



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

## HTML Como Style - version 0.2 (12250 lectures)

Per Héctor Delcourt, [Armonth](http://sigt.net) (<http://sigt.net>)

Creado el 22/07/2003 16:09 modificado el 28/07/2003 05:01

*Estoy trabajando en un documento en docbook (sgml) sobre HTML orientado al diseño, especificaciones, estandares, etc...*

*En primer instancia este documento mas que enseñar HTML enseña actitudes, modos de trabajo y metodos que mejoren la manera de trabajar y la eficiencia...*

*Actualizacion: Nuevos contenidos, correcciones, etc.*

## COMO - HTML Style.

### Introducción

Lo más normal es estar acostumbrado/a a visitar páginas web, pero es posible que nunca hayas hecho una. En este manual pretendo enseñarte tres cosas: a hacer paginas web, a mejorarlas y sobretodo como usar las herramientas mas potentes y al mismo tiempo crear una web rápida, con contenidos y siguiendo los estándares.

En este manual aprovecharemos lo que conocemos de GNU/Linux y las herramientas que nos proporciona tanto Debian, como otras distribuciones o webs.

Una vez acabado, se ofrecerá recursos para que tu pagina ofrezca servicios de valor añadido como puede ser foros, libros de visita, etc.

### Copyleft 2003 bajo GPL

Héctor Delcourt a.k.a (Also Known As - Cambien Conocido Como ) Armonth.

Héctor Delcourt ( correo nuevo aun no disponible ).

Héctor Delcourt es actualmente miembro de Debianitas y webmaster de The Tux Clock y el proyecto Adiparso localizados en [www.tuxclock.tk](http://www.tuxclock.tk)

Se puede copiar, modificar o distribuir este manual bajo las condiciones de la licencia **GNU General Public License ( GNU GPL )**.

Si se desea hacer una copia total o parcial del documento se deberá adjuntar debidamente la identidad del autor así como las secciones de copyleft y revisiones enteras.

Se recomienda, que en caso de modificación, y para un reconocimiento mas exacto de autoría, añadas una sección **ANEXO** con las modificaciones realizadas.

Y por ultimo, si no es molestia, siempre en afán de aprender y ver nuevos puntos de vista, me gustaría recibir una copia de dicha modificación en mi correo :).

El autor no se hace responsable de los daños producidos por la utilización la información del documento.



[www.debianitas.net](http://www.debianitas.net) & [tuxclock.tk](http://tuxclock.tk) Copyleft 2003.

## Sobre este COMO

Este COMO esta escrito usando docbook, en el se muestran varios modelos de generación de paginas web y las ventajas o desventajas que conllevan, este documento estará en continua revisión por lo que se corregirá a medida que se pueda y se agradecerá a los que ayuden en la corrección del mismo.

Estará disponible tanto en HTML como en PDF en la web de The Tux Clock, también si tenéis los paquetes *docbook* y *docbook-utils* y deseáis convertirlo en otros formatos (dvi, html, pagina man, pdf, ps, rtf, tex, texi, txt, etc) también tendréis disponible el \*.sgml con la sintaxis corregida para que no provoque ningún problema en la conversión.

Si modificáis este documento y decidís enviarme una copia de la modificación, prefiero una copia del sgml modificado que no otros formatos.

## Sobre Docbook

Docbook es un interprete de SGML y XML que permite la creación de documentacion técnica ordenada que gana adeptos día a día, por ejemplo la documentacion del LinuxDoc, FreeBSD o Gnome están realizado con Docbook.

Docbook es un interprete sencillo que mediante un fichero con una sintaxis parecida al HTML (lenguaje de etiquetas ya sea SGML o XML) y unas útiles, tu puedes escribir en un texto un documento y que a partir de el, genere versiones en HTML, PDF, Paginas MAN, RTF, Texto plano, dvi, etc.

Si eres usuario de Debian, mediante un simple `apt-get install docbook docbook-utils` y un documento con sintaxis SGML o XML como puede ser este COMO en su variante 'html-style-como.sgml' y un simple comando se pueden general en otros formatos, ¿que desees este manual en pdf?, `docbook2pdf html-style-como.sgml` y te genera un `html-style-como.pdf`, la posibilidad de mediante un fichero poder realizar documentacion en otros tantos formatos nos proporciona una herramienta muy potente que no puede dejarse de lado.

---

## Historia

### GML, el principio de las etiquetas

En los años 70, una gran compañía conocida como IBM tenía un serio problema: almacenar inmensas cantidades de informacion...

Por ello IBM invento el GML o General Markup Language (Lenguaje General de Etiquetas) un lenguaje con el que clasificar todo y escribir cualquier documento para luego ser procesado rápidamente.

En 1986, el ISO (International Organization for Standardization) da carácter oficial a este lenguaje, mediante la norma ISO 8879-Standard Generalized Markup Language (SGML).

