



**¿QUE ES LA WEB 2.0?**

**MILEIVY PAOLA TOBINSON TORRES**

**SUGEY ELENA PÁEZ SALEMI**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

**FACULTAD DE INGENIERÍA**

**PROGRAMA DE INGENIERÍA DE SISTEMAS**

**CARTAGENA DE INDIAS D. T. Y C.**

**2008**



**¿QUE ES LA WEB 2.0?**

**MILEIVY PAOLA TOBINSON TORRES**

**SUGEY ELENA PÁEZ SALEMI**

**Monografía presentada para optar el título de Ingeniero de Sistemas**

**Director de Monografía:**

**EDWIN PUERTA**

**Ingeniero de Sistemas**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

**FACULTAD DE INGENIERÍA**

**PROGRAMA DE INGENIERÍA DE SISTEMAS**

**CARTAGENA DE INDIAS D. T. Y C.**

**2008**

### RESUMEN

La Web 2.0 es el conjunto de avances modernos en la Web tradicional, refiriéndose con esto a un grupo de técnicas y tecnologías que han llevado a formar una cultura colaborativa y comunitaria cuyo principal objetivo es compartir información online. Todo esto se hace posible acompañado de diversas facilidades como lo son la flexibilidad y rapidez que ofrece la Web 2.0 con respecto a la versión que antecede a esta, que era una Web estática y solo permitía la lectura en la Web, mientras que en la de hoy día, interactiva, se logra un paradigma de lecto-escritura. Son muchas las ventajas que envuelve el uso de la Web 2.0 y aplicabilidades en las empresas que siempre están a la vanguardia de las nuevas tecnologías. Bajos costos y actualizaciones en las versiones de las aplicaciones Web es lo que mas atrae usuarios, así como el compartimiento de la información dejando de lado factores como espacio y tiempo.

### LISTA DE TABLAS Y FIGURAS

Tabla 1: Evolución Web 1.0 a 2.0 .....	10
Tabla 2: Evolución de la Web.....	11
Tabla 3: Ventajas y Aplicabilidades de las Tecnologías.....	46
Tabla 4: Ventajas y Aplicabilidades de las Técnicas.....	49
Tabla 5: Ventajas y Aplicabilidades de los estándares.....	50
Figura1: Paradigma tradicional-paradigma emergente.....	8
Figura 2: Web1.0 a Web 2.0.....	9
Figura 3: De Web de Escritura a Web de Lecto-Escritura.....	12
Figura 4. Modelo síncrono - modelo asíncrono.....	45
Figura 5. Modelo clásico - modelo Ajax .....	46
Figura 6. Tipos de redes P2P .....	41
Figura 7: Apis DOM .....	110
Figura 8: Transformaciones en ficheros de XML .....	118

## TABLA DE CONTENIDO

LISTA DE TABLAS Y FIGURAS.....	4
OBJETIVO.....	7
INTRODUCCIÓN.....	8
1. Marco Teórico .....	9
1.1 Antecedentes .....	9
2. Web 2.0 .....	16
2.1 Tecnologías .....	23
2.2 Técnicas .....	32
2.3 Arquitectura .....	38
2.4 Estándares .....	43
2.5 Modelos .....	48
2.6 Ventajas y aplicabilidades.....	52
2.6.1 Ventajas y aplicabilidades de las técnicas.....	52
2.6.2 Ventajas y aplicabilidades de las tecnologías .....	54
2.6.3 Ventajas y aplicabilidades de los estándares .....	55
2.7 Aplicación de las principales técnicas, tecnologías y estándares ...	56
2.7.1 Aplicación de las técnicas .....	57
2.7.2 Aplicación de las tecnologías .....	75
2.7.3 Aplicación de los estándares .....	100
3. ¿Qué es la Web 2.0? .....	118
Conclusiones.....	127
Bibliografía.....	128

## OBJETIVO

### General

Dar a conocer los principales conceptos y aplicabilidades de la Web 2.0 a través de técnicas y tecnologías diseñadas e implementadas para brindar una mejor calidad de servicio en el intercambio de información.

### Específicos

- Jerarquizar las técnicas que se utilizan para llevar a cabo aplicabilidades a la Web 2.0
- Destacar las tecnologías aplicadas a implementaciones de la Web 2.0
- Enumerar las ventajas principales de la Web 2.0
- Interpretar los principios constitutivos de las aplicaciones que caracterizan a la Web 2.0
- Reconocer los principales estándares tecnológicos sobre los que trabaja la Web 2.0
- Identificar modelos de programación implementados en desarrollo de la Web 2.0
- Asimilar la arquitectura sobre la que se construye la Web 2.0

### INTRODUCCIÓN

Cuando se escucha myspace.com, facebook.com, y páginas parecidas a estas lo primero que se piensa es lo grandiosa que es tener una propia página Web actualizada y a la que todos pueden acceder y con la cual me puedo comunicar y compartir información con conocidos y aun con gente de quien antes no he oído hablar, sin importar las distancias. Esto es muy diferente a la Web de hace algunos años en la que solo se podía hacer consultas de corporaciones comerciales y el consumo de esta era reducido a una población muy pequeña.

La Web en sus principios comenzó como un medio de comunicación expositorio, en el cual el objetivo principal era comercial, donde las empresas ofrecían servicios al usuario, pero este no podía interactuar con la página. Este concepto nombrado Web 1.0 ha evolucionado con la ayuda de nuevas tecnologías y técnicas que han surgido, las cuales han llevado a la Web a convertirse en un medio de comunicación masivo y colaborativo, donde el usuario puede interactuar a través de la Web, dirigiéndose a la creación de una nueva cultura, impactando en varios campos como el periodismo, la educación, los negocios, etc. Que han venido a formarse como consecuencia de las redes sociales, software gratuito, open source, y varios modelos de negocio.

## 1. MARCO TEORICO

### 1.1 ANTECEDENTES

La Web 2.0 no es un estándar de Internet o una nueva tecnología particular. Es un concepto algo amplio, que indica el uso de las tecnologías actuales en la red. Su inventor, O'Reilly<sup>[r01]</sup>, evocó en octubre de 2004 una transición de Internet, incluso un renacimiento. Hasta la Web 2.0, Internet estuvo marcada por dos etapas:

- La Web 1.0 corresponde hasta finales de los años 1990 con la instalación de las redes y a la adopción de los estándares de la comunicación.
- La Web 1.5 marca la llegada de un Internet comercial, era principios del año 2000, época conocida como el estallido de la “burbuja”.

La Web 2.0, presenta la evolución de las aplicaciones tradicionales hacia aplicaciones enfocadas al usuario final, que al fin y al cabo, es el principal protagonista. ¿Qué deparará la Web 2.5?, es a este concepto al que se dirige la Web actual.

La Web 2.0 es la nueva “moda” de Internet. Se podría decir que es algo que se está extendiendo como la espuma.

Este tipo de Web es el futuro de Internet, porque poco a poco todas las páginas han ido evolucionando en este sentido. De hecho páginas como *Marca* y otras páginas de renombre que ya no solo se dedican a publicar para ser vistas, sino que permiten al usuario interactuar permitiendo dejar comentarios, votar en encuestas,... en definitiva, cosas que antes estaban reservadas para foros y blogs.

### **La Web 2.0: De Sus Conocedores A Sus Hacedores**

En los tiempos de Twitter (servicio de redes sociales y MicroBlogging que permite a sus usuarios enviar mensajes de sólo texto, con una longitud máxima de 140 caracteres, vía SMS, mensajería instantánea) y de las “redes sociales”, hablar de la Web 2.0 atrasa. Hoy los “nombres sagrados” de la Web



## ¿Qué es la Web 2.0?

---

son: *hacer, aprender haciendo, conocimiento ligado a la acción, y sobre todo inteligencia colectiva o producción entre pares.*

La breve historia de la Web 2.0 tiene sus mitos fundacionales, sus relatos y sus metáforas. Una de las metáforas que está en el origen de los blogs es la de “conversación colectiva”. Existen más de 70 millones de blogs en el mundo e innumerables textos que empezaron siendo comentarios o notas al pie en otros blogs y que luego tomaron el cuerpo del texto.

Esto lleva a la inferencia de que la Web 2.0 representa algo más que nuevas tecnologías, simboliza una nueva cultura que se avecina con la evolución de la Web y el Internet.

Los sistemas de *tagging social* o de etiquetado colaborativo de la información, en los que se basan aplicaciones como del.icio.us y Flickr, y la distribución de contenidos por RSS asociada a los blogs, son una parte fundamental de la lógica de funcionamiento de los nuevos medios, centrada en los usuarios y en sus comunidades. Se podría incluir bajo esta misma lógica otras herramientas como Feevy y Twitter, que operan también como indicadores de lo que vendrá: movilidad y ubicuidad de los nuevos dispositivos de comunicación, conexión permanente y “redes sociales”.

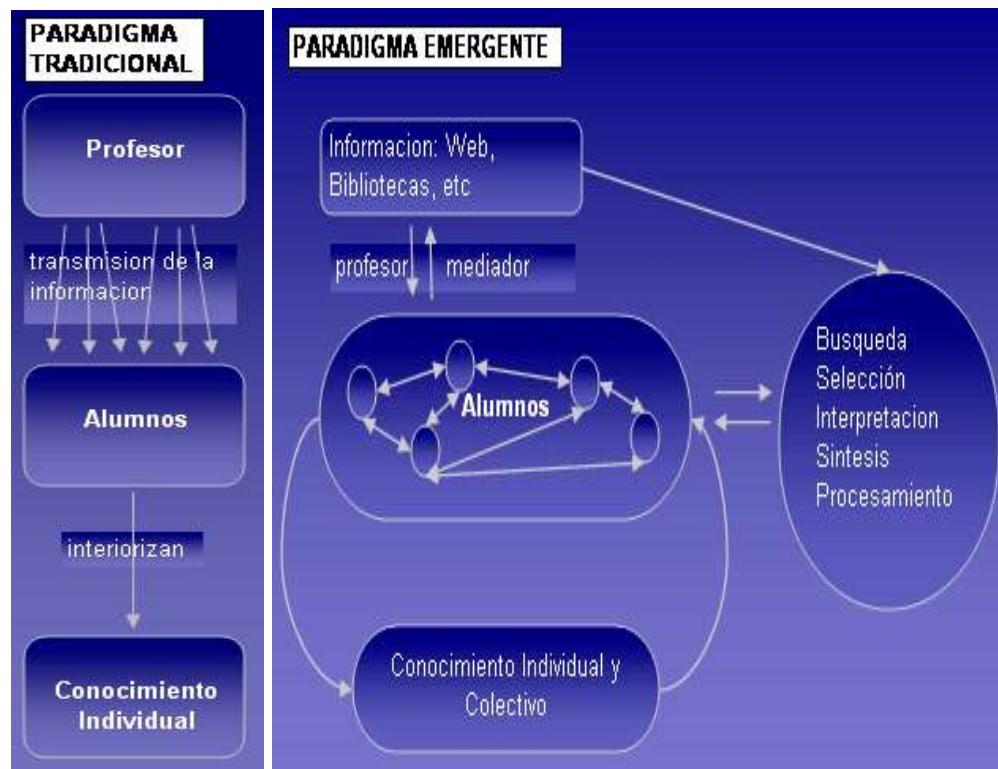
El uso de estas herramientas se encamina así hacia un nuevo modelo basado en agregar, distribuir y compartir. Más aún, el gran potencial de transformación a partir del vínculo entre las tendencias sociales de la humanidad y la eficacia de las tecnologías consiste en la posibilidad de hacer nuevas cosas al tiempo, de cooperar en escalas y modos que antes no eran posibles.

¿Surgirán nuevas formas de reflexividad a partir del uso de estos nuevos medios? En este nuevo orden de la información, de la velocidad y la inmanencia, surge también la pregunta por los modos en que se modifican nuestras formas de pensar y de conocer, ya no *a pesar de* todo ello sino *con* ello. Esta nueva reflexividad se vuelve *comunicacional, dialógica*, y pone en el centro de atención a la producción entre pares. Aparece ante todo como un tipo de “saber” inseparable del “hacer”.

## ¿Qué es la Web 2.0?

A continuación se muestran gráficas donde se ilustra la transición percibida en Internet desde las Webs tradicionales a aplicaciones Web destinadas a usuarios. La expectativa para la Web 2.0 es que los servicios de esta sustituyan a las aplicaciones de escritorio en muchos usos.

### La Web 2.0 y Su Impacto en la Educación



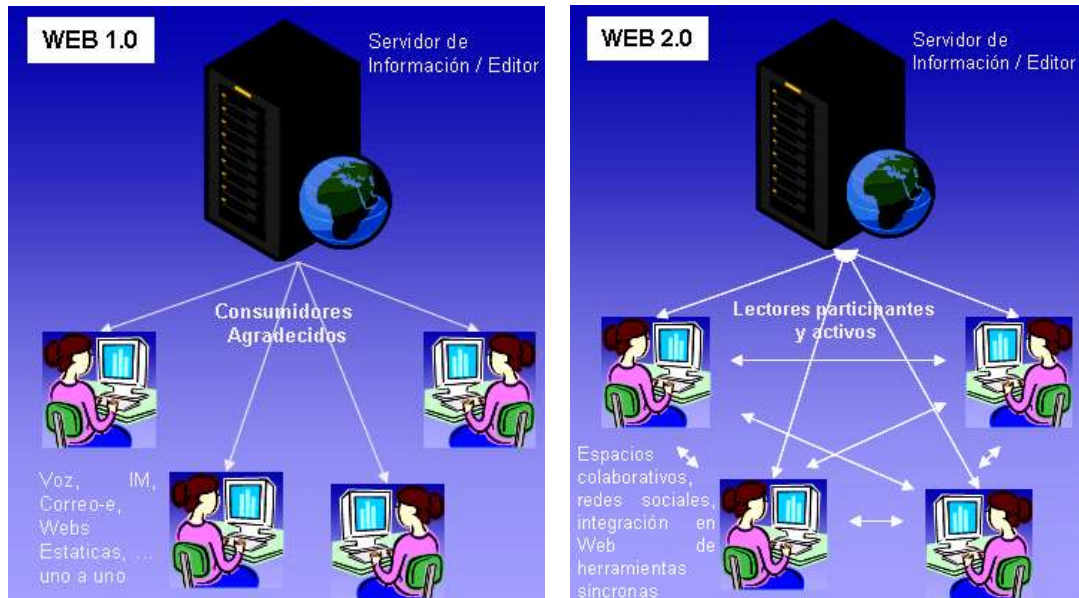
**Figura1. Paradigma tradicional-paradigma emergente. Tomado de Seminario Internacional Virtual Educa Cono Sur 2006.**

En la parte de la educación se ha visto la evolución de la Web 1.0 a la 2.0, como lo muestra la anterior gráfica. En el paradigma tradicional, el profesor transmite el conocimiento y los alumnos interiorizan la información generándose así un conocimiento individual; mientras que con el paradigma emergente se brinda la posibilidad de que el alumno busque, seleccione, interprete, sintetice y procese la información que encuentra en bibliotecas online, entre otros recursos, compartiéndola así con demás alumnos y obteniendo un conocimiento colectivo, además del individual. En el modo tradicional el profesor es la única fuente de información, mientras que en el

## ¿Qué es la Web 2.0?

emergente este se convierte en un puente entre la fuente de información y el alumno.

### De Web 1.0 a Web 2.0



**Figura 2. Web1.0 a Web 2.0. Tomado de Seminario Internacional Virtual Educa Cono Sur 2006.**

En la anterior gráfica se puede observar como en la versión de la Web 1.0 la relación entre el servidor de información y los consumidores es UNILATERAL, es decir, el servidor le brinda la información al usuario y estos simplemente la toman y agradecen por tal servicio, ya sea de voz, de correo, entre otros. En la Web 2.0 es diferente la situación, los usuarios son participativos y activos, interactúan con el servidor así como con los otros usuarios de la misma red por medio de espacios colaborativos, se presentan las redes sociales que facilitan a los consumidores tener una relación colaborativa entre ellos.

## ¿Qué es la Web 2.0?

---

### Evolución de la Web 1.0 a la 2.0

Web 1.0	Web 2.0
Doble clic	Google AdSense
Ofoto	Flickr
Akami	BitTorrent
Mp3.com	Napster
Britanica Online	Wikipedia
Sitios Webs Personales	Blogging
Evite	Upcoming.org y EVDB
Especulación con nombres de dominio	Optimización del motor de búsqueda
Page view	Coste por clic
Screen scraping	Web services
Publicación	Participación
Sistemas de gestión de contenidos	Wikis
Directorios (taxonomía)	Marcar con etiquetas (folksonomy)
Adherencia	Sindicación de contenidos

**Tabla 1. Evolución Web 1.0 a 2.0. Tomado de *Seminario Internacional Virtual Educa Cono Sur 2006*.**

En la gráfica anterior se muestra la evolución de la Web. 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 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.

## ¿Qué es la Web 2.0?

### Evolución de la Web

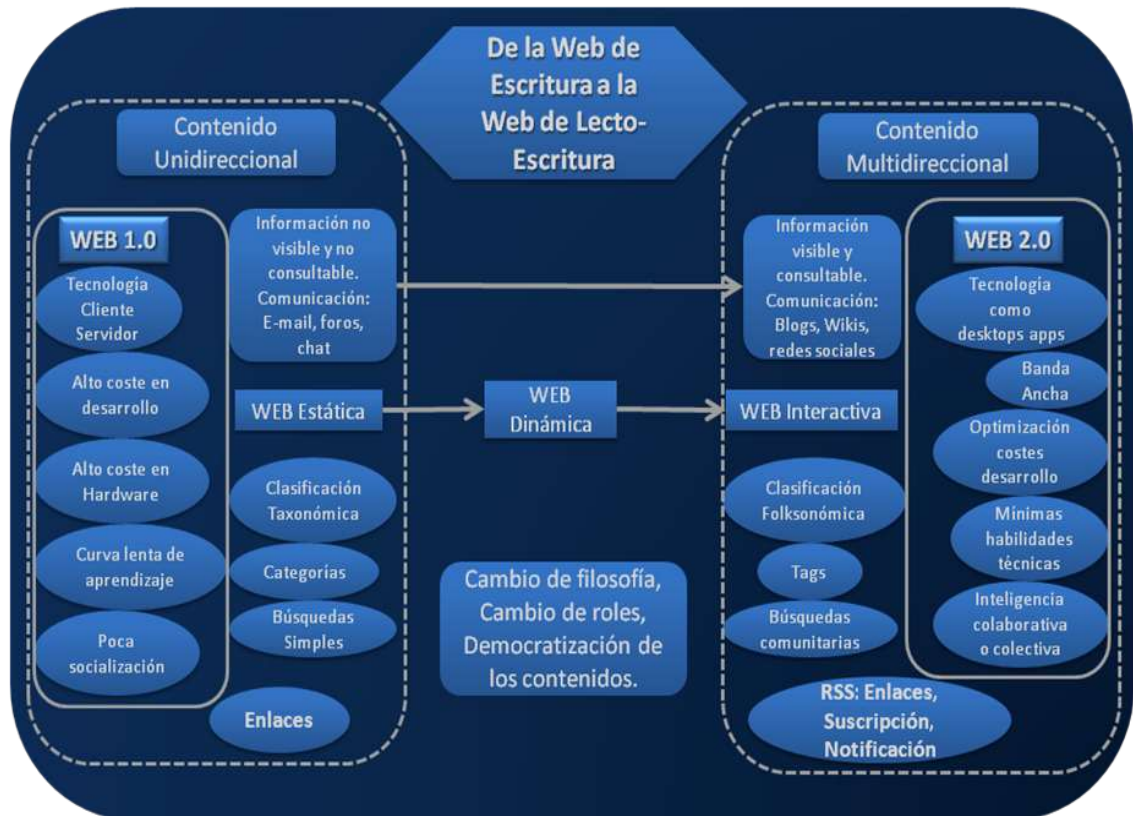
A continuación un gráfico que resume la evolución de la Web 1.0 a la 2.0 pasando por la 1.5:

CARACTERÍSTICAS	Web 1.0 (1994 - 1997)	Web 1.5 (1997 - 2003)	Web 2.0 (2003 - )
Tipo de Web	Estática	Dinámica	Colaborativa
Tecnologías asociadas	HTML, .gif	DHTML, Perl, php, asp, etc	Ajax, DOM, Soap, XML, Ruby
Arquitectura	Cliente-Servidor	Cliente-Servidor con	Servicios Web/desktops Apps
Mezclas	Es un bloque único de información /	Integración de scripts	Integrabilidad de elemento de distintas Web:
Acceso a la Información	Página a página: favoritos	Push. Microsoft Active Channel	Pull, Sindicación RSS/ATOM. Podcasting
Desarrollo	HTML	HTML/CSS con lenguajes de programación	XHTML/CSS, módulos, otros: Ajax, DOM
Intervención del usuario	Lectura	Lectura (comentarios)	Lectura/escritura publicación

**Tabla 2. Evolución de la Web. Tomado de *Seminario Internacional Virtual Educa Cono Sur 2006*.**

En la tabla anterior se muestran las características de la Web 1.0, 1.5 y la 2.0 y cómo ha evolucionado desde 1994. El tipo de Web en sus inicios era estática y ahora es colaborativa, pero antes tuvo que ser dinámica, se aprecia también las tecnologías asociadas, la arquitectura, las mezclas, el acceso a la información, el desarrollo y la intervención del usuario cómo han cambiado volviéndose complejas pero buscando la mejor interacción entre los usuarios finales ofreciéndoles así un servicio de la mejor calidad.

### Hacia una nueva generación Web



**Figura 3. De Web de Escritura a Web de Lecto-Escritura. Tomado de Seminario Internacional Virtual Educa Cono Sur 2006.**

En la anterior grafica se muestra cómo ha evolucionado la Web 1.0 a la 2.0, la Web dejó de ser estática para ser dinámica y por último interactiva que es la Web que se usa actualmente. En la Web 1.0 la comunicación era solo por correo electrónico, foros o chats; en la 2.0 aparecen también los Blogs, wikis y las redes sociales. Los costos de desarrollo de aplicaciones se optimizan, se socializa el conocimiento dejando atrás la curva lenta de aprendizaje para dar paso a la inteligencia colectiva. La Web 1.0 era solo de escritura, era un paradigma unidireccional mientras que la 2.0 es lecto-escritura haciendo que se convierta en un paradigma multidireccional. La filosofía de la Web cambia, cambian los roles de los usuarios, antes estos solo podían leer lo que en la Web estaba publicado, ahora pueden aportar y opinar y los contenidos allí expuestos se democratizan, enriqueciendo el conocimiento colectivo de quienes participan en ella.

## **2. WEB 2.0**

La Web 2.0 es una Web primordialmente colaborativa, la cual se ha ubicado en un periodo de tiempo desde el año 2003 hasta hoy, las tecnologías que utiliza son Ajax, DHTML, XML, Soap, las cuales llevan al usuario a convertirse en contribuidores, publicando informaciones y realizando cambios en los datos.

Algunas de las ventajas de la Web 2.0 son:

- Simplifica la usabilidad del sitio Web
- Ahorra tiempo al usuario
- Estandariza los lenguajes para una mejor utilización de la re-utilización del código.
- Permite una mejor interoperabilidad entre aplicaciones, y entre las aplicaciones y las máquinas (software-hardware).
- Facilita las interacciones
- Facilita el reconocimiento o detección de carencias o nuevas formas de utilización de aplicaciones.
- Facilita la convergencia entre los medios de comunicación y los contenidos.
- Facilita la publicación, la investigación y la consulta de contenidos Web.
- Estimula y aprovecha la inteligencia colectiva en beneficio de internet.

Las anteriores se derivan de la transformación que ha tenido la Web en los últimos años y algunas de estas están planteadas en el mencionado artículo original de O'Reilly<sup>[r02]</sup>, como las principales características, que se muestra a continuación como *los siete principios constitutivos de las aplicaciones Web 2.0*<sup>[r02]</sup>.

### **Siete principios constitutivos de las aplicaciones Web 2.0.**

#### **1. La World Wide Web como plataforma.**

## ¿Qué es la Web 2.0?

---

Salvo *Linux* y otras escasas alternativas, hasta hoy la industria del software computacional se enfrentaba casi con exclusividad a un modelo de negocio de paquetes con derechos propietarios y venta bajo el régimen de obsolescencia planificada. Empresas como *Adobe* (propietaria de *Photoshop*, *Flash Macromedia*, *Dreamweaver*, *Freehand*, etc), *Microsoft* (Windows Vista), o *Apple* (*iLife*, *iTunes*, *Mac OS X Tiger*) se rigen bajo ese esquema. Para que los diseñadores, empresas o instituciones puedan utilizar dichas aplicaciones en forma legal, deben comprar los derechos de uso y actualizar el software cada cierto tiempo, con los consiguientes costos adicionales por la licencia. Nada indica que en el corto plazo dicho modelo de comercialización se modifique, pero tal dinámica comienza a convivir con las aplicaciones Web 2.0, para beneficio de los consumidores. Las nuevas compañías ofrecen software gratuito, utilizando a la Web como plataforma. Así, las herramientas y sus contenidos existen en la propia Web y no en la computadora del usuario. El concepto de *webtop* opuesto al de *desktop* es apropiado para explicar este fenómeno comercial.

El modelo de negocio de la Web 1.0 se limitaba a un espacio de publicación de contenidos corporativos y de servicios, sin participación abierta ni gratuidad en contenidos o servicios de alta relevancia. Las comunidades se formaban fundamentalmente a partir de la oferta de servicios, prescindiendo de espacios para que los miembros publicaran contenidos. Es decir, los usuarios fueron relevantes en tanto eran consumidores.

Junto a la *Wikipedia*, el *peer to peer* de *Napster* fue una de las herramientas que más contribuyó a transformar el modelo de contenidos centralizados hacia uno en que los usuarios se convirtieran en servidores. Esto deja en manos de la comunidad todo el poder de la herramienta, en tanto las empresas sólo actúan como intermediarios ofreciendo un software creativo para interactuar. Las herramientas Web 2.0 utilizan su servidor para almacenar la información, y el usuario conectado a la red siempre tiene acceso a ella.

La inspiración radica en una idea clave: el producto mejora con el uso y con el crecimiento de una comunidad que no cesa de subir contenidos. Otro ejemplo



paradigmático de la Web como plataforma es *YouTube* fundada en febrero de 2005, comprada por *Google* en noviembre de 2006 y cuyas cifras de consumo son sorprendentes.

## **2. Aprovechar la inteligencia colectiva.**

La idea se encuentra en el origen de la *World Wide Web* y del *open source*. No constituye un concepto novedoso. Berners-Lee (2000) ya hablaba de estos principios cuando pensó y creó la Web a principios de la década de 1990. Aunque es evidente que la dinámica de la inteligencia colectiva está creciendo de la mano de esta nueva generación de aplicaciones Web, con software en línea diseñado a partir de una interfaz de fácil uso, escalable, de valor añadido en contenidos y de acceso gratuito.

En el entorno Web 2.0 los usuarios actúan de la manera que deseen: en forma tradicional y pasiva, navegando a través de los contenidos; o en forma activa, creando y aportando sus contenidos. *Wikipedia* es como una experiencia radical de confianza donde cualquier usuario puede aportar la definición de un término y cualquier otro puede corregirlo, transformando al usuario de un mero consumidor a un co-desarrollador en forma productiva para la plataforma. *Wikipedia* es uno de los productos más representativos de los valores de la Web 2.0; un medio ambiente igualitario con sentido de neutralidad entre pares. Sin embargo, al mismo tiempo, *Wikipedia* puede utilizarse como icono referente para señalar los problemas de dicha arquitectura de la participación, donde la estructura sitúa en el mismo nivel a escritores amateur y profesionales. En un escenario de 5.3 millones de artículos (*Business Week*, enero 2007), esto provoca algunas imprecisiones conceptuales. Lo que sucede es que no hay filtros en la entrada de contenidos, sino en la salida. La aplicación permite al usuario publicar y luego la comunidad determina la relevancia del contenido. El papel tradicional del editor queda totalmente trastocado y pierde poder en el negocio de la organización y jerarquización de la información, en un entorno donde ya no deciden que exponer y que ignorar.

