticos engloban a los de programación y a otros más.

Un lenguaje de programación permite a un programador especificar de manera precisa sobre qué datos una computadora debe operar, cómo éstos deben ser almacenados y transmitidos, y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto se realiza a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con Lexico.

Perspectiva laboral

Son muchas las posibilidades laborales que actualmente tiene un desarrollador. En principio, la mayoría de las empresas cuentan con sistemas informáticos propios. Es aquí donde se vuelve necesario tener programadores capacitados que puedan realizar el mantenimiento, la modificación y la actualización de estos sistemas. Además de poder trabajar para

una empresa como desarrolladores, también tendremos la posibilidad de hacerlo de forma independiente o de armar nuestro propio equipo de desarrollo. En forma independiente podremos ofrecer soluciones en base a las necesidades de empresas o clientes, como diseños de aplicaciones Web. Por el lado de equipos de trabajo, podremos hacer herramientas a pedido de alguna empresa.

Invertir nuestro tiempo en aprender a desarrollar sobre una de las tecnologías más usadas en el mercado (hablamos de .NET) nos permitirá obtener un empleo que hoy en día es muy bien retribuido. Todos los conocimientos adquiridos a lo largo de esta obra nos servirán para formarnos con los requerimientos necesarios para empezar en esta profesión. Además, contamos con la posibilidad de rendir los exámenes del programa **Desarrollador Cinco Estrellas, de Microsoft**, en el cual se nos darán certificados que demuestran la asimilación y el uso de determinados elementos de la tecnología .NET.

Por dónde empezar

El libro que acompaña a esta colección es de lectura obligatoria para aquellos que recién empiezan en el mundo de la programación. En él se explican los conceptos básicos, necesarios para comprender y poder realizar las prácticas presentadas a lo largo de esta colección. Además de introducirnos en lo que es la tecnología .NET, también nos permitirá estar capacitados para rendir el primero de los exámenes (Estrella 0) del programa Desarrollador Cinco Estrellas. Éste es muy importante, ya que, al aprobarlo, Microsoft nos permitirá comenzar a rendir los correspondientes a la Estrella 1 para, así, alcanzar un nivel tal que podamos empezar a realizar pequeños desarrollos de aplicaciones.

Fecha importante

El 13 de septiembre de los años no bisieptos se celebra el Día del programador. Por consenso, se eligió ese día ya que es el día número 256 del año, número importante para la programación por ser la cantidad de valores diferentes que puede tener un byte (desde el 00000000 hasta el 11111111), o lo que es más simple, 2^8 .



Introducción a .NET

Lo primero que debemos saber

Contenidos

A partir de ahora, empezaremos a conocer cuáles son los primeros pasos que debemos dar. Veremos las principales herramientas que nos permitirán desarrollar aplicaciones y construiremos nuestro primer programa totalmente funcional, que nos demostrará el potencial que podemos generar como programadores profesionales

Temas tratados

- » Qué es .NET
- » Cómo está compuesta la plataforma de desarrollo
- » Cuáles son las distintas versiones que ofrece la plataforma .NET
- » Cada una de las partes del entorno con el cual trabajaremos
- » Instalación del entorno que nos permitirá generar aplicaciones
- » Las principales bibliotecas, librerías y componentes que darán funcionalidad a nuestros proyectos
- » Desarrollo de nuestra primera aplicación totalmente funcional

Introducción a .NET

En esta etapa veremos cuáles son los principales conceptos que debemos conocer para iniciarnos como programadores .NET.

»» Qué es .NET

Para dar los primeros pasos, comenzaremos por realizar y analizar las principales características que nos ofrece la plataforma de desarrollo .NET.

- > Características que nos ofrece la plataforma
- > Principales elementos
- > Versiones

»» Plataforma de desarrollo

Nos introduciremos en el conocimiento de las herramientas con las cuales tendremos que familiarizarnos para empezar a desarrollar.

- > Entorno de ejecución
- > Bibliotecas de funcionalidad
- > Lenguajes de programación

»» El entorno de desarrollo

Conoceremos las herramientas, características y funciones del entorno de desarrollo con el que trabajaremos para convertirnos en programadores.

- > Elementos
- > Los menús del entorno
- > La barra de herramientas

»» Versiones Express

Conoceremos en detalle la versión Express de Visual Studio que se incluye en los CDs junto a esta obra. Todas las características y los pasos para hacerla funcionar.

- > Características
- > Ventajas y desventajas de cada versión
- > Modo de instalación

»» Nuestro primer programa: un navegador Web

Realizaremos nuestra primera aplicación, escribiendo sólo una línea de código, que nos permitirá crear un navegador Web.

- > Utilizar el entorno
- > Manejo de formularios
- > Ejecutar el programa



¿Qué es .NET?

A continuación, haremos una introducción a .NET, y veremos cuáles son las principales características que debemos conocer.

Para llegar a convertirnos en profesionales del desarrollo, no podemos omitir ningún concepto; por ese motivo comenzaremos por conocer cuáles son los componentes que ofrece la tecnología con la cual vamos a trabajar.

Microsoft .NET es una plataforma de desarrollo y ejecución de aplicaciones; es decir, proporciona los elementos necesarios para el desarrollo de aplicaciones de software, y todos los mecanismos de seguridad y eficiencia para asegurar su óptima ejecución.

Veamos a continuación cuáles son algunas de sus principales características:

- Las aplicaciones de .NET se ejecutan en un entorno aislado del sistema operativo denominado “runtime”, lo que lo hace flexible, seguro y portable.
- Es 100% orientado a objetos.
- Permite desarrollar aplicaciones en más de un lenguaje de programación.
- Está diseñado para permitir el desarrollo de aplicaciones corporativas complejas, robustas y flexibles.
- Proporciona un único modelo de programación consistente para el desarrollo de diferentes modelos de aplicaciones (Windows, Web, de consola, móviles, etc.) y para diferentes dispositivos de hardware (PC, Tablet PC y Pocket PC, entre otros).
- Puede integrarse fácilmente con las aplicaciones desarrolladas en modelos anteriores, como COM. Es posible utilizar elementos COM en los desarrollos .NET, y viceversa.
- Integra aplicaciones de otras plataformas y sistemas operativos, al implementar estándares como XML, SOAP, WSDL, etc.

Los principales elementos que constituyen la plataforma .NET son los siguientes:

- **.NET Framework:** Es el componente fundamental, que contiene los elementos necesarios para la creación y ejecución de las aplicaciones. Está compuesto, básicamente, por dos elementos: el entorno de ejecución de aplicaciones y las bibliotecas base.
- El entorno de ejecución de aplicaciones o runtime es el denominado *Common Language Runtime* (CLR). Entre sus principales funciones, se ocupa de:
 - Administrar la memoria de forma inteligente: Esto significa liberar y controlar la memoria de manera automática.
 - Realizar el aislamiento de aplicaciones: Si una aplicación deja de funcionar, no afecta a otra en ejecución ni al sistema.
 - Brindar seguridad en la ejecución de los componentes: Basa su ejecución en la información proporcionada por ellos (metadata), que indica cómo debe ejecutarse, qué versión utilizar y bajo qué contexto de seguridad, entre otras cosas.

