

HTML5 y la Web Semántica

¿Por qué es importante HTML5?

Hace años hablábamos de la [Web 2.0](#). Todo gracias a que Tim O'reilly acuñó el término identificando un cambio en el mercado, otro tipo de Internet que se adapta al cambio de los tiempos y a muchos nuevos perfiles que habitan en el mundo digital.

La Web 2.0 no inventaba ni proponía cambios a la web, pero sí identificaba un ciclo que estábamos viviendo donde la mentalidad para hacer proyectos en línea había cambiado.

La Web 2.0 acercó a muchos mercadólogos y hombres de negocios a Internet. Incluso a nivel marketing se intentó seguir el ciclo hablando de Web 3.0 o Web semántica. Pero no funcionó del todo esta fórmula ni se llenó de la misma gloria. Que suerte tuvimos.

Por eso quiero hablarles de HTML5.

La magia de esta unión de tecnologías agrupada en un solo término no es inventar nada nuevo, sino unir esfuerzos que por primera vez tienen a toda la industria empujando hacia una dirección y que identifican un cambio que ya sucedió en como debemos hacer la web hoy mismo.

"HTML5 es el presente de la web y si no estás asimilando lo que está pasando ya eres parte de la vieja generación de desarrolladores. Eso tendría que tenerte preocupado." Estamos viviendo el Momento HTML5, es una época maravillosa para desarrollar en la web.

Entender HTML5 es entender que hoy nos conectamos desde teléfonos móviles, tabletas, eBooks, netbooks, computadores y otra gama de dispositivos. Es entender que se acabó la profesión de webmasters siendo reemplazada por equipos multidisciplinarios. Las empresas tecnológicas hoy cuentan con frontends, backends, sysadmins, mobile devs, community managers y arquitectos de información en los proyectos que están reinventando mercados y generando tráfico e ingresos sólidos.

Podemos hablar de todas las empresas gigantes de la web, podemos enumerar a Microsoft, Google, Apple, Adobe, Facebook, Yahoo, Mozilla, Twitter, Amazon y miles de proyectos tecnológicos que independientemente de sus objetivos hoy respiran HTML5, lo apoyan y tienen propuestas que los hacen competir en código en el navegador

más cercano a tu pantalla. Es una de sus más importantes estrategias de posicionamiento y es una de sus más fuertes apuestas de reclutamiento. ¿Dije además que HTML5 tiene [logo de superhéroe](#) y que venden camisetas para apoyar al movimiento? Bienvenido al momento HTML5. Si crees que HTML sigue representando las siglas del Hypertext Markup Language no has entendido nada. HTML5 es una época maravillosa de la web que tienes el privilegio de estar disfrutando. No es tecnología, ni un estándar solamente, es una época. Disfrútala y embriágate de sus tags.

Introducción a HTML5

HTML5 es la actualización de HTML, el lenguaje en el que es creada la web. HTML5 también es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de Javascript que manejan diferentes APIs en el navegador.

La versión anterior y más usada de HTML, HTML4, carece de características necesarias para la creación de aplicaciones modernas basadas en un navegador. El uso fuerte de Javascript ha ayudado a mejorar esto, gracias a frameworks como [jQuery](#), [jQuery UI](#), [Sproutcore](#), entre otros.

Flash en especial ha sido usado en reemplazo de HTML para desarrollar web apps que superan las habilidades de un navegador: Audio, video, webcams, micrófonos, datos binarios, animaciones vectoriales, componentes de interfaz complejos, entre muchas otras cosas. Ahora HTML5 es capaz de hacer esto sin necesidad de plugins y con una gran compatibilidad entre navegadores modernos.

¿Qué pasó con XHTML?

La versión corta es que la W3C ha reemplazado sus esfuerzos por continuar el avance de XHTML para promover y difundir el estándar HTML5.