Bajo la misma noción de inteligencia colectiva, se valor y potencia la relevancia de las aplicaciones de escritura colaborativa en el nuevo periodismo. Los *blogs* proponen un tipo de noticias producidas por ciudadanos que tienen algo para decir –más allá de los *mass media*– quienes siempre tuvieron el monopolio de la primera versión de la historia. Así, se reafirma que los recursos de noticias no tradicionales permiten crear un contexto valioso alternativo a los intereses comerciales de los grandes medios, pues éstos se han convertido en “instituciones arrogantes con un conservadurismo poco crítico”.

A tono con los valores del saber compartido, los campos del conocimiento científico son como grandes *wikis*, editables todo el tiempo a modo de revisión entre pares, en los cuales las nuevas entradas representan trabajos que hacen crecer la inteligencia colectiva en forma eficiente.

### **3. La gestión de la base de datos como competencia básica.**

Este principio tiene una palabra clave: *infoware*: software más datos. Lo valioso de las aplicaciones Web 2.0 son los datos, ya que en muchos casos el software es un recurso abierto o de fácil implementación. Un ejemplo es *Amazon*; con la misma base de datos original que obtuvieron otras librerías en línea. Esta empresa tuvo desde sus comienzos una política de enriquecimiento de información, en muchos casos generados por los usuarios. Hoy su base de datos es mucho más potente que los recursos originales de los que dependió en sus comienzos. La administración de los datos en *Amazon* es su competencia básica y parte esencial de la riqueza del servicio. Es un orden misceláneo con una clasificación multifacética que tiene más que ver con la realidad. Un amigable desorden que promueve diferentes tipos de búsqueda a gusto del usuario, donde la serendipia cumple un factor relevante.

Así, el interés inicial de estos proyectos donde la gestión de la base de datos es la competencia básica es obtener una masa crítica de usuarios que produce un volumen de datos de gran valor. Poseer esta información clave y usuarios co-desarrolladores otorga valor comercial al producto.

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

Como se mencionó antes, se rompe el modelo inicial del software cerrado con derechos de uso y bajo el principio de la obsolescencia planificada, para pasar al uso del software como servicio gratuito, corriendo en la propia Web, y en combinación con los datos. Esto representa un cambio en el modelo de negocio y una de las principales ventajas que el usuario puede extraer de esta transformación.

De manera natural surgen preguntas como: ¿si se acelerara la aparición del software como servicio Web y gratuito desaparecerá el software empaquetado? ¿Qué pasará cuando se popularicen los nuevos tipos de *Photoshop* con menos complejidad de uso, gratuito, con plataforma en la Web y con espacio de servidor suficiente para almacenar datos y compartirlos en una comunidad colaborativa? Se vive una transformación del mercado en la que los actores deben reaccionar. ¿Cómo competirá *Adobe*? ¿Con un paquete completo licenciado a más de 1000 euros y con actualizaciones a 300 euros y renovación forzosa cada dos años? Los paquetes de software tradicionales siguen siendo productos costosos bajo una agresiva política de obsolescencia que en poco beneficia a sus consumidores. En tanto, nuevas aplicaciones Web 2.0 se desarrollan con el objeto de sustituir a estos productos y ganar clientes dispuestos a producir y subir los datos a nuevas plataformas. Por su parte, *Google* representa -con su motor de búsqueda y todos sus servicios complementarios- el modelo de actualización diaria del software sin costes para el usuario. Pero se puede pensar a la inversa, imaginando un potente buscador empaquetado similar *Google* para rastrear contenidos en la Web (más un *Google Maps* como extra) a cientos de euros la última versión y una política eficaz de comunicación publicitaria. ¿Comprarían un producto así? *Google* lo ofrece en forma gratuita, con una oferta de servicios novedosos, sin actualizaciones innecesarias ni incompatibilidades, teniendo a la Web como plataforma y sin requerir espacio de disco duro de un ordenador.

Por otra parte, la cultura de uso del software también comienza a modificarse con estas nuevas estrategias. Desde los lejanos tiempos del *Apple 2*, ha

resurgido una estética *beta* que se traslada a todos los procesos narrativos. Antes existía una gran diferencia entre un prototipo o boceto y un producto final. Hoy sólo se puede pensar en versiones *beta* siempre dispuestas a ser modificadas, a tono con una gestión efímera y una circulación acelerada de la información. El espíritu de las empresas Web 2.0 radica en transformar prototipos en versiones *beta* y colocarlos en línea, utilizando el método de ensayo y error para aprender del consumo de los usuarios. Así se mejora la herramienta en forma constante, sin costes adicionales para el consumidor y acelerando el proceso de mejoras a la aplicación.

### **5. Modelos de programación ligera. Búsqueda de la simplicidad.**

Siempre según los principios constitutivos de O'Reilly<sup>[r02]</sup>, esta noción consta en sustituir los diseños ideales de la arquitectura de la información, de los metadatos y de las interfaces gráficas por un pragmatismo que promueva a la vez simplicidad y fiabilidad para aplicaciones no centralizadas y escalables. Entre otras cosas, se pretende que las aplicaciones crezcan sin complicaciones para el desarrollador y que el usuario pueda ver los contenidos en la plataforma que desee a través de la sindicación y no cuando el desarrollador / proveedor disponga en su plataforma propietaria.

Los modelos de programación ligera ponen énfasis en la reducción de la complejidad, donde menos es más, evitando las excesivas especificaciones y funciones del software empaquetado. ¿Qué porcentaje de funciones del *Microsoft Word* se utilizan? La programación ligera permite también ensamblar productos con creatividad y ofrecer así mayor valor añadido, productividad y sinergias ausentes en productos por separado.

Las aplicaciones Web híbridas denominadas *mashups* son un punto de conexión entre funciones diferentes, permitiendo obtener lo mejor de cada aplicación. La simplicidad genera formas creativas de reutilizar la programación hacia la estandarización, integración y usos bajo demanda, siempre pensando en la inclusión del usuario como productor de contenidos. Un ejemplo son las aplicaciones existentes alrededor de *Google Maps*, que desde su relanzamiento en junio de 2005 (a partir de la compra del *Earth*

*Viewer de Keyhole*), es el centro de un ecosistema de miles de *mashups* que están construyendo una nueva “geoweb”, del que se espera un enorme crecimiento a partir de los servicios basados en la ubicación del entorno Mobile Web.9

Una problemática asociada a la búsqueda de la simplicidad es la actualización de contenidos. Quienes se dedican al diseño Web saben que dicha variable era hasta hace muy poco un elemento más del diseño de las aplicaciones Web, pero, en general, no era el más relevante porque los sitios se modificaban poco, especialmente los contenidos de niveles profundos. Sin embargo, el concepto de actualización ha perdido valor en los productos de la Web 2.0. La actualización es hoy la norma, alentada por el rol creciente de *blogs* y *wikis* y su propia lógica de creación de contenidos en forma dinámica.

#### **6. El software no limitado a un solo dispositivo.**

La utilización de los productos de la Web 2.0 no se limita a las computadoras. Los teléfonos móviles de tercera generación (3G) empezaron a ocupar espacios hasta ahora sólo reservado a aquellas. Aunque los primeros productos fueron pensados en forma similar a sus pares cableados, hoy se han vuelto *mobile devices*, teléfonos más plataformas para el entretenimiento, la gestión de la información, el consumo de los *media*, la producción de contenidos multimediales y el fortalecimiento de las redes sociales. Por otra parte, la sindicación –que consiste en el etiquetado de contenidos de aplicaciones Web para su distribución automática a través de diferentes plataformas– evita tener que navegar por los sitios originales, facilitando la tarea de lectura de los contenidos previamente seleccionados por el usuario. Productos como *Bloglines* o *Google Reader* facilitan la tarea.

#### **7. Experiencias enriquecedoras del usuario.**

Cuando la Web era sólo contenido textual y *gifs* animados, en 1996 apareció *Flash Macromedia* para darle al usuario una experiencia más generosa a nivel gráfico. Sus botones, clips de película, máscaras y programación en *Action Script* promovían zonas activas de interacción. Pero la interacción de *Flash*

## ¿Qué es la Web 2.0?

---

sabe a poco con la intercreatividad y experiencia de usuario que ofrecen las aplicaciones Web 2.0. Si se piensa en contenidos dinámicos, sucede lo mismo. Las clásicas herramientas para producir webs, como *Dreamweaver* y *Flash*, ofrecen la posibilidad de crear un diseño complejo de contenidos dinámicos mediante *HTML*, estilos *CSS* y programación, pero también se quedan atrás de las aplicaciones diseñadas en *Ajax*, *Ruby on Rails* o *BitTorrent* para los productos Web 2.0. Interfaces con la capacidad de acceso en todo lugar y momento a los servicios Web, con la usabilidad, familiaridad y sencillez de las interfaces de los sistemas operativos.

El caso de *Second Life* merece unas líneas como aplicación virtuosa en la experiencia del usuario. Se trata de un mundo / videojuego virtual con una estructura escalable de avatares (personajes) en el entorno de ciudades y mobiliario urbano creados por los usuarios, así como su propia historia de colonización, revuelta popular e independencia, al modo de los estados modernos. Con una población de más de 500 mil habitantes y creciendo a un 36% mensual (*Wired*, Octubre 2006), *Second Life* tiene hasta moneda propia, un mapamundi, medios de transporte, tiendas y hasta eventos culturales. Es una enriquecedora experiencia de usuario creada por *Linden Lab* que hace recordar a los juegos en red y representa un ejemplo de complejidad a nivel de comunidades en línea. Otra experiencia de usuario de mayor calado y popularidad son los *blogs*. La facilidad de creación, productividad, posibilidad de indexación y visibilidad en los buscadores justifican su éxito. A continuación se resumen los porqués de su popularidad: los *blogs* son fáciles de usar, tienen un bajo coste o a veces son gratuitos, son interactivos, humanizan a las organizaciones, son creíbles, inmediatos, directos e infecciosos, pueden ser consultados por los periodistas, no son intrusivos, otorgan autoridad e influencia, permiten llegar a audiencias que han abandonado otros medios, crean comunidad, ayudan a aumentar la notoriedad de la organización en la red, refuerzan la cultura de la organización y ayudan en momentos de crisis institucionales.

## 2.1 TECNOLOGIAS

A lo largo de los últimos años el entramado de las nuevas tecnologías ha crecido exponencialmente tanto en número como en complejidad. Un conjunto de tecnologías que englobaría en su más amplia extensión, una Red Universal Digital (RUD) que viene a constituir la infraestructura tecnológica sobre la que se desarrolla en Nuevo Entorno Tecnosocial y de la cual emergen sus propiedades características. La RUD, según este supuesto, se extendería desde la propia Web a la Internet de las cosas o, siendo más ambiciosos, en un futuro próximo hasta el propio cerebro humano, condicionando con su plasticidad natural la formación de una percepción y organización diferente para los nativos digitales. Este proceso del que se empiezan a obtener pruebas empíricas dispersas y hasta cierto punto prometedor, ha sido bautizado como noomorfosis digital.

Entre estas tecnologías, hasta cierto punto características de la Web de Nueva Generación, hay unos pocos acrónimos y neologismos cuya razón de ser sí vale la pena conocer. Para ello cabría destacar los aspectos clave que permitan su comprensión en el contexto del análisis, sin aportar profundidad técnica o tecnológica, pero incidiendo en su alcance dentro del marco que se ha fijado como referencia conceptual y guía metodológica.

Siempre que se habla de la Web 2.0, se acostumbra a poner como ejemplo ilustrativo a una serie de servicios que se ofrecen a través de la Web, que se caracterizan por ofrecer una interfaz especialmente ágil y flexible, como pueden ser todos los servicios ofrecidos entre otros por las grandes empresas de Internet como Google con GMail, Spreadsheets&Docs por ejemplo o los ofrecidos por Yahoo! tipo Flickr, del.icius, y así un largo etc. Pues bien, detrás de esas aplicaciones, cabría identificar como común denominador la tecnología AJAX (*Asynchronous Javascript And XML*). Bajo este acrónimo, se esconde una combinación creativa de tecnologías bien conocidas desde hace tiempo por los profesionales especializados que permite agilizar la interacción entre el navegador y el propio usuario. Para tal objetivo lo verdaderamente importante es el impacto que ha producido la popularización de las interfaces

## ¿Qué es la Web 2.0?

---

Web realizadas sobre las mismas porque, en definitiva, “para el usuario final, el producto es la interfaz”, dándole la vuelta a la conocida metáfora informática, el gurú del marketing.

La complejidad técnica de los productos, servicios y aplicaciones que llegan a los usuarios finales no necesariamente especializados en las nuevas tecnologías, resultaría inabordable sin un esfuerzo especializado de modelización que permita manejarla adecuadamente. En el diseño y arquitectura de sistemas se impone hace ya unos años lo que se conoce como REST (*REpresentational Status Transfer*) y que surge de la tesis doctoral del filósofo Thomas Fielding en la Universidad de California Irvine. Se trata de una aproximación novedosa a la filosofía de diseño software que extiende el concepto de hipertexto a las aplicaciones y los servicios en la Red, utilizando como metáforas los recursos (*nodos*) y los enlaces a los mismos.

Para una mejor comprensión, podría decirse que en último término lo que se pretende es ampliar la metáfora de la Web al propio diseño de aplicaciones. Interesa destacar lo que la extensión de la filosofía REST significa para el futuro de la arquitectura Web, y lo que tiene de vuelta a los orígenes de la Red, tal y como fue concebida hace ya 15 años. Es importante destacar el impacto que –sobre todo a medio y largo plazo– puede tener su adopción en la interoperabilidad entre servicios en la Web, es decir que es interesante ver cómo se consolida una tendencia que aporta elementos tecnológicos para la construcción de arquitecturas de sistemas que soporten el crecimiento de una Web más social, más interactiva y, con muchos matices, también más participativa.

Los sistemas de gestión de contenido (Content Management Systems, CMS) constituyen los cimientos esenciales de las plataformas de prestación de servicios de publicación y colaboración apoyados en blogs y wikis. Es decir que cuando se hace referencia a cualquiera de estas herramientas de colaboración, comunicación, expresión, etc. Hay que saber que detrás hay unos sistemas técnicos de cierta complejidad, que forman parte de esa infraestructura invisible que se confunde con el entramado de la RUD y que,



en términos genéricos, y sin pretender una estricta rigurosidad en su clasificación, pueden catalogarse dentro de ese tipo de sistemas, diseñados para la creación, manipulación, compartición y/o publicación de contenido. Este tipo de sistemas aparecen dentro del ámbito de la informática empresarial, en parte como una evolución de los sistemas de gestión documental y se relacionan con algunas herramientas de gestión del conocimiento, así como entornos de colaboración y trabajo en equipo, además de formar parte integrante de las más complejas plataformas para la publicación Web que soportan las tradicionales intranets corporativas.

La relativa facilidad de implementación y de uso que ofrecen esos sistemas de publicación, ha sido responsable en gran medida de la gran proliferación de información en la Red. Precisamente, al hilo de estos sistemas, y en la línea de intentar frenar o filtrar una avalancha informativa que desborda y coloca a algunos al borde del abismo de la infoxicación surgen nuevos formatos e innovadoras herramientas para su creación y gestión. Los estándares para la sindicación de contenidos, entre los que destacan el RSS (Really Simple Syndication) y Atom como formatos técnicos más populares se presentan ante el usuario medio de Internet como una forma de acceder a sus fuentes (feeds) de información de forma rápida y sencilla.

La publicación de la información siguiendo esos formatos permite que los usuarios se puedan suscribir como si de canales sintonizables en un receptor de radio o televisión se tratara. Son precisamente esas tres letras RSS, las que hacen referencia a un conjunto de tecnologías y servicios que conforman el instrumental infotecnológico habitual del usuario consumidor de información más activo, y que ha tomado parte en la faceta mediática del fenómeno blog. Como herramienta (o como servicio) es el agregador de información (en forma de fuentes publicadas en un formato adecuadamente estandarizado para su posterior sindicación) el elemento que más se acerca a la metáfora del Daily-Me, ilustración visionaria del periódico a la carta que pronosticara N. Negroponte hace ya más de una década en Being Digital.

## ¿Qué es la Web 2.0?

---

Para finalizar este repaso preliminar, cabe destacar dos neologismos más: los microformatos y las folksonomías. Los primeros surgen como alternativa, al menos en lo que a su composición se refiere, a algún tipo de Macroformato. La historia muestra que, ante la construcción normativa de una gran cantidad de estándares técnicos para representar el significado de los trocitos de información que se encuentran por la Red en un formato que pudieran comprender e intercambiar fácilmente las máquinas (los Macroformatos, que incluyen una serie de acrónimos). Un grupo de internautas capaces y motivados decidieron aplicar aquel popular corolario del principio apócrifo de la navaja de Occam –soluciones tontas a problemas simples– incorporando inteligencia y significado a la Red, a través del propio usuario. Nacen así una serie de procedimientos y formatos estandarizados de facto por los propios usuarios, conocidos como microformatos, entre los que destaca el de asignación de etiquetas al contenido que te encuentras en la Web de la manera más sencilla posible.

En definitiva, los microformatos y las folksonomías son el máximo exponente de un fenómeno con nombre propio, el etiquetado semántico en colaboración, que ha cambiado la forma en que se genera y consume un contenido multimediático y fragmentado, que se intercambia y comparte en entornos que van del grupo de trabajo y la comunidad a la red social. El elemento técnico en este caso lo constituyen determinados componentes morfosintácticos del XHTML –una variante del XML (eXtended Markup Language), considerado la lengua franca de Internet– que algunos servicios Web permiten transformar en etiquetas, una suerte de palabras clave asociadas con el contenido al que acompañan y que sirven para catalogarlo sobre la base, en principio, de un espacio plano de categorías. Un espacio constituido por las aportaciones de todos los usuarios de dichos servicios, a través de las etiquetas, sin una intervención centralizada ni más autoridad que el uso que de éstas hagan los propios usuarios. A modo de ejemplo cabría destacar, delicious iniciativa pionera en la publicación de este tipo de servicio en la Red, también lo fue en la introducción de un nuevo nivel de creatividad en la construcción de

direcciones y que se ha convertido en uno de los servicios más representativos y exitosos de la Web 2.0, una iniciativa personal de un joven desarrollador Stewart Butterfield, finalmente adquirida por Yahoo.

Hay que llamar la atención sobre el tipo de elementos tecnológicos descritos brevemente hasta el momento, todos ellos componentes tecnológicos que tienen que ver con la interfaz, la forma de presentar el producto y la información. En definitiva un bloque arquitectónico característico, así como una filosofía de diseño que extiende algunas características propias de la Red al desarrollo de la infraestructura de servicios. De la misma forma se ha repasado la manera en la que se representa la información para su consumo en unos procesos también característicos del NET, como es el etiquetado semántico en colaboración. Es importante porque este mismo esquema, con una infraestructura de servicios Web soportando una serie de procesos más o menos innovadores, que se gestionan mediante unas interfaces relativamente ágiles y sencillas, es sobre el que se insistirá en apartados posteriores de forma más sistemática y utilizándolo como base metodológica.

Para resumir, a continuación se explica con mayor detalle algunas de las tecnologías asociadas con la Web 2.0:

- **AJAX (Asynchronous JavaScript And XML, Javascript y XML Asíncronos):** Técnica de desarrollo Web que permite modificar la información de una página Web sin tener que recargarla completamente, agilizando de esta forma la interacción con el usuario, incluye varias tecnologías, las cuales se mencionarán más adelante.
- **API (Application Programming Interface, Interfaz de Programación de Aplicaciones):** Una API representa un interfaz de comunicación entre componentes software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque

## ¿Qué es la Web 2.0?

---

no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. Ejemplos:

<u>API</u>	<u>Descripción</u>	<u>Categoría</u>
Windows Live ID Web Authentication	Web based authentication service	Seguridad
Windows Live Messenger IM Control	Web based chat service	Mensajería
Windows Live Presence	Online presence service	Mensajería
PriceGrabber	Comparison shopping service	Compras
AOL Open Mail	Email services	Email
Lat49	Geo-targeted advertising for online maps	Publicidad
USAspending	Database of US government spending	Gobierno
Shopzilla	Comparison shopping service	Compras
Amazon DevPay License Service	Amazon Web services licensing	Finanzas
Right Media Exchange	Advertising management	Publicidad

- **Beta (Versión en pruebas):** Medio en broma, medio en serio, los servicios de la Web 2.0 se encuentran siempre en «beta perpetua» evolucionando constantemente, a diferencia de los ciclos cerrados de desarrollo de otros servicios y aplicaciones.

Este es el cuarto de los siete principios de O'reilly, mencionados con anterioridad, acerca de la Web 2.0 publicados en Septiembre de 2005 de la conferencia de Octubre de 2004.

- ***Mashup (Remezcla)***: Aplicación Web que utiliza información de diversas fuentes relevantes para crear un nuevo servicio en base a ellas, normalmente empleando Servicios Web y APIs públicas. Los MashUps son una parte de una tendencia más grande en la Web 2.0 respecto a la portabilidad de datos. Hay muchas tecnologías y usos diferentes de dichas tecnologías que enfatizan la capacidad de tomar datos de una fuente y utilizarlos en otro programa, u otro sitio Web u otro contexto y los mashups son algunos de los más visibles ejemplos de aquellos, de datos que son tomados de una fuente y son utilizados en otro lugar, en otro programa, en otra aplicación, en otro sitio Web, mayormente para propósitos de visualización, pero no exclusivamente.
- ***OPML (Outline Processor Markup Language, Lenguaje de Marcas para Esquemas)***: Un formato para esquemas y listas, en lenguaje XML, que se suele utilizar para compartir información, siguiendo los pasos del trabajo y filosofía colaborativo de la Web 2.0. El uso de OPML más habitual hoy en día es para listar varias fuentes RSS juntas (técnica para distribuir y agregar información y reunirla en una misma página Web); también se puede usar, por supuesto, en programas gestores personales de información. Se compone principalmente de 4 etiquetas: el nodo raíz, *opml*; *head* y *body* para indicar la cabecera y el cuerpo del documento; y *outline* para cada una de las líneas en el esquema.
- ***P2P (Peer-to-Peer, Entre Pares)***: Red informática entre pares o «iguales», normalmente sin servidores centralizados, donde se puede compartir información de cualquier tipo. Ejemplos: Skype, Windows Live Messenger, BitTorrent, Kazaa, eMule. El P2P se basa principalmente en la filosofía e ideales de la Web 2.0 de que

todos los usuarios deben compartir. Conocida como filosofía P2P, es aplicada en algunas redes en forma de un sistema enteramente meritocrático en donde "el que más comparta, más privilegios tiene y más acceso dispone de manera más rápida a más contenido". Con este sistema se pretende asegurar la disponibilidad del contenido compartido, ya que de lo contrario no sería posible la subsistencia de la red. Aquellos usuarios que no comparten contenido en el sistema y con ello no siguen la filosofía propia de esta red, se les denomina "leechers"; los cuales muchas veces representan una amenaza para la disponibilidad de recursos en una red P2P debido a que únicamente consumen recursos sin reponer lo que consumen, por ende podrían agotar los recursos compartidos y atentar contra la estabilidad de la misma.

- **Permalinks (Enlaces Permanentes):** Enlaces fijos a páginas de contenido en Internet que no varían con el tiempo, aunque ya no esté en la página principal, un tema pueda ser localizable de manera expedita, estos son idealmente fáciles de leer para las personas. Los permalinks se asignan a cada post o entrada en un blog o bitácora.
- **REST (Representational State Transfer, Transferencia de Estado Representacional):** Filosofía de diseño y arquitectura Web que se apoya en el intercambio de información mediante XML. El término *REST* se refería originalmente a un conjunto de principios. En la Web 2.0 se usa en el sentido más amplio para describir cualquier interfaz, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como el protocolo de servicios Web SOAP. Es posible diseñar sistemas de servicios Web de acuerdo con el estilo arquitectural REST de Fielding y también es posible diseñar interfaces

XMLHTTP de acuerdo con el estilo de llamada a procedimiento remoto pero sin usar SOAP.

- **SOAP (Simple Object Access Protocol, Protocolo Simple de Acceso a Objetos):** Protocolo estándar de comunicación utilizado en los servicios Web, mediante el cual dos objetos pueden intercambiar información en XML, esto es una ventaja ya que facilita su lectura por parte de humanos, esto hace que se facilite el uso de la Web que es lo que busca la Web 2.0, por y para la gente, pero también es un inconveniente dado que los mensajes resultantes son más largos. El intercambio de mensajes se realiza mediante tecnología de componentes. El término *Object* en el nombre significa que se adhiere al paradigma de la programación orientada a objetos. SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.
- **RSS (Really Simple Syndication, Sindicación Bastante Simple):** Formato estándar para la distribución de contenidos a los que un usuario cualquiera puede suscribirse mediante un programa «agregador de feeds o canales». Esta técnica se profundiza más adelante.
- **Ruby on Rails:** Entorno de programación (Rails) que se apoya en el lenguaje Ruby. Goza de gran popularidad para el desarrollo de aplicaciones de tipo Web 2.0 puesto que automatiza las tareas más comunes y permite ciclos cortos para el desarrollo de prototipos. Los principios fundamentales de Ruby on Rails incluyen “No te repitas” (del inglés Don't repeat yourself, DRY) y “Convención sobre configuración”. No te repitas significa que las definiciones deberían hacerse una sola vez. Dado que Ruby on Rails es un framework de pila completa, los componentes están

integrados de manera que no hace falta establecer puentes entre ellos. Por ejemplo, en ActiveRecord, las definiciones de las clases no necesitan especificar los nombres de las columnas; Ruby puede averiguarlos a partir de la propia base de datos, de forma que definirlos tanto en el código como en el programa sería redundante. Convención sobre configuración significa que el programador sólo necesita definir aquella configuración que no es convencional. Por ejemplo, si hay una clase Historia en el modelo, la tabla correspondiente de la base de datos es historias, pero si la tabla no sigue la convención (por ejemplo blogposts) debe ser especificada manualmente (`set_table_name "blogposts"`). Así, cuando se diseña una aplicación partiendo de cero sin una base de datos preexistente, el seguir las convenciones de Rails significa usar menos código (aunque el comportamiento puede ser configurado si el sistema debe ser compatible con un sistema heredado anterior)

- **Widget:** Pequeña aplicación o módulo que realiza una función concreta, generalmente de tipo visual, dentro de otras aplicaciones o sistemas operativos. Entre sus objetivos están los de dar fácil acceso a funciones frecuentemente usadas y proveer de información visual. los *widgets* pueden hacer todo lo que la imaginación desee e interactuar con servicios e información distribuida en Internet; pueden ser vistosos relojes en pantalla, notas, calculadoras, calendarios, agendas, juegos, ventanas con información del clima en su ciudad, etcétera. Los *widgets* de escritorio también se conocen como *gadgets*, una nueva categoría de mini aplicaciones, diseñadas para proveer de información o mejorar una aplicación o servicios de un ordenador o computadora, o bien cualquier tipo de interacción a través del World Wide Web, por ejemplo una extensión de alguna aplicación de negocios, que provea información en tiempo real



del estatus del negocio u organización, que es lo que hace una Web más actualizada, una Web 2.0

## **2.2 TECNICAS**

La arquitectura de la Web 2.0 es compleja y va evolucionando, pero incluye el software de servidor, sindicación de contenidos, protocolos de mensajes, navegadores basados en estándares, y varias aplicaciones para clientes.