Component Object Model

Es la plataforma para desarrollo de componentes de software introducida por Microsoft antes de .NET. El término COM es usado comúnmente en el desarrollo de software como un término que abarca las tecnologías OLE, OLE Automation, ActiveX, COM+ y DCOM. Podemos decir entonces que .NET es la evolución de COM.

- Generar código nativo: La ejecución de los componentes se realiza con un compilador *Just In Time* (JIT), que traduce los componentes al código nativo según la CPU en que se encuentre, y los ejecuta en su entorno.
- **Base Class Library (BCL)** o biblioteca base: Proporciona todos los componentes y clases necesarios para el desarrollo de aplicaciones en la plataforma. Éstos incluyen la mayoría de las funcionalidades que los programadores aplican de manera cotidiana y, además, un conjunto de clases específicas de acuerdo con la tecnología que se va a utilizar en la creación de las aplicaciones. Todos estos componentes están divididos en tres grupos principales:
 - Windows Forms
 - ASP.NET y Servicios Web XML
 - ADO.NET
- **Lenguajes de programación y compiladores:** Los lenguajes de programación permiten el desarrollo de aplicaciones sobre la plataforma .NET. Hacen uso de las especificaciones del CLR y de las BCL para el desarrollo de las aplicaciones, y siguen un estándar denominado *Common Language Specification* (CLS), lo que hace que sean completamente compatibles entre sí. Existen muchos lenguajes para programar en .NET, como Visual Basic .NET, C# (C-Sharp), NET.COBOL, y otros. Dado que el CLS es una especificación abierta, cualquier lenguaje que cumpla con ella puede ser utilizado para desarrollar en .NET.
- **Herramientas y documentación:** .NET proporciona un conjunto de utilitarios y herramientas de desarrollo que simplifican el proceso de creación de aplicaciones, así como también, documentación y guías de arquitectura, que describen las mejores prácticas de diseño, prueba e instalación de aplicaciones .NET.

* Sistemas operativos

El .NET Framework puede instalarse en cualquier sistema operativo de la familia Windows superior a Windows 98. Windows 2003 Server y Windows XP SP2 ya lo tienen instalado.

Tecnología .NET

- CLR (Common Language Runtime)
- BCL (Base Class Library)
- Lenguaje de programación
- Herramientas y documentación

FIGURA 001 | Éstos son los principales componentes que integran la plataforma .NET.



.NET Framework

El primer elemento que debemos conocer es el **.NET Framework** o “marco de trabajo”. Como ya mencionamos, es el componente fundamental de la plataforma Microsoft .NET, y es necesario tanto para desarrollar aplicaciones como para ejecutarlas luego. Tiene tres versiones, que pueden conseguirse en forma gratuita:

.NET FRAMEWORK REDISTRIBUTABLE PACKAGE

Es el componente de la plataforma .NET necesario para ejecutar aplicaciones. Este elemento se instala en los entornos de producción o estaciones de trabajo de los clientes y en las computadoras de los usuarios que vayan a utilizar aplicaciones .NET. Tiene dos elementos: el entorno de ejecución de la plataforma .NET (CLR, más adelante lo veremos en detalle) y las bibliotecas de funcionalidad reutilizable (BCL).

.NET FRAMEWORK SDK

Esta versión contiene herramientas de desarrollo de línea de comandos (compiladores, depuradores, etc.), documentación de referencia, ejemplos y manuales para programadores. En general, se instala en los entornos de desarrollo de aplicaciones, por lo que resulta más útil para programadores que para usuarios. Para instalar la versión SDK (*Software Development Kit*) es necesario instalar previamente el *Redistributable Package* (mencionado anteriormente).

.NET COMPACT FRAMEWORK

Se trata de una versión reducida del .NET Framework Redistributable, especialmente pensada para instalar en dispositivos móviles, como Pocket PCs y Smart Phones.

En este caso, cuando se trata de las aplicaciones de escritorio (WinForms o de Formularios) y las de consola (aplicaciones cuya interfaz de usuario es una consola de comandos

con textos solamente), el framework debe estar instalado en la PC del cliente (usuario), y en el servidor sólo si la aplicación será distribuida y tendrá parte de su funcionalidad centralizada en una única computadora. En el caso de las aplicaciones Web, el único requisito del lado del cliente o usuario es tener un navegador (browser) y una conexión de red al servidor, que debe tener instalado el .NET Framework. Finalmente, para las aplicaciones móviles, que se ejecutan sobre Windows Mobile en algún dispositivo tipo Pocket PC o Smart Phone, es preciso tener instalado el .NET Compact Framework en el dispositivo.

Versiones

Existen varias versiones de los frameworks mencionados:

- Versión 1.0: Liberada a principios del año 2002, incluía la versión 1.0 del .NET Framework, la versión 2002 de Visual Studio y varios lenguajes de programación nuevos compatibles con la plataforma (como C#.NET y Visual Basic .NET).
- Versión 1.1: Fue liberada en 2003, aproximadamente un año después que la 1.0. Esta versión introdujo el .NET Framework 1.1 junto con Visual Studio .NET 2003, la primera versión del .NET Compact Framework y un nuevo lenguaje de programación llamado J#.NET.
- Versión 2.0: Liberada a finales del año 2005, es la primera gran renovación que sufrió la plataforma en su tiempo de vida. Con la idea de ser una “evolución” en vez de una “revolución”, trajo consigo las versiones 2.0 del .NET Framework y el .NET Compact Framework, así como también, diferentes versiones de Visual Studio, lo cual representó un gran salto cualitativo en lo que respecta a productividad.

Entorno de ejecución

El CLR o *Common Language Runtime* es el entorno que administra la ejecución de código y proporciona los servicios para el desarrollo de las aplicaciones, así como también todos los elementos requeridos por los lenguajes, Visual Basic .NET, C# y otros de .NET.

Los tipos de datos son comunes a todos los lenguajes y son provistos por el *Common Type System* (CTS o sistema de tipos comunes) del CLR. Cada tipo de dato tiene su propia sintaxis en su lenguaje de programación, pero, al ser tomados por el CLR, dichos tipos son iguales. Esto se debe a que uno de los principales objetivos de la plataforma .NET fue ser independiente del lenguaje de programación elegido para el desarrollo de aplicaciones. Por eso se creó la *Common Language Specification* (CLS o es-

pecificación de lenguaje común), que define y estandariza un conjunto de las características soportadas por el CLR que son necesarias en la mayoría de las aplicaciones. Todos los componentes desarrollados y compilados de acuerdo con esta especificación pueden interactuar entre sí, independientemente del lenguaje en el que fueron escritos.

