

Servicios de componentes en Windows 2000 - COM+

© Francisco Charte Ojeda – <http://www.fcharte.com>

Sumario

El modelo de componentes COM llegó a su primera revisión importante, con la aparición de Windows 2000. Nuevos servicios, más simplicidad y un nuevo nombre: COM+.

Introducción

La llegada de Windows 2000 trajo consigo muchas novedades importantes para los programadores, siendo una de las más interesantes los nuevos servicios de componentes, a los que se ha denominado COM+. El nombre denota, de forma bastante clara, que se trata de una evolución de COM (*Component Object Model*), de una versión mejorada del modelo original y el más usado en la actualidad. Desde 1993, año en que apareció la versión original de COM, los cambios se han ido sucediendo de manera esporádica y sin afectar de modo importante a los servicios ofrecidos. Tan sólo la aparición de MTS (*Microsoft Transaction Server*) supuso una aportación cualitativa pero, siempre hay un pero, MTS no era parte de COM, sino una capa interpuesta entre el modelo de componentes y sus aplicaciones.

COM+ sigue siendo COM, por lo que los componentes que está usando actualmente, salvo excepciones, funcionarán sin problemas en Windows 2000. COM+, además, incorpora todos los servicios que aportaba MTS, así como otros desconocidos hasta el momento: balanceo de cargas en componentes, componentes con funcionamiento asíncrono, nuevo modelo de eventos, etc. Todos estos servicios forman parte de COM+, no son un añadido al modelo de componentes. Esto significa que estarán disponibles de manera nativa, tanto en Windows 2000 como en futuras versiones de Windows, sin necesidad de instalar añadidos, como ocurría con MTS.

En este artículo podrá conocer algunos de los cambios y ventajas que supone contar con estos nuevos servicios de componentes, llamados COM+. En próximas entregas entraremos con mayor detalle en aspectos puntuales, como el balanceo de cargas o el nuevo modelo de eventos, viendo en la práctica cómo aprovechar dichos servicios en sus aplicaciones.

Terminología y conceptos

Si ha usado MTS en alguna ocasión, sabrá que los componentes se agrupan formando paquetes, que son la entidad mínima de instalación y configuración. Un paquete puede contener múltiples componentes COM, que comparten muchos de sus atributos y, por regla general, se ejecutan en un mismo proceso.

En COM+ el término *paquete* ha sido sustituido por *aplicación*. Una aplicación COM+ es, por tanto, un grupo lógico formado por uno o más componentes y todos sus atributos. El componente en sí estará almacenado en su correspondiente servidor, mientras que los atributos se almacenarán separadamente, en una base de datos optimizada para trabajar con COM+.

Los componentes COM+ siempre estarán alojados en el interior de servidores dentro de proceso. Dicho en otras palabras: los componentes deben formar parte de una librería de enlace dinámico, no de un ejecutable. Esto no implica, no obstante, que todos los componentes vayan a ejecutarse en el mismo espacio de proceso que el cliente. De hecho, en la mayoría de las ocasiones esto no será así, puesto que compromete la estabilidad y limita las posibilidades.

Las aplicaciones COM+ pueden ser de dos tipos: de biblioteca o de servidor. Las aplicaciones de biblioteca son equivalentes a los servidores dentro de proceso, en el sentido de que se ejecutan localmente y utilizando el mismo espacio de direcciones que el proceso del cliente. Las aplicaciones de servidor, que en teoría será el tipo preferente, se ejecutan en un proceso subrogado de COM+, en lugar de hacerlo en el

proceso del cliente. Esto contribuirá a la estabilidad de las aplicaciones, al tiempo que aportará otras posibilidades como la ejecución remota.

Cada una de las aplicaciones COM+ puede contener, como se ha dicho antes, uno o varios componentes. Éstos, a su vez, dispondrán de atributos e interfaces. Las interfaces, por último, estarán compuestas de atributos y métodos. Estos últimos también dispondrán de atributos específicos. Los componentes de una aplicación COM+ se ejecutan en un cierto contexto, definido por todos los atributos y representado por un objeto COM conocido habitualmente como el *objeto de contexto*. Dicho objeto cuenta con interfaces que ofrecen servicios diversos al componente, de tal manera que el programador puede gestionar explícitamente, aunque no sea siempre preciso, aspectos como la seguridad o las transacciones.

