

Microsoft®

C#™

Lenguaje y aplicaciones

2ª EDICIÓN

Entornos de desarrollo:

Visual Studio 2005

•

Visual C# 2005 Express

•

Visual Web Developer
2005 Express

•

SQL 2005 Express



Elementos del lenguaje

•

Programación
orientada a objetos

•

Interfaces gráficas

•

Acceso a bases de
datos
(ADO.NET)

•

Aplicaciones para
Internet
(ASP.NET)

•

Plataforma Mono



Fco. Javier Ceballos

Incluye CD-ROM con
Microsoft .NET
Framework SDK, EDI y
los ejemplos del libro



Ra-Ma®

Microsoft C#TM

Lenguaje y aplicaciones

2.^a edición

Fco. Javier Ceballos Sierra

Profesor titular de la
Escuela Politécnica Superior
Universidad de Alcalá

<http://www.fjceballos.es>





Microsoft C#. Lenguaje y aplicaciones. 2ª edición.

© Fco. Javier Ceballos Sierra

© De la edición: RA-MA 2007

MARCAS COMERCIALES: Las marcas de los productos citados en el contenido de este libro (sean o no marcas registradas) pertenecen a sus respectivos propietarios. RA-MA no está asociada a ningún producto o fabricante mencionado en la obra, los datos y los ejemplos utilizados son ficticios salvo que se indique lo contrario.

RA-MA es una marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso, ni tampoco por cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa ni de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente.

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro, sin autorización previa y por escrito de RA-MA; su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes intencionadamente, reprodujeren o plagiaran, en todo o en parte, una obra literaria, artística o científica.

Editado por:

RA-MA Editorial

C/ Jarama, 3A, Polígono industrial Igarsa

28860 PARACUELLOS DEL JARAMA, Madrid

Teléfono: 91 658 42 80

Telefax: 91 662 81 39

Correo electrónico: editorial@ra-ma.com

Internet: www.ra-ma.es y www.ra-ma.com

ISBN: 978-84-7897-813-7

Depósito Legal: M-32337-2007

Autoedición: Fco. Javier Ceballos

Filmación e impresión: Albadalejo, S.L.

Impreso en España

Primera impresión: septiembre 2007

CONTENIDO

PRÓLOGO.....	XVII
CAPÍTULO 1. FASES EN EL DESARROLLO DE UN PROGRAMA	1
QUÉ ES UN PROGRAMA	1
QUÉ ES C#	2
REALIZACIÓN DE UN PROGRAMA EN C#	3
Cómo crear un programa	4
Interfaz de línea de órdenes	4
¿Qué hace este programa?	5
Guardar el programa escrito en el disco	6
Compilar y ejecutar el programa	6
Biblioteca de clases	8
Guardar el programa ejecutable en el disco.....	9
Depurar un programa.....	9
Entorno de desarrollo integrado	10
EJERCICIOS RESUELTOS	10
EJERCICIOS PROPUESTOS	12
CAPÍTULO 2. FUNDAMENTOS DE C#.....	13
DECLARACIÓN DE UNA VARIABLE.....	13
ASIGNAR VALORES	16
AÑADIR COMENTARIOS	18
MOSTRAR DATOS POR LA PANTALLA.....	19
EXPRESIONES ARITMÉTICAS.....	20
EXPRESIONES CONDICIONALES	21
ESCRIBIR NUESTROS PROPIOS MÉTODOS	23

EJERCICIOS PROPUESTOS	25
CAPÍTULO 3. PROGRAMACIÓN ORIENTADA A OBJETOS.....	27
PENSAR EN OBJETOS	28
Clases y objetos	28
Mensajes y métodos	29
DISEÑO DE UNA CLASE DE OBJETOS.....	30
CONSTRUCTORES	37
HERENCIA.....	40
EJERCICIOS RESUELTOS	47
EJERCICIOS PROPUESTOS	53
CAPÍTULO 4. ELEMENTOS DEL LENGUAJE	55
CARACTERES DE C#	55
TIPOS.....	57
LITERALES.....	58
Literales enteros	58
Literales reales.....	59
Literales de un solo carácter	60
Literales de cadenas de caracteres	60
IDENTIFICADORES.....	61
PALABRAS CLAVE.....	61
DECLARACIÓN DE CONSTANTES SIMBÓLICAS	62
¿Por qué utilizar constantes?	63
VARIABLES.....	63
CONVERSIÓN ENTRE TIPOS PRIMITIVOS	65
OPERADORES.....	66
Operadores lógicos	66
Operadores unitarios.....	67
Operadores a nivel de bits	67
Operadores de asignación.....	68
Operador condicional	70
PRIORIDAD Y ORDEN DE EVALUACIÓN.....	71
EJERCICIOS RESUELTOS	72
EJERCICIOS PROPUESTOS	74
CAPÍTULO 5. ESTRUCTURA DE UN PROGRAMA	75
ESTRUCTURA DE UN PROGRAMA C#.....	75
Espacios de nombres	79
Protección de una clase	81

Directriz using	82
Definiciones y declaraciones	83
Sentencia simple	84
Sentencia compuesta o bloque	84
Métodos	84
Definición de un método	84
Método Main	85
Crear objetos de una clase	86
Cómo acceder a los miembros de un objeto	87
Protección de los miembros de una clase	88
Miembro de un objeto o de una clase	90
Referencias a objetos	92
Pasando argumentos a los métodos	95
EJERCICIOS RESUELTOS	96
EJERCICIOS PROPUESTOS	98
CAPÍTULO 6. CLASES DE USO COMÚN	99
ENTRADA Y SALIDA	99
Flujos de entrada	101
Flujos de salida	102
Salida con formato	104
Excepciones	106
Trabajar con tipos de datos numéricos	107
Estructuras que encapsulan los tipos primitivos	108
Clase Leer	111
¿DÓNDE SE UBICAN LAS CLASES QUE DAN SOPORTE?	114
CARÁCTER FIN DE FICHERO	114
CARACTERES <code>\r\n</code>	116
MÉTODOS MATEMÁTICOS	118
NÚMEROS ALEATORIOS	120
EJERCICIOS RESUELTOS	120
EJERCICIOS PROPUESTOS	124
CAPÍTULO 7. SENTENCIAS DE CONTROL	127
SENTENCIA if	127
ANIDAMIENTO DE SENTENCIAS if	130
ESTRUCTURA else if	132
SENTENCIA switch	135
SENTENCIA while	138
Bucles anidados	141
SENTENCIA do ... while	143

SENTENCIA for	146
SENTENCIA foreach.....	150
SENTENCIA break	150
SENTENCIA continue	151
SENTENCIAS try ... catch.....	151
EJERCICIOS RESUELTOS	153
EJERCICIOS PROPUESTOS	159

CAPÍTULO 8. MATRICES..... 161

INTRODUCCIÓN A LAS MATRICES	162
MATRICES NUMÉRICAS UNIDIMENSIONALES	163
Declarar una matriz	163
Crear una matriz	164
Iniciar una matriz.....	165
Acceder a los elementos de una matriz.....	165
Métodos de una matriz	166
Trabajar con matrices unidimensionales.....	167
CADENAS DE CARACTERES	170
Clase String	172
String(char[]).....	172
String ToString()	173
void CopyTo(int IndFuente, char[] destino, int IndDes, int nCars).....	173
static String Concat(String str1, String str2)	174
int CompareTo(String otroString)	174
int Length	175
String ToLower().....	175
String ToUpper()	175
String Trim()	176
bool StartsWith(String prefijo).....	176
bool EndsWith(String sufijo)	176
String Substring(int pos, int ncars).....	176
Operador de indexación.....	176
int IndexOf(String str).....	177
int IndexOfAny(char[] cad).....	177
String Replace(String str, String nuevaStr)	177
String Insert(int pos, String str)	177
String[] Split(char[] seps)	177
Clase StringBuilder	178
StringBuilder([arg]).....	178
int Length	178
int Capacity	179
StringBuilder Append(tipo x).....	179

StringBuilder Insert(int índice, tipo x)	179
StringBuilder Remove(int p, int n)	179
StringBuilder Replace(String str, String nuevaStr, int p, int n)	180
Operador de indexación	180
String ToString()	180
Trabajar con cadenas de caracteres	180
MATRICES MULTIDIMENSIONALES Y DE REFERENCIAS	183
Matrices numéricas	184
Matrices de cadenas de caracteres	190
Matrices de objetos String o StringBuilder	194
SENTENCIA foreach	196
COLECCIONES	197
EJERCICIOS RESUELTOS	198
EJERCICIOS PROPUESTOS	203
 CAPÍTULO 9. MÁS SOBRE MÉTODOS	 205
PASAR UNA MATRIZ COMO ARGUMENTO A UN MÉTODO	205
MÉTODO QUE RETORNA UNA MATRIZ	207
PASAR UN ARGUMENTO POR VALOR O POR REFERENCIA	209
PASAR ARGUMENTOS EN LA LÍNEA DE ÓRDENES	212
MÉTODOS RECURSIVOS	214
LA CLASE Array	217
Clear	217
Sort	217
BinarySearch	218
Reverse	219
EJERCICIOS RESUELTOS	219
EJERCICIOS PROPUESTOS	223
 CAPÍTULO 10. TRABAJAR CON FICHEROS	 227
VISIÓN GENERAL DE LOS FLUJOS DE E/S	229
ABRIENDO FICHEROS PARA ACCESO SECUENCIAL	229
Flujos de bytes	230
FileStream	230
Flujos de caracteres	235
StreamWriter	235
StreamReader	238
Clases File, Directory y Path	239
Flujos de datos de tipos primitivos	242
BinaryWriter	242
BinaryReader	244

