

QUÉ ES LA WEB 2.0.

**PATRONES DEL DISEÑO Y MODELOS DEL NEGOCIO PARA LA
SIGUIENTE GENERACIÓN DEL SOFTWARE**

Tim O'Reilly

Traducción al español: Equipo del Boletín de la Sociedad de la Información de Telefónica

El estallido de la burbuja tecnológica en el otoño de 2001 marcó un momento crucial para la web. Mucha gente concluyó que la expectación sobre la web era exagerada, cuando de hecho [las burbujas y las consiguientes crisis económicas parecen ser una característica común de todas las revoluciones tecnológicas](#). Las crisis económicas marcan típicamente el punto en el cual una tecnología en ascenso está lista para ocupar su lugar en el escenario económico. Se descarta a los impostores, las historias de éxito verdaderas muestran su fortaleza, y comienza a comprenderse qué separa a los unos de los otros.

El concepto de 'Web 2.0' comenzó con una sesión de *'brainstorming'* realizada entre O'Reilly y MediaLive International. Dale Dougherty, pionero de la web y vicepresidente de O'Reilly, observaron que lejos de 'estrellarse', la web era más importante que nunca, con apasionantes nuevas aplicaciones y con sitios web apareciendo con sorprendente regularidad. Lo que es más, las compañías que habían sobrevivido al desastre parecían tener algunas cosas en común. ¿Podría ser que el derrumbamiento de las punto-com supusiera algún tipo de giro crucial para la web, de tal forma que una llamada a la acción tal como 'Web 2.0' pudiera tener sentido? Estuvimos de acuerdo en que lo hizo, y así nació [la conferencia de la Web 2.0](#).

Tras año y medio, el término 'Web 2.0' ha arraigado claramente, con más de 9,5 millones de menciones en Google. Pero todavía existe [un enorme de desacuerdo sobre qué significa Web 2.0](#), existiendo algunas críticas que afirman que se trata simplemente de una palabra de moda, fruto del marketing, y sin sentido, en tanto que otros la aceptan como un nuevo paradigma.

Este artículo es una tentativa de aclarar a lo que nos referimos nosotros cuando hablamos de Web 2.0. En nuestra reunión inicial de *brainstorming*, formulamos nuestra interpretación de Web 2.0 según un ejemplo:

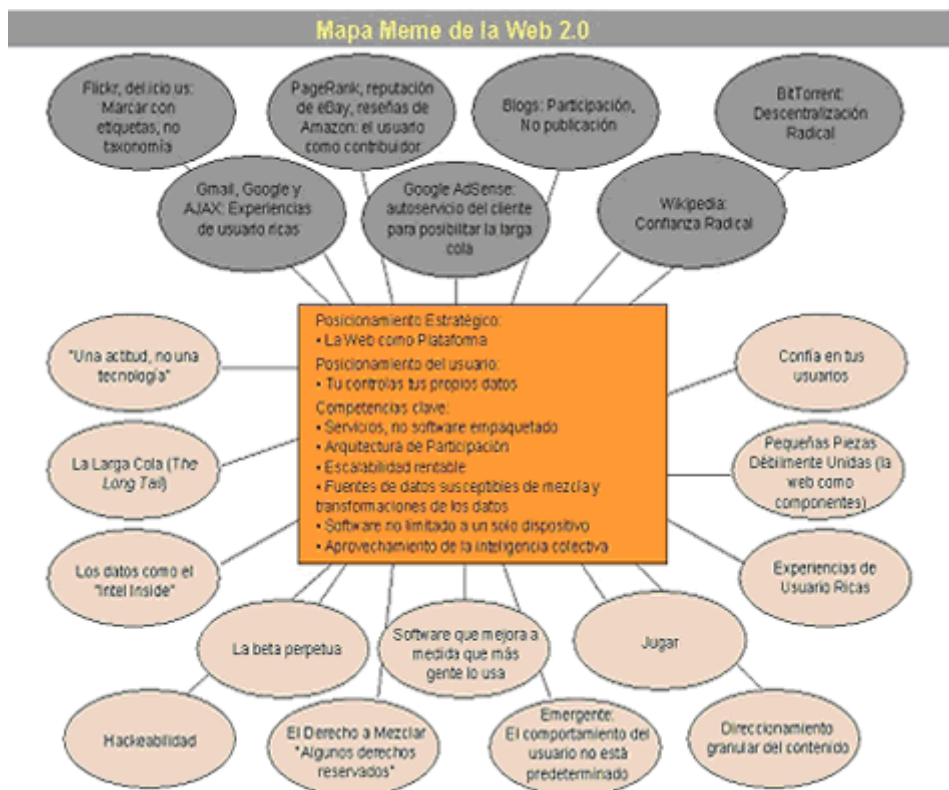
Web 1.0		Web 2.0
Doble click	-->	Google AdSense
Ofoto	-->	Flickr
Akamai	-->	BitTorrent
mp3.com	-->	Napster
Britannica Online	-->	Wikipedia
personal websites	-->	blogging
evite	-->	upcoming.org and EVDB
domain name speculation	-->	search engine optimization
page views	-->	cost per click
screen scraping	-->	web services

publishing	-->	participation
content management systems	-->	wikis
directories (taxonomy)	-->	tagging ('folksonomy')
stickiness	-->	syndication

La lista crecía y crecía. ¿Pero qué era lo que nos permitía asociar una aplicación o enfoque a 'Web 1.0' y otro a 'Web 2.0'? (La pregunta es particularmente apremiante, porque el ['meme'](#) de la Web 2.0 ha llegado a ser tan extenso, que las compañías están usando el término como una nueva palabra de moda fruto del marketing, sin comprender realmente lo que significa. Este asunto es particularmente difícil porque muchas de esas *startups* adictas a las palabras de moda no son en absoluto Web 2.0, mientras que algunas de las aplicaciones que asociamos a la Web 2.0, como Napster y BitTorrent, ¡no son, en sentido estricto, ni siquiera aplicaciones web!). Comenzamos por intentar extraer los principios que se deducen de una forma u otra de las historias de éxito de la Web 1.0 y por lo más interesante de las nuevas aplicaciones.

1. La Web como plataforma

Como muchos conceptos importantes, Web 2.0 no tiene una clara frontera, sino más bien, un núcleo gravitacional. Usted puede [visualizar Web 2.0](#) como un sistema de principios y prácticas que conforman un verdadero sistema solar de sitios que muestran algunos o todos esos principios, a una distancia variable de ese núcleo.



La figura anterior muestra un 'mapa meme' de la Web 2.0 que fue desarrollado en una sesión de *brainstorming* durante el FOO Camp, una conferencia en O'Reilly Media. Es fundamentalmente trabajo en curso, pero manifiesta las muchas ideas que irradian desde el núcleo de la Web 2.0.

Por ejemplo, en la primera conferencia de la Web 2.0, en octubre de 2004, John Battelle y yo enumeramos un conjunto preliminar de principios en nuestra charla de apertura. El primero de dichos principios era 'la web como plataforma.' Aunque esto era también un eslogan para lograr apoyos a favor del querido Netscape de la Web 1.0, que resultó derrotado tras una encendida batalla con Microsoft. Más aún, dos de nuestros ejemplos iniciales de Web 1.0, DoubleClick y Akamai, eran ambos pioneros en tratar a la web como una plataforma. La gente no piensa habitualmente en ello como 'web services', pero de hecho, la publicidad en sitios web fue el primer '*web service*' ampliamente desplegado, y el primer '*mashup*' (por utilizar otro término que ha ganado adeptos recientemente) que se desplegó extensamente. Cada *banner* actúa como elemento que facilita la cooperación transparente entre dos *websites*, proporcionando una página integrada a un lector en otro ordenador. Akamai también trata la red como plataforma, y en un nivel inferior de la pila, construyendo una red transparente de copia en caché y de distribución de contenidos que alivie la congestión del ancho de banda.