El Catálogo COM+

Hasta la aparición de Windows 2000, el único soporte de almacenamiento para los atributos de los componentes COM era el registro de configuraciones. En esta base de datos se almacenaba, básicamente, el CLSID (*Class Identifier*) del componente, su localización física y el modelo de gestión de hilos que utilizaba. No había mucho más que almacenar, puesto que los componentes COM no contaban con más atributos que los mencionados.

En COM+ se utilizan de manera importante los atributos declarativos, propiedades que definen cómo se comporta un componente y condicionan su uso y funcionamiento. Dichos atributos, lógicamente, deben almacenarse en algún soporte físico y persistente. Puesto que el registro de configuraciones de Windows no fue creado con el propósito de almacenar estos atributos, en Windows 2000 existe una nueva base de datos, exclusiva para los componentes COM, conocida como *Catálogo COM+*.

En el pasado, la mayoría de las características funcionales de un componente COM debían implementarse mediante código, siendo una responsabilidad plena del programador. Con COM+ se produce un cambio importante: muchas de esas características se definen mediante atributos, sin necesidad de alterar para nada el código del componente. Un mismo componente, por poner un ejemplo, puede ser usado de manera síncrona, como es tradicional, o asíncrona. Para conseguir esto basta con establecer un par de atributos en la entrada correspondiente del Catálogo COM+.

La tarea de establecimiento de los atributos de un componente se ha convertido, por tanto, en un trabajo administrativo y no de programación. Una vez que el componente se ha desarrollado e instalado en un determinado sistema, el administrador de dicho sistema puede fijar los atributos mediante una utilidad de administración. Lógicamente, éste es un trabajo que también puede efectuar el programador, en caso de que el administrador del sistema no cuente con los conocimientos necesarios acerca de los componentes instalados.

Seguridad basada en perfiles

El nivel de seguridad en los componentes COM+ es muy superior al existente para COM y, además, sin necesidad de que el programador del componente se preocupe por este aspecto. Cada aplicación COM+ tendrá definido uno o más perfiles, a los que se sumarán los usuarios que sean pertinentes. Suponiendo que una aplicación COM+ esté compuesta por componentes para efectuar transacciones bancarias, los posibles perfiles podrían ser administrador, interventor, cajero, etc.

Los perfiles se utilizan para permitir o denegar el acceso a los componentes, a algunas de sus interfaces o, incluso, a los métodos que implementan. De esta forma, es posible que dos usuarios distintos usen el mismo componente pero con limitaciones diferentes, teniendo uno acceso a toda la funcionalidad y otro sólo a una parte, ya sea limitando las interfaces que puede solicitar o los métodos que puede ejecutar.

Si se desea, es posible incluso afinar aún más en la seguridad de acceso a los servicios que ofrecen los componentes. Mediante las interfaces específicas del contexto del componente, el programador puede comprobar si el cliente que está accediendo al servicio pertenece o no a un cierto perfil, adaptando su funcionamiento según este parámetro.

En caso de que el usuario, que actúa como cliente respecto al componente, tenga que acceder a otros recursos, tales como bases de datos, COM+ también cuenta con servicios de delegación de la identificación. De esta forma, es el mismo servidor de aplicaciones el que se encarga de representar al usuario frente al servidor de datos, por mencionar el caso más usual.

Transacciones

De entre los servicios heredados por COM+ de MTS, seguramente el más conocido es el control de transacciones. La posibilidad de gestionar las transacciones al nivel de componentes, en el servidor de aplicaciones, permite efectuar operaciones que, de otra forma, resultarían mucho más complejas.

La mayoría de los programadores asocian el uso de transacciones al trabajo con bases de datos, al ser éstas uno de los recursos en los que es indispensable asegurar la integridad de la información. Las transacciones COM+ se sitúan en un nivel superior, pudiendo abarcar operaciones no sólo sobre una determinada base de datos, como es habitual, sino entre varias bases de datos que, incluso, podrían estar en diferentes servidores físicos.

El control de las transacciones en MTS ya era bastante simple. Básicamente, se trata de indicar qué tipo de servicios de transacción precisa un cierto componente el cual, cuando sea preciso, indicará al sistema la finalización o interrupción, según los casos. En COM+ se ha añadido una nueva posibilidad: la terminación automática de una transacción. Esto puede ocurrir cuando se produce una excepción en el componente, caso en el cual la transacción se cancelaría, o bien cuando el componente finaliza su trabajo, haciendo firmes las operaciones efectuadas durante la transacción.

