

# Contadores para Rendimiento de SQL Server

Por [Alberto López Grande](#)

## Contenido

[Contadores para Rendimiento de SQL Server](#)

[1. Introducción](#)

[2. ¿Qué son los Contadores?](#)

[3. ¿Cómo poner Contadores?](#)

[4. Áreas de Medición](#)

[5. Memoria](#)

[6. Procesador](#)

[7. Disco](#)

[8. Red](#)

[9. SQL Server](#)

[10. Conclusión](#)

## Contadores para Rendimiento de SQL Server

[Principio de la página](#)

### 1. Introducción

Una de las funciones indispensables de la administración es la monitorización del rendimiento de los servidores de bases de datos. Entre otras cosas, para eso están los **contadores de rendimiento**, que dan medida de numerosos parámetros. Existen muchísimos contadores que permiten la monitorización, tanto del sistema operativo como propios de SQL Server. Iniciarse en su conocimiento y utilización no siempre es sencillo y no sólo eso: si sabemos cuáles tenemos que poner y cómo, su interpretación tampoco es trivial. ¿Qué contadores son los más relevantes? ¿Qué tengo que medir? ¿Qué valores de referencia indican que la cosa va bien o va mal? Estas son algunas de las preguntas que este artículo pretende responder, dando una visión introductoria para aquél que se enfrenta por primera vez a este problema. Con el tiempo, en base a la profundización de estos temas, el conocimiento de las particularidades de cada entorno y el problema que se puede estar teniendo en ese momento, cada cual opta por unos u otros, con lo que se terminaría añadiendo y/o quitando contadores. Lo que aquí se ofrece es un conjunto contadores, intentando siempre observar la generalidad.

[Principio de la página](#)

### 2. ¿Qué son los Contadores?

A continuación un breve repaso sobre la forma en la que se colocan los contadores, cómo se instalan y algo acerca de su nomenclatura. Un contador, en general, es una medida de un parámetro determinado. Cada contador pertenece a un objeto. Por objeto, se entiende un ámbito de medición, existiendo muchos de estos objetos. Cada objeto tiene uno o más contadores y es necesario que se instale o active el objeto para que podamos recoger los datos de sus contadores. Hay objetos que se instalan con el sistema operativo, como el objeto **System**; otros hay que instalarlos, como los de red; y otros los instalan las aplicaciones, como los de SQL Server (es una opción habilitada por defecto en el setup, siendo más que recomendable su inclusión). También hay contadores que sólo están disponibles bajo determinadas circunstancias, como que un servicio concreto esté arrancado o se esté realizando un backup. Por último, existen contadores que permiten la medición de diferentes instancias o partes concretas o del total de las mismas. Por ejemplo, si tenemos varias CPU, podemos medir la actividad de cada una de ellas de forma independiente o tomar una media. Otro ejemplo son los contadores que miden el número de transacciones, que permiten recoger el número de transacciones de cada base de datos independientemente.

El [Figura 1](#) muestra el formulario que permite añadir contadores en **Performance Monitor**, donde se pueden ver estos tres conceptos: **Objeto**, **Contador** e **Instancia**:



Figura 1. [Volver al texto.](#)

[Principio de la página](#)

### 3. ¿Cómo poner Contadores?

El sistema operativo cuenta con una herramienta denominada **Performance Monitor** (Ir a **Inicio | Ejecutar | "perfmon"**): una interfaz para acceder a los objetos y sus contadores. Esta herramienta permite ver los contadores "en directo". Sin embargo, la forma más saludable es arrancar los contadores, acumular datos y luego estudiarlos con calma. Lo normal es realizar una captura cada cierta cantidad de minutos (1, 2 ó 5) de muchos contadores. Esto nos permitirá realizar estudios de tendencias, identificar picos y valles de carga durante el día y, sobre todo, no tener que poner contadores cuando se reporte un problema de rendimiento, momento en el que quizás ya sea demasiado tarde. Para ello existen herramientas en el mercado que te permitirán recolectar estos datos, tales como NetIQ, etc. Windows también tiene su propia herramienta de registro y alertas de rendimiento (así se llama este servicio), al cual se puede acceder desde la Administración de equipos. Se pueden configurar un conjunto de contadores, una frecuencia de los mismos, un formato de salida para los datos y luego empezar a recoger datos. El formato de los datos puede ser binario, texto plano e incluso pueden almacenarse en una base de datos, a través de un DNS. Esto nos abre un enorme abanico de posibilidades cuando pasemos a explotar la información recogida. Podemos contar también con Performance Monitor, pero también con Excel e incluso, si la cantidad de información es grande, podemos preparar un cubo OLAP para procesarlo todo.