No obstante, estos pioneros proporcionaron contrastes útiles porque competidores posteriores han llevado su solución del mismo problema aún más lejos, entendiendo de forma más profunda la naturaleza de la nueva plataforma. DoubleClick y Akamai fueron pioneros de la Web 2.0, sin embargo podemos también ver cómo es posible materializar más posibilidades adoptando [patrones de diseño adicionales de Web 2.0](#).

Vamos profundizar por un momento en cada uno de estos tres casos, desentrañando algunos de los elementos diferenciadores esenciales.

Netscape frente a Google

Si Netscape era el abanderado de la Web 1.0, Google es ciertamente el abanderado de la Web 2.0, aunque sólo sea porque sus respectivas salidas a bolsa fueron acontecimientos determinantes para cada era. Vamos comenzar con una comparación de estas dos compañías y de su posicionamiento.

Netscape ideó el concepto de 'la web como plataforma' en términos del viejo paradigma del software: su buque insignia era el navegador web, una aplicación de escritorio, y su estrategia era utilizar su dominio en el mercado de los navegadores para crear un mercado de productos de servidor de gama alta. El control sobre los estándares para visualizar el contenido y las aplicaciones en el navegador, en teoría, dio a Netscape la clase de poder de mercado del que disfrutó Microsoft en el mercado de los PCs. Al igual que el 'carro sin caballos' posicionó al automóvil como extensión de lo conocido, Netscape promovió un '*webtop*' para sustituir al escritorio (el '*desktop*'), y planeó poblar ese *webtop*

con las actualizaciones de información y *applets* insertados en el *webtop* por los proveedores de información que comprarían los servidores de Netscape.

Sin embargo, al final, los navegadores web y los servidores web resultaron ser *commodities*, y el valor se desplazó hacia los servicios ofrecidos sobre la plataforma web.

Google, por el contrario, comenzó su vida como una aplicación web nativa, nunca vendida o empaquetada, sino siempre entregada como un servicio, con clientes pagando, directamente o indirectamente, por el uso de ese servicio. Ninguna de las rémoras de la vieja industria del software están presentes. No hay programación de las actualizaciones de las versiones del software, sencillamente mejora continua. Ninguna licencia o venta, sencillamente uso. Ningún tipo de portabilidad a diferentes plataformas de forma que los clientes puedan ejecutar el software en su propio equipo, sencillamente, una colección masiva de PCs escalables en los que corren sistemas operativos de software abierto junto con aplicaciones y utilidades de su propia cosecha que nunca nadie de fuera de la compañía consigue ver.

En el fondo, Google requiere una capacidad que Netscape nunca necesitó: gestión de la base de datos. Google no es sencillamente una colección de herramientas software, es una base de datos especializada. Sin los datos, las herramientas son inútiles; sin el software, los datos son inmanejables. El licenciamiento del software y el control sobre las APIs (la palanca de poder en la era anterior) es irrelevante porque el software no necesita ser distribuido sino ejecutado, y también porque sin la capacidad de recoger y de gestionar los datos, el software es de poca utilidad. De hecho, **el valor del software es proporcional a la escala y al dinamismo de los datos que ayuda a gestionar** .

El servicio de Google no es un servidor (aunque es ofrecido por una colección masiva de servidores de Internet) ni un navegador (aunque es experimentado por el usuario a través del navegador). Ni siquiera su servicio insignia, el de búsqueda, almacena el contenido que permite encontrar a los usuarios. Como una llamada telefónica, que no tiene lugar en los teléfonos de los extremos de la llamada sino en la red que hay entre medias, Google tiene lugar en el espacio que se encuentra entre el navegador y el motor de búsqueda y el servidor de contenido destino, como un habilitador o intermediario entre el usuario y su experiencia *online*.

Aunque Netscape y Google se podrían describir como compañías de software, está claro que Netscape perteneció al mismo mundo del software que Lotus, Microsoft, Oracle, SAP, y otras compañías que surgieron durante la revolución del software de los años 80, mientras que los amigos de Google son aplicaciones de Internet como eBay, Amazon, Napster, y sí, DoubleClick y Akamai.

DoubleClick frente a Overture y AdSense

Como Google, DoubleClick es un verdadero hijo de la era del Internet. Ofrece software como un servicio, tiene una competencia básica de gestión de datos, y, según lo mencionado anteriormente, era un pionero en web services mucho antes de que los web services tuvieran un nombre. Sin embargo, finalmente DoubleClick se vio limitado por su modelo de negocio. Apoyó en los años 90 el concepto de que la web trataba de publicación, no participación; que los publicistas, no los consumidores, deben ser los que deciden; que el tamaño importaba, y que Internet cada vez estaba más dominada por los sitios web situados en la cima según las estadísticas de MediaMetrix y otras compañías que valoraban los anuncios de la web.

Como consecuencia, DoubleClick cita orgulloso en su web 'más de 2000 implementaciones exitosas' de su software. ¡Yahoo! Search Marketing (antes Overture) y Google [AdSense](#), por el contrario, ya dan cada uno servicio a centenares de millares de publicistas.

El éxito de Overture y de Google fue fruto de la comprensión de lo que Chris Anderson cita como '*the long tail*' (literalmente 'la larga cola'), el poder colectivo de los sitios web pequeños que conforman la gran mayoría del contenido de la web. Las ofertas de DoubleClick requieren un contrato formal de venta, limitando su mercado a unos pocos miles de sitios web grandes. Overture y Google se las ingeniaron para permitir la colocación del anuncio prácticamente en cualquier página web. Lo que es más, evitaron los formatos de publicidad preferidos por los publicistas y las agencias de publicidad como *banners* y *popups* (ventanas emergentes), en favor de los anuncios de texto, mínimamente intrusivos, sensibles al contexto y amigables para el consumidor.

La lección de la Web 2.0: **hacer uso del autoservicio del cliente y de la gestión de datos algorítmica para llegar a toda la web, a los extremos y no sólo al centro, a 'la larga cola' ('*the long tail*') y no sólo a la cabeza.**

Como es de esperar, otras historias de éxito de la Web 2.0 demuestran este mismo comportamiento. eBay permite las transacciones ocasionales de tan solo algunos dólares entre simples individuos, actuando como un intermediario automatizado. Napster (aunque cerrado por razones legales) construyó su red no mediante la construcción de una base de datos centralizada de canciones, sino arquitecturando un sistema en el que cada individuo que descargaba algo también se convertía en un servidor (esto es, alguien del que otros se descargaban algo), y así creció la red.

Akamai frente a BitTorrent

Como DoubleClick, Akamai está optimizado para hacer negocios con la cabeza, no con la cola, con el centro, no con los extremos. Mientras que sirve a las necesidades de los individuos en el extremo de la web facilitando su acceso a los sitios web de mucha demanda en el centro, obtiene sus ganancias de esos sitios centrales.

BitTorrent, como otros pioneros en el movimiento del P2P, adopta el enfoque radical de la descentralización del Internet. Cada cliente es también un servidor; los archivos están subdivididos en fragmentos que se pueden servir desde múltiples localizaciones, aprovechando de forma transparente la red de los individuos que se están descargando archivos para proporcionar tanto ancho de banda como datos a otros usuarios. De hecho, cuanto más popular es el archivo, más rápidamente se descarga, puesto que hay más usuarios que proporcionan ancho de banda y fragmentos del archivo completo.

BitTorrent demuestra así un principio dominante de la Web 2.0: **el servicio mejora automáticamente cuanto más gente lo use**. Mientras que Akamai debe agregar servidores para mejorar el servicio, cada consumidor de BitTorrent aporta sus propios recursos al grupo. Hay una 'arquitectura implícita de participación', una ética de cooperación inherente, en la que el servicio actúa sobre todo como intermediario inteligente, conectando los extremos entre sí y aprovechando las posibilidades que ofrecen los propios usuarios.