Un ejemplo de acceso secuencial	245
Escribir en un fichero	246
Más sobre excepciones	248
Leer de un fichero	249
ABRIENDO FICHEROS PARA ACCESO ALEATORIO	251
Propiedades y métodos para acceso aleatorio.....	252
Position.....	252
Length	252
Seek	253
La clase CPersona.....	255
La clase CListaTfños.....	256
Constructor CListaTfños	257
Escribir un registro en el fichero	259
Añadir un registro al final del fichero	260
Leer un registro del fichero	260
Eliminar un registro del fichero.....	261
¿Hay registros marcados para eliminar?.....	262
Buscar un registro en el fichero.....	262
Un ejemplo de acceso aleatorio a un fichero	263
Modificar un registro.....	266
Actualizar el fichero	267
UTILIZACIÓN DE DISPOSITIVOS ESTÁNDAR	268
EJERCICIOS RESUELTOS	273
EJERCICIOS PROPUESTOS.....	278

CAPÍTULO 11. APLICACIONES CON INTERFAZ GRÁFICA 281

PROGRAMANDO EN WINDOWS	283
ESTRUCTURA DE UNA APLICACIÓN	285
Compilar y ejecutar la aplicación	289
DISEÑO DE LA INTERFAZ GRÁFICA	290
Crear un componente.....	290
Controles más comunes.....	290
Añadir los controles al formulario.....	291
Añadir una etiqueta y editar sus propiedades	291
Añadir un botón de pulsación y editar sus propiedades.....	294
Añadir una descripción abreviada a un componente	294
MANEJO DE EVENTOS	296
Asignar manejadores de eventos a un objeto.....	297
ENTORNOS DE DESARROLLO INTEGRADOS	297
Microsoft Visual C# Express.....	297
Crear un nuevo proyecto	299
El formulario	302

Dibujar los controles	302
Borrar un control	307
Propiedades de los objetos.....	307
Bloquear la posición de todos los controles	309
Icono de la aplicación.....	309
Escribir los controladores de eventos	310
Guardar la aplicación.....	312
Verificar la aplicación	312
AÑADIR OTROS CONTROLES	314
Añadir una caja de texto	315
Añadir un botón de pulsación.....	316
Añadir una descripción abreviada	316
Añadir una casilla de verificación	317
Añadir un botón de opción	318
Añadir una barra de desplazamiento.....	319
Añadir una lista	321
Añadir una barra de menús.....	322
Mostrar cajas de diálogo.....	325
UTILIZANDO VISUAL C# EXPRESS (VISUAL STUDIO).....	327
Dibujar los controles.....	328
Escribir los controladores de eventos	329
Compilar y ejecutar la aplicación	332
Añadir una barra de menús.....	332
Mostrar cajas de diálogo.....	334
EJERCICIOS RESUELTOS	336
EJERCICIOS PROPUESTOS	340
 CAPÍTULO 12. ACCESO A UNA BASE DE DATOS.....	 341
SQL	342
Crear una base de datos	342
Crear una tabla.....	342
Escribir datos en la tabla.....	344
Modificar datos de una tabla	344
Borrar registros de una tabla.....	345
Seleccionar datos de una tabla.....	345
Crear una base de datos	347
Base de datos Microsoft SQL Server	349
ADO.NET.....	350
Componentes de ADO.NET	351
Conjunto de datos	352
Vista de datos	354
Proveedor de datos	354

Objeto conexión	355
Objeto orden	355
Objeto lector de datos	356
Adaptador de datos	356
Un ejemplo utilizando un lector	357
EJEMPLO DE ACCESO A DATOS	359
Crear una aplicación Windows	359
Proveedor de datos	360
Crear el conjunto de datos	363
Agregar un control rejilla al formulario	364
Código	365
CONTROL BindingSource	366
VINCULAR CONTROLES AL CONJUNTO DE DATOS	367
Diseño del formulario	367
Vincular las cajas de texto con el conjunto de datos	368
Controles de desplazamiento	370
Añadir, borrar y buscar datos	372
CONTROL BindingNavigator	376
EJERCICIOS PROPUESTOS	377

CAPÍTULO 13. C# E INTERNET 379

¿QUÉ ES INTERNET?	379
Intranet	380
Terminología Internet	380
SERVICIOS EN INTERNET	383
PÁGINAS WEB	386
Qué es HTML	386
Etiquetas básicas HTML	387
Etiquetas de formato de texto	388
URL	390
Enlaces entre páginas	391
Gráficos	392
Marcos	393
Formularios	394
Entrada básica de datos	395
Caja de texto	395
Caja de clave de acceso	396
Casilla de verificación	396
Botón de opción	396
Parámetros ocultos	397
Enviar datos	397
Borrar los datos de un formulario	397

Imágenes	397
Orden de tabulación.....	398
Caja de texto multilínea.....	398
Listas desplegables	398
Tablas	401
HOJAS DE ESTILO.....	402
Clases	404
Etiquetas y <div>.....	405
XML	407
XHTML.....	408
PÁGINAS WEB DINÁMICAS	408
APLICACIONES WEB CON C# .NET.....	411
FORMULARIOS WEB.....	411
Controles de servidor HTML	412
Controles de servidor Web	413
APLICACIÓN WEB ASP.NET	415
Visual Web Developer	416
Crear un nuevo formulario Web.....	417
Agregar controles y texto a la página	419
Ciclo de vida de una página.....	421
Crear controladores de eventos para los controles.....	422
Generar la aplicación Web y ejecutarla.....	425
FORMULARIOS WEB CONSTRUIDOS SIN HERRAMIENTAS RAD.....	426
Escribir la página Web	426
Escribir el fichero de código subyacente	428
CONTROLES DE VALIDACIÓN	431
SERVICIOS WEB XML.....	433
Crear un nuevo servicio Web XML.....	434
Escribir la interfaz del servicio Web	436
Crear un proyecto cliente del servicio Web.....	440
SERVICIOS WEB DESARROLLADOS SIN HERRAMIENTAS RAD	445
Crear un servicio Web XML	445
Crear un proyecto cliente del servicio Web.....	447
Escribir la página Web	448
Escribir el fichero de código subyacente.....	450
ACCESO A DATOS DESDE FORMULARIOS WEB	452
Añadir los objetos para acceso a datos	456
Responder a los eventos	458
EJERCICIOS PROPUESTOS.....	461

APÉNDICES

A. AYUDA.....	463
---------------	-----

B. ENTORNOS DE DESARROLLO.....	465
VISUAL STUDIO 2005.....	465
Ediciones Visual Studio 2005 Express.....	465
Visual Studio 2005 Standard Edition.....	466
Visual Studio 2005 Professional Edition.....	466
Visual Studio 2005 Tools para Microsoft Office System.....	466
Visual Studio 2005 Team System.....	466
APLICACIÓN DE CONSOLA.....	467
DEPURAR UNA APLICACIÓN.....	469
ARGUMENTOS EN LA LÍNEA DE ÓRDENES.....	471
SQL SERVER EXPRESS.....	471
SQL SERVER MANAGEMENT STUDIO EXPRESS.....	474
EXPLORADOR DE BASES DE DATOS.....	475
VISUAL WEB DEVELOPER.....	476
INSTALACIÓN DE ASP.NET EN WINDOWS XP.....	476
Registro manual de ASP.NET en IIS.....	476
C. CÓDIGOS DE CARACTERES.....	479
UTILIZACIÓN DE CARACTERES ANSI CON WINDOWS.....	479
JUEGO DE CARACTERES ANSI.....	480
UTILIZACIÓN DE CARACTERES ASCII.....	481
JUEGO DE CARACTERES ASCII.....	482
JUEGO DE CARACTERES UNICODE.....	483
D. .NET PARA LINUX.....	485
INSTALACIÓN DE MONO.....	486
EJECUTAR UN PROGRAMA C#.....	486
ENTORNO DE DESARROLLO DE MONO.....	487
ÍNDICE.....	489

PRÓLOGO

C#, pronunciado *C Sharp*, es actualmente uno de los lenguajes de programación más populares en Internet. Pero, además, está disponible para el desarrollo de programas de propósito general. La idea fundamental de esta obra es dar a conocer estas facetas del lenguaje C#, sin olvidar que tiene un alcance completo sobre la *Web*.

En los últimos tiempos C y C++ han sido los lenguajes más utilizados en el desarrollo de aplicaciones en general. Ambos lenguajes proporcionan al programador el nivel de abstracción preciso para abordar el desarrollo de cualquier aplicación por compleja que sea, así como mecanismos de bajo nivel para utilizar las características más avanzadas de las plataformas sobre las que se desarrolla; pero, en general, el tiempo necesario para desarrollar una aplicación resulta largo comparado con otros lenguajes como Visual Basic, que ofrecen además de facilidad, una elevada productividad en el desarrollo de aplicaciones, aunque eso sí, sacrificando la flexibilidad que los desarrolladores de C y C++ requieren. La solución que Microsoft da a este problema es el lenguaje denominado C#. Se trata de un lenguaje moderno orientado a objetos que permite desarrollar una amplia gama de aplicaciones para la nueva plataforma Microsoft .NET, la cual se caracteriza por proporcionar utilidades y servicios para sacar un provecho total tanto de la informática como de las comunicaciones.