En ambos casos, ya sea en directo o acumulando datos, no es bueno recoger estos datos desde el propio servidor que se desea monitorizar. Esto no es sólo por la alteración en los datos recogidos por la toma de los propios contadores, sino por el impacto en el rendimiento general que podemos causar. Si no se dispone de un equipo dedicado exclusivamente a la monitorización de la infraestructura, se puede recurrir a cualquier otro equipo que tengamos cerca. Usar el propio nodo en el que en ese momento está el servidor de bases de datos para estos fines debe ser siempre el último recurso.

[Principio de la página](#)

### 4. Áreas de Medición

Una de las Leyes de Murphy dice: "Todo lo que puede ir mal, irá mal". Y es cierto, por lo que debemos monitorizarlo todo de una forma u otra. Sea lo que fuere que vayamos a vigilar mediante el uso de contadores de rendimiento, aquí vamos a dividirlo en cuatro grandes áreas, a nivel de hardware (memoria, procesador, red y disco), con un quinto apartado para localizar problemas propios de los desarrollos y particularidades del sistema a estudiar, que serán sobre todo contadores de objetos propios de SQL Server (estos, en muchos casos, servirán de apoyo o ayuda para la monitorización de las áreas anteriores).

El objetivo es poder identificar cuellos de botella, es decir, dónde está principalmente el problema. Otro asunto será ver cómo salvamos ese cuello de botella. Si los contadores nos demuestran un problema de memoria, la conclusión inmediata no deberá ser añadir memoria al servidor. Si tenemos un proceso que se come los recursos, y solo aumentamos los recursos (sin mejorar ese proceso), es posible que tres meses después de realizada la inversión, nos encontremos con que nuevamente se nos ha presentado el problema en el servidor, quedando además en una posición difícil de explicar. Pedir más hardware es reconocer que no somos capaces de hacer mejor nuestro trabajo, que entre otras cosas es sacarle el máximo partido a los recursos con los que contamos. Así que cuando lo pidamos, que sea porque ya hemos revisado a conciencia todo lo demás y hemos comprobado que el coste de mejorar el desarrollo o ese diseño está sobradamente por encima del coste en hardware. Y cuando llegue ese momento, los datos recogidos de los diferentes contadores serán un punto de apoyo muy bueno para justificar la adecuación de la infraestructura.

Es necesario puntualizar, por último, que aunque lo queramos dividir en áreas, todas estas áreas están bastante relacionadas. Un problema de contención de disco suele repercutir en la CPU y puede estar causado por una mala gestión de la memoria y, finalmente, saturar la red. Es muy importante tener en cuenta estas interrelaciones a la hora de realizar un diagnóstico certero que nos permita localizar el origen (o los orígenes) del déficit de rendimiento.

#### [Principio de la página](#)

## 5. Memoria

El correcto uso de la memoria es lo que nos brindará un servidor más ágil en su respuesta, es decir, lo que va a significar que el usuario no diga la frase más temida: "La base de datos va muy lenta". SQL Server debe tener suficiente memoria disponible como para emplear un poco más si le hace falta (aunque por norma general ya la acapara casi en su totalidad), pero a la vez debe ocupar la suficiente memoria como para evitar que las operaciones se realicen contra el disco físico. Si la mejor consulta es aquella que no se realiza, la segunda mejor es aquella que se realiza contra la caché.