2. Aprovechando la inteligencia colectiva

El principio fundamental que se esconde detrás del éxito de los gigantes nacidos en la era de la Web 1.0 que han sobrevivido para liderar la era de la Web 2.0 parece ser éste, que han abrazado el poder de la web para explotar inteligencia colectiva:

- Los hipervínculos constituyen los cimientos de la web. A medida que los usuarios agregan nuevo contenido, y sitios web nuevos, se enlazan con la estructura de la web gracias a otros usuarios que descubren el contenido y enlazan con él. De forma muy parecida a la sinapsis del cerebro, donde las asociaciones llegan a ser más fuertes a través de la repetición o la intensidad, la red de conexiones crece orgánicamente como resultado de la actividad colectiva de todos los usuarios de la web.
- Yahoo!, la primera gran historia del éxito de Internet, nació como un catálogo, o un directorio de enlaces (links), un agregado del mejor trabajo de millares, después millones de usuarios de la web. A pesar de que Yahoo! ha avanzado hacia el negocio de crear muchos tipos de contenido, su papel como portal del trabajo colectivo de los usuarios de la red sigue siendo la base de su valor.
- La innovación de Google en la búsqueda, que rápidamente le convirtió en el indiscutible líder del mercado de la búsqueda, fue PageRank, un método para usar la propia estructura de enlaces de la web para proporcionar mejores resultados de búsqueda, en lugar de usar sólo las características de los documentos.
- El producto de eBay es la actividad colectiva de todos sus usuarios; como la web en sí misma, eBay crece orgánicamente en respuesta a la actividad del usuario, y el papel de la compañía es el de habilitador de un contexto en el cual pueda tener lugar esa actividad del usuario. Lo que es más, la ventaja competitiva de eBay proviene casi enteramente de la masa crítica de compradores y de vendedores,

que convierte a cualquier nuevo competidor que ofrezca servicios similares significativamente menos atractivo.

- Amazon vende los mismos productos que competidores tales como Barnesandnoble.com, y reciben las mismas descripciones del producto, imágenes de la cubierta y contenido editorial de sus proveedores. Pero Amazon ha creado una ciencia de la gestión del usuario. Tienen un orden de magnitud más que sus competidores de reseñas de los usuarios, tienen invitaciones para participar de diversas maneras en prácticamente todas las páginas, y más importantemente aún, utilizan actividad del usuario para producir mejores resultados de búsqueda. Mientras que una búsqueda en Barnesandnoble.com es probable que conduzca a los propios productos de la compañía, o a los resultados patrocinados, Amazon conduce siempre 'al más popular', mediante un cómputo en tiempo real basado no solamente en las ventas sino también en otros factores que los miembros de Amazon llaman el 'flujo' inducido alrededor de los productos. Con una participación del usuario un orden de magnitud mayor, no es ninguna sorpresa que las ventas de Amazon también aventajen a las de sus competidores.

En la actualidad, las compañías innovadoras que adoptan esta idea y quizás incluso la extienden más allá, están dejando su marca en la web:

- Wikipedia, una enciclopedia en línea basada en la inverosímil idea de que una entrada puede ser agregada por cualquier usuario de la web, y corregida por cualquier otro, es un experimento radical de confianza, aplicando la máxima de Eric Raymond (acuñado originalmente en el contexto del [software abierto](#)) de que 'con ojos suficientes, todos los fallos son superficiales' para la generación de contenido. Wikipedia está ya entre las 100 webs más visitadas, y muchos piensan que llegará a estar entre las 10 de la cima en poco tiempo. ¡Esto sí que es un cambio profundo en la dinámica de la creación de contenidos!
- Sitios como del.icio.us y [Flickr](#), dos compañías que han recibido mucha atención últimamente, han promovido un concepto que alguna gente llama 'folksonomy'⁽¹⁾ (en contraste con la taxonomía), un estilo de clasificación colaborativa de sitios usando palabras claves libremente elegidas, a menudo denominadas etiquetas (tags). El marcado con etiquetas permite la clase de asociaciones múltiples, y solapadas que el propio cerebro humano utiliza, en lugar de categorías rígidas. En el ejemplo canónico, una foto de Flickr de un cachorro puede ser marcada con la etiqueta tanto 'cachorro' como 'lindo' – permitiendo la recuperación siguiendo los mismos ejes naturales generados por la actividad del usuario.
- Los productos de filtrado cooperativo de spam como Cloudmark agregan las decisiones individuales de los usuarios del correo electrónico sobre qué es y qué no es spam, funcionando mejor que los sistemas que confían en el análisis de los propios mensajes.
- Es una perogrullada que las mejores historias de éxito de Internet no anuncian sus productos. Su adopción es impulsada por el 'marketing viral', es decir, recomendaciones propagándose directamente de un usuario a otro. Usted puede

casi asegurar que si un sitio o un producto confían en la publicidad para conseguir el impulso necesario, no es Web 2.0.

- Incluso gran parte de la infraestructura de la web (incluyendo Linux, Apache, MySQL, y Perl, PHP, o el código de Python incluido en la mayoría de los servidores web) confía en los métodos de [producción entre pares \('peer-production'\)](#) del software abierto, en sí mismo un ejemplo de inteligencia colectiva y habilitadora por la red. Hay más de 100.000 proyectos de software abierto listados en [SourceForge.net](#). Cualquier persona puede agregar un proyecto, cualquier persona puede descargarse y utilizar el código, y los nuevos proyectos emigran de los extremos al centro como resultado de ser puestos en funcionamiento por los usuarios, un proceso orgánico de adopción del software que se basa casi enteramente en el marketing viral.
- Esto no quiere decir que no haya oportunidades para asegurar beneficios y conseguir una ventaja competitiva, pero creemos que no se deben obtener mediante el control sobre el software basado en APIs y los protocolos. Hay un nuevo juego en marcha. Las compañías que tendrán éxito en la era de la Web 2.0 serán las que entiendan las reglas de ese juego, en vez de intentar volver a las reglas de la era del software de PC.
- La lección: **Las externalidades de red derivadas de las contribuciones del usuario son la clave para el dominio del mercado en la era de la Web 2.0.**

La plataforma supera a la aplicación en todo momento

En cada una de sus últimas confrontaciones con los rivales, Microsoft ha jugado con éxito la carta de la plataforma, triunfando incluso sobre las aplicaciones más dominantes. Windows permitió que Microsoft desplazara el Lotus 1-2-3 con Excel, WordPerfect con Word, y Netscape Navigator con Internet Explorer.

Esta vez, sin embargo, el choque no es entre una plataforma y una aplicación, sino entre dos plataformas, cada una con un modelo de negocio radicalmente distinto: en un lado, un solo suministrador de software, cuya base masivamente instalada y sus firmemente integrados sistema operativo y APIs le proporcionan el control sobre el paradigma de programación; en el otro, un sistema sin un dueño, agrupado mediante una serie de protocolos, estándares abiertos y acuerdos de cooperación.

Windows representa la cumbre del control propietario mediante el software basado en APIs. Netscape intentó arrebatar el control a Microsoft usando las mismas técnicas que el propio Microsoft había utilizado contra otros rivales, y falló. Pero Apache, que se aferró a los estándares abiertos de la web, ha prosperado. La batalla ya no es desigual, una plataforma contra una sola aplicación, sino plataforma frente a plataforma, siendo más bien la pregunta qué plataforma, y más profundamente, qué arquitectura, y qué modelo de negocio, se ajustan más a la oportunidad que se presenta por delante.