La palabra “NET” hace referencia al ámbito donde operarán nuestras aplicaciones (*Network* - red). C# proporciona la tecnología necesaria para saltar desde el desarrollo de aplicaciones cliente-servidor tradicionales a la siguiente generación de aplicaciones escalables para la Web, introduciendo algunos conceptos nuevos, como ensamblados, formularios Web, servicios Web, ADO.NET y el .NET Framework.

Más que otra cosa, el objetivo de C# es permitir a todos los desarrolladores en general, y a los de C y C++ en particular, abordar el desarrollo de aplicaciones complejas con facilidad y rapidez pero sin sacrificar la potencia y el control que ofrecen C y C++. Es un poco como tomar todas las cosas buenas de Visual Basic y añadirlas a C++, aunque recortando algunas de las tradiciones más ocultas y difíciles de conocer de C y C++. Esto elimina los errores de programación más comunes en C/C++. Por ejemplo:

- El recolector de basura libera al programador del peso que conlleva el manejo manual de la memoria.
- Todos los objetos creados dinámicamente así como las matrices son iniciados a cero, y aunque C# no inicia automáticamente las variables locales, el compilador avisará siempre que se intente utilizar una antes de iniciarla.
- C# unifica el sistema de tipos permitiendo ver a cada uno de ellos en el lenguaje como un objeto.

Resumiendo, C# permite escribir aplicaciones tanto para Internet como aplicaciones de propósito general; esto es, como muchos lenguajes, permite trabajar con todo tipo de datos, crear estructuras dinámicas, trabajar con ficheros, atacar a bases de datos, diseñar interfaces gráficas de usuario, etc. Más aún, C# es un lenguaje simple, potente y orientado a objetos. Su sintaxis incita al programador a generar programas modulares y fácilmente mantenibles.

Este libro ha sido escrito utilizando el paquete *Microsoft .NET Framework Software Development Kit* (SDK) que incluye todo lo necesario para escribir, construir, verificar y ejecutar aplicaciones .NET, y su propósito es ayudar al lector a aprender a programar utilizando C#. Para ello, ¿qué debe hacer? Pues simplemente leer ordenadamente los capítulos del libro, resolviendo cada uno de los ejemplos que en ellos se detallan. Evidentemente, según lo expuesto anteriormente, si el lector conoce C/C++ tendrá mucho camino recorrido.

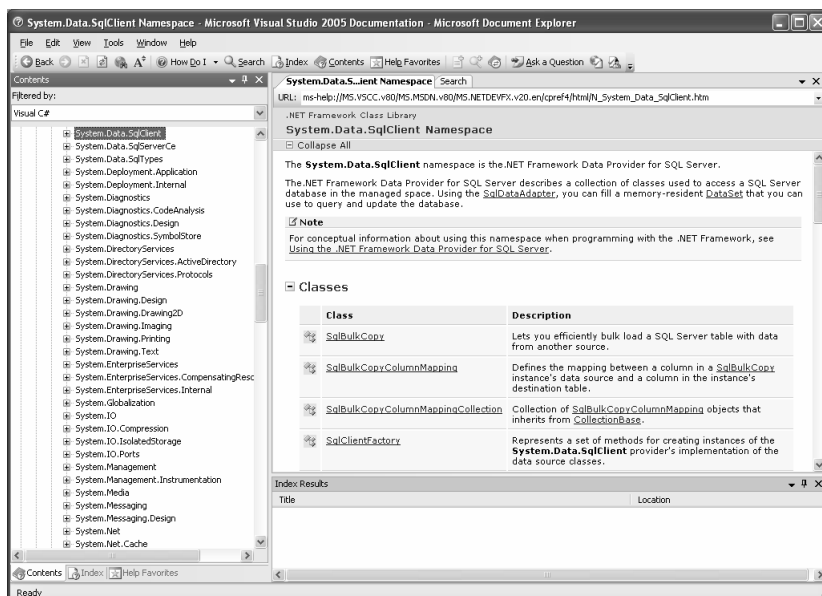
Agradecimientos

He recibido ayuda de algunas personas durante la preparación de este libro y, por ello, estoy francamente agradecido. También, deseo expresar mi agradecimiento a *Microsoft Ibérica* por poner a mi disposición, en particular, y de todos los lectores, en general, el EDI y el SDK que el estudio de esta obra requiere.

Francisco Javier Ceballos Sierra
<http://www.fjceballos.es/>

AYUDA

En este libro se ha trabajado con los espacios de nombres más comunes en Visual C# y dentro de cada espacio de nombres con las clases de uso más frecuente. Evidentemente, a medida que vaya estudiando cada capítulo le van a surgir dudas y curiosidades de por qué esta clase y no otra, o qué otros métodos tiene esta clase y, lógicamente, a muchas de estas cuestiones no va a tener respuesta en el libro; por eso este apéndice. Debe acostumbrarse a manejar la ayuda que proporciona *Microsoft .NET Framework Software Development Kit (SDK)* para tratar de dar solución a las dudas que le puedan surgir. El aspecto de la ventana de ayuda de .NET es el siguiente:



La documentación correspondiente a la API (*Application Programming Interface* - interfaz de programación de aplicaciones) de .NET Framework para Visual C# que compone la ayuda de la que estamos hablando es parte de la ayuda conocida como MSDN (*Microsoft Developer Network*) que va incluida en el CD-ROM del paquete. Cuando haya instalado Visual Studio en cualquiera de sus ediciones, asegúrese de haber instalado también la ayuda que podrá visualizar haciendo clic en *Inicio > Todos los programas > Microsoft Visual Studio 2005*.

Si observa la figura anterior, verá que tiene un aspecto similar al explorador, y que su área de trabajo se divide básicamente en las siguientes zonas:

- En la parte superior aparece una franja estrecha en la que se visualiza el título del tema que estemos viendo, así como una serie de iconos relativos a otros temas relacionados con el mostrado, requerimientos y filtros para elegir el tipo de documentación que se desea mostrar.
- En la izquierda aparece una ventana que puede visualizar la página de *contenido*, *índice* o *favoritos*, dependiendo de la pestaña sobre la que se haga clic, de las mostradas en la parte inferior de esta ventana.
- La parte inferior permite mostrar un índice con los resultados obtenidos después de una selección en la página *índice* o después de una búsqueda.
- En el resto del área de trabajo de la ventana de ayuda, la parte más grande, mostrará de forma resumida todo lo relativo al tema que haya seleccionado.

ENTORNOS DE DESARROLLO

Cuando se utiliza un entorno de desarrollo integrado (EDI), lo primero que hay que hacer una vez instalado es asegurarse de que las rutas donde se localizan las herramientas, las bibliotecas, la documentación y los ficheros fuente hayan sido establecidas; algunos EDI sólo requieren la ruta donde se instaló el compilador. Este proceso normalmente se ejecuta automáticamente durante el proceso de instalación de dicho entorno. Si no es así, el entorno proporcionará algún menú con las órdenes apropiadas para realizar dicho proceso. Por ejemplo, en los EDI que se presentan a continuación las rutas a las que nos referimos quedan establecidas durante la instalación de los mismos.

VISUAL STUDIO 2005

Visual Studio 2005 proporciona una variedad de herramientas tanto para desarrolladores individuales como para equipos de desarrollo.

<http://www.microsoft.com/spanish/msdn/vs2005/default.msp>

Ediciones Visual Studio 2005 Express

Es la nueva línea de productos que expanden Visual Studio ofreciendo herramientas ligeras y sencillas de aprender y de usar para aficionados, entusiastas y aprendices que quieren crear sitios Web y aplicaciones para Windows.

- Visual Web Developer 2005 Express Edition.
- Visual Basic 2005 Express Edition.
- Visual C# 2005 Express Edition.
- Visual C++ 2005 Express Edition.

- SQL Server 2005 Express Edition.
- y Visual J# 2005 Express Edition.

Visual Studio 2005 Standard Edition

Una flexible herramienta de desarrollo para desarrolladores de aplicaciones de línea de negocio o programadores ocasionales que estén construyendo aplicaciones móviles, basadas en Windows o para la Web.

Visual Studio 2005 Professional Edition

La completa herramienta de desarrollo para desarrolladores profesionales que trabajan solos o en pequeños equipos.

Visual Studio 2005 Tools para Microsoft Office System

La completa herramienta de desarrollo para desarrolladores profesionales construyendo soluciones sobre Microsoft Office System utilizando Excel, Word e InfoPath.

Visual Studio 2005 Team System

Herramientas de software de ciclo de vida productivas, integradas y extensibles que permiten a las empresas reducir la complejidad en la creación de soluciones orientadas a servicios.

Visual Studio Team System incluye:

- Visual Studio Team Edition para Arquitectos de Software.
- Visual Studio Team Edition para Desarrolladores.
- Visual Studio Team Edition para Testers.

