

Start programming today with the
all-new Visual Basic 2008!

Visual Basic 2008

FOR DUMMIES®

**A Reference
for the
Rest of Us!**

FREE eTips at dummies.com®

Companion Web site
has all code from
the book



Bill Sempf

Author of Visual Basic 2005 For Dummies

Visual Basic 200

PARA

TONTOS



WILEY

Wiley Publishing, Inc.

No puedo empezar a dar las gracias al increíble equipo de Wiley, que me llevó siempre tan cuidado-completamente a través del proceso de actualización de este libro. Katie Feltman mantenido en mí para

revisar las ideas que se presentan, y Enochs Como se encargó de que me limité a sched-ULE. Todo el equipo de Wiley, de hecho, es lo mejor que un autor podría esperar tener. Yo he llegado a conocerlos bien durante los últimos años, y me encanta todos ellos.

Un proyecto internacional de Wendy me llevó a pedir más orientado a los detalles persona que he conocido - Mike Spivey - revisar los detalles técnicos de esta edición. Jeff Simmons revisado el código original, e hizo un buen trabajo, pero este libro es todo lo mejor para varios conjuntos de ojos.

Mi ejército de revisores de la edición original fue fantástico: Theresa Alexander, Jim Andrews, David Deloveh, Rex Mahel, Greg McNamara, Rob Morgan, Sparkes Blake y Gary Spencer. He aquí una nota especial sobre mi padre, William E. Sempf, cuya educación fondo era de inestimable ayudar en la revisión de los conceptos iniciales para el libro. Entonces, él me dejó usar como un conejillo de indias para la Parte II! Lo que un soldado!

Por último, un grito a las muchas personas que me dieron Microsoft una mano con especiales preguntas específicas acerca de VB, Visual Studio, así como el marco en general: Jan Shanahan y Susann Ragsdale en el Grupo de Apoyo de Autor y Brad McCabe, Daniel Roth, Jay Roxe y Steve Lasker, entre muchos otros, por el los equipos de desarrollo.



Agradecimientos del editor

Estamos orgullosos de este libro, por favor, envíenos sus comentarios a través de nuestro formulario de registro en línea situado en www.dummies.com/register/.

Algunas de las personas que ayudaron a llevar este libro al mercado son los siguientes:

Adquisiciones, editoriales y medios de comunicación Servicios de Composición

Desarrollo

Editor Senior de Proyectos: Marcos Enochs

Adquisiciones Editor Senior: Katie Feltman

Copy Editor: John Edwards

Editor técnico: Mike Spivey

Editorial Manager: Leah Cameron

Desarrollo de los Medios del Proyecto:

Laura Atkinson

Ayudante de redacción: Amanda Foxworth

Sr. Ayudante de redacción: Cherie Case

Dibujos animados: Rich Tenant

(Www.the5thwave.com)

Coordinador del Proyecto: Kristie Rees

Diseño y artes gráficas: Alissa D. Ellet,

Shane Johnson, Christine Williams,

Erin Zeltner

Correctores: John Greenough, Caitie Kelly

Indexador: Indexación Potomac, LLC

Edición y Redacción para Dummies Tecnología

Richard Swadley, Vice Presidente y Editor Ejecutivo del Grupo

Andy Cummings, Vice Presidente y Editor

María Bednarek, Adquisiciones Director Ejecutivo

Mary C. Corder, Director Editorial

Publicación para Dummies

Diane Graves Steele, Vice Presidente y Editor

Joyce Pepple, Directora de Adquisiciones

Servicios de Composición

Gerry Fahey, Vice Presidente de Servicios de Producción

Debbie Stailey, Director de Servicios de Composición

Mapa de contenidos

Introducción	1
Parte I: Introducción a .NET con Visual Basic	9.
Capítulo 1: Vadeando en Visual Basic	11
Capítulo 2: Uso de Visual Studio 2008	23
Capítulo 3: Diseño de aplicaciones en VB 2008	43
Parte II: Creación de aplicaciones con Visual Basic 2008	59
Capítulo 4: Creación de aplicaciones de Windows	61
Capítulo 5: Creación de aplicaciones Web	83
Capítulo 6: Creación de bibliotecas de clases	107
Capítulo 7: Servicios de Construcción Web	123
Capítulo 8: Depuración en VB 2008	139
Parte III: Integración del Programa	155
Capítulo 9: Interpretación Strings and Things	157
Capítulo 10: Toma de decisiones en el Código	175
Capítulo 11: Cómo Loopy	187
Capítulo 12: Reutilización de Código	197
Capítulo 13: Realización de Argumentos, tener ganancias	219
Parte IV: Hurgando en el marco	237
Capítulo 14: Writing Secure Code	239
Capítulo 15: Acceso a los datos	253
Capítulo 16: Trabajo con el sistema de archivos	271
Capítulo 17: Acceso a la Internet	283
Capítulo 18: Creación de imágenes	297
Parte V: La Parte de los Diez	0.309
Capítulo 19: Diez consejos para el uso de la interfaz de usuario de VB 311
Capítulo 20: Diez ideas para tomar su siguiente paso de programación	325
Capítulo 21: Diez Recursos en Internet	337
Índice	343

Introducción

W Bienvenidos a la nueva versión de Visual Basic para el 2008. Como su nombre implica, Visual Basic es una forma visual para crear nuevos programas de Microsoft La familia Windows de sistemas operativos.

Y aunque es básico en muchos sentidos, el lenguaje Visual Basic es también muy de gran alcance. Usted puede crear nuevas bibliotecas de clases y servicios Web XML, así como los programas que puede utilizar en su PC o el explorador Web, o incluso su teléfono o PDA. Todo lo que se puede hacer en Windows se puede hacer en Visual Básica.

Programación en Visual Basic es más fácil de lo que piensas, gracias a la herramientas visuales proporcionados por Visual Studio. Usted no tiene que escribir la línea tras línea de código para crear un programa de trabajo - Visual Studio genera automáticamente algo de código para que al arrastrar y soltar los componentes de las herramientas visuales. Por supuesto, ser capaz de leer y escribir código es importante, también, y este libro ofrece un montón de ejemplos de código para que pueda entender lo interno de trabajos de sus programas.

Este libro también muestra algunas de las mejores prácticas a tener en cuenta a medida que further a lo largo de su programación. Sus primeros programas puede ser muy simple, pero cuando comienza a recibir a más aplicaciones complejas, que realmente necesita para planificar su programa antes de empezar a escribir el código.

Las versiones anteriores de Visual Basic estaban completos entornos de desarrollo. La última versión de Visual Basic en realidad es sólo una parte de una de tres partes programación estrategia:

Un idioma: Para este libro, es Visual Basic 2008. Otros lenguajes populares incluyen C #, J #, Perl, y otros 24.

Un entorno de desarrollo integrado (IDE): Para este libro, es Visual Studio 2008. Otros IDEs incluyen Borland, Adobe, y varios otras herramientas.

Un proyecto de: En este libro, se refieren a cuatro tipos de proyectos: de Windows Forms, Formularios Web, bibliotecas de clases y servicios Web XML. También puede utilizar VB la construcción de servicios de Windows, aplicaciones de consola, aplicaciones de Smart Device- ciones, Formularios Mobile Web, y muchos otros tipos de proyectos.

Acerca de este libro

No importa quién seas o lo que su fondo es, usted no es un maniquí si usted está leyendo este libro. Es posible, sin embargo, ser un muñeco de lo visual Basic es, cómo utilizarlo, o por lo que puede ayudarle a hacer su mejor trabajo.

Este libro está expresamente diseñado para hacer que un buen programador de Visual Basic. Como tal, no pasar cada página de hablar acerca de las características de la lengua, cómo utilizar Visual Studio o cómo conectarse a una base de datos. Me paso una feria cantidad de tiempo a hablar sobre cómo tomar buenas decisiones, desarrollar el derecho software para el problema que necesita resolver, y no cometer errores comunes.

Visual Basic - a pesar de todas las apariencias - es realmente muy fácil de usar. Gran parte de la complejidad del lenguaje se esconde en las herramientas proporcionadas por Microsoft. Muchas de estas herramientas no son expresamente para Visual Basic, pero se vuelven a muy importante para su éxito como programador. Este libro también trata de esas herramientas, porque hacen buenos programas de escritura, trabajando más rápido y más fácil.

Este libro también trata de usted, el programador. Soy un programador como tú. Yo se han escrito en BASIC desde 1981, y he vivido a través de todas las subidas y downs. Por cerca de 15 años, Visual Basic es un programa, no un lenguaje, y vivido eso. Ahora las cosas han cambiado - Visual Basic es de nuevo un lenguaje (Visual Studio es el programa). En este libro, me ayudará a ser un buen Programador de Visual Basic.

Convenciones utilizadas en este libro

He escrito este libro para ser fácil de leer mientras se está programando. Como tal, utilizo ciertas convenciones para hacer más fácil la lectura:

Palabras que quiero enfatizar o que estoy definiendo se colocan en cursiva.

Los términos que se utilizan en el código de Visual Basic están en fuente monotipo.

Las selecciones del menú verá así: Archivo ⇨ Nuevo. Esta es la abreviatura de "From En el menú Archivo, seleccione Nuevo."

Puedo utilizar listas numeradas para guiarle a través de un proceso secuencial, como la acumulación de ING una aplicación de Windows Forms. La parte de la audaz paso es una técnica descripción de la acción que han de cumplir, y el texto normal (no negrita) que sigue proporciona una explicación más detallada o describe cómo he implementado el paso en mis ejemplos.

Las listas con viñetas se utilizan para crear listas memorables. Para mí, uno de los más duros temas acerca de la programación es recordar los puntos clave, como características o mejores prácticas. Yo uso las listas con viñetas para ayudar con ese tipo de cosas.

Los ejemplos de código se desglosan del resto del párrafo, como sigue:

```
Si DateNumber.Text.Length > 0 Then  
    DateInterval = CInt(DateNumber.Text)  
End If  
NextWeek.Text = DateChooser.Value.Add(TimeSpan.FromDays(7)).ToString()
```

Los bloques de código se escriben normalmente de tal manera que usted puede copiar a la derecha en su programa. Ellos estarán en fuente monotipo, ya veces se tienen avances de línea (el espacio y el carácter de subrayado al final de la línea) en lugares inadecuados ya que la página impresa es sólo tan grande. Recordar que cuando se está escribiendo el código y lo están viendo en la pantalla, usted no tendrá que utilizar muchos avances de línea. Si usted tiene una pregunta acerca de dónde descanso debe ser, echa un vistazo a el código de ejemplo, que se puede encontrar en este libro compañero sitio Web, www.vbfordummies.net.

Lo que usted no tiene que leer

Si no se trabaja con gráficos en este momento, puede omitir el capítulo sobre gráficos. Si no utiliza una base de datos, puede omitir el capítulo de base de datos. Ver dónde voy? Si usted no utiliza los servicios Web, usted no tiene que leer sobre ellos.

Efectivamente, este es un libro modular. Aparte de la Parte I, que todo el mundo necesita para leer, no hay requisitos para leer cualquier cosa en cualquier orden particular. Lea lo que necesita, e ignorar el resto hasta que alguien entra en su algo oficina y las necesidades de hecho en ese tema. A continuación, puede tirar el libro de nuevo y leer ese capítulo.

Si usted tiene una copia de Visual Basic 2005 For Dummies, se encuentra simi-muchos gularidades a este libro. No hubo demasiados cambios entre 2005 y VB 2008, por suerte. Me esforzaré para documentar las diferencias entre este libro y el libro de 2005, y los publicará en www.vbfordummies.net.

Suposiciones Tontas

Supongo que con la compra de este libro y de leerlo, usted está interesado en la búsqueda de cómo programar en Visual Basic. Más allá de eso, también se supone que tiene el siguiente:

- Un PC con un cierto sabor de Windows (Windows Vista o XP, lo más probable)
- Una copia de Visual Studio 2008 Professional instalado en su PC
- El acceso a la Internet, para ejemplos de código descarga y lectura adicional

¿Cómo está organizado este libro

Este libro está destinado a ser leído como una serie de artículos, pero puede fácilmente ser utilizado como una referencia o leer directamente a través. Recomiendo su lectura a su computador, con el correr de Visual Studio.

Cada parte está diseñado para enseñarle algo que usted necesita saber. El único parte que le recomiendo encarecidamente que lea, sin embargo, es la Parte I, "Conociendo .NET Uso de Visual Basic." Despues de eso, se puede leer lo que usted necesita para hacer el trabajo hecho, o leer todo el libro hasta el final - que depende de usted.

Parte I: Introducción a .NET. Uso de Visual Basic

Después de un comienzo del salto rápido, discutir las herramientas y conceptos en esta parte. Capítulo

1

es una introducción Hello World con la lengua, que experimentó programa VB-dores sean de utilidad y los nuevos programadores encontrarán vital. El capítulo 2 es un recorrido por la herramienta de desarrollo que va a utilizar, Visual Studio 2008.

El capítulo 3 es sin duda el capítulo más importante en el libro. Se trata diseñar un buen software con Visual Basic. Es posible que desee leer que una dos veces. Quería imprimir dos veces, pero el editor no me lo permitió.

Parte II: Creación de aplicaciones con VB 2008

Esta parte le ayudará a comenzar la programación. Usted encontrará aquí un capítulo para cada de los cuatro tipos de proyectos más utilizados (Windows Forms y Web, archivos DLL y Servicios Web XML) y luego un capítulo a discutir cómo depurar todos ellos.

Parte III: Hacer sus programas de trabajo

Esta parte es realmente acerca de Visual Basic el idioma. Usted ve, los proyectos discutió en la Parte II se dispone realmente de un montón de idiomas. Usted puede escribir un proyecto de formularios Windows Forms en Java (Microsoft lo llama J #) si así lo desea. La actual lenguaje VB no se dispara hasta que se escribe un programa que necesita más que el diseño visual.

Parte IV: Excavando en el Marco

Por último, en la Parte IV, nos fijamos en la última parte del rompecabezas - lo que el NET. Marco puede hacer por usted. Toneladas de herramientas que están integradas en el marco de trabajo están disponibles para todos los idiomas, pero tienen un toque especial cuando se utiliza con Visual Basic.

Comienzo con las cosas importantes, específicamente la seguridad. Entonces me referirá a los datos, archivos, redes, y el dibujo. A lo largo de todos estos capítulos, proporciono examen de código plos que pueden ayudarle con los problemas más difíciles que pueden surgir en su carrera VB.

Parte V: La Parte de los Diez

Algunas cosas encajan en ninguna parte. Eso es lo que la Parte de los Diez es para. En esta parte, Col- seleccionado los consejos más útiles que no encajaban en otra parte y las convirtió en top- ten de las listas. Para más de estos tipos de listas, echa un vistazo a compañero de este libro
Sitio web en www.vbfordummies.net.

Iconos utilizados en este libro

Una de las cosas que más me gusta de la serie de libros para Dummies es la facilidad de la lectura. Datos importantes se distinguen fácilmente de consejos o técnicas detalladas de esta serie fresca de los iconos en los márgenes. Espero que os sean tan útil como yo.



Este es el ícono que utiliza con más frecuencia. En él se destacan una mejor práctica, un común algo de uso, o simplemente que creo que usted encontrará bueno saber acerca de una característica o herramienta.



Yo uso este ícono para señalar algo que quieras, bueno, recuerda. La gotchas famosos que todos los programadores son tan familiares con conseguir este ícono. Algunos usos no siempre son evidentes. Espero poder ayudarle a ayudarse a sí mismo señalando hacia fuera.



Este ícono señala algo que no quieras hacer, a menos que esté dispuesto a sufrir las consecuencias. Lea los párrafos junto al ícono de advertencia para que usted sabrá cómo evitar la trampa de caída, o error.



Estos íconos son enlaces a sitios donde el objeto My, nuevo en Visual Basic 2008, puede ser útil.



A veces, te dan más información que lo que realmente necesita. Cuando hago eso, Trato de usar el ícono de cosas técnicas. Va a encontrar cosas que no querían conocer el funcionamiento interno del Marco. NET, ideas de diseño, y otras cosas geek junto a este ícono.



Yo uso este ícono para seleccionar una nueva característica de Visual Basic 2008.

A dónde ir desde aquí

Si usted es completamente nuevo a Studio Visual Basic y Visual, empiece flip-ping a la página y leer el Capítulo 1. Si usted está interesado en buscar un tema particular, leerlo a través de la tabla de contenido o el índice y gire a la página indicada.

Cuando usted se siente más familiarizado con el tipo de lenguaje, herramienta, y de proyectos, ramificarse por el control de la lista de consejos en la Parte de los Diez para tomar su siguiente rutina.

Usted puede, por supuesto, leer el libro hasta el final. Otra gran manera de averiguar cómo Visual Basic obras es seguir una ruta del proyecto hasta el final a través. Por ejemplo, comience con un proyecto de Windows Forms con el sistema. Dibujo elementos, e ir a través de los ejemplos en los capítulos que tratan sobre esos temas (capítulos 4 y 18, en este caso).

Asegúrese de usar los ejemplos de código de este libro, siempre en www.vbfordummies.net. Ellos le darán un amplio punto de partida para muchos otros, programas más grandes que usted puede ser que desee escribir.

Usted también puede estar en la posición en la que tienes que aprender rápidamente a utilizar este idioma para su trabajo, y podría haber bibliotecas especializadas y las normas que usted tiene que trabajar con allí. Le recomiendo que tome el libro a casa, donde se puede trabajar sin distracciones, y se da una buena base en la idioma. Entonces usted puede tomar el libro de vuelta al trabajo y lo utilizan como una referencia para sus esfuerzos de programación en el futuro.



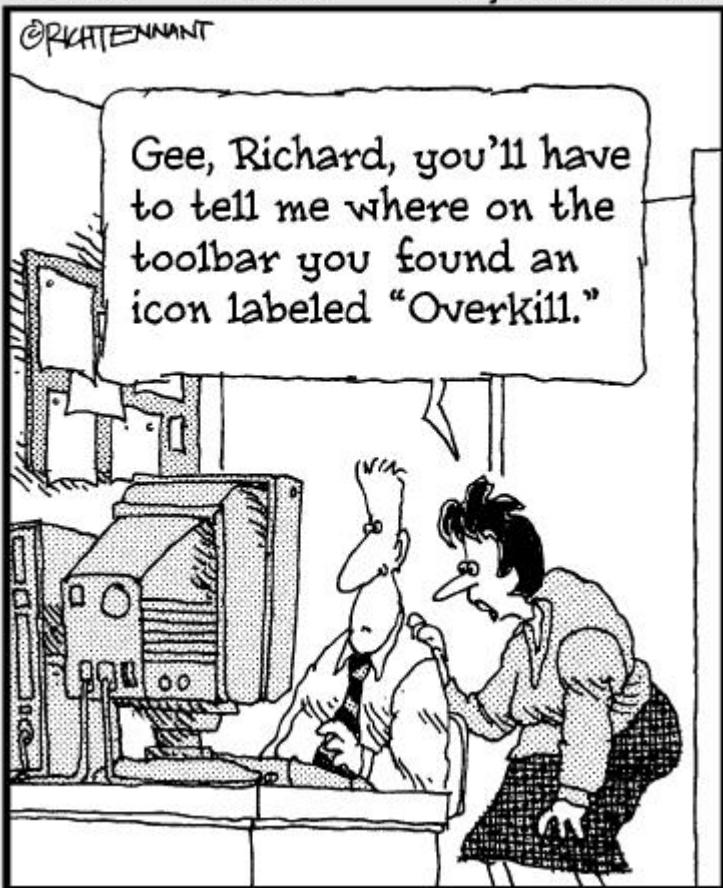
Las cosas cambian en el mundo del software y el software de Microsoft es especialmente propensos al cambio. Las cosas han cambiado probablemente desde que escribí este libro. Si el cambios en el software, no puedo actualizar los libros que ya se han impreso. Sin embargo, pueden (y lo hacen) una lista de las erratas y actualizaciones en compañero de este libro. Sitio Web, www.vbfordummies.net. Compruébelo usted mismo a menudo.

Parte I

Conociendo .NET mediante Visual Basic

The 5th Wave

By Rich Tennant



En esta parte. . .

Eoda persona tiene que empezar en alguna parte, y me pongo a la a partir de esta parte. Usted escribe su primer Visual Basic programa, y, al hacerlo, descubre algunas de las ideas detrás del marco. NET (la columna vertebral de esta versión de la lengua). A continuación, ir a hacer la necesaria sólo lectura Ing. en este libro. En primer lugar, usted se pasa de la utilización de la herramienta, Visual Studio. En segundo lugar, el diseño de la aplicación de ejemplo que se escribe en la siguiente parte.

Capítulo 1

Vadeando en Visual Basic

En este capítulo

Ver dónde encaja en Visual Basic con. NET

Escribiendo tu primer programa de Visual Basic 2008

Explorar el nuevo poder de Visual Basic

T o empezar con Visual Basic 2008, le recomiendo que saltar a la derecha y escribir software! Y para ayudarte con este enfoque assertivo, esta capítulo ofrece justo lo que necesita para poner a prueba las aguas del Visual Basic piscina y sentirse cómodo con su posición en el entorno de Visual Studio más grande.

Entonces usted puede conseguir sus pies mojados como a construir Hello World - su primera VB 2008 Windows Forms - aquí mismo en las primeras páginas! Usted averiguar cómo iniciar Visual Studio 2008 (la herramienta de desarrollo para su VB aplicaciones), la forma de iniciar un nuevo proyecto, y cómo construir una forma visualmente y hacer que funcione con el código.

También en este capítulo, le doy una mirada a la fuerza más profunda de Visual Basic. En concreto, se introduce la forma de VB 2008 se integra con Microsoft. NET Marco y ofrecen una visión sobre lo que significa para usted como programador.

Función Visual Basic en el Marco

Microsoft creó el Marco. NET para que el desarrollo de los diversos Los sistemas operativos Windows más fácil. Pero a causa de las diferencias entre los Visual Basic 6.0 y Visual Basic 7.0 (la primera. NET), la mayoría VB-devel opers encontrado un desarrollo mucho más difícil. Por ejemplo, VB 7.0 hizo todas las variables en objetos, que eliminó la capacidad del programador para definir un tipo de variable sobre la marcha.

Pero el desarrollo de aplicaciones en. NET no tiene por qué ser más difícil de lo que era en VB 6.0. El Framework. NET y Visual Basic 2008 pueden ser herramientas poderosas, y el truco es descubrir cómo trabajan juntos a través de Visual Studio Entorno de desarrollo integrado (IDE).

Parte de la dificultad que muchos programadores se enfrentan al pasar a la RED. Marco es la terminología que puede prestarse a confusión. Me gustaría poner el problema con la terminología a la cama ahora mismo, así que echa un vistazo a esta lista de los condiciones potencialmente confusos utilizados en el desarrollo de .NET.:

Visual Basic 2008: El lenguaje de programación descrito a lo largo este libro. Ya no se puede ejecutar o cargar Visual Basic como una separata de entidad. Se trata simplemente de un lenguaje de programación que se dirige a la Microsoft .NET Framework, que es el término siguiente de la lista.

.NET Framework: La capa que se encuentra entre la lengua (en este caso, Visual Basic) y el sistema operativo, que puede ser Windows 98, Windows Me, Windows 2000, Windows XP, Windows Server 2003, o cualquier de las subversiones de aquellos (como la edición Tablet PC). El .NET. Marco de capa sirve para proporcionar la funcionalidad basada en la operación del sistema de Windows en la que reside, así como para proporcionar bibliotecas para otras funciones (como cálculos matemáticos y acceso a base de datos). La figura 1-1 es una representación visual de la relación de todas las capas en el marco.

Visual Studio 2008: La herramienta que se utiliza para crear cualquier tipo de aplicación utilizando cualquier lenguaje de programación compatible. Visual Studio reemplaza el programa de Visual Basic 6.0 que anteriormente formaba parte de Visual Studio suite (todos los componentes de la suite individuales se marcaron Versión 6.0). ¿Cuándo usted va a escribir un nuevo programa en el medio ambiente. NET, se ejecuta Visual Studio 2008 y seleccionar el tipo de programa que desea escribir en el lenguaje idioma que desea utilizar. Por ejemplo, usted puede optar por crear un programa de Windows Forms utilizando el lenguaje Visual Basic, sólo Como en los viejos. O es posible que desee escribir una aplicación para un inteligente dispositivo usando C#. También puede mezclar los idiomas, por ejemplo, escribiendo el formularios en VB y las clases en C#. En este libro, voy a utilizar VB para todo - porque es un libro acerca de VB!

.NET Framework 3.5

LINQ

ASP.NET 3.5

CLR Add-in
Marco

Adicional
Mejoras

.NET Framework 3.0 SP1 +

Figura 1-1:
El .NET.

Ventanas
Presentación
Fundación

Ventanas
Comunicación
Fundación

Ventanas
Flujo de
trabajo
Fundación

Ventanas
CardSpace

Marco
jerarquía

NET Framework 2.0 SP1 +

Windows Forms: El nuevo término para un viejo anticuado Visual Basic aplicación. Este término se refiere a una aplicación que se escribe con el .NET. Marco y tiene una interfaz de usuario de Windows.

Los formularios Web Forms: El término para una aplicación con una interfaz de página Web autodiseñada con el Marco. .NET. Creación de una aplicación Web Forms es muy similar a escribir una aplicación de Windows Forms.

Servicios Web: Las bibliotecas de clases que se escriben utilizando un estándar definido por las mismas personas que se definen los estándares para la World Wide Web. Los servicios se utilizan para la interacción entre los sistemas divergentes.



El Framework. .NET es lo que usted ya sabe que la capa de Win32 en el viejo sistema de Windows DNA. Al igual que el nuevo. .NET Framework, la capa Win32 le dio la posibilidad de llegar a las funciones del sistema operativo cuando desarrollo de una plataforma de Windows. Además, el Marco. .NET incluye una gran cantidad de funcionalidad de complemento, tales como bibliotecas de matemáticas y datos, que hace que el programamiento sea una experiencia más coherente.

Básicamente, todo lo que Windows lo hace se expone en el Marco. .NET. En concreto, el Marco. .NET proporciona un nombre de programación a casi todos los objetos y el evento que Windows puede controlar. Un programador puede utilizar ese nombre para referirse a todo lo existente en el sistema operativo. ¿Es necesario decir la impresora para realizar dos copias de su documento? Trate refiriéndose a My.Computer.Printers.DefaultPrinter.PrinterSettings.Copies = 2. ¿Es usted necesitado pintar algún elemento en la pantalla azul? Trate refiriéndose a Sistema.Drawing.Brushes.Blue.

En este mundo. .NET, el lenguaje de programación se convierte simplemente en una forma de interactuar con el marco y, por lo tanto, con el sistema operativo Windows. Todos los programas tienen un conjunto de reglas establecidas para manejar el flujo (decisiones, loops, y así sucesivamente) en los programas. Visual Basic proporciona un conjunto de servicios de reglas, y el marco de trabajo proporciona los objetos y eventos para interactuar.

Decir Hola a VB 2008!

En las secciones siguientes, empezar con el clásico Hello World program. Aunque esta no es la única aplicación más emocionante que se puede construir, ayuda a asegurarse de que su entorno de desarrollo está configurado el mejor manera posible.

¿Cómo se diferencia de VB 2008 VB 6

Visual Basic 6 es un programa independiente, y Visual Basic 2008 es una lengua en un mayor desarrollo del sistema. Para volver a VB raíces, Basic es un lenguaje de programación utilizado

Hace 20 años como parte de MS-DOS. En 1985, Basic se convirtió en Visual Basic y se convirtió en una parte de la aplicación de Windows herramienta de construcción. Usted encontrar un montón más para el programa de Visual Basic 6 que sólo la lengua - su forma de construcción software, por ejemplo, se llama Ruby.

Visual Basic ha pasado por algunas revisiones desde VB 6. VB 2002 (también denominado VB 7), VB 2003 (VB 7,1), y VB 2005 (VB 8) están a sólo la revisión de

el idioma, ya que utiliza el Marco. NET. VB 2002 trajo a bordo de una forma totalmente nueva de pensar sobre la creación de aplicaciones en Windows, y VB 2005 trajo muchos de los rasgos que VB 6 programadores dependía - como la facilidad de uso.

En Visual Basic 2008, tiene una nueva forma de crear experiencias de usuario y, con él, una nueva forma para interactuar con el sistema operativo Windows. La verdadera razón para comprender el alcance de esta desarrollo de un sistema más grande - y el complejidad del Marco. NET que rodea VB 2008 - es para que la lectura de libros relacionados y documentación es más fácil.

Instalación de Visual Studio

Para seguir este ejemplo, es necesario que comience a correr Visual Studio 2008, que es el entorno de desarrollo utilizado en este libro para construir aplicaciones en Visual Basic. Antes de poder ejecutar Visual Studio, es necesario instalarlo!

Visual Studio viene en varias ediciones:

Team System: Diseñado para el personal de programación de pleno derecho en las grandes corporaciones, esta edición incluye a gran escala de aplicaciones herramientas de diseño del sistema, como desarrollo basado en pruebas y Team Foundation Server.

Professional Edition: Diseñado para los desarrolladores que trabajan con los usuarios en un ajuste independiente. La edición profesional es más común para la solo desarrollador o para el desarrollo de aplicaciones de tamaño medio. Esta es la edición que utilizo en este libro.

Standard Edition: Diseñado para la construcción de más pequeño, autónomo aplicaciones, esta versión es perfectamente funcional para 80 por ciento de las solicitudes construido. Pero si va a construir grandes sistemas que necesitan ser empresa calidad y puede tener muchos usuarios, vaya para la edición Professional.

Express Edition: Diseñado para estudiantes y aficionados. Esta versión carece muchos de los tipos de proyectos que tienen las otras versiones. Si está ejecutando Express, algunos de los ejemplos de este libro no va a funcionar para usted. En esto

sitio Web del libro (www.vbfordummies.net), he publicado algunas expresó artículos y algunos proyectos que han alterado para trabajar en la edición Express.





Si usted no tiene acceso a la biblioteca de MSDN (técnico práctico de Microsoft archivo), le recomiendo conseguirlo. Puede cargar hasta una máquina con su opción de código de ejemplo, documentación y otro material de referencia sobre Ediciones de Visual Studio, sistemas operativos y software de servidor. Usted puede encontrar fuera de la biblioteca en <http://msdn.microsoft.com>, y usted puede comprar suscripciones de varios distribuidores, incluyendo a su distribuidor de software favorito.

Instalación de Visual Studio puede ser duro, así que recomiendo ir con los valores por defecto para tu primera vez. El proceso de instalación tarda bastante tiempo, también. Aunque está utilizando el DVD, le espera para pasar dos horas de instalar el programa. Si está trabajando desde los CD, le espera para pasar cuatro horas.

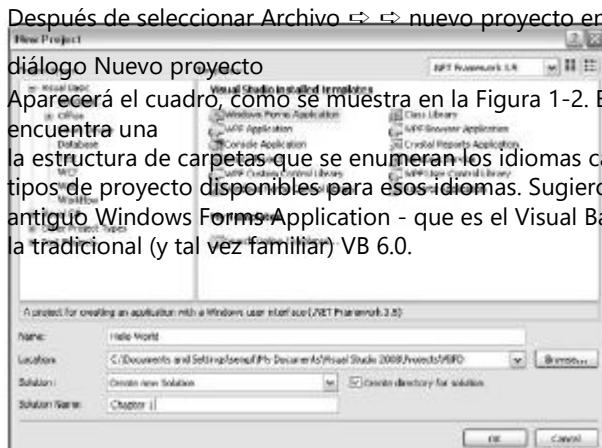
Después de instalar Visual Studio, se puede ejecutar seleccionando Inicio ⇢ Todos los programas Microsoft Visual Studio 2008 ⇢ Microsoft Visual Studio 2008. El entorno cargas, y se puede empezar a trabajar en un programa, seleccione Archivo ⇢ New Project. Después, usted necesita para tomar decisiones sobre el tipo de proyecto y el lenguaje, como se describe en la siguiente sección.

Inicio de un proyecto de formularios Windows Forms

Después de seleccionar Archivo ⇢ nuevo proyecto en Visual Studio, el cuadro de diálogo Nuevo proyecto

Aparecerá el cuadro, como se muestra en la Figura 1-2. En el panel Tipos de proyecto, se encuentra una estructura de carpetas que se enumeran los idiomas cargados con la instalación y el tipos de proyecto disponibles para esos idiomas. Sugiero comenzar con un claro antiguo Windows Forms Application - que es el Visual Basic 2008 respuesta a la tradicional (y tal vez familiar) VB 6.0.

Figura 1-2:
The New
Proyecto
Cuadro de
diálogo



Para empezar la construcción de su aplicación Hello World, siga estos pasos:

1. Seleccione el tipo de proyecto en el panel Plantillas, en el nuevo proyecto Cuadro de diálogo.

Para este ejemplo, seleccione Aplicación de Windows Forms. Además, asegúrese de que Visual Basic es el idioma seleccionado en el panel Tipos de proyecto. Si que ha cargado otros idiomas durante la instalación, es posible que tenga otros opciones.

2. Escriba el nombre que desea dar a su proyecto para reemplazar el valor predeterminado nombre en el cuadro de texto Nombre.

En este ejemplo, escriba Hello World en el cuadro de texto.

3. Haga clic en el botón Aceptar.

Visual Basic carga el formulario predeterminado (Form1 llama) y lo presenta a usted en la vista Diseño. El formulario predeterminado se completa con un espacio de trabajo, la barra de título y conocidos elementos de Windows, como los botones de cambio de tamaño y el botón Cerrar. Usted hace la mayoría del trabajo de personalizar su uso de forma este punto de vista visual.

4. Haga clic en el cuadro de herramientas palabra en el lado izquierdo de la pantalla y abra el Controles de árbol común.

El cuadro de herramientas aparece, con controles de Windows Forms cargados, como se muestra en
en
Figura 1-3.

5. Haga doble clic en el control Button.

Visual Studio carga un botón al formulario predeterminado en Vista Diseño.

6. Por defecto Form1, haga clic en el control Button y arrástrelo a reposición en el formulario.

La figura 1-4 muestra el resultado de arrastrar el botón hasta la mitad de la Form1 ventana.

Esta lista de pasos que da el inicio de la aplicación de Windows Forms, que se ve como Form1 en la vista Diseño. Pero para ver que Visual Basic entra, usted tiene que encontrar el código detrás de la forma. Visual Studio le ofrece (¡Sorpresa!) La vista de código cuando esté listo para usar Visual Basic para agregar funcionalidad al formulario.

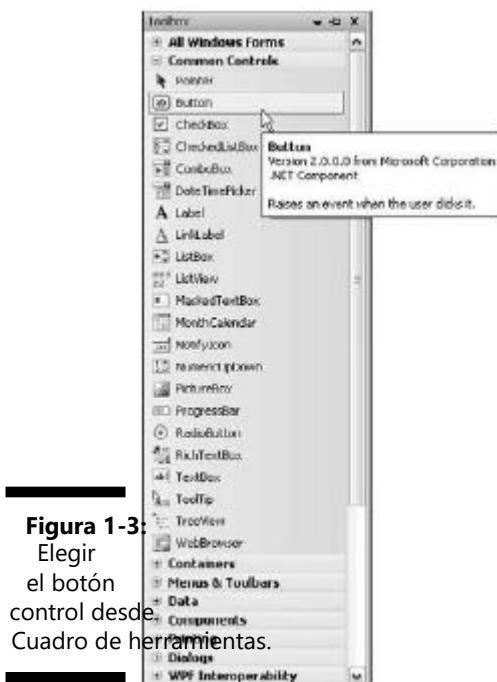


Figura 1-3:
Elegir
el botón
control desde
cuadro de herramientas.

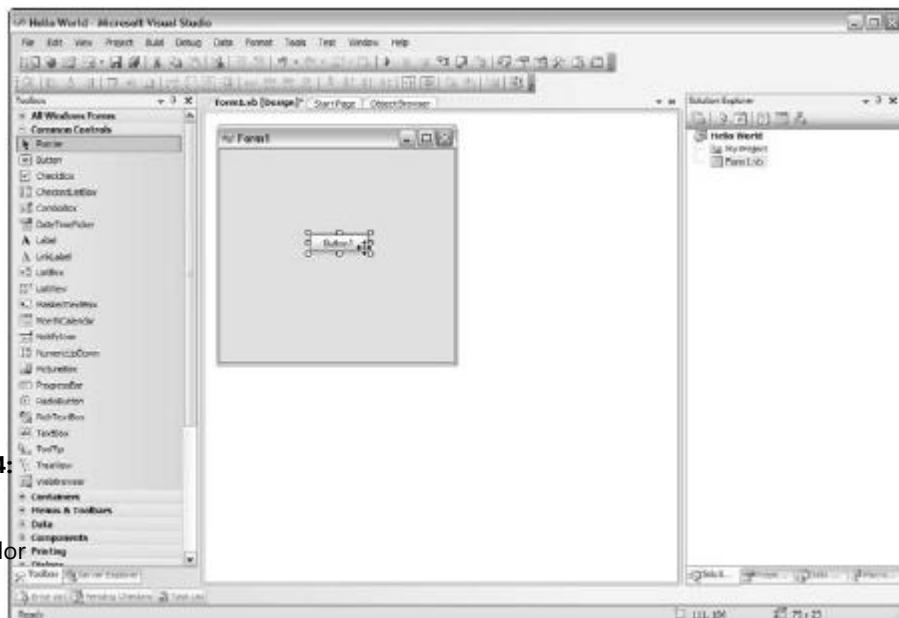


Figura 1-4:
Mover
el botón
alrededor
la forma.

Agregar funcionalidad a el formulario con código VB

Para añadir un poco de funcionalidad a Windows formar a construir en el anterior sección, siga estos pasos:

1. Haga doble clic en el control Button para entrar en Ver código.

En la ventana Vista Código, verá básico de clic de botón código que se parece a el siguiente:

```
Public Class Form1

    Private Sub Button1_Click (ByVal Como remitente _
        System.Object, ByVal e As System.EventArgs) _
        Manijas Button1.Click

    End Sub
End Class
```

Este código es una plantilla que contiene el código que se ejecuta cuando se haga clic en el botón. Visual Studio hace la parte más difícil para usted, asegurándose que el formato de la Subcomisión es correcta!

2. En la ventana Vista Código, escriba una línea de código para cambiar el texto que aparece en el control Button para Hello World.

En concreto, escriba el código siguiente en la línea que precede a la Sub End línea:

```
Button1.Text = "Hello World"
```

El código del botón ahora tiene el siguiente aspecto:

```
Public Class Form1
    Private Sub Button1_Click (ByVal Como remitente _
        System.Object, ByVal e As System.EventArgs)
        - Manijas Button1.Click

    Button1.Text = "Hello World"

    End Sub
End Class
```

Correr y operar su formulario de Windows

Así que esta experiencia está muy bien, ¿verdad? Programación con Visual Basic es tan fácil que, aquí en el capítulo 1, ya se puede escribir una aplicación de Windows Forms-ión. Pero ¿qué se puede hacer con él? Echa un vistazo a lo siguiente:

Ejecutar la aplicación de Windows Forms en Visual Studio medio-ambiente. Presione F5 y Visual Studio abre el proyecto activo como Programa de Windows. Aparece en la barra de tareas y todo. Clic el botón en el formulario, y los cambios en el texto del botón de "Hello World" (O cualquier texto que se especifica en el código). Impresionante, ¿eh? Su Formulario Windows debería parecerse a la imagen de la Figura 1-5.

Ejecute la aplicación fuera del entorno de Visual Studio. Si están todavía en modo de depuración, que tendrá que detener su programa por primera vez por utilizando el botón Detener de la barra de herramientas o cerrando la ventana del formulario.

A continuación, puede guardar y seguir adelante.

La manera sencilla de ejecutar una aplicación fuera de Visual Studio es tan sigue:

1. Seleccione Archivo \Rightarrow Guardar todo.

Visual Studio guardará su proyecto utilizando los valores por defecto proporcionados en el cuadro de diálogo Agregar proyecto.

2. Elija Generar \Rightarrow Nombre del programa Build.

En este ejemplo, elija Generar Generar solución \Rightarrow y Visual Studio compila su aplicación en un útil programa para Windows (con la extensión de archivo. exe) y lo almacena en la carpeta predeterminada.

3. Navegue a la carpeta predeterminada que contiene su nuevo Windows aplicación.

Para mi aplicación, la ruta es C:\ Documents and Settings\ Sempf\ Mis documentos\ Visual Studio 2008\ Projects\ VBFD\ Capítulo 1\ hola mundo\ bin\ Debug.

Figura 1-5:
Hola Tu
Mundo
aplicación.





Si la configuración local para el proyecto pasa a estar listo para lanzar el modo de (No se recomienda para este libro), es posible encontrar en C:\Documents and Sempf Settings\My Documents\Visual Studio 2008\Projects\VBFD\Capítulo 1\hola mundo\bin\Release.

4. Haga doble clic en el nombre de archivo para el programa compilado para ejecutarlo.

Usted puede ver una gran cantidad de archivos en la carpeta por defecto, pero en el ejemplo,

Hola World.exe es el archivo que está buscando.



Hay un método más complejo para ejecutar sus programas en VB fuera de la Entorno de Visual Studio. Utiliza un proyecto de instalación, que es una herramienta muy fresco pero más allá del alcance de este libro. Investigue el proyecto de instalación en el plazo MDSN Biblioteca cuando estés listo para averiguar más acerca de este dispositivo, que ayuda a distribuir la aplicación a otros usuarios.

Encontrar Más potencia en Visual Studio

Al principio de este capítulo, le mostró la aplicación de Windows Forms desarrollo medio ambiente y un poco de el nuevo código de Visual Basic 2008. Si usted es familiarizado con Visual Basic 6.0, la forma y el código del aspecto bastante familiar a estas alturas. De hecho, las principales herramientas de Windows de desarrollo para cualquier lenguaje de programación funcionan más o menos de esta manera.

Pero cuando se mira más allá de la forma de Windows y la estructura del código, algunos más detalles se hacen evidentes. Por ejemplo, Visual Studio tiene su código VB más allá de la forma de Windows. Las siguientes secciones le dará una visión general de el desarrollo de energía que se encuentran en Visual Studio.

Visual Studio no sólo hacer Windows!

El primer cambio evidente que fija aparte de Visual Studio como herramienta de desarrollo es la siguiente: Puede utilizar Visual Studio para escribir programas que se ejecutan en el Mundo Wide Web, así como en equipos Windows. Al hacer clic en el menú Archivo para agregar un proyecto nuevo, observe la segunda opción en el menú. Como se muestra en Figura 1-6, la opción segundo proyecto es un nuevo sitio web.

Elija esta opción para crear una aplicación web, que incorpora un conjunto serie de tecnologías - el .NET, ASP.NET, Visual Basic y HTML - que cada uno tiene un papel esencial para permitir que una aplicación se ejecute línea.

la información de la base de datos. También puede personalizar la barra de la cinta y crear e implementar complementos fácilmente.

Capítulo 1: El vadear en Visual Basic

Figura 1-6:

El Archivo
menú en
Visual
Studio.

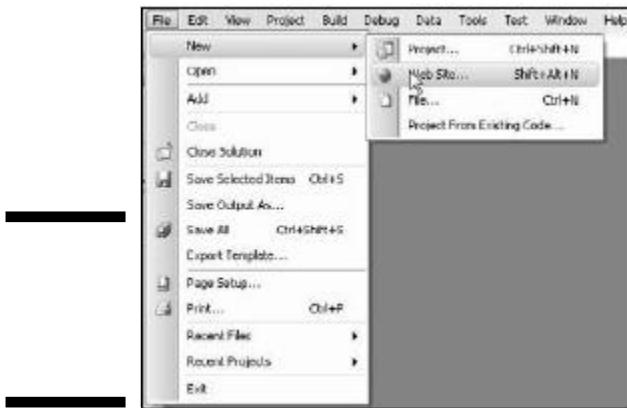
Visual Basic se vuelve móvil

La informática móvil hizo su aparición en Visual Basic 2005 y se muestra realmente ahora en Visual Studio 2008. Si usted sigue el desarrollo para dispositivos móviles, habrá dado cuenta de la gran cantidad de lanzamientos desde el equipo móvil en los últimos

pocos años. Son bien cocido en Visual Studio 2008. Pocket PC 2003, Mobile 5,0, y 6,0 móviles hacen su aparición en Visual Studio 2008, y se puede programar en VB 2008 - al igual que cualquier otro tipo de proyecto.

No doy ejemplos de estos tipos de proyectos específicos en este libro porque Puede crear una aplicación de dispositivo móvil de la misma manera que se crea un Aplicación de Windows Forms (como el programa Hello World discutido anteriormente en el capítulo). Usted debe saber que familiarizarse con el Visual Basic lenguaje tal como se presenta en este libro te pone en el camino correcto para la creación de las solicitudes para un Pocket PC. Las aplicaciones móviles de computación requieren algún prácticas especiales de programación, así que asegúrese de tomar un poco de dispositivo específico información cuando se trabaja en estos tipos de proyectos.

Escritura de rutinas para usar con otro software es más fácil con Visual Basic 2008. Puede escribir complementos para Microsoft aplicaciones de Office, incluyendo Excel y Word plantillas con código VB corriendo detrás de ellos. Estas rutinas no utilice el VBScript que usted pudo haber visto antes, una parte totalmente nueva de Office 2007 permite escribir plantillas con especial incorporado en la funcionalidad. Por ejemplo, He construido una plantilla de Word que automatiza un proceso de información pidiendo a la de usuario para un número de informe, comprobando que el número contra una base de datos de todos los informes presentados, y llenar la parte del documento en proceso con la correspondiente





VB como su tienda one-stop desarrollo

En general, Visual Studio y el Marco .NET están diseñados para ser de una sola ventana única para cualquier tipo de desarrollo en las máquinas Windows. Pero en este versión, Visual Basic 2008 también puede hacerlo todo. El lenguaje ahora se puede tocar todo el partes del marco. NET que cualquiera de los otros idiomas pueden llegar a conseguir recurriendo a la función críptica llamadas necesarias en las versiones anteriores de VB.

Las nuevas características incluidas en este libro son los siguientes:

Windows Presentation Foundation: Microsoft ha actualizado el formula para el diseño de nuevas experiencias de usuario de nuevo con aún más potencia.

El Windows Communication Foundation: Haciendo interconectado aplicaciones más potentes, el WCF es un paso avanzado que voy a tocar más adelante en el libro.

Language Integrated Query: LINQ aporta datos construye la derecha en su código con los mecanismos de consulta de nuevas colecciones de objetos.

System.XML: Si está trabajando con Extensible Markup Language, VB 2008 trae un nuevo significado a la palabra simple.

Capítulo 2

Uso de Visual Studio 2008

En este capítulo

- Repasar las herramientas de Visual Studio
- El descubrimiento de cómo el código une todos los elementos
- Personalización de las opciones
- Utilizando las herramientas de terceros

Bntes de que efectivamente puede trabajar con Visual Basic, debe conocer sus herramientas dentro y por fuera. A los efectos de este capítulo y este libro, me centraré en una sola - Visual Studio 2008. Visual Studio le permite acceder a la función de arrastrar y colocar controles que se introdujeron en versiones anteriores de Visual Basic.

A pesar de que no cubren los aspectos específicos de código de este capítulo, se cubren todos los código de generación de herramientas que proporciona Visual Studio 2008 para Visual Basic. Para ejemplo, se discute el IntelliSense nuevo y mejorado, que le puede ayudar recordar los 288.000 métodos, propiedades y eventos en el. Frame-NET trabajar, pero no cubren el propio marco.

Introducción a las herramientas de Visual Studio

Parte de la alegría de la programación en Visual Basic 2008 utiliza las herramientas que están proporcionada por Visual Studio. Desarrollo rápido de aplicaciones (RAD) es un zumbido-palabra, pero cuando Visual Basic fue desarrollado en primer lugar, se describe (entre otras cosas) la capacidad de codificar más rápido mediante la reutilización de los bits de código incorporados en el herramientas de desarrollo.

Esta posibilidad nunca ha sido más evidente de lo que es con Visual Basic 2008. A pesar de que Visual Basic es un lenguaje, y que depende de Visual Studio para su entorno, muchas herramientas RAD hacer real. En las siguientes secciones, que cubren estas herramientas. Estas herramientas son independientes del lenguaje, independiente del proyecto, y indispensable.



Tenga en cuenta que Visual Studio no es necesario realizar programas de Visual Basic. Puede, de hecho, hacer las solicitudes completas al estilo de la vieja escuela con un compilador de línea de comandos.

Además, gran parte de la documentación proporcionada por el desarrollador Microsoft Network (MSDN) Library asume una comprensión de las herramientas. El documento se refiere a las herramientas por nombre y a menudo no describe claramente ellos. Usted debe saber dónde se está trabajando para poder trabajar, por lo que el siguientes secciones le llevará en un recorrido por las herramientas de Visual Studio.



Al instalar Visual Studio, se le pidió probablemente para instalar el MSDN Library. Usted lo encontrará en una herramienta indispensable (que es lo que obtienes cuando vas al menú Ayuda, de hecho). Además, usted puede encontrar la biblioteca en línea en <http://msdn.microsoft.com/library>.

Al recorrer la vista Diseño

Cuando se inicia Visual Studio (por lo general, seleccionando su ícono en el inicio menú) y comenzar cualquier proyecto visual, se ve la vista Diseño. El Diseño View es donde la interfaz gráfica de usuario (GUI) se lleva a cabo el trabajo. En general hablando, cada vez que se trabaja con imágenes de las formas, no al código, usted es Uso de la vista Diseño. Cuando uso el término ventana del diseñador, soy refiriéndose al lugar real que haga el trabajo. La vista de diseño se refiere para el estado de la aplicación es pulg

En la vista Diseño, puede lograr lo siguiente:

Fabricación de Windows, Web y dispositivos inteligentes formas arrastrando controles directamente a la forma en un lo-que-ve-es-lo-que-hay (WYSIWYG)-tipo medio ambiente

Trabajar con bases de datos y archivos XML visualmente

Crear componentes de software de forma visual la gestión de las partes

En general, la vista de diseño es la parte principal de Visual Studio. Muchas de las otras herramientas de Visual Studio dependen de la vista de diseño, de hecho, y se desactivan cuando se utiliza otro punto de vista, como la vista Código, para trabajar en su proyecto.

Uso de las fichas Vista Diseño

Las pestanas de diseño tienen la palabra [Diseño] en el nombre de la pestaña, como se muestra en Figura 2-1, para indicar que se está utilizando la vista Diseño. Las fichas se utilizan en el diseño y las vistas de código. La pestaña gris representa los archivos que están abiertos pero no está activa. Un asterisco (*) junto al nombre de archivo significa que usted ha hecho cambios, pero aún no ha guardado el archivo.

La ficha blanca está activo y contiene el formulario editable. Cuando tenga más de un documento abierto, puede modificar sólo la forma activa. Puede arrastrar el pestañas a la izquierda y derecha para cambiar su orden. Haga clic en una ficha que da un menú desde el que puede elegir varias opciones de gestión de pantallas, como se muestra en la Figura 2-2.

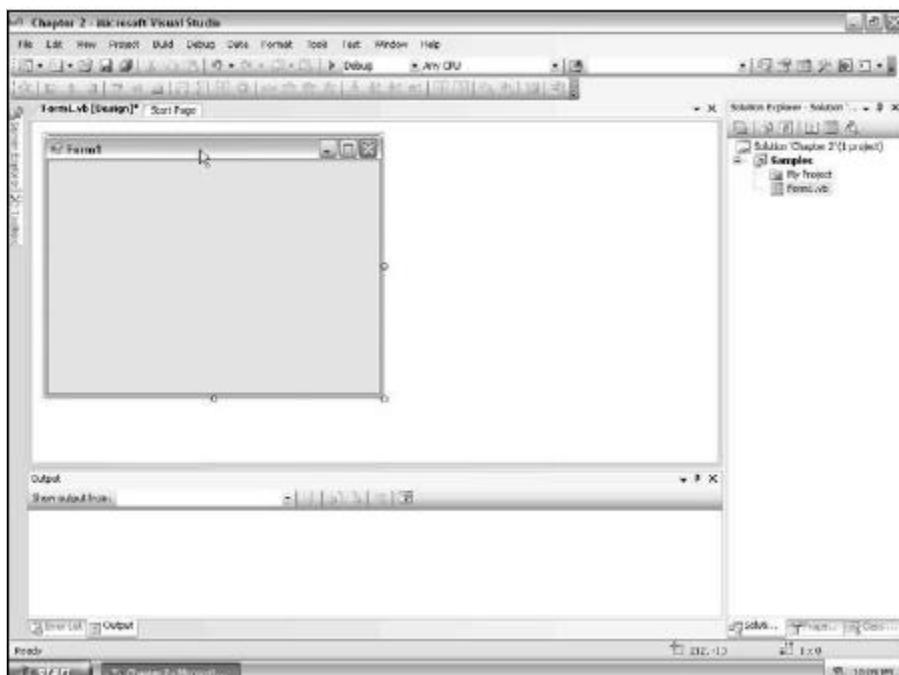


Figura 2-1:
Una forma
en
Diseño
Ver

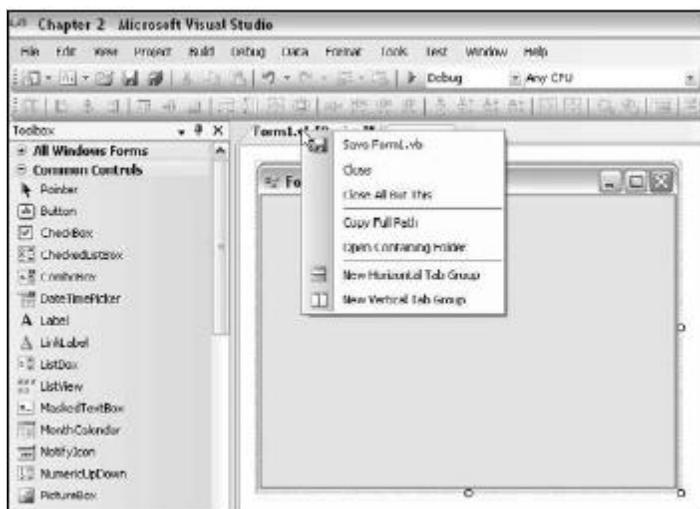


Figura 2-2:
Gerente
grupos de
pestañas.

Entender grupos de pestañas

Grupos de pestañas que sea más fácil copiar la información de una forma y en otro. Por ejemplo, usted puede tener un conjunto de páginas en la mitad superior de la pantalla y otra en la mitad inferior, y una copia de uno y pegarlo en la otro sin necesidad de cambiar las pantallas. También puede guardar y cerrar de este menú, o para obtener información, tal como la ruta actual o la carpeta que contiene.

Acceder a los controles de la caja de herramientas

Para agregar componentes de formulario, como botones y texto, para el formulario en el Diseñador ventana, simplemente arrástrelos desde el Cuadro de herramientas. La caja de herramientas, que suele presentarse en el lado izquierdo del entorno de Visual Studio y a la izquierda del diseñador ventana, es donde los componentes RAD de varios tipos de proyectos se llevan a cabo para **use hasta que los necesite**. La caja de herramientas se muestra en la Figura 2-3.

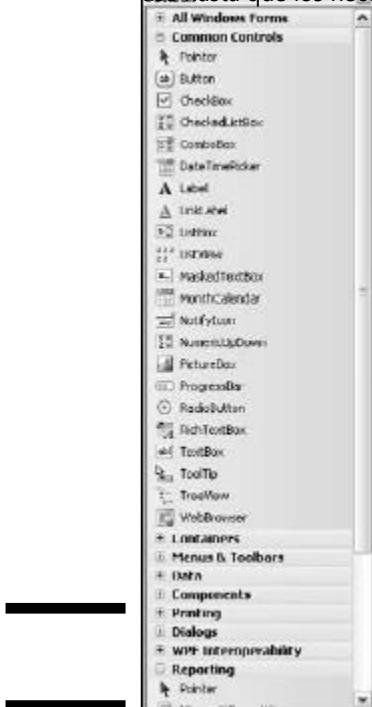


Figura 2-3:
La caja de herramientas.

Capítulo 2: Uso de Visual Studio 2008

La caja de herramientas es un proyecto sensible, lo que significa que los controles de las páginas Web y dispositivos inteligentes no aparecen cuando se escribe un proyecto de Windows Forms. Debido a que los controles no se pueden utilizar en estos tipos de proyectos, la Herramientas ni siquiera le permiten ver.

Se puede acceder a los controles de las siguientes maneras:

Haga clic y arrastre un control a un formulario, dejando caer exactamente donde usted lo desea.

Haga doble clic en el control, y aparece en la esquina superior izquierda de la forma activa.

Si pierde la caja de herramientas, o si no aparece en la pantalla, puede abrir eligiendo Vista → Caja de herramientas o presionando Ctrl + Alt + X. Haga clic en el Caja de herramientas le ofrece opciones de diseño, incluidas orden y el movimiento. Uno de los las opciones es Restablecer cuadro de herramientas, lo que lo coloca de nuevo la forma en que Microsoft tenía - una gran característica.

Los separadores grises, Data etiquetados o Windows Forms, entre otras cosas, dividir las herramientas en la caja de herramientas por categoría. Para abrir una categoría cerrada, haga clic en el divisor.

Visual Studio es extensible a causa de la Caja de Herramientas. Usted puede utilizar terceros componentes con sus programas. Por ejemplo, un reloj de control que permite al usuario configurar el tiempo se presentaba en la caja de herramientas. También puede utilizar el Caja de herramientas para almacenar piezas de uso frecuente de texto, como bloques de comentario.

En realidad se puede arrastrar un pequeño bloque de código en el cuadro de herramientas para su reutilización, tal como un bloque de comentario común o una muestra. Yo uso esta todo el tiempo cuando estoy presentando. Acabo de hacer una nueva sección, haga clic en el cuadro de herramientas y seleccionar-
ing Agregar ficha. Entonces sólo se puede seleccionar el código en la vista de Código (que cubrir en un páginas) y arrastre hacia la derecha bajo el encabezado de la sección nueva.

Cambio de los detalles con la ventana Propiedades

Después de colocar los controles en el diseñador de formularios, tendrá que editar su

son los tipos de cosas que encontrar en la ventana Propiedades. La ventana Propiedades, que se muestra en la Figura 2-4, es por lo general en el lado derecho de la pantalla. Contiene todos los valores editables asociado a un control.

Si la ventana Propiedades no está en el lado derecho de la pantalla, se puede encontrar eligiendo Ver ⇔ ventana Propiedades, o puede pulsar F4.





Figura 2-4:
La
Propiedades
ventana.

En la parte superior de la ventana Propiedades, verá el elemento cuya forma apropiada - los empates se está viendo. Puede seleccionar un control diferente, haga clic en el Diseñador ventana o seleccionando un nuevo control de la lista desplegable. En Figura 2-4, la misma forma está seleccionada.

Bajo el control seleccionado, usted encontrará algunos botones que recurren la lista o filtrar por categoría. De especial interés es el botón de rayo, que permite definir lo que el control hace bajo ciertas circunstancias - un patrón llamado eventos. Cubro los acontecimientos en profundidad a lo largo de la parte II.

La tabla que ocupa la mayor parte de la ventana Propiedades contiene la las propiedades del control. Este es un sencillo formato de tabla hash - las propiedades están a la izquierda, y los valores posibles están a la derecha. Para cambiar un valor, normalmente se puede escribir en la celda y pulse Intro. Algunas propiedades tienen un botón de puntos suspensivos (...) que se abre un formulario para simplificar la adición de detalles complejos, y
Cubro aquellos como yo discutirlas en otros capítulos.



Los botones que agrupan en la parte superior de la ventana Propiedades es una útil característica - tura. El botón de categoría es grande cuando usted apenas está comenzando, porque usted puede encontrar las propiedades en base a lo que usted necesita. La lista A-Z es mejor más adelante, cuando se sabe exactamente lo que la propiedad que busca.

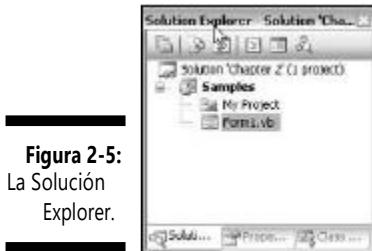
La parte inferior de la ventana Propiedades tiene una breve descripción de la propiedad que se está editando. Esta información se encuentra a la derecha de la documentación y una característica muy útil.

La organización de su proyecto con el Explorador de soluciones

Soluciones y proyectos de sostener formas y componentes como archivos en carpetas. En de datos, soluciones y proyectos están representados por carpetas en Visual Studio Proyectos de directorio de la carpeta Mis Documentos. El Explorador de soluciones es Herramienta de Visual Studio que le permite administrar los archivos que componen su proyecto.

Si usted preve sus proyectos como las carpetas, se puede imaginar que lo haría grupo como carpetas juntas en una carpeta de un nivel superior, ¿no? Eso es lo que Soluciones hacen. Ambos están física y lógicamente eso exactamente - carpetas llenas de proyectos.

En la Figura 2-5, puede ver los archivos importantes en su proyecto, y un montón de botones de arriba para ayudar a gestionar ellos.



Para abrir un archivo, haga doble clic en el ícono del archivo o el nombre. Para cambiar el nombre, copiar o borrar un archivo, haga clic en el archivo y elegir la acción deseada en el contexto menú sensible que aparece. En el Explorador de soluciones, también puede hacer nuevos carpetas y mover archivos en ellos, o haga clic en el proyecto de hacer una nueva forma o el soporte de archivos.

Los botones de arriba de los propios archivos son la parte más importante de la Solution Explorer. Son los siguientes, de izquierda a derecha:

Propiedades: Se abre la ventana Propiedades.

Mostrar todos los archivos: Muestra los archivos ocultos. Esto es más significativo en VB 2008 que antes. Aún más archivos están ocultos a la vista normal.

Actualizar: Comprueba la carpeta de soluciones para los nuevos archivos que puede haber sido añadido por otra herramienta. Este botón es muy útil cuando se está utilizando herramientas de terceros.

Ver Código: Abre el archivo seleccionado en la vista Código.

Ver diseñador: Abre el archivo seleccionado en la vista Diseño.

Ver diagrama de clases: Abre el archivo seleccionado en la vista Diagrama.



Visual Studio almacena una gran cantidad de archivos que guardan los metadatos sobre el proyecto en el

Mi proyecto carpeta en el Explorador de soluciones. Los metadatos son información sobre los datos - en este caso, información adicional sobre su proyecto. Si hace clic en Mostrar todo Archivos de botón en el Explorador de soluciones y ampliar mi proyecto, verá nada menos de siete archivos con información sobre su proyecto. Usted no tendrá que editar estos archivos con frecuencia, pero si usted no puede encontrar algo, es posible que desee comprobar hacia fuera.

Acceso a recursos externos con el Explorador de servidores

Salir a la calle de su proyecto para acceder a los recursos necesarios es uno de los más características comunes que no está soportado por la mayoría de entornos de desarrollo. Todo eso cambió con Visual Studio 2002 y ha mejorado desde entonces con la adición del Explorador de servidores, que se muestra en la Figura 2-6. Puede abrir Explorador de servidores pulsando Ctrl + Alt + S o seleccionándolo en el menú View. Ahora, para llegar a los servidores que proporcionan los servicios necesarios es más fácil que nunca.

El Explorador de servidores es una de las herramientas más dinámicas en Visual Studio ambiente. En la Figura 2-6, estoy usando mi máquina de desarrollo virtual, XP. Lo que se ve en el Explorador de servidores depende de la configuración local.



Figura 2-6
El servidor Explorer.

Capítulo 2: Uso de Visual Studio 2008

El nodo Servidores en el explorador aparece en algunas ediciones de Visual Studio. Al cierre de esta edición, la edición profesional era uno de ellos. Además, el Team System edición sin duda lo ha hecho, y usted puede conseguir que la edición de la web de Microsoft sitio en una versión de prueba.

Explorador de servidores le permite acceder a distancia (o local) los recursos de un hombre-gestión y la perspectiva del código. La vista de árbol en el interior del Explorador de servidores puede mostrar muchos servidores, y debajo de cada servidor son los bits funcionales que usted tiene acceso.

La mayoría de los desarrolladores están buscando una ventanilla única para el desarrollo, con mayor frecuencia, que incluye la necesidad de la capacidad de administrar servidor de desarrollo (s) y buscar en bases de datos. El Explorador de servidores maneja ambos, pero los nuevos datos Ventana Orígenes es aún mejor, y lo cubren en la siguiente sección.

Los servicios disponibles para usted en su Explorador de servidores dependen de su entorno-ambiente, pero aquí es una breve descripción de algunos de los servicios que son común:

Crystal Reports Servicios: Crystal es una herramienta de información de terceros que se incluido con Visual Studio. Los servicios incluyen Crystal Enterprise, Informe Servidor de aplicaciones y servicios Web.

Registros de eventos: Esto representa la normalidad antiguo Windows NT al estilo de evento registros que se puede acceder desde el Panel de Control. Los registros están disponibles tanto programación en .NET y para la gestión del Explorador de servidores.

Clases de manejo: Clases de administración de representar a Windows Management Instrumentation (WMI) clases como adaptadores de red, impresoras, y servicios, todo lo cual se puede tocar programación del mando a distancia máquinas.

Eventos de gestión: Le permiten registrar eventos WMI, como cambios en la red y alertas de hardware.

Colas de mensajes: Colas de mensajes son una manera de ayudar a controlar el número de solicitudes a una aplicación muy grande. Las colas individuales se hacen disponible aquí.

Contadores de rendimiento: Este es el acceso al Monitor de rendimiento de Windows del sistema operativo. Cada contador está disponible tanto para la visualización y programación.

ntrol están disponibles aquí. Usted
Puede detener e iniciar los servicios Web, por ejemplo.

Conexiones de datos: Esta es una categoría especial que no está directamente
relacionada

a los servidores que se muestran. Las conexiones de datos se refieren a conexiones
que
han creado a través del tiempo en la instancia de Visual Studio, y son recor-
numerados para su conveniencia.



En la vista Diseño, en realidad se puede arrastrar un registro de eventos o contadores de rendimiento en la forma de escribir código para modificar sus propiedades. Aparte de estos proporciones gramaticales, el Explorador de servidores prevé que las ventanillas únicas la gestión de compras.

El nodo Conexiones de datos le permite conectar la aplicación a un conjunto de datos fuente. Haga clic en el nodo Conexiones de datos y seleccione Agregar conexión a añadir una nueva conexión. Después de seleccionar un proveedor de datos (SQL Server Tomé), el cuadro de diálogo Agregar conexión, como se muestra en la Figura 2-7.



Visual Studio instala SQL Server Express como parte del paquete de programas proporcionado. Si usted no tiene acceso a otras bases de datos, siempre se puede utilizar esa uno. Aparecerá como Machine / Sq|Express en el cuadro de diálogo Agregar conexión.

En la Figura 2-7, seleccioné mi máquina local (XP), la instancia de SQL Express, y la base de datos Northwind. Usted probablemente no tiene la base de datos Northwind instalado por defecto. Usted puede obtener desde [www.microsoft.com / downloads](http://www.microsoft.com/downloads) o www.vbfdummies.net.

Esta selección se conecta el proyecto con una base de datos, que a su vez le permite utilizar la ventana Orígenes de datos, la gestión de los objetos de datos dentro de los datos-base, y editar los datos directamente. Cuando haya terminado de agregar los valores a la conexión, haga clic en el botón Probar conexión para asegurarse de que su proyecto Puede obtener la base de datos que ha seleccionado.



Figura 2-7:

El complemento

Conexión

Cuadro de diálogo.

Dinámicamente edición de datos con la ventana Orígenes de datos

Al iniciar un nuevo proyecto, la ventana Orígenes de datos dice "Su proyecto actualmente no tiene fuentes de datos asociados con él. "(Si no puede ver los datos Ventana Orígenes, seleccione Datos \Rightarrow datos muestran Fuentes con un proyecto abierto, o pulse Mayús + Alt + D) Para mantener los datos en el mundo. NET, como con cualquier otro medio ambiente, se debe conectar la aplicación a un origen de datos. Los Datos de Carácter Ventana Orígenes es la principal forma de hacerlo.

Para conectarse a un origen de datos (como un archivo de base de datos o XML), siga estos pasos:

1. Haga clic en el botón Agregar un Nuevo origen de datos.

Si lo hace así, se inicia el Asistente para la configuración de orígenes de datos.

2. Haga clic en el botón Siguiente.

El asistente muestra las opciones de fuente de datos. Usted puede seleccionar la base de datos opción de utilizar un servidor SQL Server o base de datos Oracle, la opción de Servicio para conectarse a un servicio Web XML, o la opción de objetos para conectarse a una capa de acceso de datos.

3. Seleccione la conexión a Neptuno que usted hizo en el Acceso a "recursos externos con el Explorador de servidores "y, a continuación, haga clic en el botón Siguiente (se muestra en la Figura 2-8).

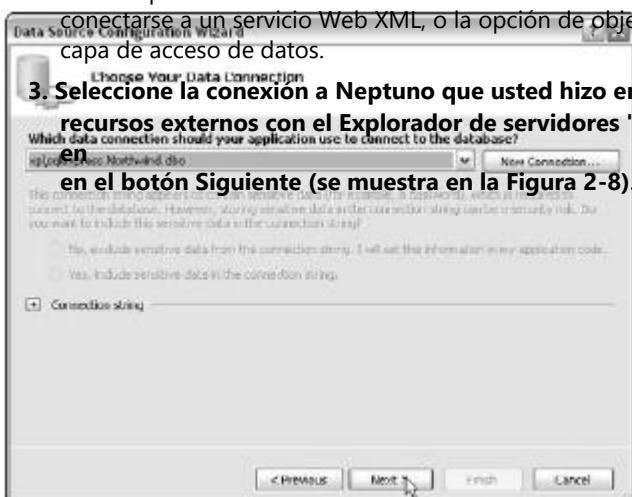


Figura 2-8:
Los Datos
de Carácter
Fuente
Configuración
Wizard

4. Acepte el nombre de la conexión por defecto (probablemente NorthwindConnection) y haga clic en el botón Siguiente.

Se le da la opción de qué objetos debe incluir en su base de datos. Cubro conjuntos de datos en el capítulo 15.

5. Por el momento, seleccionar la vista por primera vez en la lista de verificación Vistas: Lista alfabética de los productos.

6. Haga clic en el botón Finalizar.

Como se muestra en la Figura 2-9, cada una de las columnas en la vista que ha seleccionado aparece como los tipos de objetos editables que puedan representarlos. Ahora puede arrastrar a la vista de diseño para crear un control enlazado a datos.

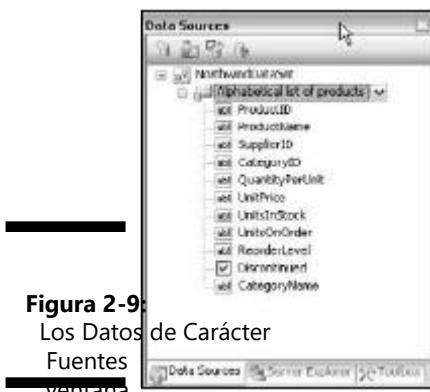


Figura 2-9:
Los Datos de Carácter
Fuentes ventanas.

Sólo por diversión, arrastre el campo QuantityPerUnit en el formulario en blanco que fue creada por usted cuando se inició el proyecto. Visual Studio creará un grupo de los datos de tuberías para usted y luego agregar el campo y una etiqueta al formulario. Uso de la Ventana Orígenes de datos como este le proporciona la funcionalidad fantástica para desarrollar rápidamente aplicaciones de datos.

En la Figura 2-10, tengo una foto de mi pantalla mientras arrastra el campo en el formulario. Se puede ver en la parte inferior vista de diseño, donde el fondo es gris, cinco componentes que se han añadido a Form1. Los componentes son funcionales elementos del cuadro de herramientas no, las cosas que son visibles en la pantalla. Los Datos de Carácter Fuentes ventanas hace que todos esos componentes y agrega el control a la Juego, en función del campo que se mudó. Ese es el punto de estas herramientas RAD.

Francamente, la Vista de diagrama es una herramienta muy sofisticada, que no cubren en este libro. Permite a los arquitectos de la empresa para construir software basado en componentes tomando bloques enteros de código y moviéndose en un entorno gráfico.

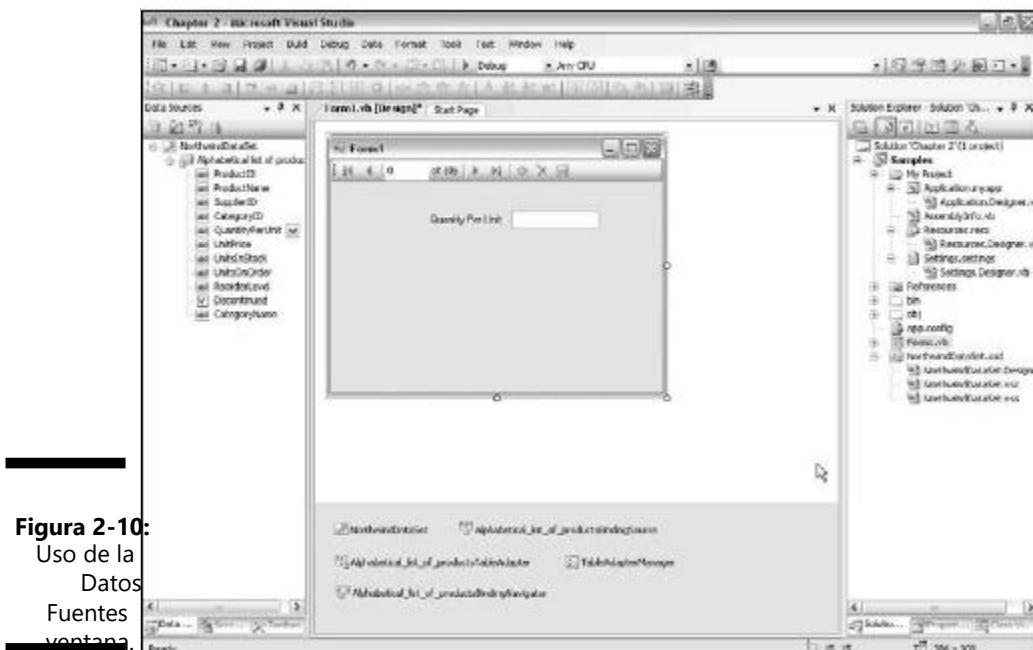


Figura 2-10:
Uso de la
Datos
Fuentes
ventana.

De vez en cuando, cuando se ha escrito algo en Visual Basic, vaya adelante y cargar la Vista de diagrama. Usted puede entrar en Vista de diagrama, haga Ver clic en un archivo en el Explorador de soluciones y seleccione en el Diseñador de la menú contextual. Se va a crear un nuevo archivo llamado ClassDiagram1.cd en el Explorador de soluciones y le mostrará la tubería detrás del software que hemos construido. Puede ser una experiencia educativa - probarlo!

Mover una ventana de herramientas

Se puede personalizar el entorno de Visual Studio para que sea más fácil para usted para trabajar, todas las herramientas, las ventanas y las vistas son parte de un sistema integrado

Development Environment (IDE), que proporciona una ubicación de inicio. Esto hace organización de su espacio de desarrollo personal mucho más fácil.

Muy a menudo, usted tendrá que mover una ventana de herramientas en torno a decirlo de una manera más

lugar conveniente. Se puede visualizar una ventana de la herramienta de las siguientes maneras:

Flotante: Una ventana flotante es muy móvil - se puede arrastrar por el asa para colocarla en cualquier lugar que desee.

Acoplable: Cuando se arrastra una ventana acoplable, sin embargo, se le da la opción por Visual Studio para acoplar la ventana. Esto se demuestra en Figura 2-11, donde estoy arrastrando la ventana Explorador de soluciones en torno a en la vista Diseño.

Organización por fichas: Usted también tiene la opción de arrastrar la ventana a la centro y la han convertido en una pestaña en la parte superior de la ventana de la vista, como los archivos de página Form1.vb y Puesta muestra en la Figura 2-11.

Usted tiene cinco opciones para acoplar la ventana. Si arrastra la ventana sobre el arriba, abajo, izquierda o flecha derecha, se acoplará a ese lado. Cuando una ventana está acoplado, tiene una chincheta que usted puede fijar o liberar. Cuando se inmovilizó, se queda en el lado, el diseño en movimiento Ver más. Cuando desprendida, se desliza fuera de la camino hacia el lado que está anclado a. Es una característica brillante.



Si arrastra la ventana hasta el centro de la estrella de cuatro puntas, la ventana se convierte en una ficha en la otra ventana central - mucho más fácil que el anterior haga clic tres veces que era muy difícil de usar. El triple clic con el botón todavía funciona, pero no lo necesitamos más!

Trabajar con código

De los programas que cree con Visual Studio, la mayor parte de lo que quiere el usuario ver son los controles y las formas. Pero usted pasará la mayor parte de su tiempo de trabajo con el código. Afortunadamente, Visual Studio tiene un montón de herramientas para ayudarle a escribir código.

Llegar a vista de código

Vista Código, al igual que en Vista Diseño o Vista de diagrama, es sólo otra manera de mirar a una archivo en el Explorador de soluciones. Para llegar a la vista de código, usted tiene varias opciones:

Puede hacer clic en un formulario en el Explorador de soluciones o en la Vista de Diseño y seleccione Ver código en el menú contextual.

Puede hacer clic en el botón Ver código en el Explorador de soluciones.

Puede hacer doble clic en un objeto en un formulario.

Cuando se trabaja en la vista Código, la mayoría de las ventanas de herramientas se convertirá en inactivos. La caja de herramientas y ventanas de propiedades, por ejemplo, tienen poco que nada disponible debido a que sus características se han diseñado para su uso con el diseño Ver en vez de la vista Código. Esto es así por diseño, para mantener el código de su camino en la construcción de la lógica de negocio.

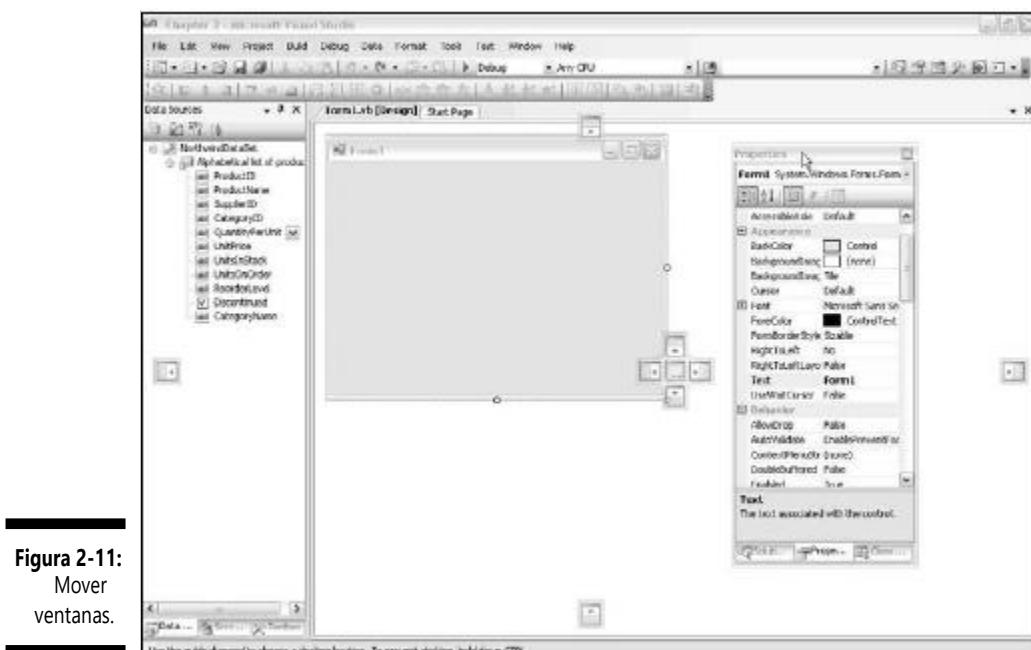


Figura 2-11:
Mover
ventanas.

Utilizar IntelliSense

El problema con el uso de la vista de código es que usted necesita saber qué escribir. Bienvenido a la programación! Para ayudarle a empezar, puede utilizar algunos muy productivos basados en códigos herramientas, como IntelliSense y etiquetas inteligentes.

El resto del libro es sobre la lengua, pero quiero darle algo para empezar para que pueda ver lo bien estas herramientas son. ¿Cuándo se está trabajando con el código como el objetivo principal, las herramientas deben ser una segunda naturaleza. Comience con estos pasos:

- 1. En el proyecto predeterminado, haga doble clic en el Form1 en blanco para ir a Ver código.**
- 2. Comience a escribir My. en el método Form1_Load. (Sólo tiene que pulsar M en el teclado, y se puede ver lo que me refiero).**

Un especial menú contextual (el menú IntelliSense) aparece cuando empezar a escribir, como se muestra en la Figura 2-12. Este menú muestra el código que está disponible para usted. Adivina lo que necesita sobre la base de lo que eres haciendo - en este caso, no tiene mucho que trabajar con lo que comenzó con el método de MID.

Finalizar 'Mi'. Mecanografía y obtendrá todos los métodos disponibles, las propiedades, y los acontecimientos en el objeto My. Es una gran manera de recordar la mejor manera para hacer las cosas.

Puede seguir utilizando los menús de IntelliSense mientras Visual Studio cree hay más tipos después del objeto seleccionado. Por ejemplo, haga doble clic Aplicación en el menú contextual y escriba. (Punto). Usted ve otro menú IntelliSense.

Usando este método, usted puede tener acceso a todo en el lenguaje Visual Basic. Incluso cuando usted escribe su propio código reutilizable, IntelliSense lo recogerá para este especial menú contextual. Esto hace que sea mucho más fácil trabajar en la .NET Framework con Visual Basic.

Lectura de la documentación

Es raro que un autor tiene que escribir acerca de cómo leer la documentación, pero en el caso de Visual Studio, hay tanto poder en la documentación modelo que se merece un poco de espacio. El uso más simple de la documentación requiere poco más que hacer clic en o en el objeto que Tienes preguntas acerca de F1 y presionando para poner en marcha la ayuda contextual.

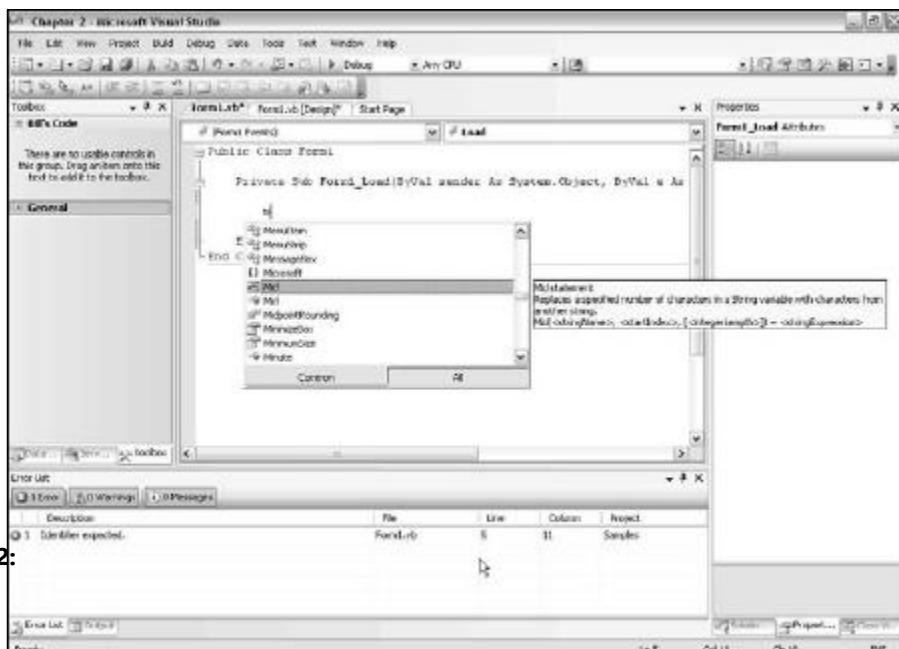


Figura 2-12:
Uso
IntelliSense.

Capítulo 2: Uso de Visual Studio 2008

Por ejemplo, en una aplicación, haga clic en algún lugar de la forma (asegúrese de que hace clic en el formulario y no un objeto en el formulario) y pulse F1. El Microsoft Explorador del documento se inicia con la documentación objeto Form cargado.

El Explorador de documentos tiene un sofisticado conjunto de herramientas, en su mayoría representados en la barra de herramientas de explorador de documentos, que se muestra en la figura 2-13. Las herramientas que se encuentran hay que dar varias maneras de acceder a la documentación, de la siguiente manera:

Figura 2-13:

La
Documento
Explorador
barra de
herramien
tas.

Primera sección La barra de herramientas tiene botones de navegación, un botón Actualizar y, a fuente mantenimiento tamaño.

Cómo se hace El botón tiene preguntas preestablecidas que se relacionan con el seleccionado tema, y puede ayudar con consultas generales sobre ciertos tipos de desarrollo si usted está atascado, le daría una oportunidad.

Al hacer clic en el botón de búsqueda permite la búsqueda frase. El botón de índice y el botón de Contenidos permite navegar a través del índice o TOC (tabla de contenido) de la documentación. Puede guardar los favoritos en la ayuda Favoritos exactamente igual que como en el Explorador de Internet. La doble flecha está práctica - se sincroniza el panel de contenidos con la página que está actualmente viendo.

El Haz una pregunta botón te lleva directamente a los grupos de noticias NNTP (Utilizando un visor de Web-based) organizado por Microsoft, donde se puede pedir preguntas que son respondidas por el MVP de Microsoft, autores y otros expertos. Si no va a participar en la comunidad de usuarios, por favor hacerlo - véase la Parte V para más información.

Usted tiene todas estas opciones, y son sólo una parte de la documentación en Visual Studio. IntelliSense muestra la información de la documentación del usuario cuando se coloca el cursor del ratón sobre una pieza de código. Las Propiedades ventana muestra la documentación de una propiedad cuando está seleccionado. Everywhere se mire, la ayuda está allí!

Y no se olvide de las herramientas en línea que proporciona Microsoft. Seleccione Ayuda →

Asistencia técnica para acceder a una gran cantidad de información disponible en la Web,



Personalizar con opciones

Las opciones disponibles en Visual Studio son increíbles. Para empezar, al igual que con muchos otras aplicaciones de Windows, las barras de herramientas y los menús son completamente editable. Seleccione Herramientas \Rightarrow Personalizar para abrir el cuadro de diálogo Personalizar. Haga clic en el Ficha Comandos para obtener una lista de todos los comandos disponibles en Visual Studio. Para agregar

un botón de comando a una barra de herramientas, arrastre un comando de la lista para la barra de herramientas.

Por ejemplo, como se muestra en la Figura 2-14, arrastré el ícono Build a una barra de herramientas para que pueda acceder a ella en cualquier momento. Es una característica fantástica para el diseño de su propio ambiente personalizado.

La personalización pareja está disponible en Herramientas Opciones \Rightarrow .

El cuadro de diálogo Opciones tiene muchas opciones que están en una vista de árbol de la izquierda lado de la caja de diálogo. Más de 100 pantallas de opciones están disponibles para la edición en categorías tales como medio ambiente, control de código fuente, herramientas de base de datos y

Desarrollador de Windows Forms.



Figura 2-14:

La

Personalizar

Cuadro de diálogo.

Capítulo 2: Uso de Visual Studio 2008

Las variables de entorno general en el cuadro de diálogo Opciones se muestran en la Figura 2-15. No hablo de todas las opciones disponibles en el cuadro de diálogo Opciones porque hay montones de ellos, pero tardará diez minutos para mirar estos opciones y ver cómo pueden ayudar a desarrollar programas. Es posible que no vea todo lo que hacen ahora, pero cuando usted los necesita, usted sabrá que están ahí.

Figura 2-15:
Las opciones de
Cuadro de diálogo.

La restauración de archivos Asociaciones botón cambia el diseño de las ventanas de herramientas cajas y paneles a los valores predeterminados de Microsoft de su perfil. Es muy bueno para usar si
¿Ha cambiado todo a su alrededor, como se describe en la sección anterior "Mover un Ventana de herramientas. "

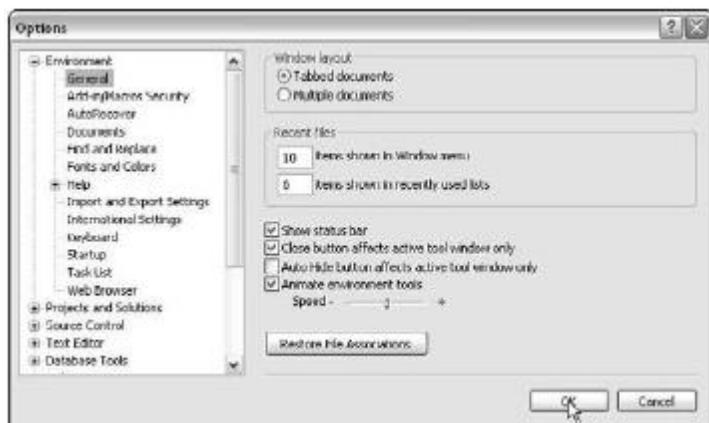
El aumento de la eficiencia con Herramientas de terceros

Visual Studio ofrece una estructura para desarrolladores de terceros para escribir a cuestas programas llamados complementos. Los complementos te dan, el programador, más flexibilidad y la funcionalidad de Visual Studio. La mayoría de las empresas realmente utilizar Visual Studio para desarrollar estos complementos. Yo no cubren terceros add-ins en este libro, pero hace un gran tema para investigaciones posteriores.

Microsoft ha incluido una serie de complementos con Visual Studio 2008 instalamiento. Mientras que Microsoft proporciona una herramienta notable, que no proporciona todos-

uncionalidad adicional en la forma de los complementos.

41



Tome Dotfuscator, por ejemplo. Por definición, .NET son auto-documentando. Cualquiera puede tener una aplicación escrita en .NET y mirar la estructura básica del código con poco esfuerzo. Con las herramientas disponibles en el Internet, puede realizar ingeniería inversa este código en Visual Basic. (Apuesto a que no lo hicieron te dicen que en el marketing.) Estoy seguro de que puedo imaginar que este hecho altera unas pocas personas.

Ingrese Dotfuscator. Esta aplicación add-in, que está completamente integrado en Visual Studio, ofrece la posibilidad de confundir compilados .NET, por lo que es muy difícil de realizar ingeniería inversa del código a Visual Basic - así el nombre del producto. Para ejecutar Dotfuscator, seleccione Herramientas → Comunidad Dotfuscator

Edición y aceptar los términos. A continuación verá la ventana de Dotfuscator defecto, como se muestra en la figura 2-16.

Esta aplicación le permite especificar un proyecto terminado y realizar algunas magia. Dotfuscator ejecución velocidades, se reduce el tamaño del paquete, y protege su propiedad intelectual. En mi opinión, Dotfuscator era una buena opción para Microsoft para incluir con Visual Studio, y se muestra el poder de terceros add-ins.

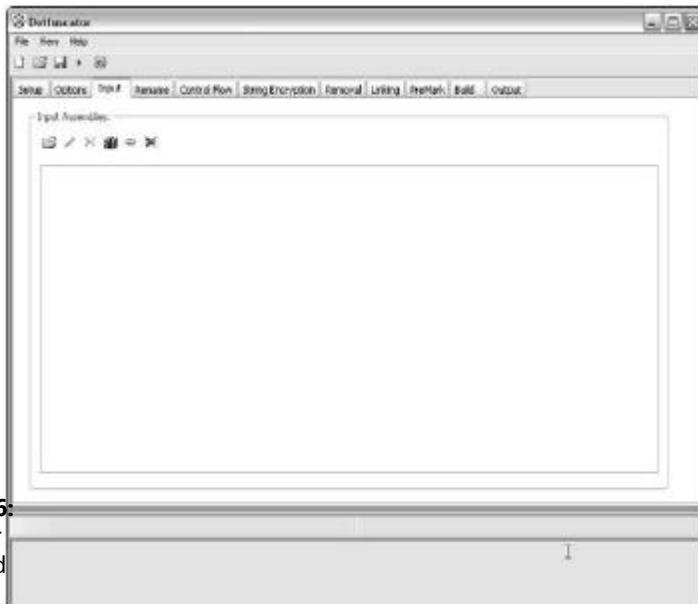


Figura 2-16:
Dotfuscator
Comunidad
Edition.

Capítulo 3

Diseño de aplicaciones en VB 2008

En este capítulo

Profundizando en el Marco. NET

La planificación de su trabajo

El trabajo de su plan de

Describir software

Bntes a descubrir la diversidad de todos los proyectos que se pueden construir, el facilidad de Visual Basic, y el poder del Marco. NET, debe saber cómo diseñar software. Por diseño, me refiero a la planificación de su trabajo. La clasic comparación es que el software es como construir una casa. ¿Podría usted contratar a un contratista que se va a construir su casa sin planos? De Por supuesto que no. Del mismo modo, no debe esperar a ser capaz de escribir software con los diseños.

En este capítulo diseño único, muestro cómo el Marco. NET hace que sea fácil para usted para encontrar la manera de construir software desde el principio. Yo también muestran que la estructura del Marco. NET y describir cómo funciona con Visual Studio y Visual Basic.

A continuación, descubrir el fundamento sobre el cual se construye el marco. NET. Yo explicar los conceptos abstractos que conforman el Marco. NET tan fácil de usar y algunos ejemplos concretos de cómo se utiliza.

Planificación para la construcción de software también se trata en este capítulo. Lo creas o no, formas aceptadas y estructuradas existen para diseñar software. Después de esta estructura Es una gran manera de conseguir su diseño previsto en el papel. En este capítulo, se discute cómo para diseñar el software que se construye en la Parte II.

Finalmente, cubrir la forma de describir el software desde la perspectiva de la lectura y escribir los diseños. Cuando termine de leer este capítulo, usted puede planear una proyecto de software real.

Haciendo uso de software simple del Marco. .NET

.NET como un concepto es una biblioteca de software conectado desarrollado por Microsoft que conecta a las personas y los sistemas y dispositivos que utilizan la información que necesitan. El Framework. NET es el entorno de desarrollo que hace posible todo esto desde la perspectiva de Visual Basic.

Visual Basic es sólo una pieza del Marco. .NET. Como se muestra en la Figura 3-1, Visual Basic sólo se utiliza para escribir el cliente, el servidor y software de conectividad que hace que todo suceda.

Aplicaciones bien diseñadas incluyen las siguientes capas:

Los clientes en el mundo. .NET incluyen dispositivos como teléfonos móviles y PDAs, PCs con Windows, un Smart Client como Microsoft Office, o un navegador Web en cualquier sistema operativo.

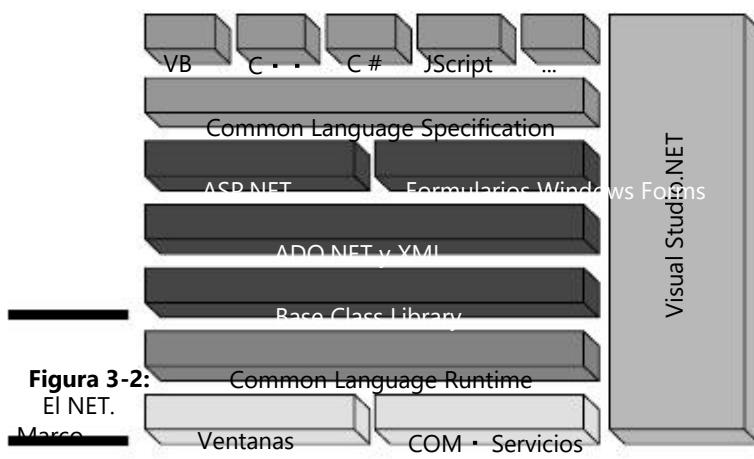
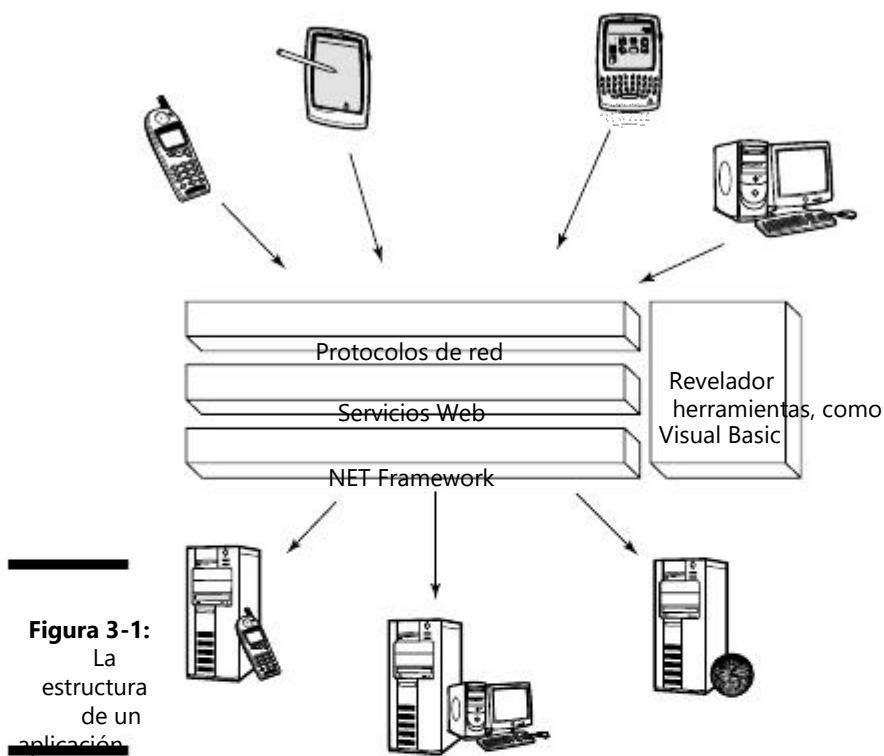
Los servidores de. .NET generalmente se ejecuta Windows Server y SQL Server. El servidor plataforma es mucho menos flexible que la plataforma de cliente en el mundo. .NET. Otras opciones están disponibles, también, como la base de datos Oracle. A veces, como servidores BizTalk o SharePoint Services se utilizan. Generalmente, los servidores prestación de servicios. Tiene sentido!

En el medio están los servicios Web XML o conectividad a otro. XML Web servicios representan una estrategia multi-plataforma para obtener información de los servidores a los clientes, los clientes a otros clientes, o incluso entre los propios servicios.

Las herramientas de desarrollo representados en la Figura 3-1 son Visual Basic y Visual Studio. Visual Basic es el lenguaje, y Visual Studio es la herramienta. La tercera pieza del rompecabezas es el plan - el tipo de proyecto. El plan es el foco de este capítulo.

En la parte superior de la figura 3-2, se puede ver todas las estructuras que componen el desarrollo de software representado en la figura 3-1. El enfoque de este libro, VB (Visual Basic) es la manera en la esquina superior izquierda.

Cómo VB interactúa con las otras partes del diagrama es muy importante, también - eso es lo que el software hace. El programa utilizará los servicios prestados en el Marco. .NET a través de las herramientas de la lengua. Esta interacción es la clave de todo - es donde tiene que enfocar su planificación. La forma de hacerlo tomar ventaja de la interacción de las piezas del marco de trabajo? Eso es lo que para diseñar.



Antes de llegar a este plan, usted necesita saber lo que el Marco. NET puede hacer por usted como parte de Visual Basic. Una de las partes más difíciles de la planificación de software es saber lo que su programa tiene que hacer, y lo que se hace por usted los servicios en el marco. NET. En las siguientes secciones, le explico lo que del Marco. NET puede hacer por usted.

Para conseguir que el sistema operativo

La función principal del núcleo del Marco. NET es la clase base Biblioteca (BCL), que proporciona acceso a las funciones de la operación sistema y los servicios como los gráficos y bases de datos. Una gran cantidad de piezas auxiliares de el marco cubrir otras cosas, pero es llegar al sistema operativo de la venta grande para Visual Basic. ¿Por qué? Programadores de Visual Basic solía tener que saltar a través de aros para llegar a estos servicios.



La ruta de acceso al sistema operativo Windows de Visual Basic es largo y viento ing. El Mi objetivo es el acceso directo a la ruta de acceso. También es un ejemplo fantástico de como el Marco. NET puede ayudarle, el programador VB, hacer el trabajo.

El Mi objetivo le da acceso a la computadora a través de los ojos de la operamiento del sistema. El objeto My.Computer permite que su programa fácilmente interactuar con todas las piezas de la computadora, tales como

- Teclado y ratón
- Impresoras
- De audio y video
- Portapapeles
- Reloj
- Sistema de archivos

Algunas de las tareas más comunes que se pueden realizar fácilmente con el objeto My incluir

- Carga y descarga de un archivo
- Leyendo, escribiendo, y limpiando el Portapapeles
- Controlar la conexión del equipo a Internet

Estas tareas son bastante difícil en la mayoría de los lenguajes de negocios, pero se hacen más fácil en Visual Basic 2008 utilizando el Marco. NET.

Capítulo 3: Diseño de aplicaciones en VB 2008

Por otra parte, se encuentran dos objetos más primarios en la mi colección. La Objeto My.Application ayuda a los programas de información sobre el medio ambiente en el que se están ejecutando. El objeto My.User le ayuda a recopilar información sobre el usuario que ha iniciado sesión en el equipo, como su nombre y e-mail.

Busque el ícono Mi objeto a través del libro de consejos sobre cómo hacer su desarrollo más fácil con el objeto My.

La integración de servidores y servicios

En la Figura 3-2, cuatro cajas se encuentran en la parte media - dos son la interfaz de usuario

orientado y dos orientado al servicio. ASP.NET y Windows Forms son fáciles de interfaz orientada, y cubrir aquellos en la siguiente sección. ADO.NET y otros componentes de la BCL son importantes, en parte debido a que ayudará a integrar parrilla servidores como bases de datos y servicios como BizTalk.

ADO.NET cubre el servidor principal que se tendrá que integrar - a los datos servidor. Bases de datos, como Microsoft SQL Server 2005, representan la mayor parte tipo común de interacción para programas de Visual Basic. Programas de negocios tienden a necesitar para conseguir información de usuario a usuario, y es que la información a menudo se almacenan en bases de datos.

ADO.NET le permite tomar los datos de la base de datos, que muestran a los usuarios, aceptar su manipulación de los datos, y actualizar la base de datos sin una gran cantidad de código de cableado. Con esto quiero decir que usted puede preocuparse por la construcción de la empresa lógica de la aplicación, y no con la forma en la conexión de base de datos sí funciona. Cubro las conexiones de base de datos y ADO.NET en el capítulo 15.

Puede encontrar más tipos de servidores que los servidores de bases de datos. Lo que le permite conectar a los servidores sin necesidad de escribir código tubería es parte del trabajo de la

Las bibliotecas de clase base. Algunas de las tuberías prebuilt proporcionado por el BCL incluye lo siguiente:

Servicios empresariales: Herramientas necesarias por aplicaciones muy grandes, como transacciones y la activación, provistos por el Service Component.

Input Output (IO): El acceso al sistema de archivos, unidades de disco, y el almacenaje en servidores de varios sistemas operativos. Cubro IO en el capítulo 16.

Mensajería: El uso del servicio de colas en Windows. (No instantánea messag- ción, debo añadir - que es diferente) Este tipo de mensajes se utiliza

ta sobre los datos y el usuario
interacción.



Gestión: El acceso a la Instrumentación de administración de Windows servicios, que le dan una idea del estado del servidor.

Net: La red e Internet. Todos los sitios Web y servidores de correo electrónico son accesibles gracias a la colección de red en el BCL. Cubro neto en el capítulo 17.

Dibujo: Hacer arte decente es dura, y el BCL le da ese poder mediante la simplificación del conjunto de herramientas de Windows gráficos conocidos como GDI +. Cubro dibujo en el capítulo 18.

Interacción con el usuario

Las otras dos cajas en la sección media de la Figura 3-2 son ASP.NET y Windows Forms. ASP.NET y Windows Forms ayudará más que cualquier otro de éstos., permitiendo la interacción con el usuario

He mencionado anteriormente que encuentre tres partes de desarrollo con Visual Básica. El primero es el idioma, Visual Basic en sí. El segundo es la herramienta Visual Studio 2008, que analizo en el capítulo 2.