Windows era una solución brillante a los problemas de la era inicial del PC. Igualó las condiciones para todos los desarrolladores de aplicaciones, solucionando una multitud de problemas que previamente habían asediado a la industria. Pero una sola aproximación monolítica, controlada por un solo proveedor, ya no es una solución, es un problema. Los sistemas orientados hacia las comunicaciones, algo que ciertamente es Internet como plataforma, requieren interoperabilidad. [A menos que un proveedor pueda controlar ambos extremos de cada interacción](#), las posibilidades de conseguir usuarios cautivos mediante el software basado en APIs son limitadas.

Cualquier proveedor de la Web 2.0 que intente asegurar los beneficios de su aplicación mediante el control de la plataforma, por definición, no estará contribuyendo al fortalecimiento de la plataforma.

Bloggging y la sabiduría de las masas

Una de las características más altamente pregonadas de la era de la Web 2.0 es el auge del bloggging. Las páginas personales han existido desde los comienzos de la web, y los diarios y las columnas diarias de opinión personal existen desde hace aún más tiempo, entonces ¿por qué tanto alboroto?

En su aspecto más básico, un blog es sencillamente una *home page* personal en formato de diario. Pero tal y como [apunta Rich Skrenta](#), la organización cronológica de un blog

'parece una diferencia trivial, pero conduce a una cadena de reparto, de publicidad y de valor completamente distinta'.

Una de las cosas que ha marcado la diferencia es una tecnología llamada [RSS](#). RSS es el avance más significativo de la arquitectura básica de la web desde que los primeros hackers se dieron cuenta de que el CGI se podía utilizar para crear sitios web basados en bases de datos. RSS permite que alguien no sólo enlace con una página, sino suscribirse a la misma, con notificaciones cada vez que la página cambia. Skrenta denomina a esto 'la web incremental.' Otros lo llaman la 'web viva'.

Ahora bien, por supuesto, los 'sitios web dinámicos' (es decir, los sitios web basados en bases de datos con el contenido dinámicamente generado) sustituyeron a las páginas estáticas de la web hace más de diez años. Lo que es dinámico en la web viva no son sólo las páginas, sino los enlaces. Se espera que un enlace a un weblog señale a una página que cambia perennemente, con los enlaces permanentes ('[permalinks](#)') para cualquier entrada individual, y con notificación de cada cambio. Una fuente RSS es, de esta manera, un enlace mucho más fuerte que, digamos un *bookmark* o un enlace a una página concreta.

RSS también significa que el navegador web no es el único medio para ver una página web. Mientras que agregadores de RSS, tales como Bloglines, son aplicaciones web, otros son clientes de escritorio, e incluso otros permiten que los usuarios de dispositivos portátiles se suscriban al contenido permanentemente actualizado.

RSS ahora se está utilizando no sólo para enviar avisos de nuevas entradas de un blog, sino también para todo tipo de actualizaciones de datos, incluyendo las cotizaciones en bolsa, información metereológica, y la disponibilidad de fotos. Este uso es realmente un retorno a una de sus raíces: RSS nació en 1997 de la confluencia de la tecnología '*Really Simple Syndication*' de Dave Winer, utilizada para informar de actualizaciones de blog, y de '*Rich Site Summary*' de Netscape, que permitió que los usuarios crearan *home pages* personalizadas en la web de Netscape con flujos de datos actualizados regularmente. Netscape perdió interés, y la tecnología fue continuada por el pionero del blogging Userland, de la compañía de Winer. En el grupo de aplicaciones actuales, podemos apreciar, no obstante, la herencia de ambos padres.

Pero RSS es solamente parte de lo que hace a un weblog diferente de una página web ordinaria. Tom Coates comenta en [el significado del enlace permanente \(el '\[permalink\]\(#\)' \)](#):

Ahora puede parecer una funcionalidad trivial, pero fue el esquema que en verdad permitió a los weblogs pasar de ser simplemente un fenómeno basado en la facilidad para publicar en un caos informal de comunidades solapadas. Por primera vez resultó relativamente fácil discutir directamente acerca de una nota sumamente concreta del sitio web de otra persona y hablar sobre ello. La discusión afloró. El chat afloró. Y, consecuentemente, las amistades surgieron o se estrecharon. El *permalink* era el primer, y el más acertado, intento de construir puentes entre weblogs.

En gran medida, la combinación de RSS y de permalinks añade muchas de las características de NNTP, el entrañable Network News Protocol del servicio de noticias ('news') de Usenet, al HTTP, el protocolo de la web. Se puede pensar en la 'blogoesfera' ('*blogosphere*') como en un equivalente nuevo, de igual a igual a Usenet y los boletines de noticias, los oasis de charla informal del antiguo Internet. La gente puede no sólo suscribirse a sitios web de otras personas, y enlazar fácilmente con los comentarios en cuestión de una página, sino que también, a través de un mecanismo conocido como *trackbacks*, puede ver cuando alguien enlaza con sus páginas, y puede responder, o con enlaces recíprocos, o agregando comentarios.

Curiosamente, los enlaces bidireccionales eran la meta de los primeros sistemas de hipertexto como Xanadu. Los puristas del hipertexto han festejado *trackbacks* por considerarlo el avance hacia los enlaces bidireccionales. Pero obsérvese que los *trackbacks* no son verdaderamente bidireccionales, sino que, realmente son enlaces unidireccionales simétricos que crean el efecto de los enlaces bidireccionales. La diferencia puede parecer sutil, pero en la práctica es enorme. Los sistemas de redes sociales como Friendster, Orkut, y LinkedIn, que requieren el reconocimiento por parte del receptor para establecer una conexión, carecen de la misma escalabilidad, al igual que la web. Según lo observado por Caterina Fake, la co-fundadora del servicio de compartición de fotos Flickr, la atención es solo fortuitamente recíproca. (De este modo Flickr permite que los usuarios creen listas de seguimiento, cualquier usuario puede suscribirse al flujo de fotografías de cualquier otro usuario a través de RSS. Se notifica el objeto en el que se fija la atención, pero no se tiene que aprobar la conexión.)

Si una parte esencial de la Web 2.0 es el aprovechamiento de la inteligencia colectiva, convirtiendo a la web en una especie de cerebro global, la blogoesfera es el equivalente de la constante charla mental en el cerebro anterior, la voz que oímos en todas nuestras cabezas. Puede no reflejar la estructura profunda del cerebro, que es a menudo inconsciente, pero es, en cambio, el equivalente del pensamiento consciente. Y como reflejo del pensamiento consciente y de la atención, la blogoesfera ha comenzado a tener un efecto cada vez más pujante.

En primer lugar, puesto que los motores de búsqueda hacen uso de la estructura de enlaces para ayudar a predecir las páginas que son útiles, los *bloggers*, como los enlazadores más prolíficos y puntuales, tienen un papel desproporcionado en la configuración de los resultados del motor de búsqueda. En segundo lugar, puesto que la comunidad de los blogs se autoreferencia tanto, los *bloggers* que prestan la atención a otros *bloggers* magnifican su visibilidad y poder. Las críticas actúan también como amplificador.

Si fuera simplemente un amplificador, el *blogging* no tendría mayor interés. Pero al igual que Wikipedia, el *blogging* explota la inteligencia colectiva como una especie de filtro. Entra en juego lo que James Suriowecki denomina '[la sabiduría de las masas](#)'. Y al igual que

el PageRank produce resultados mejores que el análisis de cualquier documento concreto, la atención colectiva de la blogoesfera selecciona por valor.

Mientras que los medios de comunicación de masas pueden considerar algunos blogs concretos como competidores, lo realmente inquietante es que la competición es con la blogoesfera en su totalidad. Esto no es sencillamente una competición entre sitios web, sino una competición entre modelos del negocio. El mundo de la Web 2.0 es también el mundo de lo que Dan Gillmor llama '[we, the media](#)' ('nosotros, los medios de comunicación') un mundo en el cual lo que antes era simplemente la audiencia ahora decide qué es importante, y no un pequeño grupo de personas desde un cuarto trastero.