Una Web se puede decir que está construida usando tecnología de la Web 2.0 si se caracteriza por las siguientes técnicas, algunas mencionadas anteriormente:

- CSS, marcado XHTML válido semánticamente y Microformatos. Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

XHTML, acrónimo inglés de eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es la versión XML de HTML, que a pesar de tener las mismas funcionalidades, cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del W3C de lograr una Web semántica, donde la información, y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo (como las hojas de estilo en cascada, CSS) y JavaScript su aspecto y diseño en distintos medios (computadoras, PDAs, teléfonos móviles, impresoras, etc). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- ✓ Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.
- ✓ Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web, con lo que

aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.

- ✓ Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- ✓ El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.
- Técnicas de aplicaciones ricas no intrusivas (como AJAX)

El DHTML, LAMP o SPA, AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente, pero se enfatizará en AJAX, la cual ya se ha mencionado en este documento.

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

- ✓ XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- ✓ Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.

- ✓ El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- ✓ XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

- Java Web Start

Java Web Start es la implementación de referencia de la especificación JNLP (Java Networking Launching Protocol) y está desarrollada por Sun Microsystems, mediante la cual permite arrancar aplicaciones Java que están en un servidor Web de aplicaciones comprobando previamente si el cliente tiene la versión actualizada de dicha aplicación. Si no es así descargará la última versión y se ejecutará en local. El arranque de dichas aplicaciones puede ser efectuado mediante enlaces en una página Web o bien a través de enlaces en el escritorio cliente. Mediante esta tecnología se asegura que una aplicación es distribuida siempre en su última versión. Los archivos que contienen la información sobre donde se encuentra la aplicación, versión, etc. tienen la extensión jnlp. Actualmente Java Web Start viene incluido en el JRE (Java Runtime Environment).

- XUL (acrónimo de XML-based User-interface Language), XUL es un lenguaje basado en XML utilizado para describir y crear interfaces de usuario, que ha sido diseñado para brindar la portabilidad de las mismas, por lo que permite desarrollar aplicaciones multi-plataforma sofisticadas o complejas sin necesidad de herramientas especiales.

Inicialmente XUL fue creado para desarrollar los productos de Mozilla (navegador y cliente de e-mail, entre otros) de una forma más rápida y fácil. Al ser un lenguaje basado en XML, contiene todas las características disponibles para XML y sus mismas ventajas.

## ¿Qué es la Web 2.0?

---

Estas aplicaciones son fácilmente personalizadas con textos, gráficos y demás, por lo que pueden ser rápidamente localizadas en diferentes mercados. Los desarrolladores Web que estén familiarizados con HTML dinámico (DHTML) podrán aprender XUL rápidamente e incorporarlo en el desarrollo de sus aplicaciones.

Una interfaz XUL es definida mediante la especificación de tres grupos de componentes distintos:

**Content:** Aquí se encuentran los documentos XUL, que definen el diseño de la interfaz.

**Skin:** Contiene las hojas de estilos (CSS) y las imágenes, las cuales definen la apariencia de la interfaz.

**Locale:** Los documentos DTD se encuentran aquí, estos documentos facilitan la localización de páginas XUL.

- Sindicación y Agregación de datos en RSS/ATOM

RSS es un sublenguaje surgido de la aplicación del metalenguaje XML. Por tanto un archivo RSS es un documento de texto compuesto por etiquetas acotadas entre los símbolos mayor y menor que similares a las utilizadas en el XHTML. RSS corresponde a Rich Site Summary o Really Simple Syndication, y está diseñado para la distribución (syndication en inglés) de noticias o información tipo noticias contenidas en sitios Web y weblogs.

Los archivos RSS comúnmente se llaman feeds RSS o canales RSS y contienen un resumen de lo publicado en el sitio Web de origen. Se estructura en uno o más ítems. Cada ítem consta de un título, un resumen de texto y un enlace a la fuente original en la Web donde se encuentra el texto completo. Además puede incluir información adicional como el nombre del autor o la fecha y la hora de publicación del contenido. Por tanto, cualquier fuente de información susceptible de poder ser troceada en ítems (los mensajes de un foro, por ejemplo) pueden distribuirse utilizando RSS.

El archivo RSS contiene además un enlace específico para cada ítem contenido en el feed que dirige a la página Web con el texto completo de la noticia. Para leer los feeds o canales RSS es necesario utilizar un programa llamado agregador. Este tipo de programas también se conocen como lectores de feeds o canales o agregadores de noticias, entre otras variaciones.

ATOM también es un sublenguaje XML. No se corresponde ni se basa en ninguna versión de RSS, pero es un formato muy similar a éste y que sobre todo tiene el mismo objetivo: permitir la distribución de contenidos y noticias de sitios Web. Las mejoras que supone respecto a RSS (en cualquiera de sus versiones) hacen que su uso se extienda rápidamente a pesar de ser algo más complicado. Un documento Atom puede contener más información (y más compleja) y es más consistente que un documento RSS.

- URLs sencillas y con significado, con palabras significativas y legibles, produce un efecto de confortabilidad en el usuario, que no se siente navegando en una máquina calculadora. Tienen cierta armonía, y se ve claramente a que se refieren.

Dado que no se concibe ahora un Web sin un gestor de contenidos (CMS), uno de los factores a tener en cuenta en ellos es la usabilidad de las url que generan. Todo CMS debe permitir construir alias para grandes secciones de una Web: por ejemplo, cambiar [http://www.komuniautomica.ess/niuweb2/servlet/integra.servlets.ControlPublico?IDCONTENIDO=67&IDTIPO=200&PLANT\\_PERSONALIZADA=/TSP/GARM/plantillasPortal/organigramas/plantillaDetalleOrganigrama.jsp&IDESTRUCTURAJERARQUICA=150&RASTRO=c\\$m120,121](http://www.komuniautomica.ess/niuweb2/servlet/integra.servlets.ControlPublico?IDCONTENIDO=67&IDTIPO=200&PLANT_PERSONALIZADA=/TSP/GARM/plantillasPortal/organigramas/plantillaDetalleOrganigrama.jsp&IDESTRUCTURAJERARQUICA=150&RASTRO=c$m120,121) por <http://www.carm.es/organizacion/medioambiente/>.

Esta administración no debe realizarse desde el servidor http, sino como una función más de gestión del proyecto Web. De este modo los contenidos propios pueden ser copiados con confianza, y mandados por correo electrónico a un amigo, o incluso pueden ser recordadores y tecleados, tanto para entrar a una página, como para navegar.

## ¿Qué es la Web 2.0?

---

- Soporte Para Postear En Un Blog

Existen varios soportes para crear y entrar a los blogs con el fin de estimular el uso de la Web 2.0, en especial de blogs entre la comunidad digital, eso incluye tutoriales, textos, videos, etc.

- JCC y APIs REST o XML

JCC, acrónimo de JavaScript Client Communication hace referencia a las técnicas de programación que, utilizando objetos JSI (JavaScript Shared Interfaces) en el navegador (en el lado cliente y no en el servidor), facilitan la integración en la misma página Web de aplicaciones y servicios a priori independientes. Como AJAX o Web 2.0, JCC no constituye una tecnología nueva en sí misma, sino que es un término que referencia a determinada técnica de programación basada en JavaScript.

En JCC la comunicación entre servicios integrados en una misma página Web se resuelve utilizando objetos JSI. Estos objetos son una modelización conceptual de APIs programadas en Javascript. Cada servicio implementa un API de este tipo (un JSI) publicando las acciones/operaciones que permite que realicen en el componente que controla otros componentes (controlados a su vez por otros JSI) que se sindicarían. Estas JSI, combinadas con la filosofía ya consolidada de despliegue de Web Services y la aplicación de las últimas técnicas respecto a creación de Aplicaciones Web como AJAX, REST, XML-RPC, SOAP, etc., dotan a las aplicaciones y servicios disponibles en la red de una potencia hasta ahora desconocida, aligerando el peso de los interfaces y mejorando notablemente la usabilidad de los interfaces Web. Contando con que el uso de JCC potenciaría el concepto de creación de sitios Web mediante partes de otros sitios (sindicación de contenidos), se podría decir que estaría encuadrada dentro de las nuevas tendencias de la Web 2.0.

## 2.3 ARQUITECTURA

Los desarrollos de aplicaciones hechos para la nueva Internet (Web 2.0) están fundamentados principalmente en la elaboración de una red colaborativa entre individuos, la cual es conocida como la arquitectura de la participación. Es decir la base más poderosa que soporta a la Web, se fortalece en la medida en que más usuarios la utilicen, donde esta arquitectura se construye alrededor de las personas y no de las tecnologías.

La estructura tecnológica se expande de manera conjunta con las interacciones sociales de los sujetos que utilizan Internet. Bajo esta idea, cada vez que una persona crea un nuevo enlace la Red se complejiza y, por tanto, se enriquece. La idea de una arquitectura de la participación se basa en el principio de que las nuevas tecnologías potencian el intercambio y la colaboración entre los usuarios.

Una de las cualidades de la Web 2.0 es que provee de innumerables instrumentos de cooperación, que no sólo aceleran las interacciones sociales entre personas que se encuentran separadas por las dimensiones del tiempo y/o el espacio, sino que además su estructura reticular promueve la gestación de espacios abiertos a la colaboración y la inteligencia colectiva gracias a que muchos interactúan con muchos (Peer-to-Peer).

Esta arquitectura de la participación, sobre la que se construye la Web 2.0, brinda nuevas herramientas de *empowerment* y, al mismo tiempo, de democratización en cuanto al intercambio del conocimiento. A fin de cuentas, todo este universo de desarrollos y avances tecnológicos tiene como pilar fundamental la valoración del usuario como pieza clave en el puzzle de la evolución tecnológica.

El poder de la plataforma de la Web 2.0 es su capacidad para servir de intermediario a la circulación de datos proporcionados por los usuarios. De manera que tras esta arquitectura de participación hay una ética de cooperación implícita, donde la Web actúa sobre todo como intermediario

## ¿Qué es la Web 2.0?

---

inteligente, conectando los extremos entre sí y aprovechando las posibilidades que ofrecen los propios usuarios.

Esta arquitectura de la participación da cuenta de un cambio tecnológico pero más aún de un cambio social que ofrece a las comunidades la posibilidad de contar con herramientas que multipliquen las formas en que se genera y distribuye el conocimiento. Desde ésta perspectiva, la apertura es la pieza clave de este círculo virtuoso de participación y colaboración.

La aplicabilidad de la arquitectura peer-to-peer a la Web 2.0 se percibe al saber que en ésta se implementan arquitecturas, que desarrollen uno de sus principales objetivos que es el de la conformación de redes sociales y colaborativas, el cual se soporta alrededor de los usuarios quienes se valen de instrumentos de cooperación que deja de lado factores como tiempo y espacio y ofrecen puntos de encuentros donde muchos interactúan con muchos compartiendo información de cualquier tipo, siendo esta la filosofía de la arquitectura peer-to-peer o arquitectura de la participación.

En estos puntos de encuentros los usuarios tienen la función de poner a disposición su información a compartir, de manera que por la ausencia de servidores centralizados, estos se convierten en servidores de otros individuos dentro de la misma red, asegurando así la disponibilidad de contenidos y mejorando las ventajas de descargas de la información fácil y con mayor velocidad.

Una aplicación peer to peer es diferente al modelo cliente servidor porque la aplicación puede actuar, como ya se menciona, tanto como cliente como servidor. Es decir ellos están capacitados para pedir información a otros servidores, y también tienen la habilidad de actuar como servidor respondiendo al mismo tiempo pedidos de información de otros clientes.

Este aprovechamiento incrementa la cantidad de valores que cada nodo en la red puede añadir, porque no sólo toma información de un solo recurso, sino que también tiene la habilidad de distribuir información con otros recursos.



Una aplicación típica P2P tiene las siguientes características claves que ayudan a definirla:

- Descubrir otro compañero: La aplicación debe ser capaz de encontrar otras aplicaciones que están dispuestas a distribuir información. Históricamente, la aplicación encuentra esos compañeros de acuerdo a los registros de un servidor central que mantiene una lista de todas las aplicaciones que frecuentemente se disponen a distribuir, dando esa lista a cualquier nueva aplicación, a medida que ellas se conectan a la red. Sin embargo hay otras, formas posibles, por ejemplo redes de emisión – difusión, o algoritmos de descubrimiento.
- Consultando compañeros por contenido: Una vez que esos compañeros son descubiertos, la aplicación puede preguntarles por el contenido que es deseado por la aplicación, o sea por ella.
- Distribuyendo contenidos con otros compañeros: Los compañeros pueden consultar a otros por su contenido, y una vez hecho esto pueden ellos también distribuir el contenido descubierto.

Los programas P2P consisten esencialmente en un motor de transferencia de archivos entre usuarios corrientes conectados entre sí. El programa gestiona los intercambios entre *peers* (usuarios) usando los recursos de sus propios ordenadores. Como se mencionó en la primera característica hay tres tipos de redes, según tengan éstas, o no, un servidor central que gestione las transacciones.

Las redes centralizadas utilizan un servidor central que gestiona todas las operaciones de intercambio. A través de este ordenador, los clientes conectados a la red pueden localizar y transferir archivos entre ellos. Es lógico pensar que una red gestionada a través de una sola máquina tiene un índice de vulnerabilidad alto ya que cualquier ataque que se produzca a dicho servidor supone la anulación de todas las operaciones.

Para evitar el problema de la vulnerabilidad de las redes con un servidor central se crearon las redes descentralizadas, entre las cuales destacan

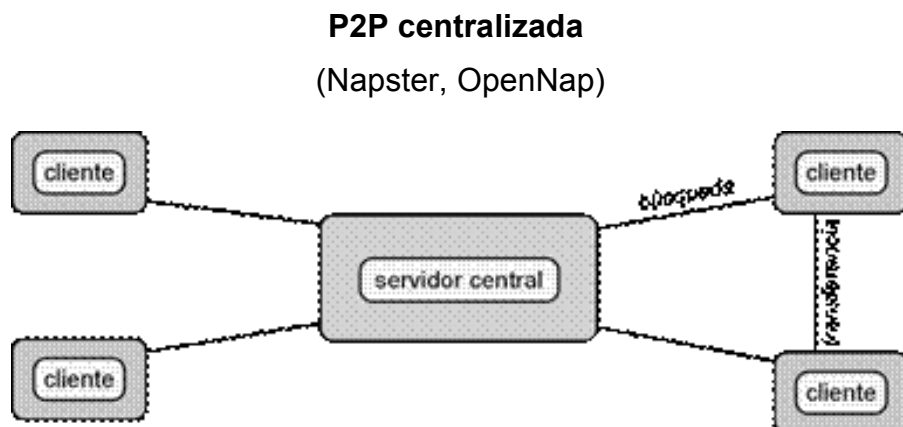
## ¿Qué es la Web 2.0?

---

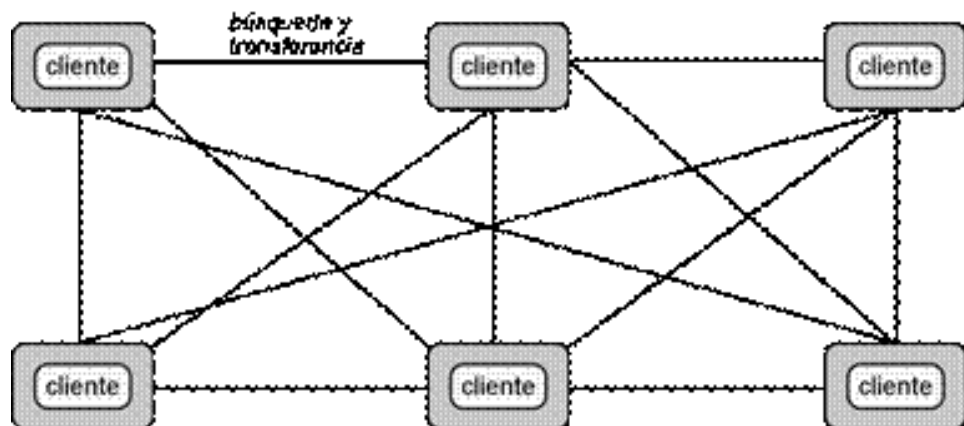
Gnutella y Freenet. Estas redes no utilizan servidor central y por tanto son mucho menos susceptibles a ataques pero, en cambio, la gestión de las operaciones de búsqueda y transferencia es mucho menos eficiente.

Las redes híbridas son las que actualmente gozan de una mayor representación. Sus creadores han sabido aprovechar las ventajas del sistema centralizado y del descentralizado consiguiendo, así, un equilibrio entre resistencia a ataques y eficiencia. Las operaciones de búsqueda y transferencia son gestionadas en este caso por los denominados *super-peers* (super-clientes) que actúan como nodos activos y agilizan el funcionamiento de la red. Kazaa y eDonkey son un buen ejemplo de gestión eficiente y optimización de los recursos.

La figura 6 representa el funcionamiento de cada uno de los tipos de redes P2P.

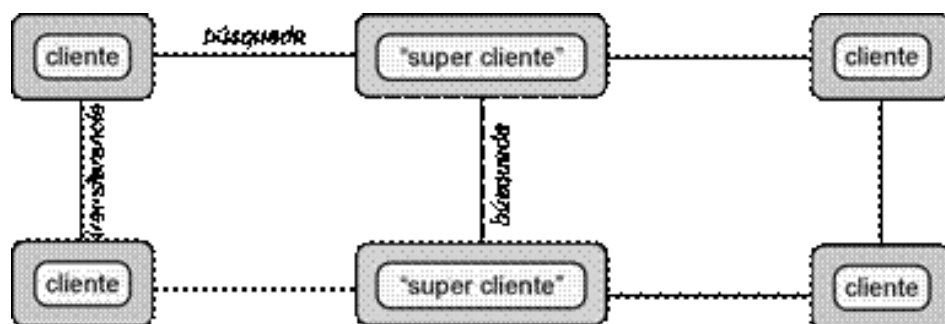


**P2P descentralizada**  
(Gnutella, Freenet)



### P2P híbrida

(Fast Track, eDonkey, WinMX)



**Figura 6. Tipos de redes P2P. Tomado de *Scripta Nova* REVISTA ELECTRÓNICA DE GEOGRAFÍA Y CIENCIAS SOCIALES.** Universidad de Barcelona. ISSN: 1138-9788. Depósito Legal: B. 21.741-98 Vol. VIII, núm. 170 (54), 1 de agosto de 2004. Dirección electrónica: <http://www.ub.es/geocrit/sn/sn-170-54.htm>

Tras los conceptos hasta aquí expuestos: Intercreatividad (Berners-Lee); Inteligencia colectiva (Lévy); Multitudes Inteligentes (Rheingold); Sabiduría de las Multitudes (Surowiecki) y Arquitectura de la Participación (O'Reilly), se

desprende como principio transversal la idea de la cooperación y *empowerment* del conocimiento a través de su colectivización e intercambio.

Se puede mencionar la existencia de dos grandes tipologías de usuarios de las nuevas tecnologías: los nativos digitales y los inmigrantes digitales. Los primeros serían aquellos consumidores y próximos productores de casi todo lo que existe (y existirá), son jóvenes que están desplegando plenamente su aprendizaje en el entorno tecnológico cuya lengua materna es la de las computadoras, los videojuegos e Internet. Frente a ellos, los inmigrantes digitales serían aquellos nacidos con anterioridad a la era tecnológica pero que sienten fascinación e interés por la tecnología, habiendo experimentado un proceso de adaptación más lento y lineal.

Mientras para los padres y educadores (generalmente inmigrantes digitales) el "conocimiento es el poder", en el emergente mundo liderado por los nativos digitales "el poder está en compartir el conocimiento". Evidentemente los principios de colectivización y apertura al intercambio del conocimiento previamente expuesto se adoptan de manera mucho más cercana a los hábitos de los nativos digitales.

### **2.4 ESTANDARES**

Los estándares Web son un conjunto de recomendaciones dadas por el World Wide Web Consortium (W3C, asociación internacional formada por organizaciones miembro del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares Web, dirigido por Berners-Lee, quien también es el fundador) y por otras organizaciones internacionales acerca de cómo crear e interpretar documentos basados en el Web. Son un conjunto de tecnologías orientadas a brindar beneficios a la mayor cantidad de usuarios, asegurando la vigencia de todo documento publicado en el Web.

El W3C trabaja en función a la creación de Estándares Web y pautas para alcanzar su objetivo. En los primeros diez años de su existencia, el W3C ha

publicado más de 80 estándares, como son las “Recomendaciones del W3C”. En esencia, la Web está basada en tres categorías de estándares:

- el 'Identificador de Recurso Uniforme (*URI*), que es un sistema universal para referenciar recursos en la Web, como páginas Web,
- el *Protocolo de Transferencia de Hipertexto* (HTTP), que especifica cómo se comunican el navegador y el servidor entre ellos, y
- el *Lenguaje de Marcado de Hipertexto* (HTML), usado para definir la estructura y contenido de documentos de hipertexto. HTML ha evolucionado durante su desarrollo y está disponible en varias *versiones*. Todas ellas son estándares, y puedes elegir una que se adapte a tus necesidades. La mayoría de las veces, la última versión será la mejor elección, a menos que te dirijas a una audiencia muy específica, o los navegadores más antiguos, o interrumpidos. La versión que elijas define los elementos y atributos que puedes usar

Los estándares Web más recientes se pueden clasificar en 5 categorías:

#### 1. Lenguajes estructurales o de contenido

- ✓ XHTML 1.0: Cabe apuntar que el XHTML 1.0 fue diseñado expresamente para ser mostrado en navegadores que soportan HTML de base. Los documentos escritos conforme a XHTML 1.0 pueden potencialmente presentar mejor rendimiento en las actuales herramientas web que aquellos escritos conforme a HTML. XHTML 1.0 sienta la base para una familia de tipos de documentos que extenderán y acotarán XHTML con objeto de soportar un amplio rango de nuevos dispositivos y aplicaciones (evolución hacia Web 2.0), definiendo módulos que especifiquen un mecanismo para combinar dichos módulos.
- ✓ XHTML 1.1: Representa una salida tanto de HTML 4 como XHTML 1.0. Lo más significativo es la eliminación de características que estaban en desuso. En general, la estrategia es definir un lenguaje de marcado que sea rico en funcionalidad estructural, pero que

delegue en las hojas de estilo para la presentación (CSS), haciendo la web más rápida y enfocándola hacia la web semántica.

- ✓ XML 1.0: El lenguaje extensible de marcas (XML) es un subconjunto de SGML. Su objetivo es permitir que SGML genérico pueda ser servido, recibido y procesado en la web en la misma manera que con HTML. XML ha sido diseñado de tal manera que sea fácil de implementar y buscando interoperabilidad tanto con SGML como con HTML.

### 2. Lenguajes de Presentación

El W3C, recomendó el uso de las CSS en diciembre de 1996 con la ratificación de la especificación Nivel 1 de CSS. El nivel 1 de CSS describía los atributos para usarse en las páginas HTML. Estos atributos reemplazaron la etiqueta font tradicional y otros marcadores de "estilo" como el color y los márgenes. En mayo de 1998, el W3C ratificó el nivel 2 de CSS que añadía capacidades adicionales a la especificación del nivel 1 e introdujo los atributos de posicionamiento. Estos atributos reemplazaron el uso incontrolado (e incorrecto) de la etiqueta table para diseñar la presentación de elementos de página. La revisión más reciente a las especificaciones CSS es CSS 2.1, que perfecciona algunos atributos y elimina otros que si acaso tenían algún uso en los navegadores actuales, era muy poco. CSS nivel 3 está en desarrollo.

Ventajas del CSS:

- Se evitan hacer a los archivos demasiado pesados.
- Se definen el estilo de un sitio entero sin necesidad de hacerlo etiqueta por etiqueta. Cualquier cambio hecho a un estilo CSS, se reflejará en todos los elementos que sean referidos a éste, automáticamente.

- ✓ CSS Level 1: Es un mecanismo simple de hojas de estilo que permite a autores y lectores asociar estilos (por ejemplo fuentes, colores y espaciado) a documentos HTML. El lenguaje CSS1 es legible y escribible por humanos, y expresa los estilos en terminología común de publicación electrónica.

Una de las características fundamentales de CSS es que las hojas de estilo están en cascada. El autor puede adjuntar una hoja de estilo preferida, mientras que el lector puede tener una hoja de estilo personal que se ajuste a sus limitaciones humanas o tecnológicas. Las reglas para la resolución de conflictos entre las diferentes hojas de estilo se definen en esta especificación.

- ✓ CSS Level 2: Incluía funciones más amplias y sofisticadas. Contiene todo de CSS level 1 y soporta el posicionamiento de contenidos, fuentes descargables, disposición de la página, aspectos para la internacionalización, contadores y numeradores automáticos, y algunas características relacionadas con la interfaz del usuario.
- ✓ CSS Level 3: Esta nueva versión viene cargada con grandes novedades y más social que nunca. Viene acompañado de HTML 5 y XHTML 2. Los cambios en estas tecnologías obedecen al fenómeno de la Web 2.0 y a las recomendaciones del creador de la World Wide Web, Tim Berners Lee de llevar a la Web a la máxima expresión del concepto de semántica.

Las ventajas principales en CSS 3 son la inclusión de nuevas propiedades especialmente en cuanto al aspecto gráfico (cabe recordar que CSS también es utilizado para describir estilos para medios de audio) aunque se prevean mejoras sustanciales en otros medios como el de sonido.

La actualización incluirá bordes redondeados, textos con sombras, la capacidad de asignar múltiples fondos, un mejor manejo de tablas incluyendo el estilo zebra, multi-columnas, etc. También se habla del hecho de que con estas nuevas propiedades la carga de la página deberá descender pues el hecho de que muchos de los efectos estén ahora bajo control del navegador hará que los recursos visuales e imágenes que ahora son empleados ya no tengan razón de seguir siendo utilizados.

### 3. Modelo de Objetos

En los años 90 tanto Netscape como Internet Explorer usaban sus propias implementaciones de JavaScript que aseguraban compatibilidad en su navegador, pero no en el ajeno. Esto obligó a tener que diseñar webs para diferentes navegadores, e incluso para diferentes versiones de un mismo navegador. El W3C decidió crear un modelo de documento único: DOM (Document Object Model). Partiendo de la premisa de que los documentos se construyen en torno a un estándar y se manipulan en torno a otro estándar (ECMAScript) la compatibilidad debiera estar asegurada.

Se está hablando de una de las claves de las tecnologías del llamado Web 2.0 (AJAX, XSLT, .NET...), que extienden el patrón **Modelo Vista Controlador** tanto en el lado del servidor como en el cliente, utilizan documentos estructurados (XHTML, XML...) para la capa Modelo e implementan una capa Controlador programada con lenguajes basados a objetos que implementan métodos de acceso al Modelo basados en DOM.

- ✓ DOM, Modelo de objetos de documento Level 1 (Core): Es una API para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes como ECMAScript (Javascript).
- ✓ DOM, Modelo de objetos de documento Level 2 (Core): Este se construye en base al DOM Level 1 Core. El DOM Level 2 Core está compuesto por una serie de interfaces de núcleo para crear y manipular la estructura y contenidos de un documento. El núcleo contiene interfaces especializadas dedicadas al XML.

#### 4. Lenguajes de Script

- ✓ ECMAScript 262 (La versión estándar de JavaScript): Es una especificación de lenguaje de programación publicada por ECMA International. El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation. Actualmente está aceptado como el estándar ISO 16262, este es utilizado por



lenguajes de desarrollo Web para crear aplicaciones interactivas o RIA, con las cuales es posible realizar cambios sobre la misma página sin necesidad de recargarla, obedeciendo a uno de los principios de la Web 2.0.

#### 5. Lenguajes de Presentación adicionales (Markup)

- ✓ MathML 2.0: Es una aplicación XML para describir notación matemática y capturar su estructura y contenido. La meta de MathML es habilitar el servicio, recibimiento, y procesamiento de las matemáticas en la WWW, así como HTML habilita estas funcionalidades para el texto. Estas funcionalidades permiten entonces que el usuario interactúe con la web, permitiéndole ser una web dinámica, de manera que entra a hacer parte de la web de lecto-escritura, parte de la Web 2.0.
- ✓ SVG 1.0: (Scalable Vector Graphics), es un lenguaje para mostrar gráficos de dos dimensiones y aplicaciones gráficas en XML, e imágenes mixtas vectoriales y convencionales. Mejora la vivencia inalámbrica porque permite incluir funciones multimedia exuberantes e interactivas, SVG será el elemento de base para las aplicaciones de móviles de nueva generación.