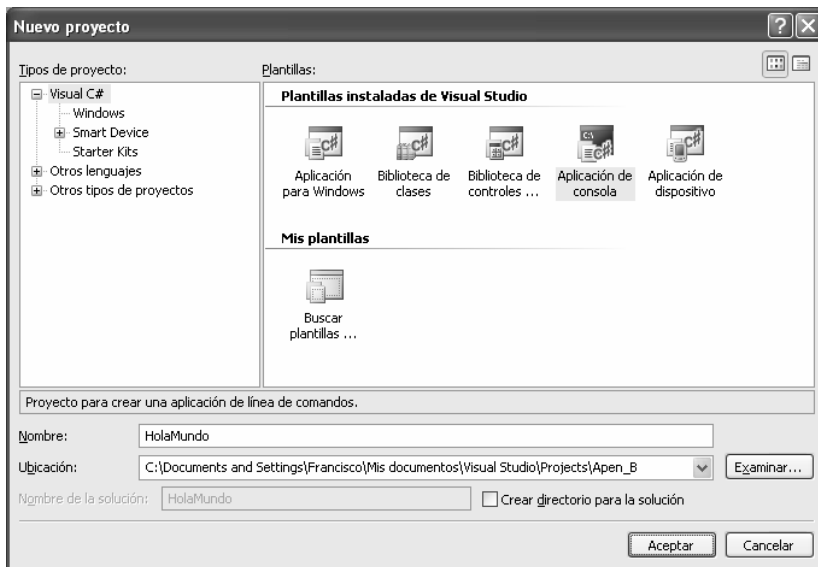
La nueva línea Visual Studio Team System también incluye Visual Studio Team Suite, que contiene los tres productos basados en estos roles, y el nuevo Visual Studio Team Foundation Server, para una colaboración en equipo mucho más eficiente.

APLICACIÓN DE CONSOLA

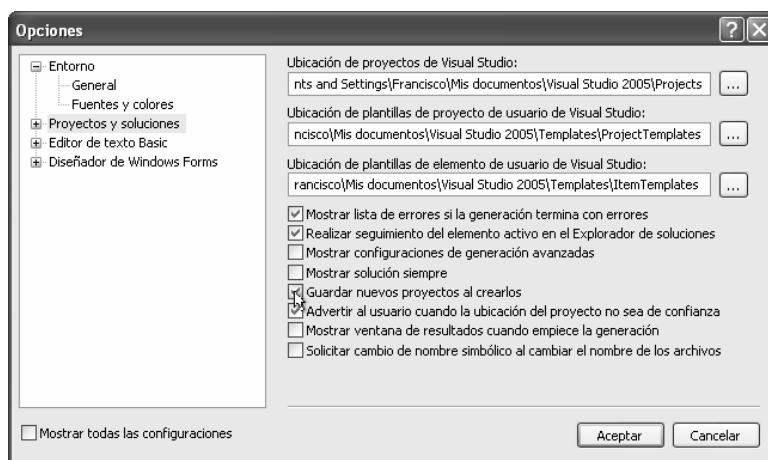
En el capítulo 2 vimos una introducción al desarrollo de aplicaciones con interfaz gráfica de usuario, pero no abordamos el desarrollo de aplicaciones de consola que en ocasiones puede ser útil. Veamos un ejemplo a continuación utilizando cualquiera de las ediciones de Visual Studio.

Para editar y ejecutar el programa *HolaMundo* utilizando cualquiera de las ediciones de Visual Studio, los pasos a seguir son los siguientes:

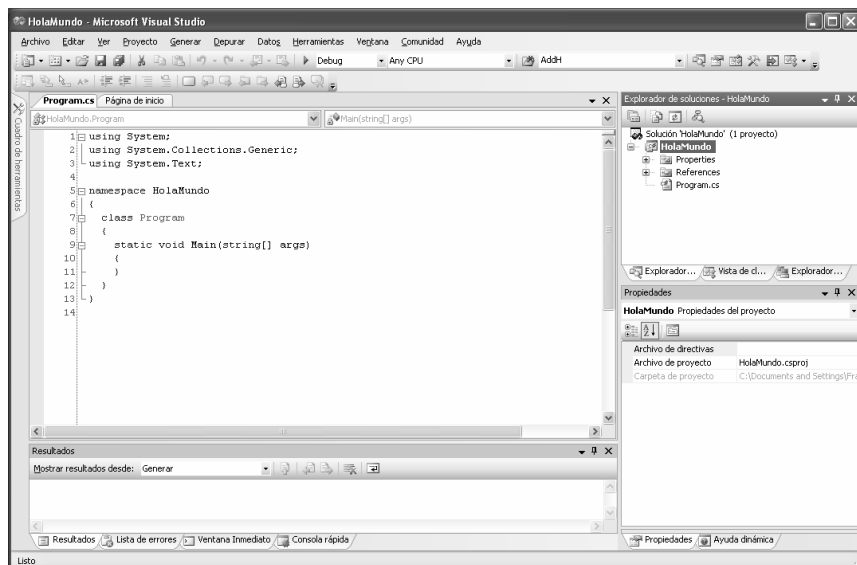
1. Partiendo de la página de inicio de *Visual Studio*, hacemos clic en *Crear: Proyecto...* para crear un proyecto nuevo, o bien ejecutamos la orden *Archivo > Nuevo > Proyecto*. Esta acción hará que se visualice una ventana que mostrará los tipos de plantillas que puede utilizar; la elección de una u otra dependerá del tipo de aplicación que deseemos construir. La figura siguiente muestra esta ventana:



Para que la ventana anterior muestre la lista *Ubicación* y la caja *Nombre de la solución* tiene que habilitar, si no lo está, la opción “Guardar nuevos proyectos al crearlos”. Para ello, ejecute la orden *Herramientas > Opciones > Proyectos y soluciones* y seleccione esta opción en la ventana que se visualiza:



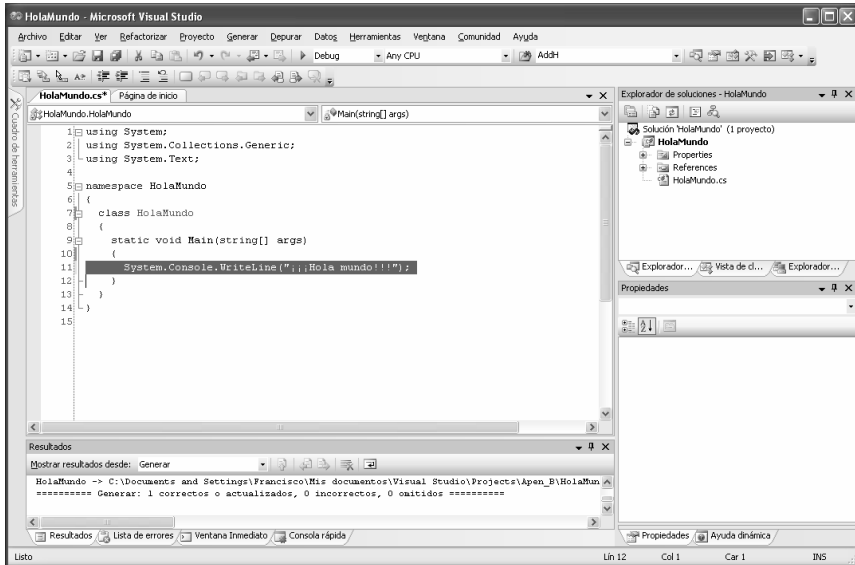
Para nuestro ejemplo, elegimos la plantilla “Aplicación de consola”. Después, especificamos el nombre del proyecto y su ubicación; observe que el proyecto será creado en una carpeta con el mismo nombre. A continuación pulsamos el botón *Aceptar* y obtendremos el esqueleto del proyecto de acuerdo con la plantilla seleccionada. Para cambiar el nombre asignado al fichero *.cs*, hacemos clic sobre *Program.cs* y utilizando el botón derecho del ratón, seleccionamos del menú contextual que se visualiza la orden *Cambiar nombre*; también podemos hacer este cambio a través de la ventana de propiedades.



Si necesitáramos añadir un fichero nuevo a este proyecto, haríamos clic con el botón derecho del ratón sobre el nombre del proyecto y seleccionaríamos

Agregar > *Nuevo elemento*. Después, elegiríamos el tipo de elemento que deseamos añadir y, finalmente, haríamos clic en el botón *Agregar*.

2. A continuación editamos el código que compone el programa. Después, antes de compilar la aplicación, podemos verificar que se trata de una aplicación de consola: *Proyecto* > *Propiedades de HolaMundo* > *Aplicación* > *Tipo de resultado* > *Aplicación de consola*.

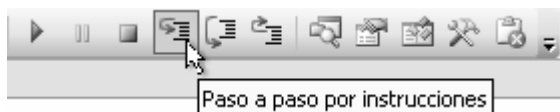


3. Para compilar el programa, ejecutamos la orden *Generar HolaMundo* del menú *Generar*. Finalmente, para ejecutar el programa seleccionamos la orden *Iniciar sin depurar* del menú *Depurar*, o bien pulsamos las teclas *Ctrl+F5*. También puede ejecutar el programa seleccionando la orden *Iniciar depuración* del menú *Depurar*, o bien pulsando la tecla *F5*. Los resultados de la compilación se muestran en la ventana *Resultados*. La acción de ejecutar el programa acarrea también su compilación si fuera necesario.