Microsoft proporciona, junto con el .NET Framework, las implementaciones de cuatro lenguajes compatibles con CLS, y sus compiladores:

- Microsoft Visual Basic .NET
- Microsoft Visual C# .NET
- Microsoft Visual J# .NET
- Microsoft Visual C++ .NET

Esto significa que una aplicación escrita en Visual Basic .NET, por ejemplo, puede incorporar

Tabla 1 | Principales servicios proporcionados por el CLR

SERVICIOS	DESCRIPCIÓN
Compilación Just In Time (o "justo a tiempo")	El CLR se encarga de compilar las aplicaciones .NET a código de máquina nativo para el sistema operativo y la plataforma de hardware en la que se está ejecutando.
Gestión automática de memoria	El CLR abstrae a los desarrolladores de tener que pedir y liberar memoria explícitamente. Para hacerlo, uno de sus componentes, llamado garbage collector (recolector de basura), se ocupa de liberar periódicamente la memoria que ya no está siendo usada por ninguna aplicación. Además, el CLR también abstrae a los desarrolladores del uso de punteros y del acceso a memoria de bajo nivel. Si bien estas características pueden ser consideradas poderosas, suelen hacer que el desarrollo y el mantenimiento de aplicaciones resulten más propensos a errores y menos productivos.
Gestión de errores consistente	Como las aplicaciones .NET no se ejecutan directamente contra el sistema operativo, cualquier error no manejado que ocurra en tiempo de ejecución será atrapado por el CLR en última instancia, sin afectar a ninguna otra aplicación que se esté ejecutando ni tener efecto alguno sobre su estabilidad.
Ejecución basada en componentes	Todas las aplicaciones .NET son empaquetadas en componentes reutilizables denominados, genéricamente, assemblies, que el CLR se ocupa de cargar en memoria y ejecutar.
Gestión de seguridad	El CLR proporciona una barrera más de contención a la hora de ejecutar aplicaciones manejadas, ya que permite establecer políticas de seguridad muy detalladas, que deberán ser cumplidas por las aplicaciones .NET que se ejecuten en una determinada computadora.
Multithreading	El CLR brinda un entorno de ejecución multi-hilos por sobre las capacidades del sistema operativo, así como también, mecanismos para asegurar su sincronización y acceso concurrente a recursos compartidos.



sin problemas nuevas partes escritas en C# o C++.NET.

Dado que la especificación CLS es un estándar público, ha permitido que otros diseñadores de lenguajes y compiladores desarrollen más de 20 lenguajes compatibles con ella y, por lo tanto, compatibles entre sí y con los lenguajes desarrollados por Microsoft. Todos los componentes y las aplicaciones creados bajo estas especificaciones se dice que son de código administrado, o *Managed Code*. La única excepción es C++.NET, que, además, tiene la capacidad de crear código no manejado, o *Unmanaged Code*, debido a que muchas aplicaciones de muy bajo nivel, como drivers de dispositivos, necesitan tener acceso directo a los recursos del sistema operativo para tener un mejor rendimiento.

Como cada lenguaje proporciona su propia sintaxis en la implementación de las directivas del CLS, es necesario que los compiladores “traduzcan” su código fuente en “algo” que sea entendible por el CLR. Ese “algo” se denomina MSIL o *Microsoft Intermediate Language*, que es un código intermedio en el que se compilan todos los lenguajes de .NET. Estos componentes y aplicaciones resultantes de la compilación se conocen como *assemblies* o ensamblados en .NET. Pueden ser archivos ejecutables (.exe) o bibliotecas de clases y componentes (.dll). Los assemblies no son ejecutables directamente, sino que son compilados al **código nativo** de la CPU en la que se encuentran corriendo por el compilador *Just In Time (JIT)* del CLR y, luego, se ejecutan en su entorno. Para que el CLR

Esquema de ejecución del CLR

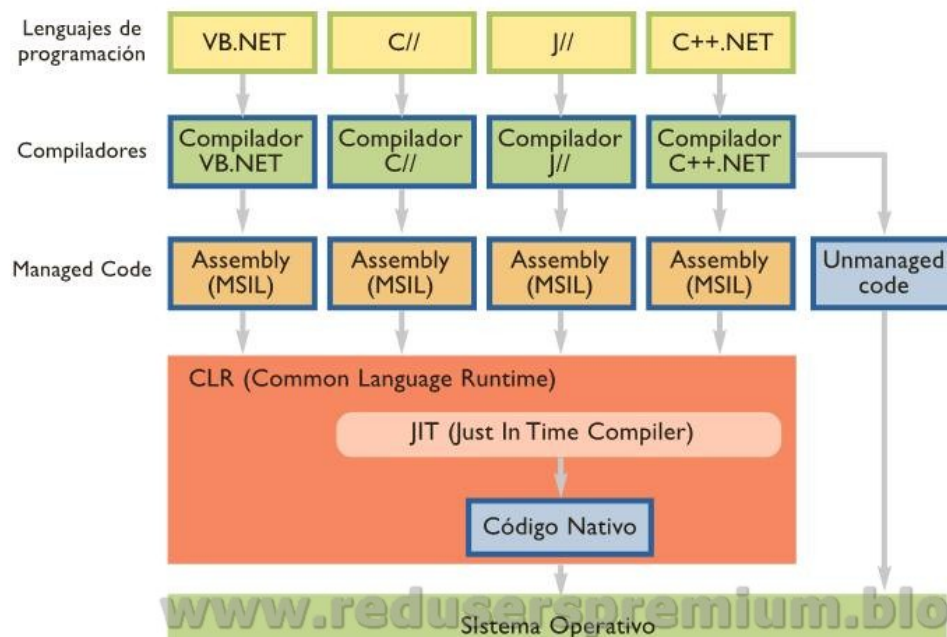


FIGURA 002 | Ejecución de las aplicaciones por medio del CLR. Los lenguajes de programación generan código administrado en MSIL a través de sus compiladores. Los assemblies generados son tomados por el JIT del CLR, traducidos a código nativo de la CPU en la que se encuentra y ejecutados en un ambiente aislado dentro del CLR.

El CLR administra la ejecución de código y permite el desarrollo de las aplicaciones.

entienda cómo debe ejecutar estas aplicaciones y componentes, los compiladores de los lenguajes de .NET incluyen en la compilación la **metadata**, que es la información que describe los objetos que forman parte de la aplicación o componente generado.

La metadata describe:

- Los tipos de datos y sus dependencias.
- Los objetos y sus miembros.
- La referencia e información (incluyendo versión) de los componentes y recursos externos que son utilizados por la aplicación o componente y que son necesarios para su funcionamiento.

Esta metadata de un componente administrado es utilizada por el CLR para proporcionar, entre otras, las siguientes funcionalidades:

- Administración de memoria.
- Localizar y crear instancias de clases.
- Administrar las referencias de los objetos y realizar el *garbage collection*.
- Resolver las invocaciones de métodos.
- Generar código nativo.
- Asegurar que la aplicación tiene los recursos necesarios para funcionar.
- Reforzar la seguridad.

Namespaces

El conocimiento de los namespaces de .NET es de fundamental importancia para aprovechar toda la funcionalidad que nos proporciona y, así, no tener que escribir código con el que ya contamos.

La inclusión de esta metadata en el componente compilado hace que éste se **autodescriba**. Esto le indica al CLR todo lo necesario para preparar y ejecutar una aplicación .NET correctamente y permitir que pueda interactuar con otros componentes.