Resumiendo, los estándares Web dividen un portal, un sitio, una Web, etc. en tres partes, su estructura (XHTML), su presentación (CSS) y su comportamiento (ECMAScript).

Un sitio construido bajo los estándares Web debe ser limpio, basado en CSS, accesible, usable y amigable a los motores de búsqueda

#### **Importancia de XHTML y CSS**

Los últimos estándares se refieren a la utilización de XHTML y CSS. Gracias al uso de XHTML para el contenido y de CSS para la apariencia, se pueden transformar rápidamente un sitio sin importar que se trate de una página Web o miles, realizando cambios en un solo lugar. Los documentos que separan apariencia de contenido usan menos código, pues optimizan su estructura y

funcionalidad. Cumplir con el estándar XHTML abre las puertas para diseñar páginas Web para dispositivos móviles o nuevas tecnologías que vayan surgiendo.

Se pretende conseguir que todos los documentos Web sean compatibles en cualquier navegador. Utilizar XHTML + CSS ayuda a mejorar la manera de diseñar una página, consiguiendo que el código sea más limpio y claro. Los documentos basados en XHTML válido son más relevantes para los motores de búsqueda, contienen mayor información y menos código, por lo que un sitio basado en estándares tendrá una mejor posición dentro de los servicios de la red.

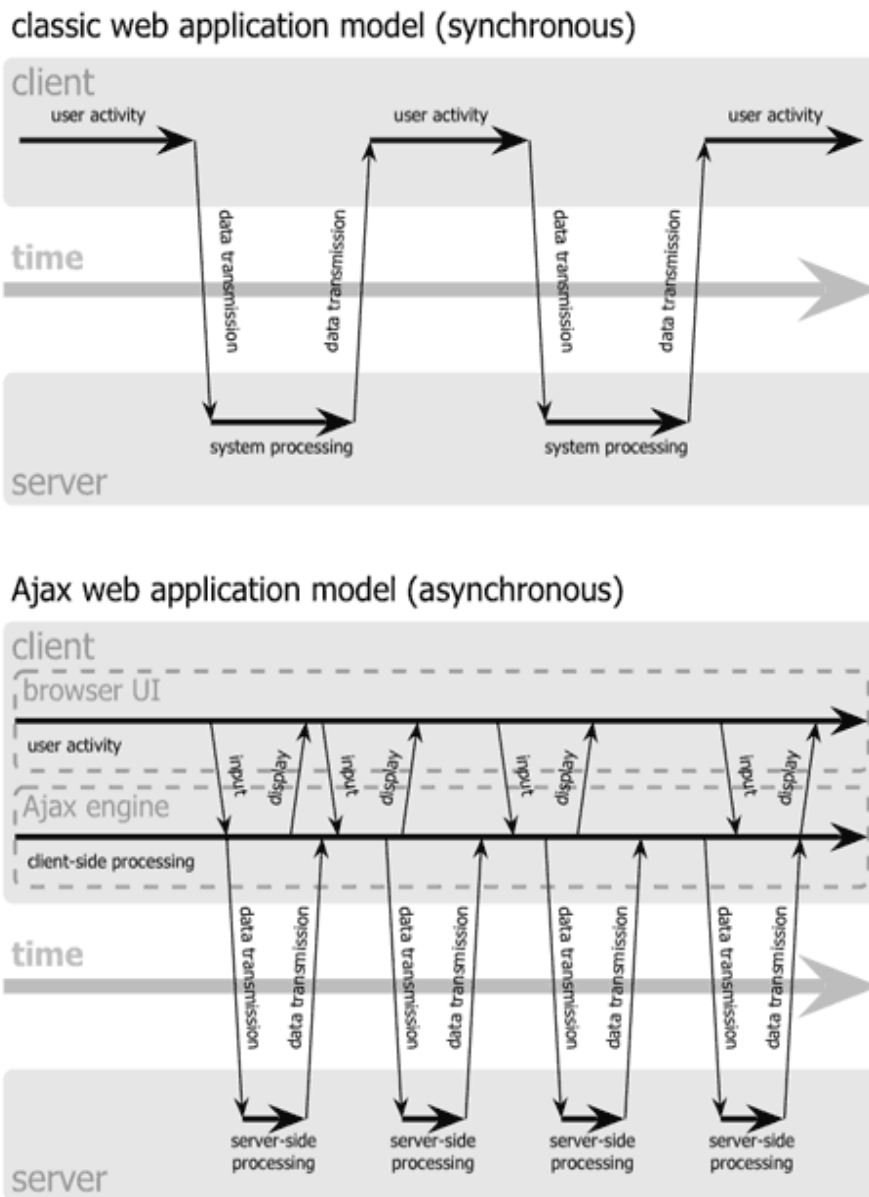
### 2.5 MODELOS

Construir aplicaciones Web 2.0 requiere de modelos ligeros de programación, pues son ensamblables por naturaleza, por ello, necesitan que su arquitectura considere la utilización de estándares tanto para el manejo de datos como para la comunicación; esto asegurará la posibilidad de reutilización de los módulos, por ejemplo para la creación de mashups.

Hay ciertos conceptos técnicos clave en la construcción de aplicaciones Web 2.0, como:

**AJAX (Asynchronous Javascript and XML):** AJAX utiliza tecnologías como Javascript y XML, y permite crear aplicaciones más interactivas en el web, con una interfaz similar al de las aplicaciones del escritorio tradicional, sin requerir características específicas de los navegadores. Asimismo, reduce el tamaño de la información que intercambia, ya que se realizan muchas “micro-peticiones”, por lo cual el flujo de datos global es inferior al normal, lo que aumenta de manera considerable la usabilidad, la velocidad y el tiempo de respuesta de un sitio web.

El modelo de aplicación clásico (síncrono) tiene la desventaja de que las páginas que se envían y reciben, lo hacen por completo, cada vez que se requiere actualización. Esto repercute en un mal aprovechamiento de los canales de comunicación, ya que un alto porcentaje de la información enviada no es necesaria debido a que ya está en el terminal del cliente y no necesita ser refrescada.



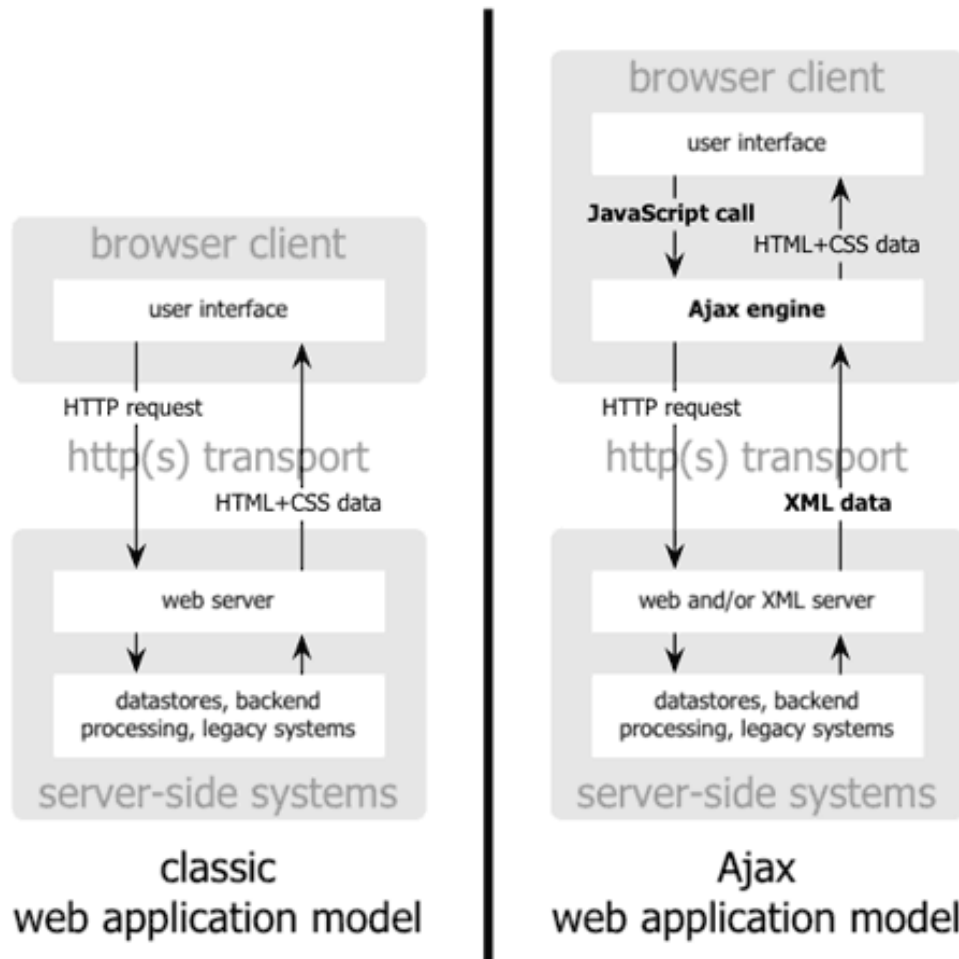
**Figura 4. Modelo síncrono - modelo asíncrono. Tomado de [web] 2.0 aproximación a las nuevas tecnologías de la web.**  
[http://jair.lab.fi.uva.es/~beasanc/PAM/practicaweb1/html/ajax\\_3.html](http://jair.lab.fi.uva.es/~beasanc/PAM/practicaweb1/html/ajax_3.html)

Al momento de usar una tecnología como AJAX se tendrá que descargar todo el código de la aplicación JavaScript durante la carga inicial de la página (lo cual representa una mínima desventaja) y al interactuar con ella, simplemente se hace la petición al servidor de aquella información que se necesita actualizar en el terminal del cliente. Con esto se consigue que una vez hecha la carga inicial, también llamada

## ¿Qué es la Web 2.0?

---

el *client engine* o *Ajax engine*, el tráfico de información cliente-servidor sea menor y por tanto más rápido, lo que repercute en un grado de interacción mayor, como muestra la figura 4.



**Figura 5. Modelo clásico - modelo Ajax. Tomado de [web] 2.0 aproximación a las nuevas tecnologías de la web.**  
[http://jair.lab.fi.uva.es/~beasanc/PAM/practicaweb1/html/ajax\\_3.html](http://jair.lab.fi.uva.es/~beasanc/PAM/practicaweb1/html/ajax_3.html)

Como se puede observar en la figura 5, la interfaz de usuario no se vuelve a recargar. Cualquier acción del usuario que requiera comunicación con el servidor se solucionará mediante una petición asíncrona en la que se suele utilizar XML, el *Ajax engine* analizará los datos recibidos y modificará lo mínimo indispensable de la interfaz. El usuario ahora no pide recargas de nuevo HTML, sino que puede seguir utilizando la aplicación mientras las comunicaciones se solucionan en segundo plano.

El Ajax es un modelo de las nuevas aplicaciones Web, cumple un papel muy importante en estas al momento de disminuir el tiempo de actualización de páginas Web, haciendo solo este proceso en partes donde realmente sea necesario. La Web 2.0 se está poblando cada vez más con aplicaciones basadas en AJAX. Encontrando los ejemplos más representativos en *Google*, que ha sido el mayor promotor de esta tecnología, con aplicaciones como *GMail*, *Google Maps*, *Google Calendar*, o el espectacular *Google Docs*, verdadero procesador de textos en línea.

## 2.6 VENTAJAS Y APLICABILIDADES

Las ventajas encontradas para el desarrollo de aplicaciones de las técnicas, tecnologías y de los estándares en la Web 2.0 son muchas, son estas las que analizan las compañías competidoras al momento de desarrollar sus productos. A continuación se mencionan las más relevantes.

### 2.6

#### 2.6.1 Ventajas y aplicabilidad de las técnicas:

Técnica	Ventaja	Aplicabilidad
<b>CSS</b>	Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.	Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web, con lo que aumenta considerablemente la accesibilidad. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario.

## ¿Qué es la Web 2.0?

---

<b>XHTML</b>	La información, y la forma de presentar la página están claramente separadas. El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.	La aplicación de XHTML tiene como objetivo avanzar en el proyecto del W3C de lograr una Web semántica.
<b>AJAX</b>	Es posible realizar cambios sobre la misma página sin necesidad de recargarla.	Aumenta la interactividad, velocidad y usabilidad en la página.
<b>Java Web Start</b>	Permite arrancar aplicaciones Java que están en un servidor Web de aplicaciones comprobando previamente si el cliente tiene la versión actualizada de dicha aplicación.	Se asegura que una aplicación es distribuida siempre en su última versión.
<b>XUL</b>	Brindar la portabilidad de las interfaces de usuario.	Permite desarrollar aplicaciones multi-plataforma sofisticadas o complejas sin necesidad de herramientas especiales.
<b>RSS/ ATOM</b>	Los archivos RSS contienen un resumen de lo publicado en el sitio Web de origen, condensando la información más solicitada en un solo sitio.	Permitir la distribución de contenidos y noticias de sitios Web.
<b>URL sencilla y con significado</b>	Produce un efecto de confortabilidad en el usuario, que no se siente navegando en una máquina calculadora.	Los contenidos pueden ser copiados con confianza, y mandados por correo electrónico a un amigo, o incluso pueden ser recordadores y teclados, tanto para entrar a una página, como para navegar.

<b>JCC</b>	Facilita la comunicación entre servicios integrados en una misma página Web.	Permite la creación de sitios Web mediante partes de otros sitios, además de aligerar el peso de los interfaces y mejorar notablemente la usabilidad de las interfaces Web.
------------	--	---

**Tabla 3: Ventajas y Aplicabilidades de las Técnicas**

**2.6.2 Ventajas y aplicabilidad de las tecnologías:**

<b>Tecnologías</b>	<b>Ventajas</b>	<b>Aplicabilidad</b>
<b>API</b>	Permite la comunicación entre componentes software.	Estas ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación.
<b>Beta</b>	Los servicios evolucionan constantemente.	Beneficia a los consumidores desarrollando productos a bajo costo y sin versiones que se vuelven obsoletas
<b>Mashup</b>	Usa información de varias fuentes para crear un nuevo servicio en base a ellas	Permite la portabilidad de datos en nuevas aplicaciones Web ya que permite tomar datos de una fuente y utilizarlos en otro contexto

## ¿Qué es la Web 2.0?

---

<b>OPML</b>	Permite listar varias fuentes RSS juntas	Se basa en la filosofía colaborativa de la Web 2.0, en la que permite distribuir, agregar y reunir información en una misma página Web para luego ser
<b>P2P</b>	Permite compartir información entre usuarios de una misma red acabando con la	Se fundamenta en la filosofía de la Web 2.0 de que todos los usuarios de
<b>Permalinks</b>	Permite tener acceso a cualquier información aunque esta ya no este disponible en la pagina principal	Se implementa al direccionar un post, una entrada, o un artículo a un enlace permanente o fijo, es decir una dirección Web particular.
<b>REST</b>	Enfoque para obtener información de un sitio Web	Define un estilo de diseño y arquitectura Web de cómo debería comportarse una aplicación Web de nueva tecnología
<b>SOAP</b>	Implementa el paradigma de la programación orientada a objetos y permite la comunicación utilizando los servicios Web	Permite la lectura de los objetos que intercambian información dentro de una aplicación Web, facilitando el uso de la misma
<b>RSS</b>	Diseñado para la distribución de contenido resumido de lo publicado en un sitio Web origen	Cualquier fuente de información puede ser fraccionada en ítems y un trozo de esta puede ser distribuido utilizando RSS, sin necesidad de publicar un artículo completo en una pagina Web.



<b>Ruby on Rails</b>	Entorno de programación que permite usar menos código de programación en aplicaciones Web	Automatiza las tareas más comunes y permite ciclos cortos para el desarrollo de prototipos
<b>Widget</b>	Da fácil acceso a funciones frecuentemente usadas y provee de información visual	Mini aplicaciones diseñadas para proveer de información o mejorar una aplicación o servicios de un ordenador, o bien cualquier tipo de interacción a través del World Wide Web, lo que hace una Web más actualizada

**Tabla 4: Ventajas y Aplicabilidades de las Tecnologías**

### 2.6.3 Ventajas y aplicabilidad de los estándares:

<b>Estándar</b>	<b>Ventajas</b>	<b>Aplicabilidad</b>
<b>XHTML</b>	La información, y la forma de presentar la página están claramente separadas. El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.	La aplicación de XHTML tiene como objetivo avanzar en el proyecto del W3C de lograr una Web semántica. Código sea más limpio y claro.

<b>XML 1.0</b>	<p>Es extensible, la vieja versión todavía puedan entender el nuevo formato.</p> <p>El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje.</p>	<p>Mejora la compatibilidad entre aplicaciones, lo que conlleva a que más información pueda ser compartida y facilite el surgimiento de las redes sociales.</p>
<b>CSS Level 3</b>	<p>Soporta el posicionamiento de contenidos, fuentes descargables, disposición de la página, aspectos para la internacionalización, contadores y numeradores automáticos, mejoras visuales.</p>	<p>Evita hacer a los archivos demasiado pesados. Se define el estilo de un sitio entero sin necesidad de hacerlo etiqueta por etiqueta. Cualquier cambio hecho a un estilo CSS, se reflejará en todos los elementos que sean referidos a éste, automáticamente.</p>
<b>DOM</b>	<p>Permite acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes como ECMAScript (Javascript).</p>	<p>Se puede manipular toda la información de manera dinámica y transparente para el usuario.</p>
<b>ECMAScript 262</b>	<p>Con este es posible crear aplicaciones interactivas o RIA.</p>	<p>Aumentar la interactividad, velocidad y usabilidad en la página web.</p>

<b>MathML 2.0</b>	Con esta se puede describir notación matemática y capturar su estructura y contenido.	Habilitar el servicio, recibimiento, y procesamiento de las matemáticas en la web, permitiendo entonces que el usuario interactúe aún más con la web.
<b>SVG 1.0</b>	Mejora la vivencia inalámbrica porque permite incluir funciones multimedia exuberantes e interactivas.	SVG será el elemento de base para las aplicaciones de móviles de nueva generación.

**Tabla 5: Ventajas y Aplicabilidades de los estándares**

## **2.7 APLICACION DE LAS PRINCIPALES TÉCNICAS, TECNOLOGÍAS Y ESTÁNDARES**

En el desarrollo de aplicaciones, para la nueva Internet (Web 2.0), se emplean técnicas, tecnologías y estándares para ofrecer a desarrolladores, usuarios y demás participantes de estos desarrollos muchos y mejores beneficios comparados con la Web tradicional. A continuación se muestran las principales aplicabilidades y ejemplos de sus usos en plataformas usadas actualmente a nivel global.

### 2.7.1 APLICACIÓN DE LAS TÉCNICAS

- ✓ **CSS:** La Web actual es algo diferente a lo que fue hace 5 años. La cantidad de técnicas usadas en la Web ha crecido rápidamente y esto es reflejado la manera como los Web sites están siendo ahora creados. En el pasado, el HTML – Que es lenguaje usado para crear las páginas Web – tenía recursos (tags), tanto para crear la estructura como para dar estilo a la página también. Hoy en día la técnica comúnmente usada para dar estilo a las páginas es el CSS o Cascading Style Sheets, que es una extensión del HTML, y está orientada básicamente a dar estilo a las páginas. Antes del CSS, solo se contaba con el HTML para poder dar estilos a cualquier elemento en una página. Como por ejemplo para colocar una letra en negrilla había que usar el tag bold:

`<b> Esto estaría en negrilla </b>`

En HTML son usados otros tipos de Tags que permiten dar estilo a las páginas, pero ellos son limitados en muchas formas. Para superar estas limitaciones y los problemas relacionados con las incompatibilidades de los browsers; los diseñadores algunas veces usan texto en imágenes de tipo GIF o JPEG en lugar del texto HTML. Por las limitaciones en diseño que presentaba HTML, los diseñadores se valían de esta técnica para conseguir la apariencia deseada.

Esto resultaría en gran cantidad de imágenes en las páginas, imágenes que no serían usadas para mostrar un logo o una foto, si no que serían usadas en su lugar para reemplazar cualquier texto el cual debería ser creado con HTML. Lo positivo acerca de este enfoque es que se puede lograr el estilo exacto que se está buscando, pero esto puede resultar contraproducente puesto que mostrar textos usando imágenes es un proceso que consume tiempo. Se tendría que estar cambiando entre HTML y el editor de imágenes cada vez que se desee cambiar el texto.

En ocasiones el proceso de arreglar un error requiere de muchos intercambios entre el editor HTML y el editor de imágenes. Si de otra parte el texto es creado en solo HTML, los cambios tomarán solo una fracción del tiempo.

Otro efecto negativo de usar imágenes para el texto en el cuerpo del documento es que ellas empiezan rápidamente a abultar la página en términos de kilobytes. Usted

deberá siempre tratar al máximo de mantener sus páginas tan “livianas” en este aspecto como sea posible, y de esta forma el usuario final podrá “download” las páginas más rápidamente. La regla general es que si una página Web no carga en 10 segundos, se perderán la mayoría de los visitantes. Aunque actualmente mucha gente tiene Internet a alta velocidad, es aconsejable de que las páginas sean al rededor de 50-60k.

CSS es muy poderoso y permite dar estilo a una página en formas que no fueron posibles antes. Hasta hace poco (Algo así como un año o dos) las implementaciones en CSS eran irregulares e inconsistentes, por esto mucha gente evadió su uso, o lo usó solo cuando era realmente necesario. Aun hay casos en que se continúa utilizando una mezcla entre la vieja manera de hacer las cosas y lo nuevo.

Generalmente se usan tablas para la estructura principal de la página y algunas imágenes para crear los encabezados de las páginas, pero nunca para el texto en el cuerpo de la página. Se puede estar seguro cuando se usa el CSS1 y CSS2, que son las dos primeras especificaciones de CSS y siendo justos la mayoría de las cosas trabajaran propiamente en los browsers. Lo anterior teniendo en cuenta que Internet Explorer tiene entre 95% y el 97% del mercado y que soporta el CSS estándar.

Cuando la Web empezó y llegó al público en general, y los browsers empezaron a soportar sofisticadas presentaciones e imágenes, la gente empezó a desenfrenarse en términos de diseño. Esto ocasionó una gran diversidad de estilos y Websites que aun persiste hoy en día y que son una evidencia de ello.

La Web se está moviendo lentamente hacia el objetivo de brindar mejor experiencia para el usuario final y desde el punto de vista del programador que se haga más fácil crear su software. CSS empieza por hacer mas consistente el dar estilo a las páginas mientras experimentados desarrolladores Web han desarrollado estructuras, que ellos ya saben trabajan bien, ya que las han usado consistentemente en todos sus proyectos.

La apariencia de las páginas tiene que ver con colores y fonts; la estructura solamente hace fácil a los usuarios navegar en la página y en el Website. Una buena imagen propiamente localizada puede hacer toda la diferencia entre una página con buena apariencia y otra con mala apariencia.

El CSS Table Gallery es un repositorio de diseños gráficos de tablas, cuyo fin es demostrar que es posible separar el contenido de la presentación (texto de imagen).

## ¿Qué es la Web 2.0?

---

Los diseñadores aportan hojas de estilo CSS a través de las cuales la apariencia de una tabla varía, de acuerdo al estilo elegido por el usuario. El desafío es no tocar el código HTML, sólo modificar el CSS.

El objetivo del trabajo en CSS1 ha sido crear un mecanismo simple de hojas de estilo para documentos HTML. La especificación actual representa un equilibrio entre la simplicidad requerida para utilizar hojas de estilo en la Web, y la presión por parte de los autores que demandan mayor y mejor control visual. CSS1 ofrece:

- Sustitución de etiquetas visuales: las extensiones HTML, p.ej. 'CENTER', 'FONT' y 'SPACER' son fácilmente reemplazadas por hojas de estilo.
- Código más limpio: en lugar de usar elementos 'FONT' para lograr el estilo de versalitas, es suficiente una declaración en la hoja de estilos. Compárese el código visual:

```
<H1>H<FONT SIZE=-1>EADLINE</FONT></H1>
```

Con la hoja de estilos:

```
H1 {font-style: small-caps}
```

```
<H1>Headline</H1>
```

- Varios niveles de integración: las reglas de estilo de CSS1 pueden obtenerse de hojas de estilo externas, incluirse en elementos 'STYLE' o ponerse dentro de atributos 'STYLE'. La última opción ofrece un método sencillo de transición desde extensiones HTML.
- Nuevos efectos: se han añadido algunos efectos visuales nuevos para dar a los usuarios "juguetes nuevos". Los pseudo-elementos tipográficos y los valores adicionales de la propiedad 'background' pertenecen a esta categoría.
- Escalabilidad: CSS1 será útil en equipos que vayan desde terminales de texto a estaciones de trabajo en color con altas resoluciones. Los

autores podrán escribir una sola hoja de estilos y estar razonablemente seguros de que el estilo buscado saldrá de la mejor manera posible.

- ✓ **XHTML:** esta técnica al estar orientada al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en la parte del código. Entre estos requisitos básicos se puede mencionar una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados, etc.

### Ejemplos

A continuación se pueden ver algunos ejemplos de los aspectos más importantes a tener en cuenta a la hora de utilizar XHTML.