### SGML y algunos problemas

SGML es un lenguaje muy trabajado, adaptable a casi cualquier problema en especial el de la catalogación, pero tal ADAPTABILIDAD creaba un problema:

**Era realmente complejo** por lo que crearon un estándar nuevo el eXtensible Markup Language (XML) o Lenguaje de etiquetas extendible, que es el usado actualmente...

Realmente el XML es una redefinicion del SGML pero simplificado al máximo por lo que realmente XML es solo un 10% de SGML.

Nota: Este COMO podéis observar que es en SGML pero realmente es por que aprendí la sintaxis para hacer documentos en docbook usando SGML, para cualquiera incluso para mi XML es mas que suficiente para elaborar



buena documentacion.

## Nacimiento del HTML

Cerca de 1989 existían dos técnicas muy interesantes para elaborar documentacion: Los hipervinculos también conocidos como links, enlaces o anclas que permitían ir de un fichero a otro y un lenguaje de etiquetas potente (SGML), un usuario que conocía ambas llamado Tim Berners-Lee físico nuclear en el Centro Europeo de Investigaciones Nucleares publico un documento que expresa que esta trabajando en un sistema que permita el acceso a informes en linea que pudiera funcionar en máquinas conectadas por redes basadas en el protocolo TCP/IP.

Así, el HTML nació como una necesidad de intercambiar informacion en linea entre investigadores

A principios de 1990-1991 Tim Berners-Lee define el HTML (HyperText Markup Language o Lenguaje de Etiquetas de Hyper Texto) como un subconjunto del SGML así también se crea el llamado World Wide Web.

En 1991 otra vez Tim Berners-Lee introduce el primer navegador de HTML que trabaja solo en modo texto y solo en plataformas Unix (LineMode).

El HTML fue adoptado rápidamente por la comunidad y estandarizado por el w3c o World Wide Web Consortium.

Luego fueron saliendo varias versiones de la norma HTML, 1.0, 2.0, etc hubo una guerra de navegadores dirigida por Microsoft Internet Explorer y Netscape que explicaremos en el siguiente paso, luego de ello en 1998 salio la ultima versión HTML 4.0

Después HTML 4.0 solo ha salido una revisión menor (HTML 4.01), y a continuación surgieron Xhtml y Dhtml (Este ultimo NO es un estándar).

---

## Guerra entre navegadores

Capitulo de la historia que requiere una mención especial

Bill Gates declaro: "Eso de Internet es solo una moda pasajera", bien Netscape no creyó lo mismo, después de el primer navegador conocido en arquitectura i386 (si no me equivoco) Mosaic tuvo un problema con una marca comercial de mismo nombre y cambio a Netscape.

Netscape Navigator tuvo un éxito fulminante, mas del 90% de usuarios empezaron lo usaban y de una compañía recién nacida en 1994 en apenas un año se convirtió en la compañía de mas rápido crecimiento de la historia.

En 1995, sobre los 12.000 desarrolladores de software se habían unido a Netscape, la versión 1.0 gano mas de 20 premios y se habían distribuido mas de 15 millones de copias en todo el mundo.

Por si fuera poco mas de la mitad de empresas que salen en la revista Fortune (que habla sobre las empresas y personas con mas capital del mundo) eran clientes suyos.

Ante tanto éxito, Bill Gates cambio rápidamente de idea y vio que el futuro de su empresa estaba en la Red, compro una empresa que desarrollaba otro navegador y en base a eso empezó su propio navegador, el cofundador y CEO de Microsoft declaro:

*Tener la mayor cuota de usuarios en el mercado de navegadores es nuestra meta principal.*

Así, en una de las acciones declaradas como 'la mayor muestra de monopolio de la empresa de Redmond' se incluyo de serie Microsoft Internet Explorer en la series Windows 98.

Poco después y aun en 1995 Netscape declaro que los directivos de Microsoft se pusieron en contacto con Netscape para que **abandonaran** sus intentos futuros de volverse su navegador en "la alternativa", Netscape se negó.

AOL, declarándose siempre enemiga de Microsoft adquirió la empresa Netscape en 1999. Empezó una guerra entre



navegadores, tanto uno como otro añadían etiquetas no-estándar como la bgsound para cargar un sonido de fondo en Internet Explorer o blink en Netscape, uno invento los marcos (Netscape) otro las capas (Explorer) y empezó el caos.

Los desarrolladores de paginas web NO sabían como hacer que las paginas fueran compatibles estaban muy limitados y por culpa de Netscape y Microsoft el desarrollo de una pagina aumento en coste y dificultad un 40%.

Microsoft fue demandado y gracias al juez Jackson, Microsoft declarada culpable de monopolio, teóricamente debían separarse en dos empresas independientes...

