

# dotNetManía

[www.dotnetmania.com](http://www.dotnetmania.com)

Dedicada a los profesionales de la plataforma .NET

## SQL Server 2005

Una nueva generación de sistemas de gestión de la información

### ENTREVISTAS



**Paul Flessner**  
Vicepresidente Senior de Aplicaciones de Servidor  
Microsoft Corporation



**César Galindo-Legaria**  
Manager del SQL Server Query Optimizer Group  
Microsoft Corporation

Alta disponibilidad  
Integración con el CLR  
Business Intelligence  
Soporte XML  
Transact-SQL 2005  
Service Broker  
SMO

### Laboratorio

SQL Server 2005 Mobile Edition  
SQL Server 2005 Express Edition

### Comunidad.net

GUSENet  
Grupo de Usuarios del SurEste

### TodotNet Q&A

SQL Server 2005

### Opinión

SQL Server 2005: Mirando hacia el desarrollador



**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.

**msdn**  
Microsoft® Developer Network

Distribuidores  
Recomendados:



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



93 540 75 01  
[www.rambalief.com](http://www.rambalief.com)



902 11 57 98  
[www.visualnet.net](http://www.visualnet.net)

# dotNetManía

Dedicada a los profesionales de la plataforma .NET

Vol. II • Número 17 • Julio/Agosto 2005

Precio: 6€ (España)

## 7-N

### Editor

Paco Marín  
(paco.marin@dotnetmania.com)

### Administración

Pilar Pérez  
(pilar.perez@dotnetmania.com)

### Asesor Técnico

Marino Posadas  
(marino.posadas@dotnetmania.com)

### Redactores

Antonio Quirós, Dino Esposito, Guillermo 'guille' Som, Jorge Serrano, José Manuel Alarcón, Luis Miguel Blanco, Miguel Katrib (Grupo Weboo) y Pedro Pozo.

### Colaboradores habituales

Ángel Esteban, Braulio Díez, Eladio Rincón, Erich Bühler, Fernando Noguera, Jorge Crespo Cano, José Miguel Torres, Miguel Egea, Octavio Hernández, Pablo Abbate, Pepe Hevia, Rodrigo Corral y Salvador Ramos.

### Además colaboran en este número

Fernando Bocigas y Fernando Guerrero

### Edición y Suscripciones

.netalia

c/ Robledal, 135

28529 Rivas-Vaciamadrid (Madrid)

Tf. (34) 91 666 74 77

Fax (34) 91 499 13 64

### Publicidad

Mediadev

Sophie Mancini (sophie@mediadev.es)

Tf. 93 426 22 57 - 670 99 74 64

Fax. 93 423 11 40

### Imprime

Gráficas Vallehermoso

www.graficasvallehermoso.com

ISSN

1698-5451

Depósito Legal

M-3.075-2004



En realidad yo creo que esto debió ocurrir hace un par de años, en el 2003. Cuántas veces los responsables de Microsoft habrán oído la misma pregunta: “¿...y no es mucho cinco años entre versiones?”. Por fin, la fecha de lanzamiento de SQL Server 2005 y Visual Studio 2005 será la semana del 7 de noviembre, según anunció **Paul Flessner** en el pasado Tech-Ed 2005 de Orlando.

Los problemas de seguridad de SQL Server 2000 por culpa de nuevos tipos de ataques no contemplados –virus *slammer*–, que hicieron que los desarrolladores de Yukon tuvieran que arremangarse para sacar nuevos *service packs* de SQL Server 2000; los inevitables retrasos que sufren proyectos de este calibre; que la integración con una tecnología que aún no estaba desarrollada (CLR) fue más difícil de lo esperado y, quien sabe, si algunos problemas con el desarrollo de WinFS, han hecho, pienso yo, que este SQL Server sea 2005 y no 2003. Por supuesto esto no es más que una suposición mía... Y además de qué sirve; porque lo que es realmente importante es que el desarrollo de SQL Server 2005 prácticamente ha finalizado y que trae muchas novedades y promete, ya lo creo que promete.

Bienvenido al número 17 de julio/agosto de 2005 de **dotNetManía**.

Éste es un monográfico de SQL Server 2005, con el que incluimos –sólo a suscriptores y hasta agotar existencias–, un DVD con la versión CTP de Abril de SQL Server de 2005. Recientemente acaba de salir la que posiblemente sea la última CTP (o la penúltima, como las copas), pero la versión que entrega-

mos –gracias al esfuerzo de Microsoft Ibérica– incluye toda la funcionalidad que tendrá la versión final, con los cambios de depuración que se precisen y que se integra perfectamente con la Beta 2 edición Standard que distribuimos el mes pasado. Con esto tiene el entorno de prueba e incluso de producción necesario para comenzar a trabajar con las versiones 2005.

La intención de este monográfico es darle, además del software, la información necesaria para asimilar los cambios más importantes de la nueva versión. Tenemos dos años por delante (se dice que la próxima versión será en 2007) para ahondar en muchos aspectos que no tienen cabida en un solo número. Aún así hemos ampliado la revista en 16 páginas.

Se han involucrado muchas personas y compañías en la confección de un número como éste. Por un lado, mi agradecimiento a **Paul Flessner** y **César Galindo-Legaria** de **Microsoft Corporation** por responder a nuestras preguntas; y al personal de prensa de **Microsoft Ibérica** que nos han ayudado hasta el último momento. Por otro, a las empresas **Alhambra-Eidos** y **Solid Quality Learning** quienes han aportado contenidos de gran calidad con algunos de los mejores consultores del país en SQL Server, y al resto de colaboradores, por supuesto, que han estado pendientes de que todo estuviera a punto desde hace tiempo; y especialmente quiero agradecer públicamente el esfuerzo a **Marino Posadas**, quien ha trabajado incondicionalmente desde el principio y los resultados pueden comprobarlos usted mismo.

Espero que le guste.





Descubre el  
Master  
ideal para los  
Desarrolladores  
Profesionales

www.alhabra-eidos.com

dnm.sumario



... ya  
tienes tu  
sitio  
reservado

<b>Una nueva generación de sistemas de gestión de la información</b>	<b>10-12</b>
<i>Presentación de Fernando Bocigas, de Microsoft Ibérica del lanzamiento de SQL Server 2005 que será, sin ninguna duda, uno de los más importantes de Microsoft hasta la fecha.</i>	
<b>SQL Server 2005: Mirando hacia el desarrollador</b>	<b>13-14</b>
<i>Si en este monográfico se muestra SQL Server 2005 desde distintas perspectivas, desde este artículo se opina sobre cómo esta nueva versión del producto atiende más que nunca las necesidades que tienen los desarrolladores de soluciones de software.</i>	
<b>Entrevista a Paul Flessner</b>	<b>15-17</b>
<i>Paul Flessner es Vicepresidente Senior de Aplicaciones de Servidor en Microsoft Corporation, donde dirige un equipo que aglutina la producción de Microsoft Windows Server System, conjunto de aplicaciones que incluye Exchange Server, SQL Server, BizTalk Server, Content Management Server, Host Integration Server y Commerce Server.</i>	
<b>Consiguiendo alta disponibilidad con SQL Server 2005</b>	<b>18-23</b>
<i>Durante las próximas líneas vamos a dar un vuelo sobre las posibilidades que tenemos en el mercado para construir sistemas de alta disponibilidad. Lo más complejo de estos sistemas muchas veces no son los conceptos, que pueden ser adquiridos con bastante agilidad si el lector tiene ya una experiencia sobre Windows Server y SQL Server como la disponibilidad.</i>	
<b>Integración del CLR con SQL Server 2005</b>	<b>24-31</b>
<i>De todo el conjunto de novedades que se incluyen con SQL Server 2005, la que probablemente llame la atención de forma más poderosa a la comunidad de programadores, sea la capacidad de escribir el código de ciertos objetos del motor de datos utilizando alguno de los lenguajes de la plataforma .NET.</i>	
<b>Entrevista a César Galindo-Legaria</b>	<b>32-35</b>
<i>César Galindo-Legaria es el responsable de la Optimización de Consultas, del grupo de Relational Engine en el equipo de desarrollo de SQL Server 2005 en Microsoft Corporation.</i>	
<b>Business Intelligence en SQL Server 2005</b>	<b>36-38</b>
<i>Un breve vistazo por todas las novedades que ha incorporado SQL Server 2005 para el desarrollo de proyectos de Business Intelligence. Se han producido cambios sustanciales en los Servicios de Transformación de Datos (DTS) y en los Analysis Services (AS), y se han incorporado nuevas herramientas de desarrollo, basadas en Visual Studio .NET.</i>	
<b>SopORTE XML en SQL Server 2005</b>	<b>40-43</b>
<i>SQL Server 2005 incorpora novedades en el soporte XML desde el lado del cliente (ADO.NET 2.0) y desde el lado del servidor (FOR XML, tipo de datos nativo, OPENXML). En este artículo veremos las posibilidades XML que tenemos en SQL Server 2000.</i>	
<b>SQL Server 2005 Mobile Edition</b>	<b>44-46</b>
<i>Microsoft SQL Server 2005 Mobile Edition es la evolución de Microsoft SQL Server CE Edition 2.0. Microsoft SQL Server 2005 Mobile Edition pretende ser una plataforma de almacenamiento de datos para soluciones móviles integradas bajo un marco de desarrollo rápido, flexible, fácil y escalable.</i>	
<b>Transact-SQL 2005. Más productivo que nunca</b>	<b>48-54</b>
<i>Este es un artículo complicado de escribir. No por los contenidos, sino por el cómo abordar algo tan amplio como Transact-SQL. Este componente de SQL Server es el pilar fundamental en el que se basan el resto de los componentes, pues es el lenguaje de programación con el que se le dice al core de SQL Server lo que debe de hacer.</i>	
<b>Proporcionando escalabilidad a SQL Server 2005 con Service Broker</b>	<b>55-59</b>
<i>Este artículo hace una introducción al nuevo Service Broker con el que SQL Server 2005 dota al motor relacional de capacidades nativas de gestión de colas, permitiendo crear aplicaciones de bases de datos escalables, seguras y confiables, basadas en procesos asíncronos.</i>	
<b>Introducción a SMO</b>	<b>60-62</b>
<i>El concepto de programabilidad del servidor se ha extendido notablemente, permitiéndonos manejar mediante código .NET casi cualquier característica de SQL Server 2005. Es el modelo SMO.</i>	
<b>SQL Server 2005 Express Edition. El menor de la familia</b>	<b>64-66</b>
<i>Dicen que las mejores cosas de la vida son gratis. SQL Server 2005 Express también lo es. Se trata del sucesor de MSDE y del hermano menor de las ediciones de SQL Server 2005. Pero no por ello está exento de interés ya que tiene el mismo motor de base de datos que sus hermanos mayores.</i>	
<b>dnm.comunidad.net</b>	<b>68-69</b>
<i>GUSE.Net, Grupo de Usuarios del SurEste</i>	
<b>dnm.todotnet.qa</b>	<b>70-72</b>
<i>SQL Server 2005</i>	
<b>dnm.biblioteca.net</b>	<b>73</b>
<i>Microsoft SQL Server 2005 New Features A First Look at Microsoft SQL Server 2005 for Developers</i>	
<b>dnm.desvan</b>	<b>74</b>

XII Edición  
Master Alhambra-Eidos  
en Tecnologías Avanzadas  
para el Desarrollo  
de Software

... ya tienes tu sitio reservado\*

... para más información: [www.alhambra-eidos.com/master/dnm](http://www.alhambra-eidos.com/master/dnm)



\*matricúlate ahora y obtendrás un **descuento del 10%** ( oferta válida hasta el 31 de Julio )



## Microsoft Tech-Ed 2005 Orlando

Tech-Ed es el mayor evento formativo para profesionales IT y desarrolladores que Microsoft organiza por todo el mundo. En Orlando, entre el 5 y el 10 de junio se celebró el primero y más importante de todos.

Orlando, FL • June 5-10



learn.solve.grow

Este es un año en el que los desarrolladores esperamos versiones importantes como Visual Studio 2005 y SQL Server 2005, pero que además debemos estar pendientes de todo lo que acontece alrededor del desarrollo de Longhorn, de los nuevos API y subsistemas de programación. Por tanto, eventos de este tipo son fundamentales, pues en ellos no sólo se aprende sobre las tecnologías actuales y futuras y se obtienen recursos, sino que también son una fuente de información sobre los roadmaps de



Paul Flessner

productos y tecnologías.

En Orlando, la *keynote* de Paul Flessner, vicepresidente de aplicaciones de servidor, fue la principal fuente de esta información. Si desea ver la transcripción completa y los PPT, puede hacerlo en la dirección: <http://www.microsoft.com/presspass/exec/flessner/06-07Tech-Ed05.msp> o en sus notas del Tech-Ed 2005 en <http://www.microsoft.com/presspass/misc/06-07flessner-teched05.msp>.

Aquí haremos un resumen de lo más importante de lo que Flessner dijo (además puede leer la entrevista que le hemos realizado en este mismo número y que nos respondió justo después de terminar este Tech-Ed).

### Microsoft pone fechas de salida de Visual Studio 2005, SQL Server 2005 y BizTalk Server 2006

Paul Flessner anunció el momento para el que se está preparando el lanzamiento de SQL Server™ 2005, Visual Studio® 2005 y BizTalk® Server 2006 y anunció que estos productos serán formalmente lanzados durante la semana del 7 de noviembre. ¡Se acabaron los rumores, pues!

Nada sorprendente, la verdad, salvo que se haya incluido en este paquete a BizTalk Server 2006 adelantando su fecha de salida. Sin embargo, han tenido en cuenta la importancia de la integración con Visual Studio 2005 y SQL Server 2005 para este adelanto.

### Primeros benchmarks de SQL Server 2005

Durante esta *keynote*, Flessner ofreció también los primeros benchmarks para SQL Server 2005. Como César Galindo-Legaria nos recuerda en su entrevista, no está permitido publicar estos benchmarks hasta que no falten menos de 6 meses para su salida. Y ya están aquí estos primeros resultados de TPC (Transaction Processing Performance Council).

El primer benchmark TPC-C para Microsoft Windows Server 2003 Datacenter Edition 64-bit está realizado en un HP

Integrity Superdome 64P con Microsoft SQL Server 2005 Enterprise Edition 64bit y reporta 1.082.203 tpmC (*Transacciones Por Minuto*) con un coste de 5,38\$ por tpmC, situándole en el primer puesto en las bases de datos bajo Windows. Este resultado es un 38% más alto que SQL Server 2000 con un 7% de mayor rendimiento y un 35% menos en coste por transacción que Oracle 10g sobre este mismo equipo bajo HP-UX (8,33\$ por tpmC y con 1.008.144 tpmC).



Uno de los Track Cabanas (Foto: Eladio Rincón)

En los benchmarks TPC-H, SQL Server 2005 ha subido el rendimiento hasta un 162% más alto que SQL Server 2000 y un 38% más que los mejores resultados de Oracle en un procesador servidor Intel Itanium 2 con un coste que es un 20% menor por consulta. En otro resultado más de benchmark TPC-H, NEC reportó un resultado de 20.261 QphH (queries por hora) y un coste de 76,00\$ por QphH en un NEC Express 5800/1320Xe (32SMP), mejorando cerca de 4 veces el mejor rendimiento de SQL Server 2000. Más información sobre estos resultados en <http://www.tpc.org>.

### Primera CTP pública de SQL Server 2005

Microsoft ha anunciado también la CTP (*Community Technology Preview*) de junio de SQL Server 2005 que es la primera pública y que puede descargarse de <http://www.microsoft.com/sql/downloads>. Esta CTP es en realidad una *pre-release* y posiblemente sea la última antes de la versión RTM (*Release To Manufacturing*), aunque Flessner no descarta alguna CTP más.

### Cambios en la línea de productos de SQL Server 2005

Para incrementar las capacidades de los clientes y socios para utilizar capacidades de negocio en profundidad, Microsoft extenderá **SQL Server 2005 Reporting Services** de las ediciones Standard y Enterprise a todas las versiones de SQL Server 2005, incluyendo las ediciones Express y Workgroup. Además, **SQL Server 2005 Report Builder** estará disponible con las ediciones Workgroup, Standard y Enterprise.

### Asistente para la migración desde Oracle

Para migrar desde Oracle a la nueva plataforma de SQL Server 2005,

Microsoft ha publicado el "SQL Server Migration Assistant". Este asistente para la migración está disponible gratuitamente en: <http://www.microsoft.com/sql/migration>

### Iniciativa RFID

Finalmente, Flessner anunció una iniciativa para ofrecer una infraestructura de servidor escalable de la tecnología inalámbrica RFID, permitiendo a los clientes y socios conectar fácilmente aplicaciones de líneas de negocio con sensores y dispositivos RFID. La infraestructura está empezando a construirse usando la tecnología de Microsoft .NET y Windows Server System y está diseñada para conectar dispositivos, capturar e interpretar datos, gestionar eventos de negocio e integrar sistemas y procesos más fácilmente que nunca.

### Tech-Ed 2005 Europe

Cuando tenga este ejemplar en sus manos ya se habrá celebrado el Tech-Ed 2005 Europe que este año vuelve a celebrarse en Ámsterdam. Allí estaremos, cómo no, y en el próximo número le contaremos todo lo acontecido.

## WSE 3.0 CTP de junio

Web Services Enhancements 3.0 for Microsoft® .NET (WSE) edición CTP es un add-on para Microsoft Visual Studio® 2005 y Microsoft .NET Framework 2.0 que permite a los desarrolladores construir servicios Web seguros basados en las últimas especificaciones de protocolo para servicios Web.

WSE 3.0 puede ser instalada en paralelo al WSE 2.0 SP3. WSE 2.0 SP3 es compatible sólo sobre .NET Framework 1.1.

Esta versión de WSE 3.0 está construida para los desarrolladores que usen Visual Studio 2005 y .NET Framework 2.0 Beta 2.

La próxima CTP será en julio, mientras que en agosto aparecerá la beta 1 y la RTM para finales de año, según **Mark Fussell** autor del artículo de MSDN "What's New in Web Services Enhancements (WSE) 3.0" y que puede leer en: <http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnwse/html/newwse3.asp>.

Puede descargar esta versión en: <http://www.microsoft.com/downloads/details.aspx?FamilyID=7591dfd2-e1b7-4624-9d5b-29c211d149fe&displaylang=en> (busque "WSE 3.0" en [www.microsoft.com](http://www.microsoft.com) y le guiará hasta aquí).

## Borland Day 2005

Borland Ibérica ha celebrado ante más de 700 asistentes la segunda edición de su macro evento Borland Day. En esta ocasión, y con la presencia del CEO de la compañía **Dale Fuller**, ha mostrado a todos los profesionales involucrados su visión sobre la optimización de los procesos, o cómo transformar el desarrollo de software en un proceso gestionado y disciplinado alineando equipos humanos y tecnologías.



Dale Fuller

La segunda edición del Borland Day ha ofrecido una serie de ponencias y tracks con un repaso al papel que Borland está desempeñando en la evolución del ciclo de vida de las aplicaciones (ALM), al diseño e implementación de su plataforma *Borland Core SDP*, y a la adquisición de la compañía líder en la metodología de procesos CMM (Teraquest).

El evento ha ofrecido de forma simultánea tracks en otras tantas salas donde se han repasado las distintas soluciones para plataformas J2EE, novedades de IDE, este año especialmente enfocado a Delphi, y todo lo que tiene que ver con el desarrollo y experiencia en CMM, a través de la visión, entre otros, de **Jacob Okyne**, *Vicepresidente Borland /Teraquest* para la región de EMEA.

Como partners estratégicos, Borland Day ha contado con los líderes más importantes del sector: Accenture, Bea Systems, HP, Mercury Interactive, Microsoft, Oracle y Telefónica Móviles que han mostrado a lo largo de la jornada la integración de sus productos con la soluciones Borland.

## Infragistics lanza NetAdvantage 2005 Volumen 2



Infragistics, uno de los fabricantes más importantes de herramientas para desarrolladores de la plataforma NET y Java, ha presentado la nueva versión de su suite **NetAdvantage 2005 Volumen 2**, un completo conjunto de herramientas para construir rápidamente la capa de presentación de sus aplicaciones para Windows Forms, ASP.NET, Tablet PC y aplicaciones COM. En NetAdvantage 2005 Volumen 2 se añaden nuevas características a los *Windows Forms Tree, Grid y TabbedMDIManager* así como los *ASP.NET Grid y Chart*.

“Con NetAdvantage 2005 Volumen 2, continuamos añadiendo las características clave que los clientes nos piden, a la vez que incrementamos la calidad. NetAdvantage 2005 aporta el framework y las herramientas de desarrollo que los equipos necesitan para construir y desplegar rápidamente aplicaciones .NET con

un aspecto fantástico y con un interfaz de usuario consistente como sus organizaciones desean”, ha dicho **Steve Dadoly**, *Director de Desarrollo* de Infragistics.

Asimismo, Infragistics ha anunciado sus planes para el lanzamiento de **Infragistics NetAdvantage 2005 Beta 3 for Visual Studio 2005 Beta 2**. Esta edición aprovechará las ventajas de las nuevas características de facilidades de uso en Visual Studio 2005 Beta 2, permitiendo a los desarrolladores crear aplicaciones consistentes con la apariencia de Microsoft a la vez que escribe menos código que el que necesitaría usando Microsoft .NET Framework 1.1. El lanzamiento de la Beta 3 de NetAdvantage 2005 se espera para estos meses de julio y agosto y contendrá aquellas características que existan en NetAdvantage 2005 Volumen 2.

Más información: <http://www.infragistics.com>

## Dotnetsolidario llega a un acuerdo con Nevant

Nevant es una empresa especializada en la gestión de procesos y metodologías en organizaciones basadas en el conocimiento, para ofrecer a las ONG servicios gratuitos de consultoría de procesos y gestión de proyectos.

Estos servicios gratuitos consistirán en:

- *Consultoría orientada a la mejora de procesos.* Implementación de soluciones de *workflow* basadas en componentes open source. Esto permite automatizar procesos de negocio, optimizando el uso de recursos.
- *Consultoría en gestión de proyectos.* Implementación de una oficina de gestión de proyectos, implantación de una plataforma basada en sistemas open source para la gestión de proyectos y ayuda a que los proyectos se terminen a tiempo y dentro del presupuesto.

## ComponentOne anuncia el lanzamiento del programa beta de Doc-To-Help 2005

*Doc-To-Help 2005 añade más flexibilidad a esta popular herramienta permitiendo a los usuarios usar HTML desde cualquier editor para sus contenidos fuente, convertir proyectos de ayuda de RoboHelp® y otros proyectos de ayuda HTML y generar NetHelp™, un explorador independiente basado en el sistema de ayudas HTML.*

**ComponentOne** “Nosotros hemos seleccionado una sección representativa de la comunidad de creación de ayudas para asegurar que Doc-To-Help 2005 incluye todas las herramientas que un autor de ayudas que trabaja con Word o HTML pueda necesitar”, ha dicho **Zuñi Wong**, *Managing Director de ComponentOne*. “Esta versión de Doc-To-Help ha sido desarrollada con las demandas de la industria en mente, y esperamos ansiosos los comentarios que recogeremos durante el período de Beta test”.

Como esta beta no necesita un editor propietario, permitirá a los autores de ayudas usar su editor favorito, tales como Microsoft FrontPage o Macromedia Dreamweaver para crear contenido HTML y configurar sus sistemas de ayuda. Los usuarios de Doc-To-Help pue-

den usar D2HML (*Doc-To-Help Markup Language*), el cual consiste en unas clases de estilo para diseñar sistemas de ayuda en ficheros HTML. Puesto que Doc-To-Help requiere sólo estos estilos para transformar los ficheros HTML a sistemas de ayuda interactivos, el fichero HTML original permanece limpio y preparado para cualquier otro uso.

Otras herramientas de creación de ayudas permiten a los autores usar su editor preferido, pero los ficheros deben ser importados por la configuración del sistema de ayudas. Adicionalmente, los usuarios de otras herramientas de creación de ayudas podrán convertir sus ficheros de proyectos a los proyectos Doc-To-Help basado en HTML con un solo clic.

Para más información: [www.componentone.com/doctohelpbeta](http://www.componentone.com/doctohelpbeta).

## Intrinsyc Software anuncia su programa beta para J-Integra Espresso

*J-Integra Espresso es un nuevo conjunto de herramientas de desarrollo y componentes runtime de alto rendimiento que une las tecnologías Java, CORBA y Microsoft .NET*

**J-Integra Espresso** es una nueva y potente solución *middleware* que permite una interoperatividad de alto rendimiento, bi-direccional y segura y escalable entre aplicaciones Microsoft .NET, Java y CORBA.

Mientras que los actuales productos J-Integra implementan los protocolos estándar de Microsoft en Java puro, J-Integra Espresso implementa una versión en .NET puro del estándar protocolo binario RMI/IIOP presente en todos los entornos J2EE y J2SE.

J-Integra Espresso está actualmente disponible como beta para clientes "cualificados". Si está interesado en participar en este programa beta, puede inscribirse en <http://j-integra.intrinsyc.com/beta>.





# dotNetManía cd Vol. 1

Este CD recopila los 11 primeros números completos de dotNetManía en formato PDF de gran calidad -todos los textos publicados tal y como se imprimieron entre febrero de 2004 y enero de 2005-. Organizado por archivos que contienen las revistas completas y por otros que contienen los artículos por separado para una mejor localización de la información a través de un interfaz HTML. Podrá acceder a cada ejemplar o artículo a través de su explorador Web y tener así la información publicada siempre a mano.

Está producido en formato digipack de tres cuerpos, en una edición limitada y dirigido principalmente a los suscriptores, que podrán obtenerlo gratuitamente con su renovación, pero también para aquellos lectores que, sin ser suscriptores, quieran tener la mejor información de la plataforma .NET. Si usted es suscriptor y quiere obtenerlo gratuitamente podrá anticipar su renovación.

En el contenido del CD podrá encontrar:

- 11 archivos con los números completos en formato PDF.
- Cerca de 250 archivos con todos los artículos por separado en formato PDF a los que podrá acceder a través del sumario de cada número o por secciones o autores.
- 660 páginas publicadas con información exclusivamente de la plataforma .NET.
- Todos los archivos con el código fuente y de ejemplo de los artículos.
- Más de 400 Mb de información.
- Todos los documentos con la posibilidad de imprimirse.

## HAZTE YA CON ÉL

Para más información visita [www.dotnetmania.com](http://www.dotnetmania.com)



Fernando Bocigas

# Una nueva generación de sistemas de gestión de la información

Desearía en primer lugar señalar que supone una gran satisfacción para mí, como responsable de negocio de SQL Server, el poder anticipar a los lectores de **dotNetManía** las mejoras introducidas en la próxima y esperada versión de esta emblemática base de datos de Microsoft, conocida hasta ahora con el nombre en código de **Yukon**, y de la que recientemente hemos hecho pública una CTP (*Community Technology Preview*) del producto, con la firme intención de lanzar la versión final del mismo durante el segundo semestre de este año.

» El lanzamiento de **SQL Server 2005** será, sin ninguna duda, uno de los más importantes de Microsoft hasta la fecha, pues ya han pasado cinco años desde que apareció en el mercado SQL Server 2000 para revolucionar las bases de datos como no se había creído posible hasta el momento, y desde ese entonces, la compañía ha desempeñado un intenso trabajo y ha realizado un esfuerzo continuado en investigación e innovación para tratar de conseguir que SQL Server 2005 suponga un giro radical y sorprenda de nuevo a los usuarios.

La llegada de SQL Server 2005, que se producirá al mismo tiempo que la de la próxima versión de Visual Studio .NET “Whidbey”, y posteriormente la de Windows “Longhorn”, serán los cimientos sobre los que se asentarán las bases de la tecnología de los próximos años, y determinarán un punto de inflexión en la manera de trabajar de los profesionales de las tecnologías de la información de todo el mundo.

Dado que el sector ha cambiado mucho desde que Microsoft lanzó SQL Server 2000, la nueva versión de nuestra solución de base de datos y análisis se ha rediseñado para responder perfectamente a las necesidades de cada uno de los diferentes segmentos de nuestros clientes, y proporcionarles así un valor y funcionalidad sin precedentes, lo que representa un gran paso hacia delante para soportar los sistemas empresariales más exigentes.



Fernando Bocigas, Responsable de Negocio de SQL Server en Microsoft Ibérica

La oferta de Microsoft en cuanto a bases de datos crece y mejora con la nueva familia de productos SQL Server 2005, con la que aumentamos el alcance de nuestras soluciones de gestión de datos para ofrecer más opciones a los clientes. Hemos incrementado la funcionalidad en las ediciones *Standard* y *Enterprise* y ampliado la plataforma con las ediciones *Workgroup* y *Express*. Ahora estamos mejor equipados para ofrecer soluciones que respondan a los requerimientos tecnológicos y presupuestarios de nuestros clientes. Nuestro objetivo es que la gestión y el análisis de datos al nivel de las grandes corpo-

**Fernando Bocigas**  
Es Ingeniero Técnico Superior de Telecomunicaciones por la Universidad Politécnica de Madrid. **Fernando Bóci**gas es el Responsable de Negocio de SQL Server en Microsoft Ibérica desde 2004.



raciones sean accesibles para una amplia gama de clientes a un precio razonable, a la vez que eliminamos la complejidad de nuestros sistemas de bases de datos, y todo a un menor coste total de propiedad.

Microsoft SQL Server 2005 se ha concebido para ofrecer al usuario una experiencia más rica, mediante mejoras sustanciales en aspectos cruciales como la escalabilidad, la disponibilidad, la seguridad y la gestionabilidad. Además, se han contemplado otros aspectos tan relevantes para las organizaciones en el competitivo entorno empresarial actual como el *business intelligence*, que se ha potenciado considerablemente en esta versión, y una integración más completa con las herramientas de desarrollo, es decir, con Visual Studio .NET.

Hoy en día, las empresas se encuentran con una gran proliferación de datos y sistemas en toda la organización, y con la necesidad de que tanto sus empleados, como sus clientes y sus *partners* tengan acceso continuo y en tiempo real a los mismos y puedan tomar las decisiones mejores y más informadas, a la vez que cumplan con el requisito de controlar los costes sin renunciar por ello a la fiabilidad, seguridad o disponibilidad de las aplicaciones. SQL Server 2005 da respuesta a estos retos de la información a los que se enfrentan las organizaciones en las tres áreas principales: la gestión de la información empresarial, la productividad de los desarrolladores y el *business intelligence*.

Las mejoras en la gestión de los datos empresariales introducidas en SQL Server 2005 permiten que la información y los sistemas que la administran estén siempre disponibles para los usuarios. Tanto éstos, como los profesionales de TI de toda la organización, se beneficiarán de la reducción del tiempo de inactividad de las aplicaciones, de una mayor escalabilidad y rendimiento y de unos estrictos controles de seguridad.

Por otra parte, SQL Server 2005 ayudará a incrementar la productividad de los desarrolladores eliminando uno de los obstáculos principales con el que éstos se encontraban hasta

## Business Intelligence

### Servicios de Análisis

Con SQL Server 2005, los *Servicios de Análisis (Analysis Services)* se mueven en el entorno del análisis en tiempo real. Desde mejoras en la escalabilidad hasta una integración profunda con Microsoft Office, SQL Server 2005 amplía el concepto de *business intelligence* a todos los niveles de su negocio.

### Integration Services

Los *Integration Services* son un conjunto de herramientas gráficas y objetos programables que pueden usarse para extraer, transformar y cargar datos (ETL) desde fuentes muy diversas y llevarlas a un destino único o múltiples destinos. *Data Transformation Services (DTS)* para Microsoft SQL Server 2005 introduce un rediseño completo para proporcionar una plataforma ETL integral.

### Data Mining

SQL Server 2005 introduce cuatro nuevos algoritmos de *Data Mining*, así como herramientas y asistentes mejorados, haciendo que el data mining sea más accesible a negocios de cualquier tamaño.

### Servicios de Reporting

*Servicios de Reporting* permite a los negocios integrar de forma sencilla datos desde fuentes heterogéneas y data warehouses en informes ricos, interactivos y gestionables, que pueden localizarse y consultarse en intranets, extranets y en Internet.

### Soporte de Servicios de Análisis en Cluster

Esta funcionalidad mejora la disponibilidad de los *Servicios de Análisis* con soporte para clusters con conmutación de nodos, soporte mejorado para múltiples instancias y soporte para backup y recuperación de objetos y datos de *Servicios de Análisis*.

### Indicadores de Rendimiento Principales

Los *Indicadores de Rendimiento Principales (Key Performance Indicators, KPI)* permiten definir métricas de negocio en formato gráfico, adaptables, para ayudar a generar y hacer el seguimiento de entornos de prueba corporativos.

### Escalabilidad y Rendimiento

Se ha mejorado la escalabilidad y el rendimiento de los *Servicios de Análisis* de SQL Server 2005 por medio de características como el proceso de partición paralelo, creación de particiones ROLAP y HOLAP remotas, cubos particionados distribuidos, cálculos persistentes y cache proactivo.

### Cubo en un solo clic

Cuando se crea un cubo en un proyecto de *Data Warehouse*, el asistente para creación de *Cubos* incluye una opción para habilitar la detección de cubos y sugerencias con un solo clic. Esta tecnología de cubos en un solo clic de SQL Server 2005 examina las relaciones en una vista del origen de datos y aporta sugerencias.

### Mejoras en la Arquitectura

SQL Server 2005 introduce una nueva arquitectura de *Servicios de Transformación de Datos (DTS)* llamada *Integration Services*. La arquitectura consiste en dos motores:

- El *Runtime de Transformación de Datos (DTR)*. Este motor ejecuta paquetes DTS, tareas DTS, hace seguimiento de la ejecución del paquete y proporciona servicios para las tareas.
- El *Pipeline de Transformación de Datos (DTP)*. El motor DTP extrae datos de las fuentes, aplica las transformaciones contra las columnas de datos y carga los datos en los sistemas de almacenamiento.

### Integración con los informes de Office System

Los reports emitidos por el servidor de reports pueden ejecutarse en el contexto de SharePoint Portal Server y aplicaciones Office, como Word y Excel. Los usuarios de SharePoint pueden utilizar las funcionalidades de SharePoint para suscribirse a informes, crear nuevas versiones de los informes y distribuirlos. Los usuarios pueden también abrir reports en Word o Excel para ver versiones en HTML de los mismos. Los usuarios de SQL Server y Servicios de Análisis pueden definir reports basados en consultas contra bases de datos relacionales y multidimensionales. Las consultas definidas con el Query Builder pueden procesarse utilizando las posibilidades de proceso de datos incorporado al servidor de informes.

## Una versión diseñada para cada tipo de cliente

### SQL Server 2005 Enterprise Edition

Es una completa plataforma de gestión y análisis de datos para aplicaciones de negocio de misión crítica de grandes empresas. Esta edición ofrecerá partición de datos, alta disponibilidad avanzada con capacidades de *database mirroring*, analíticas complejas e integración, generación de informes ad hoc con *Report Builder*, un *snapshot* de base de datos, y operaciones online y paralelas completas. Microsoft también continuará ofreciendo **SQL Server 2005 Developer Edition** para desarrolladores que quieren construir y testar aplicaciones basadas en SQL Server.

### SQL Server 2005 Standard Edition

Esta edición está pensada para las medianas empresas e infraestructuras que requieren sistemas altamente disponibles. Proporcionará funcionalidades que antes sólo estaban incluida en **SQL Server 2000 Enterprise Edition**, tales como alta disponibilidad con *database mirroring* y *clustering* y soporte integrado de 64-bit tanto para sistemas **x64** como **Itanium**, lo que les proporcionará más flexibilidad para crecer antes de invertir en la edición *Enterprise*. Soporta hasta 4 procesadores, base de datos de tamaño ilimitado y sistema de memoria ilimitado. También incluye *SQL Server Integration Services*, *SQL Server Analysis Services* y *SQL Server Reporting Services*.

### SQL Server 2005 Workgroup Edition

Es el producto más novedoso disponible tanto para SQL Server 2000 como SQL Server 2005, y proporciona una solución de base de datos asequible, fácil de usar y sencilla de gestionar diseñada específicamente para las pequeñas y medianas organizaciones. Soporta hasta dos procesadores, bases de datos de tamaño ilimitado y 2Gb de memoria.