Son esos dos aspectos, que haya memoria disponible y que se explote la caché al máximo, los que monitorizaremos en esta área:

- **Memory: Pages/sec.** Indica el número de páginas que entran y salen de la caché en cada segundo. Su valor debe situarse muy cercano a 0. Si es mayor a 20 de forma continuada, tal vez no tengamos un problema de rendimiento; pero lo que es seguro es que la memoria no está siendo gestionada adecuadamente.
- **Memory: Availability Mb (ó Kb ó Bytes , lo que más cómodo resulte).** En general, debemos contar con 5 Mb de memoria libres (y disponibles) como mínimo.
- **SQL Server: Memory Manager: Total Server Memory y Target Server Memory.** Estos dos contadores del mismo objeto nos dicen el total de memoria que tenemos y la memoria que necesitamos. "Total" debe ser igual que "Target". Si esto no es así, y "Total" es menor que "Target", es un indicio claro de un problema en la memoria, ya que tenemos menos de la que necesitamos.

#### [Principio de la página](#)

## 6. Procesador

El uso del procesador es un punto básico para la monitorización de cualquier servidor, sea o no de base de datos. Dentro de los que hay, nos concentraremos en dos de ellos: el porcentaje de uso y la cola del procesador, más un tercero que permita conocer *qué parte del pastel* se come el sistema operativo. Cuanto menor sea la utilización de las CPU que tengamos, mejor. Este uso comienza a ser un problema cuando de forma sostenida se sobrepasa el 80%. Y para solventarlos, antes de pensar en añadir más procesadores y/o cambiarlos por otros más veloces o tecnológicamente más avanzados, es preciso revisar las recompilaciones y los planes de ejecución de las consultas, entre otras cosas. Luego, para monitorizar la CPU podemos optar por los siguientes contadores:

- **Processor: % CPU Usage** (instancia \_Total si se cuenta con más de un procesador y si todos ellos están desempeñando el mismo rol): Mantener por debajo del 80%.
- **System: Processor queue length.** Es la cola de procesador, debe permanecer por debajo de 2 por CPU.

Si estos dos contadores están por encima de lo normal, podemos fijarnos en un par de ellos más para afinar el diagnóstico. Uno es **System: Context Switches/sec** (sólo en caso de tener más de una CPU). Este contador indica las veces en las que un mismo proceso, medianamente pesado, cambia de procesador para completarse. El cambio tiene lugar por cuestiones de balanceo de carga y aunque puede forzarse a que no se efectúe (con **option maxdop**, por ejemplo), lo más lógico es dejar que sea el servidor el que gestione estos saltos. Pero cada movimiento tiene un coste y ese coste puede verse reducido si configuramos el servidor para que use fibras (que se pueden definir como conjuntos de hilos o hilos gruesos), que suavizan de forma considerable el coste del cambio entre procesadores. Si este contador se sitúa por encima de 8000, es momento de pensar en modificar esta configuración.

El otro contador que puede aportarnos información adicional es **System: % Total Privileged Time**; éste indica qué porcentaje de tiempo se dedica a tareas del sistema operativo. Si está por encima del 20%, es posible que el problema no esté en el procesador, sino en el disco. Para ello, verificar si el contador **PhysicalDisk: % Disk Time** (del que hablaremos más adelante en el apartado de disco) está por encima del 55%.

[Principio de la página](#)

## 7. Disco

El disco es el punto en el que más frecuentemente se localizan los cuellos de botella, así como la causa final de los problemas de otras áreas. Discos potentes y rápidos (ya sea pinchados en el servidor o en una cabina de discos), una configuración en el RAID adecuada que asegure la disponibilidad, y una ubicación correcta de los ficheros en los diferentes grupos de discos (datos y log separados, con discos independientes también para tempdb) es la clave para que nuestros servidores rindan a plena potencia. Aún con todo esto, pueden presentarse problemas de contención, que con una adecuada monitorización detectaremos; es más, la monitorización nos dará pistas para paliar la situación. Así, además de saber si cada conjunto de discos va bien o mal, podremos determinar si el problema está en las lecturas o en las escrituras, si cambiar el fill factor de los índices ayudaría, etc.