- **Compilación Just In Time (o “justo a tiempo”)**: El CLR se encarga de compilar las aplicaciones .NET a código de máquina nativo para el sistema operativo y la plataforma de hardware en la que se está ejecutando. Esto lo hace sin intervención alguna del desarrollador o el usuario y cuando se necesita.
- **Gestión automática de memoria**: El CLR abstrae a los desarrolladores de tener que pedir y liberar memoria explícitamente. Para lograrlo, uno de sus componentes, llamado *garbage collector* (recolector de basura), se encarga de liberar periódicamente la memoria que ya no está siendo usada por ninguna aplicación. Por otra parte, el CLR también abstrae a los desarrolladores del uso de punteros y del acceso a memoria de bajo nivel. Si bien estas características pueden ser consideradas poderosas, suelen hacer que el desarrollo y el mantenimiento de aplicaciones resulten más propensos a errores y menos productivos.
- **Gestión de errores consistente**: Como las aplicaciones .NET no se ejecutan directamente contra el sistema operativo, cualquier error no manejado que ocurra en tiempo de ejecución será atrapado por el CLR en última instancia, sin afectar a ninguna otra aplicación que se esté ejecutando ni tener efecto alguno sobre su estabilidad.
- **Ejecución basada en componentes**: Todas las aplicaciones .NET son empaquetadas en componentes reutilizables denominados, genéricamente, *assemblies*, que el CLR se ocupa de cargar en memoria y de ejecutar. Profundizaremos en este tema más adelante en el curso.



- **Gestión de seguridad:** El CLR brinda una barrera más de contención a la hora de ejecutar aplicaciones manejadas, ya que permite establecer políticas de seguridad que deberán ser cumplidas por las aplicaciones .NET que se ejecuten en una computadora.
- **Multithreading:** El CLR provee un entorno de ejecución multi-hilos por sobre las capacidades del sistema operativo, así como también, mecanismos para asegurar su sincronización y el acceso a recursos compartidos.

Biblioteca de funcionalidad (Base Class Library)

La funcionalidad principal de la *Base Class Library* es proporcionar cientos de tipos básicos

(clases e interfaces) orientados a objetos, extensibles mediante herencia e independientes del lenguaje de programación que se desee utilizar (en el libro *Introducción a la programación*, se describe el significado de muchos de estos conceptos, que luego profundizaremos con mayor detalle). Este conjunto de elementos incluye la mayoría de las funcionalidades que los programadores aplican de manera cotidiana.

Dada la cantidad de clases (unos cuantos miles), es necesario organizarlas de algún modo en que sean fáciles de encontrar y que permita diferenciarlas si poseen el mismo nombre. Para lograr esto, .NET proporciona lo que se denomina namespaces o espacios de nombres. Se trata de calificadores de clases, que permiten determinar, unívocamente, qué clases utilizar dentro de .NET y, así, evitar ambigüedades. En la Tabla 2 vemos una breve lista de los principales namespaces.

Namespaces de la librería de clases base

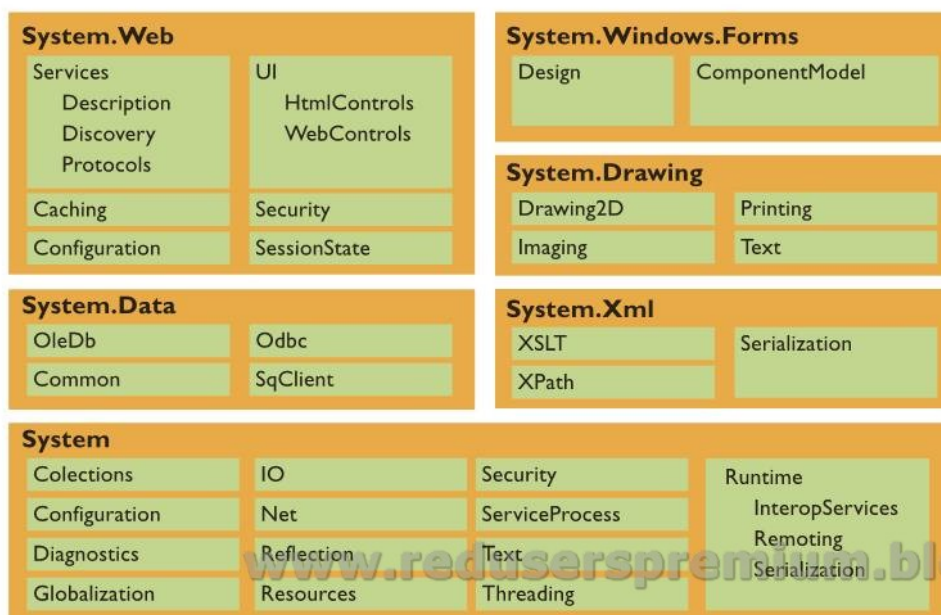


FIGURA 003 | Principales namespaces de la Base Class Library. Conocerlos nos permitirá aprovechar todo el potencial de .NET.

Lenguajes de programación

Los lenguajes de programación de .NET están basados en la *Common Language Specification* (CLS), por lo cual, ahora, la elección del lenguaje en el que debe escribirse una aplicación .NET prácticamente fue reducida a una cuestión de gustos personales y comodidad con la sintaxis. No hay motivos tecnológicos que nos lleven a escoger un lenguaje en particular, al menos, entre los ofrecidos por Mi-

crosoft. Todos utilizan el mismo runtime, todos emplean el mismo conjunto de bibliotecas de la misma manera, no existen diferencias de performance, todos tienen la misma potencia, y la misma capacidad de acceso a los recursos y servicios que expone el .NET Framework. De hecho, al cargar y ejecutar un assembly, el CLR no sabe en qué lenguaje de programación de alto nivel éste fue escrito, ya que lo que recibe finalmente es código MSIL.

Tabla 2 | Principales namespaces y su funcionalidad

NAMESPACES	FUNCIONALIDAD
System	Es el principal namespace y forma la raíz de todos los otros dentro del BCL. Entre los principales elementos que encontramos en él están: <ul style="list-style-type: none"> - Definición de todos los tipos de datos establecidos por el Common Type System del CLR. - Acceso a funciones matemáticas bajo el namespace MATH. - Clases dedicadas al acceso al entorno de ejecución de la aplicación a través del namespace Environment. - Acceso directo al Garbage Collector a través del namespace GC.
System.Collections	Contiene una serie de clases que permiten administrar un conjunto de objetos de manera ordenada. La principal es Collection, pero existen otras, como SortedList, ArrayList, Queue, Stack, etc.
System.Data	Contiene todas las clases necesarias para el procesamiento de datos desde bases de datos (ADO.NET). Entre los principales namespaces, podemos encontrar System.Data.SqlClient (exclusivo para bases de datos SQLServer), System.Data.OleDb (orígenes de datos OLEDB) y System.Data.Odbc (para otros tipos de bases de datos), entre otros.
System.Drawing	Contiene las clases para proporcionar toda la funcionalidad gráfica en el framework.
System.IO	Contiene las clases y los métodos necesarios para leer y escribir todo tipo de archivos, tanto en texto plano como a nivel de bytes.
System.Security	Proporciona toda la funcionalidad para realizar los procesos de autenticación y autorización, manejo de credenciales y criptografía, entre otras tareas.
System.Text	Contiene las clases para codificar y decodificar texto de diferentes formatos. También incluye las clases necesarias para la búsqueda y el manejo de texto a través de expresiones regulares.
System.Windows.Form	Es el namespace base para la creación de aplicaciones Windows con formularios. En él están definidas todas las funcionalidades y los controles necesarios para la creación de interfaces de usuario en este tipo de aplicaciones: cajas de texto, botones, etiquetas, diálogos y mucho más.
System.Web	Es el namespace base para la creación de aplicaciones Web (ASP.NET). En él están definidas todas las funcionalidades y los controles necesarios para la creación de interfaces de usuario en este tipo de aplicaciones. También incluye las definiciones para el tratamiento de servicios web y muchos aspectos más de este entorno.
System.Xml	Encapsula todas las clases necesarias para el procesamiento de documentos XML.