3. Los datos son el siguiente 'Intel Inside'

Hasta la fecha toda aplicación significativa de Internet ha sido desplazada por una base de datos especializada como: la base de datos de enlaces de Google, el directorio de Yahoo! (y su base de datos de enlaces), la base de datos de productos de Amazon, la base de datos de productos y vendedores de eBay, las bases de datos de mapas de MapQuest, la base de datos distribuida de canciones de Napster. Tal y como remarcó Hal Varian en una conversación privada el año pasado, el 'SQL es el nuevo HTML'. La gestión de bases de datos es una competencia básica de las compañías Web 2.0. Tanto que, en algunas ocasiones, nos hemos referido a esas aplicaciones como '[infoware](#)' en lugar de meramente software.

Este hecho conduce a una pregunta fundamental: ¿Quién es el dueño de los datos?

En la era del Internet, uno puede encontrar ciertos casos donde el control sobre la base de datos ha conducido al control del mercado y a ganancias financieras superiores a lo habitual. El monopolio en el registro de nombres de dominio concedido inicialmente por decreto del gobierno estadounidense a Network Solutions (comprada más adelante por Verisign) fue uno de los primeros grandes productos lucrativos de Internet. Aunque hemos argumentado que obtener una ventaja de negocio por medio del control del software basado en APIs es mucho más difícil en la era de Internet, el control de las fuentes clave de datos no lo es, especialmente si esas fuentes de datos son costosas de crear o favorecen el incremento de los beneficios por medio de las externalidades de red.

Eche un vistazo a los avisos de copyright en la base de cada mapa proporcionado por MapQuest, [maps.yahoo.com](#), [maps.msn.com](#) o [maps.google.com](#), y verá la frase 'Maps copyright NavTeq, TeleAtlas', o en los nuevos servicios de imágenes obtenidas vía satélite, 'Images copyright Digital Globe'. Estas compañías hicieron inversiones sustanciales en sus bases de datos (solo NavTeq ha informado de que ha invertido 750 millones de dólares para construir su base de datos de direcciones de calles y de itinerarios. Digital Globe se gastó 500 millones de dólares en lanzar su propio satélite para mejorar las imágenes proporcionadas por el gobierno). NavTeq ha llegado al extremo de imitar al conocido logotipo de *Intel Inside* de Intel: los coches con sistemas de navegación llevan la marca

'NavTeg onboard'. Los datos son, de hecho, el *Intel Inside* de estas aplicaciones, un único componente original en sistemas cuya infraestructura software es en gran medida software abierto o software que de algún otro modo se ha convertido en una *commodity*.

El ahora encendidamente competitivo escenario de los mapas web demuestra como la falta de comprensión de la importancia de poseer los datos clave de una aplicación puede, a la larga, menoscabar su competitividad. MapQuest fue el pionero en el sector de los mapas web en 1995. Aún así, cuando Yahoo!, después Microsoft, y más recientemente Google, decidieron incorporarse al mercado, pudieron ofrecer fácilmente una aplicación alternativa simplemente licenciando los mismos datos.

Como contraste, sin embargo, tomemos la posición de Amazon.com. Al igual que competidores tales como Barnesandnoble.com, su base de datos original provino del ISBN estadounidense del proveedor de registros R.R. Bowker. Pero, a diferencia de MapQuest, Amazon enriqueció incansablemente los datos, agregando datos proporcionados por la editorial tales como imágenes de la portada, la tabla de contenidos, el índice, y material de muestra. Más importante aún, sacaron provecho de sus usuarios recopilando datos, de tal forma que después de diez años, Amazon, no Bowker, es la fuente principal de datos bibliográficos sobre los libros, una fuente de referencia para eruditos y bibliotecarios, así como consumidores. Amazon también introdujo su propio identificador propietario, [ASIN](#), que equivale al ISBN cuando éste está presente, y crea una asignación de nombres equivalente para los productos sin identificador. De forma impresionante, Amazon 'abarcó y amplió' sus suministradores de datos.

Imagínese si MapQuest hubiera hecho lo mismo, aprovechando a sus usuarios para comentar los mapas y las direcciones, añadiendo capas de valor. Habría sido mucho más difícil para los competidores incorporarse al mercado simplemente licenciando los datos de partida.

La arquitectura de la participación

Algunos sistemas se diseñan para animar a la participación. En su artículo, '[The Cornucopia of the Commons](#)', Dan Bricklin señala que hay tres maneras de construir una base de datos grande. La primera, demostrado por Yahoo!, es pagar a gente para que lo haga. La segunda, inspirada en lecciones provenientes de la comunidad del software abierto, es conseguir a voluntarios para realizar esa misma tarea. El proyecto [Open Directory Project](#), un competidor de software abierto de Yahoo!, es el resultado. Pero [Napster](#) demostró una tercera manera.

Puesto que Napster se ideó para servir automáticamente cualquier música que fuera descargada, cada usuario ayudó automáticamente a construir el valor de la base de datos compartida. Esta misma aproximación ha sido seguida por el resto de servicios P2P de compartición de archivos. Una de las lecciones clave de la era de la Web 2.0 es ésta: **Los usuarios añaden valor**. Pero solamente un porcentaje pequeño de usuarios se tomará la molestia de añadir valor a su aplicación mediante el uso de medios explícitos. Por lo tanto,

las compañías Web 2.0 **usan métodos incluyentes con el fin de agregar datos del usuario y generan valor como efecto colateral del uso ordinario de la aplicación.**

Según lo observado arriba, construyen sistemas que mejoran cuanto más gente lo utiliza. Mitch Kapor una vez señaló que la 'arquitectura es política'. La participación es intrínseca a Napster, es parte de su arquitectura básica. Esta perspectiva arquitectural puede también ser más relevante para al éxito del software abierto que el frecuentemente citado llamamiento al voluntariado.

La arquitectura de Internet, y de la World Wide Web, así como los proyectos de software abierto como Linux, Apache, y Perl, es tal que los usuarios que persiguen sus propios intereses 'egoístas' generan valor colectivo como subproducto automático. Cada uno de estos proyectos tiene un núcleo pequeño, mecanismos de extensión bien definidos, y una aproximación que permite que cualquier persona agregue cualquier componente que demuestre un buen comportamiento, haciendo crecer las capas externas de lo que Larry Wall, el creador del Perl, denomina 'la cebolla' ('the onion'). En otras palabras, estas tecnologías presentan externalidades de red, simplemente por la manera en que se han diseñado.

Se puede considerar que estos proyectos tienen una arquitectura natural de participación. Pero como Amazon demuestra, mediante el esfuerzo constante (así como por incentivos económicos tales como el Programa de Asociados), es posible superponer dicha arquitectura a un sistema que no parecería poseerla en sí mismo.

La introducción reciente de Google Maps proporciona un laboratorio en vivo para la competición entre los proveedores de aplicaciones y sus suministradores de datos. El modelo de programación ligero de Google ha conducido a la creación de numerosos servicios de valor añadido en forma de lo que se conocen como *mashups* (aplicaciones web híbridas) que enlazan Google Maps con otras fuentes de datos accesibles en Internet. Por ejemplo, Housingmaps.com de Paul Rademacher, que combina Google Maps con datos de alquiler de apartamentos y compra de casas de Craigslist para crear una herramienta interactiva de búsqueda de vivienda, es el ejemplo más sobresaliente de tales *mashups*.

Por ahora, estos *mashups* son sobre todo experimentos innovadores, realizados por *hackers*. Pero los emprendedores les siguen de cerca. Y ya uno puede ver que, para al menos un tipo de desarrollador, Google le ha quitado a Navteq el papel de fuente de datos y se ha situado como intermediario preferido. Es de esperar, que en los años próximos, veamos batallas entre los suministradores de los datos y los proveedores de aplicaciones, cuando ambos se den cuenta de lo importantes que serán ciertas clases de datos que se convertirán en los componentes básicos de las aplicaciones de la Web 2.0.