La tercera y última pieza de este rompecabezas es el tipo de proyecto o de la plataforma, y que es controlada por los formularios de ASP.NET o Windows. ASP.NET tiene todos los bits para las páginas Web, Web móvil, y servicios Web XML. Windows Forms tiene todo los bits para las aplicaciones de Windows, aplicaciones de consola y el dispositivo inteligente aplicaciones.

Para obtener más información acerca de la interacción con el usuario, véanse los capítulos 4 y 5.

Comparando Conceptos Abstractos con el mundo real

Usted pasa una gran cantidad de su tiempo leyendo sobre abstracta y conceptos concretos cuando se trabaja en .NET. Aunque no es tan común en este libro, al buscar artículos o documentos en la Web, usted va a leer mucho acerca de las clases y objetos.

Clases

Clases son una construcción filosófica. Ellos son vasos que pueden ser llenados con cosas. Son marcos, esqueletos espera de su carne. Son una serie de macetas con tierra, en espera de las plantas. Ellos son conceptuales, no concreto.

Capítulo 3: Diseño de aplicaciones en VB 2008

Una clase es una definición de una cosa, con una lista de lo que se puede hacer para ella, es lo que sabe sobre él, y lo que puede hacer. Sin ser "instancia" en un "Instancia de la clase", una clase es sólo una serie de corrales para los animales que no están allí.

Objetos

Objetos son los elementos concretos que existen en su aplicación. Ellos son lo que las clases

Ilegar a ser cuando sean grandes. Al crear una instancia de una clase, la clase va y se viste y se convierte en un objeto.

Al definir algo en una aplicación, usted está haciendo una clase. "Un House tiene un color y una página inicial "sería un ejemplo de una clase. Lo es sólo la definición. Una instancia de la clase House ocurriría cuando instancias de esa clase y que se llena de datos, y se convierte en su casa, el MyHouse objeto. Usted puede construir tantas casas como quieras desde el azul-imprimir esa es la clase, ya que cada uno tiene su propio espacio en la memoria donde almacena su propia información, llama su estado.

Cubro el desarrollo de las clases en el capítulo 6, pero las clases y objetos son discutió en todo el libro y en todo el idioma. Todo en. NET es un objeto, ha desarrollado por la existencia de la propia aplicación. ¿Cuándo Microsoft desarrolló los objetos, sin embargo, no eran más que las clases!

La planificación de un proyecto utilizando Ciclo de Vida del Proyecto

Preparación para crear un nuevo proyecto consta de dos fases distintas, la planificación y el diseño. Planificación consiste en definir el proyecto y la recolección de la requisitos. Diseño consiste en escribir las pantallas y lógica que cumplir con los requisitos, y encontrar la manera de probar para ver si son derecha.

En lugar de escribir sobre cómo seguir esta receta, usted camina a través de la planificación y diseño de un proyecto que se construye en la Parte II. La proyecto de ejemplo es un programa que calcula fechas. ¿Qué Date Calculator hace el programa y cómo funciona son las cosas que figuran en el marco de la proyecto de desarrollo del ciclo de vida.

El ciclo de vida del proyecto es un proceso que se muestra mejor en el diagrama de Gantt

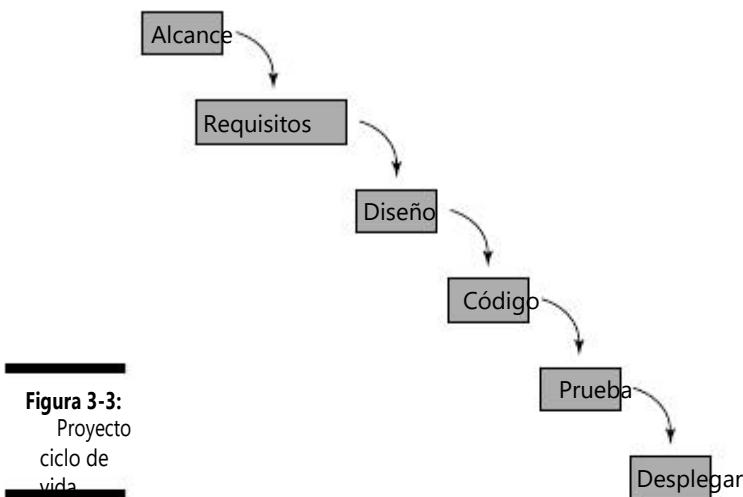


Figura 3-3:
Proyecto
ciclo de
vida

Los proyectos deben ser completados mediante este proceso. Si usted ha estado programando en Visual Basic por un tiempo, te habrás dado cuenta de que muchos de los programadores de otras lenguas son a veces desdenoso de Visual Basic. Parte de la razón es que es muy fácil escribir programas sin ningún tipo de planificación con Visual Básica. Si lo hace, no es una buena cosa.



Debo decirles que este ciclo de vida del proyecto es sólo uno de muchos procesos de diseño. Usted puede escuchar un montón de términos de la industria arrojados alrededor, y estos términos repre-enviado varios ángulos sobre el mismo paradigma básico. Sólo recuerde que no importa lo que usted llame a su proceso de desarrollo, la planificación y el diseño son cosas buenas.

Para escribir una aplicación decente, primero debe tener un plan. A pesar de que los pasos pueden superponerse, cada paso debe completarse. Incluso en pequeños proyectos, una hora gastado en el diseño vale la pena el tiempo. Los errores que usted descubra en la planificación de son cerca de diez veces más baratos en términos de tiempo más largos que los bichos que encuentres y squash en desarrollo y pruebas.

Como vas a través de los pasos del ciclo de vida del proyecto de desarrollo (véase a la Figura 3-3), usted encontrará que los tres primeros pasos que se plantean cuestiones necesidad de responder a entender las necesidades de su proyecto. Cuando se planifica la aplicación que se va a crear en la Parte II, las preguntas podrían parecer algo como esto:

Ámbito de aplicación: ¿La calculadora de la fecha necesita ninguna capacidad de cálculo de otros?

¿En qué plataforma es? ¿Es necesario ser internacional?

Requisitos: ¿Qué es exactamente el cálculo del programa? ¿Cómo se usuario, introduzca las fechas? ¿Qué resultados espera el usuario?

Capítulo 3: Diseño de aplicaciones en VB 2008

Diseño: ¿Cómo será el programa calcule las fechas? ¿Qué interfaz de usuario elementos representa mejor los datos? Qué inicia la aplicación a calcular? ¿Cómo va a mirar la pantalla?

Alcance hacia fuera del sistema

Alcance es la parte más importante del proceso de diseño, ya que define exactamente lo que hará la aplicación. Si alguien le pregunta cuál es su aplicación hace, usted debería ser capaz de decirle a él o a ella mientras estaba parado en un solo pie.

Tal vez más importante aún, el alcance define lo que su aplicación no es suficiente. El término "fuera de alcance" se refiere a esto.

Trate de escribir la definición de la aplicación en 101 palabras o menos. Si lo hace, le permite mantener el alcance corto porque usted está pensando en la el significado de cada palabra.

Para la Calculadora de la fecha, un alcance 101-palabra puede ser un poco mucho, porque la sistema es bastante simple. Listas de viñetas son formas convenientes para escribir ámbitos:

La Calculadora de la fecha es una aplicación que encuentra diferencias entre los dos EE.UU. data.

Se ejecuta en una página Web o un ordenador con Windows, o en función de cualquier tipo de aplicación.

Este ámbito define la aplicación. Cuando un usuario dice que él o ella espera que la Calculadora de la fecha se suman dos números, ya que es un calculador, puede responder que la función estaba fuera de su alcance. Si este es un esperado requerimiento, el alcance debe ser alterado, que te permite volver al tablero de dibujo en las etapas de planificación.

Recopilación de requisitos

Requisitos son las normas específicas que rigen la aplicación. Piense en requisitos en cuanto a los problemas que deben ser resueltos en la etapa de diseño del ciclo de vida del proyecto. Para la Calculadora de la fecha, estos problemas son bastante sencillo:

La Calculadora de la fecha acepta una fecha startDate EE.UU. y un entero span, y devuelve el endDate fecha que es el número de días de lapso StartDate.

aceptar dos fechas startDate y endDate y devolver el número de días entre las dos fechas como una
lapso entero.

51



La Calculadora de la fecha debe ser capaz de ser implementada como:

- Una aplicación de Windows, como un archivo ejecutable
- Una aplicación Web, se ejecutan en un entorno cliente / servidor
- Un componente reutilizable en Windows
- Un servicio Web XML

Si es posible, los últimos cinco cálculos serán almacenados por la aplicación y salvado de la utilización de su uso.



"Si es posible" requisitos son sorprendentemente comunes. Básicamente, consisten de características que pueden o no pueden encajar en el presupuesto. Deja eso para el final.

He recopilado toda la información que necesito para describir la funcionalidad de la aplicación. Esta información debe ser colocado en un documento, Consignadamente suficiente llamó un documento de requisitos. Este documento puede ser una palabra archivo, un archivo de texto, una hoja de papel de cuaderno, o una servilleta de papel. Creación y utilizando un documento de requisitos ayuda a garantizar que la aplicación finalizada hace lo que se supone que debe hacer.

Cada uno de los puntos del documento de requisitos deben ser cubiertos por un punto en la siguiente etapa, el documento de diseño. Es posible que desee numerar los puntos en el documento de requisitos y en el documento de diseño para asegurar que cada requisito tiene un diseño relacionado.

Cuando los requerimientos se resuelven, es el momento para describir el software de un punto de vista técnico: el final de la fase de diseño. En la siguiente sección, cubrir las etapas: elaboración de pantallas y la definición de la lógica.

El diseño de la calculadora de fechas

Los pasos en la parte inferior de la tabla de desarrollo del ciclo de vida del proyecto son más temas técnicos. El diseño, pasos código, probar y desplegar son generalmente tramitadas por los desarrolladores en lugar de los analistas de negocio en un gran desarrollo ir de compras. Si usted está trabajando solo, se llega a hacer de todo!

Usted debe referirse a un programa cuidadosamente ya fondo para que pudiera entregar el documento a un interno con el código. Para la Calculadora de la fecha, es necesario analizar tres puntos principales. Dividir el esfuerzo en estas secciones lógicas hace la vida más fácil cuando se construye la mayoría del software:

Diseña tus datos.

Dibuje su interfaz de usuario.

Diagrama de las conexiones entre la capa de negocio y la capa de datos.

Entender n-tier diseño

Un sistema de n niveles es el que tiene la presentación de capas, lógica de negocio y datos physicalmente se ejecutan en distintas plataformas, con menos al menos una sola capa dividida entre aquellos plataformas. Las aplicaciones Web son perfectas para n-tier arquitectura, porque la capa de presentación es dividida entre el navegador Web y Web servidor y los componentes de negocio y los datos se puede dividir - como un cliente / servidor aplicación - entre un Object Request Broker y un Sistema de Gestión de Base de Datos.

Cuando se diseña un sistema grande, me gusta definir la primera base de datos, conocida como la capa de datos. Entonces me suelen construir la interfaz de usuario, o el

Capa de presentación. Finalmente, atar los dos juntos, utilizando la capa de lógica de negocios.

El beneficio para un sistema de n niveles es doble. La modularidad de un buen n-tier diseño permite que el eliminación o sustitución de un determinado componente sin afectar la funcionalidad de la resto de la aplicación. Además, la separación de la lógica de negocio de la base de datos permite cargar equilibrio, la seguridad y la estabilidad general en sistemas altamente disponibles. La conclusión es que n-tier sistemas transaccionales a sustituir el resmas de código COBOL que dirigen el mundo economía. Si usted quiere tener un impacto, es necesario comprender los sistemas de n niveles.

Almacenamiento de datos

La Calculadora de la fecha almacena la información que se recopila y calcula. Si leer cuidadosamente los requisitos establecidos en los "requisitos Gathering" sección, anteriormente en este capítulo, se dará cuenta de que sólo tres valores existen:

- La primera cita, startDate
- La segunda fecha, endDate
- El número de días entre la StartDate y EndDate, span

También es posible que desee considerar el almacenamiento de los datos siguientes:

- La fecha en que el último cálculo se ejecuta
- El último usuario que ejecutó el programa Calculadora de la fecha
- Algunos forma de referirse a la búsqueda

Puede almacenar estos datos en una sola entidad, que se puede llamar Cálculos. Por lo general, éstas se describen usando una rejilla que se parece a la Figura 3-4. Este diagramo es una entidad-relación (ER) Diagrama porque, si hay más de una entidad existido, las relaciones se muestran como líneas entre las rejillas. Bases de datos comúnmente se muestra en esta forma.



Figura 3-4:
La Entidad
Relación
diagrama de
cálculos.

Calculations	
PK	CalculationId
	StartDate
	Span
	EndDate
	DateRan
	User

Al utilizar el diagrama ER, puede ver el tipo de información que su aplicación será el manejo - un esfuerzo muy útiles. Tres de estas piezas de información, o campos, representan la información del usuario, y tres de ellos reenvía información del sistema. En la siguiente sección, le mostraremos cómo diseñar una maqueta de pantalla que utiliza estos campos apropiadamente.

Diseño de pantallas

Volviendo a la "Reunión requisitos" la sección, usted puede ver que esta aplicación debe ser una aplicación multiplataforma. De las cuatro plataformas que los es necesario desarrollar, sólo dos de ellos (Windows Forms y Web) que el usuario pueda interfaces, y son bastante similares. Debería ser capaz de utilizar la misma diseño de interfaz de usuario para ambos.

Sobre la base de los requisitos y el diseño de datos, es necesario crear tres identificables por el usuario y controles de campo tienen alguna forma de presentar la información para el sistema. Luego hay que crear algún tipo de control para manejar el "Últimos cinco" características.

Mi recomendación es comenzar con el campo de entrada y salida. Usted necesitará tres implementaciones de los campos. La manera más simple para recoger el usuario de entrada es utilizar sólo tres cuadros de entrada de texto para esto, y etiquetar cada caja de texto adecuadamente.

Aquí es donde las piezas de interacción con el usuario de la biblioteca de clases base vienen pulg Si lo hace un poco de investigación en la documentación, usted descubrirá que tanto la ASP.NET y Windows Forms cajas de mantener un control que permite al usuario elegir una fecha en el calendario. Aquí es donde el conocimiento de lo que el sistema puede hacer por usted viene muy bien. Muchos desarrolladores puedan construir su propia fecha selector, sin saber que uno ya estaba disponible.

Así que sabiendo eso, es necesario el uso de dos selectores de fecha, una caja de entrada de texto y un botón, el usuario puede hacer clic para que el programa realice el cálculo para determinar el número de días entre dos fechas. La maqueta de estos componentes se ve algo como la ventana que se muestra en la Figura 3-5.



Figura 3-5:
La Fecha
Calculadora
inicial del
usuario
interfaz
diseño

En este punto, usted puede no saber exactamente cómo funcionan los calendarios, por lo que

No se puede crear una imagen precisa. Pero eso está bien. ¿Sabes cuáles son los calendarios

vamos a hacer - dejar que el usuario elija una fecha.



Diseño de interfaz de usuario es a veces una cuestión de experiencia. Puedo ver que esta diseño trabajará en la Web y los mundos de Windows. Eso no es obvio a todo el mundo. Si usted no está familiarizado con el diseño web y tienen un ASP.NET requisito, obtener un desarrollador web para ayudar. Nada es más difícil que tratar de escribir una aplicación de Windows utilizando la web como plataforma.

El principal requisito segunda es la posibilidad de guardar los últimos cinco búsquedas. Para

ejemplo, puede optar por mostrar este requisito en el diseño como una red ampliable que aparece para mostrar al usuario qué búsquedas habían sido guardado. Esto rompe en dos ADO.NET y los controles de la interfaz de usuario. En ADO.NET, puede obtener los últimos cinco búsquedas de este usuario preguntando al base de datos para ellos, y tanto Windows Forms y Web tienen una cuadrícula de datos espectador.



Por lo tanto, se necesita una red que se mostrará al usuario la fecha de inicio, finalización fecha, duración y fecha de búsqueda, y que permitirá que el usuario haga clic en un salvado ver y buscar en ella. Esta rejilla sólo se puede añadir a la parte inferior de la pantalla, como se muestra en la Figura 3-6.

Utiliza software, y usted sabe qué interfaces que te gusta y no les gusta. Esforzarse para el diseño de interfaces que le gustaría utilizar. Tome software populares, como Windows, Office, Quicken, y así sucesivamente, y diseñar sus propias aplicaciones de esa manera.

Recuerde que como probados desde hace varios años, no se puede patentar apariencia!

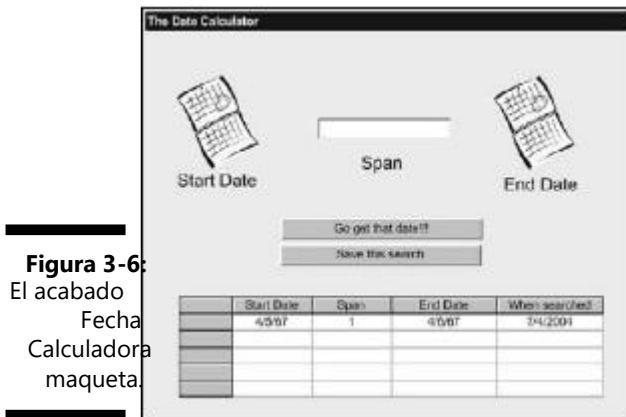


Figura 3-6:
El acabado
Fecha
Calculadora
maqueta

Definición de la lógica

Así que ya sabes lo que es el software de almacenamiento y cómo se verá el usuario.

El último paso es conectar la interfaz de usuario y los datos. Esto es generalmente llama la capa de lógica de negocios, ya veces se separa del resto de la solicitud y poner en una máquina totalmente diferente.

Usted encontrará que en el mundo del desarrollo de Windows, lo mejor es sólo averiguar todo lo que un usuario puede hacer en la aplicación y, a continuación, escribir pseudocódigo que describe que la funcionalidad.



Pseudocódigo es el lenguaje-neutral instrucciones que describen la funcionalidad. Pensar de lo que escribir el código de su programa en Inglés. El objetivo es crear una línea por descripción de la línea de lo que una interacción con el usuario dado se supone que debe hacer para que, si otra persona que le sucede código, o la figura hacia fuera después de que te hayas ido, que persona puede ver lo que quería lograr. La mayoría de las metodologías de llamar estos casos de uso, historias de usuario, o escenarios.

Un número limitado de acciones del usuario están en esta aplicación, y la aplicación puede hacer un número limitado de cosas. A continuación se presenta un detalle, y lo que su diseño se vería así:

Aplicación de carga:

- Ajuste el StartDate y EndDate igual a la fecha actual y dejar lapso vacío.
- La suposición es que el usuario desea buscar los números de días entre dos fechas dadas.
- Coloque la parrilla búsqueda guardada desde la fuente de datos, listado guardado Búsquedas en el orden inverso de fecha y tiempo ahorrado.

Capítulo 3: Diseño de aplicaciones en VB 2008

Cuando el usuario hace clic en el botón Calcular Fecha de diferencia:

- Si un valor está en los campos startDate y span, agregue el lapso número de días a la startDate y mostrar la calculada fecha en el campo endDate.
- Si un valor está en StartDate y EndDate, poner la diferencia en días entre las dos fechas en el campo de span.
- Si un valor es en todos los tres campos, se supone que el usuario desea calcular finales del tramo.

Supuestos matar a proyectos de software. Nunca haga una suposición en el diseño, como acabo de hacer. Siempre pregunte al usuario lo que él o ella quiere.

Acabo de hacer esta suposición en aras de simplificar el ejemplo.

Guardar esta búsqueda:

- Añadir el startDate, endDate, span, fecha y hora actuales, y usuario actual a los campos de datos de almacenamiento.
- Actualizar la red con la nueva búsqueda, manteniendo la más nueva búsqueda en la parte superior.

Cuando el usuario hace clic en una búsqueda en la red:

- Capacidad de carga hasta el startDate, EndDate, y valores de la amplitud en los campos.
 - Sustituya los valores actuales en el startDate, endDate, y la duración de campos con los valores de la búsqueda.
- Todo lo demás sigue siendo el mismo!

Eso de debe cubrir la funcionalidad que el usuario espera de la aplicación, y que cumple con todos los requisitos. El sistema está diseñado!

Escribir un plan de prueba

Antes de empezar a programar, escribir un plan de pruebas. Es sencillo y se asegurará de que que han afectado a todas las partes importantes de su diseño. Los pasos son simples:

1. Revisar el requisito de que un punto particular diseño soporta.

Por ejemplo, la aplicación Calculadora de la fecha tiene el requisito de que los el usuario puede introducir una fecha de inicio, fecha de finalización, y un palmo.

2. Listar el punto de diseño en cuestión.

La interfaz Calculadora de la fecha debe tener controles que aceptan entrada de datos

para asegurarse de que el punto de diseño trabaja.

Se asegura de que el usuario introduce las fechas enfocando la entrada con un Calendario control. ¿Y el tramo? Tiene que ser un número entero!



4. Describir lo que puede pasar para que el punto de quiebre de diseño o causar un error.

Una de las preguntas que usted necesita hacer en el plan de prueba es "¿Qué pasa si el usuario introduce una no entero en el campo de tiempo?" Por supuesto, en condiciones normales circunstancias, este tipo de entrada, se producirá un error de algún tipo.

Puede que tenga que modificar su diseño para asegurarse de que el usuario sólo puede entrar un número. ¿Un cuadro de texto permitir eso? Si usted hace una búsqueda rápida en Google, encontrar que un cuadro de texto existe para Windows, pero no para la Web. Para obtener más información detallada acerca de cómo Visual Basic puede ayudar a validar la entrada del usuario, véanse los capítulos 4 y 5.



Sólo asegúrese de tener un plan de prueba por escrito de que usted puede dar a un tercero para asegurarse de que su aplicación hace lo que usted espera. Es mejor para usted no poner a prueba su propia aplicación. Usted debe tener a otra persona trabajar a través del plan o utilizar un sistema automatizado de prueba como NUnit o Microsoft Team System.

Cumplir con el plan de

Ahora que ha creado el plan, el truco consiste en seguirlo. El punto siguiente res puede ayudar a mantener el plan:

No reinventar la rueda. Busque soluciones en aplicaciones similares o aplicaciones de ejemplo antes de volver a escribir algo.

Investigue y lea la documentación. No seas un usuario avanzado, un "Voy a averiguarlo yo mismo" tipo de persona. El Framework .NET es demasiado grande. Aprende a usar la documentación - Les explico en el Capítulo 2.

El código de la forma en que desea ver el aspecto de la aplicación. No te rindas. Si que cree que debería ser capaz de hacer algo, seguir cavando hasta que ver cómo se hace. Si no vale la pena, puede rediseñar.

Escribir menos código. Utilice las herramientas de la interfaz de usuario de Visual Studio que da usted. No ceda a los snobs código que cree que debería codificar manualmente todo.

Sea consistente. Utilice los mismos nombres como lo hizo en su diseño. Decidir cómo llamar a los conceptos. No utilice x para hacer referencia a un número.

Parte II

Edificio

Aplicaciones

con VB 2008

The 5th Wave By Rich Tennant

©RICH TENNANT



"Did you click 'HELP' on the VB.NET menu bar recently? It's Mr. Gates. He wants to know if everything's all right."

En esta parte. . .

Visual básica es escribir software, y, en la Parte II, escribir programas para Windows y la Web. Usted empezar por crear una aplicación de Windows tradicional, y también construir bibliotecas de clases para ir con ella. A continuación, construir una aplicación web y servicios web XML.

Capítulo 4

Construcción de aplicaciones de Windows

En este capítulo

Experimentar el poder de Visual Studio para crear aplicaciones de Windows

Creación de su primera aplicación para Windows

Capacitar a su solicitud con el código de

En cuanto a los formularios Windows Forms con mayor profundidad

B

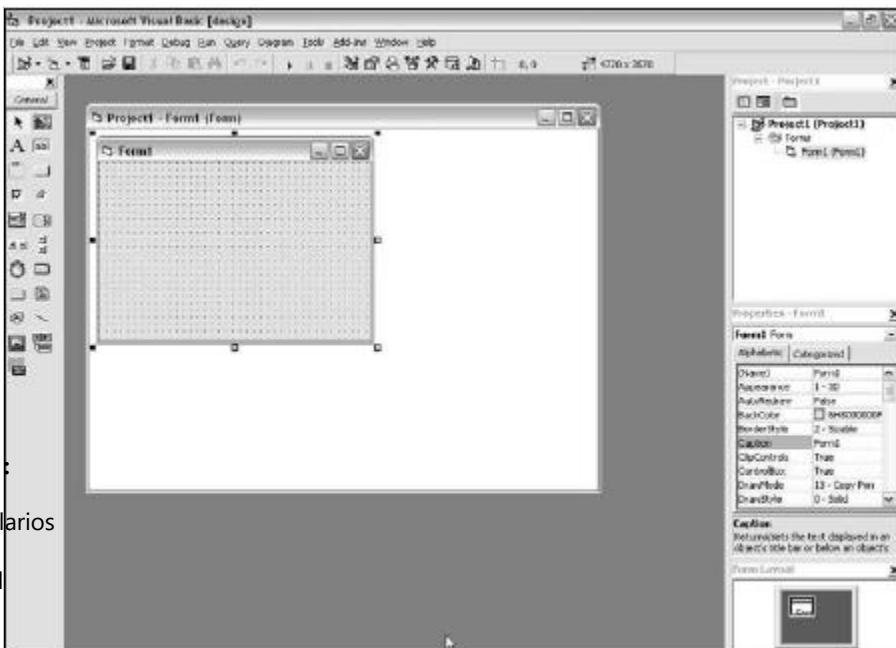
CREACIÓN un proyecto de Windows Forms con Visual Basic 2008 es una gran manera para empezar a trabajar con el idioma. Usted está familiarizado con Windows aplicaciones, tales como Microsoft Word, que estoy utilizando para escribir este libro. ¿Cuándo haya terminado de leer este capítulo, usted podría revisar su correo electrónico con Outlook Express o Groupwise. Cada programa que se utiliza en un Microsoft Windows computadora es una aplicación de Windows, por definición, pero no todos son desarrollado por Microsoft. Algunos son desarrollados por los programadores usando una herramienta como Visual Studio 2008, utilizando un lenguaje como Visual Basic 2008.

En este capítulo, me tomo una vuelta vistazo a cómo el lenguaje ha cambiado desde Visual Basic 6.0. Luego cubrir los bloques de construcción de formularios Windows - el colección de controles de Windows siempre con el idioma. Usted también encontrará la manera de construir su primera aplicación - la Calculadora de la fecha en que el diseño Capítulo 3. Por último, yo voy añadiendo las características que los usuarios esperan encontrar para aplicaciones de Windows Forms - características tales como la introducción de texto, menús, barras de estado, y las sugerencias.

A Look Back rápido en Visual Basic

Cuando usted piensa acerca de Visual Basic, es probable que también piense en Microsoft Aplicaciones de Windows. Durante 15 años, los desarrolladores han utilizado el Visual Basic programa Ruby motor formas (que se muestra en la Figura 4-1) para escribir común empresas aplicaciones empresariales. Cuando un programa se define como un programa de VB, que era, por defecto, una aplicación de Windows.

Figura 4-1:
The Ruby
Formularios
Motor
en Visual
Basic 6.0.



Sin embargo, el alcance y la versatilidad de Visual Basic creció con la introducción de Visual Basic.NET. (Consulte el Capítulo 1 para obtener información detallada acerca de este transformación.) Visual Basic.NET es un lenguaje, al igual que Java, COBOL, o C++, que puede utilizar para escribir cualquier tipo de aplicación que se apoya en una API (Application Programming Interface). Ahora, cuando un programa se define como un programa de Visual Basic, debe preguntarse: "¿Qué tipo de programa? Es un equipo con Windows aplicación? Un sitio Web? Un servicio de Windows o el Servicio Web XML?"

La aplicación de Windows - que ahora se llama una aplicación de Windows Forms - es lejos de estar muerto. Aunque las aplicaciones Web están claramente creciendo en popularidad, el rico entorno de una aplicación de Windows no sólo familiar, sino también difícil de superar para muchos usos. En este capítulo, usted descubrirá cómo utilizar el instrumento aún poderoso, Visual Studio, para crear una aplicación de Windows Forms. En concreto, les muestro los tipos de elementos que proporciona Microsoft, cómo para estructurar una aplicación de Windows Forms en Visual Basic 2008, y los detalles bajo la superficie de formularios Windows Forms.

Los formularios Windows Forms tienen muchas más funciones de las que puedo cubrir aquí en esta capítulo. Más de 60 controles de Windows Forms se integran en Visual Studio 2008, y también se puede muy simplemente crear uno propio. Visual Basic es un potente lenguaje por derecho propio, y combinándolo con los controles que se encuentran en Windows Forms lleva un control casi total sobre la experiencia del usuario.

Capítulo 4: Creación de aplicaciones de Windows

Descubriendo los controles de Windows

Con los años, una forma estándar para crear aplicaciones de Windows se ha desarrollado ya que los usuarios esperan que las aplicaciones funcionan de una manera determinada. Menús, barras de herramientas, barras de estado y cursores se han convertido en equipo estándar. Como se muestra en Figura 4-2, la calculadora tiene una barra de título, cuadros de texto y un botón, así como texto en la pantalla. Todas estas interfaces estándar de Windows se desarrollan mediante los controles proporcionados a usted por la Caja de herramientas en Visual Studio 2008.

Figura 4-2:
La Fecha
Calculadora
como
Ventanas
aplicación.

Para que el uso estándar de Windows ofrece más fácil, Visual Studio incluye todas las como equipamiento de serie para el desarrollador, también. Estas piezas de norma equipos se denominan controles. Los controles están preprogramados piezas de usuario interfaz de código que manejan una gran cantidad de la tubería para usted, el Visual Basic programática. Usted las puede utilizar en la aplicación de Windows Forms para proporcionar las características que los usuarios quieren.

La palabra de control es genérico y abusa a menudo - incluso por Microsoft. En general, sin embargo, yo uso este término para referirme al código que realiza una función a reconocer (como un cuadro de texto o un botón) funcionan de la forma esperada.

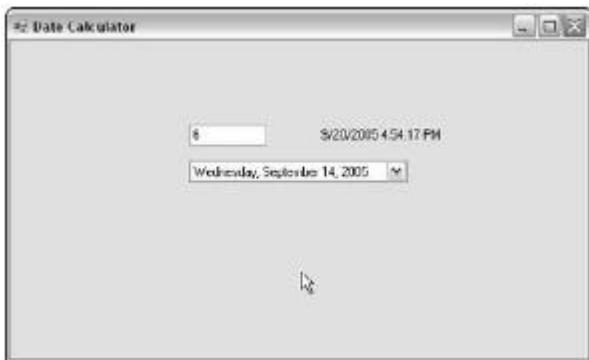
Uso de los controles es fácil. En las siguientes secciones, encontrará la manera de hacer la después:

Coloque un control arrastrándolo desde el Cuadro de herramientas hasta el formulario.

Escribir código para un control haciendo doble clic en él en la forma que toma que en vista de código.

Cambiar las propiedades de un control haciendo clic en él y cambiar los valores

edades cambiantes de un control, consulte la sección "Adición de funciones a Windows Forms", más adelante en este capítulo.



El poder de utilizar Visual Basic.NET para construir formularios Windows se encuentra en el control. La Tabla 4-1 muestra algunos controles de formulario comunes y sus usos.

Tabla 4-1**Algunos controles de formulario**

Control	Herramientas Usa	
Etiqueta	del icono	Muestra el texto en la pantalla que no es editable por el usuario
TextBox		Acepta la entrada de texto básico del usuario
RichTextBox		Ofrece tipos de procesadores de texto funcionalidad como negrita y cursiva
Botón		Hace que la aplicación para realizar una tarea predefinida
DataGridView		Muestra una tabla editable en un formulario
DateTime Recogedor		Permite al usuario seleccionar una fecha en un calendario
TabControl		Proporciona la interfaz de usuario de navegación, junto con otras herramientas, como botones y puntos de vista de árbol
MenuStrip		Muestra una barra de menú, como usted encontraría en Word o Outlook
ToolStrip		Ofrece Oficina funcionalidad de tipo barra de herramientas, incluyendo abrir / guardar y cortar / copiar / pegar
PrintDialog		Ofrece a los usuarios un fácil acceso a la impresión
ErrorProvider		Comunica a problemas de entrada de los usuarios
WebBrowser		Incluye un navegador en tu aplicación
DomainUpDown		Permite a los usuarios seleccionar de una lista

Realización de una aplicación para Windows

Un buen lugar para empezar a construir una aplicación es un formulario de entrada de usuario

(Porque eso es lo que un programador VB hace más a menudo), y las siguientes pasos que conducen a través de ese proceso. El ejemplo que uso es una calculadora de fechas

que acepta datos (una fecha) del usuario y luego vuelve a Calculado valor (una fecha futura) basada en dicha entrada. Siga estos pasos para hacer una nueva Aplicación de Windows:

- 1. Abra Visual Studio 2008 y haga clic en el botón Nuevo proyecto para acceder a el nuevo proyecto de cuadro de diálogo.**

No importa qué tipo de aplicación VB que desea hacer, usted comenzará a el nuevo proyecto de cuadro de diálogo, que se muestra en la Figura 4-3.

Figura 4-3:
The New
Proyecto
Cuadro de diálogo.

- 2. Seleccione Windows como el tipo de proyecto de Visual Basic y seleccione el Plantilla de formularios Windows Forms Application.**

- 3. Escriba un nombre de proyecto correspondiente en el Nombre y la solución cuadros de texto y haga clic en el botón Aceptar.**

Puse el nombre de mi DateCalc aplicación y luego ponerlo en un nombre de la solución

llamado el Capítulo 4, porque voy a construir una aplicación similar para los Capítulos 5, 6, y 7.

Visual Studio genera un proyecto nuevo para usted, incluyendo una nueva forma (Form1.vb) y la carpeta MyProject.

tón de trabajo para usted. Inicialmente, se

Parece como si un formulario (Form1.vb) y un archivo de proyecto (MyProject) se crean, pero en realidad mucho más se logra. Visual Studio crea lo siguiente:



- A Referencias carpeta para almacenar las partes de la estructura que se quiere a utilizar para este proyecto.
- Una carpeta bin para mantener el programa terminado y alguno de sus componentes.
- Una carpeta obj para mantener los recursos específicos a la aplicación.
- La carpeta MyProject que contiene los archivos de configuración. Puede editar este carpeta haciendo doble clic en él en el Explorador de soluciones. Visual Studio le proporciona un formulario con fichas a utilizar para editar las distintas configuraciones del racionamiento, como se muestra en la Figura 4-4.

4. Cambie el nombre del formulario predeterminado (Form1.vb), haga clic en la Explorador de soluciones y seleccionando Cambiar nombre en el menú contextual.

Puse el nombre de la DateCalc forma en la aplicación de ejemplo. En su programación, el nombre de los formularios debidamente para que sus proyectos son algunas- lo que la auto-documentado. Después de cambiar el nombre del archivo, Visual Studio le pide cambiar el nombre de todos los objetos de referencia. Seguir adelante y dejar que él.

5. Cambie el tamaño del formulario predeterminado haciendo clic en el asa en la parte inferior derecha esquina de la forma en que el diseñador y arrastrándolo a una nueva posición.

En este paso, el diseño comienza a ser importante. Si usted no sabe cómo muchos controles van en la forma, no sé qué tamaño para hacer ella. Aunque siempre se puede cambiar el tamaño después, es mucho más fácil saber exactamente lo que va a poner ahí! (I abarcará el diseño de la calculadora de fechas en el Capítulo 3.)



Figura 4-4:
La
MyProject
configuración
formulario.

Capítulo 4: Creación de aplicaciones de Windows

6. Arrastre los controles de la barra de herramientas de Windows Forms en el nombre

formulario predeterminado.

Puede agregar controles como TextBox o un MenuBar a los formularios simplemente arrastrándolos desde el Cuadro de herramientas de Windows Forms al formulario.

El tipo, la cantidad y la ubicación de los controles que agregue a su forma definir su aspecto y función eventual. Al arrastrar controles a la forma, los controles de asumir las propiedades predeterminadas que Visual Studio apoya. Específicamente, el nombre de la instancia del control se establece en la nombre dado a ese tipo de control seguido por un número. Por ejemplo, Label1 es para un control Label, o DateTimePicker1 es para un DateTimePicker control.

I utiliza un control DateTimePicker como uno de los controles primarios en el formulario. La adición de este control permite a los usuarios seleccionar la fecha de inicio.

También es necesario agregar un control Label para mostrar los resultados de la operación

ción. Un control Label se puede preestablecer ahora, en tiempo de diseño, y dejó estático.

O puede ser modificada por el código, en tiempo de ejecución, para mostrar los resultados de un operación.

Hay controles, y hay casos de controles. El control a largo plazo se aplica a un DateTimePicker. Después de arrastrar un control en el formulario, se convierte en una instancia del control, referido por un nombre de referencia. Los nombres por defecto de referencia son el nombre del tipo seguido de un número en secuencia.

7. Despues de que todos los controles adecuados se han añadido, haga clic en cada

una y pulse F4 para abrir la ventana Propiedades, donde puede cambiar cada nombre por defecto a algo memorable.

Nombro mi DateChooser control DateTimePicker y-con mi sello trol nextweek porque eso es lo que se mostrará cuando el usuario selecciona un fecha.

Dejando sus propiedades controles 'establecen en los valores por defecto (por nombre y

Contenido de texto) es una mala idea. Si no restablecer los valores a algo más lógico, es probable que no recuerde lo que controla el trabajo que está ING con cuando usted ve los valores por defecto en la vista Código.

También puede cambiar el valor de texto del formulario para mostrar algo bonito en la barra de título de la aplicación. Yo Date Calculator.

8. Guarde el proyecto haciendo clic en el botón Guardar.

buscar cuando se ejecuta. La forma en mi aplicación de ejemplo se parece a la Figura 4-5. Su forma puede tener un aspecto un poco diferente de la mía, y eso está bien. Siguiente, es necesario agregar algo de lógica de negocios para el formulario para que lo que realmente hace algo.





Figura 4-5:
La Fecha
Calculadora
con
controles
añadido

Adición de funciones a un formulario Windows Forms

Al igual que con la mayoría de otros entornos de desarrollo integrados (IDEs), Visual Studio 2008 separa el aspecto de los formularios que se crean a partir de la funcionalidad detrás de ellos. Al asignar los controles de un nombre significativo, dése una manera de hacer referencia a los controles de código. Cada control tiene funcionalidad predefinida que se puede acceder desde el código.

Puede agregar código VB 2008 con los controles en la vista Código. Usted puede obtener el Código Ver ventana de varias maneras, pero la forma más fácil es hacer doble clic en un control.

Si lo hace, genera un controlador de eventos para el evento predeterminado que el control y la cambia la interfaz de Vista Código. Por ejemplo, haga doble clic en un comando botón te lleva al evento click del botón en la vista Código.



Los controladores de eventos son métodos especiales que se ejecutan cuando un evento particular se produce, como un botón que se ha pulsado o un temporizador alcanza su tiempo asignado.

Para empezar a entrar en la lógica de negocio para los controles del formulario, siga estos pasos:

Capítulo 4: Creación de aplicaciones de Windows

1. Haga doble clic en un control para crear su controlador de eventos e ir a la vista de código.

Los controles en una aplicación de Windows que tiene que hacer más que verse bien, sino que

También hay que hacer algo en respuesta a una interacción del usuario, o un evento.

Visual Studio le ayuda a agregar funcionalidad a los controles. ¿Cuándo hace doble clic en un control, Visual Studio hace dos cosas: Crea el definición para el código que se ejecuta cuando se produce el evento predeterminado - como

como un valor de cambio o un botón que se ha pulsado - y añade un poco de vinculación

código (que se puede ver en el Diseñador de Windows Forms Generado Sección del Código) que enlaza el evento en sí con el código que necesita para funcionar.

Empecé con el DateTimePicker que nombré DateChooser.

Haga doble clic en este control crea un controlador de eventos ValueChanged. Es decir, el código del controlador de eventos se ejecuta cuando el valor en el acondicionado

control se cambia por cualquier razón (por lo general por el usuario). El subroutinamiento diente del controlador de eventos es la siguiente:

```
Public Class DateCalc
```

```
    Private Sub DateChooser_ValueChanged (ByVal sender As System.Object,  
                                         ByVal e As System.EventArgs) Handles DateChooser.ValueChanged
```

```
        End Sub
```

```
    End Class
```

2. Agregue código al controlador de eventos para realizar funciones correspondientes.

Toda la magia se basa en las palabras simples. Aparte de la funcionalidad En los controles propios, toda la funcionalidad de todos los aplicaciones que escriba estará en un controlador de eventos u otro.

Para la Calculadora de la fecha, se necesita un pedazo de código que cumpla con la requisito del Capítulo 3 - es decir, aceptar la fecha, agregar un valor y a continuación, mostrar el resultado. Con el Marco. NET, puede hacer todo eso con una línea de código. Entre la línea que comienza con la privada y la línea que comienza con Fin, escriba lo siguiente:

```
    NextWeek.Text = DateChooser.Value.AddDays (7)). ToString ()
```

He aquí un desglose de esa línea de código:

A la izquierda del signo igual es la propiedad Text del nextweek objeto, que es un control Label. Esto significa que va a establecer la texto de una etiqueta igual a algo.

En el lado derecho del signo igual es el objeto DateChooser. Es

io
se ha puesto. En este caso, va a añadir siete días y luego convertirlo en una cadena.

Así que el código terminado se ve algo así como la que se muestra en la Figura 4-6. Visual Studio insertará una gran cantidad de código para usted, y usted agregue el funcional importante código que hace que todo realizar a los requisitos funcionales. Clic el botón Iniciar depuración, que se muestra en la Figura 4-6, o presione F5 para ejecutar su nuevo aplicación.

Escoja una fecha, y la etiqueta cambiará de un valor en blanco para mostrar una fecha que es de una semana a partir de la fecha seleccionada. Es un juguete aseado, pero no hace demasiado tanto para el usuario. A continuación, te mostramos cómo añadir algunas características del programa.

Adición de funciones a Windows Forms

Para cumplir con los requerimientos funcionales de las aplicaciones, se necesitan funciones. La solicitud de la función más compleja, más complejo que el código tiene que ser. Tomar la aplicación actual, la calculadora de fechas, por ejemplo. Todas las entradas tiene que ser variable para los cálculos a ser realmente útil. Actualmente, sólo el la fecha de inicio es variable. Es necesario cambiar esta situación si desea agregar funcionalidad y características.

Inicio botón Depuración

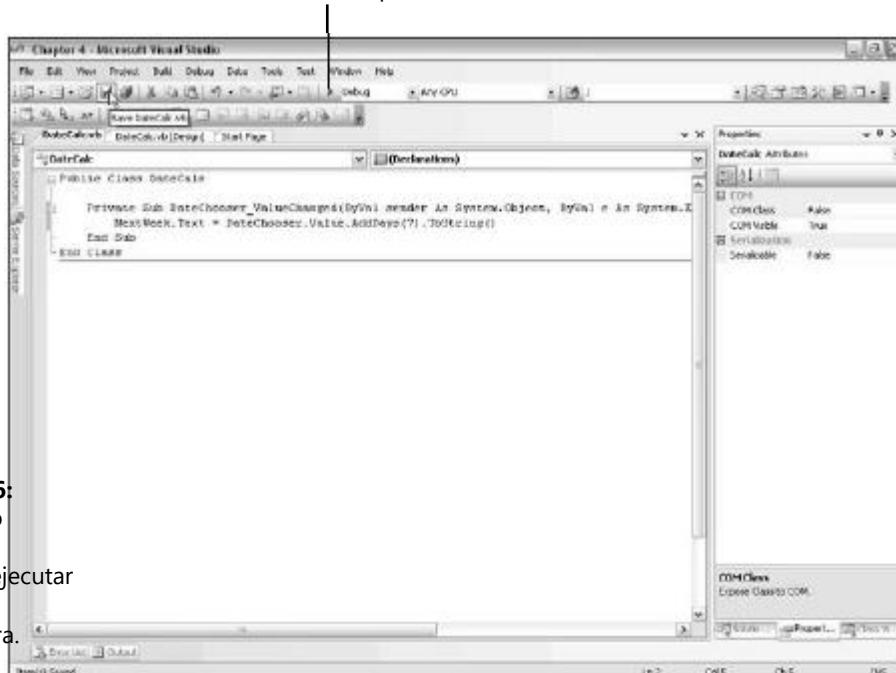


Figura 4-6:
El Código
Ver, listo
para ejecutar
la Fecha
Calculadora.

Gestión de entrada de texto con el cuadro de texto

Es evidente que el número de días (en la actualidad fijado en siete) que utiliza la calculadora de fechas

para calcular la nueva fecha debe ser variable. Siga estos pasos para aceptar la entrada del usuario, específicamente el período en el número de días:

1. Agregar una Control TextBox al formulario.

Cambie el nombre del control TextBox a DateNumber utilizando el Ventana Propiedades. Este cuadro de texto es el control donde el usuario introduce el número de días a añadir a la fecha seleccionada.

2. Alinee el cuadro de texto con otros objetos en el formulario.

Arrastre el cuadro de texto alrededor hasta el lado izquierdo se alinee con el lado izquierdo de la

DatePicker fecha - utilizar las directrices para ayudar a hacer que la adaptación, como

se muestra en la Figura 4-7.

También puede utilizar la función Alinear en el menú Formato. Esta característica hace por el que se los formularios de la forma que quieras mucho más fácil al darle opciones tales como Align Center. Todos los controles y un espacio uniforme entre los controles.

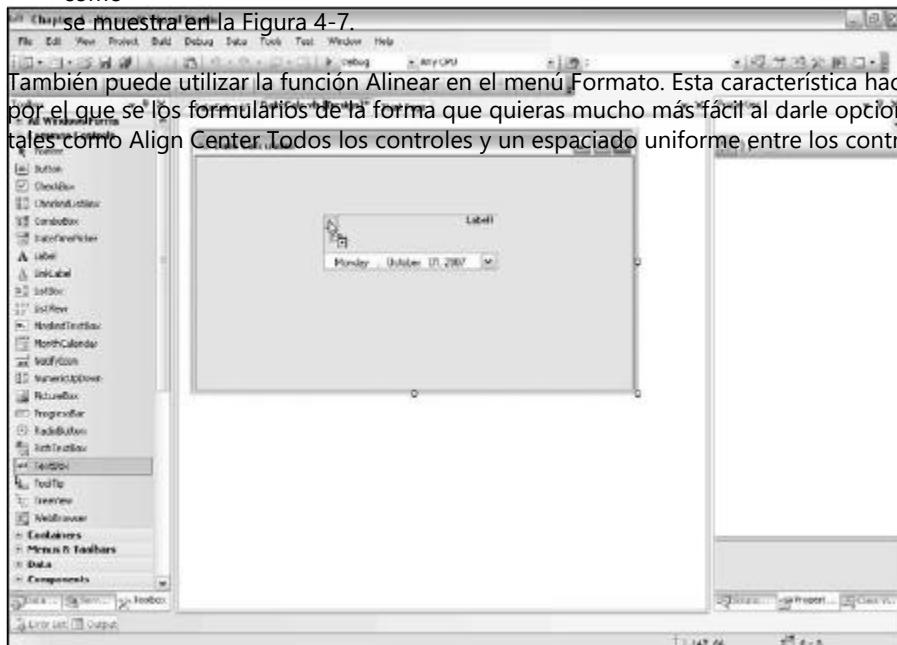


Figura 4-7:
Directrices
que ayuda
que la
línea
hasta su
controles



En este punto, usted debe tener dos pestañas en la parte superior del diseñador, Fecha Calculator.vb [Diseño] y DateCalculator.vb. La ficha Diseño representa el diseñador de formularios, que deben seleccionarse ahora. El otro Ver ficha representa el Código, que se abrió antes haciendo doble clic en un control.

Se necesita un valor predeterminado para el número de días. Si el usuario no aporta cosa en el cuadro de texto, la aplicación debe ser capaz de establecer el valor del intervalo a algo. Por el momento, definir una nueva variable en la vista Código. Bajo el Hereda definición de clase, agregue una instrucción como la siguiente dimensión:

```
DateInterval Dim As Integer = 7
```

Luego reemplazar el cuerpo del controlador de eventos DateChooser_ValueChanged con lo siguiente:

```
Private Sub DateChooser_ValueChanged (ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DateChooser.ValueChanged
    Si DateNumber.Text.Length > 0 Then
        DateInterval = CInt (DateNumber.Text)
    End If
    NextWeek.Text = DateChooser.Value.AddDays (DateInterval).ToString ()
End Sub
```

¿Qué has hecho? Si ha añadido un cuadro de texto que permite a un usuario sobre-montar el número predeterminado de días se está calculando a partir de la fecha en la fecha Selector. Haga clic en el botón Inicio para acceder al modo debug de nuevo y darle una oportunidad. Verás algo como la calculadora se muestra en la Figura 4-8.

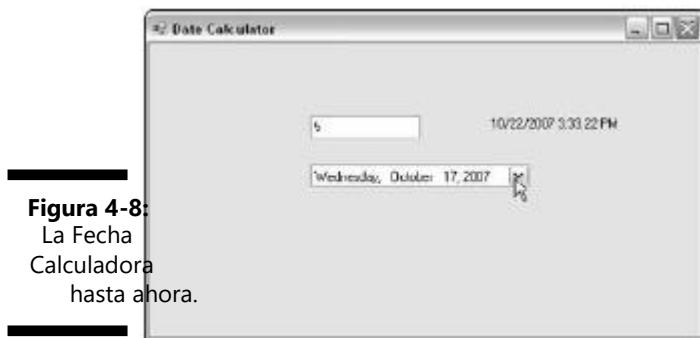


Figura 4-8:
La Fecha
Calculadora
hasta ahora.

Ahora tiene una aplicación que hace lo siguiente:

- Acepta una fecha de inicio del usuario o usa una fecha predeterminada

También tenga en cuenta el vínculo Editar elementos en la ventana Propiedades. Al hacer clic en este enlace Se abre el Editor de la colección de elementos cuadro de diálogo, como se muestra en la Figura 4-11.

Capítulo 4: Creación de aplicaciones de Windows

Acepta un tramo (número) de días desde que el usuario o usa un número predeterminado de día

Calcula y muestra la fecha en que cae el lapso entró de días desde la fecha de entrada

A continuación, sólo tiene que añadir algunas de las características que los usuarios esperan de un Aplicación de Windows, y usted casi puede hacer.

La comunicación con el usuario usando la tira de estado

Ahora que tiene la funcionalidad básica de la calculadora de fechas, deberá agregar aquellas características que los usuarios esperan de una aplicación de Windows.

Menús, estado

tiras, e información sobre herramientas mouseover son parte de la experiencia de Windows. Visual

Studio 2008 es compatible con todo esto y más.

Al momento de escribir este capítulo utilizando Microsoft Word 2007, un estado bastante complejo

Figura 4-9: El estado desnudar se Microsoft Palabra.

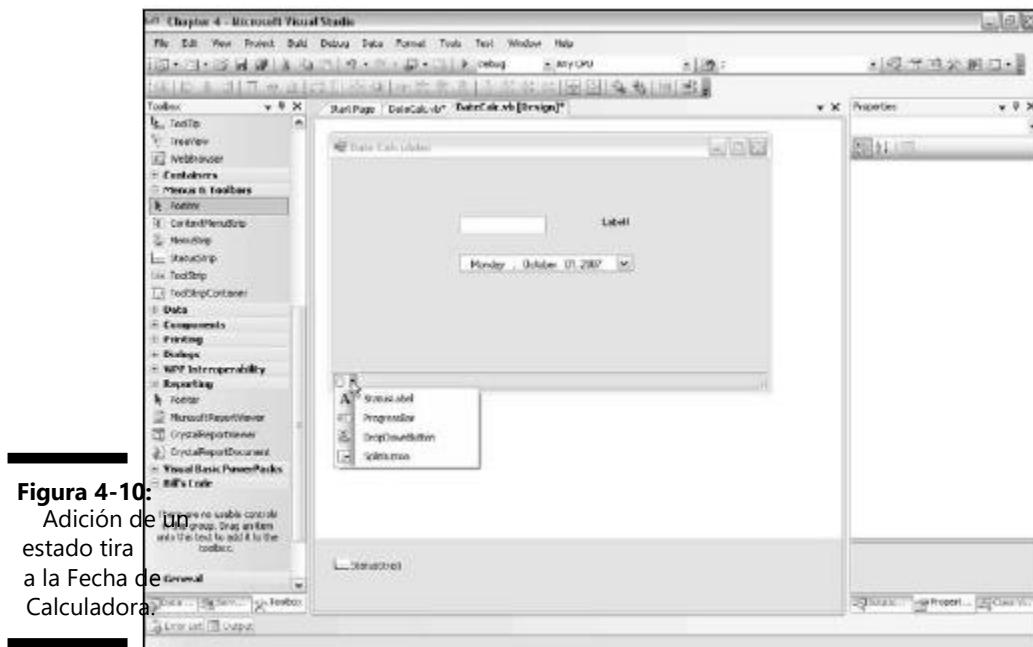
tira aparece en la parte inferior de la ventana, como se muestra en la Figura 4-9. Muestra la página actual, la sección, la línea y la columna, y cuáles son las características que tengo activo, como la grabación de macros o cambios de seguimiento.

Esta franja de estado es otro control proporcionado por el equipo que desarrolló Visual Studio. Busque el control StatusStrip en la caja de herramientas y arrástrelo Calculadora de la fecha de la forma, como se muestra en la figura 4-10. Cambiar el nombre a algo mainStatusStrip gusta.

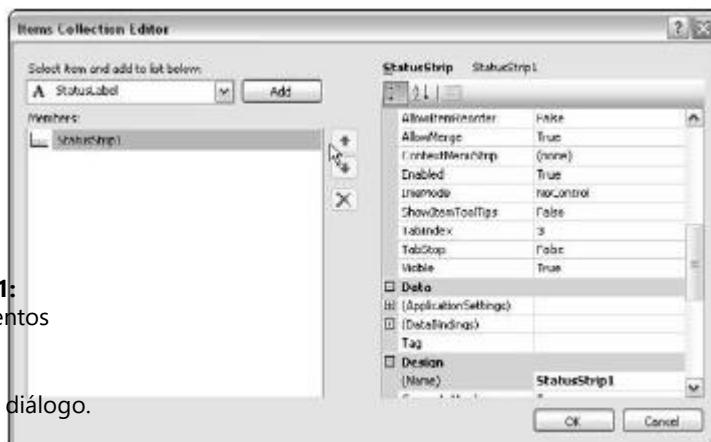
El StatusStrip es sólo un contenedor para la materia. Puedes añadir cosas a la barra como necesaria desde la ventana Propiedades, en la colección de propiedades llamado elementos. La acceso directo está disponible para la adición de estas propiedades, en la forma de una etiqueta inteligente (Véase la figura 4-10).





**Figura 4-10:**

Adición de un estado tira a la Fecha de Calculadora.

**Figura 4-11:**

Los elementos Colección Editor Cuadro de diálogo.

La colección de elementos cuadro de diálogo Editor es una herramienta que será muy familiar - es común a lo largo de Visual Studio como una herramienta para editar los elementos de colecciones. A continuación, añadir algunos elementos de referencia para la StatusStrip sólo colocado en el formulario.

Capítulo 4: Creación de aplicaciones de Windows

1. Para agregar un elemento, haga clic en el botón Agregar.

La instancia aparecerá a la izquierda, en la ventana de los Miembros, y el propiedades aparecerá a la derecha, en la ventana Propiedades.

2. Agregue dos StatusLabels, y el nombre de ellos y datePanel userPanel.

3. En la ventana Propiedades, establezca la propiedad Spring en True.

4. Haga clic en el botón Aceptar para cerrar el Editor de la colección de elementos cuadro de diálogo.

Para hacer algo con estos nuevos miembros de su StatusStrip, es necesario para establecer que el valor de texto y otras propiedades mediante programación, cuando la forma cargas. Usted puede hacer eso en otro controlador de eventos, llamado Form_Load. Para editar el controlador de eventos Form_Load, haga doble clic en el formulario.

El controlador de eventos Form_Load es una de las primeras cosas que se ejecuta cuando un nuevo forma es traída a la pantalla.

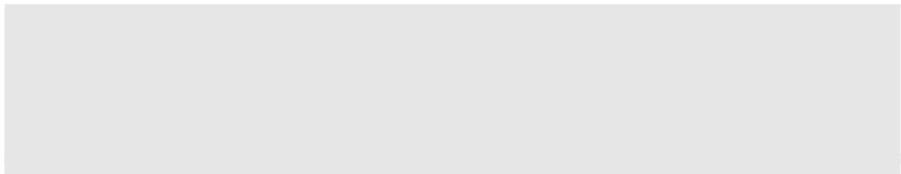
En este tipo de aplicación de una sola forma, el controlador de eventos Form_Load será lo suficientemente cerca para el primer código que se ejecuta que todo el código de instalación debe ir aquí. Debido a que usted desea configurar la instancia StatusStrip cuando la forma cargas, agregue el código siguiente al controlador de eventos que:

```
Private Sub DateCalc_Load (ByVal sender As System.Object, _ ByVal e As  
System.EventArgs)  
    Manijas MyBase.Load  
        datePanel.Text = System.DateTime.Now.ToShortDateString ()  
        userPanel.Text = My.User.Name  
End Sub
```

Dando pistas con el control ToolTip

El control ToolTip te da la posibilidad de agregar información sobre herramientas diferentes para cada controlar en cierta página y controlarlos como una colección. Se puede asignar un ToolTip objeto a cualquier conjunto de objetos, pero cada objeto individual sólo puede ser asociado con un objeto ToolTip. Tenga en cuenta que puede cambiar el individuo Información sobre herramientas.

Debido a que tiene una información sobre herramientas de referencia en el formulario, usted tiene acceso a un ToolTip propiedad de cada control en el formulario. Arrastre un objeto ToolTip en el formulario Calculadora de la fecha y observe que un componente aparezca en la Componente bandeja. Haga clic en el elemento y cambie la propiedad Name a



De los controles y los valores. . .

Mire la subrutina DateCalc_Load en aplicación de este capítulo, para ver si usted Puede encontrar algunas verdades superiores. Cuando se crea un control, que es una instancia de un objeto. La acondicionado trol sabe que puede tener un valor de texto, y-con la trol sabe dónde para mostrar el valor, pero no sabe lo que es hasta el valor que establezca. Usted puede establecer el valor en tiempo de diseño (cuando el código está escrito), o se puede establecer en tiempo de ejecución (Es decir, cuando se ejecuta el código). Tiempo de ejecución frente a las variables en tiempo de diseño son una importante consideración desarrollo, como en el siguiente ejemplos:

Ajuste de texto en tiempo de diseño se maneja como si manejado establecer el intervalo de la fecha de texto. Usted puede establecer el texto en el diseñador, y luego en gran parte no lo toque. Esto es grande para los títulos de las formas, las etiquetas de los botones de radio, y otras cosas de esa manera.

Configuración en tiempo de ejecución texto depende más de su capricho como desarrollador - que se basa en

el medio ambiente en el momento del programa es ejecutado en la máquina del usuario. En la ejemplo para la tira de estado, la fecha y el valor actual del periodo se establecen como los dos paneles. La fecha es, obviamente, una segunda tiempo de decisión porque ese valor cambia día a día. Para hacer frente a los cambios en tiempo de ejecución, se establece el valor como lo hice en la Fecha de Calc_Load controlador de eventos. Pero ¿por qué establecer el ancho de formulario en tiempo de ejecución? Debido a que el tamaño real de la forma es otra variable que tiene poco control más. Varias preferencias que el usuario puede definida en Windows puede controlar el tamaño de las nuevas formas, como exigir que se abran maximizada. La propia StatusStrip cambios con el formulario. Los paneles, debido a su naturaleza estática, no lo hacen. Por lo tanto, usted tendrá que calcular el ancho de la paneles cuando una aplicación se carga, no cuando se genera el formulario.

Ahora bien, si nos fijamos en la ventana Propiedades para, por ejemplo, el cuadro de texto DateNumber, a encontrar una nueva propiedad, información sobre herramientas en primaryToolTips, como se muestra en Figura 4-12. Seleccione el valor de la propiedad, escriba Introduzca el número de días aquí, a continuación, ejecute el programa. Al pasar el puntero del ratón sobre el texto caja de texto, la información sobre herramientas famosa aparece junto al puntero, como se muestra en la Figura 4-13.

Información sobre herramientas son sólo una de muchas maneras, una aplicación se comunica con el usuario en tiempo real.



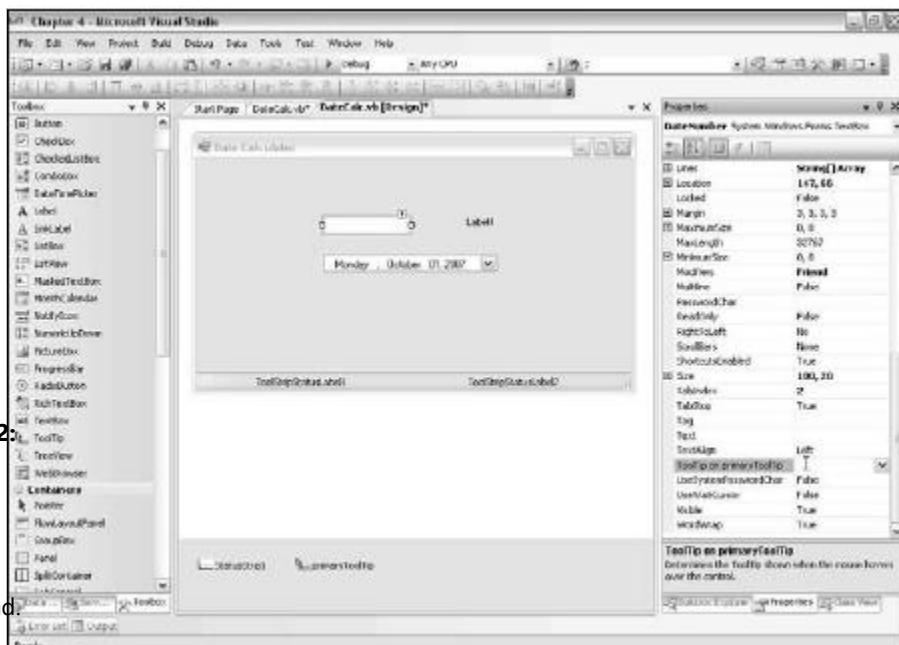


Figura 4-12:
La
ToolTip
en
primario
ToolTip
propiedad.



Figura 4-13:
La información sobre herramientas
en acción.

Algunos controles vienen con extras

Información sobre herramientas y menús contextuales se pueden arrastrar en el diseñador de la caja de herramientas, y la propiedades se pueden establecer en la ventana Propiedades mediante la selección de los objetos en la bandeja de componentes.

Echa un vistazo al objeto ToolTip, por ejemplo. El objeto ToolTip, al igual que muchos UNOB-objetos observables en el mundo de Windows Forms, proporciona funcionalidad adicional no suele disponible para un conjunto dado de controles. A diferencia de la ToolStrip objeto, que se especializa en el usuario interacción, o el objeto StatusStrip,

que se especializa en la información del usuario, el Información sobre herramientas de control de Windows Forms da controles propiedades adicionales.

Propiedades extra añadido a un objeto? ¿Cómo es que posible? El polimorfismo. Visual Basic.NET es un lenguaje orientado a objetos, y por lo tanto tiene que adherirse a cuatro reglas - los objetos deben ser heredable extensible, narrable, y polimórfico. Por lo tanto, se puede definir un objeto que redemulta las propiedades de otro, si lo desea.

Navegación con el control ToolStrip

A lo largo de la historia del software de Windows desarrollo de aplicaciones, nadie ha subido con una manera decente de hacer frente al desarrollo de menús. Visual Studio 2008 utiliza un control que pueden arrastrarse como el núcleo visual representación del menú, como se muestra en la figura 4-14.

Arrastre un control ToolStrip cualquier parte del formulario en la vista Diseño, como se muestra en Figura 4-14, y cambiarle el nombre a MainMenu. El diseñador de formularios le proporciona Escriba aquí un sistema, que le llevará a la parte clave del desarrollo de Windows Formas de aplicaciones y software con las normas aceptadas.

¿Ha notado que casi todas las aplicaciones que se ejecutan en Windows tiene un menú de la barra que muestra "Archivo, Edición, Ver, Insertar, Ventana, Ayuda" o algo por el estilo, como se muestra en la figura 4-15? Nada obliga a los desarrolladores a hacer tal barra de menús, pero lo hacen porque esta configuración es un menú aceptada standard.

Para reforzar su menú principal de acuerdo con la norma aceptada, siga estos pasos:

1. Para añadir un menú de archivo, escriba Archivo (después de añadir el control ToolStrip), como muestra en la Figura 4-14.

Aparece un mensaje a la derecha y debajo del elemento de menú sólo añadido.



elemento y el elemento ya bien llamada creará su propio controlador de eventos.

Capítulo 4: Creación de aplicaciones de Windows

Figura 4-14:
The Main
Menú en
el diseño
Ver.

Figura 4-15:
Una norma
menú del
bar.

2. Para agregar un menú Edición, haga clic en y escriba a la derecha del último elemento del menú añadido.

Para este ejemplo, sin embargo, sólo tiene que añadir una opción Salir en el menú Archivo.

Tenga en cuenta que el objeto ToolStrip añade un componente al componente Bandeja en la parte inferior de la ventana de diseño, como se muestra en la figura 4-16. A

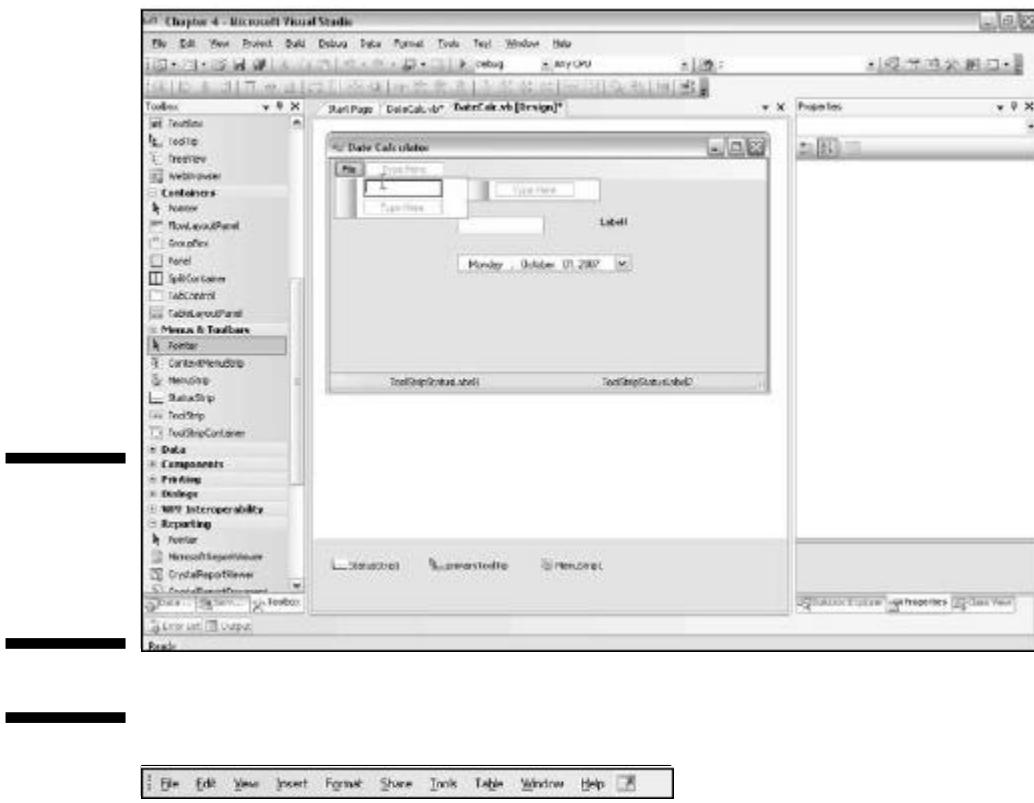
acceder a las propiedades de la instancia de menú, puede hacer clic en el control ejemplo, en lugar de los menús en la parte superior del formulario.

3. Para agregar código funcional para el menú, simplemente haga clic fuera del menú;

luego vuelve y haga clic en el menú Archivo para abrir el elemento Salir.

Muchos de los elementos del menú tienen forma de la funcionalidad, por lo que hará mucho

de uso del objeto Me. Me es un alias útiles de Visual Basic que se refiere a la objeto que es actualmente el foco de la aplicación. Haga doble clic en ese



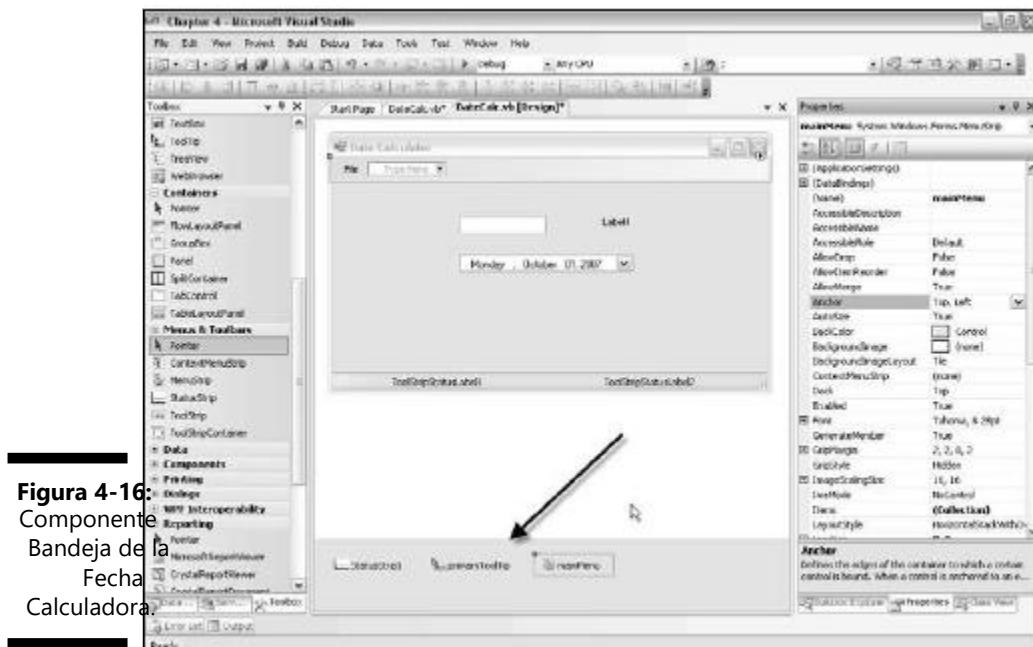


Figura 4-16:
Componente
Bandeja de la
Fecha
Calculadora2

4. Haga doble clic en el elemento de salida y añadir la Me.Close () para el de control de eventos para este artículo.

Esta es la forma en que el código de decir: "Ejecutar la función de cierre del envase objeto "- en este caso, la forma - para que se cierre la ventana cuando el elemento de menú seleccionado.

Activar el botón derecho del ratón con ContextMenuStrip

Otro tipo de menú es el menú contextual al que se accede con un clic derecho un control en una aplicación en ejecución.

Es probable que utilizar los menús contextuales constantemente, sin siquiera pensar en ello. Haga clic en una imagen en un navegador Web permite imprimir y guardar. Haga clic en una barra de desplazamiento ofrece a Down Page y Page Up opción. Usted puede proveer esta funcionalidad, como lo hice en la forma que se muestra en la Figura 4-17, con el ContextMenuStrip objeto.

Al igual que la información sobre herramientas, menús contextuales son objetos en la bandeja de componentes, y se puede asignar ciertas propiedades. Al igual que el objeto ToolStrip, puede asignar un menú contextual a la propiedad ContextMenuStrip de un formulario después de mediante el generador limpio pequeño menú en la vista Diseño.



Figura 4-17:
Un contexto
menú.

Arrastre un objeto ContextMenuStrip del cuadro de herramientas en el componente Bandeja y cambiar el nombre por defecto para primaryContextMenu. El contexto Constructor ToolStrip aparece en el diseñador de formularios, y se puede construir sólo como con el ToolStrip. Luego, en el código, se puede asignar la propiedad, al igual que con el ToolStrip. Cuando se ejecuta el formulario, el ToolStrip aparece en la la esquina superior izquierda, y el ContextMenuStrip aparece al hacer clic con el botón el formulario u otro control. Para determinar qué control obtiene qué contexto menú, añadir un poco de código en el evento Load del formulario:

```
DateNumber.ContextMenuStrip = primaryContextMenu
```

Puede usar un código como este para predefinir un par de menús contextuales y asignarlos a interfaz de usuario determinados controles basados en las opciones que el usuario necesita ver.

Capítulo 5

Creación de aplicaciones Web

En este capítulo

Entender el funcionamiento interno de ASP.NET

El conocimiento de los procesos que hacen que las aplicaciones web diferentes

Informarse acerca de las herramientas para el desarrollo Web

Creación de su primera aplicación web

La comunicación con el usuario

Web programación de aplicaciones ha cambiado drásticamente desde su origen en los años 90. Pero no importa cuánto características de programación puede han cambiado, las aplicaciones fijas de resorte de un solo concepto: Basado en un solicitud de la aplicación Web, un servidor pasa los datos a un cliente ya existente (El navegador Web), que luego hace que los datos en una interfaz que el Usuario de aplicaciones Web ve.

La adición de elementos de formulario y Common Gateway Interface (CGI) para la Web escena en 1993 aumentó la capacidad de los servidores Web a aceptar la entrada de un usuario y devolver una respuesta procesado. En los próximos 10 años, todo cambiado y sigue siendo el mismo. En la actualidad, los desarrolladores web pueden utilizar hundos de idiomas preprocesamiento en docenas de plataformas, pero todos estos instrumentos esencialmente utilizan el protocolo CGI para obtener información del navegador a la servidor y viceversa. ASP.NET - parte del Marco NET -. Es la más reciente versión del protocolo CGI original.

En este capítulo acerca de ASP.NET, que cubra la diferencia entre Windows y Aplicaciones web, y te digo cómo el Marco. NET le proporciona herramientas para simplificar la diferencia para usted como desarrollador. Explico cómo ASP.NET obras e informarle sobre los problemas que están desconectados (es decir, no tener una conexión constante entre el navegador web y el servidor Web) causas, específicamente con los problemas de las aplicaciones de Estado y la realidad de la PostBack.

Usted puede construir su primera aplicación web - una calculadora de fechas, según lo diseñado en Capítulo 3 - uso de herramientas Web y verlo en un navegador Web. Y descubres un poco sobre el inmenso poder de ASP.NET a través de los detalles acerca de la hiper-enlaces, imágenes y el objeto HttpRequest.

Al ver el funcionamiento de ASP.NET con su aplicación web

ASP.NET es un preprocesador que funciona con Internet Information Server (IIS) 5.1 o superior para servir de HTML para los navegadores Web. Lo que usted debe recordar - a lo largo de esta descripción de cómo funciona ASP.NET - es que es esencialmente CGI para la gestión de una solicitud desde el navegador al servidor, y una respuesta desde servidor al navegador. ASP.NET está integrado en el marco .NET y se utiliza para crear formularios Web Forms en Visual Studio 2008. Para su uso en la Web, ASP.NET tiene varias sobre los beneficios de Windows Forms:

Los clientes que utilizan la aplicación Web Forms no es necesario tener el .NET Framework instalado ya que la información se devuelve al navegador

que hace la solicitud, ya que sólo HTML - y como no una propiedad ASP formato.

Los clientes no tienen ni siquiera para utilizar Microsoft Windows o Internet Explorer. Se puede decir ASP.NET para representar HTML que funciona en cualquier navegador contemporánea.

Acceso complejas de procesamiento de datos o que ocurre en el servidor, que permite el navegador para residir en una estación de trabajo más simple.

Código para una aplicación ASP.NET se almacena en el servidor. Cualquier cambio para una aplicación tiene que hacerse en un solo lugar.

Por supuesto, ASP.NET tiene limitaciones, así:

El equipo cliente debe ser capaz de acceder al servidor a través de una red conexión. Esta conexión puede venir de una red local o detrás a través de un cortafuegos.

El servidor tiene poco control sobre el software que utiliza el cliente para ver la información. Web Forms diseños deben permanecer más simple que los de Windows Forms para que la mayoría de los usuarios pueden obtener una satisfacción experiencia de visualización.

Todo lo que el cliente necesita hacer es enviado en texto claro en la forma de HTML. El programador tiene que tener mucho cuidado con cómo las formas se codifican para la estabilidad y razones de seguridad.

Dos procesos fuertemente diferenciar el manejo de formularios Web Forms y Windows Forms. El primer proceso es PostBack, que es como ASP.NET controla la CGI transmisiones para la transferencia de información. El segundo proceso consiste en cómo ASP.NET administra el estado de la aplicación (es decir, la forma en que el servidor recuerda lo que el cliente está haciendo dentro de su programa).

Capítulo 5: Creación de aplicaciones Web

PostBack: No es un paquete devuelto

La devolución de datos - una invitación quasi-automática desde el navegador al servidor

-
es la magia detrás del modelo de ASP.NET. La comunicación PostBack proceso es cómo ASP.NET identifica una solicitud de la misma página con el fin de manejar una petición en la página por el usuario. Cada usuario iniciada por evento - desde escriba la dirección inicial, a hacer clic en un botón, incluso a cambio de un botón de opción selección - puede causar una devolución de datos.

Visual Studio 2008 trata a las devoluciones de datos como eventos, como un evento en un equipo con Windows
Aplicación de formularios. En el capítulo 4, se enteró de doble clic en un botón para generar el código de control de eventos para ese botón. El proceso es muy similar para formularios Web. Si está diseñando una aplicación de formularios Web Forms agregue un botón, doble clic en ese botón en la vista Diseño le consigue el código de control de eventos, también. Aunque el proceso de codificación es muy similar, el código que Visual Studio escribe para ti es diferente, y la cantidad de control que tiene es diferente.

El programador realmente no tiene control sobre la forma en que el navegador hace peticiones al servidor. Si usted ha codificado las aplicaciones web antes - usando ASP clásico o preprocesador otro - el código de devolución de datos que es automáticamente generado se sentirá muy diferente. Si la codificación de su primera aplicación Web ción, utilizando el método de ASP.NET 's automático le resultará muy fácil. De cualquier manera, vamos a Visual Studio haga su trabajo y no tratar de forzar a que el programa funcione de la manera usted está acostumbrado. Como tocar piano con un metrónomo, el marco que inicialmente parece una restricción en realidad le dan mucha libertad.

Una cuestión de Estado

Las aplicaciones Web difieren de las aplicaciones de Windows en lo que respecta a la gestión de la Estado de la aplicación. El estado de una aplicación se caracteriza por lo la aplicación sabe sobre sí mismo en cualquier momento dado. Por ejemplo, si se establecer una variable en un valor en una aplicación de Windows Forms, la variable mantiene ese valor hasta que la aplicación se cambia o se cierra (a condición de que la variable poder se declare en la sección Declaraciones del programa).

En una aplicación Web Forms, sin embargo, el momento en que el servidor termine rinda- ING una página y la envía al navegador, el servidor se olvida de cualquier aso-

desde el cliente y el servidor busca el Estado para ese usuario. La aplicación de formularios Web Forms en sí tiene ningún Estado que sea menos que el programador específicamente almacena esa variable en alguna parte.



Parte de la potencia de ASP.NET es su capacidad para guardar el estado de la aplicación en la memoria del servidor hasta que el navegador hace otra petición. En ese tiempo, el servidor recuerde la sesión de usuario que el navegador está refiriendo en la solicitud, y se llamará a los valores guardados variables.

ASP.NET ahorra diferentes elementos de la aplicación Web de diferentes maneras, como sigue:

Los valores en los controles de formularios Web - tales como cuadros de texto y las redes de datos - se guardan automáticamente a menos que específicamente pedir a los valores de no ser guardado.

Los valores de las variables no se guardan a menos que explícitamente escribir código para salvarlos e incluirlo en su aplicación Web.

Los detalles sobre el navegador que hace la solicitud se guardan cada vez, pero usted tiene que saber dónde buscar para ellos. (Estos datos se llama servidor *variables*, y usted puede encontrar más información sobre ellos en la documentación - usted no los necesitará para este capítulo.)

Devolución de datos y la gestión del Estado puede parecer confuso en teoría, pero mucho más claro después de ver en acción. Yo te mostraré cómo estos procesos relacionados con los controles y las estructuras que se utilizan para el desarrollo de las aplicaciones de formularios Web Forms.

El descubrimiento de los controles Web

ASP.NET es algo más que la suma de sus controles de formulario, pero que los controles constituyen una parte importante del total. Por ejemplo, eche un vistazo a la Figura 5-1, donde se puede ver la Calculadora de la fecha (el mismo que se encuentra en el Capítulo 3) formato de una aplicación web. Los controles, como el calendario y el texto caja, mirar lo mismo que las que se encuentran en la aplicación de Windows Forms que aparece en el capítulo 4.

La Tabla 5-1 muestra se usan con frecuencia los controles de servidor Web y sus usos principales. Además -
ción con los controles básicos estos y otros (tales como una cuadrícula de datos y un botón), Visual Basic ofrece un número de otros controles que tienen menos evidente visual impacto en una página, pero son igual de importantes en el desarrollo de aplicaciones. La Tabla 5-2 contiene una lista de categorías de estos controles menos obvias.

Tabla 5-1**Se usan con frecuencia controles de servidor Web**

Control	Herramientas del icono	Usa
Botón		Presenta una solicitud (lo que causando una devolución de datos)

Control	Herramientas del ícono	Usa
Calendario		Permite al usuario seleccionar una fecha de un calendario
RadioButtonList		Ofrece fácil acceso a una selección lista como una DDL
ImageMap		Crea una versión dinámica de la HTML clásico
FileUpload		Maneja la complejidad de la formulario multipart
Panel		Actúa como un mecanismo de intercalación para otros controles en una página

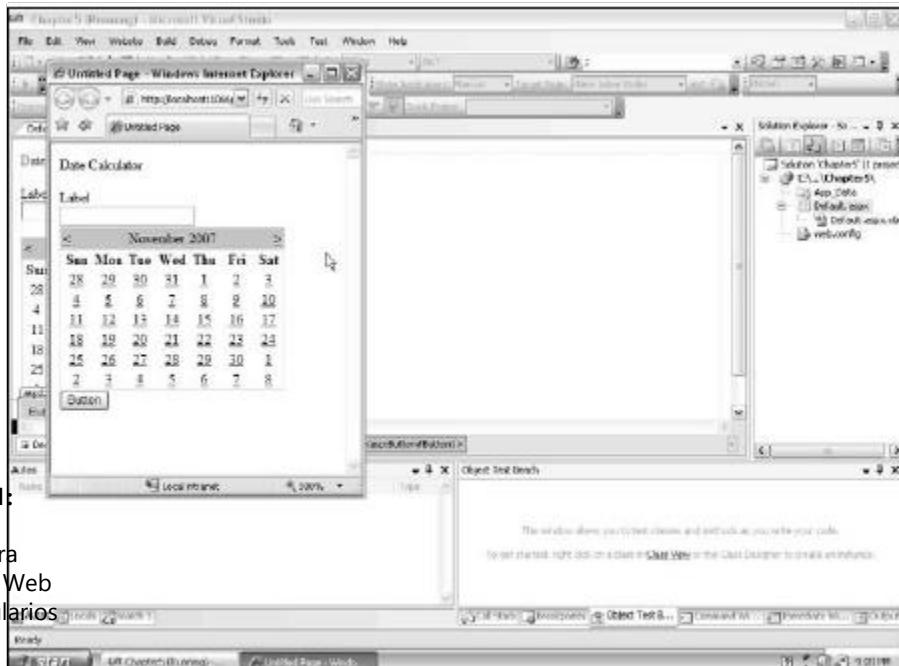


Figura 5-1:
La Fecha
Calculadora
como un Web
Formularios
aplicación.

Tabla 5-2 Otras categorías de control del servidor

Categoría de control	Usa
Los formularios Web Forms	Nivel de página controles, como el visor de Crystal Report, o Controles PDF
Datos	Datos de los proveedores de acceso, tales como las fuentes de datos y redes
Personalización	Elementos web, tales como herramientas de Business Intelligence, como des-
Seguridad	rrollado para SharePoint
Validación	Ingrera la funcionalidad que se integra con otras ventanas controles de seguridad, por ejemplo, NTFS Seguridad
Navegación	Controles de validación de entrada que producen su propio cliente código del lado del guión
HTML	Sistemas prediseñados para ir de una página a otra
	Simplemente HTML escrito previamente para facilitar el acceso

El objetivo de los desarrolladores de ASP.NET era reducir en un 70 por ciento la cantidad de código que un desarrollador Web (como usted) debe escribir. Y que le da todo esto controles de servidor para elegir va un largo camino hacia el logro de ese objetivo.

Fuera del ámbito de los controles de servidor es el espacio de nombres de formularios Web Forms que forma parte

de ASP.NET. Porque el diseño de aplicaciones Web es tan fuera del alcance normal de un esfuerzo de desarrollo regular, ASP.NET proporciona un gran número de clases para ayudar en la gestión de la aplicación. Por ejemplo, los conceptos Estado de devolución de datos y, descrito en la sección anterior, proporcionar algún-chal

fíos que ASP.NET es muy adecuado para satisfacer por esta funcionalidad.

~~Tabla 5-3 se enumeran algunas de estas clases del espacio de nombres.~~

Tabla 5-3 Algunas clases en el espacio de nombres System.Web

Clase	Usa
HttpApplication	Define las propiedades de toda la aplicación
HttpSession	Identifica propiedades de una sesión en una aplicación
HttpContext	Ofrece acceso a las propiedades específicas de HTTP de un solicitud específica
HttpBrowser Capacidades	Da acceso a la colección Server_Variables proporcionado por CGI
HttpCookie	Lee y guarda las cookies para un PC cliente
HttpRequest	Concede acceso a los valores enviados por una solicitud

Clase	Usa
HttpResponse	Proporciona acceso a los valores enviados al cliente en una respuesta
HttpUtility	Define utilidades genéricas para codificar y decodificar HTTP mensajes
HttpWriter	Permite pasar valores a un HttpResponse

La construcción de su primera aplicación web

En las secciones siguientes echar un vistazo a la construcción de la Calculadora de la fecha (como la solicitud que figura en el capítulo 3) como una aplicación Web. Esencialmente, la Red versión de la Calculadora de la fecha funciona casi de la misma manera que la de Windows Formularios versión discutida en el capítulo 4. La diferencia en la Web Forms versión es el código adicional necesario para asegurar que la aplicación se puede olvidar y volver a recordar todo entre actualizaciones de la forma. Esto es, la aplicación debe guardar Estado cada vez que el navegador devuelve la llamada al servidor en un PostBack petición.

Visualización de los extras en los formularios Web Forms

Al igual que con Windows Forms (capítulo 4), que tiene una selección de puntos de vista en la ventana del diseñador, como se muestra en la Figura 5-2. Pero a diferencia de los puntos de vista de Windows

Las formas, los formularios Web Design puntos de vista tienen el nombre, origen y Server. Al ver el diseño o la fuente, usted tiene una opción para ver juntos en una vista dividida, nuevo en 2008.

El punto de vista del servidor es para el código subyacente. Estos puntos de vista que muestran lo siguiente:

Diseño Vista: Como se puede esperar de Windows Forms, Vista de Diseño muestra formularios Web Forms en lo-que-ve-es-lo-que-hay (WYSIWYG) formato.

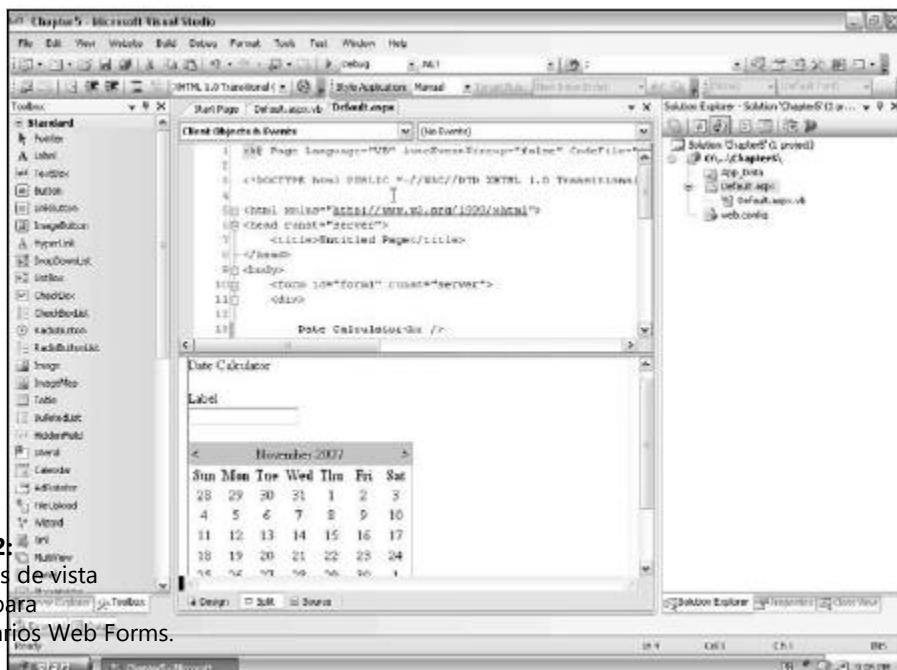
Ver código fuente: Muestra el código de visualización, que es esencialmente el código HTML que la descarga del navegador. La excepción a esto es el servidor ASP.NET controles, los cuales son prestados por IIS antes de que lleguen al navegador. Para esos controles, se ve marcado especial ASP.NET.

Servidor Vista: Muestra el código de Visual Basic 2008 que se compila y guardado en una biblioteca de clase para su uso por el servidor en el procesamiento de la

incom-
ción peticiones del navegador.



Figura 5-2:
Los puntos de vista
disponible para
Los formularios Web Forms.



Las páginas ASP.NET que acompañan a una aplicación Web son en realidad representadas por dos archivos editables. El archivo .Aspx contiene el código de presentación, es decir, el código HTML, los formatos que el material que aparece en Internet del usuario navegador. El archivo .Aspx.vb, que se llama el archivo de código subyacente, contiene la material funcional (el código de Visual Basic) que el usuario nunca ve. Usted puede mirar tanto en los archivos de la ventana del diseñador, como lo hace con los formularios Windows Forms.

La construcción de la aplicación de formularios Web Forms

Siga estos pasos para iniciar su aplicación Web Forms y rellenarla con los controles que necesitas:

1. Elija Archivo del Sitio Web \Rightarrow New en el menú principal de Visual Studio.

Los formularios Web Forms se fijan un poco diferente a Windows Forms proyectos: Aparecen como los sitios Web en lugar de proyectos. Visual Studio ofrece unas pocas opciones de la plantilla en el cuadro de diálogo Nuevo sitio Web, como se muestra en Figura 5-3.

2. Seleccione ASP.NET Web Site de las plantillas de Visual Studio instaladas lista y escriba un nombre para el sitio en el cuadro de texto correspondiente.

Capítulo 5: Creación de aplicaciones Web

Figura 5-3:
The New
Web Site
Cuadro de
diálogo.

En la Figura 5-3, observe la ubicación de la lista desplegable a la izquierda del archivo selección en la lista desplegable. Esta lista Ubicación le da la opción a publicar a un sitio de IIS o un sitio FTP. Si utiliza el sistema de archivos local - que Recomiendo durante el desarrollo - otros no serán capaz de acceder a la sitio hasta que se publica con la web Copia o despliegue otro herramienta. Búsqueda de MSDN para el término despliegue de ASP.NET para obtener más información sobre la publicación de las aplicaciones ASP.NET.

Puse el nombre de mi sitio Web DateCalcChapter5, y yo recomendamos que guarde el sitio en la ubicación predeterminada. Visual Studio crea un nuevo sitio Web de la plantilla con el nombre que especifique. El sitio incluye una página por defecto, que me fui con el nombre default.aspx. Cuando se ejecuta el sitio de Visual Studio para la prueba, se ejecutará con un servidor web personalizado especial. Si usted quiere que otros usuarios puedan ver el sitio, es necesario copiarlo en un servidor web normal.

3. Haga clic en la pestaña Diseño para ir a la vista Diseño y arrastre los controles que

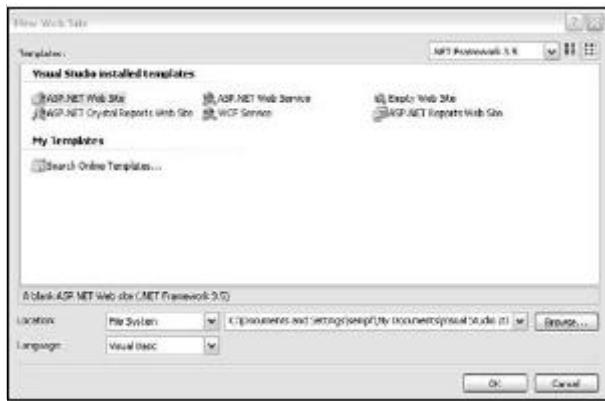
necesita de la caja de herramientas hasta el formulario.

Diseño del diseñador Web es diferente que en el diseñador de Windows. En términos generales, las páginas Web se disponen con respecto a la esquina superior izquierda esquina de la pantalla. Debido a que el desarrollador no tiene control en los tamaño de la pantalla de los usuarios (o el tamaño de fuente, o lo que sea), el diseño para una aplicación web tiene que ser mucho más flexible que un Aplicación de Windows.

En este ejemplo, se inicia con un control Label. El control Label proporciona controlada por el servidor texto en una página Web. A diferencia de Windows Forms, Web

Los formularios permiten que escriba texto estático directamente en una página Web, por lo que real-

aliado tiene dos formas diferentes de presentar el texto al usuario. Al colocar el primer control, dos características obvio de inmediato:
· El objeto en cuestión se alinea con la esquina superior izquierda de la forma, no importa donde usted lo arrastró.



- Las propiedades disponibles cuando se trabaja con un control de formularios Web Forms son muy diferentes de las de un control de formularios Windows Forms que tiene el mismo nombre.

4. Para agregar todos los controles de la aplicación Calculadora de la fecha, arrastre un control Label,

Calendario y Button al formulario.

La Figura 5-1 muestra la ubicación de estos controles en la fecha de mi muestra calculator formulario.

5. Formatear el control Calendar mediante la selección de uno de los formatos y hacer clic ING el botón OK.

Al arrastrar el control Calendar en la página, verá otra feature de las etiquetas inteligentes (que puede haber descubierto en el Windows Forms aplicación construida en el Capítulo 4) - el cuadro de diálogo Autoformato, como muestra en la Figura 5-4. El cuadro de diálogo Autoformato le da la oportunidad para implementar rápidamente una de las miradas prediseñados para una control. Para mi ejemplo, yo elijo el formato profesional 1 de la Selección una lista de Scheme. Haga su elección, y el nuevo diseño aparece en la diseñador.

6. Haga clic para colocar el cursor en la parte delantera del control Label que agregó en el superior izquierda, y pulse Enter dos veces para agregar un espacio.

7. Haga clic para colocar el cursor hacia atrás en la parte superior de la página y escriba su nombre de la aplicación.

En este ejemplo, escriba Calculadora de la fecha a nombre de mi aplicación web.

Resalta el texto y utilizar la primera lista desplegable en el formato de texto barra de herramientas para cambiar el formato de bloque a la rúbrica 1.

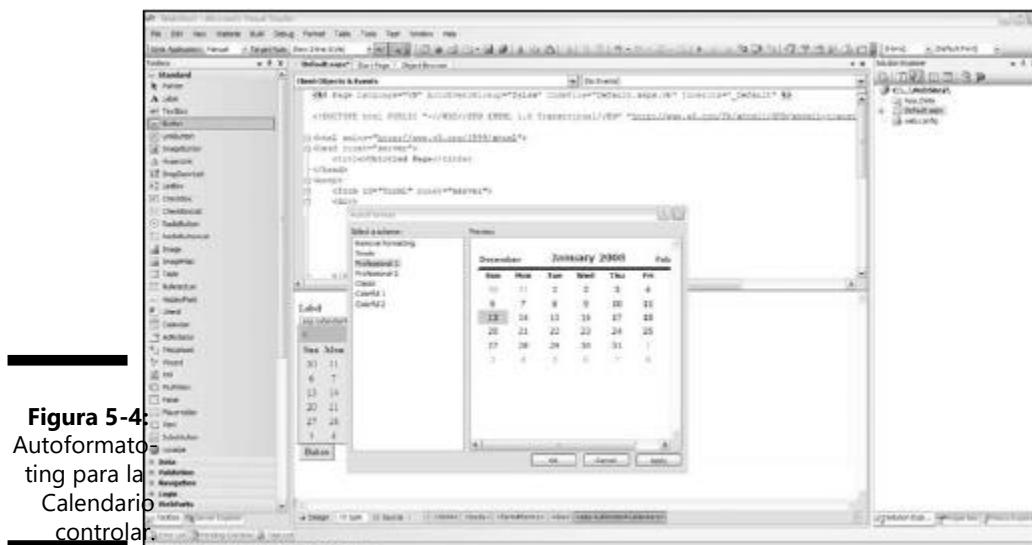


Figura 5-4:
Autoformato
ting para la
Calendario
controlar.

Capítulo 5: Creación de aplicaciones Web

8. Pulse F4 para abrir la ventana Propiedades, haga clic en cada control (el Label, Button, y el Calendario) y cambie la propiedad ID para nombres utilizables.

Yo nextweek para el sello, SubmitButton para el botón y DateChooser para el Calendario.

Al agregar los controles de formularios Web, usted ha hecho un buen comienzo en su primera aplicación web.

En Visual Studio, el desarrollo de aplicaciones Web con formularios Web Forms es muy diferente desde el desarrollo de aplicaciones de Windows con Windows Forms. Web desarrollo añade otra capa - ASP.NET - que es un tema central para la primera varias páginas de este capítulo. (Sin embargo, este es un libro acerca de Visual Basic, y no preocuparse, en este capítulo se refiere a ASP.NET VB 2008.) Si usted es completamente nuevo en el desarrollo Web, y tiene que llegar a ser muy bueno muy rápidamente, recomiendo la lectura de *ASP.NET 2 For Dummies* por Bill Hatfield (publicado por Wiley) además de terminar este capítulo.

Visualización de los resultados en vistas de origen

Para ver su trabajo en vistas de origen y también agregar alguna funcionalidad a su formulario, siga estos pasos:

1. Haga clic en la ficha Origen para cambiar a la vista Código fuente.

De repente se puede ver el diseño específico del formulario en formato HTML.

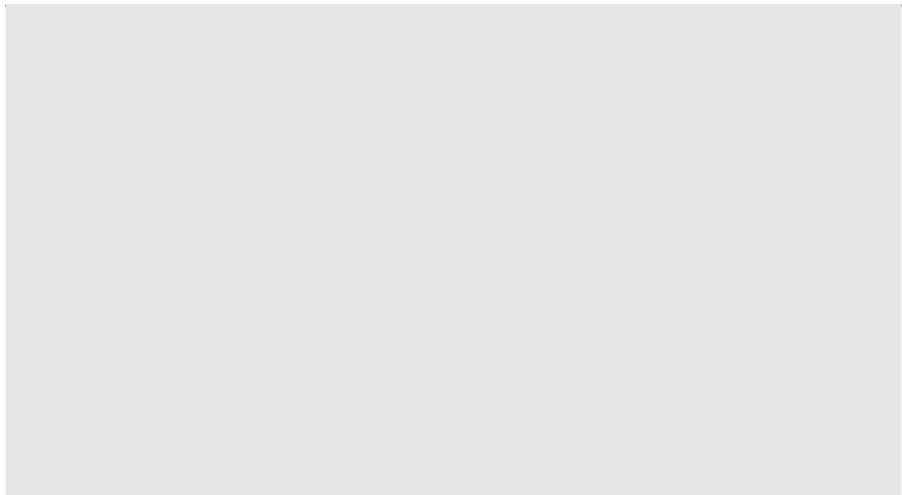
El código HTML para mi aplicación calculadora de la fecha aparece en el Listado 5-1.

Listado 5-1: El Código HTML para Default.aspx

```
<% @ Page Language = "VB" AutoEventWireup = "false" CodeFile = "Default.aspx.vb"
Inherits = "_Default"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional // EN"
 "Http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    Página <title> título </title>
</Head>
<body>
    <form runat="server"> id="Form1"
    <div>
        <asp:label ID="Label1" runat="server" Text="Label"> </asp:Label>
        <Asp: Calendar ID = "Calendar1" runat = "server" BackColor = "White"
            BorderColor = "White"
            BorderWidth = "1px" Font-Names = "Verdana" font-size = "9pt"
            ForeColor = "Negro" height = "190px"
```



Listado 5-1: (continuación)

```

NextPrevFormat = "FullMonth" width = "350px">
<SelectedDayStyle BackColor="#333399" ForeColor="White" />
<TodayDayStyle BackColor="#CCCCCC" />
<OtherMonthDayStyle ForeColor="#999999" />
<NextPrevStyle Font-Bold = "True" font-size = "8 puntos" ForeColor = "# 333333"
    VerticalAlign = "Abajo" />
<DayHeaderStyle Font-Bold="True" Font-Size="8pt" />
<TitleStyle BackColor = "White" BorderColor = "Negro" BorderWidth = "4px"
    Font-Bold = "True"
    Font-Size = "12pt" ForeColor = "# 333399" />
</ Asp: Calendar>

</ Div>
    <asp:Button ID="Button1" runat="server" /> Text="Button"
</ Form>
</ Body>
</ Html>

```



Puede trabajar con (añadir, borrar, cambiar, y así sucesivamente) los controles en este ver o en la vista Diseño - aunque usted debe saber que un error en código en la vista de origen cause un problema en la vista Diseño. Figura 5-5 muestra el mensaje resultante de un error cometido en la vista de origen.

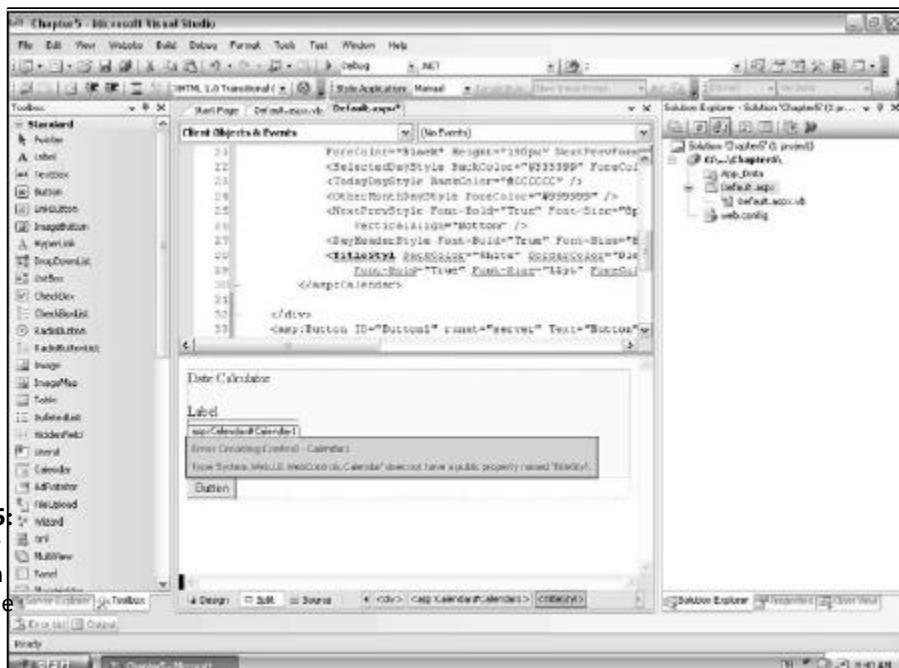


Figura 5-5:
Un Error
Creación
Controles
mensaje.

Capítulo 5: Creación de aplicaciones Web

En general, yo le aconsejaría que se centran en ser o bien un editor HTML o un editor de diseño. Si su formación es en diseño Web, utilizar y pegar con la vista de origen HTML. Si la mayor parte de su experiencia en el diseño de forma

res como Visual Basic o Delphi, se centran en el uso de la vista Diseño. Cambio entre los dos es confuso y puede causar problemas lógicos.