Para permitir que una transacción pueda tener lugar en un entorno distribuido, COM+ se sirve del Coordinador de transacciones distribuidas, conocido como MS DTC (*Distributed Transaction Coordinator*), y de unos gestores de recursos, que son los que realmente acceden a la información que se compartirá entre los distintos clientes.

Balanceo de cargas

Uno de los problemas que presentan habitualmente los servidores de aplicaciones, al igual que los servidores de datos, es la escalabilidad. En un momento u otro, el número de usuarios que requieren el acceso a un componente es tan importante que el rendimiento cae, hasta llegar a tiempos de respuesta inaceptables.

Las versiones superiores de Windows 2000, concretamente las ediciones *Advanced Server* y *DataCenter Server*, incluyen un servicio de balanceo o equilibrio de cargas para los componentes. Dicho servicio permite crear un grupo o *cluster* de servidores de aplicaciones que, de manera lógica, forman un solo servidor. De ese grupo, una de las máquinas actúa como equilibrador de cargas, analizando cuál es el tiempo de respuesta de cada servidor y derivando la creación de componentes, a medida que el cliente lo demande, al servidor que esté menos ocupado.

Que un determinado componente COM+ pueda o no ser utilizado en un entorno de equilibrio de cargas, es algo que depende tan sólo de un atributo que, como todos los demás, es almacenado en el Catálogo COM+. Aunque el programador, al crear el componente, no tiene que hacer nada especial, sí que debe tener en cuenta que el componente no podrá asumir que está ejecutándose en una cierta máquina ya que, en la práctica, el entorno de ejecución dependerá de la carga que exista.

La posibilidad de crear un *cluster* de servidores de aplicaciones, con el fin de incrementar la escalabilidad y mejorar el tiempo de respuesta global, también influye decisivamente en la disponibilidad de los servicios, parámetro no menos importante que la escalabilidad. En caso de que uno de los servidores de aplicaciones caiga por alguna causa, el servidor de balanceo de cargas repartirá el trabajo entre los que quedan en servicio, de tal forma que los clientes podrán seguir trabajando sin problema alguno.

Componentes de funcionamiento asíncrono

Otra novedad importante de COM+ es la posibilidad de utilizar componentes de manera asíncrona, de tal forma que cliente y servidor no tienen por qué ejecutarse en el mismo espacio de tiempo, algo que es habitual, sobre todo, en entornos móviles. El funcionamiento de los componentes asíncronos está basado en los servicios de MSMQ (*Microsoft Message Queuing Services*), un servidor que deberá estar funcionando en la red y que se incluye en Windows 2000.

Los servidores que contienen componentes asíncronos, las aplicaciones COM+, no se diferencian de cualquier otro servidor salvo en algunos atributos, mediante los cuales se indicará que el componente puede ser iniciado a través de una cola de mensajes MSMQ y que los métodos pueden ser invocados a

través de ese sistema. El programador del componente, por tanto y una vez más, no tiene por qué preocuparse de este aspecto.

La creación de los clientes, sin embargo, sí implica algunos cambios de filosofía, aunque realmente afectan tan sólo a la creación del componente que quiere utilizarse y poco más. Cuando el cliente no puede acceder directamente al componente, porque éste no se encuentra en ejecución o está en una red a la que no hay acceso en ese momento, lo que obtiene a cambio es una referencia a un objeto que actúa como una grabadora. La grabadora, que no es más que un objeto COM+ que muestra las mismas interfaces que el componente que se quiere usar, se encarga de registrar todas las operaciones y los parámetros asociados. Cuando el cliente se conecta a la red en el que está ejecutándose el componente, o bien cuando éste inicia su ejecución si es que estaba detenido, otro objeto, actuando como reproductor, se encargará de tomar el registro y realizar todas las operaciones detalladas.

Al funcionar de manera asíncrona, está claro que el cliente no puede esperar una respuesta inmediata del servidor, puesto que éste podría ejecutarse horas o días después, dependiendo de los casos y el contexto. Por la misma causa, no es posible utilizar parámetros de salida al llamar a los métodos, ya que éstos no pueden devolver valores de manera inmediata.