Principales lenguajes de la plataforma

Existe una gran variedad de lenguajes de programación en la plataforma .NET, pero se destacan, principalmente, dos: VB.NET y C# (C Sharp).

- VB.NET es Visual Basic .NET y es la evolución de Visual Basic 6.0. Fue rescrito por completo para la plataforma para hacerlo totalmente orientado a objetos, y puede hacer uso de todos los elementos del framework, como cualquier otro lenguaje. Para quienes hayan programado en Visual Basic 6.0, éste es el lenguaje natural para utilizar en .NET, ya que presenta la misma sintaxis que su predecesor, aunque con algunos cambios que permiten adaptarse a la plataforma.
- C# (pronunciado C Sharp) es un nuevo lenguaje que fue diseñado, específicamente, para la plataforma .NET. Tiene una sintaxis similar a las de C y Java, por lo que resulta la elección natural para quienes hayan trabajado con ellos.

Ambos lenguajes son de primer nivel en la plataforma y no presentan diferencias de rendimiento en ella.

Herramientas de desarrollo

Para que la creación de aplicaciones en .NET sea lo más productiva posible, Microsoft ofrece su propio entorno de desarrollo: Visual Studio. Se trata de un IDE (entorno integrado de desarrollo) preparado para aprovechar todas las características del framework .NET y darle al programador un entorno único desde donde realizar todas sus tareas. Dentro de Visual Studio, podemos realizar aplicaciones Windows y Web, servicios Web y bibliotecas de componentes, además de acceder a bases de

datos, y muchas opciones más, con cualquier lenguaje de la plataforma provisto por Microsoft (VB.NET, C#, C++, etc.). La idea de este IDE es que el programador no tenga que cambiar de aplicación para realizar todas las tareas involucradas en el desarrollo de un sistema. La familia de Visual Studio 2005 tiene un producto a la medida de las necesidades y posibilidades de cada tipo de desarrollador, partiendo desde una línea gratuita de versiones denominadas Express, hasta llegar a una suite completa de productos destinada a grandes equipos de desarrollo: Visual Studio Team System. A continuación, una breve descripción de las diferentes “familias” de productos:

VISUAL STUDIO 2005 EXPRESS EDITION

Incluye herramientas livianas, fáciles de usar y de aprender, destinadas a novatos, estudiantes y quienes programan como hobby. Son las versiones ideales para los que recién se inician en .NET. Son gratuitas y hay una herramienta por cada lenguaje.

Para desarrollar aplicaciones Windows, existen las siguientes alternativas:

- Visual Basic 2005 Express Edition
- Visual C# 2005 Express Edition
- Visual J# 2005 Express Edition
- Visual C++ 2005 Express Edition
- SQL Server 2005 Express Edition
- Visual Web Developer 2005 Express

⊛ ¿Qué es un IDE?

Un IDE (*Integrated Development Environment* o entorno integrado de desarrollo) es un espacio de trabajo que proporciona un conjunto de menús, toolbars y ventanas para realizar las diferentes tareas de programación.

VISUAL STUDIO 2005 STANDARD EDITION

Es un entorno de desarrollo unificado, pensado para desarrolladores que construyen aplicaciones cliente/servidor de Windows y sitios Web. En esta edición, todo el IDE se encuentra mejorado e integrado, de modo que se pueden realizar, de manera conjunta, aplicaciones Windows, Web y para dispositivos móviles, en diferentes lenguajes de programación y sin cambiar de entorno.

VISUAL STUDIO 2005 PROFESSIONAL EDITION

Es un entorno de desarrollo pensado para programadores individuales que construyen aplicaciones de alto rendimiento. Es posible aprovechar su entorno para construir una amplia gama de aplicaciones móviles, Web, Windows y basadas en Office. Presenta mejoras en el IDE, además de que proporciona soporte de depuración remoto, posibilidad de generar aplicaciones de 64 bits, y acceso completo a los servicios del sistema y a las bases de datos. Incluye Crystal Reports para la generación de reportes y proyectos de instalación basados en Windows Installer.

VISUAL STUDIO 2005 TOOLS FOR OFFICE

Esta versión permite que profesionales IT, ISVs e Integradores de Sistemas construyan soluciones Smart Client para Microsoft Office

VISUAL STUDIO 2005 TEAM SYSTEM

Se compone de herramientas para administrar todo el ciclo de vida del desarrollo de software, de manera que sean productivas, integradas y extensibles. Amplía la línea de productos de Visual Studio para que los equipos de trabajo mejoren sus capacidades de comunicación y colaboración. La familia Team System se divide en:

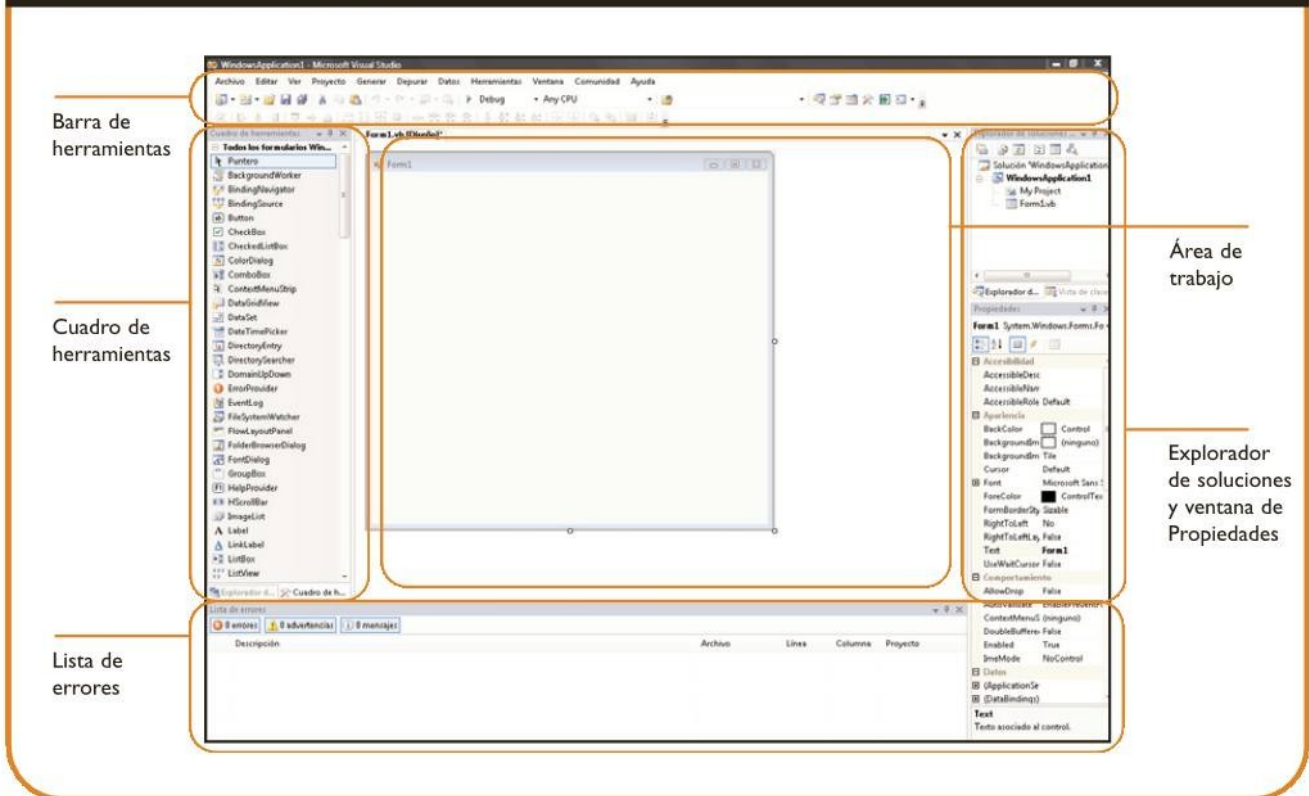
- **Visual Studio Team Suite:** Es la suite más completa, útil tanto para arquitectos como para desarrolladores y testers.
- **Visual Studio 2005 Team Suite Edición profesionales de bases de datos:** Es una edición especial que proporciona herramientas para permitir el cambio de la administración, el testeado y la implementación para las bases de datos del servidor SQL.
- **Visual Studio 2005 Team Suite Edición para arquitectos:** Brinda herramientas para el diseño visual de servicios, soluciones orientadas a servicios y validación con ambientes operacionales antes de su implementación.
- **Visual Studio 2005 Team Suite Edición para testeadores de software:** Introduce un conjunto de herramientas de testeado que se encuentran integradas dentro del ambiente del Visual Studio y ayudan a construir aplicaciones de alta calidad.
- **Visual Studio 2005 Team Foundation Server:** Es un servidor integrado que combina control de versiones, seguimiento de ítem de trabajo y reportes.

Tabla 3 | Requisitos de las versiones

Versión de Visual Studio	Procesador	RAM	Disco Rígido
Express	Pentium III 600 MHz o similar	256 MB	700 MB
Standard	Pentium III 600 MHz o similar	256 MB	Entre 2 GB y 3,8 GB
Professional	Pentium III 600 MHz o similar	256 MB	Entre 2 GB y 3,8 GB
Team System	Procesador de 2 GB o más	512 MB	8 GB



▶ Guía Visual 001 | Secciones del entorno de desarrollo



- **Visual Studio 2005 Team Test Load Agent:** Destinada a testadores de software. Permite que las organizaciones simulen más usuarios y tests más precisos de desempeño en las aplicaciones desarrolladas.

El IDE

El IDE proporciona herramientas tales como los compiladores, un depurador o debugger (que asiste en el seguimiento y la corrección de errores en los programas) y otras destinadas a administrar los proyectos. Proporciona un menú estándar, numerosas toolbars y un gran número de ventanas. El IDE está compuesto, básicamente, por cinco secciones:

- **Barra de herramientas:** Aquí se encuentran las barras de herramientas, que dan acceso a los comandos más comunes. De

acuerdo con la tarea que estemos realizando, esta sección cambiará según sean necesarios.

- **Lista de errores:** Aquí están las ventanas de errores, la de tareas y la de salida.
- **Cuadro de herramientas:** Presenta la caja de herramientas, que contiene todos los controles

☺ Paciencia

Aunque muchos de los conceptos vistos hasta el momento pueden resultarnos de difícil comprensión e incluso algo aburridos, debemos saber que es muy importante tener una base conceptual sólida antes de comenzar a "tipear código". Muchos de estos conceptos serán comprendidos con mayor claridad cuando empecemos a trabajar.

y componentes que podemos utilizar cuando realizamos aplicaciones. Según el tipo de programa (Windows o Web), se llenará con los controles y componentes apropiados para cada uno. También está el DataBase Explorer o explorador de bases de datos, que nos da acceso a los orígenes de datos por utilizar en nuestra aplicación.

- **Explorador de soluciones y ventana de propiedades:** Ambos son importantes, ya que el primero presenta todos los proyectos y archivos con que estamos trabajando; y el segundo, las propiedades de todos los elementos del entorno.
- **Área de trabajo:** En este espacio aparecen todos los documentos que estemos manejando, tanto en vista de código como en vista de diseño. El IDE es un entorno de múltiples documentos, y se accede a cada uno de ellos a través de las solapas con su nombre.

Debido a la gran cantidad de elementos que contiene el IDE, se le han agregado diferentes elementos para aprovechar al máximo el espacio disponible. Uno de ellos es el Auto-Dock, que permite mover las ventanas por todo el IDE utilizando el mouse para repositionarlas. Si por alguna razón cerramos alguna ventana, basta con ir al menú **Ver** para abrirla otra vez.

Menús y barras de herramientas

El IDE presenta un conjunto de menús que debemos acostumbrarnos a utilizar para sacarle el máximo provecho. Muchos de ellos siguen el estándar de cualquier aplicación Windows, y otros cuentan con opciones y funcionamiento similares a los de cualquier aplicación. En la Guía Visual 002 se describen los principales menús y sus funcionalidades.

▶ Guía Visual 002 | Los menús del entorno

The image shows the menu bar of Microsoft Visual Studio with the following items: Archivo, Editar, Ver, Proyecto, Generar, Depurar, Datos, Herramientas, Ventana, Comunidad, Ayuda. Below the menu bar, there are two rows of text boxes connected to the menu items by vertical lines, explaining their functions.

Proporciona las opciones para la apertura y grabación de proyectos y soluciones.	Permite acceder a las ventanas del IDE.	Destinado a la compilación del proyecto.	Configura las conexiones a los orígenes de datos.	Permite la selección y organización de las ventanas dentro del IDE.	Permita acceder a la ayuda integrada del sistema.
Brinda los comandos para la edición de texto, tales como Cut, Copy y Paste.	Permite agregar diferentes elementos al proyecto.	Permite el seguimiento del programa para la determinación de errores.	Contiene un conjunto de opciones para la configuración del entorno y de los proyectos.	Otorga acceso a los recursos en línea sobre Visual Basic.	



Visual Studio Express

Veamos y analicemos cuáles son las ventajas, requisitos y posibilidades que nos brinda esta versión de Visual Studio.