2. Haga clic en la pestaña Diseño para volver a la vista Diseño.

3. Haga doble clic en el control Button para agregar un controlador de eventos Click.

Cambios a Visual Studio Vista de servidor, donde usted tiene la oportunidad de añadir código a la instancia de dicho evento click. Los métodos, propiedades, y eventos del control Button aquí son un poco diferentes de las del control Button aplicación de Windows porque necesitan apoyo puerto de la devolución de datos modelo del motor ASP.NET.

Para este ejemplo, agregue código que se incrementa la fecha elegida por siete día, como sigue:

```
Partial Class _Default
    Hereda System.Web.UI.Page
    Protegido SubmitButton_Click (ByVal sender As Object, ByVal e As
        System.EventArgs) Handles SubmitButton.Click
        NextWeek.Text =
            DateChooser.SelectedDate.Add (TimeSpan.FromDays (7)).ToString ()
    End Sub
End Class
```

Después de completar estos pasos, usted tiene los fundamentos de una calculadora de la fecha aplicación. Al ejecutar la aplicación, puede hacer clic en una fecha del calendario, haga clic en el botón, hacer que un evento de devolución de datos y obtener una página web renovada que muestra la fecha a la semana por lo tanto como el resultado.

Ejecutar la aplicación Web

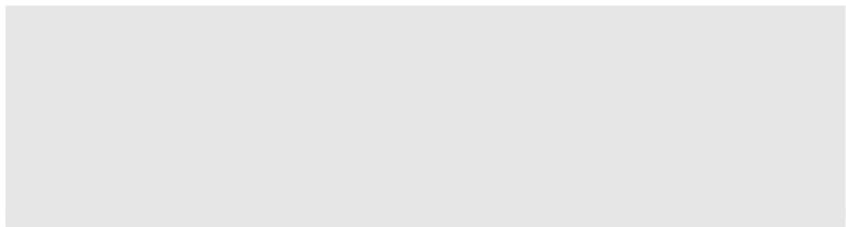
En general, Internet Information Server es necesario para ejecutar un ASP.NET Aplicaciones Web. Al desarrollar la aplicación con Visual Studio, cómo-embargo, que no es el caso. El Visual Web Developer (VWD) es un servidor web parte integral de Visual Studio 2008 y hace que el desarrollo de un no servidores plataforma sin dolor. Usted puede tomar ventaja de la estrecha integración de estos productos y probar las aplicaciones web que está en desarrollo en su localidad máquina. Con la aplicación abierta en Visual Studio, siga estos pasos para ejecutarlo en modo debug:

1. Presione F5 o haga clic en el botón Reproducir de la barra de herramientas para iniciar la Web aplicación en modo de depuración.

El modo de depuración no se configura automáticamente en un proyecto Web, por

el diálogo Depuración no habilitada caja
se muestra en la Figura 5-6.

95



2. Acepte la opción por defecto - Modificar el archivo Web.config para habilitar Depuración - y haga clic en el botón Aceptar.

Servidor VWD Web aparece en la bandeja del sistema que aloja el sitio como en la equipo local con un número de puerto aleatorio. Vea la Figura 5-7.

La aplicación se ejecuta con su página Web activa y visible en la por defecto del navegador Web se configura en las opciones de Visual Studio. (Véase el Cheat Sheet en la parte delantera de este manual para obtener información sobre la configuración de estos opciones.) Figura 5-8 muestra mi aplicación Calculadora de la fecha en abrir Internet Explorer.

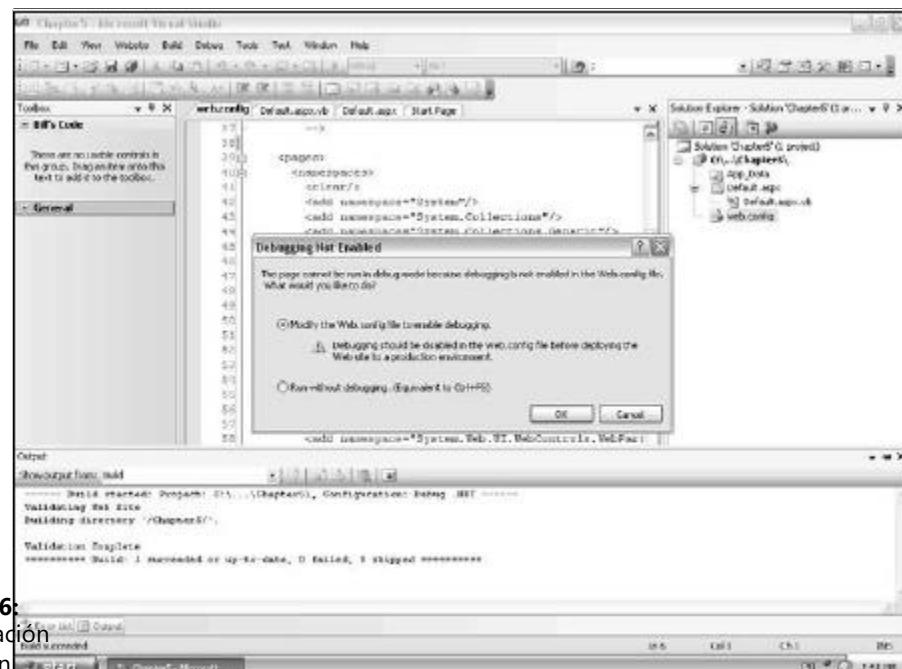


Figura 5-6:
Configuración
depuración

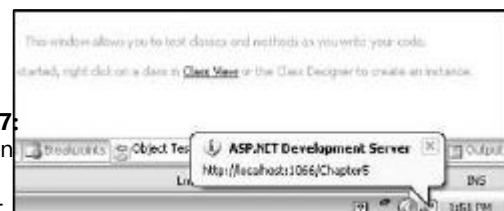


Figura 5-7:
Notificación
del VWD
Web Server.

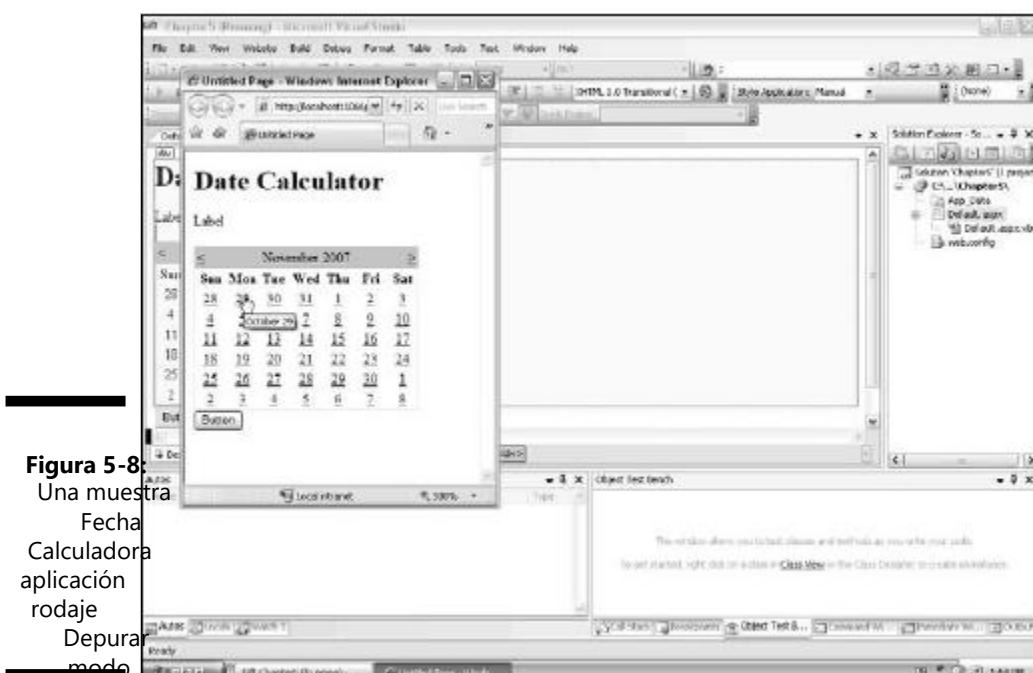


Figura 5-8:
Una muestra
Fecha
Calculadora
aplicación
rodaje
Depuración
modo.

3. Recibe en tu aplicación web (haciendo clic en un botón, escribir en un texto caja, o de otra manera interactuar con el programa) y ver el resultados.

En mi solicitud, haga clic en una fecha (29 de octubre, para ser exactos) y haga clic en el botón, y el calendario cambia para resaltar la fecha una semana fuera (Recuerden, recuerden, el cinco de noviembre), como se muestra en la Figura 5-9.

4. Cierre el explorador Web que se ejecuta la aplicación.

Visual Studio sale del modo de depuración, y el servidor Web VWD también se cierra.

Mirando bajo la superficie de formularios Web Forms

ASP.NET es un marco dentro de un marco - un servidor Web completo sistema de gestión previsto de forma gratuita como parte de .NET Framework desde Microsoft. ASP.NET es increíblemente sofisticado y poderoso, y lo hace mucho más de lo que puede cubrir aquí. Pero te puedo ayudar con una importante comprensión básica comprensión de cómo ASP.NET encapsula el CGI (Common Gateway Interface) funcionalidad que ha sido de alrededor de diez años.

Figura 5-9:
Mi muestra
Fecha
Calculadora
destacando
la nueva fecha
después de la
PostBack.

Validar entrada de usuario

Controles de entrada de usuario en los formularios web se parecen a los controles de entrada de usuario en Windows Forms. Cuadros de texto, listas desplegables, casillas de verificación, y así sucesivamente todos aceptar la entrada del usuario, y los botones de presentación de dicha información a la aplicación. La diferencia entre la Web y Windows Forms viene de cómo la formas de manejar la información de usuario bajo el capó.

Desde la perspectiva del desarrollo, cuadros de texto y otros controles de trabajo similitud particularmente para aceptar la entrada del usuario. Por ejemplo, se puede obtener el valor presentado por el usuario mediante la propiedad Text de un cuadro de texto o el SelectedValue propiedad de una lista desplegable. Una cuestión de desarrollo que difiere significativamente entre aplicaciones Web y Windows es la validación de la entrada del usuario. Debido a que (para aplicaciones Web), el cliente está separado del servidor, asegurándose que las solicitudes de los clientes tienen el formato correcto es algo que los desarrolladores queremos hacer en el cliente en lugar de en el servidor. ASP.NET permite verificar formato sencillo con los controles de validación disponibles en Visual Studio 2008.

Usted puede encontrar los controles de validación de Visual Studio, como se muestra en la Figura 5-10, en la caja de herramientas en la sección de Validación (vaya usted a saber!). Usuarios de aplicaciones Web puede olvidarse de proporcionar todos los datos que su aplicación necesita para funcionar correctamente;

también pueden escribir incorrectamente una entrada o entrar en el tipo de datos incorrecto (por ejemplo, introducir texto en un campo donde se espera un número).



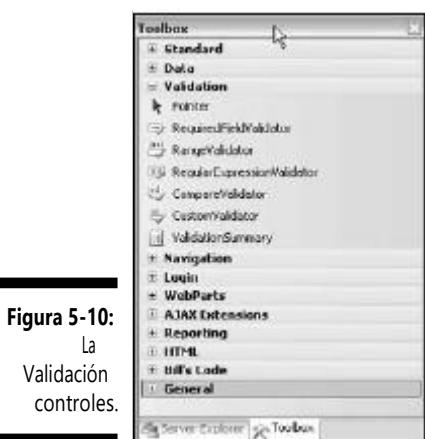


Figura 5-10:
La
Validación
controles.

Las razones más frecuentes que desea validar los datos son los siguientes:

Confirmación de que la información requerida cumple con las normas de la aplicación:

Por ejemplo, la comprobación para asegurarse de que una fecha introducida después de las caídas la fecha actual, si eso es lo que su aplicación requiere.

Verificación del tipo de datos introducidos: Asegurarse de que la fecha tiene el formato

correctamente o que se introduce un número en un campo numérico.

Para utilizar los controles de validación en la aplicación, sólo tiene que seleccionar las instrucciones en pantalla elemento que requiere la validación y arrastre el control (o controles) en la página que está diseñando. Incluyo un RequiredFieldValidator y Control ValidationSummary en mi página de Calculadora de la fecha. El requerida FieldValidator acepta un control a "ver" como parámetro, y reacciona si los requisitos establecidos para ese control no se cumplen. El ValidationSummary se encuentra en la parte superior de la página y ofrece una de esas listas con viñetas de buenas problemas, sin ningún código!

Arrastre un cuadro de texto en la página, seleccionar y cambiar el nombre de la propiedad ID para DateSpan. Puede utilizar este cuadro de texto para hacer lo mismo que el cuadro de texto

el proyecto de formularios Windows Forms (véase el Capítulo 4). En este cuadro de texto, el usuario especifica el número de días a calcular a partir de la fecha seleccionada en el calendario. La figura 5-11 muestra cómo Ver mi aplicación Web Design cuida de añadir los controles RequiredFieldValidator ValidationSummary y (Por que yo pongo el ID a DateSpanValidator y DateSummary). Puedo establecer esta validación para buscar a los usuarios introducir un número y enviarlos advertencia mensajes si no lo hacen.

Figura 5-11:

Mi aplicación Vista Diseño después de poner en el texto caja y Validación controles



Que tiene que hacer tres cosas importantes para establecer un RequiredFieldValidator controlar. Usted puede hacer todas estas cosas en el panel de propiedades con el validador seleccionado en la vista Diseño:

1. Escriba el mensaje de error.

Este mensaje es el que aparece en el control ValidationSummary cuando el usuario no ve llenado en el campo. Puse mi mensaje de error "es DateSpan Requerido".

2. Establezca el parámetro de texto.

Esto es lo que el propio validador muestra cuando se activa. Yo suelo usar un asterisco (*).

3. Ajuste el control para validar los parámetros.

Esta opción muestra el control que el validador está mirando. En este caso, el control es el cuadro de texto DateSpan.

Cuando ejecuto mi programa Calculadora de la fecha y tratar de cambiar la fecha sin introduciendo un número en el cuadro de texto DateSpan, la validación de entrada de usuario que estas medidas puestas en marcha muestra el mensaje de error que se muestra en la Figura 5-12.

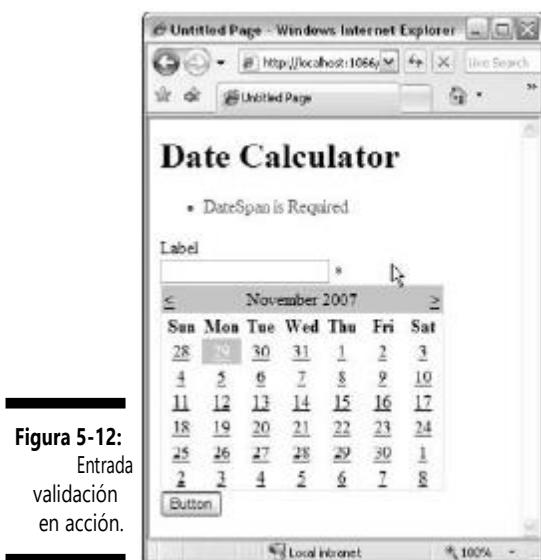


Figura 5-12:
Entrada
validación
en acción.

Lidiar con el Estado

Como menciono en la introducción de este capítulo, el Estado de una aplicación es descrito por el valor actual de los controles, variables y propiedades de objeto en cualquier momento dado. Cuando el servidor detiene el procesamiento de una página y la envía a la navegador, el servidor se da por vencido (olvida) casi todos los elementos que componen del Estado. ASP.NET proporciona algunos mecanismos para preservar Estado en un Web Forms. El más útil de estos son el ViewState y el Los objetos de sesión, que describo en las siguientes secciones.

ViewState

ViewState es un concepto nuevo para ASP.NET. En pocas palabras, el servidor de paquetes los valores de los controles de formulario que se pasaron a él, los comprime, y los guarda en un campo de formulario oculto que se representa en la información que se envía de vuelta al navegador. El usuario no sabe (o atención) que la variable está ahí, pero el servidor seguro sabe (y cuida). Esta variable forma oculta se pasa de vuelta al servidor cuando el usuario hace otra PostBack (al pedir la misma página de nuevo), y el servidor puede descomprimir la variable para saber los valores que se establecen cuando se devuelve la página al cliente.

Conocer los detalles de trabajo es menos importante que saber cómo utilizar el ViewState objeto. Además de los valores de controles de formulario, puede guardar noncontrol valores de las variables en el ViewState también. Si el programa necesita

para recordar una variable de devolución de datos para devolución de datos, puede hacerlo mediante el ahorro

y a continuación, recuperar la variable a la ViewState, como se muestra por el siguiente dos líneas de código:

```
Me.ViewState ("_NextWeek") = NextWeek.Text  
NextWeek.Text = CStr (Me.ViewState ("_NextWeek"))
```

En este sentido, el objeto de mí es sólo un atajo que hace referencia al objeto actual, que en este caso es la página Web en general. Usted ve esta estructura que se utiliza un mucho más en la Parte IV de este libro. La conversión de cadenas CStr aparece en este código, porque cuando un valor vuelve del ViewState, es sólo una *objeto tipo* y el código debe indicar a Visual Basic que el valor tipo es un String. (Para más información sobre los tipos, consulte el Capítulo 9.)

Sesión de objeto

El objeto ViewState es ideal para guardar una variable dentro de una página, pero lo que sucede si usted necesita para guardar un valor en varias páginas? Cuando un usuario cambia en las páginas, el objeto ViewState se pierde. Esta situación tiene sentido si se piensa en ello ya que su código se refiere al objeto ViewState con Mí, y que me es la página. Si el usuario se mueve a una página diferente, el programa tiene una instancia diferente de ViewState.

Así que para hacerse cargo de este problema, ejecute el objeto Session. La Sesión objeto representa un usuario en particular trabajar con una aplicación en particular. Mientras que el objeto Session dispone de varios métodos (que se puede ver si usted revisar el IntelliSense), el método importante saber lo ayuda a ahorrar valores - al igual que lo hace con objeto ViewState. Pero con la Sesión objeto, estos valores se mantengan alrededor hasta que el usuario deje de usar la aplicación, incluso si ese uso implica varias páginas.

Las siguientes dos líneas de código muestran que establecer y obtener valores de las variables con el objeto Session es tan fácil como utilizar el objeto ViewState:

```
Session ("_NextWeek") = NextWeek.Text  
NextWeek.Text = CStr (Session ("_NextWeek"))
```

Control de salida de algunos trucos frescos de la tela

La web tiene un montón de juguetes. Debido a que el Internet es una tecnología desconectado, y muchos de los juguetes son de plataforma cruzada, son una especie de fresco. Algunos de ellos incluso se espera - más o menos - por los usuarios, y algunos de ellos vale la pena conocer.

Este ejemplo muestra una instrucción If-Then-Else utilizada para manejar la navegación decisión. (Cubro con la instrucción decisión If-Then-Else en el capítulo 10.) Usted puede utilizar esta técnica de codificación de navegación del sitio (como se muestra) o incluso

Capítulo 5: Creación de aplicaciones Web

¿Cómo salir de una página a otra

Una de las características más importantes de una aplicación Web permite a los usuarios

para pasar de una página Web a otra. Si se acaba de crear directamente navegación, se puede utilizar una etiqueta de anclaje sencillo y ni siquiera se involucran con Visual Basic. La siguiente línea de código se muestra cómo se ve una etiqueta de anclaje en

la vista de origen, y la Figura 5-13 muestra cómo el enlace codificado aparece en la navegador.

```
  HREF="NextPage.aspx"> <A Esto va a la siguiente página </ A> y esto no es así.
```

Figura 5-13:

Un hipervínculo causada por un ancla tag.

ASP.NET se involucra cuando se necesita establecer el valor de la propiedad HREF de la etiqueta de anclaje en el archivo. vb (el archivo que contiene la aplicación Código de Visual Basic). Supongamos que usted necesita escoger la página que desea que el

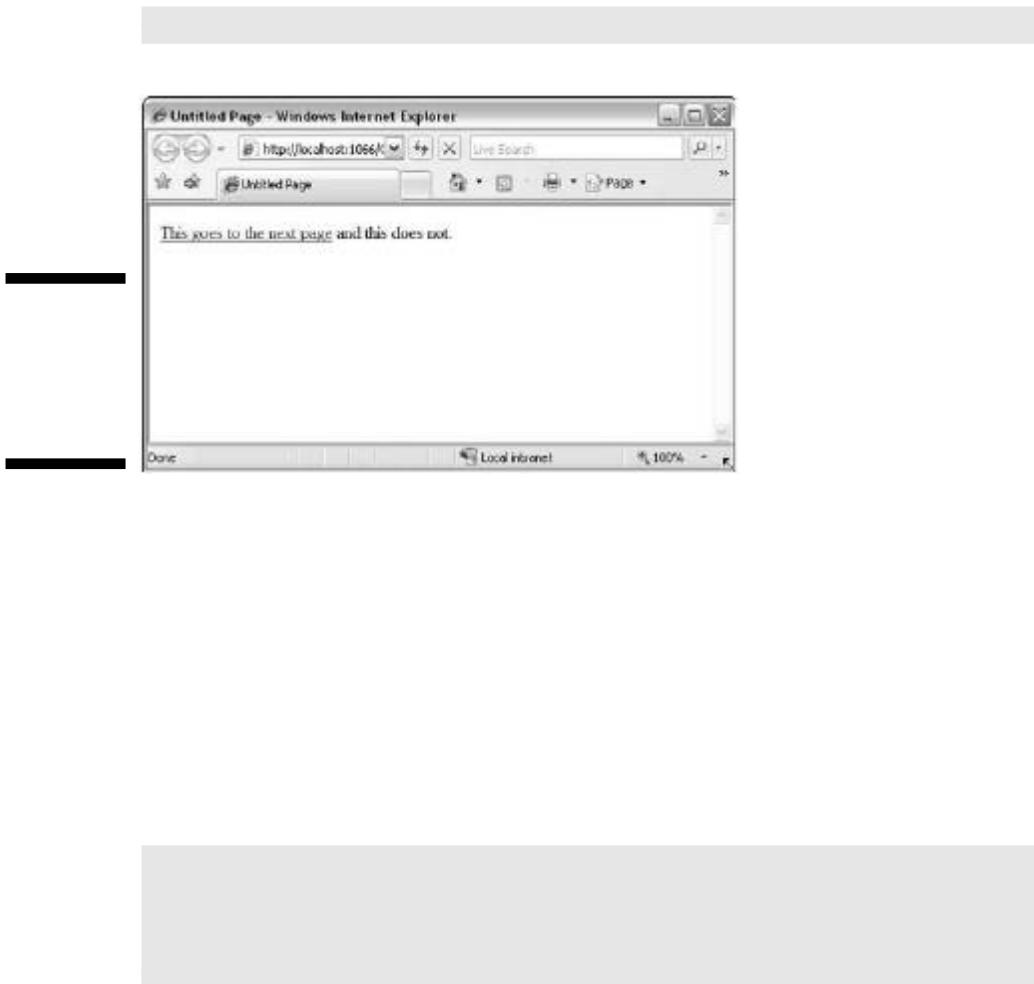
usuario alcanzar durante una devolución de datos. En Visual Basic, puede utilizar un hipervínculo

Control de usuario Web y establecer el valor de la propiedad en el NavigationUrl CodeBehind según sea necesario. Lo que sigue es un ejemplo de lo que el código se vería

como si usted deseaba establecer el HREF basado en aportaciones obtenidas por parte del usuario. En la

caso de la aplicación Calculadora de la fecha (que se verá en la sección "Creación de una aplicación Web En primer lugar," anteriormente en este capítulo), la DateSpan control (cuadro de texto) recupera el número de días para extenderse desde el usuario.

```
Si DateSpan.Text > "7" Entonces
  HyperLink1.NavigateUrl = "thispage.aspx"
Más
  HyperLink1.NavigateUrl = "thatpage.aspx"
End If
```



para la seguridad. Por ejemplo, puede buscar un ID introducido por un usuario en una base de datos-base que coincide con el ID de usuario con una lista de páginas web (URL) que el usuario está permitido el acceso.

Adición pretties

Los sitios web no son sitios web sin imágenes. Al igual que las anclas, las imágenes pueden ser manejadas sólo con HTML normal, con una etiqueta de imagen que se parece a esto:

```

```

Esta etiqueta se refiere a una imagen para Web (normalmente un archivo GIF, JPEG o PNG), que es en el mismo directorio que el archivo de código HTML referencia a la misma. El camino puede, por supuesto, cambiar en el atributo src para punto a otro directorio en el proyecto.



Tenga en cuenta que todas las rutas son relativas a la raíz del proyecto. Si siempre referencia a la ubicación de las imágenes con una ruta completa desde la raíz de la proyecto - por ejemplo, con src = "/images/navegación/imagen.gif" - no importa donde el código se utiliza, el navegador será capaz de encontrar su imagen.

Al igual que la propiedad NavigateUrl del control Hyperlink cambia el HREF atributo de una etiqueta de anclaje que lo hace, la propiedad ImageUrl de el objeto de la imagen cambia el atributo src de la etiqueta img que presta. Así pues, si arrastra un objeto de imagen a una página Web que está diseñando y desea que hacer que la etiqueta como se muestra en la línea de código anterior, podría escribir el después de agregar el archivo VB CodeBehind:

```
Sub Page_Load (ByVal sender As Object, ByVal como System.EventArgs) Handles  
    Me.Load  
    Image1.ImageUrl = "image1.gif"  
    Image1.Height = 100  
    Image1.Width = 100  
End Sub
```

Obtención de información sobre el usuario

En la sección anterior "Descubriendo los controles Web," discutir algunos controles que dar aplicaciones basadas en servidor Web de acceso al entorno del navegador. A veces la información del navegador es muy útil.

Tome los controles de seguridad, por ejemplo. Como parte de su solicitud al servidor, el navegador envía información sobre el usuario en la forma de un objeto llamado una WindowsIdentity. Recuerde, ya que ASP.NET está desconectada, el servidor no sabe qué usuario está realizando una solicitud, en cualquier momento dado a menos

Capítulo 5: Creación de aplicaciones Web

comprueba cada petición. Usted puede utilizar el objeto `HttpRequest` pasar a su aplicación del servidor para obtener un objeto `WindowsIdentity` y luego comprobar el objeto de la información del usuario, incluyendo el nombre de usuario.

El código siguiente es la vista de origen de una página de formularios Web Forms que contiene un poco texto y una etiqueta:

```
<% @ Page Language = "VB" AutoEventWireup = "false" CompileWith = "ThisPage.aspx.vb"
   ClassName = "ThisPage_aspx"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "Http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  Página <title> título </title>
</ Head>
<body>
  <form runat="server" id="Form1">
    <div>
      El usuario actual es <asp:label ID=ThisUserNameLabel runat=server> </asp: Label>
    </ Div>
  </ Form>
</ Body>
</ Html>
```

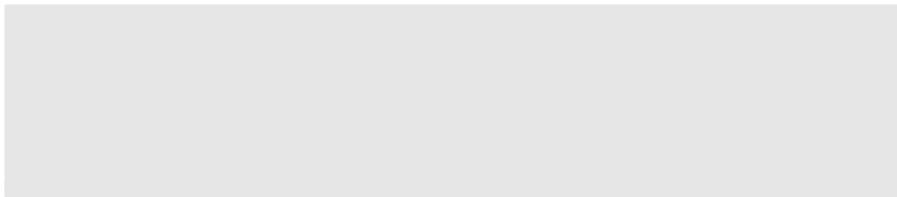
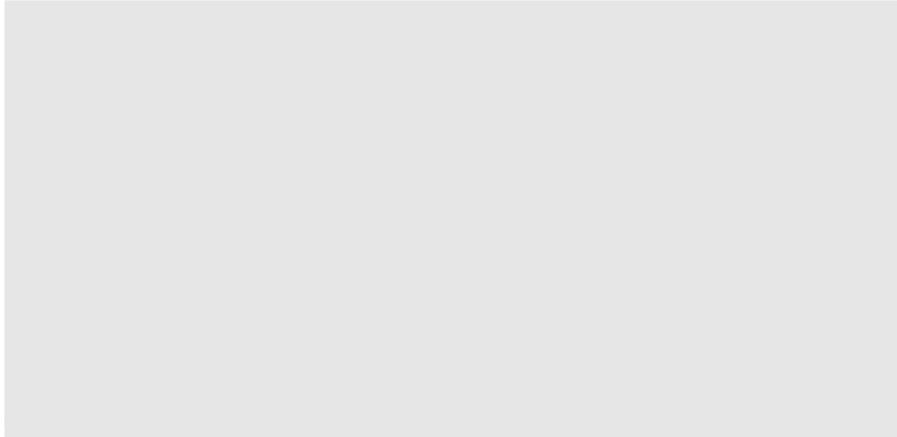
Al hacer doble clic en el formulario en la vista Diseño, se obtiene el `Page.Load` controlador de eventos (ver el siguiente código), donde se puede añadir el código de VB que permite a los objetos de acceso a aplicaciones de entorno web. Este código obtiene una copia del objeto `WindowsIdentity` del objeto `Request` y luego se hace la Nombre de la propiedad `WindowsIdentity`.

```
Sub Page_Load (ByVal sender As Object, ByVal como System.EventArgs)
  Manijas Me.Load
  ThisUser débil como System.Security.Principal.WindowsIdentity =
    Request.LogonUserIdentity ()
  ThisUserName Dim As String thisUser.Name.ToString () = ""
  ThisUserNameLabel.Text = thisUserName
End Sub
```

Este método sigue el patrón general para la obtención de información de la `http` objetos. Por lo general, estos objetos regresan algún tipo de subobjeto que necesidad de declarar en su solicitud. Para obtener más información sobre cómo obtener información de objetos, consulte el Capítulo 13.

Puede utilizar la terminología Solicitud para referirse a la actual `HttpRequest` objeción, puesto que se trata de un artefacto de versiones anteriores de ASP.

Suponga que su aplicación ha recibido una solicitud del usuario, conseguido el objeto `Request` disparado, y se extrae una copia de su `LogonUser` Identidad, que es una instancia de un objeto `WindowsIdentity`. El inicio de sesión



texto de un control Web Label (ThisUserNameLabel.Text thisUserName =).

Al crear y ejecutar esta aplicación de formularios Web Forms, verá una pantalla que (Esperemos) tiene su máquina y usuario, en lugar de la mía, como se muestra en Figura 5-14!

El objeto HttpRequest ofrece mucho más que la simple seguridad, pasa información sobre cookies, la colección de encabezados, PC del cliente, y así sucesivamente. Para más detalles sobre lo que este objeto proporciona, puede prestar atención a mi continua estímulo para leer ASP.NET 3.5 For Dummies de Ken Cox (por Wiley Publishing, Inc.) y la documentación de MSDN.

Figura 5-14:

La
Nombre de usuario
en la Web
navegador.

Otras cosas buenas de desarrollo web para saber

Usted necesita saber más detalles para escribir de

mirar a lo siguiente:

manera efectiva

Aplicaciones ASP.NET. La información sobre HTML, scripting del lado del cliente y la imagen de desarrollo

viene a la mente. ASP.NET - un servidor de preproceso-

plataforma ing - es sólo una parte del todo, y tecnologías mucho más prácticas y entrar en haciendo una aplicación web bien. En este capítulo se

le ayudará a comenzar con ASP.NET, y este libro se ocupa de los detalles de idioma CodeBehind VB lo que necesita saber. Pero usted todavía tiene una oportu-

dad de recoger mucha más información sobre la funcionamiento de las aplicaciones web antes de quedar

un experto en Internet de aplicaciones de diseño.

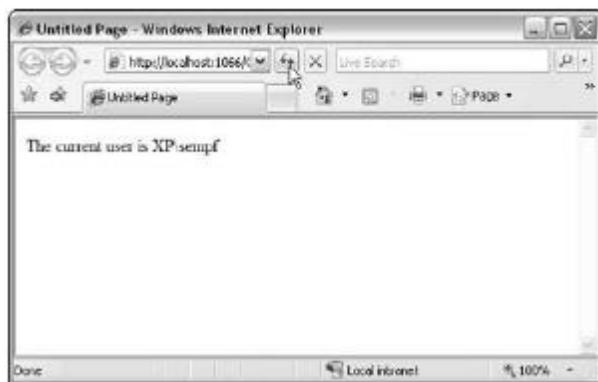
Tome un

Elementos de HTML, como tablas y listas,
que le ayudan a diseñar páginas

El impacto del procesamiento de imágenes - incluyendo
en convertir archivos al formato GIF, PNG o JPEG -
en la usabilidad de la página Web

Lenguajes de scripting como JavaScript, para
proporcionar la funcionalidad interactiva en la
cliente que no se disponga otra cosa en
ASP.NET

Configuración y administración de un sitio Web
Servidor de Información de Internet específicamente
Servidor, para ayudar a mantener el ambiente en
sus aplicaciones



Capítulo 6

La construcción de bibliotecas de clases

En este capítulo

Comprender la historia de DLLs

La construcción de una biblioteca de clases

Usando una biblioteca de clase para aislar la lógica del programa

Obtención de información dentro y fuera de los objetos

T

o decir que las bibliotecas de clases es importante entender sería un bruto eufemismo. (Una biblioteca de clases es una colección de código reutilizable organizado en grupos.) La totalidad del Marco .NET es una biblioteca de clases. Todos los grupos de controles discutidos en los capítulos 4 y 5 son bibliotecas de clases. De hecho, todos los de Windows es un conjunto de bibliotecas de clases, incluso antes .NET.

Cubro un montón de detalles sobre un tema muy complicado en este capítulo. Describo cómo usar una biblioteca de clases para encapsular la lógica, ahorrar en el uso de memoria y compartir código entre aplicaciones. Usted debe entender las partes del biblioteca de clases y las partes de la propia clase después de leer este capítulo.

En este capítulo, se construye una clase simple que se encarga de la lógica de la Fecha Calculadora diseñado en el capítulo 3. La clase se construye aquí podría ser parte de una clase más grande biblioteca (bibliotecas suelen tener múltiples clases). Discutir la diferencia entre una clase y un objeto. Además, hablo sobre algunos de los in-características de profundidad de bibliotecas de clases en este capítulo.



Espero que las clases y bibliotecas de clases se convertirá en una parte importante de su desarrollo de patrones. Usted puede, por supuesto, escribir software perfectamente funcional sin crear bibliotecas de clases, pero que realmente no debería. Usted debe diseñar su primera aplicación para que pueda decidir si el software no tiene código reutilizable o cualquier cosa que pueda ser por componentes antes de dispensar con las bibliotecas de clases. Incluso si usted acaba de construir en el proyecto y hacer no crear archivos DLL independientes, como se describe en este capítulo, usted debe usar la clase bibliotecas para la encapsulación de código. Código de encapsulación y reutilización (incluida en detalle en el Capítulo 12) hace el código mucho más fácil de mantener y fácil de construir otro software con - incluso se puede volver a utilizar el mismo código en Windows y aplicaciones web de esta manera.

Pasado y Presente: DLLs Definido

Inflamación hacia atrás para noviembre de 1985. Microsoft acaba de lanzar Windows 1.0, basado en el trabajo de Xerox y Apple. En ese momento, Windows era realmente sólo una aplicación monolítica para MS-DOS 2,0, lo que significa que todo el código de la aplicación se compila en un archivo ejecutable. Cuando se desea utilizar Windows, se ejecutó el programa. Cuando se cerró, se volvió a DOS, y el ordenador olvidó todo.

Aplicaciones de Windows 1.0, en suma, se requiere más memoria que la dura-Ware fue capaz de darles. Para obtener todos los elementos en su lugar, mantenerse dentro del restricciones de memoria, y dar a los usuarios una experiencia sin fisuras, Microsoft construyó su software en las piezas componentes. Cuando Windows se necesita un determinado carácter, se cargan sólo la pieza que se necesita, como se muestra en la Figura 6-1. Estas piezas se denominan bibliotecas de vínculos dinámicos, o DLL.

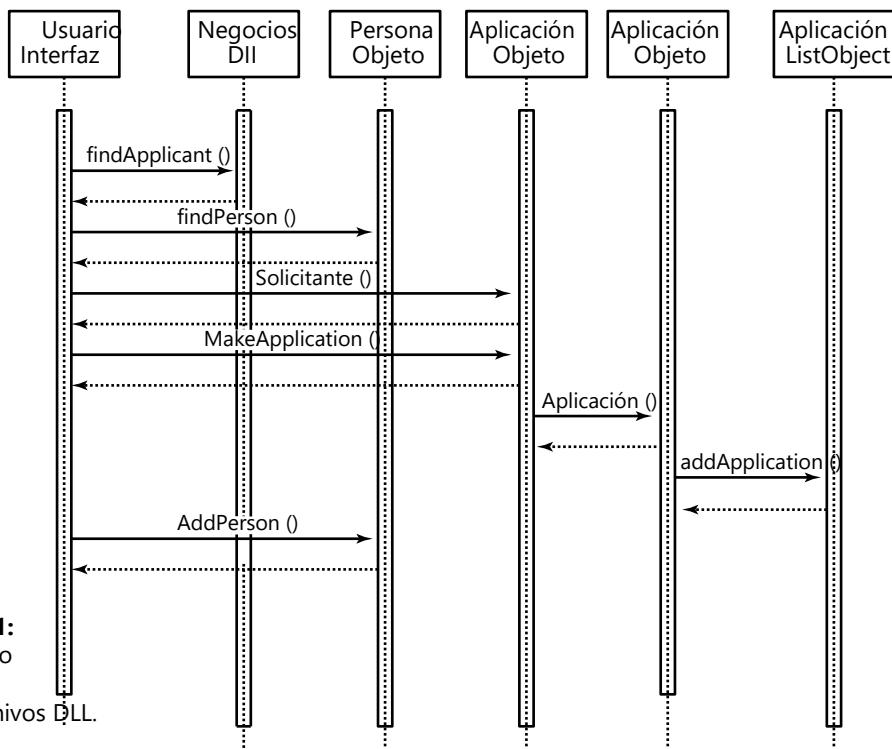


Figura 6-1:
Cómo
Ventanas
utiliza archivos DLL.

Capítulo 6: Construcción de Bibliotecas de Clase

Al final resultó que, los archivos DLL son buenos para más de la gestión de memoria. Ellos son bueno para la encapsulación de funcionalidad. La encapsulación es la localización de los conocimiento dentro de un módulo, y es uno de los inquilinos centrales de orientado a objetos de programación, así como una característica útil para tener al escribir un operativo sistema. Cuando tenga que actualizar una función, es necesario sustituir sólo el DLL, no todo el programa. Si una función particular debe ser seguro, isolando el DLL ayuda a garantizar el funcionamiento. Si tiene varios programas que utilizar la misma función, pueden compartir un archivo DLL.

Tenga en cuenta que las DLL no son programas ejecutables - no se puede hacer doble clic en ellos y ejecutarlos. Están diseñados para hacer referencia a otros programas que son ejecutable y proporcionar funcionalidad adicional. En el mundo de la programación, es común tener un programa de Windows o Web Forms en Visual Studio a lo largo de lado un proyecto de biblioteca de clase en la misma solución. La biblioteca de clases está proporcionando necesarias funciones compartidas en el programa de Windows o Web.

Una DLL es sólo un tipo particular de biblioteca de clases que es específico para el Windows del sistema operativo. Usted puede construir bibliotecas de clases para otras plataformas. Cuando se construir una biblioteca de clases para la plataforma Windows, el resultado final es un archivo DLL.

Como resultado, la cosa DLL atascado. DLL se utiliza para construir todo de la Oficina, temporales de Windows, el .NET Framework -. todo más o menos en el El mundo de Windows. DLL son la mejor manera de hacer software para Windows plataforma.

Diseño de una Biblioteca

Una biblioteca de clases en una computadora Windows es un componente de un programa y es implementado como un archivo DLL, como se describe en la sección anterior. El siguientes secciones que cubren cómo y por qué se puede construir una biblioteca de clases.

En el mundo .NET, bibliotecas de clases se utilizan para encapsular funcionalidad. Para ejemplo, tomar la Calculadora de la fecha en que discuto en el Capítulo 3. Esta aplicación tiene algunas funciones que implica la adición de un número de días a una fecha. (Esta funcionalidad no tiene nada que ver con la interfaz de usuario, que está cubierto en Los capítulos 4 y 5. No importa lo que la interfaz de usuario se parece, el programa cambia la fecha de la misma manera.)

Idealmente, usted escribe el código que hace funcionar el programa, o la lógica de

como referencia en el proyecto de la calculadora.
Esto separa la lógica y la interfaz de usuario y trae todos los beneficios que
hablamos en la sección anterior de este capítulo.



Esto es un ejemplo de funcionalidad que se puede encapsular. Aunque es obviamente un ejemplo sencillo, un programa bien estructurado de Windows - ya sea un Windows Forms o Web Forms Application - debe encapsular esta funcionalidad en una biblioteca de clases. Una aplicación que utiliza una referencia de biblioteca de clases que biblioteca como parte del código. A continuación, utiliza las funciones y las subrutinas de la biblioteca de clases como si fueran parte del programa original.

Objetos y clases

Así que tiene una biblioteca de clases, y las clases son en el mismo, como era de esperar de un biblioteca de clases. Usted espera que, debido a que es una biblioteca, puede revisar la clases dentro como usted echa un vistazo a los libros de una biblioteca regular, y usted puede hacer exactamente eso. La diferencia es que cuando se echa un vistazo a una clase de una biblioteca de clases, se obtiene una copia de la clase, por lo que la clase original permanece en la biblioteca. Esta copia se llama un objeto, y que es una instancia de esa clase.

Las clases son moldes más o menos igual. Tienen agujeros en los que ponen a la información. Cuando se obtiene información en conjunto, se puede obtener una instancia de la clase - un objeto - para almacenar la información. Tomar el examen-Date Calculator PLE. Se puede definir una clase, llamada DateCalcClass, que tiene dos propiedades y una subrutina. En tiempo de diseño, estas propiedades están vacíos, y la subrutina es sólo una herramienta. Al crear una instancia de la clase en otro programa, sin Alguna vez, se convierte en un buque activo.

El programa puede poner las cosas en el objeto, ya que es una de tres dimensiones buque, mientras que la clase era sólo un molde de dos dimensiones. Cuando el usuario se establece la fecha inicial - la primera propiedad - Calcular la subrutina llama. Que establece la segunda propiedad de la respuesta, que luego se puede utilizar en otras partes de la aplicación. La ventaja es que después de que haya terminado, usted puede eliminar esta lógica de la memoria del ordenador, y por supuesto, si usted necesita el lógica en otras partes de la aplicación, no es necesario que lo vuelva a escribir, que acaba de agregar un llamado a la biblioteca de clases.

Las partes de una biblioteca de clases

Desde la perspectiva del desarrollo, la biblioteca de clases comienza con un archivo, simplemente al igual que todos los otros proyectos. En el caso de VB 2008, que es un archivo. Vb que contiene el siguiente:

Clases: La descripción formal de un objeto

Los espacios de nombres: Un contenedor de extracto de clases, en lugar de una clase biblioteca, que es una colección concreta de clases

Funciones: Una secuencia de código que realiza una tarea específica y devuelve un valor específico

Subrutinas: Una secuencia de código que realiza una tarea específica pero no lo hace devolver los valores

Propiedades: Las cualidades de un objeto

La estructura de un archivo. Vb se muestra en la Figura 6-2. Dentro de la vb. archivo son espacios de nombres. Dentro de los espacios de nombres son clases. Finalmente, dentro de la las clases son funciones, subrutinas y propiedades, entre otras cosas.

Esta estructura tiene mucho sentido cuando se va a utilizar un archivo de clase. Para ejemplo, tomemos el System.String.Compare método (). El espacio de nombres es Del sistema, la clase es String, y la función es comparar. El siguiente código muestra esta estructura:

```
Espacio de nombres System
    Clase String
        Función Pública Comparar (ByRef como un String, ByRef b as String) As Integer
            'Código funcional estarías aquí
            "Al final, le devuelven un entero
        End Function
    End Class
Fin Espacio de nombres
```

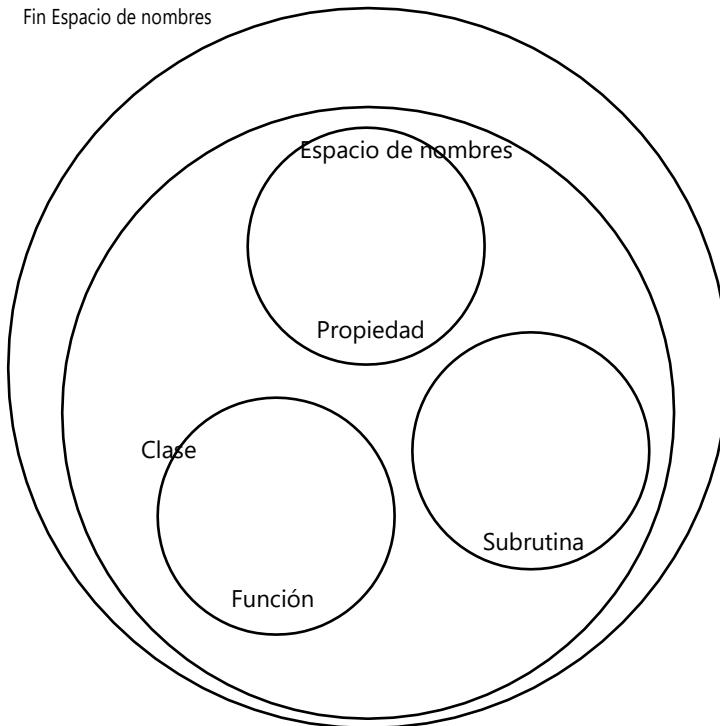


Figura 6-2:
La clase
biblioteca
de archivos
estructura

Dentro de la clase son los elementos de la construcción de código del lenguaje Visual Basic 2008. En lugar de utilizarlos ahora, sin embargo, les está construyendo:

Las funciones devuelven un valor. Generalmente, las funciones de entrada y de retorno aceptan de salida. La Calculadora de la fecha usaría una función que acepta una fecha como entrada y devuelve una fecha una semana en el futuro como salida. Funciones se designan por la palabra clave Function.

Las subrutinas no devuelven un valor. Generalmente, las subrutinas modificar algunos- lo demás en el sistema, tales como las propiedades dentro de la clase. Si escribió código para establecer el valor de una etiqueta, que sería una subrutina porque que no devuelve ningún valor cuando lo llamó. En cambio, actúa sobre algunos- lo demás en el sistema. Las subrutinas se indican mediante la palabra clave Sub.

Propiedades de mantener un valor. Una instancia de una clase mantiene su propia datos de las propiedades. Por ejemplo, cuando se establece el valor de una etiqueta en un Windows Forms o aplicaciones de formularios Web Forms (como se explica en los capítulos 4 y 5), se modifica su propiedad text. Las propiedades se denota por la Propiedad de palabras clave.

No dejes que te engañe. Hay mucho más a una clase en cualquier idioma que funciones, subrutinas y propiedades, pero los que son suficientes para empezar.

Codificación de una biblioteca de clases

La siguiente lista describe algunas de las cosas importantes que usted necesita saber sobre una clase de biblioteca:

Archivos de clase son, por naturaleza, los pesados de código dispositivos.

Las bibliotecas de clases están diseñadas y construidas con un lenguaje, y se referían a los para ser utilizado en un idioma.

Las bibliotecas de clases no se construyen generalmente usando un diseñador, como son Windows y formularios Web Forms.

Las bibliotecas de clases hacer uso de la misma. Herramientas .NET Framework que Windows o utilizar los formularios Web Forms en la vista Código. De hecho, el código de Windows o Web Formas en realidad es una biblioteca de clases.

Una biblioteca de clases consta de tres partes que son importantes para entender en este momento:

La definición de clase: Aquí es donde puede definir uno de lo que podría haber muchos clases en su biblioteca.

La declaración de la operación: Aquí se define algo acerca de esa clase para su uso posterior. Estas son las funciones, subrutinas, y las propiedades que definir en la sección anterior.

Capítulo 6: Construcción de Bibliotecas de Clase

El código funcional: Esta es la pieza más interna del puzzle, y se va dentro de una declaración de la operación. Aquí se escribe el código que hace el trabajar.

Listado 6-1 muestra las partes de la clase.

Listado 6-1: Las partes de la clase

Clase de muestra

```
SampleFunction Función Pública (entrada de ByVal como secuencia) como String
    "Aquí va el código funcional
    'La función devuelve una cadena
End Function

Public Sub SampleSub (entrada de ByVal como secuencia)
    'Código funcional estarías aquí
    'No devolver valor porque es una subrutina
End Sub

Público Propiedad SampleProperty () as String
    Obtener
        'Código para obtener el valor va aquí
        'Devuelve una cadena basada en el tipo de propiedad regreso
    End Get
    Set (ByVal como valor DateTime)
        'Código para establecer el valor va aquí
        'Utiliza el parámetro de valor para establecer la propiedad base
    End Set
    Fin de la Propiedad

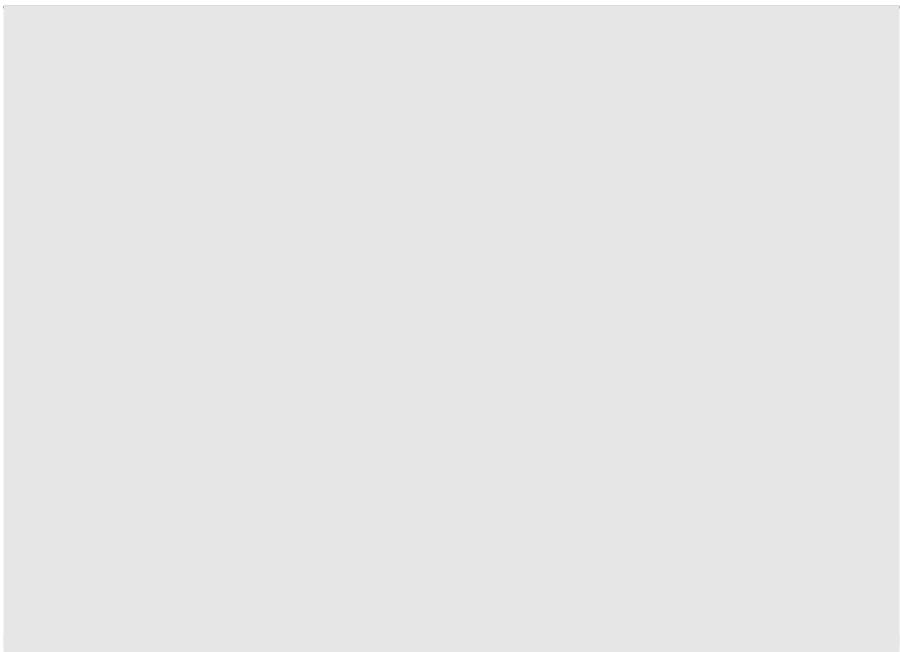
End Class
```

La dificultad en la codificación de las bibliotecas de clases se hace evidente cuando digo: "Eso es todo lo que necesitas saber. "El hecho es que casi cualquier cosa puede pasar en el funcional código, siempre que no depende de otro código en el resto del programa. Los procedimientos pueden ser definidos como deseé. Las clases pueden ser estructurados en prácticamente cualquier forma. Usted tiene un montón de espacio para hacer las cosas mal, y la única manera de encontrar la manera de codificar bibliotecas de clases de manera eficiente es práctica y de revisión de código de otros programas.

Creación de una biblioteca de clases

Un buen lugar para empezar a la hora de crear una nueva biblioteca de clases (tal vez su primero) es escribir un trozo de código que controla la funcionalidad de la Fecha

bre la Calculadora de la fecha).



Primeros pasos

Para empezar a crear una biblioteca de clases, siga estos pasos.

- 1. Abra Visual Studio y seleccione Archivo \Rightarrow Proyecto Nuevo.**
- 2. Seleccione VB.NET Biblioteca de clases de las plantillas de proyecto.**
- 3. Cambie el nombre del archivo de clase predeterminado, Class1.vb, a algo más adecuadas a su proyecto.**

Por ejemplo, he nombrado el DateCalc2008.vb biblioteca.

- 4. Agregue el código apropiado dentro del bloque de clases.**

En este caso, he añadido el código descrito en la siguiente sección de este capítulo.

- 5. Elija Generar \Rightarrow Generar solución para crear el archivo DLL.**

El archivo DLL es el archivo que se utiliza con la interfaz de usuario para implementar el código que se escribe.

El proceso de construcción de una biblioteca de clases es muy simple. Debido a que la mayoría de los el código está diseñado para que su ejecución del programa, no siempre va a obtener una gran cantidad de orientación de Visual Studio. Esto hace que averiguar lo que sucede en la clase biblioteca aún más difícil, y pone la carga sobre ti. La única guía que haya en la creación de una biblioteca de clases viene desde el diseño de su aplicación (véase el capítulo 3), lo que apunta a una de las razones por qué el diseño es tan importante.

La construcción de la Calculadora de la fecha

Cuando usted tiene un diseño, ¿sabes qué procedimientos debe definir y cuáles son las funciones de las necesidades de código. Si ha seguido los pasos de la sección anterior que está buscando a una clase en blanco similar al código siguiente, uno de los scariest cosas en toda la programación de Visual Basic - o uno de los más liberadora:

```
Public Class DateCalc2008
```

```
End Class
```

Para empezar, se necesitan tres propiedades: la fecha de inicio, fecha de finalización, y la duración de la que desea entre ellos. Para crear propiedades, es necesario para el almacenamiento local valores de las propiedades, en la forma de variables privadas.

Una variable privada es una variable que está definida fuera de una operación y que está disponible para todos los procedimientos dentro del mismo archivo de clase, variables privadas a veces se llaman campos. Por convención, las variables privadas que proporcionan almacenamiento local de las propiedades usar el mismo nombre de la propiedad, pero el comienzo con un carácter de subrayado, como sigue:

Capítulo 6: Construcción de Bibliotecas de Clase

```
_startDate Privado como fecha y hora  
_endDate Privado como fecha y hora  
Private _span As Integer
```

Después, usted necesita las propiedades mismas. Comience escribiendo Público de la Propiedad

StartDate como DateT entre las líneas de la declaración de clase, y los Característica IntelliSense aparece con DateTime seleccionado. Pulse la tecla Tab para completar la declaración.

A continuación, pulse Intro para terminar la línea y disfrutar de uno de los más bonitos, más simple FEA-ciones de desarrollo de biblioteca de clases con Visual Studio. La plantilla de código para la propiedad se completa para usted por Visual Studio, como sigue:

```
StartDate Público de la Propiedad () Como DateTime  
    Obtener  
  
        End Get  
        Set (ByVal como valor DateTime)  
  
    End Set  
    Fin de la Propiedad
```

Esto se siente un poco más como Visual Basic. Todo lo que tienes que hacer es terminar el código. Visual Studio construido dos mini-procedimientos para vosotros, las cuales son partes predefinidas de una propiedad - Get y Set. Funcionan exactamente como se esperaba: Obtener la ha código que obtiene el valor de la propiedad, y Set tiene el código que establece el valor de la propiedad. La declaración jurada de patrimonio terminado se parece a la después:

```
StartDate Público de la Propiedad () Como DateTime  
    Obtener  
        StartDate = _startDate  
    End Get  
    Set (ByVal como valor DateTime)  
        _startDate = valor  
    End Set  
    Fin de la Propiedad
```

Realiza dos propiedades más siguiendo el mismo procedimiento, pero sustituir EndDate y Span para sus nombres. Recuerde que el valor de Span debe ser Integer.

Haciendo matemáticas con una subrutina

a la biblioteca para hacer los cálculos necesarios para utilizar la propiedades. Como se describe en los capítulos 4 y 5, este código es bastante simple, y ha cambiado. En lugar de utilizar los valores de DateTimePickers y etiquetas, utiliza las propiedades, y en vez de un evento, se utiliza una subrutina.

Una subrutina es una operación que no devuelve ningún valor - que sólo afecta a inter-valores internos. En este caso, los valores internos son las variables particulares en la clase. Un ejemplo de cómo funciona la subrutina es la siguiente:

```
IncreaseDate Public Sub ()  
    EndDate = StartDate.AddDays (Span)  
End Sub
```

La lógica de esto es un poco complicado. Se supone que el usuario del sistema establece las propiedades, por lo que la lógica en la instrucción Set, el sector privado va-bles se establecen, así como en la instancia de la clase que el usuario está trabajando.

Se utilizan las variables privadas para hacer los cálculos y establecer el _endDate privado variable. Cuando el usuario va a obtener el valor final - el EndDate propiedad - la lógica de la instrucción Get se llama, y se pone al usuario la valor actual de la propiedad privada.



Estas propiedades son un ejemplo sencillo de un tema muy complejo, una idea poderosa. Esta Puede parecer una gran cantidad de código extra para hacer algo tan simple, pero cuando estás el desarrollo de aplicaciones, que rara vez crear un programa tan simple como la Fecha Calculadora. Y como desarrollador, usted encontrará a menudo - en un proyecto real - de que un código extra hace que el proyecto mucho más fácil de escribir.

Al generar el proyecto, tiene una biblioteca de clases que calcula un determinado número de días a partir de una fecha determinada. Es más, la biblioteca de clases se compila en un archivo DLL utilizable por cualquier aplicación .NET. En la siguiente sección, te muestro cómo utilizar el archivo DLL.

Ejecutar un archivo DLL

Como mencioné al principio de este capítulo, una DLL debe ser utilizado por un aplicación con una interfaz de usuario, como una aplicación de Windows Forms.

Para ejecutar el archivo DLL de nuevo, es necesario agregar un proyecto con una interfaz de usuario la misma solución que contiene el proyecto de DLL. Los pasos siguientes le ayudarán conseguir este trabajo:

- 1. Seleccione Archivo $\Rightarrow \Rightarrow$ Agregar proyecto Nuevo.**
- 2. Seleccione una nueva aplicación para Windows y darle un nombre apropiado.**

I nombrado DateCalcShell mina, lo que representa que es una capa alrededor del DLL hará referencia.

- 3. Cambie el nombre del formulario predeterminado para algo apropiado.**

Estrategias de denominación no cesan. Puse el nombre de mi calculadora de la forma.

- 4. Haga clic derecho en el nuevo proyecto y seleccione Agregar referencia.**

Aparecerá el cuadro de diálogo Agregar referencia.

Figura 6-3:
Calc
Clase
2008
en el
complement
o
Referencia
Cuadro de
diálogo.

Figura 6-4:
La
Referencias
pestaña de
la
Mi proyecto
archiv

**5. Haga clic
en la ficha
Proyectos,
seleccione el
proyecto que**

aparece allí, y haga clic en el botón OK.

En este ejemplo,

CalcClass2008 aparece en la

lista bajo el Proyecto

117

Nombre, como se muestra en la Figura 6-3.

6. Haga doble clic en el archivo

de mi proyecto en el

Explorador de soluciones y,

cuando

se abre en la vista Diseño,

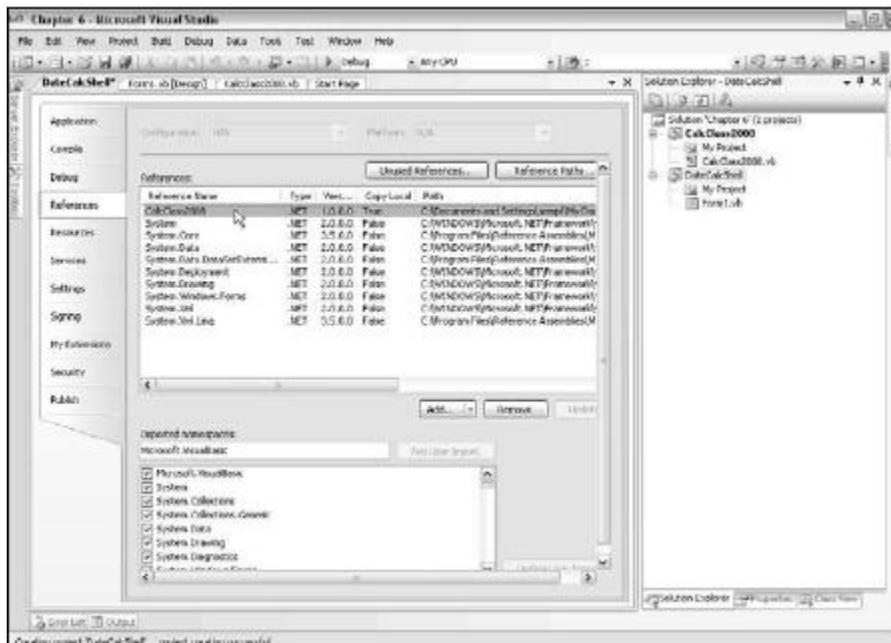
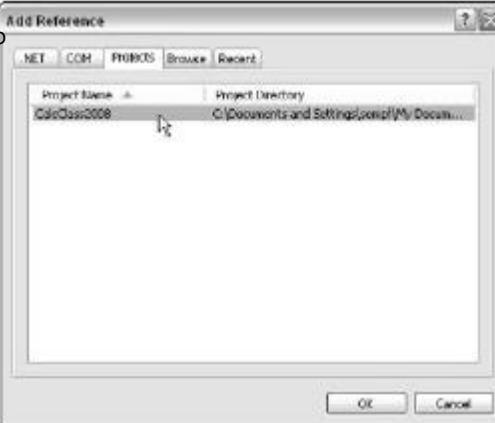
haga clic en la ficha

Referencias.

CalcClass2008 aparece en la

cuadrícula Referencias, como

se muestra en la Figura 6-4.



7. Volver al archivo de formulario y pasar los controles adecuados a la forma.

Agregar una etiqueta y un control DateTimePicker. (Vea el Capítulo 4 para obtener más sobre estos controles.)

8. Cambie los nombres de los mandos en algo apropiado.

Yo StartDatePicker para el control DateTimePicker y EndDate para el control Label.

9. Cree una instancia de la clase de la biblioteca de clases como un nuevo objeto en el Código Ver.

Conseguir un nuevo objeto DateCalc es lo mismo que recibir una nueva instancia de el objeto String. Recuerde, todo es un objeto en .NET. Un simple dimensión hace el truco: myCalc Dim como nuevo DateCalcClass Chapter6.CalcClass2008 () .

10. Haga doble clic en los controles para agregar código.

En este caso, haga doble clic en el control StartDatePicker Para agregar un valor Cambiado controlador de eventos. A continuación, utilice el nuevo componente que ha definido en el Paso 9. Puede establecer la propiedad StartDate del valor de la Control DateTimePicker, establezca la propiedad Span a 7 para representar un lapso de siete días, y luego llamar al método Calculate para que el objeto establece el EndDate para usted. Por último, establezca el texto de la etiqueta EndDate al endDate propiedad del objeto myCalc. Este código es la siguiente:

```
Private Sub StartDatePicker_ValueChanged(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles StartDatePicker.ValueChanged  
    Dim myCalc como nuevo 2008DateCalsClassChapter6.CalsClass2008()  
    myCalc.StartDate = StartDatePicker.Value  
    myCalc.Span = 7  
    myCalc.IncreaseDate()  
    EndDate.Text = myCalc.EndDate.ToString()  
End Sub
```

11. Haga clic en el botón Inicio para probar la aplicación.

Ajuste el control DateTimePicker a un valor y ver la etiqueta cambiar. Para una emoción barata, utilice el depurador, que se cubre en el Capítulo 8. Establezca un punto de interrupción en el archivo DLL, y utilizar el depurador para ver el Código de caminar a través de dos proyectos. Es limpio. Es posible que tenga que hacer clic derecho el archivo de proyecto para la aplicación para Windows y configurarlo como el inicio Aplicación.

Profundizar en DLLs

No hay más que descubrir sobre DLLs de lo que puede poner en este capítulo. Sin embargo, usted debe entender algunos puntos más acerca de DLL antes de empezar a usar ellos.

Capítulo 6: Construcción de Bibliotecas de Clase

Contando entre amigos y enemigos

A lo largo de este capítulo, se utiliza la palabra clave Public para describir la clase de procedimientos. Esta no es la única manera de describir procedimientos, sin embargo, y las diferencias entre los procedimientos son notables, especialmente cuando se trata el espinoso tema de la seguridad. Discutir la seguridad en profundidad en el capítulo 14, pero discutir aquí brevemente palabras clave de accesibilidad.

Cinco palabras clave de accesibilidad se puede utilizar para describir los procedimientos. Describen una seguridad poco a poco más restrictiva perspectiva. En general, usted desea elegir el acceso más restrictivo posible para su utilización a que el procedimiento en cuestión:

Público: Estos métodos dan esencialmente sin restricciones. Cualquier aplicación con acceso físico al archivo DLL se pueden utilizar métodos Públicas.

Protegido: Estos métodos sólo están disponibles a partir de otros métodos dentro su propia clase, o una clase que se deriva de su clase.

Amigo: Estos métodos funcionan en cualquier parte de la aplicación donde están declarado.

Protected Friend: Estos métodos son una combinación de la protegido y las palabras clave amigo, por lo que sólo están disponibles en una aplicación donde la clase y el programa de llamadas están en la misma clase y de reunión.

Privado: Estos métodos sólo son accesibles desde dentro de su propio ámbito.

Sé amable y participación

Compartido funciones son muy útiles porque el programador no tiene que instanti-comió la clase en un objeto con una declaración dimensión para usarlo. Usted puede simplemente llamar directamente a las funciones - pero usted no tiene el beneficio de un objeto con estado con propiedades y así sucesivamente. Usar la palabra clave compartida es una herramienta más en su caja de herramientas del programador.

Para construir una función compartida, tiene que aceptar y devolver valores. Por ejemplo, construí una función compartida que acepta el StartDate y Span valores y devuelve un valor de DateTime, que debería ser la fecha de finalización.

Esta función compartida es completamente diferente de la subrutina porque no utilice las propiedades. Es una función completamente diferente. Estoy mostrando

ine/prop-
solución breza son dos maneras de hacer la misma cosa.

Por lo tanto, como lo hace con las propiedades de la sección anterior "Creación de la Fecha Calculadora ", escriba el inicio de la declaración de la función, y IntelliSense recoge que está la declaración de variables, como se muestra en la Figura 6-5.

119



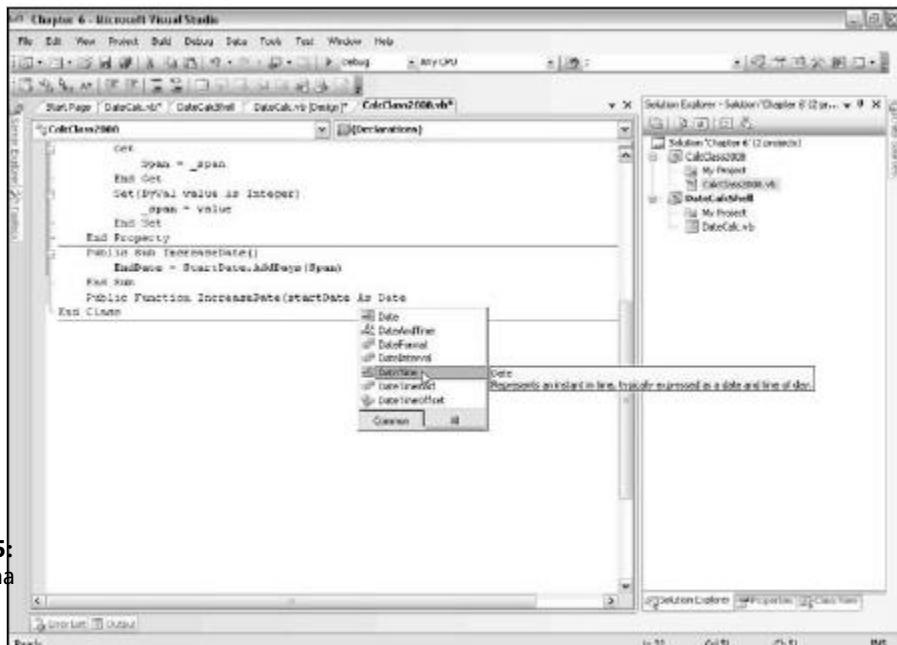


Figura 6-5:
Declarar una función.

El código funcional luego toma los valores StartDate y Span que son aprobada en, hace los cálculos fecha, y establece el valor de la función igual a el resultado. El código es el siguiente:

```
Función Pública IncreaseDate ( ByVal como startDate
    DateTime, _
    Lapso ByVal As Integer) como fecha y hora
    span = CDbL (span)
    IncreaseDate = startDate.AddDays (span)
End Function
```

Conseguir más por menos

Sin decir nada, he usado una de las características más útiles de VB 2008 clase de diseño en el libro de jugadas - sobrecarga funcional. Note algo interesante acerca de la clase de acabado, especialmente la función IncreaseDate... Oh, espera, ¿fue una subrutina? No, era a la vez! ¿Cómo es esto posible?

Es posible a través de la sobrecarga. Para simplificar el diseño de la clase, dos métodos puede tener el mismo nombre si tienen una firma de método diferente, es decir, para- tipos de medidores y el número. En este caso, tiene una función que no utiliza las propiedades, por lo que acepta la fecha de inicio y la duración como parámetros, y luego devuelve la fecha de finalización. Una subrutina tiene el mismo nombre, pero tiene cero parámetros y no devuelve nada.

Debido a las diferentes firmas, puede tener dos métodos que realizan aproximadamente lo mismo, pero de una manera diferente. El IDE está aún preparado para manejar esto con una característica especial de IntelliSense, como se muestra en la Figura 6-6.

Cuando se llama al método en el código, después de escribir el primer paréntesis, se puede ver que el método se describe con dos líneas. Utilice las teclas de flecha para moverse entre ellos. Usted no tiene que elegir una específica forma explícita; el IntelliSense está ahí como referencia.

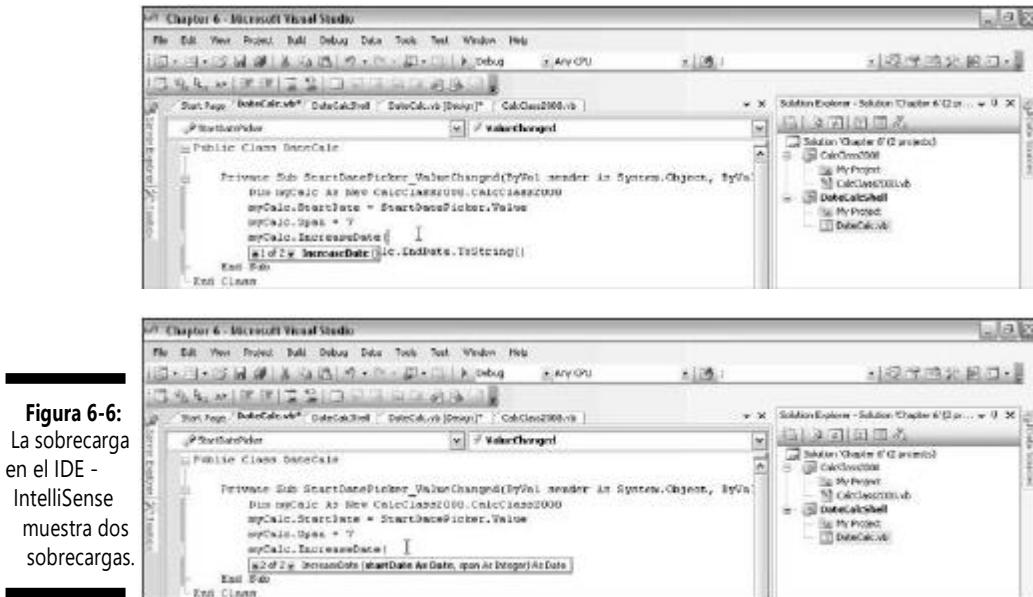


Figura 6-6:
La sobrecarga
en el IDE -
IntelliSense
muestra dos
sobrecargas.

Capítulo 7

Edificio de Servicios Web

En este capítulo

- Comprender los servicios web
- El uso de IIS para proporcionar la lógica de negocio
- Diseño de aplicaciones con servicios Web
- La construcción de su primer servicio Web

Tu capítulo cubre la cuarta parte de lo que considero los cuatro más importantes tipos de proyectos: el servicio Web XML. En el gran esquema de las cosas, una Servicio Web XML es una versión de código abierto de la biblioteca de clases (que se describe en Capítulo 6). Puede ser utilizado en una aplicación de Windows o Web (ver capítulos 4 y 5, respectivamente), según sea necesario.

Las dos partes principales del desarrollo de un servicio web están produciendo y consumiendo. La producción de un servicio web es lo que haces como programador: desarrollar un servicio para el consumo de los usuarios (como ocurre con las bibliotecas de clases construidas en el capítulo 6).

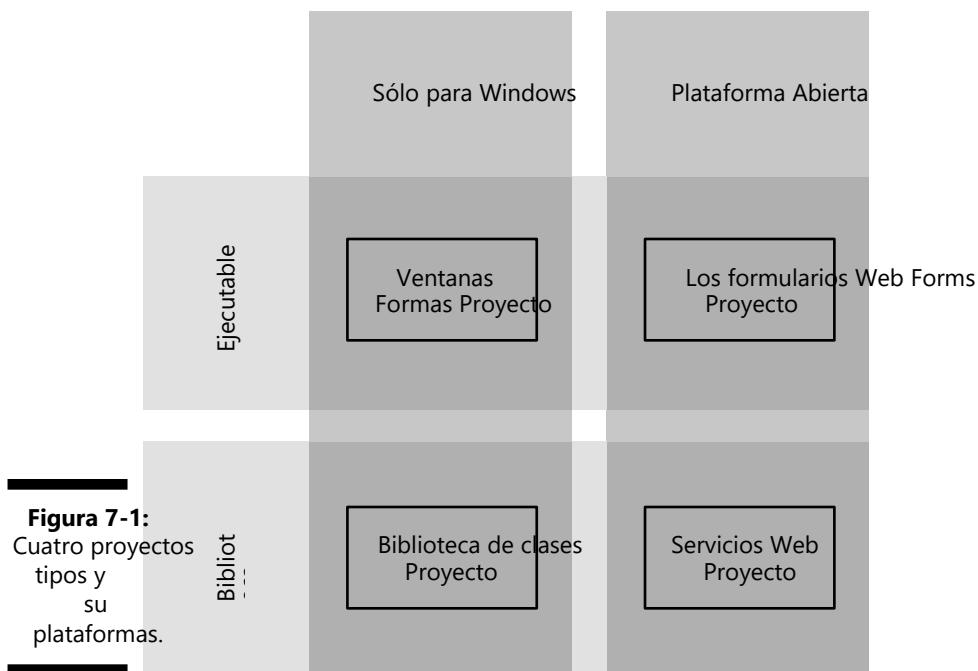
Consumidor un servicio Web es lo que el usuario final hace: hacer uso del servicio en su aplicación.

En este capítulo también examina cómo cerrar la brecha entre la producción y consumir servicios Web - por ejemplo, la documentación de su servicio para que la página por defecto que IIS crea para que realmente tenga sentido para el lector. Los jugadores de los servicios Web XML se muestran en el acto final del capítulo, incluyendo un desfile de las siglas grandes - XML, SOAP, WSDL y UDDI - y lo que significan.

Introducción a los servicios web XML

Si la creación de un servicio web parece un poco familiar, no es casualidad, si no, no problema. Un servicio Web XML es un archivo DLL ¿Qué formularios Web son a Windows Formularios - y (como un archivo DLL) es una biblioteca de clases - pero más todavía. Un servicio web es más versátil que un archivo DLL o Windows Forms - después de todo, son compilados sólo para su uso en equipos con Microsoft Windows. Servicios Web XML y Los formularios Web Forms se puede utilizar en cualquier plataforma, ya que siguen los estándares abiertos.

Figura 7-1 resume las similitudes y diferencias de estos tipos de proyectos.



La figura muestra la extensión de conceptos conocidos de Windows en código abierto XML versiones que hace utilizable en la Web:

Los formularios Windows Forms tienen su equivalente en los formularios Web Forms (detallado en el Capítulo 5), que crean una interfaz de usuario que se pueden ver en cualquier plataforma.

DLL de Windows se compilan a partir de bibliotecas de clases (véase el capítulo 6) para ayudar construir aplicaciones en plataformas Windows, un servicio Web es una abierta fuente equivalente de un DLL.

Los servicios Web XML representan una novedosa combinación de las mejores características de bibliotecas de clases y formularios Windows Forms. Lo que es nuevo aquí es XML, da Web servicios de un formato estándar que está disponible para una variedad de plataformas - incluyendo (pero no limitado a) de Windows.

Por supuesto, esta relación básica entre los servicios Web XML y sus ancestros es sólo una pequeña superficie. Se usan adecuadamente, podría ser la próxima gran Cosa en el desarrollo de aplicaciones. Para producir un servicio web sólida y tener se utiliza (consumida) correctamente, sin embargo, usted tiene que conseguir una manija en una gran cantidad de detalles prácticos - como las políticas de seguridad, la gestión de las transacciones, y la disponibilidad de recursos del sistema. Las siguientes secciones le dará una más estrecha mira lo que hace que una marca de servicio Web.

Capítulo 7: Edificio de Servicios Web

Servicios Web: Características

DLLs tiene una amplia funcionalidad en Windows, es lógico que como su hijos talentosos, servicios Web XML son ampliamente utilizable. Si usted está diseñando un amplio sistema de la empresa, esta flexibilidad significa que usted puede (y probablemente debe) desarrollar un conjunto completo de herramientas para una amplia gama de usuarios. Recuerda, sin embargo, las siguientes cuatro características esenciales de los servicios XML Web:

La Arquitectura-neutral: Los servicios Web no depende de ninguna propiedad configuración del cableado, cable tipo, formato de archivo, la descripción de esquema, que operación del sistema o estándar de descubrimiento.

Descubrimiento Es así como a otros consumidores puede averiguar lo que el servicio Web hace.

Ubicuo: Los servicios Web son "todos para uno y uno para todos", en todas partes.

Cualquier servicio Web que soporte las normas pueden apoyar el servicio que está creando.

Simple: Creación de servicios Web es fácil, rápido, e incluso (a veces) gratis. Eso es en parte porque la descripción de los datos es legible, para hacer una experiencia mejor desarrollo. Cualquier lenguaje de programación pueden participar.

Interoperable: Debido a que los servicios Web todos siguen las mismas normas, todos pueden hablar el uno al otro.

Servicios Web: Quirks

Diseño de servicios Web se siente como el diseño de bibliotecas de clases debido a que (básicamente) eso es lo que son. Estas bibliotecas de clases particulares son Web-conducido, sin embargo, así que aquí están algunas diferencias de diseño a tener en cuenta:

Toda la comunicación entre una aplicación y el servicio pasa a través de Internet. Eso significa que usted incurre por lo menos algo de sobrecarga para enviar cada mensaje individual; el programador sabio reduce el número total de los mensajes.

Latidos Chunky hablador. Envío de mensajes de menos significa hacer menos - y más grande - llamadas a funciones. En lugar de hacer varias llamadas para obtener piezas

da la docu-
ambiente. Esta práctica se llama hacer trozos en lugar de llamadas hablador.

Estas bibliotecas de clases no son estado. No hay propiedades (o-cualquier cosa como ellos) en una biblioteca de servicios Web basados en clase, en efecto, toda la opera-
ciones son compartidos.

Puesto que las funciones son compartidos, las subrutinas no son muy eficaces. Después todo, no hay propiedades o variables locales están disponibles para las subrutinas a alterar su llamado. Al igual que con las aplicaciones Web, el problema es la falta de statefulness. Su servicio, entonces, termina como un conjunto de herramientas, implementado como funciones con valores de retorno.

Diseño de Servicios Web

El objetivo general de los servicios web de construcción es conseguir que su lógica de negocio expuestos

a las masas. (Lógica de negocios es el código que se encuentra entre la interfaz de usuario forma y su fuente de datos o archivo y que indica al programa qué hacer, usted Puede leer más al respecto en los capítulos 3 y 6.) Porque existe un estándar para la con-Suponiendo servicios Web, así como para la creación de ellos, usted puede enfocar su desarrollo en proporcionar herramientas para una gama muy amplia de aplicaciones.

Planificación de la estrategia de diseño

La raíz del diseño de Web-servicio es un contraste básico: DLL son de estado - y Los servicios web no son. Al crear una instancia de un archivo DLL con una instrucción Dim en código de la aplicación, se crea una representación en la memoria de la clase. No es así con los servicios web, que utiliza IIS para mantenerse a sí mismos, por lo que no se pueden instanciar de la misma manera. Tratarlos como si estuvieras el desarrollo de un manojo de funciones estáticas, en lugar de un grupo de objetos con estado.

Esta estrategia en realidad tiene un nombre - Service-Oriented Architecture o SOA. Cuando usted está diseñando para SOA, el enfoque es diferente de lo que puede puede utilizar para: De hecho, usted lo tratan más como una caja de herramientas y menos como la vida pieza de software.

No es ninguna sorpresa que el diseño de SOA requiere una especie de mil yardas vista. Afortunadamente, Visual Studio proporciona este panorama - un sistema que arquitectos puede utilizar para atar las aplicaciones Windows y Web Forms en el mismo paquete con bibliotecas de clases construidas a partir de los dos servicios Web y archivos DLL. De esto a vista de pájaro, el mapa se ve como en la Figura 7-2.

Bueno, Figura 7-2 presenta el marco de referencia básico para su estrategia de diseño. Lo que convierte el proyecto en un servicio Web real es otra consideración eso es tan importante para el desarrollo de software: cómo usar su programming idioma. Manténgase en sintonía.

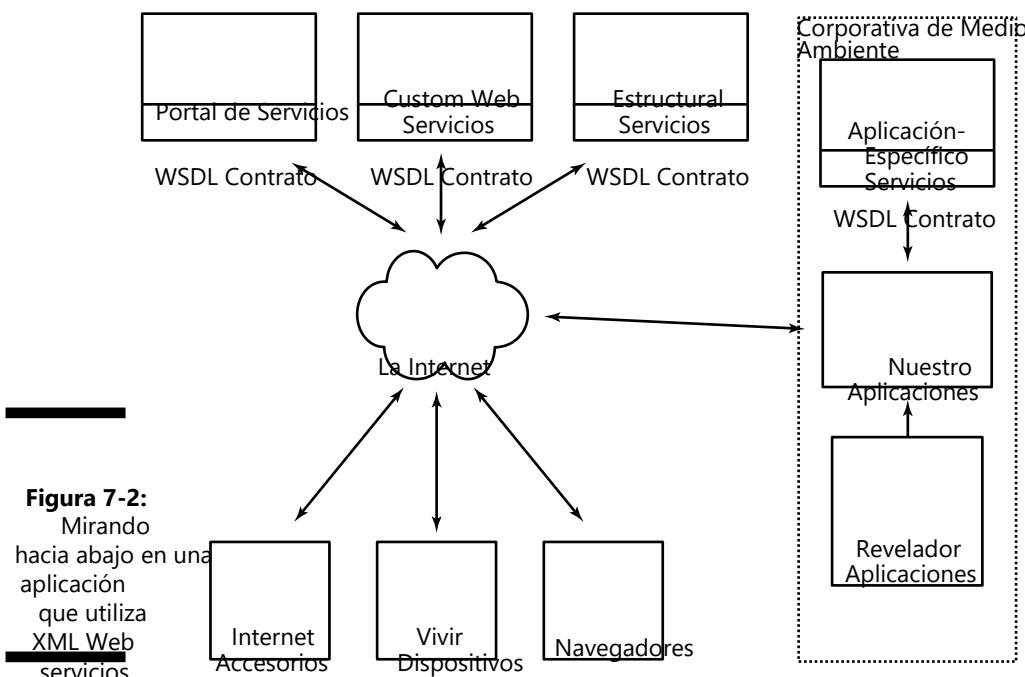


Figura 7-2:
Mirando
hacia abajo en una
aplicación
que utiliza
XML Web
servicios.

Conseguir un apretón en la táctica

El desarrollo de un servicio Web requiere de dos conjuntos muy diferentes de las tácticas: un conjunto para producir el servicio Web y un conjunto para controlar el funcionamiento del servicio es consumido. Algunas específicos - pero familiar probablemente - tácticas entran en juego cuando se utiliza Visual Basic 2008 para crear herramientas que funcionan dentro de SOA:

Las tácticas que utilizan para producir servicios Web será como los que se utilizan para crear bibliotecas de clases y compilarlos en archivos DLL.

Las tácticas que controlan cómo un servicio Web que se consume son similares a hace referencia a un archivo DLL y utilizar sus métodos y propiedades en un equipo con Windows

Formas del proyecto (como se describe en el capítulo 6).

La construcción de un Web Service

Todo el proceso de creación de un servicio web y configurarlo para su uso implica tres etapas: la producción del servicio, viendo el servicio para hacer Seguro que va a hacer lo que quieras, y hacer el servicio a disposición de los usuarios que lo consumen.

La producción de un servicio web comienza con (sorpresa, sorpresa) una plantilla de proyecto. Siga estos pasos para obtener un iniciado:

1. Abra Visual Studio y seleccione Archivo ⇔ ⇔ Nuevo Sitio Web.

El resultado de un proyecto de servicio Web es un sitio Web, al igual que la salida de un proyecto de biblioteca de clases es un archivo DLL.

2. En Visual Basic, seleccione Servicio Web ASP.NET como el Visual Studio instalado plantilla.

También te darás cuenta de que un servicio WCF está disponible. Más sobre esto más adelante.

3. En Ubicación, cambiar el nombre a algo más apropiado para la proyecto.

En este caso, he cambiado el nombre de mi proyecto de ejemplo para Chapter7DateCalc. Yo cambió el nombre de la solución a Capítulo7.

Visual Studio crea una plantilla de proyecto para usted. La nueva plantilla incluye un archivo de servicio por defecto llamada Service.asmx.vb, una carpeta de datos, y un archivo llamado Service.asmx ASMX. El archivo contiene Service.vb la clase de código de la biblioteca. El archivo contiene código de visualización Service.asmx que el servidor Web se utilizan para hacer una página automática de documentación para el servicio Web.

**4. Agregar un archivo de servicio web nuevo y cambiar el nombre de los archivos y de la clase a algunos-
lo apropiado para el proyecto.**

Aquí, he utilizado DateCalc, así que mi clase es DateCalc, mi archivo ASMX es Fecha Calc.asmx y el archivo de clase es DateCalc.asmx.vb.

5. Escriba el código en la clase como la Función Pública, sobrescribiendo la muestra método.

Observe el código predeterminado en el Listado 7-1.

Listado 7-1: El Servicio Web predeterminado

```
1: Imports System.Web
2: Importaciones System.Web.Services
3: Importaciones System.Web.Services.Protocols
4: <WebServiceBinding (ConformanceClaims:= Ws1Claims.BP10,
                     EmitConformanceClaims:= True)> _
5: Servicio Público Clase
   Hereda System.Web.Services.WebService
6:   <WebMethod()> _
7:   Public Function HelloWorld () As String
8:     Return "Hello World"
9:   End Function
10: End Class
```

¿Eso es todo lo que hay a un servicio Web? Bueno, sí, pero más que está pasando aquí que
lo que parece. He aquí un vistazo más de cerca:

Después de las declaraciones de Importaciones y antes de la declaración de clase,
una web

Declaración ServiceBinding (on line 4) sirve como una directiva de compilación.
Se le indica al marco. NET que esta clase particular se utiliza como un
Web de servicios.

La declaración de clase estándar de futuro en la línea 5 es seguido por una heren-
tancia declaración que le da el programador de la clase de acceso a
los métodos, propiedades y eventos de las clases de servicios Web.

Otra directiva de compilación aparece en la línea 6 - la directiva WebMethod,
que le da algunas opciones de documentación más tarde en y muestra también la
compilador que este método específico estará expuesto al servicio cuando
que lo tenga instalado y funcionando.

Las líneas 7, 8 y 9 son bastante estándar en Visual Basic 2008: sólo una función
que no acepta nada y devuelve una cadena: "Hello World". The End
Declaración completa de clase de la clase. Es sólo una línea de prueba de código,
asegúrese de que el cableado funciona.

Puede reemplazar las líneas 7 a 9 con casi cualquier función VB. En el Microsoft
mundo, como una función puede devolver cualquier objeto en el universo. NET. Pero
vamos a

no llegar demasiado lejos aquí. En realidad, se debe considerar que un UNIX o man-
marco ordenador puede llamar a este servicio. Si multiplataforma (o hacia atrás) com-
patibilidad es un problema, puede ser necesario limitar el valor de retorno para un
tipo primitivo. (I cubrir los gastos en el capítulo 9.)

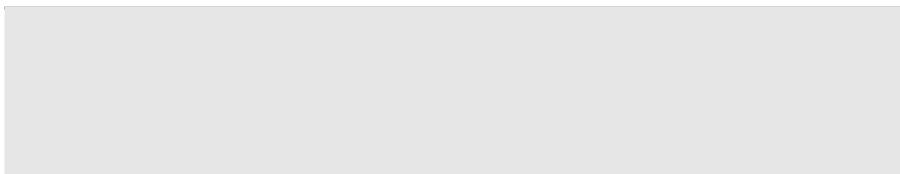
Cuando usted tiene su plantilla completa del proyecto en la mano, usted está listo para
generar el servicio Web.

Construyendo el servicio Web DateCalc

En esta sección del capítulo se basa el servicio Web como un detallado DateCalc
ejemplo. He cambiado el código de ejemplo en las líneas 7 a 9 con una función
para la biblioteca de clases en el archivo DLL (véase el capítulo 6 para más detalles de
esta fun-
ción). Listado 7-2 muestra el código de ejemplo para el servicio final.

Listado 7-2: El Servicio DateCalc

```
1 Imports System.Web
2 Importaciones System.Web.Services
3 Importaciones System.Web.Services.Protocols
4 <WebService(Namespace:="http://services.vbfordummies.com/")> _
5 Public Class DateCalc
    Hereda System.Web.Services.WebService
```



Listado 7-2: (continuación)

```

6:     <WebMethod (Description: = "Una aplicación de servicio Web de la Fecha
          Calculadora ")> _
7:     Función Pública IncreaseDate (startDate ByVal como DateTime, lapso de ByVal como
          Integer) Como DateTime
8:         IncreaseDate = startDate.AddDays (CDbl (span))
9:     End Function
10:    End Class

```

Está bien, lo admito, he cambiado algunas cosas aquí (ese fue el punto):

- Línea 4** ha pasado de ser una directiva WebServiceBinding a una web Servicio Directiva. Hice eso por lo que yo podría fácilmente describir un defecto espacio de nombres. Espacios de nombres predeterminados son importantes para el consumidor, sino que validar la ubicación esperada del servicio con su ubicación real. (La directiva WebServiceBinding se utiliza con mayor frecuencia para la empresa servicios de nivel, que no son de interés aquí.)
- Línea 6** ahora incluye una propiedad Descripción de manera que el servicio es más auto-documentado.
- Línea 7**, la declaración de la función, ahora acepta un startDate y span como entrada.
- Línea 8** contiene el código que tiene un papel estelar en los cuatro capítulos en los Parte II - la fecha de matemáticas que aumenta la fecha de inicio.

Cuando los correspondientes servicios Web características están en su lugar, dar el nuevo servicio una prueba de conducción antes de ser enviados allí para conocer a los usuarios.

Viendo el servicio DateCalc en acción

Haga clic en el botón Reproducir para iniciar el servicio. (Ahora, hay algo que no puede . hacer con un archivo DLL) Servicios Web vienen con una página de visualización por defecto, en este caso, han llamado DateCalc.asmx. IIS creará una página agradable para usted (como se muestra En la Figura 7-3), con algún tipo de documentación como se describe en el WebMethod Directiva.



Al hacer clic en el enlace IncreaseDate, verá la página de prueba para la función He creado, como se muestra en la Figura 7-4.

No lo entiendo, si usted no está trabajando con el desarrollo Web incorporado servidor. Si está utilizando IIS, se bloqueará el código de prueba, ya que es una garantía preocupación.

En este punto, introduzca una fecha de inicio y un intervalo de tiempo en el startDate y span cuadros de texto, respectivamente. (7/25/75 entré en el cuadro de texto startDate y 13

en el cuadro de texto lapso.) Haga clic en el botón Invoke, y el navegador abre un nuevo ventana con la respuesta en toda su gloria formato de la fecha, como se muestra en la Figura 7-5.

Figura 7-3:
La apertura
La página de
la
DateCalc
Asmx

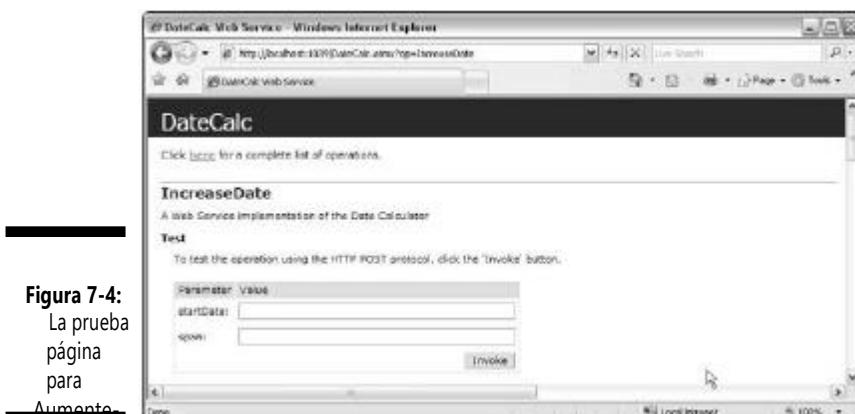


Figura 7-4:
La prueba
página
para
Aumento
Fecha.



Figura 7-5:
La
DateCalc
respuesta.

Bastante bien - pero este ejercicio no es realmente el punto del servicio Web XML He creado. Es una manera simple de probar un servicio web - incluso un sofisticado conjunto de servicios - independientemente de si los ha escrito usted mismo o tener trató de otra persona obra.



El objetivo de cualquier servicio web (como menciono en la sección "Diseño para Web Servicios ", anteriormente en este capítulo) es proporcionar a otras aplicaciones con acceso a la lógica empresarial. Observe que todas estas páginas de muestra se muestran en una Web navegador - y que el espacio de nombres del servicio es a una dirección Web. Ni de esas opciones es un accidente. Si esta función se publica, yo quiero que sea disponible en una URL en la World Wide Web - como la siguiente:

`http://yourMachine/Chapter7DateCalc/DateCalc.asmx`

Con el servicio de Web construido y probado, el siguiente paso es consumir el servicio en una aplicación. La siguiente sección le muestra cómo hacerlo.

Consumir un servicio Web

En un principio, la creación de una aplicación que consume un servicio Web parece similar a la construcción de una aplicación de prueba para una biblioteca de clases. Se empieza por construir un equipo con Windows

Formas similar a la descrita en el Capítulo 4 aplicación - pero luego referencia al servicio Web en mucho la misma manera que se hace referencia a una DLL (ver Capítulo 6).

De hecho, Visual Studio trata de un servicio web mucho el mismo que trata a una clase biblioteca en desarrollo. Cuando hace referencia a un servicio web y luego compilar un proyecto, Visual Studio crea en realidad un archivo DLL pequeño que recuerda la detalles del servicio Web. (Afortunadamente, esto ocurre de forma automática, a medida que se puede ver en los pasos siguientes.)

Para crear una aplicación que consume un servicio Web, siga estos pasos:

1. Crear un nuevo proyecto de cualquier tipo.

Aquí, he utilizado una aplicación de Windows Forms llamado DateCalcConsumer, pero cualquier proyecto. .NET puede consumir un servicio Web.

2. Haga clic derecho en el archivo de proyecto y seleccione Agregar referencia de servicio.

La Agregar referencia de servicio diálogo que se muestra en la Figura 7-6 aparece, ofreciendo para que haga referencia al servicio.

3. Escriba la dirección URL del servicio que se intenta hacer referencia a la Dirección de la caja de texto.

Bueno, si usted está haciendo ejercicio en el apartado anterior, no es necesario en realidad escribir la URL. Hace clic en el botón Discover, a continuación, abra la vista de árbol para mostrar DateCalc, a continuación, seleccione DateCalcSoap. Seleccionar



DateCalcSoap, y los espectáculos de función IncreaseDate en el derecho lado de la ventana. Por último, cambiar el espacio de nombres para Capítulo7.

Si no desea ejecutar el servicio de su máquina local - y usted quiere probar que ejecuta el servicio de Internet - probar el servicio que se estén ejecutando en www.vbfordummies.com. La dirección de dicho servicio es como sigue:

<http://www.vbfordummies.com/Chapter7/Services/DateCalc.asmx>

Después de que los ubica el servicio de navegador, la pantalla cambia a la referencia formato renuncia, como se muestra en la Figura 7-7. He utilizado el botón Discover para encontrar el otro proyecto en la solución aquí.

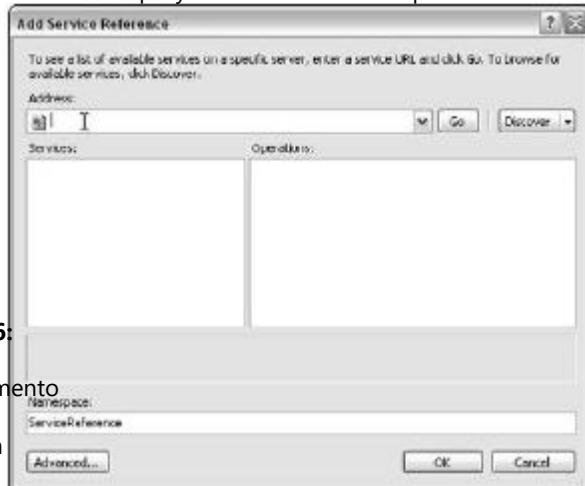


Figura 7-6:
El complemento
Servicio
Referencia
Cuadro de
diálogo.

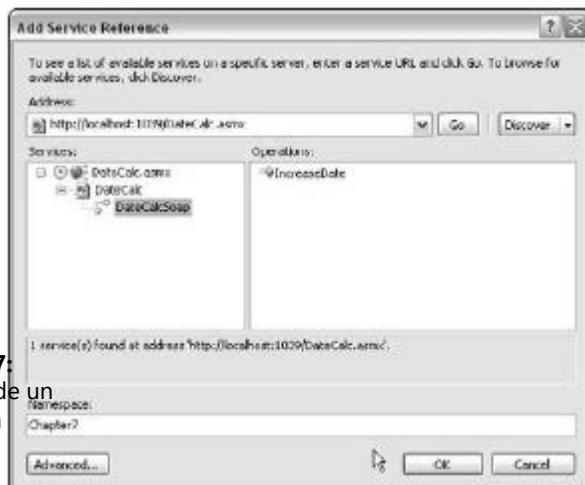


Figura 7-7:
Adición de un
referencia
al
DateCalc
servicio.

4. Si lo desea, cambie el nombre del espacio de nombres en el cuadro de la inferior a algo más apropiado para su proyecto y haga clic en Aceptar.

Tenga en cuenta que Visual Studio crea un archivo de clase para esto, al igual que usted hacerlo de forma manual en el capítulo 6. Nombrar es importante para que usted pueda encontrar el servicio con facilidad más adelante en el código. Después de hacer clic en Aceptar, Visual Studio crea-
ates una clase de proxy y te da una referencia a él en el Servicio Referencias carpeta. Un ejemplo aparece en la figura 7-8.

5. Agregue código al proyecto que hace referencia a la referencia de servicio le acaba de añadir.

En mi caso, tengo una aplicación de Windows Forms, por lo que añadir un DateTime Selector y una etiqueta, y haga doble clic en el DateTimePicker obtener el controlador de eventos ValueChanged.

6. Referencia del Servicio por instancia una copia nueva de la clase proxy.

En mi proyecto de ejemplo, esto es lo que este aspecto:

```
Dim myDateCalc Como Chapter7.DateCalcSoap
```

7. Llama a los métodos del servicio Web, tal y como lo haría con cualquier otro función en VB.

El código de acabado para la Calculadora de la fecha que utiliza el servicio Web Parece Listado 7-3.



Figura 7-8:
Una Web
referencia en
la Solución
Explorer.

Capítulo 7: Edificio de Servicios Web

Listado 7-3: La Calculadora de la fecha utilizando el servicio Web

```
Public Class DateCalc  
    Private Sub StartDatePicker_ValueChanged (ByVal sender As System.Object,  
                                         ByVal e As System.EventArgs) Handles StartDatePicker.ValueChanged  
        Dim myDateCalc Como Chapter7.DateCalcSoap  
        Label1.Text myDateCalc.IncreaseDate = (StartDatePicker.Value, 13  
    End Sub  
End Class
```

La clase de proxy tiene un estado, pero eso no quiere decir que el servicio en sí tiene ningún sentido de Estado. Si usted desea conseguir la suposición, puede utilizar este hecho de manera que los hacer que el servicio Web emular una clase - pero usted todavía está tratando con una dis-relacionadas con el servicio Web, no hace un seguimiento de su propio estado.

Servicios Web con mayor profundidad

Ah, los servicios Web - ahora, hay un tema que podría llenar varios libros (de hecho, He escrito algunos que todavía se puede encontrar flotando alrededor). Aunque usted no lo hace Hay que entender cómo todos los jugadores (sistemas, políticas, aplicaciones y usuarios) utilizar los servicios Web, ayuda si usted está trabajando con un equipo de desarrolladores para asegurarse de que todos están hablando el mismo idioma.

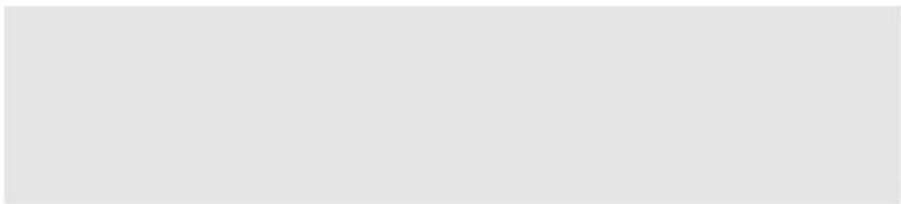
Desde la serena perspectiva a vuelo de pájaro de los servicios Web (consulte la Figura 7-2), por lo general, no tendrá que lidiar con algunas partes de la imagen grande como una des- operación - por ejemplo, en realidad la presentación de su servicio web para completar la usuarios. Servicios de Internet Information Server (IIS) es el servidor web de Microsoft aplicación que hace que estos servicios públicos. - Cuatro protocolos XML, SOAP, WSDL, y UDDI - son los que hacen ese milagro menor posible:

IIS: Internet Information Services desempeña el mismo papel con XML Web servicios como lo hace con los formularios web. La página que aparece cuando se ejecuta una servicio de Visual Studio, o escriba el URL en un navegador, es automáticamente generado por ASP.NET y se pasa al navegador por IIS. Usted puede ver este flujo en la Figura 7-9.

Figura 7-9:

IIS manejo

Consumir Web Local La Internet IIS Server IRS XML Web
servicio. Aplicaciones Puesto de trabajo Servicio



IIS trata de una aplicación de servicios web como una aplicación Web Forms. Conclusión: Se necesita el mismo tierno cuidado amoroso de su administrador. Recuerde, usted está exponiendo su lógica de negocio al mundo. La seguridad es importante y se trata en el capítulo 14.

XML: De los cuatro protocolos que hacen que los servicios funcionen Web, extensible Markup Language (XML) es el que usted está más probable encontrarse con. En general considero un protocolo de almacenamiento de datos, XML es la columna vertebral que soporta todos los mensajes que pasan como parte de los servicios Web.

SOAP: Simple Object Access Protocol describe los mensajes que son pasado. Es un meta-lenguaje XML. El Framework .NET le ahorra (Casi por completo) de tener que lidiar con SOAP.

WSDL: Web Services Description Language es el lenguaje XML meta-que describe la entrada del servicio y los parámetros de salida para el público consumo. Una vez más, los escudos Framework .NET es casi completamente de tener que lidiar con WSDL. Casi.

Ocasionalmente, sin embargo, un directorio de servicios Web tendrá un enlace a su WSDL para las cotizaciones. Para obtener el WSDL desde cualquier ASMX basada en servicio, agregar? WSDL al final de la URL. Por ejemplo, el WSDL para el URL enumero aquí sería en:

<http://www.vbfordummies.com/Chapter7/Services/DateCalc.asmx?WSDL>

UDDI: Descubrimiento Universal y Lenguaje de descripción XML es otro meta-lenguaje que ayuda a los consumidores con el descubrimiento de su Web servicios. Más información acerca de UDDI se puede encontrar en <http://uddi.microsoft.com>.