### SQL Server 2005 Express Edition

Es una versión gratuita y redistribuible del motor de base de datos de SQL Server 2005, diseñada para construir sencillas aplicaciones de base de datos, que sustituye a *Microsoft Data Engine* (MSDE) para SQL Server 2000. Se puede descargar gratuitamente de la Web e incluye una herramienta gráfica de gestión; controles de *report*; replicación, un cliente *SQL Service Broker*, encriptación de base de datos nativa; soporte XML y *Common Language Runtime* (CLR).

Con SQL Server 2005, los clientes podrán aprovechar el licenciamiento para procesadores *multicore*, que les permite licenciar el producto por procesador; frente a las ofertas de la competencia que licencian por *core*. Este nuevo modelo potencia el rendimiento informático y la funcionalidad y permite a las empresas obtener el máximo valor de su inversión en tecnología. A diferencia de sus principales competidores, Microsoft también lidera el sector en alta disponibilidad, al ser el primer fabricante de bases de datos que permite a los clientes utilizar servidores *passive failover* con SQL Server 2005 sin requerir licencias adicionales, con lo que pueden crear entornos de alta disponibilidad a bajo coste.

cionalidad de generación de informes y almacenamiento de datos. Además, también aporta mejoras en la toma de decisiones en todos los niveles de la organización, mayor seguridad y disponibilidad para que los usuarios tengan un acceso ininterrumpido a los informes y aplicaciones de *business intelligence*; y mejores capacidades de análisis en toda la empresa.

Esperamos que todas estas mejoras contribuyan a que el nuevo SQL Server 2005 se convierta en el software de análisis y administración de datos del futuro que tanto nuestros clientes, como desarrolladores y *partners* desean, ya que esta solución es una muestra evidente del continuo compromiso de Microsoft por ofrecer una nueva generación de tecnologías que ayuden a incrementar la productividad y competitividad de todos los usuarios y organizaciones. ○

ahora, y que no era otro que la falta de herramientas integradas para la depuración y el desarrollo de bases de datos. SQL Server 2005 incluye mejoras en las herramientas de desarrollo e integración con el entorno de desarrollo de Visual Studio; compatibilidad de lenguajes ampliada para seleccionar entre una serie de lenguajes conocidos para desarrollar aplicaciones de bases de datos, y admitirá XML y servicios Web.

Por último, en lo referente a las nuevas funcionalidades de *business intelligence*, éstas se han potenciado considerablemente y SQL Server 2005 ofrece una plataforma completa con análisis integrado, que incluye OLAP, extracción de datos, ETL, fun-

## Licencias

### Licencia por procesador

Una licencia para cada procesador en un servidor corriendo SQL Server.

### Servidor más CAL de dispositivo

Una licencia para cada servidor corriendo SQL Server, más una *Licencia de Acceso de Cliente* (CAL) para cada dispositivo cliente.

### Servidor más CALs de usuario

Una licencia para servidor corriendo SQL Server, más una CAL para cada usuario con acceso al servidor.





Antonio Quirós

# SQL Server 2005

## Mirando hacia el desarrollador

Si en este monográfico se muestra SQL Server 2005 desde distintas perspectivas, desde este artículo se opina sobre cómo esta nueva versión del producto atiende más que nunca las necesidades que tienen los desarrolladores de soluciones de software. Y son muchas. Y creo que nuestro amigo Yukon no las decepcionará.

>> **Uno echa de menos** aquella época en que era técnico y ante un producto como éste se lanzaba de lleno a probarlo, destriparlo y sacarle todo el jugo posible. Ahora me toca estar en otra onda y pensar en los productos con otra perspectiva. Sin embargo, el polifacético **Yukon** abona sobradamente el terreno de la reflexión, ya que rompe muchos moldes y hace evolucionar el concepto de base de datos hacia derroteros bastante interesantes.

En general, el mundo de las bases de datos ha evolucionado impulsado por numerosos factores; entre ellos podemos destacar las necesidades de mejores sistemas de almacenamiento y acceso rápido a los datos, las necesidades de seguridad transaccional, las demandas de una administración más rápida y segura, la tipología de los datos que se almacenan, el modelo centralizado o distribuido de gestión de los datos, las mejoras en SQL, el lenguaje de acceso a datos, etc. Sin embargo, rara vez se ha dado un movimiento claro a fin de mejorar los recursos que las bases de datos proporciona a los desarrolladores de software para mejor usar sus recursos. Y la prueba la tenemos ahí palpable; lenguajes como PL/SQL de Oracle o Transact SQL de SQL Server son un buen ejemplo de este tipo de situación. Ambos permiten realizar funciones básicas, pero no admiten ni la más leve comparación con cualquier lenguaje serio y formalmente bien construido. Es como remontarse a la prehistoria del desarrollo cuando tenemos lenguajes sofisticados para resolver el resto de las cuestiones con las que cada día nos enfrentamos.

Con SQL Server 2005, Microsoft se ha propuesto numerosos objetivos. Y por algo ha tardado 5 años

en generar esta nueva versión, periodo éste bastante inusual en lo que se refiere a actualizaciones de productos de software. No voy a entrar aquí en el análisis de todos estos objetivos ni en la valoración de si ha sido mucho o poco tiempo de espera. SQL Server es una tecnología madura y su evolución está respondiendo a dichos parámetros de madurez. Me detendré, pues, sólo en aquellos aspectos que contribuyen a mejorar la cara con la que un desarrollador de software conmuta entre su apreciado Visual Studio y ese árido SQL Server 2000 al que hasta ahora estábamos acostumbrados, es decir, que me concentraré en aquello que mejorará la productividad del desarrollador.

---

... ahora existe una auténtica y productiva integración entre Visual Studio y SQL Server

---

Veamos a continuación un somero análisis de estos aspectos para que nos ayuden a comprender la profundidad del cambio a que me refiero:

- SQL Server 2005 permite escribir funciones, procedimientos y triggers usando C# o Visual Basic .NET. Ahora el CLR está albergado en el motor de



la base de datos y el desarrollador tiene a su disposición la librería de clases del .NET Framework para modernizar la escritura de aquellos obsoletos procedimientos almacenados, funciones o triggers. ¿Os imagináis trabajar con objetos dentro de un procedimiento? La modernidad llega a las bases de datos al fin. Ya está bien de conmutar entre el futuro y la prehistoria cuando pasamos de escribir lógica de negocio o de presentación desde Visual Studio a escribir lógica de datos desde SQL Server.

- Mejora de Transact SQL por si, a pesar de todo, quieres seguir usándolo.
- Servicios Web integrados en la capa de datos. Ahora podemos escribir servicios Web que son gestionados por el propio SQL Server que se convierte así en un servidor HTTP lo que permite prescindir de Internet Information Server en ocasiones. Esto, junto con el soporte extendido para XML constituyen a SQL Server en una herramienta altamente autónoma para crear y gestionar servicios Web que usan datos almacenados en la base de datos y devuelven código XML aprovechable por las distintas aplicaciones.

---

## ¿Os imagináis trabajar con objetos dentro de un procedimiento? La modernidad llega a las bases de datos al fin

---

- Tipos de datos definidos por el usuario que pueden ser empleados en la definición de columnas de una tabla. Así, pues, ahora podemos complementar los tradicionales `int`, `char`, etc., con nuestros propios tipos creados según las necesidades que nuestras aplicaciones presenten.

- Un nuevo modelo de objetos (SMO - SQL Management Objects) que sustituye al antiguo DMO. Este nuevo modelo de objetos contiene la funcionalidad de la base de datos pero susceptible de ser manejada programáticamente de manera simple al estar estructurada como un *assembly* de .NET Framework. Contamos, además, con otro nuevo modelo denominado AMO (*Analysis Management Objects*) específicamente diseñado como librería de objetos para Analysis Services.
- Un entorno de desarrollo común. Ahora todo el desarrollo de la lógica de datos está integrado dentro de Visual Studio 2005, lo que quiere decir que desde dicho entorno podemos desarrollar y, ¡ojo!, depurar, por ejemplo un script de SQL conteniendo tratamientos sobre objetos de bases de datos, procedimientos almacenados, etc. Se acabó conmutar continuamente entre Visual Studio y SQL Server para escribir distintos tipos de código; ahora Visual Studio es una pieza más de los servicios que SQL Server presta a los desarrolladores de software.
- Más soporte para Business Intelligence. Ahora disponemos del Business Intelligence Development Studio, un entorno basado en Visual Studio para el desarrollo de soluciones que integran tratamiento del motor de datos con Analysis Services y Reporting Services. Un lujo para el hasta ahora árido mundo de los servicios OLAP.
- Y, para finalizar, una de las mejores cosas: SQL Server Express. Una versión específica que permite a los desarrolladores crear aplicaciones para sus clientes que no tengan un SQL Server de pago. Esta versión es distribuible de forma gratuita y tiene algunas limitaciones como que por ejemplo sólo puede ser usada en un CPU con 1GB de RAM y 4GB como máximo de tamaño de la base de datos. Sin embargo, para muchos desarrolladores de aplicaciones para pequeños entornos será una magnífica solución con un mucho mayor nivel de integración y acceso del que tenía MSDE y, desde luego, con un entorno infinitamente más robusto, rápido, seguro y fiable del que podían ofrecer las aplicaciones desarrolladas para Access.

En resumen, podemos decir que ahora existe una auténtica y productiva integración entre Visual Studio y SQL Server. Ello contribuirá sin duda a una mayor comodidad y rapidez en el desarrollo de soluciones. Nuestra productividad como desarrolladores nos lo agradecerá. Con su última política a este respecto Microsoft ha abierto una nueva línea frente a otros productos de su competencia. Está claro que el abordaje del mercado corporativo incluye hacer que los desarrolladores de aplicaciones prefieran el motor de datos de Microsoft a otro cualquiera. La integración entre Visual Studio y SQL Server es la puerta que abre este camino. ○





Marino Posadas

## Entrevista a Paul Flessner

**Paul Flessner** es *Vicepresidente Senior de Aplicaciones de Servidor* en Microsoft, donde dirige un equipo que aglutina la producción de Microsoft Windows Server System, conjunto de aplicaciones que incluye *Exchange Server, SQL Server, BizTalk Server, Content Management Server, Host Integration Server y Commerce Server*. Paul ha sido el alma de las transformaciones de las herramientas de servidor independientes de Microsoft en el conjunto de servidores integrado que se presenta hoy día. Aunque su nivel de ocupación es altísimo y no fue posible la comunicación presencial con él, sí pudimos realizar un e-Interview, que presentamos a continuación.

» **Como Vicepresidente Senior** de Aplicaciones de Servidor en Microsoft tienes que tratar con los equipos de desarrollo de muchos productos distintos. ¿Cuánta gente depende de ti y cuál es el canal usual de comunicación con ellos? ¿Hablas con los grupos personalmente?

Somos un equipo e intentamos comunicarnos cara a cara siempre que es posible. Al mismo tiempo, hacemos una actualización semanal en la que les comunico vía e-mail cuál es la situación en el momento, y dónde queremos ir. El equipo depende de mí, pero al mismo tiempo, yo dependo de ellos. Para construir con éxito un producto como SQL Server tenemos que tener muchos expertos que nos den sus distintas visiones del producto. Por ejemplo, yo no soy el experto principal en *Business Intelligence*, así que acudo a **Bill Baker**, el *Manager General de BI*, para que me ayude con ideas directoras acerca de cómo piensa que debe desarrollarse el producto. Tiene que ser una relación de toma y daca, o sino, haríamos algo que no se ajusta a las necesidades de nuestros clientes.

Dinos algo acerca de las expectativas de futuro de tu departamento. Por un lado, ¿cuál será lo próximo en ver la luz? Y, además, ¿hasta qué punto depende un producto de otro distinto a la hora de su distribución comercial?

En este momento, nuestra prioridad fundamental es la salida de SQL Server 2005. Hemos estado trabajando con la ayuda de nuestros *partners*, para desarrollar una plataforma de datos completa, que les provea de las características empresariales de gestión de datos que necesitan, pero a un precio efectivo más competitivo.



Paul Flessner

Una de las grandes ventajas de utilizar productos Microsoft, es que están contruidos para integrarse entre sí. Comenzamos la planificación de los productos mucho antes de hablar públicamente sobre ellos. Y durante la fase de planificación, evaluamos

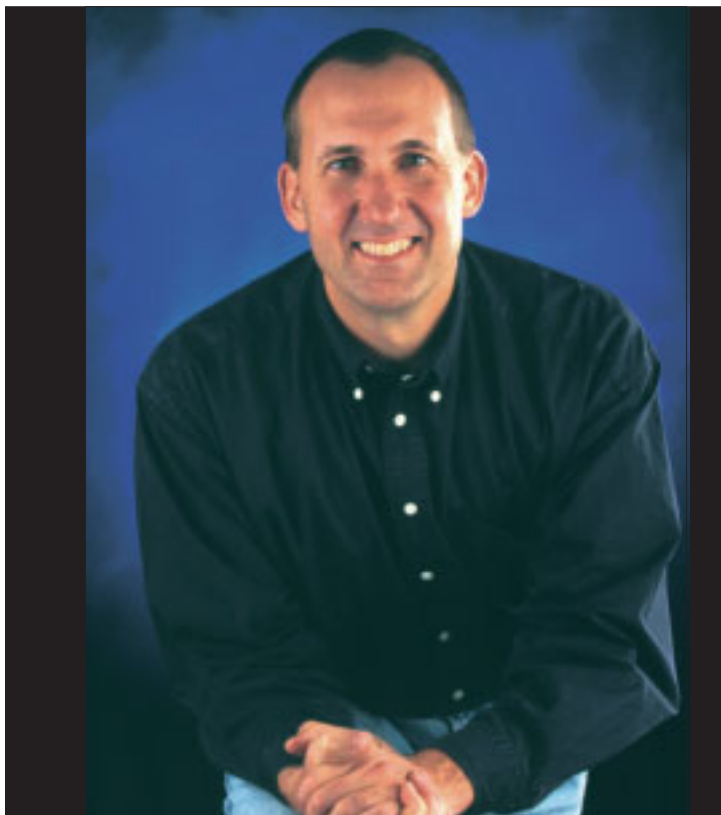
**Marino Posadas** es asesor técnico y redactor de dotNetManía, MVP de C# y formador de Alhambra-Eidos

## Pensamos que la integración de Visual Studio 2005 y SQL Server 2005, será un elemento diferenciador para ambos productos y un área de beneficio significativo para el cliente

cómo se comportará ese producto con el resto. Por ejemplo, empezamos a desarrollar SQL Server 2005 al mismo tiempo que se estaba desarrollando Windows Server 2003. Haciéndolo así, podíamos identificar fácilmente los puntos de integración de ambos productos. Ese es el beneficio real de utilizar una plataforma integrada, que somos nosotros los que realizamos la integración, no tiene que hacerlo el cliente.

**¿Crees que la salida en paralelo de Visual Studio 2005 y SQL Server 2005 será bueno o malo para SQL Server?**

Pensamos que la integración de Visual Studio 2005 y SQL Server 2005, será un elemento diferenciador para ambos productos y un área de beneficio significativo para el cliente. Microsoft se encuentra en una posición única porque es la única compañía que suministra una oferta completa de plataforma,



incluyendo bases de datos y herramientas de desarrollo integradas. Decidimos realizar una inversión en la integración completa de ambos productos, porque pensamos que los beneficios a largo plazo de una productividad de desarrollo mejorada, capacidades de misión crítica, y mejoras de la perspectiva de negocio para nuestros clientes.

**¿No crees que una distancia de cinco años entre las dos últimas versiones de SQL Server es demasiado? ¿Esperáis que los usuarios adopten todo ese conjunto de novedades ofertadas de forma inmediata?**

Creo que no esperábamos que el ciclo de desarrollo de ese producto durase cinco años, pero no creo que el ciclo de desarrollo extendido sea un detrimento. Nuestros clientes elogiaron las mejoras incluidas en SQL Server 2000, y esto ha crecido más en los últimos años respecto a su aceptación original. Ellos nos han dicho que esto es debido a que resulta fiable, escalable, de fácil administración y con buena relación precio/prestaciones. Dicho esto, hay muchos clientes deseando poder utilizar SQL Server 2005. Más de 400.000 clientes y *partners* han tenido la oportunidad de comprobar las nuevas características y hemos recibido un más que interesante *feedback* por su parte. Tenemos la seguridad de que estarán satisfechos con la nueva funcionalidad, y que pensarán que la espera ha merecido la pena.

**Y ya que hablamos de eso, ¿cuál sería tu consejo para los usuarios potenciales, desde el punto de vista de la curva de aprendizaje?**

En Octubre de 2004, presentamos la primera *Community Technology Preview* para SQL Server 2005. Estas CTP permiten a los clientes, evaluar las características de los productos de forma gradual. A aquellos que estén interesados en aprender más sobre SQL Server 2005, les animamos a probar estas CTP. Hemos presentado la última de estas CTP el 8 de Junio en el Tech-Ed de Orlando, y está disponible para descarga en la dirección [www.microsoft.com/sql/downloads](http://www.microsoft.com/sql/downloads). Y todavía habrá alguna otra disponible antes de la aparición final del producto.

Y si no estás preparado para comenzar la fase de testeo, también ofrecemos guías, borradores y *webcasts* en [www.microsoft.com/sql](http://www.microsoft.com/sql), que están diseñados para educar a la gente acerca de la nueva funcionalidad de suministra SQL Server 2005. También existe información comparativa respecto a otras bases de datos. Continuaremos actualizando estos recursos a medida que nos aproximemos al lanzamiento final, que será el 7 de Noviembre de 2005.

**Los usuarios, al menos aquí, pueden aceptar normalmente la existencia de *bugs* en aplicaciones de desarrollo o producción, pero, muy raramente en servidores. ¿Ha sido esa la razón principal del retraso en la salida? ¿Los planes de testeo?**



No hay una única razón para justificar el hecho del retraso en la salida de SQL Server 2005, pero lo que es más importante para nosotros es sacar un producto con la alta calidad que requieren nuestros clientes. Por ejemplo, en esta versión hemos cambiado totalmente nuestra visión de la seguridad, convirtiéndolo en una prioridad absoluta. Nos hemos asegurado de que SQL Server 2005 cumplirá con todas expectativas de nuestros clientes en ese sentido.

Otro punto importante tiene que ver con Longhorn. Ahora, todo parece apuntar en esa dirección. ¿Podrá SQL Server 2005 ser transferido a Longhorn de forma transparente?

Durante el ciclo de vida de SQL Server 2005 hemos trabajado en estrecha colaboración con el equipo de desarrollo de Longhorn Server, de la misma forma en que lo hizo el equipo de Windows Server 2003 con nosotros, durante su ciclo de desarrollo. Lo hemos hecho así para garantizar que muchas de las mejoras incluidas en SQL Server 2005 puedan ser utilizadas en Longhorn y viceversa. Ese trabajo hará que la transición para nuestros clientes sea muy sencilla.

¿Podrías subrayar dos o tres razones técnicas para convencer a nuestros lectores de las ventajas de la migración a SQL Server 2005? Me refiero a características funcionales, no a aspectos comerciales...

Hay muchas más de 3 características que creemos que serán de interés para nuestros clientes, pero comenzaré por aquellas que han sido más largamente solicitadas por ellos. Hemos mejorado notablemente las capacidades de administración con *SQL Server Management Studio*. Esto ofrece tremendas posibilidades de administración para SQL Server y también incluye a *Business Intelligence Development Studio*; en segundo lugar, nos hemos centrado en la seguridad, y el producto ofrecerá encriptación nativa de datos como característica avanzada de seguridad; otra novedad muy bien acogida por nuestros clientes, ha sido la replicación de datos en espejo (*Database Mirroring*). Esto permitirá la creación de soluciones de alta disponibilidad, con rápida tolerancia a fallos, y redirección automática de cliente.

---

...en esta versión hemos cambiado totalmente nuestra visión de la seguridad, convirtiéndolo en una prioridad absoluta. Nos hemos asegurado de que SQL Server 2005 cumplirá con todas expectativas de nuestros clientes en ese sentido

---



Paul Flessner en el "2004 Worldwide Partner Conference"

Esperamos que nuestros clientes estén impresionados con las nuevas posibilidades de SQL Server 2005. Y desde aquí, animamos a todos aquellos que quieran revisar la lista de todas las novedades disponibles, a que lo hagan desde la dirección <http://www.microsoft.com/sql/2005/productinfo/sql2005/features.asp>.

Finalmente, ¿consideras que SQL Server 2005 es el primero de toda una nueva generación de servidores de datos?

Completamente, pero no sólo por las mejoras tecnológicas que hemos incluido en SQL Server 2005. Cuando nos planteamos la construcción de SQL Server queríamos conseguir un sistema de bases de datos que fuera fácil de usar, fácil de manejar y con una buena relación precio/prestaciones. Diez años más tarde, hemos cambiado significativamente el panorama, suministrando una gestión de datos a nivel corporativo mucho más accesible y asequible. Como resultado, SQL Server ha crecido exponencialmente, mientras otros sólo intentan suministrar promesas parecidas. Esperamos que SQL Server tenga un impacto significativo en la industria porque aporta capacidades de gestión de datos de muy alto nivel, desarrollo integrado, y herramientas de gestión y de inteligencia de negocio, y todo ello a un TCO (*Costo Total de Propiedad*) mucho más bajo que otras plataformas de base de datos. ○



Miguel Egea

# Consiguiendo alta disponibilidad con SQL Server 2005

Durante las próximas líneas vamos a dar un vuelo sobre las posibilidades que tenemos en el mercado para construir sistemas de alta disponibilidad. Lo más complejo de estos sistemas muchas veces no son los conceptos, que pueden ser adquiridos con bastante agilidad si el lector tiene ya una experiencia sobre Windows Server y SQL Server como la disponibilidad (sé que suena un poco a paradoja pero es así).

» **Para poder tener** acceso a un cluster, es decir, a un sistema que nos proporcione *Alta disponibilidad* (HA de *High Availability* en adelante) necesitamos tener el hardware que lo soporte, o sea, un hardware que está pensado fundamentalmente para ser muy robusto, fabricado generalmente por los más afamados constructores de hardware y con un coste, por tanto, adecuado al nivel de calidad que ofrece (vamos, carísimo).

Hablando con propiedad, no solamente con un cluster podemos conseguir HA, puede conseguirse con técnicas como el *trasvase de registros* o *Log Shipping*, que ya tenemos en SQL 2000, también mediante replicación de bases de datos, y en SQL 2005 con otra técnica nueva llamada *Database Mirroring*, sin embargo, en lo que nos vamos a centrar en este artículo va a ser en lo que normalmente conocemos como HA y que realmente es un subconjunto de esto que es como supongo todos ustedes esperan la técnica de clustering.

Nos vamos a centrar en cómo conseguir con las –relativamente– nuevas técnicas de máquinas virtuales, instalar y configurar un cluster y simular por tanto con un hardware mucho más asequible este entorno. Posteriormente instalaremos, sobre ese entorno “clusterizado”, SQL Server 2005 y testaremos su funcionamiento.

Esto nos será de gran utilidad para poder realizar nuestros test en un entorno mucho menos comprometido si somos ingenieros de sistemas, y a probar adecuadamente nuestro software si somos arquitectos de software. Realmente un sistema en cluster debe

pasar desapercibido para el desarrollador, y así lo es habitualmente, sin embargo hay casos en los que no es así, o no tiene por qué serlo, en cualquier caso, parece mucho más responsable por nuestra parte, probar nuestros sistemas en un entorno lo más parecido a los entornos reales que se pueda y en este sentido parece muy adecuado poder probar las aplicaciones en un cluster, aunque éste sea virtual.

Un detalle que me gustaría dejar claro es que éste artículo no pretende ser una guía de instalación de cluster, sino un conjunto de ideas que nos permitan montarlo en máquinas virtuales para poder realizar las pruebas que estimemos oportunas. Si quiere tener una guía paso a paso sobre cómo instalar un cluster de forma detallada, puede usar el artículo de TechNet al que da acceso este link. <http://www.microsoft.com/latam/technet/articulos/200004/art05>.

Para tener acceso a la HCL (*Hardware Compatibility List*) lo mejor que podemos hacer es buscar en la propia web de Microsoft la lista de los productos soportados en esa HCL. En el momento de escribir este artículo la lista estaba en esta dirección <http://www.microsoft.com/windows/catalog/server/default.aspx?subID=22&xslt=categoryProduct&pgn=8b712458-b91c-4a7d-8695-23e9cd3ae95b>.

Como dato añadido, si alguien está interesado en profundizar en el tema de clustering hay un documento en la Web de Microsoft <http://www.microsoft.com/windowsserver2003/techninfo/overview/cluster-geo.mspx>, en el que habla de cluster geográficamente

Miguel Egea

Es Ingeniero Técnico en Informática y trabaja como consultor en Sinergia Tecnológica.

Es MVP en SQL Server y mantiene el site [PortalSQL.com](http://PortalSQL.com) y es mentor asociado de Solid Quality Learning



dispersos, basados en técnicas de mirror de discos junto con las tecnologías de clustering.

## Primeros pasos

La “clusterización” no es un servicio propio de SQL Server, sino del sistema operativo, de *Windows Server 2003 Enterprise Edition* y anteriormente de *Windows 2000 Advanced Server*. Hay otras versiones de estos productos que también admiten clustering, pero éstas son las más comunes. Tenemos, por tanto, que acudir a las características de Windows Server 2003 para tener una idea más clara de qué es y qué puede aportarnos en cuanto a HA se refiere el servicio de clustering. El servicio se llama *Server Cluster* en Windows Server 2003 y es la versión madura y avanzada del MSCS de Windows 2000 (entonces se llamaba parecido: *Microsoft Cluster Service*).

Para crear nuestro artículo usaremos Virtual Server 2005, servidores de máquinas virtuales hay varios, sin embargo la percepción que me ha causado VS2005 es francamente inmejorable. Una interfaz Web para el acceso a las máquinas virtuales y un rendimiento francamente muy, muy bueno. El aspecto es el que podéis ver en la figura 1, tomado mientras comenzaba con este artículo, instalando desde el principio un *Windows Server 2003 Enterprise Edition*. En mi día a día siempre había usado bien *Virtual PC 2004*, bien *VmWare* como herramienta de máquinas virtuales, y la verdad es que me ha dejado gratamente sorprendido.

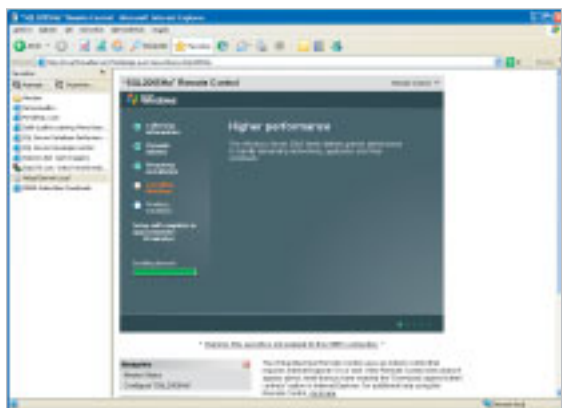


Figura 1

## Creando el cluster con Virtual Server 2005

En la propia ayuda de la máquina virtual, buscando la palabra clustering, tenemos una pequeña lista de instrucciones y pasos a seguir para configurar nuestro servicio de cluster.

### Elementos necesarios

En nuestra instalación de cluster vamos a hacer una instalación lo más parecida posible a una ins-

talación real, pero intentando ahorrar esfuerzos al máximo.

El primer paso que llevaremos a cabo será la instalación de un *Windows 2003 Enterprise Server* en el servidor de máquinas virtuales.

Usaremos tres máquinas virtuales, una de ellas nos hará de *Domain Controler* y las otras dos serán las que nos doten de HA, es decir, ahí “clusterizaremos” (perdonad la palabra) el servicio.

En nuestro caso vamos a crear un dominio llamado **SESAMO** y nuestros nodos se llamarán **EPI** y **BLAS**, estos nombres son un pequeño tributo a mi infancia, pero también a un proyecto en el que tuve la oportunidad de participar hace unos años y del que guardo un muy grato recuerdo.

---

## La “clusterización” no es un servicio propio de SQL Server, sino del sistema operativo, de Windows Server 2003 Enterprise Edition y anteriormente de Windows 2000 Advanced Server

---

Aunque se puede caer en la tentación de copiar y pegar las máquinas virtuales, es conveniente saber que el identificador de instalación queda registrado y eso nos dará problemas en nuestro directorio activo, no podremos hacer login y algunas cosas así. Es cierto que hay comandos que permiten resetear una instalación hasta casi el final del proceso pero son bastante agresivos (hablo de comandos como **sysprep**), sin embargo si buscamos en la Web de *SysInternals* podemos encontrar una utilidad llamada *NewSID* (<http://www.sysinternals.com/ntw2k/source/newsid.shtml>), que nos va a permitir cambiar solamente el SID y el nombre de la máquina.

De lo que acabamos de ver quizá alguien pueda pensar que la mejor forma, por tanto, es copiar y pegar el disco virtual y después usar *NewSID*, es cierto que es una alternativa, sin embargo, Virtual Server nos ofrece una funcionalidad llamada discos hijos que nos va a permitir crear a partir de un disco el resto simplemente de la diferencia que hay entre ambos. Ahorramos así tanto espacio físico como tiempo.

Para hacerlo tenemos que detener nuestra máquina virtual, marcar el disco (el fichero que lo contiene) como de sólo lectura y, a partir de ese disco, crear tantos discos de diferencias como necesitamos (tres en este caso). En cada uno de esos discos tendremos que ejecutar *NewSID* para limpiar el nombre y el identi-

ficador de la instalación; es conveniente realizar esta operación después de activada la copia de Windows. Cada máquina virtual apuntará al disco de diferencias que corresponda.

Posteriormente, promocionaremos una de las tres máquinas, la que hará de DC a controlador de dominio de un nuevo bosque, y lo llamaremos **SESAMO**. En la siguiente figura puede verse uno de los pasos de construcción del nuevo controlador de dominio.

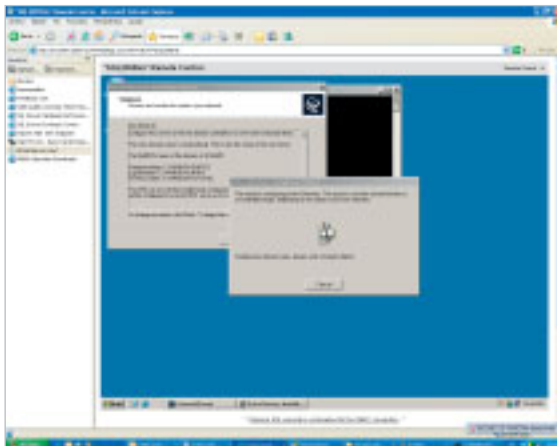


Figura 2

### Los nodos del cluster al dominio

Ahora comenzaremos a unir las máquinas del cluster al dominio, los detalles de *Active Directory* quedan muy lejos de este artículo, sin embargo, como pincelada para unir las máquinas al dominio simplemente hay que entrar en la pestaña de propiedades de la máquina y aparecen las opciones; hay que configurar la red para que el servidor de DNS resuelva los nombre (es importante este punto) pero en nuestro caso es tan sencillo como hacer que el servidor de DNS sea el controlador de dominio recién promocionado. Ya sabéis que para conseguir esto basta con especificar su dirección IP en “DNS preferido”.

En la figura 3 puede verse cómo se une cada uno de los nodos al dominio.

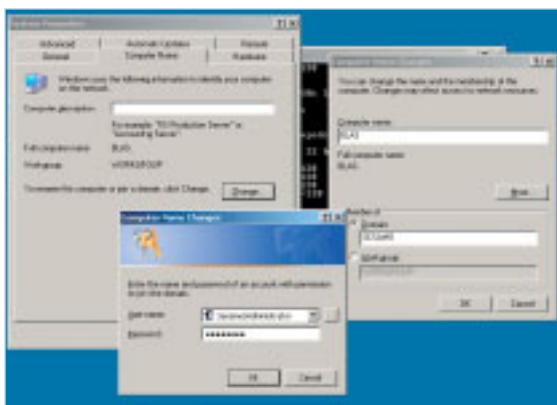


Figura 3

### La segunda tarjeta de red

Una de las necesidades que tenemos que cubrir es la de que los elementos que compongan el cluster tengan un mecanismo fiable y rápido de mandarse señales para poder comprobar si los demás están “vivos”. Esta señal se llama *heartbeat*, es decir, *latido*. Esto lo conseguiremos instalando en nuestra máquina virtual dos tarjetas de red (virtuales también).

Para hacer esta operación, lo primero que haremos será detener nuestra máquina virtual, y después entrar en su administración para poder añadir la nueva tarjeta.

Una vez añadida esta nueva tarjeta, podremos entrar a configurarla. Para nuestro ensayo utilizaremos tres máquinas virtuales que ejercerán de máquinas reales y necesitaremos además dos direcciones virtuales (simularemos un entorno activo/activo). Necesitaremos, por tanto, cinco direcciones IP que usaremos para cada una de las máquinas, más las dos virtuales y necesitaremos dos más para el *heartbeat* (es decir, para que los dos nodos del cluster que vamos a mostrar se vean). Utilizaremos las direcciones **192.168.1.x** para las máquinas donde **x** irá desde 0 hasta 4 y utilizaremos las direcciones **192.168.100.x** para el *heartbeat*.

En nuestro caso hemos sustituido **x** por el valor 2 para el servidor **BLAS** y el valor 3 para nuestro servidor **EPI**, hemos dejado el 1 reservado para nuestro *Domain Controller*. Por otra parte, hemos renombrado las redes (botón derecho, “renombrar”) para de un simple vistazo tener claro la tarjeta de red que estamos usando.

Virtual Server ofrece la posibilidad de crear redes virtuales. En nuestro caso no es necesario aunque es una opción. Para el ejemplo, hemos usado siempre tarjetas de red conectadas a la tarjeta física del equipo anfitrión, en este caso llamado “Fast Ethernet VIA PCI 10\_100Mb”.

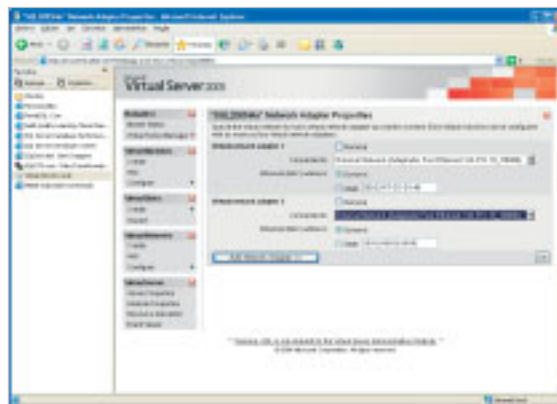


Figura 4

### El almacenamiento compartido

No puede montarse un cluster sin almacenamiento compartido. En los servidores físicos (no



virtuales como estamos usando para nuestras pruebas), este almacenamiento suele ser o bien disco SCSI que pueden ser compartidos, o bien cabinas de discos completas de distintos fabricantes.



Figura 5

Para instalar el primero añadiremos un adaptador SCSI para clustering seleccionando para ello la opción adecuada dentro de cada una de las máquinas virtuales.

Es conveniente que cada una de las máquinas tenga un *SCSI adapter ID* diferente, así pues a una de ellas le asignaremos el 7 y a la otra el número 6.

Después, crearemos un disco virtual llamado quórum; para este disco no necesitamos gran espacio, será el lugar común para comunicar el cluster y no tiene que tener un tamaño grande, aunque vale con menos, en nuestro ejemplo le hemos dado 1GB. Posteriormente y siguiendo el mismo proceso instalaremos un nuevo disco que será donde residirá al final nuestro Server SQL.

Continuaremos agregando los discos a nuestro almacenamiento, el propio sistema ya tiene como parámetro que son compartidos, por lo que nos permitirá arrancar de nuevo ambas máquinas virtuales sin problemas de uso de los discos.