Otros servicios de COM+

Además de los ya mencionados, COM+ cuenta con otros servicios también destacables. Al igual que MTS, COM+ cuenta con un depósito de objetos reutilizables que el administrador del sistema puede configurar según las necesidades. Los programadores también deben seguir unas ciertas normas al crear componentes que puedan reutilizarse, de tal forma que éstos puedan quedar en el mencionado depósito, inactivos, activándose de nuevo en cualquier momento, cuando un cliente lo requiera. Esta activación inmediata, conocida como JIT (*Just In-Time*), tiene la ventaja de que lo que se usa es un componente ya existente, no hay necesidad de crear un nuevo componente y, por tanto, el tiempo de respuesta es menor, mejorando el rendimiento.

También hay que mencionar la existencia de un nuevo modelo de apartamento para los componentes, conocida como NTA (*Neutral Thread Apartment*). Éste viene a minimizar los problemas que se presentan cuando un componente diseñado para ejecutarse en un STA (*Single Thread Apartment*) es utilizado en una aplicación con múltiples hilos de ejecución, lo cual implica cambios de contexto que consumen tiempo y afectan negativamente al rendimiento. Con el nuevo NTA este problema se minimiza, al no ser preciso un cambio de contexto, de un hilo a otro, tan caro en tiempo como el existente hasta el momento.

Administración de los servicios de componentes

Como se ha repetido en los puntos anteriores, la mayor parte de los nuevos servicios y características de los componentes COM+ no implican cambios en la codificación, tan sólo una mayor tarea de administración. Ésta se efectúa mediante el nuevo Administrador de servicios de componentes, al que se tiene acceso, como puede ver en la figura 1, a través del Panel de control. Esta utilidad, según puede verse en la figura 2, guarda una cierta similitud con el Explorador de MTS.

Desde el Administrador de servicios de componentes es posible gestionar tanto el equipo local como otros equipos remotos que ejecuten Windows 2000, lo cual simplifica la tarea de administración al no ser preciso desplazarse físicamente a cada equipo. Puede crear nuevas aplicaciones COM+, insertar componentes en ellas y editar todos los atributos. En la figura 2 puede ver dos paneles: el de la izquierda muestra los diversos elementos que es posible administrar, en forma de árbol jerárquico, mientras que en el de la derecha aparecen los atributos de las aplicaciones COM+ existentes.

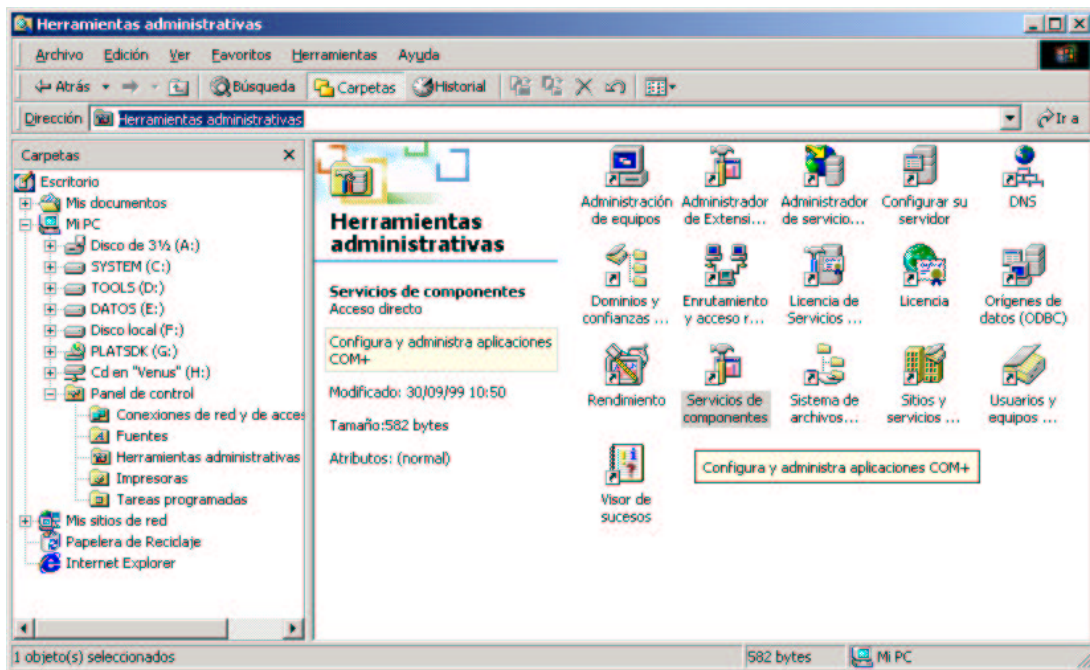


Figura 1. Desde el Panel de control es posible iniciar la utilidad de administración de los servicios de componentes

