

Optimizando nuestra web



RODRIGO GALINDEZ

rodrigo@simplelogic.com.ar
Desarrollador web, Rodrigo es un apasionado por todo lo que vaya después de http://.

EN ESTE ARTICULO HAREMOS UN REPASO DE LAS HERRAMIENTAS PARA PROBAR NUESTROS SITIOS EN DIFERENTES PLATAFORMAS Y ESCENARIOS, Y DAREMOS CONSEJOS PASO A PASO PARA MEJORARLOS Y OPTIMIZARLOS A FONDO. EL RESULTADO: NUESTROS SITIOS SERAN ACCESIBLES A CUALQUIER TIPO DE AGENTE DE USUARIO.

Navegadores alternativos: Firefox, Opera, Safari. Medios de conexión no tradicionales: celulares, PDAs. Lectores de pantalla. Navegadores en modo texto. ¿Cómo asegurarnos de que nuestro sitio web se comporta de la manera en que deseamos ante esta avalancha de agentes de usuario? ¿Cuáles son las técnicas que nos brindan los estándares web para asegurar que el contenido de nuestro sitio será accesible para ellos?

Durante el desarrollo de este artículo, comentaremos los diferentes escenarios posibles con los que se encuentra el desarrollador web y daremos consejos sencillos de implementar, a fin de **optimizar nuestras páginas para que luzcan más profesionales**. También les mostraremos técnicas para hacer que nuestro sitio sea accesible a diferentes públicos y mercados.

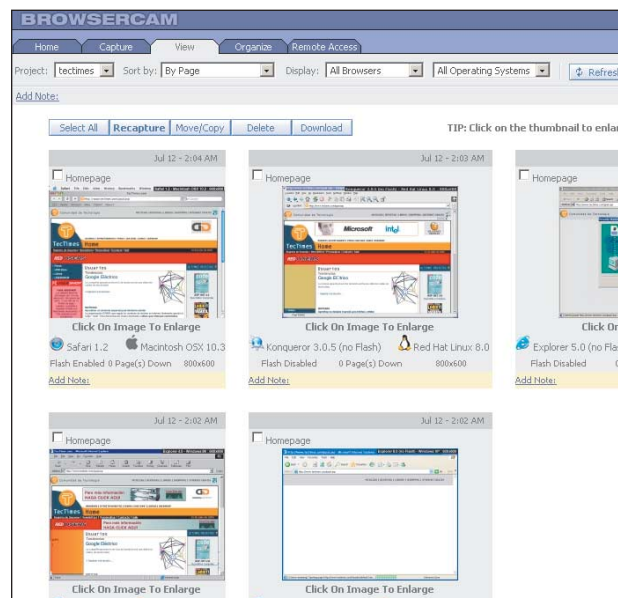
MÚLTIPLES NAVEGADORES EN NUESTRA PC

Para asegurarnos de que nuestro sitio se muestra de la manera en que lo deseamos, primero deberemos probarlo exhaustivamente en diferentes escenarios. Lo primero que podemos hacer es instalar varios navegadores en nuestra PC. Esta es una buena alternativa, ya que nos permite introducir cambios y ver los efectos en forma rápida y cómoda.

La lista de navegadores es realmente interminable, pero muchos se basan en un mismo motor de renderizado –como lo es Gecko, desarrollado por Mozilla Foundation, o KHTML, del equipo de KDE–, por lo que tener un navegador instalado como muestra representativa de un motor es suficiente. Los navegadores y sus correspondientes motores de renderizado incluyen:

- **Navegadores basados en Gecko:** Firefox, Mozilla Suite, Epiphany, K-Meleon, Netscape Browser. Todos estos navegadores se encuentran disponibles para varias plataformas (Windows, Linux, Mac) y en varios idiomas. Poseen un excelente soporte para estándares web, y son muy rápidos.

- **Navegadores basados en KHTML:** Safari y Konqueror. Safari es el navegador reinante en el mundo Mac, el cual viene instalado en las últimas versiones de OS X, específicamente a partir de Mac OS X 10.3. Konqueror es un completo navegador de archivos y navegador web, disponible en el entorno KDE. Lamentablemente, si



➤ **Browser Cam trabajando en el sitio de TecTimes. El programa nos muestra la apariencia en distintos navegadores y sistemas.**

estamos trabajando en ambientes Windows, no tendremos la opción de instalarlos. De cualquier manera, podemos usar un servicio basado en la Web para emular la salida en pantalla de éstos, el cual comentaremos en el apartado siguiente.