La versión larga tiene que ver con la misma burocracia de esta organización. Verán, la W3C (El World Wide Web Consortium) está compuesta de un comité de gente muy inteligente, humana, democrática y que se dedica a crear estándares de lo que consideran correcto en la web y se olvidaron de ver qué estaba pasando allá afuera. Luego de la versión 4 de HTML, su esfuerzo fue hacia XHTML que combinaba el estándar HTML con la sintaxis de XML. Esta sintaxis es muy

estricta, hay que cerrar todos los tags, no hay que ser desordenados y si alguna vez intentaste usar XHTML 1.1 en la versión estricta estoy seguro que todas las malas costumbres que tenías impedirían que el documento llegue a funcionar en los navegadores mostrando errores de sintaxis. De hecho, cuando pensaron en la versión de XHTML 2 pensaron en errores draconianos. Esto significa que si hacías algo mal, el navegador directamente iba a mostrar un error maldiciéndote. Y la web no funciona así.

De hecho, uno de los mayores problemas que tenemos cuando nos ponemos a codificar para que nuestras páginas sean compatibles con diferentes navegadores es prever como los navegadores iban a improvisar cuando hemos cometido errores. Los webmasters de hace muchos años no hacían todo bien. Los editores WYSIWYG no eran perfectos, existía mucha improvisación, es la misma naturaleza humana reflejada en la web y así funcionan la mayoría de páginas que hoy seguimos visitando. Y la W3C quería arreglar esto, corregir el curso de la web.

Pero era más importante pensar en evolucionarla, seguir progresando. Por eso surgió el WHATWG (Web Hypertext Applications Technology Working Group) de la mano de Ian Hickson que propuso trabajar en estándar paralelo para crear Web Apps. En el 2004 su propuesta fue rechazada y decidió hacerlo todo por su cuenta, con sus amigos. Así, la WHATWG empieza a trabajar en el estándar de HTML5 en un modelo dictatorial donde Ian Hickson tiene la última palabra de esperar a que un comité burocrático se ponga de acuerdo.

Si, una dictadura. Aunque suene terrible y todos deberíamos estar en contra, es la base misma de HTML5 y ha resultado bastante bien. Porque la dictadura de Ian Hickson ha avanzado muy rápido, sus propuestas son ágiles e incluso cuando políticamente suena incorrecto, el peso de la comunidad sigue siendo muy fuerte.

Hay una anécdota importante relacionada con el tag `<time>` por ejemplo. Este es uno de los nuevos tags semánticos incluidos dentro del estándar para encapsular información sobre horas. Hickson no estaba de acuerdo con este tag y decidió eliminarlo, diciendo que el tag `<data>` era más correcto, más amplio. Casi siempre su mando ha sido super correcto y no solemos criticarlo, pero cuando hizo lo de tag `<time>` el

Internet estuvo en su contra y le demostró que estaba mal. La decisión fue rectificadada y el tag <time> está de vuelta. Pero insisto, no es un consenso gigantesco que requiere unificar las agendas de un montón de personas. Nos guste o no, funciona mejor que la W3C en muchos sentidos.

Por eso, la W3C eventualmente se sumó al esfuerzo. Dejaron de lado el áfan por seguir adelante con el XHTML2 y están apoyando a HTML5. Incluso los grupos se mantienen, la W3C, el WHATWG y además el HTML Working Group. ¿Otro más? Bueno, sucede que el HTML WG es un grupo formado por la W3C que se encargó de tomar el esfuerzo del WHATWG y seguir adelante con HTML5. Son burocráticos y no cambiarán del todo, aunque los intentos han sido muy acertados.

Si hoy vas al sitio del WHATWG encontrarás avances importantes sobre HTML5. Y luego en la W3C, de la mano del HTML WG verás también importantes avances sobre HTML5. ¿Cuál es la versión correcta? No importa realmente, lo importante es que siga evolucionando y que las cosas que allí se proponen sean implementadas en el navegador.

Los estándares ahora son modulares

Tal vez lo más importante que ha quedado con toda la telenovela anterior es que hoy, HTML5, los APIS HTML5 e incluso CSS3 son muy extensos. Revisa los documentos del estandar HTML2 por ejemplo y compáralo con las últimas versiones, son gigantes, amplios y si los imprimimos acabarán con el papel de tu oficina. Y esto es bueno, porque son exhaustivos, hay mucho aprendizaje en el camino y todo es modular. Al hablar de que sea modular nos referimos a que se van trabajando en etapas. Hay muchas cosas de HTML5 que está escrito en piedra, otras más que todavía están en discusión y otras que poco a poco van siendo implementadas.

