

dotNetManía

www.dotnetmania.com

Dedicada a los profesionales de la plataforma .NET

Creación de asistentes en Visual Studio

Entrevista

Richard Waymire y Gert Drapers

Grupo de Visual Studio Team Edition for Database Professionals

Microsoft Corporation

Opinión

Métricas para la estimación de software: Puntos por función

Fundamentos de globalización • Mantener la sesión HTTP con .NET Framework • Impresión desde CRM con WordML y FetchXml • Seguridad: Gestión de tickets de sesión • Cómo personalizar el ribbon de Office 2007 • Delegados y eventos. Primera parte: ¿En quién delegas tú?

Comunidad.net

10 Años de El Guri

Laboratorio

Chart FX para Visual Studio 2005

TodotNet & QA

Resolución de problemas en ASP.NET

Microsoft

Tu potencial, nuestra pasión.™

TODO

lo que un desarrollador necesita:

Una idea brillante, algo para comer mientras la desarrollas y...

...Suscripciones MSDN

Porque con una Suscripción MSDN eres el primero en recibir las últimas versiones* –incluso Betas de producto– de las herramientas de desarrollo (Microsoft Visual Studio .NET, Visual FoxPro, Visual Studio Tools para Office System), versiones de desarrollo y prueba de TODOS los servidores de Microsoft (SQL, Exchange, Commerce, etc), los sistemas operativos (incluyendo Windows Server 2003), kits de desarrollo de software (SDKs) y drivers (DDKs), casi dos GB de documentación en CD o DVD, e incluso herramientas de productividad como Microsoft Office System, Project, Visio y MapPoint. Y todo ello actualizado periódicamente durante el periodo de vigencia de tu Suscripción.

Además, como beneficio adicional, los suscriptores tienen trato privilegiado en los grupos de noticias gestionados y hasta cuatro incidentes de soporte gratuito con el Servicio Profesional de Soporte para Desarrolladores.

Suscripciones MSDN te ofrece TODO

(bueno, o casi todo... aún no es posible alimentarse con CDs y DVDs)

Más información en: <http://www.microsoft.com/spanish/msdn/suscripcion>
o en el teléfono de Atención al Cliente 902 197 198

*Los productos recibidos varían en función del nivel de Suscripción MSDN adquirido

© 2004 Microsoft Corporation. Todos los derechos reservados. Microsoft, el logo Microsoft, Suscripciones MSDN, el logo MSDN y "Tu potencial, nuestra pasión" son marcas registradas o marcas comerciales de Microsoft Corporation en Estados Unidos y/o en otros países.



Distribuidores
Recomendados:



902 12 31 46
<http://ms.danysoft.com>



85 540 75 01
www.rambial.com



902 15 57 96
www.visualnet.net

dotNetManía

Dedicada a los profesionales de la plataforma .NET

Vol. III • Número 30 • Octubre 2006

Precio: 6,50€

Editor

Paco Marín

(paco.marin@dotnetmania.com)

Redactor Jefe

Marino Posadas

(marino.posadas@dotnetmania.com)

Editor técnico

Octavio Hernández

(octavio@dotnetmania.com)

Consejo de Redacción

Dino Esposito, Guillermo 'guille' Som, José Manuel Alarcón, Lorenzo Ponte, Luis Miguel Blanco y Miguel Katrib (Grupo Weboo).

Colaboradores habituales

Antonio Quirós, Braulio Díez, Carlos Quintero, Eladio Rincón, Javier Aragonés, Jorge Serrano, José Miguel Torres, Iván González, Pepe Hevia, Salvador Ramos y Sergio Vázquez

Además colaboran en este número

Andy González, Daniel Sabater, Fernando Guerrero, Joaquín Gracia, Jorge Ramo y Pablo Tilotta

Atención al suscriptor

Pilar Pérez

(pilar.perez@dotnetmania.com)

Ilustraciones

Yamil Hernández

Edición, suscripciones y publicidad

.netalia

c/ Robledal, 135

28529 Rivas-Vaciamadrid (Madrid)

www.dotnetmania.com

Tf. (34) 91 666 74 77

Fax (34) 91 499 13 64

Imprime

Color Digital

ISSN

1698-5451

Depósito Legal

M-3.075-2004



Lejos de convertirnos en cenizas, como parece sugerir la portada, estamos trabajando más que nunca en ofrecerle los mejores contenidos sin costes adicionales. Ya hemos comenzado la producción del Cuaderno Técnico nº5 de **dotNetManía** que junto con el sexto y séptimo serán gratis para nuestros suscriptores gracias en parte al patrocinio de MSDN en España. Los temas a tratar serán la optimización de aplicaciones ASP.NET, WPF y Sharepoint 2007.

Bienvenido al número 30, de octubre de 2006, de **dotNetManía**.

Muchas caras nuevas en este primer ejemplar de otoño. Por un lado, hay que apreciar en lo que vale la aportación de **Andy González**, quien nos habla de fundamentos de globalización de software. Andy es *International Program Manager* de Microsoft y durante mucho tiempo ha venido responsabilizándose de la localización de las herramientas de desarrollo de la compañía al castellano, entre otras lenguas. Y tiene valor doble porque sus honorarios serán cedidos a la *International Diabetes Federation* en su nombre.

También se estrenan **Daniel Sabater** con su "Impresión desde CRM con WordML y FetchXml", y **Joaquín Gracia** con su "Cómo personalizar el ribbon de Office 2007", y **Jorge Ramo**, de la inagotable cantera de buenos redactores que es **Raona**, con su "Seguridad: Gestión de tickets de sesión".

Este mes tenemos una entrevista de lujo a **Richard Waymire** y **Gert Drapers** del equipo de desarrollo de Visual Studio Team Edition for Database Professionals. **Fernando Guerrero** tuvo la ocasión de charlar con ellos durante el Tech-Ed 2006 de Boston.

Carlos Quintero toma protagonismo en la portada con su "Creación de asistentes en Visual Studio", una forma más de extender Visual Studio, algo en lo que Carlos es experto. No se pierdan cómo nuestra mascota intenta imitarle sin conseguirlo :-).

Si está comenzando con .NET, uno de los imprescindibles este mes es "Delegados y eventos. Primera parte: ¿En quién delegas tú?" de "El Guille".

No quiero terminar sin felicitar a mi amigo y compañero **Guillermo Som** por su décimo aniversario en la Web. Tal ha sido el empeño durante estos años en ayudar a tantos programadores, que este pequeño homenaje es lo menos que podíamos hacer por él. Si aún no lo conoce..., pero ¡qué digo! si le conoce todo el mundo :-).

Para el próximo número ya estamos preparando un pequeño monográfico sobre AJAX, ahora que Atlas, o por mejor decir, **Microsoft AJAX Library**, **ASP.NET 2.0 AJAX Extensions** y **ASP.NET AJAX Control Toolkit** están casi listos para salir del horno.

Como siempre, dentro hay más. Espero que le guste.

Paco Marín



Métricas para la estimación de software: Puntos por función 10-11

Medir o no medir, esa es la cuestión. En la Ingeniería del Software, el uso de métricas, tanto para la estimación como para el seguimiento de proyectos, se impone como una buena práctica esencial. En este artículo abordaremos esta cuestión desde la perspectiva de la estimación, dejando para otros de la serie, los aspectos relativos al seguimiento.

Entrevista a Richard Waymire y Gert Drapers 12-14

Durante la celebración de la pasada edición de Tech-Ed en Boston, tuvimos la ocasión de entrevistar a Richard Waymire, Program Manager/Architect y Gert Drapers, Development Manager/Architect en Microsoft. Ambos son los máximos responsables del nuevo Visual Studio Team Edition for Database Professionals, producto destinado a revolucionar la manera en que los administradores y desarrolladores de bases de datos y los desarrolladores de aplicaciones colaboran a un nuevo nivel cualitativo.

Fundamentos de globalización 16-18

El mayor beneficio de la globalización del software es que proporciona el acceso a usuarios de todo el mundo, porque la globalización permite el desarrollo de aplicaciones en varios idiomas y en varias culturas.

Mantener la sesión HTTP con .NET Framework 19-22

En este sencillo pero muy útil truco .NET, nos vamos a imponer el reto de subsanar una pega que muchos desarrolladores han encontrado al intentar crear aplicaciones que consumen información de servidores Web utilizando mecanismos de cierto bajo nivel.

Creación de asistentes en Visual Studio 23-28

En este artículo veremos una forma más de extensibilidad de Visual Studio, aparte de las macros y complementos que ya hemos visto con anterioridad. Se trata de los asistentes (wizards) que se introdujeron con la versión inicial de Visual Studio .NET. En artículos posteriores hablaremos de las novedades y cambios introducidos en Visual Studio 2005 (plantillas, kits de iniciación, etc.).

Impresión desde CRM con WordML y FetchXml 30-34

Se trata de una extensión de Microsoft Dynamics CRM que permite convertir la información de cualquier entidad CRM (oferta, factura etc) en un documento Word, para imprimir o enviar por correo electrónico. Esta alternativa a Reporting Services se basa en FetchXML y una plantilla WML a la que añadimos etiquetas XML que transformaremos con XSL para obtener el documento final.

Seguridad: Gestión de tickets de sesión 36-38

Supongamos que estamos trabajando en un entorno de autenticación en que se utilizan diversos datos biométricos (análisis de retina, huellas dactilares, análisis de voz, etc.), y tras aplicar una función sobre ellos se obtiene un identificador solo conocido por el sistema. Cualquier persona que quisiera suplantar nuestra identidad debería conocer todos estos datos para plantear un ataque con éxito, puesto que con un solo error en alguna de las variables el identificador resultante no sería válido. En este artículo veremos cómo plantear este escenario en nuestras aplicaciones ASP.NET imitando el modelo del protocolo Kerberos.

Cómo personalizar el ribbon de Office 2007 39-41

Siguendo los pasos de este tutorial podrás personalizar el ribbon de Office 2007 de una forma realmente sencilla y fácil. Explicamos cómo crear un add-in de Office 2007 basado en la interfaz IDTextensibility2 y la forma en la que puedes añadir pestañas y botones propios en el ribbon.

Delegados y eventos. Primera parte: ¿En quién delegas tú? 42-47

En este número vamos a tratar de los delegados, y también de los eventos (aunque de estos últimos nos encargaremos con más detalle en el próximo artículo), ya que en .NET están estrechamente relacionados; tanto es así que no podemos definir un evento sin la intervención de un delegado.

dnm.todotnet.qa. Resolución de problemas en ASP.NET 48-50**dnm.biblioteca.net** 51

*Programming ASP.NET 2.0 Core Reference. Dino Esposito
Applications = Code + Markup. Charles Petzold*

dnm.comunidad.net. 10 Años de "El Guille" 52-53**dnm.laboratorio. Chart FX para Visual Studio 2005** 54-55**dnm.desvan** 58

Somos
los mejores
mentores
al servicio de
su empresa

SQL

Solid Quality Learning

www.solidqualitylearning.com

Microsoft
GOLD CERTIFIED

Partner

Advanced Infrastructure Solutions
Data Management Solutions
Learning Solutions

dnm.sumario

LA DIRECCIÓN PRECISA



Empresas de todo el mundo confían en nosotros como partners estratégicos para el diseño y optimización de sus sistemas empresariales.

La combinación exclusiva de experiencia, conocimientos y metodología de los mentores de Solid Quality Learning garantiza el éxito de nuestros clientes a través de una adecuada transferencia de conocimiento.



- Arquitectura, diseño y optimización de sistemas basados en SQL Server 2005.
- Aplicaciones de inteligencia de negocio, ETL y minería de datos.
- Análisis de Arquitectura de Aplicaciones Empresariales en .NET 2.0
- Visual Studio Team System 2005.

Microsoft
GOLD CERTIFIED
Partner

SQL

Solid Quality Learning

Tenemos las mejores soluciones al servicio de su empresa. www.solidqualitylearning.com

Microsoft lanza la Release Candidate 1 de Windows Vista

Microsoft ha anunciado la disponibilidad de la versión Release Candidate 1 (RC1) de Windows Vista, ampliando el Customer Preview Program (CPP) a nuevos usuarios

El 1 de septiembre, Microsoft anunció la disponibilidad de la RC1 de Windows Vista (*build* 5600) a un grupo de probadores, incluyendo a los actuales participantes del *Customer Preview Program* (CPP). En total, se espera que más de cinco millones de usuarios en todo el mundo tengan acceso a la RC1 de Windows Vista.

A través de este programa, los desarrolladores y profesionales de TI que no tienen acceso a la RC1 de Windows Vista a través de otros canales pueden disponer del código para comenzar a analizarlo. La compañía también publicará la RC1 en sus espacios Web MSDN y TechNet para que pueda ser descargada por sus suscriptores.

Posteriormente, el 22 de septiembre, Microsoft liberó una nueva revisión (*build* 5728) para los probadores habituales así como para unos 100.000 partici-

pantes del CPP. Esta *build* incluye algunas mejoras y cambios relacionados con el *feedback* recibido desde el 1º de septiembre.

La información sobre los precios de las distintas versiones de Windows Vista, así como otras de sus características, se encuentra disponible en el espacio *online*: <http://www.microsoft.com/windows-vista/getready/editions>.

También la RC1 de .NET 3.0

El 1 de septiembre Microsoft liberó también la RC1 de .NET Framework 3.0. Combina la potencia de .NET Framework 2.0 con las extensiones revolucionarias antes conocidas bajo el acrónimo común WinFX: WPF, WCF, WWF y WCS. .NET 3.0 formará parte de Windows Vista, aunque también estará disponible para Windows XP y Windows Server 2003.

Visual Studio 2005 SDK versión 3.0 RTM

El pasado 15 de septiembre, Microsoft sacó oficialmente a la luz la versión 3.0 del SDK de Visual Studio 2005, una nueva “vuelta de tuerca” a la herramienta que permite a los desarrolladores integrar editores, diseñadores, nuevos lenguajes de programación y otros tipos de herramientas dentro del entorno integrado de Visual Studio 2005.

Entre las principales novedades que esta versión añade podemos mencionar:

- la versión 1.0 de DSL (*Domain Specific Language*) Tools,

- MPLEX y MPPG, herramientas para la generación de analizadores sintácticos y gramaticales en código manejado,
- nuevos contenidos relacionados con la extensibilidad de Team Test y Team Foundation Server,
- nuevos ejemplos relacionados con Iron Python.

Este SDK requiere la versión estándar o superior de VS 2005 y puede descargarse de: <http://www.microsoft.com/downloads/details.aspx?FamilyId=7E0FDD66-698A-4E6A-B373-BD0642847AB7&displaylang=en>.

Visual Basic 2005 Power Packs

Recientemente han “visto la luz” los **Visual Basic 2005 Power Packs**, un conjunto de complementos, componentes y utilidades destinados a simplificar y agilizar aún más el desarrollo de aplicaciones con Visual Basic 2005. Por el momento hay dos “packs” disponibles:

- **Microsoft Interop Forms Toolkit 1.0.** Kit de herramientas para facilitar la incorporación de formularios .NET a aplicaciones desarrolladas con Visual Basic 6. En vez de tener que actualizar todo el código de las aplicaciones, los desarrolladores ahora podrán ir migrando formulario a formulario.
- **Microsoft PrintForm Component 1.0.** Componente cuyo objetivo central es permitirnos imprimir fácilmente los formularios Windows.

Estos “packs” pueden descargarse desde <http://msdn.microsoft.com/vbasic/downloads/powerpacks/default.aspx>.

Además...

Roadmap y nombres para Atlas

La librería javascript Atlas del lado del cliente va a llamarse **Microsoft AJAX Library**. Trabaja con cualquier explorador y también soportará cualquier servidor Web. La funcionalidad de Atlas del lado del servidor que se integra con ASP.NET se llamará **ASP.NET 2.0 AJAX Extensions** y, por último, lo que hasta ahora conocemos como Atlas Control Toolkit se llamará **ASP.NET AJAX Control Toolkit**.

Según **Scott Guthrie** declara en su blog <http://weblogs.asp.net/scottgu>, la versión final para Atlas estará disponible a finales de año. Habrá que esperar a las betas y RC para conocer una fecha con más exactitud, pero en cualquier caso parece que no habrá que esperar a Orcas para tener la primera versión definitiva.

Visual Studio 2005 Tools for Office Second Edition Beta disponible

Visual Studio 2005 Tools for Office Second Edition Beta es un *add-on* de Visual Studio 2005 que permite a los desarrolladores construir aplicaciones para el 2007 Office System.

Cualquiera que tenga una licencia de VSTO o 2005 Tools for Office o Visual Studio Professional o superior puede descargarla gratuitamente desde: <http://msdn.microsoft.com/office/tool/vsto/2005SE>.

MSDN y campusMVP te invitan a certificarte



MSDN y campusMVP le ofrecen de forma conjunta esta oportunidad única de formarte a tu ritmo y desde cualquier lugar en las últimas tecnologías de desarrollo de Microsoft, y obtener la certificación oficial. Tanto si no está certificado como si ya posee una certificación anterior, este pack puede interesarle ya que está orientado a las nuevas certificaciones de .NET 2.0 que han sustituido a las antiguas.

Con el pack "Certificación" y solo hasta el 31 de octubre de 2006 obtendrá:

- Acceso a 6 cursos *online*, con las tutorías de los MVP y MCT de campusMVP.
- Un manual oficial de Microsoft (MOC) orientado a su especialidad.
- Un cupón con el que podrá examinarse gratis en cualquier parte del mundo para obtener su certificación.
- Gestión gratuita de una bonificación con el Estado para que el curso le salga realmente barato.

Para más información: <http://www.campusmvp.com> o en <http://www.microsoft.com/spanish/msdn/spain>.

"A C++/CLI Design Rationale" en castellano

Rafael Ontivero y Octavio Hernández, colaboradores de dotNetManía, han puesto a disposición del público una traducción al castellano de "A C++/CLI Design Rationale", de Herb Sutter.

Se trata de un documento de unas 50 páginas en el que se explican detalladamente las decisiones de diseño tomadas durante la concepción de C++/CLI, el enlace de C++ (el popular lenguaje de programación creado por **Bjarne Stroustrup** en los años 80 y estandarizado por ISO) con CLI (*Common Language Infrastructure*, la infraestructura común para múltiples lenguajes estandarizada por ECMA e ISO y cuya realización práctica más conocida es Microsoft .NET Framework). C++/CLI está implementado al 100% en el compilador de Visual C++ incorporado en la línea de productos de Visual Studio.

Una personalidad de enorme reputación en el mundo de la informática, **Herb Sutter** es ampliamente conocido como Secretario del Comité de Estandarización de C++ y por sus conferencias y libros relacionados con C++ y la programación concurrente. Desde hace cuatro años es Arquitecto de Software en Microsoft y fue el líder del equipo que creó C++/CLI.

La versión original del documento (en inglés) está disponible en <http://www.gotw.ca/publications/C++CLIRationale.pdf>, y la traducción al castellano puede descargarse de http://www.pokrsoft.com/cpp/a_design_rationale_es.zip.

Carta de Jim Allchin a los desarrolladores

Jim Allchin, Presidente para Plataformas y Servicios de Microsoft, ha escrito una carta abierta a los desarrolladores (titulada "Windows Vista: ¡ahora es el momento!") en la que los insta a prepararse para la próxima aparición de Windows Vista y de una nueva generación de herramientas destinadas a revolucionar la manera en que desarrollamos aplicaciones: la línea de productos Expression (acaban de

aparecer las CTP de septiembre de Interactive Designer y Graphic Designer y la beta 1 de Expression Web), las nuevas librerías que se integran en .NET Framework 3.0 y la próxima versión de Visual Studio ("Orcas").

Puede acceder a la carta desde el rediseñado centro para desarrolladores de Windows Vista en: <http://msdn.microsoft.com/windowsvista>.

Productos Turbo ya disponibles. ¡They're Back!



El Grupo de Herramientas de Desarrollo de Borland ha anunciado recientemente la disponibilidad de su nueva línea de productos Turbo, entornos integrados

para el desarrollo rápido de aplicaciones para Windows y .NET Framework (versión 1.1). Componen la línea Turbo Delphi y Turbo C++ (para el desarrollo Win32) y Turbo Delphi para .NET y Turbo C#, para el desarrollo de aplicaciones .NET.

Los productos Turbo son versiones de Borland Developer Studio 2006 basadas en un único lenguaje y plataforma y se ofrecen en dos ediciones:

- Las ediciones Turbo Explorer son de descarga gratuita y tienen algunas limitaciones, como la imposibilidad de incorporar al entorno componentes de terceros. Han sido concebidas principalmente como herramienta de aprendizaje.
- Las ediciones Turbo Professional, con un precio aproximado de 400€, sí son extensibles y personalizables y ofrecen todas las posibilidades del BDS 2006 para el lenguaje y plataforma elegidos. En el caso de los productos para .NET Framework, éstos permiten crear todo tipo de librerías, aplicaciones y servicios tanto para Windows como para la Web.

Para más información y descargas, visite www.borland.com/us/products/turbo/index.html.

Iron Python 1.0

Iron Python es la nueva implementación del lenguaje de programación Python para su ejecución bajo la plataforma .NET. Soporta una consola interactiva con compilación totalmente dinámica, así como integración con .NET Framework, con lo que pone a disposición de los programadores de Python las librerías de .NET al mismo tiempo que mantiene la compatibilidad completa con el lenguaje Python.

Puede descargarla desde el sitio CodePlex en la dirección: <http://www.codeplex.com/Release/ProjectReleases.aspx?ProjectName=IronPython&ReleaseId=423>. Más información también en el blog de Jim Hugunin: <http://blogs.msdn.com/hugunin/archive/2006/09/05/741605.aspx>.

Suscríbase y llévase los tres próximos libros gratis

dotNetManía

aparte del CD Vol. 2 con los números 12 al 22 en PDF



□ N°16 (6,50€)



□ N°17 (6,50€)



□ N°18 (6,50€)



□ N°19 (6,50€)



□ N°20 (8,50€)



□ N°21 (6,50€)



□ N°22 (6,50€)



□ N°23 (6,50€)



□ N°24 (6,50€)



□ N°25 (6,50€)



□ N°26 (6,50€)



□ N°27 (6,50€)



□ N°28 (6,50€)



□ N°29 (6,50€)

Oferta válida hasta el 31 de octubre de 2006 o hasta agotar existencias

- ☐ Deseo suscribirme a **dotNetManía** por un año (11 números) por un precio de **65,00€ IVA incluido** y recibir gratuitamente los tres próximos cuadernos técnicos que publique **Netalia**, aparte del **CD Volumen 2** con los ejemplares desde el 12 al 22 en formato PDF de alta calidad de forma gratuita. Si su dirección está fuera de España consulte los detalles en www.dotnetmania.com
- ☐ Deseo que me envíen los **números atrasados marcados** según el precio de portada. Otros: _____

DATOS DE ENVÍO

CIF/NIE Empresa
 Nombre y apellidos
 Dirección
 Población Código Postal Provincia
 Teléfono Fax email

DATOS DE FACTURACIÓN (sólo si son distintos a los datos de envío)

CIF/NIE Empresa
 Nombre y apellidos
 Dirección
 Población Código Postal Provincia
 Teléfono Fax email

FORMA DE PAGO

- ☐ Talón nominativo a nombre **NETALIA, S.L.**
- ☐ Transferencia bancaria a nombre de **NETALIA, S.L.** a:
 La Caixa - Número de cuenta **2100 4315 48 2200014696** (Indique su nombre en la transferencia)
- ☐ Domiciliación Bancaria (con renovación automática, previo aviso)
 Indique su número de cuenta: _____
- ☐ Tarjeta de crédito
☐ VISA ☐ MASTERCARD
 Número de su tarjeta: _____
 Fecha de caducidad: ____/____/____ (imprescindible)

Firma y/o sello

a de

de 2006

Puede enviar sus datos por Fax al 91 499 13 64, o por teléfono al 91 666 74 77, o por email a la dirección suscripciones@dotnetmania.com, o también puede enviarlos por correo postal a la siguiente dirección:

Netalia, S.L.
 C/ Robledal, 135
 28529- Rivas Vaciamadrid (Madrid)

Usted autoriza a la mecanización de estos datos. El responsable y destinatario de éstos es Netalia, S.L. Usted tiene derecho a acceder a sus datos, modificarlos y cancelarlos cuando lo desee. Sus datos no serán cedidos en ninguna de las formas posibles a terceras partes y no se utilizarán más que para el buen funcionamiento de su suscripción a la revista **dotNetManía** y para informarle de las actividades comerciales que realice la editorial Netalia, S.L. Si no desea recibir información comercial de **dotNetManía** marque la casilla siguiente ☐

OHMMMM



HAY MEJORES MANERAS DE ESTAR CONECTADO

acensExchange

Avanzada solución de correo y colaboración para empresas, desde dispositivos fijos y móviles

- Gestión total de correo, agenda, contactos, tareas, notas y carpetas compartidas
- Accesible desde su escritorio, cualquier navegador o dispositivo móvil
- Elevada seguridad: antivirus, antispam, copias de seguridad
- Facilidad de uso: panel de control para el administrador y cada usuario
- Reducción de costes: según el número de usuarios y sin inversión inicial
- acensExchange, basado en Exchange Server 2003 de Microsoft



Otros servicios: HOSTING DEDICADO • HOUSING
• REDES PRIVADAS • BACKUP Y CONTINUIDAD DE NEGOCIO



Data centers



en Madrid



y Barcelona

pídanos asesoramiento sin compromiso

www.acens.com 902 90 10 20 comercial@acens.com

acens
comunicaciones IP al servicio de su empresa



Antonio Quirós

Métricas para la estimación de software: Puntos por función

Medir o no medir, esa es la cuestión. En la Ingeniería del Software, el uso de métricas, tanto para la estimación como para el seguimiento de proyectos, se impone como una buena práctica esencial. En este artículo abordaremos esta cuestión desde la perspectiva de la estimación, dejando para otros de la serie, los aspectos relativos al seguimiento.

» Como ya comentábamos en el artículo de inicio de esta serie, el uso de métricas para la estimación de software es uno de los signos que con más claridad denota la madurez de la empresa que las emplea. Muchos equipos de desarrollo carecen de instrumentos eficaces para estimar la carga de trabajo que sus proyectos les supondrá. Esto, por supuesto, redundará en que no se podrá presupuestar eficazmente ni tampoco planificar de forma rigurosa el trabajo de los recursos necesarios para el proyecto.

Para subsanar este problema el método más usado es el *comparativo*. Es decir, que dado el histórico de proyectos realizados por un equipo de trabajo, cuando surge uno nuevo lo comparamos con aquellos que se parecen y sobre esta comparación estimamos el esfuerzo que se necesitará para llevarlo a cabo. De la estimación del esfuerzo saldrá de forma fácil la estimación del coste. Esto puede parecer algo rudimentario, sobre todo cuando nos movemos en un mundo técnico cuya concreción se realiza por ingenieros que, en teoría, deberían ser el paradigma de la precisión.