- Los documentos deben estar *bien formateados*: un formato correcto en un documento XHTML es muy importante. Esto quiere decir que todos los elementos deben tener etiquetas de cierre, deben estar escritos de una forma determinada y además todos los elementos deben estar anidados correctamente. A continuación una muestra de un código de elementos anidados:

```
<p>Ejemplo de elementos bien <em>anidados</em>.</p>
```

```
<p>Ejemplo de elementos mal <em>anidados</p>.</em>
```

- Los nombres de atributos y elementos deben ir en minúsculas: los elementos y los atributos deben ir en minúsculas para todos los elementos HTML y los nombres de atributos. Esto es importante ya que XML interpreta las mayúsculas y las minúsculas de forma diferente.

```
<body>Ejemplo correcto</body>
```

```
<BODY>Ejemplo incorrecto</BODY>
```

- Los elementos que no estén vacíos necesitan etiquetas de cierre: al finalizar una frase, un texto o cualquier tipo de escrito se deberá cerrar la etiqueta correctamente, como se muestra en los siguientes ejemplos:

```
<p>Ejemplo correcto.</p>
```

```
<p>Ejemplo correcto.</p>
```

```
<p>Ejemplo incorrecto.<p>Ejemplo incorrecto.</p>
```

- Los valores de las etiquetas deben ir siempre entre comillas: todos los valores de los atributos deben ir entre comillas, incluso aquellos que sean numéricos.

```
<table rows="3">
```

```
<table rows=3> ejemplo incorrecto
```

- Existen varias versiones de XHTML (1.0, 1.1, Básico, etc.): para utilizar una versión concreta, se debe incluir antes del elemento HTML del código de la página Web, la Definición del Tipo de Documento (DTD) que se pretende utilizar. Por ejemplo, un documento en XHTML 1.1 podría especificarse de la siguiente forma:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE
```

```
html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
```

```
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
```

```
<head>
```

```
<title>Título</title>
```

```
</head>
```

```
<body>
```



</body>

</html>

- ✓ **Java Web Start:** esta técnica también se presenta en aplicaciones Web 2.0. Usualmente cuando se hace necesario ejecutar aplicaciones que no se encuentran instaladas en una maquina, se descarga de algún sitio Web disponible, se instala en dicho equipo y por último se ejecuta. Java Web Start lo que pretende es cambiar este proceso de modo que el usuario lo único que tiene que hacer para lanzar una aplicación sea simplemente hacer clic en un enlace de su navegador, a partir de ese momento, Java Web Start comprueba en el servidor de Web si existe una versión nueva de la aplicación y, si existe, la descarga y la ejecuta automáticamente logrando así que todo este proceso se realice de una manera transparente.

Aunque en ocasiones se tiende a confundir Java Web Start con un Applet por su parecido en el comportamiento, estas no se relacionan ya que Java Web Start solo utiliza un *browser* para que el usuario pueda hacer uso de las aplicaciones. Una vez que el usuario hace clic en un enlace de una aplicación, ésta se ejecuta en la máquina virtual del cliente como cualquier otra aplicación.

Java Web Start no es un componente del navegador Web, es solo una aplicación independiente por lo que no requiere del navegador para su funcionamiento. Una vez que el usuario hace clic en un enlace para ejecutar una aplicación, puede continuar navegando o cerrar el navegador sin que esto interfiera en el funcionamiento de la aplicación que ha sido lanzada. Además, Java Web Start va guardando en una caché interna las aplicaciones que va ejecutando el usuario, de modo que éste pueda lanzarlas posteriormente sin la

## ¿Qué es la Web 2.0?

---

necesidad de abrir el navegador o incluso ejecutarlas localmente sin conectarse a ninguna red.

Las aplicaciones Java Web Start siguen el modelo de seguridad de la plataforma Java 2 por lo que la integridad de los datos que se obtiene a través de la red está garantizada. Comúnmente las aplicaciones que se ejecuten han de estar debidamente firmadas y se requiere siempre que el usuario autorice su ejecución.

Las ventajas de las aplicabilidades que ofrece tanto a los desarrolladores de las mismas como a los usuarios son muchas y muy importantes:

- **Transparencia:** El usuario no necesita pasar por un proceso traumático de descarga e instalación de la aplicación para poder ejecutarla. Únicamente tiene que pinchar un enlace en su navegador y la aplicación se descarga, se instala y se ejecuta de manera automática. Además, Java Web Start se encarga de crear los accesos directos correspondientes en el escritorio y menú de inicio del usuario.
- **Mantenibilidad:** Para los desarrolladores y administradores de sistema, Java Web Start es una bendición. Ahora ya no es necesario copiar la misma aplicación a todos los usuarios de una red cada vez que se realiza una pequeña modificación en la misma, sino que con actualizarla en el servidor Web es suficiente para que los usuarios puedan utilizar la última versión de la misma.
- **Control de versiones:** Java Web Start se encarga automáticamente de realizar el control de versiones de las aplicaciones. Antes de ejecutar una aplicación, Java Web Start comprueba en el servidor Web que no exista una versión más avanzada de la misma, en cuyo caso actualizará la vieja versión por la nueva automáticamente. Esto beneficia tanto a los usuarios que siempre ejecutan la última versión de su software, como a los desarrolladores que no tienen necesidad de distribuir las nuevas versiones a los usuarios o crear algún sistema interno de control de versiones.

- Independencia del servidor Web y del navegador: Java Web Start puede funcionar en cualquier servidor Web tan sólo añadiendo el tipo MIME correspondiente a los ficheros con extensión *.jnlp*, por otra parte, también funcionará en cualquier navegador aunque en algunos habrá que configurar el programa asociado a los ficheros con dicha extensión.
- Independencia del sistema operativo: Aunque Java Web Start no está disponible para todos los sistemas operativos para los que la plataforma Java se encuentra disponible, *OpenJNLP* que como se dijo está escrito en Java y que es una iniciativa *Open Source* si que es totalmente independiente del sistema operativo.
- Automatiza la gestión de *JREs*: Cada aplicación puede decidir que *JRE* quiere utilizar para ejecutarse, es más, si ese *JRE* no existiese en el equipo del cliente, Java Web Start se encarga automáticamente de su descarga e instalación en el sistema.
- Transparencia al desarrollador: No es necesario modificar las aplicaciones existentes para que aprovechen esta tecnología. Para hacer una vieja aplicación compatible con Java Web Start, tan sólo hay que crear un pequeño descriptor XML con las características de la aplicación y colocarla en un servidor Web. Las aplicaciones pueden seguir ejecutándose del modo tradicional sin ningún problema.
- Ejecución local de las aplicaciones: Java Web Start a diferencia de tecnologías como JSP/Servlets no necesita la red para ejecutar las aplicaciones. La red tan sólo es un medio para obtener dichas aplicaciones y sus actualizaciones. Una vez descargada una aplicación, ésta se ejecuta de manera local y tan sólo accede a la red si lo necesita para su funcionamiento.

La técnica Java Web Start proporciona a las aplicaciones de Internet las mismas características y ventajas descritas para las aplicaciones basadas en Web. La tecnología Java Web Start es una solución de distribución de aplicaciones para la Web. El uso de una aplicación con todas las características en lugar de un cliente basado en Web tiene diversas ventajas:

## ¿Qué es la Web 2.0?

---

- Interfaz de usuario altamente interactiva, comparable a la de las aplicaciones tradicionales, como procesadores de texto u hojas de cálculo.
- Menor necesidad de ancho de banda. Una aplicación no tiene por qué conectarse con el servidor de Web con cada clic, puede guardar en la antememoria la información ya descargada. De esta forma, ofrece una mejor interactividad en conexiones lentas.
- Admite el uso fuera de línea.

Por supuesto, seguirá teniendo que descargar la aplicación la primera vez que la utilice. Por consiguiente, una aplicación basada en HTML tiene un *coste de primera activación*. Generalmente, una página Web aparece en cuestión de segundos. Una aplicación basada en la tecnología Java™ requiere un tiempo de descarga de varios minutos en una conexión de módem habitual. Java Web Start guarda en la antememoria todos los archivos descargados en el sistema. De esta forma, aunque el coste de primera activación es más alto para las aplicaciones que para las páginas HTML, las siguientes veces la aplicación se ejecutará de forma casi instantánea, puesto que todos los recursos necesarios estarán disponibles localmente.

- ✓ **XUL:** el objetivo Inicial de la creación de la técnica XUL fue para desarrollar productos de Mozilla (navegador y cliente de e-mail, entre otros) rápida y fácilmente. Al ser un lenguaje basado en XML, contiene todas las características disponibles para XML y sus mismas ventajas.

Las aplicaciones XUL se pueden personalizar con textos, gráficos y otros componentes, por lo que pueden ser rápidamente localizadas en diferentes mercados. Los desarrolladores Web que estén familiarizados con HTML dinámico (DHTML) podrán aprender XUL rápidamente e incorporarlo en el desarrollo de sus aplicaciones.

La mayoría de las aplicaciones son desarrolladas utilizando las características propias de una plataforma específica, haciendo que el desarrollo de software multi-plataforma no sea costoso y no consuma tiempo. Es por esto que en el pasado se han desarrollado soluciones multi-plataforma que brinden dicha portabilidad (como por ejemplo Java), y que este aspecto sea su característica

más fuerte. Con XUL, una interfaz puede ser implementada y modificada fácil y rápidamente.

Para desarrollar aplicaciones portables, complejas y de fácil manejo se puede implementar XUL en lugar de HTML, ya que a diferencia de este último, XUL brinda un mejor abanico de herramientas para crear menús, paneles, barras de herramientas, entre otras. Gracias a esto, no será necesario utilizar un lenguaje de programación propietario o incluir un gran código Java Script para manejar el comportamiento de la interfaz de usuario.

Hay cuatro tipos distintos de aplicaciones, según como las mismas son creadas:

- Extensiones de Firefox: Son barras de herramientas, menús u otros documentos XUL que agregan funcionalidades al Firefox. Para hacer esto, se usa un elemento definido por XUL llamado "Overlay", este elemento permite incorporar documentos al navegador mismo.
- Aplicaciones independientes: Estas aplicaciones son creadas mediante XULRunner, el cual es una versión de la plataforma Mozilla que permite crear aplicaciones XUL independientes. No es necesario el navegador para ejecutar estas aplicaciones, tienen su propio ejecutable.
- Paquete XUL: Es un intermedio entre los dos tipos anteriores, es instalado como una extensión, pero actúa como una aplicación separada del navegador. Este tipo de aplicaciones son creadas cuando no se quiere utilizar una aplicación XULRunner completa.
- Aplicaciones XUL Remotas: Son aplicaciones XUL que se encuentran en un Servidor Web y son ejecutadas remotamente como cualquier otra página Web.

En aplicaciones para Mozilla, XUL se comporta de manera similar a HTML y a otros tipos de contenido. Cuando el usuario escribe la URL de una página HTML en el campo de dirección del navegador, éste busca el sitio Web y descarga su contenido. El motor de presentación de Mozilla toma el contenido

## ¿Qué es la Web 2.0?

---

en el formato HTML y lo transforma en un árbol de documento. Este árbol se convierte en un conjunto de objetos que pueden mostrarse en la pantalla. Los estilos CSS, imágenes y otras tecnologías son usados para controlar la presentación. El procesamiento de XUL es muy similar.

Para todo tipo de documento de Mozilla, bien sean HTML o XUL son procesados por el mismo código, de manera que las mismas propiedades CSS pueden usarse para definir el estilo tanto del HTML como de XUL, y muchas de las características pueden compartirse entre ambos. Sin embargo, hay características que son específicas del HTML como son los formularios, y otras que son específicas de XUL como son los overlays (revestimientos). Ya que XUL y HTML se procesan de la misma forma, se pueden cargar desde el sistema de archivos local del usuario, desde una página Web, desde una extensión del navegador o desde una aplicación XULRunner.

El contenido de fuentes remotas, como por ejemplo: `http://localhost/~username/`, sin importar si es HTML, XUL o cualquier otro tipo de documento, está limitado en la clase de operaciones que pueden realizar, por razones de seguridad. Por este motivo, Mozilla proporciona un método para instalar contenido en forma local y registrar los archivos instalados para que formen parte del sistema chrome. Esto permite usar una URL especial llamada la URL `chrome://`. Al acceder a un archivo usando la URL chrome, éstos reciben privilegios para acceder a archivos locales, preferencias, marcadores de página y ejecutar otras operaciones privilegiadas. Obviamente, las páginas Web no obtienen estos privilegios, a menos que estén firmadas con un certificado digital y el usuario dé el permiso para ejecutar estas operaciones.

Las aplicaciones XUL autónomas (que no necesitan el navegador) pueden incluir código XUL de la misma forma, pero este código se incluirá como parte de la instalación, en lugar de tener que instalarse de forma separada como una extensión. Sin embargo, este código XUL debe ser registrado en el sistema chrome de tal forma que las aplicaciones puedan mostrar la UI.

Vale la pena anotar que el navegador Mozilla realmente es un conjunto de paquetes que contienen archivos XUL, JavaScript y hojas de estilo. Estos

paquetes son accesibles usando una URL chrome, tienen privilegios ampliados y trabajan como cualquier otro paquete. Por supuesto, el navegador es más grande y más sofisticado que la mayoría de las extensiones. Firefox, Thunderbird y muchos otros componentes también están escritos en XUL y se puede acceder a ellos usando la URL chrome. Se puede examinar estos paquetes mirando el directorio chrome donde Firefox o cualquier otra aplicación XUL esté instalada.

- ✓ **RSS:** a pesar de que los RSS son usados generalmente en los campos de la publicación y suscripción a titulares de noticias y de alimentadores blog, muchas otras útiles e innovadoras aplicaciones de RSS han estado surgiendo en toda la red.

RSS es una técnica de impacto innovador que puede ser empleada a una gran variedad de aplicaciones posibles, bien sea para negocios, marketing, inteligencia competitiva o protección de seguridad, video entretenimientos o rastrear descargas de su producto más nuevo. Aquí hay algunas de las más interesantes:

- Monitorear Newsgroups/Usenet: "Grupos de Google" es una comunidad y servicio de grupo discusión online gratuito que ofrece el más completo archivo de la Web de los envíos Usenet. Los Grupos de Google pueden ser monitoreados utilizando alimentadores RSS. Para monitorear un grupo de noticias específico simplemente se agrega "/feed/msgs.xml" al final del URL del Grupo de Google. Se agrega el nuevo URL al lector del grupo de noticias y se recibirán actualizaciones cada vez que aparezca un nuevo envío.

Otros servicios orientados a la comunidad online y grupos de colaboración online que soportan totalmente RSS incluyen:

- YahooGroups  
<http://www.yahogroups.com/>
- SocialText  
<http://www.socialtext.net/>
- Near-Time  
<http://www.near-time.com/>

## ¿Qué es la Web 2.0?

---

- Calendarios Compartidos: se puede construir calendarios para uso público o privado y recibir actualizaciones de calendario vía alimentadores RSS. Los calendarios compartidos pueden ser creados como recordatorios personales para pagar cuentas o eventos sociales. Los calendarios que puede ser utilizado para grupos específicos como escuelas, eventos deportivos o de la industria. Un calendario compartido puede ser totalmente sindicado y compartido utilizando alimentadores RSS. A continuación, ejemplos de calendarios RSS:

- CalendarHub

<http://www.calendarhub.com>

Example Feed - <http://www.calendarhub.com/rss/21882/month>

- RSS Calendar

<http://www.rsscalendar.com/>

- Zimbra

<http://www.zimbra.com/>

- Clima: Ingresando un código postal se puede obtener información del clima para una región específica. Haciendo clic en cualquier icono RSS y se agregara el alimentador al lector RSS. Cada día, a la hora que se especifique, se recibirán reportes del clima locales regionales a medida que se emitan. Los alimentadores RSS pueden ser personalizados para vivir condiciones del tiempo local, radar Doppler local, tendencias de polen, pronósticos regionales en video con comentarios expertos y detalles del pronóstico extendido. Algunos ejemplos son:

- Yahoo Weather

<http://weather.yahoo.com/rss/>

- Weather.com

<http://www.weather.com>

- RSS Weather

<http://www.rssweather.com>



- Clasificados (Casas/Empleos): se reciben notificaciones cuando las vacantes estan disponibles. Los alimentadores RSS puede ser creados por regiones, rangos de salarios o puestos.
  - Craigslist  
<http://www.craigslist.org/>  
[Craigslist Example Feed](#)
  - RSSJobs  
<http://www.rssjobs.com>
  - CareerChannel  
<http://www.careermag.com/MKT/RSS/>
- Ego Searches: Un egosurfer es alguien que surfea la Internet en búsqueda de su propio nombre, para ver, si hay artículos sobre el. Ego búsquedas son búsquedas gratuitas y simples diseñadas para monitorear blogs y portales de noticias en búsqueda de menciones de su compañía, productos, competidores u otras palabras claves específicas. Realizar ego búsquedas, no sólo le permite estar informado sino también mantener una ventaja estratégica sobre compañías competidoras. Un número de nuevos servicios Internet están disponibles libremente que hacen que esas 'ego búsquedas' sean sencillas y sin complicaciones.

Muchos de los servicios permiten crear un alimentador basado en palabras claves. Cada vez que nueva información conteniendo la palabra clave aparece en los recursos buscados, el alimentador RSS se actualiza.

- Egosurf  
<http://www.egosurf.org/>
- Monitor This:  
<http://uckan.info/depot/monitorthis/>
- Technorati Watchlist  
<http://www.technorati.com>

## ¿Qué es la Web 2.0?

---

Seguimiento de Envíos: para recibir notificaciones vía alimentadores RSS cuando los paquetes son enviados alimentadores RSS personalizado puede ser creados para monitorear el progreso del envío. Rastrea el estado de los paquetes enviados a través de US Postal Service, UPS, DHL o Federal Express.

- Simple Tracking

<http://www.simpletracking.com/>

- Track2RSS Project

<http://track2rss.sf.net/>

- PackTrack

<http://www.packtrack.com/>

- Ofertas Al Rojo Vivo, Cupones Y Especiales: para todos los que aman una ganga, hay alimentadores RSS que promocionan descuentos que dependen del factor tiempo u ofertas se están haciendo cada vez más populares. Los restaurantes envían almuerzos especiales mientras que las compañías de viajes envían ofertas de último minuto a potenciales viajeros utilizando alimentadores RSS.

- Coupons.com

<http://www.coupons.com>

- DealNews

<http://dealnews.com/rss.html>

- HotDealsClub

<http://www.hotdealsclub.com/>

- [Sample Restaurants Feed](#)

[Sample Travel Feed](#)

- Monitoreo De Remates: Items en eBay o en otras principales casas de remates online dependen del factor tiempo y pocos individuos tienen tiempo suficiente para hacer controles diarios de productos específicos que están siendo rematados. Se puede crear un alimentador RSS utilizando palabras claves para monitorear productos a medida que se ponen disponibles para remate.

- Bidding Tools  
<http://www.freebiddingtools.com>  
[Example Feeds](#)
  - AuctionMonitor  
<http://www.auctionmonitor.net/>
  - RSSauction  
<http://www.rssauction.com/>
- Vigilancia De Precios: La vigilancia de precios le permite rastrear y monitorear cambios de precio de productos y mercados específicos. Todavía en su infancia este nicho de mercado ciertamente se convertirá en algo de mucha demanda.
    - RSStalker  
<http://www.rsstalker.com/>
    - FuelWatch  
<http://www.fuelwatch.wa.gov.au/...>
- Podcasting y Podcatching: Podcasting se trata de permitir, escuchar programas de mayor favoritismo de audio y video donde sea, cuando sea y con el dispositivo que quiera.
    - iTunes  
<http://www.apple.com/itunes/>
    - Odeo  
<http://www.odeo.com/>
    - ZiePod  
<http://www.ziepod.com/>
    - FireAnt  
<http://fireant.tv/download>

## ¿Qué es la Web 2.0?

---

- Edición de Video: Gracias a RSS de la gente común la edición de video colaborativo pronto puede ser una realidad para muchos. Kent Bye ha estado trabajando en esta idea como también los muchachos detrás de algunos de los más actualizados de los nuevos sitios de edición de video que han sido lanzados recientemente. RSS es un componente clave de todas estas herramientas que ayudan a aquellos que están trabajando sobre un tema o un proyecto específico, a mantenerse actualizados con los nuevos clips enviados para que trabajen con ellos.
  - Jumpcut  
<http://www.jumpcut.com/>
  - Eyespot  
<http://www.eyespot.com/>
- Newsmastering: Newsmastering es una nueva aplicación emergente en la cual un gran número de alimentadores RSS, aumentado por filtros específicos son combinado para crear alimentadores información temática de alto valor. Puede ser llamados radares o cuando cubren contenido de noticias newsradars.
  - MySyndicaat  
<http://www.mysyndicaat.com/>
  - Newsgator  
<http://www.newsgator.com/hostedSolutions.aspx>

RSS todavía es muy joven. Tanto editores como consumidores están buscando activamente la manera de sacar provecho de esta singular y poderosa tecnología de envío de contenido y de los usos adicionales de los alimentadores RSS que están brotando diariamente.

RSS pronto puede convertirse en una tecnología invisible accionando centenares, sino miles, de diferentes aplicaciones y usos.

- ✓ **JCC:** generalmente se presentaba que las páginas de servicios y aplicaciones disponibles en la Web que ofrecían acceso a la funcionalidad o información de distintos servidores, o bien ‘incrustaban’

trozos de otras páginas/aplicaciones sin interactuar con ellos (banners estáticos, portlets, etc.), o bien abordaban la integración mediante la comunicación directa de los servidores, es decir, el servidor de la aplicación principal establece una comunicación M-M con los servidores de cada servicio que se integra para recoger información (esto requiere protocolos de comunicación y un tráfico de red M-M adicional) y crea toda la Interfaz de Usuario de manera unificada, pudiendo incluir incluso la de los servicios que se están integrando. De esta forma, se vuelven a crear unas Interfaces de Usuario que posiblemente ya existen, lo que implica una reinención continua de la rueda.

Si se quiere ahorrar tiempo, reutilizando servicios en un navegador, debe existir un mecanismo que permita la comunicación y el intercambio de información de los servicios directamente en el navegador (parte cliente), sin necesidad de una comunicación entre servidores. Este mecanismo de comunicación es lo que resuelve JCC:

- La aplicación reutiliza de forma directa la Interfaz de Usuario (IU) de cada servicio integrado. El usuario recibe de forma transparente una IU compuesta por partes provenientes de distintos servicios. Ya no es prioritaria (ni siquiera necesaria) una comunicación M-M entre servidores. Se disminuye el tráfico de red originado por los protocolos M-M.
- Los servicios se comunican entre sí directamente en el lado del cliente. Se traslada al cliente parte de la potencia de cálculo, liberando recursos en los servidores.

En JCC la comunicación entre servicios integrados en una misma página Web se resuelve utilizando objetos JSI (JavaScript Shared Interfaces). Estos objetos son una modelización conceptual de APIs programadas en Javascript. Cada servicio implementa un API de este tipo (un JSI) publicando las acciones/operaciones que permite que realicen en el componente que controla otros componentes (controlados a su vez por otros JSI) que se sindicarán.

Estas JSI, combinadas con la filosofía ya consolidada de despliegue de Web Services y la aplicación de las últimas técnicas respecto a creación de Aplicaciones Web como AJAX, REST, XML-RPC, SOAP, etc., dotan a las aplicaciones y servicios disponibles en la red de una potencia hasta ahora desconocida, aligerando el peso de los interfaces y mejorando notablemente la usabilidad de los interfaces Web. Teniendo en cuenta que el uso de JCC potenciaría el concepto de creación de sitios Web mediante partes de otros sitios (sindicación de contenidos), se podría decir que estaría encuadrada dentro de las nuevas tendencias de la Web 2.0.

De la misma manera que *Google* es uno de los grandes responsables de la popularización de AJAX, ofrece APIS Javascript que permiten la utilización de los controles de Google Maps. También *Yahoo!* publica APIs de este tipo en alguno de sus servicios. Ambos casos suponen la utilización de objetos JSI.

La aplicación de JCC proporcionará una serie de beneficios adicionales al evidente de compartimiento de interfaces Web interactivos entre distintas aplicaciones, como: reducir la prioridad de la comunicación M-M entre servidores, reducir el tráfico en la red, reducir la necesidad de protocolos adicionales, potenciar el cálculo en el cliente para liberar servidores y liberar recursos. Aumentar la potencia de la reutilización, el nivel de abstracción o la transparencia al integrar servicios de terceros disminuirán las diferencias entre los distintos sites dotando a cualquiera que utilice JSIs de una potencia de integración de funcionalidad en sus sites hasta ahora desconocida, incidiendo además en una disminución de costes consecuencia de la reutilización de componentes.

### **2.7.2 APLICACION DE LAS TECNOLOGÍAS**

#### **API:**

En Internet, una API permite que los sitios web se ofrezcan servicios entre ellos mismos, a través de llamadas a funciones documentadas y publicadas, facilitando de esta manera el “mash-up” o mezcla de servicios. Por ejemplo, actualmente es muy fácil que desde un blog personal se puedan publicar

noticias del sitio de un famoso periódico, mezcladas con fotos que ya están alojadas en un sitio de fotografías, a través de llamadas a la API de estos dos servicios.

### **Detalles técnicos**

Un API es una serie de reglas que se usan para escribir funciones o hacer llamados a subrutinas y acceder a otras funciones en una librería. Los programas que usan estas reglas o funciones en sus llamadas API pueden comunicarse con cualquiera que use dicha API.

Las API se comunican con el proveedor a través de diálogos para obtener o actualizar información en el mismo, entre ellos se encuentran:

- Acceso a bases de datos
- Comunicación cliente/servidor
- Comunicación peer-to-peer
- Comunicación en tiempo real
- Event-driven (orientada a eventos)
- Store and forward
- Procesamiento de transacciones

Una API puede llevar a cabo varias funciones como recuperación de errores, traducción de datos, seguridad, manejo de colas y nomenclatura con una interfase fácil de asimilar, que comprende acciones y comandos simples pero con muchas opciones.

Para hacer una llamada a una API, el programa debe invocar a una función tipo “send”, especificando parámetros para el nombre de destino, indicadores de datos y opciones de confirmación.

Una vez que se llama a la API, esta toma la información y hace que todo el trabajo de comunicación sea transparente para la aplicación. Hay cuatro tipos de APIs que posibilitan el intercambio de datos entre diferentes aplicaciones de software en plataformas individuales o distribuidas:

- **Remote Procedure Calls (RPCs) o Llamadas a Procedimientos Remotos.** Con estas, los programas pueden comunicarse a través de procedimientos (o tareas) que actúan en buffers compartidos de información.
- **Standard Query Language (SQL).** Este lenguaje permite el intercambio de los datos entre aplicaciones, accediendo a bases de datos comunes.
- **Transferencia de archivos.** Habilita el intercambio de datos enviando archivos formateados entre aplicaciones.
- **Envío de mensajes.** Provee el intercambio de datos por comunicaciones de interprogramación directa, a través de pequeños mensajes formateados entre aplicaciones emparejadas.

Las APIs pueden desarrollarse para cualquier plataforma y sistema operativo o para ambos al tiempo. Una de las aplicabilidades del desarrollo de APIs es para usarlas como forma de acceder a servicios web, las más conocidas son:

- Google Search
- Flickr
- Del.icio.us
- Amazon
- Google Maps

Flickr se ha convertido en un modelo de aplicaciones web 2.0 y como tal, comparte su API con todos los usuarios que quieran emplear sus servicios en su propio dominio. De esta manera, reutilizando la **API de flickr**, se pueden crear muchísimas aplicaciones interesantes: Galerías, Mashups, Flogs y más.

Para realizar una acción usando la API de Flickr, debes seleccionar una convención de llamada, enviar una solicitud a su extremo y especificar un método y algunos argumentos, y recibirás una respuesta con formato.



Todos los formatos de solicitud, enumerados en la página de índice de API, toman una lista de parámetros con nombre.

El parámetro OBLIGATORIO **method** se emplea para especificar el método de llamada.

El parámetro OBLIGATORIO **api\_key** se utiliza para especificar tu clave API.

El parámetro opcional **format** se aplica para especificar un formato de respuesta.

Los argumentos, las respuestas y los códigos de error para cada método están enumerados en la página de especificaciones del método.

La API de Flickr expone los identificadores para usuarios, fotos, álbumes de fotos y otros objetos con identificación única. El formato de los ID. puede cambiar con el tiempo; por lo que, si te basas en el formato actual, podrías tener problemas en el futuro.

### **BETA:**

Las aplicaciones “Beta Perpetua”, como ya se había mencionado en los principios de la Web 2.0, comparten la posibilidad de acceder on-line a la función que ofrecen, en lugar de hacerlo a partir de la instalación de un programa en el disco duro.

Gracias a estas características los usuarios se suman a la producción de contenidos, convirtiéndose en codesarrolladores de la web. La web 2.0 propone un cambio de dirección: Es la web la que se acerca al usuario y no el usuario a la web.

El software pasa a ser entendido como servicio en lugar de como producto. Este evoluciona constantemente, día a día, para seguir funcionando, es decir se transforma en una versión beta perpetua (Flickr reconoció que despliega nuevas implementaciones cada media hora).

Tomando a Flickr como ejemplo, esta aplicación hace de los datos privados, públicos y además construye comunidades, redes sociales. Aunque ofrece

## ¿Qué es la Web 2.0?

---

cierto nivel de privacidad a los datos permitiendo ocultar las imágenes a aquellos usuarios que no formen parte de contactos admitidos.

Además, este sitio Web aplica algunos otros patrones: El software no se limita a un solo dispositivo (por ejemplo, se pueden usar teléfonos celulares para subir las fotos) y la compañía tiene control sobre fuentes de datos. Esto último les permite mejorar la aplicación.

Una variación de Flickr es FlickrDown, imágenes desde Flickr por usuario, grupo o etiquetas

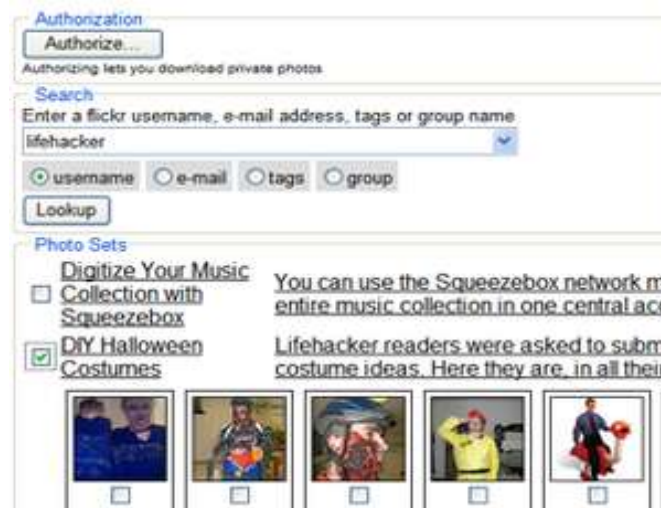


Figura xxx. Tomado de API, Interface de Programación de Aplicaciones. Miércoles, 7 de noviembre de 2007 a las 08.49. Dirección electrónica: <http://www.elwebmaster.com/referencia/api-interface-de-programacion-de-aplicaciones>

FlickrDown es una herramienta muy sencilla para extraer imágenes desde un propio usuario, email, etiqueta o grupo. Esta aplicación es de código abierto y podrás mejorar la versión si crees que podrías agregar algún extra, para seguir la versión Beta.

Otro ejemplo de una aplicación en beta perpetua es Windows Live Mail. Junto con la salida de nuevas versiones de Windows Live Messenger y Windows Live Writer, Microsoft también sacó un Beta de su futuro reemplazo de Outlook Express: Windows Live Mail Beta.

Parece que Microsoft está ahora siguiendo el ejemplo de Google y sacando betas por doquier. Estas versiones betas lanzadas muestran bastante calidad, y realmente están dando a conocer una nueva faceta de Microsoft.

### **MASHUP:**

Un mashup se refiere a la fusión de dos o más conjuntos de contenido de diferentes fuentes, utilizando una aplicación web. Un buen ejemplo de un sitio mashup es el [www.chicagocrime.org](http://www.chicagocrime.org). Este sitio web es el resultado de la mezcla o "mashup" de Google mapas de la delincuencia y la base de datos de Chicago.

Los Mashups también se han aplicado en los bienes raíces, combinando mapas de un determinado sitio web con los datos de otra fuente a fin de que la información acerca de casas en venta en una zona determinada pueda ser producida utilizando un mapa virtual.

Los Mashups de hoy podrían ser clasificados en función de su contenido y fuentes. Los tipos más comunes de mashups son los siguientes:

- **Cartografía mashups** requiere una cartografía que provea una presentación visual de la zona o lugar que está involucrado en el proceso. La fuente para el tipo de datos o información que contendrá el mashup es también necesaria. Estos datos se dibujan en el mapa de una manera gráfica o visual por la aplicación. Ejemplo: [wikimapia.org](http://wikimapia.org)
- **Noticias mashups** se presenta colocando una colección de noticias que un determinado usuario o cliente quiere o prefiere por lo general y luego los presenta en una forma colectiva. Ejemplo: Las noticias se actualizan normalmente a través de las RSS y sistemas de alimentación de datos, por lo que los mashups en esto son más limitados. [digg](http://digg.com) + [del.icio.us](http://del.icio.us) + [slashdot](http://slashdot.org) = [doggdot.us](http://doggdot.us)
- **Video y fotografía mashups** se basan en fotos y videos proveedores de contenido, junto con otra fuente de datos que pueda proporcionar cualquier información que pueda estar relacionada con este tipo de contenidos multimedia. Estos podrían incluir los lugares de las fotos y

los vídeos. Estos lugares o direcciones podrían ser utilizados para el trazado de geografía de referencia visual. Ejemplo: wikimapia + flickr + youtube = jotle.com

- **Compras y búsqueda mashups** trabaja en la idea de comparar los precios de los productos y las especificaciones mediante un método de búsqueda. Los resultados de la búsqueda de diversas fuentes de la web podría ser compilado o puré-up para el navegante de la conveniencia. Ejemplo: shopping + amazon = secretprices.com

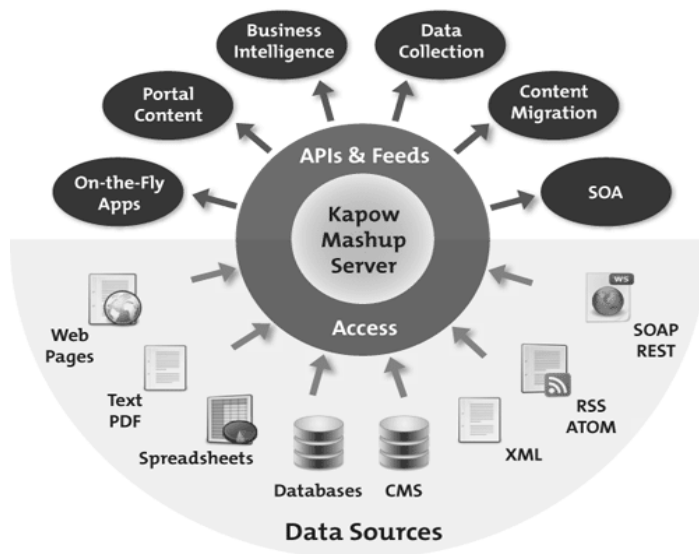
Los Mashups generalmente están en webs de aplicaciones que permitan a las páginas web actualizarse en tiempo real sin necesidad de cargar otra vez la página entera en cada actualización.

Si los tres componentes de un mashup se conectan y colaboran con éxito, este se puede considerar satisfactorio. Los componentes son el navegador del cliente, la página web mashup, y las fuentes de datos en línea o de la API de los proveedores.

Existen varias herramientas para crear mashups, entre esas, se destacan Google Maps y Kapow Mashup Server Product Family.

Las tecnologías Kapow son un mercado líder en software para crear Mashup web 2.0 que te permite conectar, coleccionar y mezclar cualquier cosa en la web. Esa tecnología permite construir y desplegar mashups en una fracción de tiempo y costo que toma el desarrollo de un software tradicional.

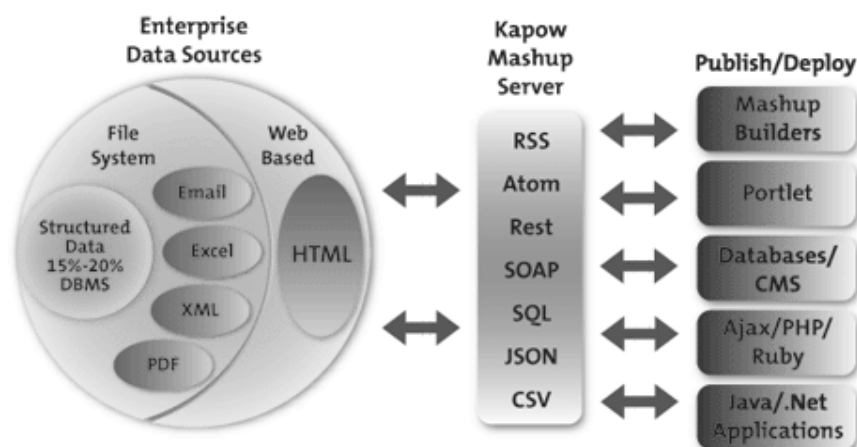
La grafica a continuación ilustra el ecosistema mashup. Hay una gran variedad de fuentes de datos estructuradas y no estructuradas que pueden convertirse fácilmente en alimentadores y servicios mashup. Estos alimentadores y servicios son entonces despachados como una variedad de soluciones de negocios utilizando diferentes estilos mashup.



Tomado de <http://www.kapowtech.com/index.html>

El Kapow Mashup Server incorpora un poderoso lenguaje script visual que soporta múltiples estilos mashup:

- Presentation-centric mashups (aplicaciones compuestas, hojas de cálculo, portales, tablero de instrumentos)
- Data-centric mashups (colección de datos, migración de contenidos, fluido de datos automatizados)



Tomado de <http://www.kapowtech.com/index.html>

## ¿Qué es la Web 2.0?

---

Algunos de los productos de la familia Kapow Mashup Server son:

- Powerful Mashup Engine for Web 2.0
- Creation of Web Services and RSS Mashup Feeds
- Programmatic Access to Any Web Based Data
- Mashup API's for Java, AJAX, and .NET Developer Tools
- Automated Web Data Collection & Content Migration
- Leverages investment in SOA Services & Infrastructure

### **OPML:**

El objetivo de diseño de OPML es tener un formato transparentemente simple, auto documentable, extensible y legible por humanos, que sea capaz de representar una gran variedad de datos fáciles de consular y editar.

El formato OPML es especialmente útil para describir información estructurada como listas de suscripción-RSS, directorios online y varios tipos de información estructurada que le gustaría compartir. Es un formato abierto para que otros outliners y desarrolladores de servicios puedan usarlo para que sea compatible con Radio UserLand, el OPML Editor, o para cualquier otro propósito.

Algunas herramientas Outliners que soportan OPML:

- Hierarchical Notebook (Unix)
- Joe (Java)
- Notetaker (Macintosh)
- OmniOutliner (Macintosh)
- OPML Editor (Windows)
- Radio UserLand (Windows, Macintosh)
- S'More (PocketPC)

- Liferea (Linux): *Linux FEed REAders* (lector de noticias para linux). Es un programa que selecciona noticias de diversas fuentes de internet y las muestra de forma centralizada. Soporta muchos formatos, entre ellos RSS/RDF, CDF, Atom, OCS y OPML. Liferea tiene como objetivo ser rápido, independiente y fácil de usar. Se respalda en las librerías GTK/GNOME pero puede ejecutarse en cualquier escritorio disponible para Linux. Entre las características de Liferea está la de importar y exportar la lista de fuentes con formato OPML.

## Implementación

Para saber cómo se lleva a cabo la implementación OPML, hay que saber que un outline es un árbol, donde cada nodo contiene una serie de atributos tipo cadena. Un documento OPML tiene elementos XML que son cuatro etiquetas, mencionadas en el capítulo 2:

**<opml version="1.0">**

Este es el elemento raíz. Este debe contener el atributo con la versión y una cabecera (*head*) y un cuerpo (*body*).

**<head>**

Contiene metadatos. Puede contener cualquiera de estos elementos opcionales: *title*, *dateCreated*, *dateModified*, *ownerName*, *ownerEmail*, *expansionState*, *vertScrollState*, *windowTop*, *windowLeft*, *windowBottom*, *windowRight*. Cada elemento es un simple elemento de texto. Los contenidos de *dateCreated* y *dateModified* conforman al formato de fecha y hora. *expansionState* contiene una lista de números separados por coma que deberían ser expandidos en despliegue. Los elementos *windowXXX* definen la posición y tamaño de la ventana en despliegue. Un procesador OPML puede ignorar todos los subelementos de *head*. Si el outline es abierto dentro

## ¿Qué es la Web 2.0?

---

de otro outline entonces el procesador debe ignorar todos los subelementos de *head*.

### **<body>**

Compuesto por el contenido del outline. Debe tener uno o más elementos outline.

### **<outline>**

Representa una línea en el outline. Puede contener cualquier número de atributos arbitrarios. Los atributos comunes incluyen *text* y *type*. El elemento outline puede contener cualquier número de sub elementos outline.

El siguiente es un jemplo de un archivo OPML:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<opml version="1.0">
<head>
<title>playlist.xml</title>
<dateCreated>Thu, 27 Jul 2000 03:24:18 GMT</dateCreated>
<dateModified>Fri, 15 Sep 2000 09:01:23 GMT</dateModified>
<ownerName>Dave Winer</ownerName>
<ownerEmail>dave@userland.com</ownerEmail>
<expansionState>1,3,17</expansionState>
<vertScrollState>1</vertScrollState>
<windowTop>164</windowTop>
<windowLeft>50</windowLeft>
<windowBottom>672</windowBottom>
<windowRight>455</windowRight>
```



```

</head>
= <body>
= <outline text="Background">
  <outline text="I've started to note the songs I was listening to as I was writing DaveNet
pieces." />
  </outline>
= <outline text="The Last Napster Sunday?">
  <outline text="Rivers of Babylon (HTC).mp3" type="song" f="Jimmy Cliff - Rivers of
Babylon (HTC).mp3" />
  <outline text="The Tide Is High.mp3" type="song" f="Blondie - The Tide Is High.mp3" />
  <outline text="Up on Cripple Creek.mp3" type="song" f="The Band - Up on Cripple
Creek.mp3" />
  <outline text="Crackerbox Palace.mp3" type="song" f="George Harrison - Crackerbox
Palace.mp3" />
  <outline text="Taxi.Mp3" type="song" f="Harry Chapin - Taxi.Mp3" />
  <outline text="Thick As A Brick.mp3" type="song" f="Jethro Tull-Thick As A Brick.mp3" />
  <outline text="Riding With the King.mp3" type="song" f="B. B. King & Eric Clapton -
Riding With the King - 11 - Hold On Im Coming.mp3" />
  </outline>
= <outline text="The Thrill is Gone?">
  <outline text="Shaft.MP3" type="song" f="Isaac Hayes - Shaft.MP3" />
  </outline>
</body>
</opml>