UDDI fácilmente podría tomar un libro completamente nuevo, y de hecho lo ha hecho. Yo por lo menos conseguir empezar por decirles cómo habilitar UDDI. El estándar de la industria es una Archivo DISCO - un archivo XML con un conjunto de etiquetas estándar que le dice a un servidor UDDI qué servicios están en su proyecto.

El proceso en el que un UDDI u otro servidor recibe servicios web (en este caso, desde un archivo DISCO) se llama descubrimiento.

Para crear un archivo DISCO, siga estos pasos:

- 1. Crear un nuevo archivo XML en su proyecto de servicio Web con un clic derecho el proyecto y seleccionando Archivo XML.**
- 2. Agregue las etiquetas estándar de descubrimiento en el documento.**

Para este propósito, este proceso puede ser tan simple como la que se muestra en la siguiendo el código, aunque será más difícil conseguir que los servicios más complex. (Todo lo que tendría que cambiar para otro proyecto sería las dos etiquetas que terminan en la referencia que yo pongo en negrita en el siguiente código.)



```
<? Xml version = "1.0">
<discovery xmlns="http://schemas.xmlsoap.org/disco/">
<discoveryRef ref="/Folder/Default.disco"/>
<ContractRef ref = "http://TheServerUrl/DateCalc.asmx?WSDL"
              xmlns = "http://schemas.xmlsoap.org/disco/scl/" />
</ Descubrimiento>
```

3. Cambie el nombre del archivo.

En este caso, se base en el archivo DISCO usando Default.disco.

Ahora, cuando usted se registra este servicio con UDDI - por ejemplo, en un Universal Empresas del registro en un servidor Windows 2003 - el portal no sabe dónde explorar. Luego, el servicio aparece en el listado de usar para todos.



UDDI es extraordinariamente útil para despliegues a gran escala de toda la empresa servicios, o los servicios públicos de cualquier tamaño. Recuerde, si usted está en un multiservidor entorno, puede crear servidores UDDI que ir a buscar a las funciones que necesitan mediante el uso de archivos de disco. Potencialmente, es un sistema muy poderoso.

Capítulo 8

Depuración en VB 2008

En este capítulo

- El uso de herramientas visuales para insectos de la calabaza
- Implementación de herramientas de depuración con el Marco. NET
- Encontrar errores en diferentes tipos de proyectos

Yn los ejemplos que figuran en la Parte II, se juega su código para ver lo ejecuta en un navegador Web o como una aplicación de Windows. Como habrás adivinado, hay más esta funcionalidad de lo que parece.

Depuración es el proceso de encontrar y solucionar problemas en una aplicación de cualquier tipo. A menudo, el código de depuración toma el tiempo escribiendo lo hizo en la primera lugar, según la mayoría de los sistemas de desarrollo de software de ciclo de vida, como por ejemplo CMM (Modelo de Capacidad de Madurez). Una de las razones más importantes para utilizando un entorno de desarrollo integrado para crear aplicaciones es tomar ventaja de las herramientas de depuración incluidos.

Visual Studio 2008 está repleto de herramientas de depuración que funcionan en algunos o todos los tipos de proyecto. A lo largo de este capítulo, te doy un golpe por golpe descripción de lo que la depuración de las herramientas están disponibles - tanto las herramientas visuales y herramientas del Marco. NET. Luego le mostrará cómo depurar cada tipo de proyecto.

Enfriar Visual Tools para depurar

La depuración es tan importante que tiene su propio menú en el entorno de Visual Studio-ambiente. La posibilidad de ver su carrera código, revise los valores de las variables, y comprobar el contenido de los objetos es la razón principal por la experiencia desaparecer utilizar un IDE como Visual Studio en lugar de escribir su código en un editor de texto.

Antes de poder utilizar las herramientas visuales, debe cumplir con los siguientes requisitos:

Usted debe tener un proyecto ejecutable abierto a depurar. Proyectos de bibliotecas de clases tales como los que hablaremos en el capítulo 6, por ejemplo, no se ejecutará sin una especie de concha visual. Ver las posteriores sección "Las bibliotecas de clases" para instrucciones.

Visual Studio debe estar en modo de depuración (también llamada en pausa o romper modo). Cuando pulse F5 o haga clic en el botón Reproducir para ejecutar el proyecto desde Visual Studio, usted está poniendo el proyecto en modo de depuración.

El proyecto debe ser una pausa para ver las variables de tiempo de ejecución. Te detienes el proyecto con un punto de interrupción, lo que me refiero en la siguiente sección. Además, puede introducir un proyecto en modo de interrupción, que cubro en la sección posterior "Depuración de los proyectos".

Visual Studio proporciona una serie de herramientas de depuración de diversa complejidad. Yo cubrir las tres herramientas más frecuentemente utilizadas:

Breakpoint: Un marcador que se coloca en una línea de código para detener la ejecución de un programa.

Vea ventana: Una ventana que muestra los valores de las variables en tiempo de ejecución y objetos.

Ventana Inmediato: Aparecerá una ventana que te permite escribir en tiempo de ejecución VB 2008 comandos y ver los resultados potenciales.

Puntos de interrupción

El punto de interrupción es tu amigo.

Permitanme sugerir un escenario. Usted tiene un complicado algoritmo que generan un precio final para el usuario. Dos propiedades de los objetos y las variables son tres utilizado para crear el precio final. Al probar la aplicación, el precio es mal.

¿Cómo se puede averiguar cuál es el problema? ¿Sabes cuál es la línea que se encuentra, pero usted no sabe los valores de las variables. Usted podría poner laboriosamente cinco etiquetas en el formulario y definir los valores de las etiquetas iguales a los dos propietarios y tres variables. Luego, cuando encuentre el problema, es necesario eliminar todas las etiquetas. Una mejor manera de encontrar el problema es establecer un punto de interrupción en ese línea y comprobar las variables mientras la aplicación está en pausa.



Para ver los valores de las variables, la aplicación debe estar en un estado de pausa, tal como el proporcionado por un punto de interrupción. Si tratas de ver valores de variables, mientras que

la aplicación se está ejecutando, no conseguirá lo que usted espera. Esta es la razón por desa-
opers utilizar puntos de corte.

La creación de puntos de interrupción

Cuando usted juega sus aplicaciones de Visual Studio, son en realidad em-
ing modo de depuración. De este modo, usted puede pedir Visual Studio para pausar la
ejecución de la aplicación en una determinada línea de código con un punto de
interrupción.

Para crear un punto de interrupción, haga clic en la barra gris a la izquierda de la línea
de código
a la que le gustaría que el programa se detenga. Esta acción deja un poco de rojo
punto en la barra gris, como se muestra en la Figura 8-1.

Cuando se reproduce el proyecto, la ejecución se detiene en ese punto. Presione F5
después de poner en un punto de interrupción y se detiene en la ejecución en esa linea
de código,
como se muestra en la Figura 8-2. Al pulsar de nuevo F5 continúa la ejecución del
programa
desde ese punto.

The screenshot shows the Visual Studio IDE with the DataCalc.vb code editor open. The code is as follows:

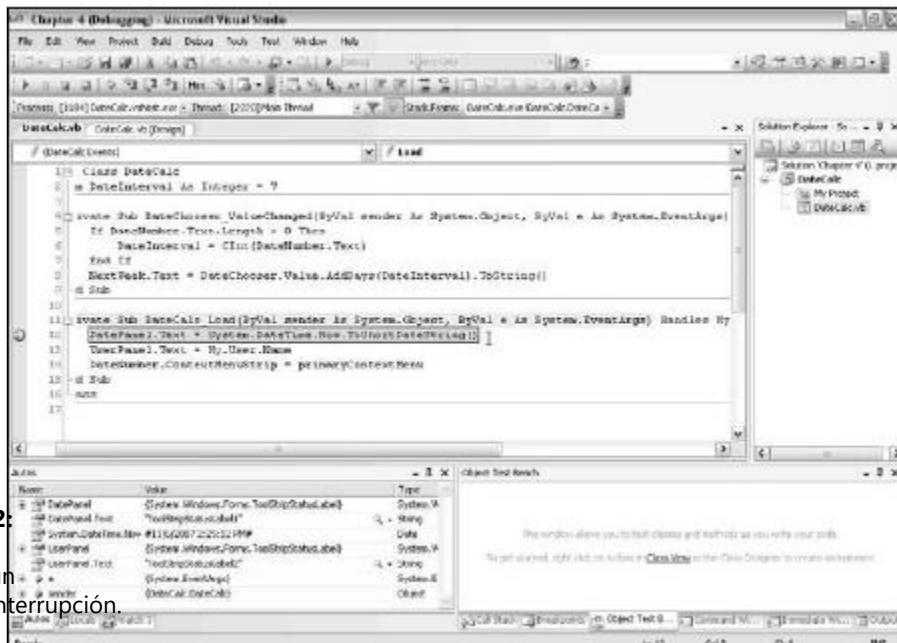
```
Class DataCalc
    Private DateInterval As Integer = 7
    Private DateChosen As DateTime = Now
    Private MemberText As String = "None"
    Private MemberName As String = "InterneUser"
    Private MemberValue As String = "InterneUser"

    Private Sub DateCalc_ValueChanged(ByVal sender As System.Object,
                                     ByVal e As EventArgs)
        If DateInterval.Text.Length > 0 Then
            DateInterval = CInt(DateInterval.Text)
        End If
        MemberName.Text = InterneUser.Value - InterneUser.DateChosen
        DateCalc.Text = DateChosen.ToString("dd/MM/yyyy")
        If DateChosen.Day < DateInterval Then
            DateChosen = DateChosen.AddDays(DateInterval)
        End If
        DateChosenContextMenuStrip = primaryContextMenu
    End Sub
End Class
```

A red dot, indicating a breakpoint, is visible on the left margin next to line 11. The Solution Explorer and Properties windows are also visible in the background.

Figura 8-1:
Realización
de una
punto de
interrupción.

Figura 8-2:
Parada
en un
punto de interrupción.



Gestión de puntos de interrupción

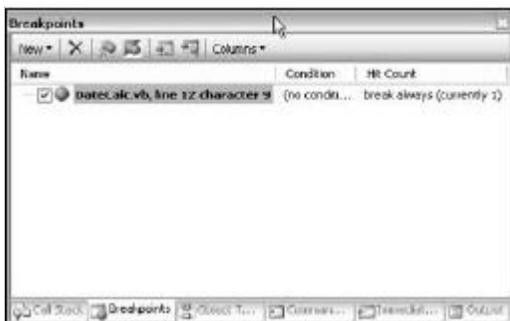
Avance rápido hasta el centro del proyecto de desarrollo, y es posible que tiene puntos de interrupción demasiados para ser útiles. Visual Studio proporciona una herramienta para que pueda administrar todos ellos - la ventana Puntos de interrupción, que se muestra en la Figura 8-3, que se invoca pulsando Ctrl + Alt + B.

El defecto de la ventana Puntos de interrupción es útil cuando está en un estado de pausa, o simplemente en modalidad de desarrollo normal. Las columnas por defecto incluyen la columna Recuento de visitas, que describe cuando al punto de interrupción durante la ejecución del código, y la columna Condición, que describe una expresión que debe evaluar como cierto para la ejecución de la aplicación para detener. Otras consideraciones para la decidir la forma de trabajar con puntos de interrupción incluyen los siguientes:

Puede agregar otras columnas de las ventanillas Breakpoint seleccionándolos de las columnas desplegables, incluidas las columnas que muestran cuál es la función en el punto de interrupción se encuentra, el idioma, y de golpe cuando. El Hit Cuando columna permite definir un mensaje a impresión o una macro se ejecute cuando el punto de parada es alcanzado.

Puede editar las funciones de depuración - tales como recuento de visitas, la condición, y Cuando Hit - haciendo clic derecho en el marcador de punto de interrupción a la izquierda de la código o haga clic en el punto de interrupción en la ventana Puntos de interrupción. La menú contextual que aparece contiene selecciones para cada una de estas opciones que le permiten manejar las funciones de punto de interrupción.

Figura 8-3:
La
Puntos de
interrupción
ventana



Breakpoint Properties simplemente hacer puntos de corte rápido de usar. Mientras que depuración, puede fácilmente acaba de establecer un punto de interrupción y de ir a buscar a valores para ver cuáles son los problemas. Uso de las opciones, sin embargo, reduce la número de pasos que debe seguir para obtener la respuesta que necesita.

Una estrategia de punto de interrupción se vuelve como un conjunto estándar de aperturas de ajedrez más tiempo. La experiencia dicta cómo se utilizan las herramientas de depuración, con base en su estilo de programación personal.

La ventana Inspección

Relojes son espías poco de programación que se pueden colocar en los objetos para mantener un ojo en sus valores de tiempo que se refuerza a través de código. Visual Studio proporciona cuatro ventanas Inspección, y sobre una base de proyecto por proyecto, se acuerdan de lo

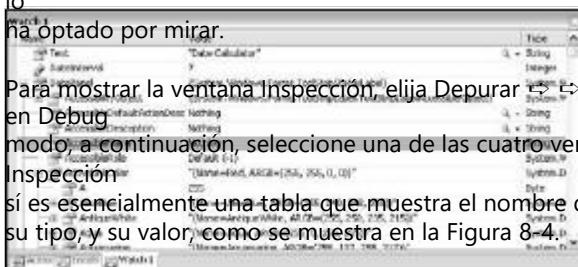


Figura 8-4:
El reloj
ventana

Para mostrar la ventana Inspección, elija Depurar → reloj de Windows, mientras que en modo, a continuación, seleccione una de las cuatro ventanas Watch. La ventana Inspección es esencialmente una tabla que muestra el nombre del objeto que se observaba, su tipo, y su valor, como se muestra en la Figura 8-4.

Para agregar un reloj a la lista de vigilancia, siga estos pasos:

- 1. Pausa el proyecto, ya sea llegando a un punto de interrupción o haciendo clic en el Pausa en la barra de herramientas.**
- 2. En la vista Código, haga clic en el objeto que desea ver y seleccione Agregar inspección en el menú contextual.**



Para ver el valor de una variable, debe estar en el ámbito. Una variable es en el ámbito cuando que existe dentro del bloque de código en ejecución. Por ejemplo, si se declara una variable dentro de un controlador de eventos para un botón, sólo cuando haga clic botón que hacen los valores de dicha variable esté disponible para ver.

Cuando en el modo de depuración, una variable es ya sea con o sin un valor, tal como es cuando se ejecuta un programa. La ventana Inspección muestra esta muy bien, como se muestra en Figura 8-5.



Figura 8-5:
Variables
dentro y fuera
de alcance.

Cuando la variable no tiene valor, aparece con el ícono de error y la siguiente texto:

Variable.Name no declarado o el módulo que lo contiene no está cargado en la sesión de depuración.

Cuando la variable tiene un valor, que se describe con todas las propiedades.

La ventana Watch es una gran manera de ver objetos enteros, en lugar de valores en las variables. Colecciones, tales como matrices y conjuntos de datos, a menudo tienen un amplia variedad de propiedades y valores que usted necesita para comprobar en cada descanso. La ventana Watch proporciona un método sencillo para un cheque estructurado en valores.

La ventana Inmediato

A veces, un reloj no es suficiente y es necesario ejecutar un comando, mientras que el aplicación está en pausa. La ventana Inmediato, que se muestra en el contexto de la Figura 8-6, está diseñado para tal situación.

Figura 8-6:
La
Inmediato
Ventana
en uso.

La sintaxis básica para la ventana Inmediato es el de la escritura de los valores de expresiones a la ventana utilizando el método Debug.Print, que me cubren en la siguiente sección. El atajo para este procedimiento es el signo de interrogación.

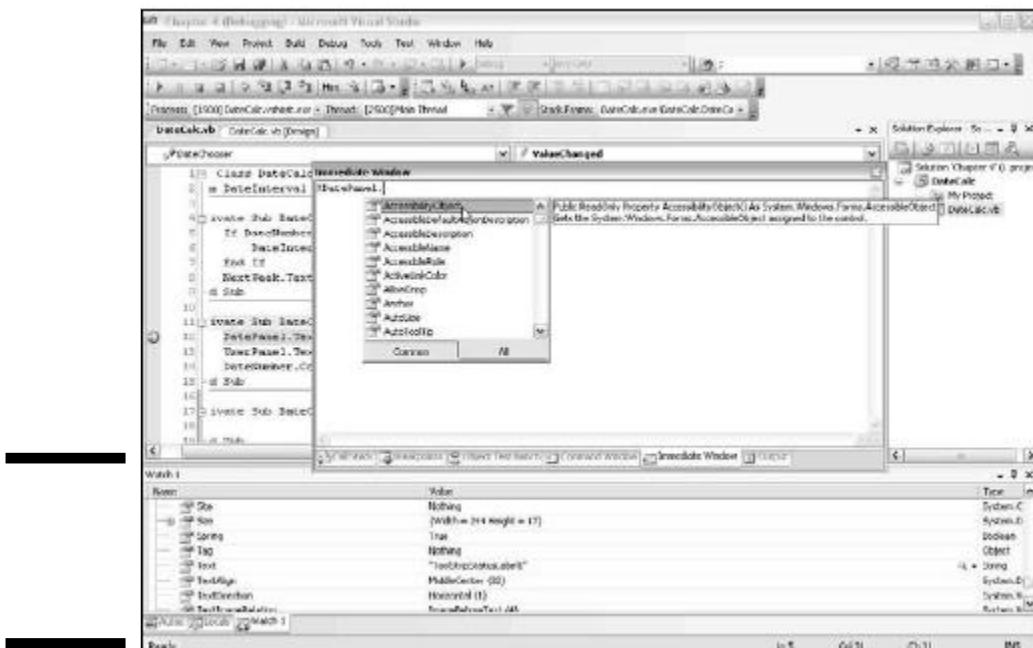
Para utilizar la ventana Inmediato, al igual que con la mayoría de otras herramientas de depuración, su proyecto es necesario detener. Para escribir el valor de una variable a la pantalla, a continuación, se

simplemente escriba? VariableName en la ventana. En el ejemplo de la Figura 8-6, He escrito? DatePanel. para obtener el valor de esa propiedad. Si el valor está fuera de alcance, me gustaría recibir el mismo fuera de alcance error que aparece en el reloj ventana.

Uso de la ventana Inmediato es más de un control al azar que usar muchas de las otras herramientas de depuración. Si usted está usando puntos de interrupción condicionales que sólo romper cuando las variables contienen ciertos valores, usted se encontrará con la ventana Inmediato para ver lo que provocó el paro.

Uso de las herramientas de depuración en el Marco. NET

rramienta para la depuración, pero el Marco .NET en sí también tiene algunas características fantásticas que hacen más fácil la depuración. Estas herramientas son más basada en código y aspectos visuales. Me acerco a algunos de ellos aquí, y



Luego cubrir un poco más cuando hable de depuración tipos específicos de proyectos después en este capítulo.

La clase Debug

¿Cuál sería la depuración sin una clase Debug, ¿verdad? Al igual que las ventanas anteriormente en este capítulo son una especie de la aplicación de Windows Forms depuración, la clase Debug DLL es la implementación de depuración.

La clase Debug tiene una gran cantidad de métodos, propiedades y eventos que ayudarle a ver lo que hace su aplicación mientras se ejecuta. El más simple ejemplo de un método de la clase de depuración es el método Write. El siguiente pasos para comenzar:

1. En Visual Studio 2008, seleccione Proyecto Archivo \Rightarrow / soluciones para abrir un Windows Aplicación del proyecto.

Por ejemplo, me abrió la DateCalcChapter4 de la Muestra aplicaciones.

2. Haga doble clic en un espacio vacío en el formulario para crear un Form_Load evento manejador.

En este ejemplo se crea una función DateCalc_Load que se encarga de la forma cargar evento.

3. Inserte un Debug.WriteLine declaración en la función DateCalc_Load, como sigue:

```
Private Sub DateCalc_Load (ByVal sender As System.Object e, ByVal args As System.EventArgs) Handles MyBase.Load
    Debug.WriteLine ("Formulario de Carga")
End Sub
```

4. Haga clic en el botón Reproducir en Visual Studio para entrar en modo de depuración.

5. La ventana de resultados muestra el mensaje.

Hay más en la clase Debug que la declaración de escritura, pero la mayoría de la uso común de la clase es para seguir la ejecución del programa, y la mejor manera de ejecución pista programa es dejar caer migas de pan como partes del programa ejecutar.

Gestión de errores

Tratamiento de errores y depuración van de la mano. La depuración es el acto de la pista-

ING por los errores y controladores de error están diseñados para, así, controlar los errores!

El manejo de errores es una gran parte de la depuración, ya que los errores que se crean por errores en el software deben ser capturados y manipulados. Una frase interesante describe el tratamiento de errores en Visual Basic - Try-Catch. Uno trata de una pieza de código.

Si se produce un problema, Visual Basic genera un error, y puedes coger ese error.

El bloque try-catch parece al código siguiente. Visual Studio escribe la mayoría de él para usted. Sólo tienes que escribir Pruebe en una función y presione Enter, y Visual Studio inserta automáticamente el resto del bloque.

```
Tratar de
Catch ex As Exception
End Try
```

Se puede ver que la instrucción Catch se está imponiendo algo en particular que merece un vistazo más de cerca - el objeto Exception.

El objeto de excepción es lo que hay detrás de Visual Basic en tiempo de ejecución cuando se produce un error. Es lo que Visual Basic arroja a usted, para que pueda atrapar con su tratamiento de errores.

Por ejemplo, siga los pasos descritos en la sección anterior para obtener una Form_Load método de control de eventos, excepto insertar el código en el Listado 8-1. Dentro de la Try bloque (línea 2 del listado 8-1), un entero es igual a una cadena, que se no puede hacer. (Para obtener más información, consulte el Capítulo 9.)

Esta prueba hace que se produzca un error, y un objeto de excepción es el resultado. En la parte Catch del bloque (línea 3 del listado 8-1), se puede llegar a la Excepción objeto con su ex declaración.

Listado 8-1: Causar un error en el controlador de eventos Form_Load

```
Tratar de
BugInCode As Integer = String.Empty
Catch ex As Exception
    Debug.Write (ex.Message)
End Try
```

En el modo de depuración, se puede ver el contenido del objeto de excepción por escribiendo? ex en la ventana Inmediato. Establezca un punto de interrupción en la depuración. Write (ex.Message) línea y ejecute el proyecto. Para obtener el código del listado 8-1, el objeto Exception devuelve toda la información útil para la inmediata Ventana:

```
Datos: {} System.Collections.ListDictionaryInternal
HelpLink: Nada
HResult: -2147467262
InnerException: {} System.FormatException
IsTransient: False
Mensaje: "La conversión de la cadena" "el tipo 'Integer' no es válida".
Fuente: "Microsoft.VisualBasic"
StackTrace: "en Microsoft.VisualBasic.CompilerServices.
    Conversions.ToInt32 (valor de cadena)
    en DateCalcChapter4.DateCalc.DateCalc_Load (
        Object sender, EventArgs e) en C:\Documents and Settings\Sempf\
        Mis documentos\Visual Studio\Projects\OSIA\DateCalcChapter4\
        DateCalcChapter4\DateCalc.vb: Línea 9 "
```

El objeto Exception devuelve una gran cantidad de información acerca de lo que pasó mal. Algunos detalles pueden estar ocultos en objetos que usted necesita para mirar separado, pero dos piezas principales de información están al frente y centro: El Mensaje propiedad (del objeto de excepción, que se ex nombre) tiene el error que ocurrido, y la propiedad StackTrace tiene el número de línea. Con esos dos piezas de información y la fecha que originalmente enviados al método, se tenemos lo que usted necesita saber el 80 por ciento de las veces.

Toda esta información se puede utilizar en el tratamiento de errores de nivel de aplicación. Usted Puede enviar esta información a ti mismo, lo devolverá al usuario, o grabarla en un registro archivo. Más información sobre las distintas maneras de hacer esto se pueden encontrar en la Web. Yo recomendaría los bloques de Excepción Microsoft Management componente, disponible a partir de los Microsoft Patterns and Practices sitio Web en <http://msdn2.microsoft.com/en-us/practices/default.aspx>.

Depuración de los Proyectos

Cada uno de los proyectos que cubren más temprano en la Parte II - Windows Forms, Web Forms bibliotecas de clases y servicios Web XML - tienen un conjunto similar de depuración herramientas. Los detalles que hablo anteriormente en este capítulo funcionan para todos los proyectos, pero cada uno de los tipos de proyectos tiene sus ajustes específicos. Cubro estos en el siguientes secciones.

Formularios Windows Forms

Aplicaciones de Windows Forms son los más fáciles de depurar, ya que son aplicaciones independientes sobre las que tiene control completo. La algunos trucos son necesarios para depurar Windows Forms, pero aprovecho esta oportunidad de cubrir la característica de depuración que utiliza con más frecuencia - la intensificación a través de código. Puede aplicar esta función a todos los tipos de proyectos, pero es mejor muestra como parte de una aplicación de Windows Forms.

Los "puntos de ruptura" anteriormente en este capítulo, se describe cómo utilizar break-points y demuestra lo poderosos que son. Usted puede usar este poder para ejecutar el código de una línea a la vez y mantener un ojo sobre los detalles del objeto propiedades y valores de variables. Los siguientes pasos a empezar paso a paso a través de código:

1. Carga de un proyecto de aplicación para Windows eligiendo Project File ⇒ / Solución

y seleccionar un archivo de proyecto.

Por ejemplo, yo uso el proyecto DateCalcChapter4.

2. Pulse F10 para iniciar la depuración del proyecto.

Esto se pone en marcha el proyecto en modo de depuración y para el proyecto en el primer punto encontrado. Para el ejemplo DateCalcChapter4, se Puede configurar un controlador en el Form_Load.

3. Para seguir paso a paso por el código, presione F10.

Esta guía a través del código de una línea a la vez en modo de interrupción. Usted puede utilizar la ventana Inmediato, o puedes pasar el ratón sobre los nombres de variables a ver sus valores.

4. Para pasar por encima de una llamada a un método, pulse F10.

Esto pasa sobre el código interno de un método para que usted permanece en el flujo del programa original.

5. Para entrar en un método, pulse F11.

Esto ingresa el código funcional de una llamada de método y puede cambiar el archivo usted está viendo.

6. Para continuar ejecutando el programa en modo de ejecución (es decir, para salir de Quiebre modo), presione F5.

Esto continúa para ejecutar el programa fuera del modo de depuración.

se puede
resolver la mayoría de los problemas de lógica y de ejecución que el programa pueda tener.

Los formularios Web Forms

Los formularios Web Forms son diferentes, como menciono en el capítulo 5, ya que se están ejecutando

en un Internet Information Server (IIS) en lugar de hacerlo directamente en su estación de trabajo ción. Mientras que usted puede estar ejecutando una aplicación de formularios Web Forms en su estación de trabajo

usando IIS o servidor de Visual Web Developer Web, todavía se considera depuración remota, ya que la aplicación está en manos de un independiente sistema.

Esto hace que todo los nuevos problemas con la depuración. En primer lugar, es posible que deba depurar

una aplicación de formularios Web Forms que no se está ejecutando en su estación de trabajo. En segundo lugar,

variables de entorno (por ejemplo, variables de sesión que discuto en

Capítulo 5) puede tener un impacto grande en su aplicación. Conociendo la

Los valores de estas variables es importante - en modo de interrupción o modo de ejecución.

La depuración remota

La depuración remota es necesaria porque si está ejecutando una Web Forms aplicación en un servidor que no tiene instalado Visual Studio, no lo hará tienen el programa necesario para permitir la depuración. Para instalar este programa, siga estos pasos:

1. Inserte el Visual Studio 2008 CD depurador remoto en el mando a distancia máquina.

2. El programa depurador Configuración remota se ejecutará automáticamente.

Si no es así, abra la unidad de CD desde el Explorador de Windows y ejecute el mando de Depurador programa de instalación.

A menudo, para depurar en un equipo remoto, debe tener un administrador cuenta en esa máquina. Cuando se abre un proyecto Web con un control remoto dirección, puede depurar el proyecto como si se estuviera ejecutando localmente en Visual Web Developer Web Server.

Rastrear

Parte de la información acerca de las aplicaciones Web no es la mejor obtenida a través de la depuración en modo de interrupción. A veces, si una página se puede ver en el modo Run, pero con información completa acerca de la ejecución de la página disponible poder, sus problemas pueden resolverse.

Traza permite precisamente eso. Trace se activa cambiando la directiva @ Page en la parte superior de un archivo ASPX. Para ello, siga estos pasos:

1. Abra un proyecto Web eligiendo Archivo de Proyecto ⇔ / soluciones y seleccionando un archivo de proyecto ASP.NET.

Para este ejemplo, yo uso DateCalcChapter5.

Capítulo 8: Depuración en VB 2008

2. Abra una página haciendo doble clic en él en el Explorador de soluciones.

Yo uso la página default.aspx que he creado con la Calculadora de la fecha en el mismo.

Tenga en cuenta la directiva @ Page en la parte superior de la página, al igual que el siguiente código.

Esto aparece en todas las páginas ASP.NET, y es lo que el servidor Web utiliza para vincular la página al archivo de código subyacente y establecer el idioma, entre otras cosas.

```
<% @ Page Language = "VB" AutoEventWireup = "false"  
    CompileWith = "Default.aspx.vb" ClassName = "Default_aspx"%>
```

3. Agregar una Traza atributo de la directiva @ Page, estableciendo Trace = "true", como sigue:

```
<% @ Page Language = "VB" AutoEventWireup = "false"  
    CompileWith = "Default.aspx.vb" ClassName = "Default_aspx"  
    Trace = "true"%>
```

4. Guardar los cambios y, a continuación, compile la aplicación seleccionando Construir => Sitio Web Generar.

5. Haga clic derecho en el archivo que ha cambiado y seleccione Ver en el explorador.

La información proporcionada por el trazo en la parte inferior de la página.

Traza funciona muy bien para los servicios Web XML también!

Al igual que el objeto Exception que discutir en este capítulo, Trace tiene una gran cantidad de información, bien organizada por ASP.NET. Las secciones incluyen la despu  s:

Detalles de la solicitud: Proporciona la informaci  n b  sica sobre la solicitud formulada a el servidor.

Traza la informaci  n: Detalles de la temporizaci  n de escal  n en escal  n en la solicitud, lo cual es muy importante para descubrir problemas de rendimiento.

Control de \'rbol: Muestra todos los controles de servidor y de usuario que utiliza el aplicaci  n.

Estado de sesi  n y estado de aplicaci  n: Muestra el contenido de la Variables de sesi  n y de aplicaci  n en el tiempo de respuesta.

Galletas de solicitud de recogida y recolecci  n de respuesta Cookies: Detalles las colecciones de galletas en el tiempo de respuesta.

Headers Colecci  n Colecci  n de respuesta y encabezados: Muestra el standar encabezados HTTP, normalmente usados para depurar problemas del servidor Web.

spuesta

Encabezados Collection es lo que salió al cliente.

Forma Colección: Da los valores de todos los campos del formulario enviado al servidor.



QueryString Colección: Muestra los valores de las variables enviadas al servidor en la URL.

Variables del servidor: Muestra un conjunto estándar de variables pasadas entre todos clientes y servidores, independientemente de la plataforma o middleware.

Se puede ver que esta información es muy valiosa para depurar problemas con las aplicaciones Web. Incluso hay más, también, porque con el seguimiento clase en el Marco .NET, puede insertar avisos en el código que solo aparecen en el modo de rastreo. Porque cambiar de una página a Trace modo sólo requiere un cambio en el archivo ASPX, incluso se puede hacer en un sistema de producción. Es una herramienta muy poderosa, más pueden ser descubiertos mediante la búsqueda de ASP.NET *Rastrear* en la biblioteca de MSDN en <http://msdn.microsoft.com/library>.

Las bibliotecas de clases

Las bibliotecas de clases son un problema de depuración interesante sólo porque son no ejecutable por sí mismos. Como menciono en el capítulo 6, las bibliotecas de clases son utilizado por otras aplicaciones para componentizar funcionalidad. Por esta razón, que solo son utilizables como parte de otras aplicaciones.

El capítulo 6 describe cómo utilizar dos proyectos en una solución. Eso es lo que hay que hacer para depurar una biblioteca de clases. Si se ejecuta una aplicación de Windows Forms-
ción que hace referencia a una biblioteca de clases, y usted tiene el proyecto de refe-
biblioteca de clases mentado en la misma solución, cuando pasa a través del código de
la aplicación de Windows Forms, la aplicación de los pasos a la derecha en la biblioteca de clases
cuando se llama a un método o una propiedad de la biblioteca de clases.

Para que esto suceda, siga estos pasos:

1. Abra una solución de biblioteca de clases en Visual Studio.

Para este ejemplo, yo uso la biblioteca de clases que he desarrollado en el Capítulo 6,
DateCalcClassChapter6.

2. Agregar un proyecto a la solución mediante Archivo ⇔ ⇔ Agregar proyecto Nuevo.

**3. Seleccione un proyecto de Visual Basic Aplicación para Windows en el nuevo complemento
Cuadro de diálogo, que sea algo apropiado y haga clic en el botón Aceptar
botón.**

Llamo DateCalcShell mío.

**4. Haga clic derecho en el nuevo proyecto y seleccione Agregar referencia en el contexto
menú.**

**5. Haga clic en la ficha proyectos, seleccione el proyecto de biblioteca de clases de archivos, y
luego
haga clic en el botón Aceptar.**

6. Haga doble clic en el Form1 formulario para obtener un controlador de carga de página.

Capítulo 8: Depuración en VB 2008

7. Agregue el código que llama a una función del proyecto de biblioteca de clases.

En mi ejemplo, me acaba de agregar una variable llamada miDate rapidito que me puse a una semana a partir de ahora mediante la función IncreaseDate, como sigue:

```
Private Sub Form1_Load (ByVal como System.Object remitente e, ByVal como  
System.EventArgs) Handles MyBase.Load  
    DateCalculator dévil como nuevo CalcClass2008.DateCalc()  
    MiDate As DateTime  
    miDate = dateCalculator.IncreaseDate (System.DateTime.Now, 7)  
End Sub
```

8. Pulse F10 para entrar en la aplicación. Continúe presionando F10 para moverse

Línea por línea en el código.

Como paso en la línea 4 del código anterior, el depurador abre la proyecto de biblioteca de clases y le guía a través del código de la Ampliación Función de fecha y, a continuación, volverá al controlador de eventos Form_Load.

Si utiliza Visual Basic 6 o 7, es posible reconocer esta característica - que ha sido alrededor de un rato. El nuevo sabor orientado a objetos de Visual Basic que hace mirar un poco diferente, pero en realidad es mucho lo mismo.

Los servicios Web

Los servicios Web son, con mucho, los más difíciles de depurar. Al igual que con la clase bibliotecas, usted tiene que utilizar algunos trucos para depurar servicios Web. Después de obtener pasado esos trucos, es muy parecida a la depuración de cualquier otro tipo de proyecto.

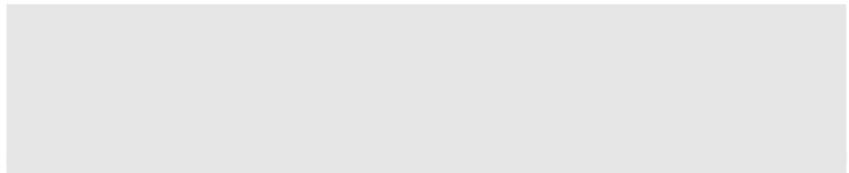
La clave es recordar acerca de la depuración de servicios Web es que usted no puede depurar a menos que tenga acceso al código fuente del propio servicio. Si está utilizando algún otro servicio, por ejemplo, y un error es en el código, el depuración no le muestra toda la información de depuración avanzada. Si tienen el código para el servicio Web y la aplicación cliente, puede depurar ambos.

Si está depurando un servicio Web que usted ha desarrollado a nivel local - como describe en el Capítulo 7 - sólo tiene que pulsar F10 e ingrese al modo de interrupción el paso a través del código, tal como en los ejemplos precedentes en este capítulo. Si el servicio se está ejecutando en un servidor Web, puede que tenga que adjuntar a el proceso que se está ejecutando en ese servidor.

Para adjuntar a un proceso para depurar un servicio Web, haga lo siguiente:

1. Abra un proyecto de servicio Web.

Para este ejemplo, yo uso el proyecto que he desarrollado en el capítulo 7 se llama



2. Seleccione Herramientas \Rightarrow asociar al proceso.

La Asociar al proceso cuadro de diálogo, como se muestra en la Figura 8-7.

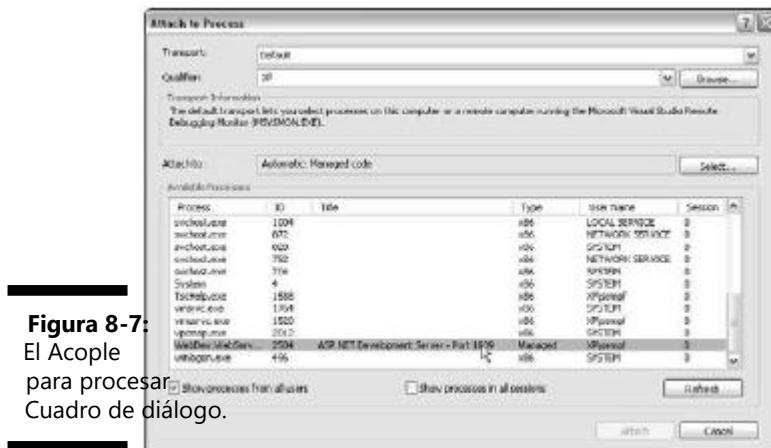


Figura 8-7:

El Acople
para procesar
Cuadro de diálogo.

3. Seleccione el servidor que ejecuta el servicio Web desde el menú Qualifier desplegable.

4. En el cuadro Procesos disponibles, elija aspnet_wp.exe o w3wp.exe.

El archivo aspnet_wp.exe se utiliza en Windows XP y los servidores de 2000; w3wp.exe es para Windows Server 2003.

5. Haga clic en el botón Adjuntar.

6. Haga clic en el botón Aceptar.

Al presionar F8 para entrar en el proyecto, Visual Studio puede ver el código running en el servidor remoto al que se haya seleccionado en el Asociar al proceso Cuadro de diálogo.



Este método también es útil para las acciones de la depuración de otros. Para obtener más información, búsqueda de depuración implementado aplicaciones ASP.NET en MSDN Library, <http://msdn.microsoft.com/library>.

Parte III

Cómo hacer que su Programas de trabajo

The 5th Wave

By Rich Tennant



"You might want to adjust the value on
your 'Nudge' function."

En esta parte. . .

Manera de la programación de los detalles del negocio lógica - Las reglas del programa que está escribiendo. Esta parte explica cómo tomar decisiones, hacer cosas más de una vez, y utilizar y reutilizar el código que ya está fuera allí, maduros para la cosecha, para hacer su trabajo más fácil.

Capítulo 9

Cadenas de Interpretación y Cosas

En este capítulo

Averiguar cómo Visual Basic almacena el más básico de información

La manipulación de la información

Validar entrada de usuario

Trabajo con números y fechas

Cambio de entrada de un tipo a otro

Té el núcleo de un lenguaje de programación orientado a objetos, tales como Visual Basic es el movimiento de la información. Algunos lenguajes de programación especializan en mantener un vínculo con el hardware y algunos se especializan en el manejo de la maquinaria, pero se especializa en Visual Basic información.



Cómo Visual Basic almacena la información internamente es de gran importancia para usted, el desarrollador. Las palabras, números, imágenes digitales y otros lugares dentro de la equipo cada uno tiene sus propios tipos especiales. Estos tipos tienen su propia métodos, propiedades y eventos porque son tratados como objetos en Visual Basic 2008.

En este capítulo, se describe esta característica del lenguaje, mostrando cómo las palabras, números y fechas (todos los tipos) se tratan cuando se refiere a los programas. Yo discutir la funcionalidad que un número entero adquiere automáticamente como parte de un programa, y las cosas interesantes que puedes hacer con el texto que el usuario introduce.

Me acerco cómo utilizar los tipos a su mejor ventaja, también. Por ejemplo, la validación de tipo es esencial para asegurarse de que el usuario introduce la información correcta (Números, texto, etc) en la aplicación. También puede manipular la información de formas interesantes mediante el uso de herramientas fantásticas llamadas expresiones regulares.

Brevemente cubrir las fechas y matemáticas fecha en este capítulo. Tienes un montón de subdestacan sobre las fechas. Para llegar allí, sin embargo, es necesario empezar por el principio - al descubrir las diferencias entre los tipos.

Cambio de un tipo a otro también se trata en este capítulo. Por ejemplo, que puede manejar el cambio de un número a una palabra de varias maneras diferentes. Algunos de estos métodos son los más utilizados en situaciones específicas, lo que describo en este capítulo.

Tipos de Información en Visual Basic

Los programas de ordenador, en su forma más básica, tienen dos tipos de almacenamiento: volátiles y no volátil. Almacenamiento volátil es el almacenamiento que utiliza el programa mientras se está ejecutando; cuando el programa se detiene, la información almacenada se ha ido. Es físicamente almacenados en la memoria de acceso aleatorio (RAM) de su PC.

Almacenamiento no volátil es el almacenamiento permanente, como bases de datos y archivos de texto, yo cubrir el almacenamiento no volátil en la Parte IV. Información no volátil se almacena generalmente en el disco duro.

Volátiles información se almacena en "cubos" marcados por el tipo de información que puede ir en ellas. Las palabras, números y fechas, por ejemplo, son diferentes tipo de información. Se almacenan en la memoria volátil en cubos llamados variables, que son ordenados por tipo.

Algunos de los tipos de información que a menudo tienen que almacenar en las variables incluyen los siguientes:

Texto: Las palabras se suelen almacenar en el tipo String.

Fechas: El tiempo y las fechas se almacenan en el tipo DateTime.

Números: Los números se pueden almacenar en varias formas diferentes - si es un especialista en matemáticas, usted podrá apreciar los tipos Integer y Double. *Enteros* son números enteros, fracciones y dobles son almacenados como decimal números.

Cuando se trabaja en Visual Basic, la mayor parte de su tiempo de programación se dedica a autRING el código para aceptar la entrada de los usuarios, averiguar qué tipo de poner el introducidos y ponerlo allí.

Descripción de los tipos en Visual Basic

Para crear una nueva variable de un determinado tipo en Visual Basic, se utiliza el Dim comunicado. Dim es sinónimo de dimensión, y se refiere a los viejos tiempos cuando necesario dejar a un lado un paquete de memoria para almacenar el contenido de la variable. Eso es realmente lo que está pasando todavía, así que los programadores siguen utilizando el término. Por ejemplo, para declarar una nueva cadena, que le codificar lo siguiente:

```
MyNewString Dim As String = "Este es el contenido de mi variable de cadena!"
```



Los valores de cadena están rodeados por comillas. Si usted necesita tener un statu-tación marca en una cadena, utilice dos comillas, por ejemplo: "Yo uso el palabra "," cuchara "," demasiado en este capítulo. " El uso de dos citas marcas se llama escapar de la marca. Otros valores de tipo, como los números y Se hace referencia a las fechas, sin comillas.



Una cadena es un objeto, como todos los tipos, y a veces el valor de otro objeto es de tipo String. Esto puede ser confuso cuando los objetos son suficientes en la página. Esencialmente, usted sólo necesita una dimensión nuevo objeto de cadena cuando se tenga que manejar individualmente.

Los tipos, tales como la cadena y entero, disponibles para su uso son muchas y variadas.

Tabla 9-1 cubre los tipos más utilizados. Usted debe saber que hay más tipos existe y que se puede crear el suyo propio.

Tabla 9-1**Tipos intrínsecos de Visual Basic**

Tipo	Descripción
Byte	Un número de un solo dígito, como 8.
Char	Un personaje único, como r.
DateTime	Una fecha y hora juntos, como 4/3/2004 12:45:54 PM.
Doble	Un número decimal, como 4,534.
Entero	Un número entero, como 56386.
Objeto	Lo que sea. Un tipo de objeto puede contener cualquier cosa en la red. Marco.
Cadena	Las palabras, como Esto es una cadena.

Cuando obtiene los datos de un usuario o de otra fuente, como una base de datos, lo hará ya tiene un tipo, por lo general String o Object. Para utilizar los datos para hacer matemáticas, por ejemplo, es necesario cambiar esa cadena o un objeto en el Integer o Tipo Double. Esto se hace con CType.

Cambio de tipos con CType

CType es la forma en Visual Basic de lo que le permite cambiar el tipo de una variable. Al ver cómo funciona esto se hace mejor con un ejemplo del mundo real. Seguir estos pasos para cambiar el tipo de una variable:

- 1. Abra Visual Studio 2008 y elija Archivo \Rightarrow New Project.**
- 2. Seleccione un proyecto de aplicación de Windows, y el nombre en el nombre cuadro de texto.**

Nombré StringsAndThings minas. También es posible usar un Web Forms proyecto para estos pasos.

3. Cambiar el nombre de la solución para el capítulo 9 y haga clic en el botón Aceptar.

Esto no es un requisito, pero le ayudará a mantener las cosas organizadas si optar por añadir otros proyectos relacionados con este capítulo.

4. Coloque dos cuadros de texto, una etiqueta y un botón en el formulario predeterminado, como

muestra en la Figura 9-1.



Usted puede utilizar el Centro de Formato \Rightarrow función en forma horizontal al centro los campos, si así lo desea.

Para este y todos los capítulos de la parte III, os dejo los nombres predeterminados para simplicidad. Nunca haga esto en una aplicación de producción. Esto hace que la aplicación difícil para depurar e imposible para alguien de mantener.

5. Haga doble clic en Button1 para crear un OnClick controlador de eventos.**6. Coloque el código siguiente en el controlador de eventos Button1:**

```
Label1.Text = TextBox1.Text + TextBox2.Text
```

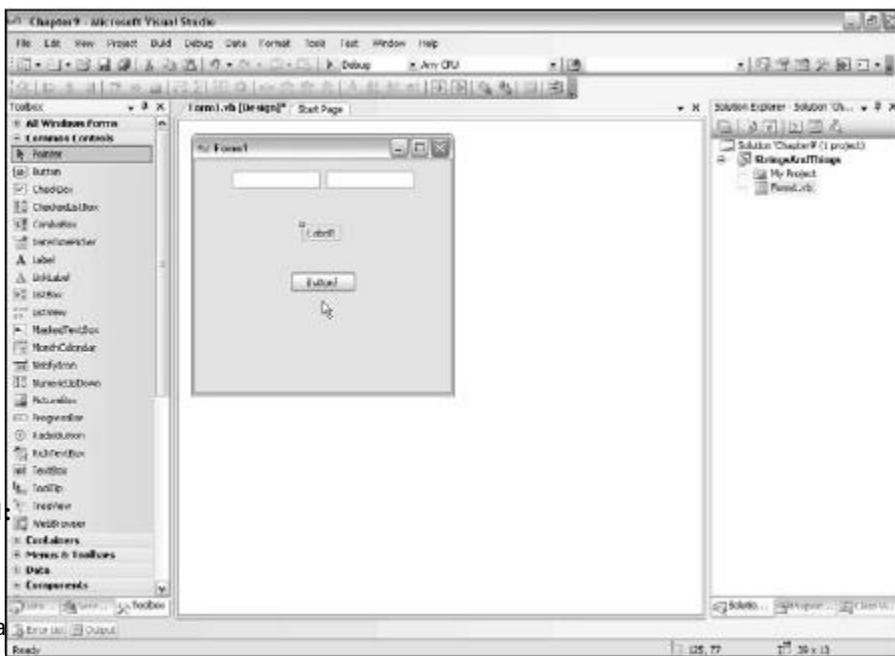
7. Presione F5 para ejecutar el programa.**8. Entrar 4 en el primer cuadro de texto y 56 en el segundo cuadro de texto.****9. Haga clic en Button1.**

Figura 9-1:
La
StringsAnd
Cosas
muestra
aplicación.

La Figura 9-2 muestra los resultados. Whoops. Debe ser nueva matemática: $4 + 56 = 456$? ¿Qué que pasó?



Figura 9-2:
Nueva
matemáticas.

Lo que sucedió fue que el programa concatena dos cadenas (puso un `,` tras otro) en lugar de la adición de dos números. Visual Basic supone que - porque no se diga lo contrario - las dos piezas de información ingresada en los cuadros de texto son cadenas. Esto es usualmente una buena suposición, pero en este caso, es incorrecto.

Entonces, ¿qué hace usted? Es necesario indicar a Visual Basic que esos valores son enteros. Para ello, se utiliza la función `CType` mencionado. Trate de poner el siguiente código en el controlador de eventos `Button1`:

```
Label1.Text = CType (TextBox1.Text, Integer) + CType (TextBox2.Text, Integer)
```

La declaración indica a Visual Basic `CType` que puede esperar enteros desde el Propiedad `Text` de `TextBox1` y `TextBox2`. Cuando se ejecuta la aplicación de nuevo y entrar en los mismos valores en los cuadros de texto, se obtiene el valor correcto en la etiqueta cuando se hace clic en el botón.



Algunos accesos directos a la cuenta de `CType` son retrocesos a versiones anteriores de BASIC:

`CStr` arroja desde un objeto a una cadena. (Para la emisión es describir la acción de utilizar un `CType`.)

`CInt` arroja desde una cadena a un entero. Usted podría utilizar `CInt` en el preceding línea de código si quieras, de la siguiente manera:

```
Label1.Text = CInt (TextBox1.Text) + CInt (TextBox2.Text)
```

`CDate` pone cadenas a fechas.

`CBool` proyecta una cadena o un entero a un Boolean - Un valor true o false.

COBL arroja algo a un objeto - útil si usted está interactuando con un mayores idioma.

CDbl arroja tipos a un doble, que es un número decimal.

Utilización de una sentencia CType tiene ciertas implicaciones, sin embargo. ¿Qué sucede si el usuario introduce algunas letras o palabras en los cuadros de texto en lugar de números? En ese caso, Visual Basic proporciona el mensaje de error se muestra en la Figura 9-3.

Visual Basic genera un error InvalidCastException porque no se puede cambiar una letra en un número entero fácilmente - o en absoluto. Como se suele decir, que no se puede hacer que.

Entonces, ¿qué hace usted? Hay que obligar al usuario a introducir sólo números enteros en este caso. Eso se maneja en el lado de la interfaz de usuario de las cosas. Lo que es más, me da la oportunidad de introducir otra nueva idea en tipos - con ayuda de la Escriba como una clase estática.

En el siguiente ejemplo, vamos a utilizar un método del objeto Integer, llamado TryParse. Es como cualquier otra función, excepto que acepta su devolución valor como un parámetro, llamado un parámetro out. Discutir clases y métodos en más detalle en los capítulos 12 y 13.

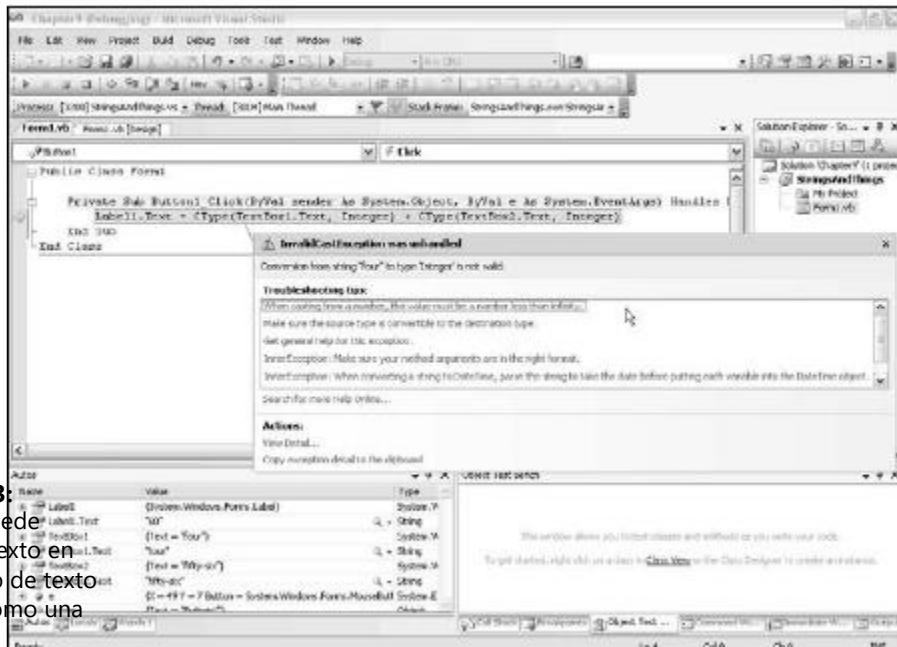


Figura 9-3:

No se puede introducir texto en un cuadro de texto elegida como una entero

Tipos de control con una validación

Usted puede hacer la validación en formularios Windows Forms en muchas formas diferentes - y en

Web Forms para el caso. He seleccionado una de las muchas maneras debido a que funciona bien en aplicaciones sencillas. Para restringir al usuario ingresar sólo una tipo específico de información (números enteros, en este caso), siga estos pasos:

- 1. En la StringsAndThings proyecto en el que hemos estado usando, cambiar a Ver código.**
- 2. Agregue dos nuevas variables en la sección Declaración, justo debajo de la Formulario de declaración:**

```
PrimerNumero Dim As Integer = 0  
SecondNumber Dim As Integer = 0
```

- 3. Cambie el Button1_Click código de evento a lo siguiente:**

```
Tratar de  
    "Buscar enteros primera  
    Si no Integer.TryParse (primerNumero TextBox1.Text), entonces  
        Lanza Nueva ApplicationException (_  
            "El primer número debe ser un entero")  
    End If  
    Si no Integer.TryParse (TextBox2.Text, secondNumber) A continuación,  
        Lanza Nueva ApplicationException (_  
            "El segundo número debe ser un entero")  
    End If  
  
    "Haz la cuenta si no hemos errado fuera.  
    Label1.Text = primerNumero + secondNumber  
  
Catch ex As Exception  
    MessageBox.Show (ex.Message, "Error de entrada", _  
        MessageBoxButtons.OK, MessageBoxIcon.Error)
```

```
End Try
```

- 4. Elija Depurar ⇒ Iniciar depuración para ejecutar el programa.**
- 5. Intenta entrar en algo que no sea un número entero en uno de los cuadros de texto.**

El nuevo código le avisará con el mensaje de error que ha introducido.

Obviamente, he añadido un montón de código para este ejemplo sencillo. Una regla no escrita dice que por cada línea de código funcional, cuatro líneas de código asegurarse de que funciona. Cuando se agrega en el código, usted tiene que añadir más código para asegurarse de que el usuario no ha introducido datos incorrectos. . . así, se obtiene la idea.

¿Cómo funciona esto es bastante sencillo. De ello se deduce que los principios de programación utilizan en los ejemplos más complejos:

Para empezar, establezca las variables de tenencia de los valores que desea trabajar.

Prueba de los valores en los cuadros de texto, y si los valores son válidos establecido, el la celebración de las variables iguales a los valores probados.

TryParse es muy conveniente para esto, porque devuelve False si el cambiar a un entero no funciona y deja que la variable que contiene fue aprobada en cambios. Discutir TryParse al final del capítulo.

Si todos los que trabajaban, hacer el código de función original, con la participación variables.

Ya hemos acabado con las cajas de texto.

Si no funciona, usted ha envuelto todo en un callejón sin Try bloquear.

Esto informa al usuario del problema y podría poner en práctica cualquier mundial manejo de errores que usted ha creado.

Las palabras hagan que el trabajo con el tipo de serie

Cuando quiero decir, no son números, se trata de la cadena de tipo. Tradicionalmente, los lenguajes BASIC - Visual Basic incluido - eran débil en el manejo de cadenas. Debido al marco. NET Visual copias de seguridad Basic 2008, muchos de estos problemas han desaparecido.

Las herramientas fantásticas construidas en cadenas

Cuando se declara una cadena y llénala, la cadena se convierte en un objeto con sus propios métodos, propiedades y eventos. Para empezar a utilizar cuerdas, abrir una nueva aplicación de Windows Forms y agregue un botón, dos cuadros de texto y un etiqueta, tal como se describe en la sección anterior "Cambio de tipos con CType." Entonces agregar una cadena de título de su programa, siga estos pasos:

- 1. En la vista Diseño, arrastre una segunda etiqueta en el formulario.**
- 2. Haga doble clic en una parte vacía del formulario para cambiar a vista de código con un Controlador de eventos Form1_Load.**
- 3. Escriba el siguiente código en el Controlador de eventos Form1_Load:**

```
TitleString Dim As String = "Este es mi programa de ejemplo".  
Label2.Text = TitleString
```

Capítulo 9: Interpretación Strings and Things

Al ejecutar la aplicación, debe tener un título en la forma en la que cayó la segunda etiqueta. Ahora que tiene un objeto String en el programa, usted tiene la oportunidad de ver algunas de las cosas interesantes que se puede hacer con un objeto String.

Por ejemplo, mire el método ToUpper. En lugar de Label2.Text = TitleString, escriba lo siguiente:

```
Label2.Text = TitleString.ToUpper.
```

El formulario muestra la versión en mayúsculas de la cadena sin cambiar el cadena original! Para reemplazar un carácter especificado con otro carácter, utilice el método Replace, como sigue:

```
Label2.Text = TitleString.Replace ("í", "I")
```

El objeto String tiene 46 métodos, propiedades y eventos integrados, y todos ellos están disponibles en cualquier cadena declarado. Más información mediante la búsqueda de "Cadena de clase, los métodos" en los archivos de Ayuda.

Sería negligente si no mencionara String.Format. Es una herramienta que se construye en el tipo String en sí - no cualquier cadena particular. Se utiliza una cadena-formato básico: se refiere a una discusión con un ordinal. Por ejemplo:

```
MessageBox.Show (String.Format ("El texto de Label2 es {0}", Label2.Text))
```

El ordinal en las llaves que se refiere al primer argumento. Si otro argumento seguido de la declaración Label2.Text, que se conoce como {1}.

Esencialmente, esta es una otra manera de construir cadenas de texto, al igual que concatenación y la clase StringBuilder. (Para más información acerca de StringBuilder, consulte la sección "La construcción de cadenas con la clase StringBuilder," más adelante en este String.Format capítulo.) es ideal para la construcción de URLs de las páginas ASP.NET.

Emptiness - manejo de valores nulos

Dado que las cadenas son objetos, se puede configurar para varios valores que uno No pienses en como cadenas. Principalmente, estos valores son diferentes formas de decir que la cadena está vacía.

Por ejemplo, un valor predeterminado se denomina String.Empty. Lo que es igual a, lo preguntas?

""

Yup. Nada. Una propiedad general para referirse a nada. ¿Por qué? En caso de que el valor de

adena = "".



El peor de estos valores vacíos son nulos, variables llenas de nada, ni siquiera cero o una cadena vacía. Nulos vienen en dos sabores - los asignados por el .NET Framework y las dadas a usted por bases de datos. Los valores nulos de base de datos, Además, vienen en un solo sabor para cada tipo de base de datos. (No, no soy chicode-ding). cubro bases de datos en el capítulo 15.

Los valores nulos utilizados por el framework son bastante simples. Configuración de una cadena a la valor nulo es lo mismo que decir que es igual a nada - ni String.Empty, "no", no es 0, pero en realidad nada.

Para ver si una cadena contiene un valor nulo, se puede utilizar el IsDBNull método integrado en Visual Basic. Cubro If-Then declaraciones en el capítulo 10, pero aquí hay una vista previa:

```
Si IsDBNull (MiCadena) A continuación,  
    MessageBox.Show ("Esa cadena es nula")  
End If
```



Francamente, los valores nulos se debe evitar debido a la necesidad de comprobar por ellos a cada paso. No es necesario el uso de valores nulos para codificar buenos programas. En el flujo de su aplicación, asegúrese de que cada variable tiene un tipo y un valor.



Para confundir más las cosas, no es cualquier variable se puede ajustar para que sea anulable. Yo no recomiendo esto, porque una variable nula es usualmente un signo de un problema y deben ser manejados como el error de que por lo general representa, un Null ReferenceException. Sin embargo, si usted lo necesita, sólo tiene que añadir una pregunta marcará el final de la declaración, así:

```
MiCadena Dim como cadena?
```

Esto es en realidad la abreviatura de la expresión genérica siguiente:

```
MiCadena Dim como Nullable (Of String)
```

Ambos hacen lo mismo. Ahora, además de estar vacía, puede ser MiCadena nulo o Nothing en dialecto VB. Es más o menos la misma que la base de datos nula, salvo que pueda probar por lo siguiente:

```
Si IsNothing (MiCadena) A continuación,  
    MessageBox.Show ("No es nada")  
End If
```

Herramientas para la Búsqueda de Entrada de usuario Gestión

Continuando con el hilo de discutir los valores recibiendo de los usuarios, es posible que

necesidad de gestionar las aportaciones que recibe de los usuarios después de la entrada se vuelve valores dentro de su sistema. Cuerdas, especialmente, están sujetos a la manipulación, ya sea mediante la creación de nuevas cadenas de producción o cambiando cadenas existentes para almacenamiento.

La clase `StringBuilder` es una herramienta fantástica que era nuevo para la RED. Framework 1.0, y se ha actualizado a la versión 3.5 de la .NET Framework. Te permite hacer sistemáticamente las decisiones sobre cómo hacer que las cadenas grandes de muchas cadenas pequeñas.

Varios de gama alta libros de programación se dedican por entero a expresión regular *siones*, que es un lenguaje especial dedicado a la manipulación de los patrones de cadena. Yo no cubrir ni el 10 por ciento de lo que hay que saber acerca de las expresiones regulares aquí, pero te dan suficiente información para que pueda leer un libro sobre expresiones regulares sin que se pierdan.

La construcción de cadenas con la clase `StringBuilder`

`StringBuilder` es una clase que está diseñado para ayudarle a manipular cadenas. A menudo, los se utiliza para la creación de cadenas de salida a partir de diversas fuentes de entrada, tales como un base de datos, una entrada de archivo de entrada, o el usuario.

La clase `StringBuilder` es parte del Marco. NET que no está incluido en el proyecto de forma predeterminada. Para utilizarlo, es necesario añadir una nueva línea de código a la parte superior de la vista de código. La nueva línea 1 es la siguiente:

```
Imports System.Text
```

Es necesario agregar esta línea porque la clase `StringBuilder` es realmente la `System.Text.StringBuilder` clase. Para hacer referencia a ella, es necesario utilizar la instrucción `Imports`. Puede hacer referencia a un objeto con la ruta completa, pero es más fácil de añadir la referencia `System.Text`.

I código
para el controlador de eventos Form1_Load. A continuación, puede crear nuevas
cadenas!

Las funciones de pequeños grandes disponibles en la clase `StringBuilder` incluir la después:

Anexar: Añade el texto proporcionado hasta el final de la cadena original.

Insertar: Coloca el texto proporcionado en la cadena original se especifica en el ubicación.

Quitar: Toma una serie de caracteres de la cadena.

Reemplazar: Al igual que en el método `Replace` que describo en la sección "La herramientas fantásticas construidas en cadenas ", anteriormente en este capítulo, este método sustituye a instancias especificadas de las cadenas con la cadena que usted provee.

La clase `StringBuilder` realiza manipulaciones de cadenas que se pueden hacer otras maneras. Sin embargo, es una solución muy elegante a un problema que se quiere frente todo el tiempo: los programadores de Visual Basic debe constantemente cadenas de puntada juntos y luego volver atrás y hacer cambios basados en requisitos cambiados. La clase `StringBuilder` hace que los cambios de ejecución a las cadenas más fácil si usted lo usa para empezar.

Por ejemplo, tomemos un vistazo al código que se puede utilizar para construir un nuevo cadena de título, como se describe en la sección anterior "las fantásticas herramientas integradas en cadenas. "Reemplazar las líneas de código que asignan el título... .

```
TitleString Dim As String = "Este es mi programa de ejemplo".
Label2.Text = TitleString
```

... con las siguientes líneas:

```
Dim sb StringBuilder
sb.Append ("Este")
sb.Append ("es")
sb.Append ("a")
sb.Append ("title").
sb.Insert (10, "nuevo")
Label2.Text sb.ToString = ()
```

Este código escribe un título en `Label2` que dice "Este es un nuevo título." El Instrucción de inserción pone la nueva palabra en el medio de la cadena - algunos- cosa que es muy difícil de hacer en el curso de la lógica de programación. La clase `StringBuilder` hace esto para usted - y lo hace más rápido y mejor que cualquier otro método.

Manipulación de cadenas con expresiones regulares

Después de una cadena se construye, a menudo es necesario buscar o modificar la cadena. Regular expresiones, una parte tradicional del lenguaje Perl, son una forma compleja para

Capítulo 9: Interpretación Strings and Things

gestionar cadenas utilizando fantásticos, cadenas complejas, codificados innovadoras para describir lo que hay que cambiar.

Las expresiones regulares se basan en patrones. Los patrones son lo que suenan como - las combinaciones de caracteres que fueron reconocidos por una definición, tal como [a-z] para todas las letras en minúscula. La complejidad viene cuando se intenta y hacer que la definición.

Las expresiones regulares se utilizan para una serie de cosas maravillosas:

- Búsqueda de una cadena de valores, a la derecha dentro de su código
- Edición de cadenas utilizando un patrón
- Validar la entrada del usuario con los valores demasiado complicadas para ser mostrado como un
- cadena

Dos componentes principales forman un patrón:

Literales: Representaciones exactas de una cadena que está buscando - como en los ejemplos anteriores en este capítulo. "0" es un literal, como "muestra".

Metacaracteres: Descripciones de las categorías de personajes. Metacaracteres se definen normalmente mediante corchetes y guiones. Por ejemplo, el rango de números de 0 a 1000 sería descrito como "[0-1000]".

Por ejemplo, si desea asegurarse de que el contenido TitleString cadena "sample", podría hacerlo usando expresiones regulares. Usted acaba de generar un nuevo objeto Regex y utilizar la función de coincidencia, como sigue:

```
Dim myPattern como nuevo Regex ("muestra")
Si myPattern.IsMatch (TitleString) A continuación,
    MessageBox.Show ("El título es válido")
End If
```

Usted probablemente tendrá que añadir las importaciones System.Text.RegularExpressions como la primera línea de su código - por encima la declaración de la clase - para expresiones regulares para trabajar.

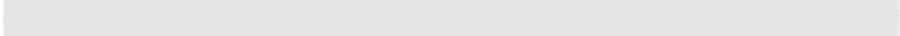
Las expresiones regulares no son sólo para la validación, tampoco. Por ejemplo, usted puede dividir una cadena con un patrón, rompiendo otra cadena en partes utilizando el caracteres y metacaracteres especificadas en el modelo.

También puede reemplazar caracteres con un patrón. El método Replace de la Regex objeto cuenta la cadena, "Hey, reemplace nada que coincide con este pagolondrina de mar con este nuevo texto. "Esto es increíblemente poderoso, como estoy seguro de que puede imaginar. Cuando se trabaja con la manipulación de datos, se le pregunta constantemente a

s con una X" es muy común. La
el siguiente código realmente funciona:

```
myPattern.Replace (newString, "[0-9]", "X")
```

169



Me pasé una sección sobre un tema que podría - y hace - llenar un libro entero. Si usted está en la manipulación de cadenas, expresiones regulares será una herramienta poderosa para usted. Para obtener más información, busque "expresiones regulares" en línea.

Cosas que no son Strings - Números y fechas

A pesar de que pasan la mayor parte de su tiempo en Visual Basic con palabras y cadenas, a veces es necesario trabajar con otros tipos. Los números y las fechas tienen una gran colocar, y son bastante difíciles de trabajar.

Los números enteros y reales y imaginarios, ¡ay!

Números están cubiertos por toda una serie de tipos. Para el propósito de este debate, que cubren sólo dos - los números enteros y decimales. Números enteros utilizar el tipo Integer. Decimales utilizar el tipo Double.

Los números son realmente bastante simple - la cosa más importante para recordar es que a diferencia cadenas, no se refieren a ellos utilizando comillas. Al establecer una variable igual a un número entero o una doble, sólo se refiere directamente al número, como sigue:

```
Dim miEntero como Integer = 65  
Dim MiDouble como Double = 6,555
```

Cualquier tipo de número puede ser manipulado con símbolos de operador, como se muestra en Tabla 9-2.

Tabla 9-2	Operadores
Operador	Descripción
+	Adición
-	Sustracción
*	Multiplicación
/	División
%	Módulo (el resto de una división)

Capítulo 9: Interpretación Strings and Things

Para sumar dos números con un operador, se genera el código que se parece a la código utilizado para concatenar la entrada de dos cajas de texto, como se describe en la

anterior sección "Cambio de los tipos con CType." Es sólo mi myAnswer = Número + MiDouble. Números de cualquier tipo usualmente se pueden añadir juntos, siempre que la variable de la respuesta es de un tipo que puede manejarlo. En este caso, myAnswer tiene que ser un tipo Double o los números después de la coma decimal punto en MiDouble se pierda.

Al igual que todas las grandes herramientas que se integran en cadenas, números tienen algunos

métodos integrados que ayuden a asegurarse de que ellos son los esperados. Para ejemplo, el tipo Integer tiene un método MaxValue y MinValue, por lo que puede asegurarse de que la operación no se sobrecargue la variable.

Las variables son posiciones de memoria y tienen una cantidad específica de espacio asignado.

Los enteros sólo puede ser entre -2147483647 y 2147483647. Esta suena como una gran gama de números, pero usted se sorprenderá de lo fácil que es puede abrumar a ese rango.

Dobles tener algunos otros métodos que representan valores silvestres matemáticas. Usted puede

prueba de los valores que representan PositiveInfinity, NegativeInfinity y Epsilon - ese número maravilloso acerca a cero, pero no alcanza cero que el estudiante de primer año años profesor de matemáticas mantienen hablando. Debido a que todos de estos números imaginarios son posibles resultados de ecuación matemáticas, es genial ser capaz de probar por ellos.

Utilización de fechas y matemáticas fecha

Las fechas son una historia completamente diferente. No sólo toda una categoría

separada de

aplicaciones utilizan fechas - en comparación con fuertes aplicaciones numéricas - pero también tiene una mentalidad diferente a utilizar los tipos DateTime de Visual Básica.

Permítanme comenzar con un hecho simple - si usted necesita una fecha o una tiempo o ambos, Visual Basic esencialmente almacena el montón en un solo tipo. El tipo que se utiliza con más frecuencia es DateTime, como se puede esperar, que posee tanto en una fecha y una hora.

En la Parte II, se muestra que la fecha de matemáticas por la construcción del programa Calculadora de la fecha en que

busca la diferencia entre dos fechas, utilizando un método integrado en la Fecha

wer para manipular
tarde el contenido de la variable de cadena.

El tipo DateTime tiene un azul millón de métodos integrados. La siguiente lista
describe sólo algunas de las más poderosas:

171

Añadir: Puede añadir cualquier tipo de palmo a una fecha en un tipo DateTime.
Para
ejemplo, Horas, Minutos, y los días son algunos de los tramos disponibles
en el método Add.



Componente: Las propiedades de los componentes le permiten obtener sólo una parte de un Tipo DateTime. Por ejemplo, la propiedad Mes obtiene sólo el mes de una fecha.

Conversión: Los métodos de conversión le ayudarán a cambiar la fecha que otro formato común. Por ejemplo, el método ToUniversalTime convierte la fecha en el objeto DateTime UTC Hora universal - útil para las aplicaciones internacionales.

Además, algunos métodos y propiedades no se encuentran en ninguna categoría, en cambio, hacen algo específico. Por ejemplo, la IsDaylightSavingsTime determina si la fecha depende del horario de verano en los Estados Unidos Estados.

Para tener una mejor idea de cómo funciona el tipo DateTime, echa un vistazo a la capturas en la Parte II. Yo lo uso extensamente mientras que el diseño y la construcción de la Fecha Calculadora de aplicación.

Usted puede encontrar una gran cantidad de fechas más que esto, sin embargo. Por ejemplo, para obtener el fecha actual en una variable DateTime, puede utilizar la función Ahora, como en el siguiendo la línea de código. Te da la fecha actual hasta el nanosegundo.

```
MiDate Dim como DateTime = Ahora
```

También puede obtener las representaciones de cadena de fechas y horas. Por ejemplo, dicen que se necesita el nombre de un mes que descubriste utilizando uno de los Categorías constitutivas de los métodos de tipo DateTime. La función MonthName le da una cadena que contiene el nombre real del mes:

```
Dim miMes As Integer = 4
Nombre As String Dim
Nombre = MonthName (miMes)
MessageBox.Show (Nombre)
```

Usted recibe un cuadro de mensaje que dice "Abril" - una herramienta muy útil para la interfaz de usuario creación, ya que los usuarios no quieren ver un número, quieren ver una palabra!

Una última cosa sobre fechas - proveedores de formato permiten mostrar las fechas en cualquier maravillosa forma que desee. El DateTimeFormatProvider es una herramienta global que le permite dar formato a las fechas de una manera flexible, universal. El más manera común de utilizar el proveedor de formato es el método ToString. La el siguiente código devuelve "Sábado, 07 de agosto 1971 12:00:00 AM":

```
MiDate As DateTime = "08/07/1971"
MessageBox.Show (myDate.ToString ("F"))
```

Algunos de los proveedores de otros formatos para fechas son las que se muestran en la Tabla 9-3. Tenga en cuenta que el caso del valor utilizado en el método ToString es importante. Más puede ser descubierto mediante la búsqueda de "método DateTime.Parse" en MSDN Library del sitio Web.

Tabla 9-3	Proveedores de formato DateTime
Formato de Proveedores	Ejemplo de salida
D	07/08/1971
D	Sábado, 07 de agosto 1971
G	07/08/1971 12:00a.m.
G	07/08/1971 12:00:00 AM
S	1971/08-07T00: 00:00
Y	08 1971

Tipos de Cambio con Parse y TryParse

Analizar es un término usado por los arquitectos de sistemas cuando necesitan algo de un formato a otro, pero no se sabe muy bien cómo. En Visual Basic, los métodos Parse y TryParse darle una forma de obtener un valor en una nueva formato mientras se controla exactamente cómo se hace.

Para utilizar el método Parse, es necesario entender algo complejo sobre Visual Basic - tipos son objetos, también. Al igual que una variable puede declararse como Tipo DateTime, el tipo DateTime es en sí mismo un objeto de tipo Type. Que tipos de medios tienen sus propios métodos, propiedades y eventos.



Tenga en cuenta que cada objeto tiene un método ToString (). El método ToString () no siempre es lo que parece. Aparentemente, está diseñado para que pueda ver el objeto como una cadena, pero no siempre es obvio lo que debe ser mostrada al pedir una versión de la cadena de un objeto (por ejemplo, el método ToString () de un Objeto Graphics, por ejemplo). No dependen del método ToString () - uso Parse, ConvertTo o CStr lugar, y permitir que el sistema produzca una error si se confunde. Mejor eso que datos incorrectos.

Uno de los problemas más comunes está tomando una cadena a partir de la propiedad de texto de un cuadro de texto y convertirlo en un tipo utilizable, como una fecha. Cuando alguien entra "07/08/1971" en un cuadro de texto, es sólo una cadena, no un tipo DateTime. Las cadenas son útiles, pero no se puede agregar un número de días a una fecha introducida como una cadena porque, de acuerdo con Visual Basic, no es una cita!

Para hacer, por ejemplo, una cadena en una fecha, se utiliza el método Parse de la Tipo DateTime. Lógicamente, se siguen una serie de pasos:

- 1. Obtenga una fecha como una cadena,** MiCadena, a partir de una entrada de base de datos o del usuario.
- 2. Declare una nueva** Tipo DateTime llamado miDate para manejar la nueva fecha.

- 3. Utilice el** DateTime.Parse método para hacer que una variable DateTime nuevo de la cadena, de la siguiente manera:

```
Dim MyString As String = "08/07/1971"  
MiDate Dim como nuevo objeto DateTime  
miDate DateTime.Parse = (MiCadena)
```

El método TryParse es muy similar, pero es más útil si se no está seguro de que el valor MyString es una cita. Debido TryParse no hace devolver un valor, sino que acepta un valor como una subrutina, no lo hará un error si el valor MyString no es susceptible de ser analizada. En su lugar, se devolverá un valor nulo. El código siguiente muestra el código anterior sería parecen utilizar TryParse:

```
Dim MyString As String = "08/07/1971"  
MiDate Dim como nuevo objeto DateTime  
DateTime.TryParse (MyString, miDate) fv
```

Capítulo 10

Toma de decisiones en el Código

En este capítulo

Diagramas de flujo del programa

Directivo de flujo con IF-THEN declaraciones

Elegir con Select-Case declaraciones

Manejo de excepciones con try-catch declaraciones

T

hey no llaman al código en las aplicaciones de la lógica de negocio para nada. Muchos aplicaciones que se escriben en Visual Basic implican la lógica, y de gran parte lógica consiste en la toma de decisiones. De hecho, la toma de decisiones representa la proceso más importante en los negocios. No podrá continuar con la producción de aplicaciones de negocio sin entender las complejidades de la replicación la toma de decisiones humanas proceso utilizando Visual Basic 2008 código.

En este capítulo, le doy un procedimiento de diseño a seguir cuando se describe una de procesos de negocio para sus aplicaciones. Este proceso, que es un derivado de diagramas de flujo básicos, lo ayuda a todos los diseños de la toma de decisiones - no sólo código de programación.

A continuación te muestro cómo trabajar con las tres herramientas de toma de decisiones en Visual Basic - solo proceso, de opción múltiple, y las excepciones - que se utilizan en el código de Visual Basic en el If-Then-Else, Select-Case, y-Prueba Catch construcciones, respectivamente. Usted ve cómo estas tres construcciones pueden ser utilizada para asegurar que sus aplicaciones de negocios más estrechamente replicar el humano de toma de decisiones que están tratando de reemplazar.

El diseño de la lógica de negocios

Tengo un cliente que describe toda la lógica de procesos de negocio como los si-entonces-goto *diagramas*. Teniendo en cuenta el número de grados que este cliente tiene en su haber, siempre asumió que él realmente sabía lo que estaba hablando. Y tenía razón!

Cuando me presenté a este caballero a cuatro páginas, flujo de procesos sofisticados (la diagramas que presento en este capítulo), eligió el único defecto en mi lógica en unos 15 minutos. (¿No sería yo el amor a todos mis clientes a pensar de esa manera?)

Utilizo este ejemplo para ilustrar que esbozar la lógica de negocio es la única situación más difícil que un programador trata sobre una base diaria. La empresa Ness lógica sirve como base para el código mecánico entre el usuario interfaz y los datos en una aplicación. Este código determina cómo el usuario considera que la información que él o ella está después, y cómo esa información se manejadas cuando se guardan.

Antes de profundizar en cada ejemplo de código en las siguientes secciones, se analizan una proceso de diseño de la lógica de negocio utilizando un diagrama de flujo. Yo repasar los conceptos básicos de diseño de la aplicación en el capítulo 3, pero el problema del diseño de la lógica es una específica situación que no todas las aplicaciones de negocios encontrará.

La razón de usar un diagrama de flujo para describir la lógica de negocio es sencillo. Modelado del proceso de decisión \Leftrightarrow sistema de dirección mediante un diagrama de flujo es exactamente lo que tendrá que hacer cuando se modela la lógica del programa. Incluso con un gran sistema, le beneficia como el programador para modelar circuitos complejos y decisiones utilizando diagramas de flujo.

Representando lógica con Componentes del Diagrama de flujo

Un diagrama de flujo es una representación "pictórica de una forma ordenada paso a paso de la solución

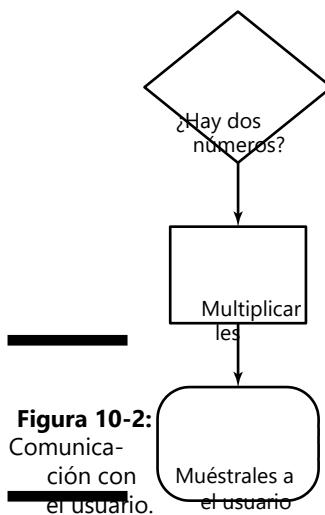
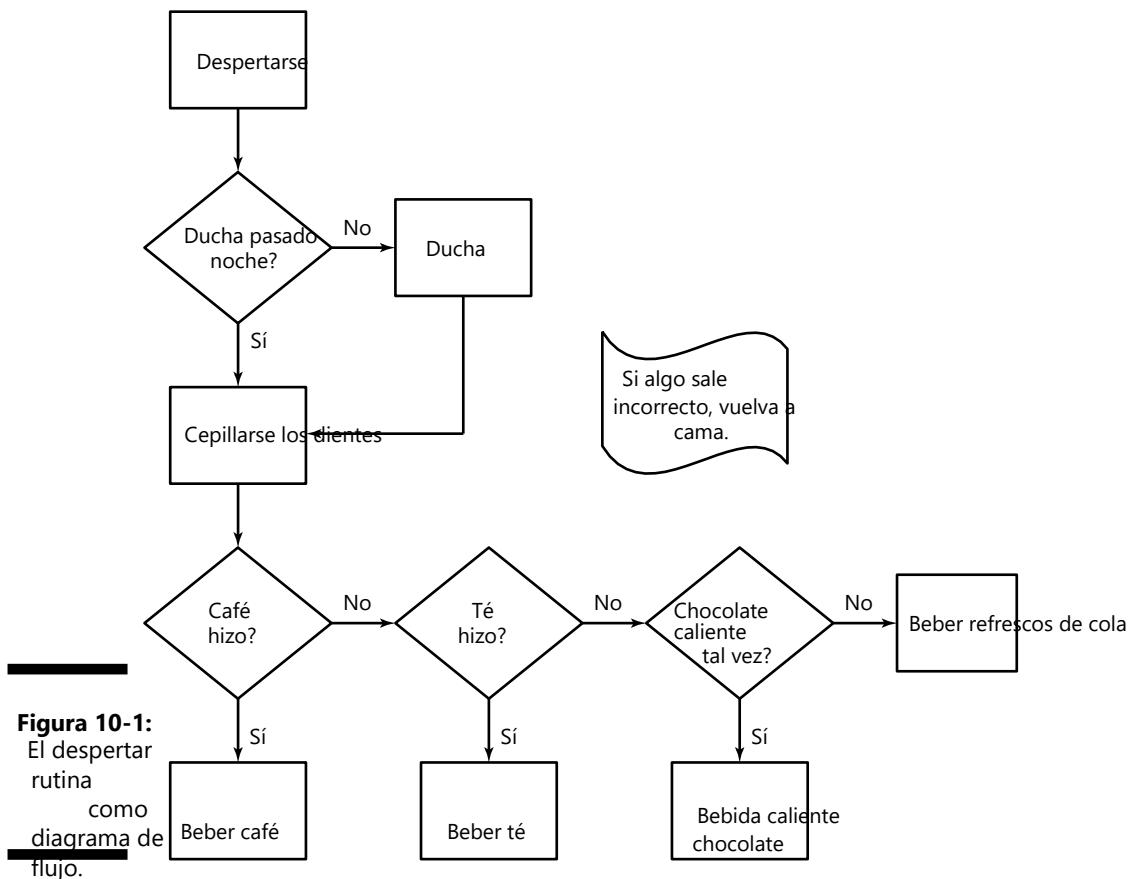
a un problema, "de acuerdo a Indiana State University. No podría estar más de acuerdo. La diagrama de flujo es simplemente líneas que conectan tres estructuras de representación de comunicación, los procesos y las decisiones. Diagrama de flujo magia - es decir, el negocio o lógica de la aplicación - es representado por cómo se combinan estos componentes (También conocidos como nodos).

Por ejemplo, un diagrama de flujo global que describe una rutina de la mañana podría parecerse a la Figura 10-1. Este diagrama de flujo particular utiliza la proceso y la decisión componentes, representados por rectángulos (como despertador) y diamantes (como la ducha anoche?), respectivamente. Obviamente, esta mañana routine es algo que no se reproduciría en el código, sino porque tiene espero que realiza un proceso similar en las últimas 24 horas, se hace un gran ejemplo!

La comunicación con el usuario

De los tres componentes de un diagrama de flujo de aplicación, se comunican con el usuario se convierte en la parte del programa visible para el mundo exterior. Al usuario, esta comunicación puede ser percibida como un mensaje escrito a la pantalla, o una copia impresa. El flujo del programa para la comunicación de usuario está representado por

una caja con esquinas redondeadas, como en el diagrama de Entrada / Salida se muestra en la Figura 10-2.



Su programa con éxito se comunica con el usuario cuando el siguiente sucede:

- El resultado es tangible.
- El programa produce una impresión, incluso si es sólo imprime en la pantalla.

El usuario podría esperar retroalimentación del programa en este punto.

Definición del proceso

Un componente de proceso de un programa de diagrama de flujo representa un bloque de código que

maneja una sola interacción con una entidad. Por ejemplo, la adquisición de entrada desde el usuario y la actualización de la base de datos son procesos que pueden ser representados en una diagrama de flujo. Los componentes del proceso (como los procesos de consumo de alcohol de la mañana-

ING rutina representada en la Figura 10-1) están representadas por rectángulos, como se muestra En la Figura 10-3.

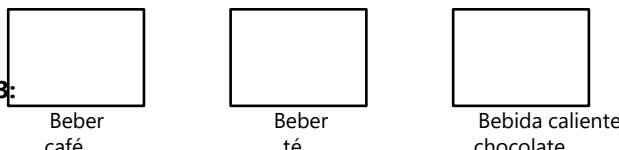


Figura 10-3:
Mañana rutina
procesos.

Los siguientes tres características identificar un componente de proceso:

- El nodo no tiene salida.
- Representa una regla de negocio.
- Se describe una función que normalmente se realiza manualmente.

Tomar una decisión

El núcleo del diagrama de flujo es el componente de decisión, que se ha asociado ramas que permiten el flujo de la gráfica para cambiar de dirección. La ramificación (es decir, después de una sucursal en el diagrama de flujo) es que la magia que aporta flexibilidad y sustancia a la lógica del programa. Sin decisiones que tomar, el diagrama de flujo es sólo una lista de cosas que hacer (procesos) y otras cosas para mostrar (o pedir) personas (Comunicación).

El nodo de decisión es físicamente muy simple, es un diamante en el diagrama que solicita una respuesta afirmativa o negativa. La figura 10-4 muestra el diamante de decisión y sus ramas. Una rama llega con la entrada, y vaya dos ramas out - uno para sí y otro para el no.

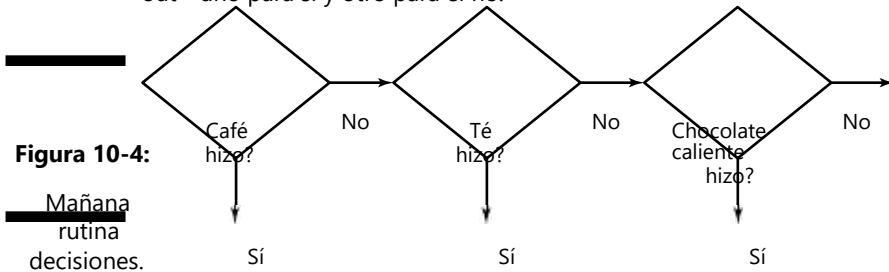


Figura 10-4:

Mañana
rutina
decisiones.

Un componente de decisión

Tiene una entrada y dos salidas.

Está formulado como una pregunta (como "respuestas" a los concursantes en Jeopardy!).

Requiere una hazaña de la lógica para pasar el nodo.

Usted puede pensar en el componente de decisión sobre la gestión de la entrada a la normal procesos de negocio. Supongamos que usted tiene un montón de empleados que realizan un conjunto de procesos día a día. Si este grupo de repente tiene que lidiar con una sola decisión inusual, una pregunta de opción múltiple, o una excepción, puede llamar a los administradores a tomar una decisión. Tal situación constituiría una decisión nodo en un diagrama de flujo de proceso humano.

Un proceso informático tienen las mismas situaciones de toma de decisiones, que puede dividir en tres categorías:

Proceso individual: Un simple "Si, pues, que, de otra seguir como siempre" una especie de decisión. Un ejemplo de un proceso de decisión de un solo estaría conduciendo en la carretera: "Si el coche delante de mí se detiene, entonces debe golpear mis frenos; otra cosa que seguir adelante".

Opción múltiple: Un proceso que tiene un montón de opciones. "Si ella lo quiere azul, y luego comprar pintura azul, y si ella quiere que verde, y luego comprar pintura verde, y si ella quiere de rojo, y luego comprar pintura roja, beber cerveza más".

Excepción: Un tipo especial de proceso único. Esta es una decisión cuando no quería tomar una decisión. "Sigue como siempre. Si se rompe, llame a la gerente".

La implementación de estos procesos en Visual Basic

Para describir los procesos en el código, lo que necesita saber cuáles son los procesos parecer en un diagrama. Correctamente diseñado, el diagrama te dice lo que se describe en el código.

Proceso individual

El mismo proceso es bastante simple - una decisión única, aislada dentro de un flujo, es por lo general un único proceso. En algunos casos, en realidad está viendo una excepción. En gran medida, sin embargo, si usted está buscando en una imagen similar a la Figura 10-5, se es un proceso único.

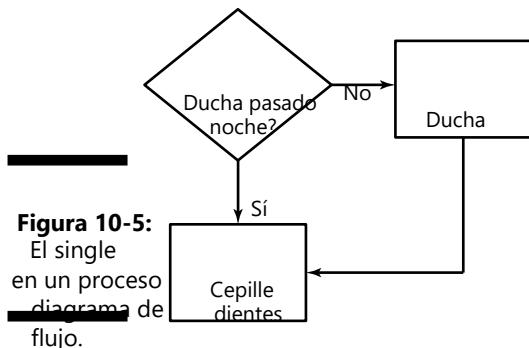


Figura 10-5:
El single
en un proceso
diagrama de
flujo.

Decisión de una sola procesos ejecutar un bloque de código si un enunciado es verdadero. En Visual Basic, es necesario proporcionar el punto del programa con una decisión *Declaración booleana* - Es decir, algo que puede ser evaluada como verdadera o falso - para decidir si el código ha de ser ejecutado. Por ejemplo, compruebe a cabo estos pasos para una rápida decisión de un solo proyecto:

- 1. Abra Visual Studio y cree un nuevo proyecto de Visual Basic 2008 Windows Aplicación del proyecto.**
- 2. Arrastre un Control TextBox y un control Button desde el cuadro de herramientas al formulario.**
- 3. Haga doble clic en el Botón de control para que Visual Studio genere su OnClick controlador de eventos.**

4. En la vista Código para la Manejador de evento OnClick, agregue el código que da

su programa una sola decisión que tomar.

En mi ejemplo, agregue el siguiente código para mostrar un cuadro de mensaje si el palabra correcta se escribe en el cuadro de texto:

```
Si TextBox1.Text = "bañado" Entonces  
    MessageBox.Show ("cepillarse los dientes!")  
End If
```

Tenga en cuenta que este código se muestra una instrucción If seguido de un End If estado

ambiente. Una declaración Si requiere una instrucción End If sólo cuando se tiene múltiples sentencias a ejecutar. Una declaración simple puede ser puesto en una línea, de este modo:

```
Si TextBox1.Text = "bañado" Entonces MessageBox.Show ("Cepillarse los dientes!")
```

Pero, en general, todas las declaraciones de flujo de control en Visual Basic tiene un principio y un

la línea final. Además, la línea de arranque (la condición Si es este ejemplo) puede convertirse

más complejo. Si cualquiera de las dos posibles respuestas pueden causar el cuadro de mensaje

para mostrar, se puede generar una declaración booleana mediante la conexión exactamente dos

declaraciones con un operador condicional.

Por ejemplo, si quieras apoyar a las duchas o baños, el código puede ser escrito diez del siguiente modo para imitar el Inglés en este requisito:

```
Si TextBox1.Text = "bañado" OR = TextBox1.Text "Tomó Bath" Entonces  
    MessageBox.Show ("cepillarse los dientes!")  
End If
```

Puede enlazar dos sentencias condicionales If. Utilice dos ligado Si los estados cuando dos resultados posibles para la decisión puede causar dos diferentes líneas de código que se ejecutará de manera exclusiva. Puede enlazar dos condicional Si declaraciones con una instrucción Else, que funciona igual que el Inglés "Si esto, entonces que, de lo contrario el otro".

La instrucción Else es para el "No" en la caja de derivación decisión en su flujo gráfico. Una instrucción Else se muestra en el siguiente código:

```
Si TextBox1.Text = "bañado" Entonces  
    MessageBox.Show ("cepillarse los dientes!")  
Más  
    MessageBox.Show ("Lluvia").  
End If
```

También puede vincular varias declaraciones If-Then-Else para manejar un múltiple-elección del proceso. La sentencia Elseif puede ayudar con eso, como sigue:

```

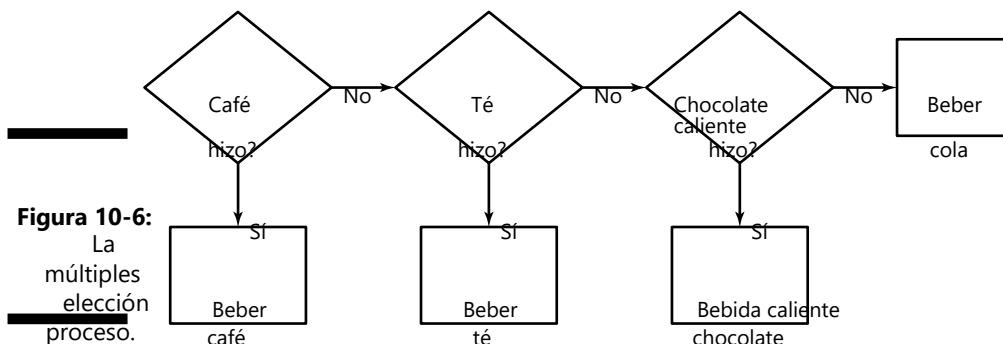
Si = TextBox1.Text "bañado anoche" Entonces
    MessageBox.Show ("Cepillarse los dientes!")
Elseif TextBox1.Text = "bañado hace dos noches" Entonces
    MessageBox.Show ("Ducha de nuevo!")
Más
    MessageBox.Show ("Shower!")
End If

```

Este ejemplo es sorprendentemente como la siguiente en una serie de enunciados condicionales que están disponibles en Visual Basic - demostrando una vez por todo lo que pueda realizar la misma tarea en más de una forma en Visual Basic. Para muchos de opción múltiple ambientes, la mejor opción en el código es el Select-Case comunicado.

Opción múltiple

Procesos de elección múltiple son igualmente obvio al código si usted es muy honesto en el diagrama. El hecho es que pocos diseñadores son lo suficientemente honesto como para escribir un diagrama como el que se muestra en la Figura 10-6. Esta estructura por niveles, sin embargo, es la firma indiscutible de un proceso de elección múltiple.



Efectivamente, este diagrama muestra varios diagramas de proceso de un solo en una fila. Visual Basic 2008 le proporciona una estructura que se encarga de este tipo de situación. La buen diseño de software se aprovecha de la mayor cantidad de la lengua en cuestión como sea posible.

Mientras que una instrucción If-Then-Elseif evalúa una serie de diferentes respuestas, la instrucción Select-Case evalúa la misma variable en contra de un número de respuestas posibles.

Por ejemplo, la instrucción If-Then-ElseIf se muestra en la sección anterior compara el mismo cuadro de texto a dos valores diferentes. Se podría tener la misma comparar fácilmente dos cuadros de texto diferentes a dos valores diferentes.

La instrucción Select-Case está diseñado para comparar la misma variable a varios valores posibles. El código siguiente muestra cómo escribir un Estado-Select-Case ción que lleva a cabo la misma cosa como una instrucción If-Then-ElseIf.

```
Seleccione TextBox1.Text
Caso "Café Made"
    MessageBox.Show ("Beba"      Café! ")
Caso "Tea Made"
    MessageBox.Show ("Beba"      Tea! ")
Caso "Hot Chocolate Made"
    MessageBox.Show ("Beba"      Chocolate caliente! ")
Case Else
    MessageBox.Show ("Beba"      Cola! ")
End Select
```



El código en una instrucción Select-Case puede hacer exactamente lo mismo que la código en una instrucción If-Then-ElseIf (como en el código anterior y el código que se muestra en la sección anterior), pero es la instrucción Select-Case mucho más fácil de leer, y que en realidad se ejecuta un poco más rápido.

También puede poner una lista separada por comas de los valores en cada caso para dar casi una rejilla de dos dimensiones sentir al proceso. Uso de una lista delimitada por comas que camino es bastante resbaladiza, y una elegante manera de codificar el proceso de elección múltiple como diseñado en el código anterior.

La instrucción Select-Case no es el único proceso que es similar a la Si-El entonces ElseIf comunicado. Otro proceso es el proceso de excepción, donde usted se encuentra escribiendo un flujo que dice: "Trate de ir a menos que algo va mal; entonces haz esto "En VB 2008, esto se llama una declaración try-catch.

Excepción

La excepción es un caso especial del modelo de un solo proceso. Cuando haya escrito ING un flujo, y de repente se necesita para poner en un proceso que dice: "Si esto no es tan era de esperar, lo que "se trata con una excepción. Figura 10-7 muestra lo que una excepción se ve como en un diagrama de flujo.

Figura 10-7:
¿Cómo
maneja una
excepción.





Una excepción es diferente de un error. Un error es un error en una de las capas de una aplicación - por ejemplo, una fila de base de datos o una mala red se desconexión, o en el peor de los casos, un error es una pieza mala de código. Una excepción es un error esperado. Es algo que usted calcula que podría ocurrir, aunque no lo quiera, y usted tiene un pedazo de la lógica para tratar con él. Para más información sobre excepciones, consulte el Capítulo 12.

La instrucción Try-Catch es diferente de las otras dos estructuras de decisión. Se asume lo siguiente:

- Usted tiene una lista de procesos que desea realizar.
- Usted desea redirigir el flujo del proceso si se produce un error.
- Usted tiene procesos a seguir si se encuentra un error.

La instrucción Try-Catch es lo mejor para ese proceso que es difícil de diagrama - "Haced esto a menos que algo salga mal, y luego hacer eso."

La lista de cosas que hacer pasa por debajo de la instrucción Try, y espera que cada error va con una instrucción Catch. Cada instrucción Catch incluye la proceso que se va a ejecutar si esa declaración se alcance. Si no hay errores ocurre, el código en las declaraciones de capturas se ignora.

El siguiente código muestra un ejemplo de una instrucción Try-Catch basa en Figura 10-1, sólo para mantener las cosas consistentes. La función MorningRoutine estaría constituida por el código de decisión en el capítulo hasta ahora.

```

Tratar de
    MorningRoutine ()
Catch somethingWentWrong como excepción
    MessageBox.Show ("Algo salió mal - volver a
        cama ")
End Try

```



Cuando se utiliza un bloque try-catch para detectar una excepción lógica (como los datos están en error) en lugar de un error técnico (como la base de datos está roto), utilice un ApplicationException en lugar de una excepción a la cláusula Catch.

El código siguiente es un uso mucho más común de una instrucción Try-Catch. Observe cómo utilizar la propiedad Message de la excepción que se llamó a avisar al usuario de lo que salió mal:

```

SmallNumber como Integer Dim
LargeNumber como Integer Dim
Tratar de
    smallNumber = 4534
    largeNumber = 7654
    largeNumber = smallNumber * LargeNumber
Catch badNumber como InvalidCastException
    MessageBox.Show ("El número era malo -" & badNumber.Message)

```

```
Catch somethingElse como excepción  
    MessageBox.Show ("Otra cosa que salió mal -" & somethingElse.Message)  
End Try
```

Las excepciones que constituyen las declaraciones de capturas son una parte emocionante de la la .NET y un poco más allá del alcance de este capítulo –, o incluso este libro. Cuando ocurre algo inesperado, tal como un error, el fotograma trabajo produce una excepción, y esa excepción es lo que tienes que coger con el código anterior.



Por último debo mencionar. Después de una instrucción Try-Catch, a veces es necesario hacer las cosas independientemente de si ocurrió un error. Si ese es el caso, poner el código en un comunicado el último. Por ejemplo, si se trata de un dato-base, se debe cerrar la conexión con la base de datos sobre el éxito o el fracaso. Esa instrucción aparecería en un bloque Finalmente, después de su última captura bloquear.

Usted puede encontrar cientos de tipos de excepción, y se puede escribir el suyo propio. Para más en la gestión de excepciones, véanse los capítulos 8 y 12; para futuras investigaciones, búsqueda de "Exception Management" en la biblioteca MSDN.

Capítulo 11

Conseguir Loopy

En este capítulo

Confirmación de los conceptos de la cuenta en el código

Excavando en sus opciones de bucle

Taquí hay dos tipos de estructuras de control en Visual Basic - Decisiones y bucles. En el capítulo 10, cubro las decisiones, que son todos acerca de la ramificación fluir sobre la base de algunos de entrada al programa.

Looping, el segundo tipo de estructura de control, está sobre la repetición de la misma comando (o una serie de comandos) hasta que una determinada condición se cumple. Negocios aplicaciones, sobre todo, tienen que repetir lógica de programa para una cierta cantidad de tiempo o un cierto número de iteraciones - resultando en la programación equivalente de "enjabonar, enjuagar, repetir." Por supuesto, usted no quisiera lavar y enjuagar el cabello todos los días, por lo que esta frase debería decir "enjabonar, enjuagar y repetir una vez." Una secuencia de eventos como este es análogo a una estructura de bucle - conocido como el bucle For-Next - en un programa de Visual Basic.

Los programadores suelen utilizar bucles y las decisiones juntos. Visual Basic proporciona un constructo para que, también. El bucle do-while es un ejemplo de esto - la probamos a través de un comando o serie de comandos (el hacer parte), siempre y como una cierta condición (la parte While) es verdadera. Tan pronto como la condición se convierte en falso, se detiene el bucle.

En este capítulo, se pasa al diseño y código de los cuatro tipos de bucles en Visual Basic 2008:

- For-Next
- For-Each
- Hágalo hasta que
- While-End

Todas estas estructuras de control de bucle tienen algunas características comunes. Repiten un bloque de código, y tomar una decisión acerca de cuándo parar. Las diferencias entre ellos son completamente basado en el recuento de la lógica.

Tratar con Zero

La toma de decisiones sobre qué circuito a utilizar es difícil. Usando el bucle incorrecto puede cambia significativamente el procesamiento del programa, y realmente puede estropear la experiencia del usuario. Así las secciones siguientes le ayudarán a determinar qué bucle estructura de usar y dónde empezar a contar sus lazos.

A partir de las cero

Todo lo importante acerca de los bucles se pueden desglosar mirando a contar. Alguna vez trató de contar el número de horas entre el final de la comida y el fin de la jornada laboral? Tú cuentas ", la una, dos, tres, cuatro, cinco! Cinco horas! "

Sin embargo, su jornada laboral por la tarde no es de cinco horas de duración, es cuatro horas de duración. A contar con el número adecuado de horas, tendrá que saltarse el incremento de tiempo en primer lugar.

Lo que hay que contar son los espacios entre horas, de la siguiente manera: "Uno para dos, de dos a tres, tres a cuatro, cuatro a cinco. Cuatro horas".

Bucle a través de código en un programa es similar: Si usted está contando los espacios entre los números, se salta el primer número. Es decir, siempre se corre el bucle por primera vez, y a continuación, comprobar la condición al final.