Para poder observar los contadores de disco físico, es necesario que éstos estén activos. A partir del sistema operativo Windows 2000, los contadores están activados por defecto. Si esto no es así, es necesaria la ejecución de **diskperf -y** en una ventana de comandos del sistema, así como también el reinicio del sistema, para que los podamos colocar.

El contador más relevante quizá sea la cola de disco, ya sea la media o la actual. Si durante períodos prolongados (más de 10 minutos) se mantiene por encima de 2, se puede decir que existe un problema, aunque teniendo en cuenta dos detalles importantes: primero, el contador funciona a nivel lógico, es decir, por unidades. Si por ejemplo tenemos una unidad en RAID 5 compuesta por 5 discos, tendremos que dividir el valor del contador entre 5; y segundo, hay que ser un poco permisivos, por sentido común, con determinados procesos, sean o no de mantenimiento. La realización de backup y las reindexaciones son operaciones que exprimen intensamente el disco. Es totalmente normal que las colas de disco se disparen, sin que por ello tengamos que perder la cabeza. Asimismo, si tenemos un DTS que realiza la actualización del diario o la actualización de los datos para un cubo OLAP, por ejemplo, hay que ser conciente de ello y esperar una cola de disco realmente alta mientras dure el proceso. La forma de proceder será encontrar un momento de baja carga para la ejecución de estos procesos, como durante las noches o los fines de semana. Tampoco significa esto que no podamos optimizar todos estos procesos, sólo que no pueden ser medidos por el mismo rasero.

Los contadores para el disco:

- **PhysicalDisk: Avg. Disk Queue Length.** Debe estar por debajo de 2 en cada unidad (tras ponderar el número de discos del RAID, si procede). La instancia común puede sernos de ayuda para calibrar el estado general.
- **PhysicalDisk: % Disk Time.** Indica qué porcentaje de tiempo se emplea en el disco. Si está por encima del 55%, puede que haya un problema, habría que mirar también **PhysicalDisk: Disk Read Time** y **PhysicalDisk: Disk Write Time** para ver si hay un importante desequilibrio no esperado entre las lecturas y las escrituras, que podría regularse variando el fill factor de los índices (si estamos hablando de una base de datos de sólo lectura o eminentemente de lectura, es lógico que haya un desequilibrio). Si las lecturas están muy por encima de las escrituras, el fill factor puede que sea muy alto. Si es al contrario y las escrituras llevan la mayor parte del peso, aumentar el fill factor podría paliar el problema. Es más una cuestión de afinar y probar hasta encontrar un equilibrio.

[Principio de la página](#)

## 8. Red

Los objetos que nos permiten colocar contadores de red, los que empiezan por **Network**, se instalan como un componente de Windows (en estos sistemas operativos), concretamente en el llamado **Herramientas de administración y supervisión**.

La red casi nunca supone un cuello de botella, pero es imposible no tenerla en cuenta, ya que los datos van y vienen por aquí. Lo que no es tan infrecuente es que se produzcan cortes en la comunicación, que pueden ser detectados por métodos tan simples como poniendo una estadística de ping (con paquetes ligeros). La aplicación de las normas más básicas de la tecnología cliente-servidor nos aportan la mejor forma de optimizar el uso de la red. Es decir, enviar sólo aquello que el cliente pide e impedir que el cliente pida más de lo que necesita (con una correcta paginación, parámetros limitados en los reports, búsquedas controladas, etc.).

Aún así, nuestra red debe estar dimensionada acorde con lo que se le va a exigir, para lo cual contamos con la ayuda de los siguientes contadores:

- **Network Interface: Bytes Total/sec.** Depende esencialmente de la red, con lo que es difícil dar una cifra que pueda ser usada de forma general. Se puede aplicar una sencilla regla, en combinación con el contador **Current Bandwidth**, del mismo objeto. **Bytes Total/sec** dividido entre **Current Bandwidth** debe ser menor que 6.
- **Network Segment: % Network Utilization.** Permite verificar el uso de cada tarjeta de red que podamos tener en el servidor.
- **Server: Bytes Received/sec y Server: Bytes Transmitted/sec.** Permite comprobar si es el servidor de base de datos el que está saturando la red, perjudicando sus otros usos.
- **SQLServer: SQL Statistics: Batch Request/sec.** Una tarjeta de red de 100Mbps soporta, aproximadamente, unos 3000 comandos por segundo. Si este contador está por encima, se precisa una segunda tarjeta o una de mayor capacidad.