```

Ejemplo tomado de <http://www.opml.org/spec2>

## Aplicaciones

Una de las aplicaciones que se le está dando al formato OPML es la de compartir los feeds a los que uno está suscrito. Bloglines y otros agregadores

te dan la posibilidad de importar y exportar información sobre feeds. OPML es excelente para mantener una copia de seguridad de tus suscripciones y cambiarte de agregador sin necesidad de tener que suscribirte a cada feed, y para, de un vistazo, comprobar a qué feeds está suscrito alguien.

En opinión de los usuarios de archivos OPML, estas son algunas de las funcionalidades que surgen:

- Un ranking de los **feeds más populares**, lógicamente, aumenta a medida que incrementa el número de usuarios que aportan su archivo OPLM.
- Navegar a través de las **suscripciones de un usuario** (si este comparte sus suscripciones) y comparar que suscripciones tienen en común. De esta manera tienes acceso a las fuentes que una persona utiliza.
- Hay un **agregador Top 100**, suscrito a los feeds más populares.
- Ver qué **personas están suscritas** a determinado feed.
- Ver el **número de feeds al que está suscrita** cada persona, esto se traduce en estadísticas que saca del sistema: Ej. Un 20% de los usuarios están suscritos a más de 180 feeds; un 47% están suscritos a entre 110 y 153 feeds; etc.

### **REST (REpresentational State Transfer):**

Este nuevo estilo arquitectónico que tiene como objetivo diseñar aplicaciones web, indica como debe ser la interacción entre el cliente y el servidor. Permite ofrecer una “interfaz uniforme”, que puedan entender todos los clientes potenciales.

Al diseñar la interfaz de los servicios de la aplicación, el REST se enfoca no en definir un API de métodos y parámetros, sino en especificar recursos de información sobre los que se podrán realizar un conjunto preestablecido de acciones (recuperar, insertar, actualizar y eliminar).

A modo de ejemplo, se supone que se está diseñando un servicio de gestión de usuarios (este ejemplo ha sido tomado de la Wikipedia).

- En el enfoque API o RPC, se definirían las acciones que se pueden realizar. En este caso en concreto se prevé que el protocolo cliente-servidor resultante sería:
  - getUser()
  - addUser()
  - removeUser()
  - updateUser()
  - getLocation()
  - addLocation()
  - removeLocation()
  - updateLocation()
  - listUsers()
  - listLocations()
  - findLocation()
  - findUser()
  
- En el enfoque REST bastaría, como se indicó anteriormente, definir los recursos sobre los que se van a realizar las operaciones. Las operaciones posibles están ya preestablecidas en REST (PUT, GET, POST y DELETE). Debido a que el protocolo HTTP no define operaciones para manejar conjuntos de recursos (operaciones list), se especifican recursos para definir los propios conjuntos.
  - User
  - Localization
  - UserList

## ¿Qué es la Web 2.0?

---

- LocalizationList.

Cada objeto REST se denomina “recurso” y la información contenida en el recurso se denomina “estado”. Este estado o información debe ser codificado para poder enviarse en los mensajes HTTP. Estos estados codificados se denominan “representaciones”. El MIME-TYPE es parte del interfaz debido a que indica el formato en el que se desea la información.

Un aspecto importante de la interacción aplicando REST es que cada mensaje se puede procesar sin requerir información de un mensaje anterior (salvo en el caso del método POST). Esto facilita a posterior el uso de los distintos tipos de caché existentes.

La siguiente tabla muestra las operaciones (interacciones) estándares (coinciden con los verbos del protocolo HTTP):

Operación	Función
<b>GET</b>	Solicitar recurso
<b>POST</b>	Crear recurso nuevo
<b>PUT</b>	Actualizar o modificar recurso
<b>DELETE</b>	Borrar recurso

Tomado de *Representational State Transfer (REST)*. PROYECTO FIN DE CARRERA. Alberto Cubo Velázquez. Dirección electrónica: <http://209.85.207.104/search?q=cache:z9coIRR2SeIJ:trajano.us.es/~antonio/REST.ppt+REST+aplicacion+funcionamiento+web&hl=es&ct=clnk&cd=4&gl=co>

La lógica de negocio inherente a los cambios de estado es transparente para el cliente del servicio REST. Esto se ejecuta como un efecto secundario del cambio de estado del sistema. Por ejemplo: Actualizar un pedido al estado pagado (mediante un PUT) puede implicar interfaz con el subsistema logístico para planificar su envío.

## Objetivos de REST

Los objetivos de REST se listan a continuación:

- Escalabilidad de la interacción con los componentes. La Web ha incrementado tanto la cantidad como la variedad de clientes (estaciones de trabajo, sistemas industriales, dispositivos móviles, etc.), manteniendo un nivel de calidad alto.
- Generalidad de interfaces. Facilidad de acceso de cualquier cliente a cualquier servidor HTTP sin ninguna configuración especial. Esto no es del todo cierto para otras alternativas, como SOAP para los Servicios Web.
- Puesta en funcionamiento independiente. Para que los servidores antiguos sean capaces de entenderse con clientes actuales y viceversa. HTTP permite la extensibilidad mediante el uso de las cabeceras, por medio de las URIs, a través de la habilidad para crear nuevos métodos y tipos de contenido.
- Compatibilidad con componentes intermedios. La compatibilidad con intermediarios (proxy, firewall, Gateway) permite reducir la latencia de interacción, reforzar la seguridad y encapsular otros sistemas.

**REST logra satisfacer estos objetivos aplicando cuatro restricciones:**

- **Identificación de recursos y manipulación de ellos a través de representaciones.** Esto se consigue mediante el uso de URIs, utilizado por el protocolo HTTP. Como se menciona antes, los recursos se convierten a estado y luego se codifican en representaciones para poder ser manipulados.
- **Mensajes autodescriptivos.** REST establece que los mensajes HTTP deben ser tan descriptivos como sea posible. Esto hace posible que los intermediarios interpreten los mensajes y ejecuten servicios en nombre del usuario. Uno de los modos que HTTP logra esto es por medio del uso de varios métodos estándares, muchos encabezamientos y un mecanismo de direccionamiento.
- **Hipermedia** como un mecanismo del estado de la aplicación. El estado actual de una aplicación Web debería ser capturada en uno o más documentos de hipertexto, residiendo tanto en el cliente como en el servidor. El servidor conoce sobre el estado de sus recursos, aunque no intenta seguirle la pista a las sesiones individuales de los clientes. Esta es la misión del navegador, el

sabe como navegar de recurso a recurso, recogiendo información que el necesita o cambiar el estado que él necesita cambiar.

Hoy en día existen millones de aplicaciones Web que implícitamente heredan estas restricciones de HTTP.

### **Beneficios obtenidos al usar REST**

- Mejora el tiempo de respuesta gracias al mecanismo Caché y los mensajes auto descriptivos
- Mejora la seguridad debido a los mensajes auto descriptivos y el uso de los métodos http

### **Implementaciones**

Los siguientes son algunos ejemplos de implementaciones caracterizadas por aplicar REST.

- **eBay**: Desarrolló una API REST en 2004, en esta consultan sus productos a través del método GetSearchResults(). Ejemplo de uso: `http://rest.api.ebay.com/restapi?CallName=GetSearchResults&RequestToken=xyz123&RequestUserId=ebayuser&Query=toy%20boat&Schema=1`
- **RESTLETS**: Su API fue desarrollada en 2006, esta acerca REST a los desarrolladores Java, realizando las mismas funciones de los Servlets pero al estilo REST.

### **SOAP:**

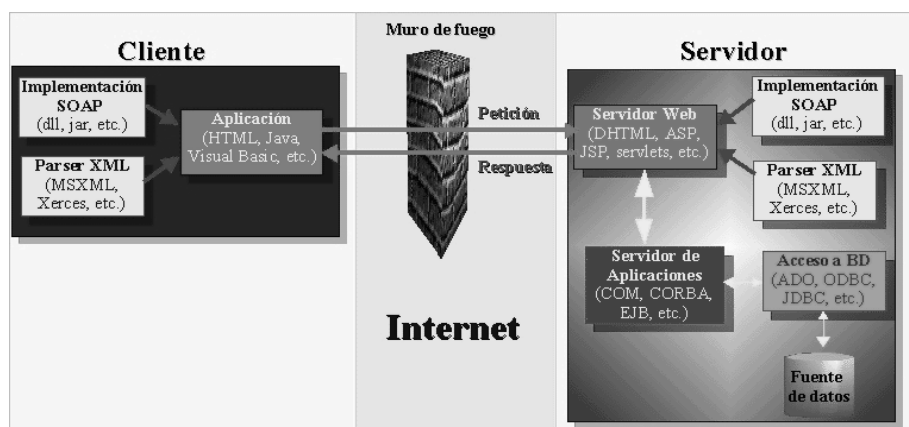
Actualmente las organizaciones llevan a cabo el desarrollo de aplicaciones distribuidas usando tecnologías diversas, como COM (Component Object Model), CORBA (Common Object Request Broker Architecture), EJB (Enterprise Java Beans) y más. Entre las propuestas surge SOAP (Simple Object Access Protocol), que plantea resolver los problemas de falta de interoperabilidad entre las opciones anteriores, tomando como base protocolos

ya establecidos y con gran aceptación en Internet, para para la construcción de servicios Web.

SOAP tiene como objetivo principal ser independiente de cualquier plataforma, lenguaje de desarrollo e implementación (modelo de objetos). Este también pretende establecer un protocolo estándar de invocación de servicios remotos, basado en protocolos estándares de Internet: HTTP (Hiper Text Transport Protocol) y XML (eXtensible Markup Language).

El protocolo de comunicación HTTP es el empleado intrínsecamente para la conexión sobre Internet. Garantiza que cualquier cliente con un navegador estándar pueda conectarse con un servidor remoto. La transmisión de datos se empaqueta (serializa) con XML, que viene a ser mediador en el intercambio de datos, salvando las incompatibilidades entre otros protocolos, tales como el NDR (Network Data Representation) o el CDR (Common Data Representation). Por otra parte, los servidores Web pueden procesar las peticiones de usuario, empleando las tecnologías de servlets, paginas ASP (Active Server Pages) o JSP (Java Server Pages), o un servidor de aplicaciones, solicitando objetos de los tipos CORBA, COM o EJB.

Un esquema del funcionamiento de SOAP se muestra en la figura:



Tomado de **Revista Digital Universitaria**. Artículo *Construcción de Servicios Web con SOAP*. Alejandro Botello Castillo. 31 de marzo del 2002 Vol.3 No.1 Dirección electrónica: <http://www.revista.unam.mx/vol.3/num1/art3/img01.html>

## ¿Qué es la Web 2.0?

---

Como SOAP es independiente del lenguaje de desarrollo las aplicaciones cliente y servidor pueden estar escritas con HTML, DHTML, Java, Visual Basic u otras herramientas y lenguajes disponibles. Lo primordial es tener alguna implementación de SOAP (dependiendo de la herramienta de desarrollo elegida) y enlazar sus librerías con la aplicación.

La especificación SOAP establece que los mensajes están compuestos de tres secciones: la envoltura (Envelope), el encabezado (Header) y el cuerpo (Body). Mientras que el encabezado es opcional (pueden no tener, tener uno o tener varios), la envoltura y el cuerpo son obligatorios y únicos. La sección Envelope contiene a las otras dos secciones como elementos. A continuación se muestra un ejemplo donde se analiza el contenido de un mensaje SOAP (se numeran las líneas como referencia para la explicación), tomado de la Revista Digital Universitaria. Artículo *Construcción de Servicios Web con SOAP*. Alejandro Botello Castillo. 31 de marzo del 2002 Vol.3 No.1 Dirección electrónica: <http://www.revista.unam.mx/vol.3/num1/art3/>:

```
1 POST /soap/StockQuote HTTP/1.1
2 Host: www.stockquoteserver.com
3 Content-Type: text/xml; charset="utf-8"
4 Content-Length: nnnn
5 SOAPAction: ""
6
ENV="http://schemas.xmlsoap.org/soap/envelope/">
SOAP-ENV:mustUnderstand="1">
11 5
12</ t:Transaction>
13 </ SOAP-ENV:Header>
14 <SOAP-ENV:Body>
15 <m:GetLastTradePrice xmlns:m="http://www.stockquoteserver.com/soap">
16 <symbol>DEF</symbol>
```



17 </m:GetLastTradePrice>

18 </SOAP-ENV:Body>

19 </SOAP-ENV:Envelope>

En el encabezado (líneas 1 a 5) se debe especificar el tipo de mensaje en Content-Type como text/xml, para que sea correctamente interpretado, así como el conjunto de caracteres a usar (utf-8). Como una extensión SOAP al encabezado se incluye SOAPAction, que define un URI desde donde se localizará el servicio, que en este caso es una cadena vacía, ya que se ha establecido como www.stockquoteserver.com/soap/StockQuote (líneas 1 y 2). Es importante remarcar que el encabezado y el cuerpo del mensaje están delimitados por dos retornos de carro/nueva línea (línea 6 en blanco).

El contenido del mensaje se describe a partir de la línea 7. Con xmlns:identificador="URI" se define un espacio de nombre (namespace), que es un identificador de ámbito para evitar ambigüedades. Se observa que el encabezado contiene un elemento Transaction, con atributo mustUndestand="1" (tiene que ser interpretado) y con un valor de 5, mientras que el cuerpo contiene un elemento GetLastTradePrice (el nombre del servicio) con el elemento symbol (el parámetro del servicio) y el valor DEF (el valor a buscar).

La respuesta obtenida será algo similar a lo siguiente:

1 HTTP/1.1 200 OK

2 Content-Type: text/xml; charset="utf-8"

3 Content-Length: nnnn

4

5 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

6<SOAP-ENV:Header>

7 <t:Transaction xmlns:t="http://www.stockquoteserver.com/soap" mustUnderstand="1">

8 5

</t:Transaction>

## ¿Qué es la Web 2.0?

---

```
9 </SOAP-ENV:Header>
10 <SOAP-ENV:Body>
11 <m:GetLastTradePriceResponse
xmlns:m="http://www.stockquoteserver.com/soap">
12 <Price>34.5
13 </m:GetLastTradePriceResponse>
14 </SOAP-ENV:Body>
15 </SOAP-ENV:Envelope>
```

El cuerpo del mensaje tiene un similar esquema, sólo diferenciado en que el cuerpo contiene al elemento *GetLastTradePriceResponse* (que es el nombre del servicio, añadiendo el sufijo *Response*), que a su vez contiene al elemento *Price* (el parámetro de retorno) con un valor de 34.5, el cual es el resultado del servicio.

### Aplicaciones

SOAP no se centra en un software de desarrollo específico, los distintos fabricantes ofrecen sus paquetes que incluyen asistentes, código de ejemplo y manuales para los desarrolladores. Entre las alternativas están la de Microsoft (usando su tecnología de componentes basados en el modelo COM), IBM (que permite usar componentes COM, EJB y clases de Java sobre servlets), DevelopMentor (usando clases de Java o scripts de Perl) y otras empleando el lenguaje C++.