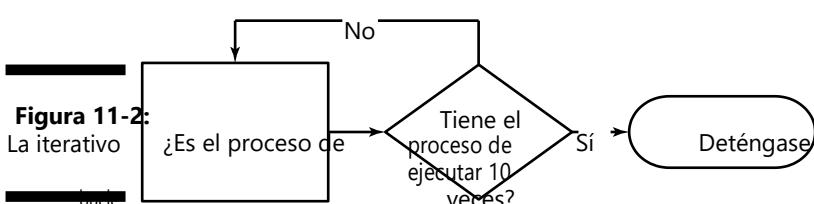
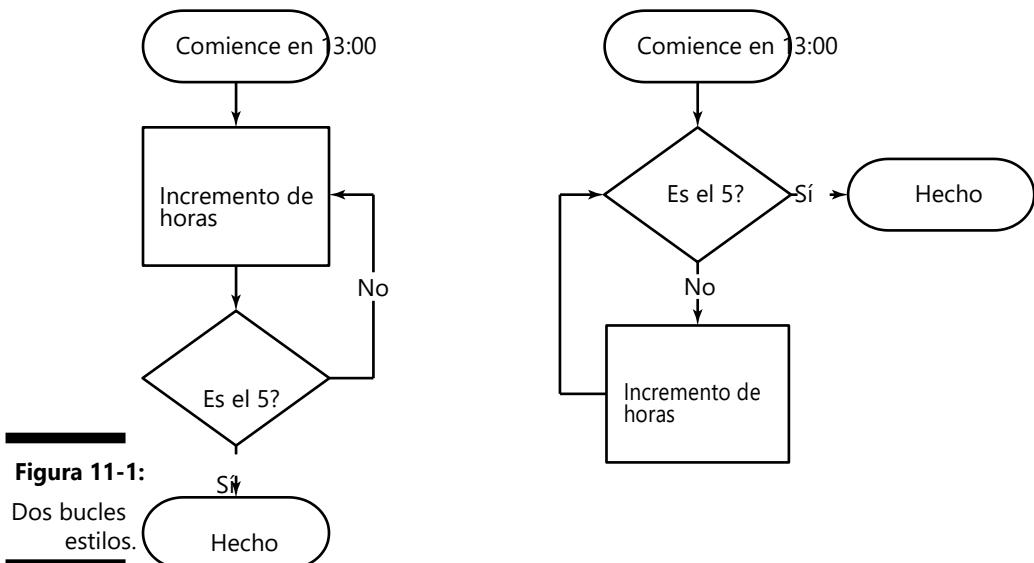
La figura 11-1 muestra dos diagramas. La de la derecha es el camino equivocado para contar las horas después del almuerzo. En este diagrama se empieza a contar a las 1:00 PM. El diagrama de la izquierda empieza a contar a las 2:00 PM, que es el mismo que contar el espacios entre los números.

Comparando los lazos específicos y los bucles indefinidos

Otra diferencia entre los diferentes tipos de bucles es si el bucle se repite para un número específico de veces o repite para un número indefinido de veces. Un bucle específico se reproduce un número definido de veces; un bucle indefinido toma una decisión, ya sea al principio o el final del bucle, para detener.

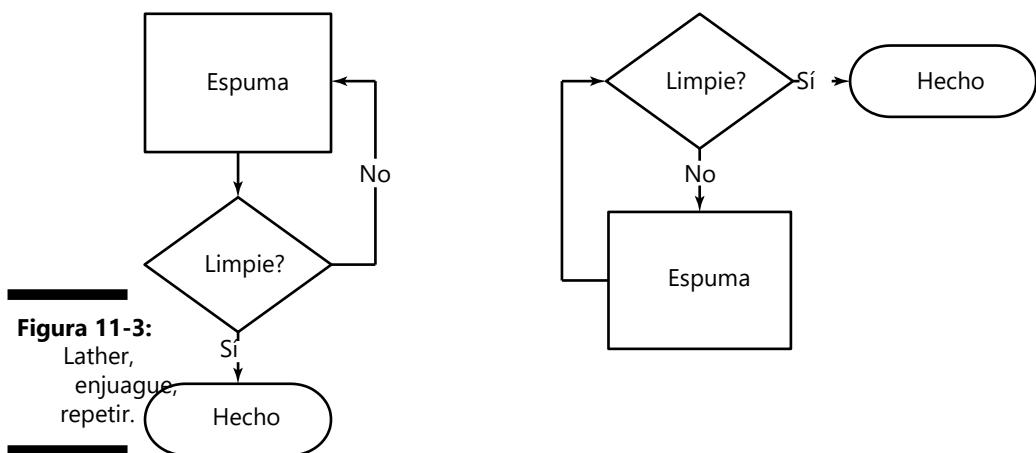
Un bucle que se ejecuta un número específico de veces que es como "diez latigazos con un húmedo fideos. "Este concepto se aplica con un For-Next o For-Each bucle.

Efectivamente, se está traduciendo el ejemplo: "Porque cada uno en una colección de diez, azotar con un video mojado. "En este ejemplo se muestra en la Figura 11-2. Que Parece un poco por la borda, pero tiene mucho sentido en el contexto.



Un bucle que tiene una cantidad indefinida es como "hacer espuma, enjuague, repita". ¿Cuántos veces que se repite? Bueno, esa es la broma en el ejemplo - no lo sé. Usted asumir que significa repetir hasta que quede limpio, pero no lo sé.

Suponiendo que el objetivo es repetir hasta que quede limpio, has vuelto a la cuenta problema. ¿Se supone que es sucio, y luego comenzar con la espuma? ¿O es que revise primero antes de enjabonar la primera vez? Figura 11-3 muestra cómo Figura 11-1 podría ser cambiado para mostrar las dos opciones para el champú ejemplo.



Escribir Loops con For-Next

El bucle For-Next es un bucle iterativo. Al principio del bucle, se define una cantidad, y el bucle se repite muchas veces. Cuando el bucle se realiza repitiendo el código después del bucle se ejecuta. El siguiente es un ejemplo de un sencillo bucle For-Next que muestra el número en un cuadro de mensaje:

```

Contador Dim como Integer
Para contador = 1 a 5
    MessageBox.Show ("El número es" & contador)
Contador Siguiente

```

Usted puede hacer mucho con un bucle For-Next. Cuando vaya a manipular algunas cosa que un conjunto de números de veces, cuando usted necesita para asegurarse de que algo ocu plumas de un número determinado de veces, o cuando tenga que recuperar un número determinado de artículos de un grupo, este es su lazo.



For-Next loops pueden hacer unas pocas truquitos, también. Por ejemplo, imagine que usted hay que hacer algo para cada dos líneas de una colección. Usted podría probar en el mitad del bucle para ver si el contador es par, o usted podría utilizar la declaración Paso lo mismo que yo en el siguiente código:

```

Contador Dim como Integer
Para contador = 2 a 10 Paso 2
    MessageBox.Show ("El número es" & contador)
Contador Siguiente

```

Este bloque de código muestra 2, 4, 6, 8, y 10 en el cuadro de mensaje. Bastante resbaladiza. Usted puede usar esto para acceder a todos los demás elementos de una colección mediante Contador en el índice de la colección.

Asimismo, la declaración Step puede ser usado para contar hacia atrás, como muestro en la siguiente código. Cuando lo haga, asegúrese de que el primer número es más grande que el segundo, o el bucle no se ejecuta!

```
Contador Dim como Integer
Para contador = 5 a 1 Step -1
    MessageBox.Show ("El número es" & contador)
Contador Siguiente
```

Como estoy seguro que lo habéis adivinado, este código le da 5, 4, 3, 2 y 1. Supongo que esto añade otro uso - cuenta atrás para el lanzamiento de un cohete o algo así.

Una última cosa acerca de bucles For-Next: A veces es necesario salir de un bucle antes de que el bucle está hecho. Esta situación suele ocurrir cuando el inicio y valores finales son variables, y usted no sabe ir en exactamente lo que son.

Por ejemplo, digamos que usted no quiere ir por debajo de 0 en el ejemplo siguiente.
Usted

Puede utilizar una instrucción If-Then y una instrucción Exit-Para detener el bucle.

```
Oscur Contador como Integer
0 valorIncial como Integer = 5
Oscur EndValue como Integer = -1
0 Contador = valorIncial a EndValue Step -1
Oscur
0
Para
    Si el contador < 1 Then Exit para
        MessageBox.Show ("El número es" & contador)
Contador Siguiente
```

Este código se detendrá cuando el contador se pone a 0 y se mueve a la línea después de la Declaración contra Siguiente.

Uso de la For-cada lista con colecciones

Una colección es una construcción especial del Marco. NET que contiene una número de objetos y se accede con un índice que se refiere al artículo en la colección. A pesar de una colección no sólo se encuentra en la carpeta Windows mundo, los detalles son algo único en el Marco. NET.

No tengo espacio suficiente en este capítulo para entrar en los detalles de colección ciones. Se les ve en los ejemplos del libro, por lo general como una propiedad plural de un objeto. Por ejemplo, todos los controles de un formulario (por ejemplo, botones y texto cajas) se celebran en una colección llamada ControlCollection.



La colección se implementa utilizando una interfaz llamada `IEnumerable`. Esta biblioteca de código especifica que el código usando `IEnumerable` debe ser capaz repetirse utilizando el For-cada anuncio. Si necesita saber si puede utilizar For-Each para recorrer en iteración una colección, vea la documentación de la oponerse a averiguar si implementa `IEnumerable`. Por ejemplo, el Control colección - que es iterativo - se parece a esto en la documentación:

```
Control.ControlCollection _ Public Class
    Hereda ArrangedElementCollection _
    Implementa IList, ICollection, IEnumerable,
    ICloneable
```

Para recorrer la colección de control utilizando el For-Cada lista, es necesario crear una aplicación de ejemplo poco siguiendo estos pasos:

- 1. Abra Visual Studio y comenzar un nuevo proyecto de aplicación para Windows.**
- 2. Arrastre cuatro cuadros de texto al formulario.**
- 3. Arrastre un botón al formulario.**
- 4. Haga doble clic en el botón para encender el OnClick controlador de eventos.**

La solicitud debe ser similar a la Figura 11-4.

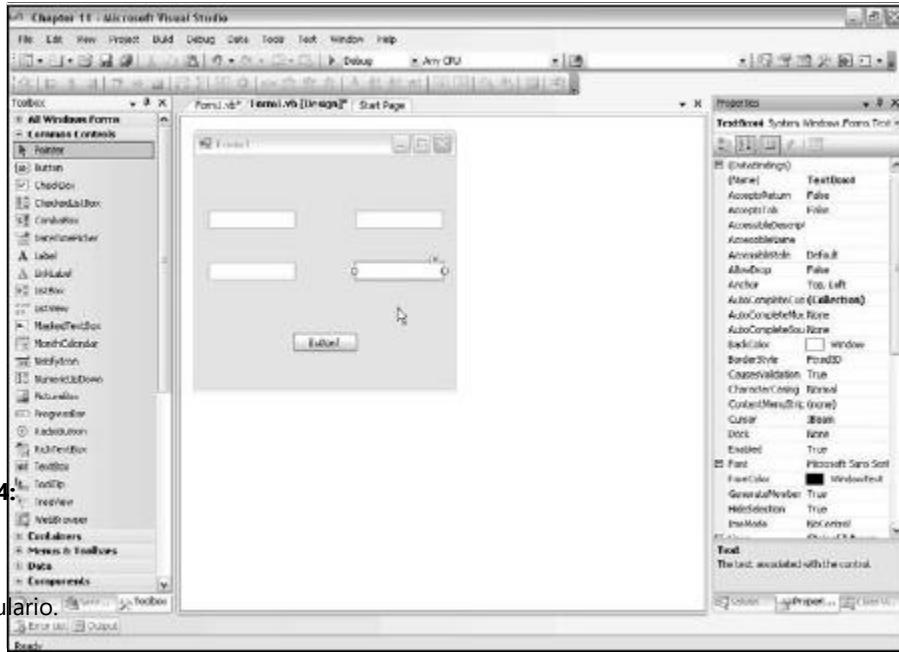


Figura 11-4.

El por-
Cada
ejemplo
formulario.

Capítulo 11: Cómo Loopy

5. Agregue el código siguiente al método de creación:

```
Private Sub Button1_Click (ByVal como System.Object remitente,_
    ByVal e As System.EventArgs) Handles Button1.Click
    For Each myControl Como System.Windows.Forms.Control En Me.Controls
        Si myControl.GetType Es TextBox1.GetType Entonces
            myControl.Text = myControl.Location.ToString
        End If
    Próximo
End Sub
```

6. Haga clic en el botón para iniciar el código en el método.

Tenga en cuenta que el texto se cambia en cada uno de los cuadros de texto.

A pesar de que no le dijo al bucle cuántas veces a correr, era todavía un con-número de veces tensas, debido a que la colección contiene un número discreto de controles de formulario. Observe también cómo me aisló más el número de controles actúa sobre la comprobación del tipo en el bucle con una instrucción If-Then.

¿Por qué tengo que hacer eso, usted pregunta? Debido a que ninguna colección de cuadros de texto existe, sólo una colección de controles de formulario. Si sólo desea los controles TextBox, necesario filtrar utilizando un si-entonces la forma en que lo hizo.

Escribiendo Bucles indefinidos con Do-Loop

Bucles indefinidos son lazos que no se cuentan, pero que continuará indefinidamente hasta algo sucede. Bucles indefinidos son un poco más difícil de escribir, porque dos las cosas pueden suceder a encerrar a su aplicación - olvidarse de pasar a la siguiente elemento en un grupo, o establecer los criterios de cuándo detener el bucle para un acondicionado condición que nunca será verdad.

Usted puede utilizar el Do-Loop de cuatro maneras:

Loop mientras que algo es verdad, comprueba antes de empezar.

Loop mientras que algo es verdad, nos registramos después de la primera iteración.

Loop hasta que algo se hace realidad, nos registramos antes de empezar.

Loop hasta que algo se hace realidad, nos registramos después de la primera iteración.

El bucle Do-es, con mucho, el bucle más flexible, ya que se encarga de construir

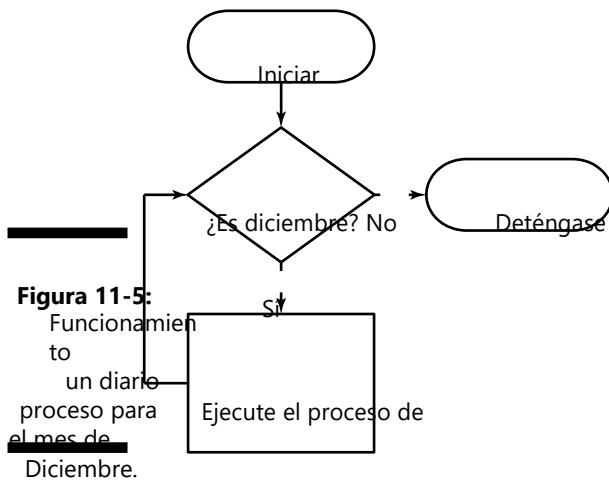
rma manual, un Do-Loop puede sustituir a un bucle For-Next.

En general, sin embargo, los programadores utilizar el Do-Loop, como último recurso, ya que es tan propenso a errores. A Do-Loop es tan amplia que es muy fácil crear una situación en la que el bucle terminaría prematuramente nunca, correr o ir en fin.

El Do-Loop es un concepto muy útil, sin embargo, vale la pena aprender a usar también. Discutir cada una de las cuatro opciones, primero en un diagrama de flujo y luego en el código.

Do-While, comprobado en el arranque

El siguiente ejemplo de la utilización de la do-while describe ejecuta una rutina para todos los días en un mes, cuando no está seguro de si se encuentra en el mes que desee. Por ejemplo, digamos que usted tiene un proceso que le quería correr una vez para cada día del mes, pero sólo en el mes de diciembre. Trazado por fuera, esto se vería como en la Figura 11-5.



La ejecución de este se vería como el siguiente código:

```

MiDate Dim como DateTime = Date.Now ()
Do While myDate.Month = 12
    RunTheProcess ()
    miDate = myDate.AddDays (1)
Bucle

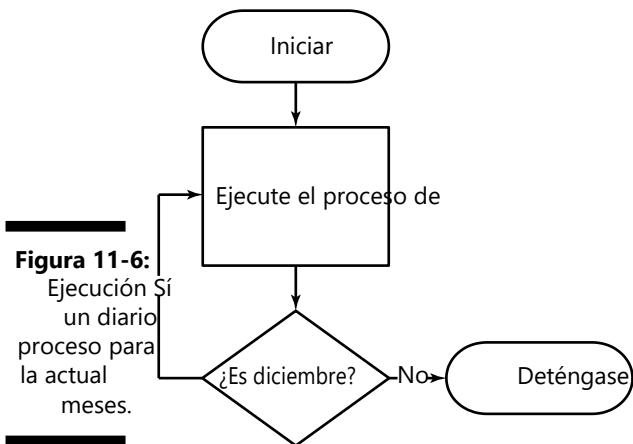
```



No te olvides de incrementar su iterador bucle!

Do-While, comprobar al final

Tomando la perspectiva opuesta de este ejemplo diario-proceso supone una proceso que usted sabe que siempre desea ejecutar una vez, sin importar el meses. Por ejemplo, digamos que el proceso se ejecuta por lo menos una vez por cada mes, en todos los meses, y lo que desea que el bucle se detendrá cuando el contador de días no cae en diciembre, como se muestra en la Figura 11-6.



La construcción de la Figura 11-6 en el código se vería como la siguiente:

```

MiDate Dim como DateTime = Date.Now ()
Hacer
    RunTheProcess ()
    miDate = myDate.AddDays (1)
Loop While myDate.Month = 12
    
```

Do-Until, comprobado en el arranque

Siguiendo con el tema de la fecha, supongamos que tiene un proceso que desea ejecutar todos los días de la semana hasta el sábado. Si comienza un sábado, sin embargo, no debe correr, ¿no? Así que la lógica dice así: "Ejecutar la rutina de una vez por todos los días hasta el sábado."

La ejecución de este se vería como el siguiente código:

```

MiDate Dim como DateTime = Date.Now ()
Hacer hasta myDate.DateOfWeek = DayOfWeek.Saturday
    RunTheProcess ()
    miDate = myDate.AddDays (1)
Bucle
    
```

Do-Until, comprobar al final

Puede ejecutar la rutina al menos una vez cada hora, y hágalo funcionar hasta el día muestra como el sábado. Esto significa que si comienza un sábado, se correría hasta el próximo sábado, es decir, ocho días en total. Tal vez eso es lo que quiere - pero asegúrese primero!

Al ejecutar esta rutina se vería como el siguiente código:

```
MiDate Dim como DateTime = Date.Now ()  
Hacer  
    RunTheProcess ()  
    miDate = myDate.AddDays (1)  
Repetir la reproducción hasta myDate.DayOfWeek = DayOfWeek.Saturday
```

Comprobación en el principio con bien

En el código en el "Do-While, comprobado en el arranque" y "Do-While, comprobar al final "secciones, anteriormente en este capítulo, se puede ver el bien del Estado ción en el Do-Loop. ¿Por qué, entonces, hay un bucle While-End?

La diferencia está en la salida mientras comunicado. No se puede salir de un Do-Loop. La While-End loop puede salir como un bucle For-Next posible mediante el uso de la salida While. Aparte de eso, como se puede ver en el siguiente código, el diferencias entre un bucle do-while y un bucle While-End son mínimos:

```
MiDate Dim como DateTime = Date.Now ()  
Mientras myDate.Month = 12  
    RunTheProcess ()  
    Si MyDate.DayOfWeek = Salir DayOfWeek.Saturday bien  
        miDate = myDate.AddDays (1)  
End While
```

Capítulo 12

Reutilización de Código

En este capítulo

- Escritura de funciones y subrutinas para su reutilización
- La reutilización de código en comparación con hacer que sea sencillo
- Tomando ventaja de los programas existentes
- Hablando con DOS

muchas funcionalidades están flotando por ahí. Antiguos VB 6 programas de DOS, aplicaciones, archivos de otras personas DLL y archivos incluso que el último módulo de programática izquierda son todas las posibles fuentes de código de reutilizar para la nueva VB 2008 aplicación.

En el capítulo 6, te muestran cómo escribir programas de la biblioteca, llamada Dynamic Link Bibliotecas o archivos DLL. Y aunque te recomiendo un archivo DLL como el tipo de proyecto para convertir para cuando tenga que escribir código reutilizable, está lejos de ser la única manera de reutilizar su código de funcionalidad. También puede crear reutilización escribiendo funcional código dentro de sus propios programas. Es decir, usted puede agregar fácilmente ayudante subrutinas y funciones a sus programas sin tener que utilizar una DLL proyecto.

Un programa que está perdiendo protagonismo, pero aún está por ahí es el antiguo Sistema operativo, o DOS. Muchas personas que empezaron temprano con personal computadoras todavía juran por la interfaz de línea de comandos. Ciertamente, algunos las cosas son mucho más fáciles de utilizar la línea de comandos, y .NET le permite tocar los viejos comandos de DOS desde tu código VB 2008.

En este capítulo, se tratará cómo obtener los valores de entrada y salida de las funciones y subrutinas. También averiguar dónde se encuentran los ganchos para los programas externos y cuando a utilizar DOS.

La reutilización de código para crear software

Cada pieza de código en un programa en Visual Basic debe estar dentro de un procedimiento como una función o una subrutina. Los controladores de eventos que se presentan

a través de los programas de este libro (véanse los capítulos 4 y 5, por ejemplo)



L

son todas las funciones. Sin embargo, el control de la operación del programa mediante el uso de caso handlers no es la única manera de construir software. Mover código repetido - es decir, las mismas líneas de código que se muestran en el controlador de eventos con más de un - en procedimientos auxiliares es la forma aceptada para construir aplicaciones. La razón para esto es simple: la depuración y el mantenimiento de una aplicación que se creó de esta manera son más rápidos y menos propenso a errores introducidos. Específicamente, haciendo cambios en las líneas de código que aparecen en el mismo lugar (el procedimiento reutilizable) es mucho más eficaz que realizar cambios en las mismas líneas de código que aparecen en varios lugares (los controladores de eventos individuales).

Creación de funciones o subrutinas reutilizables no es más difícil que la codificación manejadores de eventos específicos, pero sí requiere un enfoque un poco diferente. Usted construir código reutilizable mediante la construcción de una función independiente o en una subrutina clase de archivo y luego llamarlo desde el evento que lo requiera. Tenga presente lo siguiente artículos en cuenta a la hora de salir a hacer que el código realmente reutilizable:

Conozca la diferencia entre encapsular código y crear

código reutilizable. *Encapsulado* código es un código común que se pone en un archivo o lugar lejos de su programa principal - en un archivo de clase, para ejemplo - por conveniencia y separación lógica. Código encapsulado mantiene una pieza específica de lógica juntas por el uso de un conjunto específico de código funcional. Código reutilizable es el código que usted encapsular y utilizar otra vez sin cambios. Una biblioteca de datos en toda la empresa es un ejemplo de código reutilizable que se puede utilizar en varios programas.

Entender que el código reutilizable bien contiene un elemento de abstracción

ción. Esto es, el código reutilizable no debe depender de los nombres específicos de los controles que se llaman o, de hecho, sobre los controles en absoluto. Siendo medios abstractos que el código reutilizable opera en los parámetros pasados y devueltos valores. Necesita aceptar tipos de bases (como enteros y cadenas) o conocido construcciones (como bases de datos y colecciones) y regresar el mismo.

Hacer la parte reutilizable función de un archivo separado, en lugar de colocar que dentro del proyecto que lo llama. Esta es la parte de encapsulación. Usted Desea encapsular el código de modo que puede fácilmente pasar de una proyecto a otro.

La construcción de funciones con la reutilización en mente

Para mostrar cómo se ve el código reutilizable en una aplicación real, empiezo con el código de Calculadora de la fecha de la construcción en el capítulo 4 y alterarlo para cumplir con el requisito de reutilización. La Calculadora de la fecha contiene el código del controlador de eventos - `LateDate_ValueChanged` y `EarlyDate_ValueChanged` - es que la código real que hace los cálculos de fecha y define el valor devuelto. El seguimiento de código muestra dos controladores de eventos:

Capítulo 12: Reutilización del código

```
Private Sub EarlyDate_ValueChanged (ByVal sender As System.Object e, ByVal como
System.EventArgs) Handles EarlyDate.ValueChanged
    NumberOfDays.Text = CStr (System.Math.Abs (CInt ((EarlyDate.Value -
LateDate.Value). Días)))
End Sub

Private Sub LateDate_ValueChanged (ByVal sender As System.Object e, ByVal como
System.EventArgs) Handles LateDate.ValueChanged
    NumberOfDays.Text = CStr (System.Math.Abs (CInt ((EarlyDate.Value -
LateDate.Value). Días)))
End Sub
```

Observe que la línea de código mismo aparece como la línea media en los dos submarinos privados
(Que es la línea en negrita del código en las dos funciones). Repitiendo esta línea de código no es

bueno, porque viola el concepto de reutilizar la funcionalidad. Si necesita cambiar la funcionalidad de esta línea de código más tarde, usted tiene que cambiar el misma línea en ambos lugares. En su lugar, usted puede poner el código funcional repetida en un lugar en su programa, le dan un nombre público, y llamarlo desde el evento manejador. Para ello, puede crear un procedimiento - al igual que en la biblioteca de clases en Capítulo 6 - y colocarlo justo en el código del formulario.

Ahora, cuando usted quiere hacer los cálculos la fecha, sólo se puede llamar a esta función, y que hace el trabajo. El código siguiente muestra el funcionamiento FindDateDiff:

```
Public Sub FindDateDiff ()
    NumberofDays.Text = CStr (System.Math.Abs (CInt ((EarlyDate.Value -
LateDate.Value). Días)))
End Sub
```

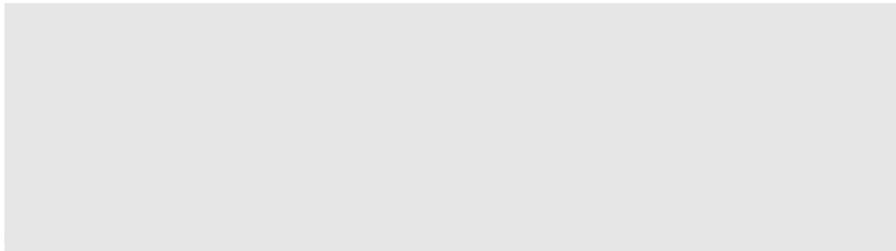
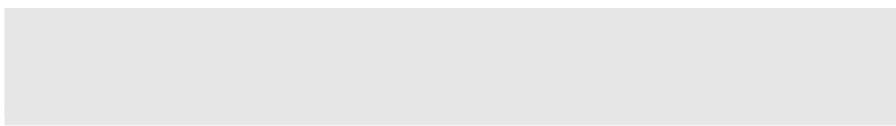
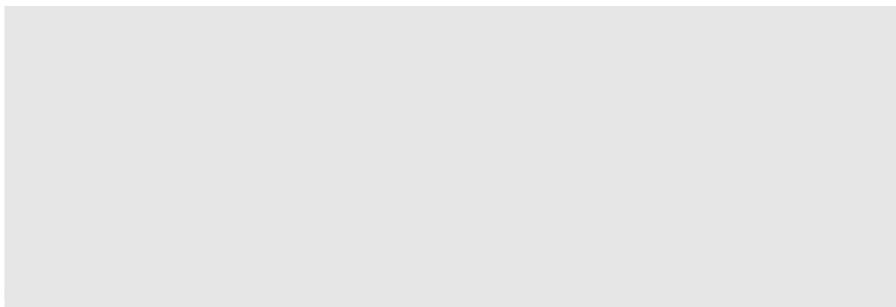
Y el código siguiente muestra los controladores de eventos parece ahora que se llama a la función FindDateDiff en lugar de repetir la línea de código:

```
Private Sub EarlyDate_ValueChanged (ByVal sender As System.Object e, ByVal como
System.EventArgs) Handles EarlyDate.ValueChanged
    FindDateDiff ()
End Sub

Private Sub LateDate_ValueChanged (ByVal sender As System.Object e, ByVal como
System.EventArgs) Handles LateDate.ValueChanged
    FindDateDiff ()
End Sub
```

Al colocar esta subrutina pública nueva en su forma Calculadora de la fecha, es un ejemplo de código encapsulado (código común que aparece en uno subrutina dentro de una forma por conveniencia). Si usted se muda fuera de la rutina forma, se puede utilizar en otra forma sólo si el nombre del otro formulario controla exactamente la misma medida que los nombrados en esta solicitud. Esa situación

í misma como abstracto (es decir,
independiente de los controles) como sea posible es mejor.





Aún con sus limitaciones, les muestro un ejemplo de código encapsulado porque a veces encapsulación es más importante que la reutilización. No todos los función necesita ser abstraído a la enésima potencia. De hecho, la mayoría de su código no estará en la forma de funciones abstractas. Entender cuando es necesario reutilizar el código y cuando no importa depende de su modelo de negocio - es decir, el propósito general de su programa. Por ejemplo, si una función particular es sólo código de mantenimiento dentro de la propia aplicación - conversión de un formato de datos local a otro, por ejemplo - que probablemente no tendrá que ser reutilizado fuera del programa.

Para hacer una función apropiada para su reutilización por lo que es independiente de la específica nombres de la rutina de llamada, es necesario pasar parámetros a la función y aceptar los valores devueltos de la misma. En la aplicación de Calculadora de la fecha ejemplo, a lograr esta abstracción por:

- | Pasando la fecha de inicio
- | Pasando la fecha de finalización
- | Al aceptar el intervalo devuelto (en días) como un entero

En efecto, se establece el valor del cuadro de texto NumberofDays.Text igual a el valor de retorno de la función FindDateDiff pasándolo a los valores de los dos selectores de fecha. Muestro esta función verdaderamente reutilizable en el listado 12-1.

Listado 12-1: Abstracción Usando una función en lugar de una subrutina

```
Private Sub EarlyDate_ValueChanged (ByVal sender As System.Object e, ByVal como
    System.EventArgs) Handles EarlyDate.ValueChanged
    NumberofDays.Text = CStr (FindDateDiff (EarlyDate.Value, LateDate.Value))
End Sub

Private Sub LateDate_ValueChanged (ByVal sender As System.Object e, ByVal como
    System.EventArgs) Handles LateDate.ValueChanged
    NumberofDays.Text = CStr (FindDateDiff (EarlyDate.Value, LateDate.Value))
End Sub

Public Function FindDateDiff (startdate ByVal como Date, ByVal como enddate Fecha) Como
    Entero
    Diferencia Dim As Integer
    = diferencia (startdate - enddate). Días '# 1
    Volver diferencia
End Function
```



Observe el elemento interesante en la línea numerada como 1. Esta línea se ventaja de la propiedad de la fecha de cálculo. Esto es, porque (iniciar fecha - enddate) es una resta de dos fechas, devuelve un objeto de la DateSpan tipo de datos. Un objeto de este tipo incluye una propiedad de los Días, que la línea # 1 se conecta a. (Para saber más acerca de los tipos y la forma en que trabajar, buscar en el capítulo 9.)

Reutilización Extender con archivos de clase

Para la reutilización real, que necesita para hacer la parte de la función de un archivo independiente denominado un archivo de clase. Estos archivos de clases son una especie de bibliotecas de proyecto dentro y son útiles cuando el código puede reutilizarse dentro de un proyecto, pero el código no es probable que se utilicen fuera de un proyecto. Un archivo de clase en un proyecto es exactamente el mismo tipo de criatura como un archivo de clase dentro de una DLL.

Siga estos pasos para configurar y utilizar un archivo de clase:

1. Con el programa abierto en Visual Studio, cree una carpeta nueva en su proyecto.

Le recomiendo que almacenar las clases en una carpeta separada - con el nombre inteligente de clases - en el interior de su proyecto. De esta manera, la clase archivos son fáciles de encontrar.

2. Haga clic derecho en el Explorador de soluciones y seleccione Agregar clase para hacer una nuevo archivo de clase. (Vea la Figura 12-1.)

3. Escriba un nombre para el archivo de clase cuando se le solicite, hacer que el nombre de algunos-

lo apropiado para el tipo de código que se espera.

Puse el nombre de mi DateMath.vb nuevo archivo.

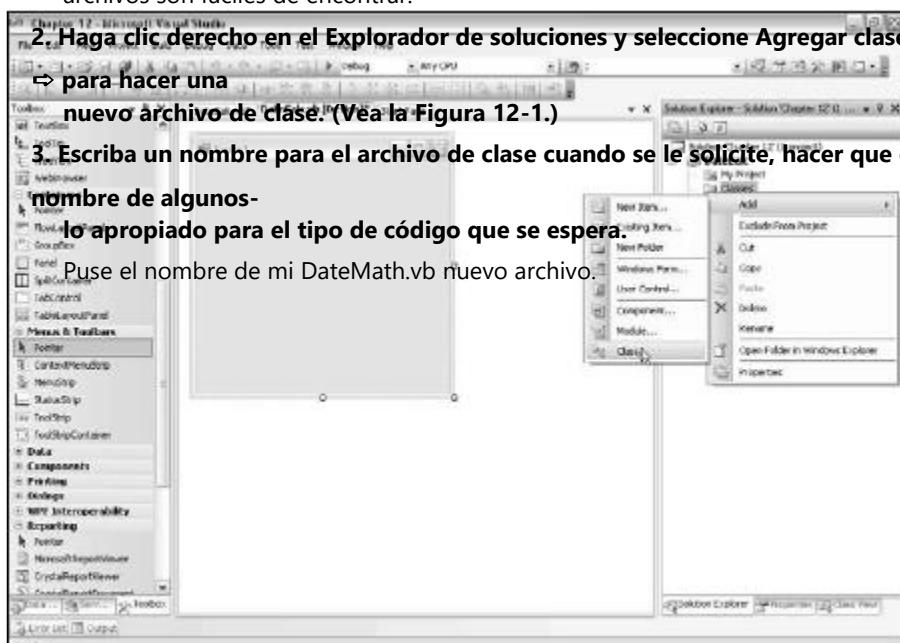


Figura 12-1:
Adición de
un
clase de
archivos para
su proyecto.

**4. Copia y pega el código reutilizable (su función) de su aplicación
ción del código del formulario en el archivo de clase.**

Mi código de acabado archivo de clase sería la del listado 12-2.

Listado 12-2: El código de función en un archivo de clases

```
Public Class DateMath
    Public Function FindDateDiff (startdate ByVal como Date, ByVal como enddate Fecha) Como
        Entero
        Diferencia Dim As Integer
        Diferencia = (startdate - enddate).días
        Volver diferencia
    End Function
End Class
```

Al crear funciones reutilizables y ponerlos en archivos de clase, que, naturalmente, mover el código relacionado en un archivo separado física en su proyecto. En la Fecha ejemplo Calculadora, teniendo la función de fecha y cálculo de la fecha Código del formulario Calc significa que Visual Studio no puede encontrar el FindDate Dif. método. Se puede decir tanto porque una línea ondulada azul aparece bajo el nombre del método. La posibilidad de llamar al código de fecha de cálculo es efectivamente perdida porque el método ya no se encuentra en la misma clase que llama.

Visual Basic le ofrece dos maneras de solucionar el problema. Usted obtiene ningún beneficio real por la elección de una forma de referencia sobre el otro, sino que es simplemente preferencia personal de cómo usted quiere que su código para mirar. Dispone de las siguientes opciones:

Usted puede agregar una referencia en su forma a la nueva clase que ha creado para mantener sus funciones reutilizables. Este proceso es similar a añadir un referencia a una nueva biblioteca de clases, como lo describo en el Capítulo 6. Añadir la necesaria código por encima del nombre de la clase con la instrucción Imports, como muestro en el siguiente código:

```
Option Strict On
Option Explicit On
Las importaciones DateCalc2008.DateMath
Public Class CalculatorMain
```

Usted puede hacer referencia directamente a la función utilizando el nombre de clase en delante del nombre de la función cuando se llama a la función, como se ha demostrado por DateCalc2008.DateMath.FindDateDiff en el siguiente código:

```
Private Sub EarlyDate_ValueChanged (ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles EarlyDate.ValueChanged
    Dim dateCalc como nuevo DateCalc2008 ()
    NumberofDays.Text = CStr (dateCalc.DateMath.FindDateDiff (
        EarlyDate.Value, LateDate.Value))
End Sub
```

Cosas Pública y Privada

Las palabras claves públicas y privadas para archivos de clases y funciones a ser especialmente importante al crear código reutilizable.

Ha estado usando todo el tiempo, como parte de formas, pero que adquieren un nuevo significado cuando se trata de llamar a una función fuera de la clase en la que se declaró.

Una función pública puede ser visto y llamado por cualquier programa que hace referencia a la clase en la que reside. La instrucción Imports en el ejemplos en la sección cercano "Ampliación de reutilización con archivos de clase "hace referencia a la clase, y luego

usted puede utilizar todas las declaraciones públicas en ella. Una clase privada sólo puede ser visto y utilizado por los otras funciones dentro de la misma clase. En la ejemplo en la reutilización "Extender con archivos de clase "de su Sección si eran FindDateDiff Privado, sería invisible incluso si una Instrucción Imports fueron utilizados. Amigo y Protegido otras opciones, pero-con control de acceso en las asambleas (bloques compilados de código). Público y Privado son los opciones que veremos más a menudo.

Para hacer referencia directamente a un método en una clase, el método debe ser compartido, o debe Dim una nueva instancia de la clase y el método de referencia de la nuevo nombre de variable. En el caso de que usted está escribiendo un archivo de clase que es principalmente orientado a compartir, marcar los métodos como compartida que permite utilizarlos sin instanciar una instancia de la clase. Si el objeto de representada por la clase se debe crear en primer lugar, sin embargo, usted tiene que Dim un nuevo instancia de la clase antes de llamar al método. En ese caso, se utiliza la variable nombre capaz de hacer referencia al método en cuestión. Para ver un ejemplo de Clases compartidas, echa un vistazo a File y FileInfo en el nombre System.IO- espacio cubierto en el capítulo 16.

Evitar la trampa de código Complejidad

Complejidad, es decir, escribir más código que es necesario resolver un problema-dilema ming, es un problema para todas las aplicaciones, sin importar la plataforma o idioma que se construyen con. Causa problemas de complejidad, como los programas no se puede mantener, y también hace que los problemas existentes más difícil de encontrar, porque es difícil seguir la lógica representada por el código.

Como desarrollador, es necesario equilibrar la complejidad con el sentido común. Fabricación

código tan sofisticada que nunca se puede encontrar errores es un peligro, especialmente situación de las unidades organizativas. Imagine una función que llama a una función que llama a una función... usted consigue la idea. Puede perderse en el flujo de el código muy rápido.



Sin embargo, no se puede poner todo el código de procedimientos en un archivo grande tampoco. Si su método Button1.Click tiene toda la lógica en él para manejar un auto, contra el facturación del envío, usted tendrá un controlador de eventos 1000-line. Este no es el camino ir tampoco.

Se puede emplear dos buenas prácticas de programación que ayudan a evitar com-problemas de complejidad en sus procedimientos:

Cómo prevenir los problemas de lógica mediante la protección de los parámetros que se pasan a la funciones en los procedimientos.

Incorporar el manejo de errores para ayudarle con la programación procedural.

La protección de los valores de los parámetros



Cuando se usa un procedimiento, tiene dos formas de pasar los parámetros. Usted puede pasar una copia de la información en poder de la variable que se utiliza como parámetro, o puede pasar la variable real.

Determinar cómo pasar los parámetros se remonta a la programación idiomas que utilizan punteros. Por ejemplo, el lenguaje C hizo uso de punto res a información de referencia prevé que en determinadas posiciones de memoria. Usted podría pasar una copia de la información o un puntero a la información real.

Hágase esta pregunta para decidir cómo manejar el paso de parámetros: Haga está usando los datos originales a estar protegido contra el procedimiento que está llamada ción, o está esperando el llamado procedimiento para cambiar sus datos? Responder esta pregunta y luego elegir la estructura de Visual Basic que te lleva a la resultado que usted desea. Visual Basic se encarga de pasar parámetros con los siguientes palabras clave:

ByVal se refiere a términos de valor, y asegura usando esta palabra clave que una copia de datos del parámetro (su valor) se hace antes de que se pasa a la función ción. Si la función cambia el valor del parámetro, que cambió valor se desecha al final de la vida funcional del parámetro (Que se encuentra al final de la subrutina o función). En general, las funciones no cambie los valores que se pasan a ellos, pero si quieres ser absolutamente seguro, debe proteger esos valores al optar por pasar parámetros ByVal. Esta es la opción por defecto.

ByRef representa como referencia, y esta palabra clave se encarga de que una referencia a la variable original pasa a la función como parámetro.

La referencia apunta a la ubicación real del valor de la variable, por lo que f la función cambia el parámetro, la variable en el programa de acogida cambios también. Usted puede realmente deseas que la función de cambiar el ori-valores Nal - por ejemplo, una operación de facturación siempre quiera a cero una variable de honorarios acumulativo pasa a una función. En este caso, utilizando ByRef que la función pueda acceder directamente a la variable original y le ahorra una gran cantidad de código.

Usted puede notar que si no escribe en el mismo palabra clave, Visual Basic establece el parámetro por defecto para pasar ByVal para todas las funciones y subroutines. Tener ByVal como el estado por defecto es más seguro (ya que la variable de original valor se conserva) y evita los errores lógicos (por ejemplo, accidentalmente sobrescribir un valor que el código de llamada espera ser el mismo) que pueden resultantes del uso ByRef y que son difíciles de encontrar más tarde.

Gestión de errores con eficacia en un entorno abstracto

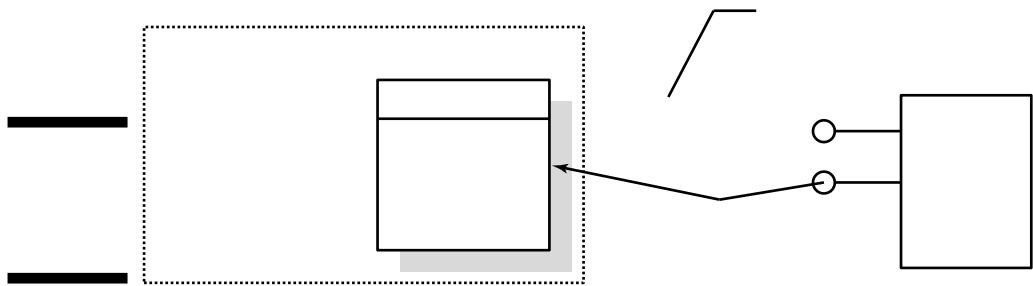
Otra consideración para limitar la complejidad de la programación procedural se refiere a errores de manipulación. Cubro el tratamiento de errores utilizando el bloque try-catch en Capítulo 10. En esta sección, me refiero a los detalles de la determinación de cómo tratar con errores arrojados en la programación procedural.

El problema con los errores en el código procesal es uno de abstracción (el mismo separación funcional que apuntas al crear una clase reutilizable). Si crear código reutilizable utilizando ficheros de clases y procedimientos, se puede perder la capacidad de comunicarse con la interfaz de usuario y en consecuencia a los usuarios cuando se produce un error. Figura 12-2 ilustra este concepto.



Entonces, ¿qué sucede cuando se produce un error en el procedimiento que se abstrae de la aplicación? Usted maneja esta situación al permitir excepciones a burbujas desde el procedimiento para la interfaz de usuario, en lugar de tratar para manejarlos en el archivo de clase en sí. Por ejemplo, en vez de manejar un error de base de datos de conversión en una función reutilizable con una instrucción Try-Catch, sólo debe permitir que el error que se arrojen a la aplicación de origen.

Abstracción, por lo tanto, tiene un impacto más allá de facilitar el intercambio de código. ¿Cuándo mueve un procedimiento que contiene código reutilizable en un entorno aislado



aparecerá un cuadro de mensaje si (y cuándo) se produce una excepción! Usted debe hacer buenas decisiones acerca de cuándo y cómo tratar las excepciones. Aquí están algunos punteros:

Deje que la mayoría de los errores que brotan en el código que llama a la procedimiento. No trate de controlar los errores que usted puede dejar que el sistema de manejo para usted. Cuando se produce un error que está cubierta por un sistema de excepción (Por ejemplo, la base de datos de estar fuera de lugar), el error hará que el procedimiento detenga la ejecución y luego pasar de nuevo a la ejecución llamando al procedimiento. Este proceso automático es suficiente para hacer frente a la mayoría de los errores.

Organice su lógica de negocios para evitar excepciones en los archivos de clase.

Los procedimientos que se diseña para ser compatible necesidad de ser mejor de lo que tiene que ser. En otras palabras, sus funciones y subrutinas reutilizables debe ser lo más infalible posible. ¿Las pequeñas cosas que reflejan una cuidadosa práctica de programación - como asegurarse de que los bucles no se sobreestiran sus límites o que sus datos no estarán sujetos al tipo confusión. Compruebe que los valores de su procedimiento acepta como parámetros son lo que realmente necesita el código de modo que usted no tiene que preocuparse de errores de versiones. El cuidado de los detalles como este hace mucho código reutilizable más útil.

Cuando un error de lógica de negocio se produce, informar al programa de llamada por la creación de un error de los tuyos. Esta práctica se llama programación *arrojando un error*, y se maneja como era de esperar por la palabra clave Throw. Por ejemplo, supongamos que el procedimiento quiere asegurarse de que el parámetro pasado llamado startdate es siempre anterior a la apropiado parámetro llamado enddate. El código siguiente muestra cómo producir un error después de comprobar los dos valores:

```
Public Shared Function FindDateDiff (ByVal startdate como la fecha, _  
ByVal Como enddate Fecha) As Integer  
    Diferencia Dim As Integer  
    Si enddate > startdate Entonces  
        Lanza Nueva ApplicationException ("Fecha de finalización no puede" + _  
            "Ser antes de la Fecha de Inicio")  
    End If  
    Diferencia = (startdate - enddate). días  
    Volver diferencia  
End Function
```

Si el startdate no sea anterior a la enddate, este código hace que el ejecución del procedimiento para detener, y el control se devuelve al invocador-programa con el ApplicationException a remolque. Si el programa se ajusta el llamar al procedimiento en el bloque try-catch (que trataremos en el capítulo 10), el programa maneja la excepción. Si no es así, Windows maneja para usted!

Encontrar otras maneras de reutilizar el código

Otras formas de escribir código reutilizable se integran en el entorno de Visual Studio ambiente. Los controles de servidor (componentes, tales como cuadros de texto y botones) que se puede utilizar en Windows y Web Forms se integran en el entorno, y usted puede crear sus propios componentes reutilizables. Usted también encontrará un sencillo concepto de control, un control de usuario en páginas ASP.NET. Y usted puede construir fácil de reutilizar plantillas de página, como las páginas maestras de ASP.NET, también.

Creación de controles personalizados

Los controles de servidor - como el TextBox y Button que muestro en muchos ejemplos - son fáciles de construido utilizando un tipo de proyecto especial en Visual Studio. Aunque la creación de controles personalizados es un poco más allá del alcance de esta libro, siga estos pasos para crear un control personalizado simple y hacerla valer- poder a sus proyectos de Windows Forms:

- 1. Abra Visual Studio y seleccione una nueva Biblioteca de Windows Forms Control proyecto.**

El editor se abre con un espacio de trabajo que se parece mucho a la normal Diseñador de Windows Forms, pero con sólo una zona gris para trabajar, lo nombre mi ejemplo de muestra de control de Windows.

- 2. Arrastre uno o más controles del cuadro de herramientas a la zona gris, y cambiar el tamaño del área para encajar alrededor de ellos.**

Me arrasta tres cuadros de texto a la zona gris en mi proyecto y organizarlos como se muestra en la Figura 12-3.

- 3. Haga clic con el defecto . Vb archivo en el Explorador de soluciones, y dar a su controlar un nombre que pueda recordar.**

Llamo a mi PhoneNumber.vb archivo.

- 4. Elija Generar ⇒ Generar solución en el menú principal para compilar el proyecto.**

- 5. Seleccione Archivo ⇒ Agregar proyecto nuevo y seleccionar una nueva aplicación para Windows**

Proyecto para agregar otro proyecto a la solución.

Llamo a mi nuevo proyecto de prueba de muestra de control.

o y elija Establecer como proyecto de puesta en marcha.

7. Haga clic en el nuevo proyecto y elija Agregar referencia.

Aparecerá el cuadro de diálogo que se muestra en la Figura 12-4 aparece.

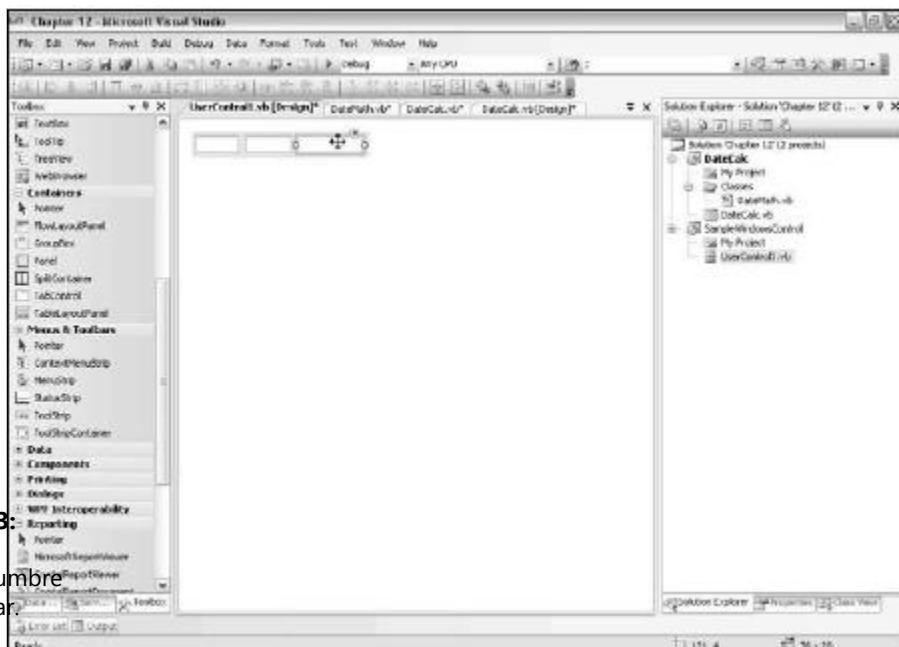


Figura 12-3:
El inicio de
una costumbre
controlar la

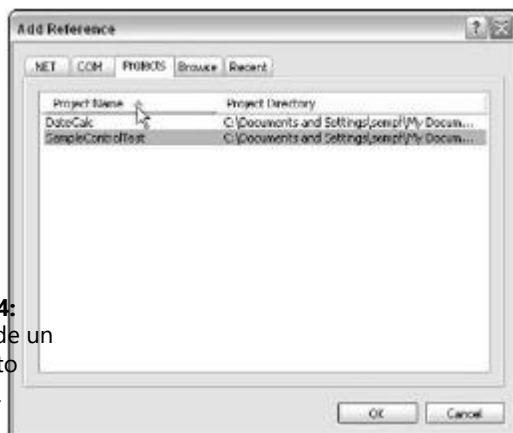


Figura 12-4:
Adición de un
proyecto
referencia.

8. Haga clic en la ficha Proyectos, seleccione el proyecto SampleControlTest y haga clic en el botón OK.

9. Expanda la caja de herramientas y tenga en cuenta la adición de su nuevo control.

En mi ejemplo, yo veo el control PhoneNumber. Debe estar en una nueva grupo por encima del grupo All Windows Forms.

10. Arrastre el nuevo control hasta el formulario predeterminado.

El nuevo control aparece en su forma tal como si se tratara de cualquier de los incorporados en los controles.

Cómo hacer controles personalizados le ayuda a compartir código es obvio. Si se define un conjunto común de controles que tienen la lógica común, se puede crear con-custom controles y escribir el código para estos controles sólo una vez. Su lógica está protegido por la compilación, y usted puede redistribuirlo los controles personalizados en su organización. En pocas palabras, lo que hace controles personalizados es una tremenda herramienta en términos de código compartido.

Agregar controles de usuario

Otra gran herramienta en el campo de código compartido es el control de usuario

ASP.NET. La *control del usuario* es una versión más simple del control personalizado y está representado por un archivo especial - un archivo ASCX - en el proyecto Web ASP.NET.

De lo contrario, un control de usuario que funciona y actúa igual que un control personalizado. La control del usuario

Encapsula otros controles y la lógica que les rodea.

Se muestra como un objeto separado dentro del proyecto.

Está construido en un espacio de diseño independiente.

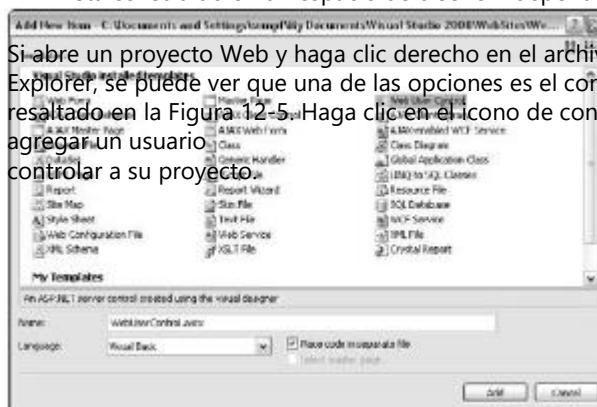


Figura 12-5:
Selección de
un
Web del
usuario
Control.

El proceso para desarrollar un control de usuario es igual que el proceso de des-
ING una página de formularios Web Forms. Arrastre los controles en la pantalla y haga doble clic para
agregar
código para el control. Para agregar el nuevo control a una página web, sólo tienes que arrastrar el
ASCX archivo a la página ASPX en la vista Diseño.

Usted debe tener en cuenta, sin embargo, que los controles de usuario ASP.NET no son tan ver-
satile como controles personalizados. Los controles de usuario no se compilan en archivos de clase,
pero
están integrados en la propia Web. Para volver a utilizar el código de control de usuario en otro
proyecto, tiene que copiar el código de un proyecto a otro. Y cuando
agregar controles de usuario a una página, que no aparezcan exactamente como se ha diseñado
ellos, pero están representados por un marcador de posición.

A pesar de sus limitaciones, controles de usuario pueden ser partes importantes de una
Web del proyecto. Muchos elementos de la página Web se repite - por ejemplo, la navegación
ción controles y encabezados, pies de página, etc. Todos estos elementos hacen fantas-
tic usuario controla.

Creación de páginas maestras

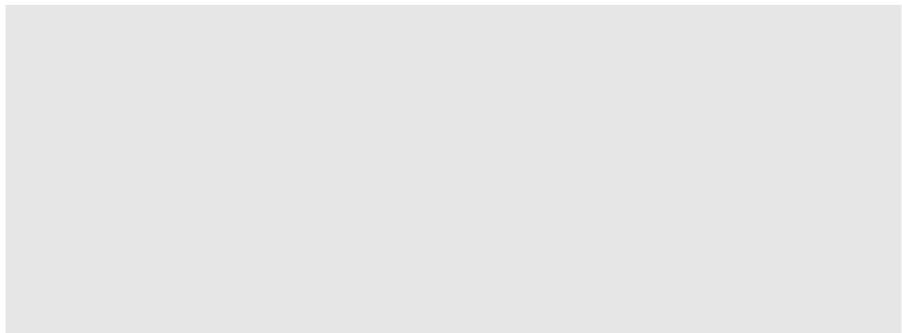
Otra de las características de ASP.NET, a lo largo de las líneas de un control de usuario, es una página
principal. La
página maestra es efectivamente una plantilla de página para los sitios web, por lo que no se
comparte código
tanto como se trata de un marco común para un proyecto.

Para agregar las páginas principales de los proyectos así como de agregar controles de usuario. Puede
hacer
haga clic en el archivo de proyecto y seleccione el ícono de página maestra para agregar uno a un
proyecto.
A partir de ese momento, se edita un documento que es como un Microsoft Word tem-
plata para páginas Web.

La página maestra utiliza un constructo llamado un marcador de posición de contenido para
determinar
el diseño y la colocación de contenidos de las páginas ASPX. El ContentPlace
Titular de la construcción permite estructurar lugar de la página el contenido
a partir de una página de contenido se coloca. El control de contenido en las páginas ASPX define
el contenido que va con cada control. Por ejemplo, el código siguiente muestra
una página principal con dos controles ContentPlaceHolder justo al lado de un
otro en una tabla:

```
<% @ Language Master = "VB" CompileWith = "MasterPage.master.vb"  
    AutoEventWireup = "false" ClassName = "MasterPage_master"%>  
  
<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
    "Http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">
```

Página <title> título </ title>
</ Head>
<body>
<form runat="server" id="Form1">



Capítulo 12: Reutilización del código

```
<table>
<tr>
<td>
<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
</Asp: ContentPlaceHolder>
</Td>
<td>
<asp:ContentPlaceHolder id="ContentPlaceHolder2" runat="server">
</Asp: ContentPlaceHolder>
</Td>
</Tr>
</Table>
</Form>
</Body>
</Html>
```

Para ejecutar esta página maestra con el contenido, la página de contenido debe tener

Un atributo de Maestría en la directiva Page.

Dos controles de contenido en la página que tengan el contenido para ser
colocado en
los ContentPlaceHolder1 y ContentPlaceHolder2 controles
en la página maestra.

Utilizar una página maestra para controlar el diseño que evita tener que recodificar
las etiquetas HTML en varias ocasiones de una página a otra. También puede realizar
esta
estructura reutilizables con controles de usuario, pero las páginas maestras son mucho
más
eficiente.

La reutilización de programas externos del Marco

Conozco dos razones para pensar en la reutilización de la funcionalidad del programa
fuera de
NET Framework.:

Es posible que los programas más antiguos - código legado escrito antes de la RED.

Marco existía - que tiene la lógica de negocio que usted aún desea utilizar.

Puede que tenga que trabajar con las partes del sistema operativo Windows que
no están disponibles en el Marco. NET.

trabajar. Mientras que usted puede volver a escribir la mayor parte de su código heredado en VB 2008, no siempre tendrá la tiempo o energía. La posibilidad de aplicar directamente el mundo del COM (el Modelo de objetos componentes arquitectura utilizado antes. NET) en Visual Basic y del Marco. NET simplificará en gran medida de su ambiente de trabajo.

211

Afortunadamente para los programadores de VB 2008, trabajar con objetos COM es ahora más fácil que nunca. Además, los desarrolladores de Visual Basic tenía un muy realista vista de la plataforma Windows. Mientras que los desarrolladores de Microsoft están muy concentrados en el Marco .NET, no todos los productos vendidos por sus proveedores de terceros es .NET listo. Además, ciertos elementos de Windows (por ejemplo, propiedades de archivos NT) todavía no tenemos .NET asociados a ellos.

Empiezo mi discusión sobre la conexión con código antiguo con la aplicación de COM porque ese tema abarca una gran cantidad de los problemas heredados de reutilización de código que tienen que resolver. Luego hable de ponerse manos a la del sistema operativo operaciones existentes. No puedes conectar a las viejas operaciones OS mucho, pero algunas situaciones - como tratar con hardware antiguo - todavía requieren que saber cómo.

Tomando como referencia la edad de componentes Modelo de objetos

El Modelo de objetos componentes (COM) es la biblioteca de código que permite la desarrollo de los archivos DLL antes. .NET existido. No hay nada malo con el COM per se, pero .NET es mucho más apropiado para el desarrollo en el actual ambiente.

Sin embargo, a veces usted tendrá que escribir el código que utiliza la lógica en COM viejo objetos. Visual Basic 2008 proporciona una función, llamada CreateObject, es decir, integradas en el lenguaje. Esta función permite a la dimensión de un objeto y luego asignarla a una clase COM existente si conoce la referencia a esa clase.

Por ejemplo, para crear un nuevo archivo de Microsoft Word, puede utilizar el siguiente código. El código define simplemente un marcador de posición objeto nuevo y lo asigna a la Word.Application clase, que es la clase COM viejo para Microsoft Word funcionalidad. La clase expone varios métodos, propiedades y eventos, pero Este código utiliza el método Save y suministra el objeto recién creado con un nombre de archivo.

```
Public Sub MakeWordFile ()  
    Myword Dim como objeto  
    myword = CType (CreateObject ("Microsoft.Office.Interop.Word.Application"),  
                    WordApplication)  
    myWord.Save ("c: \ NewFile.doc")  
End Sub
```

Antes de tener acceso a cualquiera de estas funciones, es necesario hacer referencia a el archivo DLL en su proyecto. En este caso, haga clic en la ficha COM en el complemento Referencia del cuadro de diálogo (ver Figura 12-4). Desde este cuadro de diálogo, es necesario añadir una referencia a la versión local de Word. En mi PC, es la versión 11, y el archivo DLL se llama Microsoft Word 11.0 Object Library.

Capítulo 12: Reutilización del código

Cuando se utiliza la función CreateObject de Visual Basic y un enlace a COM viejo clases, usted puede conseguir un poco de retroceso desde Visual Studio. En la vista Código, Visual Studio puede advertirle que la Resolución Consolidada Late podría causar errores. Esta advertencia aparece porque no se ha utilizado Visual Studio para crear el nuevo objeto, y eso significa que Visual Studio no puede confirmar que la Guardar el método, en realidad existe.

Llamar a métodos de objetos COM

Si usted no necesita una instancia real del objeto definido por una clase COM, Puede utilizar la función CallByName para ejecutar sólo un método como el definido por una clase. Por ejemplo, el código siguiente muestra cómo se puede llamar al método Save en el ejemplo anterior sin utilizar un objeto:

```
CallByName (myword, "Guardar", CallType.Method,  
           "C:\NewFile.doc")
```

Con otros programas con la clase Process

De lejos, la mejor manera de obtener información a otro programa desde un VB 2008 programa es la clase Process. La clase de proceso hace uso de la archivo-asignaciones de extensión de Windows para determinar qué aplicación para el lanzamiento. La gran ejemplo de este uso consiste en la herramienta de búsqueda de Google. La herramienta de búsqueda de Google Tool es una aplicación para Windows muy simple que muestra al usuario un cuadro de texto y un botón. Devuelve una URL basada en el término de búsqueda introducido en el cuadro de texto y te permite decidir qué aplicación para lanzar a ver la URL.

Para construir su propia herramienta de búsqueda de Google, siga estos pasos:

1. Abra Visual Studio y comenzar una nueva aplicación de Windows Forms proyecto.

Por sorprendente que parezca, yo nombro mi aplicación Herramienta de búsqueda de Google.

2. Haga clic derecho en el proyecto y seleccione Agregar una referencia.

Aparecerá el cuadro de diálogo Agregar referencia.

3. En la ficha. .NET, agregue una referencia a la System.Web componente.

4. Arrastre un cuadro de texto y un botón del cuadro de herramientas al

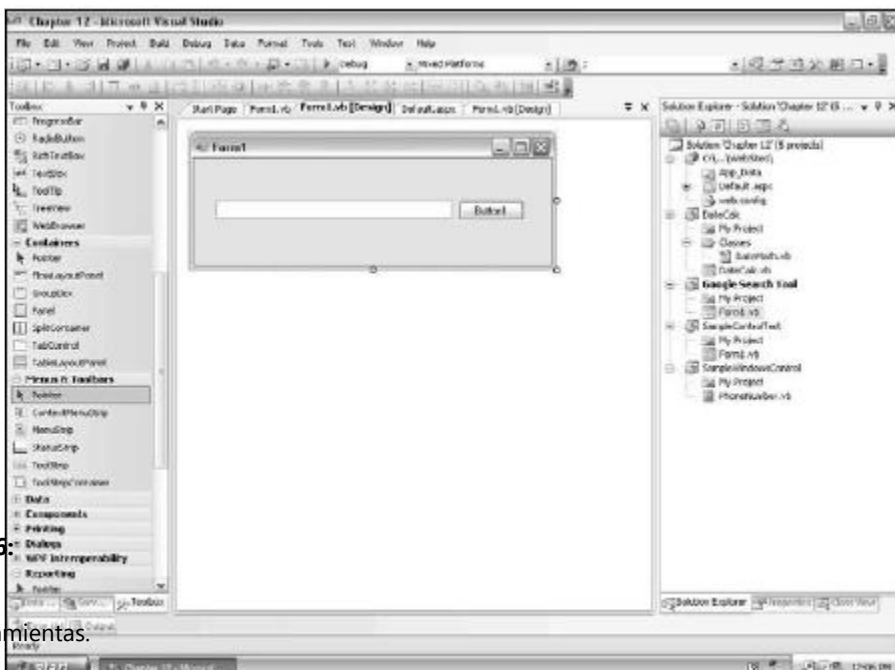
e muestra en la Figura 12-6.



Figura 12-6.

El sencillo
Google

Buscar herramientas.



5. Haga doble clic en el botón para volver a la OnClick controlador de eventos y añadir el código para incorporar el programa fuera.

El código es el siguiente:

```
Dim myProcess Como New Process()
SearchString Dim As String = _
    System.Web.HttpUtility.UrlEncode(TextBox1.Text)
 urlString Dim As String = "http://www.google.com/search?q=" + searchString
 myProcess.StartInfo.FileName = urlString
 myProcess.Start()
```

6. Haga clic en el botón Reproducir para ejecutar la aplicación. Escriba un término de búsqueda en el

cuadro de texto y haga clic en el botón para hacer que el navegador por defecto para cargar.

La figura 12-7 muestra mis resultados. Para este ejemplo, utilice Mozilla Firefox en lugar de Internet Explorer, y la clase Process todavía funciona muy bien!

El código para incluir esta aplicación fuera es sorprendentemente simple. Las líneas funcionan como sigue:

Obtener el término de búsqueda en el cuadro de texto, y combinarlo con la búsqueda URL que Google ofrece para crear una nueva variable llamada urlString:

```
SearchString Dim As String =
    System.Web.HttpUtility.UrlEncode(TextBox1.Text
)
```

```
UrlString Dim As String =
    "Http://www.google.com/search?q =" +
    SearchString
```

Establecer la nueva variable igual a la propiedad de la StartInfo.FileName objeto de proceso nuevo que he creado. Si hubiera querido un documento de Word en lugar de una URL, que podría haber proporcionado un nombre de archivo Word:

```
myProcess.StartInfo.FileName = urlString
```

Llame al método Start, y VB 2008 se ve en el registro para ver qué aplicación está configurada para manejar una URL:

```
myProcess.Start()
```

Visual Basic a continuación, ejecuta la aplicación para mí, establece la dirección URL en la barra de direcciones, y lejos va!



Aunque yo no lo uso, el programa Windows Forms creado para incorporar la herramienta de búsqueda de Google ahora tiene el control del proceso del explorador que tiene engendrado. Puedo revisar su avance, pausa su ejecución, o terminarla de la

programa si así lo desean. Para una lista completa de la funcionalidad del proceso clase, puede buscar "Clase Process" en la documentación de MSDN.



Figura 12-7:
El Proceso
clase en
trabajar
carga
un URL.

Acceso a DOS: Pero sólo como último recurso

DOS está muerto. El sistema operativo que muchos programadores experimentados cortar sus dientes en el mundo del PC es simplemente emulado en los sistemas operativos más recientes tales como los sistemas Windows Vista y el aún más reciente de Windows Server System que están en desarrollo, mientras escribo este libro.

No obstante, es posible que desee utilizar bits de DOS y mayores aplicaciones Windows en sus programas y VB 2008 es compatible con eso. Por una serie de razones hijos (tales como la posibilidad de eliminar la función de Microsoft en una tarde versión de Windows), excavando en los sistemas operativos más antiguos no es cierto. Lo que quieras hacer - a menos que no tienen otra opción. Pero es bueno para Sé que si tienes que ir tan bajo, se puede.

Ejecución de la línea de comandos programas con Shell

Si ha hecho algún trabajo de administración de sistemas, usted sabe que una gran cantidad de aplicaciones poderosas y necesarias están disponibles sólo desde el mando línea. También, usted puede escribir una aplicación de línea de comandos en Visual Studio utilizando el tipo de proyecto de consola. Para ejecutar estos programas desde sus programas de VB 2008, puede utilizar el comando Shell.

El siguiente código muestra un ejemplo muy sencillo que lanza el Windows Calculadora utilizando su nombre de comando, calc.exe. He implementado este código ponerlo en el controlador OnClick de un botón en un formulario en blanco.

```
Private Sub Button1_Click (ByVal como System.Object remitente, _
    ByVal e As System.EventArgs) Handles
    Button1.Click
    Dim As Integer CALCid
    CALCid = Shell ("C: \ Windows \ system32 \ calc.exe", _
        AppWinStyle.NormalFocus)
    End Sub
```

El ID de proceso retorna a su programa en la forma de un número entero (en este caso, CALCid), y el programa puede hacer referencia a la aplicación o el proceso, siempre y cuando dicho proceso sigue funcionando. Como con la clase de proceso, el ID del proceso le da la capacidad de mantener hasta su aplicación durante el proceso de referencia carreras, controlar su progreso, o matar a tu gusto.

El comando Shell toma unos pocos parámetros. El primero es (obviamente) el nombre de que el proceso se ejecute, completo con los parámetros del proceso puede aceptar. El segundo parámetro es el AppWinStyle, que es una colección que define cómo el proceso aparecerá al usuario. Se incluye el siguiente:

- Ocultar: No muestra en la barra de tareas.
- NormalFocus: Muestra normalmente, basado en el sistema por defecto.
- MinimizedFocus: se minimiza en la barra de tareas y tiene el foco, como si el usuario ha hecho clic en él.
- MaxamizedFocus: Llena la pantalla y tiene el foco.
- NormalNofocus: Muestra normalmente y sin foco.
- MinimizedNoFocus: se minimiza sin tener el enfoque.

El tercer parámetro de Shell, que no usé en mi ejemplo, es un valor booleano parámetro que indica al programa si el programa de llamada debe esperar el proceso llamado para terminar. El cuarto, y último, el parámetro es un tiempo de espera valor que le dice al programa que llama al dejar de lado la referencia.

Obtención de enfoque con AppActivate

No importa qué herramienta se utiliza para ejecutar un programa de su VB 2008 aplicación - si se trata de una clase de proceso, el comando Shell, o la Función Interop - usted puede dar el foco del programa con AppActivate. La función AppActivate acepta un nombre de la ventana de un programa en ejecución o un identificador de proceso y sitúa el foco a esa aplicación.

Por ejemplo, ejecutando calculadora como se muestra en el siguiente código establece inicialmente la aplicación no tener el enfoque. La línea AppActivate luego da la aplicación enfoque utilizando el ID de proceso que fue devuelto por el comando Shell.

```
Private Sub Button1_Click (ByVal como System.Object remitente, _
    ByVal e As System.EventArgs) Handles Button1.Click
    Dim As Integer CALCid
    CALCid = Shell ("C:\Windows\\system32\calc.exe", _
        AppWinStyle.MinimizedNoFocus)
    AppActivate (CALCid)
End Sub
```

En el ejemplo de procesos en la sección "Uso de otros programas con el Clase Process ", anteriormente en este capítulo, se puede dar el foco del explorador si saber exactamente lo que el nombre de la ventana será. El siguiente código demonio-STRATES esto:

```
Dim myProcess Como New Process()
SearchString Dim As String = System.Web.HttpUtility.UrlEncode (TextBox1.Text)
UrlString Dim As String = "http://www.google.com/search?q=" + SearchString
myProcess.StartInfo.FileName = urlString
myProcess.Start()
AppActivate ("Google Search: Bill Sempf - Mozilla FireFox")
```

Uso de AppActivate no sería la mejor solución en muchas circunstancias. En este ejemplo, si usted no está utilizando Mozilla Firefox, este código no funciona para usted! ¿Quieres evitar el uso de la cadena de nombre de Windows cuando el nombre varía basado en el usuario. Como resultado, el objeto del proceso tiene un ProcessId parámetro que da un resultado mucho más flexible.

Capítulo 13

Haciendo Argumentos, Devoluciones Rentables

En este capítulo

El descubrimiento de trucos para el uso de clases

La construcción de los controladores de eventos

Nivelar el rendimiento con rosca

Encontrar flexibilidad en los genéricos

Visual Basic 2008 sigue siendo, en el fondo, un lenguaje muy funcional. Gran parte de El beneficio de usar VB gira en torno a llamar a las funciones y conseguir valores espalda. Incluso con el movimiento hacia más objetos y servicios uso orientado de la lengua, que siempre tendrá que saber cómo llamar a un función y obtener un valor de retorno - no importa cuál sea la causa más elevada resulta ser.

El lenguaje Visual Basic se basa principalmente en el diseño, definición y llamando procedimientos, que pasan argumentos, y obtener valores de retorno que son útiles para su programa. La comprensión de cómo realizar funciones muy complejas y subrutinas hace que sus programas funcionen mejor y su código más legible.

En este capítulo, se analizan el diseño de procedimientos avanzados - sofisticadas funciones ciones y subrutinas. En el capítulo 6, diseñar archivos de clase, y en el capítulo 12, usted hacer que el código reutilizable realista. La información de este capítulo le da el último de los detalles que usted necesita para hacer que el código funcional óptima usted puede.

Este capítulo cubre los detalles últimos de diseño y creación de la función que no se han discutido en los capítulos anteriores. Usted encontrará lo siguiente:

- Una descripción detallada de los parámetros y valores de retorno de las funciones
- Procedimientos que aceptan diferentes conjuntos de valores mediante la sobrecarga
- Trucos para la llamada y el uso de clases
- Una introducción a la fabricación de objetos flexibles con los genéricos

Una descripción de la gestión de eventos

Un ejemplo divertido de la temporización de los objetos que utiliza hilos

Utilización de clases de manera efectiva

A pesar de que discutir el diseño de bibliotecas de clases, la programación y las pruebas en los capítulos

6 y 8, en este capítulo, se analizan las formas más eficaces de utilizar las clases. La lógica contenida en las clases es importante, ya que lo utilizan una y otra vez. Usted puede mejorar significativamente la fiabilidad global de sus esfuerzos de programación si usted sigue algunas de las sugerencias que van más en las siguientes secciones.

Por ejemplo, ¿cómo instanciar y destruir objetos pueden tener una clara impacto en la gestión de memoria de la aplicación. Saber si un recurso caro está siendo utilizada es importante. Por último, debe conocer algunos trucos del oficio que puede hacer su código más limpio.

Hacer y destruir objetos

Detrás de las escenas, cuando se crea un objeto, se está tomando la información de el archivo de programa y almacenarlos en la memoria RAM de la máquina. Usted tiene mucho más espacio en el disco duro que en RAM, por lo que debe tener cuidado con su RAM.

Cada instrucción Dim ocupa más memoria en la máquina. Algunas cosas, tales como gráficos y conexiones de red, ocupan más memoria de la que otros. Debido a este hecho, se debe tener cuidado lo que usted hace y cuando lo destruye.

Para los pequeños programas que voy otra vez en este libro, usted no tiene que preocuparse acerca de decisiones y destruir objetos en el momento oportuno. Debido a que los programas están tan pequeña, la gestión de memoria es insignificante.

Sin embargo, es fácil crear un programa que tendría que usar toneladas de memoria y donde la gestión de memoria se vuelven importantes. Por ejemplo, imaginar un programa que enrollado a través de un gran número de registros en un archivo y comenzó un mecanismo para confirmarlos. Al final del bucle, el proceso será tienen una copia del mecanismo para cada línea del archivo viva en la memoria!

.NET tiene algo llamado la recolección de basura que se deshace de objetos no utilizados. Por lo general toma objetos que fueron utilizados en un método y las destruye tras ese método se ha ejecutado. Si el método es particularmente larga, sin embargo, es posible que quiera deshacerse de un objeto inicial. Os muestro un ejemplo de esta recolección de basura en el siguiente código. Para deshacerse de un objeto inicial, se utiliza un método que todos los objetos heredan de la .NET: Finalizar.

Capítulo 13: Realización de argumentos, retorna Rentables

```
"Obtener una nueva instancia de la Calculadora de la fecha  
CurrentCalculator Dim como nuevo Calculadora  
"Deshazte de él para siempre!  
currentCalculator.Finalize ()
```

Utilización de recursos

Otra consideración para decidir cuándo destruir un objeto es lo que sucede a los recursos manejados por la clase que crea una instancia. Cubro recursos (Por ejemplo, conexiones de red y la base de datos) con mayor detalle en los capítulos 15

y 17. El recurso puede ser bloqueada por el objeto, y si usted está en función de que tiene ese recurso después, este bloqueo podría ser una mala cosa.

Las siguientes restricciones deberían regir la forma de hacer y destruir objetos. Si se encuentra en una situación de escasez de recursos, considere las siguientes recomendaciones:

Utilice una instrucción Dim o Private dimensionar el objeto correcto antes de usted está listo para su uso, en lugar de al principio de su código.

Cuando haya terminado con el objeto, llame a su método Finalize.

Tenga en cuenta que el uso de Finalización es para situaciones específicas. Trato de implementación de Administración de Utilización de Recursos cuando estoy teniendo un problema con un recursos, o estoy trabajando con objetos como redes, archivos o bases de datos. En términos generales:

El recolector de basura se ejecuta después de cada firma de método.

La mayoría de las clases de .NET son muy ligeros, y será el recolector de basura manejarlos bien.

Con y Uso

Al trabajar con objetos, no los escriba con tanta frecuencia. Realmente - echar un vistazo a el siguiente código! La palabra clave con VB dice que las siguientes líneas de código se van a utilizar "con" un objeto determinado - que es bastante cool.

```
Con myArray  
    . Add (1)  
    . Agregar (3)  
    . Sort ()  
Extremo con
```

Con gusta, Usar define un nuevo recurso que se debe utilizar como parte de la

ión de MSDN tiene una gran ejemplo utilizando una fuente en las clases de dibujo, que os muestro aquí:

221



```
Público makeBig Sub (myControl ByVal As Control)
Usando MyFont como nuevo System.Drawing.Font ("Garamond", 18.0F, FontStyle.Normal)
    myControl.Font = MyFont
    myControl.Text = "Big Garamond título!"
End Using
End Sub
```

Utilizando controladores de eventos

Los controladores de eventos son métodos que se ejecutan automáticamente cuando se produce un evento que del Marco. NET conoce. El ejemplo más obvio es un botón click - cuando el usuario hace clic en un botón, el marco lo conoce como un OnClick evento. Se ve en el código para que la pantalla de un método diseñado para que evento OnClick del botón. Este método es un controlador de eventos. Un ejemplo es el siguiente:

```
Private Sub Button1_MouseEnter (ByVal sender As Object,
    ByVal e As System.EventArgs) Handles
    Button1.MouseEnter
End Sub
```

Dos cualidades designar un controlador de eventos:

La Instrucción Handles: La instrucción Handles cuenta el marco que este método en particular está diseñado para hacer frente a un acontecimiento específico para un objeto específico.

Los tipos especiales de parámetros que requieren controladores de eventos: Estos son los El remitente y los argumentos del evento. El Remitente representa una referencia al objeto que envió el evento, y los argumentos de eventos son una costumbre colección de propiedades que se relacionan con la solicitud - como la posición de los el ratón en un clic o la hora exacta de un evento de red.

Para apoyar un controlador de eventos, el evento debe ser expuesto por un objeto. Uso Visual Studio, puede ver los eventos que están expuestos por un objeto en Diseño Ver a través de la ventana Propiedades y mediante el uso de IntelliSense. El siguiente secciones mirar a ambos.

Gestión de eventos usando el Ventana Propiedades

Con mucho, la forma más fácil de trabajar con controladores de eventos es mediante el diseño Ver. La ventana Propiedades, que yo voy en el capítulo 2, tiene una especial panel diseñado para trabajar con eventos.

Para empezar, pruebe este pequeño ejemplo:

1. Abra Visual Studio 2008 y cree un nuevo Windows de Visual Basic Aplicación del proyecto por proyecto eligiendo Archivo \Rightarrow Nuevo.

2. En el diseñador de Form1, agregue un botón y un temporizador.

El control Timer se encuentra bajo la ficha Componentes del Cuadro de herramientas.



3. Seleccione el botón y, a continuación, abra la ventana Propiedades.

4. Haga clic en el botón Eventos, que se muestra en el margen.

5. Tenga en cuenta los eventos disponibles para su uso.

La Figura 13-1 muestra lo que puede suceder a un botón en el entorno del usuario.

¿Quieres que el código se ejecute cuando el usuario pasa el cursor del ratón sobre el botón, y no cuando se hace clic en? Si es así, utilice el MouseEnter evento. ¿Quieres que el código se ejecute cuando el usuario arrastra algo más el botón? Utilice el evento DragDrop. El panel de eventos se muestra en la Figura 13-1.

6. Para ver cómo se puede utilizar, haga doble clic en el área de propiedad de la derecha de la MouseEnter evento.

Visual Studio generará automáticamente un controlador de eventos para usted y enviará a la vista Código. El evento probablemente será llamado Button1_MouseEnter. Observe la instrucción Handles? Cuenta la .NET.

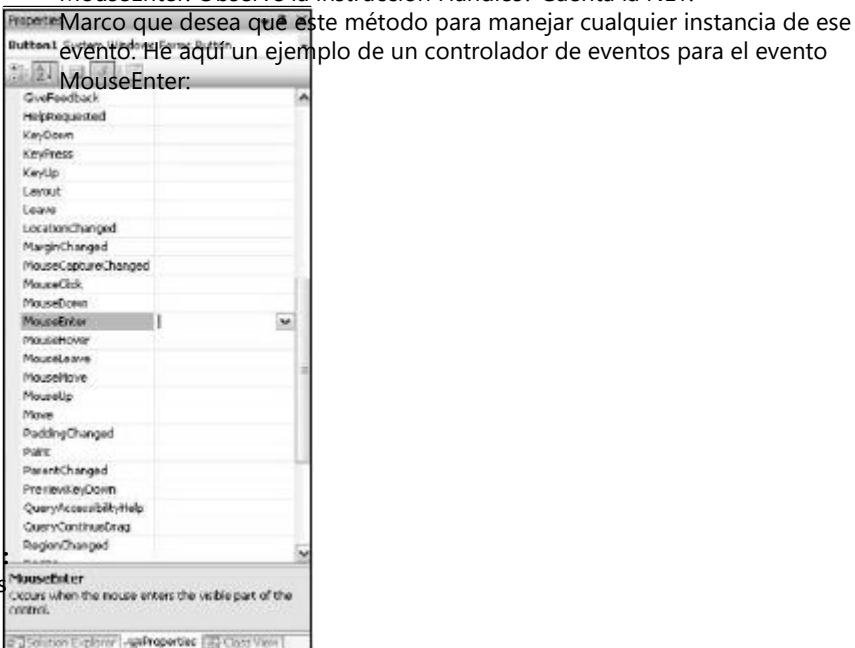


Figura 13-1:
Los Eventos
panel.

```
Private Sub Button1_MouseEnter (ByVal sender As Object,
    ByVal e As System.EventArgs) Handles
    Button1.MouseEnter
End Sub
```

Gestión de eventos con IntelliSense

La segunda forma de crear métodos para controladores de eventos está utilizando IntelliSense y el Código View. Mientras que en la vista de código, puede utilizar los selectores en la parte superior de la pantalla para seleccionar los controladores de eventos! Por ejemplo, elegir el temporizador en el lista desplegable en la parte superior de la ventana de código, en el lado izquierdo, como se muestra en

Figura 13-2. Los de la derecha desplegables lista cambia para mostrar todos los eventos de el contador de tiempo que están disponibles.

Ahora que la derecha en la lista desplegable tiene todos los eventos que se muestran en la Eventos del panel. Simplemente elija uno (como se muestra en la Figura 13-3) para generar un evento controlador para ello.

Si no está seguro de lo que el evento se trata, generar el controlador, resalte el nombre del evento en la instrucción Handles y presione F1 para obtener ayuda. Usted Siempre se puede eliminar el método sin cargo.



Manejadores de eventos Relajado

Está claro que los controladores de eventos son subrutinas. Al igual que cualquier otro subroutinante, que puede o no puede ser que necesite parámetros de entrada. El hecho es que si va a agregar la fecha en un datetimepicker con la fecha en otro datetimepicker, es posible que no sólo tiene los parámetros de entrada.

Hasta ahora, el compilador ha requerido que se incluyen las entradas para metros: el remitente y los argumentos del evento. Ahora, con clases parciales, no son realmente necesarios, y usted no tiene que ponerlos pulg Si usted recibe un defecto controlador, haciendo doble clic en un control, por ejemplo, obtendrá los parámetros de forma predeterminada. Si usted está escribiendo código, usted no los necesita. Por ejemplo:

```
Private Sub Button1_MouseEnter () Maneja Button1.MouseEnter
    'Código funcional aquí
End Sub
```

Cuando esto realmente ayuda es cuando se acaba de crear un evento para algunos- cosa a nivel de código, y está utilizando IntelliSense para ayudarle a lo largo.

A veces sólo quiero escribir privadas manijas NewFunction Sub y ver IntelliSense lo pondrá a disposición para resolver un determinado problema. Antes de esto, usted necesita para entender el formato de los parámetros de entrada. Ahora, usted no lo has.

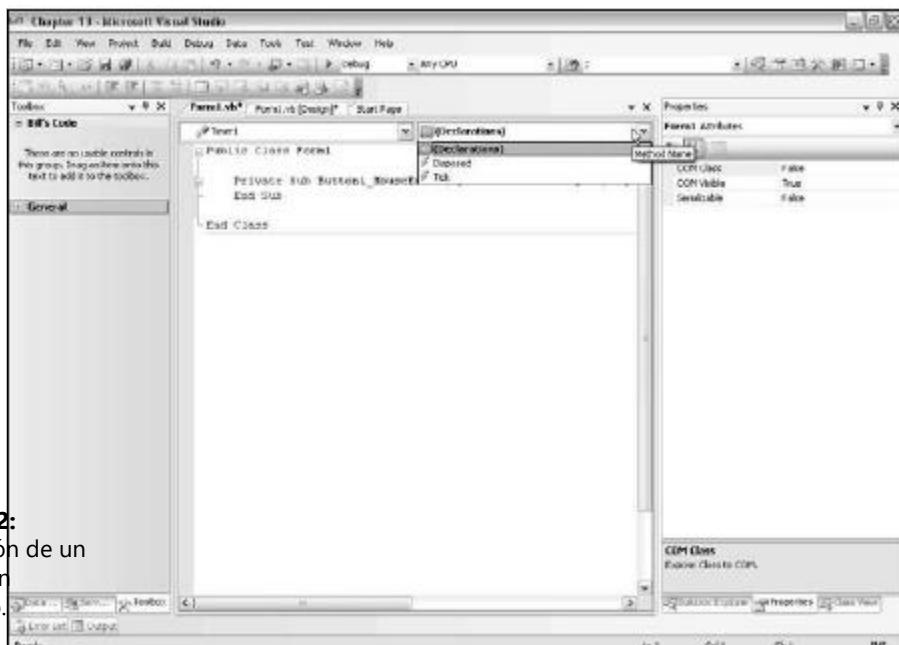


Figura 13-2:
La selección de un
objeto en
Ver código.

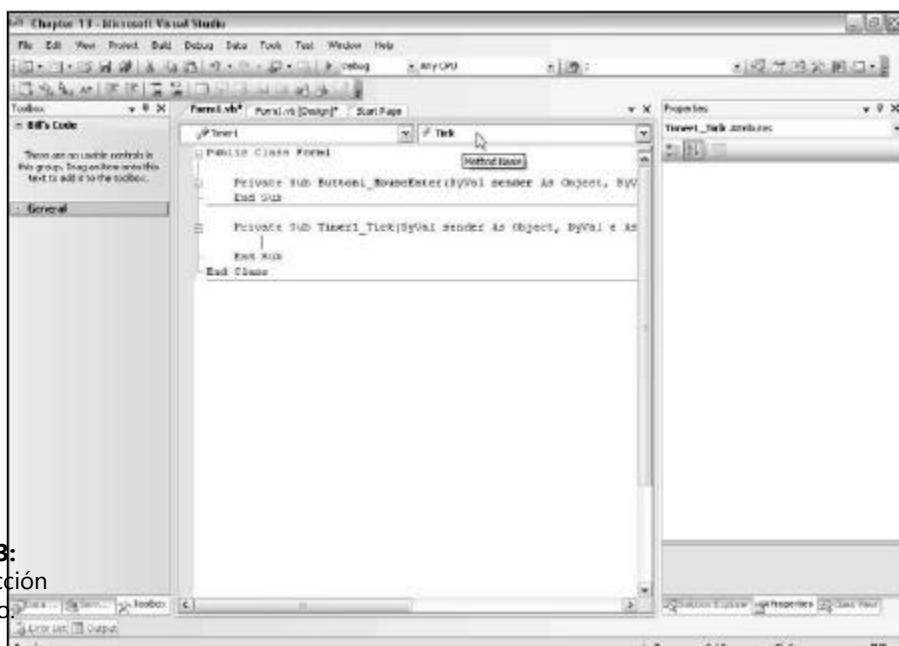


Figura 13-3:
Recolección
un evento

Procedimiento de adopción Sensible con sobrecarga

La sobrecarga es una característica organizativa que permite una variedad de parámetro recuentos y tipos que se utilizan en un procedimiento lógico. Por ejemplo, usted podría tener un método que suma dos números o números tres. Cuando se utiliza el procedimiento, que parece funcionar como una función que toma dos o tres números. Al escribir el procedimiento, sin embargo, es en realidad dos funciones.

A pesar de la sobrecarga no hace nada no se puede hacer mediante la codificación múltiple procedimientos, se puede utilizar para hacer su código más sentido.

Volviendo a usar los nombres de los procedimientos

Déjame mostrarte un ejemplo de reutilización de un procedimiento que tiene ya escrito con sobrecarga. Tomar la especificación se ha mencionado anteriormente - un añadir la función que puede agregar dos o tres números. Se puede imaginar la función en VB, sino que se vería como la siguiente:

```
Añadir la Función Pública (NumberOne ByVal As Integer, ByVal  
    numberTwo As Integer) As Integer  
    Volver NumberOne + numberTwo  
End Function
```

Con el requisito en la introducción, usted tendrá que crear un complemento procedimiento que acepta tres números enteros. Usted puede hacer una función completamente nueva, pero

¿no le gustaría sólo tiene que utilizar el nombre de complemento de nuevo, con el nuevo método signatura? Como se puede ver en el siguiente código, con sobrecarga puede:

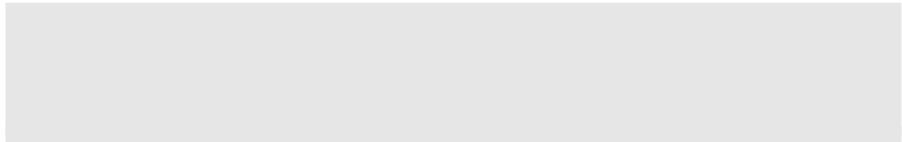
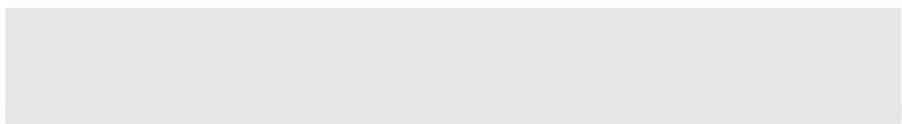
```
Añadir la Función Pública (NumberOne ByVal As Integer, ByVal  
    numberTwo As Integer numberThree, ByVal como  
    Integer) As Integer  
    Volver NumberOne + + numberTwo numberThree  
End Function
```

Si tiene experiencia en VB.NET 1.0 o 1.1, te darás cuenta de que no usé Las sobrecargas de palabras clave. Ya no es necesario a menos que se sobrecarga una función incorporada.

¿Qué hace esto para usted en el entorno de desarrollo? Principalmente, se IntelliSense hace ganar mucho más... sentido. Puedo mostrar esto en la figura 13-4.

Si utiliza IntelliSense, se puede ver los dos métodos que se muestran como un add método. Esto no tiene ningún impacto real en la funcionalidad - es sólo una conveniencia -

pero ¡qué es importante para su código de sentido mucho más!



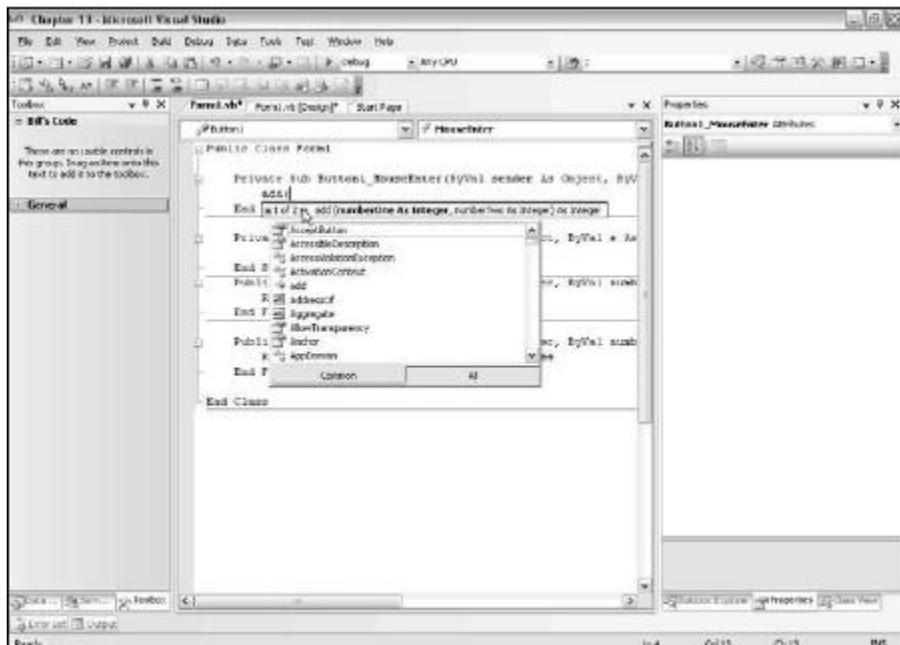


Figura 13-4:
La sobrecarga
y
IntelliSense.

Cambio de funciones incorporadas con la sobrecarga de operadores

La sobrecarga no se limita a los métodos de creación propia. Usted puede sobre-cargar métodos integrados y operadores, también! Como explico en el capítulo 9, los operadores son en su mayoría los símbolos matemáticos, con un número de bits lógicos booleanos pocos arrojados pulg Operadores son accesos directos a las expresiones matemáticas más.

Por ejemplo, en lugar del método add se muestra en la sección anterior, se sólo puede escribir answer = NumberOne + + numberTwo numberThree. Es mucho más fácil crear algo así.

Si desea que la función de complemento de hacer algo diferente, como en los avisos de si está agregando un número negativo a un número positivo, se podría escribir un funcionan para hacer esto, ya que podría sobrecargar el operador +. El siguiente código muestra un ejemplo de esto:

```
Operador Público Compartido + (ByVal NumberOne como Integer,
    NumberTwo ByVal como entero) como entero
    Si (NumberOne <0) o (numberTwo <0) Then
        MessageBox.Show ("Usted está agregando un negativo
            número! ")
    End If
Fin del operador
```

Al hacer esto, el operador complemento funciona como se diseñó originalmente, pero se ha esta funcionalidad añadida que usted ha puesto en él - y sólo cuando se agrega dos enteros.

Esta es una nueva funcionalidad en VB 2008, aunque ha estado disponible en otro Microsoft lenguas por un tiempo. La sobrecarga podría parecer ser un teórico programación concepto, pero en general lo que realmente tiene un lugar importante en la escritura código claro y conciso. Ves lo que digo en la siguiente sección.

Diseño para sobrecarga

A pesar de que la sobrecarga no tiene un impacto directo en la funcionalidad, se puede tienen un gran impacto en la forma de pensar sobre el software. Cuando usted está escribiendo Windows o Web Forms, la sobrecarga no tiene un gran impacto, pero cuando se está escribiendo bibliotecas de clases o un archivo DLL, que es muy importante.

En el último ejemplo, mira el Marco. NET en sí. Recuerde arrays, las listas de cosas que uno puede tener en la memoria? Usted ordenar unos pocos en la Parte II. De todos modos, Ordenar subrutina que cuenta con 18 versiones diferentes, cada uno aceptar un poco conjunto diferente de parámetros basados en las necesidades del programador.

Cada una de esas versiones de la subrutina Ordenar se codificaron por separado y se ve como subrutinas diferentes en el código fuente del Marco. NET. Pero a usted, el usuario de la estructura, no es sólo un método, orden y sólo pasa a tener exactamente los parámetros que usted necesita!

Sin sobrecarga, la clase Array tendrían 18 subrutinas Organizar, en lugar de uno solo. Y cuando se codifica una matriz, que tendría que Recuerdo sólo la subrutina Ordenar particular que usted quería o cavar a través de todos 18 en IntelliSense o la documentación.

Veinticuatro métodos son ya parte de la clase Array. Con toda la sobrecargas, mi recuento aproximado indica que habría 107 métodos - funciones y subrutinas - en la clase Array. Eso es más de cuatro veces más. Ahora, 220.000 métodos, propiedades y eventos de existir en el Marco. NET, por lo que sin sobrecarga, casi un millón existiría si la relación se mantenía. Eso es significativo!

Tenga en cuenta que puede sobrecargar demasiado. Si se encuentra una sobrecarga método 250 veces para hacer frente a una gran cantidad de parámetros, es posible que desee comprobar en una matriz de parámetros. Estas matrices permiten pasar de una cantidad variable de parámetros.

Así sobrecargas son realmente un problema de diseño. Cuando usted está construyendo una biblioteca de clases, pensar en cómo los métodos son nombrados y si sus patrones hacen

sentido. Haga que otro programador mirar por encima de ellos. Compárese con lo que tiene ya se ha hecho en el Marco .NET. Luego vea si la sobrecarga puede ayudar a mejorar el diseño de las clases.

Los parámetros opcionales

El uso de parámetrosopcionales es otra forma de estructura de denominación procedimiento, pero raramente lo tiene beneficios sobre la sobrecarga. Desde las primeras versiones de Visual Parámetros básicos, opcionales están disponibles para su uso al escribir subrúpulas o funciones. De hecho, Visual Basic es el único lenguaje contemporáneo que permite que los parámetrosopcionales.

Los parámetrosopcionales se utilizan mediante la inclusión de parámetros al final de la firma de método que no son necesarios para el método a ejecutar. Por ejemplo, Podría implementar parámetrosopcionales en mi método add, como se muestra en la el siguiente código:

```
Añadir la Función Pública (NumberOne ByVal As Integer, ByVal  
    numberTwo As Integer, Optional ByVal  
    numberThree = 0) As Integer  
    Dim como resultado Integer  
    Si numberThree > 0 entonces  
        resultado = NumberOne + + numberTwo numberThree  
    Más  
        resultado = NumberOne + numberTwo  
    End If  
    Volver resultado  
End Function
```

Las diferencias entre los parámetrosopcionales y procedimientos sobrecargados son bastante claro:

A encontrar un tercer parámetro llamado numberThree, que tiene un Palabra clave Optional.

El parámetroopcional tiene un valor predeterminado.

Tuve que incluir lógica en el código para controlar la posibilidad de que el parámetroopcional quedó como el valor predeterminado.

Debido a las reglas de la suma, podría haber utilizado el parámetroopcional ter no importa qué - que habría sido o bien un número o un 0, ¿no? Adición nada a 0 devuelve el valor original. Pero que es específica para este ejemplo. Si esto fuera un método de división, que no sería el caso.

Generalmente, se utiliza la sobrecarga en lugar de los parámetrosopcionales. La sobrecarga hace mucho más sentido para el programador, usando el método.

Objetos flexibles con los genéricos

Generics son exactamente lo que parecen - objetos que aceptan su propia descripción como un parámetro. En los capítulos 6, 9, y 12, menciono que las propiedades de objetos de ciertos tipos, tales como cadenas o enteros. Con los genéricos, que puede hacer que un objeto que contiene elementos de un tipo genérico para que pueda definir cuando usted lo utiliza, más que cuando lo codificar.

¿Confundido? No tengas. La palabra clave que desea es recordar Of. De es su mejor amigo. Cuando se construye una nueva clase genérica, debe ser declarada como una de cierto tipo. A continuación, una lista dentro de ese objeto puede ser una lista de todo lo que necesites el objeto a ser en tiempo de código. En tiempo de ejecución, a continuación, el objeto puede ser declarado ser de un tipo, como números enteros o manzanas, para asegurarse de que obtiene el derecho tipos de valores.

Medicamentos genéricos de construcción

Tengo un ejemplo muy simple en el siguiente código. El objetivo del personal es una lista de personas. Es posible que desee mantener los nombres en la lista, o sus identificaciones, o incluso Objetos Person si tuviera que crear una. Es posible que no sabe cuando construir el objeto Staff, por lo que hacerlo genérico, como sigue:

```
"En primer lugar, declarar el objeto con el tipo genérico.  
"El nombre puede ser cualquier cosa, me acabo de inventar el staffType  
Personal Public Class (De staffType)  
    "Se necesita una amplia privada para celebrar su lista de diez  
        Personas  
    Privada peopleArray (10) como staffType  
    'El método Add agrega uno de lo que tengas  
    'Como una instancia del objeto a la colección.  
    Public Sub Add (ByVal como persona staffType)  
        peopleArray.SetValue (persona, peopleArray.Length + 1)  
    End Sub  
End Class
```

Así que ahora cuando vas a utilizar el objeto de Personal, usted tiene que declarar qué tipo de cosas que usted va a mantener en el mismo. Esto se muestra en el siguiente código:

```
"Esto podría tener una lista de nombres  
MyStaffofStrings débil como nuevo personal (Of String)  
  
"Esto podría tener una lista de los ID  
MyStaffOfIntegers débil como nuevo personal (Of Integer)  
  
¿O incluso una lista de personas para una persona previamente codificados  
objeto  
MyStaffOfPeople Dim como nuevo Estatuto (de la persona)
```

Lo que ustedes han creado aquí es una lista personal genérico capaz de mantener lo que- cosa que quiera estar en él cuando lo utiliza. Al declarar lo que eres va a poner en ella, usted se aferra a ella. En la Figura 13-5, se puede ver que cuando El personal declarado el objeto como la celebración de números enteros, incluso apareció en IntelliSense.

El diseño de los genéricos

Tal vez usted se está preguntando cuál es el punto, y puedo entender eso. Básicamente, usted está evitando tener que escribir clases dos veces. Si el personal puede ser un colección de nombres o números de identificación, usted tendría que escribir dos veces (sin genéricos) y el nombre de dos cosas diferentes. Con los genéricos, que ya no es necesario.

Desde una perspectiva de diseño, esto es como una sobrecarga. La sobrecarga impide de tener que escribir dos métodos para manejar dos parámetros diferentes tipos. Los genéricos le impedirán tener que escribir dos clases diferentes de poseen colecciones de dos tipos diferentes de tipos.



Figura 13-5:
Uso
genéricos.

Si usted piensa en las clases como moldes y objetos como los elementos que salen de los moldes, se puede pensar de los genéricos como una forma de modificar el molde sobre la marcha. Es otra herramienta en su caja de herramientas, y no es un requisito para el diseño de la clase. Cuando se tiene un problema que sólo se pueden resolver mediante el uso de genéricos, sin embargo, usted lo sabrá.

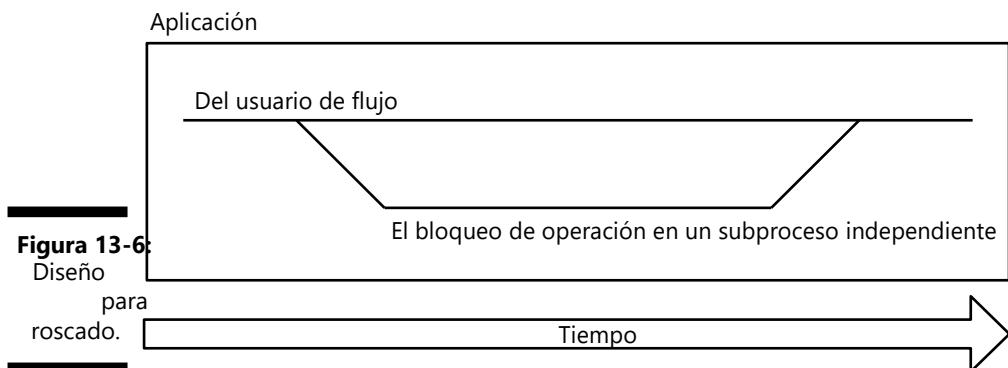
Objetos de control con Roscado

Te muestro mucho sobre la construcción y el uso de objetos, pero no mucho acerca de su alimentación y cuidado. En general, el Marco .NET se encarga de los objetos para usted, pero a veces, es necesario tomar el control. Ahí es cuando lo que necesita saber acerca de roscado.

Diseñar para roscar

Desde una perspectiva de diseño, roscado es muy simple. Si usted tiene un tiempo muy operación que consume, es posible que tenga que poner en un segundo plano y volver controlar al usuario. ¿Alguna vez has hecho algo en Word y tenía la hora vidrio aparece? Eso es un ejemplo de una operación de bloqueo. La aplicación entera tuvo que esperar a que la operación se complete antes de devolver el control a el usuario.

Si la operación es tal que la aplicación no tiene que esperar, como la programador puede ejecutar esta operación en un subproceso independiente, paralelo al la aplicación como un todo, y dejar el control del usuario de la aplicación en el hilo original. (I demostrar este concepto en la Figura 13-6.) El usuario tal vez ni siquiera saben que otro proceso está en marcha!