Netscape por ultimo saco la versión 6.0 al mismo tiempo que la nueva versión de Internet Explorer, fue un rotundo FRACASO, mucha gente perdió la fe en ellos.

Mas tarde nació Mozilla, un navegador de código libre, a diferencia de Navigator, al no depender de una empresa, no se puede sobornar o derribar mediante las típicas tácticas de las empresas (nombre se la que usted desee, sea Microsoft o paquito-esta-forrado S.A) de chantaje, extorsión, etc.

¿Que hizo que fuera demandada Microsoft por monopolio?, realmente esto es una cuestión algo profunda: Un sistema que poseía (y aun posee) una GRAN cuota de mercado no puede imponer de serie el uso de ciertos productos ya que puede perjudicar gravemente varios sectores, os imagináis si incluyera de serie un cortafuegos, un antivirus, un programa de contabilidad, etc? bien: La mayoría de usuarios por desconocimiento o por pereza lo usaría, cuando hay alternativas y competencia se debe ganar por el medio de ser MEJOR producto no por venir de serie.

Ahora, año 2003 hace poco se anuncio que [Microsoft y AOL finalizan la guerra de los navegadores](#)<sup>(1)</sup> enlace de vnunet.es en muy resumidas cuentas: Microsoft ha pagado a AOL una suma nada despreciable por daños y prejuicios (631 millones de euros) y le permitirá utilizar durante 7 años sus programas.

Y así se acaba extrajudicialmente la guerra del monopolio, poco después toda la plantilla de Netscape es despedida o introducida a otros proyectos como podemos leer en barrapunto en el enlace [AOL se carga su grupo de Mozilla \(o eso parece\)](#)<sup>(2)</sup> pero no es todo lo parece...

En base a esto, se crea la fundación (sin animo de lucro) Mozilla en base a esto por fin los desarrolladores dejaron de ver a Netscape como una alternativa, ya que queramos o no, tener que realizar 2 versiones para dos navegadores pudiendo hacer una 100% estándar pero que no funcionara en ciertos navegadores antiguos (Netscape en las versiones no estándar) es una alegría para muchos.

Así bien, Netscape Navigator ha muerto después de meses y meses de ser un cadáver de una vieja gloria que no hay que olvidar, es una pena pero una ventaja: La gente no veía mas allá de Netscape y Internet Explorer, habiendo todos los basados en el motor Gecko (Mozilla, Firebird, Netscape, Galeon), otros como Konqueror, Opera o si saltamos a algunos como los MAC: Safari basado en Konqueror...

Por no hablar que hay navegadores en modo consola: Links, Lynx, Links con soporte gráfico, etc.

A destacar la opinión de **su** que ofrece sobre la guerra de navegadores en la versión 0.1b publicada en primera instancia en Bulma:

*La guerra de navegadores, no fué necesariamente mala. Quedó en evidencia, que el HTML, que tan solo estaba pensado para que físicos nucleares se pasaran datos entre sí, no estaba preparado para un consumidor que quería otra cosa. Entonces, cada compañía, añadió elementos propios al HTML, que solo soportaba su navegador.*

*Lo peor, es que se mezcló, en un lenguaje pensado para la estructura, una gran cantidad de elementos presentacionales (esto sí fué malo), y toda la maquetación, se empezó a hacer con tablas (con los problemas de accesibilidad que eso conlleva). Pero en el fondo, es normal que sucediera eso, ya que no había mecanismos para evitarlo (el W3C aún no existía, y el HTML 3.2, que era un desastre, fué su primera recomendación). Ahora ya existen tecnologías como CSS, que permiten lo que el público y los webmasters querían.*

## XML, HTML y problemas

Desde 1996 hasta hoy el w3c ha intentado normalizar el HTML, en 1998 empezó a trabajar y aun sigue en el XML.



Pero el HTML tiene una serie de problemas MUY graves uno de ellos es quizá el hecho de mezclar continuamente presentación y contenido, generando mucho código a procesar y bajar, duplicación de esfuerzos y otros...

La guerra entre los navegadores no ayudo, se hizo prácticamente imposible hacer una pagina que se viera bien para los dos navegadores, la comunidad se fragmento, y (repito) el coste de desarrollo aumento sobre el 40% según varios estudios.

Hoy en día, aun padecemos residuos de esos problemas: Webmaster's que no hacen sus paginas 100% compatibles o no se rigen por unas normas mínimas, paginas desarrolladas "optimizada para XXX navegador" (principalmente Internet Explorer o Netscape).

Y en resumen, planificación nula para separar contenido y estructura.