Usaremos el administrador de discos para formatear el disco duro y asignarles unidad como puede verse en la siguiente figura:

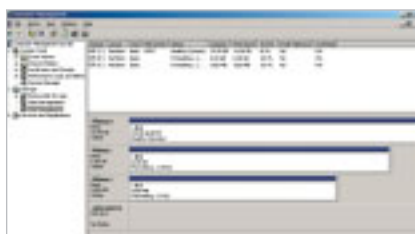


Figura 6

## Instalando Server Cluster

Para instalar el servicio de cluster no hay que hacer nada especial, basta con ir al menú herramientas administrativas y abrir la aplicación “Cluster Administrator”. El “Cluster Administrator” es la aplicación que nos va a servir para montar nuestro cluster.

Al abrirla nos ofrecerá tres opciones: en la primera de ellas podemos crear un nuevo cluster, en la segunda podemos añadir nodos a un cluster ya existente y en la tercera podemos conectarnos a un cluster para administrarlo.

Hay literatura que recomienda que el segundo nodo esté apagado cuando se comience con esta operación.

Comenzaremos con la opción “Crear Cluster”. En esta opción nos aparecerá un asistente que nos guiará en todo el proceso de creación del cluster. Nos irá solicitando una dirección IP nueva –ésta será la nueva dirección virtual de cluster–,

después chequeará nuestro sistema para comprobar si tenemos todos los recursos necesarios (doble tarjeta de red, almacenamiento compartido para quórum y recursos...) y después nos pedirá un nombre de usuario y una contraseña, que deberá ser el usuario que levante los servicios del cluster.

Nosotros, para simplicidad hemos elegido un administrador del dominio, pero esto, en general, no es buena idea de cara a la seguridad, es decir, si os toca montar un cluster en producción consultad la documentación de Windows y seguid las recomendaciones de seguridad al respecto.

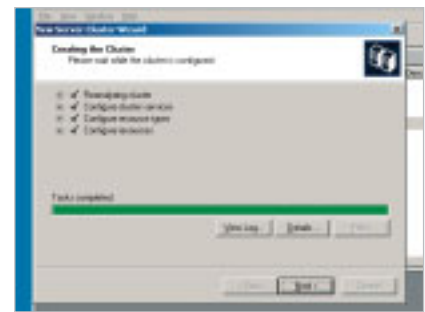


Figura 7

Hay que asegurarse no dejar como puede sugerir el sistema por defecto el quórum como “localquorum”, si se hace así, al no poder ser compartido no podrán añadirse nuevos nodos al cluster. Al acabar la configuración del sistema la respuesta será algo parecido a la figura anterior.

Cuando el asistente termina nos muestra toda la información de cómo ha quedado montado nuestro cluster. En nuestro caso la información es la siguiente:

```
Cluster name:
    BARRIOSESAMO.sesamo.local

Cluster IP address:
    192.168.1.10\255.255.255.0

Cluster network:
    lanExterna - Private and Public
    Intel 21140-Based PCI Fast Ethernet Adapter (Generic)
    Primary Address: 192.168.1.2 \ 255.255.255.0

Cluster service account credentials:
    Name: ClusterUsr
    Password: *****
    Domain: sesamo.local

Cluster node membership:
    EPI

Resource configuration:
    Local Quorum - NOT managed
    Disk E: - Managed - Quorum
    Majority Node Set - NOT managed

Network configuration:

lanExterna - Private and Public
    Intel 21140-Based PCI Fast Ethernet Adapter (Generic)
    Primary Address: 192.168.1.2 \ 255.255.255.0

HeartBeat - Private and Public
    Intel 21140-Based PCI Fast Ethernet Adapter (Generic)
    Primary Address: 192.168.100.1 \ 255.255.255.0
```

Terminado este proceso, unas cuantas comprobaciones son necesarias, por ejemplo, mover el único cluster group

que tenemos de nodo (a través de la opción “Move Group” del “Cluster Administrator”) y ver que el sistema queda confiable.

Para añadir el segundo nodo, lo arrancaremos y desde el primero podemos usar el menú “Open” del “Cluster Administrator” y ahí seleccionar la opción “Add Nodes to Cluster”. Ahí elegiremos nuestro segundo nodo (**BLAS**) y el asistente se encargará de hacerlo formar parte del sistema. Aseguraos eso sí de que la configuración es idéntica, doble tarjeta de red, discos compartidos,...

### Añadiendo nuevos discos al sistema

Para poder instalar nuevos servicios, será necesario que dotemos al cluster de nuevos discos, en ellos, como comentábamos antes, será donde instalemos nuestro SQL Server 2005.

Si en lugar de una máquina virtual esto se hace en un entorno real, no olvidéis seguir las recomendaciones sobre tipo de almacenamiento y ficheros a introducir. Se trata de maximizar el ancho de banda que sea capaz de proporcionar una cabina de discos y de minimizar los tiempos de espera de IO debidos a escrituras, por lo que lo más conveniente es situar los ficheros de log en discos en RAID 1+0 o 0+1. Los ficheros de datos ya será una cuestión de economía; lo que importa son las lecturas, que éstas sean rápidas, por lo que un RAID 5 puede ser suficiente.

En cualquier caso y volviendo a nuestra máquina virtual, añadiremos un nuevo disco con tamaño fijo a través del menú de Virtual Server asegurándonos que el disco sea un disco de longitud fija.



Figura 8

A continuación entraremos en la configuración de la máquina a la que le vamos a instalar el nodo y le añadiremos un nuevo dispositivo SCSI compartido para cluster y con el mismo SCSI ID que le determinamos en la primera configuración (en nuestro caso al nodo EPI le pusimos el ID 6 y al nodo **BLAS** el ID 6 también).

Añadiremos a continuación nuestro disco en el nuevo bus SCSI y arrancaremos nuestra máquina (apagando posteriormente el otro nodo para repetir los pasos). Al entrar, no veremos cambios, hay que acceder al administrador de discos (para los que no lo encontréis, botón derecho sobre “My Computer”, y luego “Manage”). Al entrar inmediatamente nos aparece el asistente para configurar el disco. Seguiremos las opciones por defecto y después lo particionaremos y formatearemos para que esté listo para ser usado.

Repetiremos los pasos de configuración en el segundo nodo, añadiendo tanto el nuevo dispositivo SCSI como el disco compartido en nuestro segundo nodo (tiene que estar apagado para que Virtual Server nos permita hacer cambios en la configuración).

En este instante, una vez arrancadas ambas máquinas estamos en disposición de configurar una aplicación (así se llama en el entorno de cluster) en el que al final vivirá SQL Server. Hay que meter el disco como recurso de cluster, esto implica un nombre virtual, una dirección IP virtual y el disco recién creado, para que la rutina de instalación de SQL detecte el recurso y se instale en esta unidad. Para conseguir esto nos situaremos en el “Cluster Administrator”, botón derecho encima del nombre de nuestro cluster, “Configurar nueva aplicación” y seguimos los pasos del asistente, añadiendo como recurso el disco que acabamos de crear y formatear.

El resultado después del asistente será algo como queda en el siguiente gráfico.



Figura 9

Hecho esto, estamos en disposición de comenzar la instalación de nuestro SQL 2005 en cluster. Tendremos que asegurarnos que ambos nodos están arrancados y funcionando antes de comenzar la instalación de SQL Server (aseguraos de poner online el grupo).

### Instalando SQL Server 2005 en cluster

Después, la tarea es bastante sencilla. Para la instalación hemos usado la versión Beta 2 de SQL Server

2005, aunque a fecha de la escritura del artículo ya tenemos la CTP de junio, pero hemos preferido instalarlo con Beta 2 ya que es potencialmente a la que mayor acceso ha habido.

Al ejecutar la instalación los primeros pasos son exactamente iguales en un entorno de cluster que en un entorno no “clusterizado”. Cuando llega el momento en que el programa de instalación nos solicita los componentes a instalar es cuando tenemos que especificar que se instale como un servidor virtual, para nuestro caso vamos a hacer esa operación tanto para el motor relacional como para el motor de OLAP. En la figura no aparece seleccionado *Reporting Services*; esto es debido a que no hemos instalado los servicios de IIS en las máquinas virtuales.



Figura 10

En el siguiente paso del asistente nos aparecerá si queremos instalar en la instancia por defecto o con nombre de instancia, en nuestro caso elegiremos instalar con nombre de instancia, no es necesario pero si al final acabásemos instalando un cluster en *activo-activo* queda bastante más elegante que ambos servidores tengan instancias con nombre que si uno es la instancia por defecto y el otro tiene un nombre de instancia.

A continuación nos solicita el nombre de nuestro servidor virtual, siguiendo la línea de Barrio Sésamo vamos a nombrarlo *Triqui*, así tenemos prácticamente entero el barrio.



Figura 11

El siguiente paso es uno de los más importantes en esto del clustering. Ahí nos va a solicitar la dirección IP virtual, es decir, a través de la cual accederán a nuestro servidor virtual todos los clientes y que podrá estar activa en un nodo o en el otro resultando de esta forma transparente a los usuarios si se están conectando a nuestro servidor *EPI* o a nuestro servidor *BLAS*. En este caso hemos elegido una configuración de direcciones IP muy redondeadas (vamos a elegir la **192.168.1.50** como dirección virtual), en los entornos de producción esto puede depender de la política de red, evidentemente es transparente, solamente necesita tener una dirección IP fija para ser accedido, sin necesidad de que esa dirección cumpla otro requisito (bueno tiene que estar libre por razones obvias).

A continuación nos solicita los nodos en los que va a poder residir nuestro servicio, en nuestro ejemplo sólo hay dos nodos, uno de ellos es obligatorio, y el otro aparece seleccionado por defecto, en un entorno con muchos nodos podemos decidir que un servicio no pueda residir en un servidor, esto dependerá de la configuración que deseemos, del número de almacenamientos disponibles etc.

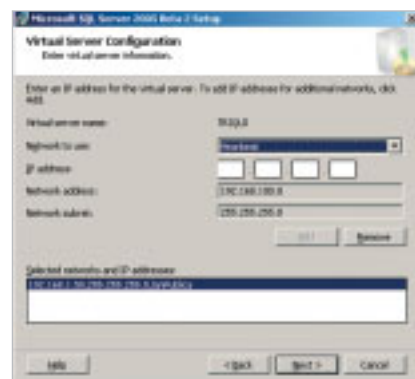


Figura 12

En la figura puede verse cómo hemos configurado la *lanPublica* para que sea el método de acceso, y no vamos a configurar la red de *heartbeat* que para nuestro caso será privada entre ambos servidores (en entornos de producción algunas veces se trata de un cable cru-

## La parte más compleja de todo el asunto, más que en el propio SQL Server, está en el sistema operativo

zado de una a otra máquina para garantizar la disponibilidad).

Una vez pasado este proceso, el instalador de SQL Server aparece ligeramente distinto que en una instalación que no vaya contra un cluster, ya que en la parte superior permite elegir el nodo y ver cómo va el progreso de la instalación en cada uno de los nodos.

Acabado el asistente tendremos nuestro servidor SQL Server 2005 instalado en HA, un cluster recién instalado. Ya podremos empezar con las pruebas. En el caso de los desarrolladores puros, nuestras aplicaciones deberían superar al menos las siguientes:

1. Que se mueva el servidor de nodo en medio de una transacción.
2. Que se mueva el servidor de nodo en un momento de inactividad.

Todas estas operaciones, que suponen cortes en el servicio deberíamos ser capaces de que pasasen inadvertidas al usuario (no a las aplicaciones naturalmente) y deberíamos recoger en un log todos esos datos. Notificar al *Event Viewer* es una opción bastante adecuada, sin embargo, los usuarios de aplicaciones no están consultando el *Event Viewer* habitualmente, así que podrían establecerse mecanismos adicionales de control sobre las aplicaciones.

## Conclusiones

Gracias a las tecnologías de máquinas virtuales tenemos un mecanismo asequible de probar tecnologías realmente complicadas de probar debido al gran coste económico que suponen.

La parte más compleja de todo el asunto, más que en el propio SQL Server, está en el sistema operativo, en la instalación y configuración de dispositivos de hardware y en conocer unos cuantos conceptos.

Además SQL 2005 mantiene la sencillez de instalación en entornos “clusterizados” de su antecesor SQL 2000 simplificando además la instalación de *Analysis Services* en este tipo de estructuras. ○





Luis Miguel Blanco

# Integración del CLR con SQL Server 2005

## Ejecutando código administrado desde el núcleo del motor de datos

De todo el conjunto de novedades que se incluyen con SQL Server 2005, la que probablemente llame la atención de forma más poderosa a la comunidad de programadores, sea la capacidad de escribir el código de ciertos objetos del motor de datos utilizando alguno de los lenguajes de la plataforma .NET Framework (Visual Basic .NET, C#, etc.), aspecto éste bautizado con el nombre de “Integración del CLR con SQL Server 2005”.

### » ¿Qué es la integración del CLR en SQL Server 2005?

Una de las primeras cuestiones que deberíamos preguntarnos sería “¿En qué consiste o qué debemos entender por integración del CLR en SQL Server 2005?”. La definición que podemos dar acerca de este nuevo aspecto de nuestro conocido gestor de datos sería la siguiente: “se trata de aquella característica de SQL Server por la cual, el CLR (*Common Language Runtime*), es decir, el motor de ejecución de la plataforma .NET Framework, se encuentra albergado dentro de la maquinaria del servidor de datos como un elemento más, permitiendo que SQL Server pueda beneficiarse de toda la potencia que brinda la tecnología .NET al desarrollador”.

### ¿Qué beneficios aporta esta integración?

El objetivo principal de incorporar el CLR a la arquitectura de SQL Server, consiste en dotar al programador de aplicaciones de bases de datos de un conjunto de herramientas que potencien su trabajo, entre las cuales se encuentra la posibilidad de emplear cualquiera de los lenguajes de la plataforma .NET, para escribir los siguientes objetos del motor de datos: procedimientos almacenados, funciones definidas por el usuario, tipos definidos por el usuario, agregados y triggers.

Los mencionados objetos, hasta este momento, sólo se podían crear utilizando Transact-SQL (T-SQL), o en el caso de procedimientos almacenados, mediante el API de programación de procedimientos almacenados extendidos. Sin embargo, a partir de ahora, podemos ampliar

su potencia gracias a la rica jerarquía de clases que ofrece .NET Framework para su programación; con ella, es posible desarrollar aspectos antes impensables debido a las limitaciones de T-SQL, y obtener así las ventajas de la ejecución bajo el entorno de código administrado (*managed code*) de .NET.

Por otro lado, esta capacidad de integración significa que a partir de ahora, podemos escribir código administrado para SQL Server utilizando Visual Studio .NET, el mismo entorno de desarrollo que empleamos para el resto de aplicaciones habituales de esta plataforma, incrementando nuestra productividad gracias a la multitud de asistentes y ayudas para la programación, sobradamente conocidas por todos los usuarios de esta herramienta.

La seguridad es otro de los apartados favorecidos por la integración, ya que elementos como la *Seguridad de Acceso al Código* (CAS, *Code Access Security*), la seguridad a nivel de tipos, etc., harán que nuestro código administrado se ejecute de modo más seguro en SQL Server.

Con respecto a las versiones de los productos utilizadas para elaborar este artículo, se han utilizado SQL Server 2005 Beta 2 y Visual Studio 2005 Beta 2, los cuales aparecerán en sus correspondientes versiones definitivas a primeros de noviembre.

### Objetos administrados de base de datos

Como ya hemos apuntado anteriormente, la integración con el CLR permite escribir los conocidos procedimientos almacenados, funciones, triggers, etc., propios de T-SQL, empleando cualquiera de los len-

Luis Miguel Blanco es redactor de dotNetManía. Es consultor en Alhambra-Eidos. Ha escrito varios libros y decenas de artículos sobre la plataforma .NET ([alibrenadigital.com](http://alibrenadigital.com))

guajes de la plataforma .NET. Dentro de este contexto de desarrollo y ejecución, a estos elementos se les denomina *Objetos Administrados de Base de Datos* o *Managed Database Objects*; para abreviar, nosotros emplearemos a partir de ahora las siglas MDBO para referirnos a estos objetos.

Dependiendo de la naturaleza de cada uno de estos objetos su implementación varía, así pues, los procedimientos almacenados, funciones y *triggers* se escriben como métodos compartidos (*Shared* o *static*, según el lenguaje utilizado), mientras que los agregados y tipos definidos por el usuario se escriben como clases o estructuras.

## Acceso a datos desde objetos MDBO

En un alto porcentaje de ocasiones, cuando escribamos el código de un MDBO, este objeto necesitará acceder también a la información de la base de datos en la que se encuentra alojado.

Para facilitar nuestro trabajo en este sentido, el CLR proporciona al programador un proveedor de datos (*in-process data provider*), que se ejecuta integrado junto al código de nuestros objetos MDBO, dentro del proceso de SQL Server.

El conjunto de tipos que conforman este proveedor integrado de datos se encuentra en el espacio de nombres *System.Data.SqlServer*, que se halla dentro del ensamblado *Sqlaccess.dll*. Entre las clases contenidas por este proveedor de datos podemos mencionar las siguientes:

- **SqlCommand**. Permite la creación y ejecución de sentencias DDL y DML en la base de datos.
- **SqlDataReader**. Proporciona capacidades de navegación de sólo lectura y avance por las filas de un conjunto de resultados, obtenido a partir de una consulta contra la base de datos.
- **SqlPipe**. Es el objeto encargado de enviar a la aplicación cliente el resultado de la ejecución de un MDBO, en forma de conjunto de resultados, cadena de caracteres, etc.
- **SqlContext**. Representa el entorno o contexto dentro del cual se ejecutan los objetos MDBO. Proporciona objetos preconstruidos del proveedor de datos integrado, tales como *Connection*, *Command*, *Pipe*, etc., que aportan una mayor agilidad en la ejecución. Por ejemplo, si desde un MDBO necesitamos ejecutar un comando contra la base de datos en la que se encuentra instalado, no es necesario crear primeramente una conexión y luego un comando; llamaremos directamente al método *SqlContext.GetCommand()*, con el que obtendremos un objeto *Command*, cuya propiedad *Connection* ya está configurada para conectar con la base de datos en curso.

Como podrá comprobar el lector, el uso de estos objetos resultará muy familiar para aquellos programadores que ya hayan trabajado con el modelo de clases de

## El CLR proporciona al programador un proveedor de datos integrado junto al código de nuestros objetos MDBO, dentro del proceso de SQL Server 2005

ADO.NET, lo que facilitará la curva de aprendizaje con el proveedor de datos integrado de SQL Server.

Para una descripción en profundidad de todas las clases pertenecientes al espacio de nombres *System.Data.SqlServer*, recomendamos al lector la consulta de la documentación de SQL Server 2005.

## Creación de un procedimiento almacenado MDBO

Evidentemente, no todo en este artículo va a ser teoría y conceptos sobre la integración entre el CLR y SQL Server 2005. Durante los siguientes apartados también vamos a abordar, como es natural, la vertiente práctica en la creación de objetos MDBO; y lo vamos a hacer comenzando por los procedimientos almacenados, uno de los objetos más sobradamente conocidos y utilizados habitualmente en T-SQL.

Antes de comenzar, es necesario puntualizar que el código de los ejemplos de este artículo está desarrollado en VB.NET, estando disponible para su descarga en la dirección [www.dotnetmania.com](http://www.dotnetmania.com).

En primer lugar abriremos Visual Studio 2005 y seleccionaremos la opción de menú “File + New + Project”. En el cuadro de diálogo “New Project”, dentro del panel de tipos de proyecto, haremos clic sobre el nuevo tipo “Database”; como plantilla elegiremos “SQL Server Project”, la cual dispone por defecto de todos los ajustes necesarios para crear objetos MDBO. El nombre del proyecto será *ProcAlmacConsulta* como vemos en la figura 1.

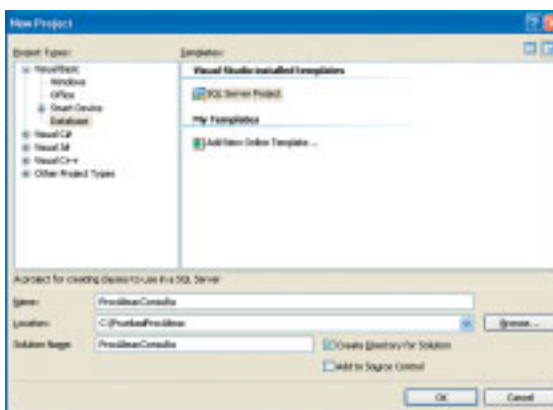


Figura 1

Una vez aceptado el cuadro de diálogo de creación de proyecto, la primera acción a realizar será establecer una conexión con un origen de datos, para lo cual, y de modo automático, Visual Studio mostrará el cuadro de diálogo de conexión a una base de datos, en el que introduciremos los valores necesarios para conectar con nuestro servidor SQL Server local, y utilizar la base de datos **Northwind**, empleando seguridad integrada de Windows, como vemos en la figura 2.

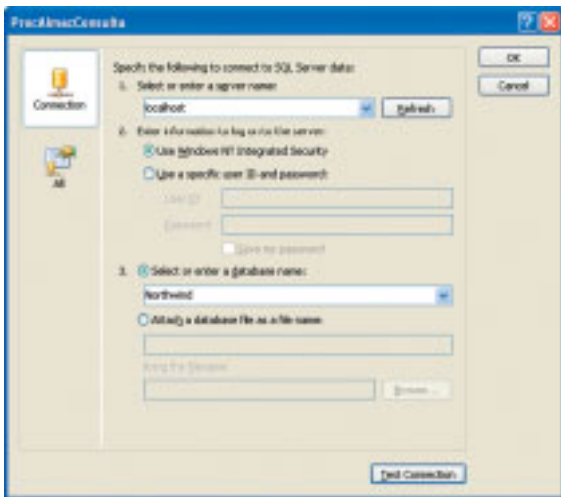


Figura 2

A continuación, mediante la opción de menú “Add + New Item...”, abriremos el cuadro de diálogo para añadir un objeto MDBO de tipo “Stored Procedure”, al que llamaremos **spProductosOrdenNombre**, como vemos en la figura 3.

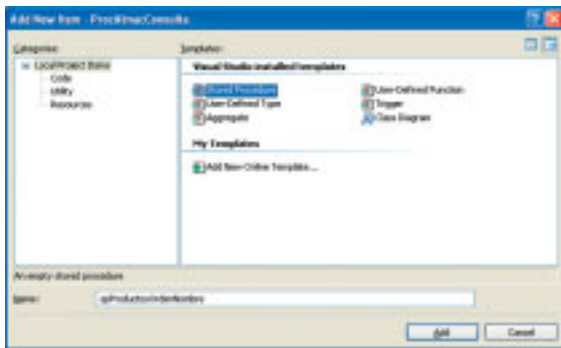


Figura 3

Al aceptar esta ventana, se creará una nueva clase con el nombre **StoredProcedures**, que tendrá un método con el nombre **spProductosOrdenNombre**. Llegados a este punto, debemos comentar algunas características del lenguaje referentes a esta clase:

- La clase está definida con la palabra clave **Partial**. Esta es una novedad incluida en los lenguajes de la plataforma .NET, que nos permite repartir los miembros de una misma clase en varios archivos de código. Hasta la presente versión de .NET, el

código de una clase tenía que escribirse en un único archivo.

- El método se define como **Shared** o compartido, para que SQL Server pueda ejecutarlo directamente sin tener que instanciar un objeto.
- A nivel de método se aplica el atributo **SqlProcedure**, que proporciona información a SQL Server acerca del tipo de objeto de base de datos que vamos a crear. De esta manera, en función del tipo de objeto a escribir, existen diferentes atributos que deberemos aplicar: **SqlFunction**, **SqlUserDefinedType**, etc.

Seguidamente escribiremos el método de la forma mostrada en el fuente 1.

```
Imports System
Imports System.Data
Imports System.Data.Sql
Imports System.Data.SqlClient
Imports System.Data.SqlTypes

Partial Public Class StoredProcedures
    <SqlProcedure() > _
    Public Shared Sub spProductosOrdenNombre()
        Dim oCommand As SqlCommand
        Dim oDataReader As SqlDataReader
        Dim oPipe As SqlPipe

        oCommand = SqlContext.GetCommand()
        oCommand.CommandType = CommandType.Text
        oCommand.CommandText = _
            "SELECT ProductID, ProductName, QuantityPerUnit " & _
            "FROM Products ORDER BY ProductName"

        oDataReader = oCommand.ExecuteReader()

        oPipe = SqlContext.GetPipe()
        oPipe.Send(oDataReader)
        oPipe.Send("Resultados enviados al cliente")
    End Sub
End Class
```

Fuente 1

En el código que acabamos de ver, utilizamos el objeto **SqlContext** para obtener una instancia de un objeto **SqlCommand**, que emplearemos para ejecutar una consulta contra la base de datos. Observemos que, gracias al diseño de las clases del proveedor de datos integrado, no es necesario crear un objeto **Connection**, ya que por defecto se asume que vamos a trabajar con la base de datos que hemos especificado al crear el proyecto; por lo tanto, dicha conexión ya se encuentra incluida en aquellos objetos que lo necesiten, como es el caso del comando.

Asignaremos al comando el texto de la consulta y la ejecutaremos, obteniendo el conjunto de resultados como un objeto **SqlDataReader**; hasta ahora nada nuevo para todo aquel que haya programado con ADO.NET. A partir de aquí es donde entramos en la parte más interesante, ya que necesitamos devolver el objeto **SqlDataReader** de alguna manera al cliente que ejecuta el procedimiento almacenado; esto lo



conseguiremos utilizando un objeto **SqlPipe**, que obtenemos también de **SqlContext**.

La clase **SqlPipe** dispone del método **Send()**, que será el empleado para enviar resultados al cliente. Gracias a las distintas sobrecargas de este método, es posible devolver información de diferente tipo; en este caso lo usamos para retornar un objeto **SqlDataReader**, y a continuación una cadena de caracteres.

### Compilación, despliegue y ejecución de un objeto MDO

Terminada la escritura del objeto vamos a proceder a ejecutarlo para probar su funcionamiento.

En primer lugar, mediante la opción de menú "Build + Build Solution" compilaremos el código, obteniendo el ensamblado **ProcAlmacConsulta.dll** en el directorio bin del proyecto. A continuación, la opción "Build + Deploy Solution" desplegará e instalará el ensamblado generado dentro de SQL Server.

Si queremos verificar que la instalación del objeto ha tenido éxito, abriremos SQL Server Management Studio y realizaremos las siguientes acciones: en la ventana "Object Explorer" nos situaremos sobre la base de datos **Northwind**, expandiremos el nodo "Programmability", y dentro de este, los nodos "Assemblies" y "Stored Procedures", que mostrarán respectivamente, el ensamblado y procedimiento almacenado que hemos creado desde Visual Studio, como vemos en la figura 4.

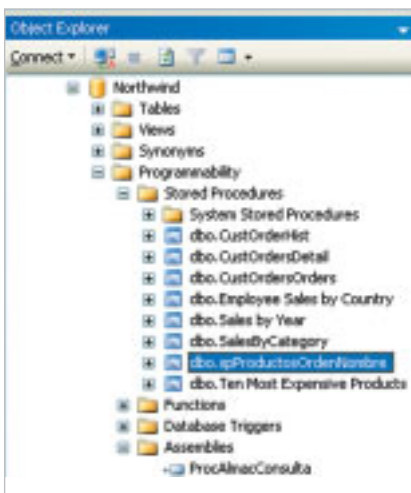


Figura 4

Para ejecutar nuestro procedimiento almacenado desde Visual Studio pulsaremos [F5], o el botón "Start" de la barra de herramientas, obteniendo el resultado en la ventana "Output", como vemos en la figura 5, en la que también señalamos cada uno de los elementos devueltos por el objeto **SqlPipe**. En el caso de que esta ventana no muestre la información deseada, debemos seleccionar, de su lista desplegable "Show output from", el valor "Database Output" (figura 5).

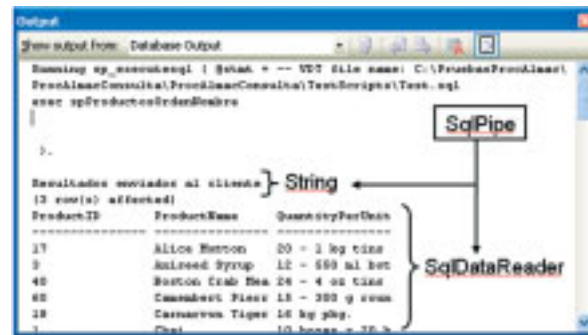


Figura 5

Lo que realmente hace Visual Studio cuando ponemos en funcionamiento el proyecto es ejecutar el archivo de script **Test.sql**, que contiene la instrucción **exec spProductosOrdenNombre**. Este archivo está situado en la carpeta **TestScripts** del proyecto, y es creado automáticamente cuando desarrollamos proyectos de tipo SQL Server. Según el elemento de nuestro proyecto que deseemos probar al ejecutar el proyecto, deberemos escribir la/s instrucción/es adecuadas en este archivo.

También podemos ejecutar este procedimiento almacenado desde SQL Server Management Studio, escribiendo la instrucción antes mencionada en una nueva ventana de consultas, en cuya parte inferior, las pestañas "Results" y "Messages", mostrarán respectivamente los datos devueltos por el procedimiento y la cadena enviada con el objeto **SqlPipe**.

### Agregar y registrar manualmente un ensamblado y su contenido en SQL Server

Aunque como acabamos de ver, el medio más sencillo y directo de instalar un ensamblado .NET en SQL Server 2005 es desde el IDE de Visual Studio 2005, quizá nos encontremos en algún momento con la necesidad de realizar esta operación, pero no dispongamos del mencionado entorno de desarrollo. ¿Qué hacemos entonces, nos quedamos sin instalar el ensamblado?.

No se preocupe el lector, ya que es perfectamente posible agregar manualmente el ensamblado y registrar su contenido en el servidor de datos, puesto que la nueva versión de SQL Server incorpora los comandos apropiados a tal efecto.

Vamos a comprobar esta característica de la integración con el CLR, eliminando primeramente desde SQL Server Management Studio los elementos creados en el apartado anterior. Para ello, en el nodo "Stored Procedures" haremos clic derecho sobre el procedimiento almacenado creado anteriormente, y seleccionando la opción de menú "Delete" lo eliminaremos de la base de datos. A continuación, en el nodo "Assemblies", procederemos exactamente igual con el ensamblado.

Posteriormente, para añadir un ensamblado manualmente a la base de datos, deberemos utilizar la instrucción **CREATE ASSEMBLY**, proporcionando la ruta en la que reside el archivo del ensamblado, y especificando el conjunto de permisos de seguridad que otorgamos al mismo, como vemos en el siguiente fuente:

```
CREATE ASSEMBLY ProcAlmacConsulta
FROM 'C:\EnsambladosCLR\ProcAlmacConsulta.dll'
WITH PERMISSION_SET = SAFE
```

VS 2005 proporciona una nueva plantilla para proyectos de SQL Server, y dentro de estos, plantillas para cada uno de los objetos de datos administrados.

## Todo proyecto VS de SQL Server se encuentra ligado a una base de datos sobre la que se desplegará el ensamblado resultante

La cláusula `WITH PERMISSION_SET` será la que empleemos para establecer el grupo de permisos asignados al ensamblado; dispone de tres valores que describimos seguidamente:

- **SAFE**. Representa el nivel con mayor grado de seguridad, ya que restringe el acceso del ensamblado a recursos tales como el registro, sistema de archivos, redes, etc. En el caso de no especificar ningún valor de seguridad para el ensamblado, se tomará por defecto este nivel.
- **EXTERNAL\_ACCESS**. Cuando establecemos este tipo de permisos, el ensamblado puede acceder a recursos como archivos, el registro, variables de entorno, etc.
- **UNSAFE**. El ensamblado tiene pleno acceso a todos los recursos. Al ser la opción menos restrictiva ha de emplearse con sumo cuidado.

### Depuración de objetos MDBO desde Visual Studio

Al igual que ocurre durante el desarrollo de otros tipos de aplicaciones, la creación de objetos administrados para SQL Server es una labor que no está exenta de errores. Es por ello que Visual Studio nos proporciona también la capacidad de depurar este tipo de código, aunque se trata de una característica que no está activada por defecto.

Para habilitar la depuración en un proyecto de tipo SQL Server debemos seguir estos pasos: abrir la ventana “Server explorer”, expandir el nodo “Data Connections”, hacer clic derecho sobre la conexión de datos que hayamos establecido para el proyecto, y finalmente, seleccionar la opción de menú “Allow SQL/CLR Debugging”. Aparecerá una ventana con el siguiente mensaje: “SQL/CLR debugging will cause all manager threads on the Server to be stopped. Do you want to continue?”, a la que deberemos responder afirmativamente, con lo que ya tendremos disponible la depuración del código en nuestro proyecto.

### Personalizando el conjunto de resultados a devolver

En el anterior ejemplo hemos podido ver que la clase `SqlPipe` es la encargada, dentro de un procedimiento almacenado, de devolver la información resultante al cliente en forma de objeto `SqlDataReader`. Aunque este

modo de trabajo será el adecuado en muchas ocasiones, es posible que nos encontremos con requerimientos adicionales a la hora de enviar la información.

Supongamos que vamos a crear un procedimiento almacenado MDBO con una consulta sobre la tabla `Employees`, y por un lado debemos traducir los valores del campo `TitleOfCourtesy`, mientras que por otra parte tenemos que devolver `FirstName` y `LastName` como un único campo. Daremos el nombre `spEmpleadosFormato` a este nuevo procedimiento almacenado.

En este caso no podemos retornar directamente el objeto `SqlDataReader` producto de ejecutar el comando, lo que debemos hacer es construir nuestra propia estructura de almacenamiento de datos, recorrer el `SqlDataReader` para manipular los campos necesarios, y devolver estos datos transformados utilizando el objeto `SqlPipe`.

Todo lo anterior lo conseguiremos definiendo un modelo de registro y campos personalizados. Para el registro utilizaremos la clase `SqlDataRecord`, mientras que para los campos emplearemos la clase `SqlMetaData`; en estos últimos crearemos un objeto por campo, y los añadiremos todos a un array, que posteriormente pasaremos al constructor de `SqlDataRecord`.

Antes de iniciar la manipulación de datos, indicaremos el comienzo del envío de resultados llamando al método `SqlPipe.SendResultsStart()`, cada vez que enviemos un registro personalizado llamaremos a `SqlPipe.SendResultsRow()`, y cuando hayamos terminado de procesar todos los registros ejecutaremos `SqlPipe.SendResultsEnd()`. Vea el fuente 2, que contiene el código necesario.

Los datos obtenidos al ejecutar el anterior procedimiento los podemos ver en la figura 6.