Por otro lado, el modo comparativo de hacer las cosas carece de la objetividad necesaria si siempre se compara con los resultados históricos del mismo equipo de trabajo, ya que sus ratios de productividad pueden estar totalmente sesgados. Este método sería solo válido si realmente pudiera realizarse una comparación entre distintos equipos. Con esta finalidad surge el denominado método de los *Puntos por función*, descrito inicialmente por Allan Albretch de IBM en 1979. Este método está

La objetivación del proceso de estimación hace que podamos hablar con nuestro cliente un lenguaje común

basado en el criterio comparativo que hemos mencionado, pero de modo universal, lo que le dota de los criterios de objetividad que echábamos en falta anteriormente.

Con *Puntos por función*, lo que hacemos es puntuar los distintos componentes del software que realizamos según ciertos parámetros: tipo de proceso, número de ficheros que se ven afectados, entradas y salidas, etc. Esto nos lleva a que cada función unitaria tenga un valor en puntos según su tipo. Pero, además existen coeficientes correctores para cada tipo de aplicación, para cada lenguaje de programación empleado, etc. Todo esto nos lleva a que, dada una aplicación, tenemos un número de puntos que miden su complejidad. A partir de aquí, sólo tenemos que hacer las extrapolaciones en cuanto a las horas de trabajo; es decir, equivalencias del tipo: 1 punto se corresponde con n horas de trabajo, o 1 punto de función son n euros a presupuestar al cliente.

Ventajas

- La objetivación del proceso de estimación hace que podamos hablar con nuestro cliente un lenguaje común. Nadie piensa que le está engañando nadie, cosa común en los procesos de desarrollo de software, donde para un mismo alcance de proyecto es bastante frecuente que los presupuestos de distintas compañías sean netamente diferentes.
- Esta misma objetivación hace que la planificación de la carga de trabajo pueda realizarse con rigor. Los miembros del equipo del proyecto se sienten más cómodos con estimaciones bien hechas, se evitan los continuos esfuerzos adicionales por una estimación dudosa y se pueden fijar reglas claras de incentivos con un entorno fiable y común entre la empresa y el empleado.
- La valoración en puntos por función de una aplicación nos permite adicionar un conjunto de métricas posteriores muy interesantes. Por ejemplo, los defectos por punto, las intervenciones de mantenimiento por punto, la media histórica de esfuerzo para desarrollar un punto, etc. Tradicionalmente, estos indicadores se han calculado por LOC (*Line of code*), sin embargo la visión de *Puntos por función* permite un acercamiento al ámbito funcional de la aplicación que aporta mucha más riqueza que la mera unidad de *peso* que supone la línea de código.
- Cuando se dan procesos de subcontratación, el punto por función es una unidad de medida más que razonable para valorar la aportación de la empresa subcontratada. En este ámbito existen experiencias interesantes. Por ejemplo, recientemente he estado en contacto con una compañía que ha desarrollado una, así denominada, *máquina de programar*, basada en MDA y que permite bajo un sistema Web recoger todas las instrucciones de diseño, de forma que éstas son valoradas en puntos por función y si queremos que la máquina genere el código correspondiente, el importe económico que dicha empresa nos repercute está basado en este estándar de medida. Así, el sueño de poder encargar por la Web una aplicación esta tarde, pagarla con la tarjeta de crédito y disponer de ella de forma inmediata parece que no está tan lejano.

Inconvenientes

- Es necesario tener bien determinadas las funciones que va a tener una aplicación, lo que nos ha debido conducir de forma previa e inexorable a realizar una parte importante del esfuerzo de análisis antes de realizar la estimación global del proyecto. Esto nos lleva a un modelo de contratación diferenciado entre

En cualquier caso, es una excelente práctica para cualquier compañía habituarse a estimar sus aplicaciones de forma rigurosa

diseño y construcción que muchas compañías no están dispuestas a seguir.

- La valoración de un proyecto en puntos por función es una sobrecarga adicional de trabajo cuyo coste hay que soportar.
- Cada vez que se producen cambios en el alcance de la aplicación, la estimación en puntos por función tiene que ser reconsiderada, si queremos tener una versión actualizada de la misma. Esto es tiempo de trabajo adicional.
- Debemos dotar a nuestros analistas y jefes de proyecto con una competencia nueva, la de saber estimar los puntos para cada función de sus aplicaciones.

En cualquier caso, creo que es una excelente práctica para cualquier compañía que desarrolle software el hecho de que forme a sus profesionales en esta técnica y se habitúe a estimar sus aplicaciones de forma rigurosa. Seguro que las ventajas que obtendrá serán enormes y ganará en rigor y precisión con sus proyectos. No podemos estar continuamente hablando de lo problemático que resulta el proceso de desarrollar software sin ir dividiendo el gran problema en pequeños subproblemas e ir atacando cada uno de ellos de forma racional e inteligente. Y no cabe duda que la delimitación clara del alcance de una aplicación y disponer de un criterio objetivo para la estimación de la carga de trabajo es uno de los elementos clave que contribuirá de forma impactante en la calidad de lo que hacemos como empresas que desarrollan software y, por tanto, en la confiabilidad que el mercado deposita en nosotros. ☺

Referencias

- [1] Garmus, David y Herron, David. *Function Point Analysis. Measurement Practices for Successful Software Projects*. Addison Wesley, 2000.
- [2] IFPUG (*International Function Point User Group*): www.ifpug.org (aquí puede encontrarse un amplio catálogo bibliográfico sobre el tema).
- [3] Asociación Española de Métricas del Software: www.aemes.fi.upm.es.



Fernando Guerrero

Entrevista a Richard Waymire y Gert Drapers

Durante la celebración de la pasada edición de Tech-Ed en Boston, tuvimos la ocasión de entrevistar a Richard Waymire, Program Manager/Architect y Gert Drapers, Development Manager/Architect en Microsoft. Ambos son los máximos responsables del nuevo Visual Studio Team Edition for Database Professionals, producto destinado a revolucionar la manera en que los administradores y desarrolladores de bases de datos y los desarrolladores de aplicaciones colaboran a un nuevo nivel cualitativo.

>> **Vamos a empezar** a empezar hablando un poco sobre Team Edition para Profesionales de Bases de Datos. ¿Qué nos puedes decir en general sobre este nuevo producto?

Richard Waymire: Empecemos esclareciendo por qué lo estamos haciendo. Si se analiza el mercado de bases de datos actual, hay una gran necesidad de un mayor control de versiones. Para un servidor de producción de SQL cualquiera, ¿cuánta gente sabe qué es lo que de verdad ha sido puesto en producción?, ¿cuánta gente sabe qué versión del esquema se está utilizando actualmente?, ¿existe en general una estrategia de versionado de esa base de datos? Estas son preguntas que la gran mayoría de la gente no puede responder, y los que pueden es porque han desarrollado soluciones a medida que les han requerido un gran esfuerzo y frecuentemente no cubren todas las necesidades. Lo que estamos haciendo es crear un producto que ofrezca una vía para responder a estas preguntas. Y por ello tenemos una gran integración con el resto de Team System, para no obtener una solución aislada que resuelva únicamente el problema de SQL, sino para poder integrar a desarrolladores de bases de datos y desarrolladores de aplicaciones en el mismo ciclo de desarrollo. El trabajo que se estaba haciendo hasta el momento no tenía en cuenta esa integración. A partir de ahora podremos tener el esquema de la base de datos *offline* para poder trabajar con él y compararlo con la versión de producción, para saber las diferencias o para saber periódicamente si alguien ha realizado algún cambio. De este modo se hace posible trabajar sin interrumpir la producción de SQL.

¿Crees que la comunidad de DBA va a recelar de esta herramienta, pensando que como proviene del lado de los desarrolladores éstos los van a llevar al “lado oscuro”?

RW: Sí, eso es cierto. Mucha gente va a ser muy escéptica. Hay algunas cosas que van a hacerse desde Visual Studio y eso es algo a lo que los DBA no están acostumbrados, aunque ya estén usando BI Development Studio y SQL Server Management Studio. Lo que intentamos decir a este respecto es que Visual Studio es un *shell* perfecto para la administración y el desarrollo. Lo que yo creo que va a ocurrir es que los DBA van a ver un gran valor en esta integración con Visual Studio y con los equipos de desarrollo. La integración se va a producir de manera que el DBA pueda comunicar a los desarrolladores el estado de la base de datos, sea capaz de hacer pruebas y despliegue al servidor junto con los desarrolladores, y éstos puedan hacer sugerencias y optimizaciones que el DBA pueda testear y validar fácilmente. De esta forma se rompen las barreras entre DBA y desarrolladores.

Una de las cosas que me alegran de vuestra participación en este producto es que generalmente este tipo de herramientas suelen ser desarrolladas desde una visión de desarrollador. ¿El hecho de que vosotros dos estéis en el equipo de desarrollo asegura una visión de DBA?

RW: Sí, ambos venimos de SQL, y somos muy afortunados de tener un equipo mixto; gente que viene de todas partes del Team Suite, e incluso personas que han trabajado en el núcleo de .NET Framework. Gert y yo tenemos una visión muy clara de lo que queremos, al igual que el resto de los miembros del equipo, vengan de donde vengan.

¿Cómo de grande es vuestro equipo?

RW: No es muy grande: algo más de 40 personas. No llega al tamaño del equipo de herramientas de SQL.

¿Eso significa que Microsoft no lo considera un producto muy importante?

RW: Me han preguntado ya esto anteriormente durante esta semana. Simplemente hemos comenzado a trabajar de la siguiente forma: alguien decidió que merecía la pena desarrollar este producto, nos dieron los recursos para contratar a gente clave en la industria, además de gente con mucha experiencia en Microsoft, y nos pusimos en marcha.

No sé si tenéis planes al respecto, pero creo que hay un gran campo por delante en lo relativo a educar a los profesionales de bases de datos no solo sobre las herramientas, sino también sobre los procesos.

RW: Ese es uno de nuestros retos más grandes. Hoy frecuentemente el propio DBA es la versión actual de lo que hay en producción, y no es bueno para las organizaciones que una sola persona sea quien tenga el control de la versión actual. Lo que tiene sentido es tratar las versiones de una base de datos SQL como al código fuente y almacenarlo en un sistema de control de versiones. Lo que estamos haciendo no es sino poner la base de datos allí donde está el resto de cosas en una organización, es decir, bajo control de versiones.

Pasando a un tema más personal, ¿cómo te ves en el futuro luego de ser un gurú en seguridad y disponibilidad de SQL Server y ahora estar en una posición más de desarrollador?

RW: Pues creo que he podido aportar algo al equipo; soy una persona con un conocimiento profundo de SQL Server, y conozco también a muchos clientes. Además, no estoy en un ambiente de desarrollo puro, sino que mi trabajo también contiene aspectos de administración de bases de datos y de administración del ambiente de desarrollo. Según mi experiencia, no creo que exista un administrador “puro” de SQL Server; los que conozco siempre están escribiendo código SQL. Soy muy feliz por estar trabajando en el desarrollo de una gran experiencia y por tener la oportunidad de pensar no solo en lo

Visual Studio Team Edition para Profesionales de Bases de Datos

Visual Studio Team Edition para Profesionales de Bases de Datos es un nuevo producto de Microsoft que tiene como objetivo central la integración del diseño y mantenimiento de las bases de datos de la empresa dentro del proceso integral de gestión del ciclo de vida de las aplicaciones corporativas mediante Visual Studio Team System. Las nuevas posibilidades que ofrece, como son, por ejemplo, la gestión del versionado de las bases de datos dentro del control de versiones común o la aplicación de las tecnologías de pruebas unitarias y de otros tipos a las bases de datos del sistema harán de este producto una herramienta fundamental de cara a la mejora de la calidad del software.

Para más información, visite <http://msdn.microsoft.com/vstudio/teamssystem/dbpro>.

que estamos haciendo en este momento, sino también en lo que vamos a hacer dentro de 1, 2 ó incluso 5 años.

¿Cómo ves el futuro del producto?

RW: Con que solo nos pusiéramos a trabajar en la lista de cosas que la gente aquí en Tech-Ed nos ha solicitado, tendríamos trabajo para 5 años. Aunque en realidad lo que me preocupa no es si vamos a tener trabajo o no, sino más bien que sepamos priorizarlo de la mejor manera para hacer las cosas en el orden óptimo. Pero ahora en lo que estamos centrados realmente es en entregar esta versión a nuestros clientes a final del año.

Este producto ayuda a mejorar los procesos y las aplicaciones. Tu experiencia de trabajo en el mundo real te ha dado una gran ventaja en este aspecto, ¿no?

RW: Lo cierto es que sí, y espero poder seguir ayudando con mis conocimientos. Ahora voy a estar un tiempo viajando, conociendo a algunos clientes y estudiando sus procesos de negocio y sus procesos de desarrollo. Estamos seguros de que ningún producto es perfecto para todo el mundo, pero tenemos que lograr que el producto sea lo suficientemente extenso y flexible para la mayoría de nuestros clientes.

Entra Gert en la entrevista...

Estábamos muy interesados en tener una entrevista con usted. ¿Cuáles han sido los obstáculos más grandes que habéis encontrado al desarrollar este producto?

Gert Drapers: Uno de los retos más grandes que hemos tenido que afrontar consiste en que para poder definir lo que teníamos que hacer teníamos que tener un completo conocimiento del esquema (tablas, vistas, procedimientos, funciones, índices, triggers, ensamblados...). La única manera de llegar ahí es analizando

todo de un modo integral; de modo que el mayor reto ha sido el de crear una infraestructura de análisis que nos permita ver el T-SQL del usuario e interpretarlo, para crear un grafo de dependencias que luego se utiliza a través de todo el producto. Esto ha sido muy difícil por las siguientes razones:

1. T-SQL es un lenguaje muy extenso; no es como C#, que ofrece una cantidad limitada de sentencias diferentes.
2. T-SQL es muy ambiguo.

El código T-SQL no es muy predecible, ¿cierto?

GD: Exacto. Hay que hacer mucho análisis y revisar hasta 4 niveles de código para poder entender lo que significa el código leído; algo realmente muy complejo.

Otro gran reto ha sido el cambio de mentalidad que nos lleva desde la idea de que todo reside dentro de la base de datos hasta el nuevo concepto en el que todo reside en el Team System. Todos los cambios se realizan en modo *offline*, desconectados de la base de datos, lo que hace posible que podamos hacer pruebas sin afectar nada, para luego, cuando estemos satisfechos con los resultados, ponerlos en producción.

Un proyecto como éste surge del hecho de que muchas personas quieren probar diferentes formas de modificar el esquema y las consultas para mejorar el rendimiento antes de ponerlas en práctica. ¿Qué herramientas de este tipo tiene Team System para los desarrolladores?

GD: En este momento, desde el punto de vista de rendimiento no proveemos muchas herramientas. Estamos añadiendo un plan de ejecución para el editor de consultas de modo que esa información esté disponible; sin embargo, si se dese-



Fernando Guerrero, Gert Drapers y Richard Waymire

ara utilizar esta herramienta desde la perspectiva del testeo se debería comenzar por hacer cosas como medir los tiempos de ejecución de ciertas consultas —esta posibilidad la tenemos ya incorporada al producto—. También están disponibles algunas posibilidades de efectuar test guiados por los datos (*test driven*); pero nos queda algún trabajo aún en lo relativo a las pruebas unitarias. Definitivamente, queremos que los usuarios puedan incorporar el análisis del rendimiento al ciclo de desarrollo; esto es algo en lo que estamos trabajando.

Ya están disponibles en la Web algunos documentos y una versión preliminar (*Community Preview*). ¿Qué sugerencia tiene para alguien que quiera comenzar a utilizar el producto de inmediato?

GD: El producto requiere un cambio mental, de tal forma que un primer paso en la dirección correcta sería comenzar a pensar en el mantenimiento del esquema fuera de la base de datos. Este cambio mental será más difícil para las personas que aún mantienen todo dentro de sus bases de datos, y que utilizan una herramienta de conexión para poder hacer cambios en su infraestructura. Team System será capaz de importar sus scripts para realizar una ingeniería inversa, de modo que el usuario pueda trabajar con la base de datos en un ambiente aislado, sin que se cambie el script de producción.

Uno de los problemas potenciales que veo es que algunos DBA no están acostumbrados a aprender procesos; ellos únicamente escriben código en Management Studio, sin haber apren-

dido que deben realizar pruebas antes de salir a producción.

GD: En los últimos años ha habido grandes cambios en la industria en la línea de la regulación del uso de procesos. Existen herramientas que ofrecen cierta ayuda para realizar cambios directamente sobre el esquema, pero que no ayudan en cada uno de los pasos del proceso y no se integran con un sistema de seguimiento de procesos o un administrador de procesos como es Team Foundation. Nosotros no obligamos a utilizar una metodología en particular; muchos clientes nos han preguntado si existe un proceso al que deban amoldarse, o si hay alguna sugerencia al respecto; pues aún no lo tenemos. Una de las preguntas que nos han hecho en las presentaciones ha sido si pensamos proveer a los usuarios de un estándar de codificación y si se desarrollará una herramienta para forzar a utilizarlo. Este es el tipo de cosas que los clientes quieren; no son tan difíciles, solo que conllevan mucho trabajo.

¿Qué nos puedes contar acerca del desarrollo de este producto?

GD: La razón principal por la que me incorporé a este equipo es que podía hacer algo impactante para la industria en 18 meses. Creo en los pequeños pasos y eso es lo que hemos estado haciendo: escuchamos a los clientes, implementamos algo, hacemos demos y luego escuchamos de nuevo, lo que asegura una gran retroalimentación. Acabamos de hacer algunos cambios sustanciales basados en algunos comentarios que hemos recibido, y eso demuestra nuestra flexibilidad para tra-

bajar en un producto destinado a satisfacer las necesidades del cliente.

¿Cómo se compara este producto con el de otras empresas?

GD: La oferta de nuestro producto no se basa en características, sino que más bien es un producto integrado; se trata de cómo se integran sus partes. Si se analiza lo que hay en el mercado, podremos encontrar productos con algunas funcionalidades similares a las de nuestro producto; pero nadie lo integra todo en el ciclo de desarrollo. Nuestro objetivo central como parte del equipo de Team System es suministrar herramientas de gestión del ciclo de vida para el desarrollador de la base de datos. No obstante, hay que señalar que somos los primeros en la industria en incorporar características como la refactorización de objetos del esquema. Las herramientas de comparación de esquemas existen desde hace algún tiempo y muchas personas las tienen a disposición; sin embargo nadie ha utilizado aún una de esas herramientas en un proyecto *offline*, como nadie permite aún comparar el contenido de un archivo T-SQL con una tabla física dentro de la base de datos.

Mirando un poco hacia atrás, ¿te habría sido útil este producto anteriormente?

GD: Por supuesto; en algunas de estas cosas he estado pensando desde hace mucho tiempo. Esta herramienta habría sido ideal para mí en muchos de los proyectos en los que he trabajado, porque habría podido tener una representación fuera de línea de la base de datos para poder trabajar en cualquier momento. Es una herramienta sumamente útil, independientemente de que se trabaje de forma individual o en equipo.

¿Qué significa para ti pertenecer ahora al equipo de Team System después de tantos años dentro del equipo de SQL?

GD: Aunque formalmente soy miembro del equipo de Team System, para mí este trabajo sigue teniendo mucho que ver con el mundo de las bases de datos, y eso me mantiene motivado. Queremos cumplir con los objetivos trazados y terminar este producto, y sin lugar a dudas lo lograremos.

¡Muchas gracias a ambos! ○

Naturally Integrated.[™]

(Naturalmente Integrado.)



Chart FX en Visual Studio 2005

ABOX

C/Manso 26-28, 2a planta
08015 Barcelona
Tel.: 93 426 22 57

www.abox.com

La Herramienta para Visualización de Datos más Fuertemente Integrada a Visual Studio 2005

Evidencia de ello es el "Smart Tag Wizard", que expone fácilmente al programador muchas de las propiedades disponibles. También existe el "Data Wizard" que le permite conectar rápidamente su gráfico a cualquier fuente de datos (data source), configurar los campos e indicar cómo utilizar los datos. La nueva interfaz de programación de aplicaciones (API) fue diseñada tomando en cuenta el modelo de objetos de Visual Studio 2005, para simplificar el acceso a funciones complejas. Chart FX for Visual Studio 2005 muestra sus gráficos utilizando DHTML y la técnica avanzada de desarrollo Web llamada AJAX (Asynchronous JavaScript and XML), para permitir una amplia interactividad en gráficos mostrados como imágenes. El nuevo Manejador de Extensiones (Extensions Manager) estandariza la infraestructura para utilizar las extensiones de Chart FX, tales como Financial, Statistical, Maps y OLAP, entre otras. El Centro de Recursos de Chart FX (Resource Center) provee en una simple herramienta, la Guía del Programador, el API de Chart FX, una Referencia de Internet y un sinnúmero de ejemplos y gráficos con código. Para mayor información acerca de la integración sin esfuerzo y otras características poderosas de este producto, visite nuestro sitio en la Web www.softwarefx.com.


Chart FX
for Visual Studio[®] 2005

www.softwarefx.com

Software FX, todos los derechos reservados.
Chart FX es una marca registrada de Software FX, Inc.
Todos los otros marcas son propiedad de sus
respetivos propietarios.



Andy González

Fundamentos de globalización

El mayor beneficio de la globalización del software es que proporciona el acceso a usuarios de todo el mundo, porque la globalización permite el desarrollo de aplicaciones en varios idiomas y en varias culturas.

» La globalización es el proceso que habilita al software para manipular, procesar y mostrar información en culturas internacionales más fácilmente; esto quiere decir que la información con que se trabaja es relativa a la referencia cultural, incluyendo lo relacionado al idioma, el país o región, los calendarios utilizados, los modelos de formato para las fechas, la moneda y los números y el criterio de ordenación de las cadenas. La clase **CultureInfo** de .NET Framework proporciona este tipo de funcionalidad útil para desarrollar aplicaciones globalizadas (internacionalizadas).

La mejor manera para entender la globalización es utilizando la clase **CultureInfo**. La siguiente serie de ejemplos son una colección de escenarios comunes que muestran cómo se puede utilizar esta clase de .NET Framework para aprovechar de una manera óptima las diferentes culturas.

NOTA

Para implementar estos ejemplos, es necesario hacer referencia a los espacios de nombres **Globalization** y **Threading**:

```
Imports System.Globalization
Imports System.Threading
```

Ejemplos

Calendarios

Los calendarios se representan en divisiones de tiempo como semanas, meses y años. El espacio de

```
Dim ci As New CultureInfo("th-TH")
Dim i As Integer = ci.OptionalCalendars.GetUpperBound(0)

Debug.WriteLine("The default calendar for the culture " & _
    ci.EnglishName & " is: " & ci.Calendar.ToString)
Debug.WriteLine("The optional calendars for " & _
    ci.NativeName & " are:")
For i = 0 To ci.OptionalCalendars.GetUpperBound(0)
    Debug.WriteLine(ci.OptionalCalendars(i).ToString)
Next
```

RESULTADO

```
The default calendar for the culture Thai (Thailand) is:
System.Globalization.ThaiBuddhistCalendar
The optional calendars for ไทย (ไทย) are:
System.Globalization.ThaiBuddhistCalendar
System.Globalization.GregorianCalendar
```

Fuente 1

nombres **System.Globalization** incluye las siguientes implementaciones de la clase base **Calendar**: **GregorianCalendar**, **HebrewCalendar**, **HijriCalendar**, **JapaneseCalendar**, **JulianCalendar**, **KoreanCalendar**, **TaiwanCalendar** y **ThaiBuddhistCalendar**.

El fuente 1 muestra cómo obtener la lista de calendarios que puede utilizar la referencia cultural **th-TH** (Tailandés-Tailandia).

Notas

1. La propiedad **CultureInfo.OptionalCalendars** obtiene la lista de calendarios que puede utilizar la referencia cultural seleccionada.
2. Diferencia entre **EnglishName** y **NativeName**: **EnglishName** muestra el nombre de la cultura en inglés, mientras que **NativeName** muestra el nombre localizado.

Andy González es International Program Manager en Microsoft. Es responsable de la calidad de las herramientas de desarrollo para Latinoamérica y España, entre las que se incluyen Visual Studio, Visual Studio Team System, .NET Framework y Visual Studio Tools for the Office System. Ha trabajado en Microsoft durante 6 años, siempre centrado en la globalización de estos productos.

```

Dim dt As DateTime = DateTime.Now
Dim ci As CultureInfo = New CultureInfo("ar-EG")

Thread.CurrentThread.CurrentCulture = New CultureInfo("en-US")
Debug.WriteLine("The Short Date format for: " & _
    Thread.CurrentThread.CurrentCulture.NativeName & " is: ")
Debug.WriteLine(dt.ToString("d"))
Debug.WriteLine("The first day of the week is in " & _
    Thread.CurrentThread.CurrentUICulture.NativeName & " is: " & _
    Thread.CurrentThread.CurrentUICulture.DateTimeFormat.FirstDayOfWeek)

Debug.WriteLine("The Short Date format for: " & ci.NativeName & " is: ")
Debug.WriteLine(dt.ToString("d", ci))
Debug.WriteLine("The first day of the week is in " & ci.NativeName & _
    " is: " & ci.DateTimeFormat.FirstDayOfWeek)

```

RESULTADO

```

The Short Date format for: English (United States) is:
5/30/2006
The first day of the week is in English (United States) is: 0
The Short Date format for: العربية (مصر) is:
30/05/2006
The first day of the week is in العربية (مصر) is: 6

```

Fuente 2

```

Dim theTime As DateTime = DateTime.Now
Dim gmtTime As DateTime = DateTime.Now.ToUniversalTime
Dim timeDiff As TimeSpan = theTime.Subtract(gmtTime)

Dim tMinutes As Integer = timeDiff.TotalMinutes
Dim Minutes As Integer = timeDiff.Minutes
Dim Hours As Integer = timeDiff.Hours
Dim myTimeZone As TimeZone = TimeZone.CurrentTimeZone

txtTime.Text = theTime.ToString
txtGMT.Text = gmtTime.ToString
txtTimeDiff.Text = "hours: " & Hours & " Minutes: " & Minutes
lblTminutes.Text = "(" & tMinutes & " Total minutes" & ")"
lblTimeZone.Text = myTimeZone.StandardName

```

RESULTADO



Fuente 4

Fechas

El código del fuente 2 permite obtener el formato de fecha para las culturas Inglés-Estados Unidos y Árabe-Egipto, y muestra no solo cómo el formato es diferente, sino también la diferencia en el primer día de la semana para ambas culturas.