---

## Nuestra área de trabajo

### Herramientas básicas

Para realizar paginas web debemos tener un par de herramientas que son las siguientes:

- Un editor de texto, para crear las paginas: Nano, Vi o cualquier otro
- Un navegador como Phoenix, Mozilla, Konqueror, Galeon, para visualización
- \*Opcional solo si se usa PHP: Servidor Web con soporte PHP
- Verificador de Sintaxis: El parser para validar webs online del w3c

### ¿\*.htm o \*.html?

Existen varias web's que usan indistintamente una o otra extensión, la razón de ello es mas por asimilación y historia que otra cosa, en los Sistemas Operativos basados e incluyendo a MS-DOS (MS-DOS y Windows 3.x concretamente) las extensiones estaban (están) limitadas a 3 caracteres, GNU/Linux no tiene limitaciones de ese tipo.

El método correcto es .html, en caso de tener código php incluido se deberá procesar como \*.php o \*.phtml esto siempre que lo soporte el servidor.

### ¿Que es Internet?, ¿Que es la WWW?.

Algo que se debe tener en cuenta es que **LA WWW NO ES INTERNET**, una web, es una pagina con código html, que se trasmite por un protocolo (HTTP) en una aplicación de Internet tipo cliente/servidor, esto es (explicación poco técnica):

El cliente (navegador) pide una pagina (web) al host (servidor que aloja la web) y mediante Internet y el protocolo HTTP llegan a un acuerdo de negociación y por ultimo recibes la pagina, tu navegador tiene un motor de renderizado el cual representa las imágenes y los textos traduciendo la sintaxis HTML.

Existen muchos otros protocolos que seguramente usaras, http es para documentos como paginas web o imágenes, FTP para enviar/recibir ficheros sean o no html el "irc" es un sistema de reenvío de mensajes que permite una conversación en tiempo casi real (con un ligero lapsus o lag).

Existen muchos servicios aparte, cada máquina puede correr hasta 65335 puertos TCP/UDP por lo que hablamos de cientos de miles de servicios, algunos mas conocidos que otros, algunos considerados por la IANA como estándares ( por debajo del puerto TCP 1024 como puede ser el 117 IDENT, el 21 FTP o el que tratamos 80 HTTP).

Esto es una explicación resumida, para saber mas sobre Internet, redes y demás "parafernalia" aconsejo [google](#)<sup>(3)</sup> ya que (repito) esto es SOLO un MANUAL de PAGINAS WEB enfocado a realizar paginas de una manera cómoda y compatible.

---



## Estándares

Se debe elegir un estándar a seguir y cumplir en la elaboración de las paginas esto es importante, la primera lección a aprender seria:

*i. Una web debe ser accesible, si una persona tiene problemas para entrar posiblemente no vuelva a entrar y te borre de sus favoritos*

Los estándares del World Wide Web son llevados acabo por el w3c (World Wide Web Consortium) existen actualmente varios estándares de los cuales los principales son: Html 3.01, 4.01 y Xhtml 1.0.

Dentro de cada estándar hay casos comunes que se pueden usar o no: Cascade Style Sheet (CSS), tablas, marcos, etc.

Dentro de cada estándar existen tres categorías:

**Transitional:** Se permiten etiquetas antiguas y de estilo, interesante para cuando existen migraciones, pero se recomienda siempre que sea posible usar Strict.

**Strict:** Estándar estricto, no se permiten etiquetas antiguas o que pertenezcan a otro estándar.

**Frameset:** Versión del estándar aplicado a paginas con marcos.

Luego ahí que tener en cuenta unas cuantas cosas: Html 3.01 y otras revisiones están actualmente desactualizadas y estas incluida la versión 3.2 te causaran mas problemas que otra cosa.

Html 4.0 y su revisión menor 4.01 es un estándar maduro, pero a partir de las series 4.0 el uso de etiquetas de estilo esta desaconsejado ¿que significa esto?, pues que cualquier día podrías encontrarte con la agradable (aunque para ti no) situación de que todos los navegadores siguen los estándares y que tu pagina no se vea en ninguno.

En Html 4.0 y revisiones menores el estilo va en ficheros de estilo en Cascada (CSS) por lo tanto si usas Html 4.0 seria recomendable empezar a plantear el uso de CSS.

Xhtml es la reformulacion del Html como aplicación XML.

Este manual en primer lugar se basara en Xhtml Strict, con unas nociones básicas también usaremos CSS y quizá algunas técnicas en PHP.

## ¿WYSIWYG?

En GNU/Linux no existen muchos editores HTML tipo WYSIWYG (What You See Is What You Get - Lo que tu ves es lo que tu recoges) esto es, aquellos editores HTML que pones un código o haces click en un sitio y sale en pantalla como queda esa imagen en la web o ese texto, existen algunos como Quanta, Mozilla Composer, pero realmente no tanto.

Realmente me quedo con la frase de un amigo mio WYSIWY(don't)G o - Lo que tu ves es lo que tu NO recoges - ya que aunque en los editores de texto normal si suele ser así, en el HTML nunca sera así ya que se vera según resolución, medio, navegador.