TituloCortesia	NombreEmpleado
Señorita	Nancy Davolio
Doctor	Andrew Fuller
Señorita	Janet Leverling
Señora	Margaret Peacock
Señor	Steven Buchanan
Señor	Michael Suyama
Señor	Robert King
Señorita	Laura Callahan
Señorita	Anne Dodsworth

Figura 6

### Desarrollo de un tipo definido por el usuario MDBO

A continuación vamos a abordar una característica, que si bien ya se encuentra presente en T-SQL, se ve potenciada gracias a la integración del CLR; nos referimos a la creación de tipos de datos propios (*User Defined Type*) para SQL Server, usando código administrado.

```

Partial Public Class StoredProcedures
    <SqlProcedure()> _
    Public Shared Sub spEmpleadosFormato ()
        Dim oCommand As SqlCommand
        Dim oDataReader As SqlDataReader
        Dim oPipe As SqlPipe
        Dim aMetadatos(1) As SqlMetaData
        Dim oRegistro As SqlDataRecord
        Dim sTitCortesia As String
        Dim sNombre As String

        ' crear comando, ejecutarlo y obtener DataReader
        oCommand = SqlContext.GetCommand()
        oCommand.CommandType = CommandType.Text
        oCommand.CommandText = "SELECT TitleOfCourtesy, FirstName, LastName " & _
            "FROM Employees"
        oDataReader = oCommand.ExecuteReader()

        ' definir metadatos y registro para devolver datos resultantes
        aMetadatos(0) = New SqlMetaData("TituloCortesia", SqlDbType.NVarChar, 15)
        aMetadatos(1) = New SqlMetaData("NombreEmpleado", SqlDbType.NVarChar, 40)
        oRegistro = New SqlDataRecord(aMetadatos)

        oPipe = SqlContext.GetPipe()

        ' indicamos que va a comenzar el envío de resultados
        oPipe.SendResultsStart(oRegistro, False)
        While oDataReader.Read()
            ' obtener valores de campos y modificarlos
            sTitCortesia = oDataReader.GetString(0)
            Select Case sTitCortesia
                Case "Ms."
                    sTitCortesia = "Señorita"
                Case "Dr."
                    sTitCortesia = "Doctor"
                Case "Mrs."
                    sTitCortesia = "Señora"
                Case "Mr."
                    sTitCortesia = "Señor"
            End Select

            sNombre = oDataReader.GetString(1) & " " & oDataReader.GetString(2)

            ' pasar valores modificados a registro
            oRegistro.SetString(0, sTitCortesia)
            oRegistro.SetString(1, sNombre)

            ' enviar registro al cliente
            oPipe.SendResultsRow(oRegistro)
        End While
        ' aquí termina el envío de resultados
        oPipe.SendResultsEnd()
    End Sub
End Class
    
```

Fuente 2

Los tipos de usuario MDBO nos permiten extender los tipos de datos nativos de SQL Server, y entre sus características podemos destacar las siguientes:

- Se deben codificar como una clase o estructura.
- Permiten la creación de tipos de datos complejos sin la necesidad de estar basados en un tipo nativo de SQL

Server; en este sentido aportan mayor flexibilidad que los tipos de usuario tradicionales de T-SQL.

- En función de lo expresado en el anterior punto, estos tipos se ejecutan, manipulan y almacenan como objetos reales, lo cual permite al código cliente que los utiliza, llamar a sus métodos y manipular sus propiedades.

## Pedido, un nuevo tipo de datos para SQL Server

Partiendo de las anteriores premisas, nuestro siguiente ejemplo va a consistir en realizar el desarrollo de **Pedido**, un nuevo tipo de dato MDBO. Este tipo estará basado en el supuesto de una empresa de transportes, que distribuye pedidos de material a diversos puntos del país. Por cada pedido necesita registrar la descripción del material, y la distancia recorrida en kilómetros hasta el punto de destino, para poder calcular, en base a una tarifa de desplazamientos, el importe a facturar al cliente por el envío del pedido.

En primer lugar, crearemos en Visual Studio un nuevo proyecto de tipo SQL Server Project, con el nombre **TiposUsr**. Acto seguido tendremos que establecer la referencia con la base de datos asociada al proyecto; como ya creamos una conexión hacia **Northwind** en el anterior proyecto, Visual Studio nos ofrece una caja de diálogo con las bases de datos disponibles. En nuestro caso volveremos a elegir la misma, aunque siempre podemos pulsar el botón “Add Reference”, y abrir la ventana de selección para establecer referencia con base de datos diferente. Ver la figura 7.

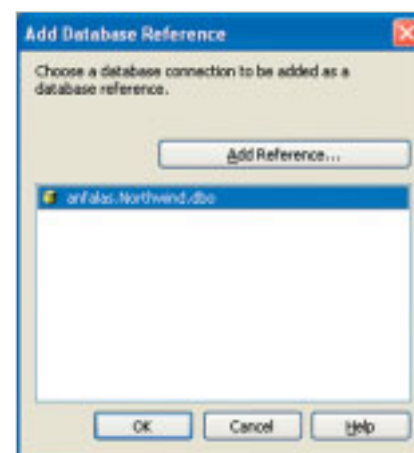


Figura 7

A continuación, agregaremos al proyecto un elemento de tipo “User-Defined Type” al que llamaremos **Pedido**. El entorno de desarrollo creará entonces una clase con ese mismo nombre y el esqueleto básico de la misma, a



la que deberemos añadir nuestro propio código para completar su funcionalidad. El código al completo podemos verlo en el fuente 3.

De todo el código necesario para crear el tipo Pedido destacaremos los siguientes aspectos:

- Aplicamos a nivel de clase los atributos `Serializable` y `SqlUserDefinedType`, para controlar el modo de serialización del tipo.
- Implementamos el interfaz `INullable`, y por consiguiente, las propiedades `IsNull` y `Null`, para que la clase disponga de la capacidad de manejar valores nulos.
- Al igual que las clases nativas de la plataforma .NET soportan la conversión a/desde cadenas, en nuestra clase creamos los métodos `ToString()` y `Parse()`, en los que desarrollamos la lógica necesaria para disponer también de esta característica.

Finalizada la escritura de esta clase, la compilaremos e instalaremos en SQL Server. Para trabajar con este nuevo tipo de dato, iniciaremos SQL Server Management Studio y nos situaremos sobre la base de datos `Northwind`; seguidamente abriremos una nueva ventana de consultas, en la que ejecutaremos las instrucciones mostradas en el fuente 4.

Observe el lector que en los ejemplos del anterior código, al declarar una variable de tipo `Pedido`, el acceso a sus miembros se realiza usando la sintaxis habitual del operador punto (`.`), con excepción del método `Parse()`, para el que utilizamos el operador dos puntos (`::`). Esto se debe a que `Parse()` es un método calificado como `Shared`, y dadas las características de este tipo de métodos, tendremos que emplear el mencionado operador cuando los invoquemos desde SQL Server.

Siguiendo con el método `Parse()`, dado que es posible que la cadena que le pasamos como parámetro no tenga el formato esperado, por ejemplo: `Pedido::Parse('Grabadoras DVD-abc-65')`, creamos una excepción con un mensaje personalizado, de forma que cuando se produzca un error por este

```
<Serializable(> _
<SqlUserDefinedType(Format.SerializedDataWithMetadata, MaxByteSize:=512)> _
Public Class Pedido
    Implements INullable
    Private bIsNull As Boolean
    Private sDescripcion As String
    Private nDistancia As Integer
    Private nTarifa As Integer

    Public Property Descripcion() As String
    Get
        Return sDescripcion
    End Get
    Set(ByVal value As String)
        sDescripcion = value
    End Set
    End Property

    Public Property Distancia() As Integer
    Get
        Return nDistancia
    End Get
    Set(ByVal value As Integer)
        nDistancia = value
    End Set
    End Property

    Public Property Tarifa() As Integer
    Get
        Return nTarifa
    End Get
    Set(ByVal value As Integer)
        nTarifa = value
    End Set
    End Property

    Public ReadOnly Property IsNull() As Boolean Implements INullable.IsNull
    Get
        Return bIsNull
    End Get
    End Property

    Public Shared ReadOnly Property Null() As Pedido
    Get
        Dim oPedido As Pedido = New Pedido()
        Return oPedido
    End Get
    End Property

    Public Overrides Function ToString() As String
    Dim sResultado As String
    If Me.IsNull Then
        sResultado = "NULL"
    Else
        sResultado = sDescripcion & "-" & CType(nDistancia, String) & "-" & _
            CType(nTarifa, String)
    End If

    Return sResultado
    End Function

    Public Shared Function Parse(ByVal s As SqlString) As Pedido
    Dim oPedido As Pedido
    Dim aElementos() As String
    Try
        If s.IsNull Then
            Return Pedido.Null
        Else
            oPedido = New Pedido()
            aElementos = s.ToString().Split("-"c)

```

Fuente 3 (sigue...)

```

oPedido.sDescripcion = aElementos(0)
oPedido.nDistancia = CType(aElementos(1), Integer)
oPedido.nTarifa = CType(aElementos(2), Integer)
Return oPedido
End If
Catch ex As Exception
Throw New Exception("Error al convertir")
End Try
End Function

Public Function ImporteFacturar() As Integer
Dim nResultado As Integer
nResultado = nDistancia * nTarifa
Return nResultado
End Function
End Class

```

(...continuación) Fuente 3

motivo, podamos identificarlo adecuadamente. Ver la figura 8.

mas de diseño y rendimiento a la hora de desarrollar aplicaciones de gestión de datos.

La integración del CLR es una excelente característica que mejora enormemente el gestor de datos, pero Transact-SQL sigue existiendo, y ambos elementos ocupan un sitio necesario en SQL Server; en ningún momento se pretende que la posibilidad de escribir código administrado en una base de datos sea su sustituto.

A la hora de plantear el uso de una u otra característica, debemos sopesar las ventajas e inconvenientes de cada una dentro del escenario de ejecución bajo el cual van a trabajar, y elegir la que proporcione un rendimiento más eficaz.

Dentro del modelo de desarrollo basado en *n-capas*, la integración del CLR difumina el límite existente entre la capa de datos y la capa de lógica de negocio, ya que ahora podemos integrar, si así lo queremos, una gran parte de la lógica de proceso en el interior del almacén de datos.

Cuanto más código de negocio traslademos a la capa de datos, dicho código tendrá un acceso más veloz a los datos, lo

```

/* crear objeto de tipo Pedido, asignar valores
y llamar a sus métodos */
DECLARE @oP Pedido
SET @oP.Descripcion = 'Tarjetas Sintonizadoras TV'
SET @oP.Distancia = 214
SET @oP.Tarifa = 87
SELECT @oP.ImporteFacturar()
SELECT @oP.ToString()

/* crear objeto a partir de una cadena mediante
el método compartido Parse */
DECLARE @oP Pedido
SET @oP = Pedido::Parse('Grabadoras DVD-384-65')
SELECT @oP.Descripcion AS DESCRIPCION,
@oP.Distancia AS DISTANCIA,
@oP.Tarifa AS TARIFA,
@oP.ImporteFacturar() AS IMPORTE

/* crear tabla con un campo de tipo Pedido y añadir
filas, hemos de utilizar una variable del mismo tipo
para almacenar los valores a insertar en dicho campo*/
CREATE TABLE MisPedidos (
IDPedido int NOT NULL,
DatoPedido Pedido NULL)

DECLARE @oPedido Pedido

SET @oPedido.Descripcion = 'Monitores TFT 17'
SET @oPedido.Distancia = 125
SET @oPedido.Tarifa = 6
INSERT INTO MisPedidos VALUES (1,@oPedido)

SET @oPedido.Descripcion = 'Micros P.III'
SET @oPedido.Distancia = 95
SET @oPedido.Tarifa = 4
INSERT INTO MisPedidos VALUES (2,@oPedido)

/* consulta simple contra la tabla */
SELECT IDPedido,
DatoPedido.Descripcion AS DESCRIPCION,
DatoPedido.Distancia AS DISTANCIA,
DatoPedido.Tarifa AS TARIFA,
DatoPedido.ImporteFacturar() AS IMPORTE
FROM MisPedidos

/* obtener el campo Pedido de una de las filas de la
tabla y llamar a un método del tipo */
DECLARE @oPed Pedido

SELECT @oPed = DatoPedido
FROM MisPedidos
WHERE IDPedido = 1

SELECT @oPed.ImporteFacturar() AS RESULTADO

```

Fuente 4

cual quiere decir que se mejorará la velocidad de proceso de nuestro código. No obstante, esto puede ser un arma de doble filo, ya que situar demasiado código en la capa de datos, puede penalizar seriamente el rendimiento y las posibilidades de extensibilidad de nuestro diseño.

Por todo ello, debemos hacer un uso racional de la integración, realizando un análisis de nuestros requerimientos, y aplicándola en aquellas situaciones que resulte conveniente. ○

## La clase `SqlPipe` dispone del método `Send()`, que permite devolver resultados de distinto tipo al cliente

Por último, cuando utilizamos `Pedido` como campo de una tabla, dado que se trata de un tipo complejo, una consulta del estilo `"SELECT * FROM MisPedidos"` daría error, por lo que hemos de ser más precisos, indicando a qué propiedad o método del objeto contenido en el campo queremos acceder, mediante la sintaxis `NombreCampo.Miembro`.

## Algunas reflexiones sobre la integración del CLR como conclusión

A lo largo del presente artículo hemos repasado algunos de los principales aspectos de la integración del CLR en el nuevo SQL Server 2005. Ahora bien, el conjunto de características aquí expuestas, no se traducen en que esta nueva funcionalidad de SQL Server sea el remedio que vaya a solucionar todos nuestros proble-

```

Msg 4021, Level 16, State 2, Line 2
A .NET Framework error occurred during execution of user defined routine or aggregate "Pedido":
System.Exception: Error al convertir
at TiposUsr.Pedido.Parse(String s)
at SQLCLR_Usr.ImporteFacturar()

```

Figura 8



Marino Posadas

# Entrevista a César Galindo-Legaria

**César Galindo-Legaria** es el responsable de la *Optimización de Consultas*, del grupo de *Relational Engine* en el equipo de desarrollo de SQL Server 2005 en Microsoft Corp. Tuvimos la ocasión de charlar con él con ocasión de su visita a España como ponente del SQLU Summit celebrado recientemente en Madrid.

>> **SQL Server 2005** es un producto muy grande y además han transcurrido 5 años desde la versión anterior. ¿Cuál es tu papel en el equipo de desarrollo de SQL-Server 2005?

SQL Server es un producto grande y es un equipo grande, en total trabajamos como unas 800 personas. Tiene muchos componentes diferentes: hay componentes de OLAP, de *Data Transformation*, herramientas, interfaces de usuario... Yo trabajo en el grupo *Relational Engine*, que es de procesamiento básico de consultas SQL y almacenamiento de datos. Dentro de este grupo, yo soy el *manager* de optimización de consultas del usuario.

O sea, que esos planes de ejecución que vemos cuando montamos una *query*...

Sí, esos son los que mi equipo genera. Y personalmente he trabajado en ese grupo de desarrollo

durante 10 años, desde antes de la versión de SQL Server 7.0.

Luego siempre has estado vinculado al área de datos y consultas...

Sí, datos, consultas, planes de ejecución eficiente..., etc.

¿Cómo es que el producto ha sufrido tantos retrasos (si es que los ha sufrido de verdad)? ¿O han sido más bien cambios de dirección en la política o el contenido del producto? Cinco años parece mucho tiempo...

Cinco años es muchísimo tiempo. Hubo varios elementos ahí: Uno son los retrasos. Nosotros habíamos planeado para 3 años, pero los planes no se ajustaron a la realidad; Otro son los imprevistos, por ejemplo, algunos problemas de seguridad con SQL Server 2000, en el que hubo que efectuar revisiones de segu-



César Galindo-Legaria

**Marino Posadas** es asesor técnico y redactor de dotNetManía, MVP de C# y formador de Alhambra-Eidos





César Galindo-Legaria y Marino Posadas

ridad, comprobaciones de código, agregar características nuevas, etc.

O sea, que a vosotros os afectó aquel famoso parón de dos o tres meses que hubo cuando todos os tuvisteis que estudiar el “Writing Secure Code” de **Howard y LeBlanc**?

Efectivamente (risas)... y ahí hubo personas y equipos de desarrollo que ya estaban preocupadas desde hacía un año de que esto podía pasar y estuvimos viendo qué podíamos hacer, cómo responder, qué cambios hay que hacer en el proceso de desarrollo, y demás. Después vino el virus *slammer* y eso supuso incluso la necesidad de hacer cambios en el desarrollo de SQL Server 2000, cambios para los *service packs*...

Asentemos primero lo que ya está en producción con los *service pack* y luego ya seguiremos con los nuevos desarrollos...

Eso es; y luego hay una tercera parte: es la integración tan cercana con el CLR y con Whidbey y el hecho de que estábamos tomando una dependencia de una tecnología que estaba todavía desarrollándose. Esa parte ha sido difícil de manejar para los dos lados, porque teníamos una dependencia mutua. Coordinar la entrega de código y de betas... No pensábamos que fuese a ser tan complicada como ha resultado. Pero lo que tenemos claro es que después de SQL Server 2005 viene SQL Server 2007, estamos haciendo los ajustes necesarios para que esto sea así. Ahí aprendimos varias lecciones prácticas sobre cómo abordar estos problemas.

¿Quizá porque la salida de Longhorn Server se espera para 2007? No olvidemos que se trata de una versión de servidor y su integración con otros servidores es fundamental...

la propia Microsoft. ¿Vosotros seguís las normas establecidas por MSF?

No. En los grupos más grandes de desarrollo en Microsoft (Windows, Office y SQL-Server), estamos tomando tiempo y analizando activamente los procesos que necesitamos seguir para que la construcción del software sea más predecible, en cuanto a tiempos, calidad y defectos. Y también que el producto mismo sea más robusto. Para ello existen comités de tipo general y grupos de trabajo más locales, llamados *engineer excellence*, que estudian qué es lo que ha fallado, qué es lo que ha funcionado y qué ajustes se pueden hacer. Después hacen reuniones de consolidación para extraer pautas a seguir por todos los grupos en general y ver las diferencias operativas de cada grupo.

De hecho, en el último año de desarrollo con Yukon, hemos estado usando algunas de las conclusiones a las que se han llegado, y en la transición entre Yukon y la siguiente versión, utilizaremos plenamente todas estas propuestas.

---

## ...lo que tenemos claro es que después de SQL Server 2005 viene SQL Server 2007, estamos haciendo los ajustes necesarios para que esto sea así

---

El calendario de *release* de los otros servidores es importante, pero estamos tomando una actitud conservadora respecto a tomar dependencias mutuas con respecto a tecnologías que no estén ya hechas. Así que quizá sea bueno que salgan más o menos en la misma fecha pero que no haya dependencia entre los dos. Que cualquier dependencia sea con tecnología que ya esté en ciclo de producción.

A propósito de ciclos de producción, la creciente popularización de MSF como marco de trabajo para controlar el ciclo de vida de las aplicaciones, y su inclusión como parte del MOC (*Microsoft Official Curriculum*), ha generado más de una vez preguntas acerca de la utilización que de éste marco hace

Es imperativo que la programación se vea como una forma más de ingeniería en la que existen normativas bien claras y establecidas de cómo deben de hacerse las cosas.

He leído que los retrasos de Longhorn eran debidos a los retrasos de SQL Server ¿Esto es cierto?

**Paco Marín:** O al revés, yo he leído que Longhorn se estaba retrasando por el retraso de WinFS y éste por el de SQL Server 2005 y por eso, incluso se dijo que la primera versión de Longhorn saldría sin WinFS ¿Es cierta alguna de las dos afirmaciones?

Dependiendo de a quien le preguntéis yo creo que obtendréis diferentes versiones, echándose la culpa unos a otros.

Lo que sí es objetivo es que varias partes de Longhorn tenían dependencia nuestra; la misma dependencia estrecha que teníamos nosotros con Whidbey. Son grupos grandes, de tecnologías complejas, y hacer que la tecnología funcione bien entre sí iba a llevar esfuerzos y en cierto momento hubo que cambiar los planes ahí... Hay cuestiones básicas a considerar, como cuáles con los servicios de un sistema operativo que pueden depender de código administrado y cuáles no, y todo ello ha conllevado retrasos porque se trata de una tecnología muy novedosa y muy interrelacionada.

En 2001, hablando con **Tony Goodhew** sobre las BB.DD. orientadas a objetos, él mencionaba que la tecnología no parecía interesante en ese momento. Tres años más tarde, **Andrew Conrad**, del equipo de desarrollo de *Object Spaces*, indicaba justamente lo contrario. ¿Cómo está la situación en este momento?



con el CLR está tomada desde el año 1999 ¿antes que saliera SQL Server 2000!?

No sé exactamente la fecha, pero es cierto que esto estaba presente desde el principio de la planificación de Yukon. Lo que ha pasado es que ha supuesto más esfuerzo de lo previsto.

seguir considerándolos como independientes.

¿Tú crees que la curva de aprendizaje para SQL Server 2005 va a ser para “tener que ponerse las pilas”?

Hay nuevas características del producto que van a requerir bastante trabajo. Una de ellas será la integración con el CLR, otra será integración activa de XML, y probablemente algunas otras más pequeñas, pero estas dos son las más grandes, desde el punto de vista del *engine*. Luego, Reporting Services y otras herramientas han cambiado y otras son nuevas, pero desde mi punto de vista, del *engine*, los cambios mayores son los citados.

¿Y crees que el gran público utilizará desde el principio los editores de *Business Intelligence*, que hemos podido ver en la Beta 2?

Ahí lo que no estoy seguro es cuáles son los modelos que los usuarios ya tienen y esperan incorporar con más facilidad a sus sistemas. Y qué tanto corresponde a las nuevas interfaces y herramientas y qué parte es heredada de sistemas anteriores. Ahí veo yo la dependencia. Respecto a Reporting Services, sin embargo, el asunto estaba claro.

¿Crees que la integración de XML realmente es muy importante para el público? y, ahora que sé a qué te dedicas, la existencia de un nuevo tipo de datos os tiene que haber afectado bastante en vuestro trabajo ¿no?

La implementación de la integración de XML llevó bastante tiempo y a pesar de ser una primera versión funciona bastante bien. Es muy claro que como formato de intercambio de datos

---

...varias partes de Longhorn tenían dependencia nuestra; la misma dependencia estrecha que teníamos nosotros con Whidbey. Son grupos grandes, de tecnologías complejas, y hacer que la tecnología funcione bien entre sí iba a llevar esfuerzos y en cierto momento hubo que cambiar los planes ahí...

---

Pues puede decirse que hay un movimiento a favor de esta última línea. Una de las motivaciones de eso es subir el nivel de abstracción de forma que permita la gestión de datos de manera sencilla, más rápida, etc. Pero sí es una tecnología que tendrá el impacto comercial que algunos piensan, sólo el tiempo podrá decirlo. Es un área importante para SQL Server, aunque no esté muy claro en este momento la forma que vaya a tomar como producto definitivo. Hace 10 años estaba **Object Store**, y otras que no llegaron a cuajar, quizá por el estado de la tecnología en ese momento.

**Paco Marín:** ¿Es cierto que la decisión sobre la integración de SQL Server

WinFS es un sistema de almacenamiento para Longhorn. Las tecnologías del nuevo sistema parecen estar tan claras, que hemos visto como dos de sus API principales (*Avalon e Indigo*), van a poder instalarse “hacia atrás”, en Windows XP y Windows 2003 y ya existen versiones beta que lo permiten, pero no hemos oído nada de la posibilidad de hacer lo mismo con WinFS. ¿Sabes por qué es esto?

Pues no sé realmente el camino que esto va a seguir. En realidad WinFS y SQL Server son caminos independientes. SQL Server está basado en un modelo relacional mientras que WinFs tiene componentes que están mucho más arriba, y es mejor

## La CTP 15 está planeada para principios de junio y no habrá una última CTP 16 como se esperaba, sino que iremos directamente a la versión RTM. A partir de la beta 2, las CTP están funcionando bastante bien

no hay duda que está ahí, tan claro como que almacenar XML como un nuevo tipo de datos y servirlo desde ahí era una necesidad. La integración se refiere más a consultas y transformación de XML dentro de la base de datos. Hay que ver cómo se recibe comercialmente y cómo se desarrolla, porque parte de la tecnología está ahí, tal vez no al 100% pero sí lo suficiente para que se empiecen a hacer cosas. El rendimiento de las bases de datos relacionales se da cuando más o menos se entiende el hecho de que haya regularidad en las estructuras. Cuando empezamos a manejar datos menos estructurados y esquema abierto y queremos buscar en cualquiera de los niveles..., todas estas operaciones sí se pueden implementar y sí se pueden optimizar, pero a mí me parece que la naturaleza del esquema abierto tiene un costo a la hora de la eficacia. Qué es lo que va a pasar, a fin de cuentas, no está muy claro para mí.

**Por cierto, cómo se almacenará el XML, ¿en texto, en binario?**

En binario, en un formato propio. De hecho otras bases de datos asociadas a XML, como **Tamino** de Software AG que almacenan en binario. Es una cuestión de rendimiento, asociada con el *streaming*<sup>1</sup>.

**¿Se puede anticipar ya una fecha de salida? ¿Y, sobre todo, están SQL Server o Visual Studio 2005 condicionados mutuamente de cara a su aparición oficial?**

Ambos saldrán en los últimos meses del año<sup>2</sup>.

**Siempre que sale un producto de este tipo, Microsoft hace un gran esfuer-**

**zo por convencer a los usuarios potenciales de la necesidad y las ventajas de la migración. Desde un punto de vista no comercial, ¿cuáles serían esas ventajas en tu opinión?**

Yo creo que hay tres partes en esta pregunta: Una es toda la funcionalidad nueva que hay en SQL Server y dentro de esa funcionalidad nueva hay cosas en OLAP, hay cosas en *Data Mining*, hay cosas en el relational engine como la integración con CLR y XML, hay database mirroring, hay *snapshot*... Hay muchas características



nuevas que creo que son valiosas para el desarrollo; otra, los cambios en SQL Server para un mejor soporte (ésta fue una de las tareas donde invertimos bastante). Toda la información que hay cuando algo va mal se guarda en unas tablas virtuales, no necesariamente para el usuario final, pero sí para el ingeniero de soporte; y la tercera parte, el rendimiento y la escalabilidad.

Nosotros hace ya que no publicamos datos mientras la competencia sí porque según las normas no pueden publicarse antes de 6 meses de anticipación a la disponibilidad; en los próximos meses publicaremos nuestros números y espero que sean buenos<sup>3</sup>.

**Paco Marín.** Lo cual quiere decir que aún quedan más de 6 meses para que salga al mercado (la entrevista se realizó a finales de abril, por tanto, debía ser en noviembre o diciembre).

Bueno ahí, ahí estamos (risas).

**Paco Marín.** ¿Las ediciones que se entregarán a partir de ahora hasta la puesta a la venta serán CTP o queda alguna beta más?

No. La CTP 15 está planeada para principios de junio y no habrá una última CTP 16 como se esperaba, sino que iremos directamente a la versión RTM (*Release To Manufacturing*), que es la versión final a falta de todo lo necesario para ponerla a la venta). A partir de la beta 2, las CTP están funcionando bastante bien.

**Paco Marín.** Existen rumores de que la CTP que tú llamas 14, la de abril para el público, llevará todas las funcionalidades y sólo queda depurar el código en las siguientes releases. ¿Qué hay de cierto en esto?

Básicamente es así. Tal vez haya algunos ajustes pendientes, pero, en cuanto a funcionalidad, la CTP de abril ya está completa. ○

<sup>1</sup> En la fecha de impresión de este número, ya sabemos que la fecha de lanzamiento será el 7 de noviembre de 2005.

<sup>2</sup> En el artículo de Eladio Rincón "Soporte XML en SQL Server 2005" puede leer acerca de comentarios de Michael Rys (Program Manager de XML en el equipo de SQL Server) sobre rendimiento en el pasado Tech-Ed 2005 de Orlando

<sup>3</sup> Los nuevos datos de TPCH y TPCC ya comenzaron a salir y sí son buenos. En cuanto los tengamos, los publicaremos.





Salvador Ramos

# Business Intelligence en SQL Server 2005

Vamos a echar un breve vistazo por todas las novedades que ha incorporado SQL Server 2005 para el desarrollo de proyectos de *Business Intelligence*. Se han producido cambios sustanciales en los *Servicios de Transformación de Datos* (DTS) y en los *Analysis Services* (AS), y se han incorporado nuevas herramientas de desarrollo, basadas en Visual Studio .NET.

» Ya en la versión anterior nos encontrábamos con herramientas para el desarrollo de proyectos de *Business Intelligence* (BI), pero es en esta nueva versión (SQL Server 2005) cuando tenemos a nuestra disposición una herramienta con gran cantidad de mejoras en cada uno de los servicios que tenemos disponibles, en su funcionalidad, y en las nuevas herramientas de desarrollo.

Como todos sabemos se ha producido en esta versión del producto un cambio sustancial; esto incluye también importantes cambios en los servicios y herramientas que nos ofrece para BI.

Estos son, básicamente, los componentes que nos proporciona SQL Server 2005 en ámbito del BI, y que trataremos en detalle más adelante:

- *SQL Server Reporting Services* (SSRS).
- *SQL Server Integration Services* (SSIS).
- *SQL Server Analysis Services* (SSAS).
- *Business Intelligence Development Studio* (BIDS)

Algunos son nuevos, otros se han mejorado, otros han sufrido grandes cambios, incluso los hay que han sido rediseñados en su totalidad y codificados de nuevo. Vamos a ir presentando cada uno de ellos, indicando cuál es su equivalente en SQL Server 2000, y su finalidad.

## SQL Server Reporting Services

Comenzaremos con la herramienta que nos proporciona para reporting empresarial, *SQL Server Reporting Services*, que aunque es una herramienta disponible en SQL Server 2000 desde hace un tiempo, realmente se ha diseñado para la versión 2005, aun-

que debido a la gran aceptación que tuvo por parte de los usuarios, Microsoft decidió anticipar su salida, sacando una versión para SQL Server 2000 sin costes de licencia adicionales.

Por el tema tratado en este artículo, no podemos pasar por alto una herramienta como esta, que nos va a permitir desplegar informes por toda nuestra organización, ofreciéndonos posibilidades de integración con otros productos como Sharepoint, para la inclusión de informes en nuestra intranet.

Pero como ya se han publicado algunos artículos sobre ella en esta revista, y los cambios que ha sufrido han sido mínimos, debido a las razones antes citadas, nos vamos a limitar a nombrarla sin más.

Como puede comprobar el lector en la interfaz gráfica todo resulta muy familiar, si ya conoce la versión anterior.

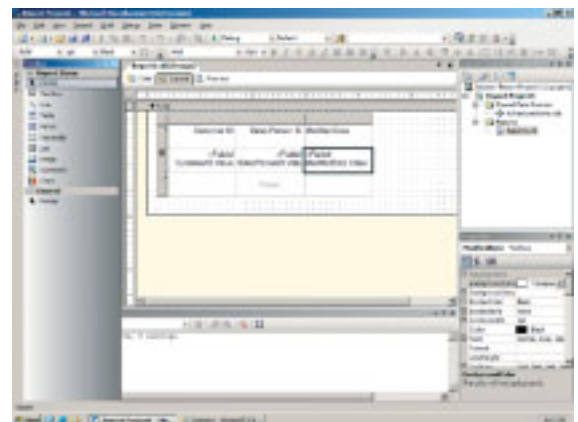


Figura 1

## SQL Server Integration Services (SSIS)

SQL Server Integration Services (SSIS) es la herramienta ETL (*Extract, Transformation and Loading*) que nos proporciona el producto. Es la evolución de los *Servicios de Transformación de Datos (DTS)*. Esta nueva herramienta ha sido rediseñada y desarrollada desde cero, utilizando la plataforma .NET. Va a tener disponible toda la funcionalidad que ya tenían los DTS, y va a incorporar gran cantidad de mejoras, aportando una buena cantidad de componentes para ETL y una nueva interfaz para el desarrollo de proyectos mucho más completa, basada en .NET. También ha sido mejorado su rendimiento y escalabilidad.

Como ya sabe el lector, para el almacenamiento y ejecución de paquetes DTS, se utilizaba el propio servicio de SQL Server; en la nueva versión, en cambio, hay un servicio totalmente independiente, que debe estar ejecutándose para poder almacenar y ejecutar los paquetes.

## Business Intelligence Development Studio

Para el diseño de paquetes, tenemos el *Business Intelligence Development Studio*, en el cual tenemos diferentes tipos de proyectos (reporting, análisis, y transformación-integración- de datos). Es importante destacar que en todos ellos dispondremos de un depurador, que tanto añoramos en la versión anterior, y que nos permitirá hacer un seguimiento exhaustivo de la ejecución de nuestros proyectos.

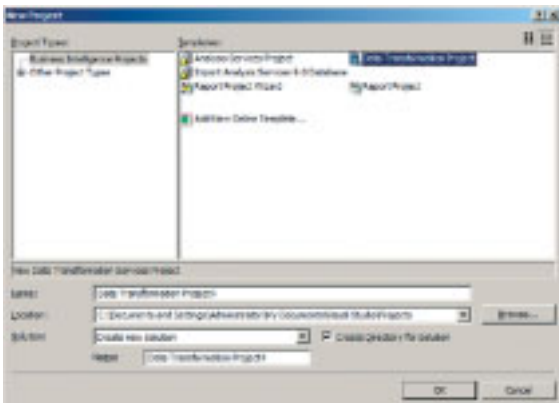


Figura 2

Por supuesto también se han incluido diversos asistentes que nos facilitan las labores, siguiendo la línea de la versión anterior, y mejorando las posibilidades que nos ofrecen.

Vamos a ver una primera imagen del nuevo entorno de desarrollo que sustituye al anterior diseñador de paquetes, el BIDS trabajando con un proyecto de transformación, -o, mejor dicho, ahora según el nom-

bre asignado en la última CTP, con un proyecto de integración-, y comentar algunas de sus novedades más destacables.

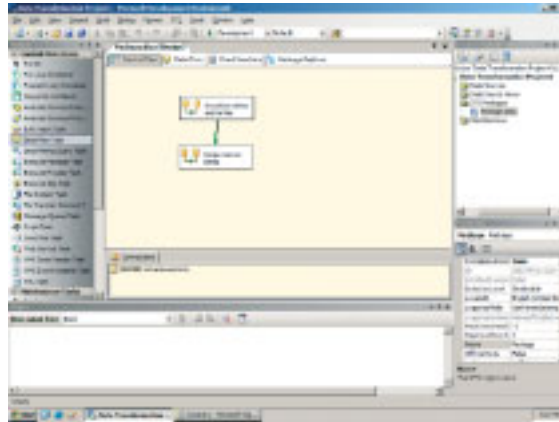


Figura 3

Como puede comprobar el lector, el nuevo entorno es muy diferente, no tiene nada que ver con el diseñador de paquetes, incluido en el Enterprise Manager, que nos ofrecía la versión anterior. Una de las primeras diferencias destacables es que ya no está orientado a diseñar un paquete, sino a soluciones y proyectos, y en éstos podremos incluir los paquetes que deseemos. Otra novedad destacable que nos encontramos es la separación entre el flujo de control y el flujo de datos.

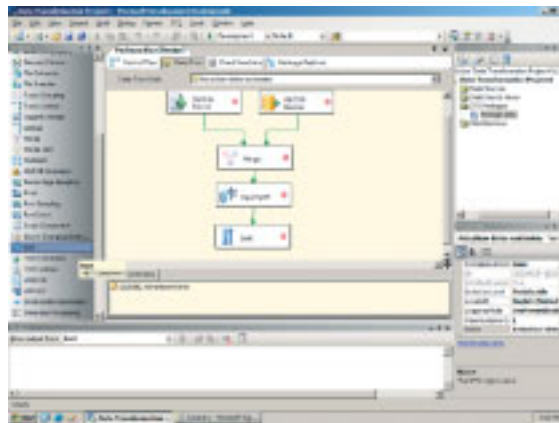


Figura 4

También puede apreciar que hay un montón de nuevos elementos disponibles en el *toolbox*, tanto para el control de flujo de ejecución, como para el flujo de datos, y que éstos son componentes .NET. Ahora muchas de las tareas que antes implicaban un desarrollo basado en ActiveX o en una gran cantidad de código en Transact-SQL se simplifican, reduciéndose a la inclusión del componente correspondiente para dicho proceso y a la configuración de sus propiedades.

Por último es conveniente indicar que hay una herramienta para la migración de nuestros paquetes desarrollados con SQL Server 2000 a la versión 2005, y que también dispone de la posibilidad de ejecutar paquetes de dicha versión.

## Analysis Services (SSAS)

Nos vamos a centrar a hora en los *Analysis Services* (SSAS), que nos proporcionan dos funcionalidades básicas: OLAP y Data Mining.

Antes de nada, es importante comentar que la herramienta de desarrollo también será el *Business Intelligence Development Studio* (BIDS), mucho más potente para el desarrollo que el antiguo *Analysis Manager*, y que incluye un potente depurador de scripts MDX. Allí podremos abrir un proyecto de Analysis Services.

Ya centrándonos en el modelado, vamos a comentar algunas de las novedades de la versión 2005:

Tenemos los *Data Source Views*, que proveen una capa de abstracción entre las bases de datos y los cubos, pudiendo contener tablas de diferentes Data Sources, además de permitirnos trabajar de forma desconectada.

En cuanto a la construcción de los cubos también tenemos notables mejoras, ya que tenemos un asistente llamado *Intellicube* que nos va a facilitar las tareas y nos va a ir guiando paso a paso y haciéndonos sugerencias.