Notas

1. La propiedad `Thread.CurrentCulture` establece la cultura para el subprocesso actual.
2. La propiedad `Thread.CurrentUICulture` establece la cultura que utiliza el gestor de recursos para utilizar los recursos culturales en tiempo de ejecución.

Hora

El ejemplo que se presenta en el fuente 3 es básico y trata sobre los formatos de hora. A continuación, el ejem-

```

Dim en As CultureInfo = New CultureInfo("en-US")
Dim es As CultureInfo = New CultureInfo("es-ES")
Dim dt As DateTime = DateTime.Now

Thread.CurrentThread.CurrentCulture = en
Debug.WriteLine("The Time format for: " & _
    Thread.CurrentThread.CurrentUICulture.NativeName & " is: ")

Debug.WriteLine(dt.ToString(en.DateTimeFormat.LongTimePattern))
Debug.WriteLine("The Time format for: " & ci.NativeName & " is: ")
Debug.WriteLine(dt.ToString(ci.DateTimeFormat.LongTimePattern))

```

RESULTADO

```

The Time format for: English (United States) is:
9:48:54 AM
The Time format for: español (España) is:
9:48:54

```

Fuente 3

plo del fuente 4 utiliza el método `ToUniversalTime` para convertir el valor del objeto `DateTime` actual a la hora universal coordinada (UCT) y mostrar la diferencia entre la hora local y la hora universal.

Notas

1. El método `DateTime.ToUniversalTime` convierte el valor del objeto `DateTime` actual a la hora universal coordinada (UCT).
2. La propiedad `TimeZone.CurrentTimeZone` establece la zona horaria del equipo.

Números y monedas

.NET Framework permite aprovechar todas las personalizaciones dependientes de la cultura posibles para los valores numéricos, como son:

- Separador de bloques.
- Separador decimal.
- Números negativos.
- Agrupación de dígitos.
- Colocación del símbolo de número negativo.
- Símbolo monetario

El fuente 5 presenta un ejemplo de código.

Ordenación

El ejemplo que se presenta en el fuente 6 muestra la diferencia entre la ordenación utilizando el alfabeto tradicional y el alfabeto moderno en idioma español.


```

Dim i As Integer = 1000000
Dim j As Integer = -1000
Dim en As CultureInfo = New CultureInfo("en-US")
Dim es As CultureInfo = New CultureInfo("es-ES")

Thread.CurrentThread.CurrentCulture = en

Debug.WriteLine("The initial amount is: " & i)
Debug.WriteLine("The amount in " & _
    Thread.CurrentThread.CurrentCulture.NativeName & "is: " & _
    i.ToString("c"))
Debug.WriteLine("The amount in " & es.NativeName & "is: " & _
    i.ToString("c", es))
Debug.WriteLine("The initial amount is: " & j)
Debug.WriteLine("The amount in " & _
    Thread.CurrentThread.CurrentCulture.NativeName & "is: " & _
    j.ToString("c"))
Debug.WriteLine("The amount in " & es.NativeName & "is: " & _
    j.ToString("c", es))

```

RESULTADO

```

The initial amount is: 1000000
The amount in English (United States)is: $1,000,000.00
The amount in español (España)is: 1.000.000,00

The initial amount is: -1000
The amount in English (United States)is: ($1,000.00)
The amount in español (España)is: -1.000,00

```

Fuente 5

Culturas personalizadas (Custom Cultures)

La clase **CultureInfo** proporciona información específica sobre la cultura como el idioma, país, región, calendario, etc., y utiliza clases como **DateTimeFormatInfo**, **NumberFormatInfo**, **CompareInfo** y **TextInfo** para manejar características tales como los formatos de fechas y números, y las comparaciones de cadenas de caracteres.

Las aplicaciones que se ejecuten en un equipo solo podrán utilizar las culturas definidas de antemano en la clase **CultureInfo**. Ahora bien, es posible utilizar la clase **CultureAndRegionInfoBuilder** para crear una nueva cultura o reemplazar una ya definida, e instalarla en un equipo.

Esta puede ser utilizada por cualquier aplicación que se ejecute en el equipo donde se haya instalado. Para instalar una cultura, es necesario tener derechos de administrador.

Microsoft Locale Builder

Microsoft Windows Vista soporta más de 200 localidades y más de 100 idiomas, pero esto aún no es suficiente para cubrir todas las necesidades del mundo del software. **Microsoft Locale Builder** es una herramienta que proporciona la posibilidad de cambiar la información local predefinida por información regional específica. Esta herramienta ha sido creada para satisfacer las necesidades de clientes en regiones para la cual Windows Vista no ha definido una configuración regional.

```

Dim stringArray(3) As String

stringArray(1) = "color"
stringArray(2) = "calor"
stringArray(3) = "calle"

Debug.WriteLine("The Array contains: ")
Debug.WriteLine(stringArray(1))
Debug.WriteLine(stringArray(2))
Debug.WriteLine(stringArray(3))

Thread.CurrentThread.CurrentCulture = New CultureInfo(3082)

Array.Sort(stringArray)

Debug.WriteLine("After sorting in the array in Spanish"&_
    " - Spain (Modern Sort): ")
Debug.WriteLine(stringArray(1))
Debug.WriteLine(stringArray(2))
Debug.WriteLine(stringArray(3))

Thread.CurrentThread.CurrentCulture = New CultureInfo(1034)
Array.Sort(stringArray)
Debug.WriteLine("After sorting again in " & _
    "Spanish-Spain (Traditional Sort): ")
Debug.WriteLine(stringArray(1))
Debug.WriteLine(stringArray(2))
Debug.WriteLine(stringArray(3))

```

RESULTADO

```

The Array contains:
color
calor
calle

After sorting in the array in Spanish - Spain (Modern Sort):
calle
calor
color

After sorting again in Spanish - Spain (Traditional Sort):
calor
calle
color

```

Fuente 6

Life for a Child with Diabetes

El importe correspondiente a los derechos de autor de este artículo ha sido cedido por el autor a la **International Diabetes Federation**. ¡Gracias Andy!

El objetivo del proyecto "Life for a Child" es dar a los niños con diabetes los cuidados clínicos y la educación necesaria sobre su enfermedad para seguir con vida. Los centros de diabetes beneficiados están en países como Fiji, Nueva Guinea, Filipinas, India, Rumania, Bolivia, Azerbaijón, Montenegro, Sri Lanka, República Democrática del Congo o Ruanda.

Si considera ésta una causa justa y solidaria y desea hacer una donación como ha hecho Andy, puede hacerlo en www.lifeforachild.idf.org.



Conclusión

En este artículo hemos presentado, a través de ejemplos prácticos, las amplias posibilidades que nos ofrece .NET Framework para la globalización de nuestras aplicaciones. ○

Referencias

[1] Para encontrar más información, visite [http://msdn2.microsoft.com/es-es/library/system.globalization\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/system.globalization(VS.80).aspx).

[2] Para obtener más información acerca de **Microsoft Locale Builder**, visite <http://www.microsoft.com/downloads/details.aspx?FamilyId=12ADFFC8-A4DA-424B-8D62-17C1E0FFC116&displaylang=en>.



Pepe Hevia

Mantener la sesión HTTP con .NET Framework

En este sencillo pero muy útil truco .NET, nos vamos a imponer el reto de subsanar una pega que muchos desarrolladores han encontrado al intentar crear aplicaciones que consumen información de servidores Web utilizando mecanismos de cierto bajo nivel...

» **Como por ejemplo,** al simular desde un formulario Windows el envío de datos mediante POST, para entonces recoger el resultado HTML y visualizarlo en una vista Web dentro de un formulario Windows.

El problema que se plantea es el de cómo no perder la sesión para que las páginas ASP, JSP, etc., sepan en todo momento quiénes somos, a pesar de no que no utilizamos ningún navegador estándar para estas tareas. El caso es que vamos a escribir un *User-Agent* propio aprovechando los conocimientos del protocolo HTTP y sus cabeceras, de tal forma que el servidor nunca pierda la pista de quiénes somos.

Un poquito de... conceptos HTTP

Desde la aparición de las aplicaciones Web, uno de los problemas que se plantearon fue el de cómo mantener una referencia del navegador que estaba conectado al servidor. ¿Por qué? Pues porque las aplicaciones Web ofrecen normalmente un sistema de navegación por un conjunto de opciones que requieren el almacenamiento de información acerca de quién está conectado. De esta forma es posible mantener el nombre del usuario en las páginas por las que pasa, hacer un seguimiento de los productos que selecciona, mantener información el tiempo suficiente como para poder hacer cálculos, procesos, accesos a datos...

Todo lo anterior no es posible tal y como lo plantea el protocolo HTTP 1.0. ¿Y eso? Pues porque HTTP 1.0 es como la Dorita de la película "El pequeño Nemo": se olvida de quiénes somos en cada conversación que tenemos. HTTP es un protocolo desconectado, es decir, que responde al patrón "conectamos-descargamos la página-nos desconectamos". Y cada petición es un evento completamen-

te nuevo (hay que recordar que HTTP fue pensado originalmente para la descarga de documentos). Esto llevó a los cabezones que desarrollaban el estándar a romperse la cabeza y desarrollar una mejora.

El protocolo HTTP 1.1 establecía una nueva cabecera de tipo *cookie* que almacenaba un identificador que el servidor Web generaba con cada nueva conexión. De esta forma, esta *cookie* de sesión era enviada en cada petición-respuesta, conociéndose así en todo momento al cliente de la conexión. Esto llevó al protocolo HTTP a ser persistente entre llamadas. Y los subsistemas CGI de los servidores pudieron comenzar a almacenar datos en su memoria acerca de la sesión de los clientes conectados. Comenzaba, pues, lo rico del juego que todos conocemos ahora....

Ahora bien, en la actualidad los servidores Web ya no son solo contenedores de documentos, sino completos servidores de aplicaciones. Y el uso cotidiano de la Web nos lleva a pedir más. De hecho, el problema es el propio HTML: es demasiado limitado para poder "ir más allá". Es por eso que se plantean dos iniciativas:

- a) Clientes relativamente inteligentes, gráficamente potentes, basados en servicios de lógica distribuida (*Smart Clients*).
- b) Clientes ligeros AJAX, mediante los cuales (empleando tecnologías abiertas: JavaScript y XML, junto con los servicios de lógica distribuidos), se hace posible escribir clientes Web más accesibles y ricos.

La cosa es que Internet 2.0 está a la vuelta de la esquina y la cosa promete revolucionar. Y digo yo... ¿qué tal un término medio? Me explico: ni a) ni b), sino un punto intermedio.

Pepe Hevia

colabora habitualmente con dotNetManía. Es arquitecto de software senior en Alhambra-Eidos. Tiene las certificaciones de MCAD, MCSD.NET y MCT en .NET Framework. Ha escrito varios libros sobre la plataforma .NET (lalibreriadigital.com)

El problema de a) es que pasa por un modelo de servicios Web que no guardan estado de sesión alguno por la propia filosofía del paradigma SOA; el problema de b) es que, aunque AJAX plantea un interfaz HTML rico, es relativamente complejo a la hora de desarrollar, ya que hay que conocer muy a fondo JavaScript, DOM y HTML dinámico. Y no siempre el resultado es efectivo en todos los navegadores.

TIP

Ahora que viene Internet 2.0, planteo yo: si .NET tiene un despliegue tan sencillo como ClickOnce; una ejecución segura garantizada gracias a los permisos de seguridad; un tamaño compacto, etc., ¿por qué no orientarse a desarrollar clientes ricos firmados digitalmente para hacer más rico el uso del cliente y la red, empleando protocolos abiertos? ¡Es justo el camino de en medio!

Así que, ¿por qué no usar los interfaces ricos de las plataformas clientes, y hacer que las páginas ASP.NET se encarguen del modelo de negocio? Quitamos HTML y dejamos una plataforma con los mismos elementos que ya conocemos: sesión, aplicación, etc. Es decir, reemplazamos el navegador de Internet, y creamos “navegadores” específicos de la aplicación Web. Pero claro, si el Internet Explorer se comunica con los servidores y se emplea la sesión para mantener la navegación, ¿cómo podemos hacer que una navegación de Windows Forms haga lo mismo que las páginas ASP.NET? Pues bien, haciendo que nuestro aplicativo procese las *cookies* de sesión HTTP 1.1 de forma correcta. Y el truco: conocer el protocolo HTTP y sus RFC. Esto nos permitirá:

- Generar menús Win32 e interfaces gráficas basadas en potentes *TreeViews* u otros controles más avanzados a partir de XML de servicios de un servidor Web. Nos quitamos de encima la pesadilla de los diseños de menús HTML.
- Navegar desde los menús empleando ventanas MDI, cuadros de diálogo, etc., que enviarán datos POST a servicios HTTP.
- Validar la información a enviar al servidor empleando todos los mecanismos de validación de los controles Windows.
- Mantener la sesión entre ventanas Windows, empleando las *cookies* de sesión de un servidor HTTP para mantener la coherencia en los servicios.
- Aprovechar la potencia de un entorno GDI/DirectX para poder dotar a los interfaces de servicios nuevos.
- La lógica queda encapsulada en sencillas páginas ASP.NET centralizadas. No deberemos preocuparnos por los cambios en los clientes.
- El despliegue de clientes, gracias a las tecnologías PUSH de .NET, es... ¡pan comido!
- Un largo etcétera.

Para que HTTP sea un protocolo *Keep-Alive*, la primera vez que nos conectamos a una página ASP.NET se nos estampa una cabecera *Set-Cookie* con el identificador que nos hace únicos en un contexto del servidor Web. Es a partir de ese momento que gozamos del objeto **Session** en las páginas ASP/ASP.NET, etc. Lo importante es que el navegador, en cada diálogo petición-respuesta, debe enviar continuamente al servidor dicha *cookie* para que el servidor nos tenga localizados. Y ¡voilà! Os aconsejo que veáis un poco más adelante cómo queda el protocolo, para que veáis cómo se implementa el control de sesión HTTP en nuestro ejemplo (figura 3).

TIP

Muchas veces lo que parece complicado es muy sencillo. Con un poco de HTTP, un poco de imaginación y – sobre todo – buenas herramientas, se puede hacer prácticamente todo.

La arquitectura

Para demostrar cómo funciona nuestra propuesta, podríamos complicarnos mucho la vida montando cualquier tipo de aplicación típica de la Web. Pero nos vamos a limitar a crear una sencilla aplicación que nos permitirá introducir quiénes somos, y nos dará la información de nuestra sesión. Simplificando el análisis (vea figura 1):

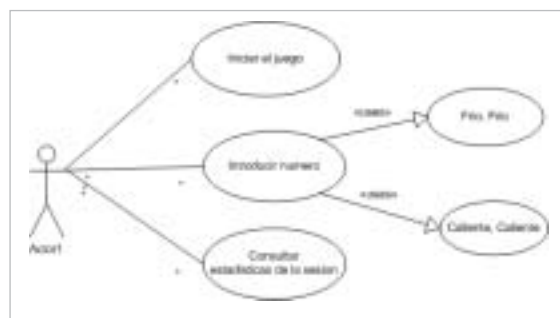


Figura 1. Los casos de uso de nuestro ejemplo.

TIP

Me gusta mucho analizar antes de poner manos a la obra. Fijaros cómo, a partir de un sencillo diagrama de lo que quiero hacer (figura 1), obtengo un diagrama de arquitectura (figura 2).

Para poder implementar la lógica de negocio necesitaremos dos capas y un **CommandPattern** para el protocolo de comunicaciones basado en HTTP (a todo el conjunto, modelo orientado a servicios).

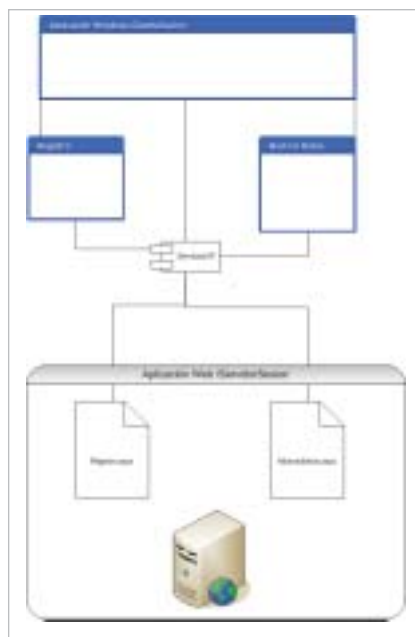


Figura 2. Arquitectura en capas. Fijaos en la sincronía de cada elemento. Una preciosidad, el modelo orientado a servicios.

Finalmente, el protocolo que encapsula nuestro manipulador se describe en la figura 3.

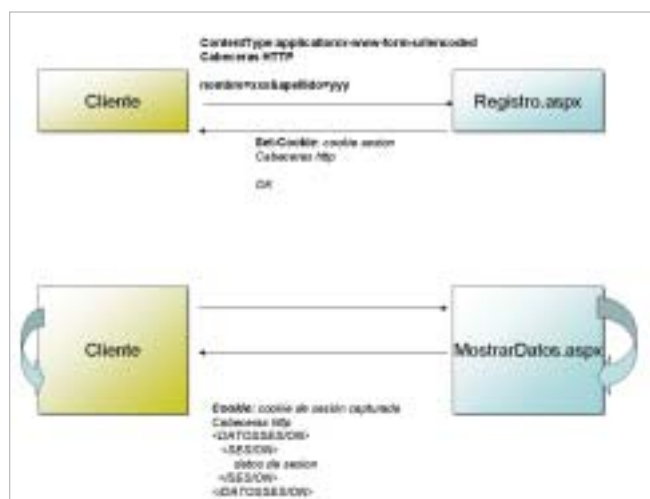


Figura 3. Diagrama del protocolo empleado.

Lo que necesitamos de la capa de cliente

Con el patrón de diseño **CommandPattern** nos será posible encapsular lo que necesitamos de HTTP tanto en petición como en respuesta desde los servicios de los controles de los formularios. Así encapsulamos su funcio-

```
Public Class ServiciosCP
    Private mSession As Cookie = Nothing

    Public Shared Function ParserCookie(ByVal cabecera As String) As String
        Dim tokens As String() = cabecera.Split(";")
        Return tokens(0)
    End Function

    Public Function Registrar(ByVal nombre As String, _
        ByVal apellido As String) As String
        Dim cliente As HTTPWebRequest = _
            WebRequest.Create("HTTP://localhost/ServidorSesion/Registro.aspx")
        cliente.Method = "POST"
        cliente.ContentType = "application/x-www-form-urlencoded"
        Dim post As New StreamWriter(cliente.GetRequestStream())
        post.Write(String.Format("nombre={0}&apellido={1}", nombre, apellido))
        post.Close()

        'Recoger la respuesta, entre ella, la cookie de sesión
        Dim resp As HTTPWebResponse = cliente.GetResponse()
        mSession = ParserCookie(resp.Headers("Set-Cookie"))
        Dim msg As New StreamReader(resp.GetResponseStream())
        Dim result As String = msg.ReadToEnd()
        msg.Close()
        Return result
    End Function

    (...)
End Class
```

Fuente 1. Comando Registrar.

nalidad en un comando, y obtenemos una respuesta independientemente de cómo funcione por dentro (de manera similar a como trabajan objetos tan conocidos como los de ADO.NET, por poner un ejemplo). La clase responsable de este patrón es la clase **ServiciosCP**, que se encarga de:

- Registrar los datos del usuario empleando la página ASPX **Registro.aspx**. Codifica los datos en un cuerpo POST, realiza la petición HTTP y de su respuesta extrae dos cosas: la *cookie* de sesión y el mensaje de res-

puesta del servidor. Como muestra el fuente 1, es necesario extraer la cookie de sesión de la cabecera HTTP *Set-Cookie*, y extraer de su formato (definido en las RFC W3C) la cadena que guarda el par **clave=valor**. Es importante indicar el **ContentType** para que el servidor ASP.NET reconozca correctamente el formato del POST.

- Recoger los datos de información de la sesión del usuario, encapsulando el funcionamiento del servicio que ofrece la página ASPX **MostrarDatos.aspx**. Como se puede ver en el fuente 2, el envío de la sesión se realiza como una cabecera HTTP *Cookie* con el valor obtenido del proceso de registro (que generó la cabecera *Set-Cookie*).

Es importante recalcar que no se están empleando servicios Web ASMX, sino páginas ASPX puras y duras (es posible hacerlo con servicios Web, pero en tal caso es necesario activar el soporte de *Stateful* y control de sesión con las directivas adecuadas). Es por ello que todo el peso de la serialización de los comandos recae directamente sobre HTTP, y que debemos emplear las maravillosas clases HTTP de .NET: **WebRequest**, **HTTPWebRequest** y **HTTPWebResponse**, del espacio de nombres **System.NET**; clases que encapsulan la manipulación del protocolo.

Nuestro manipulador lo definirá el programa principal y se lo pasará a los formularios que lo requieran. Ahí está el mantenimiento de la sesión: no perder bajo ningún concepto la *cookie* de sesión.

```

Public Class ServiciosCP
    Private mSession As Cookie = Nothing
    (...)
    Public Function MostrarDatos() As DataSet
        If mSession Is Nothing Then
            Throw New Exception("La sesión no está establecida")
        End If

        Dim cliente As HTTPWebRequest = _
            WebRequest.Create("HTTP://localhost/ServidorSession/MostrarDatos.aspx")
        cliente.Method = "GET"
        cliente.CookieContainer.Add(mSession)

        'Recoger la respuesta, entre ella, la cookie de sesión
        Dim resp As HTTPWebResponse = cliente.GetResponse()
        mSession = resp.Headers.Add("Cookie", mSession)
        Dim msg As New StreamReader(resp.GetResponseStream())
        Dim result As String = msg.ReadToEnd()
        msg.Close()

        'Obtener el DataSet del XML de retorno
        Dim dsDatos As New DataSet("INFO")
        dsDatos.ReadXml(new StringReader(result), XmlReadMode.InferSchema)
        Return dsDatos
    End Function
End Class

```

Fuente 2. Comando **MostrarDatos**.

Lo que necesitamos de la capa de servidor...

Sólo nos queda poner el condimento en el plato, como hacen nuestros maestros cocineros de la tele, para dejar casi todo terminado. Necesitamos solo coordinar el cliente con los servicios de negocio de las páginas ASPX. En ninguno de los casos la página devolverá HTML, así que limpiaremos el documento HTML dejando sólo la directiva `@Page`, y escribiremos el código en el evento `Page_Load` del *code-behind*.

La página de **Registro** recogerá el POST y creará la sesión (simplemente usando el objeto integrado `Session`, ver fuente 3). La página de **MostrarDatos** compondrá un documento XML con la información que podemos sacar de la sesión ASP (fuente 4). Y eso es todo. Lo bueno de ASP es que la propia plataforma gestiona la persistencia de la sesión de forma nativa. Nosotros nos limitamos solo a usar sus servicios.

En el código fuente, además de la sesión es posible enviar cualquier dato relevante almacenado en los objetos integrados ASP.NET. Ya puestos... Además, tal cual está diseñado el sistema, como el documento devuelto es XML, podremos ampliar la lista tanto como queramos. En nuestro caso, hemos añadido la ruta del *host* cliente

y un nombre establecido en la cabecera HTTP: el agente de usuario (que muestra que no todo es Internet Explorer).

```

Public Class Registro
    Inherits System.Web.UI.Page

    (...)

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles MyBase.Load
            'Preparar la sesión.
            Session.Add("nombre", Request.Form("nombre"))
            Session.Add("apellido", Request.Form("apellido"))

            'Generar la respuesta. Ya viajará la cookie de sesión.
            Response.Write("OK")
        End Sub
    End Class

```

Fuente 3. Code-behind de **Registro.aspx**

```

Public Class MostrarDatos
    Inherits System.Web.UI.Page

    (...)

    Private Sub Page_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles MyBase.Load
        Response.ContentType = "text/xml"
        Response.Write("<DATOSSESION><SESION>")
        Response.Write(String.Format("<nombre>{0}</nombre>", Session("nombre")))
        Response.Write(String.Format("<apellido>{0}</apellido>", Session("apellido")))
        Response.Write(String.Format("<idSesion>{0}</idSesion>", Session.SessionID))
        Response.Write(String.Format("<esNueva>{0}</esNueva>", Session.IsNewSession))
        Response.Write(String.Format("<timeOut>{0}</timeOut>", Session.Timeout))
        Response.Write(String.Format("<ip>{0}</ip>", Request.UserHostAddress))
        Response.Write("</SESION></DATOSSESION>")
    End Sub
End Class


```

Fuente 4. Code-Behind de **MostrarDatos.aspx**

Todo funcionando...

Bien pues ya solo nos queda el perejil. Ver todo funcionando. Simplemente se ejecuta el **Registro**, y accedemos a la ventana de "Mostrar Datos" bien desde el propio formulario de alta, bien desde el menú. El caso es que se trata de instancias distintas. Y lo que se puede ver es que todas comparten la misma sesión ASP.NET en todo momento. Ya tenemos las bases para montar aplicaciones inteligentes que suplen la función de un navegador HTML. Ahora podremos montar navegadores de servicios en la futura Web 2.0. Javascript mola, pero .NET mola más.

Vosotros podéis ir ya imaginando arquitecturas. Os propongo algo que resolveremos en futuros artículos: menús dinámicos, formularios dinámicos de entrada y respuesta. ¡Todo dirigido por la Web!

El código fuente de este proyecto se puede obtener desde la página Web de la revista (<http://www.dotnetmania.com>), o bien desde mi sitio Web (www.beviatec.net). 



Carlos Quintero

Creación de asistentes en Visual Studio

En este artículo veremos una forma más de extensibilidad de Visual Studio, aparte de las macros y complementos que ya hemos visto con anterioridad. Se trata de los asistentes (*wizards*) que se introdujeron con la versión inicial de Visual Studio .NET. En artículos posteriores hablaremos de las novedades y cambios introducidos en Visual Studio 2005 (plantillas, kits de iniciación, etc.).

» En entregas anteriores hemos hablado de dos formas de extensibilidad de Visual Studio que eran las macros y los complementos (*add-ins*). Existe una tercera forma de extensibilidad de alto nivel, aunque de alcance más reducido que las anteriores, que son los asistentes (*wizards*). Todos sabemos lo que es un asistente, pues se popularizaron mucho desde Windows 95: es un cuadro de diálogo que a través de una serie de pasos (pulsando el típico botón “Siguiente”) nos permite realizar una tarea más o menos compleja. En el caso más sencillo, un asistente puede constar de un único paso y no requerir información del usuario. Los asistentes de Visual Studio permiten generar de forma automática código repetitivo, que empleamos una y otra vez en nuestros proyectos. Preguntando al usuario algunos datos, el asistente puede personalizar el código generado con dichos valores.