Sinceramente algo que me he dado cuenta es que el tan nombrado "Dreamweaver" de Macromedia y todos los editores HTML tipo WYSIWYG producen código poco legible y muchas veces sucio, no es una buena idea: Cada carácter de mas puede provocar una subida de tamaño, a mas tamaño mas tardara en cargar eso nos lleva a la siguiente lección:

*ii. Toda web debe ser de acceso rápido, no existen trucos milagrosos, la única manera de que una pagina sea rápida es que se DESCARGUE rápidamente.*

Otro problema muy común es la duplicación de código al que suele pasar por usar estas herramientas, un ejemplo sencillo es que si pones un párrafo de digamos 50 palabras saldrá como ejemplo:



<font face="Verdana, Arial, Georgia" size="1"> las 50 palabras </font>

Pero si luego volvemos a darle formato es posible que acabe como:

<font face="Verdana, Arial, Georgia" size="1"><size="1"> las 50 palabras </font> </font>

En ese ejemplo hay un tag que define el tipo de fuente y su tamaño con su tag de cierre y otro que define el tamaño (otra vez) y su respectivo cierre, esta duplicación se agrava con el desarrollo de web's cada vez mas complejas, tablas cada vez mas anidadas...

*¿Tienes un html de 60KB con menos de pagina y media de texto? ¿No sabes por que? pregunta te por que viendo el código de mas arriba, en algunos casos puedes llegar a tener repetidos los tag hasta ¡100 veces! (cantidad típica hasta que uno se da cuenta en realidad podría ser miles de millones de tags repetidos) y aunque 100 etiquetas no sea nada para el motor de renderizado de nuestros navegadores puede ocasionar un retardo muy grande de descarga y en CPU lentas si provoca un retardo de renderizado.*

La única solución a eso es o bien usar el editor WYSIWYG y luego eliminar (limpiar) el código o directamente no ensuciarlo, optaremos por la segunda opción y además separaremos la presentación del contenido.

Este problema NO podremos provocarlo ya que en el caso anterior se habría creado un fichero CSS tipo:

```
p { text-font: Verdana, Arial, Georgia; font-size: 10pt; }
```

Y luego un <p>Este texto es un párrafo</p> por lo que las únicas etiquetas que realmente no tendremos que poner en todos los textos los Font Face, los Size, etc si no solo un <p> y </p> esto eliminara a la larga cientos y cientos de líneas de código, tablas, marcos y otros que ralentizarían la descarga y renderizacion.

Otra ventaja que esto ofrece es que si luego deseamos que los párrafos sean en otro tamaño o fuente solo deberemos modificar el fichero CSS y no TODAS las paginas.

## ¿Que Tecnologías usar?

Mentiría si dijera "una LAMP (Linux - Apache - Mysql - Php)" no seria partidista, pero intentare analizar las soluciones actuales desde mi punto de vista, a la larga acaba siendo la mejor solución salvo excepciones...

Un buen resumen y una buena lección seria:

*iv. Aprovecha y potencia las tecnologías que se ejecuten del lado del servidor, es muy posible que el usuario no tenga algún plugin correspondiente a la tecnología que usas pero si esta se ejecuta transparentemente (independientemente del Software que tenga el usuario) erradicaras muchas posibilidades de que el usuario no acceda a la informacion o no acceda como tu quisieras.*

## ¿Flash?

Sin dudarlo: NO, el flash es mas colorido y puede entrar al usuario por el lado artístico o visual al igual que han entrado por nuestros ojos las Interfaces Gráficas de Usuario (GUI), pero mejor veamos pro's y contra's.

- Pro: Entra por los ojos (visualmente)
- Pro: Tiene una buena cuota de mercado (clientes con soporte Flash)
- Contra: El peso una animación sencilla ya es bastante pesada
- Contra: Aun el peso siendo aceptable, se requieren mucha máquina\*
- Contra: Flash no es código libre, lo peor de eso en realidad es que no existen plugins para Flash en arquitecturas no-i386 (PowerPC como los Mac de Apple, Sparc, estaciones SUN, etc)...

\* Una animación estática requiere mas procesado gráfico que 5 imágenes grandes, un flash puede pesar "solo" 20KB y no procesarse bien (o saturar) un ordenador antiguo (inferior a un Pentium I 266 MHz).



Mi recomendación sería usar el Flash *\_solo\_* para animaciones pequeñas, o en casos que por algún motivo sea imprescindible pero nunca para el público, otra lección es:

*iv. No se debe moldear al usuario mediante nuestra web a nuestros gustos, no existe para el usuario nada más molesto (y que probablemente ocasione más posibilidades de que no vuelva a vuestro sitio) que se le obligue a bajar un plugin que no desea.*

Y esto yo lo aplico como: Tengo el plugin de Flash en mi Mozilla/Firebird pero aplicando una extensión (Flash click for view) que impide que se descargue y procese cualquier flash si luego me interesa el susodicho flash se descarga y procesa con un simple click, esto también evita esas miles de animaciones en flash que pueden llegar a saturar nuestra línea siendo la mayoría de publicidad.