La competición se centra en poseer ciertas clases de datos clave: localización, identidad, agenda de acontecimientos públicos, identificadores de producto y espacios de nombres.

En muchos casos, cuando crear los datos supone un coste significativo, puede existir una oportunidad para aplicar un juego del estilo *Intel Inside*, con una única fuente para los datos. En otros, el ganador será la compañía que alcance antes una masa crítica mediante la agregación de usuarios, y convierta esa agregación de datos en un sistema de servicios.

Por ejemplo, en el área de la identidad, PayPal, 1-click de Amazon, y los millones de usuarios de sistemas de comunicaciones, pueden ser todos ellos legítimos competidores a la hora de construir una base de datos de identidad que abarque toda la red. (En este sentido, la reciente tentativa de Google para utilizar los números de teléfono móvil como identificador para las cuentas de Gmail puede ser un paso hacia la adopción y extensión del sistema de telefonía). Mientras tanto, nuevos competidores como [Sxip](#) están explorando el potencial de la identidad federada, en busca de una especie de '1-click distribuido' que proporcione un subsistema Web 2.0 de identidad transparente. En el área de la elaboración de agendas, [EVDB](#) es un intento de construir la agenda compartida más grande del mundo por medio de una arquitectura basada en el estilo wiki de participación. Mientras que el jurado todavía está analizando el éxito de cualquier nueva empresa o aproximación particular, está claro que los estándares y las soluciones en estas áreas, convirtiendo con eficacia ciertas clases de datos en subsistemas fiables del 'sistema operativo de Internet', posibilitarán la siguiente generación de aplicaciones.

Otro punto que se debe tener en cuenta respecto a los datos es la preocupación del usuario por la privacidad y los derechos con respecto a sus propios datos. En muchas de las primeras aplicaciones web, el copyright apenas se tenía en cuenta. Por ejemplo, Amazon reclama para sí cualquier reseña que se incluye en su sitio web, pero, en ausencia de los medios para hacer cumplir tal pretensión, la gente puede incluir la misma reseña en cualquier otra parte. Sin embargo, a medida que las compañías comiencen a darse cuenta de que el control sobre datos puede ser su principal fuente de ventaja competitiva, podremos encontrar mayores intentos de control.

Al igual que el auge del software propietario provocó el movimiento [Free Software](#), podemos esperar que el auge de las bases de datos propietarias dé lugar en la próxima década a un movimiento Free Data. Uno puede apreciar indicios prematuros de esta tendencia inconformista en proyectos de datos abiertos tales como Wikipedia y Creative Commons, y en los proyectos software como [Greasemonkey](#), que permiten que los usuarios controlen cómo se muestran los datos en su ordenador.

4. El fin del ciclo de las actualizaciones de versiones del *software*

Según lo mencionado anteriormente en la discusión de Google frente a Netscape, una de las características que definen al software de la era Internet es que se entrega como un servicio, no como un producto. Este hecho conduce a un número de cambios sustanciales en el modelo de negocio de tal compañía:

- 1. Las operaciones deben convertirse en una competencia central de la compañía (core competence).** La maestría de Google o de Yahoo! en el desarrollo de productos se debe corresponder con una maestría en la gestión del día a día. La transformación del software como artefacto al software como servicio es tan sustancial **que el software dejará de funcionar a menos que se mantenga diariamente**. Google debe rastrear continuamente la web y poner al día sus índices, filtrar continuamente los enlaces de spam y otros intentos de influir en sus resultados, responder continua y dinámicamente a los centenares de millones de peticiones asíncronas de usuario, haciéndolas corresponder simultáneamente con anuncios adecuados según el contexto. No es ninguna casualidad que las técnicas de administración del sistema, la tecnología de red, y las técnicas de balanceo de carga de Google sean posiblemente secretos aún mejor guardados que sus algoritmos de búsqueda. El éxito de Google en la automatización de estos procesos es una parte clave de su ventaja en costes sobre sus competidores. No es tampoco una casualidad que los lenguajes de scripting como [Perl, Python, PHP, y ahora Ruby, desempeñen un papel tan importante](#) en las compañías Web 2.0. El Perl fue genialmente descrito por Hassan Schroeder, el primer webmaster de Sun, como 'la cinta adhesiva de Internet'. Los lenguajes dinámicos (a menudo llamados lenguajes de scripting y despreciados por los programadores de la era de los artefactos software) son la herramienta elegida por los administradores de sistemas y redes, así como por los desarrolladores de aplicaciones que implementan sistemas dinámicos que requieren cambios constantes.
- 2. Los usuarios deben ser tratados como co-desarrolladores**, considerando las prácticas de desarrollo del software abierto (incluso aunque sea poco probable que el software en cuestión sea liberado bajo una licencia abierta). De hecho, la máxima del software abierto 'libera pronto y libera frecuentemente', ha derivado hacia una posición aún más radical, 'la versión beta perpetua', en la cual se desarrolla el producto continuamente, incorporando nuevas funcionalidades mensual, semanal, o incluso diariamente. No es ninguna casualidad que servicios como por ejemplo Gmail, Google Maps, Flickr, del.icio.us y similares sigan siendo 'Beta' durante años. Así, la supervisión en tiempo real del comportamiento del usuario para detectar qué funcionalidades nuevas se utilizan, y cómo se utilizan, se convierte en otra competencia central (core competence) requerida. Un desarrollador web de un importante servicio en línea comentó: 'Ofrecemos dos o tres nuevas funcionalidades en alguna parte del sitio web cada día, y si los usuarios no las adoptan, las eliminamos. Si les gustan, las introducimos en todo el sitio web'.
Cal Henderson, el desarrollador principal de Flickr, ha revelado recientemente que ellos [despliegan nuevas implementaciones incluso hasta cada media hora](#). Esto es claramente un modelo de desarrollo radicalmente distinto. Aunque no todas las aplicaciones web se desarrollan con un estilo tan extremo como el de Flickr, casi todas las aplicaciones web tienen un ciclo de desarrollo radicalmente diferente del de cualquier cosa de la era del PC o del cliente-servidor. Esta es la razón por la que un editorial reciente de ZDnet [concluyó que Microsoft no podrá batir a Google](#): 'El

modelo del negocio de Microsoft depende de que cada dos a tres años todo el mundo actualice su entorno informático. Google depende de que todo el mundo explore cada día qué hay de nuevo en su entorno'.

Mientras que Microsoft ha demostrado una capacidad enorme de aprender y últimamente ha mejorado su competitividad, está claro que, esta vez, la competencia requerirá que Microsoft (y por extensión, cualquier otra compañía de software existente) se convierta en un tipo de compañía completamente distinta. Las compañías nativas Web 2.0 gozan de una ventaja natural, ya que no tienen viejos patrones (y los correspondientes

Una Tesis de Inversión Web 2.0

Paul Kedrosky, inversionista que se dedica a prestar capital de riesgo, [escribe](#): 'la clave es encontrar las inversiones susceptibles de ser denunciadas donde usted disienta del consenso'. Es interesante ver cómo cada faceta de la Web 2.0 implica disentir: todo el mundo hacía énfasis en mantener los datos privados, Flickr, Napster *et al* los hacen públicos. No se trata precisamente de discrepar para ser desagradable, es discrepar cuando se puede construir algo a partir de las diferencias. Flickr construye comunidades, Napster construyó variedad de colecciones.