Tipos de asistentes

Visual Studio .NET permite crear tres tipos de asistentes:

- *Asistente para nuevo proyecto.* Es el que se ejecuta cuando seleccionamos un tipo de proyecto en el diálogo que aparece al pulsar el menú “Archivo - Nuevo - Proyecto...” de Visual Studio. En ese cuadro de diálogo aparecen los distintos tipos de proyectos que podemos crear, agrupados por lenguaje (VB.NET, C#) o por tipo (proyectos de instalación, de base de datos, etc.).

En la mayoría de los proyectos, el asistente que se ejecuta al seleccionar un tipo de proyecto no muestra ninguna ventana para pedir información adicional, porque sólo con el nombre del proyecto, tipo y carpeta ya puede generar el código fuente. Imaginemos que en nuestra empresa tenemos un *framework* de desarrollo y todos los proyectos que se crean siguen una serie de pautas: referencias que se han de incluir, archivos obligatorios (tal vez almacenados en una carpeta central), etc. En este caso nos puede interesar un asistente para un nuevo tipo de proyecto que ahorre a nuestros programadores el tener que realizar esas tareas manualmente cada vez que quieran crear una nueva aplicación. Aunque esto se puede realizar desde una macro o un complemento, el *Asistente para nuevo proyecto* de Visual Studio se creó específicamente para esto.

- *Asistente para nuevo archivo.* Es similar al asistente para nuevo proyecto. Cuando ya tenemos un proyecto creado, si pulsamos el menú “Proyecto - Agregar nuevo elemento...” se muestra un cuadro de diálogo que ofrece distintos tipos de archivo que podemos añadir



Carlos J. Quintero es Microsoft MVP de Visual Basic desde principios del año 2004. Es autor del popular complemento MZ-Tools 3.0 para VB6, VB5 y VBA, de MZ-Tools 4.0 para Visual Studio .NET 2002/ 2003, y de MZ-Tools 2005 para Visual Studio 2005. Estas herramientas están disponibles en <http://www.mztools.com>

a nuestro proyecto. Al seleccionar uno de ellos, se invoca el asistente para ese tipo de archivo que, como en el caso anterior, la mayoría de las veces no preguntará ninguna información adicional. Supongamos que en nuestras aplicaciones tenemos que añadir muchos archivos con código que se repite entre ellos (por ejemplo, que todos los archivos incluyan un encabezado de documentación). De nuevo, aunque esto puede realizarse con una macro o complemento, puede interesarnos que aparezca un archivo de este tipo en el cuadro de diálogo “Nuevo elemento” de Visual Studio.

- *Asistente a medida (custom wizard)*. Es el típico asistente que sí pregunta información adicional y muestra los botones “Anterior” y “Siguiente”. En el caso de Visual Studio .NET, los asistentes de este tipo permiten realizar cualquier tarea, y no solamente crear nuevos proyectos o archivos (a costa de requerir la codificación necesaria para llevar a cabo dicha tarea, claro).

En este artículo nos centraremos sobre todo en los dos primeros tipos ya que son los más específicos y sencillos. Estos son muy similares entre sí, así que los explicaremos conjuntamente, aunque en los ejemplos usaremos un *Asistente para nuevo archivo*.

Visual Studio .NET permite crear tres tipos de asistentes: para nuevo proyecto, nuevo archivo, o a medida

asistente para crear el proyecto o archivo en la solución de Visual Studio. El asistente creará un nuevo proyecto en la solución o añadirá un archivo al proyecto existente a partir de unas plantillas.

Físicamente, un asistente es simplemente una DLL registrada como componente COM que implementa la interfaz `EnvDTE.IDTWizard` (del ensamblado `EnvDTE.dll` que tendremos que añadir al proyecto del asistente). Esta interfaz sólo contiene un método llamado `Execute` que es llamado por Visual Studio para ejecutar el asistente. El asistente es libre de hacer lo que quiera en este método (especialmente si es un asistente a medida), pero en el caso de asistentes para nuevo proyecto o nuevo archivo básicamente el asistente creará un nuevo proyecto o archivo.

Más adelante hablaremos de la interfaz `IDTWizard` y su método `Execute`, pero primero vamos a enumerar los pasos básicos para crear un Asistente para nuevo proyecto o nuevo archivo que desarrollaremos en los siguientes apartados:

- Creación de las plantillas de proyecto o archivo que usará el asistente. Al

Studio para crear nuevos proyectos o archivos a partir de plantillas.

- Reemplazar los símbolos (*placeholders*) de las plantillas por sus valores reales, que ha especificado el usuario al invocar el asistente (nombre del proyecto o de la clase, etc.).
- Registro COM (ActiveX) de dicha DLL. Ya hemos mencionado en otras ocasiones que Visual Studio .NET no es una aplicación .NET, sino que sigue usando COM para su extensibilidad.
- Registro del asistente en Visual Studio mediante un archivo `.vsz`. Mediante este archivo, el asistente se da a conocer a Visual Studio y hace que aparezcan las plantillas en los cuadros de diálogo “Nuevo proyecto” o “Nuevo elemento” del IDE.

Creación de plantillas

Como hemos mencionado, el propósito de un *Asistente para nuevo proyecto o nuevo archivo* es generar el código del proyecto o del archivo. Aunque el asistente podría generar este código directamente añadiendo un proyecto o archivo en blanco y usando el modelo de extensibilidad para escribir código (línea a línea o párrafo a párrafo), un enfoque más adecuado consiste en que use plantillas de código externas al propio asistente. De esta manera se puede actualizar el contenido de dichas plantillas sin necesidad de cambiar y recompilar la DLL del asistente. Con este enfoque, lo que el asistente tiene que hacer es obtener la carpeta donde están instaladas dichas plantillas y mediante la función `AddFromTemplate` del modelo de extensibilidad copiarlas a la carpeta de destino especificada por el usuario y añadirlas a la solución o al proyecto, según el tipo de asistente (proyecto o archivo, respectivamente).

Por tanto, nuestro primer paso para crear un asistente de este tipo será crear las plantillas que usará el asistente. ¿Cómo se hace esto? Pues la

Físicamente un asistente es simplemente una DLL, registrada como componente COM, que implementa la interfaz `EnvDTE.IDTWizard`

Creación de asistentes

Nuestro objetivo es, por tanto, hacer que en los cuadros de diálogo “Nuevo proyecto” o “Nuevo elemento” aparezcan plantillas de proyectos o archivos proporcionados por nosotros (junto a las que proporciona Visual Studio) y que cuando el usuario las seleccione, se invoque nuestro

distribuir nuestro asistente, tendremos que distribuir no solamente el asistente sino dichas plantillas.

- Creación de la DLL del asistente con una clase que implemente la interfaz `IDTWizard`.
- En la implementación del método `Execute` de la interfaz `IDTWizard`, usar los métodos `AddFromTemplate` del modelo de extensibilidad de Visual

forma más fácil es crear un nuevo proyecto o archivo como hacemos normalmente y personalizarlo como deseemos para que nos sirva de plantilla. Una vez creadas las plantillas, se pueden almacenar en cualquier carpeta, siempre que el asistente pueda luego encontrarlas. Por ello, lo más fácil es guardarlas en una subcarpeta de la carpeta donde está instalado el asistente. En el fuente 1 se muestra una plantilla muy sencilla para crear un archivo de clase VB.NET personalizado con un encabezado. Como se puede observar, hemos usado unas variables (con los caracteres %) que más tarde reemplazaremos por sus valores (como veremos en otro apartado).

```

'-----
' Autor   : %AUTOR%
' Fecha   : %FECHA%
'-----
Public Class %NOMBRE_CLASE%

End Class

```

Fuente 1

Creación de la DLL

Para crear la DLL del asistente tenemos que crear un proyecto en Visual Studio del tipo *Biblioteca de clases* (*Class Library*), al que añadiremos un archivo de clase y una referencia al ensamblado `EnvDTE.dll`. Como hay que registrar el asistente como componente COM (ActiveX), esta clase tiene que tener los siguientes atributos del espacio de nombres `System.Runtime.InteropServices`: `ComVisible(True)`, `ProgId(<progid>)` y `Clsid(<clsid>)`, donde `<progid>` puede ser por ejemplo “AsistentesVSNET.Asistentes” y `<clsid>` es un GUID único. Asimismo, es conveniente cambiar en el archivo `AssemblyInfo.vb` el valor del atributo `AssemblyVersion` a “1.0.0.0”, para indicar que el ensamblado siempre es compatible hacia atrás (podemos versionar el archivo de la DLL con el atributo `AssemblyFileVersion` en su lugar). Finalmente, para poder depurar el asistente, en las páginas de propiedades del proyecto seleccionaremos “Arrancar programa externo” como acción de inicio e indicaremos la ruta al ejecutable de Visual Studio .NET (`C:\Archivos de programa\Microsoft Visual Studio .NET 2003\Common7\IDE\devenv.exe`) y marcaremos la casilla “Registrar para interoperabilidad COM” en la sección “Compilación” para que se registre la DLL como componente ActiveX.

La interfaz IDTWizard

La clase principal del asistente tiene que implementar la interfaz `EnvDTE.IDTWizard`, que tiene un único método llamado `Execute`:

```

Public Sub Execute(ByVal Application As Object, _
    ByVal hwndOwner As Integer, _
    ByRef ContextParams As Object(), _
    ByRef CustomParams As Object(), _
    ByRef retval As EnvDTE.WizardResult)

```

Sus parámetros son los siguientes:

- **Application**: es el objeto `EnvDTE.DTE` que representa la instancia del IDE. Ya hemos hablado de este objeto en los artículos sobre complementos o macros.
- **hwndOwner**: el manejador (*handle*) de la ventana que muestra el cuadro de diálogo del asistente, para el caso de que el asistente tenga que mostrar algún cuadro de diálogo que tendríamos que emparentar con esa ventana.
- **ContextParams**: es un array de elementos de tipo `Object` con información de contexto que proporciona Visual Studio.
- **CustomParams**: es un array de objetos de tipo `Object` con información definida por el creador del asistente.
- **retval**: el resultado de la llamada a la función.

Valor del array		
Posición	ContextParams	Descripción
0	Tipo de asistente	El valor <code>EnvDTE.Constants.vsWizardNewProject</code> .
1	Nombre del proyecto	Nombre del proyecto que se ha de crear (sin extensión).
2	Carpeta local	Carpeta en la que se ha de crear el proyecto.
3	Carpeta de instalación	Carpeta donde está instalado Visual Studio.
4	Exclusivo	<ul style="list-style-type: none"> • Si es True, el asistente debería cerrar la solución actual y crear una nueva. • Si es False, el asistente ha de añadir el nuevo proyecto a la solución actual.
5	Nombre de la solución	El nombre de la solución que se ha de crear si el usuario especificó “Crear carpeta para la solución” en el cuadro de diálogo “Nuevo proyecto”.
6	Silencioso	Si es True, el asistente no debe preguntar nada al usuario, usando valores por defecto.

Tabla 1

Valor del array		
Posición	ContextParams	Descripción
0	Tipo de asistente	El valor <code>EnvDTE.Constants.vsWizardAddItem</code> .
1	Nombre del proyecto	El nombre del proyecto al que se está añadiendo el nuevo archivo.
2	Archivos de proyecto	La colección <code>EnvDTE.ProjectItems</code> a la que se debería añadir el nuevo archivo.
3	Carpeta nuevo archivo	La carpeta de destino del nuevo archivo.
4	Nombre nuevo archivo	El nombre que introdujo el usuario en el cuadro de diálogo “Nuevo archivo”.
5	Carpeta instalación del producto	La carpeta en la que está instalado el lenguaje de programación del proyecto.
6	Silencioso	Si es True, el asistente no debe preguntar nada al usuario, usando valores por defecto.

Tabla 2

```
Imports System.Runtime.InteropServices
Imports System.Windows.Forms

<ComVisible(True), _
    ProgId("AsistentesVSNET.Asistentes"), Guid("{17F0492C-7060-4fc1-B2DE-B7C37F367D10}")>_
Public Class Asistente
    Implements EnvDTE.IDTWizard
    Public Sub Execute(ByVal Application As Object, _
        ByVal hwndOwner As Integer, _
        ByRef ContextParams() As Object, _
        ByRef CustomParams() As Object, _
        ByRef retval As EnvDTE.wizardResult) Implements EnvDTE.IDTWizard.Execute

        Dim sTipoAsistente As String
        Dim objDTE As EnvDTE.DTE
        Dim colProjectItems As EnvDTE.ProjectItems
        Dim sNombreProyecto As String
        Dim sCarpetaDestino As String
        Dim sNombreSolucion As String
        Dim sNombreArchivo As String
        Dim bExclusivo As Boolean
        Dim bSilencioso As Boolean
        Dim bContinuar As Boolean

        ' Valor por defecto
        retval = EnvDTE.wizardResult.wizardResultFailure
        Try
            sTipoAsistente = CType(ContextParams(0), String)
            Select Case sTipoAsistente.ToLower
                Case EnvDTE.Constants.vsWizardNewProject.ToLower
                    sNombreProyecto = CType(ContextParams(1), String)
                    sCarpetaDestino = CType(ContextParams(2), String)
                    bExclusivo = CType(ContextParams(4), Boolean)
                    sNombreSolucion = CType(ContextParams(5), String)
                    bSilencioso = CType(ContextParams(6), Boolean)

                    ' Si es un nuevo proyecto exclusivo, hay que dar al usuario la
                    ' oportunidad de guardar la solución actual
                    bContinuar = True
                    If bExclusivo Then
                        If objDTE.ItemOperations.PromptToSave = _
                            EnvDTE.vsPromptResult.vsPromptResultCancelled Then
                            bContinuar = False
                        End If
                    End If
                    If bContinuar Then
                        retval = AñadirProyecto(objDTE, sNombreProyecto, _
                            sCarpetaDestino, bExclusivo, sNombreSolucion, bSilencioso)
                    End If

                Case EnvDTE.Constants.vsWizardAddItem.ToLower
                    sNombreProyecto = CType(ContextParams(1), String)
                    colProjectItems = CType(ContextParams(2), EnvDTE.ProjectItems)
                    sCarpetaDestino = CType(ContextParams(3), String)
                    sNombreArchivo = CType(ContextParams(4), String)
                    bSilencioso = CType(ContextParams(6), Boolean)
                    retval = AñadirArchivo(objDTE, sNombreProyecto, colProjectItems, _
                        sCarpetaDestino, sNombreArchivo, bSilencioso)
            End Select
        Catch objException As Exception
            MessageBox.Show(objException.ToString)
        End Try
    End Sub

    Private Function AñadirProyecto(ByVal objDTE As EnvDTE.DTE, _
        ByVal sNombreProyecto As String, ByVal sCarpetaDestino As String, _
        ByVal bExclusivo As Boolean, ByVal sNombreSolucion As String, _
        ByVal bSilencioso As Boolean) As EnvDTE.wizardResult
        Dim eResultado As EnvDTE.wizardResult = EnvDTE.wizardResult.wizardResultSuccess
        Dim sArchivoPlantilla As String
```

Cuando Visual Studio llama a este método del asistente para ejecutarlo, en ese punto el asistente puede hacer lo que quiera sin restricciones. Para ello dispone de la información que se le pasa en los distintos parámetros de la función. El parámetro más importante es **ContextParams**, ya que proporciona información que el asistente puede emplear para generar código. Como hemos dicho, se trata de un array de objetos de tipo **Object**, pero su longitud y tipo real dependen del tipo de asistente. La tabla 1 muestra los valores para un *Asistente de nuevo proyecto*, y la tabla 2 para un *Asistente nuevo archivo*.

Lo primero que debería hacer un asistente es comprobar el valor del primer elemento del array **ContextParams**, para determinar si es del tipo de asistente esperado. El fuente 2 muestra el código del asistente.

Uso de las plantillas

Una vez recuperados los valores de los parámetros que recibe el método **Execute** de la interfaz **IDTWizard**, el asistente tiene que recuperar la plantilla de proyecto o de archivo que va a usar de la carpeta donde residen que, como indicamos antes, normalmente será una subcarpeta de la carpeta donde está la DLL del asistente. En .NET un ensamblado puede saber la ubicación de su ejecutable usando el método **System.Reflection.Assembly.GetExecutingAssembly.Location**. Por tanto, el asistente podría extraer la parte del directorio (excluyendo la parte final del archivo) y añadir la subcarpeta de las plantillas para obtener la ruta completa a las mismas. La función **ObtenerCarpetaPlantillas** muestra cómo realizar este paso.

La forma de añadir un proyecto a una solución es mediante el método **AddFromTemplate** de la clase **EnvDTE.Solution**:

```
Public Function AddFromTemplate(_
    ByVal FileName As String, _
    ByVal Destination As String, _
    ByVal ProjectName As String, _
    Optional ByVal Exclusive As _
        Boolean = False) As EnvDTE.Project
```

Fuente 2 (continúa...)


```

Try
    sArchivoPlantilla = ObtenerCarpetaPlantillas() & "PlantillaProyecto"
    objDTE.Solution.AddFromTemplate( sArchivoPlantilla, sCarpetaDestino, _
                                    sNombreProyecto, bExclusivo)

Catch objException As Exception
    MessageBox.Show(objException.ToString)
    eResultado = EnvDTE.wizardResult.wizardResultFailure
End Try
Return eResultado
End Function

Private Function AñadirArchivo(ByVal objDTE As EnvDTE.DTE, _
    ByVal sNombreProyecto As String, ByVal colProjectItems As EnvDTE.ProjectItems, _
    ByVal sCarpetaDestino As String, ByVal sNombreArchivoSinExtension As String, _
    ByVal bSilencioso As Boolean) As EnvDTE.wizardResult

    Dim eResultado As EnvDTE.wizardResult = EnvDTE.wizardResult.wizardResultSuccess
    Dim objProjectItem As EnvDTE.ProjectItem
    Dim sArchivoPlantilla As String
    Try
        sArchivoPlantilla = ObtenerCarpetaPlantillas() & "Plantilla.vb"
        objProjectItem = colProjectItems.AddFromTemplate(sArchivoPlantilla, _
                                                         sNombreArchivoSinExtension & ".vb")
        ReemplazarSimbolos(objProjectItem, sNombreArchivoSinExtension)

    Catch objException As Exception
        MessageBox.Show(objException.ToString)
        eResultado = EnvDTE.wizardResult.wizardResultFailure
    End Try
    Return eResultado
End Function

Private Sub ReemplazarSimbolos(ByVal objProjectItem As EnvDTE.ProjectItem, _
    ByVal sNombreArchivoSinExtension As String)

    Dim eFindOptions As EnvDTE.vsFindOptions
    Dim objWindow As EnvDTE.Window
    Dim objTextDocument As EnvDTE.TextDocument

    ' Abre la ventana
    objWindow = objProjectItem.Open(EnvDTE.Constants.vsViewKindTextView)

    ' Obtiene el documento de texto
    objTextDocument = DirectCast(objWindow.Document.Object, EnvDTE.TextDocument)

    ' Establece las opciones de búsqueda
    eFindOptions = EnvDTE.vsFindOptions.vsFindOptionsFromStart Or _
        EnvDTE.vsFindOptions.vsFindOptionsMatchCase Or _
        EnvDTE.vsFindOptions.vsFindOptionsMatchWholeWord

    ' Realiza los reemplazos
    objTextDocument.ReplacePattern("%AUTOR%", System.Environment.UserName, _
                                    eFindOptions)
    objTextDocument.ReplacePattern("%FECHA%", Now.ToShortDateString, eFindOptions)
    objTextDocument.ReplacePattern("%NOMBRE_CLASE%", _
                                    sNombreArchivoSinExtension, eFindOptions)

End Sub

Private Function ObtenerCarpetaPlantillas() As String
    Dim sRutaEnsamblado As String
    Dim sCarpetaEnsamblado As String
    Dim sPlantillaCarpeta As String

    sRutaEnsamblado = System.Reflection.Assembly.GetExecutingAssembly.Location()
    sCarpetaEnsamblado = System.IO.Path.GetDirectoryName(sRutaEnsamblado)
    sPlantillaCarpeta = sCarpetaEnsamblado & "\Plantillas\"
    Return sPlantillaCarpeta
End Function
End Class

```

(...continuación) Fuente 2

donde los parámetros usados son los siguientes:

- **FileName**: la ruta completa a la plantilla.
- **Destination**: la carpeta de destino del proyecto (el asistente debe crear esta carpeta antes de llamar a esta función).
- **ProjectName**: el nombre del archivo de proyecto sin extensión.
- **Exclusive**: si es True, la solución actual es cerrada y se crea una nueva. En caso contrario, se añade el proyecto nuevo a la solución actual.

Y la función devuelve el proyecto creado (del tipo `EnvDTE.Project`).

La forma de añadir un proyecto a una solución es mediante el método `AddFromTemplate` de la clase `EnvDTE.ProjectItems`:

```

Public Function AddFromTemplate(_
    ByVal FileName As String, _
    ByVal Name As String) As _
    EnvDTE.ProjectItem

```

Donde los parámetros usados son los siguientes:

- **FileName**: la ruta completa a la plantilla.
- **Name**: el nombre del archivo (incluyendo su extensión).

Y la función devuelve el archivo creado (del tipo `EnvDTE.ProjectItem`).

Reemplazos de variables

Cuando creamos un proyecto o archivo con las funciones `AddFromTemplate` que hemos visto, el archivo que se genera tiene unos valores por defecto que no serán adecuados. Por ejemplo, el nombre de una clase será siempre `Class1` o el nombre que hayamos escrito en la plantilla (ver fuente 1). Por ello, es muy común usar en la plantilla símbolos que tienen que ser sustituidos en este punto por sus valores. Los símbolos pueden tener cualquier formato, pero en nuestro caso hemos usado el carácter % para delimitar el nombre del símbolo, por ejemplo `%NOMBRE_CLASE%`. La función `ReemplazarSimbolos` muestra cómo reemplazar los símbolos de la plantilla

Un enfoque más adecuado consiste en que use plantillas de código externas al propio asistente

por sus valores adecuados usando la función `TextDocument.ReplacePattern` del modelo de extensibilidad. En este punto ya podemos compilar el asistente para ver si tiene errores.

Registro del asistente

Una vez compilado, nuestro asistente ha de ser registrado de dos maneras distintas:

- *Como componente COM (ActiveX)*: durante la etapa de depuración esto lo hace Visual Studio si hemos marcado la casilla “Registrar para interoperabilidad COM” en las propiedades del proyecto, pero para la instalación final deberá ser nuestro programa de instalación el que lo haga en la máquina del usuario final.
- *Como asistente de Visual Studio*: de igual manera que los complementos (*add-ins*) se han de dar a conocer a Visual Studio, los asistentes también tienen que informar a Visual Studio de su existencia. Esto se realiza mediante un archivo de texto con la extensión `.vsz` cuya primera línea es la cadena “VSWIZARD 7.0” (incluso si estamos usando Visual Studio .NET 2003, cuya versión es 7.1). La segunda línea es de la forma:

```
Wizard=<ProgId>
```

o

```
Wizard=<CLSID>
```

y especifica el `ProgId` (como “AsistentesVSNET.Asistentes”) o el `CLSID` (como “17F0492C-7060-4fc1-B2DE-B7C37F367D10”) que identifica al componente COM (ActiveX) de nuestro asistente (estos atributos se establecieron en la clase principal del asistente). A continuación puede haber más líneas opcionales de la forma:

```
Param=<valor1>
```

```
Param=<valor2>
```

donde estos valores serán pasados al método `Execute` de la interfaz `IDTWizard` en el parámetro `CustomParams` (que es un array de objetos de tipo `Object`).

Por último, sólo queda indicar dónde se coloca este archivo `.vsz`:

- Para un *Asistente de nuevo proyecto*, se ha de colocar en la carpeta `C:\Archivos de programa\Microsoft Visual Studio .NET 2003\VB7\VBProjects` para proyectos VB.NET o en la carpeta `C:\Archivos de programa\Microsoft Visual Studio .NET 2003\VC#\CSharpProjects` para proyectos C#. Si miramos en esas carpetas veremos unos cuantos archivos `.vsz` proporcionados por Visual Studio. Al colocar un archivo `.vsz` en estas carpetas, cuando abrimos el cuadro de diálogo “Nuevo proyecto” de Visual Studio, éste añade nuestro asistente como plantilla en la parte derecha del cuadro de diálogo para el lenguaje de programación seleccionado en el árbol de la izquierda.
- Para un *Asistente de nuevo archivo*, las carpetas son `C:\Archivos de programa\Microsoft Visual Studio .NET 2003\VB7\VBProjectItems\Local Project Items` para VB.NET y `C:\Archivos de programa\Microsoft Visual Studio .NET 2003\VC#\CSharpProjectItems\Local Project Items` para C#, pero en este caso debemos crear una subcarpeta donde dejar el archivo `.vsz`.

Para cambiar la imagen del asistente que aparece en los cuadros de diálogo de Visual Studio, basta con que creamos un archivo de icono (`.ico`) con el mismo nombre que el archivo `.vsz` y lo coloquemos en esa misma carpeta.

Depuración del asistente

Con todo esto, ya podemos poner un punto de ruptura en el método `Execute` de nuestro asistente y empezar a depurar. Se abrirá una segunda instancia de Visual Studio (ya que lo configuramos así anteriormente) y en esta segunda instancia podremos abrir los cuadros de diálogo de “Nuevo proyecto” o “Nuevo archivo”. Si el asistente está registrado correctamente, aparecerá en esos cuadros de diálogo y al pulsar sobre él la ejecución se detendrá en nuestro punto de ruptura y podremos depurar el asistente. Cuando todo esté probado, sólo nos quedará crear el programa de instalación para distribuirlo a otros usuarios. ☺

Bibliografía

- [1] Creating a wizard (MSDN Visual Studio .NET)
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vsintro7/html/vxconCreatingWizard.asp>
- [2] Capítulo 9 (“Visual Studio .NET Wizards”) del libro “Inside Microsoft Visual Studio .NET 2003”, de **Craig Skibo** y otros: <http://www.microsoft.com/mspress/books/6425.asp>.

Ha Llegado el momento de entrar en pista

MSDN y CampusMVP te ofrecen una oportunidad única

Sólo durante este mes de octubre de 2006, matricúlate en nuestro Pack "Certificación".

Podrás formarte en las últimas tecnologías de desarrollo **Microsoft** y obtener de forma **gratuita** tu **certificación oficial**.

Con el Pack "Certificación" tendrás: acceso a **seis cursos on-line** de la especialidad que elijas, **tutorías** por parte de los **Trainers MVP** y **MCT**, un manual oficial de **Microsoft** (MOC), un **Exam Voucher** que te permitirá hacer tu **examen oficial** de certificación **Microsoft** antes del **31 de Diciembre de 2006**.