[Principio de la página](#)

## 9. SQL Server

Casi todos los contadores mencionados hasta ahora son del sistema operativo. SQL Server posee un importante número de objetos, con sus contadores, que permiten monitorizar con el detalle que uno desee un servidor, y más concretamente los desarrollos o peculiaridades que corren en él. Tenemos objetos para SQL Server Agent, replicación, cursores, bloqueos, memoria y caché, conexiones, paginación, backup, etc. No hay más que ir al monitor de rendimiento y en la lista de objetos ver todos los que empiezan con **SQLServer**. Aquí se citarán algunos de ellos, los más generales quizás. Pero en función del uso que tengamos en cada servidor deberemos ampliar o cambiar los contadores a seguir. Y si en general lo de los contadores es una cuestión de gusto o comodidad, ya que se puede medir lo mismo con varios contadores en casi todos los casos, en el caso de los contadores propios de SQL Server, en los que tenemos más variedad, más aún si cabe.

Los siguientes son algunos de los contadores de uso más extendido, intentando hacer un barrido por lo más significativo para conseguir una monitorización básica:

- **SQLServer: Access Methods: Page split/sec.** Si está por encima de 100, viene acompañado de problemas de disco. Un aumento en el fill factor puede resolver la situación.
- **SQLServer: Buffer Manager: Cache Hit Ratio.** Indica el porcentaje de veces que el motor usa la caché frente al disco. Es un valor medio desde el último reinicio. Este valor debe permanecer por encima del 99%, casi en 100, para servidores OLTP. En servidores OLAP, debe estar por encima del 80%.

Los siguientes 5 contadores sirven para afinar problemas de caché y memoria:

- **SQLServer: Buffer Manager: Page Life Expectancy.** Tiempo en segundos que permanece una página en memoria sin tener ninguna referencia que la retenga allí. Cuanto más tiempo, mejor. Un valor de referencia, por encima de 300, es decir 5 minutos.

- **SQLServer: Buffer Manager: Lazy Write/sec.** Páginas que salen de la caché por segundo. Al contrario que el anterior, cuanto más alto, peor. Por debajo de 20.
- **SQLServer: Buffer Manager: Checkpoint Pages/sec.** Un checkpoint obliga a bajar a disco todas las páginas que se tengan en memoria. Sólo debe ejecutarse en determinadas circunstancias, la mayoría de ellas, tareas administrativas, por lo que si se ejecuta con mucha frecuencia, estaremos mal utilizando la memoria.
- **SQLServer: Buffer Manager: Procedure Cache Pages.** Este contador indica las páginas de memoria dedicadas a almacenar planes de ejecución de procedimientos almacenados. Un descenso brusco en este contador puede venir acompañado de un descenso del rendimiento, causado por la recompilación de procedimientos almacenados.
- **SQLServer: Databases: Log Flushes/sec.** Indica las veces por segundo que las páginas pasan de caché al fichero de log. Funciona muy en paralelo al número de transacciones, y como el número de transacciones, cuanto menor sea, mayor rendimiento.