- **Navegadores basados en Presto:** por ser un motor de renderizado no libre, el único navegador que lo usa es Opera. Sin embargo, se encuentra implementado en la versión CS 2 de Adobe GoLive, como también en Macromedia Dreamweaver.

- **Navegadores basados en Trident:** Internet Explorer y Maxthon. Maxthon es un navegador que se apoya en el motor de renderizado de Internet Explorer, y agrega funcionalidades extra, como gestos del mouse y capacidad para instalar plug-ins.

Como sabemos, el navegador más utilizado es Internet Explorer. Por esta razón, deberemos prestarle especial atención, y también tener en cuenta que muchos usuarios no navegan con la última versión de éste (hasta la fecha, la versión 6). Por lo tanto, sería deseable que tuviéramos instaladas las versiones 5, 5.5 y 6 de IE en nuestra PC,

porque, aunque se trate del mismo navegador, cada versión interpreta de manera diferente las hojas de estilo en cascada, o CSS. Para tener las diferentes versiones de Internet Explorer, no es necesario instalar varias versiones de Windows. Simplemente deberemos descargar cada versión de Internet Explorer y copiarla a una carpeta separada de nuestro disco, desde donde las ejecutaremos. En el repositorio de navegadores de Evolt, en <http://browsers.evolt.org>, podremos encontrar todas las versiones que nos hagan falta, empaquetadas junto a sus respectivas DLL para poder ejecutar varias ediciones del navegador sin problemas.

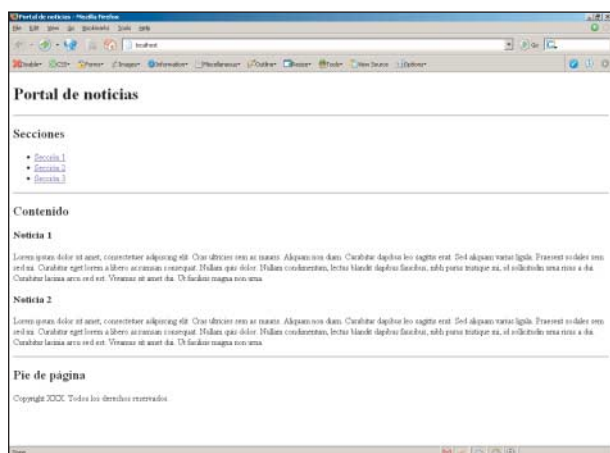
Entonces, en resumen, en nuestro entorno de desarrollo sería deseable tener instalados: Internet Explorer, Firefox y Opera, lo cual es una muestra bastante representativa de la población de navegadores de Internet.

LA ALTERNATIVA EN LA WEB: SERVICIOS DE CAPTURA DE PANTALLAS

Si bien podemos instalar varios navegadores en nuestro ambiente de trabajo, también tenemos la alternativa de utilizar algún servicio en la Web que nos genere capturas de pantallas en diferentes plataformas y navegadores. Esto es especialmente útil cuando, por ejemplo, queremos ver cómo se comporta nuestro sitio en ambientes Mac o Linux.

El más conocido es Browser Cam (www.browsercam.com), el cual captura la apariencia de nuestro sitio en los sistemas operativos y navegadores que le indiquemos. La lista de configuraciones disponibles es realmente grande, e incluye versiones antiguas y actuales de varios navegadores. Como característica especial, también tiene las versiones de los navegadores sin el plug-in de Flash instalado. Esto es realmente interesante, puesto que si hemos desarrollado una versión "no Flash" de nuestro sitio (que sería lo ideal), también podríamos probarla. El servicio es pago; sin embargo, se puede probar por 24 horas sin ninguna limitación en el número de capturas.

Por otro lado, iCapture (www.danvine.com/icapure) se enfoca en brindar solamente capturas de pantalla bajo la configuración Mac/Safari. Aunque no tiene las funcionalidades de Browser Cam, iCapture es totalmente gratuito.



➤ Como vemos, el esqueleto de nuestro portal de noticias se entiende perfectamente aunque el navegador no interprete CSS.

NAVEGADORES EN MODO TEXTO Y LECTORES DE PANTALLA

Aunque tal vez la mayoría de los visitantes a nuestro sitio estén usando navegadores en modo gráfico, no debemos olvidarnos de aquellas personas que emplean lectores de pantalla o navegadores en modo texto para acceder a su contenido.