© 2006 Microsoft

Recuerda que si trabajas en una **empresa española** tendrás aún **más ventajas**

Pide más información llamando al **902 100 986** o entrando en **www.campusmvp.com**

Especialidad Desarrollo Web

CURSOS BASE

- 1 Lenguaje de programación C# e introducción a .NET 2.0
- 2 Programación multi-hilo en .NET

MANUAL OFICIAL

- 1 Advan. Web Applications Technologies with Visual Studio .NET

CURSOS DE ESPECIALIZACIÓN

- 1 Desarrollo aplic. Web con ASP.NET 2.0
- 2 Desarrollo de servicios Web con .NET
- 3 Desarrollo aplic. Web para dispositivos móviles
- 4 Seguridad avanzada de Internet Information Server

Especialidad Acceso Datos

CURSOS BASE

- 1 Lenguaje de programación C# e introducción a .NET 2.0
- 2 Programación multi-hilo en .NET

MANUAL OFICIAL

- 1 Advan. Data Access with Visual Studio .NET

CURSOS DE ESPECIALIZACIÓN

- 1 Lenguaje SQL y T-SQL para SQL Server .NET
- 2 Admin. de SQL Server .NET para programadores
- 3 Acceso a datos con ADO.NET 2.0
- 4 Diseño y distribución de informes con Visual studio .NET

Especialidad Aplic. Empresariales

CURSOS BASE

- 1 Lenguaje de programación C# e introducción a .NET 2.0
- 2 Programación multi-hilo en .NET

MANUAL OFICIAL

- 1 Advan. Distributed Appl. Development with Visual Studio .NET

CURSOS DE ESPECIALIZACIÓN

- 1 Diseño e implementación aplic. datos en N-capas con .NET
- 2 Desarrollo de servicios Web con .NET
- 3 Desarrollo aplic. Web para dispositivos móviles
- 4 Gestión de proyectos de software con Team System

Especialidad Aplic. Móviles

CURSOS BASE

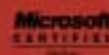
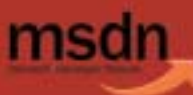
- 1 Lenguaje de programación C# e introducción a .NET 2.0
- 2 Programación multi-hilo en .NET

MANUAL OFICIAL

- 1 Advan. Data Access with Visual Studio .NET

CURSOS DE ESPECIALIZACIÓN

- 1 Acceso a datos con ADO.NET 2.0
- 2 Desarrollo aplic. móviles con .NET Compact Framework 2.0
- 3 Desarrollo aplic. Web para dispositivos móviles
- 4 Técnicas de escritura de código seguro





Daniel Sabater

Impresión desde CRM con WordML y FetchXML

Se trata de una extensión de Microsoft Dynamics CRM que permite convertir la información de cualquier entidad CRM (oferta, factura, etc.) en un documento Word, para imprimir o enviar por correo electrónico. Esta alternativa a Reporting Services se basa en FetchXML y una plantilla WML a la que añadimos etiquetas XML que transformaremos con XSL para obtener el documento final.

» **En este artículo describimos** una manera de aumentar las posibilidades de MS-CRM con una funcionalidad que en muchos casos “se le supone” a la herramienta: la impresión de datos. No me refiero a una impresión “plana” que sirva para uso interno, sino a un formato más presentable: con gráficos, tablas etc. El tipo de documento que presentaríamos a un cliente sin avergonzarnos. Vaya por delante que la herramienta número uno para esta tarea es SQL Reporting Services. Por potencia, flexibilidad y robustez es la mejor manera de volcar a papel cualquier información contenida en nuestro CRM. De todas formas, no está mal saber que existen alternativas como la que presentamos en este artículo. Se trata de aprovechar la coincidencia de Office y CRM en un lenguaje de datos: XML. Vamos por partes: desde la versión 2003, Office integra el lenguaje **WordML** para expresar documentos Word en un lenguaje XML. Por otro lado, CRM permite a través del lenguaje **FetchXML** lanzar consul-

Desde la versión 2003, Office integra el lenguaje WordML para expresar documentos Word en un lenguaje XML

tas a la aplicación que devuelven a su vez datos en XML. Seguimos dentro de XML: la tecnología XSL permite hacer transformaciones XML entre un documento origen y uno destino. De modo que si generamos la XSL necesaria, los datos de CRM se pueden convertir en un documento Word de la complejidad que necesitamos.

La pregunta que surge ahora es ¿cómo generamos esta transformación XSL? Ahí radicaba el problema de esta aproximación antes de la aparición de la WordML Inference Tool (**wml2xslt**). Esta herramienta permite generar una XSL a partir de un documento Word cualquiera: a este documento lo llamaremos a partir de aquí la semilla. Este es el punto fuerte de esta aproximación: mientras que con Reporting Services el cliente tiene que dar una idea aproximada de qué aspecto debe tener su informe, con la aproximación WML el cliente puede darnos directamente un documento ejemplo que

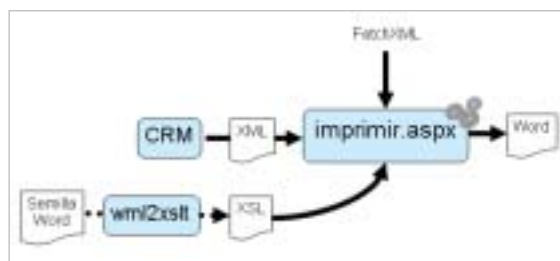


Figura 1. XML + XSL = WML

SWREUSER *Optimizando su proceso de desarrollo de software*

It's your knowledge, reuse it.



- **Una herramienta CASE única**
- **Soporte a la reutilización de software (metodología IRM)**
- **Cobertura completa del ciclo de vida: requisitos, riesgos, modelado UML, generación de código, pruebas, trabajo colaborativo, trazabilidad total**
- **Pensada por y para diseñadores .Net**



contact@reusecompany.com | www.reusecompany.com

muestra el aspecto exacto que quiere que tenga su máscara de impresión. El desarrollador sólo tendrá que agregar los campos XML en los lugares adecuados del documento. A partir del documento semilla, la herramienta `wml2xslt` genera la transformación XSL necesaria. Sólo falta un punto: ¿cómo conseguimos los campos XML a insertar en el documento Word? Una opción es consultar el esquema de CRM en la dirección `http://<miser-vidorcrm>/sdk/list.aspx` y construir un ejemplo XML “a mano”. Una opción más rápida es fabricar un pequeño programa que vuelque desde CRM esta información, que es lo que veremos en la próxima sección.

FetchXML es un lenguaje XML para realizar consultas integrado en la plataforma de CRM. Permite establecer criterios de búsqueda, columnas deseadas, ordenación, etc. A cualquiera que haya hecho una consulta SQL le resultará familiar su funcionalidad –aunque no su sintaxis–. Por otro lado, cualquiera que haya hecho una búsqueda avanzada desde la interfaz de usuario está usando FetchXML sin saberlo. Una búsqueda guardada no es otra cosa que una consulta FetchXML almacenada que se puede lanzar a petición del usuario. Lo que véis en el fuente 1 es un ejemplo de FetchXML que devuelve información de todas las ofertas (`quotes`) cuyo `quoteid` sea “xxx” (en negrita).

```
<?xml version='1.0' encoding='utf-8' ?>
<fetch mapping='logical' distinct='true'>
  <entity name='quote'>
    <filter>
      <condition attribute='quoteid' operator='eq'
        value='xxx' />
    </filter>
    <attribute name='quotenumber' />
    <attribute name='name' />
    <attribute name='totalamount' />
    <attribute name='totaltax' />
    <attribute name='totallineitemamount' />
    <attribute name='customerid' />
    <link-entity name='systemuser' from='systemuserid'
      to='owninguser'>
      <attribute name='fullname' />
    </link-entity>
    <link-entity name='account' from='accountid'
      to='customerid'>
      <attribute name='name' />
    </link-entity>
  </entity>
</fetch>
```

Fuente 1. FetchXML

Su sintaxis es sencilla de manejar y nos permite acceder de manera segura, a través del servicio Web de CRM (`crm.service.asmx`), a consultas complejas. Y

FetchXML es un lenguaje XML para realizar consultas integrado en la plataforma de CRM

cuando digo segura, me refiero a que tiene en cuenta el nivel de seguridad definido en CRM para el usuario que hace la consulta. Por ejemplo, aunque pida todas las cuentas de CRM, la consulta sólo devolverá aquellas para las que el usuario que la lanza tiene permisos de lectura.

Ya sólo nos falta un pequeño programa que lance todo este montaje. En nuestro caso, se trata de una página ASPX que lanza la consulta, recoge los resultados, aplica la transformación y vuelca el resultado al navegador del cliente. Para más comodidad, hemos creado una serie de parámetros que nos permiten escoger la consulta (FML) y la transformación (XSL). El atributo que contiene el identificador del registro lo añade CRM automáticamente con el nombre `oid` gracias al parámetro `PassParams=1`. Copiamos la carpeta `imprimir` con el fichero `imprimir.aspx` al directorio de la Web de CRM. Solo un detalle: la consulta debe cambiar dependiendo del registro que queramos imprimir, algo que configuraremos en el atributo `value` del elemento `condition` dentro de `filter`. Es decir, accedemos al nodo `/fetch/entity/filter/condition/@value` y cambiamos su valor por el identificador que nos pase CRM.

```
string id = Request['oid'];
XmlElement nodoCondicion = (XmlElement)docFetch.SelectSingleNode(
  '/fetch/entity/filter/condition');
nodoCondicion.SetAttribute('value', id);
```

Fuente 2. Detalle de `imprimir.aspx`

Otro detalle de `imprimir.aspx` en el que podemos fijarnos es la realización de la consulta. El SDK de CRM nos permite lanzar consultas al sistema de manera muy sencilla a través del servicio Web `crm.service.asmx` y el método `fetch`:

```
CrmService crm = new CrmService();
crm.Url = 'http://maquinacrm/mscrmservices/2006/crm.service.asmx';
string consulta = '<fetch>...</fetch>';
string resultado = crm.Fetch(consulta);
```

Fuente 3. Detalle de `imprimir.aspx`

Por último, el espacio de nombres `Xml.Xsl` nos proporciona clases que dan acceso al motor de transformaciones XSL de .NET. Solo tenemos que llamar al método `Transform` para aplicar una transformación XSL a un documento XML. Una opción cómoda es volcar el resultado de la transformación al *stream* asociado al objeto `Response` del formulario Web:

```
XslTransform xsl = new XslTransform();
xsl.Load(ficheroXsl, null);
...
xsl.Transform(docXml, null, Response.OutputStream,
null);
```

Fuente 4. Detalle de `imprimir.aspx`

La configuración de nuestra aplicación Web solo requiere un parámetro: la dirección del servidor CRM contra el que va a trabajar. Esto se configura como en cualquier aplicación ASP.NET, en el fichero `web.config`. Hemos añadido una sección `appSettings` donde añadimos dicho parámetro:

```
<appSettings>
<add key='UrlServidorCRM' value='http://maquinacrm:9999/'>
</appSettings>
```

Fuente 5. Detalle de `web.config`.

Entramos ahora en la configuración de CRM. CRM permite agregar a su interfaz de usuario nuevas secciones, opciones de menú y botones. El fiche-

```
<configuration version='3.0.0000.0'>
<Entities>
<Entity name='quote'>
<ToolBar ValidForCreate='0' ValidForUpdate='1'>
<Button Title='Imprimir' I
con='_imgs/ico_18_debug.gif'
Url='/imprimir/imprimir.aspx?xsl=
oferta&fml=oferta'
PassParams='1' />
<Button Title='Generar Semilla'
Icon='_imgs/ico_18_debug.gif'
Url='/imprimir/imprimir.aspx?xsl=oferta&fml=
oferta&semilla=1'
PassParams='1' />
</ToolBar>
</Entity>
</Entities>
</configuration>
```

Fuente 6. `isv.config.xml` durante el desarrollo

El espacio de nombres `Xml.Xsl` nos proporciona clases que dan acceso al motor de transformaciones XSL de .NET

ro `isv.config.xml` es el que especifica cuáles son estas nuevas opciones y en qué consisten. Inicialmente, agregamos dos botones a la ficha de la entidad que nos interesa. Supongamos que es el objeto `oferta` (`quote`) lo que queremos imprimir. Fabricamos un botón para ayudarnos a generar la semilla y otro para imprimir el documento. El primer botón sólo lo usaremos durante la fase de desarrollo, más adelante lo inhabilitaremos (véase el fuente 6).

Con el primer botón ejecutaremos la página `imprimir.aspx` con el parametro `semilla=1`. Esto generará un fichero en el directorio `imprimir` con nombre `semillaOferta.xml`. Es importante que lancemos este proceso desde un registro de CRM que tenga todos los datos que queramos imprimir; de otra manera no aparecerán en la semilla. Abrimos este fichero con Word 2003: veremos los distintos campos devueltos por CRM como etiquetas XML (ojo: los campos en blanco no aparecerán).

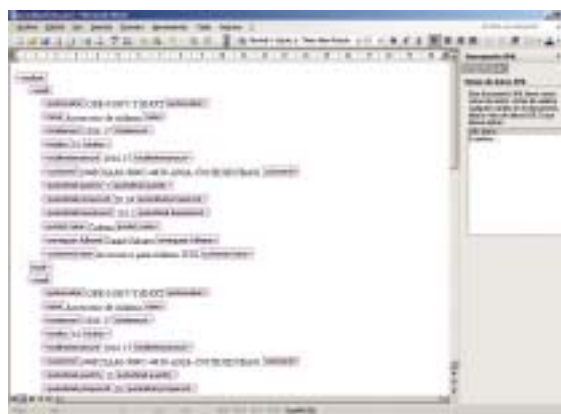


Figura 2. `SemillaOferta.xml` en Word.

Por otro lado, sin cerrar el otro fichero, abrimos en el mismo Word el ejemplo de impresión que nos ha proporcionado el cliente. Seleccionamos todo el fichero y lo copiamos al fichero `semilla.xml` que aún mantenemos abierto. Ahora queda la parte más tediosa: colocar cada cosa en su sitio trasladando las etiquetas XML a su lugar correspondiente.

Esta forma alternativa de imprimir mediante XML se puede convertir en la práctica en una herramienta de productividad muy ventajosa



Figura 3. **Semilla.xml** con el fichero agregado.

Ya estamos terminando: nuestro fichero **semilla.xml** casi está listo para ser utilizado. Sólo tenemos que guardarlo con formato XML a través del menú “Fichero” de Word en la opción “Guardar como...” escogiendo como formato XML (ojo: la casilla “Guardar datos solamente” debe estar inhabilitada). La semilla ya está lista: podemos cerrar todos los documentos de Word.

Lo siguiente es lanzar la aplicación **wml2xslt** que, como su nombre sugiere, convierte un documento WML en una transformación XSL. Su sintaxis es:

```
wml2xslt semillaOferta.xml -o oferta.xsl
```

Con este último punto acabamos de generar la transformación XSL que necesita nuestra página **imprimir.aspx** con el siguiente formato (esto es lo que configuramos previamente en el fichero de configuración **isv.config.xml**):

```
imprimir.aspx?fml=oferta&xsl=oferta
```

Para probar la impresión, vamos al registro de CRM que hemos creado para las pruebas, abrimos su formulario y pulsamos el botón “Imprimir”. Aparecerá un diálogo de confirmación para guardar o abrir el documento, que

tendrá como título el nombre de la XSL pero con extensión **.doc**.

Por último, si todo ha ido bien, ya podemos quitar de CRM el botón de “Generar semilla”. Lo más cómodo sería comentar la línea, lo que en sintaxis XML tiene el siguiente aspecto:

```
<configuration version='3.0.0000.0'>
  <Entities>
    <Entity name='quote'>
      <ToolBar ValidForCreate='1' ValidForUpdate='1'>
        <Button Title='Imprimir' Icon='/_imgs/ico_18_debug.gif'
          Url='/imprimir/imprimir.aspx?xsl=oferta&fml=oferta'
          PassParams='1' />
        <!-- Button Title='Generar Semilla'
          Icon='/_imgs/ico_18_debug.gif'
          Url='/imprimir/imprimir.aspx?xsl=oferta&fml=oferta&semilla=1'
          PassParams='1' / -->
      </ToolBar>
    </Entity>
  </Entities>
</configuration>
```

Fuente 7.isv.config.xml definitivo.

Conclusiones

En este artículo hemos visto una forma alternativa de imprimir desde CRM a través de la tecnología XML y gracias a la inclusión del lenguaje de consulta FetchXML en la aplicación. En un principio puede parecer largo, pero con la práctica veréis cómo se puede convertir en una herramienta de productividad muy ventajosa. Recordad que el trabajo de decir “dónde va cada cosa”, de elegir gráficos, diseñar tablas, etc., lo está realizando el cliente, que además estará más satisfecho por ver que el resultado es exactamente como él lo quiere, y que los cambios que haya que realizar se aplican rápidamente: solo hay que enviarle la semilla y pedirle que cambie lo que quiera para luego repetir el paso de generación con **wml2xslt**. Toda la estructura que hemos montado puede permanecer intacta, cambiando como mucho el fichero **.fml** para incluir o excluir atributos. ○

Microsoft Sharepoint Products & Technologies

CUADERNOS TÉCNICOS

dotNetMania

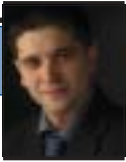


Gracias a las tecnologías de Microsoft Sharepoint podremos implantar desde herramientas colaborativas a auténticos portales corporativos con el mínimo esfuerzo, centralizando la información en servidores de alta escalabilidad y con la garantía de poder gestionar la información con las herramientas de Microsoft Office 2003. Este documento está dirigido a los profesionales de TI que necesitan conocer a fondo Sharepoint, su implantación, su explotación, su extensibilidad y su personalización empleando las herramientas de plataforma .NET



PVP: 24,50 €
20% descuento
para suscriptores

www.dotnetmania.com



Jorge Ramo

Seguridad: Gestión de tickets de sesión

Supongamos que estamos trabajando en un entorno de autenticación en que se utilizan diversos datos biométricos (análisis de retina, huellas dactilares, análisis de voz, etc.), y tras aplicar una función sobre ellos se obtiene un identificador solo conocido por el sistema. Cualquier persona que quisiera suplantar nuestra identidad debería conocer todos estos datos para plantear un ataque con éxito, puesto que con un solo error en alguna de las variables el identificador resultante no sería válido. En este artículo veremos cómo plantear este escenario en nuestras aplicaciones ASP.NET imitando el modelo del protocolo Kerberos.

>> Introducción a Kerberos

Internet es un medio inseguro... Así empieza la introducción al protocolo Kerberos: invitándonos a reflexionar sobre la necesidad de implementar mecanismos de seguridad en nuestras aplicaciones. Una buena práctica consiste en que estos mecanismos estén validados por expertos en seguridad, dado el enorme riesgo de vulnerabilidades no controladas si los construimos nosotros mismos desde cero.

Kerberos es un protocolo de autenticación desarrollado por el MIT basado en criptografía asimétrica y tickets de sesión con una más que probada eficacia en sistemas como, por ejemplo, la autenticación de usuarios bajo Windows 2000. Así pues, ¿por qué no adaptar su modelo al sistema de seguridad de nuestras aplicaciones?

Nuestra solución

Planteemos ahora un entorno de un portal Web desarrollado en ASP.NET, con multitud de usuarios accediendo desde Internet a una serie de recursos internos y con autenticación basada en credenciales de usuario. Una buena práctica para aumentar la seguridad de este entorno consiste en el uso de tickets de sesión protegidos por criptografía y que otorguen acceso temporal a los recursos.

Dependiendo del grado de seguridad que necesitamos, podemos utilizar criptografía de clave asimétrica

Kerberos utiliza criptografía asimétrica, de forma que la única credencial presentada por el usuario es su firma

o simétrica. El uso de la primera opción implica que cada usuario que accede al portal necesita de una clave pública y privada facilitada por una entidad certificadora. Kerberos utiliza criptografía asimétrica en su protocolo, de forma que la única credencial presentada por el usuario es su firma, dato suficiente para su reconocimiento. No obstante, el uso de certificados y criptografía de clave asimétrica, pese a ser más seguro, no siempre es posible, sobre todo si queremos dar de alta a usuarios de manera ágil y dinámica.

Supondremos en nuestro ejemplo el uso de criptografía de clave simétrica en el servidor, de forma que nuestra implementación será transparente para el usuario de la aplicación.

El ticket de sesión

Estudiemos ahora qué es y de qué se compone un ticket de sesión. De la composición del ticket dependerá el grado de seguridad de la solución. En Kerberos, por ejemplo, un ticket posee 6 componentes.

En nuestra implementación adaptaremos la definición de ticket a nuestras necesidades, dependiendo del contexto y del grado de seguridad que necesitamos:

Ticket Kerberos	Nuestro Ticket
<ul style="list-style-type: none"> Nombre de cliente Nombre de servidor Dirección de red Sello de tiempo Tiempo de vida Clave de sesión 	<ul style="list-style-type: none"> Clave de ticket (sirve como identificador) Nombre de usuario (el introducido por el usuario para validarse) Sello de tiempo del momento de generación del ticket Sello de expiración del ticket (en blanco mientras el ticket esté activo)

Tabla 1

Para la generación de la clave del ticket utilizaremos criptografía, encriptando los diferentes datos de usuario: IP de acceso, identificador de sesión asignado por ASP.NET, cadena del navegador y, en general, cualquier dato identificativo del origen del usuario. La clave de ticket obtenida, por tanto, dependerá de todos los datos recopilados.

Para obtener un entorno de alta seguridad completo deberemos implantar unas credenciales seguras de acceso

Funcionamiento del sistema

Una vez definido el ticket, veamos paso a paso cómo se comporta el sistema:

1. El usuario introduce sus credenciales en el formulario de acceso.
2. Las credenciales se validan contra el sistema de registro de usuarios (Active Directory, base de datos, formulario, etc.).
3. Si la validación es correcta, el servidor recoge los datos necesarios (usuario, IP, agente, ID de sesión)

y, mediante criptografía simétrica (en nuestro ejemplo), se genera una clave de ticket, consistente en una cadena alfanumérica.

4. El sistema almacena los datos del ticket y su clave asociada en la base de datos, registrándolo como ticket válido y activo.
5. Por cada acción que realice el usuario, el sistema deberá recoger los datos necesarios y volver a generar la clave de ticket, comprobando que sea la misma que está almacenada en la base de datos, y que el ticket a la cual pertenece está aún activo. Además, debemos comprobar también que la diferencia entre el momento de generación de la última acción y el actual no superen el plazo de expiración (*timeout*) previsto en nuestra aplicación. En tal caso, se debe denegar la acción, informar que el ticket ha expirado y rellenar su fecha y hora de expiración.
6. El usuario se desconecta y el ticket pasa a ser inactivo.

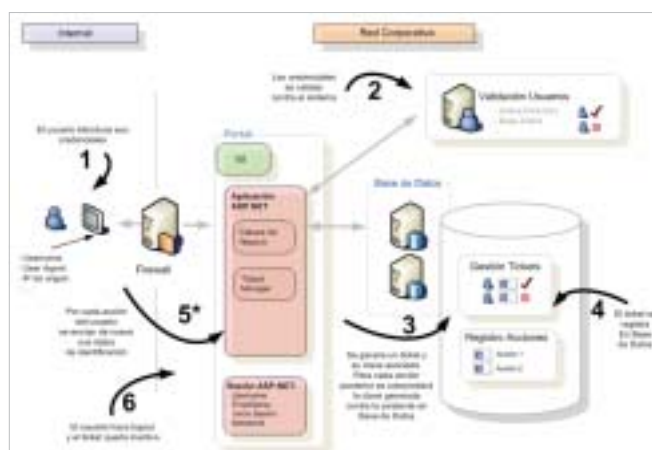


Figura 1

Credenciales de acceso

La implementación de tickets de sesión que hemos visto protege de ataques de robo de sesión (*session hijacking*) una vez el usuario ha entrado en el sistema. Como ya comentábamos inicialmente, un usuario que desee hacerse pasar por nosotros deberá suplantar todos los datos contenidos en el ticket, puesto que un cambio en alguno de ellos generaría un ticket no válido.

Para obtener un entorno de alta seguridad completo deberemos proteger también el acceso al sistema implantando unas credenciales seguras de acceso. Una posible solución consistiría en utilizar los datos de generación de la clave del ticket y asociarlos al usuario como parte de sus credenciales. No obstante, debemos tener en cuenta que si alguno de estos datos cambia, no será

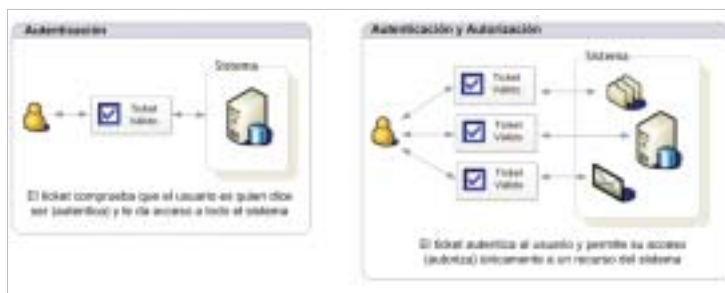


Figura 2

posible el acceso del usuario. Por ejemplo, si asociamos la dirección IP de un usuario como parte de sus credenciales, le será imposible acceder a la aplicación desde una ubicación con diferente IP.

La implementación con más alto grado de seguridad consiste en el uso de certificados de clave asimétrica y una entidad certificadora que actúe de intermediario entre el portal y los usuarios. Las funciones principales de este tercero de confianza serían generar las claves pública y privada de los usuarios y certificar su validez. Esta última implementación sería la más parecida al modelo Kerberos.

```
// Valida las credenciales del usuario,
// Genera y registra el ticket de usuario
Login(Credentials usercredentials) : Bool
{
    if (ValidateUser(userCredentials))
    {
        Ticket userTicket = GenerateTicket(userCredentials.Username);
        Register(userTicket);
        return true;
    }
    else
    {
        return false;
    }
}

// Genera un ticket seguro
GenerateTicket(Credentials usercredentials) : Ticket
{
    Ticket generateTicket = new Ticket();
    generateTicket.Ticketid = Encrypt(userCredentials.Username);
    generateTicket.GenerationTimeStamp = Now;
    return generateTicket;
}

// Valida al usuario a través del ticket y ejecuta una cierta acción.
DoAction(userParams, Action action)
{
    Ticket tempTicket = GenerateTicket(userParams);
    Ticket userTicket = GetActiveTicket(userParams.Username);
    if ( tempTicket.Ticketid == userTicket.Ticketid )
    {
        BeginTransaction();
        RegisterAction(userTicket, Now);
        // Implementación de la acción.
        EndTransaction();
    }
}
```

Fuente I. Pseudo código de la clase `TicketManager`.