Un ejemplo de una aplicación de este tipo es la herramienta gratuita SOAP ToolKit de <http://msdn.microsoft.com/library/default.asp?url=/downloads/list/websrv.asp> Microsoft que contiene un asistente de generación de definiciones, documentación y ejemplos denominado Web Service References Tool que permite realizar búsquedas de descripción, descubrimiento e integración (UDDI, Universal Description Discovery and Integration) de aquellos servicios registrados o abrir directamente su archivo de descripción. (WSDL, Web Services Description Language).

Ejemplos:

- **Amazon.com:** Permite acceder directamente a sus servicios, manejando directamente el “carrito de compra”. Amazon.com fue

pionera en el uso de REST en 2002, poseyendo una base de datos con todos los productos que vende. En ese año sus productos se accedían como recursos, no como métodos de búsqueda. Pero cuando comenzó a realizar servicios más sofisticados debió migrar a SOAP.

- **Cowbell:** Es un elegante organizador de música basado en álbumes. Soporta la mayoría de formatos de audio incluidos ogg/FLAC, mp3 y MusePack. Cowbell está fuertemente integrado con Amazon, usando el servicio gratuito SOAP. Cowbell en combinación con inteligentes algoritmos “adivina” la información de cada canción de los títulos de las canciones. También guarda las imágenes de las cubiertas en el mismo directorio donde están los álbumes.
- **Google:** Puedes actualizar tus páginas con su información, e invocar la búsqueda directamente usando una API.

### **RUBY ON RAILS (ROR):**

Ruby on Rails es un framework Model View Controller(MVC) de reciente creación, especialmente apropiado para el desarrollo web recurriendo a metodologías ágiles. Está perfeccionado para permitir el desarrollo veloz de aplicaciones web 2.0 manteniendo al mismo tiempo un alto nivel de calidad.

#### **Arquitectura: Patrón MVC**

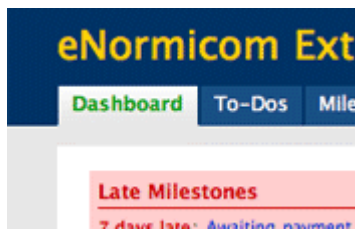
ROR facilita el diseño y desarrollo de aplicaciones web al dividir todos los componentes de la aplicación en 3 capas (Modelo, Vista y Controlador).

- Modelo es todo acceso a base de datos, y las funciones que aplican la “lógica de negocio”.
- La vista es la que expone la información al usuario final: HTML, XML.
- El controlador se encarga de unir la vista con el modelo, contiene toda la lógica de programación. Almacena las funciones que toman los valores de un formulario, delega consultas de base de datos al modelo y produce valores que solicitarán a la vista adecuada.

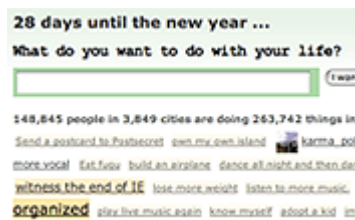
## ¿Qué es la Web 2.0?

---

En conclusión con ROR se centra en lo que verdaderamente importa: la funcionalidad de las aplicaciones. Estas llegan a ser más complejas y de funcionamiento más “suave” con muchísimo menos esfuerzo. Algunos ejemplos de grandes aplicaciones hechas con RoR son: Basecamp, La Coctelera, Odeo, 43things, Shopify, Fluxiom, Typo, Bigcartel, WriteBoard, SoapBX, Campfire, Backpack, entre otras.



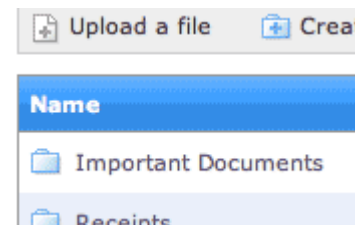
Basecamp: Project management.  
The original Rails app by 37signals.



43things: Achieve your goals in life.



Shopify: E-commerce made easy.

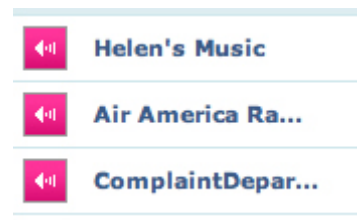


Strongspace: Secure file hosting.

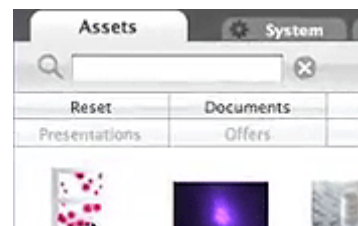
Tomado de <http://www.rubyonrails.org/>



Campfire: Group chat for biz.  
Pushing the limits of Ajax in Rails.



ODEO: Record and share audio.



Fluxiom: Digital asset management.



Typo: Put your weblog on Rails.

### **2.7.3 APLICACION DE LOS ESTÁNDARES**

#### **XML 1.0:**

XML es un lenguaje compuesto por marcas (tags) que ofrece un formato para la descripción de datos estructurados. Estos tags son definidos por el usuario y facilitan declaraciones de contenido más precisas y resultados de búsquedas más significativos en varias plataformas. También, XML habilitará una nueva generación de aplicaciones para ver y manipular datos basados en la Web. XML es un estándar de la industria y es independiente de cualquier empresa privada, esto permite intercambiar datos entre aplicaciones y plataformas. Lo cual ha resultado en una nueva generación de aplicaciones de comercio electrónico en la Web, con formas nuevas de acceso a las bases de datos existentes y de entregar datos a clientes de Web. Las aplicaciones se pueden generar más ágilmente, su mantenimiento es más sencillo y pueden ofrecer fácilmente varias vistas de los datos estructurados.

XML, que proporciona un estándar de datos que puede codificar el contenido, la semántica y los esquemas de una gran variedad de casos, sirve para marcar lo siguiente:

- Un documento normal.
- Un registro estructurado, como un registro de citas o un pedido de compra.
- Un objeto con datos y métodos, como el formulario permanente de un objeto Java o de un control ActiveX.
- Un registro de datos, como el conjunto de resultados de una consulta.
- Metacontenido sobre un sitio Web, como el formato de definición de canal (CDF).
- Representaciones gráficas, como la interfaz de usuario de una aplicación.
- Entidades y tipos de esquema estándar.

## ¿Qué es la Web 2.0?

---

- Todos los vínculos entre datos y personas que hay en la Web.

Cuando los datos llegan al escritorio del cliente, se pueden manipular, editar y presentar en varias vistas, sin tener que regresar al servidor.

XML es un lenguaje muy similar a HTML pero su función primordial es describir datos y no mostrarlos. HTML y XML se complementarán el uno al otro, anudándose ambas gramáticas (XHTML).

Algunos de los objetivos planteados por el Grupo de Trabajo XML y el W3C son:

- XML debe ser directamente útil sobre Internet.
- XML debe tolerar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- El número de características opcionales en XML debe ser absolutamente mínimo, idealmente cero.
- Los documentos XML deben ser claros y legibles por los usuarios de este lenguaje.
- El diseño de XML debe ser formal, conciso y preparado ágilmente.
- Los documentos XML deben ser fácilmente creables.
- La brevedad en las marcas XML es de mínima importancia.

### **Estructura de XML.**

Físicamente XML está compuesto por unidades llamadas entidades. Una entidad puede hacer referencia a otra, produciendo que esta se incluya en el documento. Cada documento comienza con una entidad documento, también llamada raíz. Lógicamente, el documento está compuesto de declaraciones, elementos, comentarios, referencias a caracteres e instrucciones de procesamiento, todos los cuales están etiquetados por una marca explícita. Las estructuras lógica y física deben ajustarse de manera adecuada:

Los documentos XML se dividen en dos grupos, documentos bien formados y documentos válidos.

Un documento XML se dice que está bien formado si, considerándolo como conjunto, ajusta con las especificaciones XML de producción, lo que implica:

- Contiene uno o más elementos.
- Hay exactamente un elemento, llamado raíz o elemento documento, de forma que ninguna parte del mismo aparece en el contenido de ningún otro elemento. Los elementos delimitados por una etiqueta de inicio y otra de final, se encuentran anidados correctamente.
- Cumple todas las restricciones que proporciona su especificación a través del DTD. Si no se utiliza DTD, el documento se obliga a comenzar con una Declaración de Documento único (SDD, Standalone Document Declaration) que indique:

`<?XML version="1.0" standalone="yes"?>`

- Cada una de sus partes procesadas está bien formada.
- Todas las etiquetas deben estar balanceadas: esto es, todos los elementos que contengan datos de tipo carácter deben tener etiquetas de principio y fin (no está permitida la omisión excepto para los elementos vacíos.);
- Todos los valores de los atributos deben ir entrecomillados (el carácter comilla simple [el apóstrofe] puede utilizarse si el valor contiene caracteres comillas dobles, y viceversa): si necesitas ambos, utiliza `&apos;` y `&quot;`;
- Cualquier elemento VACÍO (p.e. aquellos que no tienen etiqueta final como `<IMG>`, `<HR>`, y `<BR>` y otros de HTML) deben terminar con `'/>'` o debes hacerlos no VACÍOS añadiéndoles una etiqueta de fin; Ejemplo: `<BR>` se convertirá en `<BR/>` o en `<BR></BR>`.

- No debe haber etiquetas aisladas (< ó &) en el texto (p.e. debe darse como &lt; y &amp;), y la secuencia ]]> debe darse como ]]&gt; si no ocurre esto como final de una sección marcada como CDATA;
- Los elementos deben anidar dentro de sí sus propiedades (no se deben sobreponer etiquetas, como en el resto de SGML);
- Los Archivos bien-formados sin-DTD pueden utilizar atributos en sus elementos, pero éstos deben ser todos del tipo CDATA, por defecto.

Dado que XML está diseñado para ser un subconjunto de SGML, cualquier documento XML válido debe ser también un documento SGML válido.

### **Aplicaciones de XML**

Algunas de las aplicaciones de XML son:

- Ofrecer mecanismos más versátiles de mostrar datos. Uno de estos es DOM (Document Object Model), una API soportada por todos los procesadores de XML y HTML. El objetivo de esta es que se pueda representar (a través de javascripts o JavaApplets) documentos XML en los navegadores Web, ya que XML proporciona una sintaxis y una semántica.
- Buscadores inteligentes. Se consigue a través de información etiquetada por su significado de forma precisa en documentos XML, así puede ser localizada de forma mucho más clara que en documentos HTML. Con DTDs estandarizados para distintas aplicaciones (librerías, tiendas de deporte, catálogos de componentes,...) se podría programar buscadores Web que recuperasen información sobre un producto de cualquier website en el mundo.
- Intercambio de información entre sistemas heterogéneos. La base es la misma que para los buscadores inteligentes. Debido a que el DTD proporciona un formato estándar para representar la información de un tema específico, puede emplearse para facilitar el intercambio de información entre distintas fuentes (actualmente existen ya dos DTD



estandarizados uno para fabricantes de chips y otro para industrias químicas, llamado CML).

### **Ventajas de XML**

- Los autores y proveedores pueden diseñar sus propios tipos de documentos usando XML, en vez de limitarse a HTML.
- XML puede dar más y mejores facilidades para la representación en los visualizadores.
- La información será más accesible y reutilizable, porque la flexibilidad de las etiquetas de XML pueden manipularse sin tener que adaptarse a reglas determinadas por un fabricante, como es el caso de HTML.

### **XML es la base de RDF y de la Web Semántica**

El Resource Description Framework (RDF) de la W3C es un formato de texto XML que soporta aplicaciones de descripción de recursos y metadatos, tales como listas de temas musicales, colecciones de fotos, y bibliografías. Por ejemplo, RDF podría permitirle identificar las personas en un álbum de fotos Web usando información de una lista de contactos personales; entonces, su cliente de correo podría enviar automáticamente a esas personas un mensaje diciendo que sus fotos están en la Web. Así como el ser humano se pone de acuerdo en el significado de las palabras, las computadoras deben hacer lo mismo con los significados de los términos para comunicarse efectivamente. Cuando se hace esto en una cierta área (compras o manufactura, por ejemplo) se llama ontología, y es una parte necesaria de la Web Semántica. RDF, las ontologías, y la representación del significado de modo que las computadoras ayuden a las personas a realizar el trabajo, son tópicos de la Actividad de la Web Semántica.

### **DOM:**

Haciendo uso del estándar DOM, se logra el acceso a documentos en un formato de árbol que es por otra parte el tipo de estructura que más procesadores XML y XSL admiten. Cualquier implementación requiere que el

contenido completo del documento sea recibido y almacenado en memoria. DOM se usa sobre todo en aplicaciones que requieran acceder a elementos del documento para ser manipulados de manera repetitiva e impredecible. Si la aplicación involucra una lectura o escritura secuencial o selectiva, el uso de DOM representa una considerable ventaja.

XML DOM es una API que se encarga del acceso y la manipulación a documentos XML. Proporciona unos elementos de programación relativamente simples de utilizar y que permiten hacer todo lo que se quiera con los documentos XML... todo ello a través de numerosos interfaces (con sus métodos y atributos perfectamente documentados), aunque los verdaderamente útiles son sólo unos pocos. La navegación a través de los elementos del XML se hace a través de su árbol DOM que se carga en memoria, en consecuencia hay que tener presente que utilizar XML DOM implica consumos de recursos nada despreciables

Los APIs SAX y DOM están determinados por el grupo XML-DEV y por el W3C, respectivamente. Las librerías que definen dichos APIs están incluidas en el fichero parser.jar, que también contiene la implementación de referencia de Sun, Project X.

El "API Simple" para XML (SAX) es el manejador de eventos, mecanismo de acceso serie que hace el proceso elemento-por-elemento. El API para este nivel lee y escribe XML al repositorio de datos o la Web. Para aplicaciones del lado servidor y de alto rendimiento, se quiere entender completamente este nivel. Pero para muchas aplicaciones, un mínimo entendimiento es suficiente.

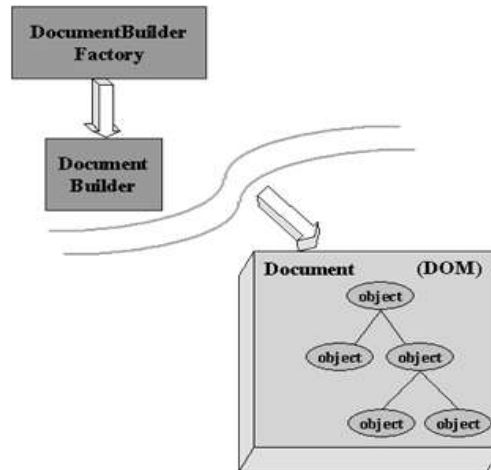
El API DOM es habitualmente fácil de usar. Proporciona una relativamente familiar estructura de árbol de objetos. Se puede usar este API para manipular el árbol de objetos que encapsula la aplicación. El API DOM es ideal para aplicaciones interactivas porque el modelo de objeto completo se presenta en memoria, donde puede ser accedido y manipulado por el usuario.

Por otro lado, construir el DOM requiere la lectura de toda la estructura XML y mantener el árbol de objetos en memoria, lo que significa un mayor consumo

de CPU y de memoria. Por esta razón, el API SAX tiende a ser el preferido en aplicaciones del lado del servidor y filtros de datos que no requieren representación de datos en memoria.

- **Los APIs "Document Object Model" (DOM)**

El siguiente diagrama muestra los APIs JAXP en acción.



**Figura 7: Apis DOM**

Se usa la clase `javax.xml.parsers.DocumentBuilderFactory` para obtener un ejemplar **DocumentBuilder** (superior izquierda), y se usa esto para producir un Document (un DOM) que cumple la especificación DOM (inferior derecha). De hecho, el constructor que se obtiene está determinado por la propiedad del sistema, `javax.xml.parsers.DocumentBuilderFactory`, que selecciona la implementación de la factoría que se usa para producir el constructor. (El valor por defecto de la plataforma puede sobrescribirse desde la línea de comandos).

Se puede usar el método `newDocument()` para crear un Documento vacío que implemente el interface `org.w3c.dom.Document`. De forma alternativa, se puede usar uno de los métodos analizadores del constructor para crear un Documento desde datos XML existentes. El resultado es un árbol DOM como el que se ve en la esquina inferior derecha del diagrama.

- **Paquetes**

## ¿Qué es la Web 2.0?

---

La implementación de DOM está definida en los siguientes paquetes.

Paquete	Descripción
<b>org.w3c.dom</b>	Define los interfaces de programación DOM para documentos XML (y, opcionalmente HTML), según lo especificado por W3C.
<b>javax.xml.parsers</b>	Define la clase <b>DocumentBuilderFactory</b> y la clase <b>DocumentBuilder</b> , que devuelve un objeto que implementa el interface Document W3C. La factoría que se usa para crear el constructor está determinada por la propiedad del sistema javax.xml.parsers, que puede ser seleccionada desde la línea de comandos o sobrescrita cuando se llama al método <b>newInstance</b> . Este paquete también define la clase <b>ParserConfigurationException</b> para reportar errores.
<b>com.sun.xml.tree</b>	La implementación Java XML de Sun de las librerías DOM, incluyendo las clases <b>XmlDocument</b> , <b>XmlDocumentBuilder</b> , y <b>TreeWalker</b> .

### ECMASCRIPT 262:

ECMAScript especifica las normas para la implementación de un lenguaje script, es decir, define todas las propiedades, métodos y objetos.

ECMAScript describe:

- Sintaxis
- Tipos
- Sentencias
- Palabras claves
- Palabras reservadas
- Operadores
- Objetos

## **Aplicaciones que contienen ECMAScript y lenguajes en los que están escritos**

- Navegadores basados en el motor de navegación Gecko (Firefox, Mozilla, Netscape, Camino, Flock) (Javascript)
- Internet Explorer (JScript)
- Opera (Javascript y JScript)
- Konqueror (Javascript)
- Microsoft .NET Framework (JScript .NET)
- Adobe Flash y Adobe Flex (ActionScript)
- Adobe Acrobat (JavaScript)

Cada implementación de ECMAScript contiene sus propias librerías. A pesar de estas implementaciones que no necesariamente funcionan en cualquier ambiente que interprete ECMAScript, se puede crear cierta compatibilidad.

## **JavaScript y ECMAScript**

JavaScript es un lenguaje tipo script compacto, orientado a objetos y guiado por eventos, creado para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Comenzando a marcar la etapa de la Web 2.0.

Los programas JavaScript van embebidos en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos, etc.

## **Versiones**

El navegador es el que interpreta los programas JavaScript, lo que quiere decir que si alguno no soporta JavaScript, no se podrá ejecutar las funciones que se programen.

Desde luego, Netscape y Explorer (los más usados) lo soportan, el primero desde la versión 2 y el segundo desde la versión 3.

Navegador	Version de JavaScript	Soporte ECMA
Nestcape 2	Javascript 1.0	--
Internet Explorer 2	No soporta JavaScript	--
Nestcape 3	Javascript 1.1	--
Internet Explorer 3	Javascript 1.0	--
Nestcape 4	Javascript 1.2 - 1.3 incompleta	ECMA-262-compliant hasta la version 4.5
Internet Explorer 4	Javascript 1.2	ECMA-262-compliant
Nestcape 6	ECMA compliant Javascript 1.4	Full ECMAScript-262
Internet Explorer 5	ECMA compliant Javascript 1.3	Full ECMAScript-262

Tomado de [http://developer.mozilla.org/es/docs/Gu%C3%ADa\\_JavaScript\\_1.5:Concepto\\_de\\_JavaScript](http://developer.mozilla.org/es/docs/Gu%C3%ADa_JavaScript_1.5:Concepto_de_JavaScript)

Las diferentes versiones de JavaScript están ahora integradas en el estándar ECMAScript-262. A continuación se describirá JavaScript según este estándar.

### Como y donde incluir Java Script

Existen distintos modos de incluir lenguaje JavaScript en una página.

La forma más frecuente de hacerlo es utilizando la directiva <script> en un documento HTML (se pueden incluir tantas directivas <script> como se quiera en un documento). El formato es el siguiente:

```
<script language="Javascript 1.3">
```

El atributo lenguaje hace referencia a la versión de JavaScript que se va a utilizar en dicho script. Otro atributo de la directiva script es src, que puede usarse para incluir un archivo externo que contiene JavaScript y que quiere incluirse en el código HTML.

```
<script language="JavaScript" src ="archivo.js"> </script>
```

El archivo externo simplemente es un archivo del texto que contiene código JavaScript, y cuyo nombre acaba con la extensión js.

Puede incluirse también código JavaScript como respuesta a algún evento:

```
<input type="submit" onclick="alert('Acabas de hacer click');return false;" value="Click">
```

**Nota:** Los scripts pueden incluirse como comentarios para asegurar que su código no es "visto" por navegadores viejos que no reconocen JavaScript y así evitar errores.

```
<script ....>
```

```
<!--
```

```
Código JavaScript
```

```
//-->
```

```
</script>
```

### Gramática de JavaScript

Si conoce algún otro lenguaje de programación ya estará familiarizado con el uso de variables, operadores, declaraciones, etc. Lo que se resume en la tabla siguiente son los elementos principales de la gramática de JavaScript.

Es importante resaltar que JavaScript es sensible a mayúsculas y minúsculas, todos los elementos de JavaScript deben referenciarse cómo se definieron, no es lo mismo "Salto" que "salto".

<b>Variables</b>	Etiquetas que se refieren a un valor cambiante.
<b>Operadores</b>	Pueden usarse para calcular o comparar valores. Ejemplo: pueden sumarse dos valores, pueden compararse dos valores.
<b>Expresiones</b>	Cualquier combinación de variables, operadores, y declaraciones que evalúan a algún resultado. Ejemplo: intTotal=100; intTotal > 100

<b>Sentencias</b>	Una sentencia puede incluir cualquier elemento de la gramática de JavaScript. Las sentencias de JavaScript pueden tomar la forma de condicional, bucle, o manipulaciones del objeto. La forma correcta para separarlas es por punto y coma, esto sólo es obligatorio si las declaraciones múltiples residen en la misma línea. Aunque es recomendable que se acostumbre a terminar cada instrucción con un punto y coma, se ahorrará problemas.
<b>Objetos</b>	Estructura "contenedora" de valores, procedimientos y funciones, cada valor refleja una propiedad individual de ese objeto.
<b>Funciones y Métodos</b>	Una función de JavaScript es bastante similar a un "procedimiento" o "subprograma" en otro lenguaje de programación. Una función es un conjunto que realizan alguna acción. Puede aceptar los valores entrantes (los parámetros), y puede devolver un valor saliente. Un método simplemente es una función contenida en un objeto.

Tomado de [http://developer.mozilla.org/es/docs/Gu%C3%ADa\\_JavaScript\\_1.5:Concepto\\_de\\_JavaScript](http://developer.mozilla.org/es/docs/Gu%C3%ADa_JavaScript_1.5:Concepto_de_JavaScript)

### **Variables**

Las variables guardan y recuperan datos. Estas pueden referirse a un valor que cambia o se cambia. Las variables son referenciadas por su nombre, y este debe ser conforme a ciertas reglas (debe empezar con una letra o ("\_")); los caracteres siguientes pueden ser números (0-9), letras mayúsculas o letras minúsculas).

A continuación se analizan algunos ejemplos tomados de [http://developer.mozilla.org/es/docs/Gu%C3%ADa\\_JavaScript\\_1.5:Concepto\\_de\\_JavaScript](http://developer.mozilla.org/es/docs/Gu%C3%ADa_JavaScript_1.5:Concepto_de_JavaScript):

Ejemplos de definiciones **erróneas** :



```
var Mi Variable, 123Probando, $Variable, for, while;
```

Ejemplos de definiciones **correctas**:

```
var _Una_Variable, P123robando, _123, mi_carrooo;
```

Por supuesto, se puede inicializar una variable al declararla:

```
var Una_Variable="Esta Cadenita de texto";
```

Las variables en JavaScript pueden ser de alcance global o local. Una variable global es accesible desde cualquier lugar del programa, en este caso, desde cualquier <script> de la página, mientras que una variable local sólo lo es desde la función o método en la que fue declarada.

Normalmente, usted establece una nueva variable global asignándole simplemente un valor:

```
globalVariable=5;
```

Sin embargo, si usted está codificando dentro de una función y usted quiere implantar una variable local que sólo tenga alcance dentro de esa función, debe declarar la nueva variable haciendo uso de var:

```
function newFunction()  
{  
var localVariable=1;  
globalVariable=0;  
...  
}
```

### **Tipos de datos**

Javascript reconoce seis tipos de valores diferentes: numéricos, lógicos, objetos, cadenas, nulos e indefinidos.

JavaScript tiene la singularidad de que una variable puede cambiar de tipo durante su vida, por ejemplo uno puede declarar una variable que ahora sea un entero y más adelante una cadena.

```
MiVariable=4;
```

## ¿Qué es la Web 2.0?

---

Y después:

```
MiVariable="Una_Cadena";
```

En Javascript no es necesario declarar las variables especificando el tipo de dato que contendrán, será el propio interprete el que le asignará el tipo apropiado. (la idea crear programas pequeños y lograr que el programador realice los scripts de la manera más rápida posible).

Tipos de Datos:

<b>Números</b>	Enteros o coma flotante.
<b>Boleanos</b>	True o False.
<b>Cadenas</b>	Los tipos de datos cadena deben ir delimitados por comillas simples o dobles.
<b>Objetos</b>	Obj = new Object();
<b>Nulos</b>	Null
<b>Indefinidos</b>	Un valor indefinido es el que corresponde a una variable que ha sido creada pero no le ha sido asignado un valor.

Tomado de [http://developer.mozilla.org/es/docs/Gu%C3%ADa\\_JavaScript\\_1.5:Concepto\\_de\\_JavaScript](http://developer.mozilla.org/es/docs/Gu%C3%ADa_JavaScript_1.5:Concepto_de_JavaScript)

### **MATHML 2.0:**

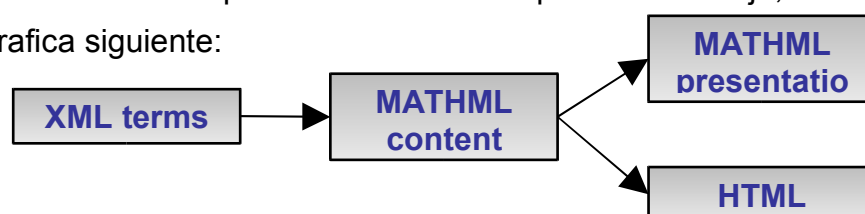
El World Wide Web Consortium (W3C) recomienda la especificación de MathML (Mathematical Markup Language) como aplicación XML (eXtensive Markup Language) apta para codificar signos matemáticos, símbolos científicos y contenidos en Internet.

MathML es un lenguaje de bajo nivel que tiene en cuenta la comunicación máquina a máquina de datos estructurados como información de bases de datos. El W3C ha tratado de definir un lenguaje XML dócil que duplique el

contenido y la presentación de expresiones matemáticas, algo que hasta el momento ha estado demorado en Internet.

El lenguaje MathML hace uso de dos series de códigos progresivos: mientras la primera presenta los signos matemáticos en series crecientes, la segunda transmite el significado semántico de las expresiones matemáticas, lo que posibilita la codificación de símbolos y signos tanto matemáticos como científicos.