El repositorio ha dejado de estar almacenado en una base de datos Access (aunque podía migrarse a una base de datos de SQL Server) y ahora estos metadatos se encuentran almacenados en XML. Otra característica destacable es que disponemos de herramientas para el despliegue de nuestros proyectos, cuando antes sólo teníamos mecanismos como el *backup/restore* de las bases de datos multidimensionales.

Otra novedad muy destacable en cuanto a la arquitectura es que ya no va a existir la diferencia actual en cuanto a fuentes de datos. Gracias al *Unified Dimensional Model* (UDM) que combina las mejores características de los modelos de datos relacional y multidimensional, ofreciendo grandes ventajas a las aplicaciones analíticas.

Como puede apreciar el lector, hay una gran cantidad de cambios, que harían que este artículo se extendiese mucho más. Pero hay que ceñirse a ciertas limitaciones, por lo que no voy a seguir describiendo otras muchas mejoras, que siguen siendo muy importantes y a su vez interesantes de conocer, por lo que dejaré esta tarea en manos del lector, y me voy a limitar simplemente a citarlas: *Proactive caching*, *Key*

*Performance Indicators (KPI) framework*, *translations*, *MDX scripts*, *XMLA (XML for Analysis)*, nuevos tipos de dimensiones, posibilidad de que un cubo contenga varias tablas de hechos, mejoras en la seguridad, en el lenguaje MDX, en rendimiento, en escalabilidad, y un largo etcétera...

Como muestra valga esta pantalla, donde el lector puede comprobar los grandes cambios que se han producido en el entorno de desarrollo, y quedan palpables algunas de las novedades citadas anteriormente.

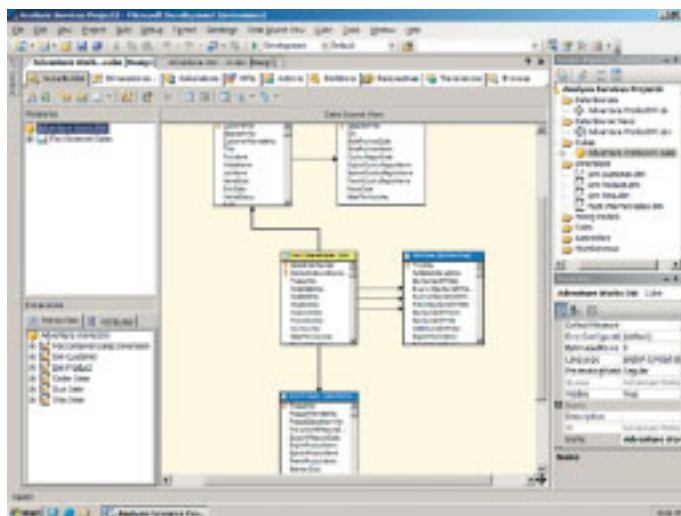


Figura 5

Con respecto a las novedades en *Data Mining*, voy a citar las dos más destacables:

Disponemos de nuevos algoritmos de minería de datos, pasando de dos que había en la versión 2000 a ocho en esta nueva versión.

Disponemos de un potente lenguaje DMX (no lo debe confundir el lector con el lenguaje MDX citado anteriormente), que nos va a facilitar las tareas típicas de creación de modelos de minería de datos, entrenamiento y predicciones sobre el modelo.

## Conclusión

Nos encontramos ante una nueva versión de SQL Server, donde se han producido una gran cantidad de cambios muy importantes, aún así, me atrevería a decir, que en lo que respecta a *Business Intelligence*, ha sido una de las partes que más ha evolucionado el producto, incluyendo unos servicios con una seguridad, disponibilidad, funcionalidad y escalabilidad muy superior, y unas nuevas herramientas para el desarrollo de proyectos mucho más potentes y productivas. ○

Todas las pantallas mostradas en este artículo pertenecen a la Beta 2 de SQL Server 2005. Entre los cambios en la última CTP quiero destacar que ya no aparece nada referente a Data Transformation Services, sino que aparece como Integration Services, que es el nuevo nombre que le han asignado a este servicio.



**¿ Piensas que estás seguro en la red?**

[www.hakin9.org/es](http://www.hakin9.org/es)



# hakin9

Hard Core IT Security Magazine

¿cómo defenderse?

También en nuestra tienda virtual:

[www.shop.software.com.pl/es](http://www.shop.software.com.pl/es)



Eladio Rincón

# Soporte XML en SQL Server 2005

SQL Server 2005 incorpora novedades en el soporte XML desde el lado del cliente (ADO.NET 2.0) y desde el lado del servidor (FOR XML, tipo de datos nativo, OPENXML). En este artículo veremos las posibilidades XML que tenemos en SQL Server 2000 y presentaremos las novedades en SQL Server 2005.

## » Soporte XML en SQL Server 2000

Antes de ver las novedades que aporta SQL Server 2005, debemos conocer cuales son las posibilidades en SQL Server 2000. SQL Server 2000 aporta funcionalidades XML divididas en dos grupos: posibilidad de representar datos relacionales en formato XML y la posibilidad de representar datos XML en formato relacional.

El soporte XML actual en SQL Server 2000 ofrece un “puente” entre el mundo relacional y el mundo XML; los datos se almacenan en formato relacional, y tenemos la posibilidad de representar/modificar en formato XML.

### Representación de datos relacionales en formato XML

Para representar datos relacionales en formato XML debemos añadir a la sentencia T-SQL la cláusula **FOR XML**. Ésta debe usarse con uno de los siguientes sufijos:

- **RAW**: Devuelve cada fila del resultado como elemento XML.
- **AUTO**: Devuelve cada tabla de la consulta como elementos independientes.
- **EXPLICIT**: Permite personalizar de manera explícita el formato del XML resultante.

No es el propósito de este artículo analizar cada uno de los métodos, pero debemos quedarnos con la idea de que dada una sentencia T-SQL que devuelve datos en formato tabular, la cláusula **FOR XML** convierte el resultado en formato XML; tradicionalmente se devuelve un XML a una aplicación cliente, y ésta, con hojas de estilo y/o validando el XML contra un esquema (XSD), pre-

senta los datos y/o realiza modificaciones en el XML usando XPath y/o API específicas para manipular XML.

Usando **FOR XML** es importante el orden de las columnas que aparecen en la **SELECT** y el orden de las columnas en la ordenación; como ejercicio podéis cambiar el orden de las columnas y veréis cuán diferente es el XML resultante.

---

SQL Server 2005 aporta el nuevo tipo de datos XML. Éste nuevo tipo de datos puede formar parte de columnas de una tabla, ser una variable, o incluso ser argumento de un procedimiento almacenado

---

### Representación de datos XML en formato relacional

El caso contrario es: dado un XML, convertirlo en formato relacional para poder “trabajarlo” con T-SQL. La aplicación cliente ha hecho ciertas modificaciones en el XML recibido (creado, por ejemplo, con **FOR XML**), y ahora hay que reflejar los cambios del XML en la base de datos. Para ello tenemos la función **OPENXML** que partiendo de una cadena de caracteres que representa un XML, lo convierte en formato tabular. Una vez convertido el XML a formato tabular, podemos operar con

---

Eladio Rincón

Es colaborador habitual de dotNetManía. Es MVP en SQL Server y uno de los fundadores del grupo de usuarios de SQL Server en España GUSE. Eladio es mentor asociado de Solid Quality Learning

el resultado utilizando tablas derivadas, *joins*, etc. Para usar la función `OPENXML` necesitaremos hacer uso del procedimiento almacenado de sistema `sp_xml_preparedocument` que hace uso del *parser* MSXML para representar en memoria una cadena. Una vez en memoria, la cadena es usada por `OPENXML`; después de usar `OPENXML` habrá que quitar de memoria la referencia al XML con `sp_xml_removedocument`.

## Mejoras de funcionalidades XML existentes en SQL Server 2000 en 2005

Las mejoras en `FOR XML` vienen enfocadas principalmente en facilitar la compleja tarea de “personalización” los XML generados con `FOR XML EXPLICIT`. Algunas de las mejoras son las siguientes:

- Posibilidad de definir el elemento raíz del documento XML.
- Crear consultas `FOR XML` como tablas derivadas... del orden `SELECT columna1, (SELECT ... FOR XML AUTO), columna2 ... FOR XML`, eliminando parte de la complejidad que suponía el uso de `FOR XML EXPLICIT`.
- Soporte para columnas nulas con la palabra clave `XSINIL`. Actualmente si un atributo es nulo, no se muestra el atributo.
- Convertir el resultado de `FOR XML` a tipo de datos XML nativo de manera que el resultado pueda ser modificado/formateado con consultas XQuery; para ello usaremos la opción `TYPE`. Más adelante veremos ejemplos de las funcionalidades que aporta XQuery.
- Modo `PATH`. Quizás la mejor novedad. Permite utilizar el formato X-Path para poder indicar la estructura del resultado. Mirad el ejemplo siguiente:

```
SELECT
'Articulo XML' AS 'NombreArticulo'
,'20050510' AS 'Fecha/Revision'
,CAST ('20050620' AS SMALLDATETIME) AS 'Fecha/Publicacion'
,'Eladio' AS 'Autor/@Nombre'
,'Rincon' AS 'Autor/@Apellido'
FOR XML PATH ('Articulo')
```

Resultado:

```
<Articulo>
<NombreArticulo>Articulo XML</NombreArticulo>
<Fecha>
<Revision>20050510</Revision>
<Publicacion>2005-06-20T00:00:00</Publicacion>
</Fecha>
<Autor Nombre="Eladio" Apellido="Rincon" />
</Articulo>
```

Fijaros cómo se definen en las columnas de la sentencia `SELECT` los elementos (o atributos) necesarios para definir la estructura del documento XML haciendo uso de extensiones X-Path.

## XML nativo en SQL Server 2005

SQL Server 2005 aporta el nuevo tipo de datos XML. Éste nuevo tipo de datos puede formar parte de columnas de una tabla, ser una variable, o incluso ser argumento de un procedimiento almacenado. Como primera apreciación, debemos quitarnos la idea de que por el hecho de que el XML se represente como una cadena de caracteres, internamente vaya ser guardado como una copia exacta del documento XML; SQL Server 2005 lo guardará internamente en formato binario (ver entrevista a César Galindo, del equipo de desarrollo de SQL Server 2005 en este mismo número), siendo su acceso más efectivo y teniendo la posibilidad de definir la estructura de los datos de manera más eficiente; como ejercicio proponemos jugar con los comandos `DBCC` para ver la estructura interna de una columna de tipo XML.

El tipo de datos XML puede tener como argumento la referencia a un esquema XML. Digamos que podemos obligar a que una columna (variable, argumento, ...), esté validada contra un esquema XML, o por el contrario definirlo “limpio” (sin validar contra ningún esquema XML). Aunque no sea validado contra un esquema, el dato deberá estar bien formado (*well formed*, <http://www.w3.org/TR/REC-xml/#sec-well-formed>). Como ejemplo, vamos a crear una tabla que contenga dos columnas XML una de ellas validada contra un esquema XML. Para ello, primero usaremos la sentencia `CREATE XML SCHEMA` que define el esquema de un documento XML que se ajusta al siguiente formato:

```
'<articulo>
<nombre>...</nombre>
<autor>...</autor>
</articulo>'
```

En la definición del esquema XML definimos las etiquetas XML necesarias en el documento, el tipo de datos soportado e incluso el número de ocurrencias que debe tener:

```
CREATE XML SCHEMA COLLECTION schema1
AS
N'<?xml version="1.0" ?>
<xsd:schema
targetNamespace="http://schemas.mi_base.com/schema1"
xmlns="http://schemas.mi_base.com/schema1"
elementFormDefault="qualified"
attributeFormDefault="unqualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
<xsd:element name="articulo">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="nombre" type="xsd:string"
minOccurs="1" maxOccurs="1"/>
<xsd:element name="autor" type="xsd:string"
minOccurs="1" maxOccurs="1"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>'
```



Y a continuación crearemos una tabla con los dos tipos de datos XML:

```
CREATE TABLE dbo.t1 (
    v1 xml (schema1)
    , v2 xml
)
```

Donde la columna `v1` será validada contra el esquema XML `schema1`, y la columna `v2` no será validada. A continuación insertaremos algunas filas:

```
INSERT INTO dbo.t1 ( v1, v2 ) select
'<?xml version="1.0" ?>
<articulo xmlns="http://schemas.mi_base.com/schema1">
<nombre>XML</nombre>
<autor>Eladio</autor>
</articulo>',
'<articulo nombre="XML">
<autor>Eladio</autor>
</articulo>'
GO
INSERT INTO dbo.t1 ( v1, v2 ) select
'<?xml version="1.0" ?>
<articulo xmlns="http://schemas.mi_base.com/schema1">
<nombre>SMO</nombre>
<autor>Marino</autor>
</articulo>',
'<articulo nombre="SMO">
<autor>Marino</autor>
</articulo>'
GO
```

Fijaros que para la columna `v1`, hemos definido en el dato XML el esquema contra el que será validado (`xmlns="http://schemas.mi_base.com/schema1"`).

En caso de que el dato XML no defina cuál es el esquema contra el que se valide, o no cumpla las restricciones del esquema, la operación generará una excepción y no se insertará la fila en la tabla. Al igual que las restricciones que se definen con las columnas de una tabla (*check*, *foreign key*, *primary key*, *unique* y *not null*) ayudan al optimizador de consultas a conocer la mejor manera de la distribución de los datos de una tabla, obligar a que una columna XML se ciña a un esquema XML, también ayuda al optimizador en el momento de generar planes de ejecución para consultas sobre el dato XML (consultas XQuery).

## Indexación de columnas XML

Al igual que las columnas “tradicionales” se pueden indexar, las columnas XML también se pueden indexar; las columnas XML pueden tener dos tipos de índices: *primario* y *secundario*. Por diseño, un índice secundario no puede existir si no hay un índice primario porque el método de indexación del índice secundario se basa en el índice primario; a su vez, una columna XML no se podrá indexar si la clave prima-

ria no es un índice agrupado (*clustered*). Para crear el índice XML añadiremos a la tabla creada una columna *integer* que será clave primaria, y después crearemos el índice XML:

```
ALTER TABLE dbo.t1
ADD id INT IDENTITY PRIMARY KEY
GO
CREATE PRIMARY XML INDEX
    p_xml_index ON dbo.t1 (v1)
```

Una vez creado el índice XML primario, podremos crear índices XML secundarios. Aquí tendremos tres tipos de índices XML secundarios:

- **PATH:** Beneficiará consultas XQuery que usan XPATH, por ejemplo, `/articulo[autor=" José Antonio"]`.
- **PROPERTY:** Cuando se accede a propiedades del primer elemento de un nodo XML, por ejemplo, `/articulo/@fechaPublicacion`.
- **VALUE:** Cuando la consulta XQuery hace referencia a cualquier elemento del nodo, por ejemplo, `//autor="José Antonio"`. Este caso devolvería todos los elementos cuyo autor sea `José Antonio`, independientemente de que sea hijo de `artículo` u otro elemento.

```
CREATE XML INDEX Path_p_xml_index ON dbo.t1(v1)
USING XML INDEX p_xml_index FOR PATH
```

Según comentarios de **Michael Rys** (*Program Manager* de XML en el equipo de SQL Server) en sus sesiones de Tech-Ed Orlando 2005, dependiendo de los tipos de consultas XQuery, las mejoras de rendimiento usando índices primarios XML pueden ser de hasta un 200%; además, usando un índice secundario XML la mejora sería de hasta un 50% más. Evidentemente, al igual que en los índices “normales”, deberemos tener en cuenta el coste que suponen los índices, y deberemos elegir los índices acorde al tipo de consultas XQuery que se vayan a realizar en nuestro sistema.

## XQuery

XQuery es el lenguaje utilizado para realizar consultas sobre instancias XML. De la misma manera que Transact-SQL se utiliza para realizar consultas sobre tablas de la base de datos, XQuery se utiliza para consultar sobre datos de instancias XML nativas; es un lenguaje estandarizado por los fabricantes y sus especificaciones se pueden consultar en W3C (<http://www.w3.org/XML/Query>). El tipo de datos nativo XML tiene los siguientes métodos:

- **query()** para realizar consultas sobre una instancia XML. El valor devuelto no se ajusta a

ningún esquema, pero sí está bien formado, por ejemplo:

```
select v1.query ( '
declare namespace my="http://schemas.mi_base.com/schema1";
/my:articulo/my:autor')
from dbo.t1
```

Devuelve como resultado:

```
<my:autor xmlns:my="http://schemas.mi_base.com/schema1">
  Eladio
</my:autor>
<my:autor xmlns:my="http://schemas.mi_base.com/schema1">
  Marino
</my:autor>
```

- **value()** para recuperar valores de la instancia XML.
- **exist()** para comprobar si una consulta devuelve un resultado no vacío. Por ejemplo:

```
select v2
from dbo.t1
where v2.exist ( '
/articulo[@nombre="SMO"]') = 1
```

Devuelve:

```
<articulo nombre="SMO">
  <autor>Marino</autor>
</articulo>
```

- **modify()** para hacer modificaciones. Podemos usar las funciones XQuery **insert**, **delete** y **replace value of**. Veamos un ejemplo: reemplazar el atributo XML por “Intro. XML” en la columna **v2** de la tabla **dbo.t1**.

```
update dbo.t1
set v2.modify ( '
  replace value of (/articulo/@nombre)[1]
  with "Intro. XML" ')
where v2.exist ( '
/articulo[@nombre="XML"]') = 1
cambia
```

- **nodes()** para devolver cada elemento XML como filas separadas de un elemento raíz. Por ejemplo, en:

```
DECLARE @x xml
SET @x='<doc>
  <fila>1</fila>
  <fila>2</fila>
  <fila>3</fila>
</doc>'
SELECT c.query('.') AS result
FROM @x.nodes('/doc/fila') as T(c)
```

Devolverá:

```
<fila>1</fila>
<fila>2</fila>
<fila>3</fila>
```

## Conclusiones

SQL Server 2005 acerca más si cabe XML a la base de datos. Sin embargo, deberemos ser lo más objetivos posible a la hora de decidir su utilización en los distintos escenarios que se nos planteen. El uso de XML en nuestros sistemas es una cuestión con la que afloran ciertos prejuicios antes siquiera de conocer el escenario: administradores de base de datos frente a desarrolladores expertos en XML, que al poder representar datos de una forma más integrada con la base de datos ven todo un abanico nuevo de posibilidades. Como referencia, tenéis una serie de artículos publicados en MSDN que nos ayudarán a identificar cada escenario y que podéis ver en la tabla adjunta. ○

### Artículos relacionados

XML Best Practices for Microsoft SQL Server 2005	<a href="http://msdn.microsoft.com/sql/2005/2005articles/default.aspx?pull=/library/en-us/dnsq90/html/sql25xmlbp.asp">http://msdn.microsoft.com/sql/2005/2005articles/default.aspx?pull=/library/en-us/dnsq90/html/sql25xmlbp.asp</a>
XML Support in Microsoft SQL Server 2005	<a href="http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq90/html/sql2k5xml.asp">http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq90/html/sql2k5xml.asp</a>
What's New in FOR XML in Microsoft SQL Server 2005	<a href="http://msdn.microsoft.com/sql/2005/2005articles/default.aspx?pull=/library/en-us/dnsq90/html/forxml2k5.asp">http://msdn.microsoft.com/sql/2005/2005articles/default.aspx?pull=/library/en-us/dnsq90/html/forxml2k5.asp</a>
XML Options in Microsoft SQL Server 2005	<a href="http://msdn.microsoft.com/sql/2005/2005articles/default.aspx?pull=/library/en-us/dnsq90/html/sql2k5xmloptions.asp">http://msdn.microsoft.com/sql/2005/2005articles/default.aspx?pull=/library/en-us/dnsq90/html/sql2k5xmloptions.asp</a>



José Miguel Torres

# SQL Server 2005 Mobile Edition

Microsoft SQL Server 2005 Mobile Edition es la evolución de Microsoft SQL Server CE Edition 2.0. Microsoft SQL Server 2005 Mobile Edition pretende ser una plataforma de almacenamiento de datos para soluciones móviles integradas bajo un marco de desarrollo rápido, flexible, fácil y escalable.

» **Las características** más destacadas de **SQL Server 2005 Mobile Edition (SQL Mobile en adelante)**, son en primer lugar la optimización del procesador de consultas y la utilización de transacciones. SQL Mobile es una base de datos relacional, con lo que la integridad de dominio, valor y referencial están presentes. SQL Mobile puede ser instalado independientemente de la plataforma de desarrollo utilizada, tanto para aplicaciones .NET Compact Framework como para aplicaciones eVC++ (nativas). Tanto para una como para la otra se integran herramientas de acceso a datos que permitirán la creación, modificación y selección de datos desde nuestras aplicaciones.

## La evolución de SQL Server CE 2.0

La característica más visible es la integración de SQL Mobile en lo que conocemos en Microsoft SQL Server 2005 como **SQL Server Management Studio**. Ésta posibilita la creación de una base de datos SQL Mobile bajo interfaz Windows sin necesidad de tener que acceder a aplicaciones de terceros. Asimismo, también está integrado en Visual Studio .NET 2005, con lo que las posibilidades son variadas. De esta manera, no sólo podremos crear y modificar las bases de datos desde las aplicaciones anteriormente mencionadas, sino que características como el Plan de ejecución gráfica, los asistentes para subscriptores de replicación y de configuración Web, e incluso del servicio de transformación de datos (DTS) (ahora *Integration Services*), el cual permite transformaciones tanto a/desde Microsoft Access como a/desde Oracle (entre otros), y una herramienta de implantación del fichero de la base de datos **.sdf**, estarán disponibles de manera sencilla.

SQL Mobile lleva el control de estadística para la optimización de consultas automáticamente, creando planes de consulta beneficiando de esta manera al motor de almacenamiento.

---

**El motor de almacenamiento, totalmente optimizado para dispositivos móviles, ha sido escrito de nuevo, posibilitando que cada una de las aplicaciones utilice un espacio de memoria compartida común**

---

Otra de las novedades destacadas que encontramos es la posibilidad de abrir la base de datos por dos usuarios, permitiendo el bloqueo a nivel de registro, a nivel de página y a nivel de aislamiento para mantener la integridad referencial en todo momento; posibilidad no operativa en versiones anteriores. Se mantienen la capacidad de replicación (como subscriptor) así como el acceso remoto a datos (RDA), para una total integración con su hermano mayor, SQL Server 2005.

El motor de almacenamiento, totalmente optimizado para dispositivos móviles, ha sido escrito de nuevo, posibilitando que cada una de las aplicaciones utilice un espacio de memoria compartida común (*pool*).

---

José Miguel Torres colabora habitualmente con dotNetManía. Es técnico superior en desarrollo de aplicaciones informáticas y trabaja como arquitecto de software en el departamento de tecnologías de la información de MRW



SQL Mobile reutiliza espacios vacíos en página permitiendo así el *Auto-Shrink* de la base de datos.

SQL Mobile, requiere el sistema operativo Windows CE 4.3 o superior y dispositivos PPC 2003. De este modo, también se ha capacitado a SQL Mobile para funcionar sobre Tablet PC, así como futuras versiones de Microsoft Windows Smartphone. De esta manera podrá soportar un mayor número de dispositivos independientemente de la plataforma operativa, es decir, nuestra base de datos SQL Mobile, funcionará tanto para Tablet PC, como para Pocket PC o Windows Smartphone, indistintamente.

Para los que tenemos numerosas bases de datos en SQL Server CE 1.0 y 2.0, es interesante saber que Microsoft adjunta una herramienta de actualización llamada **upgrade.exe** como se muestra:

```
upgrade.exe
/s "\dirOrigen\BDDorigen.sdf"
/sp "pwdOrigen"
/d "\dirDestino\BDDdestino.sdf"
/dp "pwdDestino" /e /q
```

En la tabla 1 podrán encontrar todas las posibilidades de **upgrade.exe**.

Parámetro	Descripción
/s	El path de origen de la base de datos.
/sp	La password de la base de datos de origen.
/d	El path de destino de la base de datos.
/dp	La password de la base de datos de destino.
/e	Si se especifica, la base de datos de destino será encriptada.
/q	Instalación silenciosa, sin ningún interfaz gráfico
/?	Muestra los posibles parámetros

Tabla 1. Parámetro de **upgrade.exe**

## Manos a la obra

Cuando creamos la base de datos y a diferencia de SQL Server, ésta se almacena en un único archivo con extensión **.sdf**. La capacidad máxima está limitada a 4 Gb, algo trivial teniendo en cuenta las capacidades de almacenamiento de los dispositivos móviles actuales. La base de datos, como dije antes, puede ser accedida

desde cualquier tipo de dispositivo móvil así como con el SQL Server Management Studio desde nuestro PC de desarrollo.

Antes de empezar a crear la primera base de datos en SQL Mobile tengamos en cuenta ciertas limitaciones además de la del tamaño de archivo. En los libros por pantalla de SQL Server 2005 pueden encontrar las limitaciones de todos los objetos de SQL Mobile, desde el tamaño de página y su incremento en Kb, pasando por número de sesiones máximas, hasta índices por páginas. Asimismo sería interesante conocer los diferentes tipos de datos que utiliza SQL Mobile, sobretodo en contextos de utilización de replicación por SQL Server, para evitar errores de conversión.

SQL Mobile utiliza bases de datos temporales siempre y cuando se especifique de manera explícita. El tamaño de ésta dependerá del tipo de operación que ejecutemos. Aunque en un marco de desarrollo, digamos, convencional como puede ser la de creación de una aplicación escritorio, las bases de datos temporales son “transparentes” en la mayoría de casos, en un marco de desarrollo móvil el espacio de almacenamiento sea ROM o RAM

es muy limitado, así que cuando decidamos utilizar la base de datos temporal, pensemos que en operaciones de **UPDATE** o **DELETE** de muchas filas o en operaciones de **SORT** sobre columnas no indexadas, puede crear una base de datos temporal grande ya que sólo existe una sola base de datos temporal para todas las base de datos. La base de datos temporal también almacenará información en el caso de que una de las bases de datos se acerque, según el crecimiento de la misma, al límite anteriormente especificado.

Ahora sí, veamos un ejemplo de creación de una base de datos. Para ello utilizaremos el SQL Server Management Studio. Una vez nos conectemos a SQL Mobile, deberemos indicarle que queremos la creación de una nueva base de datos (figura 1).

Como información básica indicaremos la localización del fichero de la

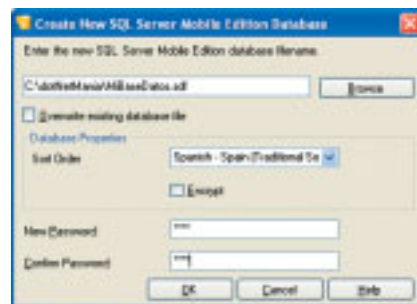
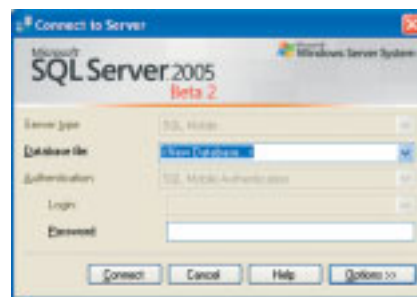


Figura 1. Conexión a SQL Mobile y creación de una nueva base de datos.

base de datos en nuestro PC de desarrollo, así como el nombre de la misma. Seleccionaremos en las propiedades de la base de datos la información sobre la cultura, la utilización o no de que la base de datos esté encriptada, así como su contraseña (figura 1).

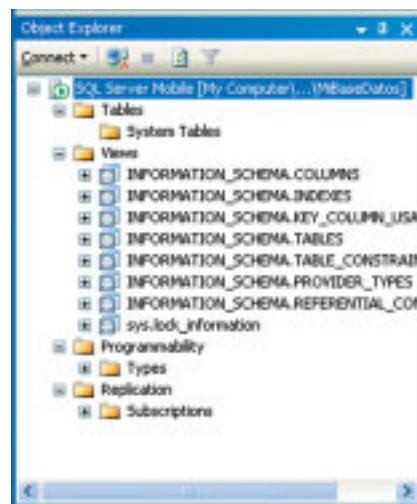


Figura 2. Object Explorer de SQL Server Management Studio.

Para la mayoría que hemos utilizado alguna interfaz de administración para nuestro gestor de base de datos, no nos supondrá ningún problema la creación de tablas, vistas, y sus respectivos atributos. Como pue-

den observar en la figura 3, la creación de una tabla es sencillamente rápida y flexible.

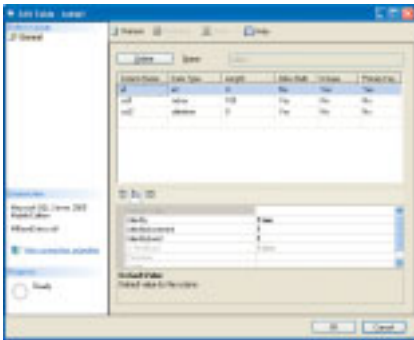


Figura 3. Edición de una tabla.

## Consultas

Con SQL Mobile, podemos realizar consultas desde el SQL Server Management Studio utilizando la mayoría de operaciones que se pueden utilizar en SQL Server como el plan de ejecución estimado (figura 4).

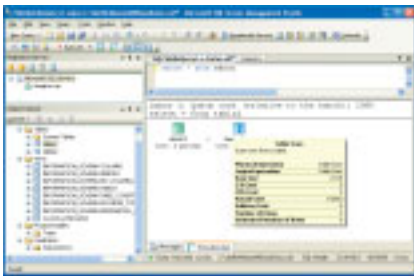


Figura 4. Plan de ejecución estimado desde SQL Server Management Studio.

Aunque la sintaxis de consulta y modificación basada en el lenguaje SQL es extensa, lamentablemente no podemos utilizar Transact-SQL en SQL Mobile, permitiendo sólo una consulta simple por comando, eso sí, SQL Mobile soporta la ejecución parametrizada de consultas. De la misma manera, tampoco podemos ejecutar scripts, ya que SQL Mobile no posee lo que en SQL Server es conocido como SQL Manager Objects (SMO), que no es más que la evolución de SQL DMO en las versiones anteriores (vea el artículo dedicado a SMO en este mismo número).

De todas formas, desde el SQL Server Management Studio o desde el

SQL Query Analyzer del SQL Mobile podemos crear/modificar o consultar tablas, modificar o crear índices, compactar o reparar la base de datos, verificar accesos remotos RDA y aplicaciones replicadas... En definitiva, todo lo que SQL Mobile permite.

## Multiusuario

Una de las nuevas características de SQL Mobile es el acceso multiusuario. Para los que hemos desarrollado con SQL Server CE 1.0 o 2.0, las propiedades de concurrencia y bloqueo son nuevas en esta versión y como tal debemos tenerlas presentes. Con SQL Mobile, podemos tener más de una aplicación accediendo a la base de datos realizando varias operaciones sobre la base de datos e incluso con la misma tabla, o tener varias conexiones desde una sola aplicación. Es aquí donde el bloqueo y la consideración de concurrencia entran en escena. La cantidad de conexiones abiertas está restringida a 256, siempre y cuando los recursos del sistema lo permitan.

La posibilidad de abrir una base de datos de forma exclusiva está permitida. En este caso el acceso multiusuario queda inhabilitado permitiendo sólo la conexión que de forma explícita requirió la exclusividad.

Aunque la capacidad multiusuario está implantada, no existe ningún tipo de implantación de rol a nivel de seguridad y acceso. Todas las conexiones utilizan la misma autenticación y autorización, que de forma predeterminada, ofrece SQL Mobile.

A nivel de concurrencia, existen dos tipos de bloqueo, el optimista y el pesimista. En una transacción dentro de SQL Mobile y con un nivel de concurrencia pesimista, SQL Mobile bloquea los recursos de la base de datos hasta que la transacción haya finalizado. En un control optimista, intenta solucionar los conflictos de acceso a recursos, no dejando inoperativa la base de datos. El bloqueo pesimista viene implementado en SQL Mobile, sin embargo el bloqueo optimista no. Esto significa que aunque el bloqueo

optimista no este implementado (pero si soportado), a nivel programático debemos realizar las operaciones que permitan la integridad y disponibilidad de los datos. Como resultado de la utilización de un bloqueo optimista nos podemos encontrar con lecturas fantasmas de registros, pérdidas de datos actualizados, lecturas "sucias" de registros (*Dirty Read*). Aunque no explicaremos, por su extensión, todos los casos, queda evidente que la implantación de acceso multiusuario permite una flexibilidad que en versiones anteriores nos limitaba, pero el precio de su utilización es el exhaustivo control por parte de las aplicaciones del acceso a la base de datos.

Podemos utilizar transacciones desde SQL, así como desde código administrado o código nativo. Las transacciones son atómicas, durables, aislables y consistentes (ACID). Sin embargo, las transacciones no son distribuidas, y condicionan la utilización de cursores dentro su propio contexto en sentencias SQL.

El manejo de transacciones en SQL Mobile, por defecto, es de forma automática, aunque podemos hacerlo de forma explícita. Cuando SQL Mobile lo hace automáticamente, realiza el autocommit él mismo.

## Conclusión

Hemos introducido de una manera rápida las principales características de la nueva versión de SQL Mobile. Como tal, ofrece una serie de nuevas funcionalidades que deberemos tener en cuenta para el desarrollo de nuestras aplicaciones; de las cuales destaco la integración tanto con Visual Studio .NET como con SQL Server Management Studio además de las posibilidades multiusuario que, aunque ofrecen una mayor flexibilidad, deberemos entender y replantearnos el escenario de implementación. ☺



### Más enlaces

<http://www.microsoft.com/sql/ce/mobility/beta1.asp>

<http://www.microsoft.com/sql/ce/default.asp>

<http://www.desarrollomobile.net/dotNetMania/SQLMobile2005>



# **DOTNETSOLIDARIO**

**WWW.DOTNETSOLIDARIO.COM**



**INTERNET SE ESTA CONVIRTIENDO EN EL MEJOR MEDIO PARA DAR A  
CONOCER IDEAS, PROYECTOS, INICIATIVAS.**

**DOTNETSOLIDARIO ES UN PORTAL SIN ANIMO DE LUCRO QUE TRATA  
DE PONER A DISPOSICION DE LOS USUARIOS DE INTERNET LOS PROYECTOS  
SOLIDARIOS QUE LLEVAN A CABO LAS ONGS.**

**PARA ELLO OFRECE GRATUITAMENTE SOLO CON ACREDITAR SER ONG:**

**🕒 HOSTING ASP .NET**

**🕒 SOFTWARE MICROSOFT .NET**

**🕒 FORMACION**





Pepe Hevia

# Transact-SQL 2005

## Más productivo que nunca

Este es un artículo complicado de escribir. No por los contenidos, sino por el cómo abordar algo tan amplio como Transact-SQL. Este componente de SQL Server es el pilar fundamental en el que se basan el resto de los componentes, pues es el lenguaje de programación con el que se le dice al core de SQL Server lo que debe de hacer.

» **Y el problema es** que T-SQL es el conjunto de instrucciones que todos los subsistemas de SQL emplean para orquestar sus acciones. Esto quiere decir que cualquier novedad en cualquier subsistema, por descontado, genera nuevas instrucciones SQL y mejora las antiguas.

Así que, bueno, antes de empezar a contar nada, voy a intentar delimitar lo que vamos a tratar en este artículo, ya que T-SQL abarca muchos subsistemas, y no hay espacio para todos... Haremos lo siguiente: introduciré las novedades más relevantes de T-SQL desde la perspectiva del desarrollador; el soporte integrado y nativo de XML; el nuevo motor de ejecución integrado de .NET y las nuevas herramientas y opciones interesantes relacionadas con el desarrollo sobre T-SQL. Con esto imagino que conseguiré daros una idea (¡y desbordaros!) con las novedades que hay en la mayor revisión de SQL Server de todos los tiempos. Y el resto de T-SQL, seguro que lo averiguaréis usando las herramientas de administración del nuevo SQL Server Management Studio. Valor... ¡y al toro!