Para tener una idea del número de usuarios que manejamos, los siguientes 3 contadores son muy útiles:

- **SQLServer: General Statistics: User connections.** Indica el número de conexiones. Sirve para saber identificar horas de alta y baja actividad y para saber si un pico en otras áreas puede estar relacionado con un mayor número de usuarios en ese momento.
- **SQLServer: Databases: Transaction/sec.** Además de un uso similar al contador anterior, nos permite determinar qué bases de datos tienen más carga de transacciones. Un caso a tratar de forma particular es el de tempdb, ya que es muy común que sea ésta una de las bases de datos que más transacciones por segundo soporta. Es necesario vigilar y optimizar en lo posible este hecho. Su reducción puede venir de muchas formas, siendo algunas tan obvias como la revisión de los procesos que usan tablas temporales, pero no sólo eso, también hay que observar las consultas muy pesadas y complejas, que usan tempdb para completarse.
- **SQLServer: SQL Statistics: SQL Compilations/sec.** Da una idea de la carga real del servidor, en cuanto a compilaciones se refiere. Si está entorno a 100, es buena señal. Si pasa de ahí, se estarán consumiendo muchos recursos en la preparación de planes de ejecución.

Por lo general, la detección de bloqueos no se observa de una forma fácil con contadores, salvo que la situación sea realmente caótica. Los siguientes contadores aportan datos que si destacan, es mejor estudiar con profiler:

- **SQLServer: Access Methods: Full scans/sec.** Este contador nos da una idea de las veces en las que se realizan recorridos de índice, mucho peor que realización de búsquedas en los mismos. Si arroja cifras relevantes, es mejor capturar con profiler y estudiar planes de ejecución de las consultas con más lecturas lógicas.
- **SQLServer: Locks: Number of Deadlocks.** El número de interbloqueos debe ser 0. Si se detecta alguno, hay que revisar y erradicar las sentencias que estén provocando ese problema. En ocasiones, el negocio obliga a que se den con más frecuencia, será cuando más cuidado haya que tener con el nivel de aislamiento.
- **SQLServer: Locks: Avg. Wait Time (ms).** Indica la media de milisegundos que hay que esperar para la liberación de un bloqueo.
- **SQLServer: Latches: Average Latch Wait Time (ms) , Latch Waits/sec y Total Latch Wait Time (ms).** Estos tres contadores nos hablan de los **minibloqueos**, es decir, bloqueos que son tan cortos que no llegan ni a serlo. Pero eso no significa que no estén ahí. Un elevado número de latches suele venir acompañado de un importante número de bloqueos. Así que estos contadores son de gran ayuda para anticiparse a problemas de este tipo, ya que lo que hoy son latches, en el futuro pueden ser bloqueos. La forma de gestionar estos valores es obtener una línea base durante un periodo de tiempo que consideremos normal, y que luego usemos como referencia, para poder saber si la situación se degrada. Si nos alejamos de esa línea base (por arriba), además de la revisión de las consultas y el nivel de aislamiento, es frecuente que el



problema esté en la memoria o en el disco. Otros contadores, ya mencionados, nos ayudarán en este sentido.

Por último, un contador para medir el rendimiento de los backup. Cuando se lanza un backup, lo más normal es que se produzca un importante pico en la cola de disco. Si contamos con varias bases de datos, es típico lanzarlos todos a la vez (además, justo a las doce de la noche). Podemos tener un problema de rendimiento que es preciso vigilar, que podríamos evitar lanzando los backup uno a continuación del anterior. Un contador al que podemos acudir es **SQLServer: Backup Device: Device Throughput Bytes/sec**. Este contador mide el rendimiento de los backup, y con una regla de tres simple también lo podemos usar para saber cuánto nos queda para que un backup finalice, por ejemplo. Tiene la pega de que sólo se puede arrancar para los backup que estén realizándose en ese momento. Pero si se dispone de una utilidad de monitorización es posible explotarlo sin problema.

Hay que tener un poco de cuidado, porque a veces puede arrojar resultados engañosos, apareciendo lecturas cuando hay una importante actividad, generalmente relacionada con tempdb (pero no exclusivamente con ello), y no hay ningún backup en curso. Los backup del log propios de tempdb, algo interno del funcionamiento de esta base de datos, pueden ser detectados en este contador cuando existe un fuerte uso de esta base de datos (tablas temporales creadas intencionadamente o como resultado de procesos muy pesados que requieren de tablas intermedias, o worktables, para su funcionamiento). Podemos distinguirlos de los backup *normales*, además de por saber que no estamos haciendo ningún backup en ese momento, porque el contador arroja picos muy breves e intensos, en lugar de valores intermedios durante periodos prolongados (lo que dure el backup).