Jamás pienses que HTML5 está listo, jamás pienses que está implementado del todo en todas partes. Lo importante es entender que muchos módulos están implementados, muchos módulos se están implementando y nos tocará ver más cambios y evolución en cada nueva versión de tu navegador favorito.

HTML5 es un montón de cosas. Para muchas de ellas ya estamos listos y hay que aprovecharlas. Además el estandar se subdivide bastante. Un ejemplo concreto es el estandar de LocalStorage, un API del navegador

que permite que guardes información en el disco del usuario. LocalStorage es parte de HTML5 pero no está en su estandar. Es un estandar diferente. ¿Por qué? Algunos dicen que es porque ya el estandar de HTML5 era muy grande y no querían hacerlo crecer más, otros dicen que es para que el grupo de trabajo pueda desarrollarlo más calmadamente, en su propio ecosistema. Y todo esto está funcionando muy bien para nosotros que nos toca codificar al final del día. Recuerda además, una cosa es que tengamos estándares y otra cosa que funcionen. Hasta que no se implementen en versiones de navegadores en producción, no podemos usarlo. Un ejemplo es CSS4. Aja, ya se está trabajando en la versión 4 de esta tecnología sin que tengamos todo el estandar CSS3 funcionando. Tiene propuestas super interesantes que discutiremos más adelante, pero aún no funciona en ningún navegador. Cuando Microsoft, Mozilla, Google, Apple, Opera y demás productores de navegadores empiecen a aprovechar estas ideas y llevarlas a la práctica va a ser más divertido. Piensa que con los estándares tenemos la oportunidad de ver el futuro antes de que este implementado.

"HTML5 estará listo en el 2014"

El último y más famoso recurso de los detractores de HTML5 es que "no está listo". ¿Ustedes creen que HTML5 es un software que una empresa "termina"? HTML5 es un spec vivo, cuya implementación depende de los navegadores y del cual gran parte se puede usar ya mismo sin problemas. Eso incluye CSS3. Es [posible usar CSS3 en Internet Explorer viejos](#). Es posible usar HTML5 [desde ya](#).

El draft HTML5 de la W3C no se cerrará hasta dentro de unos años, pero para entonces nuestros navegadores tendrán nuevas tecnologías, porque no es Flash o Visual Basic, es una tecnología viva que crece con su uso.

Maduren, sean profesionales, usen HTML5 y no teman llamarle a las cosas por su marca.

Web Semántica y nuevas etiquetas HTML5

¿Tengo que volver a hacer todos mis sitios con HTML5?

No. HTML4 y HTML5 son 100% compatibles entre sí. Todo el código que tienes en HTML normal seguirá funcionando sin problemas en HTML5. La retrocompatibilidad es una de las mejores decisiones que tiene esta

propuesta. Las cosas buenas que aprendimos con el estricto XHTML están disponibles, la improvisación que hacen los navegadores para mostrar errores se está regularizando.

¿Cómo empezar?

Para empezar a usar HTML5 lo único que tienes que hacer es colocar este: [DOCTYPE](#) antes de la etiqueta <html>:

```
<!DOCTYPE html>
```

Es un DOCTYPE mucho más simplificado que [XHTML](#) (cuyas reglas siguen siendo usadas) y te permite usar todas las habilidades de HTML5 sin que nada de lo que ya tienes programado deje de funcionar.

Incluso, este doctype es lo único que requieres para hacer un documento en HTML5. Si incluyes eso en un archivo .html, lo guardas (ocupará 15 bytes) y vas al validador de la W3C, te dirá que es un documento totalmente válido. En HTML5 no se revisa ni siquiera si abriste o no un tag <html>, un tag <head>, etc. Dejemos esto solo como una anécdota curiosa, aunque de poca utilidad, ya que no tiene sentido hacer un archivo solo con el doctype.