### Las novedades del T-SQL

Son muchas. Y repartidas a muchos niveles. Y van desde nuevas instrucciones del lenguaje SQL a la integración nativa con estándares abiertos como HTTP, SOAP, XML...etc, pasando por la integración con el motor de ejecución y los tipos del CLR de .NET. Como ya vino siendo habitual desde las primeras versiones de SQL Server; las primeras novedades siempre vienen en forma de nuevas instrucciones del lenguaje para abordar los grandes grupos de sentencias SQL: bloques DDL

(*lenguaje de definición de datos*), DCL (*lenguaje de control de datos*) y DML (*lenguaje de manipulación de datos*). Que se adecuan a las novedades de ejecución del motor de la base de datos. Y aunque la mayoría están basadas en los estándares ANSI, es con T-SQL con lo que saca un mayor provecho a estas sentencias, como es lógico.

---

Las novedades de TSQL son muchas y repartidas en muchos niveles. Van desde nuevas instrucciones del lenguaje hasta integración nativa con estándares abiertos, pasando por la integración con .NET Framework

---

Obviamente, cualquier cambio en un tipo de dato hace que todas las sentencias se vean afectadas. Y eso es lo que ha ocurrido en SQL Server 2005. Aparecen nuevos tipos de datos...

### Las novedades en DDL

Todas las novedades del lenguaje DDL se recogen agrupando todas las sentencias en varios grupos, que

---

#### Pepe Hevia

colabora habitualmente con dotNetManía. Es arquitecto de software senior en Alhambra-Eidos. Tiene las certificaciones de MCAD, MCS.D.NET y MCT en .NET Framework. Ha escrito varios libros sobre la plataforma .NET ([alibriedigital.com](http://alibriedigital.com))

corresponden a los diferentes subsistemas de SQL Server. Entre esos grupos, las novedades vienen de la mano de la integración de .NET Framework en la plataforma, el nuevo Service Broker, el nuevo sistema de indexación y su capacidad de particionamiento, los sinónimos, las gestión de eventos de los Triggers y, como no, de la seguridad. El resto de las novedades, son, en realidad, mejoras a lo que ya conocemos: ampliaciones de las instrucciones de gestión de bases de datos, de la creación de tablas e índices... etc.

Pero vamos por partes, como las integrales. En DDL, y la creación de nuevas bases de datos, las novedades comienzan con los tipos de datos. En las versiones anteriores de SQL Server, existía una amplia variedad de tipos, lo que proporciona ventajas e inconvenientes. El inconveniente venía de la mano de los tipos `varchar`, `image`, `text` y `binary`. `Varchar` almacena cadenas de caracteres de un máximo de casi 8Kb; `text` era utilizado para subsanar la limitación de los `varchar` de almacenar caracteres, llegando hasta la friolera de 2GB; `image` proporciona un campo `BLOB` para almacenar imágenes fundamentalmente; `binary` por la contra, es realmente el campo base de los dos anteriores. Luego, entonces, por qué distinguirlos. Qué ha cambiado... que los nuevos tipos de datos resultado de optimizar lo ya existente han sido los nuevos `varchar(max)`, `nvarchar(max)` y `varbinary(max)`. Bajo esta denominación tengo el orgullo de presentaros los tipos que reemplazan a los anteriores: `Varchar` y `NVarchar` proporcionan un nuevo sistema de almacenamiento capaz de almacenar hasta 231 bytes, con la diferencia de ser uno ASCII estándar y el otro un sistema de almacenamiento de caracteres UNICODE; `varbinary` ahora queda con el rol de almacenar cualquier tipo de información `BLOB` (imágenes, sonidos, objetos...etc). Con la ventaja añadida de que podemos limitar el tamaño máximo de la información que necesitamos guardar. Ahora, un detalle... Si especificamos un valor `n`, éste estará limitado a un máximo de 8Kb. Si especificamos el valor `max`, indicaremos al sistema el tamaño máximo de almacenamiento (de nuevo, 231 bytes). En todos los casos, por supuesto, variable. Ni que decir tiene que en la sentencia `Update` existen nuevas extensiones que permitirán al desarrollador poder trabajar con estos nuevos tipos de datos, e incluso optimizar su uso con la nueva cláusula `WRITE`. Aunque luego lo comentaremos con más detalle, no puedo evitar hacer mención a ello ahora. Dentro de los nuevos tipos de datos aparecen dos nuevas estrellas: XML y .NET. Podremos definir columnas que de por sí son documentos XML instanciables y analizables y tipos de datos puros de .NET Framework... es decir, objetos. Pero lo dejamos por ahora...

Otra novedad más interesante: *los sinónimos*. Este concepto que ya existía en otras bases de datos, permite al desarrollador el poder definir *alias* que apuntan a otros objetos del sistema. La idea, encapsular nombres de objetos o tipos avanzados de datos, que pueden estar en constante cambio. Por ejemplo,

podremos integrar un ensamblado dentro de la lista de procedimientos almacenados de SQL. En lugar de acceder a su nombre empleando el espacio de nombres, podemos crear un sinónimo que nos simplifique la tarea. Otra estimada utilidad de este concepto es el de poder apuntar a objetos de diferentes esquemas. Por ejemplo, si trabajamos desde la base de datos **A** y queremos acceder a una tabla de la base de datos **B**, necesitamos indicar toda la ruta hasta dicho objeto si queremos realizar trabajos distribuidos (*servidor.basedatos.esquema.objeto*). Con los sinónimos podríamos crear un “objeto virtual” en la base de datos **A** de forma que –al igual que las vistas– apuntara a un objeto de la base de datos **B**, sin tener que preocuparnos a partir de este momento de cómo acceder a la base de datos **B**. Miremos como queda el ejemplo del código fuente 1:

```
--Crear el sinónimo
USE BaseDatosA
GO
CREATE SYNONIM ProductosB
FOR BaseDatosB.dbo.Productos

--Usar el sinónimo...
USE BaseDatosA
GO
SELECT IdProducto,Nombre,Precio
FROM ProductosB
```

Fuente 1

Para acabar con este punto acerca de las novedades de DML –y aunque hay muchas más en forma de revisiones– no puedo dejar de comentar que aparece algo que hacía falta: *los triggers DDL*. Esta potente utilidad nos permitirá capturar con “eventos” de base de datos hechos tales como la modificación del esquema de la base de datos, así como acciones personalizadas. Esto es, que podremos actuar cuando alguien intente crear una nueva tabla en la base de datos, y podremos aprobar dicha creación o no en base a lógica de negocio. Lo dicho, una auténtica utilidad.

```
CREATE TRIGGER controlEsquema
ON MIBDD FOR CREATE_TABLE
AS
INSERT INTO RegistroOperaciones VALUES(EVENTDATA())
GO

CREATE TABLE TmpCalculos ( /* campos de la tabla */ )
```

Fuente 2

## Las novedades en DML

Dentro de este apartado, simplemente comentar las nuevas cláusulas que nos pueden ser útiles a la hora de

recabar información o bien actualizarla. Una de las novedades es la posibilidad de crear CTE o *Common Table Expressions* (*Expresiones Comunes de Tablas*). Esto que suena tan *fashion* es algo entre las funciones de SQL Server y las cláusulas **WITH** conocidas de VB. Nos permiten poder crear expresiones calculadas sobre tablas que podemos reutilizar en consultas de más alto nivel. Una especie de vistas temporales. Pero no sólo son un filtro para encapsular vistas sencillas, sino que permiten en toda su potencia la posibilidad de crear consultas recursivas, maestro-detalle e incluso consultas jerárquicas... ¡impresionante! Algo sencillo quedaría de la forma como se ve en el código fuente 3.

```
USE AdventureWorks;
GO
WITH DirReps(ManagerID, DirectReports) AS
(
    SELECT ManagerID, COUNT(*)
    FROM HumanResources.Employee AS e
    WHERE ManagerID IS NOT NULL
    GROUP BY ManagerID
)
SELECT ManagerID, DirectReports
FROM DirReps
ORDER BY ManagerID;
GO
```

Fuente 3

Y si os fijáis en la figura 1, una consulta CTE jerárquica que muestra una relación jerárquica entre los empleados de la base de datos *AdventureWorks* (que reemplaza a la ya más que veterana y omnipresente *Northwind*).

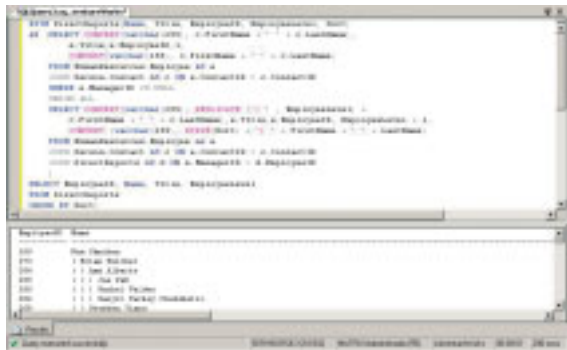


Figura 1. Captura de pantalla de una consulta CTE jerárquica

La cosa queda en que podemos realizar subqueries dentro de la cláusula **WITH** que actúan de detalle, maestro y la query principal, de orquestadora (figura 2).

Otras dos novedades muy interesantes, pero un poco complejas, son las nuevas cláusulas-operadores relacionales **APPLY** y **PIVOT/UNPIVOT**. Ambas pertenecen al grupo de optimizaciones realizadas a nivel de consultas cruzadas entre tablas, conocidas más comúnmente como

```
WITH descendientes(padre, id, cantidad) AS
(SELECT padre, id, cantidad
 FROM maestro WHERE id = @start
 UNION ALL
 SELECT E.padre, E.id, E.cantidad
 FROM maestro AS P INNER JOIN
 descendientes AS D ON D.id = E.padre
 )
SELECT id FROM descendientes
```

Figura 2. Esquema de las consultas recursivas

**JOINS**. **PIVOT** nos permitirá convertir filas en columnas (extremadamente útiles para informes de sumario) en lo que **UNPIVOT** hace el trabajo contrario, es decir, convertir columnas en filas. La utilidad de usar **PIVOT** estriba en la posibilidad que nos brinda de crear esquemas abiertos, esto es, que en lugar de definir tablas con numerosas columnas de propiedades –todas por defecto a **NULL**, porque puede que no tengamos todos los datos–, se dividen las propiedades/atributos en múltiples filas, y sólo aquellos de los cuales tenemos datos consistentes (ver figura 3, queda muy claro).

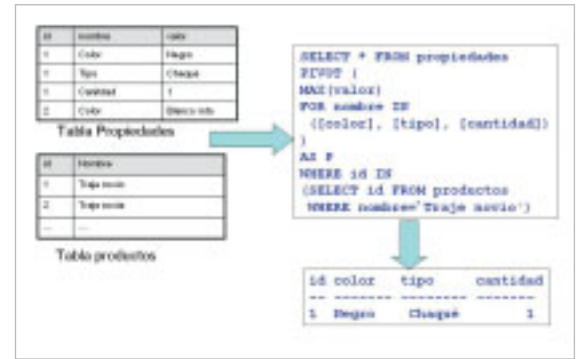


Figura 3. Utilización de PIVOT

**APPLY** es el encargado de permitir el **JOIN** de tablas con resultados obtenidos de funciones tabulares. Para poder delimitar quién de ellos tiene la potestad de controlar los **NULL** en el caso de no encontrarse equivalencias, es por lo que podremos usar diversos modificadores (**CROSS APPLY** o **OUTER APPLY**).

### Nuevo sistema de control de errores: TRY-CATCH

¡Impresionante! No sólo hemos visto la evolución de los lenguajes .NET para adquirir ese maravilloso sistema de control de excepciones que es el bloque **try-catch**, sino que además se nos regala esa estructura en T-SQL. Mucho mejor que estar continuamente jugando con la variable **@@ERROR** y la función **RAISERROR** tras evaluar el valor de error con un sencillo **IF**, podemos jugar ya con un bloque de código del estilo a lo que podemos ver en el código fuente 4.

Y con el conjunto de funciones que nos permitirán conocer en todo momento la información de la excepción producida: **ERROR\_NUMBER()**, **ERROR\_SEVERITY()**, **ERROR\_MESSAGE()**. Lo dicho, que ya podemos



```
BEGIN TRY
    -- lista de instrucciones que queremos controlar
END TRY
BEGIN CATCH
    -- lista de instrucciones del manejador de posibles
    excepciones
END CATCH
```

Fuente 4

escribir un código más estructurado que nunca, ya que además estos bloques pueden estar anidados. Nos quitamos de encima así, las eternas estructuras IF anidadas de control de errores -que tantos errores nos han producido... seguro-.

---

El lenguaje T-SQL tiene nuevas instrucciones, optimizaciones que se demandaban desde las versiones anteriores y nuevos manipuladores de excepciones, que lo hacen más robusto y productivo que nunca

---

### MARS. Una optimización necesaria

Antes que pensar en implantaciones extraterrestres, MARS representa las siglas de *Multiple Active Result Sets*. Esto quiere decir que se pueden mantener activas varias instrucciones/BATCH en el la misma conexión, sin tener la necesidad de cancelar un BATCH en ejecución antes de poder procesar el siguiente. Algo parecido a algo muy típico como cuando alguien, por ejemplo, está leyendo datos de una tabla y emplea otro comando para actualizar datos de otras tablas en base a los valores de la primera. Hasta que el SQL no libera el cursor de la primera consulta, el planificador no le permite continuar con la siguiente. Esta nueva tecnología permitirá al desarrollador ejecutar múltiples bloques T-SQL en la misma conexión, sin tener que preocuparse de si ya existen o no transacciones pendientes de finalización, que le forzaban a instanciar nuevos objetos `Connection`. De un modo inteligente SQL irá planificando las peticiones MARS y devolviendo múltiples resultados si es menester de los procesos establecidos. Lógicamente, sólo las últimas versiones de los clientes SQL (ODBC, OLEDB o los proveedores gestionados `SqlClient`) podrán gozar de estas características.

## XML. Un nuevo tipo de datos

De XML creo que hay que hablar poco en estos días. Ya sabemos todas las ventajas que nos ha aportado XML a nivel de “pegamento” entre capas en arquitecturas de software (flexibilidad, adaptación, reutilización, transparencia, y un largo etcétera). Ventajas incluso a nivel de estándares como el de los servicios Web y su SOAP, WDSL, etc. Y sobre todo, su ventaja a la hora de aportar una estructura autodefinida en los modelos de datos. La sencillez que proporciona a la hora de estructurar de forma tabular, estructuras de información. Y, como no, resultados de bases de datos.

XML no es una novedad en SQL Server. Por lo menos no, en cuanto a su integración con muchas de las funciones y sentencias de T-SQL. Recordemos que la instrucción `SELECT` de la versión 2000 ya era capaz de devolver *resultsets* en perfecto XML, con la directiva `FOR XML AUTO`. Pero esa integración era casi más una capa de servicio de datos, que una característica más de SQL Server. Es por eso que en SQL Server 2005, se ha subido al siguiente escalón: integrar XML en la arquitectura del propio sistema de tipos de SQL Server. De tal forma que podamos contar con columnas que ya almacenan de forma nativa XML, sin necesidad de lógica adicional. Vea el artículo de Eladio Rincón en este mismo monográfico dedicado específicamente a XML.

Qué nos ofrece este nuevo nivel de integración: Tipo de dato XML nativo en las tablas, integración de esquemas en el modelo de objetos de la base de datos, el potente lenguaje XQuery de consultas XSQL y la integración del motor XPath para manejo avanzado de consultas XML, la potencia de las consultas XMLSQL y sus resultados en XML y las mejoras de las sentencias que ya estaban en el ajo en la versión anterior: `FOR XML` y `OPENXML`. Rápidamente...

### XML: Un nuevo tipo de dato y la integración de los Esquemas como Objetos en la base de datos.

Es una pasada. XML integrado en la definición de nuevas tablas como un tipo de datos más, a la altura del `INTEGER` o el `VARCHAR`. Pero con muchas ventajas asociadas. No es simplemente un campo de tipo `VARCHAR` renombrado. Los tipos de datos XML tienen como responsabilidad el garantizar que los documentos XML están bien formados –o sintácticamente correctos–... pero más allá, validarlos –es decir, comprobar que la semántica del documento XML coincide con la establecida en un documento de esquema o XSD–. Ahora, hay que aclarar que el esquema XSD no es necesario, dándonos la posibilidad y la potencia de crear columnas XML que disponen de esquemas dinámicos, flexibles y adaptables a los cambios. Y que además, podemos guardar documentos completos, o sólo fragmentos de los mismos. A aquellos XML con XSD les llamaremos *Typed XML* y a los que no dependen de un esquema, *Untyped XML*.

Cuando hablo de un nuevo tipo de datos, también me refiero a la inherente capacidad que tienen la mayoría de los tipos de datos de SQL: la capacidad de ser indexado... ¡SÍ!, acelerando en mucho los procesos sobre dichos tipos de datos. Y además, se proporcionan todas las herramientas para poder acceder a sus valores. En este caso, podemos contar con las expresiones XQuery embebidas en el propio lenguaje T-SQL. Una maravilla. Un ejemplo sería algo del estilo a lo que podemos ver en el código fuente 5.

### El acceso a los datos XML

Es aquí donde nos metemos de lleno ya en el proceso de los documentos XML y donde T-SQL echa ya el resto. XQuery embebido en las propias sentencias T-SQL. Esto dota al lenguaje de una facilidad de proceso de los documentos XML sin precedencia (sólo equiparable a la potencia que nos proporcionaría el propio API XML de .NET, mediante el uso de lenguajes de programación). Este lenguaje XQUERY hay que decir que es un estándar *de facto*, ya que está basado en el DRAFT 4 del W3C y tiene su base en XPATH. La mayor diferencia entre ambos es que

XQUERY tiene extensiones que permiten la manipulación de los datos, en lo que XPATH es de consulta. Mucha de la funcionalidad viene de las extensiones del tipo de dato XML: **query**, **exists**, **value** y muchos otros. Donde **query** es la que nos permite poder ejecutar sentencias XQUERY.

Habría mucho que contar acerca de estos dos monstruos, pero dejaremos esto para futuros artículos. Por ahora, abriremos boca con un ejemplo como el de los fuentes 6 (y su pantalla en la figura 4) y 7.

```
CREATE XML SCHEMA COLLECTION FacturaSchemaCol
AS
N' <?xml version="1.0" encoding="UTF-16"?>
<xsd:schema
  targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManuInstructions"
  xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManuInstructions"
  ...
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:complexType name="FacturaCT" mixed="true" >
    <!--definición de las reglas del esquema de la factura -->
  </xsd:complexType>
</xsd:schema>
GO

CREATE TABLE Pedidos(
  id INT PRIMARY KEY,
  factura XML(FacturaSchemaCol)
  ...
)
```

Fuente 5

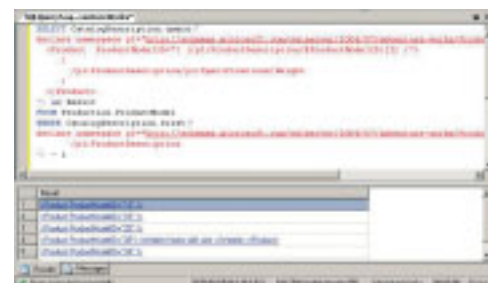


Figura 4. La ejecución del código fuente 6. XQuery en acción

```
select Factura.query(
'for $elem in /Factura/DatosPersonales
return
  <Nombre>
  { sql:column("Nombre") }
  </Nombre> '
) from Pedidos
```

Fuente 7

```
SELECT CatalogDescription.query('
  declare namespace p1="(...) /adventure-works/ProductModelDescription";
  <Product ProductModelID="{ (/p1:ProductDescription/@ProductModelID)[1] }">
  {
    /p1:ProductDescription/p1:Specifications/Weight
  }
</Product>
') as Result
FROM Production.ProductModel
WHERE CatalogDescription.exist('
  declare namespace
    p1="(...) /adventure-works/ProductModelDescription";
    /p1:ProductDescription
  ') = 1

--nota: los (...) indican que se ha obviado la ruta al esquema. Ejemplo extraído de los libros en línea de SQL Server 2005
```

Fuente 6

Ni qué decir tiene que parte de las optimizaciones que proporciona SQL Server 2005 para el proceso de este tipo de información, se apoyan en los índices. Podremos crear índices en columnas XML y los índices podrán ser de varios tipos: basados en atributos, valores y en estructuras XPath.

Otra de las grandes utilidades del nuevo soporte de XML en T-SQL es la utilización de las consultas SQL basadas en la gramática SQLXML, por la cual podremos embeber SQL dentro de documentos XML. Esto proporcionará un nivel más abstracto de interacción con el SQL Server 2005 a la par que proporciona una herramienta muy valiosa a todos aquellos arquitectos que modelan sus capas de servicios con servicios Web o con otras plataformas no

## XML está integrado en todos los recovecos del sistema. Podremos crear tipos de datos, hacer uso de funciones y procedimientos del sistema, e incluso crear índices sobre esos tipos de datos... Tan sencillo como manejar un VARCHAR

Microsoft. Sin controladores complicados, simplemente envíame el documento XML con lo que tengo que hacer, y yo te diré "sí, cariño". Un ejemplo de este concepto, en el fuente 8.

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-sql">
  <sql:query client-side-xml="0">
    SELECT *
    FROM VistaFacturacion
    FOR XML AUTO
  </sql:query>
</ROOT>
```

Fuente 8

Todo esto, unido a la integración con los estándares HTTP, SOAP y WSDL convierten SQL en un auténtico .NET Server. Y sus funciones perfectamente encapsuladas en servicios Web, lo dotan de la potencia necesaria para formar una pieza robusta más dentro de las arquitecturas empresariales.

### El CLR y .NET Framework. Una alternativa, no el reemplazo.

En el artículo de **Luis Miguel Blanco** de este monográfico de SQL Server 2005 ya habréis leído acerca de la integración de éste y el .NET Framework. Obviamente, en este punto tengo que comentarlo por su implicación en las mejoras que esto proporciona al propio lenguaje T-SQL. Ni qué decir tiene que las posibilidades que nos brinda una tecnología tan amplia como .NET a nivel de servidor son ilimitadas, pues supera barreras que el propio T-SQL tenía a nivel de lógica de negocio. Un lenguaje diseñado para trabajar de un modo sencillo con complejos conceptos relacionales, no está orientado a soluciones que requieran de cierta lógica más estructurada o integrada (como la OOP o el acceso a servicios externos al propio concepto de base de datos).

En el nuevo T-SQL, .NET Framework se convierte en un tipo de objeto más de la base de datos, de tal forma que nos permite poder crear funciones de usuario, procedimientos almacenados... incluso ¡tipos de datos! Y esto dota al SQL de un nuevo nivel de proceso y representación de información. Que tendrá claramente ventajas en conceptos como servicios de capa de datos, integración y lógica de negocio. Pero, la cuestión es ¿.NET es un reemplazo de T-SQL? La respuesta es no. .NET actuará como una opción más a la hora de abordar un problema. Y proporcionará valor añadido al propio T-SQL, que seguirá siendo el lenguaje por excelencia de la base de datos.

Pero como eso se discutirá más adelante en otros artículos, me limitaré a poner algunos ejemplos de esta integración y las ventajas que eso aporta.

Para empezar, la integración de .NET aporta nuevos tipos de datos (UDT o *User Defined Types*). Modelados a partir de clases y objetos, podemos decir que se aporta la filosofía OOP al esquema de las tablas de la BD. Haciendo aún más transparente los conceptos de estructuración de la información. Cada columna podrá ser de un tipo definido como clase en un ensamblado, permitiendo así el acceso al mismo empleando sus propiedades y/o métodos como si de un registro se tratase. Extremadamente interesante el temita. Eso sí, deberemos tener en cuenta, que las clases deberán cumplir un mínimo de requisitos. Pero eso ya verá con Visual Studio 2005. En el fuente 9 os pongo el ejemplo en el que se agrega como tipo una clase de tipo punto y su implicación en operaciones DML.

```
CREATE ASSEMBLY Point
FROM 'c:\tipos\Punto.dll'
GO

CREATE TYPE TipoPunto
EXTERNAL NAME Punto.TipoPuntoC
GO

CREATE TABLE Puntos
(
  idPunto INT NOT NULL IDENTITY(1,1),
  ElPunto TipoPunto
)

SELECT ElPunto.m_x, ElPunto.m_y
FROM Puntos
go

UPDATE Puntos SET ElPunto.SetXY(20, 30)
WHERE ElPunto.m_x = 0

DECLARE @Pnt TipoPunto
SET @Pnt = (SELECT Pnt FROM Puntos
           WHERE ID = 2)

SELECT @Pnt.ToString() AS Pnt
GO
```

Fuente 9



## El CLR y .NET Framework nos permite dotar a T-SQL de un valor añadido ilimitado. Pero no es un reemplazo, sino una característica más que nos ayudará a ofrecer mejores soluciones

Esta es una de las funcionalidades del CLR. Las demás, casi salen por sentido común: aplicación en la creación de procedimientos almacenados, funciones definidas por el usuario... y ¡¡¡tachaaán!!!... ¡Triggers! Lo que está claro es que la potencia está asegurada, y las aplicaciones... las que se nos pase por la imaginación –siempre respetando los requisitos del usuario, claro–.

Y además, para finalizar casi ya, es importante tener en cuenta que todo esto funciona en un entorno optimizado como es el CLR integrado en el núcleo de SQL Server y con todas sus capacidades de ejecución como el JIT, pools, etc., y la importantísima seguridad. Reforzada, eso sí, con los propios mecanismos del motor de SQL. Casi nada.

### Con buena herramienta... bien se trabaja

Para finalizar ya con esta parte del especial de T-SQL 2005, no puedo irme sin comentar nada acerca de las herramientas con las que contará un desarrollador de datos para trabajar con esta capa, con la sencillez y comodidad a la que Microsoft siempre nos tiene acostumbrados. Para esto, hablar de la que será la consola por excelencia: El SQL Server Management Studio o SSMS cariñosamente. Esta consola integra todo lo necesario para un DBA. Entre muchas de sus funciones, muy integrada está la capacidad de poder ejecutar Querys SQL a diferencia de las versiones anteriores en las que para cosas avanzadas contábamos con el Analizador de consultas. Ahora todo eso pasó, y contaremos con muchas opciones que iremos descubriendo cuanta más experiencia tengamos con las nuevas herramientas.

Desde SSMS podremos ejecutar cualquier tipo de query genérica, siendo responsabilidad nuestra el conectarnos contra un origen de datos, o bien podremos emplear el Object Explorer para que la consola habilite esas conexiones por nosotros. A partir de ese momento, y con un rico juego de colores y extensiones –tales como el arrastrar y soltar–, podremos diseñar nuestras queries con mucha más productividad que en versiones anteriores.

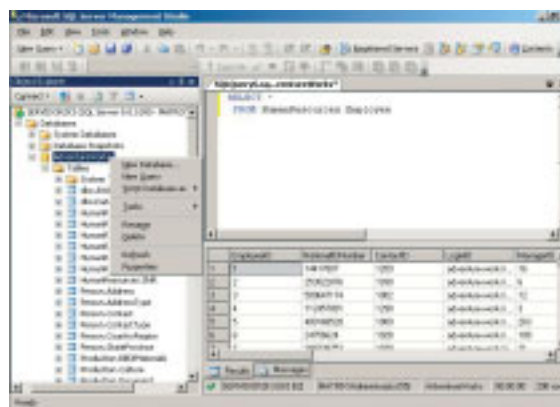


Figura 5. SSMS en acción

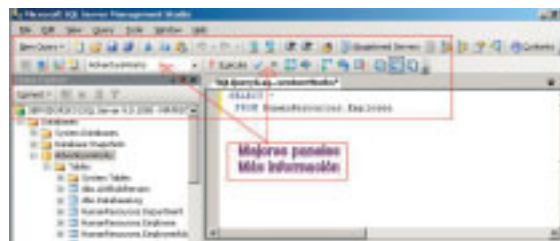


Figura 6. Toolbars de SSMS

Una novedad –además de integrar el analizador en la consola de adminis-

tración– es la de los visores de datos, que reaccionan al tipo de dato mostrado tras ejecutar una consulta. En el caso de salidas XML, por ejemplo, cada fila aparecerá adornada con hipervínculos que nos mostrarán el contenido visualmente formateado del documento XML almacenado. Una preciosidad.

Además de todo lo visto, es importante añadir también la integración existente con el nuevo Visual Studio .NET

2005. Integrado en un solo entorno de desarrollo, aparecerán múltiples herramientas para la gestión, desarrollo y depuración (sí, sí..., depuración) de objetos SQL de la base de datos... tanto a nivel de T-SQL como a nivel de CLR. Dispondremos pues de las plantillas de soluciones de bases de datos, permitiéndonos abarcar el desarrollo de la capa de datos como una pieza más de la arquitectura de una solución.

Lo dicho, que como podemos ver Microsoft nos sigue sorprendiendo a cada paso que da. Hemos visto sólo una parte de las novedades que nos ofrecen en la mayor revisión de SQL de la historia. Y esto confirma aún más la apuesta de MS por sus arquitecturas .NET Servers. SQL Server 2005 nos ofrece muchas novedades en todos sus sistemas, pero lo que está claro, es que su nuevo T-SQL es más productivo que nunca. ☺



Fernando Guerrero

# Proporcionando escalabilidad a SQL Server 2005 con Service Broker

Este artículo hace una introducción al nuevo Service Broker con el que SQL Server 2005 dota al motor relacional de capacidades nativas de gestión de colas, permitiendo crear aplicaciones de bases de datos escalables, seguras y confiables, basadas en procesos asíncronos.

## » Mensajes y procesos de negocio

La gestión asíncrona de mensajes no es nueva en la industria informática. Prácticamente desde el principio de la computación, los ordenadores se diseñaron para gestionar colas de peticiones, precisamente porque esto refleja muy fielmente los procesos reales que las empresas implementan en sus negocios.

Empecemos por el principio. Pepa acaba de montar su propia empresa de gestión de seguros y anda feliz con su nueva oficina, su mesa de despacho y su resplandeciente teléfono. Tiene un sistema informatizado que le conecta a las diferentes compañías de seguros y cuando un cliente le llama es capaz de conectarse en línea y proceder con la gestión inmediatamente.

Todo va de maravilla siempre y cuando los clientes entiendan que sólo pueden llamar cuando ella está desocupada, de modo que pueda ir atendiendo uno a uno en secuencia. Digamos que esto es como tener SQL Server en modo de usuario único, atendiendo peticiones desde una sola conexión.

Sin embargo, algunos clientes se quejan a Pepa de que cuando llaman, sólo consiguen una línea ocupada, y que esto les obliga a acudir a otros gestores de la competencia. Pepa analiza el problema y decide que debe instalar más de una línea de teléfono de modo que pueda ver cuándo alguien le está llamando y poder pasar así de una llamada a otra cuando sea necesario.

Dicho y hecho, ahora Pepa puede pasar de una llamada a otra compartiendo su tiempo entre varias llamadas simultáneas, y procesando varios asuntos al mis-

mo tiempo (al fin y al cabo su condición femenina le permite efectuar diferentes tareas a la vez).

Esto sería algo similar a una instancia de SQL Server admitiendo varias conexiones simultáneas, donde el sistema asignaría hilos de ejecución independientes a cada proceso, permitiendo una ejecución en paralelo de las peticiones de dichas conexiones.

Sin embargo, incluso con las mejores dotes telefónicas e informáticas de Pepa, aún hay clientes que no llegan a conectarse y deben dejar un mensaje en el contestador. El problema es que estos mensajes no suelen tener toda la información necesaria para proceder con el asunto que solicitan y en muchos casos no son útiles en absoluto.

Pensemos que SQL Server permite recibir peticiones de consultas por email, que serían procesadas fuera de línea mediante el procedimiento `sp_processmail`. Sin embargo, los mensajes recibidos no siempre contienen una consulta válida, y esto hace que el sistema no resulte muy eficaz realmente.

Entonces a Pepa se le ocurre una idea. Contrata un secretario que reciba las llamadas y se encargue exclusivamente de rellenar un formulario con los datos necesarios para poder procesar la petición, e instruye al secretario para que ponga todos los formularios rellenos en una bandeja de entrada. De este modo, los clientes no tienen que esperar mientras Pepa procesa las peticiones de otros clientes, ya que ha separado por completo los procesos de relleno del formulario del proceso de ejecutar la petición. Acaba de inventar el proceso asíncrono de peticiones de seguro, mediante la gestión de colas, además de que estas peticiones se atienden a un formato

Fernando Guerrero es MVP en SQL Server, y un ponente habitual en la mayoría de las conferencias internacionales de este campo. Viaja permanente alrededor del mundo como consultor, profesor y director ejecutivo de Solid Quality Learning.

estricto que le permite garantizar que no habrá peticiones ambiguas. Así que está contentísima, como podréis imaginar. La bandeja de entrada va creciendo conforme llegan nuevas peticiones, y ella las va procesando durante el día a su ritmo.

El problema es que el secretario se le queja de que hay veces que no da abasto con las llamadas. Pepa se acaba de dar de bruces de nuevo con un problema de escalabilidad. Sin embargo, se da cuenta de que podría disponer de otros secretarios a diferentes horas del día para poder atender esos picos de llamadas, por lo que configura la centralita telefónica de modo que las nuevas llamadas entrantes se dirijan automáticamente al primer secretario que esté disponible. De este modo los formularios pueden rellenarse en paralelo, y van apilándose en la bandeja de entrada a lo largo del día. Pepa acaba de inventar el escalado del servicio de relleno de formularios de entrada.

la bandeja de entrada se mantenga en niveles aceptables, y que entre los dos pueden perfectamente procesar todas las peticiones del día.

Como Pepa es una persona ambiciosa y las peticiones no paran de llegar, se da cuenta inmediatamente de que puede contratar otros agentes de seguros que colaboren en el proceso de esta bandeja de entrada. Acaba de aplicar una solución de escalabilidad al procesado de solicitudes, del mismo modo que logró escalar el relleno de formularios.

Claro que el negocio no para de crecer y ya no es posible procesar todas las peticiones en la misma oficina, por lo que llega a acuerdos de colaboración con otras oficinas a las que reenvía ciertas peticiones para su proceso. Dicho en términos informáticos, está “enrutando” las peticiones a otros procesos.

Haciendo estudios de eficiencia y reingeniería de procesos, detecta que estos mismos principios se pueden aplicar a

Este método de trabajo existe hace bastante tiempo en los grandes ordenadores corporativos, y Microsoft lo introdujo en el entorno Windows con el *Option Pack* de NT4, mediante el servicio MSMQ (*Microsoft Message Queue*).

Este servicio es compatible funcionalmente con los servicios de gestión de colas de los grandes ordenadores corporativos, como *MQSeries*, y hasta permitía conectividad con los mismos mediante el *Microsoft Host Integration Server*.

Sin embargo, la utilización de estos sistemas de colas requiere un diseño distinto de las aplicaciones y esto ha hecho que no todos los programadores hayan encaminado la arquitectura de sus aplicaciones por esta ruta.

Quizá se haya sobreestimado la dificultad de programación que estos sistemas conllevan, pero la realidad es que no ha tenido el alcance que debiera entre la comunidad de programadores.

La llegada de .NET facilita la utilización de estos servicios mediante el espacio de nombre **System.Messaging** diseñado para ofrecer soporte mediante código gestionado para los servicios de MSMQ.

Sin embargo, esto está muy lejos del alcance típico de un programador de bases de datos. Si trabajar con .NET para un programador de Transact-SQL le supone un serio problema, intentar lidiar con MSMQ ya está fuera de lugar completamente.

---

## ...al desarrollar SQL Server 2005 se decidió dotar al motor relacional de capacidades nativas de gestión de colas, de modo que estas aplicaciones pudieran beneficiarse de la escalabilidad y prestaciones que este sistema proporciona

---

Como todo en esta vida, nada sale gratis, y todas las soluciones traen de la mano problemas inesperados. El servicio escalable de relleno de formularios funciona tan eficientemente, que cada día hay más clientes que llaman a la empresa de Pepa para solicitar seguros. Pepa no tiene problemas contratando nuevos secretarios, pero la bandeja de entrada va creciendo y creciendo y no da abasto para poder procesar todas estas peticiones durante el día.

Piensa y piensa hasta que se le ocurre contratar a un amigo, profesional de seguros como ella, para que le eche una mano en procesar esta siempre creciente bandeja de entrada. Y se da cuenta de que con la ayuda adecuada, consigue que

la subdivisión de ciertos procesos de modo que pueda obtener beneficios del procesado asíncrono de peticiones, y de la flexibilidad que la gestión de colas le proporciona.

Desde el punto de vista del usuario, la transacción de negocio comienza en el momento en que envía la petición a la empresa de Pepa, y finaliza cuando recibe una respuesta final del resultado de su petición. Esto puede llevar algún tiempo, y no sería adecuado implementar este proceso completo como una única transacción física a nivel de base de datos. Sin embargo, podría ser implementada como un conjunto de transacciones físicas que formaran parte de la misma transacción lógica.

## ¿Y qué pinta SQL Server en toda esta discusión?

Para muchos programadores de aplicaciones basadas en plataformas de Microsoft, la única razón por la cual se crearon sistemas de colas en sus aplicaciones fue para escalar peticiones a SQL Server.

Además, en muchos de estos casos, se seleccionó que las colas se implementaran físicamente como tablas en una instancia de SQL Server.

Así que si tenemos que todo el proceso, desde la implementación física de las colas hasta los procesos de negocio que requieren de estos servicios, se relacionan directamente con SQL Server, ¿para qué involucrar a un servicio externo como MSMQ en este trabajo?





Por esta razón, al desarrollar SQL Server 2005 se decidió dotar al motor relacional de capacidades nativas de gestión de colas, de modo que estas aplicaciones pudieran beneficiarse de la escalabilidad y prestaciones que este sistema proporciona. Este sistema se denomina *Service Broker*, y no es un servicio de Windows independiente del servicio de SQL Server, sino que forma parte del motor relacional mismo.

Los mensajes son enviados entre servicios mediante TCP/IP, por lo que la arquitectura está diseñada nativamente para soportar entornos distribuidos.

¿A que suena muy parecido a MSMQ? Naturalmente. *Service Broker* es una evolución del paradigma definido por MSMQ, pero diseñado para optimizar las aplicaciones de bases de datos.

Una vez que SQL Server 2005 implementa un sistema como este, ¿por qué no utilizarlo para otros menesteres? De hecho, la notificación de eventos, así como la notificación de cambios en los resultados de las consultas (nueva posibilidad de ADO.NET 2.0) se implementan mediante *Service Broker*.

## Service Broker es una evolución del paradigma definido por MSMQ, pero diseñado para optimizar las aplicaciones de bases de datos

*Service Broker* permite crear aplicaciones de bases de datos escalables, seguras y confiables, basadas en procesos asíncronos, tanto en sistemas de una sola instancia de SQL Server, como en complejos sistemas distribuidos con múltiples instancias de SQL Server geográficamente dispersas.

*Service Broker* se basa en la creación de servicios independientes, que interactúan entre sí mediante mensajes, transmitidos asíncronamente mediante colas, y formateados de acuerdo a contratos preestablecidos. Ojo, al hablar aquí de servicios no nos estamos refiriendo a servicios de Windows, sino a instancias de procesos que se encargan de procesar mensajes gestionados por *Service Broker*. Estos servicios podrían ser activados automáticamente por *Service Broker*, si así se prefiere.

Por supuesto *Service Broker* utiliza las nuevas características de seguridad de SQL Server 2005, como encriptación de mensajes, seguridad basada en certificados y claves simétricas, para que el proceso de mensajes sea tan seguro como requieren los entornos en los que se utilizan este tipo de aplicaciones.

### ¿Y qué pasa con las aplicaciones que ya tengo escritas?

Me alegra que me haga esta pregunta: Si tienes aplicaciones que siguen la recomendación de interactuar con la base de datos mediante procedimientos almacenados, estás de enhorabuena. En este caso te estás conectando a la base de datos mediante interfaces bien definidos, y no debes cambiarlos, para no romper tu aplicación.

Mantén estos interfaces, manteniendo los mismos nombres de tus procedimientos y los mismos parámetros, pero traslada la lógica de estos procedimientos (su implementación) a otros procedimientos que vas a invocar como servicios de *Service Broker*.

Tus procedimientos originales recibirán las peticiones de tu aplicación mediante los parámetros de entrada, y los enviarán a unas colas creadas a tal efecto. Estos procedimientos de entrada no harán nada más.

Tras esto, los nuevos procedimientos que has creado para albergar la lógica que ya tenían implementada, leerán peticiones de estas colas y ejecutarán las acciones correspondientes, enviando los resultados a una o varias colas de salida.

Deberán crear otros procedimientos que sirvan como servicios de recogida de resultados, y que comuniquen el estado de estos procesos a la aplicación de cliente.

¿La ventaja de este proceso? Tu aplicación de cliente se mantiene más o menos intacta. Y los cambios introducidos en la base de datos se implementan mediante instrucciones Transact-SQL que se encargan de gestionar el proceso de los mensajes disponibles en las colas de procesos.

¿Otra ventaja fundamental? Puedes perfectamente implementar los servicios de proceso mediante aplicaciones externas a SQL Server, incluso como servicios Windows, ejecutándose en máquinas remotas, escalando muy rápidamente el sistema, sin cambiar drásticamente el modo en que se diseñó originalmente la aplicación de cliente.

Hay un cambio fundamental en la aplicación de cliente: debe consultar el estado de las peticiones enviadas, ya que estas peticiones se procesan asíncronamente. Pero esta consulta no es necesario efectuarla inmediatamente tras el envío de la petición.

### Objetos de Service Broker

*Service Broker* utiliza diferentes tipos de objetos, y cada uno de ellos proporciona una funcionalidad muy específica. Describir en detalle cada uno de estos objetos sería demasiado largo para un artículo de introducción como éste,

pero al menos es interesante entender el cometido de cada uno de ellos.

## MESSAGE TYPE

Esta es la definición de un tipo de mensaje, donde especificamos si se trata de un mensaje binario o XML, y en ese caso, si se trata de un mensaje *XML typed* (con un formato específico definido por un esquema conocido) o no.

El tipo de mensaje se define mediante una instrucción como esta (para mensajes XML con estructura válida):