Capítulo 13: Realización de argumentos, retorna Rentables

Un montón de aplicaciones que utilizan el uso diario independiente Threading. Word utiliza cuando le revisar la ortografía o imprimir. Outlook lo utiliza cuando envía o recibe e-mail. Excel se utiliza al calcular los valores en las celdas. Todas estas cosas siguen mientras se está escribiendo a máquina, en su mayor parte. El número de simultánea hilos sólo está limitado por la cantidad de memoria en la máquina.

Los ejemplos de oficina son buenos porque muestran el más de uso frecuente razón para implementar threading - acceso a un recurso. Conexión de red diccionarios y bases de datos sólo puede aceptar una conexión a la tiempo. Si desea que el usuario pueda seguir utilizando el programa mientras que la aplicación está procesando, es necesario utilizar hilos.

Implementación de rosca

Para tener una idea de cómo funciona una operación de bloqueo de entrada y salida de un hilo, intente este sencillo ejemplo utilizando un temporizador para emular un funcionamiento problemático:

1. Inicie Visual Studio y cree un nuevo proyecto de aplicación para Windows en Visual Basic.

Llamé ThreadingExample mío. Lo puedes encontrar en este libro compañera sitio Web en www.vbfordummies.net.

2. Agregue dos botones, llamadas StartThread y TestLocking, a la forma.

3. Cambiar el texto de StartThread a "iniciar el cronómetro."

4. Cambiar el texto de TestLocking a "Test The Lock".

Figura 13-7 muestra un ejemplo de cómo la forma debe mirar.

5. Haga doble clic en el botón Iniciar el temporizador para iniciar la vista Código y obtener el OnClick controlador de eventos.

6. Añade un importaciones declaración a la parte superior del código -

Importaciones

System.Threading.

Esto pondrá a disposición de los métodos nuevos temporizador para que el temporizador para ejecutar en un subproceso independiente.

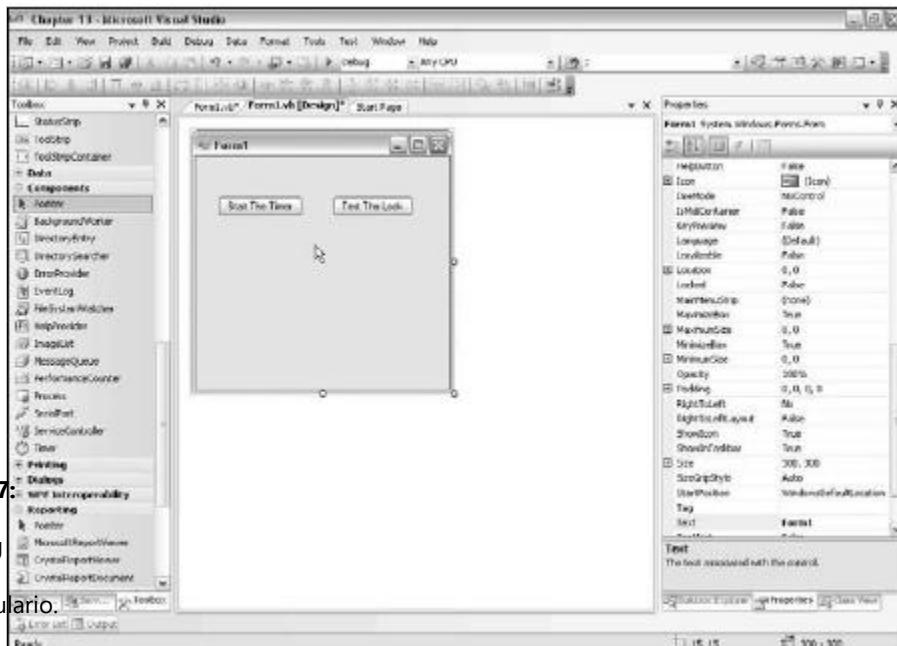
7. Agregue el código siguiente a la StartThread_Click controlador de eventos:

```
NetworkEmulator As New Timer (New _  
    TimerCallback (AddressOf FakeNetworkCall), _  
    Nada, 0, 4000)
```

**ase para generar el FakeNetworkCall
que está emulando con el temporizador:**

```
Public Sub FakeNetworkCall (estado ByVal como objeto)
    MessageBox.Show ("Esta es una llamada de red!")
End Sub
```

Figura 13-7:
La
-Threading
Ejemplo
formulario.



Esto hará que aparezca un cuadro de mensaje cada cuatro segundos - por lo general no recomendado.

9. Agregue el código siguiente en la vista de código para un controlador de eventos para el evento click del TestLocking botón:

```
TestNumber Dim As Integer = 0
Private Sub TestLocking_Click (ByVal Como remitente
    System.Object, ByVal e As System.EventArgs)
    Manijas TestLocking.Click
    TestNumber = TestNumber + 1
    Me.Text = Me.Text + TestNumber.ToString
End Sub
```

Cuando se ejecuta este código, se dará cuenta de que no pasa nada hasta que haga clic El comienzo del botón del temporizador. A continuación, cada cuatro segundos, aparecerá un cuadro de diálogo caja con el mensaje dentro prueba. Trate de mantenerse al día con ellos. De vez en de nuevo, haga clic en el botón de bloqueo de la prueba. El contador debe incrementar en la nombre de la forma. Usted puede ver mi ensayo loco en la Figura 13-8.

¿Qué hace este programa de prueba? Esto demuestra que una aplicación puede ejecutar dos cosas al mismo tiempo - de verdad. Al mismo tiempo, el programa fue condensing a 4.000 y otra vez, y se sigue permitiendo que usted trabaje con el formulario.

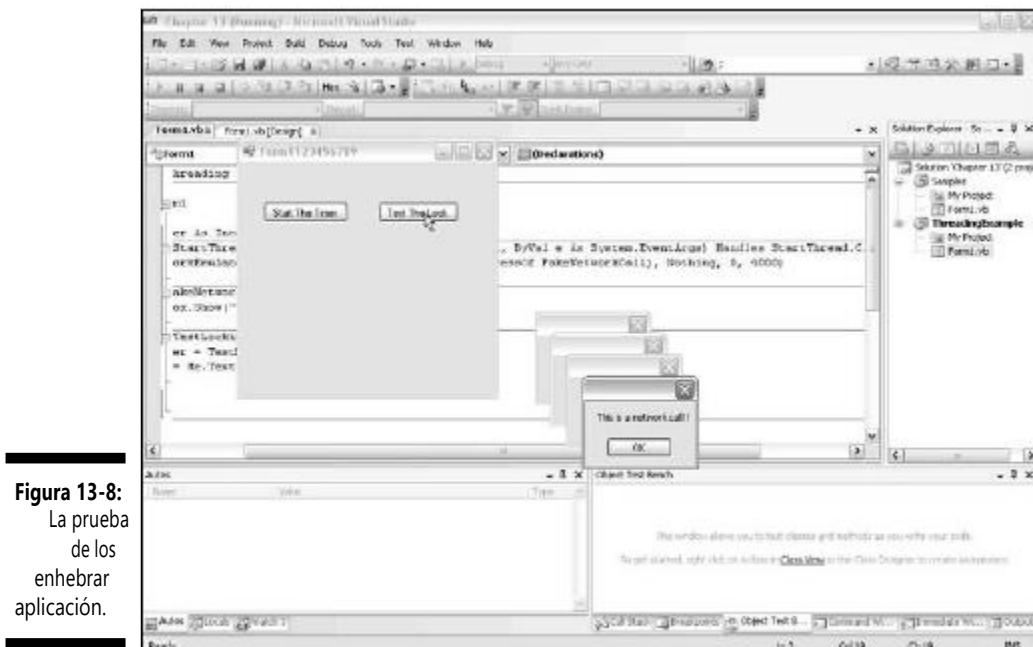


Figura 13-8:
La prueba
de los
enhebrar
aplicación.

Todo esto fue debido al objeto TimerCallback, que es un hilo encapsulado-ción. Usted le dijo a Marco. NET, "Hey, lanzar un hilo que tiene un temporizador cada cuatro segundos y llama a este método. "funciona bastante bien. Hay otras cosas que harán threading para usted, también, incluyendo los siguientes:

Muchas llamadas de red tienen una serie asíncrona de métodos, que permiten roscado.

La prioridad está integrada, por lo que al tener varios hilos, se puede decir que hilo es el más importante.

Archivo lectura y la escritura pueden ser automáticamente enhebrada.

Puede definir un bloque de código como roscado.

Todo esto se encuentra en la documentación de MSDN, por supuesto. Sólo tienes que buscar por-hilo "ción ", y comenzar con Sobre Threading. Si usted está escribiendo a gran escala de Windows aplicaciones que hacen más que leer y escribir en una base de datos, usted será interesado - ¡lo prometo!

Parte IV

Hurgando en el Marco

The 5th Wave

By Rich Tennant



"...and that's pretty much all there is to
creating a dynamic array."

En esta parte . . .

Té. NET Framework es la columna vertebral de Visual Basic 2008. Permite el acceso a bases de datos, gráficos, seguridad, archivos, y todo a punto de que su programo posible que desee utilizar. En esta parte, se utilizan las herramientas que hacer ese trabajo por usted, y, créeme, te sorprenderá la forma en que el marco hace que algunas características muy difíciles parece muy, muy fácil.

Capítulo 14

Writing Secure Code

En este capítulo

Diseño para la seguridad

La construcción de Windows y aplicaciones seguras Web

Excavando en System.Security

Seguridad es un gran tema. Ignorando por un momento todas las palabras de moda rodeadas de seguridad, estoy seguro que te das cuenta de que usted necesita para proteger su aplicación de ser utilizado por personas que no deberían estar usándolo. Usted también sabe que es necesario para evitar que su aplicación sea utilizada por las cosas que no debería ser usado para.

Al comienzo de la era electrónica, la seguridad se realizó generalmente por ofuscación. Si usted tuviera una aplicación que no quería que la gente mira a escondidas de los casos, simplemente lo escondió y nadie sabría dónde encontrarla. Por lo tanto, sería asegurar. (Recuerda la película Juegos de guerra? Los militares sólo supone que no uno podría encontrar el número de teléfono para conectarse a sus ordenadores centrales -, pero Carácter Matthew Broderick hizo.)

Eso, obviamente, no se corte nunca más, y ahora es necesario considerar seguridad, como exigencia propia de cada sistema que usted escribe. Su aplicación podría no tener los datos sensibles en el mismo, pero puede ser utilizado para llegar a otra información en la máquina? Puede ser utilizado para obtener acceso a una red que no habría de hacerlo? Las respuestas a estas preguntas asunto.

Las dos partes principales de seguridad son la autenticación y autorización. *Autenticación* es el proceso de asegurarse de que un usuario es auténtico - que es decir, que el usuario es que él o ella dice ser. El método más común de autenticación es requerir el uso de un nombre de usuario y una contraseña, aunque existen otras maneras, como por ejemplo los análisis de huella digital. La autorización es el acto de hacer

Asegúrese de que el usuario tiene la autoridad para hacer lo que él o ella le pide que hacer. Presentar permisiones son un buen ejemplo de esto - los usuarios no pueden borrar archivos sólo del sistema, por ejemplo.

El socio silencioso de seguridad es asegurarse de que su sistema no puede ser engañado en la creencia de que un usuario es auténtica y / o autorizado. Debido a este requisito, hay más en la seguridad de nombre de usuario y la contraseña sólo insertar

cuadros de texto en su programa. En este capítulo, te digo qué herramientas están disponibles en el Marco. NET para ayudarle a asegurarse de que sus aplicaciones son seguras.

Diseño de Software seguro

Software de seguridad es una buena cantidad de trabajo para diseñar con precisión. Si se rompe el proceso en pedazos, usted encontrará que es mucho más razonable que acomodar. Los patrones y las prácticas del equipo (un grupo de arquitectos de software en Microsoft que idean las mejores prácticas de programación) ha creado una sistemática enfoque para el diseño de programas de seguros que creo que le resultará muy sencillo, así que lo describen en las secciones siguientes.

La determinación de lo que debe proteger

Las diferentes aplicaciones tienen diferentes artefactos que necesitan protección, pero todo las aplicaciones tienen algo que necesita protección. Si usted tiene una base de datos en su aplicación, que es el elemento más importante que proteger. Si su aplicación es una aplicación basada en servidor, el servidor debe calificar bastante alto cuando que está determinando qué proteger.

Incluso si su programa es sólo un poco de un solo usuario de la aplicación, el software debe hacer nada malo - un extraño no debería ser capaz de utilizar la aplicación para romper en el equipo del usuario.

Documentación de los componentes del programa

Si usted piensa que el título de este artículo suena similar a la "documentación" parte del proceso de diseño descrito en el capítulo 3, tienes razón. Una gran cantidad de amenaza-modelado se acaba de entender cómo funciona la aplicación y la descripción de bien.

En primer lugar, describir lo que hace la aplicación. Esta descripción se convierte en una panorámica internacional. Si usted sigue los pasos establecidos en el capítulo 3, los casos de uso, requisitos, o historias de usuario de documentos (dependiendo de su personal metodología) debe darle un buen punto de partida.

A continuación, se describe cómo la aplicación obtiene todo eso hace al más alto nivel. Una visión Arquitectura de Software (SAO) diagrama es una gran manera de hacer esto. Este diagrama muestra que las máquinas y servicios hacen lo que en su software.

Capítulo 14: Writing Secure Code

Si le sucede a estar utilizando Visual Studio Team System, la construcción de un diagrama en el la versión Enterprise Architect es el último diagrama de SAO y es un buen modelo.

A veces, la SAO es un diagrama muy simple - si usted tiene un stand-alone Windows Forms programa como un juego, eso es todo lo que hay! Un stand-alone programo no tiene ninguna conexión de red y no hay comunicación entre el software partes. Por lo tanto, la arquitectura de software es sólo la máquina de uno.

Descomposición de los componentes en funciones

Después de tener un documento que diga lo que el software está haciendo y cómo, Necesitamos romper los pedazos individuales funcionales del software. Si usted tiene configurar el software de forma componente, las clases y los métodos de mostrar la descomposición funcional. Es realmente sencillo de lo que parece.

El resultado final de romper el software en piezas individuales está teniendo un bastante matriz decente de los componentes que necesitan ser protegidos, qué partes de la software interactuar con cada componente, qué partes de la red y de disco duro sistema de cerámica interactuar con cada componente, y cuáles son las funciones del soft-vajilla hacer lo que con cada componente.

La identificación de las posibles amenazas en esas funciones

Una vez que tenga la lista de componentes que se necesitan para proteger, se llega a hacer

la parte difícil: poner dos y dos juntos. La identificación de las amenazas es el proceso que obtiene los consultores de seguridad mucho dinero, y es casi totalmente una factor de la experiencia.

Por ejemplo, si la aplicación se conecta a una base de datos, usted tendría que imaginar que la conexión podría ser interceptada por un tercero. Si utiliza un archivo para almacenar información sensible, es teóricamente posible que el archivo puede estar en peligro.

Para crear un modelo de amenazas, es necesario categorizar las amenazas potenciales a su software. Una manera fácil de recordar las diferentes categorías de amenazas es como el acrónimo STRIDE:

n.

La manipulación de los datos o archivos: Los usuarios editar algo que no debe ser editado.

241



El repudio de la acción: Los usuarios tienen la oportunidad de decir que no lo hicieron hacer algo que realmente hizo.

Divulgación de información: Los usuarios ver algo que no debe ser visto.

Denegación de servicio: Los usuarios evitando que los usuarios legítimos el acceso a la sistema cuando lo necesitan.

Elevación de privilegios: Los usuarios obtener acceso a algo que no deberían tener acceso.

Todas estas amenazas deben ser documentados en un esquema de acuerdo con las funciones que exponer la amenaza. Esta estrategia no sólo le da una buena lista, discreta de amenazas, sino que también se centra en la seguridad de su endurecimiento aquellas partes de la aplicación que plantean el mayor riesgo de seguridad.

Valoración del riesgo

El paso final en el proceso es de tasa de los riesgos. Microsoft utiliza el DREAD modelo para evaluar el riesgo para sus aplicaciones. DREAD es un acrónimo que define cinco atributos clave que se utilizan para medir cada vulnerabilidad:

Daño potencial: El dólar costará a la empresa por una infracción.

Reproducibilidad: ¿Existen condiciones especiales para la brecha que podría hacer más difícil o más fácil de encontrar?

Explotabilidad: ¿Qué tan lejos en un sistema corporativo que un hacker podría conseguir?

Los usuarios afectados: ¿A quién afecta? ¿Cuántos usuarios?

Detectabilidad: ¿Qué tan fácil es encontrar la posible infracción?

Usted puede investigar el modelo DREAD en <http://msdn.microsoft.com/> Security, o simplemente colocar el modelo de amenazas a considerar esos atributos. La clave es determinar qué amenazas son más propensos a causar problemas y para mitigar de la mejor manera que pueda.

La construcción de Windows seguro Formularios

El marco vive en un entorno limitado estrictamente controlada cuando se ejecuta en un cliente computadora. Debido a las realidades de este entorno limitado, la configuración de seguridad política para su aplicación se vuelve muy importante.

Capítulo 14: Writing Secure Code

El primer lugar donde hay que buscar la seguridad por escrito en formularios Windows Forms es el mundo de la autenticación y la autorización. La autenticación se confirma la identidad de un usuario, y la autorización es determinar lo que él o ella puede y no se puede hacer dentro de una aplicación.

Cuando esté modelado de amenazas, es fácil considerar todos los posibles autenticación y autorización amenazas usando el acrónimo STRIDE. (Ver la sección anterior "Identificar las amenazas potenciales en las funciones" para Más información sobre STRIDE.)

Autenticación mediante inicio de sesión de Windows

Para ser franco, tengo que decir que la mejor manera para que una aplicación autorizar a un usuario es hacer uso de la conexión de Windows. Una gran cantidad de argumentos existen para esta y otras estrategias, pero todo se reduce a la simplicidad: Las cosas simples son más seguras.

Durante gran parte del software desarrollado con Visual Studio, la aplicación utilizarse en una oficina por los usuarios que tienen diferentes funciones en la empresa, para ejemplo, algunos de esos usuarios podrían estar en las ventas o departamento de Contabilidad mentos. En muchos entornos, los usuarios más privilegiados son los gerentes o administradores - otra serie de funciones. En la mayoría de las oficinas, cada empleado tiene su propia cuenta de usuario, y cada usuario se le asigna a la Windows NT grupos que son apropiados para los papeles que él o ella desempeña en la empresa.

Con la seguridad de Windows sólo funciona si el entorno de Windows está configurado correctamente. Usted no puede efectivamente construir una aplicación segura en un espacio de trabajo con un montón de equipos con Windows XP en el que todos los registros en el Administrador, porque no se puede decir que está en cuál es el papel.

Construir una aplicación de Windows Forms para tomar ventaja de la seguridad de Windows es bastante sencillo. El objetivo es comprobar que ha iniciado sesión (Autenticación) y a continuación, comprobar el papel de dicho usuario (autorización).

Los pasos siguientes muestran cómo crear una aplicación que protege la menú del sistema para cada usuario al mostrar y ocultar botones:

- 1. Inicie un nuevo proyecto de aplicación para Windows eligiendo Archivo → New**

Proyecto, y darle a su nuevo proyecto un nombre descriptivo.

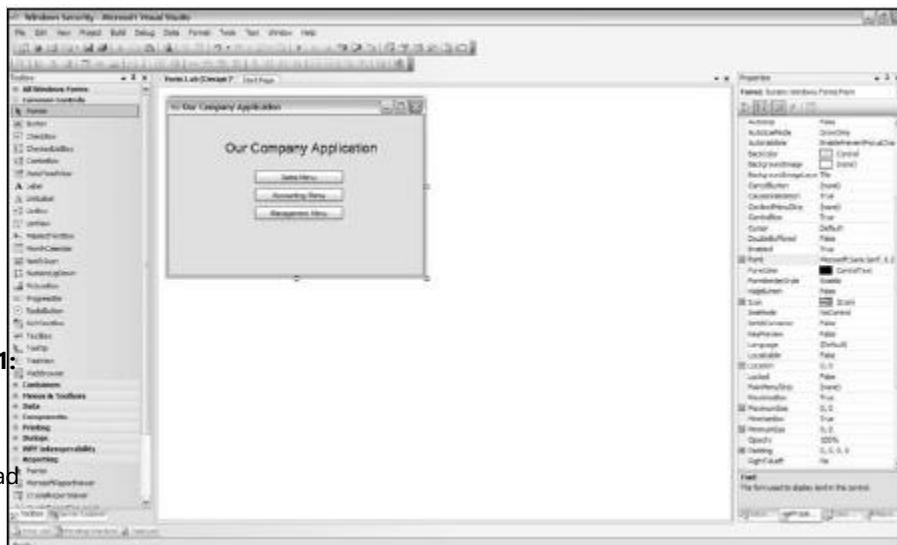
Por ejemplo, me llamó por mi proyecto de seguridad de Windows.

**completo Menú Sales, uno para
Menú Contabilidad, y uno para el menú de administración.**
Mi ejemplo se muestra en la Figura 14-1.

243



Figura 14-1:
La
ejemplo
Ventanas
Seguridad
aplicación.



3. **Establezca todas las propiedades de los botones visibles para False para que no se ver en el formulario de forma predeterminada.**
4. **Haga doble clic en el formulario para llegar a la Controlador de eventos Form1_Load.**
5. **Por encima de la Declaración de clase, importe el System.Security.**
Espacio de nombres principal, como sigue:
Imports System.Security.Principal
6. **En la Declaración de clase, un objeto de dimensión nueva identidad que representa el usuario actual con la Método de la GetCurrent WindowsIdentity objeto agregando el siguiente código:**
MyIdentity dévil como WindowsIdentity =
WindowsIdentity.GetCurrent
7. **Obtenga una referencia a esta identidad con el WindowsPrincipal clase, sigue:**
Dim myPrincipal Como WindowsPrincipal = Nuevo
WindowsPrincipal (myIdentity)
8. **Por último, en la Subrutina Form1_Load, correr un poco si-entonces para determinar qué botón para mostrar. Todo el código se muestra en el Listado 14-1.**

Listado 14-1: La aplicación de Windows Security Código

```
Public Class Form1
```

```
    MyIdentity dévil como System.Security.Principal.WindowsIdentity =
        System.Security.Principal.WindowsIdentity.GetCurrent
```

```

Dim myPrincipal Como WindowsPrincipal = Nuevo
    System.Security.Principal.WindowsPrincipal (myIdentity)
Private Sub Form1_Load (ByVal como System.Object remitente e, ByVal como
    System.EventArgs) Handles MyBase.Load
    Si myPrincipal.IsInRole ("Contabilidad") Then
        AccountingButton.Visible = True
    ElseIf myPrincipal.IsInRole ("Ventas") Then
        SalesButton.Visible = True
    ElseIf myPrincipal.IsInRole ("Administración") Then
        ManagementButton.Visible = True
    End If
End Sub
End Class

```

Para ejecutar correctamente el código, debe tener un entorno que tiene Contabilidad, Ventas y Gestión de grupos de usuarios de NT.

En algunos casos, usted no necesita este tipo de diversificación de roles. A veces usted sólo necesita saber si el usuario está en una función estándar. Sistema. Security proporciona eso, también. Usando el enumerador WindowsBuiltInRole, se puede describir acciones que deben tener lugar cuando, por ejemplo, la Administrador ha iniciado sesión:

```

Si myPrincipal.IsInRole (WindowsBuiltInRole.Administrator) A continuación,
    "Haz algo
End if

```

Cifrar la información

El cifrado es - en el centro - un proceso increíblemente sofisticado. Cinco nombres espacios están dedicados sólo a diferentes algoritmos. (Dado que el cifrado es tan complejo, yo no voy a entrar en los detalles de este libro.)

No obstante, debe comprender un elemento de cifrado para un elemento clave ción de la seguridad - cifrado de archivos. Cuando se trabaja con un archivo en un equipo con Windows

Forms, se corre el riesgo de que alguien acaba de cargar en un editor de texto y mirando a ella, a menos que haya cifrado el programa.

DES (Data Encryption Standard) es un esquema de cifrado común que es implementar simplemente en. NET. No es el cifrado más potente en estos días máquinas de escritorio de 64-bits, pero es lo suficientemente fuerte como para cifrar los archivos de datos para una aplicación de Windows. Usted puede encontrar los métodos para cifrar los ST en la DESCryptoServiceProvider en la System.Security.Cryptography espacio de nombres.

Despliegue de seguridad

Si va a implementar la aplicación mediante ClickOnce, es necesario definir el acceso a la PC que la aplicación solicitará. ClickOnce es una estrategia de implementación basada en el servidor que permite a los usuarios ejecutar Windows Forms aplicaciones desde un navegador Web o un recurso compartido de archivos. Esto se logra con la Seguridad en la pestaña Mi archivo de configuración del proyecto, como se muestra en la Figura 14-2.

Para conseguir que el archivo de proyecto Mi configuración es bastante sencilla. Seguir estos pasos:

- 1. A partir de un proyecto abierto, vaya al Explorador de soluciones pulsando Ctrl + Alt + L.**
- 2. Haga doble clic en el archivo de mi proyecto.**
- 3. Haga clic en la ficha Seguridad.**

Aquí, puede definir las características que utiliza la aplicación para que el usuario instalación que recibirá una advertencia durante la instalación en lugar de un error de seguridad cuando se ejecuta la aplicación.

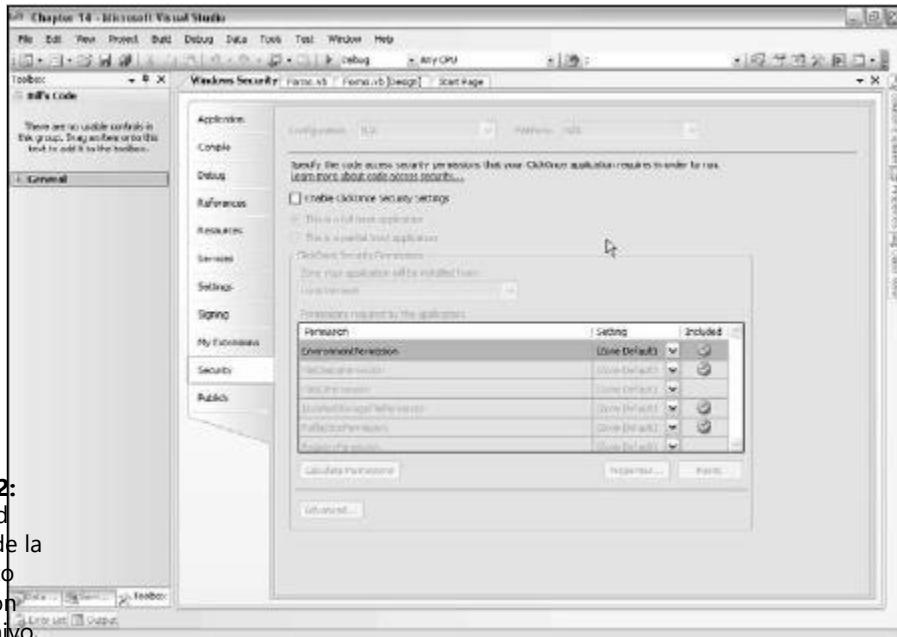


Figura 14-2:
La Seguridad
pestaña de la
Mi proyecto
configuración
archivo.

La construcción de Secure Web Forms Aplicaciones

Web Forms aplicaciones están desconectadas, programas de acoplamiento flexible que exponer a un servidor a ataques potenciales a través de los puertos expuestos utilizados por el aplicaciones. Por débilmente acoplados, es decir que tienen una instrucción de Transact-y-esperar relación con el servidor.

Debido a este acoplamiento, la construcción de la seguridad se vuelve más importante que siempre con una aplicación de formularios Web Forms. Un efecto secundario de esto es que su aplicación ción puede ser menos funcional debido a consideraciones de seguridad.

Cuando la construcción de aplicaciones basadas en Web, que gastan menos de su tiempo preocupándose acerca de la autenticación (especialmente si su aplicación se hace pública disponibilidad de tiempo) y más preocupaciones acerca de las galletas. Debido a que usted está haciendo una servidor - por lo general algo que puedes mantener en privado - a disposición del público, los programas están sujetos a un nuevo conjunto de normas para la seguridad.

La clave para proteger un servidor público es la honestidad. Hay que ser honesto con Infórmese sobre las debilidades del sistema. No piense: "Bueno, un cracker podría averiguar la contraseña haciendo XYZ, pero nadie lo haría." Confía en mí, alguien lo encontrará.

Los dos principales tipos de ataques que debería estar preocupado por una Web Forms son los ataques de inyección SQL y exploits de script.

Ataques de inyección SQL

Un ataque de inyección SQL se produce cuando un hacker entra en una línea de código SQL en un campo de entrada se utiliza para consultar una base de datos en un formulario en una página Web (tal como la Nombre de usuario y Contraseña de texto en un formulario de inicio de sesión). Esta malicioso SQL código está escrito para hacer que la base de datos para actuar de una manera inesperada o permitir al hacker acceder, alterar o dañar la base de datos.

La comprensión de inyección SQL

acker utiliza una inyección SQL es ver a un ejemplo. Por ejemplo, una página Web tiene código en el lugar que acepta un producto ID del usuario en un cuadro de texto así como los detalles de productos basados en la Identificación del producto introducido por el usuario. El código en el servidor podría tener este aspecto:

```
'Get productId de usuario
ProductId Dim As String = TextBox1.Text
'Obtener información de la base de datos.
Selectstring Dim As String = "SELECT * FROM productos WHERE
    ProductID = ""& productId &"" ";
Dim cmd Como SqlCommand SqlCommand = Nuevo (selectstring, conn)
conn.Open ()
Dim myReader Como SqlDataReader = cmd.ExecuteReader ()
'Resultados del proceso.
myReader.Close ()
conn.Close ()
```

Normalmente, un usuario introduzca la información correspondiente en el cuadro de texto. Pero un cracker intenta un ataque de inyección SQL escribir lo siguiente cadena en textBox1:

"Foobar ', DELETE FROM Artículos, -"

El código SQL que se ejecuta por el código se vería así:

SELECT * FROM productos WHERE ProductID = 'foobar', DELETE FROM
Artículos, - '

El servidor SQL se ejecuta algún código que no esperaba, en este caso, el código todo lo suprimido en la tabla de artículos.



La forma más sencilla de evitar la inyección de SQL es nunca utilizar la concatenación de cadenas para generar SQL. Usar un procedimiento almacenado y los parámetros de SQL. Usted puede leer más sobre esto en el capítulo 15.

Los ataques mediante scripts

Una explotación guión es una falla de seguridad que aprovecha el motor de JavaScript en el navegador Web del usuario. Los ataques mediante scripts tomar ventaja de uno de los más características comunes de las aplicaciones de formularios Web públicos - que permite la interacción entre los usuarios. Por ejemplo, una aplicación Web Forms puede permitir a un usuario publicar un comentario que los demás usuarios del sitio pueden ver, o puede permitir a un usuario para llenar un perfil en línea.

Entendiendo los ataques mediante scripts

Si un usuario malintencionado pusiera algo de código script en su perfil o ambiente, que un hacker podría asumir el navegador del usuario siguiente que vino a el sitio. Varios resultados son posibles, y ninguna de ellas son buenas.

Por ejemplo, la colección de cookies está disponible para JavaScript cuando un usuario llega a su sitio. Un usuario malintencionado podría poner un poco de código script en su perfil que podría copiar la cookie de su sitio a un servidor remoto. Esto podría dar al usuario acceso malintencionado a la sesión del usuario actual, porque el sesión identificador se almacena como una cookie. El usuario malintencionado podría suplantar la identidad del usuario actual.

La prevención de los ataques mediante scripts

Afortunadamente, ASP.NET impide a los usuarios escribir más código de secuencia de comandos en un formulario campo y publicarla en el servidor. Prueba con un Web Forms proyecto básico por siguientes pasos (obtendrá el error que se muestra en la Figura 14-3):

- 1. Crear un nuevo proyecto de Web Forms.**
- 2. Agregar un cuadro de texto y un botón para la página predeterminada.**
- 3. Ejecute el proyecto.**
- 4. Tipó msgbox <script> () </ script> en el cuadro de texto.**
- 5. Haga clic en el botón.**



Además, puede utilizar el método para codificar cualquier Server.HtmlEncode-cosa que la aplicación de formularios Web Forms envía a la pantalla - esto hará que código de script aparezca en el texto real y no real en HTML.



Figura 14-3:
Guion
hazañas son
bloqueadas
por
predeterminado.

Las mejores prácticas para la seguridad de su Formularios Web

Además de asegurarse de que su Web Forms evitará SQL

Ataques y vulnerabilidades de inyección de secuencias de comandos, debe tener en cuenta algunas buenas

Prácticas recomendadas para proteger sus aplicaciones Web.

La siguiente lista describe algunas de las prácticas más importantes para garantizar aplicaciones Web:

Mantenga su caja de IIS al día.

Copia de seguridad de todo.

Evite el uso de una variable de cadena de consulta.

No deje comentarios HTML en el lugar. Cualquier usuario puede ver el código HTML y ver tus comentarios seleccionando Ver código fuente ↛ en un navegador.

No depende de la validación del lado del cliente para la seguridad - que puede ser falsificada.

Utilice contraseñas seguras.

No dé por sentado lo que el usuario le envió vino de su forma y es seguro. Lo Es fácil falsificar un envío de formulario.

Asegúrese de que los mensajes de error no dar al usuario toda la información acerca de su aplicación. E-mail si mismo los mensajes de error en lugar de la pantalla-ING ellos para el usuario.

Usar Secure Sockets Layer.

No guarde nada útil en una cookie.

Cierre todos los puertos no utilizados en su servidor Web.

Apague SMTP en IIS a menos que usted lo necesita.

Ejecute un programa antivirus si permite cargas.

No haga funcionar su aplicación como administrador.

Usar cookies temporales, si es posible, mediante el establecimiento de la fecha de caducidad a una fecha pasada. La cookie sólo se mantendrá vivo durante la duración del período de sesiones.

Ponga un límite de tamaño para la carga de archivos. Usted puede hacer esto en el archivo

Web.Config,
como sigue:

```
<configuration>
  <system.web>
    <httpRuntime maxRequestLength="4096" />
  </System.web>
</Configuration>
```

Recuerde que el ViewState de Web Forms es fácilmente visible.

Usando System.Security

Aunque gran parte de las herramientas de seguridad están integrados en las clases que los usan, algunas clases desafían toda descripción o clasificación. Por esa razón, System.Security es la olla de retención para cosas que no encaja en ningún otro lugar.

Los espacios de nombres más comunes para System.Security se describen en Tabla 14-1. Muestro cómo utilizar el espacio de nombres en el Security.Principal sección anterior "Autenticación mediante inicio de sesión de Windows."

Tabla 14-1 Los títulos y Reglas

Espacio de nombres	Descripción	Clases comunes
Seguridad	Sirve como la base clase de seguridad	CodeAccessPermission, SecureString
AccessControl	Los anfitriones más sofisticado control de autorización	AccessRule, AuditRule
Autorización	Contiene las enumeraciones que describen la seguridad de una solicitud	CipherAlgorithmType
Criptografía	Contiene varios espacios de nombres que ayudar con el cifrado	CryptoConfig, DESCryptoServiceProvider
Permisos	Controla el acceso a recursos	PrincipalPermission, SecurityPermission
Política	Defiende repudio con clases para evidencia	Evidencia, sitio Url
Principal	Define el objeto de que representa la corriente contexto de usuario	WindowsIdentity, WindowsPrincipal

Capítulo 15

Obtener acceso a datos

En este capítulo

Entender el espacio de nombres System.Data

Conexión a una fuente de datos

Trabajar con datos de bases de datos

Haciendo rápidos pantallas de gestión de datos

No te predisponen al contenido de este capítulo, pero se le Problema encontrar que el acceso de datos es la parte más importante de su uso del .NET Framework. Es probable que utilizar las diversas funciones del sistema. Los datos de espacio de nombres más que cualquier otro espacio de nombres.

Sin lugar a dudas, uno de los usos más comunes de Visual Basic es la creación de aplicaciones empresariales. Las aplicaciones de negocios están a punto de datos. Esta es el blanco y negro de desarrollo con Visual Basic 2008. Mientras que enparado un poco de todo es importante comprender y completa del Espacio de nombres System.Data es muy importante cuando usted está construyendo negocios aplicaciones.

Usted puede mirar en las herramientas de datos en VB 2008 de tres maneras:

Base de datos de conectividad: Obtención de información de y hacia una base de datos una parte fundamental del espacio de nombres System.Data.

Sosteniendo los datos en contenedores dentro de sus programas: El conjunto de datos, Data Ver y contenedores DataTable son mecanismos útiles para la realización de datos. Si usted es un Visual Basic 6 o un programador ASP, te acuerdas de De registros, que han sido sustituidos por los nuevos constructos.

El Query Language Integrated ahora le permite obtener los datos de los datos contenedores utilizando Lenguaje de Consulta Estructurado (SQL) en lugar de cumplido lenguaje objeto.



Integración con controles de datos: El System.Web y System.Windows espacios de nombres funcionar para integrarse con los controles de datos. Control de datos integración utiliza conectividad de base de datos y contenedores de datos ampliamente. Esto hace que los datos controla un gran objetivo para la lectura de este capítulo.

Conozca System.Data

Los datos de .NET es diferente de los datos en cualquier otra plataforma de Microsoft tiene usado antes. Microsoft ha sido y sigue cambiando las formas como son manejada de Marco. .NET. ADO.NET, cuya adaptación se ha realizado en el nuevo System.Data la biblioteca de datos, proporciona todavía otra nueva forma de pensar acerca de los datos desde una perspectiva de desarrollo:

Desconectado: Después de obtener los datos de un origen de datos, el programa ya no ya conectado a dicha fuente de datos. Usted tiene una copia de los datos. Esta cura un problema, y es causa de otro:

- Ya no es un problema de bloqueo de fila. Debido a que tiene un copia de los datos, usted no tiene que limitar la base de datos de hacer cambios.
- Usted tiene el último problema victorias. Si dos instancias de un programa obtener los mismos datos, y los dos lo actualice, la última vuelta a la base de datos sobrescribe los cambios realizados por el primer programa.

XML impulsado: La copia de los datos que se obtiene de la fuente de datos es en realidad XML bajo el capó. Puede ser movido alrededor en una costumbre formato que Microsoft considere necesario para el desempeño, pero es igual XML de cualquier manera, haciendo que el movimiento entre plataformas, aplicaciones o bases de datos mucho más fácil.

Base de datos genéricos de los contenedores: Los recipientes no dependen del tipo de la base de datos - que se puede utilizar para almacenar datos desde cualquier lugar.

Específicas de base de datos adaptadores: Las conexiones a la base de datos son específicos la plataforma de base de datos, por lo que si desea conectarse a una base de datos específica, se necesitan los componentes que trabajan con la base de datos.

El proceso para la obtención de datos ha cambiado un poco, también. Usted solía tener un conexión y un comando que devuelve un conjunto de registros. Ahora, usted tiene una adaptador, que utiliza una conexión y un comando para llenar un contenedor DataSet. Lo que ha cambiado es la forma en que la interfaz de usuario le ayuda a conseguir el trabajo hecho.

System.Data tiene las clases para ayudarle a conectarse a una gran cantidad de diferentes datos bases y otros tipos de datos. Estas clases se dividen en el nombre-los espacios se muestran en la Tabla 15-1.

Tabla 15-1 Los espacios de nombres System.Data

Espacio de nombres	Propósito	Las clases más utilizadas
System.Data	Clases comunes a todos de ADO.NET	El DataSet contenedores, DataView, DataTable DataRow
System.Data. Común	Clases de utilidades utilizados por específicas de base de datos a clases	DbCommand, DbConnection OdbcCommand, OdbcAdapter
System.Data. ODBC	Las clases para conexiones a bases de datos ODBC tal como dBASE	OleDbCommand, OleDbAdapter
System.Data. OleDb	Las clases para conexiones a bases de datos OLEDB tales como Access	OracleCommand, OracleAdapter
System.Data. OracleClient	Las clases para conexiones para Oracle	SqlCommand, SqlDataAdapter
System.Data..SqlClient	Las clases para conexiones para Microsoft SQL Server	SqlDateTime
System.Data. SqlTypes	Para hacer referencia a los nativos tipos comunes de SQL Servidor	

Aunque hay una gran cantidad de espacio de nombres System.Data y herramientas relacionadas, lo concentrarse en la forma en Visual Studio implementa estas herramientas. En las versiones anteriores del software de desarrollo de todas las marcas y modelos, las herramientas visuales sólo hizo las cosas más difíciles a causa del problema de recuadro negro.

El problema de recuadro negro es la de tener un entorno de desarrollo de hacer las cosas para usted sobre lo que no tienes control. A veces, es bueno tener las cosas hecho por ti, pero cuando el entorno de desarrollo no es construir cosas exactamente como usted las necesita, termina generando código que no es muy útil.

Afortunadamente, ese no es el caso más. Visual Studio genera ahora completamente abierta y sensata código VB cuando se utilizan las herramientas de datos visuales. Creo que

usted estará satisfecho con los resultados.



Cómo Ajustar las clases de datos en el Marco

Las clases de datos son todos acerca de almacenamiento de información. En el capítulo 13, me refiero a colecciones, que son para el almacenamiento de información mientras se ejecuta una aplicación-principio. Tablas hash son otro ejemplo de almacenamiento de información. Colecciones mantener listas de objetos y tablas hash sostener pares de nombre y valor.

Los contenedores de datos contienen datos en grandes cantidades y ayudarle a manipular que los datos. Los contenedores de datos son las siguientes:

DataSet: Tipo del abuelo de todos ellos, el contenedor DataSet es una representación en memoria de una base de datos completa.

DataTable: Una sola tabla de datos almacenados en la memoria, el DataTable contenedor es lo más cercano que puede encontrar a un conjunto de registros, si usted es un VB 6 programador y se busca. Contenedores DataSet se componen de DataTable contenedores.

DataRow: Como era de esperar, se trata de una fila en un recipiente DataTable.

DataView: Una copia de un objeto DataTable que se pueden utilizar para ordenar y filtrar datos con fines de visualización.

DataReader: A sólo lectura, sólo hacia delante flujo de datos que se utiliza para una sola vez, los procesos, como el relleno cuadros de lista. Por lo general, se llama una manguera de bomberos.

Para llegar a sus datos

Todo en el espacio de nombres System.Data gira en torno a la obtención de datos a partir de una base de datos, como Microsoft SQL Server y llenar estos datos contenidos. Se puede acceder a estos datos de forma manual. En términos generales, el proceso pasa en etapas que miran algo como esto:

1. Se crea un adaptador.
2. Usted le dice al adaptador de cómo obtener información de la base de datos (el conexión).
3. El adaptador se conecta a la base de datos.
4. Usted le dice al adaptador de qué información es necesaria a partir de la base de datos (el comando).
5. El adaptador llena el recipiente DataSet con datos.
6. La conexión entre el adaptador y la base de datos se cierra.
7. Ahora dispone de una copia desconectada de los datos en el programa.

Capítulo 15: Obtener acceso a datos

No poner un punto demasiado fino en él, pero que no debería tener que pasar por eso proceso. Visual Studio tiene un montón de la gestión de datos para usted si usted lo permite, y recomiendo que lo hagas.

Utilizando el espacio de nombres System.Data

El espacio de nombres System.Data es otro espacio de nombres que se mezcla entre el mundo de código y el mundo herramientas visual. Aunque es más de un relación entre los controles de formulario y el espacio de nombres de datos, a menudo Parece que los datos vive justo en el interior de los controles, sobre todo cuando estás tratar con Visual Basic.

En las secciones siguientes, se refieren principalmente a las herramientas visuales, que son los una parte importante de la experiencia de Visual Basic como el código. En primer lugar, yo voy a conectar ING a fuentes de datos y, a continuación os muestro cómo escribir una aplicación rápida con una de esas conexiones. Por último, quiero repasar un poco de la parte del código.

Conexión a una fuente de datos

No hay más que conectar a una base de datos que establece una sencilla conexión a Microsoft Access. Desarrolladores de Visual Basic tiene que conectarse a mancuadros, archivos de texto, bases de datos inusuales, servicios Web y otros programas. Todos de estos sistemas dispares se integran en las ventanas y las pantallas de Web, con la actualización, añadir y eliminar funcionalidad para arrancar.

Para llegar a estas fuentes de datos depende sobre todo de las clases de adaptador de la espacios de nombres individualizados base de datos. Oracle tiene su propio, al igual que SQL Server. Las bases de datos que son ODBC (Open Database Connectivity) compatible (por ejemplo, Microsoft Access) tienen sus propias clases de adaptador, y el más reciente OLEDB (Object Linking and Embedding Database) protocolo tiene uno, también.

Afortunadamente, un asistente se encarga de la mayor parte de esto. La configuración de orígenes de datos Wizard es accesible desde el panel Orígenes de datos, en la que pasan gran parte de su tiempo al trabajar con datos. Para empezar a utilizar el origen de datos Asistente de configuración, siga estos pasos:

chivo \Rightarrow New

Proyecto. Seleccione una aplicación para Windows de Visual Basic y asígnelle el nombre apropiado.

Para este ejemplo, el nombre del proyecto de aplicación de Windows Acceso Datos.

2. Para abrir el panel Orígenes de datos, seleccione Datos ⇒ datos muestran Fuentes, o pulse Mayús + Alt + D.

Cabe decir que usted no tiene fuentes de datos, como se muestra en la Figura 15-1.



Figura 15-1:
Los Datos de Carácter
Fuentes
panel.

3. Haga clic en el vínculo Agregar nuevo origen de datos en el panel Orígenes de datos.

Esto nos lleva a la fuente de datos Asistente para la configuración. El asistente tiene una variedad de tipos de fuentes de datos que usted puede elegir. El más interesante de ellos es el origen del objeto, lo que le da acceso a un objeto en una asamblea para enlazar los controles a.

Haga clic en el tipo de fuente de objetos para ver las opciones que hay, como se muestra en la Figura 15-2, y haga clic en el botón Anterior para volver a la pantalla anterior.

Usted puede escoger un servicio Web para conectarse a una función en otra computadora. Cubro la creación de servicios Web y el consumo en el capítulo 7, pero esta funcionalidad te prepara para tener una fuente de datos junto con la Web servicio de referencia. Es muy bueno. He seleccionado la USZipSoap de WebServiceX como un ejemplo en la Figura 15-3.

Cuando haya terminado de mirar a su alrededor, haga clic en el botón Cancelar para volver.



Figura 15-2:
El uso de
un
para un
objeto
datos fuente.

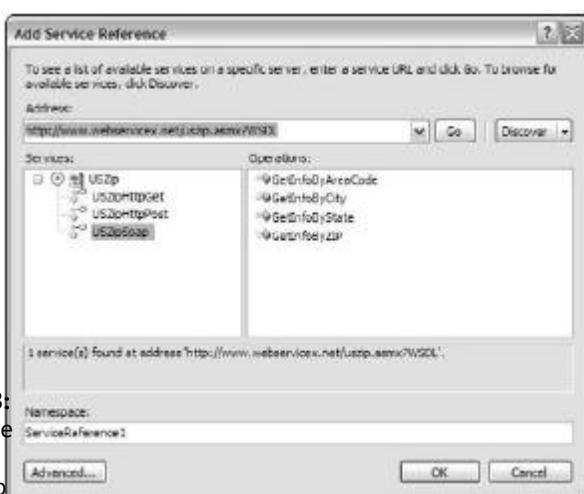


Figura 15-3:
El uso de
un
Servicio Web
para un
conjunto de
datos

4. Haga clic en el tipo de fuente de datos de base de datos que deben adoptarse para el elegir su

Los datos de pantalla de conexión, como se muestra en la Figura 15-4.

El punto de acceso más común es una base de datos.

5. Si usted tiene una conexión de datos existente, aparecerá en el menú desplegable

lista. De lo contrario, tendrá que hacer clic en el botón Nueva conexión para abrir

el cuadro de diálogo Agregar conexión, como se muestra en la Figura 15-5.

Para este ejemplo, haga clic en el botón Nueva conexión y seleccione Northwind, la base de datos de ejemplo de Microsoft.

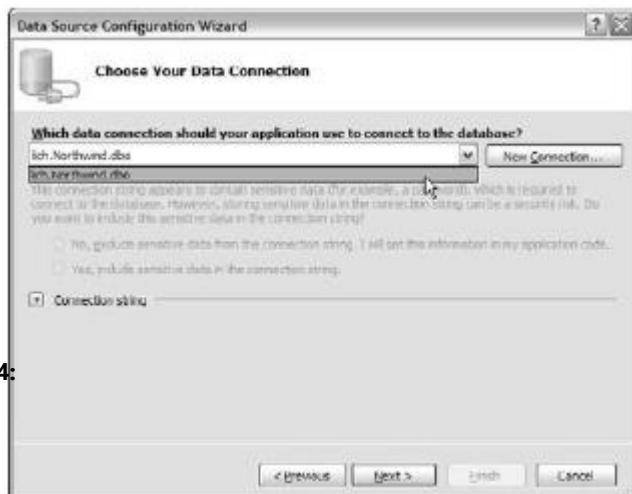


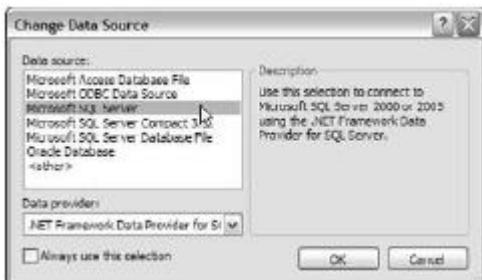
Figura 15-4:
Elegir
sus datos
conexión.



Figura 15-5:
El complemento
Conexión
Cuadro de diálogo.

Aparecerá el cuadro de diálogo Agregar conexión se supone que se va a conectar a un servidor SQL. Si ese no es el caso, haga clic en el botón Cambiar para seleccionar una base de datos diferente en el cuadro de diálogo Cambiar origen de datos, como se muestra En la Figura 15-6. Para este ejemplo, he elegido Microsoft SQL Server y clic en el botón Aceptar.

Figura 15-6:
El Cambio
Fuente de
datos
Cuadro de
diálogo.



6. Seleccione un servidor en el nombre del servidor en la lista desplegable.
 7. Seleccione la base de datos Northwind en la selección o Ingresa una base de datos
 - Nombre de la lista desplegable.
 8. Haga clic en el botón Aceptar.
- Vuelve a la pantalla Elegir tu conexión de datos.
9. Haga clic en el botón Siguiente para guardar la cadena de conexión a la aplicación
 10. Acepte los valores predeterminados, haga clic en el botón Siguiente.

Ve la pantalla Choose Your Database Objects. Aquí se puede elegir las tablas, vistas o procedimientos almacenados que deseé utilizar.

11. En tablas, seleccione Orders y Order Details (como se muestra en la Figura 15-7), y haga clic en el botón Finalizar.

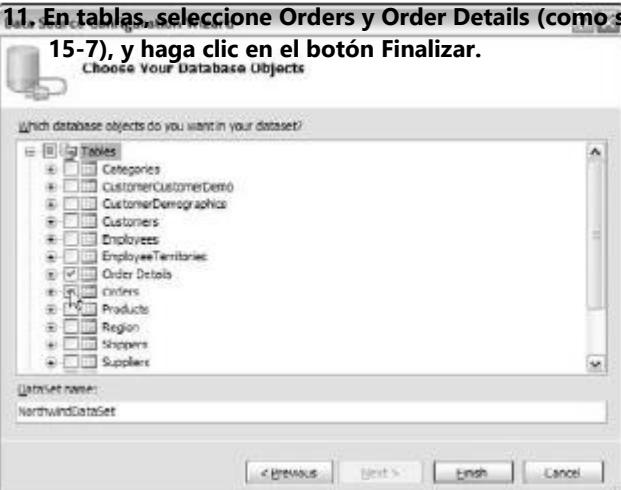


Figura 15-7:
Selección
sus datos
objetos.



¡Ya está! Si nos fijamos en el panel Orígenes de datos, usted encontrará que los nuevos datos conexión se añadió, como se muestra en la Figura 15-8.

Tenga en cuenta que el panel Orígenes de datos tiene las tablas Orders y los datos Panel de conexiones tiene todas las tablas. Esto es porque el contenedor DataSet que generó en el asistente sólo tiene la tabla Pedidos y tablas relacionadas en el mismo. El panel de conexiones de datos muestra todo en la base de datos.

Siguiendo los pasos anteriores, se crean dos entidades significativas en Visual Studio:

Se crea una conexión con la base de datos, se muestra en la Base de Datos Explorer. Usted encontrará que se pega alrededor - es específico para esta instalación de Visual Studio.

También se crea un origen de datos del proyecto que es específica para este proyecto, y no va a estar allí si se inicia otro proyecto.

Ambos son importantes, y que proporcionan una funcionalidad diferente. En este capítulo, me centro en el origen de datos específico del proyecto se muestra en los datos Fuentes de panel.

Trabajar con las herramientas visuales

Las herramientas de datos de RAD para Visual Basic son una gran mejora sobre lo que tiene previamente proporcionada por Microsoft. Las herramientas RAD de datos en Visual Basic 2008 son utilizables, hacer lo que usted necesita, y realmente escribir código decente para usted.

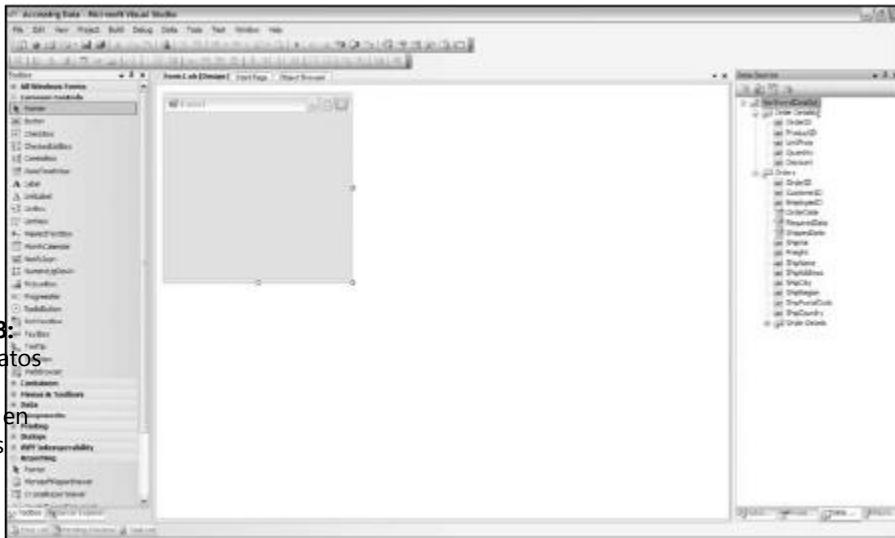


Figura 15-8:

Nuevos datos
conexiones
aparecen en
los datos
Fuentes
panel.



Usted necesita saber que yo nunca, nunca mostrar este tipo de magia negra si no era una buena práctica. En el pasado, las herramientas que hizo algo que no se podía ver con qué frecuencia su trabajo mal. Usando las herramientas, a la larga, en realidad hizo su programa peor. Las nuevas herramientas, sin embargo, son una muy buena manera de construir software. La gente puede decir que estoy equivocado, pero en realidad no es malo. Pruébelo!

Si hace clic en una tabla en el panel Orígenes de datos, una flecha hacia abajo aparece. Selecciónelo y ves algo muy interesante, como se muestra en la Figura 15-9. La lista desplegable aparece que le permite elegir la forma en que la tabla es integrado en formularios Windows Forms.

Cambie la tabla Orders a una vista Detalles. Se utiliza para crear un tipo de detalle forma - que fácilmente le permite al usuario ver y cambiar los datos. A continuación, arrastre la tabla a la forma, y la vista de detalles se ha creado para usted, como se muestra en Figura 15-10.

Un montón de cosas que sucedió cuando se le cayó de la mesa en su formulario:



Los campos y nombres de campo se han añadido.

Los campos están en el formato más apropiado - tenga en cuenta que la Orden La fecha es un selector de fechas.

El nombre del campo es una etiqueta.

Visual Studio agrega automáticamente un espacio en el que cambia de caso.

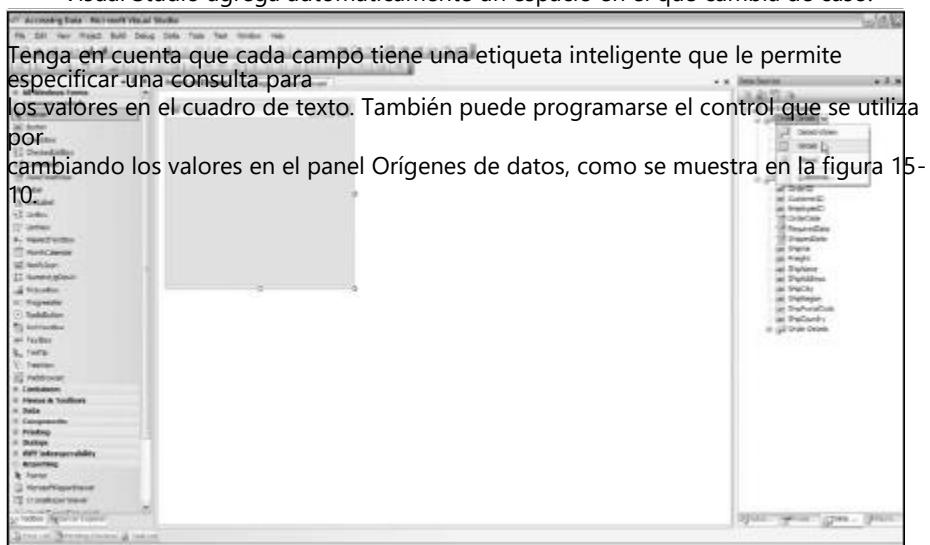


Figura 15-9:
Mesa
Opciones
desplegable
lista.

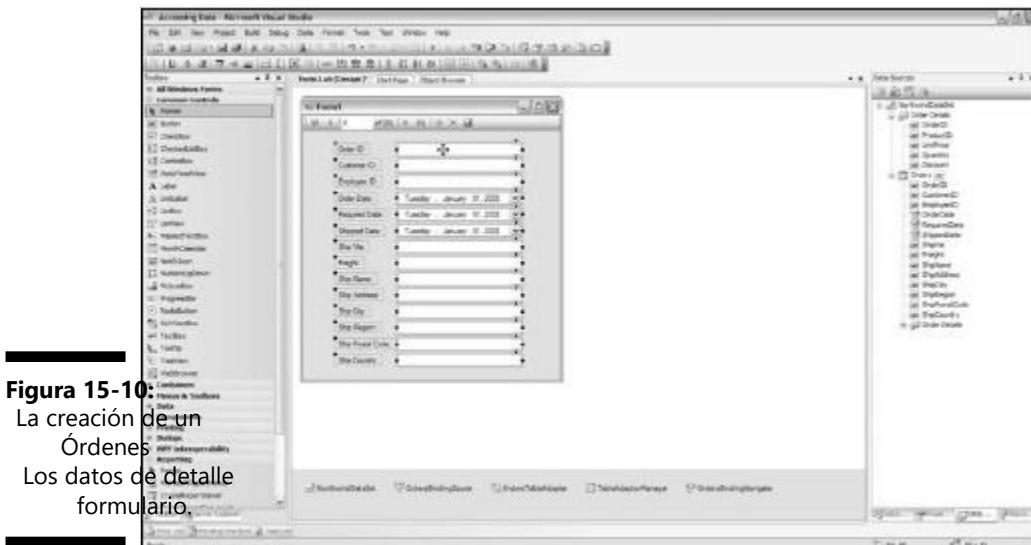


Figura 15-10:
La creación de un
Órdenes
Los datos de detalle
formulario.

También, un VCR Bar (llamado técnicamente la BindingNavigator) se añade a la parte superior de la página. Al ejecutar la aplicación, se puede utilizar la barra de VCR para ciclo entre los registros de la tabla.

Finalmente, cinco completamente basados en códigos objetos se añaden en la Bandeja de componentes en la parte inferior de la página: el conjunto de datos denominado NorthwindDataSet, la BindingSource llamado OrdersBindingSource, los TableAdapter llamado OrdersTableAdapter, TableAdapterManager y el BindingNavigator llamados objetos OrdersBindingNavigator.

Haga clic en el botón de reproducción y se puede fácilmente ver la obra Bar VCR. Se puede caminar a través de los elementos de la base de datos sin problemas, como se muestra en la figura 15-11. Es como trabajar en Access o FoxPro, pero con la calidad de la empresa!

Se pone mejor. Siga estos pasos para crear una interfaz de tabla secundaria:

1. Abra la tabla Order en el panel Orígenes de datos, haga clic en el signo más firme al lado de la mesa.
2. Desplácese hacia abajo hasta que vea la tabla Detalles de pedidos anidado en el Reglamento mesa.
3. Arrastre el ejemplo de la tabla a la forma y colocarla debajo de la Pedidos campos que se colocan en la forma anteriormente en esta sección (consulte la Figura 15-10).
4. Haga clic en el botón Reproducir para ejecutar el ejemplo, como se muestra en la Figura 15-

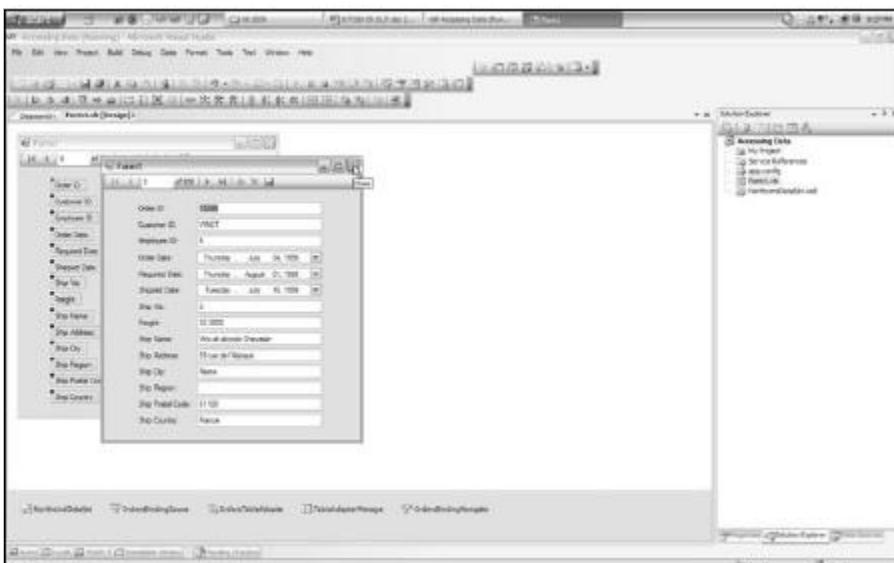


Figura 15-11:
Ejecución de
la
ejemplo

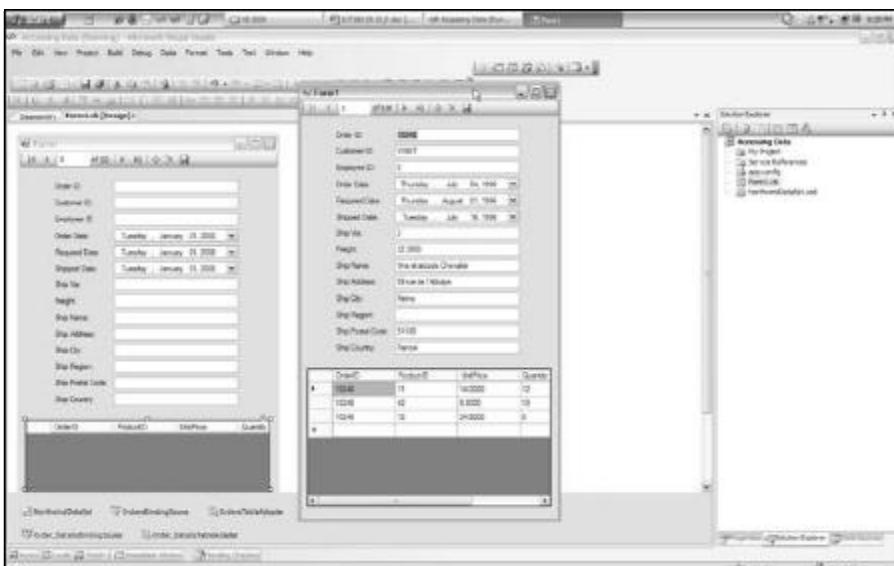


Figura 15-12:
Un completo
formulario
de edición.

Tiene un funcionamiento, fácil de usar padre / hijo forma, con órdenes y el orden detallados. La creación de este tipo habría requerido a escribir 100 líneas de código, incluso en las versiones anteriores de VB. Con la capacidad de elegir un conjunto para una fuente de datos que Visual Basic 2008 le concede, la forma es incluso empresa lista. Es algo bastante resbaladiza.

Escribiendo código de datos

En la mayoría de los entornos de desarrollo empresarial, usted no va a utilizar el visual herramientas para construir software de acceso a datos. Generalmente, una infraestructura ya está en lugar.

La razón de esto es que el software de la empresa a menudo tiene muy requisitos específicos mentos, y la forma más fácil de manejar esas especificaciones es de única y código personalizado. En resumen, algunas organizaciones no quieren hacer las cosas de la forma en que Microsoft les hace.

La producción de las herramientas visuales

La razón por la que las herramientas visuales no son de uso frecuente en las empresas del medio mentos es que el código de las herramientas apagar es bastante sofisticado. Si cambiar a la vista Código, haga clic en una instancia de un objeto (por ejemplo, la CustomersTableAdapter objeto) y seleccione Ir a definición, te vas para el código detrás del diseñador. Cuento con 212 líneas de código en el archivo - gran parte de ella va a definir los objetos de datos que se utilizan, como se muestra en la el siguiente código:

```
'NorthwindDataSet
Me.NorthwindDataSet.DataSetName = "NorthwindDataSet"
'CustomersBindingSource
Me.CustomersBindingSource.DataMember = "Clientes"
Me.CustomersBindingSource.DataSource Me.NorthwindDataSet =
'CustomersTableAdapter
Me.CustomersTableAdapter.ClearBeforeFill = True
'CustomersBindingNavigator
Me.CustomersBindingNavigator.AddNewItem = Me.bindingNavigatorAddNewItem
Me.CustomersBindingNavigator.BindingSource = Me.CustomersBindingSource
Me.CustomersBindingNavigator.CountItem = Me.bindingNavigatorCountItem
Me.CustomersBindingNavigator.CountItemFormat = "de {0}"
Me.CustomersBindingNavigator.DeleteItem = Me.bindingNavigatorDeleteItem
Me.CustomersBindingNavigator.Items.AddRange (Nueva
    System.Windows.Forms.ToolStripItem ())
    {Me.bindingNavigatorMoveFirstItem,
    Me.bindingNavigatorMovePreviousItem, Me.bindingNavigatorSeparator,
    Me.bindingNavigatorPositionItem, Me.bindingNavigatorCountItem,
    Me.bindingNavigatorSeparator1, Me.bindingNavigatorMoveNextItem,
    Me.bindingNavigatorMoveLastItem, Me.bindingNavigatorSeparator2,
    Me.bindingNavigatorAddNewItem, Me.bindingNavigatorDeleteItem,
    Me.bindingNavigatorSaveItem})
Me.CustomersBindingNavigator.Location = Nueva System.Drawing.Point (0, 0)
Me.CustomersBindingNavigator.MoveFirstItem = Me.bindingNavigatorMoveFirstItem
Me.CustomersBindingNavigator.MoveLastItem = Me.bindingNavigatorMoveLastItem
Me.CustomersBindingNavigator.MoveNextItem = Me.bindingNavigatorMoveNextItem
Me.CustomersBindingNavigator.MovePreviousItem =
    Me.bindingNavigatorMovePreviousItem
Me.CustomersBindingNavigator.Name = "CustomersBindingNavigator"
Me.CustomersBindingNavigator.PositionItem = Me.bindingNavigatorPositionItem
```

```
Me.CustomersBindingNavigator.Size = Nuevo System.Drawing.Size (292, 25)
Me.CustomersBindingNavigator.TabIndex = 0
Me.CustomersBindingNavigator.Text = "BindingNavigator1"
'BindingNavigatorMoveFirstItem
Me.bindingNavigatorMoveFirstItem.DisplayStyle =
    System.Windows.Forms.ToolStripItemDisplayStyle.Image
Me.bindingNavigatorMoveFirstItem.Image =
    CType (resources.GetObject ("bindingNavigatorMoveFirstItem.Image"),
    System.Drawing.Image)
Me.bindingNavigatorMoveFirstItem.Name = "bindingNavigatorMoveFirstItem"
Me.bindingNavigatorMoveFirstItem.Text = "Mover primero"
```

No hay nada malo con este código, pero es muy genérico a propósito de apoyar algo que cualquier persona que quiera hacer con él. Los clientes empresariales a menudo quieren asegurarse de que todo se hace de la misma manera. Por esta razón, suelen definir un formato de datos de código específico y esperan que su software deseadores de usar que, en lugar de las herramientas visuales.

Datos de código Basic

El código del proyecto de ejemplo es bastante simple:

```
Private Sub Form1_Load (ByVal como System.Object remitente e, ByVal como
    System.EventArgs) Handles MyBase.Load
    'TODO: esta línea de código carga datos en la' NorthwindDataSet.Customers 'mesa.
    Puede mover, o eliminar, según sea necesario.
    Me.CustomersTableAdapter.Fill (Me.NorthwindDataSet.Customers)
End Sub

Private Sub bindingNavigatorSaveItem_Click (ByVal remitente como System.Object, ByVal
    e como System.EventArgs) Handles bindingNavigatorSaveItem.Click
    Si Me.Validate Entonces
        Me.CustomersBindingSource.EndEdit ()
        Me.CustomersTableAdapter.Update (Me.NorthwindDataSet.Customers)
    Más
        System.Windows.Forms.MessageBox.Show (Me, "se produjo errores de validación."
            "Guardar", System.Windows.Forms.MessageBoxButtons.OK,
            System.Windows.Forms.MessageBoxIcon.Warning)
    End If
End Sub
```

Si bien esto es bastante sencillo, es evidente que no es todo lo que necesita. El resto del código está en el archivo que genera la forma visual en sí, supportar los componentes visuales.



Esto se vuelve especialmente útil cuando se desea crear un servicio Web o un biblioteca de clases - aunque hay que señalar que todavía se puede utilizar el visual herramientas de ese tipo de proyectos.

Una vez llegue el momento que desee conectarse a una base de datos sin necesidad de utilizar el herramientas visuales. Discutir los pasos de la sección anterior "Cómo las clases de datos El ajuste en el Marco ", y aquí les muestro el código para ir con ella:

```
MyConnection As New SqlConnection  
myConnection.ConnectionString =  
    "Server = (local); database = Northwind; Trusted_Connection = True"  
MyAdapter SqlDataAdapter Dim  
myAdapter = New SqlDataAdapter ("SELECT * FROM Clientes", myConnection)  
MyDataSet As New DataSet  
myConnection.Open ()  
myAdapter.Fill (myDataSet)  
myConnection.Close ()
```

Después de ejecutar este código, usted tendría tabla del Cliente en un DataSet contenedor, tal como lo hizo en las herramientas visuales en la sección anterior "Cómo el Clases de datos encajan en el marco. "Para acceder a la información, se establecería el valor de un cuadro de texto con el valor de una celda en el contenedor DataSet, así:

```
TextBox1.Text = myDataSet.Tables (0).Filas (0) ("CustomerName")
```

Para cambiar al siguiente registro, usted tendría que escribir código que cambia el Filas (0) en filas (1) en el siguiente ejemplo. Como se puede ver, sería una feria cantidad de código.

Es por eso que algunas personas utilizan el código de datos básicos para obtener las bases de datos. Cualquiera de los dos se utilizan las herramientas visuales, o se utiliza un agente de datos de algún tipo.

Con corredores de datos

Un corredor de datos es un bloque de código que hace que los datos de acceso más simple. Básicamente, se pone toda la tubería complicado de la conexión de base de datos en un archivo de clase para puede llamar - de un lado - el código que es común a todos sus datos-base de accesos.

Los patrones y las prácticas del equipo de Microsoft (<http://msdn.microsoft.com / prácticas>) creó un corredor llamado Enterprise Library de acceso a datos Aplicación del bloque que hace exactamente esto. Está diseñado para su uso por la gran empresas que menciono en este capítulo, pero el corredor es excelente para utilizar en aplicaciones independientes y, sobre todo, si usted encuentra que el visual herramientas no hacer el truco por alguna razón.

Con el Enterprise Library Data Access Application Block datos corredor, sólo tiene que preocuparse por las partes más personalizadas de los datos proceso de acceso, concretamente estos tres pasos generales:

1. Crear el objeto de base de datos.
2. Suministrar los parámetros para el comando, si son necesarios.
3. Llame al método apropiado.

Estos pasos pueden ser ejecutados en una línea de código que se ve algo como esto:

```
myDataSet = DatabaseFactory.CreateDatabase ("Neptuno"). ExecuteDataSet ("SELECT *  
FROM Clientes");
```



Estos pasos se supone que tiene la biblioteca de empresa ya instalada, referencia, y configurado. Usted puede encontrar la biblioteca Enterprise en <http://msdn.microsoft.com/practices/default.aspx?pull=/library/en-us/dnpag2/html/entlib.asp>, o se puede obtener a partir de esta libro compañero sitio web en www.vbfordummies.net.

Usar LINQ

LINQ es un conjunto de clases en el Marco. NET diseñado para ayudarle con datos manipulación - al igual que algunas de las clases en System.Data. Es más o menos una nueva manera de manejar datos en. NET, en lugar de los métodos de ADO.NET

Conjuntos de datos. El objetivo es hacer bases de datos relacionales, como las que se encuentran en

Microsoft Access y SQL Server, se parecen más a las clases discutidas en Capítulo 6.

LINQ es un tema muy amplio. Incluso el artículo introductorio sobre él en el MSDN es 50 páginas impresas. Aunque una discusión completa está más allá del alcance de este libro,

he aquí una breve introducción.

En el ejemplo de esta sección, se obtiene la información del pedido desde NorthWind. Usar LINQ, primero debe definir lo que vas a obtener de la tabla con una clase, tal y como lo hicimos con el DateCalculator en el capítulo 6. Adición de un modificadores de la clase hace pocos datos-aware para que podamos consultar la información con

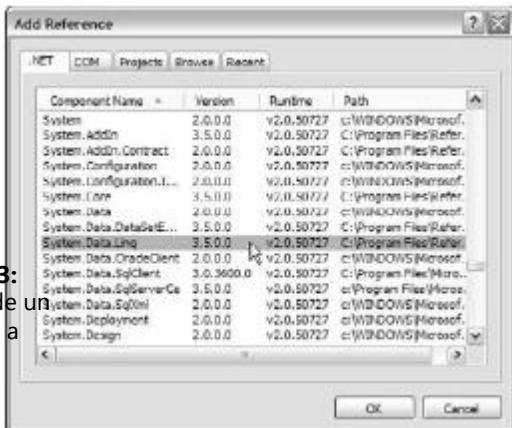
LINQ. Utilice los siguientes pasos para empezar con un proyecto básico de LINQ:

1. Agregue una referencia al System.Data.Linq, utilizando el Agregar referencia panel (como se muestra en la Figura 15-13).

2. Añadir unos pocos declaraciones Importaciones en el código que tendrá el LINQ declaraciones:

```
Imports System.Data.Linq  
Las importaciones System.Data.Linq.Mapping
```

Figura 15-13:
Adición de una
referencia a la
Sistema.
Data.Linq.



3. Hacer una clase del proyecto que hace referencia a las Atributo de la tabla, como se muestra en el siguiente código:

```
<Table(Name:="Orders")>_
Las órdenes Clase pública
<Column()>_
    OrderDate público Como DateTime
    <Column()>_
        Público Destinatario As String
End Class
```

4. Generar una manera de obtener datos de la base de datos utilizando un DataContext objeto, como se muestra aquí:

```
Dim currentContext Como DataContext = Nuevo _
DataContext ("server = (local); database = Northwind; Trusted_Connection = True")
MyOrders As (Tabla de órdenes) currentContext.GetTable = (de las órdenes) ()
```

5. Ahora usted tiene una colección extraída de una base de datos.

Si quieres recorrer ellos, se puede poner una orden en For Each latestOrders, y los procesa de esa manera.

Como se puede ver, LINQ es otra forma de tratamiento de datos. Si usted está construyendo sistemas de gran escala, LINQ tiene mucho sentido. Por pequeña aplicación de Windows ciones, LINQ no es mucho mejor que ADO.NET.

Para obtener más información, echa un vistazo a LINQ to SQL: .NET Language Integrated Para datos relacionales, en <http://msdn2.microsoft.com/en-us/library/bb425822.aspx>. El código VB en el artículo tiene algunos errores, pero los principios discutidos son sólidos.

Capítulo 16

Trabajar con el sistema de archivos

En este capítulo

Entender las clases y controles en System.IO

Uso de los controles de gestión de archivos System.IO

Abrir, guardar, listado y ver archivos en sus aplicaciones

Storing información en archivos es una de las tareas más comunes de un ordenador programa, y Visual Basic hace que sea más fácil de realizar esas tareas en el Versión 2008. Visual Basic no se conoce como un fuerte manejo de archivos de lenguaje. Con la adición del Marco .NET, sin embargo, un número sorprendente de archivos manejo de herramientas están disponibles para su uso.

En este capítulo, le mostraré cómo trabajar con los directorios y archivos en el computadora. El espacio de nombres System.IO y sus clases, junto con el My. Clase Computer.FileInfo, contienen las herramientas que necesita para leer directorios, analizar los archivos, guardar información en archivos, obtener información del archivo y más.

También se describen los controles que Visual Studio proporciona para la manipulación de archivos en aplicaciones Windows Forms. El OpenFileDialog, SaveFileDialog, FolderBrowserDialog y acelerar el desarrollo de programas que gestionan archivos. El componente FileSystemWatcher hace que sea más fácil de mantener comunicación entre una aplicación y sus archivos.

La clave para trabajar con archivos en formato .NET se está familiarizado con ellos. Aunque requiere un poco de esfuerzo para diseñar un formato de archivo o de averiguar una ya existente formato, le recomiendo que utilice los archivos cuando usted necesita los archivos. Los controles en Visual Basic 2008 hacen uso de archivos mucho más sencillo que nunca antes.

Conozca System.IO

System.IO tiene dos categorías de clases y un conjunto de componentes que se quiere familiarizarse con: las clases de flujos, el archivo y el directorio clases, y los controles de diálogo.

Clases Stream permiten manejar el contenido de archivos como una secuencia de personajes. Tabla 16-1 describe algunas de las clases de flujos comunes disponibles capaz de Visual Basic.

Tabla 16-1 Las clases System.IO Stream en	
Clases	Descripción
BinaryReader, BinaryWriter	Se utiliza para leer y escribir archivos no son de texto, como imágenes, en una corriente
FileStream	Se puede usar para hacer cualquier archivo en una corriente
TextReader, TextWriter	En concreto se utiliza para leer y escribir texto en los arroyos

Streams son herramientas complicadas, y yo no los cubren mucho aquí. En su mayoría son utilizado para el movimiento de la información en los diversos estados de conexión, y si bien esto es importante, está más allá del alcance de este libro. I lugar de centrarse en la archivos y directorios herramientas en System.IO, que son mucho más propensos a necesitas sobre una base diaria.

Las clases de mantenimiento de archivos y directorios están parcialmente compartida, lo que significa que

no es necesario que obtenga una copia de usarlos porque están siempre disponibles.

También puede utilizar las implementaciones de instancia, que aceptan la ruta de acceso a la archivo o directorio en cuestión como un parámetro. Las clases de gestión de archivos son muestran en la Tabla 16-2.

Tabla 16-2 Las clases de administración de archivos en System.IO	
Clases	Descripción
Directorio, DirectoryInfo	Compartida y herramientas de instancia para el mantenimiento de los directorios
DriveInfo	
File, FileInfo	Ayuda con el mantenimiento de una unidad de disco (tal como una unidad de disco duro o una unidad de disco virtual)
FileSystem Watcher	Compartida y herramientas de instancia para el mantenimiento de archivos
Camino	Una clase muy cool que mantiene un ojo en el sistema de archivos y provoca eventos cuando suceden cosas específicas
	Ayuda a mantener las rutas UNC a los archivos y así sucesivamente



Las clases de gestión de archivos en System.IO sustituir a la antigua FileSystemObject en VB 6 y VBScript.

Las clases que terminan en los datos fueron clases de instancia, lo que significa que usted necesita para dimensión que antes de utilizarlos y darles un camino para comenzar con. Esto es útil cuando usted está haciendo una gran cantidad de operaciones en un solo archivo o directorio historia. Las clases sin información al final son compartidos, lo que significa que usted puede utilizar siempre que lo desee, como una caja de herramientas digital. Las clases son compartidas ideal para un cambio rápido a un archivo o directorio.

Usted consigue la mayoría del uso de las clases File y Directory. Por ejemplo, ambas clases de apoyo un método Exists, que acepta una ruta de acceso y devuelve una Valor booleano que indica si el archivo o directorio existe - muy simple, útil, y es algo que no puedes vivir sin él.

La tercera categoría importante de las herramientas que proporcionan las clases System.IO no está realmente formado por clases - que se compone de los controles. Lo que solía ser llamado el control CommonDialog es ahora un conjunto de controles, entre los genéricamente categorizados como Diálogos.

Estos controles hacen mucho uso de las clases de archivos y directorios, y hacen que sea mucho más fácil para que usted pueda dar a los usuarios el control sobre los archivos que refieren a la aplicación. Usted ha visto estos controles de diálogo en otro programas (como los programas de Microsoft Office). Tabla 16-3 describe el diálogo controles disponibles en Visual Basic.

Tabla 16-3**Los controles de diálogo**

Clase	Descripción
FolderBrowser Diálogo	Muestra un cuadro de diálogo que permite al usuario navegar por y elegir un directorio
OpenFileDialog	Muestra un cuadro de diálogo abierto que permite al usuario seleccionar un archivo del sistema de archivos local
SaveFileDialog	Muestra un cuadro de diálogo Guardar que permite al usuario guardar una archivo en el sistema de archivos local

Utilizando el espacio de nombres System.IO

Algunas de las aplicaciones que escriben requerirá una gran cantidad de acceso a archivos. Algunas

aplica-
caciones necesita prácticamente ningún acceso a archivos. No importa qué, pudiendo rápidamente
configurar una aplicación para ir a un archivo, mover, copiar, leer, o borrar se trata de un
parte importante de la programación del día a día.

En las secciones siguientes se ofrece una serie de las tareas más comunes que su
programas tienen que realizar con los archivos y describir cómo System.IO y la
herramientas relacionadas en el lenguaje VB hacer que los archivos de fácil manejo.

Cómo guardar los archivos en un servidor Web

Los tipos que se describen en la Tabla 16-3 se Controles de Windows Forms. Las aplicaciones Web que

Necesitamos trabajar con archivos están sujetos a los caprichos del navegador y está atrapado usando HTTP Cargar.

Subir HTTP le permite acceder a algunos controles, a saber, el control FileUpload. La FileUpload de control permite al usuario seleccionar un archivo en su sistema de archivos local para subir a

el servidor. Recuerde, la Web se desconecta conectados, por lo que los archivos que el

usuario ve en la navegador se encuentran en su máquina.

Después de que el archivo se carga en el

servidor, la

Clases File y Directory se utilizan para

guardar el archivo en el servidor. Por ejemplo, el siguiente código podría ser usado para manejar un archivo cargado con el control FileUpload:

```
myPath = "C:\Archivos\"  
fileName As String =  
    FileUpload1.FileName  
myPath += fileName  
FileUpload1.SaveAs(myPath)
```

Detrás de las escenas, este código utiliza una corriente Escritor para guardar el archivo, que es parte de la System.IO clase. Esto se ve mucho, y son a salvo de lo mucho, a lo largo de la RED. Marco.



Todos los ejemplos siguientes - en su caso - Use un archivo de texto en el C:\ directorio llamado inputFile.txt y escribir en un archivo de texto llamado outputFile.txt.

Abrir un archivo

Hay varias formas de abrir un archivo en Visual Basic, pero sólo hay una buena manera que el usuario seleccione un archivo para abrirlo - el control OpenFileDialog. El control es un control OpenFileDialog bandeja de componentes - que no va derecho en un formulario, pero tú lo llamas desde otro objeto firme, como un botón.

Inicie un nuevo proyecto de aplicación de Windows, seleccione Archivo → Proyecto Nuevo. Nombre su nuevo Windows algo aplicación adecuada; he usado el nombre del archivo De gestión de este ejemplo. Cuando usted tiene una nueva aplicación para Windows proyecto listo, siga estos pasos para utilizar el control OpenFileDialog para permitir al usuario abrir un archivo:

1. Arrastre un OpenFileDialog control desde el Cuadro de herramientas al formulario.

El componente OpenFileDialog está en la porción de la Diálogos Caja de herramientas. Aparece en la bandeja de componentes.

2. Arrastre un Etiqueta de control de la caja de herramientas hasta el formulario. Cambie el

Texto valor en blanco, y el nombre de FileName. Ajuste el AutoSize a Falso, y cambiar el BackColor ActiveCaption.

3. Arrastre un Control TextBox al formulario. Cambiar el Multiline propiedad a Verdadero. Cambie el nombre a FileContents.

4. Arrastre un nuevo botón al formulario y establezca la Texto valor para abrir un File. Asigne un nombre al OpenFileDialog botón.

El entorno debe parecerse a la Figura 16-1.

5. Haga doble clic en el botón para entrar en el editor de código. Visual Studio entonces

hace que el OpenFileDialog_Click controlador de eventos.

6. Agregue el código siguiente en OpenFileDialog_Click para obtener el contenido del **el archivo en un** Variable String:

```
Imports System.IO
Public Class Form1
    "Esto es para los contenidos del archivo.
    MyFileContents Dim As String
    Private Sub OpenFileDialog_Click (ByVal como System.Object remitente e, _ ByVal como
        System.EventArgs) Handles OpenFileDialog.Click
        'Abre el cuadro de diálogo y asegúrese de que se ha realizado correctamente
        Si OpenFileDialog1.ShowDialog () = DialogResult.OK Entonces
            'Abre un objeto StreamReader con el nombre de archivo en el cuadro de diálogo
            MyStreamReader As New _
            StreamReader (OpenFileDialog1.FileName)
            "Leer el archivo con el StreamReader
            MyFileContents Dim As String = myStreamReader.ReadToEnd ()
            'Cierra el StreamReader - que utiliza los recursos
            myStreamReader.Close ()
            'Establecer los campos de salida
            FileName.Text = OpenFileDialog1.FileName.ToString
            FileContents.Text = myFileContents
        End If
    End Sub
End Class
```

7. Ejecute la aplicación y haga clic en el botón.

Si todo está bien, cuando se hace clic en el botón, usted será capaz de seleccionar un archivo

y ver su contenido. Si coloca el archivo inputFile.txt en el directorio C:\ directorio, se pueden ver los resultados en la Figura 16-2.



Usted puede elegir entre una gran cantidad de opciones para el control OpenFileDialog. Usted puede establecer el tipo de archivo predeterminado que se va a abrir, el título del cuadro de diálogo, y el directorio de inicio, sólo para empezar. Echa un vistazo a la ventana Propiedades para el control de lo que quiero decir.

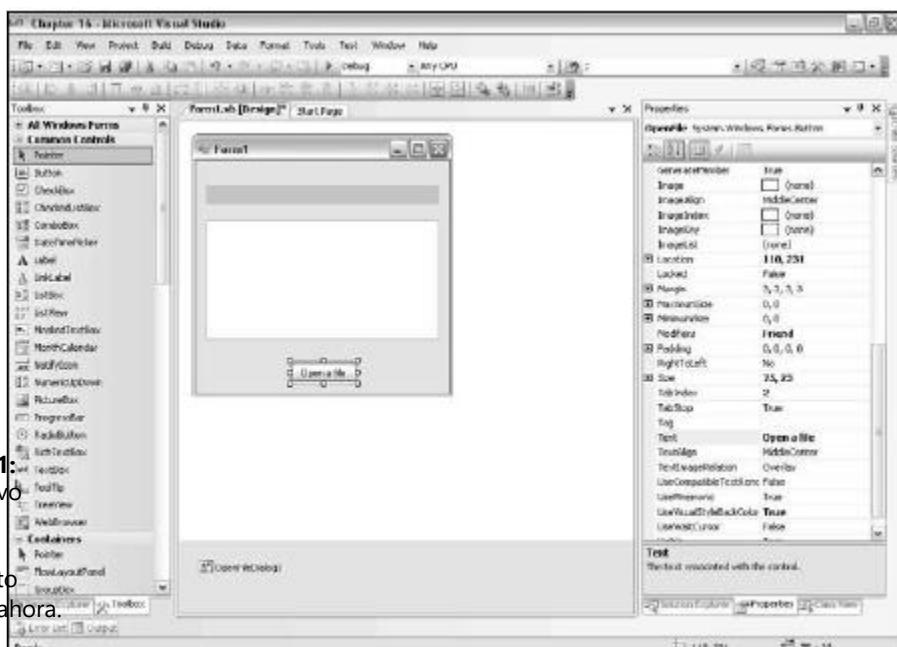


Figura 16-1:
El Archivo
Manage-
ción
proyecto
hasta ahora.

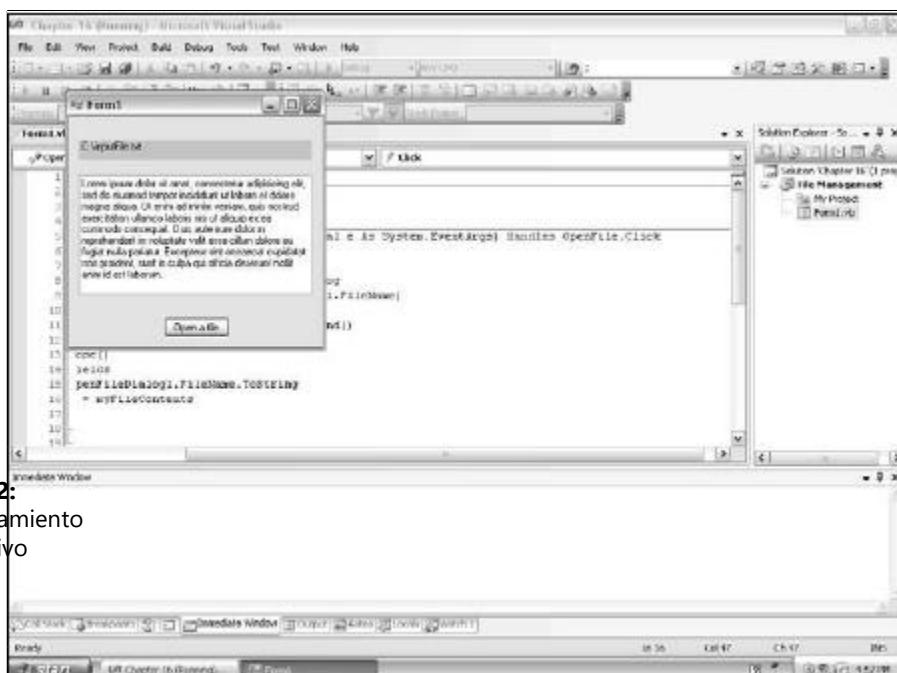


Figura 16-2:
Funcionamiento
el Archivo
Manage-
ción
aplicación.

Cambiar el contenido de un archivo

Si ha seguido los pasos de la sección anterior, usted tiene una cadena con un el contenido del archivo, y usted necesita conseguir algo más en el archivo. Se utiliza el StreamReader para obtener la información de un archivo, y no es sorprendente que, utilizar el StreamWriter para obtener la información de nuevo en el archivo.



Aunque los ejemplos de este capítulo utilizan archivos de texto, debe ser consciente de dos advertencias. En primer lugar, cualquier archivo con formato de texto pueden ser manejados de esta manera.

(Abra el archivo en el Bloc de notas primero para asegurarse de que efectivamente tienen un texto

archivo.) En segundo lugar, los archivos binarios, como imágenes se pueden manejar con corrientes, también,

pero la salida será algo más que una cadena (un mapa de bits, por ejemplo).

Para este ejemplo, el contenido del archivo están en una cadena llamada myFile Contenidos. Para obtener el nombre del archivo, se utiliza el SaveFileDialog, que está discute en la siguiente sección. Para utilizar el mismo nombre de archivo, puede seguir estos pasos:

- 1. Agregue un botón al formulario y asígnele el nombre SaveFile.**
- 2. En el código, hacer una nueva StreamWriter y darle el nombre de archivo y contenido:**

```
Private Sub SaveFile_Click (ByVal como System.Object remitente e, ByVal como  
System.EventArgs) Handles SaveFile.Click  
    "Usted puede obtener el nombre de archivo de la etiqueta  
    MyStreamWriter As StreamWriter StreamWriter = Nuevo (FileName.Text)  
    'A continuación, utilice el cuadro de texto para obtener el contenido.  
    myStreamWriter.Write (FileContents.Text)  
    myStreamWriter.Close ()  
End Sub
```

- 3. Ejecute la aplicación. Cambiar el contenido en el cuadro de texto y haga clic en el**
- Botón Guardar archivo para guardar el contenido.**

Guardar un archivo

Si desea que el usuario pueda guardar el archivo con un nombre diferente, Puede utilizar el SaveFileDialog. Esto es igual que el OpenFileDialog, excepto que permite al usuario realizar un nuevo nombre de archivo y guardar el archivo renombrado en un directorio diferente.

Para usarlo, sólo tienes que arrastrar una SaveFileDialog al formulario y cambie el código en la subrutina SaveFile_Click a lo siguiente:

```
Private Sub SaveFile_Click (ByVal como System.Object remitente e, ByVal como
                           System.EventArgs) Handles SaveFile.Click
    Si SaveFileDialog1.ShowDialog () = DialogResult.OK Entonces
        MyStreamWriter As StreamWriter Nuevo (SaveFileDialog1.FileName)
        Si no (myStreamWriter Is Nothing) Entonces
            'A continuación, utilice el cuadro de texto para obtener el contenido.
            myStreamWriter.Write (FileContents.Text)
            'Cerrar el StreamWriter - que utiliza los recursos
            myStreamWriter.Close ()
        End If
    End If
End Sub
```

Al ejecutar la aplicación, usted tiene la oportunidad de dar un nombre al archivo nuevo. Tenga en cuenta que la SaveFileDialog, como el OpenFileDialog, tiene mucha deliciosas opciones para definir lo que el usuario puede y no puede hacer. Puedes ver algunas de ellos en la Figura 16-3.

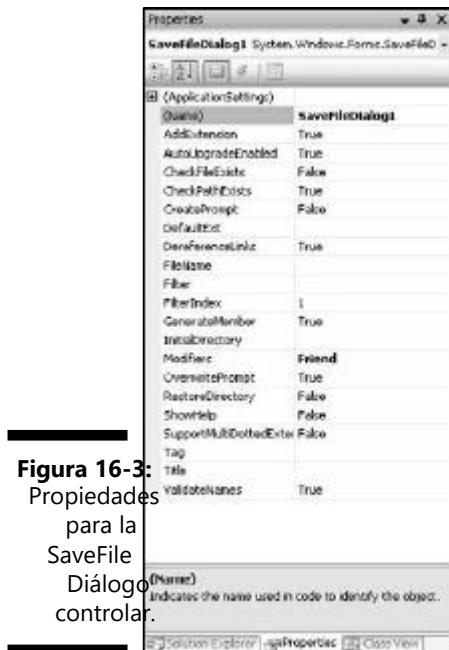


Figura 16-3:
Propiedades
para la
SaveFile
Diálogo
controlar.

Listado de los directorios y archivos

Directorios y archivos se encuentran en colecciones como parte de la System.IO.File y Objetos System.IO.Directory. Cuando se llama a uno de estos métodos, obtener una matriz de cadenas que son los subdirectorios del directorio especificado.

Así, por ejemplo, si se escribe un código como éste...

```
Dim As String subdirectorio  
En cada subdirectorio En Directory.GetDirectories ("C:\")  
    Console.WriteLine (subdirectorio)  
Próximo
```

... usted consigue algo como esto:

```
C:\Documents and Settings  
C:\Archivos de programa  
C:\Windows
```

Lo mismo funciona para los archivos. Nótese, sin embargo, que no tenía por qué dimensionar una copia de la clase Directory - Acabo de usarlo. Esto se debe a la Clase Directory se comparte, lo que le permite utilizarlo sin instanciar ella. Esto reduce el desorden en el código - y el uso de la memoria también.

Por otra parte, las clases compartidas en System.IO comprobar su seguridad acceso cada vez que se les llama, mientras que las clases instanciadas marque sólo una vez, cuando se crea una instancia. Si usted va a utilizar las clases de repetidamente, se debe utilizar la clase DirectoryInfo FileInfo y que se describen en la siguiente sección.

Visualización de la información del archivo

Si, por ejemplo, se necesita más información sobre un directorio que el otro directorios en el mismo, es posible que desee utilizar la DirectoryInfo (o FileInfo, para los archivos) de clase. Esta clase es similar a la clase relacionada sin la etiqueta de información, pero se requiere una nueva declaración, como se muestra en el siguiente código:

```
Dim As String subdirectorio  
En cada subdirectorio En Directory.GetDirectories ("C:\")  
    Console.WriteLine (subdirectorio)  
    Midirectorio = New DirectoryInfo (subdirectorio)  
    Console.WriteLine (myDirectory.CreationTime)  
    Console.WriteLine (myDirectory.LastAccessTime)  
    Console.WriteLine (myDirectory.Parent)  
    midirectorio = Nothing  
Próximo
```

El propósito de utilizar DirectoryInfo en la línea 4 es para aclarar el uso del archivo sistema y crear un objeto, aunque sea brevemente, que tiene un directorio base. Si Vamos a utilizar el objeto de directorio (aquí, midirectorio) más de una vez, vale la pena utilizar las clases y DirectoryInfo FileInfo.

Mantener un ojo en los archivos

FileSystemWatcher es una gran herramienta poco que mantiene un ojo en los archivos para usted. Si algo sucede, por ejemplo un cambio de nombre o contenido de archivos, que le permite saber lanzando un evento que se puede tomar con el código.

Para ver cómo se puede utilizar FileSystemWatcher, puede agregar una característica poco el editor de archivos a construir en los apartados anteriores. La adición de un archivo SystemWatcher le informa cuando algo sobre el cambio de archivos. Seguir estos pasos para agregar un FileSystemWatcher:

1. Agregar una FileSystemWatcher para la aplicación de gestión de archivos.

2. Agregar una Etiqueta bajo la etiqueta que agregó previamente.

Puede que tenga que ampliar el formulario.

3. Cambie el BorderStyle de la etiqueta para Fixed3D, y establecer el AutoSize valor a false.

4. Estire la Etiquetar a un tamaño apropiado. Establezca el valor TextAlign a MiddleCenter.

Su aplicación debería parecerse a la Figura 16-4.

5. Agregue las siguientes dos líneas de código para el OpenFileDialog evento handler:

```
FileSystemWatcher1.Path = "C:\\"  
FileSystemWatcher1.EnableRaisingEvents = True
```

6. Haga clic en el objeto en la lista desplegable en la vista Código y seleccione el FileSystemWatcher, a continuación, en el evento en la lista desplegable, seleccione el Cambiado evento.

Visual Studio crea una subrutina de control de eventos.

7. Agregue la siguiente línea en negrita del código que escribe los detalles del archivo cambiar a la Etiqueta de objeto en el FileSystemWatcher1_Changed controlador que se ha generado para usted:

```
Private Sub FileSystemWatcher1_Changed (ByVal sender As  
Object, ByVal e As  
System.IO.FileSystemEventArgs) Maneja  
FileSystemWatcher1.Changed  
Label1.Text = String.Format ("{0} ha sido escrito para a  
{1}", e.Name, DateTime.Now.ToString)  
End Sub
```

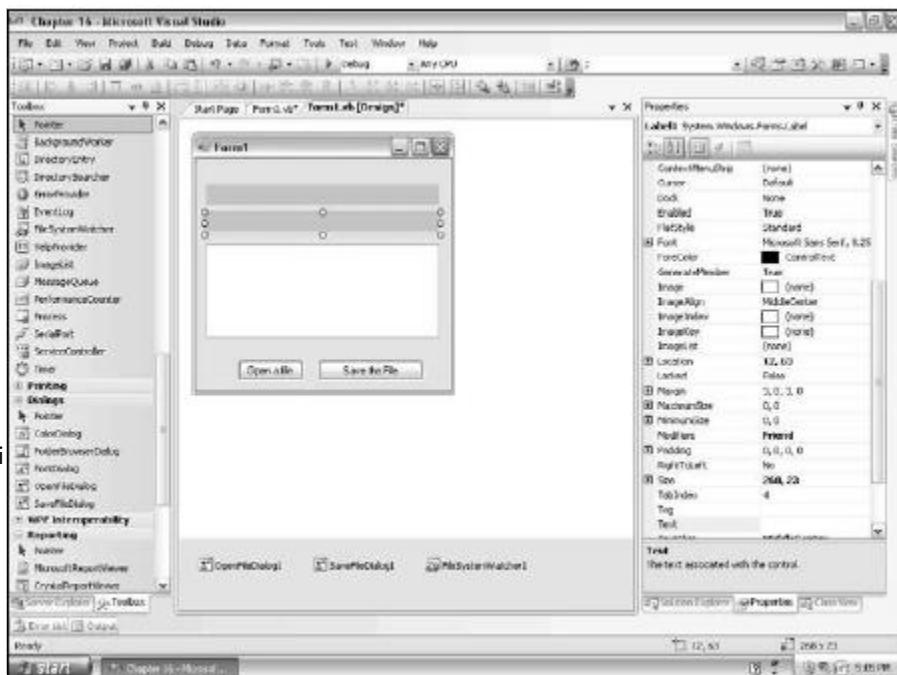


Figura 16-4:
Adición de la
Expediente
Sistema
Watcher
para el
archivo
Manage-
ación
aplicación.

Los argumentos del evento proporcionar el nombre y otros detalles. Otros eventos, Renombrado como, incluso proporcionan el OldName. Esto sería excepcionalmente útil para el registro de acceso a archivos o cambios en los archivos del sistema. Es un potente objetos para la gestión del sistema.

Capítulo 17

El acceso a Internet

En este capítulo

Hacer un recorrido del espacio de nombres System.Net

Utilizando las herramientas integradas para acceder a la red

Haciendo las herramientas de la red para usted

Y mi honesta opinión, la razón por la que Microsoft tuvo que crear la RED. Marco en el primer lugar fue la falta de interoperabilidad de Internet dentro de la infraestructura existente. Visual Basic 6 no podía manejar la Internet. El Internet funciona de forma diferente que la mayoría de plataformas, como PC. La Internet se basa en protocolos - cuidadosamente definidas y acordadas formas para hacer las cosas como las transferencias de correo y de archivos de trabajo. Entorno de Microsoft antes de 2002 claramente no manejaba esos también.

Como se puede ver a lo largo de este libro, el Marco. NET se ha diseñado a partir de la base para tomar la Internet y las redes en general en consideración. No es sorprendente que ninguna parte es más evidente que en el sistema. Espacio de nombres de red. El Internet tiene la primera silla aquí, con herramientas Web ocupar nueve de las clases en el espacio de nombres.

En esta segunda versión de la estructura, aún más funcionalidad a Internet al horno pulg Si bien en la versión 1, la atención se centró en las herramientas utilizadas para construir otras herramientas (Funciones de bajo nivel), que ahora contiene características que son útiles para usted, como Web, correo y FTP. Secure Sockets Layer - Seguridad de Internet de transporte - es mucho más fácil de utilizar en esta versión, como es FTP y electrónico, que previamente otros requieren más difícil de usar clases.

Un espacio de nombres System.Net grande, carnoso, y encontrar su camino en lo posible ser difícil. Mi objetivo en este capítulo es tomar las cosas que se hacen con frecuencia y mostrar los fundamentos, y luego le dará las herramientas para investigar el más complejo características de las clases.