Como mencionamos anteriormente, existen diversas versiones de Visual Studio según la necesidad de cada corporación o equipo de trabajo. El problema es que cada una es muy costosa, y es difícil que dispongamos de algunos miles de dólares para adquirir una y comenzar a estudiar. Por eso, la empresa Microsoft puso a disposición una versión de su entorno Visual Studio totalmente gratuita, llamada Express. La pregunta es por qué ofrece sin costo un entorno que cuesta miles de dólares. La respuesta es sencilla: esta versión Express no contiene todo el potencial de sus hermanas mayores, pero nos permite aprender usando sus mismas posibilidades. Antes de continuar, es importante que conozcamos cuál es la diferencia que existe entre esta versión y la versión comercial. La versión Express está dividida por lenguajes, es decir que si queremos utilizar Visual Basic .NET, deberemos instalar dicha versión de Visual Studio Express; en el caso de C#, deberemos hacer lo mismo, y así podremos completar todos los lenguajes disponibles. Y es acá dónde tenemos la principal diferencia. Las versiones superiores a las Express utilizan el mismo entorno para todos los lenguajes, y permiten instalar una sola vez el producto, para disponer de todos ellos. Incluso, es posible realizar aplicaciones empleando más de un lenguaje de programación al mismo tiempo (siempre que hayamos instalado antes todos los entornos). Como nosotros estamos comenzando a programar y no vamos a realizar aplicaciones multilingüaje, esta última característica realmente no nos beneficia.

Cómo obtener las versiones Express

Si bien estas versiones están disponibles en forma gratuita en Internet, no vamos a asumir que todos los usuarios cuentan con banda ancha para realizar la descarga. Por eso, en los CDs que acompañan a la obra incluimos las versiones de Visual Basic .NET Express, C# Express y SQL Server 2005 Express, lo que nos permitirá seguir sin inconvenientes el desarrollo de nuestra capacitación. A lo largo de la obra, iremos indicando el lenguaje utilizado en el código fuente, para que cada uno recurra a la versión Express adecuada y pueda realizar con éxito todas las prácticas.

Como podemos apreciar, las versiones Express tienen un gran potencial, ya que permiten efectuar desarrollos de nivel profesional. Y el día de mañana, cuando estemos delante de una versión profesional, veremos que todo está exactamente en el mismo lugar y no tendremos que perder tiempo en aprender a manejarla nuevamente.

⚠ Requisitos mínimos para las versiones Express

Procesador: Pentium III, 600 MHz

RAM: 256 MB

Disco duro: 700 MB aproximadamente para cada lenguaje

Sistema operativo: Windows XP, 2000 o superior

PRACTICA 01

Nuestro primer programa: un navegador Web

Veremos la potencia de .NET para realizar una aplicación que requiere que escribamos ¡una línea de código!

Llegó el momento de realizar nuestra primera aplicación. La idea es programar un navegador sencillo que nos permita navegar por Internet. Para realizarlo, utilizaremos como lenguaje Visual Basic .NET, por lo que deberemos tener instalado Visual Studio o Visual Basic Express. Es importante aclarar que las indicaciones dadas servirán para todas las versiones de Visual Studio. Lo primero que haremos será abrir Visual Studio.

1 – Una vez abierto el IDE, seleccionamos del menú **Archivo** la opción **Nuevo proyecto**. Si utilizamos una versión que no es Visual Basic .NET Express, cuando se abra la aplicación, deberemos seleccionar Visual Basic en **Tipos de proyecto** (panel izquierdo de la ventana **Nuevo proyecto**).

2 – En la ventana **Nuevo proyecto**, elegimos **Aplicación para Windows** (Figura 004); en la

parte inferior aparecerá la posibilidad de asignarle un nombre a nuestro trabajo. En este caso, escribimos **MiniBrowser** en el cuadro de texto. Podemos poner cualquier nombre, pero siempre tengamos en cuenta que esté relacionado con el objetivo de la aplicación final.

3 – Una vez creado el proyecto, veremos el entorno completo de Visual Studio. El panel que vamos a utilizar es el de **Propiedades**, para modificar las funciones que realizará la ventana de nuestra aplicación. Mientras estemos desarrollando la aplicación, a estas ventanas las llamaremos **Formularios**. En ellos pondremos los componentes o elementos que serán parte de nuestra aplicación, tal como si fuera un editor de imágenes. Pero antes de colocar elementos, debemos hacer clic en el formulario (**Form1**) y, en el panel de propiedades, buscar los siguientes datos, para darles el valor indicado:

- **Text:** MiniBrowser. El texto escrito en esta propiedad será el que mostrará la ventana en la parte superior, en la misma línea de los botones de minimizar, maximizar y cerrar.
- **Size:** 640,480. Para poner estos valores, hacemos clic en el signo + que tiene esta propiedad, y escribimos el ancho y el alto. Esta propiedad indica que, al abrirse el proyecto, lo hará con un tamaño de ventana de 640 por 480 píxeles.
- **StartPosition:** CenterScreen. Esta propiedad hace que la ventana o aplicación aparezca centrada en la pantalla.

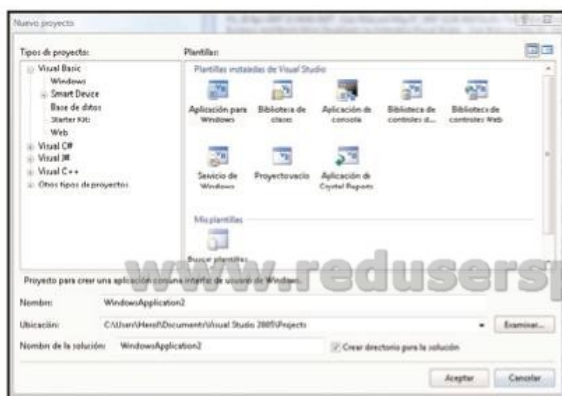


FIGURA 004 | Creación de un nuevo proyecto: aplicación para Windows.



- `FormBorderStyle: FixedSingle`. Con esta propiedad, evitamos que, al ir con el mouse sobre los extremos de la ventana, se modifique su tamaño, con lo cual no podremos maximizar la aplicación.

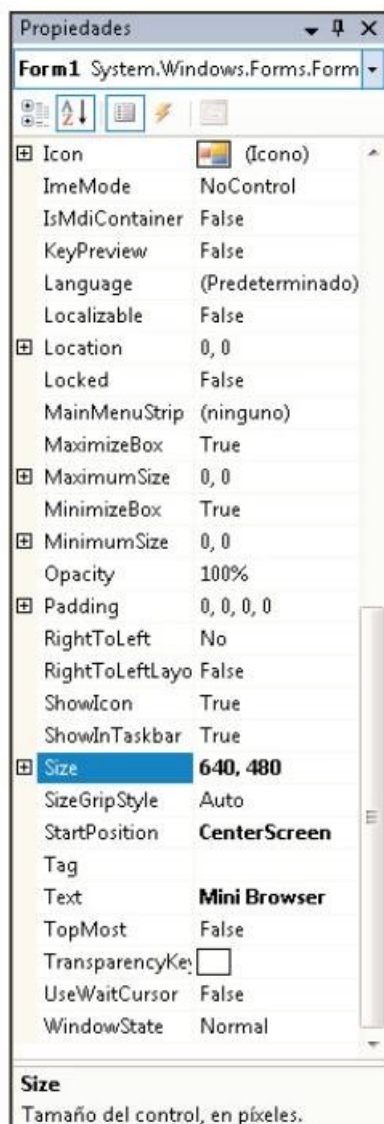


FIGURA 005 | Propiedades del formulario (Form1).