```
CREATE MESSAGE TYPE
[//Adventure-Works.com/Gastos/EnviarGastos]
VALIDATION = WELL_FORMED_XML;
```

## CONTRACT

Un contrato define qué tipo de mensaje puede enviar y recibir cada servicio. Por ejemplo:

```
CREATE CONTRACT
[//Adventure-Works.com/Expenses/ExpenseSubmission]
( [//Adventure-Works.com/Expenses/SubmitExpense]
  SENT BY INITIATOR,
[//Adventure-Works.com/Expenses/ExpenseApprovedOrDenied]
  SENT BY TARGET,
[//Adventure-Works.com/Expenses/ExpenseReimbursed]
  SENT BY TARGET
) ;
```

## QUEUE

Las colas son los objetos que almacenan los mensajes y a los cuales acceden los servicios para leer y enviar dichos mensajes. Estas colas pueden especificar un servicio de activación automática cuando lleguen mensajes nuevos, aunque esto no es necesariamente lo más común. Un ejemplo de creación de una cola podría ser este:

```
CREATE QUEUE ExpenseQueue
WITH STATUS=ON,
ACTIVATION (
  PROCEDURE_NAME = expense_procedure,
  MAX_QUEUE_READERS = 5,
  EXECUTE AS 'ExpenseUser' ) ;
```

En el que definimos su nombre, *ExpenseQueue*, la activamos y además definimos que puede lanzar hasta 5 instancias del servicio *expense\_procedure* para leer mensajes de esta cola, y que la ejecución de este servicio (un procedimiento almacenado en este caso) se va a efectuar bajo las credenciales de el usuario *ExpenseUser*.

## SERVICE

Un servicio de *Service Broker* es el nombre que se le da a una tarea o conjunto de tareas. Por ejemplo:

```
CREATE SERVICE [//Adventure-Works.com/Expenses] ON QUEUE ExpenseQueue
([//Adventure-Works.com/Expenses/ExpenseSubmission],
[//Adventure-Works.com/Expenses/ExpenseProcessing]) ;
```

Este servicio, *Expenses*, se crea en una cola determinada, *ExpenseQueue*, y permite dos contratos distintos: *ExpenseSubmissions* y *ExpensesProcessing*.

## Diálogos y Conversaciones

Todos los mensajes enviados por *Service Broker* forman parte de una conversación. Un diálogo es una comunicación entre dos servicios, o más precisamente una corriente persistente y confiable de mensajes entre dos servicios distintos.

Por ejemplo, este sería el modo de comenzar una conversación:

```
DECLARE @dialog_handle UNIQUEIDENTIFIER ;

BEGIN DIALOG CONVERSATION @dialog_handle
FROM SERVICE [//Adventure-Works.com/ExpenseClient]
TO SERVICE '//Adventure-Works.com/Expenses'
ON CONTRACT [//Adventure-Works.com/Expenses/ExpenseSubmission] ;
```

En este ejemplo, establecemos este diálogo entre dos servicios, *ExpenseClient* y *Expenses*, utilizando un contrato determinado.

El valor del puntero *@dialog\_handle* nos servirá luego para enviar y recibir mensajes como parte de esta conversación.

Las conversaciones se pueden agrupar en grupos de conversaciones, las cuales típicamente agrupan diferentes conversaciones dentro de una misma transacción, lo cual es muy útil para complejos entornos distribuidos.

## Enviar Mensajes

Como hemos dicho antes, los mensajes se envían como parte de una conversación, con un tipo de mensaje válido para el contrato declarado para dicha conversación, como en este ejemplo:

En este caso, el contenido de la variable

```
SEND ON CONVERSATION @dialog_handle
MESSAGE TYPE [//Adventure-Works.com/Expenses/
  SubmitExpense]
(@ExpenseReport) ;
```

*@ExpenseReport*, podría ser un texto XML o cualquier dato binario que el servicio de destino sea capaz de procesar.

## Recibir Mensajes

Los mensajes los leemos de las colas mediante la instrucción **RECEIVE**, que devuelve un conjunto de registros. De algún modo la instrucción **RECEIVE** sobre una cola se comporta como una instrucción **SELECT** sobre una tabla (de hecho algo así sucede), pero extrayendo los mensajes de la cola para su procesado.

```
RECEIVE TOP 1 *
FROM ExpenseQueue
WHERE conversation_handle = @conversation_handle ;
```

En este caso estamos recuperando sólo el primer mensaje de la cola **ExpenseQueue** relativa a la conversación definida por el puntero **@conversation\_handle**.

## Usos típicos de Service Broker

Podemos pensar en muchas aplicaciones prácticas para *Service Broker*, pero estas son algunas de las más importantes:

### Gestión asíncrona de disparadores

La ejecución de disparadores (*triggers*) supone siempre una sobrecarga para cualquier aplicación. Esta sobrecarga será mayor o menor en función de la complejidad de este disparador. Sin embargo, si este disparador pudiera enviar un mensaje a un servicio de *Service Broker* para su proceso asíncrono, el disparador podría devolver control a la aplicación inmediatamente, mejorando sensiblemente el tiempo de respuesta.

Esto supone una mejora sustancial en la concurrencia de la aplicación, ya que la transacción finalizará más rápidamente, liberando los bloqueos que pudiera mantener, y dejando así vía libre a otros procesos para que pudieran utilizar estos recursos.

### Proceso asíncrono de colas

En muchos casos, las aplicaciones deben ser diseñadas de tal modo que sean insensibles a problemas de comunicación debidos a fallos de comunicaciones, problemas de falta de energía eléctrica, fallos

de servidores y demás interrupciones de servicio. Implementando estas aplicaciones alrededor de servicios de *Service Broker*, cada vez que un mensaje debe ser procesado y esto no es posible por alguno de estos problemas, la transacción física que se encarga de procesar este mensaje tendría que cancelarse, llevando el mensaje de vuelta a la cola, y el servicio podría intentar procesarlo de nuevo más adelante.

La gran ventaja de este sistema es que este proceso se efectuaría automáticamente sin necesidad de una programación específica, ya que la infraestructura la proporciona *Service Broker*.

### Recopilación de datos

Aplicaciones que deben recopilar datos desde diferentes fuentes, pueden aprovecharse de la naturaleza asíncrona de *Service Broker*, ya que no es necesario que origen y destino de los datos estén disponibles simultáneamente. Este tipo de aplicaciones pueden conectarse al origen de datos y enviar los datos como mensajes a un servicio de *Service Broker* para su proceso posterior. *Service Broker* podría crear tantas instancias de estos servicios como fuera necesario para procesar estos datos e irlos enviando a su destino final.

De hecho, algo similar ha sido implementado en SQL Server 2000 por DTS y por replicación, para incorporar sistemas de colas en la importación de datos y en la replicación.

### Procesos distribuidos

Supongamos que un sistema de pedidos debe coordinar piezas y componentes de diferentes proveedores, los cuales es más que probable que no estén en línea al mismo tiempo. Este tipo de procesos sería un excelente candidato para ser gestionado por *Service Broker*, ya que el sistema podría aceptar pedidos, independientemente de cómo se procesan estos pedidos. El sistema definiría servicios que se encargarían de procesar las peticiones individuales a cada uno de los proveedores, pero todas estas peticiones formarían parte de la misma transacción lógica.

## Consolidación de datos

Todos estamos acostumbrados a utilizar páginas Web para encontrar información. Si la información a mostrar se obtiene de diferentes fuentes, sería posible enviar peticiones en paralelo a esas fuentes, utilizando *Service Broker*, de modo que el tiempo de respuesta global se minimizara, incrementando el rendimiento de la aplicación desde el punto de vista del usuario.

### Proceso por lotes para grandes aplicaciones

Este quizá sea el escenario más tradicionalmente aceptado para ser implementado con *Service Broker*, especialmente en sistemas bancarios. Las aplicaciones de cliente ejecutándose en computadoras distribuidas, envían peticiones a una cola, donde esperan ser procesadas más tarde. Otros ordenadores podrían ejecutar servicios a intervalos regulares que se encarguen de procesar estas peticiones.

No estamos hablando aquí del típico proceso nocturno por lotes, sino de la posibilidad de procesar colas en tiempo prácticamente real, ya que cada capa de estos sistemas podría escalar-se para darle mayor capacidad bajo demanda.

## Conclusiones

Estamos seguros de que *Service Broker* cambiará la forma en que muchos programadores diseñan aplicaciones de bases de datos con SQL Server 2005.

La posibilidad de diseñar procesos asíncronos, que permitan flexibilizar el proceso de cada etapa dentro de los procesos de negocio, dará una mayor agilidad y escalabilidad a las aplicaciones de bases de datos, incrementando notablemente la satisfacción de los usuarios.

Esto no es una afirmación de un experto en marketing, sino la opinión personal de un arquitecto de sistemas de bases de datos que, tras llevar años lidiando con este tipo de problemas, ve en esta nueva tecnología unas posibilidades extraordinarias. ○





Marino Posadas

# Introducción a SMO

En versiones anteriores de SQL Server, se suministraba un mecanismo de automatización del servidor, basado en una librería de tipos COM (SQL-DMO), para permitir el acceso mediante programa a características de funcionamiento del servidor y sus objetos dependientes. En esta versión, el concepto de programabilidad del servidor se ha extendido notablemente, permitiéndonos manejar mediante código .NET casi cualquier característica de SQL Server 2005. Es el modelo SMO.

» **De hecho, el cambio** de nomenclatura a SMO ya nos sugiere las nuevas posibilidades: *Objetos para gestión del sistema*. De todo el sistema: y esto incluye a versiones anteriores de SQL Server (7 y 2000). Paralelamente a estas librerías, Microsoft incluye también la librería para la gestión de *Réplicas de Bases de Datos* (RMO, o *Replication Management Objects*), aunque ésta no sea el objetivo principal de este artículo.

Aunque el conjunto de librerías vinculadas a SMO incluye otros elementos, dos de ellas forman la parte troncal del sistema:

- `microsoft.sqlserver.smo.dll` (librería principal).
- `microsoft.sqlserver.smo.connectioninfo.dll` (datos de conexión).

En concreto, la primera suministra un paquete funcional muy variado, con mención especial a los siguientes apartados:

- Servicios Generales de SMO.
- Instanciación optimizada.
- Ejecución diferida.
- Funcionalidad WMI (Windows Management Instrumentation).
- *Scripting* avanzado.
- Configuración de opciones del Servidor.

La segunda librería suministra servicios de información y configuración de conexiones. Genéricamente hablando, podríamos dividir conceptualmente la amplísima jerarquía de clases de SMO en dos tipos: *clases de instancia* y *clases de soporte de utilidades*.

Las primeras, representan los objetos principales que encontramos en el servidor: *Server*, *DataBase(s)*, *Schema(s)*, *StoreProcedure(s)*, *Table(s)*, *View(s)*, *Column(s)*,

etc. En la segunda categoría encontramos clases del tipo: *Backup*, *Restore*, *Scripter* y *Transfer* (para transferencias de datos entre instancias de SQL Server a través de canales DTS).

## Programación con SMO

La programación con SMO desde Visual Studio 2005 (usamos la Beta 2 para este artículo), permite comenzar el trabajo desde cualquier aplicación (Consola, Windows o Web) siempre que hagamos referencia a las librerías citadas más arriba. En la figura 1 puede verse.

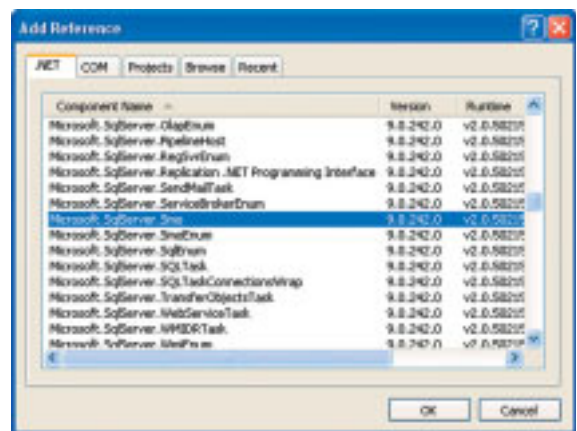


Figura 1. Referencia a las librerías principales de SMO desde V.Studio 2005

La ventana de “Referencias” mostrando las distintas librerías programables que SQL Server 2005 pone a disposición del desarrollador. Note el lector que en la columna *Runtime* se especifica la versión del Framework necesaria para su funcionamiento (2.0.50215). Esto respon-

**Marino Posadas** es asesor técnico y redactor de dotNetManía, MVP de C# y formador de Alhambra-Eidos

de a la pregunta de si dichas librerías serían utilizables desde Visual Studio .NET 2003. Las librerías son instaladas en el GAC (*Global Assembly Cache*) por la opción de instalación de herramientas de cliente de SQL Server 2005.

Una vez adjuntas las librerías básicas, las sentencias de referencia nos dan acceso al nutrido conjunto de clases de SMO. Pongamos un primer ejemplo básico de utilización de SMO para conectarnos a una instancia de SQL Server 2005 y obtener información diversa del servidor.

Los dos fragmentos interesantes de código son la referencia a las librerías (ver fuente 1), y las 5 líneas de código que incluimos en el constructor de la clase del formulario, para instanciar un objeto de la clase `ServerConnection` y leer propiedades del servidor (ver fuente 2).

```
using Microsoft.SqlServer.Management.Smo;
using Microsoft.SqlServer.Management.Common;
```

Fuente 1

```
//Código de acceso al Servidor SQL-Server 2005
ServerConnection SC = new ServerConnection(@"MARINO-XP-PRINC\SQL05CTP");
this.richTextBox1.Text = "Cadena de Conexión: " + SC.ConnectionString + "\n";
this.richTextBox1.Text += "Nombre del servidor: " + SC.TrueName + "\n";
this.richTextBox1.Text += "Protocolo de red: " + SC.NetworkProtocol.ToString() + "\n";
this.richTextBox1.Text += "Número de Versión: " + SC.ServerVersion.BuildNumber.ToString() + "\n";
this.richTextBox1.Text += "Perfil de usuario: " + SC.UserProfile.ToString() + "\n";
```

Fuente 2

Lo que produce la siguiente salida en ejecución:



Figura 2. Salida en ejecución del fuente anterior

Es curioso que la única referencia para el acceso sea el propio nombre de la instancia de SQL Server, asumiéndose en este caso las credenciales de confianza y la desactivación de MARS (*Multiple Active Result Sets*) (vea la sección *TodoNet Q&A* de **Dino Esposito** en este mismo número).

```
Server Servidor = new Server(@"MARINO-XP-PRINC\SQL05CTP");
this.richTextBox1.Text += "Nombre del servidor: " + Servidor.Name + "\n";
this.richTextBox1.Text += "Estado del servidor: " + Servidor.State.ToString() + "\n";
this.richTextBox1.Text += "Lenguaje por defecto: " +
    Servidor.Configuration.DefaultLanguage.Description + "\n";
this.richTextBox1.Text += "Plataforma: " + Servidor.Information.Platform + "\n";
this.richTextBox1.Text += "BB.DD: " + Servidor.Databases[0].Name.ToString() + "\n";
this.richTextBox1.Text += "Plataforma: " + Servidor.Information.Platform + "\n";
this.richTextBox1.Text += "Memoria: " + Servidor.Information.PhysicalMemory + " Mb. \n";
this.richTextBox1.Text += "Versión: " + Servidor.Information.VersionString + "\n";
this.richTextBox1.Text += "Perfil de usuario: " +
    Servidor.ConnectionContext.UserProfile.ToString() + "\n";

foreach (Database db in Servidor.Databases)
{
    this.richTextBox1.Text += db.Name + ':' + db.Size + "K \n";
}

foreach (Table t in Servidor.Databases[0].Tables)
{
    this.richTextBox2.Text += t.Name + "\n";
}

foreach (Column c in Servidor.Databases[0].Tables[0].Columns)
{
    this.richTextBox3.Text += c.Name + "\n";
}
```

Fuente 3

Más potente y apropiada para estas tareas resulta la clase `Server`. Nos presen-

código, por otro lado, no puede ser más simple. La esperada salida del programa se muestra en la figura 3.



Figura 3. Información extendida mediante la clase `Server`

ta todas las características del propio servidor y además, las de las bases de datos manejadas por él, junto a todos sus objetos relacionados: *Tablas*, *Columnas*, etc.

En el siguiente ejemplo, utilizamos esta clase, no sólo para leer información similar a la anterior, sino para acceder a estructuras de metadatos de la base de datos predeterminada, recuperando nombres de tablas y columnas en sendas solapas de un control `TabControl`. El código fuente utilizado es el del fuente 3.

Como puede apreciar el lector, la misma clase servidor es capaz de acceder a información del sistema, como la memoria instalada (vía WMI, en este caso), a la información propia de configuración, y también a la de las bases de datos que gestiona, incluyendo diversos niveles de profundidad jerárquica. El

Pero aquí no terminan (casi no hacen más que comenzar), las posibilidades de SMO. También es posible crear objetos, recuperar datos, y –en suma– gestionar cualquier cosa que una aplicación de gestión pueda necesitar de un servidor.

## Creación de objetos en el servidor

El proceso de creación de objetos complejos, tales como tablas de bases de datos, sigue una dinámica similar a la que .NET ya nos tiene acostumbrados. Todos los objetos se articulan en forma de colecciones, algunos de cuyos miem-

bros, son –a su vez– otras colecciones. El código fuente, sólo tiene que ir creando los objetos de más bajo nivel (columnas), para ir añadiéndoles a las colecciones de tablas, que, por su parte, serán finalmente añadidas a las bases de datos. Puede ver un ejemplo sencillo en el fuente 4.

```
//Creación de objetos en el servidor
Database BDDAdventureWorks = Servidor.Databases["AdventureWorks"];
Table Descuentos = new Table(BDDAdventureWorks, "Descuentos");
Descuentos.Schema = "Sales";
Column IDDescuento = new Column(Descuentos, "IDDescuento", DbType.Int);
Column Concepto = new Column(Descuentos, "Concepto", DbType.NVarChar(40));

//Asignación de propiedades específicas de columnas
IDDescuento.Identity = true;
Concepto.DefaultSchema = "Sales.Descuentos";
//Construcción de la jerarquía de objetos y creación final.
Descuentos.Columns.Add(IDDescuento);
Descuentos.Columns.Add(Concepto);
Descuentos.Create();
```

Fuente 4

Como puede verse, operamos de forma escalonada. Primero, una referencia a la base de datos. A partir de ahí la creación del objeto `Table`, indicando su nombre. Como cada tabla del servidor puede pertenecer a un *schema* o al predeterminado *dbo*, asignamos el esquema correspondiente, perfilamos las características de comportamiento del campo identidad, añadimos los objetos `column` a la colección (`Columns`), y cuando todo está listo, lanzamos el método `Create()`, que se encarga de la creación física de los objetos en el servidor, como podemos ver en la ventana de administración de SQL Server Management Studio de la figura 4.

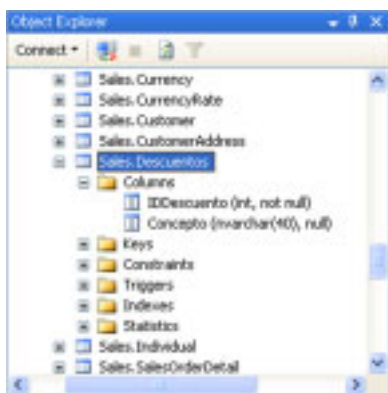


Figura 4

Sólo una puntualización: si la conexión está operando en el modo predeterminado, la sentencia de creación se envía inmediatamente al servidor. Si –por el contrario– hubiéramos preferido operar en modo *CaptureSQL*, las sentencias se almacenarían para un uso posterior.

Por otro lado, las clases manejadas hasta aquí disponen de mecanismos de modificación y borrado de objetos, cuya sintaxis sigue reglas de coherencia mnemotéc-

nica con las sentencias T-SQL asociadas en el servidor (`Alter()`, para modificar, `Drop()` para borrar, etc.).

Pero, como apuntamos más arriba, no es ésta la única funcionalidad posible, y dentro del apartado de “Utilidades” encontramos clases que pueden ser de extrema utilidad para los administradores de bases de datos. Un buen ejemplo de esto está en las capacidades de copia de seguridad y restauración, que se exponen mediante las clases `Backup` y `Restore`, respectivamente.

Concluimos, pues, esta breve introducción a SMO con un par de ejemplos de estas capacidades.

```
//Asumimos que disponemos de un objeto Server (Servidor)
Backup CopiaSeguridad = new Backup();
CopiaSeguridad.DataBase = "AdventureWorks";
CopiaSeguridad.DeviceType = DeviceType.File;
CopiaSeguridad.Devices.Add(@"C:\Backups\CopiaAW.bak");
CopiaSeguridad.Action = BackupActionType.Database;
CopiaSeguridad.SqlBackup(Servidor);
```

Fuente 5

Como vemos, sigue la misma línea de sencillez que el código anterior. Se crea un objeto `CopiaSeguridad` y se asigna su origen, tipo de almacenamiento, ubicación física y acción a realizar. Finalmente, el método `SqlBackup()` genera las órdenes y produce la copia.

Para la restauración el proceso, es muy similar:

```
//Asumimos que disponemos de un objeto Server (Servidor)
Restore CopiaSeg = new Restore();
CopiaSeg.DeviceType = DeviceType.File;
CopiaSeg.Devices.Add(@"C:\Backups\CopiaAW.bak");
bool Verificación = CopiaSeg.SqlVerify(Servidor);
if (Verificación == true)
    MessageBox.Show("Restauración correcta");
else MessageBox.Show("La copia no es operativa");
```

Fuente 6

Como se ve, el mecanismo inicial es similar, siendo el método `Add()` del objeto `Restore` el encargado de cargar la copia en el sistema. Para más seguridad, añadimos un pequeño control de verificación que el objeto `Restore` permite mediante `SqlVerify()` sobre el servidor correspondiente. Todo puro .NET Framework.

## Conclusión

Estamos ante un subsistema de gran potencia a la hora de manipular las bases de datos. Nos permite acceder a cualquier recurso del servidor (también a los propios datos) desde un programa .NET realizado en cualquiera de los 3 modelos (Consola, Windows o Web), capacitando para la creación de soluciones .NET de alto rendimiento, que incluyan mecanismos de mantenimiento del servidor sin recurrir a SQL Server Management Studio, de forma clara y sencilla. ○



Todas las piezas de su negocio son importantes. Internet también.



La presencia en Internet de su empresa es una pieza fundamental para su negocio. Sin ella, todo puede caer. Por esta razón, usted debe elegir un socio tecnológico con infraestructura propia, con un soporte técnico profesional, con una gran experiencia. Usted debe elegir acens.

dominios desde **18** €/año

- .com, .net, .org, .info, .biz...
- Agente Registrador (Red.es) para .es, .com.es, .nom.es y .org.es
- servicios legales (LSSI, LOFD, marcas, etc.)

alojamiento web desde **23** €/mes

- Windows Server 2003 y Linux
- arquitectura de alta disponibilidad y balanceo de carga
- correo, estadísticas, streaming, comercio electrónico, etc.
- potente y exclusivo panel de control para gestión propia

correo profesional desde **2** €/mes

- antivirus, redirecciones, webmail trilingüe, etc.
- desde uno hasta miles de buzones por dominio
- asignación flexible (por panel de control) de espacio por buzón

servidores desde **120** €/mes

- privados y dedicados de última generación
- múltiples servicios: estadísticas, firewall, monitorización, backup, etc.
- plataformas a medida: en cluster, balanceo, etc.

ADSL desde **39** €/mes

- diseñado para empresas y profesionales
- router Zykel wi-fi o Cisco gestionado
- IP Fija
- con soluciones de correo y hosting profesional

hágase distribuidor

- la gama más amplia de servicios profesionales: hosting compartido, dedicado, housing y conectividad (ADSL y Redes Privadas Virtuales)
- soporte técnico-comercial dedicado en exclusiva al canal
- sistema personalizable (incluso marca blanca)
- descuentos de hasta el 60% respecto al PVP acens

Precios sin IVA. \*Costes mensuales de referencia por contratación anual



Pedro Pozo

# SQL Server 2005 Express Edition

## El menor de la familia

Dicen que las mejores cosas de la vida son gratis. SQL Server 2005 Express también lo es. Se trata del sucesor de MSDE y del hermano menor de las ediciones de SQL Server 2005. Pero no por ello está exento de interés ya que tiene el mismo motor de base de datos que sus hermanos mayores.

» Lo primero que debemos aclarar es que esta edición Express, no se trata de una versión demo ni de una versión de pruebas con fecha de caducidad, se trata de un completo sistema de base de datos, incluso tiene el mismo motor de base de datos que las otras ediciones superiores. Además es totalmente gratuita y podemos distribuirla junto con nuestros desarrollos.

Pero por supuesto tiene sus limitaciones, que están a nivel de sistemas, ya que esta versión sólo se ejecuta en máquinas con un hardware limitado, no permitiéndonos la escalabilidad o el alto rendimiento que nos permiten las ediciones superiores de este mismo producto. Aunque sí que nos resultará muy sencillo escalar bases de datos SQL Server 2005 Express a SQL Server 2005 Workgroup, Standard o Enterprise ya que se trata del mismo motor de base de datos.

Este producto está dirigido a desarrolladores no profesionales o estudiantes que gracias a esta edición gratuita pueden utilizar y conocer las características de SQL Server. Por tanto en la mayoría de los casos puede ser muy útil ya que se trata de un completo sistema de base de datos, pero para entornos de producción se recomienda una de las ediciones superiores de SQL Server.

### Características

SQL Server 2005 Express puede considerarse el sucesor de MSDE, aunque con mejoras como una mayor sencillez para su gestión y un mejor rendimiento.

Este producto tiene el mismo motor de base de datos que toda la familia SQL Server 2005, con todas las ventajas que eso implica, y que por supuesto utiliza el mismo lenguaje Transact-SQL. Podemos crear los mismos procedimientos almacenados, *triggers*, *views* y demás funcionalidades de SQL Server. Incluso podemos utilizar esa nueva característica tan esperada por todos en SQL Server 2005, la de programar nuestros procedimientos almacenados con C#.

---

...no se trata de una versión demo ni de una versión de pruebas con fecha de caducidad, se trata de un completo sistema de base de datos, incluso tiene el mismo motor de base de datos que las otras ediciones superiores

---

Otra nueva característica es la movilidad de las bases de datos de un servidor a otro con **xcopy**. Gracias a esta nueva utilidad podemos mover un fichero de base de datos MDF de una máquina a otra, y ya estará lista nuestra base de datos para seguir funcionando. Esta utilidad es muy práctica cuando queremos distribuir nuestra aplicación junto con la base de datos, de forma que adjuntamos nuestro fichero de base de

datos en la distribución de la aplicación y ya estará listo para usar sin necesidad de realizar ninguna configuración adicional. Para poder utilizar esta base de datos deberemos utilizar la opción **AttachDBFilename** en nuestra cadena de conexión, que nos permite conectar con nuestra base de datos que ha sido cambiada de localización.

En cuanto a las posibilidades para trabajar con XML, se ha incluido soporte nativo del mismo. Disponemos de un nuevo tipo de datos XML, integrándose así perfectamente el sistema de base de datos con este lenguaje universal, permitiéndonos hacer desarrollos mucho más versátiles.

Entre las nuevas características también están el soporte de replicación y *SQL Service Broker*. El soporte de replicación consiste en tener una base de datos duplicada en varios servidores utilizando el modelo *editor-suscriptor* que se sincroniza en unos intervalos de tiempo predefinidos; mientras que *SQL Service Broker* nos permite trabajar de modo desconectado, aunque realmente *Service Broker* sólo funciona en combinación con otras ediciones de SQL Server 2005. Este nuevo servicio consiste en la comunicación entre servidores SQL Server 2005 mediante contratos de mensajería llamados diálogos. Puede ver mucho más sobre *Service Broker* en el excelente artículo de **Fernando Guerrero**.

Además, Microsoft ha creado una nueva aplicación que sirve de interfaz de usuario para gestionar bases de datos y que podemos descargar de forma gratuita de Internet. Esta nueva aplicación se llama *SQL Server Express Manager* y la comentaremos más adelante en este mismo artículo.

En cuanto a los requisitos hardware y software necesarios para instalar SQL Server 2005 Express, podemos verlos en la tabla siguiente:

Requerimientos Hardware	Requerimientos Software
Microprocesador 600mHz (recomendado 1GB)	Windows 2000, Windows XP Professional ó Windows 2003
Memoria RAM 256MB (recomendado 512)	Microsoft .NET Framework 2.0
170MB de espacio libre en disco duro	Microsoft Internet Explorer 6.0 SP1 o mayor

Tabla 1

## Diferencias

Ya hemos comentado con anterioridad que esta edición Express tiene el mismo motor de base de datos que las ediciones superiores, por lo cual pasar una base de datos creada con SQL Server 2005 Express a SQL Server 2005 Workgroup, Standard

o Enterprise es un proceso inmediato que no supone ningún problema.

Por entonces ¿en qué se diferencia la edición Express del resto de ediciones?. Pues bien, principal-

---

**Otra nueva característica es la movilidad de las bases de datos de un servidor a otro con XCOPY. Gracias a esta nueva utilidad podemos mover un fichero de base de datos MDF de una máquina a otra, y ya estará lista nuestra base de datos para seguir funcionando**

---

mente en que tiene limitaciones en cuanto al hardware en el que puede ser instalado, y que no tiene características de alta disponibilidad e inteligencia de negocios.

Esta edición Express tiene sus limitaciones a nivel de hardware, que son las siguientes:

- Tamaño máximo de bases de datos de 4 GB.
- Tamaño máximo de memoria RAM de 1 GB.
- Un único microprocesador (aunque puede funcionar en un sistema multiprocesador sólo utilizará uno).

También tiene otras diferencias con respecto a las ediciones superiores, que son:

- No dispone de características de alta disponibilidad como *data mirroring* o *clustering*.
- No dispone de búsqueda de texto completa.
- No tiene *SQL Agent*.
- No dispone de servicios de notificación y *data-warehouse*.

En la tabla siguiente podemos ver una comparación de las limitaciones de hardware que tiene cada una de las ediciones de SQL Server 2005.

Express	Workgroup	Standard	Enterprise
1 CPU	2 CPU	4 CPU	Ilimitada
1 GB RAM	3 GB RAM	Ilimitada	Ilimitada
4GB Base Datos	Ilimitada	Ilimitada	Ilimitada

Tabla 2

Sin embargo, SQL Server Express, es totalmente gratuito, y podemos incluirlo en la distribución de nuestras aplicaciones, sin que nos suponga ningún incremento en los gastos de nuestros desarrollos.



Podemos considerar, por tanto, a Microsoft SQL Server 2005 Express como una base de datos ligera, portable y de sencillo manejo, que podrá ser utilizada por una amplia variedad de usuarios debido a su carácter gratuito

## SQL Express Manager

Al instalar SQL Server 2005 Express podemos comprobar que la única herramienta de usuario que nos proporciona es *SQL Computer Manager* que sirve para gestionar los servicios básicos de SQL Server y configurar los protocolos de red. Por tanto, Microsoft también ha desarrollado una aplicación para gestionar nuestras bases de datos, que es también gratuita y muy sencilla de utilizar: se trata de *SQL Express Manager*.

Esta aplicación tiene un interfaz de usuario muy sencillo de utilizar, con la que podemos realizar las tareas básicas de gestión de una base de datos, ayudados en muchas ocasiones por asistentes. Además de permitirnos conectar con bases de datos SQL Server 2005 Express, también podemos hacerlo con otras ediciones de SQL Server 2005, SQL Server 2000, y MSDE 2000.

Como podemos ver en la figura 1, esta herramienta incluye un explorador de objetos, que recuerda en su aspecto al *Enterprise Manager*, aunque con una funcionalidad más reducida. Así, podemos hacer tareas típicas como gestionar las tablas, procedimientos almacenados, crear usuarios, entre otras. También incluye un editor de Transact-SQL para poder realizar consultas a la base de datos.

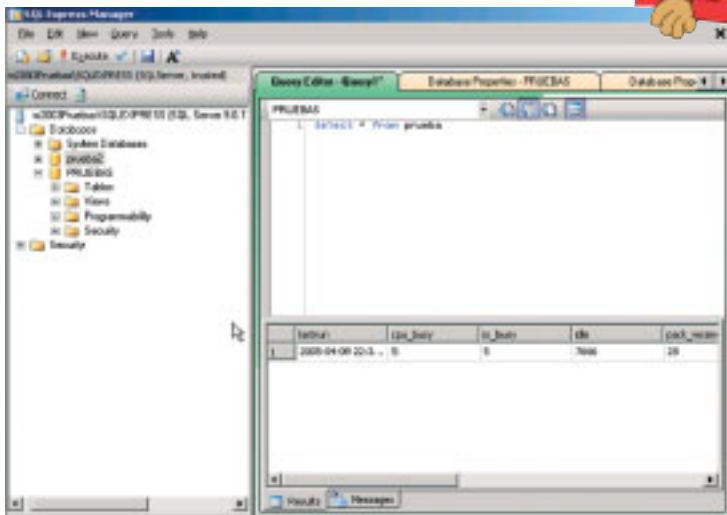


Figura 1. SQL Express Manager

## Integración con Visual Studio .NET 2005

La integración de SQL Server Express con Visual Studio .NET es casi total, se nota que se ha hecho un esfuerzo para aumentar la facilidad de uso y la sencillez en nuestros desarrollos.

Tan solo arrastrando y soltando una tabla de nuestra base de datos sobre un formulario podemos tenerla disponible en nuestra aplicación, sin escribir una sola línea de código o, por ejemplo, también se puede hacer del mismo modo la unión de un control **list-box** a un campo de nuestra base de datos. Todo de una forma visual e intuitiva, moviendo los campos de la base de datos con nuestro ratón sobre el control que nos interesa.



Una nueva característica es que ahora podemos utilizar objetos como fuente de datos de un control. Por ejemplo, un control *datagrid* puede enlazarse con un objeto de nuestra aplicación que representara a algún elemento de nuestra base de datos, por ejemplo, una tabla, vista o procedimiento almacenado. De esta forma en los desarrollos en *n-capas* podemos establecer una clara separación entre nuestro interfaz de usuario, la lógica de negocio y la capa de datos.

## Conclusiones

Podemos considerar, por tanto, a Microsoft SQL Server 2005 Express como una base de datos ligera, portable y de sencillo manejo, que podrá ser utilizada por una amplia variedad de usuarios debido a su carácter gratuito. Sin duda será de gran utilidad para desarrolladores, estudiantes o aplicaciones que no exigen una alta disponibilidad o un alto rendimiento y quizá un primer paso para abordar proyectos más grandes debido a su facilidad para migrar a ediciones superiores de SQL Server 2005. ○

Te contamos todo lo que sabemos de **.net**  
Lo que no sabemos, en el próximo número.

# dotNetManía

Revista especializada en la plataforma .net



Inscríbete en [www.dotnetmania.com](http://www.dotnetmania.com)

# GUSE.NET

## Grupo de Usuarios del Sureste de España

Muchas veces la inquietud es la fuerza mayor que hay en la naturaleza, así nace GUSE, el Grupo de Usuarios del Sureste de España, por la inquietud que algunos sentimos hacia la tecnología en la que trabajamos todos los días.

» Ya hace mucho tiempo que después de participar activamente en comunidades virtuales, *Callsql.com*, *helpdna.net*, *siquelnet.com*, *portalsql.com* y en los grupos de noticias públicos, pensamos que sería buena idea ponerle cara a esa gente, hacer más humano si se puede este mundillo. Así pues, podría decirse que **GUSE** nació hace muchos años en Torrevieja una calurosísima noche en que nos juntamos a cenar al final 3 familias que no se conocían más que por Internet. Después de esto y de fortalecer esa amistad creímos que sería buena idea, unir a más gente en reuniones con carácter técnico y hablar de nuestras cosillas :-), quedar a comer alguna vez, en fin esas cosas que hacen más humana esta profesión.

De entonces a ahora hay un largo camino que, apoyados por Microsoft, hemos recorrido y ahora, casi 50 profesionales estamos ilusionados con este proyecto que ya tiene nombre y forma y que esperamos nos brinde muchísimos éxitos profesionales.

Os cuento la historia más reciente, si puede ponerse un título sería “y de cómo dio sus primeros pasos GUSENET”:

El 27 de mayo en la Facultad de Informática de Murcia y gracias a la colaboración de la propia facultad y de Microsoft, tuvimos la oportunidad de reunirnos por segunda vez a tratar temas relacionados directamente con .NET y SQLServer.

En la primera reunión<sup>1</sup>, allá por julio del año pasado en Torrevieja, tuvimos la oportunidad de



disfrutar de dos charlas magistrales a cargo de **Ron Talmage** y **Mike Hotek**, en esa reunión nos hicieron de anfitriones **Fernando G. Guerrero** uno de los dos Regional Director de Microsoft y **Eladio Rincón MVP** en SQL. Tras aquella primera experiencia, muy positiva, nos propusimos constituir firmemente el grupo de usuarios, aún sin nombre. Sin embargo las circunstancias hicieron que no nos constituyéramos realmente hasta mayo de este año.

La segunda reunión, este mismo Mayo, fue espectacular, los tres ponentes que estuvieron con nosotros, **David Carmona**, **Antonio Soto** y **Fernando G. Guerrero** dieron realmente el espíritu y el conocimiento que esperamos todos obtener de estas reuniones.

David Carmona, contó las novedades de *Visual Studio Team System* –si habéis tenido la oportunidad de disfrutar de alguna charla de él sabréis el magnífico nivel que da a sus presentaciones y lo amenas que resultan–. En un evento como este, con el número ideal en mi opinión de asistentes (algo menos de 50) la cercanía que imprimió a su charla hizo que no sólo técnicamente fuese impresionante, sino que también fuese algo así como una reunión entre amigos.

**Antonio Soto** vino desde La Coruña a contarnos las novedades sobre seguridad que trae SQL Server 2005, es un tema realmente denso, certificados, claves, encriptación

<sup>1</sup> Nota del editor: La primera reunión se llamó “1 Reunión PASS Spanish Group” y dimos cuenta de ella en el número 7, el de septiembre de 2004 en dotNetManía. Fue un 9 de julio de 2004.



tación de campos, sin embargo, Antonio supo darle este tono amable que hace que las cosas complejas parezcan menos complejas. Nos contó cuales eran las formas para encriptar /desencriptar y como podía afectar esta nueva funcionalidad a los servicios SQL.

**Fernando G. Guerrero** hizo una especie de preguntas al experto mezcladas con ese saber estar que da la experiencia internacional del “Maestro Guerrero”, creo que simplemente nos “enamoro” tecnológicamente hablando. Fue capaz de transmitir de forma muy sencilla lo que son buenas prácticas y lo que no lo son, ahondando en aquellos temas en que la audiencia parecía más interesada.

En este punto establecimos un diálogo ameno y fluido entre los integrantes del grupo de usuarios, debatimos asuntos técnicos, por supuesto, pero también asuntos relacionados con la logística del grupo. Entre otras cosas, por ejemplo, quedamos que todas las sesiones tendrían una parte de ofertas/demandas de empleo, en las que las empresas interesadas (estén o no involucradas en el grupo) puedan encontrar profesionales motivados y bien formados en tecnología.

Al final lo que vamos a hacer son reuniones mensuales, sea invierno o verano, en ellas trataremos las últimas tendencias de la tecnología (menudo fin de año nos espera :-), y todos aquellos temas que puedan resultar de interés para el trabajo del día a día.

En cuanto al formato, decidimos que sería mixto, es decir, un par de horas serán de presentaciones formales, con Powerpoint, demos y todas esas cosas, y la segunda parte será de preguntas y respuestas, mezcladas con casos prácticos.

La idea es que cada uno de los integrantes del grupo pueda sentirse libre de consultar lo que no sabe, o exponer cómo está haciendo las cosas, simplemente para validar que no está inventando la rueda de nuevo y escu-

---

...vamos a seguir pisando fuerte, es muy posible que en julio contemos con una sorpresa muy agradable, en este momento estamos cerrando poder contar con el gran Itzik Ben-Gan, escritor habitual de SQLServer-Magazine

---

un *overview* de servicios que tenemos con SQL –será algo rápido pero muy práctico– y veremos también aspectos de migración de ASP a ASP.NET, después las secciones que os hablaba, ofertas/demandas de empleo, y mesa redonda con casos prácticos. Algunos de estos casos ya los estamos discutiendo *off* para garantizar que sea muy ágil y amena la sesión.

Y después de esto ¿qué? Bueno pues la verdad es que parece que vamos a seguir pisando fuerte, es muy posible que en julio contemos con una sorpresa muy agradable, en este momento estamos cerrando poder contar con el gran **Itzik Ben-Gan**, escritor habitual de **SQLServer-Magazine** y que ha aceptado nuestra invitación a venir. Es muy probable que esto cuaje, y que cerremos unas sesiones muy interesantes.

Así que ya sabes, si estás a un tiempo razonable en tu desplazamiento y te apetece tratar la tecnología desde un punto de vista fresco y dinámico, puedes acudir sin ningún problema a nuestras reuniones, que serán alternativamente en Alicante y Murcia y a la que quedáis todos, formalmente invitados.

Me gustaría terminar estas líneas agradeciendo a Microsoft la gran acogida que han dado a esta iniciativa,



Eladio Rincón



Fernando Guerrero



Salvador Ramos



Miguel Egea



Antonio Soto

Algunas de las personas que participan o han participado en las reuniones de GUSE.NET

char puntos de vista de otras personas más alejadas de el día a día este que nos somete a tanta presión.

Para la tercera reunión (que hemos bautizado como **Episodio III**), vamos a contar con **Juan Salvador Díaz** y con **Salvador Ramos** y con **Sergio Vázquez**, que nos van a hablar de SSIS, es decir, de los nuevos retos a los que nos enfrentaremos con los servicios de integración de Microsoft SQL 2005 (antes DTS), también haremos

a los miembros del grupo que están haciendo posible que la ilusión de unos cuantos “locos” de la informática se concrete y a **dotNetManía** que nos presta su espacio para que os contemos nuestras ilusiones.

Nos vemos en el sureste en la próxima reunión del **GUSE.NET**. ☺

.....  
Miguel Egea



Dino Esposito

# SQL Server 2005

>> **Incluso desde la sola** perspectiva del desarrollador, SQL Server 2005 introduce mejoras significativas en varias áreas, incluyendo soporte de tipos de datos, consultas dependientes y modificación, y múltiples resultados activos (MARS). El proveedor de datos ADO.NET para SQL Server también tiene nuevas posibilidades vinculadas a las mejoras de SQL Server 2005. Algunas de estas mejoras han sido algo sobrevaloradas y corren el riesgo de ser mal utilizadas.

**¿No debería dar el paso y comenzar a explorar MARS? Soy un tanto tradicional y no me fio demasiado cuando se comienza a dar mucho bombo a una tecnología dada. ¿Salvará MARS a la Tierra?**

Te comprendo. Yo también soy un tanto escéptico cuando veo eso en cualquiera de sus formas. En mi humilde opinión, *Multiple Active Result Set* (MARS), como muchas otras iniciativas de cara al desarrollo en SQL Server 2005, no salvará nuestro amado mundo; ni siquiera lo conquistará. Nos ayudará a manejar mejor algunos problemas de programación. Así que ajústese el cinturón, ponga el motor en marcha y vamos a explorar.

Cualquier versión de SQL Server anterior a la 2005, soporta solamente un conjunto de resultados abierto por conexión. Otras bases de datos han soportado esta característica desde hace mucho –sin importar la implementación– y ésta es una de las características que los programadores han solicitado desde mucho tiempo atrás al equipo de desarrollo de SQL Server. En ADO.NET 1.x, tanto el proveedor nativo de SQL Server como el proveedor OLE DB para SQL Server soportaban únicamente un conjunto de resultados por conexión. Curiosamente, el proveedor OLE DB no administrado y la vieja versión de la librería de ADO, sí parecen soportarlo. ¿Qué sucede? Que es sólo una ilusión. En OLE DB el efecto de múltiples resultados activos se consigue abriendo nuevas conexiones no manejadas por el *pool* asociado.

En SQL Server 2005, la característica MARS está implementada de forma nativa y permite a una aplicación tener más de un *SqlDataReader* abierto en la misma conexión, cada uno lanzado desde un comando distinto. Tener más de un *SqlDataReader* abierto por conexión ofrece potencialmente mejoras en el rendimiento, ya que múltiples *Readers* son menos pesados que múlti-

ples conexiones. Al mismo tiempo, MARS añade ciertas penalizaciones ocultas por operación, debido a su implementación interna. Considerar los inconvenientes y realizar una decisión sensata, es, como puede imaginarse, labor personal suya.

Para habilitar MARS en una conexión SQL Server 2005 (o incluso en una SQL Server Express 2005), no hay que hacer nada. Está activa por defecto. Para deshabilitarla, se necesita configurar el atributo **MultipleActiveResultSets** a **false**, en la cadena de conexión:

```
string conn = "MultipleActiveResultSets=false; ... ";
```

El uso canónico de MARS (su escenario de utilización más apropiado), es cuando se utiliza un *DataReader* para navegar por un conjunto de resultados, mientras otro comando sobre la misma conexión lanza sentencias de actualización sobre la base de datos, a medida que vamos leyendo. En el mundo anterior a .NET, probablemente haya usado un cursor de servidor para esas labores. Pero en el deslumbrante mundo de ADO.NET los cursores de servidor se han eliminado. Si necesita leer datos al tiempo que actualiza, está en dificultades. Consideremos un simple ejemplo de código para leer los empleados e invertir los nombres, pongamos de *Nancy* a *ycnaN* (Ver fuente 1).

**¿No debería dar el paso y comenzar a explorar MARS? ¿Salvará MARS a la Tierra?**

Lo primero, obtenemos un *DataReader* y, a continuación, recorremos todos los registros. Leemos el contenido de la columna **FirstName**, la guardamos en una cadena e invertimos el orden de sus caracteres. Después leemos el valor de la clave primaria de ese registro, y lanzamos un comando **UPDATE** para actualizar los resultados modificados. Se lanza el comando **UPDATE** sobre la misma conexión que se está utilizando con el *Reader*. Al final

```
using (SqlConnection conn = new SqlConnection(connString))
{
    SqlCommand cmd1 = new SqlCommand("SELECT * FROM employees", conn);
    cmd1.Connection.Open();
    SqlDataReader reader = cmd1.ExecuteReader();

    // Navegar por el data reader
    while (reader.Read())
    {
        // Invertir el nombre
        string firstNameReverted = reader["firstname"].ToString();
        char[] buf = firstNameReverted.ToCharArray();
        Array.Reverse(buf);
        firstNameReverted = new string(buf);

        // Ver el nombre invertido en la misma conexión
        int id = (int)reader["employeeid"];
        SqlCommand cmd2 = new SqlCommand(
            "UPDATE employees SET firstname=@newFirstName WHERE
            employeeid=@empID", conn);
        cmd2.Parameters.AddWithValue("@newFirstName", firstNameReverted);
        cmd2.Parameters.AddWithValue("empID", id);
        cmd2.ExecuteNonQuery();
    }
    reader.Close();

    // Obtener un nuevo Reader para refrescar la Interfaz de Usuario
    grid.DataSource = cmd1.ExecuteReader();
    grid.DataBind();
    cmd1.Connection.Close();
}
```

Fuente 1

del bucle, cerramos el *Reader*. Para refrescar alguna interfaz de usuario, digamos un *DataGrid*, obtenemos otro *Reader* a partir del mismo comando de consulta y mostramos los nombres actualizados. Si se ejecuta sobre un SQL Server 2005, funciona como un campeón.

Si intentas hacer esto en ADO.NET 1.x o en ADO.NET 2.0 con MARS deshabilitado, obtienes una excepción, acusando que el *DataReader* asociado con la conexión está abierto, y debiera cerrarse antes de intentar ejecutar un comando *UPDATE*. En otras palabras, leer y escribir al mismo tiempo, por la misma conexión, es una característica que sólo MARS puede habilitar. Como comentario paralelo, considera que, para que funcione MARS, es preciso utilizar diferentes objetos *SqlCommand*, para las diferentes sentencias que se quieran ejecutar. En el fuente 1, utilizo el mismo objeto *Command* (la variable *cmd1*), para leer la lista de los empleados 2 veces. La primera para navegar, la segunda para recuperar los nombres modificados. Para que esto funcione tengo que usar distintos *Command* o cerrar previamente el *Reader*.

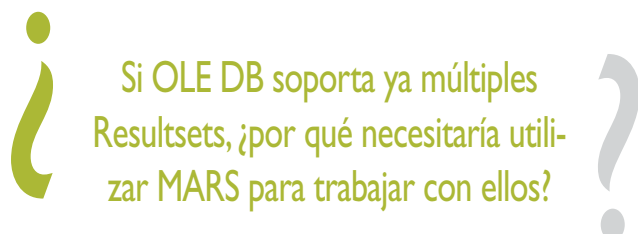
Otra gran ventaja de MARS es que, si estamos procesando una transacción, te

permite ejecutar código en el mismo ámbito del nivel de aislamiento que la conexión original. No se obtiene lo mismo abriendo una segunda conexión por debajo.

Existen algunos “costos” asociados con MARS: Revisemos el proceso de ejecución de un comando. Cuando se crea una conexión, ADO.NET crea una conexión física y una lógica. La conexión lógica va a parar a una especie de *pool* para su reutilización posterior. Cuando intentas ejecutar un método de un objeto *SqlCommand*, ADO.NET busca una conexión lógica disponible. Si se encuentra una, se vincula al objeto *SqlCommand* de forma que pueda ejecutarse. Si no, se crea y devuelve una nueva conexión lógica. No hace falta decir que ese *SqlCommand* requiere de una conexión lógica y que esta conexión se asigna un instante antes de la ejecución. La distinción entre conexiones físicas y lógicas es parte del funcionamiento interno de SQL Server en lo que concierne a la ejecución del comando. Cuando se completa el comando, la conexión lógica es liberada nuevamente en el *pool* de conexiones.

Las conexiones lógicas son recursos “caros” y valiosos, y se agrupan en una zona de memoria de 10 elementos. En tiempo de ejecución los comandos son agrupados en paquetes y ejecutados en transacciones internas contra tablas de la base de datos. Hay que recordar que MARS está definido en términos de agrupamiento secuencial, no en términos de ejecución paralela. La ejecución actual, sin embargo, sólo puede ser lanzada en puntos bien definidos. La mayoría de las sentencias, de hecho, deben ejecutarse de forma atómica en un proceso *batch*. Las excepciones son *SELECT*, *BULK INSERT* y *FETCH*. Para el resto de comandos, la ejecución debe completarse antes de que otro comando pueda ser lanzado sobre la misma conexión. El manejo excesivo de comandos, puede originar una utilización sobrecargada de conexiones. Incluso más sobrecargada que utilizar distintas conexiones o cursores de servidor.

Este comportamiento de MARS está disponible en la versión 2.0 de .NET Framework para los proveedores administrados OLE DB y Oracle. El proveedor de Oracle, no soporta el atributo *MultipleActiveResultset* en la cadena de conexión, pero habilita esa característica automáticamente. El proveedor OLE DB



Si OLE DB soporta ya múltiples Resultsets, ¿por qué necesitaría utilizar MARS para trabajar con ellos?

no soporta tampoco ese atributo en la cadena de conexión, pero simula múltiples resultados cuando se conecta a versiones anteriores de SQL Server o cuando la librería MDAC 9.0 no está presente. Si operamos a través de OLE DB en una versión SQL Server 2005 equipada con MDAC 9.0, la característica está activa de forma nativa.

Si OLE DB soporta ya múltiples Resultsets, ¿por qué necesitaría utilizar MARS para trabajar con ellos? O, dicho de otra forma, ¿por qué Microsoft no lo implementó como en OLE DB?

La lectura entre líneas de esta pregunta, sería algo así como ¿En qué forma difieren los múltiples *Resultsets* de



OLE DB y MARS en SQL Server 2005? Imagine el mismo ejemplo canónico que consideramos antes, actualizar mientras leemos. Si escribe ese código para usar OLE DB funciona normalmente. Sin embargo, lo que sucede internamente, merece algo más de atención. El punto clave es que, en OLE DB, el comando para actualizar el registro es enviado a través de una segunda conexión, invisible y no perteneciente al *pool* de conexiones. MARS en SQL Server 2005 funciona diferente y utiliza la misma conexión. La pregunta real sería ¿cuál es mejor? ¿cuáles fueron los inconvenientes que llevaron al equipo de desarrollo de SQL Server a implementar MARS de forma distinta? (Lo que ha supuesto un considerable trabajo adicional).

Cuando usamos OLE DB, SQL Server no trata las dos conexiones usadas de forma especial, son sólo dos conexiones más. Funcionan en entornos de ejecución diferentes y pueden entrar en mutuo conflicto. Intente envolver el código anterior dentro de una transacción definida de forma explícita: seguirá funcionando en MARS, pero no en OLE DB. De hecho, el proveedor OLE DB impide la creación de una nueva conexión desde una transacción local, o distribuida.

Incluso puede ser peor. Si la transacción comienza con uno de los comandos, el proveedor no lo detecta, y le permite lanzar una segunda conexión. En este punto, los comandos que aparentemente se encuentran en la misma transacción, acaban ejecutándose en transacciones distintas. En situaciones extremas, esto podría desembocar incluso en el temido “abrazo mortal” (*deadlock*), sobre todo si el nivel de la transacción es alto (pongamos **REPEATABLE READ** o **SERIALIZABLE**) y los tiempos de ejecución de los distintos comandos provocan su entrelazado.

**¿Habría que usar MARS en cursores de servidor? Bueno, creo que sé la respuesta: no hay cursores de servidor en ADO.NET, ¿no?**

Tiene razón. No hay cursores de servidor en ADO.NET, aunque su introducción fue repetidamente anunciada e incluso existió una implementación de ella, en una fase muy temprana de ADO.NET 2.0. Sin duda, los cursores de servidor son la forma más común de obtener la misma

## ¿Habría que usar MARS en cursores de servidor?

funcionalidad que permite MARS hoy en SQL Server 2005. Sin embargo, aunque ambos tienen ciertos puntos en común, no se trata de lo mismo. Como si fueran manzanas y naranjas, quizá tengan formas similares, tengan el mismo color, se produzcan y sean degustadas por la misma gente, pero son cosas bien distintas, en realidad.

Los cursores de servidor suministran una forma de procesar los resultados de una consulta, trabajando con una fila cada vez. Un cursor se declara por encima de una consulta, y a través de una operación *fetch* se obtienen las filas individuales a partir del conjunto de resultados.

Una vez que las filas han sido procesadas, o ya no se necesitan los resultados, el cursor es destruido, lo que libera los recursos de servidor asociados con él. Durante las operaciones *fetch*, no hay código ejecutándose en el servidor; sólo se preserva el estado. La principal limitación de los cursores de servidor es el rendimiento. Los cursores requieren un viaje de ida y vuelta al servidor por cada operación *fetch*, mientras que los resultados de MARS son accesibles a medida que se encuentran disponibles.

Al final, la existencia de MARS no elimina la necesidad de los cursores. Hay escenarios que sólo pueden ser procesados mediante cursores. Debido a esto, no todas las aplicaciones que utilizan cursores de servidor pueden ser cambiadas para usar MARS. Y, lo que es más importante acerca de MARS: no utiliza ejecución paralela de sentencias.

**He leído que SQL Server 2005 soporta XML de forma nativa. ¿Eso no existía ya? ¿Cuál es la diferencia con FOR XML?**

El que SQL Server 2005 soporte de forma nativa el tipo de datos XML, significa que ahora podemos almacenar datos XML en columnas individuales. Estoy de acuerdo en que, a primera vista, esto no parece nada nuevo en el sentido de que XML es texto plano y para

almacenar datos XML en una columna lo único que se necesita es que el tipo de columna acepte entradas de texto. Lo que te permite hacer SQL Server 2005, sin embargo, es algo distinto. Ahora podrías crear una columna del tipo XML. XML se convierte así en un tipo nativo del mismo nivel que **int** o **varchar**. Es reconocido por la nueva sintaxis de T-SQL y soportado por las herramientas de SQL Server, incluyendo el administrador de conexiones del Explorador de Servidores de Visual Studio 2005.

## He leído que SQL Server 2005 soporta XML de forma nativa. ¿Eso no existía ya?

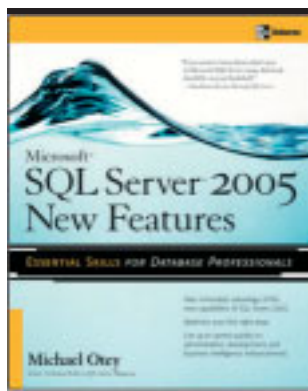
**SqlXml** es el nombre oficial del nuevo tipo en ADO.NET. El tipo pertenece al espacio de nombres **SqlTypes** como cualquier otro tipo de dato SQL. **GetSqlXml()** es el nuevo método de la clase **SqlDataReader** que devuelve el valor de la columna en formato fuertemente tipado.

```
SqlDataReader reader = cmd.ExecuteReader();
while (reader.Read())
{
    // El campo #1 contiene datos XML

    // Leer datos y hacer algún proceso
    SqlXml xml = reader.GetSqlXml(1);
    ProcessData(xml);
}
```

Este fragmento de código muestra cómo ejecutar una consulta y recuperar el valor de un campo particular declarado como XML. La clase **SqlXml** puede exponer los contenidos en cualquiera de las dos formas: como texto plano (propiedad **Value**) o a través de un **XmlReader** (con el método **CreateReader**).

En este punto, debiera estar clara la diferencia con **FOR XML**. La cláusula **FOR XML** fuerza a que toda la salida de una consulta se convierta en XML; el soporte nativo XML de SQL Server 2005 te permite trabajar con XML a nivel de columna. ○



## Microsoft SQL Server 2005 New Features

*Michael Otey*

Editorial: McGraw-Hill Osborne Media

ISBN: 0072227761

Páginas: 352

Publicado: Junio, 2004

Prologada por uno de los propios *Jefes de Proyecto* de SQL-Server 2005, **Otey** (ya con más de 6 obras en su haber, y *Senior Technical Editor* de SQL Server Magazine) presenta una visión de las novedades de esta versión dividida en 3 apartados principales: *Administración de bases de datos*, *Desarrollo* y *Business Intelligence*.

De especial interés resultan los apartados de integración con el CLR, mejoras del lenguaje T-SQL, Integración de XML, mejoras de seguridad, y los nuevos modelos de desarrollo: AMO, SMO, ODL, XMLA. Es de apreciar la concisión obtenida, en poco más de 350 páginas.

## A First Look at Microsoft SQL Server 2005 for Developers

*Bob Beauchemin, Dan Sullivan, Niels Berglund*

Editorial: Addison-Wesley

ISBN: 0321180593

Páginas: 693

Publicado: Junio, 2004

Conocíamos a **Beauchemin** por su anterior “Essential ADO.NET”, donde ya había probado su mérito divulgador, habiendo estado entre las mejor consideradas respecto a su temática. Vuelve ahora con esta propuesta temprana, pero bien acogida, para revisar los aspectos de interés para el desarrollador de Bases de Datos, teniendo el mérito de centrarse en aquellas cosas que -siendo necesarias- considera que pueden resultar más complejas al desarrollador “clásico” de BB. DD.: la integración con CLR y su acceso, el soporte XML en T-SQL y como tipo de datos nativo, SQL-Server 2005 como plataforma de servicios Web, las mejoras de ADO.NET y el uso de la plataforma como servidor de aplicaciones.



Marino Posadas

**Las presentaciones de la conferencia PASS, disponibles en inglés en el sitio de Gert Drappers**

El pasado mes de mayo, la conferencia PASS (una de las más importantes en su género en Europa), se celebró con una gran participación. Muchas de las presentaciones están disponibles gratuitamente en el sitio Web de **Gert Drappers** (ver "Sitios recomendados"). Las conferencias (así como otro material similar) se encuentran en [http://www.sqldev.net/events.htm#2005\\_PASS\\_European\\_Conference\\_May\\_11-13\\_2005,\\_Munich\\_\(Germany\)](http://www.sqldev.net/events.htm#2005_PASS_European_Conference_May_11-13_2005,_Munich_(Germany)).

**Cada neurona es una célula pensante que funciona como un sofisticado ordenador**

Al contrario de lo que se pensaba, (que el cerebro utilizaba amplísimas zonas para el almacenamiento de

recuerdos, ahora parece que el funcionamiento neuronal tiene una "personalidad individual", tal y como se deduce de las experiencias realizadas por el Instituto Tecnológico de California, y publicadas en la prestigiosa revista Nature, recientemente. Algún investigador ya sugiere a la vista de esto, las posibles implicaciones del descubrimiento en los sistemas de Inteligencia Artificial y Redes Neuronales. Para detalles sobre la noticia, ver la siguiente dirección : [http://www.tendencias21.net/index.php?action=article&id\\_article=165499](http://www.tendencias21.net/index.php?action=article&id_article=165499).



**Documentos y código en la Red**

"**The enterprise applications Design Pattern**" y "**The SqlDataAccessor Class**", son dos capítulos descargables de la obra "Design Patterns for Scalable Microsoft .NET Applications", de **Sten y Per Sundblad** disponibles en formato digital en <http://www.2xsundblad.com/ProductDetails.aspx?ProductID=1>.



**SQL Server 2005 White Papers: el sitio "oficial" de Microsoft** (<http://www.microsoft.com/sql/2005/techinfo/default.asp>) es un excelente lugar para comenzar a recopilar información sobre las nuevas características del producto.



"**Choosing Between the CLR and T-SQL in SQL Server 2005**", es un documento aclaratorio sobre las posibilidades y beneficios que nos pueden ofrecer las nuevas capacidades de integración con el CLR en SQL-Server 2005 frente al tradicional T-SQL. Por **Andy Holloway**, disponible en [http://www.devx.com/dbzone/Article/28412?trk=DXRSS\\_LATEST](http://www.devx.com/dbzone/Article/28412?trk=DXRSS_LATEST).

**Sitios recomendados**



**SQL Skills:** Se trata del sitio oficial de una de las divulgadoras de la herramienta (no me resigno a llamarle "evangelista") más populares en el mundo, **Kimberley Tripp**, ponente habitual en Tech-Ed y PDC, y autora de numerosos artículos y obras sobre el tema. El sitio está plagado de información: artículos, recursos, información de eventos, scripts de SLQ, etc. <http://www.sqlskills.com>.



**SQL Team** Un sitio pseudo-oficial, con cantidad de artículos, recursos, trucos, diseño de bases de datos, consultas, procedimientos almacenados, administración, OLAP, seguridad y muchísimo más. Sus webmasters: **Bill Graziano** (graz) y **Sean Baird**. <http://www.sql-team.com>.

**SQLDev.net:** El sitio de **Gert Drappers**, otro de los gurús que recomendamos en este número. En este caso, mucho contenido y poco continente: todo artículos y código fuente. <http://www.SQLDev.net>.



**El sitio oficial de Bill Vaughn**, otro de los clásicos de las bases de datos. Aparte de la información sobre sus actividades, mantiene una base de datos de artículos publicados por él en las revistas internacionales más populares, como *SQL-Server Magazine*, *MSDN Magazine*, *Pinnacle*, *Visual Studio Magazine*, *VS7 Magazine*, y un buen conjunto de información adicional. <http://beta.v.com/BLOG/billva>.



**NOTA** Además de los citados, no se pierda los sitios de **Fernando Guerrero** ([www.solidqualitylearning.com](http://www.solidqualitylearning.com)), **Eladio Rincón** (<http://www.siquelnet.com>), **Miguel Egea** (<http://www.portalSQL.com>) y **Salvador Ramos** (<http://www.helpdna.net>), todos ellos autores de artículos en este ejemplar e integrantes de lo que ya se denomina universalmente "Escuela Mediterránea de SQL-Server" (y no "Escuela Murciana de SQL-Server", como hemos podido leer en algunos sitios extranjeros, sin duda desconocedores de nuestra geografía.).



**Consiga dos usuarios gratuitos de Advantage  
llamando al 902 160 145**



**Tenemos LA solución para usted:**

Advantage Database Server (ADS) le da mucho más tiempo y mucho más dinero para las cosas importantes de la vida.

Advantage Database Server es una alternativa real a las bases de datos SQL Cliente/Servidor, a una fracción del coste. ADS tiene ya más de 2 millones de usuarios!

## **Advantage Database Server**

...más tiempo libre y más dinero para las cosas importantes...

**ABOX**

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

 **Extended  
Systems®**

# LOS PECES LUCHADORES DE SIAM SE MATAN ENTRE ELLOS AL INTRODUCIRLOS EN EL MISMO ENTORNO.

CON FRECUENCIA, PASA LO MISMO CON LAS APLICACIONES EMPRESARIALES.

## WINDOWS SERVER SYSTEM CON .NET REDUCE LAS DIFICULTADES DE INTEGRACIÓN ENTRE APLICACIONES.

Las soluciones CRM y de cadena de suministro pueden convivir perfectamente. ¿Cómo? Con Windows Server System™ y .NET.

.NET Framework, un componente esencial de Windows Server System, es el entorno de desarrollo que permite un perfecto funcionamiento conjunto de los diversos componentes y aplicaciones. Así las aplicaciones resultan más sencillas de crear, gestionar, implementar e integrar. .NET Framework utiliza los estándares del sector como

XML y Web Services los cuales permiten conectar las aplicaciones empresariales a infraestructuras de cualquier tipo.

Además, las características de mejora de la productividad de .NET, como la asignación automática de datos desde y hacia XML, permiten simplificar la integración reduciendo la cantidad de código necesario.

Para obtener más información sobre Windows Server System y .NET, visite [www.microsoft.com/spain/consys](http://www.microsoft.com/spain/consys)



WINDOWS SERVER SYSTEM Y .NET LE PERMITEN SOLUCIONAR LOS PROBLEMAS DE INTEGRACIÓN AUNQUE NO PUEDE HACER NADA CON RESPECTO AL PEZ LUCHADOR DE SIAM.

Microsoft  
**Windows Server System™**