HTML5 es HTML, así que todos los tags que seguramente recuerdas: <body>, <p>, <div>, , <input> siguen funcionando como antes. Pero para implementar más semántica, se han incorporado muchas etiquetas nuevas. Las principales etiquetas HTML5 nuevas no tienen una representación especial en pantalla. Todas se comportan como un <div> o un . Pero cada una tiene un significado semántico superior a un simple div o span. <header>

Hacer cosas como <div id="header"> es un poco estúpido cuando el 99% de los proyectos web tienen una cabecera. <header> está diseñada para reemplazar la necesidad de crear divs sin significado semántico.

<hgroup>

Muchos headers necesitan múltiples títulos, como un blog que tiene un título y un tagline explicando el blog. <hgroup> permite colocar un h1, h2 y h3 dentro del header sin afectar el SEO, permitiendo usar otro h1 en el sitio. A nivel SEO, recomendamos utilizar h1 una vez por sitio o el h1 pierde prioridad de acuerdo a muchos comentarios que ha hecho Google directamente.

<nav>

Igual que `<header>`, `<nav>` está diseñado para que ahí coloques la botonera de navegación principal. Puedes colocar cualquier etiqueta dentro, aunque lo recomendado es usar listas ``.

`<section>`

Define un área de contenido única dentro del sitio. En un blog, sería la zona donde están todos los posts. En un vídeo de youtube, habría un section para el vídeo, uno para los datos del vídeo, otro para la zona de comentarios.

`<article>`

Define zonas únicas de contenido independiente. En el home de un blog, cada post sería un article. En un post del blog, el post y cada uno de sus comentarios sería un `<article>`.

`<aside>`

Cualquier contenido que no esté relacionado con el objetivo primario de la página va en un aside. En un blog, obviamente el aside es la barra lateral de información. En el home de un periódico, puede ser el área de indicadores económicos.

`<footer>`

Este es obvio. Es el pie de página y todo lo que lo compone.

Atención `<div>` no está muerto:

Estas nuevas etiquetas no significan que ya no se use `<div>`. Div siempre debe usarse cuando necesites una caja con objetivos de diseño gráfico o cualquier cosa que no tenga significado semántico. Sólo usa las etiquetas semánticas de HTML5 donde sean necesarias. Son fáciles de usar, no las olvides en tus próximos proyectos.

Las etiquetas nuevas importantes de HTML5

Las etiquetas semánticas y la costumbre a utilizarlas serán claves para posicionamiento en buscadores y buen desarrollo web. Eso si, no son la razón principal por la que todo el mundo habla de HTML5. Vídeo, audio, canvas y animación vectorial están en la lista de prioridades y en la boca de todas las personas que evangelizan su uso.

Otras etiquetas nuevas que se han popularizado son:

`<video>`

Inserta video sin necesidad de plugins. Es muy fácil usarla, pero cada navegador soporta codecs diferentes de vídeo, lo que hace necesario

recodificar un video en múltiples codecs. En un futuro capítulo hablamos un poco del drama que este tag está generando.

<audio>

Lo mismo que video, pero sin video. Puede usar múltiples formatos, en especial mp3, pero también depende del navegador. El MP3 tiene además protecciones de patentes, así que luego evaluaremos algunas cosas que debes considerar al usarlo en tus proyectos.

<input *>

Input ya existía como la etiqueta para insertar cajas de texto y botones. Ahora es más poderosa, con la capacidad de insertar cajas tipo "email" que se autovalidan, calendarios tipo "date", sliders, números, entre otras. Los formularios solíamos modificarlos y regularlos con mucho javascript y ahora desde el mismo HTML5 hacemos todo esto.

<canvas>

Un área de dibujo vectorial y de bitmaps con Javascript. Es un API de dibujo entero para Javascript. Mucha magia HTML5 se está produciendo en esta etiqueta.

<svg>

Una etiqueta, igual que , para insertar dibujos y animaciones vectoriales al estilo de Flash. Todo basado en el estándar abierto SVG (Scalable Vector Graphics), derivado de XML.

Más adelante hablamos de la magia en HTML5 y mencionamos ejemplos concretos con canvas y SVG. También hay capítulos dedicados a video y audio.