DEPURAR UNA APLICACIÓN

¿Por qué se depura una aplicación? Porque los resultados que estamos obteniendo con la misma no son correctos y no sabemos por qué. El proceso de depuración consiste en ejecutar la aplicación paso a paso, indistintamente por sentencias o por métodos, con el fin de observar el flujo seguido durante su ejecución, así como los resultados intermedios que se van sucediendo, con la finalidad de detectar las anomalías que producen un resultado final erróneo.

Por ejemplo, para depurar una aplicación utilizando el depurador del entorno de desarrollo de *Visual Studio*, ejecute la orden *Depurar > Paso por instrucciones* y utilice las órdenes del menú *Depurar* o los botones correspondientes de la barra de herramientas (para saber el significado de cada botón, ponga el puntero del ratón sobre cada uno de ellos).

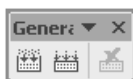


De forma resumida, las órdenes disponibles para depurar una aplicación son las siguientes:

- *Iniciar* o *F5*. Inicia la ejecución de la aplicación en modo depuración hasta encontrar un punto de parada o hasta el final si no hay puntos de parada.
- *Alternar puntos de interrupción* o *F9*. Pone o quita un punto de parada en la línea sobre la que está el punto de inserción.
- *Detener depuración* o *Mayús+F5*. Detiene el proceso de depuración.
- *Paso a paso por instrucciones* o *F11*. Ejecuta la aplicación paso a paso. Si la línea a ejecutar coincide con una llamada a un método definido por el usuario, dicho método también se ejecutará paso a paso.
- *Paso a paso por procedimientos* o *F10*. Ejecuta la aplicación paso a paso. Si la línea a ejecutar coincide con una llamada a un método definido por el usuario, dicho método no se ejecutará paso a paso, sino de una sola vez.
- *Paso a paso para salir* o *Mayús+F11*. Cuando un método definido por el usuario ha sido invocado para ejecutarse paso a paso, utilizando esta orden se puede finalizar su ejecución en un solo paso.
- *Ejecutar hasta el cursor* o *Ctrl+F10*. Ejecuta el código que hay entre la última línea ejecutada y la línea donde se encuentra el punto de inserción.
- *Inspección rápida* o *Ctrl+Alt+Q*. Visualiza el valor de la variable que está bajo el punto de inserción o el valor de la expresión seleccionada (sombreada).

Para ejecutar la aplicación en un solo paso, seleccione la orden *Iniciar sin depurar* (*Ctrl+F5*) del menú *Depurar*.

Además de la barra de herramientas *Depurar*, dispone también de la barra de herramientas *Generar* reducida que se muestra en la figura siguiente:

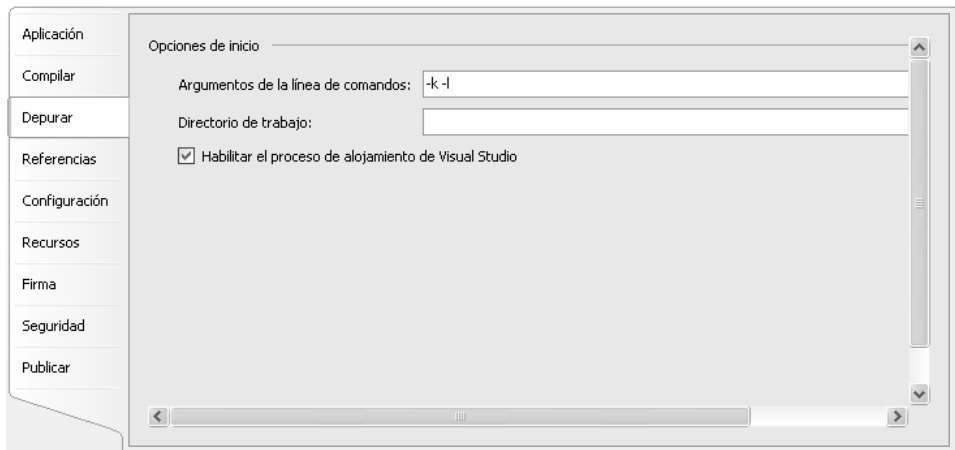


Esta barra de herramientas pone a su disposición las órdenes siguientes:

- *Generar aplicación.* Compila la aplicación y genera el fichero ejecutable correspondiente.
- *Generar solución.* Compila y genera todos los proyectos que componen la solución.
- *Cancelar.* Detiene el proceso de compilación o de generación del fichero ejecutable.

ARGUMENTOS EN LA LÍNEA DE ÓRDENES

Si ejecuta una aplicación desde el EDI que requiere argumentos en la línea de órdenes, ¿dónde se introducen estos argumentos? Haga clic sobre el nombre del proyecto con el botón derecho del ratón y seleccione la orden *Propiedades*. Después, en la ventana que se visualiza, seleccione el panel *Depurar* y escriba los argumentos según puede observar en la figura:



SQL SERVER EXPRESS

SQL Server 2005 Express es el motor de base de datos gratuito, potente, pero sencillo, que se integra perfectamente con el resto de productos Express. Se trata de una versión aligerada de la nueva generación de SQL Server.

Este producto tiene el mismo motor de base de datos que toda la familia SQL Server 2005 y utiliza el mismo lenguaje SQL. Otra característica interesante es la movilidad de las bases de datos de un servidor a otro con **XCOPY**. Con esta utilidad podemos mover un fichero MDF de una máquina a otra a cualquier ubicación

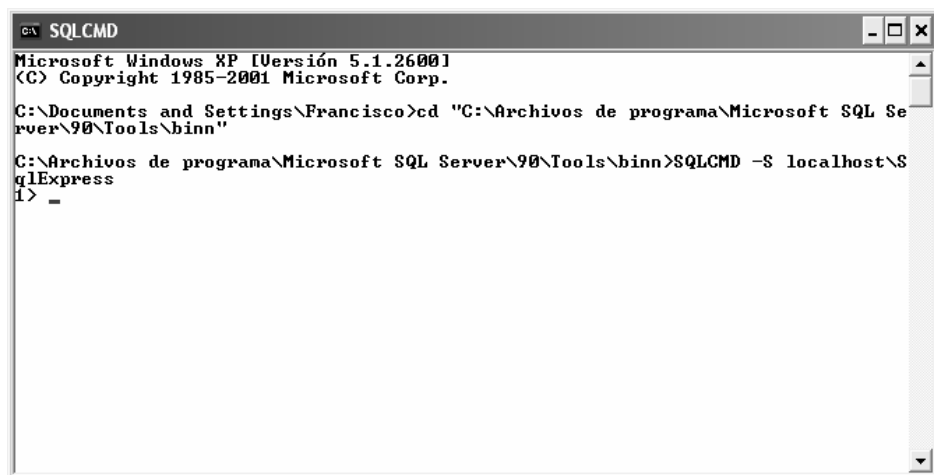
dentro de su sistema de ficheros, quedando la base de datos movida lista para trabajar. Para utilizar esta base de datos deberemos hacer uso de la opción **AttachDBFilename** en la cadena de conexión, según se muestra a continuación:

```
connectionString="Data Source=.\sqlexpress;Initial Catalog=;  
Integrated Security=True;AttachDBFileName=C:\bd\bd_editorial.mdf"
```

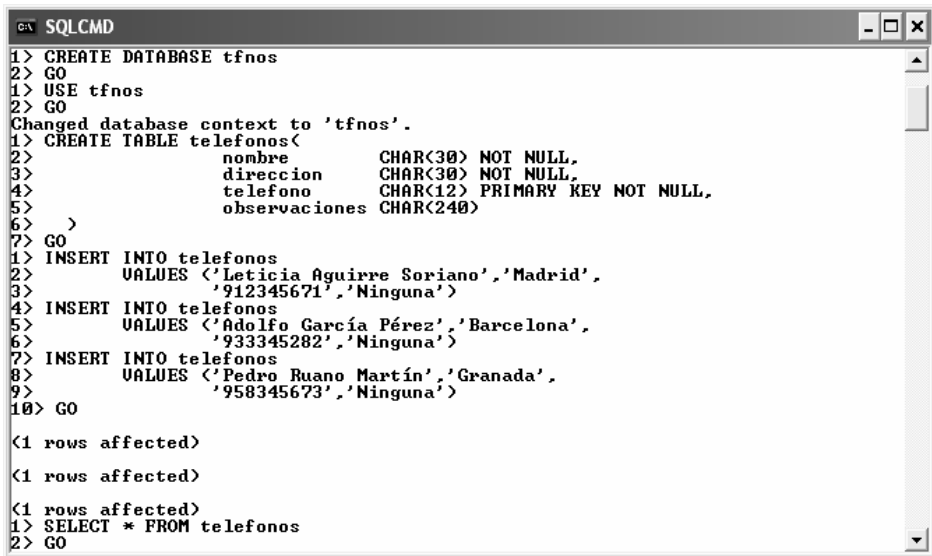
La entrada *Data Source* especifica el servidor de base de datos que vamos a utilizar y **AttachDBFilename**, la localización del fichero de base de datos. Obsérvese que la entrada *Initial Catalog* está vacía.