Otro punto que me guardo para el final es que tres de las potenciales herramientas de una web no son desde el punto clásico "explotables" en una web en flash.

La primera son los **enlaces**, un enlace permite saltar de una web a otra para hacer la información más importante lo más accesible posible pero, si tenemos un flash con 2 o 3 artículos en flash y queremos enlazar a uno de ellos ya no podemos hacerlo desde otra web (al menos no fácilmente).

La segunda es **la copia de contenido**, si un artículo de opinión, un manual o cualquier información que creamos interesante no se puede copiar se pierde uno de los grandes sentidos de la web, esto es del todo incorrecto: Se puede copiar, pero para ello se debe bajar el fichero \*.swf o desde los temporales (cache) del navegador, crackearlo o en su defecto copiar la información palabra a palabra...

La tercera y casi tan importante como las demás es **los buscadores** un buscador es un motor que se encarga de buscar y poner en un índice las páginas interesantes como google es quizá uno de los buscadores más conocidos, bien los buscadores no pueden *leer* flash.

¿Y que importa si un buscador puede o no meter en su directorio tu página?: Mucho, si no se está en los buscadores se pierde una gran cantidad de visitas, estar en los buscadores es sinónimo de importancia, de estar entre las páginas importantes y eso se consigue solo de una manera: Esforzándose.

Flash es una tecnología interesante pero que no llega a todos los usuarios, por el lado artístico está interesante pero si se quiere llegar a los internautas con la ayuda de los buscadores, que te enlacen y en general "que encuentren tu página interesante" debe ser accesible, no poner muros y flash por desgracia las pone.

Resumiendo: Flash para webs artísticas puede servir, pero no lo recomiendo, por otro lado quizá sería interesante el uso de Flash MX + Soporte XML pero no lo he probado y al no tener soporte para visualizarse en PDA's, PowerPC y cualquier no-i386 le quita parte por no decir todo su encanto.

## ¿Java & Javascript?

Si: Solo cuando sea necesario, tanto Java como Javascript tienen un problema actualmente: la guerra de máquinas virtuales de Java que existe entre las distintas empresas, cada una ofreciendo una distinta y muchas veces el código hecho en una no funciona en la otra y a la inversa, por lo que no se puede asegurar que vaya a funcionar.

Actualmente es un cachondeo (que no se me vayan a cabrear los programadores de Java) el show que hay montado: Sun (creadores de Java) denunciando a Microsoft por que su máquina virtual java es incompatible, Microsoft no se sabe si le da largas o directamente lo ignora y de mientras otras como la máquina virtual java de IBM ha estado durante un cierto tiempo con frecuentes incompatibilidades.

Por lo tanto: Recomendaría Java para hacer applet's y añadidos que se ejecuten en el servidor y de la forma más transparente posible en los usuarios por otro lado NO y digo NO se os ocurra usar esos applet's java tipo pijada-java-only-ie como por ejemplo ese código java que hace como que nieva en la web; eso solo causa que la web vaya lenta, no pone ningún valor añadido a la página y provoca problemas.

Por otro lado recuerda: Mucha gente suele tener desactivado las máquinas virtuales JV.



- Pro: Potente.
- Pro: Arraigado (tecnología famosa, muchos usuarios contentos con java, etc.)
- Contra: Para una web te arriesgas a hacer algo incompatible con una u otra máquina virtual java.
- Contra: Se ejecuta del lado del cliente, es posible que el usuario no tenga máquina virtual java de ningún tipo (ni la de Microsoft, ni la de Java, ni la de IBM, etc.) y por tanto lo que quieras mostrarle no puedas mostrárselo.\*

Resumiendo: Es una tecnología potente, útil y usable, pero te arriesgas a que no llegue a todos los usuarios, valora razonablemente bien si merece la pena utilizar java en la parte de tu sitio que creas necesario utilizarla y no uses esta tecnología para pijadas como esa que he dado de la nieve (que a los usuarios les molestan los java pesados y inútiles).

\* Realmente esto no es culpa de JAVA, ni siquiera de SUN si no de las máquinas virtuales que hay que no son compatibles, por otro lado si todo el mundo usara la máquina virtual de SUN seria perfecto: La VM de SUN es potente, multiplataforma, etc.

## ¿ASP?

En diciembre de 1996 Microsoft introdujo la tecnología ASP (Active Server Pages), tiene algunos puntos fuertes pero uno muy bajo.