Otra forma de verlo es que todas las compañías de éxito renuncian a algo costoso pero considerado crítico, para conseguir gratis algo valioso que antes era caro. Por ejemplo, Wikipedia renuncia al control editorial central para obtener velocidad y amplitud. Napster renunció a la idea del 'catálogo' (todas las canciones que el proveedor vendía) y consiguió variedad y amplitud. Amazon renunció a la idea del tener una tienda física, pero consiguió dar servicio al mundo entero. Google renunció (inicialmente) a los clientes grandes y consiguió el 80% de los que no tenían sus necesidades cubiertas. Hay algo muy del aikido (que usa la propia fuerza del oponente en contra suya) en decir 'sabe, usted tiene razón. Absolutamente cualquier persona en el mundo entero puede actualizar este artículo. Y adivine, eso son malas noticias para usted'.

modelos del negocio y fuentes de ingresos) de los que deshacerse.

Nat Torkington

5. Modelos de Programación Ligeros

Una vez que la idea de los *web services* se convirtió en habitual, las grandes compañías se lanzaron a la lucha con una compleja pila de *web services* diseñada para crear entornos de programación altamente fiables para aplicaciones distribuidas.

Al igual que la web tuvo éxito precisamente porque derrocó muchas de las teorías del hipertexto, sustituyendo un pragmatismo simple por un diseño ideal, RSS se ha convertido

quizás en el *web service* más ampliamente desplegado debido a su simplicidad, mientras que las pilas de complejos *web services* corporativos todavía tienen que lograr un amplio despliegue.

Análogamente, los *web services* de Amazon.com se proporcionan de dos formas: una que se adhiere a los formalismos de la pila de *web services* SOAP (Simple Object Access Protocol), la otra que simplemente proporciona datos XML sobre HTTP, se trata de un enfoque ligero denominado a veces REST (Representational State Transfer). Mientras que las conexiones de alto valor B2B (como las que hay entre Amazon y socios de venta al por menor tales como ToysRUs) utilizan la pila SOAP, Amazon informa de que el 95% del uso corresponde al servicio ligero de tipo REST.

Esta misma búsqueda de la simplicidad se puede encontrar en otros *web services* 'orgánicos'. El reciente lanzamiento de Google, Google Maps, es un ejemplo. La interfaz simple AJAX (Javascript y XML) de Google Maps fue descifrada rápidamente por los *hackers*, los cuales procedieron a remezclar los datos para dar lugar a nuevos servicios.

Durante algún tiempo, proveedores GIS como ESRI, al igual que MapQuest y Microsoft MapPoint, han proporcionado *web services* relacionados con mapas. Pero Google Maps ha causado una revolución debido a su simplicidad. Mientras que experimentar con cualquiera de los *web services* formales proporcionados por un proveedor requería un contrato formal entre las partes, debido a la forma en la que Google Maps se implementó, los datos eran algo muy fácil de conseguir, y los *hackers* pronto encontraron maneras creativas de reutilizar esos datos.

De todo esto se desprenden varias lecciones significativas:

1. Apoye los modelos de programación ligeros que permiten sistemas débilmente acoplados . La complejidad de la pila de los *web services* apoyados por las corporaciones se diseña para permitir un fuerte acoplamiento entre las partes. Aunque esto es necesario en muchos casos, muchas de las aplicaciones más interesantes pueden ciertamente continuar siendo débilmente acopladas, e incluso frágiles. ¡La mentalidad de la Web 2.0 es muy diferente de la mentalidad tradicional de las TI (Tecnologías de la Información)!

2. Piense en sindicación, no en coordinación . Los *web services* simples, como RSS y aquellos *web services* basados en REST, tratan de la sindicación de datos hacia el exterior, sin controlar lo que sucede cuando llegan al otro extremo de la conexión. Esta idea es inherente a Internet en sí misma, un reflejo de lo que se conoce como el [principio extremo a extremo\(end-to-end\)](#).

3. Diseñar para ser 'hackeados' y 'remezclados' . Los sistemas como la web original, RSS, y AJAX tienen algo en común: las barreras para su reutilización son extremadamente bajas. Gran parte del software útil es realmente software abierto, pero incluso cuando no lo es, hay poca protección de la propiedad intelectual. La opción del navegador web 'Ver código

fuente' ('*View Source*') hizo posible que cualquier usuario copiara la página web de cualquier otro usuario; RSS fue diseñado para permitir al usuario visualizar el contenido que quiera y cuando lo desee, no según lo que dicte el proveedor de información; los *web services* más exitosos son los que han sido más fáciles de llevar en nuevas direcciones inimaginables para sus creadores. La frase 'algunos derechos reservados' ('*some rights reserved*') que fue popularizada por Creative Commons como contraste a la frase más típica 'todos los derechos reservados' ('*all rights reserved*'), es un indicador útil.

Innovación en el Ensamblado

Los modelos de negocio ligeros son concomitantes con la programación ligera y las conexiones ligeras. La disposición mental de la Web 2.0 es buena para la reutilización. Un servicio nuevo como Housingmaps.com se construyó simplemente ensamblando juntos dos servicios existentes. Housingmaps.com no tiene un modelo del negocio (todavía), pero para muchos servicios de pequeña escala, Google AdSense (o quizás los honorarios de los asociados de Amazon, o ambos) proporciona el equivalente de un modelo de ingresos.

Estos ejemplos proporcionan una idea de otro principio clave de la Web 2.0, que llamamos 'innovación en el ensamblado'. Cuando los componentes *commodity* son abundantes, se puede crear valor simplemente ensamblándolos de forma novedosa o eficaz. Al igual que la revolución del PC proporcionó muchas oportunidades para la innovación en el ensamblado del hardware de tipo *commodity* (con compañías como Dell, la cual hizo de dicho ensamblado una ciencia, derrotando consecuentemente a compañías cuyo modelo de negocio requería de innovación en el desarrollo del producto), creemos que la Web 2.0 ofrecerá oportunidades a las compañías para superar a la competencia siendo mejores en el aprovechamiento y la integración de los servicios proporcionados por otros.

6. El software no limitado a un solo dispositivo

Otra característica de la Web 2.0 que merece mencionarse es el hecho de que ya no se limita a la plataforma PC. Dave Stutz, que durante mucho tiempo fue desarrollador de Microsoft, en su recomendación de despedida de Microsoft señaló que 'el [software útil escrito sin circunscribirse a un solo dispositivo](#) proporcionará altos márgenes durante mucho tiempo'.

Por supuesto, cualquier aplicación web se puede considerar *software* no limitado a un solo dispositivo. Después de todo, incluso la aplicación web más simple implica por lo menos a dos ordenadores: el que hospeda el servidor web y el que alberga el navegador. Y, como hemos comentado, el desarrollo de la web como plataforma extiende esta idea a aplicaciones sintéticas compuestas de servicios proporcionados por múltiples ordenadores.

Pero como en muchas áreas de la Web 2.0, donde el ser 2.0 no es algo nuevo, sino más bien una realización más completa del verdadero potencial de la plataforma web, esta

frase nos da una idea clave de cómo diseñar las aplicaciones y los servicios para la nueva plataforma.

Hasta la fecha, iTunes es el mejor ejemplo de este principio. De un modo transparente, esta aplicación va desde un dispositivo portátil a un *back-end* web masivo, con el PC actuando como caché local y estación de control. Ha habido muchos intentos anteriores de llevar el contenido de la web a dispositivos portátiles, pero la combinación de iPod/iTunes es una de las primeras aplicaciones diseñadas partiendo de cero para abarcar múltiples dispositivos. TiVo es otro buen ejemplo.

iTunes y TiVo también presentan muchos de los otros principios básicos de la Web 2.0. No son aplicaciones web en sí mismas, sino que hacen uso del poder de la plataforma web, haciendo de ella una parte integrada total y transparentemente. La gestión de datos es claramente la parte más importante de su oferta. Son servicios, no aplicaciones empaquetadas (aunque en el caso de iTunes, puede ser utilizado como una aplicación empaquetada, gestionando solamente los datos locales del usuario). Lo que es más, tanto TiVo como iTunes muestran un cierto uso incipiente de la inteligencia colectiva, aunque en ambos casos, sus experimentos están en guerra con el grupo de presión (*lobby*) IP. En iTunes hay sólo una arquitectura de participación limitada), aunque la reciente adición del [podcasting](#) cambia esa ecuación sustancialmente.