Para crear una base de datos utilizando SQL Server 2005 Express tiene que hacerlo desde la línea de órdenes (véase también el capítulo titulado *Acceso a una base de datos*). Para iniciar la consola que le permita trabajar contra el motor de base de datos SQL Server, localice en su instalación el fichero SQLCMD.EXE (o bien SQLCMD90.EXE), cambie a ese directorio y ejecute la orden:

```
SQLCMD -S nombre-del-ordenador\SqlExpress
```



Una vez iniciada la consola, puede escribir órdenes SQL a continuación del símbolo ">". Para ejecutar un bloque de sentencias escriba GO. Para salir, escriba QUIT. Por ejemplo, el guión que muestra la figura siguiente crea la base de datos *tfnos* con una tabla *telefonos*, añade tres filas a la tabla y, finalmente, selecciona todas las filas de la tabla con todas sus columnas:



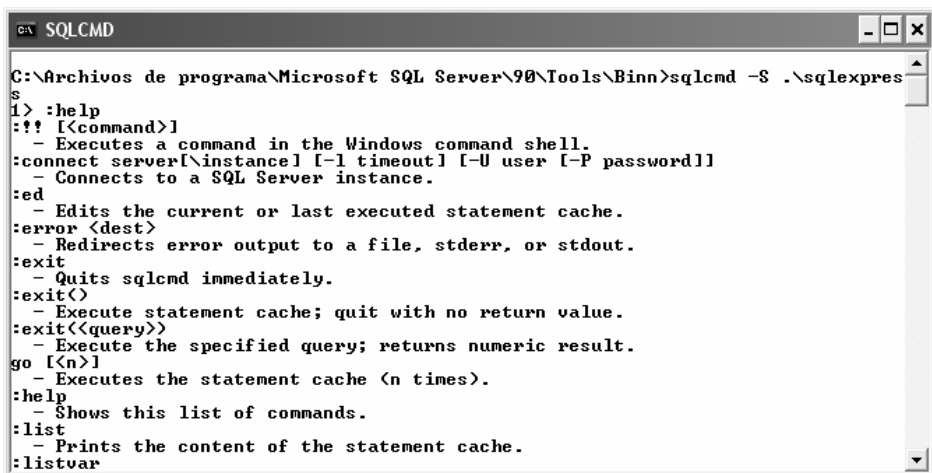
```

C:\ SQLCMD
1> CREATE DATABASE tfnos
2> GO
1> USE tfnos
2> GO
Changed database context to 'tfnos'.
1> CREATE TABLE telefonos(
2>     nombre          CHAR(30) NOT NULL,
3>     direccion       CHAR(30) NOT NULL,
4>     telefono        CHAR(12) PRIMARY KEY NOT NULL,
5>     observaciones   CHAR(240)
6> )
7> GO
1> INSERT INTO telefonos
2>     VALUES ('Leticia Aguirre Soriano','Madrid',
3>             '912345671','Ninguna')
4> INSERT INTO telefonos
5>     VALUES ('Adolfo García Pérez','Barcelona',
6>             '933345282','Ninguna')
7> INSERT INTO telefonos
8>     VALUES ('Pedro Ruano Martín','Granada',
9>             '958345673','Ninguna')
10> GO

<1 rows affected>
<1 rows affected>
<1 rows affected>
1> SELECT * FROM telefonos
2> GO

```

Para ver la relación de órdenes que puede utilizar a través de la aplicación *SQLCMD* ejecute la orden **help** como se muestra en la figura siguiente. Obsérvese que cada orden va precedida por dos puntos (:).



```

C:\Archivos de programa\Microsoft SQL Server\90\Tools\Binn>sqlcmd -S .\sqlserver
1> :help
:!! [<command>]
- Executes a command in the Windows command shell.
:connect server[instance] [-l timeout] [-U user [-P password]]
- Connects to a SQL Server instance.
:ed
- Edits the current or last executed statement cache.
:error <dest>
- Redirects error output to a file, stderr, or stdout.
:exit
- Quits sqlcmd immediately.
:exit()
- Execute statement cache; quit with no return value.
:exit(<query>)
- Execute the specified query; returns numeric result.
go [<n>]
- Executes the statement cache (<n times>.
:help
- Shows this list of commands.
:list
- Prints the content of the statement cache.
:listvar

```

También, la utilidad *SSEUtil* que se proporciona en el CD del libro le permite utilizar una consola como la anterior y, además, proporciona otras opciones de interés.

SQL SERVER MANAGEMENT STUDIO EXPRESS

Si instaló SQL Server 2005, habrá comprobado que la única herramienta que proporciona al usuario es *SQL Computer Manager* que sirve para gestionar los servicios básicos de SQL Server y para configurar los protocolos de red. Por tal motivo, Microsoft también ha desarrollado una nueva aplicación para gestionar bases de datos que puede obtener de forma gratuita de Internet en la dirección especificada a continuación (en el CD del libro se adjunta una copia):

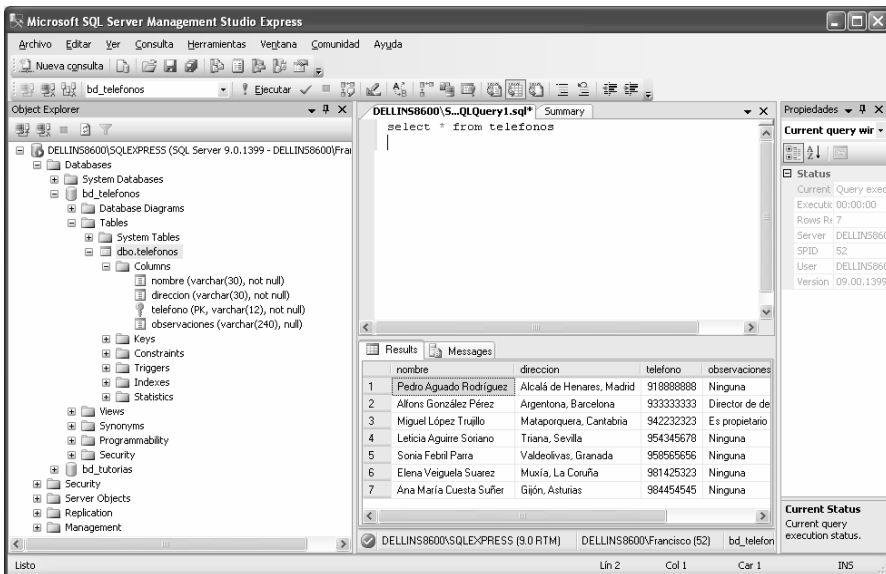
<http://www.microsoft.com/downloads>

Esta aplicación presenta una interfaz gráfica, muy sencilla de utilizar, para realizar tareas típicas como crear bases de datos, gestionar las tablas de la base, los procedimientos almacenados, crear usuarios, etc.

Cuando inicie *SQL Server Management Studio Express*, le serán solicitados el nombre del servidor de bases de datos, el tipo de autenticación, y el usuario y la contraseña sólo si eligió autenticación SQL Server:

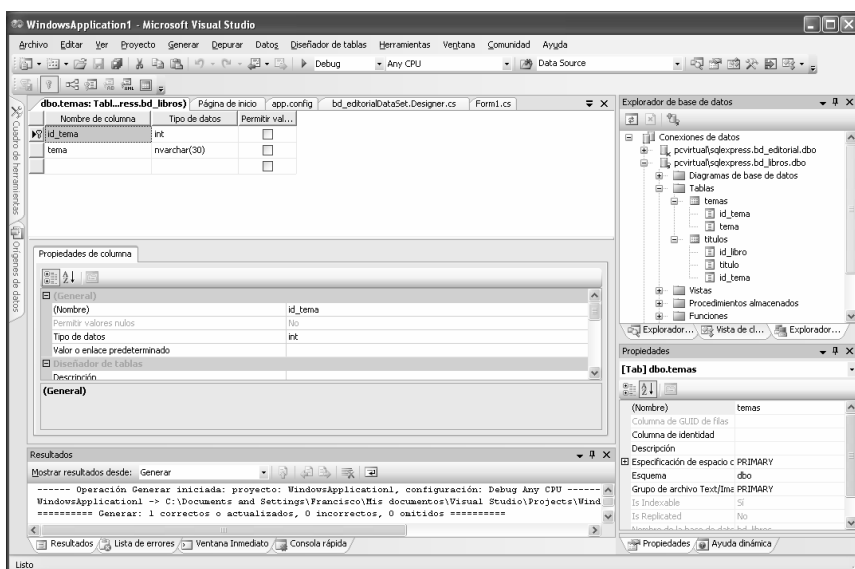


Una vez realizada la conexión con el gestor de bases de datos, le será mostrada la ventana de la figura siguiente. Seleccione en la lista del panel de la izquierda la base de datos con la que desea trabajar, haga clic en el botón *Nueva consulta* de la barra de herramientas y, después, escriba en el mismo las sentencias SQL que desee ejecutar. Para ejecutar una sentencia SQL haga clic en el botón *Ejecutar* de la barra de herramientas.



EXPLORADOR DE BASES DE DATOS

Otra forma de crear bases de datos es a través del explorador de bases de datos del EDI de Visual Studio. Si instaló Visual C# 2005 y SQL Server Express, entonces es posible añadir al proyecto elementos nuevos de tipo base de datos utilizando el explorador de bases de datos. Para más detalles, véase en el capítulo titulado *Acceso a una base de datos* el apartado *Ejercicios resueltos*.