- Pro: Es parecido a PHP, puedes incluir trozos de html y código asp en un mismo fichero sin problemas.
- Pro: Funciona del lado de servidor por lo cual es transparente al usuario
- Pro: Aunque su punto de desarrollo es el VBScript se puede utilizar otros lenguajes como Perl o JavaScript.
- Pro: Al ser programación de lado del servidor si tu no lo deseas el código resultante utilizado para la generación dinámica de paginas web esta oculto.
- Pro: Permite trabajar con bases de datos como SQL.
- Contra: Solo se puede ejecutar bajo Windows.

Este ultimo y único CONTRA es quizá el mas serio, ya que de entrada la mayoría de hostings tanto gratuitos como de pago trabajan sobre una LAMP (Linux - Apache - Mysql - Php) además los hostings de pago suelen ser mas caros en su variante WAMP ( Windows - Apache - Mysql - Php) que bajo LAMP y por si fuera poco aun son mas caros y difíciles de conseguir con Windows + IIS + ASP + SQL

Una buena tecnología sobretodo si tienes un servidor BLINDADO y ya preparado...

## ¿PHP?

Tiene las mismas características que ASP y aparte de ello es libre, esta bajo Windows, GNU/Linux, \*BSD's, etc, recomendable y perfectamente factible para el uso en paginas dinámicas.

El único contra real es el que tienen todas las tecnologías basadas en programación tales como ASP: Que si programas mal, puedes provocar bug's que si no se tienen en cuenta con medidas extra de blindado en el servidor pueden acabar siendo la causa de un ataque conseguido, pero no debería por que, un servidor mínimamente decente tendrá bien programado los servicios de PHP y sera difícil que TU provoques un bug que permita entrar.

(Obviamente las empresas de Hostings y mas las gratuitas no se van a dejar piratear todo su negocio solo por que un aprendiz de programador en php se equivoque en una linea).

Por lo que este ultimo contra por favor "olviden lo" solo piensen que podría pasar en servidores montados "de cualquier manera" pero esto no suele ser la tónica diaria.

## ¿ColdFusion y JSP?

ColdFusion de Allaire y Java Server Pages (JSP) de Sun, son dos tecnologías propietarias parecidas también a ASP y a PHP, no he tenido oportunidad nunca de probarlas por lo que me guardare comentarios...



## CSS, CGI y otras tecnologías

CSS (Cascade Style Sheet) es una tecnología aprobada en el estándar w3c que permite dar formato rápido (Separar presentación de contenido) y ahorrar trabajo al webmaster, también permite aligerar las paginas... en caso que un navegador no soporte CSS (a estas alturas casi ninguno no lo soporta) lo único que puede pasar es que vea la pagina sin formato, pero seguirá teniendo la chicha de la pagina a su alcance (el contenido).

Por otra parte CGI es una tecnología para establecer comunicación entre un servidor web y un programa, de tal modo que este último pueda interactuar con Internet. También se usa la palabra CGI para referirse al programa mismo, aunque lo correcto debería ser script.

CGI fue durante años la única alternativa para tener foros y paginas dinámicas, pero actualmente están algo desfasados, a la larga programar en CGI es programar en una tecnología que quizá desaparezca, ya que actualmente da mas problemas un script en CGI que uno en cualquier otra tecnología.

También los CGI tiene muy mala fama y tampoco puede uno fiarse por la fama, pero cuando mas de un conocido que en programación aplicada al web me supera en creces asegura que son auténticos agujeros de seguridad en potencia y viendo la de problemas que han causado en varios de mis trabajos hace tiempo no dice mucho en su favor.

Existen otras tecnologías, pero creo que ya he tratado las mas conocidas, sin olvidar claro esta las bases de datos como MYSQL, SQL, Oracle, etc.

---

## Sintaxis y básico: Doctype's, html, cabecera, menu, cuerpo...

### Sintaxis

Abrimos nuestro editor de texto y guardamos el fichero como prueba1.html luego dentro tendremos la sintaxis básica de toda pagina web:

```
<html> <head> </head> <body> </body> </html>
```

Explicamos que significa cada etiqueta: **html** define el documento, esto nos dice que es un documento HTML (pagina web) ;absolutamente toda nuestra web estará entre estas dos etiquetas!.

*v. Es una buena idea considerar cada pagina como cinco partes principalmente: el "todo", la "cabecera" (head), "cuerpo" (presentación), "road map" y "contenido".*

El "todo" es el Doctype que define que estándar usamos y las otras cuatro partes del documento que están entre <html> y </html>.

La "cabecera" o head es todo lo que esta entre <head> y </head> en ella se define toda la informacion "invisible" al contenido como los metadatos, informacion de que documento es, datos para los buscadores, enlaces al/los documento/s CSS, etc.