Vulnerabilidades

El grado de seguridad del sistema depende de:

1. La facilidad de ser suplantados los datos en los que se basa el sistema para garantizar que el usuario que realiza la petición es el que ha iniciado sesión. Variables como el agente de usuario (*User Agent*) del navegador son fácilmente obtenibles por un usuario malicioso que intente suplantarlos. Por otra parte, la dirección IP del usuario también es fácilmente obtenible pero no tan fácil de suplantar, a menos que el atacante se encuentre detrás de nuestro mismo proxy.
2. El nivel de seguridad de las credenciales de acceso. De nada nos sirve proteger mediante validación por ticket las acciones del usuario si el acceso al sistema se realiza mediante una contraseña frágil y fácil de obtener.
3. El método utilizado para generar la clave del ticket en el servidor. Si utilizamos criptografía simétrica, debemos proteger la clave privada utilizada en el proceso de encriptación.
4. El nivel de seguridad del servidor del portal. Debemos proteger la validación y generación de tickets de ataques externos, puesto que si un atacante consigue comprometer esta parte puede tener acceso a todo el sistema.
5. Otra opción recomendable a implantar es el uso de encriptación en todas las comunicaciones. De esta forma se evita que cualquier atacante pueda “escuchar” nuestra red y recoger los datos del usuario necesarios para la elaboración de la clave de ticket.

Aplicación de la solución

La implementación mediante tickets no es únicamente aplicable a aplicaciones Web, sino que es perfectamente extensible a cualquier tipo de aplicación cliente/servidor que requiera autenticación entre usuarios o bien entre servidores.

Otra opción consiste en extender nuestra solución y generar tickets solo válidos para ciertos recursos, convirtiendo así a los tickets en una solución válida para la autenticación y autorización de usuarios. ○



Joaquín Gracia

Cómo personalizar el ribbon de Office 2007

Siguendo los pasos de este tutorial podrás personalizar el ribbon de Office 2007 de una forma realmente sencilla y fácil. Explicamos cómo crear un add-in de Office 2007 basado en la interfaz IDTExtensibility2 y la forma en la que puedes añadir pestañas y botones propios en el ribbon.

» **Uno de los problemas** que Office 2003 tiene es que hay un gran número de sitios en donde uno tiene que mirar para encontrar la funcionalidad que desea: menús, submenús, barras de herramientas visibles, barras de herramientas que sólo se acceden desde algunos menús, barras de herramientas emergentes, paneles de tareas, paneles de tareas que sólo aparecen bajo un determinado contexto, etc.

Para realizar algunas tareas los usuarios tienen que buscar en diferentes sitios las acciones que necesitan, algo que se convierte a veces en una tarea pesada de realizar.

En la nueva interfaz de usuario de Office 2007 se introduce el *ribbon*. El *ribbon* es una región en la parte superior de la pantalla que muestra de una forma sencilla el conjunto de comandos que se pueden realizar con un programa.

El *ribbon* está organizado por pestañas, y éstas a su vez tienen grupos de controles agrupados por funcionalidad, como se puede ver en la figura 1.



Figura 1. El ribbon de Office 2007.

Una de las características que ofrece Office 2007 es que se puede personalizar la interfaz de usuario de las aplicaciones que lo componen añadiendo elementos al *ribbon*. La personalización del *ribbon* de Office 2007 se puede realizar a dos niveles:

- *A nivel de documento/plantilla*. A este nivel sólo veremos la personalización que hayamos realiza-

do si abrimos un documento o una plantilla específica. Para realizar esta personalización modificaremos nuestro documento o plantilla, incluyendo en ellos el XML de personalización de la interfaz de usuario.

- *A nivel de aplicación*. De esta forma, la personalización de la interfaz de usuario estará siempre disponible, independientemente del documento que tengamos abierto. Para obtener la personalización a nivel de aplicación deberemos crear un *add-in* que se cargará siempre que se cargue la aplicación.

En el ejemplo que presentaremos a continuación, realizaremos la personalización de la interfaz de usuario de Office 2007 a nivel de aplicación; concretamente, vamos a personalizar el *ribbon* de Excel 2007.

Creando un add-in de Office 2007

Como hemos dicho antes, lo primero que necesitamos es crear un *add-in* compartido (*shared add-in*) que se instancie cuando Excel 2007 se cargue.

Abrimos Visual Studio 2005 y creamos un nuevo proyecto de *add-in* compartido:

- Seleccionamos la opción del menú “File->New->Project” (figura 2).
- Nos aparece el asistente de *add-ins*. Pulsamos “Siguiente”.
- Seleccionamos el lenguaje de programación que deseamos usar (en este caso C#). “Siguiente”.
- Luego seleccionamos las aplicaciones en las cua-

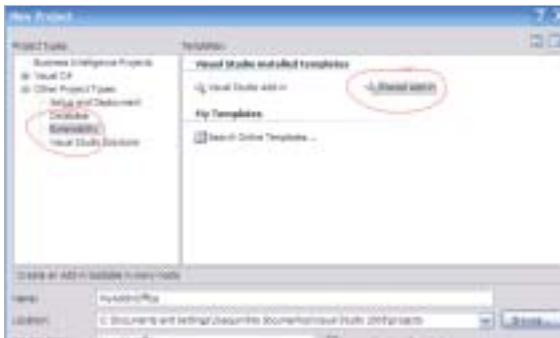


Figura 2

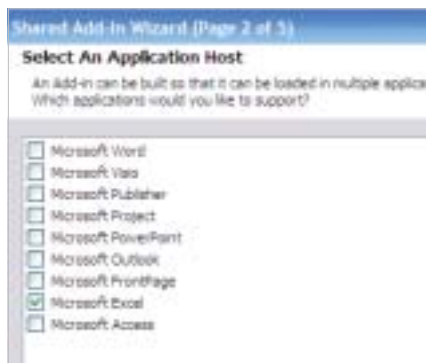


Figura 3

les nuestro *add-in* se cargará. En este ejemplo, voy a seleccionar solo “Excel” (figura 3). Pulsamos “Siguiente”.

- Ponemos un nombre a nuestro *add-in*, así como una descripción. “Siguiente”.
- Por último, marcamos las dos opciones finales:
 - Quiero que mi *add-in* se cargue cuando la aplicación se cargue.
 - Mi *add-in* debería estar disponible para todos los usuarios de la máquina en la que se instale.
- Pulsamos “Siguiente”.
- Pulsamos “Finalizar”.

Este asistente nos creará dos proyectos en una solución (figura 4):

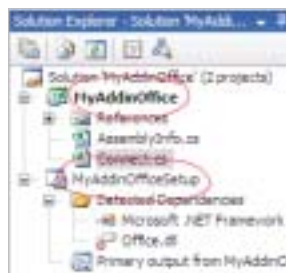


Figura 4

- El proyecto del *add-in* propiamente dicho. Contiene el código de nuestro *add-in*.

Método	Objetivo
<code>void OnAddInsUpdate(ref Array custom)</code>	Método que se llama cuando otro <i>add-in</i> se carga.
<code>void OnBeginShutdown(ref Array custom)</code>	Método que se llama cuando la aplicación <i>host</i> se va a descargar.
<code>void OnConnection(object Application, ext_ConnectMode ConnectMode, object AddInInst, ref Array custom)</code>	Método que se llama cuando el <i>add-in</i> se carga.
<code>void OnDisconnection(ext_DisconnectMode RemoveMode, ref Array custom)</code>	Método que se llama cuando el <i>add-in</i> se va a descargar.
<code>void OnStartupComplete(ref Array custom)</code>	Método que se llama cuando el <i>add-in</i> se ha cargado por completo.

Tabla 1. Miembros de IDTExtensibility2

- El proyecto de la instalación del *add-in*. Este crea un instalable a partir de resultado del proyecto anterior.

Para que el *add-in* sea manejado correctamente por Excel debemos implementar la interfaz **IDTExtensibility2**, que contiene los métodos que se muestran en la tabla 1.

Cuando creamos un proyecto de *add-in* se nos proporciona una implementación por defecto de esta interfaz, así que generalmente no tenemos que preocuparnos por ella.

Personalizando la interfaz de usuario

Lo siguiente que tenemos que hacer es implementar la interfaz **IRibbonExtensibility**, que nos permite personalizar la interfaz de usuario de la aplicación huésped, en este caso Excel, cargando la definición del mismo en XML.

Esta interfaz sólo tiene un miembro:

```
public string GetCustomUI(string ribbonID)
```

que debe devolver una cadena XML con la definición de nuestra interfaz de usuario personalizada.

Para nuestro ejemplo vamos a usar el XML del fuente 1.

En el XML anterior se define una nueva pestaña llamada “MiEmpresa” con el elemento `<tab id="MiEmpresa" label="MiEmpresa">`. A continuación, indicamos que queremos crear un grupo de controles que llamaremos “Herramientas”, lo que realizamos con el elemento `<group id="Herramientas" label="Herramientas">`; dentro de este grupo de controles añadimos un botón, estableciendo que cuan-

```
<customUI xmlns=
"http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon>
    <tabs>
      <tab id="MiEmpresa" label="MiEmpresa">
        <group id="Herramientas" label="Herramientas">
          <button id="MisBotones" onAction="HolaMundo"
            size="normal" label="MisBotones" />
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

Fuente 1

do éste se pulse se deberá llamar a la función `HolaMundo`; de esto se encarga el elemento `<button id="MisBotones" onAction="HolaMundo" size="normal" label="MisBotones" />`.

El código que genera el XML del fuente 1 se presenta en el fuente 2.

Añadiendo acciones personalizadas

Ahora crearemos una acción que se ejecutará al pulsar sobre el botón “MisBotones”, para lo cual añadiremos una referencia a la librería de Excel. Pulsamos con el botón derecho sobre la carpeta **Referencias** del proyecto, seleccionamos “Agregar referencia” y añadimos una referencia a **Microsoft.office.Interop.Excel**. También es conveniente añadir la siguiente línea al principio de nuestro programa:

```
using Excel =
    Microsoft.Office.Interop.Excel;
```

```
public string GetCustomUI( string ribbonID)
{
    string customui;

    customui = "<customUI xmlns=\"http://schemas.microsoft.com/office/2006/01/customui\" >";
    customui += "<ribbon>";
    customui += "<tabs>";
    customui += "<tab id=\"MiEmpresa\" label=\"MiEmpresa\">";
    customui += "<group id=\"Herramientas\" label=\"Herramientas\" >";
    customui += "<button id=\"MisBotones\" onAction=\"HolaMundo\"
        size=\"normal\" label=\"MisBotones\" />";

    customui += "</group>";
    customui += "</tab>";
    customui += "</tabs>";
    customui += "</ribbon>";
    customui += "</customUI>";

    return customui;
}
```

Fuente 2

```
public void HolaMundo(IRibbonControl control)
{
    Excel.Application app = (Excel.Application) this.applicationObject;
    app.ActiveCell.Value2 = "¡¡¡Hola Mundo!!!";
}
```

Fuente 3

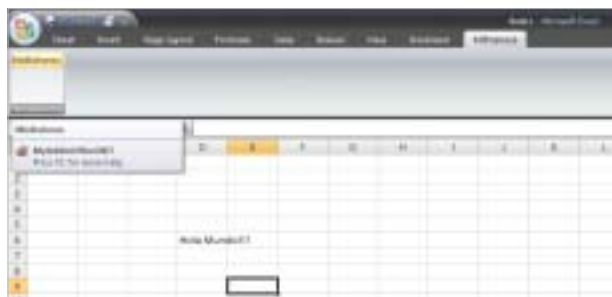
Y ya estamos listos para implementar el método `HolaMundo` que hemos definido en el documento XML como acción asocia-

da al botón (fuente 3). Nuestra acción de ejemplo inserta el texto “Hola Mundo” en la celda de Excel que esté activa en ese momento.

Para finalizar, sólo nos queda compilar nuestra solución y ejecutar el proyecto de instalación. Al ejecutar el proyecto de instalación, el *add-in* se registrará para que sea cargado cuando arranquemos Excel.

Como se puede observar en la figura 5, después de ejecutar la instalación tendremos en Excel una nueva pestaña con un botón con la funcionalidad antes descrita.

En todo momento, podemos ver qué *add-ins* tenemos instalados en las opciones de Excel (figura 6). Desde ahí podemos activarlos y desactivarlos cuando queramos.

Figura 5. Excel después de la instalación del *add-in*.Figura 6. *Add-ins* instalados en Excel.

Conclusiones

Como hemos mostrado en este artículo, la creación de un *add-in* para personalizar el *ribbon* de cualquier aplicación de Office 2007 es realmente sencillo. No obstante, cuando salga la nueva versión de *Visual Studio Tools for Office* (VSTO), el proceso será aún mucho más fácil, ya que esta herramienta incorporará proyectos de *add-in* predefinidos para cada una de las aplicaciones que componen Office 2007. Mientras tanto, podemos desarrollar nuestros prototipos de integración con el *ribbon* siguiendo las pautas que hemos detallado aquí. ☺



Guillermo "Guille" Som

Delegados y eventos

Primera parte: ¿En quién delegas tú?

En este número vamos a tratar de los delegados, y también de los eventos (aunque de estos últimos nos encargaremos con más detalle en el próximo artículo), ya que en .NET están estrechamente relacionados; tanto es así que no podemos definir un evento sin la intervención de un delegado.

>> Delegados y eventos en Visual Basic y C#

Seguramente los programadores de Visual Basic estarán pensando que lo dicho en la entradilla no es totalmente cierto. Bueno, posiblemente no, porque si están leyendo **dotNetManía** sabrán lo que se esconde tras los eventos. Pero no está de más aclararlo para que no queden dudas.

Lo cierto es que Visual Basic es un lenguaje muy protector, y nos “libera” de ciertas tareas para facilitarnos el trabajo real, de forma que nos podamos concentrar en lo que de verdad importa y olvidarnos un poco de ciertos “asuntillos” que en parte solo nos hacen teclear más.

Aclaremos un poco todo esto que acabo de comentar. En Visual Basic, para definir un evento solo hay que escribir la instrucción **Event** seguida de la definición que daremos al método que recibirá los eventos. Por ejemplo, si nuestra clase es del tipo **Button**, podemos definir el evento **Click** de la siguiente forma:

```
Public Event Click( ByVal sender As Object, _  
                  ByVal e As EventArgs )
```

Como podemos comprobar, esa es la definición que usaremos en cualquier formulario que quiera interceptar la pulsación en un botón. Simple, ¿verdad? Veamos ahora cómo tendría que definir ese mismo evento un programador de C#:

```
public delegate void ClickEventHandler(  
    object sender, EventArgs e);  
public event ClickEventHandler Click;
```

En este caso, primero se define un delegado y a continuación hay que definir el evento que debe ser del tipo de ese delegado. Complicado, ¿verdad? Ahora mismo no entraremos en muchos detalles sobre esto, antes veamos cómo se lanzaría ese evento tanto desde Visual Basic como desde C#:

```
' En Visual Basic:  
RaiseEvent Click(sender, e)  
  
// En C#:  
if( Click != null )  
{  
    Click(sender, e);  
}
```

Indudablemente en Visual Basic sigue siendo mucho más fácil, más simple, menos complicado, no tenemos que comprobar nada... Y es cierto, a eso es a lo que me refería con lo de que Visual Basic es muy “protector” y nos libera de ciertos detalles que en realidad no necesitamos saber, o al menos, no es obligatorio que sepamos. Por otra parte a los programadores de C#, seguramente por aquello de que les gusta “escribir más” código, pues... ¡que escriban más! Aunque, como veremos en este artículo, eso ya está cambiando, y ahora podrán hacer también ciertas cosas sin necesidad de escribir tanto, ya que el propio compilador de C# se encargará de algunos aspectos, digamos, de trasfondo. Pero al final, tanto C# como Visual Basic deben seguir las reglas de .NET, y aunque nosotros como usuarios no tengamos que preocuparnos, los compiladores sí que lo harán.

Independientemente de las bromas, tenemos que ser conscientes (sobre todo los programadores de Visual Basic) de que algunas veces el que nos “mimen” tanto no es bueno, ya que nos acostumbramos mal, y cuando creemos que todo va a ser sencillo, llega la versión 2005 y nos dicen que si queremos usar la nueva instrucción **Custom Event** debemos saber manejar los delegados, además de que también debemos saber en qué medida están relacionados con los eventos. Esto a los programadores de C# no les pillaría tan desprevenidos. Así, si ahora les dicen que pueden crear métodos anónimos, y que esos métodos anónimos los podrán crear donde se pueda usar un delegado, o que ya no es necesario usar un constructor para crear un tipo delegado o que por medio de la *covarianza* o la *contravarianza* podrán usar de forma más óptima los delegados, simplemente estarán preparados y sabrán soportar el cambio...

Pero como siempre hay gente nueva, (tanto en C# como en Visual Basic), no está de más que algunos puntos estén totalmente claros, así que eso es lo que vamos a intentar en este primer artículo dedicado a los delegados y a los eventos. Y en los que seguirán, terminaremos por aclarar casi cualquier duda que posiblemente se nos pueda presentar a la hora de trabajar con los eventos y con los delegados.

¿Qué son y para qué sirven los delegados?

Como sabemos, .NET Framework se caracteriza por ser un entorno de código administrado (*managed code*) o lo que es lo mismo, a .NET no le gustan las sorpresas. Si una función tiene que devolver un valor de tipo **String**, debe devolver un valor de tipo **String**; si un método debe recibir dos parámetros de tipo **Object**, eso es lo que recibirá. Y todo esto los lenguajes adscritos a .NET deben respetarlo, ya que de no ser así, .NET no permitirá la ejecución del código. Y esto es aplicable a todo lo que .NET controla, es decir, a todo lo que está bajo su influencia. Lo que no quiere decir que no podamos hacer cosas que .NET no permita; pero si lo hacemos, debemos hacerlo por la puerta falsa. De

esto saben mucho los que han desarrollado con C, incluso los que desarrollan con C#. Aunque en todas las puertas falsas de .NET siempre hay alguien que “está por allí” y revisa que en realidad no hagamos demasiadas trastadas. Esto en otros lenguajes no es así, y por error o porque así lo hayamos previsto, podemos crear grandes problemas, si no, ¿por qué aparecen los fallos de protección general? (las típicas pantallas azules de las versiones anteriores de Windows, que ahora simplemente están remozadas y han cambiado de *look* por un cuadro de diálogo más “mono”).

arbitrario, es decir, que no nos “colemos” donde no debemos, ya que, como ya he dicho, a .NET no le gustan las sorpresas, por eso impone reglas que debemos cumplir; si las cumplimos, pasamos, si no las cumplimos, no nos deja seguir.

Y esto es así por todo lo comentado anteriormente, ya que el CLR quiere seguridad y la única forma de tenerla es creando normas de conducta y de utilización, en este caso, de la memoria o del acceso a esas partes de la memoria en la que están las definiciones de los métodos o funciones.

Un delegado permite acceder a una función de forma casi anónima, ya que simplemente tiene la dirección de memoria de dicha función

Este tipo de problema se debe a un acceso indebido a la memoria, normalmente causado por un acceso a una posición de memoria que no estaba dentro del rango que teníamos permitido. .NET es más estricto y menos permisivo para estas cuestiones, por tanto, si queremos estar bajo el abrigo de la seguridad de .NET debemos seguir sus normas.

Como sabemos, .NET define una serie de tipos de datos, los cuales podemos usar indistintamente desde un lenguaje u otro, ya que independientemente del nombre que cada compilador le dé, en realidad estamos trabajando con los tipos definidos en la librería de clases, o mejor dicho, en el sistema de tipos comunes (CTS). Y los delegados no son una excepción.

Pero... ¿qué es un delegado? Un delegado es una referencia a una función, lo que también se conoce como un puntero a una función, es decir, un delegado permite acceder a una función de forma casi anónima, ya que simplemente tiene la dirección de memoria de dicha función. Y sabiendo la dirección de memoria, podemos acceder a ella. Pero en .NET esto debe estar controlado, de forma que ese acceso no sea

Por tanto, si queremos acceder a un método, tenemos que hacerlo por medio de un puntero controlado, y la forma de controlar ese acceso es definiendo un prototipo en el que indiquemos de qué tipo es ese método, si recibe parámetros, y de hacerlo cuántos y de qué tipo son. Una vez que tenemos definidos todos estos requerimientos, es cuando le decimos al *runtime* de .NET que nos permita acceder a esa función. De esta forma, podrá controlar que estamos accediendo al sitio correcto.

Esa definición del prototipo de función (o método) al que queremos acceder lo hacemos por medio de un delegado. Podemos pensar que un delegado es en cierto modo similar a las interfaces (ver **dotNetManía** n° 16), que como sabemos definen un contrato que debemos respetar. Sabiendo esto, si queremos acceder a una función que devuelve una cadena y que recibe un parámetro de tipo **Cliente**, debemos definir un delegado con esas características, y cuando posteriormente queramos acceder a ese método, en lugar de hacerlo directamente o por medio de un puntero directo, usaremos un objeto del tipo definido por el delegado. Lo que nos lleva a una segunda definición de lo

Los delegados definen la “firma” que los métodos a los que queremos acceder deben tener

que es un delegado, en la que podemos decir que es un tipo especial que nos permite definir la forma de acceder a una función.

Veamos el ejemplo del método que devuelve una cadena y recibe un parámetro de tipo `Cliente`. Ese método lo podemos definir en C# de esta forma:

```
public string MiFuncion(Cliente c)
{
    return "...";
}
```

La forma de usar ese método sería algo así:

```
string s = MiFuncion( new Cliente() );
```

Por supuesto aquí no estamos usando ningún puntero, simplemente estamos accediendo a ese método de forma directa. Es más, debido a que no estamos indicando dónde está dicha función, suponemos que estamos accediendo desde la propia clase en la que está definido. Pero si quisiéramos acceder de una forma más anónima, podríamos definir un delegado que tenga esa definición y solo tendríamos que indicar dónde está definido para permitirnos el acceso, sin necesidad de usar una instancia de la clase que lo define.

¿Por qué tanta complicación? Porque debemos suponer que en ciertas circunstancias no tenemos una forma directa de acceder a ese método, ya que si la tuviéramos, no tendríamos necesidad de complicarnos la vida y usaríamos la forma directa mostrada anteriormente. Por ejemplo, si queremos que desde otra parte del código alguien pueda acceder a ese método sin necesidad de saber si está definido en una clase o en otra, podemos usar un delegado y por medio de dicho delegado acceder al método.

Veamos la definición del delegado que nos permitiría acceder a esa función

y cómo podemos usarlo. Primero definimos el delegado usando la misma “firma” que tiene el método al que queremos acceder:

```
public delegate string MiFuncionDelegado(Cliente c);
```

Fuente 1. Definición de un delegado para acceder a un método definido con la misma firma.

Para usar el delegado, definimos una variable de ese tipo, y como los delegados en realidad son como clases, podemos usar el mismo código que usamos para crear cualquier tipo, con la diferencia de que en el constructor debemos indicarle la función a la que queremos acceder:

```
MiFuncionDelegado mfd = new
    MiFuncionDelegado(MiFuncion);
```

En este caso, también estamos accediendo desde la propia clase al método `MiFuncion`, pero lo dejamos así para mantener las cosas simples.

Para acceder a esa función por medio del delegado que acabamos de crear, lo haremos así:

```
string s = mfd( new Cliente() );
```

Que como podemos comprobar, es un código muy parecido al usado anteriormente, pero con la diferencia de que en este código no usamos el nombre de la función, sino el del objeto creado a partir del delegado.

Ahora supongamos que este último código lo queremos usar en cualquier método de cualquier clase, sin importar cómo y dónde se haya definido, ¿cómo podríamos hacerlo? porque para poder definir una variable como en este caso, deberíamos tener “cerca” la función a la que queremos acceder. Y si la tenemos cerca, pues no necesitamos usar un dele-

gado. Veamos el siguiente código y seguro que esa cercanía no es tan necesaria:

```
public static void usarMiFuncionDelegado(
    MiFuncionDelegado mfd)
{
    string s = mfd( new Cliente() );
}
```

En este caso, tenemos un método que define un parámetro del tipo del delegado, por tanto, podemos llamar a ese método pasándole la dirección de una función que cumpla con la definición del delegado y no importará donde estén definidos, ni el método ni la función a la que accederemos por medio del delegado.

La única condición es que desde donde hagamos la llamada, tengamos acceso a ambos métodos (o funciones), pero no tienen por qué estar todas definidas en una misma clase, ¡ni

siquiera en un mismo ensamblado!

Para acceder a este método lo podemos hacer de esta forma:

```
static void Main(string[] args)
{
    Cliente c = new Cliente();
    usarMiFuncionDelegado(c.MiFuncion);
}
```

Debemos notar en la forma en que llamamos al método que recibe el delegado, ya que simplemente le hemos pasado el nombre de la función. Esto es nuevo en C# 2.0 (aunque no en Visual Basic), y la forma en que tendríamos que hacerlo en las versiones anteriores sería esta otra:

```
usarMiFuncionDelegado(new
    MiFuncionDelegado(c.MiFuncion));
```

Es decir, pasando como argumento un objeto creado a partir del “tipo” del delegado.

De igual forma, el código que mostramos anteriormente en el que usábamos también un constructor de ese tipo lo podríamos haber escrito de esta otra forma, que como vemos, es más simple e igualmente comprensible.