Networking es una gran parte del Marco .NET, y toda la funcionalidad está en este espacio de nombres - un libro entero puede ser (y ha sido) escrito sobre el tema. A los efectos de esta introducción a la creación de redes con VB, te muestro las siguientes características:

- Obtención de un archivo de la red
- Envío de e-mail
- Registro de las transferencias
- Comprobación en el estado de la red en torno a la aplicación en ejecución

Tenga en cuenta que no estoy negando la importancia de las tomas, IPv6, y otros protocolos de Internet avanzadas. Sólo estoy sugiriendo que las partes de la espacio de nombres que va a usar todos los días. Como siempre, hay más conocer System.Net.

Conozca System.Net

El espacio de nombres System.Net está llena de clases que son muy confusas si visto en la documentación, pero hacen mucho sentido cuando se usa en una aplicación. El espacio de nombres elimina toda la complejidad de tratar con la variabilidad protocolos de las unidades organizativas utilizadas en Internet.

Más de 2.000 RFCs existen para protocolos de Internet (RFC es una solicitud de Comentarios, documento que se envía a un cuerpo de normas que se revisará compañeros antes de que sea una norma), y si usted tiene que aprender todo de ellos separadamente, usted nunca conseguirá su proyecto terminado. El espacio de nombres System.Net es trata de hacer que sea menos dolorosa.

System.Net no es sólo para proyectos Web. Al igual que con todo lo demás en la base biblioteca de clases, puede utilizar System.Net con todo tipo de proyectos. Usted puede hacer el siguiente:

- Obtener información de páginas Web en Internet y lo utilizan en su programas
- Mover archivos a través de Internet usando FTP (File Transfer Protocol)
- Enviar por e-mail fácilmente
- Usar estructuras de red más avanzadas
- Las comunicaciones seguras a través de Internet utilizando el protocolo SSL

Si necesita verificar la conectividad de una computadora desde un Windows aplicación, puede utilizar System.Net. Si usted necesita para construir una clase que por voluntad cargar un archivo desde un sitio Web, es el espacio de nombres System.Net usted necesita. Sólo

Capítulo 17: El acceso a Internet

porque la mayoría de las clases se refieren a la Internet no significa que sólo en la web aplicaciones pueden utilizar. Esa es la magia de System.Net. Cualquier aplicación puede ser una aplicación conectada. Mientras que algunas partes de la función de espacio de nombres de hacer que el desarrollo de aplicaciones web más fácil, el espacio de nombres en general está diseñado para que cualquier aplicación funcione con la Web.

Cómo los Cursos por Internet Fit en el Marco

El espacio de nombres System.Net contiene 62 clases y seis espacios de nombres más pequeñas.

Mientras escribo esto, estoy abrumado. Sin embargo, si se mira de cerca, se puede ver patrones.

Si usted necesita ayuda con las clases, usted puede encontrar más información en los capítulos 1 y 3.

Las clases son muy bien su nombre, y se dará cuenta de que algunos protocolos obtienen una número de clases cada uno. Luego de traducir, se puede reducir lo que usted necesita en función de la forma en que se nombró el protocolo:

Autenticación y autorización: Estas clases proporcionan seguridad.

Cookie: Esta clase administra las cookies de los navegadores web y por lo general utilizado en las páginas ASP.NET.

DNS (Servicio de Nombres de Dominio): Estas clases ayudan a resolver dominio nombres en direcciones IP.

Descargar: Esta clase se utiliza para obtener los archivos de los servidores.

EndPoint: Esta clase ayuda a definir un nodo de red.

FileWeb: Este conjunto brillante de clases describe los servidores de archivos de red como clases locales.

FTPWeb: esta clase es una simple transferencia de archivos aplicación del Protocolo.

HTTP (HyperText Transfer Protocol): Esta clase es el protocolo Web.

IP (Protocolo de Internet): Esta clase ayuda a definir los puntos finales de red que están específicamente relacionados con Internet.

IrDA: Esta clase es un extremo infrarrojo. Los puertos de infrarrojos son redes, también!

NetworkCredential: esta clase es otra implementación de seguridad.

as conexiones de red.

Socket: Esta clase trata de la más primitiva de las conexiones de red.



Upload: Este conjunto de clases le ayuda a subir la información a la Internet.

Web: Estas clases ayudan con la World Wide Web - en gran parte implementaciones de las clases http que son más orientado a las tareas.

Esta lista es tan extensa porque las clases se complementan entre sí. The End Clases de punto son utilizados por las clases socket para definir de red determinada específicos, y las clases de IP que sean específicos de la Internet. La Web clases son específicos de la World Wide Web. Rara vez se utiliza más alto clases de nivel, pero a menudo es difícil ver lo que se necesita cuando.

La mayoría de las funciones que se utilizan todos los días, sin embargo, están encapsulados en su mayoría dentro de los siete nuevos espacios de nombres en el espacio de nombres System.Net:

Cache: Esta función tiene una gran cantidad de encuestadores que gestionan el navegador y almacenamiento en caché de la red funciones integradas en el espacio de nombres.

Configuración: Esta función permite el acceso a las propiedades que se debe configurar para que muchos de los otros trabajos System.Net clases.

Mail: Esta función toma el relevo para System.Web.Mail para facilitar la el envío de correo electrónico de Internet.

Mime: Esta función de archivo de paquetes de archivos adjuntos con el espacio de nombres de correo.

NetworkInformation: Esta función obtiene información acerca de la red en torno a su aplicación.

Seguridad: Esta función implementa la seguridad de la red gestionada por muchas clases de System.Net.

Sockets: Esta función utiliza la más básica de las conexiones de red disponible para Windows.

Utilizando el espacio de nombres System.Net

El espacio de nombres System.Net es muy orientado a código, lo que significa que muy pocos implementaciones son específicamente para las interfaces de usuario. Casi todo lo que usted hacer con estas clases está detrás de las escenas. Tiene pocos arrastrar y soltar usuario controles - el espacio de nombres System.Net se utiliza en la vista Código.

Para demostrar esto, en el resto de este capítulo, voy sobre la construcción de un equipo con Windows

Formas de aplicación que tiene los siguientes requisitos:

Compruebe el estado de la red.

Obtener un archivo específico desde Internet.

Enviarlo por correo electrónico a una determinada dirección de correo electrónico.

Registrar la transacción completa.

Este no es un conjunto de requisitos insignificante. De hecho, incluso en el 1,0 y 1,1 versiones de VB.NET, esto sería muy difícil. Uno de los principales objetivos de la Espacio de nombres System.Net en esta versión es hacer este tipo de tareas - tareas muy comunes - mucho más fácil. Puede empezar por la carga de la ejemplo de código o iniciando un nuevo proyecto y siguiendo los pasos de los siguientes secciones.

Comprobación del estado de la red

En primer lugar, es necesario informar al usuario sobre la conectividad de red siguiendo estos pasos:

1. Crear un nuevo proyecto de aplicación para Windows en Visual Studio.

Llamé NetworkTools minas.

2. Haga referencia a la Espacio de nombres System.Net sumando las importaciones de línea

System.NET a la parte superior del código.

3. Agregar una StatusStrip control al formulario arrastrándolo desde la **Caja de herramientas**.

4. Seleccione la SmartTag que aparece y añadir un ToolStripLabel.

5. De vuelta en la vista Diseño, haga doble clic en el formulario para obtener el Form_Load

control de eventos y se mueven a la vista Código.

6. Agregue el código en negrita de la lista siguiente para ver si el red está disponible y lo mostrará en la barra de estado:

```
Imports System.Net  
Public Class Form1  
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs)  
        Handles MyBase.Load  
        If NetworkInformation.NetworkInterface.GetIsNetworkAvailable Then  
            ToolStripStatusLabel1.Text = "Conectado"  
        Else  
            ToolStripStatusLabel1.Text = "Desconectado"  
        End If  
    End Sub  
End Class
```



Eso es todo lo que hay que hacer. La clase NetworkInformation contiene un montón de información sobre el estado de la red, las direcciones IP actuales, la puerta-manera de ser utilizado por el equipo actual, y más.

Tenga en cuenta que la clase NetworkInformation sólo funcionará en un local máquina. Si utiliza esta clase en una aplicación Web ASP.NET Forms, usted a obtener información sobre el servidor.

Descargar un archivo de Internet

Así, el próximo, es necesario obtener un archivo desde Internet. Esto se puede lograr una número de maneras, pero una de las más comunes es el uso de FTP. FTP es una luz-peso protocolo que se ve favorecida porque es seguro y compatible con muchos sistemas.

Para crear una aplicación que utilice FTP, siga estos pasos:

- 1. Arrastre un botón al formulario desde el Cuadro de herramientas.**
- 2. Haga doble clic en el botón para obtener la** Haga clic en controlador de eventos.
- 3. Agregue las importaciones requeridas** - System.Net, System.Net.Mail, y System.IO - a la parte superior del código.
- 4. Crear una nueva subrutina llamada** Descargar archivo que acepta un mando a distancia **nombre de archivo y un nombre de archivo local como cadenas.**
- 5. En la subrutina nueva dimensión, una nueva** FileStream (llamadas locales FileStream) y FtpWebRequest (llamado miRequest), como se muestra en **Listado 17-1.**

Las referencias FileStream un archivo local y acepta el archivo local que es transfiere a la subrutina. El FtpWebRequest es lo mismo para el archivo remoto.
- 6. Ajuste el** Método parámetro de la FtpWebRequest a WebRequestMethods.Ftp.Downloadfile.
- 7. Ajuste el** Credenciales característica de la FtpWebRequest para un nuevo NetworkCredential con información anónima, como lo hice en **Listado 17-1.**
- 8. Crear un nuevo** WebResponse objeto del método miRequest.

Esto nos lleva la cuenta de nuevo desde el servidor FTP en cuanto a cómo su se atenderán las peticiones.
- 9. Obtener el** Stream desde el objeto respuesta.
- 10. Lea el archivo en un búfer de 1024 bytes, un bloque a la vez, utilizando un Mientras bucle, como se muestra en el extremo del listado 17-1.**

Listado 17-1: El Método DownloadFile

```
Protegido DownloadFile (ByVal remotefile As String, _
ArchivoLocal ByVal como secuencia)
```

```
LocalFileStream As New FileStream (archivoLocal, FileMode.OpenOrCreate)
MiRequest dévil como FtpWebRequest = WebRequest.Create (remotefile)
myRequest.Method = WebRequestMethods.Ftp.DownloadFile
```

```
myRequest.Credentials = Nuevo NetworkCredential ("Anónimo", _  
    "Bill@sempf.net")  
Dim myResponse ComoWebResponse = myRequest.GetResponse  
MyResponseStream dévil como secuencia = myResponse.GetResponseStream  
Tampón Dim (1024) As Byte  
BytesRead As Integer = myResponseStream.Read (buffer, 0, 1024)  
Mientras BytesRead > 0  
    localFileStream.Write (buffer, 0, BytesRead)  
    BytesRead = myResponseStream.Read (buffer, 0, 1024)  
End While  
localFileStream.Close ()  
myResponseStream.Close ()  
  
End Sub
```

11. Llame a la DownloadFile método desde el evento Button1_Click controlador, como muestra en el siguiente código:

```
Private Sub Button1_Click (ByVal como System.Object remitente, _  
    ByVal e As System.EventArgs) Handles Button1.Click  
    DownloadFile ("ftp://ftp.vbfodummies.com/sampleFile.bmp", _  
        "C:\ sampleFile.bmp")  
End Sub
```

Este es un ejemplo FTP muy aguada, pero se pone el punto a través. La WebRequest yWebResponse clases del espacio de nombres System.Net son plenamente utilizado para crear la FtpWebRequest más completa, por ejemplo. Propiedades tales como el método de descarga y Credenciales lo convierten en un decisión fácil.

De hecho, la parte más difícil de este proceso se trata de un objeto FileStream, que sigue siendo la mejor manera de mover los archivos y no es específico para el System.Net espacio de nombres. Las secuencias se discute en el capítulo 16, que abarca el sistema. IO espacio de nombres, pero no tienen importancia para las clases de red, también. Streams representan un flujo de datos de algún tipo, y un flujo de información desde el Internet cumple los requisitos.

Eso es lo que está haciendo cuando usted consigue una página Web o un archivo de Internet - recoger un flujo de datos. Si usted toma un segundo para pensar en ello, tiene sentido que se trata de un flujo, debido a que la barra de estado en una aplicación muestra un porcentaje edad en que cesa. Es como echar agua en un vaso, el flujo de datos es un corriente, por lo que el concepto se llama Stream.

Este concepto es válido para obtener un archivo de la World Wide Web, también. HTTP, el protocolo de la Web, es sólo otro protocolo que define cómo un documento se traslada de un servidor en Internet en el equipo local. En hecho, el código aún se ve muy similar a la del ejemplo FTP, como se puede ver en el siguiente ejemplo. La misma corriente se recupera, sólo el formato es diferente.

```

    Protegido DownloadWebFile (ByVal remotefile As String, _
    ArchivoLocal ByVal como secuencia)
        LocalFileStream As New FileStream (archivoLocal, FileMode.OpenOrCreate)
        MiRequest dévil como WebRequest = WebRequest.Create (remotefile)
        myRequest.Method = WebRequestMethods.Http.Get
        Dim myResponse ComoWebResponse = myRequest.GetResponse
        MyResponseStream dévil como secuencia = myResponse.GetResponseStream
        Tampón Dim (1024) As Byte
        BytesRead As Integer = myResponseStream.Read (buffer, 0, 1024)
        Mientras BytesRead > 0
            localFileStream.Write (buffer, 0, BytesRead)
            BytesRead = myResponseStream.Read (buffer, 0, 1024)
        End While
        localFileStream.Close ()
        myResponseStream.Close ()
End Sub

```



Usted tendrá que pasar en una dirección Web, para que su llamada de subprograma se vería de esta manera:

```
DownloadWebFile ("http://www.vbfordummies.com/sampleFile.bmp",
    "C:\sampleFile.bmp")
```

Tenga en cuenta los cambios, que están marcados en negrita. miRequest es ahora una WebSolicitar más que un FtpWebRequest. Además, la propiedad Method de miRequest se ha cambiado a WebRequestMethods.Http.Get. Finalmente, la propiedad Credentials se ha eliminado debido a que las credenciales son ya no es necesario.

Envío por correo electrónico un informe de situación

El correo electrónico es un requisito común de sistemas en red. Si está trabajando en un entorno empresarial, que se va a escribir una aplicación a gran escala para manejar todos los requisitos de correo electrónico, en lugar de hacer que cada individuo aplicación ción de e-mail-aware.

Sin embargo, si usted está escribiendo un producto independiente, podría requerir de correo electrónico sup- puerto. Porque sucede que escribir una aplicación independiente, eso es exactamente lo que voy a hacer.

El correo electrónico es una operación basada en el servidor, así que si usted no tiene un servidor de correo electrónico que que puede utilizar para enviar desde, esto puede ser difícil. Muchos ISP ya no permitir la retransmisión, que es el envío de un mensaje de salida sin tener primero una cuenta y iniciar la sesión. Por lo tanto, es posible que tenga problemas al ejecutar esta parte de la muestra.

Si se encuentra en un entorno corporativo, sin embargo, por lo general, puede hablar con su administrador de correo electrónico y obtener el permiso para utilizar el servidor de correo electrónico. Porque solicitudes salientes en general sólo se aprovecha dentro del firewall, la retransmisión es a menudo disponibles. Para generar la función de correo electrónico, siga estos pasos:

1. Agregar un cuadro de texto al formulario predeterminado en Vista Diseño y, a continuación, cambie a

Ver código.

2. En la parte superior de la vista de código, asegúrese de que ha indicado el
System.Net.Mail espacio de nombres.

3. Crear una nueva subrutina llamada SendEmail.

Debe aceptar la dirección de correo electrónico, el e-mail a la dirección, el asunto del e-mail, y el cuerpo del correo electrónico.

4. Una nueva dimensión MailMessage y pase el FromAddress,
a dirección, parámetros sujetos, y el cuerpo, de la siguiente manera:

Mensaje débil como nuevo MailMessage (FromAddress, a dirección,
asunto, el cuerpo)

5. Una nueva dimensión SmtpClient, y pase la dirección de su correo
servidor.

Esta puede ser una dirección IP, nombre de la máquina, o URL.

6. Utilice el Enviar método del objeto SmtpClient ha creado para enviar
la MailMessage, que se pasa como un parámetro.

Cuando haya terminado, asegúrese de que ha establecido los valores del correo
Mensaje y SmtpClient para nada, ya que ocupan
recursos.

Listado 17-2 muestra la subrutina completado.

Listado 17-2: La subrutina SendEmail

```
Sub SendEmail (ByVal como secuencia FromAddress, a dirección ByVal As String, _  
           Sujeto ByVal como secuencia, cuerpo ByVal como secuencia)  
    Mensaje débil como nuevo MailMessage (FromAddress, a dirección, _  
                                         asunto, el cuerpo)  
    ClienteDeCorreo débil como nuevo SmtpClient ("localhost")  
    mailClient.Send (mensaje)  
    mensaje = Nothing  
    ClienteDeCorreo = Nothing  
End Sub
```

Tenga en cuenta que he usado localhost como nombre del servidor de correo electrónico. Si usted tiene correo electrónico software servidor instalado localmente, incluso sólo con SMTP IIS 6.0, esto funcionará. La mayoría de las veces, usted tendrá que poner otro nombre del servidor de correo electrónico en la SmtpClient constructor. El nombre del servidor de correo electrónico a menudo se pueden encontrar en su Preferencias de Outlook.

Después de haber escrito su método, usted tiene que llamar después de que el archivo está en el suelo- cargado en el controlador de eventos Button1_Click. Cambie el código de ese sub-rutina a lo siguiente para llamar a ese método:

```
Private Sub Button1_Click (ByVal como System.Object remitente, _
    ByVal e As System.EventArgs) Handles Button1.Click

    DownloadFile ("ftp://ftp.vbfordummies.com/sample
        File.bmp ", _
    "C: \ sampleFile.bmp")
    SendEmail (TextBox1.Text, TextBox1.Text, "FTP
        Exitoso ", _
    "FTP descargado correctamente")
End Sub
```

Observe que he enviado en el valor del cuadro de texto dos veces: una para la dirección a la y otra para la dirección del remitente. Esto no siempre es necesario, ya que puede tener una situación en la que desea que el correo electrónico a venir sólo de un Webmaster abordar o ir solamente a su dirección.

Usted debe tener suficiente código en lugar de ejecutar la aplicación ahora. Presione F5 para iniciar la aplicación en modo de depuración y darle una oportunidad.

Al hacer clic en el botón, la aplicación debe descargar el archivo en el unidad local y luego por correo electrónico para informarle de que la descarga se ha completado. Una gran cantidad de cosas pueden ir mal con las aplicaciones de red sin embargo, y usted debe ser consciente de ellos. Éstos son sólo algunos:

Para más actividad de red, el equipo que ejecuta el software debe ser conectado a una red. Esto no es un problema para usted, como desarrollador, pero hay que ser consciente de los usuarios finales, quienes pueden necesitar la conectividad dad de tener acceso a las funciones que deseé utilizar. El uso de la red código de estado puede ayudar a informar a los usuarios sobre la disponibilidad de estas características.

Los firewalls y otros dispositivos de red a veces bloquear el tráfico de red de aplicaciones legítimas. Algunos ejemplos de esto son:

- FTP es a menudo bloqueada de las redes corporativas.
- Funciones de análisis de la red de .NET son a menudo bloqueados en las empresas servidores. Si el servidor está disponible para el público, estas aberturas pueden hacer agujeros para los hackers para gatear por ellos.

Capítulo 17: El acceso a Internet

- Hablando de hackers, asegúrese de que si usted hace uso de red entrante trabajar características en su aplicación, han asegurado adecuadamente su aplicación. Más sobre esto se puede encontrar en el libro excelente *Writing Secure Code*, Segunda edición, por Michael Howard y David C. LeBlanc (publicado por Microsoft Press).
- El correo electrónico es especialmente frágil. A menudo, los proveedores de servicios de Internet se bloquear el correo electrónico desde una dirección que no está registrado en un servidor de correo. Esto significa que si usted está utilizando su servidor localhost (como en el ejemplo del Listado 17-2), el ISP podría bloquear el e-mail.

El tráfico de red es muy difícil de depurar. Por ejemplo, si la muestra funcionamiento de la aplicación, pero nunca recibe un e-mail de la SmtpServer usted codificadas, lo que salió mal? Es posible que nunca se sabe. Servicios Web XML (En el Capítulo 7) tienen un problema similar - es espectacularmente duro para ver el código real en el sobre SOAP para saber qué salió mal.

Registro de actividad de red

Esto me lleva al siguiente tema, que es el registro de la red. Debido a que la red problemas de actividad son tan difíciles de depurar y reproducirse, Microsoft ha construido en varias herramientas para la gestión de seguimiento de la actividad de la red.

Lo que es más, al igual que la disposición trazado ASP.NET, espacio de nombres System.Net seguimiento se gestiona completamente mediante el uso de los archivos de configuración. Esto significa usted no tiene que cambiar y volver a compilar el código cuando desee utilizar el funciones. De hecho, con una gestión poco, incluso se puede mostrar depurar información al usuario mediante la gestión de los archivos de configuración que utiliza su aplicación.

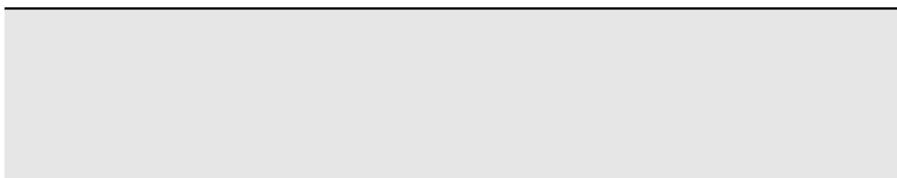
Cada tipo de aplicación tiene un tipo diferente de archivo de configuración. Para Windows

Aplicaciones de formularios, el que lo utiliza aquí, el archivo se llama app.config y se almacena en el directorio de proyecto de desarrollo. Cuando se compila, el nombre del archivo se cambia al nombre de la aplicación, y se copian en el directorio bin para correr.

Si abre el archivo app.config ahora, se ve que algunas diagnóstico información ya está en allí, como se muestra en el listado 17-3. Usted va a añadir un poco de información para que.

Listado 17-3: El archivo app.config defecto

```
<? Xml version = "1.0" encoding = "UTF-8">
<configuration>
```



Listado 17-3: (continuación)

```

<Nombre de origen = "Microsoft.VisualBasic.Logging.Log.WindowsFormsSource"
switchName = "DefaultSwitch">
<listeners>
<add name="FileLog"/>
<- Eliminar el comentario de la sección a continuación para escribir a la Aplicación
Registro de sucesos ->
<- <add Name="EventLog"/> ->
</Oyentes>
</Source>
</Sources>
<switches>
<add name="DefaultSwitch" value="Information" />
</Switches>
<sharedListeners>
<Add name = "FileLog"
      type = "Microsoft.VisualBasic.Logging.FileLogTraceListener,
      Microsoft.VisualBasic, Version = 8.0.0.0, Culture = neutral,
      PublicKeyToken = b03f5f7f11d50a3a, processorArchitecture = MSIL "
      initializeData = "FileLogWriter" />
<- Elimine el comentario de la sección siguiente y reemplazar con APPLICATION_NAME
      el nombre de la aplicación para escribir en el registro de sucesos de aplicación
      ->
<- <Nombre add = "EventLog"
      type = "System.Diagnostics.EventLogTraceListener"
      initializeData = "APPLICATION_NAME" /> ->
</SharedListeners>
</System.diagnostics>
</Configuration>
```

En primer lugar, es necesario agregar una nueva fuente para el espacio de nombres System.Net. Usted ve que una fuente que ya existe para mi objeto (presentado en el capítulo 3); se agrega una para el espacio de nombres System.Net también.

A continuación, agregue un interruptor en la sección de interruptores de la fuente ha agregado. Por último, se agrega un SharedListener a esa sección y establecer el archivo para eliminar la información de seguimiento de forma automática.

El archivo app.config acabado, con la añade en negrita, se muestra en el listado 17-4. También se encuentra en el código de ejemplo en el sitio de este libro en la Web.

Listado 17-4: El archivo app.config Terminado

```

<? Xml version = "1.0" encoding = "UTF-8">
<configuration>
  <system.diagnostics>
    <sources>
      <Nombre de origen = "Microsoft.VisualBasic.Logging.Log.WindowsFormsSource"
        switchName = "DefaultSwitch">
        <listeners>
          <add name="FileLog"/>
```

```
</ Oyentes>
</ Source>
<source name="System.Net">
  <listeners>
    <add name="System.Net"/>
  </ Oyentes>
</ Source>
</ Sources>
<switches>
  <add name="DefaultSwitch" value="Information" />
  <add name="System.Net" value="Verbose" />
</ Switches>
<sharedListeners>
  <Add name = "FileLog"
    type = "Microsoft.VisualBasic.Logging.FileLogTraceListener,
    Microsoft.VisualBasic, Version = 8.0.0.0, Culture = neutral,
    PublicKeyToken = b03f5f7f11d50a3a, processorArchitecture = MSIL "
    initializeData = "FileLogWriter" />
  <Add name = "System.Net"
    type = "System.Diagnostics.TextWriterTraceListener"
    initializeData = "My.Log" />
</ SharedListeners>
<trace autoflush="true" />
</ System.diagnostics>
</ Configuration>
```

Ejecute de nuevo la aplicación y ver la ventana de resultados. Avanzada la tala información se muestra allí a causa de los cambios en el archivo de configuración. Además, un archivo de registro fue escrito. En el entorno de desarrollo, esto es en la carpeta bin / debug del proyecto. Puede que tenga que hacer clic en el Mostrar todo Archivos en la parte superior del Explorador de soluciones para ver.

En esa carpeta, verás un archivo llamado My.Log. Aquí es donde la Shared Oyente que agregó al archivo app.config dirigió la infor-logging mación. Mi copia de ese archivo se muestra en el listado 17-5 - su kilometraje puede variar.

Listado 17-5: La información de registro

```
Información System.Net: 0:
  WebRequest :: Create (ftp://ftp.vbfordummies.net/sample.bmp)
Información System.Net: 0: Salida de WebRequest :: Create () ->
  FtpWebRequest # 37460558
Información System.Net: 0: FtpWebRequest # 37460558 :: GetResponse ()
Información System.Net: 0: Salir FtpWebRequest # 37460558 :: GetResponse ()
System.Net Information: 0: Mensaje # 59487907 Asociar con
  HeaderCollection # 23085090
Información System.Net: 0: HeaderCollection # 23085090 :: Set (MIME-version = 1,0)
Información System.Net: 0: Asociar MailMessage # 6964596 con
  Mensaje # 59487907
System.Net Information: 0: :: SmtpClient ctor (host = 24.123.157.3)
```

(Continuación)

Listado 17-5: (continuación)

```
System.Net Information: 0: SmtpClient Asociar # 17113003 con
    SmtpTransport # 30544512
    Información System.Net: 0: Salida de SmtpClient :: ctor () -> SmtpClient # 17113003.
    Información System.Net: 0: SmtpClient # 17113003 :: Send (MailMessage # 6964596)
    Información System.Net: 0: SmtpClient # 17113003 :: Send (DeliveryMethod Network =)
    System.Net Information: 0: SmtpClient Asociar # 17113003 con
        MailMessage # 6964596
    Información System.Net: 0: Asociar SmtpTransport # 30544512 con
        ConexionSmtp # 44365459
    System.Net Information: 0: ConexionSmtp Asociar # 44365459 con
        Servicepoint # 7044526
    System.Net Information: 0: ConexionSmtp Asociar # 44365459 con
        SmtppooledStream # 20390146
    Información System.Net: 0: HeaderCollection # 30689639 :: set (content-transfer-
        codificación base64 =)
    Información System.Net: 0: HeaderCollection # 30689639 :: set (content-transfer-
        encoding = quoted-printable)
    Información System.Net: 0: HeaderCollection # 23085090 :: Remove (x-receptor)
    Información System.Net: 0: HeaderCollection # 23085090 :: set (de = bill@sempf.net)
    Información System.Net: 0: HeaderCollection # 23085090 :: set (a = bill@sempf.net)
    Información System.Net: 0: HeaderCollection # 23085090 :: set (fecha = 01 de abril 2008
        16:32:32 -0500)
    Información System.Net: 0: HeaderCollection # 23085090 :: set (subject = FTP
        Correcto)
    Información System.Net: 0: HeaderCollection # 23085090 :: get (MIME-Version)
    System.Net Information: 0: HeaderCollection # 23085090 :: get (desde)
    System.Net Information: 0: HeaderCollection # 23085090 :: get (a)
    System.Net Information: 0: HeaderCollection # 23085090 :: get (fecha)
    Información System.Net: 0: HeaderCollection # 23085090 :: get (subject)
    System.Net Information: 0: HeaderCollection # 30689639 :: get (content-type)
    System.Net Information: 0: HeaderCollection # 30689639 :: get (content-transfer-
        codificación)
    System.Net Information: 0: SmtpClient que sale # 17113003 :: Send ()
```

La lectura de este archivo, se puede ver que los números de referencia que coinciden con el peticiones en el servidor de todos aparece, mejorando notablemente la facilidad de depuración Ging. También, porque todo está en orden de acción, saber exactamente dónde se produjo el error en el proceso es mucho más fácil.

Capítulo 18

Creación de imágenes

En este capítulo

- Entender el espacio de nombres System.Drawing
- Encontrar la manera de encajar las clases de dibujo en el .NET Framework
- Utilizando System.Drawing para crear una aplicación simple juego

No se va a escribir la próxima edición de Bioshock utilizando Visual Basic. Lo que es el tipo de lenguaje que se utiliza para escribir gráficos intensivos aplicaciones tales como disparar-em-para arriba juegos.

Dicho esto, packs de Visual Basic una buena cantidad de energía en el sistema. Clases de dibujo. Mientras que estas clases son algo primitivo en algunas áreas, y el uso de ellos puede hacer que usted tenga que escribir unas cuantas líneas más de código lo que debe, no hay mucho que estas clases no se puede hacer con suficiente trabajar.

La capacidad de dibujo que proporciona el marco .NET se divide en cuatro áreas lógicas por el diseño de espacio de nombres proporcionada por Microsoft. Todo el general dibujo capacidad se encuentra en el espacio de nombres System.Drawing. A continuación, encontrar varios espacios de nombres especializados:

- System.Drawing.2D tiene funciones avanzadas de dibujo vectorial.
- System.Drawing.Imaging es principalmente sobre el uso de mapa de bits para la esteras, como. bmp y. archivos jpg.
- Ofertas System.Drawing.Text con tipografía avanzada.

En este capítulo, me centro en el espacio de nombres base y cubrir lo básico de dibujo en Visual Basic. (Discutir cada aspecto de dibujo fácilmente podría llenar un libro entero.)

Conozca System.Drawing

Incluso en el nivel más alto, la programación de gráficos se compone de polígonos de dibujo, llenándolos de color, y etiquetarlos con el texto - todo en un lienzo de algunos tipo. Como era de esperar, esto te deja con cuatro objetos que se encuentran son el núcleo del código gráfico que escribir: gráficos, bolígrafos, cepillos y texto.

Gráficos

En términos generales, la clase Graphics crea un objeto que es su paleta. Es el lienzo. Todos los métodos y propiedades del objeto Graphics son diseñado para hacer que el área que aprovechar más apropiado para sus necesidades.

Además, la mayoría de los métodos de gráficos y relacionados con la imagen de otras clases en el marco proporcionará el objeto Graphics como salida. Por ejemplo, usted puede llamar al método System.Web.Forms.Control.CreateGraphics de un Aplicación de Windows Forms y obtener una copia de objeto Graphics que permite dibujar en un control de formulario en su proyecto. También puede controlar la Pintura evento de un formulario, y echa un vistazo a la propiedad Graphics del evento.

Objetos gráficos utilizan lápices y pinceles - explica más adelante en este capítulo en las "plumas" y las secciones "Pinceles" - para dibujar y rellenar. Los objetos gráficos tienen métodos tales como los siguientes:

- DrawRectangle
- FillRectangle
- DrawCircle
- FillCircle
- DrawBezier
- DrawLine

Estos métodos aceptan lápices y pinceles como parámetros. Se podría pensar "¿Cómo es un círculo me va a ayudar?" Pero hay que recordar que incluso com-objetos gráficos complejos, como el Pacto en Halo 2 se acaba de hacer de círculos y rectángulos - solo miles y miles de ellos. El truco para arte útil es el uso de las matemáticas para juntar un montón de círculos y cuadrados hasta que tener una imagen completa. La aplicación de ejemplo se describe más adelante en este capí-ter es un ejemplo muy simple de eso.

Plumas

Utiliza lápices para dibujar líneas y curvas. Gráficos complejos están hechos de polígonos, los polígonos se componen de líneas, y dichas líneas se generan por plumas. Plumdas tienen propiedades tales como

- Color
- DashStyle
- EndCap
- Ancho

Usted consigue la idea: utilizar plumas para dibujar las cosas. Estas propiedades son utilizadas por los los corrales para determinar cómo las cosas se dibujan.

Pinceles

Cepillos de pintar el interior de los polígonos. Mientras utiliza los lápices para dibujar el formas, se utiliza pinceles para llenar las figuras con colores, patrones, o gradiente padres. Por lo general, los cepillos se pasan en un parámetro a un método DrawWhatever

de los objetos del corral. Cuando el lápiz dibuja la forma que se le pidió que dibujar, usa el pincel para llenar la forma - tal y como lo hizo en el jardín de la infancia con crayones y libros para colorear (el objeto pincel siempre permanece dentro de la líneas de embargo).

No busques la clase Brush, sin embargo. Se trata de un área de espera para el real cepillos, que tienen clase de nombres extraños. Los pinceles se hacen para ser cus- personaliz, pero se puede hacer mucho con los pinceles que vienen con el marco como es. Algunos de los cepillos incluyen

- SolidBrush
- TextureBrush
- HatchBrush
- PathGradientBrush

Mientras que las plumas se utilizan para pasar a los métodos de estiramiento de los gráficos objeto, los cepillos se utilizan para pasar a los métodos de relleno que forman polígonos.

Texto

El texto está pintado con una combinación de fuentes y pinceles. Cepillos funcionan como las plumas, la clase Font utiliza pinceles para llenar las líneas de una operación de texto.

System.Drawing.Text cuenta con colecciones de todas las fuentes instaladas en el sistema el funcionamiento de su programa, o se instala como parte de su solicitud. Sistema.Drawing.Font tiene todas las propiedades de la tipografía, tales como los siguientes:

- Audaz
- Tamaño
- Estilo
- Subrayar

El objeto Graphics, una vez más, ofrece la escritura del texto en la paleta.

Como las clases de dibujo Colocar en el Marco

El espacio de nombres System.Drawing rompe dibujo en dos pasos:

- Crear un objeto System.Drawing.Graphics.
- Utilice las herramientas en el espacio de nombres System.Drawing para dibujar sobre ella.

Parece sencillo, y lo es. El primer paso es obtener un objeto Graphics. Los objetos gráficos provienen de dos lugares principales - las imágenes existentes y Windows Forms.

Para obtener un objeto Graphics a partir de una imagen existente, mira el objeto Bitmap. El objeto de mapa de bits es una gran herramienta que te permite crear un objeto utilizando un archivo de imagen existente. Esto le da una nueva paleta que se basa en un mapa de bits imagen (archivo JPEG, por ejemplo) que ya se encuentra en su disco duro. Es un muy conveniente, especialmente para imágenes web. Así es como se carga un mapa de bits:

```
MyBitmap As New Bitmap ("c:\images\myimage.jpg")
MyPalette As Graphics = Graphics.FromImage (myBitmap)
```

Impresión de una forma

En VB6 y versiones anteriores, una de las más comunes formas de obtener información sobre el papel era sólo imprimir un formulario. Esta funcionalidad no existe en VB 2002, 2003 y 2005, pero regresó en un Poder Pack y ahora se integra en Visual Studio 2008. Lo está disponible para todos los idiomas, pero debería ser más útil para los programadores de VB.

Si necesita crear un informe, debe utilizar Microsoft Report Viewer, que yo no cubren en este libro. Si lo que desea es obtener un texto y imágenes a la impresora del usuario, sin embargo, el Componente PrintForm debe hacer el truco.

Para utilizar el componente PrintForm,

arrástrelo desde el Cuadro de herramientas al formulario en la vista Diseño. Aparecerá en la bandeja de componentes. En la

de control de eventos para la función de impresión (el MenulItem.Click función, por ejemplo), establecer la propiedad Form del componente, la Imprimir en Acción y luego llamar al comando Imprimir. Se parece a esto:

```
Con PrintForm1
    .Forma = 'Me TheFormIWantPrinted utilizar
              para la forma actual
    .PrintAction = PrintToPrinter
              "Hay otras opciones de archivos o
              Avance
              .Imprimir ()
Extremo con
```

El formulario será enviado a la impresión de Windows funcionamiento, al igual que si se hubiera usado el cuadro de diálogo Imprimir para imprimir un archivo.

Ahora el myPalette objeto es un objeto Graphics cuya altura y anchura se basan en la imagen en myBitmap. Lo que es más, la base de la mi Paleta de imagen se ve exactamente como la imagen de referencia en el myBitmap objeto.

Puede utilizar los lápices, pinceles y fuentes en la clase Graphics para dibujar derecho en que la imagen, como si se tratara de un lienzo en blanco. Yo lo uso para poner el texto en imágenes antes de que les muestro en las páginas web y para modificar el formato de las imágenes en la volar, también.

La segunda manera de obtener un objeto Graphics es conseguirlo de formularios Windows Forms. El método que usted está buscando es

```
System.Windows.Forms.Control.CreateGraphics
```

Este método le da una nueva paleta que se basa en la superficie de dibujo de el control se hace referencia. Si se trata de una forma, hereda la altura y la anchura

de la forma y el color de fondo tiene forma. Puede utilizar lápices y pinceles para dibujar directamente en el formulario.

Cuando se tiene un objeto Graphics, las opciones son casi infinitas. Dibujo sofisticado no está fuera de la cuestión, a pesar de que tendría que para hacer una obra tonelada para crear gráficos como los que ves en Halo 3 usando Visual Basic. (No es una clase de Master Chief que sólo puede generar automáticamente).

Sin embargo, incluso los gráficos 3D más complejos son polígonos sólo color, y usted puede hacer las personas con la clase System.Drawing. En la siguiente secciones, voy a construir un tablero de cribbage con un objeto Graphics, bolígrafos, cepillos, y fuentes.

Utilizando el espacio de nombres System.Drawing

Aplicaciones buenas vienen desde lugares extraños. Gabrielle (mi esposa) y yo disfrutar juegos, y uno de nuestros favoritos es el cribbage juego de cartas. Estábamos en vacación en Disney World cuando ella tenía ganas de jugar, pero no teníamos una cuna bage bordo. Tuvimos tarjetas, pero no la junta directiva.

Sin embargo, yo tenía mi laptop, Visual Studio y System.Drawing la espacio de nombres. Después de sólo una o dos horas de trabajo, construí una aplicación que sirve como un tablero de cribbage trabajando!



Se trata de una aplicación bastante completa, y no tengo suficientes páginas para caminar que a través de él paso a paso. Cargue la aplicación desde el sitio Web en www.vbfordummies.net, y sigue con el resto de este capítulo. Esto no es una aplicación compleja, pero es larga.

Primeros pasos

Cribbage es un juego de cartas donde se cuentan las manos para arriba en puntos, y el primer jugador en anotar 121 puntos gana. Todo depende de los jugadores para contar los puntos, y el resultado se guarda en una tabla.

Tableros de Cribbage se componen de dos líneas de agujeros para clavijas, por lo general por un total de

120, pero a veces se utilizan 60 hoyos y jugar a través de dos veces. Figura 18-1 muestra un tablero de cribbage típico. Cribbage tarjetas vienen en un montón de diferentes estilos - echa un vistazo a www.cribbage.org si usted es realmente curioso, tiene una gran galería de casi 100 tablas, desde la básica hasta caprichosa.

Para este ejemplo, acabo de crear la imagen del tablero para una aplicación que mantiene puntuación de un juego de naipes - pero no sería más allá de Visual Basic para escribir las cartas en el juego, también!

Así que el consejo para esta aplicación cuenta con 40 hoyos en cada uno de los tres pares de líneas, que es la configuración de la placa estándar para dos jugadores jugando a 120, como se muestra en

Figura 18-2. La primera tarea es dibujar el tablero, y luego sacar las clavijas como las puntuaciones de los jugadores - introducidos en los cuadros de texto - CAMBIAR.

La premisa es la siguiente: Un jugador juega una mano y entra en las puntuaciones resultantes en el cuadro de texto debajo de su nombre (vea la Figura 18-2). Cuando la puntuación para cada mano se introduce, la puntuación al lado del nombre del jugador se actualiza y el clavija se mueve en el tablero. La próxima vez que un mismo jugador marca de mano, la clavija se mueve hacia adelante y la clavija se mueve de nuevo en su lugar. ¿No te mencionar la vinculación de nuevo? Oh, sí, el inventor de cribbage era un paranoico de trampa - si usted no está familiarizado con siete y medio, es posible que desee comprobar hacia fuera el reglas en www.cribbage.org.



Figura 18-1:
Un tradicional
cribbage
bordo, foto
por Turtle AJ.

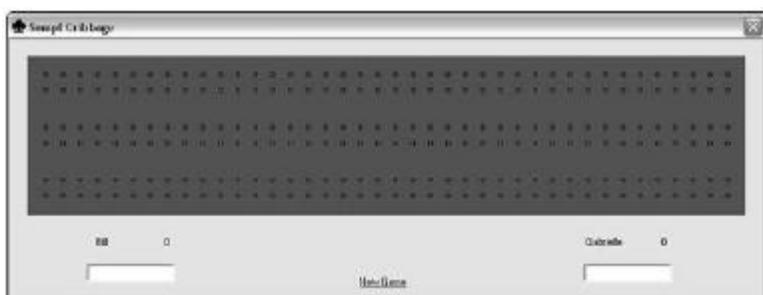


Figura 18-2:
Lo digital
cribbage
bordo.

Configuración del proyecto

Para comenzar, cree una superficie de juego. De hecho, me creó la tabla se muestra en la Figura 18-2 sin llamar la propia placa - Pinto que en adelante con System.Drawing. Mi consejo se parecía mucho a la figura 18-3 cuando estuviera lista comenzar con las reglas de negocio.

He utilizado una subrutina poco para manejar los cambios de puntaje llamando a la subrutina de eventos OnChange los dos cuadros de texto. El código que llama al-subrou tine sigue:

```
Private Sub HandleScore (ByVal scoreBox como TextBox, puntos de ByVal como Label,
    OtherPlayer ByVal como Label)
    Tratar de
        Si 0 > CInt (scoreBox.Text) o CInt (scoreBox.Text) > 27 Y
            ScoreCheck.SetError (scoreBox, "Score debe estar entre 0 y 27")
            scoreBox.Focus ()
        Más
            ScoreCheck.SetError (scoreBox, "")
            "Añadir la partitura escrita a los puntos
            points.Text = CInt (points.Text) + CInt (scoreBox.Text)
    End If
    Catch ext Como System.InvalidCastException
        "Algo más que un número
        Si scoreBox.Text.Length > 0 Then
            ScoreCheck.SetError (scoreBox, "Score debe ser un número")
        End If
    Catch ex As Exception
        "Eek!
        MessageBox.Show ("Algo salió mal!" + Ex.Message)
    End Try
    'Comprobar la puntuación
    Si CInt (points.Text) > 120 A continuación,
        Si CInt (points.Text) / CInt (otherPlayer.Text) > 1.5 Después
            WinMessage.Text scoreBox.Name.Substring = (0,
                scoreBox.Name.Length - 6) y "skunked 'em!"
        Más
            WinMessage.Text scoreBox.Name.Substring = (0,
                scoreBox.Name.Length - 6) y "Won!"
    End If
    WinMessage.Visible = True
End If
End Sub
```

Todo este cambio de valores de la pantalla hace que el evento Paint del formulario fuego - cada vez que VB tiene que cambiar el aspecto de un formulario por cualquier razón, esta activa el evento - por lo que sólo tiró un poco de código en ese controlador de eventos que lo haría dibujar mi consejo para mí:

```
Private Sub CribbageBoard_Paint (ByVal sender As Object, ByVal e As  
    System.Windows.Forms.PaintEventArgs) Maneja MyBase.Paint  
    PaintBoard (BillsPoints, GabriesPoints)  
End Sub
```

A partir de ese momento, mi mayor preocupación es el dibujo de la placa misma.

Señalando a la junta

Tengo que pintar directamente en un formulario para crear la imagen de la tarjeta para mi cuna aplicación bage, así que uso el método CreateGraphics de control de formulario. A partir de ahí, tengo que hacer lo siguiente:

- Pintar el marrón placa con un cepillo.
- Dibuja seis filas de pequeños círculos con un lápiz.
- Rellene el hoyo si ese es el resultado correcto.

Limpie mis provisiones.

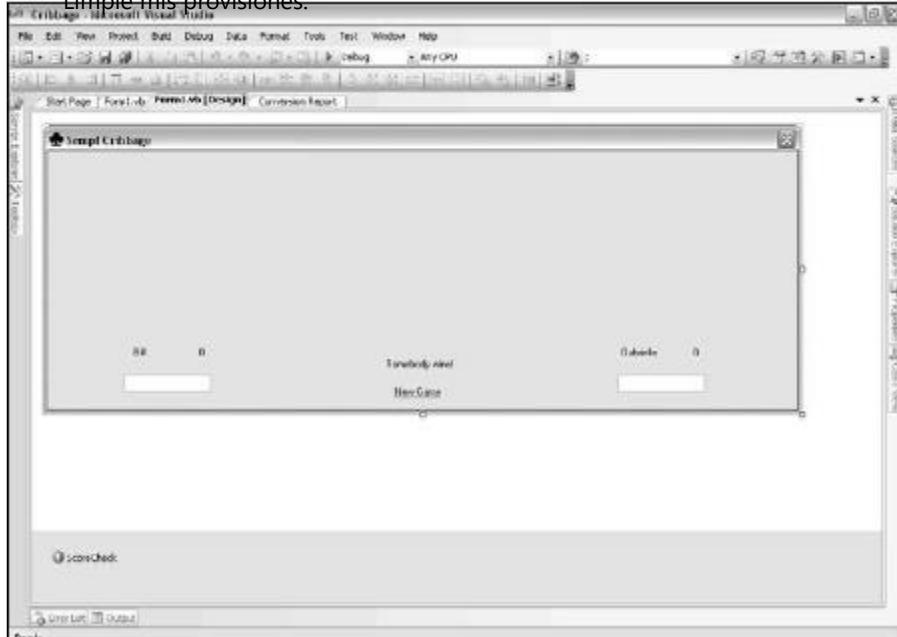


Figura 18-3:
El básico
bordo.

A tal fin, se me ocurrió con el método PaintBoard, que acepta la etiquetas que contienen los resultados de pie para ambos jugadores. Se muestra en la Listado 18-1.

Listado 18-1: El Método PaintBoard

```

PaintBoard Private Sub (Bill ByRef como Label, ByRef Gabrielle Como Label)
    Paleta As Graphics = Me.CreateGraphics
    BrownBrush dévil como nuevo SolidBrush (Color.Brown)
    palette.FillRectangle (brownBrush, New Rectangle (20, 20, 820, 180))
    'OK, ahora tengo que pintar los pequeños agujeros.
    "Hay 244 pequeños agujeros en el tablero.
    "Tres filas de 40 veces dos, con los arranques y paradas pequeñas en cada extremo.
    "Vamos a empezar con los 240.
    Filas As Integer
    Columnas Dim As Integer
    ScoreBeingDrawn Dim As Integer
    BlackPen As New Pen (System.Drawing.Color.Black, 1)
    Prieto dévil como nuevo SolidBrush (Color.black)
    RedBrush dévil como nuevo SolidBrush (Color.Red)

    'Hay 6 filas, a continuación, en 24 y 40, 80 y 100, luego 140 y 160.
    Para las filas = 40 a 160 Etapa 60
        "Hay 40 columnas. Son cada 20
        Para las columnas = 40 a 820 Paso 20
            'Calcular puntuación está elaborando
            scoreBeingDrawn = ((columnas - 20) / 20) + (((filas + 20) / 60) - 1)
                * 40)
            'Bill Draw
            'Si el puntaje está elaborando = factura de relleno, de lo contrario dibujar
            Si scoreBeingDrawn = CInt (Bill.Text) A continuación,
                palette.FillEllipse (prieto, columnas, filas - 2 - 2, 6, 6)
            ElseIf scoreBeingDrawn = BillsLastTotal Entonces
                palette.FillEllipse (redBrush, columnas, filas - 2 - 2, 6, 6)
            Más
                palette.DrawEllipse (blackPen, columnas, filas - 2 - 2, 4, 4)
            End If
            "Dibujar Gabrielle
            'Si el puntaje está elaborando = Gabrielle relleno, de lo contrario dibujar
            Si scoreBeingDrawn = CInt (Gabrielle.Text) A continuación,
                palette.FillEllipse (prieto, columnas, filas - 2 + 16, 6, 6)
            ElseIf scoreBeingDrawn = GabriellesLastTotal Entonces
                palette.FillEllipse (redBrush, columnas, filas - 2 + 16, 6, 6)
            Más
                palette.DrawEllipse (blackPen, columnas, filas - 2 + 16, 4, 4)
            End If
            Próximo
            Próximo
            palette.Dispose ()
            brownBrush.Dispose ()
            blackPen.Dispose ()
End Sub

```

A parte de las matemáticas, tenga en cuenta la toma de decisiones. Si el partido está elaborando es la puntuación en la etiqueta, llenar el agujero con una clavija roja. Si es la última puntuación dibujado, llenar el agujero con una estaca negro. Si no, bueno, dibujar un círculo.

Es difícil de entender, pero así es exactamente como grandes juegos se escriben. Es cierto que los motores gráficos más grandes hacen muchos If-Then decisiones, pero la premisa es la misma.

Además, los juegos grandes, utilice imágenes de mapas de bits a veces, en lugar de dibujar todo el tiempo. Para la aplicación de puntuación cribbage, por ejemplo, podría usar un imagen de mapa de bits de un tipo de cambio en lugar de sólo llenar una elipse con un negro o rojo cepillar!

Parte V

La Parte de los Diez

The 5th Wave

By Rich Tennant



"It's a horse racing software program. It analyzes my betting history and makes suggestions. Right now it's suggesting I try betting on football."

En esta parte. . .

Y en esta parte, usted encontrará consejos sobre todo lo que se mueva después de la lectura de este libro a la búsqueda de recursos en línea. Hay aún más información en el sitio web: www.vbfordummies.com. Espero que usted encontrará que es una herramienta útil referencia!

Capítulo 19

Diez consejos para el uso de la interfaz de usuario de VB

En este capítulo

Encontrar maneras múltiples para generar código de control de eventos

Copiar el texto de la forma en que realmente lo quiere

Hacer buen uso de las barras de herramientas

Extender VB con tus propias creaciones

Utilizando los mismos trucos fresco como a los profesionales

V

isual Studio es una gran herramienta para la escritura de código de Visual Basic, pero es así va a utilizar esta herramienta que no puede encontrar las características realmente ordenada hasta la próxima versión está disponible! Para ayudarle a encontrar características interesantes para la escritura de VB código, he compilado esta lista de (casi) diez consejos para trabajar con el Visual Estudio interfaz. Espero que estos consejos hacen su codificación más agradable!

La generación de controladores de eventos de la ventana de propiedades

Cuando se trabaja en la vista Diseño para Web o Windows Forms y doble clic en un control, Visual Studio te trata con el código para el valor predeterminado controlador de eventos. Pero ¿qué pasa si usted no desea que el controlador de eventos predeterminado? Cualquier dado objeto a menudo tiene varios eventos que puede que desee tener acceso, y Visual Studio puede con la misma facilidad (y automágicamente) generar código para cualquier uno de esos eventos.

Siga estos pasos para darle una oportunidad:

- 1. Crear un nuevo proyecto de Visual Basic 2008 aplicación para Windows. (Se obras en los formularios Web Forms, también.)**
 - 2. Arrastre un control al formulario predeterminado.**
- Puedo utilizar un cuadro de texto en este ejemplo.
- 3. Seleccione el control y luego presione F4 para ampliar las Propiedades ventana.**
 - 4. En la ventana Propiedades, haga clic en el botón de cita (que se parece a una rayo poco).**

Verá una lista de eventos asociados, algo así como la lista que se muestra en Figura 19-1.

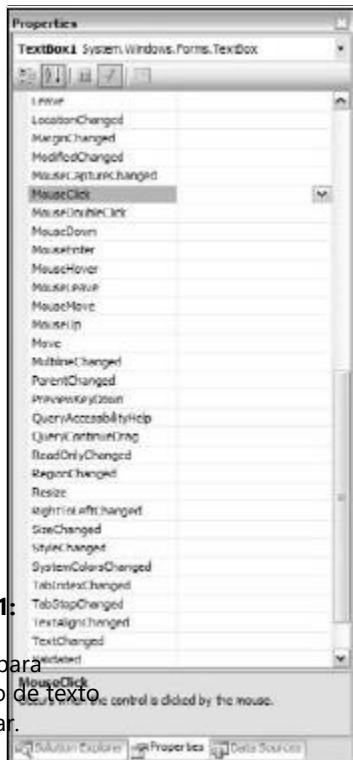


Figura 19-1:
El listado
de eventos para
un cuadro de texto
controlarán.

Capítulo 19: Diez consejos para el uso de la interfaz de usuario de VB

5. Haga doble clic en un evento de la lista para generar el código de controlador de eventos.

Los eventos listados en la Figura 19-1 son todos los eventos expuestos por el TextBox objeto. He seleccionado el evento MouseClick. Esto significa que cualquier tiempo clics del usuario en el cuadro de texto, el código se ejecutará.

Al hacer doble clic en un evento en la ventana de eventos, el código generado se ve algo como lo siguiente:

```
Private Sub TextBox1_MouseClick (ByVal como System.Object remitente, _
    ByVal e As System.Windows.Forms.MouseEventArgs) _
    Manijas TextBox1.MouseClick

End Sub
```

Compruebe la instrucción Handles al final de la declaración, sino que debe coincidir el evento seleccionado.

La generación de controladores de eventos desde la vista Código

En la sección anterior se muestra cómo crear controladores de eventos de la Diseño Ver - pero la vista de diseño no es la única parte del Visual Basic interfaz con funciones automáticas. La vista de código tiene un par de trucos, también. En concreto, los eventos se enumeran cómodamente a lo largo de la parte superior de la vista de código, por tanto la navegación entre los eventos y la creación de otras nuevas. Les muestro este lugar- ción en la Figura 19-2.

Figura 19-2 también muestra el controlador predeterminado nuevo que he creado con un doble clic en un botón y haciendo clic en la lista desplegable de arriba a la derecha de la ventana de código. Se puede ver que MouseClick está en negrita, lo que significa que el Método MouseClick existe. Todos los demás eventos que expone el objeto Button se muestran también.

La lista desplegable a la izquierda se muestran los objetos instanciados en el formulario. En este ejemplo, la lista sólo mostrará el botón y los objetos de formulario. Los siguientes pasos describen el proceso para la creación de un nuevo evento en este punto de vista:

1. Haga clic en el formulario que desea editar y seleccione Ver código para ir a Ver código.

**a manejar un evento de más a la izquierda de la
en la lista desplegable.**

313

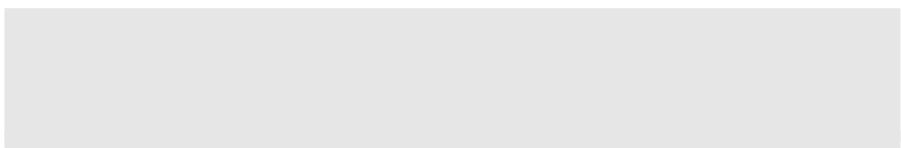
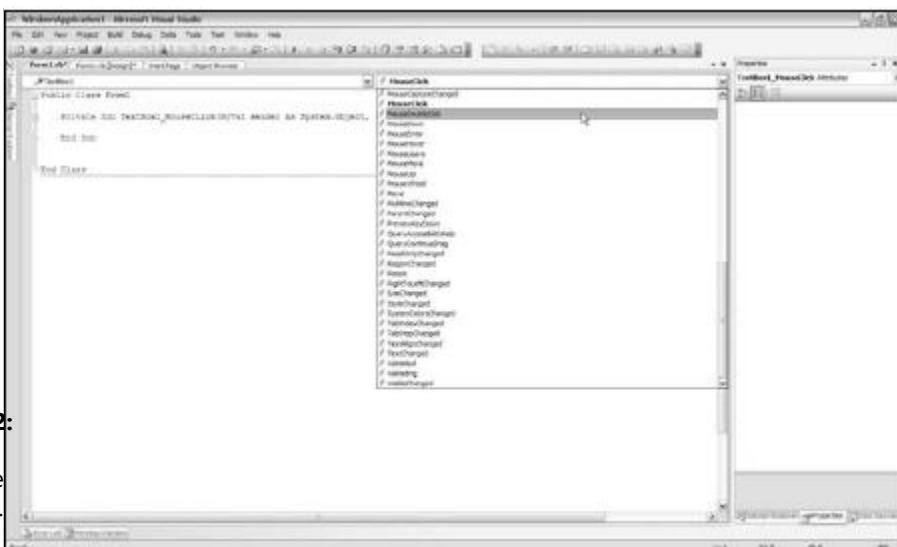


Figura 19-2:
El evento
selector de
Ver código.



3. Seleccione el evento que desea gestionar desde el menú del extremo derecho desplegable.

En este punto, Visual Studio crea el código auxiliar de la subrutina para usted, y usted puede agregar el código que necesitan al alcance de la misma.

Más tarde, si el código en el formulario se larga, puede navegar de vuelta al evento controlador de la misma manera y hacer cambios.

Pegar texto como HTML

Al hacer páginas web en los formularios Web Forms, que a menudo puede encontrarse tirando el contenido de otras páginas web abiertas en Internet Explorer o en la Oficina documentos. Por ejemplo, al crear sitios Web estáticas, el contenido puede vengo a ti en la forma de un documento de Word.

El problema de llevar el contenido de esta forma proviene de la interactividad entre los programas de Microsoft. ¿Te has dado cuenta de que, cuando se corta a partir de Excel y pegar a Word, la estructura de la tabla queda? Productos de Microsoft intenta para mantener el formato siempre que sea posible. Echa un vistazo a este ejemplo:

1. Abra cualquier documento de Word que contiene texto con formato.

Para este ejemplo, sólo tiene que utilizar el documento de Word en este capítulo.

Capítulo 19: Diez consejos para el uso de la interfaz de usuario de VB

- 2. Abra Visual Studio y crear un sitio Web nuevo.**
- 3. En la página por defecto, haga clic en la ficha Diseño para cambiar a la vista Diseño (si aún no se encuentra en la vista Diseño).**
- 4. Selecciona el texto desde el documento de Word y elija Editar Copiar ⇔.**
- 5. De nuevo en Visual Studio, coloque el cursor en la página web predeterminada y elija Editar Pegar ⇔.**
- 6. Haga clic en la ficha de código para cambiar a la vista Código y mirar el texto de su cortar-y-pegar trabajo quede atrás.**

Mi ejemplo deja el texto que aparece en el listado 19-1.

Listado 19-1: Messy HTML del comando Pegar

```
<h1>
    Pegar como HTML </ h1>
<p class="MsoNormal">
    Al hacer páginas web en los formularios Web Forms, que a menudo puede encontrarse tirando contenido
    de otro
    Abrir páginas Web en Internet Explorer o de documentos de Office
    .<span Style="mso-spacerun: yes"> </ span> Por ejemplo, cuando estás
        la creación de Web estático
    sitios, el contenido puede
    <? Xml namespace = "" prefix = "st1"?
    <st1:state w:st="on"> <st1:place w:st="on"> venir </ st1: place> </ st1: State>
    a usted en la forma de un documento de Word. </ p>
```

La mayoría de las veces, usted no desea conservar el formato de otros programas cuando se trata de una aplicación web. ¿Quieres copiar el contenido de el otro programa (documento Word, hoja de cálculo Excel, etc) y pegar sólo el texto para que pueda aplicar estilos adecuados para una página Web. Si ese es el caso, elija el comando Edición ⇔ Pega alternativo en lugar de la Edición ⇔ Pega comando en el paso anterior 5.

Personalizar las barras de herramientas para cada tipo de archivo

Mientras que en el constructor de formularios Web Forms de Visual Studio, es posible que note algo

sta Código,
las barras de herramientas - los botones debajo de los menús - el cambio. De hecho, su
trabajo-
área ing podría incluso cambiar el tamaño de las barras de herramientas porque se
añaden o eliminan.

Esta cambiante de las barras de herramientas, se debe a que Visual Studio es compatible con una diferente barra de herramientas de configuración diferente para cada tipo de archivo y la vista. HTML y páginas ASPX puede tener totalmente diferentes barras de herramientas, y la vista Diseño y la vista Código puede tener barras de herramientas diferentes para cada tipo de archivo.

Puede personalizar su espacio de trabajo, especificando las barras de herramientas que desea ver para cualquier tipo de archivo determinado o vista. Basta con abrir un archivo del tipo deseado y haga clic derecho en la barra de la cinta (la zona gris en los menús). Cuando lo haga, se obtiene una larga lista de barras de herramientas disponibles, y usted puede simplemente haga clic en los que desee. Enumero algunas barras de herramientas y los contenidos en la Tabla 19-1.

Tabla 19-1 Barras de herramientas para diferentes tipos de archivos y vistas

Barra de herramientas	¿Qué es bueno para	
Nombre	Tiene botones que coinciden con el menú Generar y le permiten compilar el proyecto con varias opciones.	
Construir	Contiene herramientas de diseño que son apropiados para la fabricación de DLL archivos.	
Diseñador de clases	Depurar	Tiene botones que funcionan de manera similar al menú Debug y le permitirá entrar en el modo de depuración o depurar otros run-Ning programas.
Dispositivo	Contiene herramientas para ayudarle a trabajar con dispositivos inteligentes	
Disposición	(Como teléfonos móviles) y le dará acceso a la emuladores para Pocket PC.	
Diseñador de consultas	Contiene herramientas (tales como el elemento de alineación se menciona en Capítulo 4) que son útiles para la estructuración de formas.	
Hoja de estilo	Tiene herramientas para ayudarle a crear consultas SQL y XML de datos.	
	Contiene botones para aplicar estilos CSS a HTML.	

Adición de Proveedores extensores

Extender los proveedores podría usar su propio capítulo, pero debido a todos-Lo más acerca de Visual Basic que tengo que decirte, te doy una breve mirada a ellos aquí. Un proveedor extensor proporciona una extensión - en concreto nuevo - propiedades a un objeto o grupo de objetos. Prueba esto:

Capítulo 19: Diez consejos para el uso de la interfaz de usuario de VB

1. Abra Visual Studio y comenzar un nuevo proyecto de aplicación para Windows VB.

2. Arrastre un Botón objeto en el formulario.

Supongamos que usted desea mejorar su botón con una información sobre herramientas - la pequeña ventana flotante que aparece en algunas aplicaciones cuando se coloca el ratón sobre un objeto en la pantalla. En una aplicación de Windows VB proyecto, no se puede agregar información sobre herramientas directamente al objeto Button.

3. Para agregar una propiedad ToolTip del botón, añada una ToolTip objeto a la bandeja de componentes.

Usted puede encontrar el objeto ToolTip en la caja de herramientas. La información sobre herramientas aparece en la bandeja de componentes.

4. Vuelva a la ventana de propiedades del botón.

Observe que, en la parte inferior de la ventana en la categoría Varios, una información sobre herramientas ahora existe!

Debido a que la información sobre herramientas es un proveedor extensor, que está diseñado para dar a todos los objetos que llenar una interfaz de una nueva propiedad, en este caso, una información sobre herramientas. Aunqueadir una propiedad a un objeto puede parecer un poco extraño, en realidad es un gran manera de extender la funcionalidad de un conjunto de controles. Y creo que Microsoft está haciendo así que era bastante brillante.

Usted puede crear sus propios proveedores extensores para hacer todo, desde agregar cadenas de texto para hacer referencia a todos el camino para hacer nuevo y funcional, y elementos gráficos. Estas propiedades le dan una forma poderosa de extender la interfaz de usuario proporcionada por Microsoft.

Uso de componentes visuales Que no son tan Visual

Faltaría a mí deber si yo no te dije acerca de la Bandeja de componentes y, especialmente, los componentes visuales que no son tan visual. Aunque Visual Basic es configurado como una herramienta rápida, el desarrollo de apuntar y hacer clic, un montón de objetos (que no son del todo) visible todavía debe ser y son administrados por el desarrollo visual. Mientras que las herramientas.

ario que aparece cuando se utiliza un componente no visual, como el temporizador o conjunto de datos. Figura 19-3 muestra la Bandeja de componentes, que muestra sólo unos pocos fea real- ras, ya que la mayor parte del desarrollo de apuntar y hacer clic con el botón implica mover componentes visuales por la pantalla.

Otras partes del libro se refieren a componentes no visuales un número de veces. En Parte II, a averiguar sobre el objeto de menú. En la Parte III, os muestro el temporizador objeto. La Parte IV contiene información acerca de los objetos de datos. En anteriores de este capítulo sección, os muestro el objeto ToolTip. No todos ellos tienen partes utilizables que se ve en pantalla, pero sí aparecen en la bandeja de componentes.

Y, aunque no se puede cambiar la posición de estos componentes en la pantalla, usted puede hacer el siguiente:

Haga clic en el componente para obtener un menú sensible al contexto, al igual que usted para hacer un botón o componente otra ventana.

Abra la ventana Propiedades y haga clic en el componente en la bandeja para seleccionarlo para su edición. En la ventana Propiedades, puede cambiar el nombre del componente y otras propiedades comunes.



Esto es una advertencia a tener en cuenta: Declarar el objeto en la vista Código no inmediatamente lo convierten en un objeto bandeja de componentes. En general, si usted Para editar un componente en la vista Diseño, es necesario crearlo en Diseño Ver. La mayoría de los desarrolladores son los principales desarrolladores ya sea la vista Código o Diseño

Ver los desarrolladores. Al encontrar su estilo personal, podrás conocer cómo frecuencia con la que voy a utilizar la bandeja de componentes.

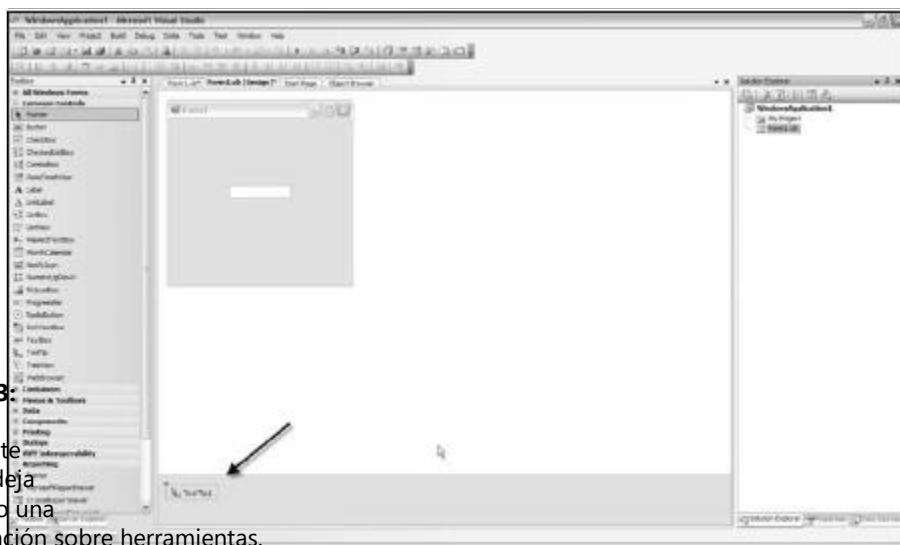


Figura 19-3:
La
Componente
Bandeja
mostrando una
información sobre herramientas.

Grabación de macros

Las macros son tan cool que consiguen su propia interfaz de usuario, lo que muestro en Figura 19-4. Y macros son tan poderosas y flexibles, tienen el potencial para cambiar totalmente sus patrones de desarrollo. Aquí les presento una pequeña parte de su poder. (Dejo la experimentación de usted.)



Una de las mejores cosas que puede hacer es grabar una macro de teclas de ahorro para su posterior reproducción. Si usted tiene una tarea (por ejemplo, el formato de texto) que usted espera hacer más de una vez, guardar como macro.

Una de mis formas favoritas para utilizar una macro es grabar el formato HTML. Tengo toda una serie de macros que grabé para dar formato a cadenas de párrafos y listas. Formateo con macros pregrabados hace que la gestión del sitio con Visual Estudio de una brisa.

Para llegar a las macros, puede hacer algunas cosas:

Pulse Alt + F8 para abrir el Explorador de macros en Visual Studio.

Seleccione Herramientas → Macros Macros IDE desde la barra de menú.

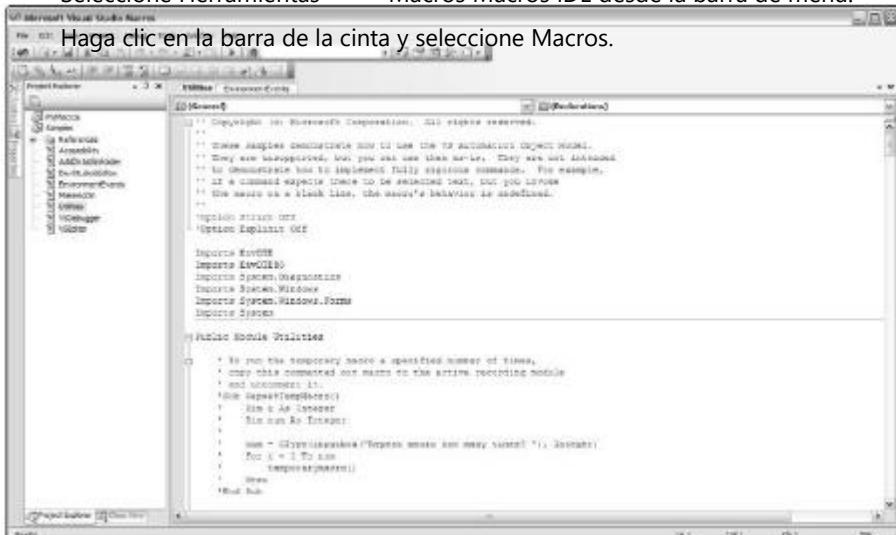


Figura 19-4:
El Macro
IDE.

Arranca el Explorador de macros (presione Alt + F8) y siga estos pasos para ver cómo se hace:

- 1. Comience con un nuevo proyecto Web en Visual Studio. Haga clic con el HTML Diseñador y seleccione Ver código para cambiar a la vista Código.**
- 2. Abra la página Web predeterminado y pegar unas pocas líneas de texto en el Código Ver.**

Por ejemplo, copiar un texto desde el Bloc de notas, como si lo ha recibido de un cliente, y pegar el texto en la vista Código. Cada párrafo se ser una larga línea de texto en la pantalla Vista Código.

- 3. Haga clic a la izquierda de la primera línea del texto para dejar el cursor al principio de la primera línea que desea dar formato.**
- 4. Seleccione Herramientas ⇔ Macro Registro Temporal.**

El panel Grabar aparece y comienza a capturar todos los com-significativo ordeno dar Visual Studio.

- 5. Escriba lo siguiente para registrar el formato de un párrafo HTML:**

- Escriba la etiqueta HTML párrafo, <p>, al comienzo del texto que deseé dar formato. Si Visual Studio agrega el resto de la etiqueta de párrafo, simplemente borrarla pulsando Eliminar.
- Pulse Finalizar para mover el cursor hasta el final de la línea de texto.
- Escriba la etiqueta HTML párrafo estrecha, </ p>.
- Pulse la tecla de flecha derecha, que mueve el cursor al inicio de la siguiente línea.

- 6. Haga clic en el botón Detener grabación en el panel de grabación.**

La macro se registra ahora y, lo que es más, en realidad es el código que reside en el lugar macro temporal en el Explorador de macros. Listado 19-2 muestra este código.

Listado 19-2: El Macro Párrafo

```
Las importaciones EnvDTE
Las importaciones EnvDTE80
Imports System.Diagnostics

RecordingModule Módulo Pública
Sub TemporaryMacro ()
    DTE.ActiveDocument.Selection.Text = "<P>" 
    DTE.ActiveDocument.Selection.Delete (4)
```

```
DTE.ActiveDocument.Selection.EndOfLine ()  
DTE.ActiveDocument.Selection.Text = "</ P>"  
DTE.ActiveDocument.Selection.CharRight ()  
End Sub  
End Module
```

Si hace clic derecho y seleccione Editar TemporaryMacro, se puede ver el código el IDE de macros, donde se puede hacer lo siguiente:

Realizar cambios en la macro.

Eliminar líneas de código innecesarias (como la línea creada cuando he eliminado el párrafo generado automáticamente etiqueta de cierre en mi ejemplo) para guardar una macro limpiador.

Haga la macro permanente copiando el código en otro módulo en la grabadora de macros y guardarla.

Haga clic derecho y selecciona Guardar como para guardar el archivo con un nombre diferente en el Explorador de macros.

Ejecutar la macro desde el Explorador de macros haciendo doble clic, o utilizar el Menú contextual en la vista Código, como se muestra en la Figura 19-5.

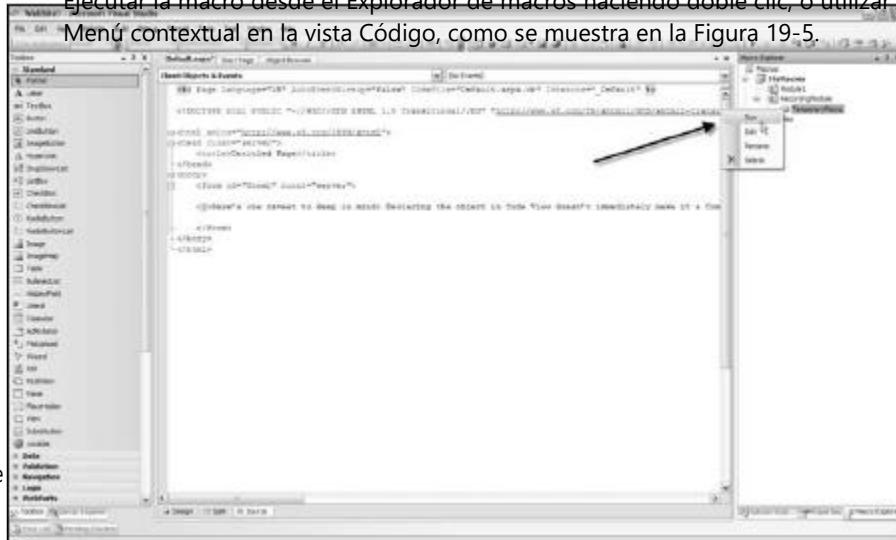


Figura 19-5:
Ejecución de
una
macro

Uso de la lista de tareas

La lista de tareas es una característica muy fresco personal de gestión del proyecto de la Visual Studio IDE. Para mostrarlo, seleccione Vista \leftrightarrow Otras ventanas Lista de tareas o \leftrightarrow pulse Ctrl + Alt + K. Para usarlo, simplemente haga clic en el botón Crear nueva tarea de usuario y a continuación, comience a escribir. Cuando haya completado la tarea en su lista de tareas, puede acaba de marcar.

Usted puede hacer mucho más con esta herramienta que acaba de hacer listas de tareas pendientes. Es totalmente integrado con control de código fuente, también, así que espera de check-ins, etc aparecen en la lista. Además, puede agregar fichas en el código que aparece en la lista de tareas.

Para agregar una ficha, abra cualquier proyecto y añadir un comentario que comienza con TODO. Se dará cuenta de que después de agregar un comentario TODO, un comentario se presenta con Los comentarios en la lista de tareas en la lista desplegable, como se muestra en la Figura 19-6. Double-click en una tarea con comentario le trae al archivo y la línea donde el comentario fue insertado. Usted puede usar esto para no perder de vista las cosas que usted necesita recordar a medida que los hacen.

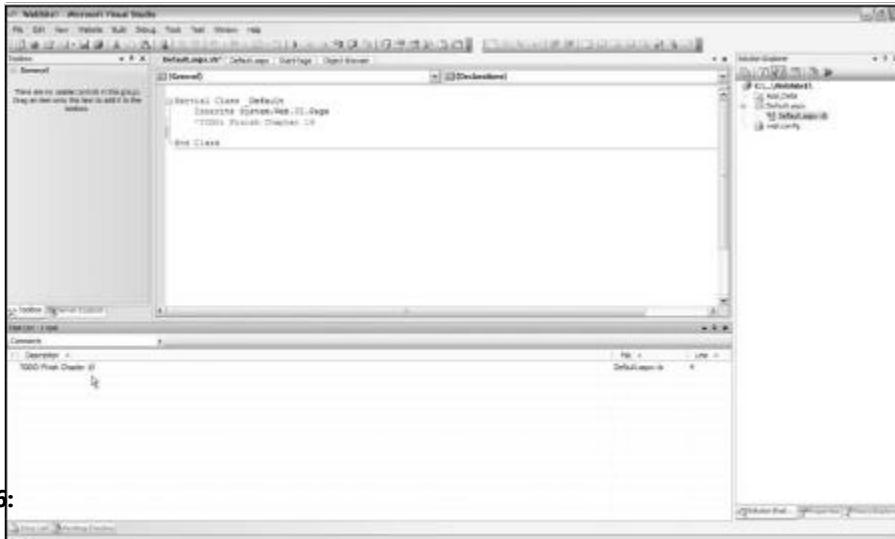


Figura 19-6:
La Tarea
lista.

Las fichas incluyen otros HACK y deshacer. Estos no suelen salirse con la propia filtro Lista de tareas, pero usted puede usar para marcar el código cuestionable que puede que tenga que volver a visitar, o el trabajo que necesita ser completado. Usted puede agregar nuevas fichas en la sección de Medio Ambiente del panel de opciones.

Fragmentos de Inserción en el código

Una de las características de Visual Studio genial, al igual que CodeSwap, es la característica Insertar fragmento-tura. Esta herramienta de organización tiene varios fácil de usar, pero difícil de recordar trozos de código en varias categorías, incluyendo los que se muestran en la Figura 19-7.

Para utilizar la función Insertar fragmento de código, simplemente haga clic en el código VB en el Código Ver y seleccione Insertar fragmento. El código está bien factorizado e incluye una gran cantidad de

Tipo de plantilla cosas, también, como un bucle de matriz y algoritmos complejos. Todos el nuevo material en VB 2008 está ahí, también, como LINQ y Windows Presentation Fundación. Si usted tiene un problema difícil de resolver, eche un vistazo a la inserción Fragmento de función.

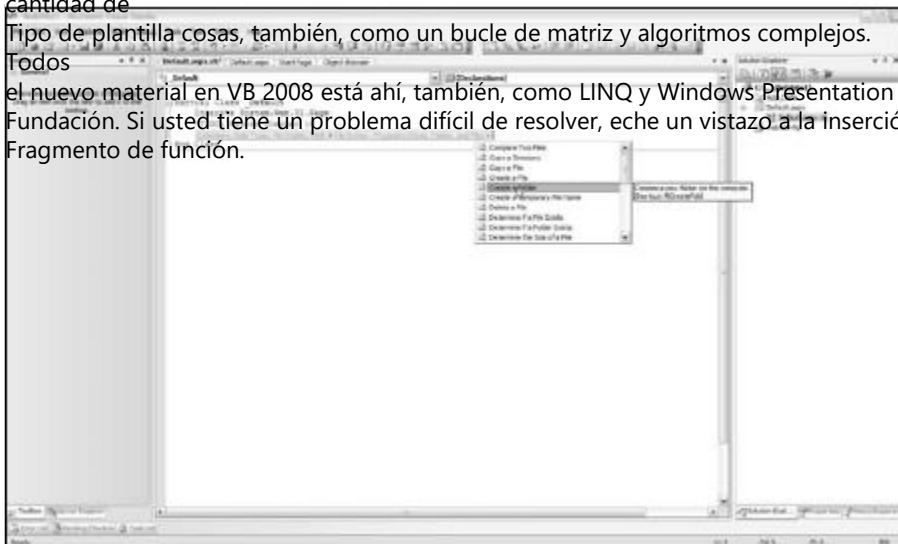


Figura 19-7:
Uso de la
Insertar
Fragmento
característica.

Capítulo 20

Diez ideas para tomar su próxima Programación Paso

En este capítulo

- Hacer nuevos tipos de proyectos
 - Participar en los concursos
 - Probar las ideas de otros
-

Without lugar a dudas, por el tiempo que ha trabajado a través de los ejemplos en este libro, usted será un experto. Expertos brillar por salir en su propio y probar cosas nuevas. Este capítulo tiene que ver con las cosas que usted puede ir a probar en el mundo real después de haber conseguido abajo los fundamentos de Visual Basic y ya está listo para encontrar algunos de los retos de programación.

Cómo Visual Basic 2008 para uso en el hogar

Si se utilizan principalmente Visual Basic para el trabajo, es posible que desee obtener una copia de Visual Basic 2008 Express Edition para uso en el hogar para que pueda divertirse con proyectos fuera de horas. Mientras que usted puede descargar el SDK. NET y utilizar la línea de comandos, le recomiendo usar la Edición Express.

La Edición Express es una copia de Visual Studio 2008 para los aficionados. No se encuentra algunas de las características de la versión profesional que discuten en otra parte de este libro, pero nada significativo se quede fuera. La edición Express es perfecto para algún que otro proyecto de código abierto.

Ediciones Express son - en el momento de escribir estas líneas - libre. Tenga en cuenta que el Professional Edition contiene un montón de cosas que no puedes hacer en Express, razón por la cual elegí profesional para este libro. Sin embargo, es un gran herramienta de aprendizaje.

Actualización de código para utilizar las nuevas herramientas

Esto solía ser una sección sobre cómo escribir sus propias herramientas, y tenía un poco de muestra Generador de código de mi objeto de datos. Honestamente, yo no uso esa herramienta más, porque creo conjuntos de datos y el Diseñador de DataSet Visual son mucho mejores. De hecho, he vuelto a una serie de proyectos antiguos y actualizados a utilizar el diseñador de DataSet.

En sí mismo, esto es un gran consejo. Actualice sus aplicaciones para que utilicen el nuevo herramientas. Si está trabajando en una aplicación basada en gran medida objeto, utilizar LINQ en su código de capa de negocio para obtener los datos correctos en el momento correcto. Si contar con un programa que ha mantenido durante años que notifica gerentes que necesitan para aprobar los documentos, que se actualizará para utilizar Windows Workflow Fundación.

Las nuevas herramientas que ofrece Microsoft son sólo un reflejo de lo que nosotros, el comunidad, han demostrado que es necesario sobre una base regular. No pase por alto ellos. Tabla 20-1 muestra una lista de las cosas nuevas que he actualizado los proyectos con poco, y lo reemplacé. Como siempre, su kilometraje puede variar, pero lo elegir un proyecto y darle una oportunidad.

Tabla 20-1

Actualizar Este

Mano con código de objetos de datos
Lógica para mover información en torno a entre los grupos interesados
Viejo, cansado de Windows Forms código
COM +
Archivos INI
Complejo objeto de asignación de código

Cosas para actualizar

Con este

Conjuntos de datos
Windows Workflow Foundation
Windows Presentation Foundation
Windows Communication Foundation
Los archivos de configuración
Linq

Únase a una competencia en línea en

TopCoder

Otra gran cosa que hacer para estirar las piernas codificación es participar en una competencia en línea. TopCoder (www.topcoder.com) es lo mejor que puedes encontrar - a la libre competencia en la que puede responder a problemas usando VB y competir

Capítulo 20: Diez ideas para tomar su siguiente paso de programación

por la fama, la fortuna y proyectos. Por ejemplo, uno de los problemas más fácil solve consiste en tomar datos sobre los estudiantes y descubrir una manera de extraer el más antiguo estudiante mediante el uso de una clase especificada y el método.

La simple interfaz de usuario Round Match es una de las mejores interfaces de Java que he visto, como se muestra en la Figura 20-1. Usted está en una habitación virtual con otros programadores, que le da un cierto nivel de conocimiento sobre cómo otros están abordando los problemas. Es muy emocionante.

Figura 20-1:
La
TopCoder
arena.

Otro uso popular de TopCoder son sus concursos de desarrollo de componentes. Estos no son tiempos, pero tienen fechas de vencimiento, y si se juzga mejor por sus pares, puede reunir un poco de dinero. Es una buena prueba de habilidades.

Los problemas que encuentre en TopCoder se parecen mucho a los cuestionarios en un programa-ming clase, y eso es una comparación justa. El objetivo es escribir el mejor código que solucione el problema y hacer que su código en comparación con el de otros programadores.

Participar en un proyecto de código abierto

Software de código abierto se rige por un software de una miríada de licencias libres, que las reglas del estado, tales como "Este software es gratuito para uso, pero cualquier

ta a la comunidad. "El código abierto es producto más famoso, Linux, puede obtener toda la prensa, pero un montón de código abierto .NET proyectos están disponibles, también.



Antes de que usted acaba de saltar, sin embargo, echar un vistazo alrededor. Dos de los mejores lugares encontrar proyectos son SourceForge.net (<http://sourceforge.net>) y CodeZone (www.codezone.com / MyCodezone.CodezoneCom). Inicie sesión y mirar a los proyectos. Busca una que te interese, obtener el código y jugar un rato con él. Vea si usted puede hacer mejoras.

A continuación, inicie sesión en el tablón de anuncios y hablar con los diseñadores. Estos deben estar personas con las que podían ir a tomar una copa, ¿sabes? Hable acerca del proyecto. Ver si es algo que te gustaría donar un puñado de horas meses a.

Usted puede recoger un montón de beneficios de trabajar en proyectos de código abierto. En primer lugar, podría ser la única oportunidad que te dan, dependiendo de su situación, para trabajar con expertos. .NET los programadores. En segundo lugar, se obtiene una verdadera oportunidad de practicar codificación en un proyecto revisado por pares. En tercer lugar, usted puede terminar encima de ser capaz de señalar a una aplicación de público que ha participado en la construcción. Este es un gran Resume Builder.

Por encima de todo eso, sin embargo, los proyectos de código abierto son divertidos. Mientras escribo esto, un rápido ver las aplicaciones disponibles en una pocas áreas del proyecto son los siguientes:

- Un proyecto de portal Web
- Documentación de las herramientas XML
- Un marco relacional objeto
- Web de IIS administradores
- Línea de conteo utilidades
- Un proxy HTTP

Esto señala otro de los beneficios - algunos programas realmente fresco está disponible, gratis para la descarga. Todo lo que estamos moralmente obligados a hacer para pagar por ello es ayudar un poco. Es un precio justo, eso es seguro!

Usar Herramientas de terceros en sus proyectos

Cuando usted está buscando para hacer cosas más sofisticadas con VB, mire algunas herramientas de terceros para integrar en sus proyectos. Grandes ejemplos de otros fabricantes herramientas son controles de usuario de formularios Web Forms para proyectos y controles de formulario para Aplicaciones de Windows Forms.

El software creado por terceros individuos es una gran manera de bajo costo

ampliar sus horizontes sin tener que gastar cientos de horas de programación y pruebas. No te dejes llevar por "Not-construido-en este caso el síndrome" - software de terceros tiene sus beneficios.



Capítulo 20: Diez ideas para tomar su siguiente paso de programación

Para empezar, el software de terceros es a menudo revisados por pares. Echa un vistazo a www.windowsforms.net. Los diez primeros clasificados del usuario controles están ahí en la página principal. Búsqueda de una forma que usted pueda necesitar. Todos los controles tienen una descarga y clasificación vista, y muchos de ellos son revisados por los usuarios.

Para obtener los formularios Web Forms, echa un vistazo a www.asp.net. Al igual que con los formularios Windows Forms. NET Sitio Web, los ejemplos de código en la galería de controles de la página ASP.NET son clasificar y contados por otros usuarios.

Los programadores usan software de terceros, y cuando usted sabe su manera alrededor de el idioma, también debería hacerlo. Si usted todavía no está seguro, busque un Microsoft-certificado de proveedor - llamado un ISV (Independent Software Vendor). Mi com- empresa es uno - muchos otros, también. Es la marca de Microsoft para una empresa que sabe lo que se está construyendo.

Para buscar proveedores certificados, echa un vistazo a la guía de recursos en <http://directory.microsoft.com/mprd>. En esta página puede encontrar un socio por especialidad (usted quiere un ISV) y la ubicación (que tal vez no importa para sólo encontrar constructores de control).

También en el ámbito de software de terceros para los programadores son los complementos para Visual Studio. El Programa de Integración de Visual Studio (disponible en <http://msdn.microsoft.com/vstudio/extend>) proporciona a los proveedores con un montón de excelentes herramientas para la integración de sus productos en Visual Studio, y sólo le ayuda a moverse a la derecha a lo largo.

Dotfuscator, que mencioné en la Parte I, es un ejemplo de un complemento en. Se trata de un Aplicación de Windows Forms que se integra perfectamente en Visual Studio ayudarle con las tareas de programación - en este caso, para proteger su fuente código. Usted puede encontrar más información acerca de Dotfuscator en los archivos de Ayuda de Visual Studio.

Tratando de complementos de vendedores es muy parecido a tratar de un control - la investigación, comprobar los exámenes y pruebas, test de prueba,. Y diviértete! La mitad de la razón para probar cosas nuevas es por la emoción.

Integrar un servicio Web Pública

El Capítulo 7 describe cómo crear e integrar un servicio Web, pero la integración

empresas con especial información han facilitado dicha información en los servicios públicos de Web XML para usted pruebe. Usted tiene que pagar para utilizar algunos de estos servicios, mientras que otros son gratuitos. Incluso si es sólo para un proyecto de lado la diversión, no debes dejar de tratar de integrar un servicio público una vez o dos veces.

329

Los servicios públicos de Web están pasando por algo de un cambio en el poder a partir de este por escrito, con las nuevas empresas pequeñas poco a poco siendo reemplazados por la red Internet Monstruos como Amazon y Google. Los principios siguen siendo los mismos, sin embargo.

Al observar esto, tenga en cuenta que usted está saliendo de la RED. mundo en muchos casos. Los servicios Web XML son multiplataforma, así como muchos Perl y Java servicios como servicios ASMX están ahí fuera. Algunos de ellos podrían devolver algo inesperado y, como tal, estropear un buen perfectamente programado. Esté preparado, y con frecuencia los errores de captura.

Uno de los pequeños hosters gratuitos de inicio de servicio que aún está en torno a que no se requieren un montón de registro (como Google y Amazon lo hacen) es WebServicesX. Tiene un número de servicios de muestra buenos que son fáciles de consumir. Voy a empezar con el servicio postal Info Código, para obtener un nombre de ciudad a partir del código postal.

Para utilizar un servicio que usted encuentra en un sitio de servicios de host público, primero asegúrese de que usted está conectado a Internet, y descargar y utilizar el servicio de siguiendo estos pasos:

- 1. Abra Visual Studio y comenzar un nuevo proyecto de aplicación para Windows.**
- 2. En el formulario, agregue un objeto de etiqueta y mantener el nombre predeterminado.**
- 3. Haga clic derecho en el proyecto y seleccione Agregar referencia de servicio.**
- 4. Introduzca la dirección del servicio en el cuadro de texto URL y haga clic en Go botón.**

Por ejemplo, el archivo WSDL para el servicio que uso para este ejemplo está en www.webservicex.net/uszip.asmx?WSDL.

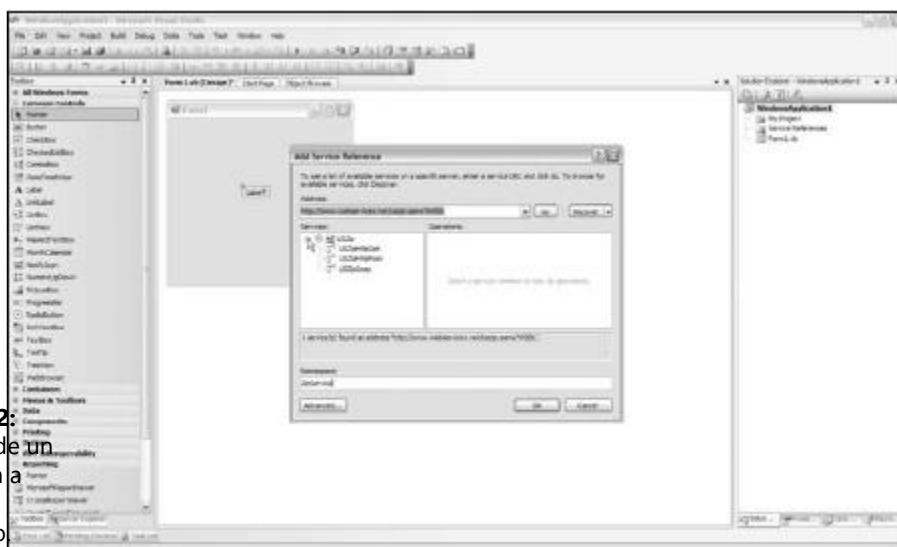


Figura 20-2:

Adición de una referencia a el USZip Código

Capítulo 20: Diez ideas para tomar su siguiente paso de programación

5. Escriba un nombre para el servicio en el cuadro de texto Nombre de referencia Web y haga clic en el botón Agregar referencia.

Por ejemplo, entré en la ZipService nombre.

6. Haga doble clic en el formulario para obtener el Controlador de eventos Form1_Load.

7. Agregar el código para activar el servicio.

Entré en el código siguiente para habilitar USZipCode:

```
Private Sub Form1_Load ( ByVal como System.Object remitente e, ByVal como  
System.EventArgs) Handles MyBase.Load  
MyZipService dévil como ZipService.USZipSoapClient = Nuevo  
ZipService.USZipSoapClient ()  
MyCity dévil como XmlNode =  
myZipService.GetInfoByZIP ("43123").FirstChild  
MyCity.FirstChild.InnerText.ToString = Label1.Text  
End Sub
```

8. Ejecute el código pulsando F5.

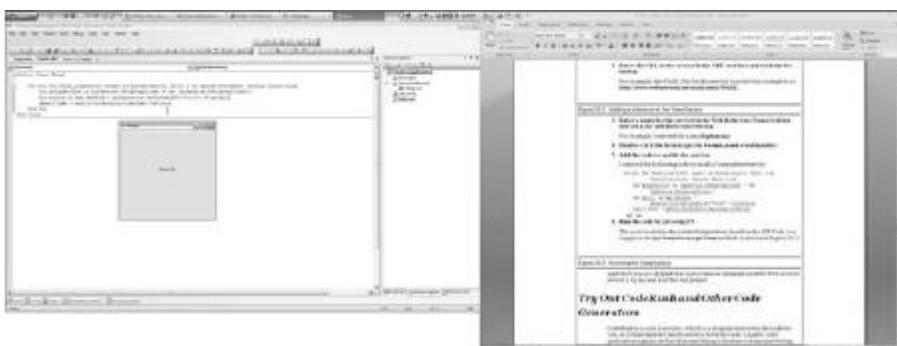
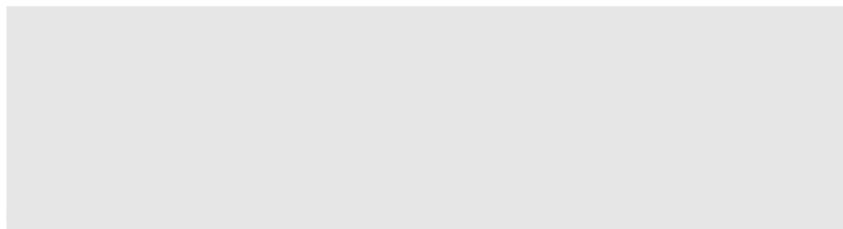
El servicio devuelve la temperatura actual, basado en el código postal que de suministro en el método myTempService.getTemp, como se muestra en la Figura 20-3.

Figura 20-3:
Ejecución de la
USZip
Código.

Y ahí lo tienes - eso es todo lo que se necesita para integrar un servicio Web público! Dar una oportunidad en su siguiente proyecto por diversión.

Pruebe CodeRush y Generadores de código

CodeRush es un generador de código, que es un programa que escribe el código para



programas se basan en un esquema de base de tomar y escribir el acceso código. Algunos generadores de código, sin embargo, hacer oh mucho más.

El objetivo es escribir un mejor código más rápidamente, y disfrutar de ella. CodeRush - y otros programas generadores de código como ella - son una gran manera de empezar. No son un sustituto para saber cómo hacerlo usted mismo, pero son una gran herramienta cuando usted está en el mundo de la producción. Puede encontrar más información sobre CodeRush en www.coderush.com.

Para ser justos, CodeRush es más que un generador de código simple. Por ejemplo, CodeRush tiene las mejores características siguientes:

Le permite ver el código que ha construido en una herramienta visual, incluso si el código es una biblioteca de clases!

Se selecciona de manera inteligente código para usted, en lugar de seleccionar una línea.

Tiene una función de copia muy sofisticado que se asegura de que usted obtenga el bloque de código. Por ejemplo, si se selecciona un bloque try-catch, se copias a la instrucción End Try.

Es, por supuesto, genera un código con un conjunto de plantillas de código grandes.

Cuando usted está creando un proyecto, y estamos en el punto en el que realmente saber lo que está en contra, usted debe buscar en el uso de una herramienta de productividad como CodeRush para ayudarle a código más rápido y con mayor precisión. El uso de un código generador puede mejorar la experiencia de escribir un programa!

Escriba un elemento Web

SharePoint es un servidor de portal de Microsoft para el conjunto de las empresas. Por portal *servidor*, Me refiero a una página de intranet prediseñado que incluye espacio para almacenar documentos, horarios, etc. No se estremecen - en realidad es un muy buen producto, a diferencia de muchos esfuerzos similares.

Elementos web son los pequeños agujeros en los que los documentos y anexos son colocado. Básicamente, en lugar de escribir una página o control de usuario personalizado cada vez necesita una nueva página o control de la intranet, puede invocar una norma Elemento web y configurarlo ligeramente para adaptarse a sus necesidades.

SharePoint viene con 15 o menos elementos web, que incluyen lo siguiente:

Discusión bordo

Fax de control

Documento de almacenamiento

Enlace lista

Help Desk

¿Qué le interesa, sin embargo, como un programador de VB, es la capacidad de escribir un nuevo elemento Web. En total, el proceso de escritura de un elemento Web desde cero es realmente muy complicado. Microsoft hace que sea mucho más fácil, proporcionando una nuevo tipo de proyecto de elementos web - al igual que un proyecto de aplicación para Windows o un proyecto de servicio Web.

La salida de un elemento web es una DLL, al igual que un proyecto de biblioteca de clases (de hecho, es una biblioteca de clases). El truco para conseguir balanceo con un nuevo elemento Web, sin embargo, es descargar el SDK desde el sitio de SharePoint <http://msdn.microsoft.com/sharepoint> y lo instale. Entonces usted tendrá acceso a un nuevo tipo de proyecto cuando abre Visual Studio y seleccione Archivo → New Proyecto.

Un nuevo producto, llamado SharePoint Designer, también está ahí fuera, sino que hace que sea fácil incluso para los usuarios finales para desarrollar elementos web de SharePoint. Es parte de la MSDN Biblioteca, y se puede descargar desde <http://office.microsoft.com/en-us/sharepointdesigner/default.aspx>. SharePoint Designer, que se muestra en la Figura 20-4, es un diseñador HTML de alto poder que tiene todo el sistema de tuberías de SharePoint construido adentro

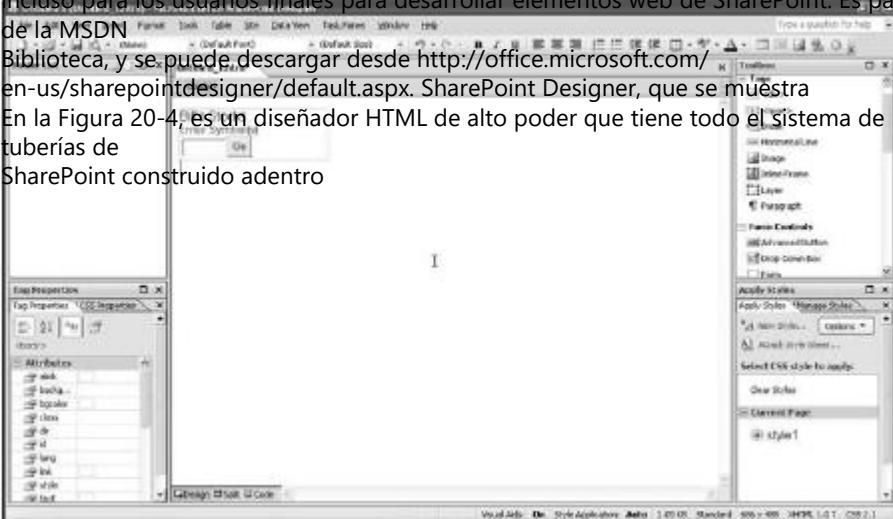


Figura 20-4:
SharePoint
Designer.

Fuera de elementos web son los flujos de trabajo, lo que es más la dirección en la que SharePoint va. Éstas definen el proceso alrededor de la distribución de los información, y se pueden desarrollar utilizando Windows Workflow Foundation.

Para echar un vistazo a un flujo de trabajo de arranque, consulte el Windows Workflow Foundation tipo de proyecto en proyecto crear en Visual Studio.

Utilice el DTE

El DTE (Environment Design Time) es el modelo de objetos de automatización - el biblioteca de clases que se utiliza para crear nuevos complementos y asistentes para Visual Studio sí mismo. Sí, es cierto, estoy recomendando que avanzar en su comprensión ción de Visual Basic mediante la creación de un add-in para Visual Studio sí mismo.

Yo podía entrar en las implicaciones filosóficas de esto, pero lo voy a dejar esto: Cuanto más se trabaja en profundidad con algo, mejor que podemos encontrar en ella.

Para utilizar el DTE, usted necesita comenzar de nuevo un tipo de proyecto nuevo seleccionando Archivo \Rightarrow New Project. En el cuadro de diálogo Nuevo proyecto, busque en los tipos de proyectos lista de la izquierda, haga clic en el signo más situado junto a Otros tipos de proyectos y, a continuación, seleccione Extensibilidad. En el cuadro Plantillas de la derecha, seleccione el complemento de Visual Studio en (que se muestra en la Figura 20-5) y luego haga clic en el botón Aceptar.

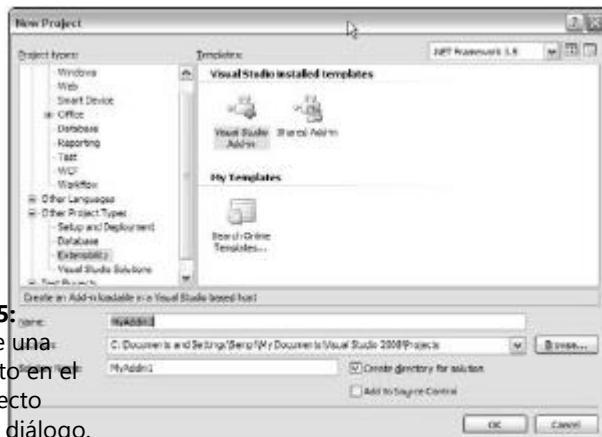


Figura 20-5:
Selección de una complemento en el Nuevo Proyecto
Cuadro de diálogo.

Lo importante de esto es que al igual que el elemento Web de SharePoint, Microsoft tiene hecho toda la plomería para usted. Una gran cantidad de código necesario para presentar una de complemento trabajar, pero gran parte del código es automáticamente suministrada por el tipo de proyecto.

Una vez que haya cargado el complemento de Visual Studio en el proyecto, se tiene acceso a casi todas las partes de un proyecto de Visual Studio. Necesidad de bucle a través de los archivos en un proyecto? No hay problema. ¿Quieres ver una pieza selecta de código? Hecho. ¿Quiere cambiar una línea de texto en un párrafo HTML? Mira esto:

```
MakeParagraph Sub ()  
    DTE.ActiveDocument.Selection.Text = "<P>"  
    DTE.ActiveDocument.Selection.EndOfLine()  
    DTE.ActiveDocument.Selection.Text = "</ P>"  
    DTE.ActiveDocument.Selection.CharRight()  
End Sub
```

Al igual que el generador de objetos de datos, el DTE puede ser un poderoso aliado. Vale la pena dándole un tiro!