Según la teoría de la prueba, las derivaciones formales son solo otra categoría de expresiones matemáticas. Esta observación simple abre posibilidad para explotar MathML no solamente para las declaraciones lógicas o expresiones matemáticas tradicionales sino también para la más ambiciosa meta de pruebas de codificación y entonces, por las extensiones convenientes, expresa documentos matemáticos. Por otro lado, entrando en ediciones básicas, hay que tener mucho cuidado de respetar el ambiente multilingüe de las diferentes posibilidades de formalización matemática: cada sistema básico requiere su propio dialecto lógico con un margen de beneficio específico. Al respecto de esto, XML proporciona a una flexibilidad derecha. Así, MathML, introduce los detalles, su margen de beneficio contenido se puede utilizar para darle interés al lenguaje intermedio entre la codificación formal baja (expresada en un cierto dialecto específico de XML) y el lenguaje del formato de la ficha. En este nivel intermedio que se podría conseguir ya librados de muchas molestias de detalles sintácticos que afligen típicamente a las matemáticas formales, transformando en frases entendibles para los seres humanos. En cualquier caso, el contenido formal sería preservado por medio apuntadores de respaldo a un nivel de especificación bajo, como se aprecia en la grafica siguiente:



**Figura 8:** Transformaciones en ficheros de XML: Las flechas posteriores representan las conexiones del contenido y de los ficheros de la presentación

## ¿Qué es la Web 2.0?

---

a los ficheros correspondientes de XML. Tomado de [http://mx.geocities.com/c\\_sinuhe/mathml.html](http://mx.geocities.com/c_sinuhe/mathml.html)

La aplicación de MathML como formato de archivo y de la presentación para los depósitos grandes del conocimiento matemático estructurado abre nuevas y emocionantes posibilidades y agranda su campo de la aplicación. De hecho, los sistemas lógicos existentes no son convenientes para la creación de depósitos grandes del conocimiento matemático estructurado accesible vía el Web. De hecho, las bibliotecas en armazones lógicos se salvan generalmente en representaciones sea eso claramente insatisfactorio puesto que se orientan también a las aplicaciones específicas. La información no está directamente disponible si no por medio de las funciones ofrecidas por el sistema mismo, y por otra parte, la misma información falta una forma convincente de presentación. En el contrario, XML es una aplicación natural independiente de formato de la especificación para las bibliotecas formales, proporcionar una tecnología central para salvar, extraer, y procesar documentos matemáticos y abarcar los mecanismos Web-Web-publicación sofisticados (hojas del estilo). De la misma manera, MathML se puede explotar naturalmente no solamente como (intermedio) formato de archivo interesante sino también como un lenguaje privilegiado de formato de fichas, rendimiento de un puente natural entre las herramientas para la mecanización de razonamiento formal y el Web.

Incluso si es contenido el margen de beneficio se restringe a la codificación de un detalle conjunto de fórmulas (los utilizaron hasta los primeros dos años de la universidad dentro de los Estados Unidos), se ha concebido como extensible y así, potencialmente, lenguaje infinito.

El pasar a través de una representación semiformal intermedia (por ejemplo contenido de MathML) mejora la modularidad de la configuración total: muchos dialectos lógicos específicos se pueden asociar en el mismo intermedio lenguaje (o en extensiones convenientes de él), y semejantemente la misma representación intermedia se puede transformar en diferentes formatos de la presentación.

Por el hecho de ser un estándar, MathML puede ser explotado a las expresiones de corte y de la goma a partir de una aplicación a otra.

El contenido de MathML captura bien la "semántica informal" (o, mejor, el uso *semiformal*) de bien conocidos operadores, por ejemplo, por ejemplo, igualdad. En el nivel foundational, hay muchos de diversas igualdades (intensional, extensional, Igualdad higher-order de Leibniz, igualdades tipo-específicas, etc.), pero hay poca punta para preservar la distinción en el intermedio nivel si no como puntero a la definición específica.

Cuando se decida usar XML como el idioma descriptivo de datos para representación de términos matemáticos, transformaciones de XSL (hojas del estilo) se deberá proporcionar a un mecanismo estándar del convertidor o generar automáticamente representaciones asociadas de MathML de un documento matemático de XML, para asociar la notación al contenido formal. Existen dos fases de la aplicación de las hojas del estilo: el primer genera el MathML representación contenta del XML uno; el segundo genera de esta representación intermedia el MathML margen de beneficio de la presentación (o un diverso margen de beneficio de la blanco).

La transformación del formato formal bajo de XML al contenido de MathML requiere típicamente la extensión del último lenguaje a cubrir, por medio de conveniente `csymbols`, los constructores primitivos del específico dialecto lógico, en el cual los buenos sistemas foundational son pocos. La potencia del contenido de MathML comienza a ser explotada completamente en la codificación de las entidades matemáticas definidas en el sistema lógico, y especialmente cuando pertenecen al conjunto básico de elementos contenidos de MathML. En este caso, el identificador bajo puede ser asociado directamente al operador correspondiente, automáticamente recuperación su presentación prevista. Un puntero a la definición formal, baja del operador puede ser guardado en `definitionURL` atributo.

En el nivel de la presentación, hay que extender las hojas (ya disponibles) del estilo de MathML para cubrir el nuevo `csymbol` que se introdujo en el nivel de

contexto. Una vez más se guardan punteros a la información formal baja asociada en `xlink:href` atributo, usando la maquinaria de XLink

Al hacer uso de MathML, hay que tener presente que este estándar requiere hacer una unión de dos objetivos complementarios del margen de beneficio: a disposición de la presentación o de paginación y significado matemático. La necesidad algo más allá de una disposición de paginación pura puede ser considerada ya al intentar el proceso para entender algo tan simple como  $a ( b + c )$ . sin la información adicional no se puede decidir si esto significa que la función  $a$  está aplicada a la suma *de*  $b$  y *de*  $c$  , o si es el producto a el cual se amplía  $ab + ac$  . Mientras que dentro de un ambiente controlado la variación heurística o convencional se puede aplicar, clarificando esto la interpretación es exacto el papel del margen de beneficio contenido-base. De un punto de vista procesando automáticamente, los autores deben poder indicar exacto su interpretación y uso de la expresión dada en árbol. Deben poder registrar los significados de sus símbolos (nodos de la hoja del árbol de la expresión) y lazos matemáticos entre ellos de manera que sea autónomo las expresiones se pueden reconocer y procesar por el software.

### **3. ¿QUE ES LA WEB 2.0?**

La Web 2.0 es la evolución que se ha dado de aplicaciones tradicionales hacia aplicaciones que funcionan a través del Web enfocadas al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio. La Web 2.0 ofrece servicios básicos internos como sitios de redes sociales, wikis (tipo de paginas de Internet que permiten a las personas agregar, modificar o remover contenido disponible sin necesidad de un previo registro), herramientas de comunicaciones y folksonomies que permite a las personas compartir información en línea de maneras que antes no estaban disponibles.

Se empezó a hablar por primera vez de la "Web 2.0" en la "Conferencia sobre Web 2.0" en el año 2004. Por aquel tiempo se suponía que había que usar "la Web como plataforma", lo que quería decir "aplicaciones Web". Mientras que la primera ola de la Web dependía del navegador, la segunda ola extiende el uso de las aplicaciones a la Web y permite una nueva generación de servicios y oportunidades de negocio.

La Web 2.0 surgió a partir de la Web 1.0, la vieja Internet, en la que primaba el desarrollo tecnológico y se refería a una Web de "solo lectura"; el término "Web 2.0" nació a raíz de una lluvia de ideas realizada en una sesión entre O'Reilly<sup>[r03]</sup> y Medialive International que son productores de exhibiciones tecnológicas y conferencias. O'Reilly<sup>[r03]</sup> quería organizar una conferencia sobre la Web, y se estaban preguntando qué nombre darle. Y para que las novedades de la Web en ese momento cobraran importancia y captaran la atención de todos, O'Reilly y Medialive le llamaron la Web 2.0.

Constantemente estaban surgiendo nuevas aplicaciones y sitios con sorprendentes funcionalidades. Y así se dio la pauta para la Web 2.0 conference de 2004. Esta conferencia no solo fue exitosa sino que ya tuvo seguimiento en la Web 2.0 Conference del 2005 celebrada en Octubre, siendo la más reciente en Octubre de 2007 en California, EEUU.

## ¿Qué es la Web 2.0?

---

En la charla inicial del Web Conference se habló de los principios que tenían las aplicaciones Web 2.0:

- La Web es la plataforma
- La información es el procesador
- Efectos de la red movidos por una arquitectura de participación.
- La innovación surge de características distribuidas por desarrolladores independientes.
- El fin del círculo de adopción de software ("Servicios en beta perpetuo")

La Web 2.0 está caracterizada principalmente por un enfoque social, en el cual nuevas tecnologías aportan para producir cambios importantes en la forma de publicar las webs, de encontrar información, de comunicarnos. La infraestructura tecnológica de la Web 2.0 incluye "software servidor, sindicación de contenidos, protocolos de mensajería, aplicaciones cliente, navegadores estándares con plug-ins y extensiones". Todas estas tecnologías le permiten ofrecer capacidades distintas de las que ofrecen los sitios Web tradicionales, como son almacenamiento de información, creación y capacidades de disseminación.

La tecnología que hay detrás de la Web 2.0 trata, sobre todo, de tecnologías que están siendo utilizadas por personas en el mundo, y de la velocidad con la que se asimilan muchas de ellas.

Hay ocho funcionalidades tecnológicas que aparecen típicamente en un sitio Web 2.0:

- Sindicación y agregación de datos en formato RSS/ATOM RSS es el acrónimo de Really Simple Syndication y de Rich Site Summary. Es un formato utilizado para distribución de contenido, y también es un estándar definido para dar actualizaciones a contenidos Web.



- Técnicas de programación de aplicaciones con Ajax que aumentan la velocidad Ajax significa Asynchronous JavaScript y XML” es una técnica de desarrollo que sirve para crear aplicaciones Web interactivas.” Permite que solo se actualice la parte que se necesita y no todo, haciendo este proceso más fácil y rápido, y que las aplicaciones Web puedan funcionar, cada vez más, como escritorios.
- Hojas de estilo CSS para añadir estilo a las páginas Web.
- Representación DOM de HTML y XHTML. Se crea para asegurar un estándar común para páginas Web animadas que se crean con tecnologías que combinan HTML, hojas de estilo y scripting.
- Representational State Transfer (REST) es un enfoque para obtener contenido de información de un sitio Web. El objetivo detrás de REST es definir un estilo de arquitectura Web de cómo debería comportarse una aplicación Web bien definida.
- XML para construir lenguajes de marcado a medida para describir tipos de datos.
- Combinación de aplicaciones “mashup” (amalgama) para crear nuevos servicios Los mashups o amalgamas son combinaciones de aplicaciones Web que integran el contenido en un sitio único tomando distintos pedacitos de distintos lugares.
- Web semántica, lenguaje que pueden entender las máquinas también. Según la W3C, la Red Semántica es una evolución de la Red existente. La Red Semántica aportará estructura al contenido fundamental de las páginas Web. Los agentes de software buscarán páginas y ejecutarán tareas sofisticadas en nombre de los usuarios (Tim Bernes-Lee et al, 2001). Esta es una de las áreas que más se está investigando en estos momentos, inclusive se están adentrando tanto y se ha vuelto quizás tan amplio que algunos como Wikipedia mencionan que “podría hablarse de la Web 3.0 para la Web semántica”.

## ¿Qué es la Web 2.0?

---

Unidos con las tecnologías existen dos aspectos filosóficos que marcan la Web 2.0:

El primero es "democracia". Ahora se tienen grandes ejemplos que muestran que el trabajo realizado por los aficionados puede superar al de los profesionales, cuando tienen el sistema adecuado para canalizar sus esfuerzos. La Wikipedia puede que sea el más famoso, y es libre, cosa que significa que la gente realmente puede leerla.

Otro punto en el que la democracia parece ganar es en decidir qué son noticias y qué no. No es recomendable leer noticias sino son de páginas confiables y reconocidas. Reddit fue fundada por Y. Combinator, Reddit es como un lector de RSS de toda la red, con un filtro de calidad. Páginas Web similares son Digg, una Web sobre tecnología que es casi tan popular como Slashdot, y del.icio.us, una Web recolectora de enlaces que provocó el movimiento del "tagging". Mientras que el mayor atractivo de la Wikipedia es que es muy buena y libre, estas Webs sugieren que los votantes tienen un papel más importante que los editores humanos.

El ejemplo más dramático de la democracia en la Web 2.0 no está en la selección de ideas, sino en su producción. Se presentan casos en que la información que aparece en Webs personales es tan buena o mejor que lo que aparecen en periódicos y revistas. Y ahora es evidente que los links más populares de Reddit son de Web personales en vez de artículos de revistas o de noticias nuevas.

El otro aspecto filosófico es no maltratar a los usuarios. No sólo de formas obvias, como hacer que se registren, o someterlos a que vean publicidad molesta. Muchísimas de las Webs más populares en los 90 estaban plagadas de molestas marcas que hacían que la Web cargase muy despacio y que le enviaban al usuario mensajes como: "esta es nuestra Web, no la tuya". La raíz del problema estaba en que las Webs sentían que estaban dando algo gratis, incluso hoy día, una compañía que ofrece algo gratis se comporta de forma arrogante.

Una forma es darle a los usuarios más poder, darles lo que quieren. Si no lo hacen pero su competidor sí, están en problemas.

iTunes es del estilo Web 2.0 en este sentido. Finalmente uno puede comprar canciones sueltas en vez de comprar el CD completo. La industria discográfica detestaba la idea y se resistió cuanto pudo. Pero era obvio lo que los usuarios querían, así que Apple se impuso. Aunque realmente debería decirse que iTunes tiene el estilo "Web 1.5", el término "Web 2.0" aplicado a la música probablemente signifique que los grupos dejen de emitir sus canciones a través de la radio y en su lugar las hagan libres.

Finalmente la tendencia da un muy buen ejemplo de la aplicación general de la Web 2.0. Google fue la pionera en tener todos los componentes de la Web 2.0: su negocio principal está muy de moda si se habla en términos de Web 2.0. y por supuesto Google desencadenó el fenómeno Ajax con Google Maps.

Web 2.0 significa usar la Web para lo que realmente fue diseñada, y Google lo hace. Ese es su secreto. Ellos van al compás de las tendencias en lugar de permanecer estáticos, predicando su modelo de negocio en los medios de comunicación escritos o intentando ir en la dirección contraria al llevar a sus clientes a juicio.

Alrededor de la Web 2.0 se ha visto que las personas pueden compartir, negociar y colaborar entre ellas mismas. El principal enfoque de esta es el social, y se mueve en varios contextos (Web 2.0 Antonio Fumero y Genis Roca, 2007) el sociotécnico y el económico – empresarial.

En el contexto sociotécnico la Web 2.0 está moviendo a la gente a que participe en la Web, a que no sea una Web de lectura solamente, sino una en la que el usuario pueda expresar su opinión, dar su toque personal a las páginas, y compartir sus recursos e información con otros usuarios, un ejemplo de esto son los *blogs*, y los *wikis*, los cuales se volvieron muy populares por la facilidad para cambiar el contenido por parte del usuario y la masificación del internet. La Web 2.0 surge de la acción social en un contexto tecnológico nuevo. Antes en lo que llaman la Web 1.0 el usuario solo se dedicaba a leer los contenidos

## ¿Qué es la Web 2.0?

---

de las páginas y no interactuaba mucho con estas, ahora con este revolucionario cambio, el webmaster no es el único que puede modificar el contenido de las páginas, gracias a las distintas aplicaciones que manejan el contenido de estas, como AJAX por ejemplo.

La Web 2.0 no solamente se mueve en un contexto sociotécnico, esta se mueve también en un contexto económico empresarial, que tiene tres aspectos (Web 2.0 Antonio Fumero y Genis Roca, 2007):

- Los intereses de las personas
  - ✓ Iniciativas sin expectativa de retorno económico
  - ✓ Iniciativas con expectativa de retorno económico centrado en la persona
  - ✓ Iniciativas con expectativa de retorno económico centrado en el proyecto
- Los intereses de las organizaciones
  - ✓ Modelos de negocio basados en la audiencia
  - ✓ Modelos de negocio basados en la tecnología
- Los intereses del capital

Cada uno de estos intereses es distinto, responde si hay o no un tipo de negocio detrás del uso de la Web 2.0 y cual podría ser. Con respecto al interés de las personas algunos ofrecen servicios gratis, otros para ganar fama y a través de esto dictar conferencias, etc. Y otros para efectuar un proyecto financiado por empresas y llevar a cabo un negocio. Las empresas mantienen siempre su interés en la rentabilidad que la Web les pueda ofrecer para multiplicar su capital, comprando empresas que ya funcionan o financiando distintos proyectos.

Sin embargo, este no es el final de los intereses de las compañías respecto a la Web, por el contrario ya la Web 2.0 tiene su fecha de vencimiento y se abre paso a la Web 3.0, que trae consigo aún más beneficios que los mencionados

anteriormente. Mientras el mundo piensa en la forma más dinámica para mudar sus sistemas a SOA y como integrar plataformas diversas para concretar la Web 2.0, algunos CIOs de carácter más competitivo e innovador ya trabajan el concepto de Web 3.0.

La Web 3.0 o más popularizada como Web Semántica es una Web que acerca al usuario a los resultados más satisfactorios de la tecnología, dotada de mayor significado, en la que los usuarios de Internet pueden hallar respuestas a sus necesidades específicas de manera fácil y veloz, producto de una definición mas enfocada y rica. Al proveer a la Web de la semántica como herramienta, se facilitan los caminos a soluciones de problemáticas habituales en la localización de información, producto de la utilización de una infraestructura común, que le permite al usuario compartir, procesar y transferir información de forma sencilla. Este concepto innovador está basado en el significado, se soporta en lenguajes universales que dinamizan un modelo sin semántica que en muchas ocasiones solo genera un sentimiento de frustración y pérdida de tiempo ante la dificultad de encontrar lo que el usuario busca puntualmente.

El concepto de "mundo plano" ha cambiado para siempre la forma de hacer negocios, multiplicando las oportunidades y también las amenazas, los clientes y los competidores están a solo un clic de distancia. La Web ha dinamizado de manera determinante las comunicaciones, eliminó las distancias y mutó la forma en la que los individuos se desempeñan en cada una de las actividades. El paradigma de reacción en tiempo real se ha metabolizado con costos sumamente atractivos. El pago de impuestos o la compra de valores en Wall Street desde el lobby de un hotel en Santiago de Chile, o una playa en Cartagena solo depende de access point de Wi-Fi, pero ya no sorprende a nadie. El acceso a cientos de miles de recursos es algo descontado, sin que el idioma o ubicación sea determinante. El crecimiento de la Web ha sido caótico, la cantidad de páginas que se suben a Internet se mide por segundo, lo cual ha provocado una real sobrecarga de información,

## ¿Qué es la Web 2.0?

---

expresada en formatos múltiples y heterogéneos provocando serios problemas de interoperabilidad.

La Web 3.0 apunta a resolver estas dos problemáticas permitiendo al usuario delegar tareas que hoy realiza de forma manual al software. La transformación semántica de la Web, permite al software procesar su contenido, metabolizarlo, combinarlo y generar deducciones lógicas para evitar problemas cotidianos de forma automática.

Con el advenimiento de la Web Semántica, los sistemas de medición de audiencia (Ej: [www.realmedia.com.br](http://www.realmedia.com.br)) en Internet cobrarán vital importancia, los sistemas de medición de audiencias por internet, son la primera expresión inteligente que conllevan el espíritu de la Web semántica, proveer al usuario satisfacción al navegar internet, de la forma que más le agrada y con resultados concretos. Las empresas, buscan herramientas que le permitan refinar cada vez más el comportamiento y preferencias de los usuarios.

La Web va a adoptar rápidamente este nuevo cambio y el mundo open source será un catalizador de esta innovación: La información que aparece en Internet va a poder interpretarse desde la PC sin necesidad de intervención humana, para que esto suceda, es necesario normalizar la semántica de las páginas Web. En la medida que esta semantización de la Web se realice con código abierto, el resultado será más rápido y efectivo.

Conforme sigan evolucionando las herramientas de gestión de contenidos empresariales (ECM) y el concepto sea adoptado masivamente (mainstream), se verá el nacimiento de una plataforma unificada basada en la Web, donde van a converger diversas tecnologías para administrar información estructurada (workflow, BPM, Digital Asset Management, Document Management, Web Content Management, Imaging) e información no estructurada (Online Meetings, Groupware, Webconferencing, Knowledge Management, Email, Instant Messaging, Search Technologies). Será el nacimiento de una nueva cultura tecnológica donde la información más que la tecnología per sé (Information Technologies - IT) será el centro de la sociedad que se conoce hoy en día, el principal activo a gestionar, entendiendo y

aceptando su ubicuidad gracias a la Internet como el nuevo habilitador de un verdadero mundo colaborativo, lo que algunos llaman Web 3.0.

La tecnología XMP es una pieza importante que va a posibilitar la concreción de la Semantic Web.

XMP quiere decir Extensible Metadata Platform o Plataforma Extendible de Metadatos y básicamente es una tecnología de etiquetado que permite incluir datos sobre un archivo, metodología conocida como metadata, dentro del archivo mismo. Con XMP tanto los programas de escritorio como los sistemas basados en servidor que trabajan con archivos adquieren un nuevo método de capturar, compartir y potenciar los metadatos de cualquier archivo. Estos metadatos pueden ser autor del archivo, fecha, idioma, palabras llave, y en general cualquier dato que describa al archivo y sus derechos de uso. XMP es un estándar del W3C.

La Web semántica es objeto de análisis en los departamentos de negocios de las empresas más competitivas del mundo (léase bien, "de negocios") es sin duda la transformación más importante de Internet que se enfrentará, desde ARPAC

### CONCLUSIONES

- La Web desde sus inicios ha estado encaminada a brindar un mejor servicio a los usuarios finales, surgiendo la Web 2.0 pero también evolucionando hacia versiones más avanzadas.
- La Web de hoy día funciona más como punto de encuentro que como fuente de información, formando así redes sociales donde el principal fin es el compartimiento de datos.
- La Web 2.0 representa el cambio evolutivo de las tecnologías y técnicas que ha surgido alrededor de la Web, marcando una nueva etapa en este mundo joven en tecnología.
- La Web 2.0 no solo representa un cambio tecnológico, sino como este afecta los hábitos humanos y su interacción creando una nueva cultura digital
- Los estándares son un conjunto de normas creadas por organizaciones internacionales que permiten crear e interpretar documentos en la Web 2.0.
- La arquitectura participativa es aquella donde los integrantes de una misma red comparten todo tipo de información mutuamente.
- Los modelos de programación permiten el mejor manejo de datos y la comunicación entre integrantes de una misma red



## BIBLIOGRAFÍA

- FUMERO, Antonio, ROCA, Genís con la colaboración de SÁEZ VACAS, Fernando. *Web 2.0*. Fundación Orange. España., mayo de 2007. 136 p.
- COBO ROMANÍ, Cristóbal; PARDO KUKLINSKI, Hugo. 2007. *Planeta Web 2.0. Inteligencia colectiva o medios fast food*. Grup de Recerca d'Interaccions Digitals, Universitat de Vic. Flacso México. Barcelona / México DF. Septiembre 2007. 162 p.
- Seminario Internacional Virtual Educa Cono Sur 2006. *Implicancias en el entorno educativo y algunas de sus herramientas*. Buenos Aires, 16 de noviembre de 2006. 88 p.
- <http://www.maestrosdelweb.com/editorial/web2/>
- <http://www.mkm-pi.com/mkmpi.php?article260>
- Definición de Web 2.0 en Wikipedia:

[http://es.wikipedia.org/wiki/Web\\_2.0](http://es.wikipedia.org/wiki/Web_2.0)

- [http://www.estandaresweb.es/introduccion\\_estandares.html](http://www.estandaresweb.es/introduccion_estandares.html)
- <http://www.cioal.com/cioaldocs.nsf/0/21EFCD299ECEDECF8525728600114B5E>
- [r01] <http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-Web-20.html> pág. 1. Capítulo 1, What Is Web 2.0.
- [r02] Cobo Romani, Cristóbal;Pardo Kuklinski, Hugo. 2007. *Planeta Web 2.0. Inteligencia colectiva o medios fast food*. Grup de Recerca d'Interaccions Digitals, Universitat de Vic. Flacso México. Barcelona / México DF. Pág. 28.
- [r03] ¿Qué es Web 2.0?, traducción del artículo de Tim O'Reilly «What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software». [http://es.wikipedia.org/wiki/Web\\_2.0](http://es.wikipedia.org/wiki/Web_2.0) pág. 1.

- **Scripta Nova REVISTA ELECTRÓNICA DE GEOGRAFÍA Y CIENCIAS SOCIALES.** Universidad de Barcelona. ISSN: 1138-9788. Depósito Legal: B. 21.741-98 Vol. VIII, núm. 170 (54), 1 de agosto de 2004. Dirección electrónica: <http://www.ub.es/geocrit/sn/sn-170-54.htm>
- **TRABAJO DE INVESTIGACIÓN - Hernan Folgueira.** "REDES COMPAÑERO A COMPAÑERO COMO SOPORTE DE SISTEMAS DISTRIBUIDOS". Dirección electrónica: <http://www.tyr.unlu.edu.ar/tyr/TYR-p2p/folgeira-p2p.htm>
- <http://www.killersites.com/translations/spanish/porQueCSS.htm>
- <http://www.laWebera.es/manuales/css1/apendices2.php>
- [http://www.javahispano.org/contenidos/es/exprimiendo\\_javaWebstart/](http://www.javahispano.org/contenidos/es/exprimiendo_javaWebstart/)
- [http://java.sun.com/products/javaWebstart/1.2/es/docs/Readme\\_es.html](http://java.sun.com/products/javaWebstart/1.2/es/docs/Readme_es.html)
- [http://developer.mozilla.org/es/docs/Tutorial\\_de\\_XUL:Estructura\\_XUL#C.C3.B3mo\\_se\\_procesa\\_XUL](http://developer.mozilla.org/es/docs/Tutorial_de_XUL:Estructura_XUL#C.C3.B3mo_se_procesa_XUL)
- [http://www.masternewmedia.org/es/2006/04/11/rss\\_usos\\_y\\_aplicaciones\\_no.htm](http://www.masternewmedia.org/es/2006/04/11/rss_usos_y_aplicaciones_no.htm)
- <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>
- <http://www.mmcdesign.com/blog/2007/12/18/entendiendo-el-funcionamiento-de-z-index-en-css/>
- API: ARTICULO: API, Interface de Programación de Aplicaciones. Miércoles, 7 de noviembre de 2007 a las 08.49. Dirección electrónica: <http://www.elwebmaster.com/referencia/api-interface-de-programacion-de-aplicaciones>
- <http://www.ilhn.com/blog/2007/12/29/interlink-headline-news-n%C2%BA-4716-del-sabado-29-de-diciembre/>
- <http://www.bajenlo.com/107/descargar/opera/>
- <http://www.w3c.es/divulgacion/guiasbreves/XHTML>

- <http://en.wikipedia.org/wiki/OPML>
- <http://www.opml.org/spec2>
- Representational State Transfer (REST). PROYECTO FIN DE CARRERA. Alberto Cubo Velázquez. Dirección electrónica: <http://209.85.207.104/search?q=cache:z9colRR2SeIJ:trajano.us.es/~antonio/REST.ppt+REST+aplicacion+funcionamiento+web&hl=es&ct=clnk&cd=4&gl=co>
- <http://www.atsistemas.com/listados.aspx?s=articulos&n=1#>
- Rafael Navarro Marset. Modelado, Diseño e Implementación de Servicios Web 2006-07 REST vs Web Services. <http://www.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>
- Revista Digital Universitaria. Artículo *Construcción de Servicios Web con SOAP*. Alejandro Botello Castillo. 31 de marzo del 2002 Vol.3 No.1 Dirección electrónica: <http://www.revista.unam.mx/vol.3/num1/art3/>
- Artículo: Ruby on Rails: agilice el desarrollo de aplicaciones web. 28 Sep, 2007 <http://www.towebs.com/blog/index.php?op=ViewArticle&articleId=44&blogId=1>
- [http://jair.lab.fi.uva.es/~beasanc/PAM/practicaweb1/html/ajax\\_4.html](http://jair.lab.fi.uva.es/~beasanc/PAM/practicaweb1/html/ajax_4.html)
- <http://www.programacion.net/tutorial/jaxp/3/#DOM>
- [http://mx.geocities.com/c\\_sinuhe/mathml.htm](http://mx.geocities.com/c_sinuhe/mathml.htm)