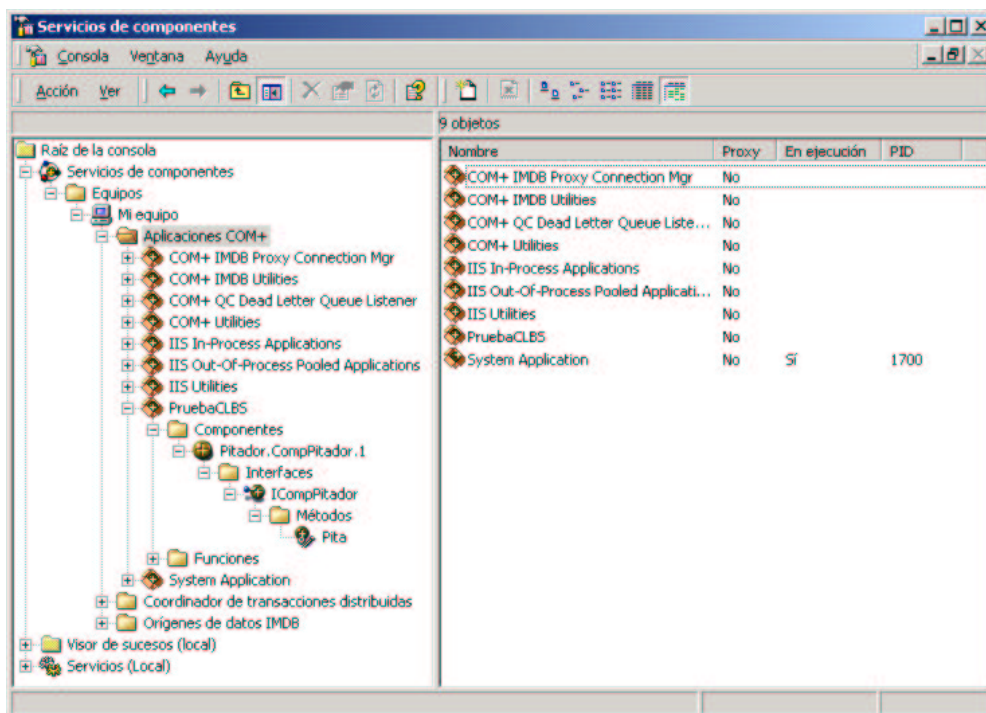


Figura 2. La utilidad de administración de los servicios de componentes es una interfaz para manipular el Catálogo COM+, creando y editando los atributos de las aplicaciones COM+

Anteriormente se indicaba la existencia de dos tipos de aplicaciones COM+: de biblioteca o de servidor. Una vez creada la aplicación, se puede establecer este atributo desde una página de propiedades (véase la figura 3). Observe que con esta acción estaría estableciéndose el método de activación de los componentes, sin que por ello sea preciso modificar para nada su código.

El anterior es un atributo establecido para toda la aplicación. También existen atributos al nivel de componente. En la figura 4, por ejemplo, puede ver cómo se elige el uso de transacciones que precisa un cierto componente de la aplicación previa. De igual manera, podrían establecerse los parámetros de seguridad o sincronización.

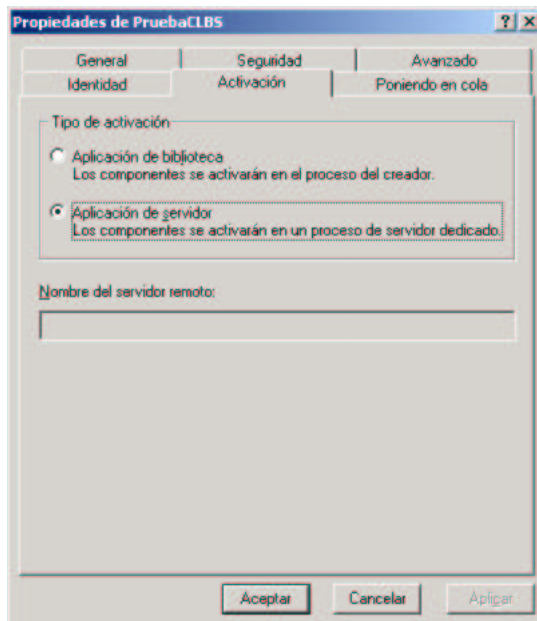


Figura 3. Una vez creada la aplicación, puede especificarse el método de activación simplemente eligiendo una opción