```
MiFuncionDelegado mfd = MiFuncion;
```

Fuente 2. C# 2.0 permite asignar directamente la función sin necesidad de un constructor.

NOTA

En Visual Basic, la forma de acceder a una función siempre es por medio de la instrucción `AddressOf` y la podemos usar también de las dos formas que acabamos de ver, es decir por medio de un objeto del tipo del delegado o de forma directa, en cuyo caso, (al igual que ocurre con C# 2.0), será el propio compilador el que determinará si esa función cumple o no los requisitos del parámetro que espera el método:

```
usarMiFuncionDelegado(AddressOf c.MiFuncion)

usarMiFuncionDelegado(New
    MiFuncionDelegado(AddressOf c.MiFuncion))
```

En las asignaciones también podemos usar cualquiera de las dos formas:

```
Dim mfd As MiFuncionDelegado
mfd = New MiFuncionDelegado(AddressOf MiFuncion)

Dim mfd As MiFuncionDelegado = AddressOf MiFuncion
```

En cualquier caso, lo que estamos pasando es la dirección de memoria de la función.

De todo lo que hemos visto debemos concluir que los delegados definen la “firma” que los métodos a los que queremos acceder deben tener, y que podemos acceder a esos métodos por medio de instancias creadas a partir del delegado, y si estamos usando C# 2.0 el compilador se puede encargar de averiguar qué delegado es el que debe usarse y lo usará de forma transparente para nosotros.

Antes de ver qué relación tiene todo esto con los eventos, repasaremos algunas otras características de los delegados que en el fondo también están relacionadas con los eventos, o en la forma que finalmente las utilizan los eventos, pero que no necesariamente usaremos para trabajar con los eventos.

Usos prácticos de los delegados

Para que nos quede más claro el funcionamiento de los delegados, vamos a ver en qué situaciones podemos usarlos. Algunos de los usos que vamos a ver son exclusivos de C# 2.0, y aunque ya se han tratado en otros números de esta revista, no viene mal darles un repaso. Debido a esa exclusividad de uso en C#, decirle a los lectores que prefieren Visual Basic que no pasen al siguiente artículo, ya que aún quedan cosas que explicar que también son válidas para ese lenguaje, aunque (como es costumbre en esta sección), el código mostrado será prácticamente en

```
ldftn instance string Cliente::MiFuncion(class Cliente)
newobj instance void MiFuncionDelegado::.ctor(object,native int)
stloc.0
ldloc.0
newobj instance void Cliente::.ctor()
callvirt instance string MiFuncionDelegado::Invoke(class Cliente)
stloc.1
ret
```

Fuente 4. El código IL es el mismo para las dos formas mostradas en el fuente 3.

exclusiva para C#. En otro artículo trataremos temas que son específicos de Visual Basic y por consiguiente todo el código

será en ese lenguaje, ya que como comenté anteriormente los delegados son una pieza clave para los eventos, y debemos saber cómo funcionan los delegados para comprender mejor cómo funcionan los eventos.

Novedades de C# respecto a los delegados

En la versión 2.0 de C# (tal como explicó **Octavio Hernández** en el número 20 de esta revista) podemos usar los delegados de forma directa, es decir, sin necesidad de que tengamos que crear una instancia de la clase del delegado al que le pasamos como parámetro del constructor la función a la que queremos apuntar, tal como hemos visto en el código del fuente 2. En esos casos, el compilador comprueba el tipo de la variable que recibe el puntero a la función y si tiene la misma firma que la función, será el propio compilador el que haga el uso adecuado del delegado correspondiente. De hecho, si examinamos el código IL generado por el compilador será el mismo en los dos casos.

```
void usarMiFuncion2(){
    MiFuncionDelegado mfd = MiFuncion;
    string s = mfd(new Cliente());
}
void usarMiFuncion3(){
    MiFuncionDelegado mfd = new
        MiFuncionDelegado(MiFuncion);
    string s = mfd(new Cliente());
}
```

Fuente 3. Las dos formas equivalentes de asignar el puntero a una función.

En el código del fuente 3 tenemos las dos formas de realizar la asignación y en el fuente 4 vemos el código IL que crea el compilador, el cual es exactamente el mismo en ambos casos.

Métodos anónimos

Los métodos anónimos son otra de las novedades de C# que están relacionadas con los delegados. Y es que en C# 2.0 podemos usar una definición de un delegado en cualquier sitio que el compilador esperaría que se asignara un delegado.

El ejemplo más claro del uso de los métodos anónimos es para relacionar un evento con un código, pero como aún no hemos tratado con detalle los eventos, vamos a verlo con el código que estamos usando últimamente. En el código del fuente 5 podemos ver cómo crear un método anónimo con la misma firma que el delegado `MiFuncionDelegado`, pero al definirlo nosotros de forma independiente podemos escribir en el cuerpo de la función anónima lo que creamos conveniente.

```
public static void usarMiFuncion4()
{
    MiFuncionDelegado mfd;
    mfd = delegate(Cliente c)
        {return "Hola " + c.Nombre;};
    string s = mfd(new Cliente("Pepe"));
    Console.WriteLine(s);
}
```

Fuente 5. Definición de un método anónimo.

Como es evidente, en este caso no hace falta usar un método anónimo, ya que sería más fácil mostrar directamente el saludo, en lugar de dar tantas vueltas, pero lo importante es ver cómo se

pueden usar los métodos anónimos, los cuales tienen mayor utilidad con los eventos o cuando queramos simplemente definir la función “apuntada” de forma directa, ya que como hemos visto anteriormente, esa variable que apunta al método la podemos pasar como argumento a otro método.

En cualquier caso, lo que no nos estará permitido hacer es modificar la firma del delegado; por tanto, si se nos ocurre la brillante idea de añadir un nuevo parámetro a la función, tal como vemos en el siguiente código fuente, el compilador nos avisará (entre otras cosas), que `MiFuncionDelegado` no tiene dos argumentos.

```
mfd = delegate(Cliente c, string saludo){
    return saludo + " " + c.Nombre;
};
```

Es importante saber que aunque estemos declarando un “método” anónimo, en realidad no es un método, al menos en el sentido de los ámbitos o cobertura de las variables, ya que desde ese método anónimo podemos acceder a cualquier variable que hayamos declarado anteriormente, y la variable definida en el parámetro también tendrá el mismo ámbito que el método o propiedad que contiene esa definición anónima. Debemos pensar en que el cuerpo del método anónimo en realidad es como cualquier otro bloque de código que podamos incluir dentro de un par de llaves.

En el código del fuente 6 podemos ver ese conflicto entre la variable definida directamente y la definida como parámetro del método anónimo.

```
Cliente c = new Cliente();
usarMiFuncionDelegado(c.MiFuncion);

MiFuncionDelegado mfd;
mfd = delegate(Cliente c) {return "Hola " + c.Nombre;};
string s = mfd(new Cliente("Manolo"));
Console.WriteLine(s);
```

Fuente 6. El ámbito de un método anónimo es el mismo que el del bloque en el que se define.

Covarianza y contravarianza

La primera vez que leí estas dos palabras, pensé que el tema tratado debía ser muy complicado, seguramente para gente más experta que yo en C#. Y como soy un catetillo de pueblo que no tiene estudios, después de leer la descripción de la ayuda de Visual Studio 2005 pensé en la suerte que tenía de que mi lenguaje materno no fuese el C#. Pero si iba a hablar de esto en este artículo, lo lógico era que me empapara del tema, al menos si lo iba a tratar. Y con esto pasa como con casi todo, hasta que no lo tienes entre las manos, no sabes el tacto que tiene.

Sobre la *covarianza*, la documentación nos dice que: *Cuando un método delegado tiene un tipo de valor devuelto que es más derivado que la firma de delegado, se denomina covariante*. En realidad, al leerlo sabiendo qué es lo que significa no era tan rebuscada la definición. Pero para dejarlo en un lenguaje más llano, diremos que esto significa que podemos usar delegados que devuelvan un tipo y el receptor de ese valor puede ser cualquier clase de ese mismo tipo o de cualquier otra clase derivada.

Por ejemplo, en el siguiente código, tenemos la definición de una clase `Persona` y un delegado que devuelve un valor de ese tipo:

```
public delegate Persona PersonaCallback();

public class Persona
{
    // Omitidas las definiciones
    // de las propiedades Nombre y Apellidos
}
```

Y definimos una clase derivada, a la que llamaremos `Colega`, en la que definimos un método estático que devuelve un objeto de ese mismo tipo:

```
public class Colega : Persona
{
    public static Colega NuevoColega()
    {
        return new Colega();
    }
    // Omitida la definición de la propiedad Correo
}
```

Podemos usar esa función como el método al que apuntará el delegado `PersonaCallback`, que como hemos visto, devuelve un objeto del tipo `Persona`:

```
PersonaCallback nColega;
nColega = Colega.NuevoColega();

Colega unColega = (Colega)nColega();
// Omitidas las asignaciones a las propiedades
```

En la asignación a la variable `unColega` debemos hacer una conversión (*cast*) ya que el valor devuelto por el delegado es del tipo `Persona`, independientemente del tipo que devuelva en realidad la función a la que hace referencia ese delegado.

En las versiones anteriores, para conseguir esto mismo teníamos que definir un delegado para cada una de los tipos que quisiéramos devolver. Ni que decir tiene que esto solo lo podemos hacer con C#; en Visual Basic no está permitida esta forma de usar los valores devueltos por un delegado.

Veamos ahora qué nos dice la documentación sobre la *contravarianza*: *Cuando una firma de método delegado tiene uno o más parámetros de tipos que derivan de los tipos de los parámetros de método, ese método se denomina contravariante*. Lo que viene a significar que los tipos de

datos que podemos usar como parámetros al llamar a un delegado pueden ser del mismo tipo que está definido en el delegado (así era hasta la versión 2.0), o de cualquier tipo derivado. Si el compilador ve una relación de herencia entre el tipo usado y el definido en el delegado, lo permitirá.

Siguiendo con el ejemplo del delegado que recibe un parámetro de tipo `Cliente`, (ver el fuente 1), vamos a rediseñar la clase y el método `MiFuncion` para que devuelva un saludo al nombre indicado en la propiedad `Nombre` de la clase. A continuación creamos la clase `ClienteOro` que se deriva de `Cliente`. Tanto en una como en otra clase hemos definido un constructor que recibe como parámetro el nombre a usar. En el fuente 7 vemos esas dos definiciones de estas clases.

```
public class Cliente
{
    // Omitida la definición de la propiedad Nombre
    public Cliente() { }
    public Cliente(string nombre)
    { this.Nombre = nombre; }
    //
    public virtual string MiFuncion(Cliente c) {
        return "Que tal " + c.Nombre;
    }
}

class ClienteOro : Cliente
{
    public ClienteOro() { }
    public ClienteOro(string nombre)
    { this.Nombre = nombre; }
}
```

Fuente 7. Definición simplificada de las clases `Cliente` y `ClienteOro`.

NOTA

El hecho de definir un constructor con parámetro en las clases del fuente 7 es para facilitar el uso de las mismas, de forma que en una sola instrucción podamos asignar el valor de la propiedad `Nombre`, de esa forma nuestro código de ejemplo podrá mostrar algo. Pero la razón de que lo hayamos tenido que hacer en las dos clases es porque los constructores no se heredan; por tanto, si queremos esa funcionalidad en las dos clases, debemos definirlos en ambas.

Ahora podemos usar cualquier función que reciba un delegado que apunte a `MiFuncion` y el compilador (realmente el *runtime*) usará la clase que corresponda.

```
ClienteOro co = new ClienteOro("Paco");
MiFuncionDelegado mfd2;
mfd2 = co.MiFuncion;
string sco = mfd2(co);
Console.WriteLine(sco);
```

Incluso podemos crear un método anónimo que use un objeto del tipo `ClienteOro` como parámetro,

pero siempre y cuando el método anónimo tenga la firma correcta, es decir, el parámetro en la definición del método anónimo debe ser del mismo tipo que el indicado en la definición del delegado, pero a la hora de usarlo podemos indicar cualquier objeto de un tipo `Cliente` o derivado:

```
MiFuncionDelegado mfd3;
mfd3 = delegate(Cliente co2) {return "A sus pies " +
                                co2.Nombre;};

Console.WriteLine(mfd3(co));
```

Pero el uso más práctico de esta característica será (como casi todo lo relacionado con los delegados), cuando lo apliquemos a los eventos.

Cuando utilizamos los métodos de eventos de los controles de Windows Forms, el segundo parámetro suele (o debería) ser una clase derivada de `EventArgs`, pero dependiendo del evento, ese parámetro será del tipo adecuado para el evento en cuestión. Por ejemplo, el evento `KeyPress` recibe un parámetro del tipo `KeyPressEventArgs`, pero en C# podemos definir el método que intercepta ese evento de cualquiera de estas dos formas:

```
private void txtNombre_KeyPress( object sender,
                                KeyPressEventArgs e)
{ ... }

private void txtNombre_KeyPress( object sender,
                                EventArgs e)
{ ... }
```

La segunda forma, a pesar de ser menos específica, seguramente la usaremos en casos muy concretos y siempre que necesitemos esa “generalidad”, pero no adelantemos acontecimientos, ya que cuando tratemos el tema de los eventos veremos algunas aplicaciones prácticas de esta posibilidad que tiene C#, ya que en Visual Basic siempre tendremos que definir los parámetros de los eventos del tipo exacto.

Conclusiones

Aún no hemos terminado con algunas de las cosas interesantes o importantes de los delegados, pero será en el próximo artículo donde veremos otra característica interesante de los delegados: la multidifusión. Esa forma de usar los delegados la comprenderemos mejor cuando sepamos más sobre la estrecha relación de estas clases especiales con los eventos.

Como es costumbre en esta sección, en el sitio Web de [dotNetManía](http://dotNetMania.com) está disponible el código de ejemplo para poder bajarlo, aunque en el caso de Visual Basic no será equivalente al de C#, simplemente porque aquél no soporta algunas de las características de los delegados que hemos tratado. ☺



Dino Esposito

Resolución de problemas en ASP.NET

Este mes respondemos algunas preguntas relacionadas con el desarrollo Web y ASP.NET en particular. Las cuestiones versan sobre importación de datos de recursos externos y sitios Web, tareas enlazadas de tipo asíncrono, y cómo mostrar imágenes almacenadas en bases de datos.

>> **Algunos sitios que** visito regularmente que muestran información actualizada desde otros sitios, por ejemplo noticias deportivas. Y no creo que ninguno de mis clientes realmente quiera enriquecer su sitio Web con tal tipo de información. No obstante, eso no es óbice para que piense que la importación de datos externos desde otros sitios es una característica poderosa y debiera considerarse...

Hay dos formas típicas de implementar una característica similar:

- Una es utilizando un servicio Web para obtener la información expuesta por el sitio. Esto significa, normalmente, que algún tipo de acuerdo ha sido suscrito entre ambas partes. Cuanto más valiosa sea la información, más posibilidades hay de que tengas que pagar por ella. Desde el punto de vista tecnológico, obtienes la información que llega del sitio (normalmente, cadenas), y la muestras donde te apetece.
- La segunda aproximación al problema supone la suscripción a una fuente RSS (sindicación). Un sitio de esa clase publica la información y la pone a disposición del público de forma gratuita. En este caso, se abre un *socket* de control del punto original del RSS, se lee la información —una cadena XML—, y se incorpora a las páginas propias.

En ambos casos, existe un acuerdo explícito o implícito entre el sitio Web y tú. Pero, para comentarlo todo, existe una tercera aproximación al problema: el típico *screen scraping* (arañar la pantalla). Nacido en los entornos de *mainframe* hace 25 años, ha vuelto a renacer con el advenimiento de la Web.

“ Si el sitio Web dispone de una oferta RSS, esa sería mi opción preferida. La técnica de *screen scraping* debería considerarse solamente como la última opción ”

Consiste en la capacidad de descargar la página completa y analizarla en busca de información de interés. Por ejemplo, podríamos descargar una página con resultados de la bolsa de valores, y configurarla para mostrar solamente los índices de nuestro interés. Después, se analizan todos los gráficos y el texto y se capturan los dos o tres elementos que buscamos.

Si el sitio Web dispone de una oferta RSS, esa sería mi opción preferida. La técnica de *screen scraping* debería considerarse solamente como la última opción. En la práctica, lo normal es utilizar herramientas como **RssBandit** o **SharpReader** para leer los fuentes RSS.

¿Cómo puedes leer una fuente RSS mediante programa e incorporarla a una página Web? La mayoría de los sitios Web (incluyendo MSDN) lo hacen mediante un control compuesto. El núcleo de este control contiene un código similar al del código fuente 1.

```

WebRequest req;
string rssData = String.Empty;
req = WebRequest.Create(RSSFEED);
using (WebResponse response = req.GetResponse())
{
    StreamReader reader;
    string text = String.Empty;
    using (reader = new
        StreamReader(response.GetResponseStream()))
    {
        text = reader.ReadToEnd();
    }

    // Process the RSS data
    rssData = ProcessFeed(text);
}

```

Fuente 1

La variable **Text** contiene la información en formato nativo XML. El contenedor **ProcessFeed** recorre los elementos XML y construye texto HTML publicable. Mediante otra opción más elegante, podríamos devolver un array de objetos personalizados desde **ProcessFeed** que solo almacene líneas individuales para transformarlas después en texto de marcado. El código del fuente 2 muestra cómo construir una lista de enlaces a las últimas noticias.

```

string ProcessFeed(string feed)
{
    StringBuilder sb = new StringBuilder();

    XmlDocument doc = new XmlDocument();
    doc.LoadXml(feed);

    XPathNavigator nav = doc.CreateNavigator();
    XPathNodeIterator iterator;

    // Title and description
    iterator = nav.Select("/rss/channel/title");
    iterator.MoveNext();
    sb.AppendFormat("<h3>{0}</h3>", iterator.Current.Value);
    iterator = nav.Select("/rss/channel/description");
    iterator.MoveNext();
    sb.AppendFormat("<i>{0}</i>", iterator.Current.Value);

    XmlNodeList items = doc.SelectNodes("/rss/channel/item");
    sb.Append("<ul>");
    foreach(XmlNode n in items)
    {
        XmlNode title = n.SelectSingleNode("title");
        XmlNode link = n.SelectSingleNode("link");
        XmlNode date = n.SelectSingleNode("pubDate");
        sb.AppendFormat("<li><a href=\"{0}\">{1}</a> on {2}</li>", link.InnerText, title.InnerText, date.InnerText);
    }
    sb.Append("</ul>");