Escribir un artículo sobre lo Que has descubierto

Digamos que he vuelto bastante bien informado acerca de Visual Basic, o tal vez que haya descubierto una forma inteligente de solucionar un problema que encontró mientras creando un programa. Especialmente cuando usted averiguar algo sobre su propia la creación de un único bloque de código para resolver un problema, es posible que desee compartir su conocimiento al escribir un artículo.

Un montón de grandes sitios, populares aceptan artículos breves de nuevos autores. No esperar el dinero, pero usted consigue dos cosas importantes si su artículo es aceptado y publicado: el reconocimiento de sus pares y la educación firme que viene con la investigación de algo sólido como para escribir sobre ella. A empezar, echa un vistazo a los siguientes sitios:

[www.dotnetjunkies.com / community.aspx](http://www.dotnetjunkies.com/community.aspx)
www.asptoday.com/info.aspx?view=WriteForUs
www.4guysfromrolla.com/authors.shtml
www.asp101.com/resources/submit.asp

Si estás interesado en escribir un artículo, escribir! No inventes excusas como eres demasiado tímido o desconocido para publicar un artículo. En 2001, escribí un artículo para Intranet Journal que aún está (www.intranetjournal.com/articles/200107/ia_07_25_01a.html), y yo no podía creer que el furor se inició. Ahora mira donde estoy!

Capítulo 21

Diez Recursos en la Internet

En este capítulo

Investigar con los sitios Web

Compartir ideas con las comunidades en línea

Encontrar código de ejemplo

W sombrero haríamos sin el Internet para recurrir? En este capítulo, mostrarle mis sitios favoritos en línea para obtener información acerca de Visual Básica. Usted puede encontrar una gran cantidad de basura que hay en Internet, sino que también puedes encontrar un montón de cosas buenas, también. El truco consiste en decir los dos separados.

MSDN Library

<http://msdn.microsoft.com/library>

La biblioteca MSDN es sin lugar a dudas el principal recurso para .NET investigación. MSDN significa Microsoft Developer Network, y el nombre de MSDN *Biblioteca* adapte el sitio muy bien, es efectivamente un gran almacenamiento de documentos en línea ubicación.

La función de búsqueda del sitio funciona muy bien (escriba su codiciado en el plazo Cuadro de texto Buscar y haga clic en el botón Go), pero no hay que ignorar a sólo cejas-ING a través del sitio. El árbol de la izquierda tiene un diseño muy inteligente tabla de contenidos incorporados, y usted puede encontrar un montón de código de ejemplo y recursos con sólo escarbar. Tenga cuidado, sin embargo, el contenido incluye algunas cosas más, y quiere concentrarse en VB 2008 documentos.

VBForDummies.net

www.vbfordummies.net

VBForDummies.net es mi sitio. Mi meta para el uso de este tipo de sitio es publicar bits del libro, así como otros documentos y códigos. Estoy usando Graffiti, el

ASP.NET sistema de gestión de contenido, para construir el sitio. Permite para el registro de participación de la comunidad que usted está probablemente acostumbrado a estas alturas, por ejemplo, comentarios a mi blog y los foros para hacer preguntas.

Canal 9

<http://channel9.msdn.com>

Una extraordinaria plataforma de debate, Canal 9 responde a todas las preguntas difíciles - ciones sobre el funcionamiento interno de Microsoft. Todos los geeks de programación en la com - empresa pasar el rato allí y las discusiones son francas y esclarecedor.

Además, usted encontrará una gran cantidad de características que reemplazan el GotDotNet ahora - difunto, como el SandBox, donde se puede encontrar muestras de código de usuario a la vez simple e imposiblemente compleja.

Mi parte favorita del sitio, sin embargo, es la entrevista de tarjetas azules. El sitio era originalmente tripulado por el extraordinario blogger Robert Scoble, y el sentimiento en el sitio es como si nunca hubiera salido. Las entrevistas están relajados y natural, y tienen un montón de gran información.

ASP.NET Web

www.asp.net

Al igual que con GotDotNet, la comunidad ASP.NET fue iniciado por un equipo de Microsoft - pero esta vez fue el equipo de ASP.NET. ASP.NET Web es el lugar para ir a cualquier descarga, control o pregunta sobre el desarrollo Web Forms.

<http://quickstarts.asp.net/QuickStartv20/default.aspx>

También - no te lo pierdas - el sitio ASP.NET es el lugar de alojamiento de la QuickStart Tutorials. Es posible que estos tutoriales son similares a este libro, con explicaciones a la de punto y un montón de ejemplos. El QuickStart Los tutoriales son una solución impulsada, también, para que responda real y no teórica preguntas. Los ejemplos de código se basa en el modelo de línea de scripting, que yo no usar en este libro. Este modelo ofrece otra manera de construir ASP.NET Web Formularios proyectos. Usted puede coger a este modelo con bastante rapidez al ver los ejemplos en el Inicio rápido.

Los grupos de noticias públicos de Microsoft

<http://msdn.microsoft.com/newsroups/managed>

Yo era un gran fan de la Usenet, pero con la proliferación de los spammers, realmente ha ido cuesta abajo. Microsoft ha hecho un gran esfuerzo, sin embargo, para obtener

Usenet comenzó de nuevo para los desarrolladores de Microsoft mediante el Managed Modelo de grupos de noticias. Si usted tiene una suscripción a MSDN y se han inscrito con el servicio de grupos de noticias administrados, se obtiene una respuesta a cualquier cuestión enviada

ción dentro de las 48 horas (o dos días hábiles) de un empleado de Microsoft o MVP. Es un gran plan, y es gratis con la suscripción MSDN.

.NET 247

www.dotnet247.com/247reference/default.aspx

Lo admito - Yo voy a promover otro autor aquí. Matt Reynolds es un fantástico escritor y programador, y se ha puesto un recurso fantástico. .NET 247.

Este sitio se ve a través de los mensajes de Usenet que en realidad tienen algo interesantes en ellos, y entonces los índices de estos puestos para que la información útil. Le recomiendo que tome ventaja de los contenidos de .NET 247.

Google CodeSearch

[www.google.com / codesearch](http://www.google.com/codesearch)

Aunque gran parte del repositorio de Google Code no es código de Microsoft, sigue siendo un tremadamente valioso recurso. Además de la búsqueda de frases de tono (Que es tremadamente divertido), se pueden encontrar algoritmos que pueden ser fácilmente traducido a VB con un poco de conocimiento.

Las expresiones regulares de Perl fuera son muy útiles, por ejemplo. Perl es un lenguaje que hace un uso intensivo de las expresiones regulares, y nuestra. NET Regular Expresiones son similares. Con criterios de búsqueda bien, es fácil de encontrar Perl regular expresiones que las cosas que pueda necesitar partido - la búsqueda "Correo Regex" por ejemplo.

kbAlertz

www.kbalertz.com

¿Cuál es kbAlertz.com? El sitio kbAlertz lo dice mejor:

"KbAlertz.com es un sistema de notificación por correo electrónico que escanea todo el Microsoft Knowledge Base todas las noches, y e-mail cuando haya actualizaciones o adiciones se hacen a las tecnologías que está suscrito. Como nos escanear la base de conocimientos de todo, también tenemos un registro bastante bueno sistema para su uso en el menú de la izquierda."

Usando kbAlertz no se trata de VB 2008 como mucho, ya que se trata de tener acceso a información que es útil para cualquier persona en un entorno Windows. Me suscribo para las alertas y obtener información sobre los cambios en las mejores prácticas de seguridad, entre otras cosas.

CodePlex

www.codeplex.com

Hablando de código fuente abierto (? Si no nosotros), no pase por alto CodePlex - Nuevo Código Abierto de Microsoft System Management. Se trata de una biblioteca de todos los .NET de los proyectos de código abierto que puede unirse, usar o navegar.

Yo uso los proyectos de código abierto como mi propio medio de educación personal. A lo largo de mi carrera, he visto cómo otras personas han hecho las cosas, tanto buenas y lo malo, y formulado mis propios planes. CodePlex es un gran lugar para hacer eso.

Es también un montón de diversión. En estos momentos, el proyecto más popular es un conductor. NET para los nunchakus de la Wii. Cool stuff.

<Microsoft> Google para la búsqueda

www.google.com / microsoft

Con todos estos sitios especiales para encontrar y compartir código, no se olvide sobre el bueno de Google. El truco para usar Google para encontrar los bits útiles de código es perfeccionar su técnica de búsqueda. Busco para la clase exacta que necesita el código de y añadir el término VB.NET, y por lo general encontrar lo que necesito. Por ejemplo, necesitaba

para conectarse a una hoja de cálculo de Excel como fuente de datos recientemente. Yo podría tener reconstruido el código necesario de la documentación de MSDN, pero en su lugar Me volví a Google. Busqué con la frase OleDbConnection "vb.net Excel "y tiene cuatro ejemplos como los primeros cuatro hits.



Ahora, para hacer una búsqueda adecuada, es necesario conocer un poco acerca de lo que son buscando. Así que no se apagan con el seguro echado, o sus resultados serán demasiado amplia. Para acotar la búsqueda y obtener buenos resultados en Google, utilice uno muy específico plazo (como OleDbConnection) con uno o más términos generales (como vb.net y Excel).

Índice

Requisitos para el,
51-52
capas de, 44

• Símbolos •

- ... (Puntos suspensivos) botón, en una propiedad, 28
- / (División) del operador, 170
- (Resta) operador, 170
- + (Suma) operador, 170, 227-228
- * (Asterisco), junto a un nombre de archivo
 - en vista de diseño, 24
- % (Módulo) del operador, 170
- * (Multiplicación) del operador, 170
- _ (Guion bajo),
 - a partir variables privadas, 114-115

• A •

- conceptos abstractos, comparando con el mundo real, 48-49
- entorno abstracto, manejo de errores en, 205-206
- abstracción, contenido por código reutilizable, 198
- acceder a un recurso, implementación de rosca favor, 233 en
- AccessControl espacio de nombres, 251
- palabras clave de accesibilidad, 119
- AccessRule clase, 251
- Adaptador de clases, 257
- adaptadores
 - la creación y uso, 256
 - base de datos específica, 254
- Agregar un nuevo origen de datos botón, 33
- Agregar cuadro de diálogo
- Conexión, de 32 años, 259-260
- añadir la función, 226
- complementos
 - describe, 41
 - incluido con Visual Studio 2008, 41-42
 - para Microsoft Office aplicaciones, 21
- Agregar método, con fecha y hora, 171-172
- Añadir un nuevo enlace de origen de datos, 258-259
- Agregar referencia cuadro de diálogo, 116-117, 207-208, 212
- Agregar referencia de panel, 269, 270
- Cuadro de diálogo Agregar referencia de servicio caja, 132-133, 330
- Añadida la opción Tab, en el Cuadro de herramientas, 27
- Add Watch, seleccionando en el Código
 - Ver, 144
- Además operador (+), 170, 227-228
- ADO.NET
 - búsquedas de recolección, 55
 - en comparación con LINQ, 270
 - describe, 47
 - aplicación de, 254
- Los usuarios afectados, con el miedo, 242
- Alinear todos los controles Center opción, el 71
- Alinear característica, en el formato menú, 71
- Lista alfabética de productos vista, 34
- Alt + F8, 319-320
- Ficha del ancla, en vistas de origen, 103
- AppActivate función, 217-218
- archivo app.config, 293-295
- Anexar la función, en la StringBuilder clase, 168
- Estado de la aplicación, en Trace, 151
- ApplicationException
 - lanzando, 206
 - usando, 184
- aplicaciones
 - asegurar adecuadamente, 293
 - construcción para Windows, 65-68
 - compilar en Windows
 - programas, 19
 - la conexión a una base de datos fuente, 32
- creando en Visual Studio 2008, 12
- diseño
 - pantallas para, 54-56
 - en Visual Basic 2008, 43-58
- en desarrollo .NET, 11

- correr, 19-20, 70
- estado de, 85-86
- almacenar los datos para, 53-54
- redacción de planes de prueba para, 57-58
- Parámetro AppWinStyle, de
 - el comando Shell, 217
- arquitectura de la neutralidad, de la Web
 - servicios, 125
- Clase Array, métodos, 228
- artículos, escritos, 335
- Ascx, 209-210
- Haz una pregunta botón, el
 - el explorador de documentos
 - barra de herramientas, 39
- ASMX servicio basado, consiguiendo
 - WSDL, 136
- ASP.NET
 - que acompaña a Web aplicaciones, 90
 - aplicaciones, la edición, el 91
 - ejemplos de código, 329
 - en comparación con Windows Forms, 84
 - limitaciones, 84
 - descripción, 83-84, 93
 - permitiendo la interacción con el usuario, 48
 - encapsulando CGI
 - funcionalidad, 97-102
 - guardar la aplicación Estado, 86
 - controles de servidor, 89
 - controles de usuario. Vea usuario
 - controles
 - como orientada a la interfaz de usuario, 47
 - Web Forms espacios de nombres, 88-89
- ASP.NET 2.0 Para Dummies* (Hatfield), 39, 93, 106
- Seguimiento de ASP.NET en MSDN
 - Biblioteca, 152
- ASP.NET Web
 - información general, 338
 - Servicio plantilla, 128
 - Plantilla del sitio, 90-91
 - archivo aspnet_wp.exe, 154
 - archivo aspx.vb, 90
 - . Aspx.vb archivo, 90
 - asamblieas, el control de acceso dentro, 203
 - supuestos, software matando proyectos, 57

asterisco (*), junto a un nombre de archivo
 en vista de diseño, 24
 Asociar al proceso de diálogo caja, 154
 AuditRule clase, 251
 autenticación define, 239, 243
 con inicio de sesión de Windows, 243-245
 Clase de autenticación, en System.Net, 285
 autorización, que se define, 239, 243
 Clase de autorización, en System.Net, 285
 Autorización de espacio de nombres, 251
 Autoformato cuadro de diálogo, en la Web
 Formas, 92
 sistemas automatizados de prueba, 58

• B •

Bibliotecas de clases base (BCL)
 acceder, 46
 tubería código precompilado, 47-48
 piezas que el usuario la interacción de, 54
 datos de código básico, 267-268
 Programación básica idioma, 14
 BCL (Bibliotecas de clase base)
 acceder, 46
 tubería código precompilado, 47-48
 piezas que el usuario la interacción de, 54
 las mejores prácticas, para garantizar Web
 Formularios, 250
 carpeta bin, 66
 BinaryReader clase, 272
 Clase BinaryWriter, 272
 BindingNavigator objeto, 264. Ver también VCR Bar
 Objeto BindingSource, 264
 Objeto Bitmap, 300
 mapas de bits, carga, 300-301
 problema de recuadro negro, 255
 operación de bloqueo, 232
 entrevistas tarjeta azul, 338
 libro
 aproximadamente, 2
 supuestos autores, 4
 convenciones utilizadas en, 2-3
 direcciones futuras, 6-7
 iconos utilizados en, 6
 organización de, 4-5
 en la programación general estrategia, 1-2

piezas que se pueden omitir, 3
 Declaración booleana, 180
 Valor booleano, echando a, 161
 caja con esquinas redondeadas, en un diagrama de flujo, 176-177
 ramificación, 178
 El modo de interrupción, 149. Véase también Depurar el modo
 Breakpoint Properties, 143
 puntos de interrupción
 crear, 141
 descrito, 140-143
 gestión, 142-143
 creación, 141-142
 Puntos de interrupción ventana, 142-143
 navegadores
 dando un enfoque, 218
 hacer peticiones a la servidor, 85
 cepillos
 pintar el interior de polígonos, 299
 tipos de, 299
 utilizado por los gráficos objetos, 298
 insectos, descubriendo mientras planificación, 50
 Construir ícono, al arrastrar a un barra de herramientas, 40
 Construir barra de herramientas, 316
 edificio
 bibliotecas de clases, 107-121
 Fecha de aplicación Calculadora, 89-95, 114-115
 DateCalc servicio Web, 129-130
 función de correo electrónico, 291
 funciones con la reutilización en la mente, 198-200
 genéricos, 230-231
 Google Search Tool, 213-215
 Hello World aplicación, 16-17
 informes, 301
 reutilización de código para construir software, 197-203
 funciones compartidas, 119-120
 URLs de las páginas ASP.NET, 165
 experiencias de usuario, 14
 formulario de entrada de usuario, 65-68
 Aplicaciones web, 89-95
 Aplicaciones de formularios Web Forms, 247-250
 Servicios Web, 127-132
 Aplicaciones de Windows Forms, 17-18, 242-246, 286-296
 De seguridad de Windows aplicación, 243-245
 funciones integradas, cambiando, 227-228
 listas de viñetas, ámbitos de escritura, 51
 aplicaciones de negocio, creación de de, 253

la lógica de negocio
define, 126
diagrama de flujo que representa con componentes, 176-179
diseño, 175-176
entrar en los controles, 68-70
organizando para evitar excepciones a clase
archivos, 206
proporcionar otras aplicaciones acceder a, 132
la separación de la base de datos, 53
escribir en un archivo DLL independiente archivo, 109

Capa de lógica de negocios
en un sistema de n-tier, 53
diseño, 56
modelo de negocio, 200
clic de botón, que se describe, 222

Botón de control
añadiendo en los formularios Web Forms, 95
describe, 64
seleccionar de la caja de herramientas, 16
para aplicaciones Web, 86

Objeto Button, eventos expuestos por, 313-314
botones, se mueve sobre una forma, 16-17

ByRef (por referencia) palabras clave, 204

Tipo Byte, 159

ByVal (en valor) de palabras clave, 204-205

• C •

Función de caché, en System.Net, 286
calc.exe, 216
Calcular método, llamado, 118
Calendario de control
formato de los formularios Web Forms, 92
para aplicaciones Web, 87
Función CallByName, 213
llamadas, hablador paliza grueso, 125
fundición, 161
Catch, 147, 184, 185
categorías, de servidor Web
controles, 88
Botón de categorías, en el Ventana Propiedades, 28
CBool, 161
CDate, 161
CDbl, 162
Centro opción Formulario, 160

Índice 345

- proveedores certificados, buscando, 329
- CGI (Common Gateway Interface)
 - ASP.NET encapsulación, 97-102
 - protocolo, 83
- Cambiar origen de datos cuadro de diálogo, 260-261
- Canal 9, 338
- Tipo Char, 159
- caracteres, sustituyendo utilizando un patrón, 169
- llamadas hablador, en comparación con
 - 125 grueso,
 - Cheat Sheet, en el frente del este libro, 96
- hijo interfaz mesa, 264-265
- Elegir la conexión de datos pantalla, 259-260
- Elija los objetos de base de datos pantalla, 261
- llamadas gruesos, en comparación con
 - hablador, 125
- CInt, 161
- CipherAlgorithmType clase, 251
- Clase (s)
 - descrito, 48-49, 110
 - sosteniendo reutilizable funciones, 202
 - compartida, 272-273
 - utilizando eficazmente, 220-222 en un archivo. vb, 110-111
 - definición de clase, en una clase biblioteca, 112
- Clase barra de herramientas de diseño, 316
- archivos de clase
 - reutilización extiende, 201-203 establecer y utilizar, 201-202
- bibliotecas de clases
 - edificio, 107-121
 - codificación, 112-113
 - en comparación con los servicios Web,
 - 125, 132
 - crear, 113-118
 - depuración, 152-153
 - define, 107
 - diseño, 109-113
 - encapsulando
 - funcionalidad, 109
 - partes de, 110-113
 - proporcionando compartido funciones, 109
- Haga clic en controlador de eventos, añadiendo en
 - Los formularios Web Forms, 95
- ClickOnce, la implementación de un aplicación utilizando, 246
- validación en el cliente, no dependiendo de, 250
- clientes en el mundo. NET, 44
- COBL, 162
- código
 - añadiendo a un menú, 79
 - para las aplicaciones ASP.NET, 84
 - arrastrando hacia la Caja de herramientas para reutilización, 27
 - ejecución de una línea a la tiempo, 149
 - fragmentos de inserción en, 323
 - avances de línea, 3 en
 - recorriendo, 188-189
 - movimiento repetido, 198
 - salida por herramientas visuales, 266-267
 - reutilización de construir software, 197-203
 - ingeniería inversa a Visual Básico, de 42
 - muestras en Visual Basic para Dummies sitio Web, 7
- actualizar a utilizar nuevas herramientas, 326
- trabajando en Visual Studio, 36-39
- escrito
 - acceder a los datos, 266-269
 - 163 funcional,
 - segura, 239-251, 293
- abstracción del código, la baja de, 205
- basado en la codificación objetos, añadiendo en la Bandeja de componentes, 264
- código trampa complejidad, evitando, 203-206
- código encapsulado
 - ejemplo de, 200
 - utilizando bibliotecas de clases, 107
- generadores de código, 23, 331
- un código orientado a los aspectos, de System.Net, 286
- ejemplos de código, para este libro, 7
- Ver código
 - generación de controladores de eventos de, 313-314
 - IntelliSense en, 224
 - abrir un archivo en, 30
 - fichas, 24
 - trabajando en, 36
- Código ventana Vista
 - entrando, 18
 - apertura, 68
- CodeAccessPermission clase, 251
- Archivo CodeBehind en ASP.NET, 90
- CodePlex, 340
- CodeRush, 331-332
- CodeZone, 328
- codificación, bibliotecas de clases, 112-113
- colecciones, 191-192, 256

COM +, puesta al día, 326
COM (Component Object
 Modelo)
Ficha COM, en el complemento
 Referencia del cuadro de diálogo, 212
aplicación en Visual Basic,
 211-212
objetos, métodos de llamada
 en, 213
referencia, 212-213
Lista delimitada por comas, de los valores,
 183
línea de comandos de los programas,
 corriendo con Shell,
 216-217
Common Gateway Interface
 (CGI)
 ASP.NET encapsulación,
 97-102
 protocolo, 83
Control CommonDialog, 273
la comunicación, el programa de flujo
 para el usuario, 176-178
Función de comparación, 111
compilación, las aplicaciones en
 Programas de Windows, 19
complejidad, 203
Componente de Desarrollo
 Concursos, 327
Método de componentes, con
 DateTime, 172
Component Object Model
 (COM)
Ficha COM, en el complemento
 Referencia del cuadro de diálogo, 212
aplicación en Visual Basic,
 211-212
objetos, métodos de llamada
 en, 213
referencia, 212-213
Componente de bandeja
 adición de componentes, 79-80
controles, 274
describe, 317
componentes no visuales en, 318
componentes
 en los orígenes de datos
 ventana, 34
descomposición en
 funciones, 241
 de programas, 240-241
proceso informático, la toma de
 situaciones de toma, 179
concatenación de cadenas, 161
Columna Condición, en el
 Puntos de interrupción ventana, 142
Los archivos de configuración, 326
archivos de configuración, 293
Función de configuración, en
 System.Net, 286

conexión, creando a una base de datos, 262
cadena de conexión, ahorrando a la configuración de la aplicación archivo, 261
Consola tipo de proyecto, la escritura aplicaciones de línea de comandos usando, 216
servicios de consumo, Web, 123, 132-135
contenedores base de datos-generic, 254 el almacenamiento de datos en, 253
control de contenido en las páginas ASPX, 210 marcador de posición de contenido, en ASPX , páginas 210
ContentPlaceHolder control, 210-211
Sumario botón, en la Document Explorer barra de herramientas, 39
menús contextuales, activar, 80-81
Ayuda sensible al contexto, lanzamiento, 38-39
ContextMenuStrip constructor, en el diseñador de formularios, el 81
ContextMenuStrip objeto, la activación de un clic derecho 80-81
ContextMenuStrip propiedad, de una forma, 80-81
Control de colección, 192-193
estados de flujo de control, 181
Control de Galería, de la ASP.NET sitio, 329
estructuras de control, 187
Control para validar los parámetros, 100
Estructura del Control, en el seguimiento, 151
ControlCollection colección, 191-192
controles accediendo desde Toolbox, 26-27 agregar propiedades adicionales para el 78 la creación personalizada, 207-209 doble clic para agregar código, 118 la edición de las propiedades de, 27-29 para formularios, 64 incluidas en Visual Studio, 63 utilizando, 63 para aplicaciones web, 86-89 convenciones utilizadas en el libro, 2-3

Método de conversión, con DateTime, 172
Clase Cookie, en System.Net, 285
galletas, utilizando temporal, 250 galletas colección, disponible a JavaScript, 249
contando el número de horas, 188-189
Botón Crear nueva tarea de usuario, 322
CreateGraphics método, de el control de formulario, 305
Función CreateObject, 212 creando adaptadores, 256 add-in para Visual Studio 2008, 334
aplicaciones en Visual Studio 2008, 12 puntos de corte, 141
aplicaciones de negocio, creación de, 253
bibliotecas de clases, 113-118 componentes reutilizables, 207 la conexión a una base de datos, 262 Crear tarea Eres nuevo botón, 322
CreateGraphics método, de el control de formulario, 305
Función CreateObject, 212 controles personalizados, 207-209 datos corredor, 268 Archivo DISCO, 136-137 DLL, 114
Form_Load evento controlador, 146 funciones o subrutinas, 198 reutilizable, imágenes, 297-307 carpeta obj, en Visual Studio, 66 Las órdenes Detalle formulario de datos, 263-264 proyecto de fuente de datos, 262 plantilla de proyecto, en Visual Studio, 128 propiedades de una biblioteca de clases, 114-115 Referencias carpeta, invisual Estudio, 66 reutilización, 197-198, 207 controles de servidor, 207-209 modelo de amenaza, 241 Evento ValueChanged manejador, 69 Aplicaciones Web en Visual Studio 2008, 20 Objeto WebResponse, de miRequest método, 288 Windows Forms programa para incorporar a Google Herramienta de búsqueda, 215 Credenciales de propiedad, de FtpWebRequest, 288

tableros de cribbage, 302-303,
305-307
cribbage juego de cartas, 302-303
Clase CryptoConfig, 251
Criptografía de espacio de nombres,
251

Crystal Reports Services, en
Explorador de servidores, 31
CStr cadena de conversión, 102,
161
Ctrl + Alt + K, 322
Ctrl + Alt + L, 246
Ctrl + Alt + S, 30
Ctrl + Alt + X, 27
Función CType, los tipos de cambio
con, 159-162
controles personalizados, crear,
207-209
personalización, Visual Studio
2008, 35

• D •

Posibles daños en temor,
242
datos
acceder, 256-257
editar dinámicamente, 33-35
recoger un flujo de, 289
sostiene en recipientes, 253
almacenamiento para aplicaciones, 53-54
tomar de una base de datos, 47
datos de proceso de acceso,
piezas a medida de,
268-269
aplicaciones de datos, el desarrollo de
rápidamente, 34
datos corredores, 268-269
clases de datos, 256
código de datos, escritura, 266-269
Nodo Conexiones de datos, en
Explorador de servidores, 31-32
Conexiones de datos panel, 262
contenedores de datos, 256
datos de los controles
integración con, 254
para aplicaciones Web, 88
Data Encryption Standard
(DES), 245
Capa de Datos, en una n-tier
sistema, 53
Generador de datos de objetos, 326
Objetos de datos, selección 261
Data Source Configuration
Mago
acceder, 257-262
Tipos de datos de origen, 258-259
utilizando, 33-34

Índice 347

- fuentes de datos, conectar, 33, 254, 257-262
- Orígenes de datos del panel
 - el acceso a la fuente de datos
 - Asistente para configuración de 257-262
 - hacer clic en un cuadro en, 263
 - contenido de, 262
 - apertura, 258
- Ventana Orígenes de datos, 33-35
- de datos, tipos de herramientas, 253
- conectividad de base de datos, herramientas
 - para, 253
- Database Management System, en un sistema de n-tier, 53
- nulos de base de datos, 166
- bases de datos
 - la conexión a sin utilizar
 - herramientas visuales, 268
 - interacción con, 47
 - mira, 31
 - caminando a través de los elementos de, 264, 265
- Objeto DataContext, consiguiendo datos de una base de datos, 270
- Control DataGridView, 64
- DataReader contenedor, 256
- DataRow contenedor, 256
- DataSet contenedor, 256
- Diseñador de DataSet, actualización proyectos para usar, 326
- Objeto DataSet, añadiendo, 264
- DataTable contenedor, 256
- DataView contenedor, 256
- Fecha de aplicación Calculadora
 - la adición de código para el 69
 - edificio, 89-95, 114-115
 - lógica para definir, 56-57
 - diseño, 52-58
 - ejemplo declaración del alcance, de 51
 - Requisitos para el, 51-52
 - en Internet Explorer, 96-97
 - decisiones adecuado para reutilización, 200
 - diseño de la pantalla, 54-56
 - almacenar los datos para, 53-54
 - formulario de entrada de usuario, 65-68
 - como una aplicación de formularios Web Forms, 86, 87
 - en los servicios Web, 135
 - como una aplicación de Windows, 63
 - escribir un plan de pruebas para, 57-58
- Conocer la forma calculadora, la colocación de un
 - subrutina pública en el interior, 199
- selectores de fecha, 54-55
- DateCalc. Asmx, 130-131
- DateCalc servicio Web
 - añadir una referencia a, 132-133
 - edificio, 129-130
 - ver en acción, 130-132
- Objeto DateChooser, añadiendo un valor, 69
- fechas
 - almacenar en variables, 158
 - representaciones de cadena de, 172
 - trabajar con, 171-173
- DateSpan tipo de datos, devolviendo un objeto de, 200
- Tipos DateTime
 - describe, 159
 - en Visual Basic, 171
- Variable DateTime, para obtener el fecha actual en, 172
- DateTextFieldProvider
 - herramienta, 172-173
- DateTimePicker control, 64
- Clase Debug, 146
- Depurar el modo
 - colocar en Visual Studio, 140
 - ejecutar aplicaciones web en, 95-97
- Depurar barra de herramientas, 316
- depuración
 - bibliotecas de clases, 152-153
 - definidos, 139
 - funciones de edición en la
 - Puntos de interrupción ventana, 142
 - problemas de tráfico de red, 293
 - proyectos, 148-154
 - remoto, 150
 - herramientas. NET Framework, 145-154
 - herramientas visuales para la depuración, 139-145
 - Web Forms, 95-96, 150-152
 - Servicios Web, 153-154
 - Aplicaciones de Windows Forms, 149
- Depuración Desplegado
 - ASP.NET
 - Las solicitudes, en el MSDN Library, 154
- Depuración de diálogo no habilitada caja, en los formularios Web Forms, 95, 96
- Método Debug.Print, en la Ventana Inmediato, 145
- Debug.Write declaración, insertando, 146
- decimales, utilizando el doble tipo, 170
- componentes de decisión
 - características de, 179
 - en un diagrama de flujo, 176-179
- la toma de decisiones, en la
 - Método PaintBoard, 307

toma de decisiones herramientas, 175
nodo de decisión, en un diagrama de flujo,
179

decisiones, descritos, 187
predeterminado archivo app.config,
293-294

forma predeterminada
el cambio de nombre en la Solución
Explorador, 66

cambio de tamaño en Visual Studio, 66
en Visual Basic, 16
espacios de nombres por defecto, en la web
servicios, 130

Default.aspx código, HTML
para, 93-94

Denegación de servicio, con calma,
242

despliegue, ASP.NET, 91
despliegue de seguridad, 246

DES (Data Encryption
Estándar), 245

Propiedad Description, para un
Servicios Web, 130

DESCryptoServiceProvider
clase, 245, 251

diseño de un proyecto, 51

documento de diseño, 52

problema de diseño, sobrecargas como,
228-229

estrategia de diseño, la planificación de
Servicios Web, 126-127

tiempo de diseño, creación de texto en el 76

Diseño Time Environment
(DTE), 334-335

Vista Diseño

acceder a los formularios Web Forms, 91

descrito, 24-26

arrastrando a un campo en blanco

forma, el 34

generación de controladores de eventos
de, 311-313

menú principal, 79

abrir un archivo en, 30

fichas, 24-26

en formularios Web Forms, 89

trabajando con evento

manipuladores, 222

fichas de diseño, 24

ventana del diseñador, de 24 años

diseño

aplicaciones, 52-58

aplicaciones en Visual Basic

2008, 43-58

la lógica de negocio, 175-176

Capa de lógica de negocios, 56

bibliotecas de clases, 109-113

Diseñador de DataSet, actualización
proyectos para usar, 326

Fecha de aplicación Calculadora,

52-58

145-154
Papel en Visual Basic, 11-13

- diseño (continuación)
 - define, 49
 - describe, 43
 - en proceso de desarrollo, 50
 - del documento, de 52
 - formar diseñador con ContextMenuStrip
 - constructor, el 81
 - para genéricos, 231-232
 - por sobrecarga, 228-229
 - de un proyecto, 51
 - pantallas para aplicaciones, 54-56
 - software seguro, 240-242
 - ajuste de texto en tiempo de, 76
 - software, 43, 240-242
 - estrategia, la planificación para la Web
 - servicios, 126-127
 - para roscar, 232-233
 - Objetos ToolTip, 75
 - interfaz de usuario, 55
 - variables, en tiempo de ejecución en comparación a tiempo de diseño, 76
 - Botón Ver Designer, en El Explorador de soluciones, 30
 - Diseñador web, diseño en el 91
 - Web Forms diseños, en comparación con Windows Formas, 84
 - Servicios Web, 126-127
 - Detalles de vista, el cambio de la Las órdenes de tabla para, 263
 - desarrollo de procesos, planificación y diseño en, 50
 - desarrollo del servidor (s), gestión, 31
 - Barra de herramientas del dispositivo, 316
 - Ver diagrama
 - descrito, 34-35
 - abrir un archivo en, 30
 - Controles de diálogo, 273
 - diamantes, en un diagrama de flujo, 176-177, 179
 - Instrucciones Dim
 - describe, 158
 - objetos de acotación, 221
 - memoria utilizada por, 220
 - directorios, listado, 279
 - Clase Directory, 272-273, 279
 - Clase DirectoryInfo, 272, 279-280
 - Archivo DISCO, crear, 136-137
 - Descubrimiento, en DREAD, 242
 - descubrimiento
 - define, 125, 136
 - proceso de, 136-137
 - Disco de sistema operativo (DOS), 216-218
 - mostrar código, en los formularios Web Forms, 89
 - división (/) del operador, 170
- DLL (Librería de Enlace Dinámico)
 - archivos
 - crear, 114
 - impacto de la sobrecarga en escrito, 228
 - funcionamiento, 116-118
- DLLs (Librerías de Enlace Dinámico)
 - en comparación con los servicios Web, 123-124, 126
 - definido, 108-109
 - descrito, 118-121
 - como una especie de biblioteca de clases, 109
 - como la salida de Web Las piezas, 333
 - como stateful, 126
 - usos de, 109
- DNS (Servicio de Nombres de Dominio)
 - clase, en System.Net, 285
- Do-Loop, 193-196
- Do-Until, 195-196
- Do-While
 - comprobación al final, 194-195
 - comprobar en el arranque, 194
 - describe, 187
- ventana acoplable, mostrando una ventana de herramientas, 36-37
- Document Explorer, de 39
- documentación
 - de los componentes de programas, 240-241
 - lectura, 38-39, 58
- Servicios de Nombres de Dominio (DNS)
 - clase, en System.Net, 285
- DomainUpDown de control, 64
- DOS (Disk Operating System), acceder, 216-218
- Dotfuscator Community Edición, 42, 329
 - .NET 247, 44, 339
 - .NET, ofuscadoras compilado, de 42
 - .NET Framework
 - funcionalidad adjunto, 13
 - como una biblioteca de clases, 107
 - define, 12
 - describe, 11, 44-46
 - dando programático nombres, 13
- interacción de las piezas de, 44-45
- Internet funcionalidad de, 283
- en capas, 12
- nulos usados por 166,
- reutilización de programas fuera de,
 - 211-215
- terminología, 12
- usos de, 46-48
- utilizando herramientas de depuración en,

Tipo Doble, 158-159, 170-171
dobles, 158, 162, 171
Descarga de clase, en
 System.Net, 285
DownloadFile método,
 llamadas, 289
DownloadFile subrutina,
 288-289
Evento DragDrop, 223
Dibujar los métodos, de la
 Objetos gráficos, 299
dibujo
 proporcionado por el BCL, 48
 nos brinda la red.
 Marco, 297
System.Drawing ruptura
 en pasos, 300
Clases de dibujo, utilizando un tipo de letra
 en, 221-222
Modelo DREAD, por clasificación de riesgo,
 242
DTE (tiempo de diseño
 Medio Ambiente), 334-335
Las bibliotecas de enlace dinámico (DLL)
 en comparación con los servicios Web,
 123-124, 126
 definido, 108-109
 descrito, 118-121
 como una especie de biblioteca de clases, 109
 como la salida de Web
 Las piezas, 333
 como stateful, 126
 usos de, 109
Dynamic Link Library (DLL)
 archivos
 crear, 114
 impacto de la sobrecarga en
 escrito, 228
 funcionamiento, 116-118

• E •

e-mail
 función edificio, 291
 obtener permiso para utilizar
 servidor, 291
ISPs bloqueo, 293
 como una operación basada en el servidor,
 290
Editar enlace de elementos, en la
 Ventana Propiedades, 73
Editar menú, añadiendo, 79
ediciones de Visual Studio
 2008, 14
Elevación de privilegios, en
 STRIDE, 242
botón de puntos suspensivos (...), en un
 propiedad, 28
Else, 181

- utilizando, 222-225
 - manejo de eventos, utilizando IntelliSense, 224-225
 - registros de eventos
 - arrastrando en los formularios, 32
 - en el Explorador de servidores, 31
 - eventos
 - describir, 28, 69
 - doble clic para generar código del controlador de eventos, 313
 - exponiendo por objetos, 222
 - el tratamiento de las devoluciones de datos como, 85
 - Eventos de botón, al hacer clic, 223
 - Eventos del panel, 223
 - Clase Evidence, 251
 - Excepción objeto
 - describir, 147
 - mirar el contenido de, 148
 - excepciones, 179, 183-185, 206
 - .Exe extensión de archivo, 19
 - imagen existente, para conseguir una Objeto Graphics a partir de, 300
 - Existe método, con el apoyo de Archivos y directorios, 273
 - Exit-Por declaración, en For-Next loops, 191
 - Opción Salir en el menú archivo, 79
 - Declaración ExitWhile, en un While-End loop, 196
 - Explotabilidad, en DREAD, 242
 - Express Edition, Visual de Studio 2008, 14, 325
 - proveedores extensores, y agregó: 316-317
 - eXtensible Markup Language (XML)
 - Copiar datos impulsada por, 254
 - describir, 136
 - Versiones XML de Windows
 - conceptos, 124
 - F •
 - F1, 38-39
 - F4
 - en Visual Studio 2008, 27, 67, 312
 - en formularios Web Forms, 93
 - F5
 - en Visual Studio 2008, 70, 140-141, 149
 - en formularios Web Forms, 95
 - F8, 154
 - F10, 149, 153
 - F11, 149
 - características, añadiendo a Windows
 - Los formularios, 70-81
- Elself declaración, 182
- encapsulado código, 198-199
- encapsulación, de funcionalidad, 109
- cifrado, 245
- línea de fondo, en un flujo de control declaración, 181
- Declaración final, después de un caso de declaración, 181
- Clases EndPoint en System.Net, 285
- utilizado por las clases de socket 286
- Enterprise Library, acceso, 269
- Enterprise Library de acceso a datos
- Datos de la aplicación de bloques
 - corredor, 268-269
- Enterprise Services, siempre por el BCL, 47
- software empresarial, específico
- requisitos de, 266
- Entidad-relación (ER)
- diagrama, 53-54
- Epsilon, las pruebas de 171,
- ER (entidad-relación)
- diagrama, 53-54
- Error al crear el control
- mensaje, en los formularios
 - Web Forms, 94
- mensajes de error
- para un ValidationSummary control, 100
 - para las aplicaciones de formularios
 - Web Forms, 250
- ErrorProvider de control, 64
- errores
- en resumen ambiente, 205-206
 - en comparación con excepciones, 184
 - define, 184
 - manipulación, 147-148, 205-206
- escapar de la marca, 158
- Incluso espacio entre
- Controles opción, el 71
- Argumentos del evento, 222
- Botón de sucesos, en las propiedades ventana, 312
- controladores de eventos
- la adición de código para el 69
 - en la Fecha de Cálculo aplicación, 198-199
 - define, 68
 - describir, 222
 - como funciones, 197-198
 - generando
 - del Código View, 313-314
 - desde la ventana de Propiedades, 311-313
 - 224 relajado,

campos. Véanse también las variables privadas
describe, 54, 114
arrastrando a un formulario en blanco en

Ver el diseño, 34

implementaciones de, 54

Clase File, 272-273

Archivo de la aplicación de gestión
la adición de un sistema de archivos

Watcher para, 280-281

apertura, 274-276

clases de administración de archivos, en

System.IO, 272

Menú Archivo

añadir una opción Salir en el 79

añadir a un formulario, 78-79

sistema de archivos, trabajar con,
271-281

File Transfer Protocol (FTP)

bloqueado corporativo

redes, 292

la construcción de una aplicación que utiliza,
288

describe, 288

tipos de archivos, personalizar

barras de herramientas, 315-316

FileInfo clase, 272, 279-280

archivos

cambiando el contenido de, 277

descarga de la

Internet, 288-290

cifrado, 245

lista, 279

monitoreo, 280-281

apertura, 29-30, 274-276

ahorro, 274, 277-278

almacenar la información en, 271

información ver, 279-280

FileStream clase, 272, 288

FileStream objeto, 289

FileSystemObject, en VB 6

y VBScript, 272

FileSystemWatcher clase,

272

FileSystemWatcher

componente, 271

FileSystemWatcher herramienta,

añadiendo, 280-281

FileUpload control, 87, 274

FileWeb conjunto de Clases, en

System.Net, 285

Rellene métodos, 299

Método Finalize, 220-221

Finally, 185

FindDateDiff función, 199

manguera de fuego, 256

cortafuegos, bloqueo de red

tráfico, 292

ventana flotante, 35-36

flujo de datos, como una corriente, 289

- componentes de diagrama de flujo,
que representa la lógica de
negocio,
176-179
- diagramas de flujo, 176
- FolderBrowserDialog
control, 273
- carpetas, en el Visual Studio
Proyectos de directorio, 29
- Clase Font, el uso de cepillos, 300
- For-Cada lista, de enlace
a través de Control,
192-193
- Bucle For-Next, 187, 190-191
- colección de formularios, en Trace,
151
- componentes de la forma,
añadiendo a
formularios, 26
- controles de formulario comunes, 64
- diseñador de formularios,
ContextMenu
Pele constructor en el 81
- proveedores de formato, mostrando
fechas, 172-173
- Controlador de eventos Form_Load
crear, 146
edición, 75
- formas
agregar funcionalidad a, 18
que codifica para la estabilidad y
razones de seguridad, 84
impresión, 301
- Palabra clave Friend, 119
- FTP (protocolo de transferencia de
archivos)
bloqueado corporativo
redes, 292
- la construcción de una aplicación
que utiliza,
288
- describe, 288
- Sitio FTP, publicando, 91
- FTPWeb clase, en System.Net,
285
- FtpWebRequest clase, 288
- Palabra clave Function, 112
- código funcional, en una clase
biblioteca, 113
- sobrecarga funcional, 120-121
- Descripción de las funciones, de una
aplicación, 240
- piezas funcionales, rompiendo
componentes individuales
en, 241
- funcionalidad
añadiendo a Windows Forms,
68-70
- concepto de reutilización, 199
- describiendo las aplicaciones, 57
- funciones
edificio
con la reutilización en mente,
198-200
- compartida,
119-120
- cambiando con el operador
sobrecarga, 227-228
- fabricación
como abstracta, 199
adecuado para su reutilización, 200
- referencia directamente, 202-203
- la devolución de valores, 112
- en la clase StringBuilder,
168
- pruebas en los servicios Web,
130-131
- en un archivo. vb, 110-111
- G •**
- recolección de basura, en .NET,
220-221
- Las variables generales del medio ambiente,
41
- generando
generadores de código, 23, 331
Generador de datos de objetos, 326
- controladores de eventos
del Código View, 313-314
desde la ventana de Propiedades,
311-313
- la concatenación de cadenas, nunca
utilizando para generar SQL, 248
- genéricos
edificio, 230-231
define, 230
diseñando para, 231-232
- Cómo parte de una propiedad, 115
- Método GetCurrent, de la
WindowsIdentity
objeto, 244
- Google, utilizando, 340-341
- Google Code Repository, 339
- Google CodeSearch, 339
- Google Search Tool, construcción,
213-215
- GotDotNet, 338
- Graffiti, 337-338
- Gráficos clase
describe, 298
lápices, pinceles y fuentes, en
301
- Gráficos objeto
consiguiendo, 300-302
métodos y propiedades de,
298
- gráficos de programación, 298
- pestana gris, en la vista Diseño, 24, 25
- directrices, ayudando a alinear
controles, 71

Muestra Hack, añadiendo, 323
Instrucción Handles
en un controlador de eventos, 222
describe, 223
tablas hash, 256
Hatfield, Bill
 ASP.NET 2.0 Para Dummies, 39,
 106
Encabezados Collection, en Trace,
 151
Hello World aplicación,
 edificio, 16-17
variable de formulario oculto,
 ViewState como objeto, 101
Hit Count columna, en el
 Puntos de interrupción ventana, 142
uso en el hogar, consiguiendo Visual Basic
 2008 para, 325
¿Cómo hago botón, en el
 Document Explorer
 barra de herramientas, 39
Propiedad HREF, del anclaje
 etiqueta, 103
HTML
 ASP.NET renderizado, 84
 elementos de, 106
 pegar texto, 314-315
El código HTML, para
 Default.aspx, 93-94
Comentarios de HTML, no dejando en
 lugar, 250
Controles HTML, para la Web
 aplicaciones, 88
Formato HTML, grabación,
 319
Párrafo HTML
 cambiar una línea de texto en,
 335
 formato de grabación, 320
HttpEncode método, 249
Http (HyperText Transfer
 Protocol) de clase, en
 System.Net, 285
objetos HTTP, obtener
 información de, 105
HTTP Subir, 274
Clase HttpApplication, en
 ASP.NET, 88
HttpBrowserCapabilities
 clase, en ASP.NET, 88
HttpContext clase, en
 ASP.NET, 88
Clase HttpCookie, en ASP.NET,
 88
HttpRequest clase, en
 ASP.NET, 88

método, 162

- HttpRequest objeto
 - conseguir un WindowsIdentity objeto, 105
 - información en, 106
 - refiriéndose a la corriente 105,
- HttpResponse clase, en
 - ASP.NET, 89
- Clase HttpSession, en
 - ASP.NET, 88
- Clase HttpUtility, en
 - ASP.NET, 89
- HttpWriter clase, en
 - ASP.NET, 89
- Web hipervínculo de control del usuario, 103
- | •
- iconos que se utilizan en el libro 6,
- Propiedad ID en formularios Web Forms, 93
- IDE (Integrated Development Medio Ambiente), 1, 35
- identidad, suplantación de la corriente
 - usuario, 249
- Interfaz I Enumerable, 192
- "Si es posible" Requisitos, 52
- If, 181
- If-Then-Else-If,
 - en comparación con Select- Case, 182-183
- If-Then-Else declaraciones
 - manejar una navegación decisión, 103
 - combinando varias, 182
- Si-Then, 191
- IIS (Internet Information Server)
 - describe, 84
 - la presentación de los servicios Web, 135
 - requerido para la ejecución un Web ASP.NET aplicación, 95
 - papel con los servicios Web XML, 135
 - Sitio IIS, publicar en el 91
 - procesamiento de la imagen, el impacto en
 - Páginas web, 106
 - ImageMap control, para la Web aplicaciones, 87
 - imágenes
 - crear, 297-307
 - manejo con HTML normal, 104
 - configurando rutas de, 104
 - imágenes etiquetas, 104
 - ImageUrl propiedad, 104
 - Ventana Inmediato, 140, 144-145
- la aplicación de
 - acceder a un recurso, 233
 - ADO.NET, 254
 - COM (Component Object Model), 211-212
 - campos, 54
 - procesos en Visual Basic, 180-185
 - Utilización de recursos Administración, 221
 - SOAP (Simple Object acceso Protocol), 136
 - roscado, 233-235
- Imports
 - añadiendo a LINQ código, 269
 - referencia
 - una clase, 203
 - una función, 202
 - la clase StringBuilder, 167
 - bucles indefinidos, 188, 193
- Proveedores de software independientes (ISV), 329
- Botón de índice, en el documento Explorador de la barra de herramientas, 39
- información
 - divulgación, en STRIDE, 242
 - cifrado, 245
 - de objetos HTTP, 105
 - sitios en línea, para Visual Basic información, 337-341
 - almacenar en archivos, 271
 - tipos de información en Visual Basic 2008, 157-164
 - ver la información del archivo, 279-280
- declaración de herencia, en un Servicios Web, 129
- Archivos INI, actualización, 326
- línea de scripting modelos, 338
- de entrada, la recopilación de usuario, 54
- Input Output (IO), proporcionada por el BCL, 47
- parámetros de entrada para el evento,
 - manipuladores, 224
- Insertar función, en la StringBuilder clase, 168
- Inserte característica de fragmentos, 323
- Inserte declaración, colocando una palabra en una cadena, 168
- instancia
 - de clases, 49, 110
 - de los controles, 67
 - define, 49
 - de objetos, 76
- clases de instancia, 273
- clases instanciadas, en System.IO, 279
- Integer objeto, TryParse

Tipo Integer
descripción, 158-159, 170
MaxValue y MinValue
método, 171

enteros, 158

Integrado de Desarrollo

Environment (IDE), 1, 35

IntelliSense

manejo de eventos utilizando, 224-225
sobrecarga en, 226
mostrando una sobrecarga, 121
mostrando la documentación del usuario
información, 39
utilizando en Ver código, 37-38

Menú IntelliSense, apertura,
37-38

Internet

acceder, 283-296
descarga de archivos de
288-290
recursos sobre, 337-341
funcionamiento de los servicios Web de,
133

Internet Information Server
(IIS)

describe, 84
la presentación de los servicios Web, 135
requerido para la ejecución
un Web ASP.NET
aplicación, 95
papel con los servicios Web XML,
135

Protocolo de Internet (IP) de clase,
285-286

interoperabilidad de la Web
servicios, 125

InvalidOperationException
error, Visual Basic
lanzando, 162

IO (Input Output), proporcionado por
el BCL, 47

IP (Protocolo de Internet) de clase,
285-286

IrDA clase, en System.Net,
285

IsDaylightSavingsTime
método, en el DateTime
objeto, 172

Método IsDBNull, construido en
Visual Basic, 166

ISV (Independent Software
Vendor), 329

Colección Artículos de diálogo Editor
caja, apertura, 73-74

bucle iterativo, 189

. J .

Motor de JavaScript, 248

describe, 175

• K •

kbAlertz, 340

• L •

Etiqueta de control
añadir a una página Web, 91-92
añadiendo a formularios, 67
describe, 64
Etiqueta de control Web, texto de,
106
idioma, Visual Basic 2008
como, 1-2
Columna de idioma, en la
Puntos de interrupción ventana,
142
Language Integrated Query
(LINQ)
describe, 253, 326
LINQ to SQL:. NET
Integrated Query para
Datos relacionales, 270
como novedad, el 22
utilizando, 269-270
último problema victorias, 254
Resolución Late Bound
advertencia, 213
Barra de herramientas Diseño, 316
código heredado, 211
bibliotecas de clases, el impacto de
sobrecarga en la escritura, 228
rayo botón, 28
saltos de línea, en el código, 3
enlazar el código, añadido por Visual
Estudio, 69
LINQ (Language Integrated
Query)
describe, 253, 326
como novedad, el 22
utilizando, 269-270
LINQ to SQL:. NET
Integrated Query para
Datos relacionales, 270
listado, directorios y archivos, 279
literales, en los patrones de, 169
Cargue evento, agregando código
para para
una forma, 81
localhost, el e-mail
nombre del servidor, 292
Vivo en la lista desplegable, en
el Web Site de diálogo Nuevo
caja, 91
bloqueo, recursos, 221
la tala, la actividad de la red,
293-296
lógica. Véase también la lógica de
negocio
contenida en clases, 220
definiendo para una aplicación,
56-57

errores lógicos, como resultado de
utilizando ByRef, 205
LogonUserIdentity, 105
bucle, 187
bucles, escribiendo con For-Next,
190-191
programas débilmente acoplados, 247

• M •

Máquina / SqlExpress, en el
Cuadro de diálogo Agregar conexión
caja, 32
Explorador de macros, apertura, 319,
320
Macro IDE, 319, 321
macros, 319-321
Función Correo, en
System.Net, 286
Menú principal, en la vista Diseño, el 79
Gestionado modelo de grupos de noticias,
339
gestión, suministrado por el
BCL, 48
Las clases de gestión, en el servidor
Explorador, 31
Eventos de gestión, en el servidor
Explorador, 31
Atributo Maestro, en la página
directiva, 211
páginas maestras
sumarse a proyectos, 210
describe, 210
haciendo, 210-211
corriendo con el contenido, 211
Coincidir función, objeto Regex,
169
matemáticas, subrutinas para, 115-116
Me objetar
describe, 79
refiriéndose a la actual
objeto, 102
Me.Close () declaración,
añadiendo, 80
gestión de la memoria,
importancia de, 220
barra de menú, estándar, 78-79
sistema de menús, para proteger
cada usuario, 243-245
MenuBar control, añadiendo a
formularios, 67
menús editables en Visual
Estudio, 40
El control ToolStrip
describe, 64
navega con, 78-80
MenuStrip objeto, 78
Propiedad de mensaje, de la
Objeto Exception, 148

Colas de mensajes en el servidor
Explorador, 31
mensajería, suministrado por el
BCL, de 47
metacaracteres, en los patrones de, 169
metadatos, almacenar, 30
Parámetro de método, de
FtpWebRequest, 288
firma de método, 120-121
métodos, llamando en COM
objetos, 213
Certificado por Microsoft vendedor, 329
Microsoft Excepción
Bloques de Gestión
componente, 148
Los programas de Microsoft,
interactividad entre, 314
Microsoft públicos grupos de noticias,
339
Microsoft Report Viewer, 301
Microsoft Team System, de 58
Microsoft Word
crear un archivo nuevo, 212
estado tira en el 73
Mime función, en
System.Net, 286
Mobile 5.0, 21
Mobile 6.0, 21
computación móvil, 21
Equipo móvil, libera de, 21
modularidad, de una de n niveles
sistema, 53
módulo (%) del operador, 170
Función MonthName, utilizando,
172
MouseEnter evento, 223-224
MSDN Library
acceder, 15
suponiendo una comprensión
de herramientas, 24
describe, 24, 337
instalar, 24
Sitio Web, 15
decisión de opción múltiple, 179
procesos de selección múltiple,
182-183
multiplicación (*) del operador,
170
Mi colección, objetos, 47
Mi objeto, 46
Mi objeto ícono, de 47
Mi Proyecto archivo de configuración,
246
Objeto My.Application, 47
Objeto My.Computer, 46
archivo My.Log, 295-296
MyProject configuración
forma, el 66
Carpeta MyProject, 66
My.User objeto, 47

• N •

n niveles del sistema, 53
 Nombre de la propiedad de un LogonUserIdentity objeto, 105-106
 de WindowsIdentity, 105
 las clases del espacio de nombres, en ASP.NET, 88
 espacios de nombres, en un archivo. vb, 110,
 111
 nombrando
 Espacios de nombres en los servicios Web,
 134
 Aplicaciones Web, 92
 navegar, con un ToolStrip control, 78-80
 Controles de navegación Web, por aplicaciones, 88
 NavigationUrl propiedad, Marco, 103
 NegativeInfinity, pruebas para, 171
 .NET, 44
 .NET 247, 339
 .NET, ofuscadoras
 compilado, de 42
 Cobros netos, en el BCL, 48
 .NET Framework
 funcionalidad adjunto, 13
 como una biblioteca de clases, 107
 define, 12
 describe, 11, 44-46
 dando programático
 nombres, 13
 interacción de las piezas de, 44-45
 Internet funcionalidad de, 283
 en capas, 12
 nulos usados por 166,
 reutilización de programas fuera de,
 211-215
 terminología, 12
 usos de, 46-48
 utilizando herramientas de depuración en,
 145-154
 Papel en Visual Basic, 11-13
 actividad de la red, la tala,
 293-296
 características de análisis de red de .NET, 292
 aplicaciones de red, el potencial problemas con, 292
 conexiones de red,
 gestión, 285
 estado de la red, comprobación, 287
 tráfico de la red, depuración, 293
 Clase NetworkCredential, en System.Net, 285

NetworkInformation clase,
 287
 NetworkInformation función, en System.Net,
 286
 Nuevo proyecto cuadro de diálogo acceso en Visual Studio 2008, 65
 seleccionar un add-in, 334
 en Visual Studio 2008, 15
 Nuevo Sitio Web del cuadro de diálogo,
 opciones de la plantilla, 90, 91
 NNTP grupos de noticias,
 acceder, 39
 nodos, en los diagramas de flujo, 176
 noncontrol valores de las variables, ahorro en ViewState,
 101-102
 componentes no visuales, usando, 317-318
 no volátil de almacenamiento, 158
 Northwind base de datos,
 instalar, 32
 Ahora la función, usando, 172
 NullReferenceException error, 166
 nulos, 165-166
 números
 métodos integrados, 171
 almacenar en variables, 158
 tipos, que cubren 170
 Sistema de pruebas NUnit, 58

• O •

carpeta obj, creado por Visual Estudio, 66
 Object Linking and Embedding Base de datos (OLEDB) protocolo, 257
 lenguaje orientado a objetos, reglas de, 78
 Objeto Request Broker, en un sistema de niveles, 53
 Fuente de objetos, en los datos La configuración de orígenes Wizard, 258-259
 Tipo de objeto, 159
 objetos
 casting para cuerdas, 161
 controlar con rosca,
 232-235
 describe, 49, 110
 haciendo y destruyendo,
 220-221
 tipos como, 173
 viendo todo, 144
 Espacio de nombres de ODBC, en System.Data, 255

ODBC (Open Database Conectividad) el cumplimiento, 257
 De palabra clave, 230
 OleDb espacio de nombres, en System.Data, 255
 OLEDB (Object Linking and Incorporación de base de datos) protocolo, 257
 OnClick evento, 222
 Controlador de eventos OnClick, 160
 competencia en línea, en joing TopCoder, 326-327
 sitios en línea, para Visual Basic información, 337-341
 6 herramientas online, proporcionados por Microsoft, 39
 Open Database Connectivity (ODBC) el cumplimiento de, 257
 Gestión de Fuente Abierta Sistema nuevo de Microsoft, 340
 proyecto de código abierto, participar en, 327-328
 software de código abierto, 327
 versiones de código abierto, de Ventanas conceptos, 124
 OpenFileDialog evento controlador, 280
 El control OpenFileDialog en comparación con SaveFileDialog, 278
 descrito, 273-274
 opciones para, 275
 apertura, archivos, 274-276
 sistema operativo
 describe, 12
 llegar a, 46-47
 Declaración de operación, en un biblioteca de clases, 112
 operaciones, que se ejecuta en hilos separados, 232
 símbolos de operador, números 170, manipulando operadores, 227-228
 Palabra clave opcional, 229
 parámetros opcionales, 229
 opciones disponibles en Visual Estudio, 40
 Cuadro de diálogo Opciones, en Visual Studio 2008, 40-41
 OracleClient espacio de nombres, en System.Data, 255
 Las órdenes Detalle formulario de datos, crear, 263-264
 organización del libro, 4-5
 fuera de alcance, 51
 Parámetro de salida, 162
 recursos externos, el acceso a con el Explorador de servidores, 30-32

sobrecarga
en comparación con los genéricos, 231
descripción, 120-121, 226
para diseñar, 228-229
procedimientos sobrecargados,
en comparación con el opcional
parámetros, 229
Sobrecargas palabra clave, 226

• P •

Directiva @ Page, 150-151
Page.Load controlador de eventos, 105
Método PaintBoard, 306-307
Panel de control, para la Web
aplicaciones, 87
El apartado macro, 320-321
matriz de parámetros, 228
parámetros
229 opcional,
que pasa, 200, 204
la protección de los valores de,
204-205
tipos, de controladores de eventos, 222
Método Parse, 173-174
análisis, 173
clases parciales, aportaciones
parámetros para, 224
Pegue comando alternativo,
315-316
pegar texto como HTML, 314-315
Clase Path, 272
patrones, 169
Patrones y Prácticas de equipo
corredor de datos creado por, 268
enfoque sistemático para
diseño seguro
programas, 240
Detenido modo. Ver el modo de
depuración
plumas
dibujar líneas y curvas, 299
propiedades de, 299
utilizado por los objetos gráficos,
298
PerfMon, acceso, 31
Contadores de rendimiento
arrastrando a una forma, 32
en el Explorador de servidores, 31
Perl, las expresiones regulares fuera
de,
339
Permisos de espacio de nombres,
251
Controles de personalización, para
Aplicaciones Web, 88
búsqueda de frases, de la
Document Explorer
barra de herramientas, 39
la planificación. Véase también el

diseño de
define, 49
para los
proyectos, 49-
51
Botón de reproducción, el lanzamiento de una web
aplicación, 95
Pocket PC 2003, 21
punteros, 204
Política de espacio de nombres, 251
polimorfismo, 78
Portal Server, 332
PositiveInfinity, pruebas
para, 171
Proceso PostBack, 84-85
amenazas potenciales, identificando,
241-242
el código de presentación, en
ASP.NET, 90
capa de presentación, en un n-tier
sistema, 53
Espacio de nombres principal, 251
PrincipalPermission clase,
251
Comando Imprimir, llamadas, 301
PrintDialog control, 64
Componente PrintForm, 301
 impresión, formularios, 301
prioridad, construido para roscar,
235
Clase privada, 203
Palabra clave Private, 119, 203
Declaración privada,
objetos de acotación, 221
variables privadas
definido, 114-115
utilizando para hacer matemáticas, 116
procedimientos
decisiones con sobrecarga,
226-229
programación procedimental, 205
nombres de reutilización, 226-227
Clase de procesos, 213-215
componentes del proceso
características de, 178
en un diagrama de flujo, 176
de diagramas de flujo de programas, 178
ID de proceso
describe, 216
devuelto por un Shell
comando, 217
Objeto de proceso, ProcessId
parámetro, 218
procesos
adjuntar a, 153-154
aplicación en Visual Basic,
180-185
productores, los servicios Web, 123,
127-128
Professional Edition
Servidores de nodo en, 31
de Visual Studio 2008, 14
nombres programáticas, en. NET
Marco, 13

lenguaje de programación,
interactuar con la
marco, 13

prácticas de programación,
evitando la complejidad,
204-206

programas
documentar los componentes de,
240-241

imprecisa, 247

reutilización fuera de. NET,
211-215

funcionamiento compilado, 20

utilizando la clase Process,
213-215

proyecto de código de datos, la creación,
262

ciclo de vida del proyecto, 49-51

sensibilidad proyecto, de la
Caja de herramientas, 27

plantilla de proyecto, creado por
Visual Studio, 128

tipo de proyecto, la selección, el 65

Tipo de proyecto o plataforma,
controlada por ASP.NET o
Formularios Windows Forms, 48

Panel Tipos de proyecto, de la Nueva
Proyecto de cuadro de diálogo, 15

proyectos
conexión con bases de datos, 32

depuración, 148-154

organización con la solución
Explorer, 29-30

pausa, 140

planificar, 49-51

tipos de, 1

utilizando herramientas de terceros en,
328-329

propiedades
cambiando para un control, 63

la creación de una biblioteca de clases,
114-115

el mantenimiento de los valores, 112

ninguno en un servicio Web basado en
biblioteca de clases, 125

de plumas, 299

puesta a cero para los controles, 67

en un archivo. vb, 111

Ventana Propiedades
cambiar los detalles con, 27-29

manejo de eventos utilizando, 222-224

expansión, 312

generación de controladores de eventos
de, 311-313

apertura, 29, 67, 93

Propiedad palabra clave, 112

Protected Friend palabra clave,
119

Palabra clave Protegido, 119

- protección, los artefactos que necesitan, 240
- protocolos de Internet, basado en 283
- clase de proxy
- instancias de los servicios Web, 134
 - estado de los servicios Web, 135
- pseudocódigo, 56
- Función pública, 203
- Palabra clave Public
- para archivos de clases y funciones, 203
 - describen los procedimientos de clase, 119
- servidor público, la protección, 247
- servicio público de sitio de acogida, 330
- Los servicios públicos de Web, integradora, 329-331
- Q •**
- Barra de herramientas del Diseñador de consultas, 316
- Colección cadena de consulta, en Trace, 152
- QueryString variable, evitando, 250
- Queue Server, en Windows, 47
- QuickStart Tutorials, en el ASP.NET sitio, 338
- comillas, en torno a variables de cadena, 158
- R •**
- RAD Data Tools, 262
- RAD (Rapid Aplicación Desarrollo), 23, 262
- RadioButtonList control, para Aplicaciones Web, 87
- RAM, uso, 220
- Desarrollo rápido de aplicaciones (RAD), 23, 262
- Panel Grabar, 320
- Grabar Macro Temporal opción, 320
- grabación, macros, 319-321
- De registros. Ver los contenedores rectángulos, en diagramas de flujo, 176-178
- elementos de referencia, sumándose a una
- StatusStrip, 74-75
- nombres de referencia, para las instancias de los controles, 67
- Referencias carpeta, creado por Visual Studio, 66
- Referencias pestaña de Mi El archivo de proyecto, 117
- Botón Actualizar, en la solución de Explorador, 30
- Regex objeto, en el partido función, 169
- registro de eventos de WMI, 31
- expresiones regulares
- descrito, 167-169
 - encontrar, 339
 - cadenas de manipulación, 168-170
 - usos de, 169
- retransmisión, 290
- Modo de lanzamiento, las configuraciones ajustado en, 20
- Configuración remota Depurador programa, funcionamiento, 150
- depuración remota, 150
- Eliminar la función, en la StringBuilder clase, 168
- Reemplace la función, en la StringBuilder clase, 168
- Reemplace método
- del objeto Regex, 169
 - con la cadena, 165
- informes, creación, 301
- Reprodicibilidad, en DREAD, 242
- El repudio de las acciones, STRIDE, 242
- Galletas de solicitud de recogida, en Trace, 151
- Detalles de la solicitud, en seguimiento, 151
- Solicitudes de Comentarios (RFC), 284
- RequiredFieldValidator
- controlar
 - añadiendo en los formularios Web Forms, 99-100
 - creación, 100
- requerimientos
- de reunión para una aplicación, 51-52
 - de un proyecto, 50
- Documento de requerimientos, de 52
- Cambiar opción Toolbox, 27
- Utilización de recursos
- Gestión, aplicación, 221
- recursos
- en Internet, 337-341
 - bloqueo, 221
 - utilizando, 221
- Las cookies de respuesta Collection, en Trace, 151
- Cabeceras de Respuesta Collection, en Trace, 151
- Restaurar asociaciones de archivo botón, en las opciones de cuadro de diálogo, 41
- valores devueltos, aceptando de funciones, 200

reutilización
código, 197-218
la creación de componentes, 207
extendiendo con archivos de clase,
201-203

nombres de procedimientos, 226-227
código reutilizable, 198
funciones reutilizables
 crear, 198
 hacer parte de un separado
 archivos, 198
 poniendo en archivos de clase, 202

Reynolds, Matt, 339

RFC (Solicitudes de
 Comentarios), 284

Control RichTextBox, 64

botón derecho del ratón, activación con
 ContextMenuStrip,
 80-81

riesgo, rating, 242

roles, en oficinas o empresas a,
 243

rutinas, escribiendo para su uso con
 otro software, 21

fila de bloqueo de problema, 254

Rubí forma de construcción de software, 14

Rubí Forma del motor, en Visual
 Basic 6.0, 62

Modo de ejecución, ejecutar un programa
 en, 149

proyecto ejecutable, como visual
 requisito de herramienta, 140

tiempo de ejecución, estableciendo en el texto, 76

• S •

SAO (Software Architecture
 Información general) diagrama,
 240-241

SaveFileDialog control, 273,
 277-278

escenarios, 56

Scoble, Robert, 338

alcance
 de un proyecto, 50, 51

variable en, 144

pantalla

 diseño, 54-56

opciones de gestión, en
 Ver el diseño, 25

código de secuencia de comandos, que muestra como
real

 texto, 249

los ataques mediante scripts

 describe, 248

 prevenir, 249

 comprensión, 248-249

lenguajes de scripting, 106

Botón Buscar, en el documento

 Explorador de la barra de herramientas, 39

búsqueda, en Google, 341
código de seguridad, la escritura, 239-251,
293
software seguro, el diseño, 240-242
SecureString clase, 251
seguridad, 239
Los controles de seguridad, para la Web
aplicaciones, 88
La función de seguridad, en System.Net, 286
Seguridad de espacio de nombres, 251
política de seguridad, la configuración de una aplicación, 242-243
Pestaña de Seguridad, del Mi Proyecto
archivo, 246
SecurityPermission clase, 251
Select-Case declaración, 183
SelectedValue propiedad, de un lista desplegable, 98
SendEmail subrutina, 291-292
Argumento del remitente, 222
Explorador de servidores, 30-32
Variables del servidor, en Trace, 152
variables de servidor, el ahorro, 86
Vista de servidor, en formularios Web Forms, 89
Server.HTMLEncode método, 249
servidores
configuración y gestión de, 106
controles, 86-89
la creación de controles personalizados, 207-209
en .NET, 44
guardar archivos, 274
Servidores nodo, en el servidor Explorador, 31
La clase de servicio, en System.Net, 285
Arquitectura Orientada a Servicios (SOA), 126-127
Archivo Service.asmx, 128
Archivo Service.asmx.vb, 128
servicios disponibles en el servidor Explorador, 31
Objeto Session, en ASP.NET, 102
Estado de la sesión, en Trace, 151
Establecer parte, de una propiedad, 115
Proyecto de instalación, 20
clases compartidas
descrito, 272-273
ejemplo de, 203
en System.IO, 279
funciones compartidas, 119-120

palabra clave
compartida, 119
Métodos
compartidas, 203
SharePoint Web Parts
incluido, 332-333
palabra clave
compartida, 119
SharePoint Designer, 333
Shell de comandos, 216-217
Shift + Alt + D, 33, 258
Botón Mostrar todos los archivos, en la solución de Explorer, 29-30, 295
Mostrar datos de opción Fuentes, 258
sencillez, de los servicios Web, 125
decisión de un solo proceso, 180-182
decisión solo proceso, 179-182
Usuario individual Ronda Partido interfaz, 327
Sitio clase, 251
Etiqueta inteligente, para cada campo Vista Detalles, 263
SmtpClient objeto, 291
fragmentos, inserción en el código, 323
SOA (Service-Oriented Architecture), 126-127
SOAP (Simple Object acceso Protocol), 136
Toma clases, en System.Net, 285
Sockets funcionar, en System.Net, 286
software
describe una técnica de perspectiva, 52
diseño, 43
diseño seguro, 240-242
reutilización de código para construir, 197-203
Arquitectura de Software
Resumen (SAO) diagrama, 240-241
El Explorador de soluciones
la edición de los Mis proyectos carpeta, de 66
con la organización de proyectos, 29-30
cambiar el nombre del formulario predeterminado, 66
Ordenar subrutina, para arreglos, 228
Ver código fuente
en el acceso a los formularios Web Forms, 93-95
Anchor pestaña, 103
trabajo visualización en, 93-95
en formularios Web Forms, 89
SourceForge.net, 328
bucles específicos, 188
Dividir vista, en los formularios Web Forms, 89
Suplantación de identidad, en STRIDE, 241
SQL: NET Language Integrated
Para datos relacionales, 270
SQL Injection, 247-248
SQL Server Express, acceder, 32
Espacio de nombres SqlCommand, en System.Data, 255

SqlTypes espacio de nombres, en
System.Data, 255
atributo src, cambiando la
ruta de acceso a una imagen, 104
Propiedad StackTrace, de la
Objeto Exception, 148
Personal de objeto, como una lista de
personas, 230-231
Standard Edition de Visual
Studio 2008, 14
Inicio botón Generar,
clic, 70
la línea de salida, en un flujo de control
declaración, 181
StartDatePicker control,
la adición de un ValueChanged
controlador de eventos, 118
StartThread_Click
controlador de eventos, en
ThreadingExample, 233
estado, de un objeto, 49
estado de una aplicación
ASP.NET gestión de, 84
tratando, 101-102
descripción, 85-86
funciones estáticas, en comparación con los
stateful objetos, 126
texto estático, escribiendo directamente en un
Página web, el 91
informe sobre la situación, e-mailing, 290-293
tira estado, 73-75
StatusStrip Control, 73-74
StatusStrip objeto, 78
Paso declaración
en bucles For-Next, 190
utilizando para contar hacia atrás, 191
Stop Recording, en el
Panel Grabar, 320
de almacenamiento, tipo de, 158
StreamReader, consiguiendo
información de un archivo, 277
corrientes
describe, 289
como herramientas complicadas, 272
StreamWriter, 277
Acrónimo STRIDE, 241-243
La clase String, 111
la concatenación de cadenas, nunca
utilizando para generar SQL, 248
Objeto String, métodos,
propiedades y eventos, 165
Tipo String
descrito, 164-166
como el valor de otro objeto,
159
StringBuilder clase, 167-168
String.Empty valor de preselección,
165

- T** .
 System.String.Compare
 método, 111
 System.Text referencia, para
 la clase StringBuilder,
 167
 System.XML, como un nuevo
 característica de, 22
 grupos de pestañas, en la vista de
 diseño, 26
 documentos con fichas, mostrando
 una
 ventana de herramientas, 36-37
 TabControl control, 64
 Atributo de la tabla, por lo que una
 clase
 referencia, 270
 Opciones de tabla en la lista
 desplegable,
 en datos de panel Fuentes, 263
 Objeto TableAdapter, y agregó:
 264
 TableAdapterManager
 objeto, añadiendo, 264
 tácticas, para el desarrollo Web
 servicios, 127
 La manipulación de los datos o
 archivos, en
 STRIDE, 241
 Lista de tareas, utilizando, 322-323
 Team System edición
 Servidores de nodo en, 31
 de Visual Studio 2008, 14
 plantillas
 seleccionando en formularios Web
 Forms, 90-91
 escribir para Office 2007, 21
 Plantillas panel, en el Nuevo
 Proyecto de cuadro de diálogo,
 16
 Botón de prueba de conexión, en el
 Cuadro de diálogo Agregar
 conexión
 caja, 32
 plan de pruebas, escribiendo para
 una
 aplicación, 57-58
 TestLocking botón, haga clic en
 caso de, 234
 texto
 pintado con las fuentes y
 cepillos, 300
 pegar como HTML, 314-315
 presentar a los usuarios de Web
 Formas, 91
 almacenar en variables, 158
 Cuadro de texto
 añadiendo en los formularios Web
 Forms, 99-100
 alineando con otros objetos, 71
 reemplazando el número por defecto
 de día, 72
 entrada de texto, manejo, 71-73
 Parámetro de texto, para una
 ValidationSummary
 control, 100

Texto propiedad
del objeto nextweek, 69
teniendo una cadena de, 173
de un cuadro de texto, 98

Control TextBox
añadiendo a formularios, 67
describe, 64
la gestión de texto de entrada, 71-73

Objeto TextBox, eventos
expuesto por, 313

TextReader clase, 272

Clase TextWriter, 272

terceros, pruebas
aplicaciones, 58
herramientas de terceros, utilizando, 41-42,
328-329

encapsulación hilo, 235

enhebrar
para diseñar, 232-233
implementación, 233-235

ThreadingExample, en la
Sitio Web para este libro, 233

modelo de amenaza, creando, 241

amenazas potenciales, identificar,
241-242

Tira palabra clave, 206
arrojando un error, 206

temporizador, emulando una problemática
funcionamiento, 233

Control Timer, añadiendo, 223

Objeto TimerCallback, 235

veces, las representaciones de cadena
de, 172

título de cadena
añadir a un programa, 164
la construcción, 168

listas de tareas, por lo que, 322

TODO comentario, añadiendo, 322

tokens, añadiendo en el código, 322

herramienta de ventana
mostrar, 35-36
mover, 35-36
opciones para el atraque, 36

barras de herramientas, editable en Visual
Estudio, 40

Caja de herramientas
acceder a los controles con, 26-27
apertura, 27
como proyecto y minúsculas, de 27 años
Los controles de validación en, 98-99
con controles de Windows Forms
cargado, 16-17

instrumentos
para la gestión de la entrada del usuario,
167-170

la actualización del código de usar, 326

con terceros, 41-42
en Visual Studio 2008, 23-35
para escribir código en Visual
Estudio, 36-39

- ToolStrip de control, 64
 - ToolTip controles
 - mejorando con botones, 317
 - dando pistas con, 75-78
 - ToolTip objetos
 - añadiendo a las formas, 317
 - describe, 78
 - diseño, 75
 - ToolTip propiedad, en el Ventana Propiedades, 76-77
 - TopCoder, la competencia en línea, 326-327
 - ToString
 - de todos los objetos, 173
 - El proveedor de formato con, 172
 - Método ToUniversalTime, en el objeto DateTime, 172
 - Método ToUpper, con String, 165
 - Trace, 150-152
 - Atributo Trace, 151-152
 - Trazar la clase, en la RED. Marco, 152
 - seguimiento de la actividad de la red, gestión de, 293
 - Bloque try-catch
 - la captura de una lógica excepción, 184
 - describe, 147
 - Try-catch, 184
 - Try, 184
 - Método TryParse
 - describe, 164
 - del objeto Integer, 162
 - utilizando, 173-174
 - Escriba, como una clase estática, 162
 - Tipo Type, tipo DateTime como un objeto de, 173
 - tipos
 - cambio
 - con CType, 159-162
 - con Parse y TryParse, 173-174
 - con el control de validación, 163-164
 - de información en Visual Basic 2008, 157-164
 - como objetos, 173
- U •
- UDDI (Universal Discovery y Descripción del lenguaje), 136-137
 - carácter de subrayado (_), a partir variables privadas, 114-115
- V •
- validar
 - tipos de control, 163-164
 - la entrada del usuario, 98-101
 - Los controles de validación en Visual Studio 2008, 98-99
 - para aplicaciones Web, 88
- Muestra deshecho, añadiendo, 323
 - Descubrimiento y universal
 - Descripción del lenguaje (UDDI), 136-137
 - actualizaciones, en Visual Basic para Dummies sitio Web, 7
 - Sube conjunto de clases, en System.Net, 286
 - Url clase, 251
 - URL
 - construcción en las páginas ASP.NET, 165
 - cargar con el Proceso clase, 215
 - casos de uso, 56
 - utilizar la sobrecarga, 229
 - Usenet, 339
 - controles de usuario
 - añadiendo, 209-210
 - características de, 209
 - en comparación con las páginas maestras, 211
 - describe, 209
 - desarrollo, 210
 - de entrada de usuario
 - controles en formularios Web, 98
 - forma, la construcción, 65-68
 - herramientas para la gestión, 167-170
 - validar, 58, 98-101
 - la interfaz de usuario
 - excepciones que permiten a brotan de, 205
 - en un sistema de n-tier, 53
 - diseño, 55
 - perder la capacidad comunicarse con, 205
 - para las macros, 319
 - herramientas para minimizar código, 58
 - usuarios
 - comunicarse con, 176-178
 - experiencia en la construcción, 14
 - obtener información sobre, 104-106
 - interactuar con, 48
 - historias de, 56
 - eventos iniciados por el usuario, causando Devoluciones de datos, 85
 - Uso de palabra clave, 221-222
 - USZipCode servicio, 331
 - ValidationSummary control, añadiendo en formularios Web Forms, 99-100
 - Evento ValueChanged controlador, crear, 69
 - valores, el ahorro a través de varios páginas 102
 - las variables
 - dentro y fuera de alcance, 144
 - tiempo de ejecución en comparación con el diseño tiempo, 76
 - almacenar información volátil en, 158
 - Visualizar los valores de, 141
 - VB 2002, 14
 - .Vb
 - contenido de, 110-111
 - HREF en propiedad, 103
 - estructura de, 111
 - VBForDummies.net, 337-338
 - vbfdordummies.net sitio Web
 - aproximadamente, 3, 7
 - DateCalc servicio Web, 129-130, 132-133
 - DownloadFile método, 288-290
 - Enterprise Library Data
 - Acceda a Application Block
 - datos corredor, 269
 - información de registro, 295-296
 - Northwind base de datos, 32
 - como recursos, 337-338
 - SendEmail subrutina, 291-292
 - System.Drawing espacio de nombres, 302
 - ThreadingExample, 233
 - Visual Studio Express Edition, 14
 - WSDL (Web Services Descripción del lenguaje), 136
 - VB.NET Class Library
 - plantilla, seleccionar, 114
 - VCR Bar, 264
 - Botón Ver diagrama de clase, en El Explorador de soluciones, 30
 - Botón Ver Código, en la solución de Explorador, 30
 - Botón Ver Designer, en El Explorador de soluciones, 30
 - visualización, información de archivo, 279-280
 - ViewState objeto, en ASP.NET, 101-102
 - Visual Basic
 - como una pieza de la red. Marco, de 44 años
 - revisiones de, 14

- Visual Basic 6
en comparación con Visual Basic 2008, 14
Rubi Formas de motor, 62
Visual Basic 2008. Véase también *temas específicos*
depuración, 139-154
Express Edition para el hogar usar, 325
consiguiendo para uso en el hogar, 325
en bucles, 187
nuevas características de, 22
como un desarrollo de ventanilla única tienda, 22
sitios en línea para información aproximadamente, 337-341
Código de Visual Basic, añadiendo funcionalidad a un formulario, 18
Visual Basic For Dummies (Www.vbfordummies.net)
Sitio web
aproximadamente, 3, 7
DateCalc servicio Web, 129-130, 132-133
DownloadFile método, 288-290
Enterprise Library Data
Acceda a Application Block
datos corredor, 269
información de registro, 295-296
Northwind base de datos, 32
como recursos, 337-338
SendEmail subrutina, 291-292
System.Drawing espacio de nombres, 302
ThreadingExample, 233
Visual Studio Express Edition, 14
WSDL (Web Services
Descripción del lenguaje), 136
Lenguaje Visual Basic, 1
Visual Basic.NET, 62
Visual Studio 2008
add-ins de, 329, 334
generación de código, 23
herramientas 23
la creación de un complemento para, 334
herramientas de depuración, 139-145
describe, 12
desarrollo de la energía en, 20-22
ediciones de 14,
extensibilidad de, 27
instalar, 14-15
no se requiere para Visual Basic programas, 24
en funcionamiento, 15
herramientas proporcionadas por, 1
- Los controles de validación, 98-99
Trabajando con el usuario interfaz, 311-323
Entorno de Visual Studio personalización, 35
aplicaciones que se ejecutan en y fuera de, 19
Integración de Visual Studio Programa, 329
herramientas visuales para la depuración, 139-145
salida de, 266-267
trabajar con, 262-265
Visual Web Developer (VWD)
 Web Server, 95-96
cadenas volátiles, 158
VWD (Visual Web Developer)
 Web Server, 95-96
- W •
- archivo w3wp.exe, 154
Reloj ventanas, 140, 143-144
relojes, 143
WCF (Windows Communication Foundation), 22, 326
Web, trucos para, 102-106
Dirección Web, pasando, 290
Las aplicaciones Web
 ASP.NET ahorro elementos de, 86
 edificio, 89-95
 creando en Visual Studio 2008, 20
 nombramiento, el 92
 corriendo, 95-98
 funcionamiento de, 106
Navegadores web
 dando un enfoque, 218
 hacer peticiones a la servidor, 85
Internet, en clases System.Net, 286
Diseñador web, diseño en el 91
Los formularios Web Forms
 debajo de la superficie de, 97-102
 en comparación con Windows Forms, 124
 define, 13
 extras en, 89-90
 puntos de vista, 89-90
Formularios Web
 mejores prácticas para la asegurar, 250
 edificio seguro, 247-250
 construir, 90-93
 depuración, 95-96, 150-152
 Estado en la preservación, 101-102
 Estado de, 85
- Formularios controles Web
propiedades disponibles de trabajo con, 92
valores salvaguardados por ASP.NET, 86
para aplicaciones Web, 88
Formularios diseños Web, en comparación a formularios Windows Forms, 84
Web Forms espacio de nombres, de ASP.NET, 88-89
Páginas Web
por el que se fuera, 91
decisiones en los formularios Web Forms, 314 pasando de uno a otro, 103-104
Elementos web, 332-334
Servidores Web
configuración y gestión de, 106
controles, 86-89
en .NET, 44
guardar archivos, 274
Servicio web del proyecto, la producción de, 128
Los servicios Web
edificio, 127-132
características de, 125
como una combinación de clase bibliotecas y Ventanas Formas, 124
consume, 127, 132-135
depuración, 153-154
define, 13
descrito, 44-45, 123-126, 135-137
para diseñar, 126-127
desarrollo, 127
partes de desarrollo, 123
produciendo, 128
peculiaridades, 125-126
utilizando las fuentes de datos, 258, 259
Descripción de servicios Web Language (WSDL), 136
Sitios Web
ASP.NET, 90-91, 128, 338
Canal 9, 338
Codeplex, 340
.Net247, 339
Google CodeSearch, 339
Google para la búsqueda, 340-341
kbAlertz, 340
Microsoft Pública de grupos de noticias, 339
MSDN Library, 15, 337
como una opción de proyecto en Visual Studio 2008, 20
Visual Basic For Dummies
aproximadamente, 3, 7
DateCalc servicio Web, 129-130, 132-133

Sitios Web (continuación)
DownloadFile método, 288-290
Enterprise Library Data
Acceda a Application Block
datos corredor, 269
información de registro, 295-296
Northwind base de datos, 32
como recursos, 337-338
SendEmail subrutina, 291-292
System.Drawing
espacio de nombres, 302
ThreadingExample, 233
Visual Studio Express
Edition, 14
WSDL (Web Services
Descripción del lenguaje), 136
Formularios Web que aparecen
como, 90
Web de Control de Usuario icono, 209
Control WebBrowser, 64
Directiva WebMethod, en una Web
servicio, 129
WebRequest clase, en
System.Net, 289
WebResponse clase, en
System.Net, 289
Objeto WebResponse, creando
método de miRequest, 288
Directiva WebService, 130
WebServiceBinding
directiva, 130
WebServiceBinding
declaración, 129
WebServicesX, 330
Cuando Hit columna, en la
Puntos de interrupción ventana, 142
While-End loop, saliendo, 196
Si bien comunicado, en un Do-
Loop 196,
ficha blanca, en vista de diseño, 25
números enteros, usando el
Tipo Integer, 170
Win32 capa, 13
De Windows 1,0, 108
Aplicaciones de Windows,
edificio, 65-68
Calculadora de Windows,
lanzamiento, 216
Windows Communication
Foundation (WCF), 22, 326
DLL de Windows, 124
Formulario Windows, la ejecución y
operativo, 18-20
Formularios Windows Forms
añadiendo
características a, 70-81
funcionalidad a, 68-70
comparado
a ASP.NET, 84
a formularios Web Forms, 124
controles en Visual Studio
2008, de 62
define, 13
permitiendo la interacción con el
usuario, 48
obtener un objeto Graphics
desde, 301
programa creado para
incorporar a Google
Herramienta de búsqueda, 215
como orientada a la interfaz de
usuario, 47
Aplicación de Windows Forms
plantilla, seleccionar, 65
Aplicaciones de Windows Forms
edificio
con capacidades de Internet,
286-296
descripción, 17-18
asegurar, 242-246
depuración, 149
describe, 62
funcionando dentro de Visual
Estudio, 19
el uso de controles en, 63
Proyecto de formularios Windows
Forms,
de partida, 15-17
Barra de herramientas de Windows
Forms,
arrastrar controles desde el 67
Ventanas interfaces estándar,
63
Inicio de sesión de Windows, la
autenticación
utilizando, 243-245
De administración de Windows
Instrumentation (WMI)
clases, 31
Windows Presentation
Fundación, 22, 326
Aplicación de seguridad de Windows,
edificio, 243-245
Windows Workflow
Fundación, 326, 334
WindowsBuiltInRole
enumerador, 245
Clase WindowsIdentity, 251
WindowsIdentity objeto,
104-105
WindowsPrincipal clase,
244, 251
Con la palabra clave, 221
WMI (Windows Management
Instrumentación) clases,
31
Documentos de Word, con la
apertura de
texto con formato, 314-315
Clase Word.Application, 212
Los flujos de trabajo, 334
espacio de trabajo, la

World Wide Web (WWW),
escribir programas en ejecución
en, 20

Escribir método, en la depuración
clase, 146

escrito

código de acceso, 266-269
artículos, 335

la lógica de negocio en un aparte
Archivo DLL, 109

código, 163, 239-251, 266-269

Aplicaciones de línea de comandos
con proyecto de consola
tipo, 216

código de datos, 266-269

DLL, 109, 228

código funcional, 163

bibliotecas de clases, el impacto de
sobrecarga en la escritura, 228

bucles, escribiendo con For-Next,

190-191

rutinas, escribiendo para su uso con
otro software, 21

ámbitos para las listas de viñetas, 51

secure code, 239-251, 293

plantillas para Office 2007, 21

planes de pruebas para las aplicaciones,

57-58

herramientas para escribir código en
Visual Studio, 36-39

World Wide Web por escrito,

programas

que se ejecuta en, 20

Writing Secure Code (Howard

y LeBlanc), 293

WSDL (Web Services

Descripción del lenguaje), 136

WWW (World Wide Web),

escribir programas en ejecución

en, 20

• X •

XML (eXtensible Markup
Language)

Copiar datos impulsada por, 254

describe, 136

Versiones XML de Windows

conceptos, 124

Servicios Web XML. Ver Web

servicios

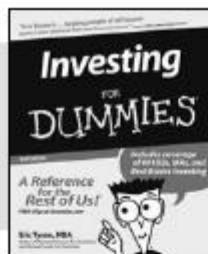
• Z •

Código Postal Servicio de información, 330

CARRERAS COMERCIALES, y finanzas personales



0-7645-9847-3



0-7645-2431-3

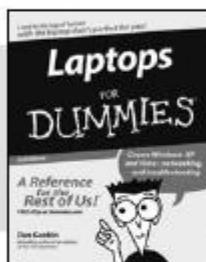
También disponible:

- ✓ Planes de Negocio Kit For Dummies
0-7645-9794-9
- ✓ Economics for Dummies
0-7645-5726-2
- ✓ Grant Writing For Dummies
0-7645-8416-2
- ✓ Home Buying for Dummies
0-7645-5331-3
- ✓ Gestión for Dummies
0-7645-1771-6
- ✓ Marketing For Dummies
0-7645-5600-2

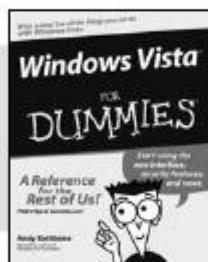
También disponible:

- ✓ Finanzas Personales Para Dummies
0-7645-2590-5 *
- ✓ Curriculum For Dummies
0-7645-5471-9
- ✓ Venta For Dummies
0-7645-5363-1
- ✓ Six Sigma For Dummies
0-7645-6798-5
- ✓ Kit Small Business For Dummies
0-7645-5984-2
- ✓ Comenzar un negocio eBay For Dummies
0-7645-6924-4
- ✓ Your Dream Career For Dummies
0-7645-9795-7

HOME & BUSINESS Computación Básica



0-470-05432-8



0-471-75421-8

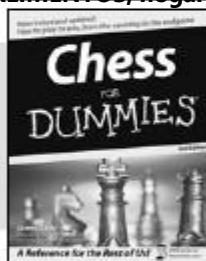
También disponible:

- ✓ Limpieza de Windows Vista Para Dummies
0-471-78293-9
- ✓ Excel 2007 Para Dummies
0-470-03737-7
- ✓ Mac OS X Tiger For Dummies
0-7645-7675-5
- ✓ MacBook For Dummies
0-470-04859-X
- ✓ Macs For Dummies
0-470-04849-2
- ✓ Office 2007 para Dummies
0-470-00923-3

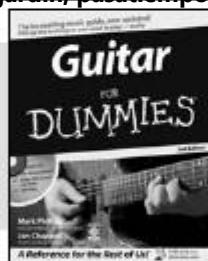
También disponible:

- ✓ Outlook 2007 para Dummies
0-470-03830-6
- ✓ PCs Para Dummies
0-7645-8958-X
- ✓ Salesforce.com For Dummies
0-470-04893-X
- ✓ Upgrading y fijación de los ordenadores portátiles para Dummies
0-7645-8959-8
- ✓ Word 2007 para Dummies
0-470-03658-3
- ✓ Quicken 2007 For Dummies
0-470-04600-7

ALIMENTOS, hogar, jardín, pasatiempos, música y mascotas



0-7645-8404-9



0-7645-9904-6

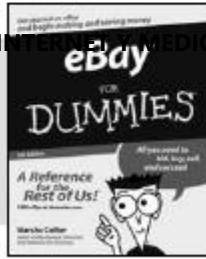
También disponible:

- ✓ Cómo hacer caramelos For Dummies
0-7645-9734-5
- ✓ Card Games For Dummies
0-7645-9910-0
- ✓ Ganchillo para Dummies
0-7645-4151-X
- ✓ Dog Training For Dummies
0-7645-8418-9
- ✓ Libro de Cocina Saludable Carb For Dummies
0-7645-8476-6
- ✓ Mantenimiento de la Vivienda Para Dummies
0-7645-5215-5

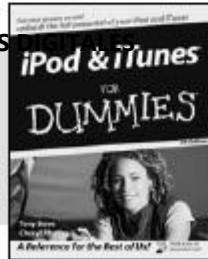
También disponible:

- ✓ Horses For Dummies
0-7645-9797-3
- ✓ Joyería & partida For Dummies
0-7645-2571-9
- ✓ Orchids For Dummies
0-7645-6759-4
- ✓ Puppies For Dummies
0-7645-5255-4
- ✓ Rock Guitar For Dummies
0-7645-5356-9
- ✓ Costura para Dummies
0-7645-6847-7

INTERNET Y MEDIOS



0-470-04529-9



0-470-04894-8

También disponible:

- ✓ Blogging For Dummies
0-471-77084-1
- ✓ Fotografía Digital para Dummies
0-7645-9802-3
- ✓ Fotografía digital All-in-One Desk Referencia Para Dummies
0-470-03743-1
- ✓ Digital SLR Cámaras y Fotografía For Dummies
0-7645-9803-1
- ✓ eBay Business All-in-One Desk Referencia Para Dummies
0-7645-8438-3
- ✓ HDTV For Dummies
0-470-09673-X

También disponible:

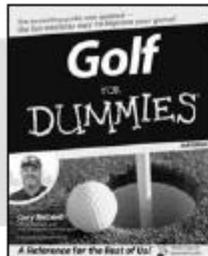
- ✓ Singing For Dummies
0-7645-2475-5
- ✓ Home Entertainment PCs For Dummies
0-470-05523-5
- ✓ MySpace For Dummies
0-470-09529-6
- ✓ La optimización del Search Engine para Dummies
0-471-97998-8
- ✓ Skype For Dummies
0-470-04891-3
- ✓ El Internet For Dummies
0-7645-8996-2
- ✓ Cableado del Hogar Digital para Dummies
0-471-91830-X

* Separe edición canadiense también está disponible

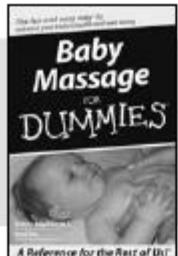
† Separada edición U.K. también disponible



DEPORTE, FITNESS, PADRES, RELIGIÓN Y ESPIRITUALIDAD



0-471-76871-5



0-7645-7841-3

También disponible:

- ✓ Catolicismo Para Dummies
0-7645-5391-7
- ✓ Ejercicio Balls For Dummies
0-7645-5623-1
- ✓ Fitness For Dummies
0-7645-7851-0
- ✓ Football For Dummies
0-7645-3936-1
- ✓ Judaísmo For Dummies
0-7645-5299-6
- ✓ Potty Training For Dummies
0-7645-5417-4
- ✓ Budismo For Dummies
0-7645-5359-3

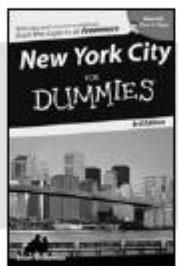
También disponible:

- ✓ Embarazo Para Dummies
0-7645-4483-7 +
- ✓ Diez Minuto Tone-Ups para Dummies
0-7645-7207-5
- ✓ NASCAR For Dummies
0-7645-7681-X
- ✓ Religión For Dummies
0-7645-5264-3
- ✓ Fútbol For Dummies
0-7645-5229-5
- ✓ Las mujeres en la Biblia para Dummies
0-7645-8475-8

VIAJAR



0-7645-7749-2



0-7645-6945-7

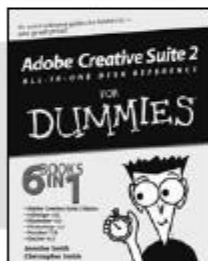
También disponible:

- ✓ Alaska For Dummies
0-7645-7746-8
- ✓ Vacaciones en crucero para Dummies
0-7645-6941-4
- ✓ Inglaterra For Dummies
0-7645-4276-1
- ✓ Europa For Dummies
0-7645-7529-5
- ✓ Alemania For Dummies
0-7645-7823-5
- ✓ Hawaii For Dummies
0-7645-7402-7

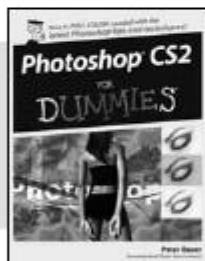
También disponible:

- ✓ Italia For Dummies
0-7645-7386-1
- ✓ Las Vegas For Dummies
0-7645-7382-9
- ✓ Londres For Dummies
0-7645-4277-X
- ✓ París For Dummies
0-7645-7630-5
- ✓ Vacaciones RV Para Dummies
0-7645-4442-X
- ✓ Walt Disney World y Orlando
For Dummies
0-7645-9660-8

GRÁFICOS, DISEÑO Y DESARROLLO WEB



0-7645-8815-X



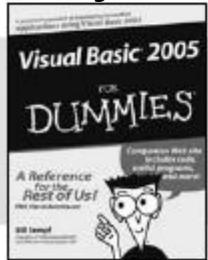
0-7645-9571-7

También disponible:

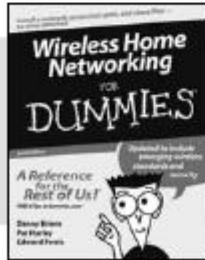
- ✓ Animación 3D Game For Dummies
0-7645-8789-7
- ✓ 2006 Para Dummies AutoCAD
0-7645-8925-3
- ✓ La construcción de un sitio Web para
Dummies
0-7645-7144-3
- ✓ Creación de Páginas Web Para Dummies
0-470-08030-2
- ✓ Creación de páginas Web All-in-One Desk
Referencia Para Dummies
0-7645-4348-8
- ✓ Dreamweaver 8 para Dummies
0-7645-9649-7

- ✓ InDesign CS2 para los maniquíes
0-7645-9572-5
- ✓ Macromedia Flash 8 para Dummies
0-7645-9691-8
- ✓ Photoshop CS2 y Digital
Fotografía para Dummies
0-7645-9580-6
- ✓ Photoshop Elements 4 para Dummies
0-471-77483-9
- ✓ Sindicar sitios web con RSS Feeds
For Dummies
0-7645-8848-6
- ✓ Yahoo! SiteBuilder para Dummies
0-7645-9800-7

Redes, seguridad, PROGRAMACIÓN Y BASES DE DATOS



0-7645-7728-X



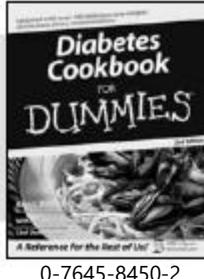
0-471-74940-0

También disponible:

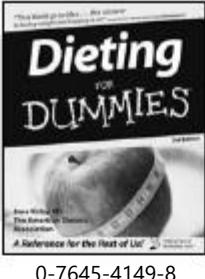
- ✓ Access 2007 For Dummies
0-470-04612-0
- ✓ ASP.NET 2 For Dummies
0-7645-7907-X
- ✓ C # 2005 For Dummies
0-7645-9704-3
- ✓ Hackear For Dummies
0-470-05235-X
- ✓ Hackear Redes Inalámbricas
For Dummies
0-7645-9730-2
- ✓ Java For Dummies
0-470-08716-1

- ✓ Microsoft SQL Server 2005 para Dummies
0-7645-7755-7
- ✓ La creación de redes All-in-One Desk
Reference
- ✓ For Dummies
0-7645-9939-9
- ✓ Prevenir el robo de identidad para Dummies
0-7645-7336-5
- ✓ Telecom For Dummies
0-471-77085-X
- ✓ Visual Studio 2005 All-in-One Desk
Referencia Para Dummies
0-7645-9775-2
- ✓ XML Para Dummies
0-7645-8845-1

SALUD Y AUTOAYUDA



0-7645-8450-2



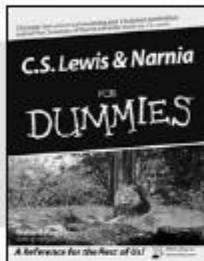
0-7645-4149-8

También disponible:

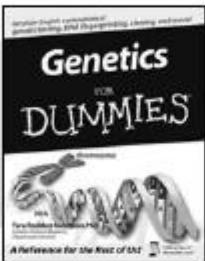
- ✓ Trastorno bipolar For Dummies
0-7645-8451-0
- ✓ La quimioterapia y la radiación For Dummies
0-7645-7832-4
- ✓ Controlar el colesterol For Dummies
0-7645-5440-9
- ✓ Diabetes Para Dummies
0-7645-6820-5 * †
- ✓ Divorce For Dummies
0-7645-8417-0 †

- ✓ Fibromialgia For Dummies
0-7645-5441-7
- ✓ Dieta baja en calorías para Dummies
0-7645-9905-4
- ✓ Meditación para Dummies
0-471-77774-9
- ✓ Osteoporosis For Dummies
0-7645-7621-6
- ✓ Superar Ansiedad para Dummies
0-7645-5447-6
- ✓ Reiki For Dummies
0-7645-9907-0
- ✓ Manejo del estrés para Dummies
0-7645-5144-2

Educación, historia, de referencia y de preparación de exámenes



0-7645-8381-6



0-7645-9554-7

También disponible:

- ✓ El ACT For Dummies
0-7645-9652-7
- ✓ Algebra For Dummies
0-7645-5325-9
- ✓ Algebra Workbook For Dummies
0-7645-8467-7
- ✓ Astronomy For Dummies
0-7645-8465-0
- ✓ Calculus For Dummies
0-7645-2498-4
- ✓ Química Para Dummies
0-7645-5430-1
- ✓ Forensics For Dummies
0-7645-5580-4

- ✓ Masones For Dummies
0-7645-9796-5
- ✓ Francés para Dummies
0-7645-5193-0
- ✓ Geometría For Dummies
0-7645-5324-0
- ✓ Química Orgánica I For Dummies
0-7645-6902-3
- ✓ El SAT I For Dummies
0-7645-7193-1
- ✓ Españoles For Dummies
0-7645-5194-9
- ✓ Estadística Para Dummies
0-7645-5423-9

A screenshot of the Smart@Dummies.com website. The header features the "DUMMIES.COM" logo and the tagline "The Online Resource for the Rest of Us®". The main content area includes sections for "It's Dummies Month!", "This Week's Selections", and "Dummies Promotions". On the left, there's a sidebar with links to various Dummies book categories like "Everyday Computing", "Advanced Computing", "The Internet", etc. At the bottom, there's a footer with links to "About Dummies", "Privacy Policy", and "Contact Us".

Cómo smart@dummies.com

®

- Encuentra una lista completa de títulos Dummies
- Busque en cargas gratuito en el artículo
- Regístrate GRATIS Etips e-mail a usted cada semana
- Ver qué otros productos llevan el nombre Dummies
- Compre directamente desde la tienda de libros Dummies
- Participa para ganar nuevos premios cada mes!



* Separe edición canadiense también está disponible

† Separada edición U.K. también disponible

h Dummies



**DVDs didácticos · Recopilaciones Música
Juegos y Novedades · Kits Culinarias
Manualidades y patrones de costura
Mejoras para el hogar / DIY Kits · y mucho
más!**

 **WILEY**

Echa un vistazo a la tienda especializada en maniquíes en www.dummies.com para más información!