VISUAL WEB DEVELOPER

Visual Web Developer 2005 Express (VWD) es una herramienta enfocada exclusivamente al desarrollo de aplicaciones Web dinámicas con ASP.NET 2.0. Para ello, proporciona:

- Diseñadores visuales para crear las páginas Web.
- Un editor de código potente que muestra referencias del lenguaje a medida que se escribe código, sin tener que dejar el editor. Con este editor podrá escribir código en los diferentes lenguajes .NET (Visual VB o C#) y código HTML.
- Desarrollo rápido de aplicaciones Web integradas con bases de datos. Para ello proporciona acceso integrado a SQL Server Express.

Crear un sitio Web con VWD y probar su funcionamiento es algo sencillo porque integra un servidor Web, por lo que no es necesario desplegar la aplicación en el servidor IIS de Windows (*Internet Information Server*). En realidad, cualquier carpeta con páginas *.aspx* puede ser considerada virtualmente como un sitio Web. Este servidor Web integrado será utilizado automáticamente cuando elijamos como tipo de sitio Web “sistema de archivos”. Otros tipos de sitios Web son HTTP y FTP, ambos necesitan tener instalado IIS.

La forma de trabajar con esta herramienta es la misma que hemos explicado en los últimos capítulos de este libro para crear sitios y servicios Web con Visual Studio, ya que esta herramienta está integrada en Visual Studio.

INSTALACIÓN DE ASP.NET EN WINDOWS XP

ASP.NET queda instalado automáticamente cuando instaló Visual Studio o, en su defecto, Visual Web Developer. Ahora bien, para crear y ejecutar aplicaciones para Internet necesitará un servidor de aplicaciones, en el caso de la plataforma Windows éste es IIS (*Internet Information Server*). La instalación de IIS debe ser anterior a la de .NET Framework. Si no se hizo así, ASP.NET no estará habilitado en IIS y no podremos realizar aplicaciones ASP.NET. En este caso, la solución es registrar manualmente ASP.NET en IIS.

Registro manual de ASP.NET en IIS

Para registrar manualmente ASP.NET en IIS, abra una ventana de consola, sitúese en la carpeta *C:\WINDOWS\Microsoft.NET\Framework\vx.x.xxxx* y ejecute:

```
aspnet_regiis.exe -i -enable
```

Esto habilitará IIS para la ejecución de ASP.NET, registrará las extensiones de ASP.NET (*aspx*, *asmx*, *asax*, etc.) y ya podremos empezar a trabajar con ASP.NET.

Si echa una ojeada a todas las opciones disponibles (*aspnet_regiis.exe -help*) podrá observar algunas muy interesantes, como *-c*, que instala las secuencias de órdenes del cliente de esta versión en el subdirectorio *aspnet_client* de todos los directorios de sitios IIS.

Si la operación anterior no solucionó el problema, pruebe a reparar la instalación actual a partir del CD de instalación del producto. Si esto tampoco solucionara el problema, sólo queda desinstalar el producto y volverlo a instalar.

CÓDIGOS DE CARACTERES

UTILIZACIÓN DE CARACTERES ANSI CON WINDOWS

Una tabla de códigos es un juego de caracteres donde cada uno tiene asignado un número utilizado para su representación interna. Visual C# utiliza Unicode para almacenar y manipular cadenas, pero también puede manipular caracteres en otros códigos como ANSI o ASCII.

ANSI (*American National Standards Institute*) es el juego de caracteres estándar más utilizado por los equipos personales. Como el estándar ANSI sólo utiliza un byte para representar un carácter, está limitado a un máximo de 256 caracteres. Aunque es adecuado para el inglés, no acepta totalmente otros idiomas. Para escribir un carácter ANSI que no esté en el teclado:

1. Localice en la tabla que se muestra en la página siguiente el carácter ANSI que necesite y observe su código numérico.
2. Pulse la tecla *Bloq Núm* (Num Lock) para activar el teclado numérico.
3. Mantenga pulsada la tecla *Alt* y utilice el teclado numérico para pulsar el 0 y a continuación las teclas correspondientes al código del carácter.

Por ejemplo, para escribir el carácter \pm en el entorno Windows, mantenga pulsada la tecla *Alt* mientras escribe 0177 en el teclado numérico. Pruebe en la consola del sistema (línea de órdenes).

Los 128 primeros caracteres (códigos 0 a 127) son los mismos en las tablas de códigos ANSI, ASCII y Unicode.

JUEGO DE CARACTERES ANSI

DEC	CAR	DEC	CAR	DEC	CAR	DEC	CAR
33	!	89	Y	146	'	202	Ê
34	"	90	Z	147	``	203	Ë
35	#	91	[148	"	204	Ì
36	\$	92	\	149	o	205	Í
37	%	93]	150	-	206	Î
38	&	94	^	151	—	207	Ï
39	'	96	`	152	☒	208	Ð
40	(97	a	153	☒	209	Ñ
41)	98	b	154	☒	210	Ò
42	*	99	c	155	☒	211	Ó
43	+	100	d	156	☒	212	Ô
44	,	101	e	157	☒	213	Õ
45	-	102	f	157	☒	214	Ö
46	.	103	g	159	☒	215	×
47	/	104	h	160		216	Ø
48	0	105	i	161	¡	217	Ù
49	1	106	j	162	¢	218	Ú
50	2	107	k	163	£	219	Û
51	3	108	l	164	¤	220	Ü
52	4	109	m	165	¥	221	Ý
53	5	110	n	166	¦	222	Þ
54	6	111	o	167	§	223	ß
55	7	112	p	168	"	224	à
56	8	113	q	169	•	225	á
57	9	114	r	170	•	226	â
58	:	115	s	171	"	227	ã
59	;	116	t	172	¬	228	ä
60	<	117	u	173	-	229	å
61	=	118	v	174	•	230	æ
62	>	119	w	175	-	231	ç
63	?	120	x	176	•	232	è
64	@	121	y	177	±	233	é
65	A	122	z	178	²	234	ê
66	B	123	{	179	³	235	ë
67	C	124		180	´	236	ì
68	D	125	}	181	µ	237	í
69	E	126	~	182	¶	238	î
70	F	127	☒	183	·	239	ï
71	G	128	☒	184	•	240	ð
72	H	129	☒	185	•	241	ñ
73	I	130	☒	186	•	242	ò
74	J	131	☒	187	"	243	ó
75	K	132	☒	188	¼	244	ô
76	L	133	☒	189	½	245	õ
77	M	134	☒	190	¾	246	ö
78	N	135	☒	191	¿	247	÷
79	O	136	☒	192	À	248	ø
80	P	137	☒	193	Á	249	ù
81	Q	138	☒	194	Â	250	ú
82	R	139	☒	195	Ã	251	û
83	S	140	☒	196	Ä	252	ü
84	T	141	☒	197	Å	253	ý
85	U	142	☒	198	Æ	254	þ
86	V	143	☒	199	Ç	255	ÿ
87	W	144	☒	200	È		
88	X	145	•	201	É		

UTILIZACIÓN DE CARACTERES ASCII

En MS-DOS y fuera del entorno Windows se utiliza el juego de caracteres ASCII. Para escribir un carácter ASCII que no esté en el teclado:

1. Busque el carácter en la tabla de códigos que coincida con la tabla activa. Utilice la orden **chcp** para saber qué tabla de códigos está activa.
2. Mantenga pulsada la tecla *Alt* y utilice el teclado numérico para pulsar las teclas correspondientes al número del carácter que desee.

Por ejemplo, si está utilizando la tabla de códigos 850, para escribir el carácter π mantenga pulsada la tecla *Alt* mientras escribe 227 en el teclado numérico.