Por ahora, ingresamos esta información sin comprender demasiado qué es lo que estamos haciendo. No nos preocupemos, ya que a lo largo de esta obra, profundizaremos en cada uno de estos conceptos.

4 – Llegó el momento de empezar a colocar los objetos. Desde el **Cuadro de herramientas**, buscamos el objeto llamado **TextBox** y, haciendo un clic sin soltar el botón del mouse, lo arrastramos al formulario. Al soltarlo, veremos que el objeto dibujado aparece en él. Este objeto nos permite ingresar texto cuando la aplicación esté en ejecución. En nuestro caso, allí colocaremos la dirección del sitio que queremos visitar. Para que quede prolijo, lo ubicamos en la parte superior.

5 – En la misma barra de donde obtuvimos el **TextBox**, buscamos el objeto **Button** y lo colocamos de la misma manera en el formulario, tratando de que quede al lado del cuadro anterior.

6 – Hacemos clic sobre el **TextBox** que está en el Formulario y veremos que aparecen unos puntitos alrededor, que nos permiten variar su tamaño. Lo extendemos para que ocupe casi todo el ancho de la pantalla. A continuación, modificamos algunas propiedades del **TextBox**, para lo cual, primero, lo seleccionamos en la ventana de propiedades. Entonces, colocamos lo siguiente:

- **Name:** `txtURL`. El nombre de esta propiedad nos permitirá, más adelante, programar acciones para este objeto desde cualquier lenguaje. Se utilizan las tres primeras letras para indicar el tipo de objeto, y el resto, para el nombre. Por ejemplo, `txt` indica que es un **TextBox**, y `URL`, el nombre de ese **TextBox**. Cada vez que hagamos referencia en el código a este objeto, pondremos `txtURL`.

Ahora seleccionemos el objeto **Button** y, en la ventana de propiedades, cambiamos:

- **Name:** `btnNavegar`. Cumple la misma función que la propiedad **Name** en el **TextBox**.

- Text: Navegar. Esta propiedad permite poner un texto que se muestra sobre el botón, como Sí, Cancelar, Aceptar, y otros.



FIGURA 006 | Disposición del cuadro de texto y el botón para nuestro MiniBrowser.

7 – Vamos al **Cuadro de herramientas** y arrastramos un control **WebBrowser** sobre el formulario. Si al soltarlo ocupa toda su superficie y oculta el resto de los controles, seleccionamos la pequeña flecha que aparece en la parte superior derecha del control y marcamos la opción **Acoplar en contenedor principal**. A continuación, cambiamos su tamaño para que quede de acuerdo con las dimensiones del formulario. Finalmente, modificamos la propiedad **Name** a **WebBrowserCtl** (la propiedad Name cumple la misma función en todos los objetos; ver propiedad Name en TextBox).



FIGURA 007 | Disposición final de los controles de nuestro minibrowser.

8 – Ya tenemos todos los elementos dispuestos en el formulario y estamos a un paso de hacer funcionar el navegador. Al ejecutarlo, el usuario deberá escribir una dirección de Internet en el TextBox y, al hacer clic sobre el botón Navegar, la aplicación mostrará la página Web en el control WebBrowser. Entonces, necesitaremos escribir el código correspondiente para cuando se realice la acción clic. La programación se lleva

a cabo haciendo dos clics en el botón Navegar que está en el formulario. Veremos que Visual Studio muestra una nueva ventana con código ya armado, correspondiente al evento clic del botón. Entre las líneas Private Sub y End Sub escribimos el siguiente código:

```
WebBrowserCtl1.Navigate(txtURL.Text)
```

El código nos dice que, cuando se presiona el botón Navegar, el texto escrito (dirección Web) en el objeto txtURL será utilizado para navegar por Internet en el objeto WebBrowserCtl.

9 – Llega el momento esperado. Para probar la aplicación, presionamos la tecla <F5>. Cuando el formulario aparece como una ventana de aplicación en el centro de la pantalla, escribimos una dirección de Internet en el cuadro de texto y presionamos el botón Navegar. Ya tenemos nuestro navegador funcionando. Obviamente, para que lo haga correctamente, debemos tener una conexión activa a Internet.



FIGURA 008 | Minibrowser en funcionamiento.

Ahora que sabemos que nuestra aplicación funciona, probemos variar las propiedades Size, StartPosition y FormBorderStyle del formulario y ver las diferencias que se producen. De esta manera, nos iremos familiarizando con algunos objetos que usaremos con frecuencia.

USERS



CURSOS.REDUSERS.COM

CURSOS INTENSIVOS



Los temas más importantes del universo de la tecnología desarrollados con la mayor profundidad y con un despliegue visual de alto impacto: Explicaciones teóricas, procedimientos paso a paso, videotutoriales, infografías y muchos recursos mas.

Brinda las habilidades necesarias para planificar, instalar y administrar redes de computadoras de forma profesional. Basada principalmente en tecnologías Cisco, es una obra actual, que busca cubrir la necesidad creciente de formar profesionales.

- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 3 CDs / 1 Libro



- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 4 CDs

Curso para dominar las principales herramientas del paquete Adobe CS3 y conocer los mejores secretos para diseñar de manera profesional. Ideal para quienes se desempeñan en diseño, publicidad, productos gráficos o sitios web.

Obra teórica y práctica que brinda las habilidades necesarias para convertirse en un profesional en composición, animación y VFX (efectos especiales).

- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 2 CDs / 1 DVD / 1 Libro



- ▶ 26 Fascículos
- ▶ 600 Páginas
- ▶ 2 DVDs / 2 Libros

Obra ideal para ingresar en el apasionante universo del diseño web y utilizar Internet para una profesión rentable. Elaborada por los máximos referentes en el área, con infografías y explicaciones muy didácticas.

www.reduserspremium.blogspot.com.ar

Llegamos a todo el mundo con OCA * y DHL **

✉ usershop@redusers.com ☎ +54 (011) 4110-8700

usershop.redusers.com.ar

** Válido en todo el mundo excepto Argentina. * Sólo válido para la Republica Argentina

Argentina: \$5,90 (recargo al interior \$0,20) México: \$30

USERS

Microsoft®

Curso teórico y práctico de programación

Desarrollador .net

Con toda la potencia de **Visual Basic .NET** y **C#**

25 fascículos | 600 páginas | 1 Libro
3 CDs con las versiones express de Visual Basic .NET, C# y SQL Server

1

La mejor forma de aprender a programar desde cero



Basado en el programa Desarrollador Cinco Estrellas de Microsoft



9 789871 347438