[Principio de la página](#)

## 10. Conclusión

A modo de resumen, se muestra a continuación un cuadro con los contadores sobre los que se ha hablado, y sus valores de referencia:

Contador	Valor/Uso
Memoria Memory: Pages/sec	< 20
Memoria Memory: Availability Mb	> 5 Mb
Memoria SQLServer: Memory Manager: Total Server Memory vs. SQLServer: Memory Manager: Target Server Memory	Si Total = Target -> Bien
Si Total < Target	-> Falta RAM
Procesador Processor: % CPU Usage	< 80 %
Procesador System: Processor queue length	< 2 por CPU
Procesador System: Context Switches/sec	Si es > 8000, probar Fibras
Procesador System: % Total Privileged Time	Si es < 20 %, posible problema de disco
Disco PhysicalDisk: Avg. Disk Queue Length	Avg./(#Discos del RAID) < 2
Disco PhysicalDisk: % Disk Time	< 55%
Disco PhysicalDisk: Disk Read Time vs. PhysicalDisk: Disk Write Time	Si Read >> Write -> Bajar Fill Factor
Si Read << Write	-> Subir Fill Factor
Red Network Interface: Bytes Total/sec vs. Network Interface: Current Bandwidth Bytes / Bandwidth	< 6
Red Network Segment: % Network Utilization	Comprobación de uso de las distintas tarjetas de red
Red Server: Bytes Received/sec vs. Server: Bytes Transmitted/sec	Saber si el servidor está perjudicando el resto de la red
Red SQLServer: SQL Statistics: Batch Request/sec	3000 por tarjeta de 100 Mbs
SQL Server (General) SQLServer: Access Methods: Page split/sec	Si es < 100, posible problema de disco
SQL Server (General) SQLServer: Buffer Manager: Cache Hit Ratio	En OLTP > 99 %; en OLAP > 80 %, uso de caché
SQL Server (Memoria) SQLServer: Buffer Manager: Page Life Expectancy	> 300
SQL Server (Memoria) SQLServer: Buffer Manager: Lazy Write/sec	< 20
SQL Server (Memoria) SQLServer: Buffer Manager: Checkpoint Pages/sec	No debe aparecer con frecuencia
SQL Server (Memoria) SQLServer: Buffer Manager: Procedure Cache Pages	Debe permanecer constante
SQL Server (Memoria) SQLServer: Databases: Log Flushes/sec	A menor valor, mayor rendimiento
SQL Server (Usuarios) SQLServer: General Statistics: User connections	Identificar horas punta y horas valle de usuarios
SQL Server (Usuarios) SQLServer: Databases: Transaction/sec	Saber qué base de datos es más usada
SQL Server (Usuarios) SQLServer: SQL Statistics: SQL Compilations/sec	< 100 (si no, los planes de ejecución están consumiendo mucho)
SQL Server (Bloqueos) SQLServer: Access Methods: Full scans/sec	Si es relevante, estudiar con profiler
SQL Server (Bloqueos) SQLServer: Locks: Number of Deadlocks	Mínimo (estudiar con profiler)
SQL Server (Bloqueos) SQLServer: Locks: Avg. Wait Time (ms)	Mínimo posible

10/8/2009

Contadores para Rendimiento de SQL S...

SQL Server (Bloqueos) SQLServer: Latches: Average Latch Wait Time (ms), Latch Waits/sec y Total Latch Wait Time (ms) Establecer línea base y observar qué se está entorno a ella  
SQL Server (Backup) SQLServer: Backup Device: Device Throughput Bytes/sec Rendimiento de backup, fuerte actividad en tempdb

**Alberto López Grande** posee 6 años de experiencia trabajando con diversos ámbitos de SQL Server en Madrid. Cuenta con la certificación MCP de SQL Server 2000, tanto en administración como en diseño.

[Principio de la página](#)