    return sb.ToString();
}

```

Fuente 2

“ La necesidad de procesamiento asíncrono surge cuando el tiempo invertido en obtener los datos de entrada o salida resulta excesivo respecto al requerido para procesarlo ”

Y este código se puede incorporar fácilmente en un control compuesto hecho a partir de una tabla o una rejilla.

He construido varias soluciones utilizando tareas asíncronas, pero confieso que no presto mucha atención al término “vinculado a entrada/salida” (*I/O Bound*). ¿Se supone que es peligrosa cualquier operación asíncrona no estrictamente relacionada con actividad de entrada/salida?

Cualquier operación puede ser etiquetada de forma sencilla de dos maneras posibles: vinculada a la CPU o a entrada/salida. En el primer caso, estamos ante una operación cuya duración se determina por la velocidad del procesador y la cantidad de memoria disponible. La vinculación a entrada/salida indica la situación opuesta, cuando la CPU mayormente espera a que otros dispositivos terminen sus tareas.

Ahora bien, ¿cuál es el propósito de las tareas asíncronas? Normalmente, resolver situaciones en las que la IU debe de tener un alto nivel de respuesta, mientras espera que una tarea en *background*, potencialmente larga, concluya. Desde esta perspectiva, no hay diferencias relevantes entre los dos tipos de operaciones. Esto es correcto cuando estamos considerando escenarios con clientes ricos, tales como las aplicaciones Windows Forms. Para aplicaciones ASP.NET, las cosas son algo distintas, pero la conclusión viene a ser que si el código ha sido escrito para Windows, probablemente servirá también en este caso.

¿Qué operaciones son candidatas a ser implementadas mediante páginas asíncronas? O, dicho de otra forma, ¿qué operaciones fuerzan o sugieren la adopción de una página asíncrona en una aplicación ASP.NET? Vamos a formalizar someramente el concepto de proceso asíncrono en el contexto de ASP.NET.

La necesidad de procesamiento asíncrono surge cuando el tiempo invertido en obtener los datos

de entrada o salida resulta excesivo respecto al requerido para procesarlo. En tales situaciones, la CPU se encuentra en estado de espera, o de infrutilización, esperando la mayor del tiempo que algo suceda. En particular, las operaciones vinculadas a entrada/salida en el contexto de ASP.NET son todavía más dañinas, porque los procesos del servidor pueden bloquearse y el *pool* de subprocesos del servidor es un recurso crítico y limitado.

Nos damos cuenta de las ventajas reales cuando usamos el modelo asíncrono en las operaciones vinculadas a entradas/salidas. En la implementación de .NET Framework, cuando se realiza una llamada asíncrona, ningún subproceso se bloquea mientras la operación está pendiente.

Ejemplos típicos de este tipo de operaciones son las que requieren acceso a algún tipo de recurso remoto o interacción con dispositivos de hardware externos. Procesos con bases de datos remotas o llamadas a servicios Web no locales, son ejemplos comunes de esta clase de operaciones, para las que debería considerarse seriamente la construcción de páginas asíncronas.

En ASP.NET, las operaciones de larga duración asociadas a la CPU (por ejemplo, algoritmos complejos que actúen sobre datos en memoria), también deberían implementarse de forma asíncrona para preservar los recursos disponibles en el servidor en un nivel suficientemente alto. De todas formas, en este escenario, la CPU está siempre cargada con algún trabajo y la ganancia es considerablemente menor.

En resumen, cualquier tarea pesada debería programarse de forma asíncrona, tanto en Windows como en ASP.NET. Pero en este caso, resulta especialmente importante para las operaciones vinculadas a procesos de entrada/salida. Si estas operaciones (como llamar a un servicio Web remoto), se implementan de forma asíncrona, la CPU del servidor Web no se ve afectada, y puede procesar otras peticiones con rapidez.

Para operaciones vinculadas a la CPU, los procesos asíncronos también resultan en una ligera mejora de las aplicaciones Windows a causa de su ejecución en una hebra separada, si bien para el caso de aplicaciones ASP.NET no deberían esperarse grandes mejoras. Todavía es mejor el proceso asíncrono, pero las páginas se generarán a la velocidad que la CPU sea capaz de procesarlas. Quizá, un mejor abordaje de esta situación pasaría por separar esas operaciones de larga duración a otro servidor especializado y transformar las tareas vinculadas a la CPU en tareas vinculadas a procesos de entrada/salida.

En los primeros días de ASP.NET 2.0 –he estado trabajando con él desde las primeras versiones alfa, en verano 2003– era muy fácil hacer referen-

“ En ASP.NET 2.0 debes crear un manejador similar y diseñarlo para que acepte en la cadena de consulta cualquier información que pueda ser útil para identificar la imagen y su origen ”

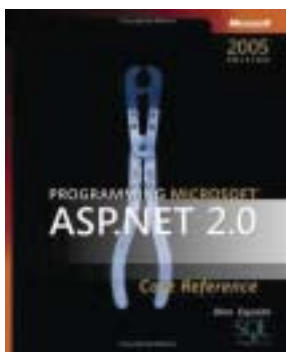
cia a imágenes almacenadas en una base de datos SQL Server usando un control a medida llamado `<asp:DynamicImage>`. El control fue eliminado de las versiones posteriores y también de la versión final. Sin embargo, sigo teniendo el mismo problema que me llevó a probar esa versión. ¿Cómo creo mi propio control para carga dinámica de imágenes en ASP.NET 2.0?

En ASP.NET 2.0 existen dos controles de servidor para mostrar imágenes: `Image` y `HTMLImage`. Ambos suministran una fina capa de abstracción sobre algunas de las propiedades del elemento `` de HTML. Al final, ambos permiten, simplemente, referenciar imágenes por URL y establecer algunos atributos de presentación.

Como indicas, en el mundo real las imágenes pueden tener orígenes muy diversos. Pueden estar en ficheros o bases de datos, o ser generadas dinámicamente en memoria utilizando usando una API gráfica. En el primer caso, la solución es directa a través de la etiqueta `` y la API de ASP.NET. En el resto, los programadores tienen que escribir su propio código para hacerlo accesible vía URL. En ASP.NET, las URL solo pueden apuntar a un manejador personalizado (`HTTPHandler`). El control `DynamicImage` se basaba en uno de estos manejadores –`cacheimageservice.axd`– que se ha eliminado en la versión final.

En ASP.NET 2.0 debes crear un manejador similar y diseñarlo para que acepte en la cadena de consulta cualquier información que pueda ser útil para identificar la imagen y su origen. El control `DynamicImage` eliminado encapsulaba buena parte de esa información (cadenas de conexión, consultas, ID, etc.), ocultando su complejidad a los desarrolladores.

El manejador HTTP deberá ejecutar cualquier código necesario para leer la imagen y establecer el tipo MIME adecuado, devolviendo sus bytes. Para evitar consultas repetidas y ahorrar algunos ciclos de CPU, el manejador puede almacenar imágenes en caché y recuperarlas posteriormente. ☺



Programming ASP.NET 2.0 Core Reference

Dino Esposito

Editorial: Microsoft Press

ISBN: 0735621764

Páginas: 800

Primera edición: 2006

Idioma: Inglés

Ya habíamos comentado en esta sección la obra anterior de nuestro colaborador **Dino Esposito** y su buena acogida entre los lectores. En esta ocasión, Dino hace una cobertura completa de los entresijos de ASP.NET sin dejar cabo suelto, en un volumen extenso y con abundantes ejemplos.

Asumiendo que el lector es conocedor de las técnicas de ASP.NET en general, aborda todos los temas desde el punto de vista del desarrollador –que él también es– dando soluciones concretas a problemas concretos, recorriendo desde el modelo de compilación y la construcción de elementos personalizados, hasta la generación de controles y los consejos necesarios para su óptimo diseño y funcionamiento. Como dice su título: una referencia fundamental.

Applications = Code + Markup:

A Guide to the Microsoft Windows Presentation Foundation

Charles Petzold

Editorial: Microsoft Press

ISBN: 0735619573

Páginas: 1.020

Primera edición: 2006

Idioma: Inglés



Sabíamos por nuestro admirado **Jeff Prosise** que **Charles Petzold** estaba escribiendo un par de libros de envergadura. Este es el primero, dedicado a la programación para Windows Vista, dentro del marco WPF (antes llamado Avalon), la API gráfica que dará soporte al nuevo sistema operativo. No es una introducción porque su longitud hace que se presente a los lectores como una referencia de todo lo que es posible hacer en el contexto de esa nueva interfaz de usuario.

Los lectores que lo hemos tenido en nuestras manos, sentimos (otros comentarios también lo avalan así) que nos encontramos ante un nuevo mundo dentro del desarrollo, de una complejidad enorme y también de enormes posibilidades. Menos mal que, como ya ha hecho otras veces, el maestro Petzold vuelve a presentar lo difícil como si fuera evidente.



Pablo Tilotta

10 Años de “El Guille”

En este artículo rendimos un sencillo pero merecido homenaje a “El Guille” por el décimo aniversario de su sitio ElGuille.info, posiblemente el sitio más conocido por los desarrolladores de Visual Basic del mundo entero. Le saludan sus amigos de Microsoft en España y Latinoamérica, de Solid Quality Learning, de INETA y, cómo no, de esta revista desde la que les martiriza cada mes.

» En el mundo de las estadísticas y los números, no creo que ningún programador tenga en claro cuántos sitios Web sobre programación hay en Internet. Solo los mejores permanecen en la mente de miles de desarrolladores.

La competencia es feroz, las luchas por el posicionamiento son insalubres. Pero dentro de los miles y miles de sitios sobre programación que existen, solo uno está al margen de todo eso; un sitio que no necesita de presentaciones, de posicionamientos, no necesita emprender luchas para captar usuarios; un sitio que estuvo ahí cuando lo hemos necesitado, cada día desde hace 10 años atrás (cuando casi nadie se animaba a montar un sitio así); un sitio que podría estar inundado de publicidad y ganar así muchísimo dinero, sin embargo su *webmaster* quiere otro tipo de calidad para sus usuarios, a costa de no ganar tanto como merece... Ojalá que la venta de sus dos libros solucionen esa injusticia.

Desde la época del *dial up* hasta la navegación ADSL, quienes teníamos dudas de Clipper, Visual Basic 3, 4, 5, 6 y -desde hace unos años- .NET, sabíamos que con concurrir a www.elguille.info saciaríamos nuestra sed de conocimientos.

Me refiero al famoso sitio de **El Guille (Guillermo Som Cerezo)** quien además de todo eso, nos daba y nos da lecciones de sencillez, de calidez y de ética, tan solo con su ejemplo.

Sabemos que “El Guille no lo sabe todo” y que su día tiene 24 horas igual que el nuestro, pero sin embargo se toma el trabajo de contestar los correos electrónicos, de dar la cara, de recorrer el mundo evangelizando a quienes prefieren los “;” del C# para que se vuelquen a Visual Basic. Y sobre todo para regalar su don de gente.

Es casi imposible no apreciarlo. Una de mis mayores satisfacciones en mis años de *webmaster* fue lograr su amistad y confianza.

Creo que este es un buen momento para que quienes aprecian y admiran a El Guille puedan hacerlo. Es por eso que esta nota estará conformada por las vivencias, opiniones, saludos de mucha gente que quiere rendirle un sincero y sentido homenaje.

¡¡¡ENHORABUENA GUILLE !!!



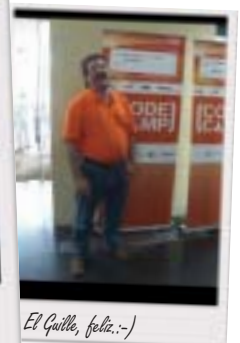
“El Guille es un tipo especial, desde luego. Lleva diez años con el sitio Web -posiblemente- más famoso de habla hispana dedicado a la programación, allá por donde va, siempre hay alguien que quiere hacerse una foto con él, ha recorrido medio mundo divalgando tecnologías de Microsoft..., y sigue siendo tan humilde y sencillo como siempre. Cuántos jovenzuelos me he encontrado, amigo

Guillermo, que con mucho menos hay que hablarles de asted. Gracias por ser como eres y te deseo de corazón muchos años más dando la lata como solo tú sabes :-). Aquí les dejo unas fotografías de mi álbum”.

Paco Marín, editor de dotNetManía.



Guille posando con uno de sus múltiples fans en el pasado CodeCamp de Vio, en España.



El Guille, feliz:-)

Pablo Tilotta

es Analista de Sistemas desde hace casi 20 años.

Luego de ser usuario de varios foros y disconforme con la falta de compromiso de muchos sitios, crea en 2003

VB-MUNDO (www.vb-mundo.com), un portal profesional para desarrolladores.

Microsoft



"Desde Microsoft Ibérica y especialmente desde el área de desarrolladores y MSDN es un placer haber contado con la amistad y el buen hacer de Guille durante estos 10 últimos años. Muchas gracias Guille por todo lo que nos has enseñado y por tu dedicación a la comunidad de desarrolladores sin descanso. Un abrazo" (Alfonso Rodríguez, Responsable de programas para desarrolladores en Microsoft Ibérica).



"Desde Microsoft Cono Sur, felicitamos al Guille no solo por sus 10 años, sino también por sus infinitas ganas de compartir su conocimiento. Realmente es increíble y admirable su voluntad. Te enviamos desde aquí un gran saludo, todos lo que hacemos MSDN Cono Sur." (María Fernanda Pérez Marino, Gerente de comunidades MSDN - Microsoft Cono Sur).



"Yo aún no había nacido y El Guille ya era MVP, el primero en España, allá por el 97. Con él descubrí el significado de MVP: no sé qué hacía yo por el stand de "Ask the experts" en el TechEd 2003, pero recuerdo que se me acercó un muchacho muy emocionado a preguntarme: "¿está el Guille?, ¡tengo que conocerlo como sea!, ¡he aprendido tanto en su Web!... ¡Muchas gracias Guille!, deja el Visual Basic que ya te lo sabes y vente al Gündous Zerver :-)" (Alberto Amescua, TechNet España).

La otra cara de El Guille



Si bien el Guille es muy conocido por sus charlas, artículos, libros y su sitio Web, tuve la suerte de compartir con él muchos viajes por Latinoamérica. No es que cambie mucho... A ver cómo les explico, sigue siendo muy simpático y amable pero, por ejemplo, no comer carne en Argentina porque no es finita como la que come en su casa y muy pero muy cocida, ¡¡¡es increíble!!! Sí, de verdad, cuando nos visitó se la pasó comiendo pan y verduras.

Otra... En la región andina se la pasaba pidiendo bocadillos o pan con tomate y aceite de oliva... ya le dije que para la próxima se traiga su aceite.

No se imaginan en Perú, ya ni recuerdo en qué ciudad, lo que fue convencer a la persona que prepara el desayuno que por favor le consiguiera pan tostado (porque OJO, si no es pan de panadería no come), que corte un tomate, lo frote sobre el pan y que le traiga aceite de oliva... ¡¡¡eso para el desayuno!!! Por suerte tomó café y el famoso zumo (se conformó con zumos de frutas locales que no existen en España).

Ahora hablando de informática, si por casualidad te has olvidado algún cable, o algo para tu computadora, El Guille abre su maleta y ¡ahí lo tiene! Te salva siempre :-D.

Y lo que no se imaginan es al Guille en el gran país del norte, ¡hablando en inglés! Y aquí la cosa se agudiza, ya que en las reuniones anuales de los MVP en Seattle, hemos tenido la dicha de encontrarnos muchos de habla de hispana y bueno, no hay quien se salve.

Pero nunca lo vi bailar o cantar, ¿por qué será?

Realmente para mí ha sido un lujo poder compartir con él todas estas experiencias, siempre haciendo chistes y contento, no tengo más que palabras de agradecimiento. Y claro, ahora me doy cuenta, como cumple 10 años, es como un niño y dentro de INETA me dicen que soy como la mamá de todos, ¿será por eso que me pone tantos problemas con la comida?

Y ahora les dejo unas fotos de nuestros viajes, donde podrán ver la otra cara del Guille... :-D ¡¡¡Feliz Cumple Guille!!!

Nilda Beatriz Díaz, INETA Latam - Comité Web (www.inetalatam.org), Gerente de Proyectos Latinoamérica. Desde el tercer planeta a partir del Sol.



En Ica-Perú, el Guille en estado puro



Aquí lo pueden ver subiendo una duna, si ese que está de espaldas es él, en Ica-Perú



En el aeropuerto de Guayaquil, colocándole gotas en los ojos a Pablo Norvárez



La cuestión es llevar Visual Basic al último rincón del planeta :-)



El Guille, justo antes de ser rescatado (es broma)



"Un recuerdo de un fabuloso día en el Río Pacuaré en Costa Rica. El Guille, José Berríos (INETA LATAM) y Adolfo Wiernik (Solid Quality Learning) mostrando sus musculaturas y habilidades para el deporte. ¡Felicitaciones Guille!, Un abrazo"

(Adolfo Wiernik, mentor y director de operaciones de Solid Quality Learning en América Latina.





Octavio Hernández

Chart FX para Visual Studio 2005

Este mes analizamos Chart FX para Visual Studio 2005, una de las herramientas líderes del mercado para la integración de gráficos comerciales y científicos en las aplicaciones para .NET Framework.

- » Si necesita incorporar gráficos comerciales o científicos de alta calidad a sus aplicaciones Windows o Web basadas en .NET Framework, Chart FX para Visual Studio 2005 puede ser una excelente opción. Un producto de amplia solera (el fabricante, Software FX, produce componentes gráficos para las principales herramientas de desarrollo desde 1993), este paquete permite crear de una manera sencilla y consistente más de 100 tipos diferentes de gráficos basados en datos para integrar en las aplicaciones .NET. En este artículo ofrecemos un primer acercamiento a esta excelente librería.

Primera impresión

Instalación

Gracias a las bondades incorporadas por Microsoft a la plataforma .NET, raro es el caso en el que hay que hacer algún señalamiento especial con relación a los procesos de instalación o despliegue de librerías de código manejado. La instalación de Chart FX para Visual Studio 2005 no es una excepción, y basta con pulsar unas cuantas veces el botón “Siguiente” para lograr la instalación de las librerías que componen el producto, la combinación de los archivos de ayuda con la colección de ayudas MSDN instaladas, y la instalación y el registro de los asistentes específicos que ofrece la herramienta como parte de su integración en el entorno de desarrollo de Visual Studio 2005.

Es de destacar que al finalizar la instalación, se lanza automáticamente el *Samples & Resource Center* (Centro de Ejemplos y Recursos), una aplicación de escritorio con conexión a servicios Web del fabricante para suministrar todo tipo de información de referencia y ejemplos de utilización del producto. Desde esta interfaz también es posible obtener soporte técnico, conocer la disponibilidad de nuevas actualizaciones o conectarse a los foros relacionados con el producto y mantenidos por Software FX.



Documentación y ejemplos

La documentación de iniciación y referencia que acompaña al producto es excelente. Eso sí, debemos señalar que se encuentra en perfecto inglés; pero eso es algo a lo que los desarrolladores de habla hispana nos hemos ya acostumbrado. Tanto la Guía del Programador como la Referencia de la API pueden consultarse cómodamente desde el Centro (filtrándolas tanto por lenguaje - C# o VB – como por tecnología– Windows Forms o ASP.NET), así como desde dentro de la ayuda en línea de MSDN.

En cuanto a los ejemplos, hay que decir que el paquete también brilla en este aspecto; incluye un total de 40 ejemplos completos, divididos más o menos equitativamente entre aplicaciones Windows y Web; más ejemplos pueden descargarse del sitio Web de Software FX.

Utilizando el producto

Una primera aplicación

Para comenzar a trabajar en modo visual con los componentes que integran Chart FX, es conveniente personalizar el Cuadro de herramientas de Visual Studio, insertando los componentes contenidos en los ensamblados **ChartFX.WinForms.dll** y **ChartFX.WebForms.dll**. A partir

de ese momento, se podrá arrastrar esos componentes para soltarlos sobre los formularios de nuestras aplicaciones.

La vía más rápida y simple para familiarizarse con Chart FX es insertar un gráfico en un formulario Windows o Web. Las tareas asociadas al componente cubren todas las necesidades básicas de configuración del gráfico. Adicionalmente, a través de la Ventana de propiedades de Visual Studio es posible asignar valores a todas las propiedades de tiempo de diseño que ofrece el componente. Y finalmente, mediante código se puede acceder a todo el extenso modelo de objetos de Chart FX utilizando cualquiera de los lenguajes .NET.

Por supuesto, los componentes soportan perfectamente las técnicas de enlace a datos que proponen .NET Framework 2.0 y Visual Studio 2005.

Formatos de salida en las aplicaciones Web

Un elemento a destacar en la funcionalidad que ofrece Chart FX para el desarrollo Web es la amplia variedad de formatos de salida soportados. Estos se presentan en la tabla 1.

Despliegue de aplicaciones y temas relacionados

Despliegue de las aplicaciones

Como ya hemos mencionado antes, tampoco hay mucho que decir en lo relativo al despliegue de las aplicaciones que utilicen ChartFX; tratándose de una librería desarrollada al 100% utilizando código manejado, el despliegue es cuestión simplemente de copiar los ensamblados correspondientes al directorio de destino, en los casos de despliegue manual, o incluirlos en el programa de instalación para implantarlos en el equipo de destino.

Con respecto a las licencias, hay que decir que el despliegue de los ensamblados redistribuibles de ChartFX es totalmente libre de *royalties* en el caso de aplicaciones Windows. En el caso de las aplicaciones Web, una licencia de desarrollo permite el despliegue sobre un único servidor Web, no importa la cantidad de procesadores que tenga; se deberán adquirir licencias adicionales para instalar los componentes Web en un segundo y sucesivos servidores Web.

Internacionalización

Otra de las ventajas importantes de ChartFX está perfectamente globalizado, tanto en el caso de los ensamblados para Windows como para la Web. El fabricante ofrece ensamblados satélites para numerosos idiomas (incluido el castellano), y adicionalmente ofrece toda la información técnica necesaria para que sea posible crear nuevos recursos localizados.

Otros productos relacionados

Extensiones de ChartFX

Los creadores de ChartFX han dotado al producto de una arquitectura abierta y adaptable, de modo que sea relativamente sencillo extender las librerías básicas con nuevas funcionalidades. La propia empresa ofrece diferentes extensiones, algunas de las cuales son gratuitas, mientras que otras no. La tabla 2 enumera las principales.

Chart FX Gauges

Chart FX Gauges es una librería de controles que simulan paneles digitales, lineales, radiales, etc. y que ofrecen la posibilidad de presentar en tiempo real datos obtenidos de manera dinámica. También están disponibles para Windows y la Web, utilizando en ambos casos una misma API.

Conclusiones

Es evidente que ChartFX para Visual Studio 2005 es un producto sólido, en cuyo desarrollo se han cuidado hasta los más mínimos detalles para garantizar una experiencia de usuario satisfactoria inclu-

Formato	Explicación
.NET	Permite al usuario final un alto nivel de interactividad con relación al gráfico. El cliente debe utilizar Internet Explorer y tener .NET Framework instalado y configurado.
PNG	Formato de mapa de bits portable y con muchas ventajas sobre GIF, es el formato de salida por defecto recomendado por Software FX.
JPG	Formato fotográfico que no ofrece en general buenos resultados al mostrar gráficos comerciales. Generado por defecto cuando el cliente Web no soporta PNG.
SVG	Formato vectorial (<i>Scalable Vector Graphics</i>). Requiere soporte adecuado en el cliente, pues los gráficos se dibujan en él. Muy eficiente gracias al pequeño tamaño de los ficheros.
Flash	Formato muy popular; requiere que el reproductor de Flash esté instalado en el cliente.
DHTML	HTML dinámico, que utiliza scripts para los menús, etc.
Accesibilidad	Presenta los resultados en forma textual, para que puedan ser leídos por un lector de pantalla a personas disminuidas.

Tabla 1. Formatos de salida que ofrece ChartFX para aplicaciones Web.

Extensión	Explicación
Chart FX Annotations	Extensión que hace posible incorporar anotaciones de diferentes tipos a los gráficos.
Chart FX OLAP	Ofrece elementos de interfaz de usuario orientados específicamente a las operaciones de pivotado, corte y profundización típicas de los sistemas de <i>Business Intelligence</i> .
Chart FX Financial	Permite ejecutar análisis técnicos de datos financieros y presentarlos utilizando tipos de gráficos especiales para el análisis financiero.
Chart FX Maps	Librería de mapas dinámicos que pueden integrarse con Chart FX para mostrar los valores de un gráfico sobre un mapa.
Chart FX Statistical	Amplía la potencia de Chart FX añadiendo posibilidades adicionales de análisis estadístico.

Tabla 2 Extensiones de ChartFX.

so al programador más exigente. Esperamos mediante estas primeras impresiones haber despertado el interés del lector, que tiene en sus manos la posibilidad de evaluar esta excelente herramienta (junto con otras también mencionadas en este artículo) gracias al CD que acompaña este mes a la revista. ●

Ficha técnica	
Nombre	Chart FX
Versión	Chart FX for Visual Studio® 2005
Fabricante	Software FX
Web	www.softwarefx.com
Distribuidor	ABOX (www.abox.com)
Precio	Desde 799\$
Valoración	●●●●●

SWREUSER



Una herramienta CASE pensada por y para diseñadores .Net
Con soporte a la reutilización de software (metodología IRM)
Con cobertura completa del ciclo de vida: requisitos, riesgos,
modelado UML, generación de código, pruebas,
trabajo colaborativo, trazabilidad total...



It's your knowledge, reuse it!

Descargue éste y otros productos orientados a reutilización de <http://www.reusecompany.com>

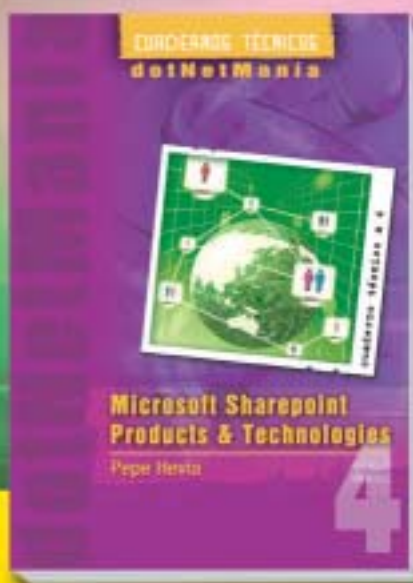
Y por ser lector de dotNetManía, consiga un 15% de descuento.

Envíe un e-mail a contact@reusecompany.com con el asunto "Compra swREUSER: dotNetManía"

CUADERNOS TÉCNICOS

dotNetManía

Microsoft Sharepoint Products & Technologies



PVP.: 24,50 €
20% descuento
para suscriptores

Información:

www.dotnetmania.com

CUADERNOS TÉCNICOS

dotNetManía

Diseño de informes con SQL Server Reporting Services



PVP.: 24,50 €
20% descuento
para suscriptores

Información:

www.dotnetmania.com



Serás el primero en certificarte en MCTS

Microsoft Certified Technology Specialist

Ante la llegada de las **Nuevas Certificaciones Microsoft** para Profesionales TIC, Alhambra-Eidos, dentro de su programa de innovación, presenta una serie de **Cursos Intensivos de Verano**, para obtener las Nuevas Certificaciones MCTS actualmente disponibles: **MCTS en Web Applications**, **MCTS en Windows Applications** y **MCTS en SQL Server 2005**.

Ref. CIV 101 (*)	Fundamentos de Lenguaje y Microsoft Framework 2.0, preparatorio para el examen 70-536 de Microsoft	9 - 24 octubre 2006
Ref. CIV 102 (*)	Workshop ADO .NET (2541 y 2542)	25 oct. - 13 nov. 2006
Ref. CIV 103	MCTS en WEB APPLICATIONS: Workshop 2543 y 2544	14 - 27 noviembre 2006
Ref. CIV 104	MCTS en WINDOWS APPLICATIONS: Workshop 2546 y 2547	28 nov. - 14 dic. 2006
Ref. CIV 105	MCTS en SQL Server 2005 Transact + MOC 2779 y 2780	16 - 31 octubre 2006 4 - 21 diciembre 2006 (segunda convocatoria)

Cada Curso dura 25 horas, repartidas de Lunes a Viernes de 16:00 a 21:00h. Todos los Cursos incluyen en el precio Manuales Oficiales y Examen correspondiente. 5% de descuento sobre el precio final para los lectores de dotNetMania.

(*) Estos Cursos Intensivos son necesarios para ambas Certificaciones

Curso de 32 horas, impartidos de lunes a jueves de 16:00 a 20:00 h. Se incluyen en el precio los Manuales Oficiales, ampliado en Transact SQL por Alhambra-Eidos, así como el examen correspondiente. 5% de descuento sobre el precio final para los lectores de dotNetMania.

más información en: www.alhambra-eidos.es/MCTS



Custom Development Solutions
Infrastructure Solutions
Learning Solutions
Security Solutions



La manera más rápida
de desarrollar aplicaciones Web.



Llámenos ahora y consiga la oferta exclusiva
para lectores de DotnetMania



C/Manso 26-28, 2a planta - 08015 Barcelona
Telf.: 934 262 257 - Fax: 934 231 140
E-mail: abox@abox.com - Web: <http://www.abox.com>



Marino Posadas

Expectación ante la posible presentación de Windows Vista en el Tech-Ed '06 de Barcelona



Considerando que la compañía no ha modificado su calendario de salida del producto, la disponibilidad de la RC1 y algunos otros factores, casi todas las voces de la red se hacen eco de la más que posible salida del producto para ISV coincidiendo con el evento de Barcelona.

De hecho, los contenidos de Tech-Ed Developers van a centrarse en la API de Vista (.NET Framework 3.0), según se desprende de la Agenda Oficial, y sus ponentes: lo mejor de lo mejor.

Desde el propio arquitecto principal de .NET (y autor del lenguaje C#), **A. Hejlsberg** (que tiene 4 presentaciones programadas, sobre C# y direcciones futuras del lenguaje), pasando por **Ivar Jakobson** (co-autor del estándar UML, que hablará sobre Team System), **Kimberly Tripp** (SQL Server 2005), **Bob Beauchemin** (SQL Server 2005), **David Chappell** (*Business Process Management*), **Roy Oshero** (que hablará sobre metodologías ágiles y Team System, y cuyo blog recomendábamos en el número anterior), **Fritz Onion** (ASP.NET 2.0, pero con una sesión sobre .NET 3.0), **Ted Pattison** (SharePoint), **Ingo**

Rammer (Windows Workflow Foundation), **Jeff Prosise** (AJAX y Atlas), **Steve Swartz** (extensísima intervención, junto a **Clemens Vasters** con 6 presentaciones fragmentadas sobre Microsoft Application Server Platform), y un larguísimo etcétera que daría para varias páginas. Todos bajo la tutela del presentador oficial de la *keynote*, **Eric Rudder**, Senior Vice President de Estrategia Tecnológica de la compañía, quien definirá el año venidero como “el más importante y prolífico en la historia de Microsoft”.

Presentado un chip que transfiere datos mediante láser



Científicos de la Universidad de California e Intel Corporation, han presentado un nuevo modelo de chip capaz de transferir información mediante un haz láser, generado por el propio chip, eliminando de esta forma la necesidad de cableado en la transferencia. El avance es de una trascendencia notable para **John E. Bowers**, director del Centro de Óptica Multidisciplinar de la citada universidad, quien reconoce que su disponibilidad deberá esperar dos o tres años, pero que estará presente en todas partes, incluidas las conexiones a domicilio mediante fibra.

Documentos en la Red

Practical Examples of Namespaces in VB.NET 2005: Un práctico dossier sobre la herencia en la nueva versión de VB, que forma parte de una serie de documentos gratuitos publicados en *ASPFree.com* (<http://www.aspfree.com/c/a/VB.NET/Practical-Examples-of-Namespaces-in-VBNET-2005>)



Windows Scripting: Ten Windows XP Performance Tweaks: Y desde la misma fuente, también tenemos este documento especialmente dedicado a los muchos que hemos actualizado nuestro hardware recientemente, con diez recomendaciones para la mejora (notable) en el rendimiento (<http://www.aspfree.com/c/a/Windows-Scripting/Ten-Windows-XP-Performance-Tweaks>).

Blogs del mes



Jusep.NET: Sitio en castellano dedicado principalmente al desarrollo mediante XAML y .NET Framework 3.0. También dispone de noticiarios, pero destacamos sobre todo el código fuente. Accesible en <http://jusep-net.blogspot.com>.



“A la Vista”, Blog de Catherine Heller, evangelista técnica ubicada en Redmond, y bien conocida en España, donde residió muchos años. Especialmente dedicado a la programación con Windows Vista. Accesible en <http://blogs.msdn.com/cheller>.

Utilidades del mes



Dimio's Utilities: Se trata de un paquete de utilidades *freeware*, disponibles en el sitio Web de Dimio (<http://dimio.altervista.org/eng>) donde podemos encontrar desde herramientas que permiten guardar el MBR del sistema (*Master Boot Record*), hasta sincronizadores de dispositivos, gestores programados de actividad y muchos otros.



XNA Studio Express Beta 1: Aunque no se trata de una utilidad, sino de una herramienta de desarrollo, no podemos dejar de mencionar su disponibilidad en el sitio de descargas <http://msdn.microsoft.com/directx/xna/gse>. Para más datos sobre su funcionamiento es recomendable visitar la página oficial (<http://msdn.microsoft.com/directx/XNA>), si bien anticipamos que un requisito imprescindible es instalarse Visual Studio C# Express Edition (gratuita). Además, en la página de *Microsoft Game Developer presentations* (<http://msdn.microsoft.com/directx/presentations>), pueden encontrarse ya abundantes materiales expuestos en el evento Gamefest 2006.

desarrolla software

será la **primera** en alcanzar
el nivel más alto en la **carrera**
de desarrolladores

Convierte en pionero en la nueva carrera para desarrolladores Microsoft: más dinero, mejora profesional. ¡Te capacitamos para que lo consigas!



Conviértete en uno de los primeros profesionales con la más alta certificación de Microsoft, en la nueva carrera oficial para desarrolladores. Insíbete en la XIII Edición del Master Alhambra-Eidos en Desarrollo de Software.

MCPD Enterprise Application Developer

Con motivo del lanzamiento de las nuevas Tecnologías de Microsoft y las nuevas certificaciones que propone el fabricante, Alhambra-Eidos lanza la XIII Edición de su Master en Desarrollo de Software, adecuando su finalidad a la obtención de la certificación de más alto nivel para los desarrolladores: MCPD Enterprise Applications Developer (Microsoft Certified Professional Developer).

Formato Blended, al estilo Executive

Para complementar a la perfección tu actividad diaria con este programa formativo, hemos preparado la XIII Edición del Master para impartirlo en formato **BLENDED**, al modo de los programas Executive, siendo el 40 por ciento de las horas presenciales de clase, y el 60 por ciento restante de autoestudio, orientación al caso, y todo ello con la tutorización de la mano de los mejores profesionales.

Nuevas Tecnologías Microsoft:

Microsoft
SQL Server 2005

Microsoft
Visual Studio 2005

Microsoft
BizTalk Server 2006

Microsoft
GOLD CERTIFIED
Partner

Custom Development Solutions
Information Worker Solutions
Learning Solutions
Security Solutions

Comienzo viernes 3 de Noviembre de 2006

Tu potencial, nuestra pasión.™
Microsoft



Nuevo Visual Studio 2005. La diferencia es obvia.

¿Encuentras la diferencia? La verás desde la primera línea de código. Visual Studio® 2005 incluye más de 400 nuevas funcionalidades que te ayudarán a crear aplicaciones escribiendo menos código, a reducir el tiempo de depuración y pruebas, y a acelerar el desarrollo sobre .NET. Así consigues lo que realmente importa. Encuentra las 10 diferencias y busca las restantes en www.microsoft.es/vstudio

Microsoft
Visual Studio 2005

© 2005 Microsoft Corporation. Reservados todos los derechos. Microsoft, Visual Studio, el logotipo de Visual Studio y "Tu potencial, nuestra pasión" son marcas registradas o marcas de comercio de Microsoft Corporation en Estados Unidos y/o otros países.