JUEGO DE CARACTERES ASCII

VALOR DECIMAL	VALOR HEXA-DECIMAL	CONTROL CARACT.	CARACT.	VALOR DECIMAL	VALOR HEXA-DECIMAL	CARACT.	VALOR DECIMAL	VALOR HEXA-DECIMAL	CARACT.	VALOR DECIMAL	VALOR HEXA-DECIMAL	CARACT.	VALOR DECIMAL	VALOR HEXA-DECIMAL	CARACT.	VALOR DECIMAL	VALOR HEXA-DECIMAL	CARACT.
000	00	NUL		043	2B	+	086	56	V	129	81	ü	172	AC	¼	215	D7	#
001	01	SOH	☺	044	2C	,	087	57	W	130	82	é	173	AD	í	216	D8	≠
002	02	STX	☹	045	2D	-	088	58	X	131	83	ð	174	AE	«	217	D9	┘
003	03	ETX	♥	046	2E	.	089	59	Y	132	84	å	175	AF	»	218	DA	┐
004	04	EOT	♦	047	2F	/	090	5A	Z	133	85	ä	176	B0	▒	219	DB	■
005	05	ENQ	♣	048	30	0	091	5B	[134	86	å	177	B1	▒	220	DC	■
006	06	ACK	♠	049	31	1	092	5C	\	135	87	æ	178	B2	▒	221	DD	■
007	07	BEL	•	050	32	2	093	5D	}	136	88	è	179	B3		222	DE	■
008	08	BS	■	051	33	3	094	5E	^	137	89	é	180	B4	—	223	DF	■
009	09	HT	○	052	34	4	095	5F	_	138	8A	è	181	B5	≡	224	E0	α
010	0A	LF	☉	053	35	5	096	60	`	139	8B	ï	182	B6	≡	225	E1	β
011	0B	VT	♂	054	36	6	097	61	a	140	8C	î	183	B7	≡	226	E2	γ
012	0C	FF	♀	055	37	7	098	62	b	141	8D	ï	184	B8	≡	227	E3	π
013	0D	CR	♪	056	38	8	099	63	c	142	8E	Ë	185	B9	≡	228	E4	Σ
014	0E	SO	♫	057	39	9	100	64	d	143	8F	Ë	186	BA		229	E5	σ
015	0F	SI	☼	058	3A	:	101	65	e	144	90	É	187	BB	≡	230	E6	μ
016	10	DLE	▶	059	3B	;	102	66	f	145	91	æ	188	BC	≡	231	E7	τ
017	11	DC1	◀	060	3C	<	103	67	g	146	92	Æ	189	BD	≡	232	E8	φ
018	12	DC2	↑	061	3D	=	104	68	h	147	93	ó	190	BE	≡	233	E9	Θ
019	13	DC3		062	3E	>	105	69	i	148	94	ö	191	BF	≡	234	EA	Ω
020	14	DC4	¶	063	3F	?	106	6A	j	149	95	ö	192	C0	⌞	235	EB	δ
021	15	NAK	§	064	40	@	107	6B	k	150	96	ú	193	C1	⌞	236	EC	∞
022	16	SYN	—	065	41	A	108	6C	l	151	97	û	194	C2	⌞	237	ED	∅
023	17	ETB	↓	066	42	B	109	6D	m	152	98	ÿ	195	C3	⌞	238	EE	€
024	18	CAN	↑	067	43	C	110	6E	n	153	99	ÿ	196	C4	—	239	EF	∩
025	19	EM	↓	068	44	D	111	6F	o	154	9A	Ü	197	C5	+	240	F0	≡
026	1A	SUB	—	069	45	E	112	70	p	155	9B	€	198	C6	≡	241	F1	±
027	1B	ESC	—	070	46	F	113	71	q	156	9C	£	199	C7	≡	242	F2	≥
028	1C	FS	⌞	071	47	G	114	72	r	157	9D	¥	200	C8	⌞	243	F3	≤
029	1D	GS	↔	072	48	H	115	73	s	158	9E	Pt	201	C9	⌞	244	F4	↑
030	1E	RS	▲	073	49	I	116	74	t	159	9F	f	202	CA	⌞	245	F5	↓
031	1F	US	▼	074	4A	J	117	75	u	160	A0	á	203	CB	≡	246	F6	+
032	20	SP	Space	075	4B	K	118	76	v	161	A1	í	204	CC	≡	247	F7	≈
033	21		!	076	4C	L	119	77	w	162	A2	ó	205	CD	≡	248	F8	°
034	22		"	077	4D	M	120	78	x	163	A3	û	206	CE	≡	249	F9	•
035	23		#	078	4E	N	121	79	y	164	A4	ñ	207	CF	≡	250	FA	·
036	24		\$	079	4F	O	122	7A	z	165	A5	Ñ	208	D0	⌞	251	FB	√
037	25		%	080	50	P	123	7B	{	166	A6	°	209	D1	≡	252	FC	∩
038	26		&	081	51	Q	124	7C		167	A7	°	210	D2	≡	253	FD	'
039	27		'	082	52	R	125	7D	}	168	A8	è	211	D3	⌞	254	FE	•
040	28		(083	53	S	126	7E	~	169	A9	—	212	D4	⌞	255	FF	
041	29)	084	54	T	127	7F	⌞	170	AA	—	213	D5	≡			
042	2A		*	085	55	U	128	80	Ç	171	AB	½	214	D6	≡			

JUEGO DE CARACTERES UNICODE

UNICODE es un juego de caracteres en el que se emplean 2 bytes (16 bits) para representar cada carácter. Esto permite la representación de cualquier carácter en cualquier lenguaje escrito en el mundo, incluyendo los símbolos del chino, japonés o coreano.

Códigos Unicode de los dígitos utilizados en español:

\u0030-\u0039 0-9 ISO-LATIN-1

Códigos Unicode de las letras y otros caracteres utilizados en español:

\u0024	\$ signo dólar
\u0041-\u005a	A-Z
\u005f	_
\u0061-\u007a	a-z
\u00c0-\u00d6	À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð Ñ Ò Ó Ô Õ Ö
\u00d8-\u00f6	Ø Ù Ú Û Ü Ý Þ ß à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö
\u00f8-\u00ff	ø ù ú û ü ý þ ÿ

Dos caracteres son idénticos sólo si tienen el mismo código Unicode.

.NET PARA LINUX

Los desarrolladores sobre GNU/Linux pueden ahora también desarrollar aplicaciones multiplataforma basadas en .NET gracias al proyecto Mono.

¿Qué es Mono? Es un proyecto para construir una plataforma de desarrollo libre, basada en Linux, compatible con Microsoft .NET. Durante su desarrollo, ha contado con el apoyo de algunos pesos pesados de la industria informática, sobre todo del mundo Linux. Es el caso de Ximian, empresa dirigida por el mejicano Miguel de Icaza, que se mostró interesada en colaborar con el desarrollo de este entorno de programación desde el principio. Más tarde, la compañía norteamericana Novell adquirió esta empresa y se constituyó en uno de sus principales patrocinadores.

Mono, en su versión 1.2, incluye un compilador para C# y bibliotecas de ejecución (*runtimes*) para Java y C#, así como otras herramientas de ayuda al desarrollo de aplicaciones multiplataforma (*cross-platform*; para varias computadoras/sistemas operativos).

Mono 1.2 se encuentra disponible para Linux, Mac OS X y Windows, pudiéndose disponer del código fuente o bien de paquetes precompilados para la última versión de Mac OS (10.3, Panther), Windows 2000 o superior, y las distribuciones Red Hat/Fedora y SuSE (aunque también es instalable en otras).

Para más detalles sobre el futuro de este proyecto visite la página Web <http://www.mono-project.com>. En esta página encontrará también un enlace, *Downloads*, desde el cual podrá descargarse el software correspondiente a Mono para las distintas plataformas anteriormente comentadas, así como instrucciones para su utilización.

INSTALACIÓN DE MONO

La forma más sencilla de instalar Mono es ejecutando el asistente *mono-1.2.4_4-installer.bin* válido para todas las distribuciones Linux. Para ello:

1. Abra en su explorador de Internet la página:

<http://www.mono-project.com/Downloads>

2. Descargue el paquete *mono-1.2.4_4-installer.bin* y proceda a su instalación así:

- a) Cambie a *root* (administrador): `$ su -`
- b) Cambie a la carpeta donde está *mono-1.2.4_4-installer.bin*.
- c) Ejecute la orden: `# chmod +x mono-1.2.4_4-installer.bin`
- d) Ejecute la orden: `# ./mono-1.2.4_4-installer.bin`
- e) Siga las instrucciones dadas por el asistente para la instalación.
- f) Salga de la sesión de *root*: `# exit`

Si desea hacer una instalación personalizada, siga un procedimiento análogo al siguiente. Supongamos que nuestra máquina tiene instalada la distribución *SUSE Linux*. Para instalar la plataforma Mono para esta distribución, diríjase a la página *Downloads* de Mono, descargue el paquete *mono-1.2.4.zip* y proceda a su instalación así:

- a) Cambie a *root* (administrador): `$ su -`
- b) Descomprima *mono-1.2.4.zip* en una carpeta, por ejemplo en *mono-1.2.4*.
- c) Desde la carpeta anterior, ejecute la orden: `# rpm -Uvh *.rpm`
- d) Ahora puede borrar la carpeta *mono-1.2.4*.
- e) Salga de la sesión de *root*: `# exit`

Otra alternativa es descargar una imagen de la máquina virtual *VMware* e instalarla. La imagen de *VMware* contiene una instalación de *SUSE Linux* con Mono y varias herramientas de desarrollo y programas necesarios para probar las aplicaciones .NET en Linux usando la máquina virtual de Mono.

EJECUTAR UN PROGRAMA C#

Una vez instalado Mono, escriba el programa *HolaMundo.cs*:

```
class HolaMundo
{
```



```
public static void Main(string[] args)
{
    System.Console.WriteLine("Hola mundo!!!")
}
```

Ahora compile el programa escribiendo la orden:

```
mcs HolaMundo.cs
```

El resultado del proceso de compilación es el fichero *HolaMundo.exe*. Para ejecutarlo, escriba la orden:

```
mono HolaMundo.exe
```

ENTORNO DE DESARROLLO DE MONO

En el CD de este libro hemos incluido el entorno de desarrollo integrado *Sharp-Develop* para C# y Visual Basic sobre Windows. Pues bien, los creadores de este EDI lo han reescrito para Linux y para Mac OS X. El resultado es *MonoDevelop*.

¿Cómo podemos disponer de este EDI? Si cuando instaló Mono siguió la alternativa 1 expuesta anteriormente, *MonoDevelop* ya estará instalado. Para abrirlo, ejecute la orden:

```
monodevelop
```

El resultado después de ejecutar la orden anterior, puede verlo en la figura siguiente (en el momento de escribir este libro, aún no era posible realizar desarrollos en Visual Basic):