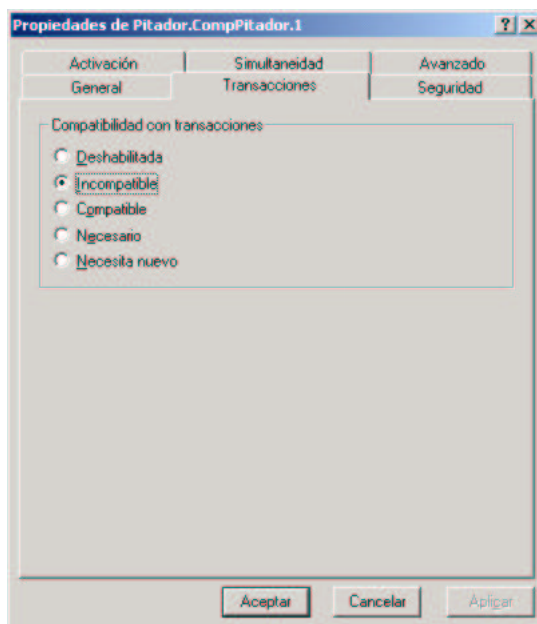


Figura 4. El uso de transacciones que hará un componente debe indicarse eligiendo una opción de su página de propiedades, de tal forma que COM+ asocie el contexto adecuado al ejecutar dicho componente

Una vez que se ha creado una aplicación, incluido en ella los componentes que hagan falta y establecido todos los atributos, desde el propio Administrador de servicios de componentes es posible exportar la aplicación, creando un paquete para instalación en otros sistemas. Esta funcionalidad simplifica la distribución de aplicaciones, ya que los receptores tan sólo tienen que instalar el paquete recibido, sin preocuparse de tareas de registro, configuración o similares. Estas acciones, además, pueden efectuarse de forma remota, como se ha dicho anteriormente.

Una técnica llamada interceptación

¿Cómo podría COM+ ofrecer todos los servicios descritos, hacer su trabajo, si el cliente obtuviese un acceso directo a las interfaces del componente? Está claro que en algún punto intermedio, interrumpiendo

ese canal de comunicación entre ambos extremos, debe existir algún elemento que se encargue de realizar el trabajo que el programador no tiene que hacer.

La existencia de los *proxy* y *stub* no es algo nuevo, han existido desde siempre para facilitar la comunicación con componentes fuera de proceso y componentes remotos. A esos elementos, que siguen utilizándose cuando se precisan, hay que añadir ahora unos agentes intermedios, colocados en el cliente o en el servidor según corresponda. Dichos agentes que preocupan de muchas de las tareas descritas, como el control de las transacciones o la seguridad.

Para denominar a esta técnica, en la cual el canal de comunicación entre cliente y componente se encuentra *pinchado* por diversos agentes, se utiliza el término *interceptación*. Es bastante lógico, puesto que el trabajo de esos agentes es, precisamente, interceptar las llamadas y retornos, interpretando la información y actuando según proceda.

Resumiendo

Tras todas las novedades y aspectos comentados en este artículo, es fácil darse cuenta de que COM ha dejado de ser una simple especificación, poco más que eso era hasta ahora, para convertirse en un conjunto de servicios que simplifican de manera notable la tarea de crear componentes y administrar servidores de aplicaciones. Hasta la llegada de Windows 2000, el sistema ofrecía unos servicios muy limitados: prácticamente la creación del componente y poco más. Con COM+ los servicios disponibles son muchos e interesantes: gestión de transacciones, equilibrado de carga en los servidores, componentes con funcionamiento asíncrono, etc. Y lo mejor de todo, es que para aprovechar muchos de estos servicios el programador no tiene que conocer nada nuevo.