Ésta es una de las áreas de la Web 2.0 donde esperamos ver alguno de los mayores cambios, a medida que más y más dispositivos se conecten a la nueva plataforma. ¿Qué aplicaciones se hacen posibles cuando nuestros teléfonos y nuestros coches no consumen datos sino que los proporcionan? La supervisión del tráfico en tiempo real, los [flash mobs](#) (grupos de gente que se reúnen de improviso en un sitio público para llevar a cabo un acto inusual o significativo y luego se dispersan), y el periodismo ciudadano son solamente algunas de las primeras señales de aviso acerca de las capacidades de la nueva plataforma.

7. Experiencias Enriquecedoras Del Usuario

Ya en 1992 con el [navegador Viola](#) de Pei Wei, la web era utilizada para descargar '*applets*' y otras clases de contenido activo dentro del navegador web. La introducción de Java en 1995 estuvo enmarcada en la descarga de tales *applets*. Javascript y después DHTML se introdujeron como formas ligeras de proporcionar programabilidad en el lado cliente y experiencias más enriquecedoras para el usuario. Hace varios años, Macromedia acuñó el término 'Rich Internet Applications' (que también ha sido adoptado por el competidor de Flash de software abierto Laszlo Systems) con el fin de destacar las capacidades del Flash para proporcionar no sólo contenido multimedia sino también experiencias de aplicación del estilo GUI.

Sin embargo, el potencial de la web de proporcionar una gama completa de aplicaciones no tuvo un éxito masivo hasta que Google introdujo Gmail, seguida rápidamente por

Google Maps, aplicaciones web con interfaces de usuario ricas y con interactividad equivalente a la del PC. El conjunto de tecnologías usadas por Google fue [bautizado como AJAX](#), en un ensayo de gran influencia de Jesse James Garrett, de la empresa de diseño web Adaptive Path. El cual escribió:

'Ajax no es una tecnología. Es realmente varias tecnologías, cada una floreciendo con derecho propio, agrupándose en nuevas formas de gran poder. Ajax incorpora:

- [presentación basada en estándares](#) usando XHTML y CSS;
- visualización e interacción dinámicas usando el modelo [Document Object Model](#);
- intercambio y manipulación de datos usando [XML y XSLT](#);
- recuperación de datos asíncrona usando [XMLHttpRequest](#);
- y [JavaScript](#) aglutinándolo todo junto.

AJAX es también un componente clave de aplicaciones Web 2.0 tales como Flickr, ahora parte de Yahoo!, las aplicaciones Basecamp y Backpack de 37signals, así como de otras aplicaciones de Google como Gmail y Orkut. Nos estamos adentrando en un período sin precedentes en cuanto a la innovación en el campo de las interfaces de usuario, ya que los desarrolladores web pueden finalmente construir aplicaciones web con interfaces de usuario tan ricas como las de las aplicaciones locales de PC.

Curiosamente, muchas de las capacidades que están siendo exploradas en este momento existen desde hace muchos años. A finales de los 90, Microsoft y Netscape tenían una visión acerca del tipo de capacidades que finalmente ahora se están desarrollando, pero su batalla sobre los estándares que se debían utilizar hizo difícil la existencia de aplicaciones que funcionaran en múltiples navegadores. Sólo cuando Microsoft ganó definitivamente la guerra de los navegadores, y sólo existía un navegador estándar de facto para el que escribir, este tipo de aplicaciones llegaron a ser posibles. Aunque [Firefox](#) ha vuelto a introducir la competencia en el mercado de los navegadores, al menos hasta el momento no hemos visto la destructiva competencia sobre los estándares web que retrasó el progreso en los 90.

Esperamos ver muchas nuevas aplicaciones web en los próximos años, tanto aplicaciones verdaderamente novedosas, como ricas reimplementaciones web de las aplicaciones para PC. Hasta el momento, cada cambio de plataforma ha generado también oportunidades para que se efectúe un cambio en el liderazgo de las principales aplicaciones de la plataforma anterior.

Gmail ha proporcionado ya [algunas innovaciones interesantes en el correo electrónico](#), combinando las fortalezas de la web (accesible desde cualquier parte, enormes capacidades de base de datos, posibilidades de búsqueda) con las fortalezas de las interfaces de usuario que se aproximan a las interfaces del PC en usabilidad. Mientras tanto, otros clientes de correo en la plataforma PC están mordisqueando el problema desde el otro extremo, agregando mensajería instantánea y capacidades de presencia

(*presence*). ¿Cómo de lejos estamos de un cliente de comunicaciones integrado que combine lo mejor del correo electrónico, la mensajería instantánea y el teléfono móvil, usando [VoIP](#) para agregar servicios de voz con el fin de enriquecer las capacidades de las aplicaciones web? La competición está en marcha.

Es fácil ver cómo la Web 2.0 también transformará la libreta de direcciones del correo electrónico (*address book*). Una libreta de direcciones Web 2.0 considerará la libreta de direcciones local en el PC o en el teléfono simplemente como una caché de los contactos que usted ha pedido explícitamente que el sistema recuerde. Mientras que, un agente web de sincronización, del estilo de Gmail, recordaría cada mensaje enviado o recibido, cada dirección de correo y cada número de teléfono usados, y ejecutaría algoritmos heurísticos de relaciones sociales para decidir cuáles debe ofrecer como alternativas cuando no se ha encontrado una respuesta en la caché local. Si la caché careciera de una respuesta, el sistema preguntaría a una red social más amplia.

Un procesador de texto de la Web 2.0 soportaría la edición al estilo colaborativo de las wikis, no sólo documentos autónomos. Pero también soportaría las ricas posibilidades de formato que esperamos de los procesadores de textos para PC. [Writely](#) es un buen ejemplo de este tipo de aplicaciones, aunque todavía no ha logrado tener gran fuerza en el mercado.

La revolución de la Web 2.0 no se limitará a las aplicaciones para PC. Salesforce.com demuestra cómo la web se puede utilizar para ofrecer software como servicio, en aplicaciones de escala empresarial como el CRM.

La oportunidad competitiva para los nuevos competidores es abrazar por completo el potencial de la Web 2.0. Las compañías que tengan éxito crearán aplicaciones que aprendan de sus usuarios, usando una arquitectura de participación para construir una abrumadora ventaja derivada no sólo de la interfaz software, sino de la riqueza de los datos compartidos.

Competencias Centrales de las Compañías Web 2.0

Al explorar los siete principios descritos anteriormente, hemos destacado algunas de las características principales de la Web 2.0. Cada uno de los ejemplos que hemos explorado ilustran uno o más de esos principios clave, pero les pueden faltar otros. Finalicemos, por tanto, resumiendo lo que creemos serán las competencias clave (*core competencies*) de las compañías Web 2.0:

- Servicios, no software empaquetado, con escalabilidad rentable.
- Control sobre fuentes de datos únicos y difíciles de replicar que se enriquezcan a medida que más gente las utilice.
- Confiar en los usuarios como co-desarrolladores.
- Aprovechar la inteligencia colectiva.

- Sacar partido de la larga cola (the long tail) mediante el autoservicio del cliente.
- Software no limitado a un solo dispositivo.
- Interfaces de usuario, modelos de desarrollo Y modelos de negocio ligeros.

La próxima vez que una compañía asegure que es 'Web 2.0', compare sus características con esta lista. Cuantos más puntos se anoten, más dignos son del nombre. Sin embargo, recuerde que la excelencia en un área puede ser más efectiva que algunos pequeños pasos en las siete.

Referencias:

1. Artículo original:
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
2. Traducción al español:
<http://sociedaddelainformacion.telefonica.es/jsp/articulos/detalle.jsp?elem=2146>