El "road map" o mapa de navegación es una parte que debemos dejar como menu de navegación entre secciones, recomiendo que sea corto (no mas de 10 secciones) esto estará entre <body> y </body>.

Y "contenido" esta zona sera la que estará entre <body> y </body> a diferencia del road map aquí se mostraran las imágenes, textos o lo que desees mostrar.

## DocType ¡MUY IMPORTANTE!

**Situación:** Tu pagina esta perfectamente, cumple todos los estándares y debería verse bien en todos los navegadores los cuales también soportan estos estándares, pero algo falla ¿el que?.



**Respuesta:** El Doctype. Doctype es una abreviación de declaración de tipo de documento (DOCument TYPE), normalmente se valida una pagina mediante un parser como el del <http://validator.w3.org> el cual muestra que errores has provocado...

Es muy posible que tu pagina se vea bien en varios navegadores pero eso no significa que no haya errores por no seguir los estándares, en una conexión de cliente-servidor entre navegador y servidor web, este le manda al navegador la web, el navegador advierte al servidor que la descarga se ha realizado satisfactoriamente y mientras renderiza (muestra en pantalla) mediante su motor gráfico lo que el código html muestra realmente.

¿Que suele pasar?, una etiqueta DocType incorrecta o directamente sin ella mete en un problema al navegador, el navegador se pregunta: **¿y esto que es?**, por la extensión sabe que es una web pero **no sabe que estándar usa**; Esto habilita a un modo de compatibilidad hacia atrás, esto solo provoca errores, renderizados incorrectos y ralentización.

Por ello ¡pon un DocType correcto! un Doctype contiene solo dos lineas, una que especifica que tipo de fichero es y otro que da una URI de muestra.

Doctype's correctos son:

En el propio DocType se explica cual es: Html 4.01 Strict o Transitional y Xhtml 1.0 Strict o Transitional, también hay un Html Frameset y Xhtml Frameset pero no encontré un Doctype correcto para este.

Recuerda, el DocType se pone JUSTO antes de la primera etiqueta de la pagina web que SIEMPRE sera <html> recuerda que hay que RESPETAR las mayúsculas y minúsculas del DocType.

A continuación DocType validos para: Html 4.01 Strict, Html 4.01 Transitional, Xhtml 1.0 Strict y Xhtml 1.0 Transitional respectivamente.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Los Doctype siempre se deben poner ANTES de la etiqueta <html>

---

## Creditos, versión y actualizaciones

### \$ID == html-style-como.sgml, versión 0.2

Ultima actualización Jueves 24 Julio 2003 20:03:15 PM

- 28/07/2003 Correcciones varias, algo de ortografía y añadido mejoras
- 25/07/2003 Agrego un comentario sobre Cold Fusion y JSP
- 24/07/2003 Agrego la opinión de syu sobre la guerra de navegadores
- 24/07/2003 Correcciones de concepto realizas siguiendo recomendaciones vertidas en Bulma mediante comentarios
- 22/07/2003 Publico la versión 0.1b en Bulma con ID 1826
- 21/07/2003 Añadido resumen de la "guerra de navegadores"
- 04/07/2003 Añadida Historia del XML y HTML
- 30/06/2003 Añadida Historia del GML/SGML
- 26/06/2003 Añadida explicación sobre \*.htm y \*.html
- 29/06/2003 Añadido resumen de tecnologías: Flash, Java y Javascript



- 01/07/2003 Añadido resumen de tecnologías: Asp y Php
- 03/07/2003 Añadido resumen de tecnologías: CSS, CGI y otros
- 05/07/2003 Añadida la importancia de uso de DocType

## Falta:

Añadir lo importante a saber por orden de las 5 categorías:

1º Categoría (Doctype añadido) falta <html xmlang=>.

Añadir categoría de Cabecera, roadmap, contenido etc tratando temas como:

Metadatos

Añadir varios CSS

Título

Pragma y Cache control para ahorrar problemas en Proxy-Caches

También se debe añadir información de CSS, uso de foros tipo PHP+Mysql (instalación básica), tema ASCII & BINARY MODE de los FTP, modo de realizar una conexión FTP "simple" (sin GUI por en medio, lo básico: ftp, user \*\*\*\*, pass \*\*\*, mput, mget, get, put, etc)

Otras cosas interesantes a añadir:

Tipo de letras: Letras mejores para lectura y letras mejores para impresión

Tipo de usuario: Actitudes, síntoma del niño ciego (curiosa evolución anti-molestias de los navegantes :P), metodología...

---

### Lista de enlaces de este artículo:

1. <http://www.vnunet.es/Actualidad/Noticias/Inform%C3%A1tica%20profesional/Empresas>
2. <http://barrapunto.com/articles/03/07/15/2240249.shtml>
3. <http://www.google.com>

---

E-mail del autor: armonth\_ARROBA\_gmail.com

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1826>