Muchos lectores se preguntarán: ¿quién navegaría la Web en modo texto? Terminales viejas sin capacidad de procesamiento gráfico, algunos modelos de teléfonos celulares y lectores de pantalla para gente con discapacidad visual, son sólo algunos ejemplos.

Ahora bien, quizá no podemos asegurar que nuestro sitio luzca como lo deseamos en estos casos, pero ¿podemos hacer algo para mejorar la experiencia de usuario? Claro que sí, y es acá donde entran en juego las nociones de usabilidad y accesibilidad para sitios web. Veamos un par de recomendaciones sencillas:

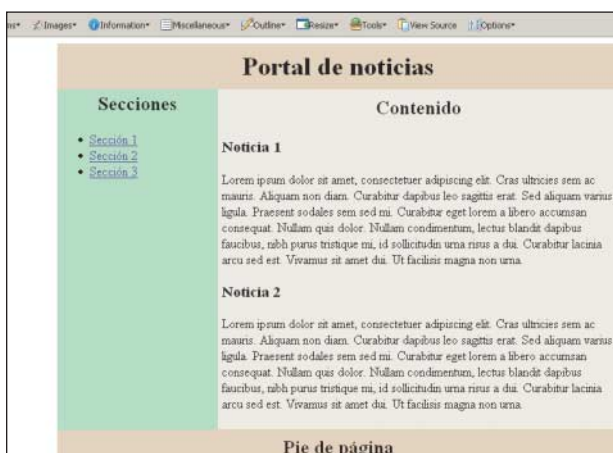
■ **Proveer textos descriptivos para nuestras imágenes.** Esta es la regla de oro que debemos seguir. Cuando una imagen de nuestro sitio no se carga, o cuando el usuario tiene la carga de imágenes deshabilitada, el texto alternativo describe en pocas palabras la imagen que está faltando. El atributo que se debe colocar en el elemento IMG es ALT. Veamos un ejemplo:

```

```

De esta manera, cualquier navegador en modo texto, como Lynx, mostrará el texto alternativo de la imagen sin inconvenientes. Como dato adicional, no nos olvidemos de que Google también hace un uso exhaustivo del atributo ALT en, por ejemplo, su servicio de búsqueda de imágenes.

■ **Asegurar que el contenido tiene sentido sin CSS.** Un sitio sin los estilos CSS aplicados debe poder leerse y entenderse perfectamente, como si se tratara de un documento estándar de Word. Es decir, sus secciones deben ser fácilmente identificables por sus títulos y párrafos, y éstos deben tener una coherencia lógica. Para ello, deberemos primero pensar en el esqueleto del sitio sin los estilos, y



➤ Nuestro portal diseñado con estilos en CSS, perfectamente "entendible" para los browsers actuales.

después en su diseño. Tomemos el ejemplo de un portal de noticias con dos columnas; su esqueleto sería de la siguiente manera:

- Cabecera
- Columna izquierda (secciones)
- Columna derecha (contenido)
- Pie de página

Para lograr esto, debemos olvidarnos de las tablas para maquetar. **Las tablas fueron creadas para representar datos tabulares, no para maquetar un sitio entero.** Hoy, la interpretación de XHTML y CSS en los navegadores actuales es lo suficientemente sólida como para que podamos dejar las tablas atrás. Agrupando las secciones de nuestro sitio mediante el uso de etiquetas <DIV> y luego colocando estos bloques de texto mediante CSS, conseguiremos un diseño accesible para lectores de pantalla y navegadores en modo texto.

■ **Incrementar la navegabilidad.** Esto quiere decir que el sitio debe contener ayudas, generalmente en la forma de enlaces, que le permitan acceder a determinadas secciones del sitio en forma rápida y sin perderse. Una buena idea es implementar la técnica conocida como **breadcrumb navigation**, la cual sirve para orientar al usuario dentro del sitio web, mostrándole claramente en qué página se encuentra y cuáles han sido las páginas visitadas anteriormente antes de llegar a la actual. Debido a su antigüedad, Lynx (<http://lynx.browser.org>) es tal vez el navegador en modo texto más conocido del mercado. Está disponible para varias plataformas y es totalmente gratuito. Sin embargo, si no deseamos instalarlo, también podemos ver cómo se comportaría nuestro sitio en estos navegadores, utilizando el servicio Lynx Viewer (www.delorie.com/web/lynxview.html).

EL MISMO RESULTADO EN VARIOS NAVEGADORES

Si estamos utilizando CSS para maquetar nuestro sitio (eso sería lo ideal), podemos recurrir a los siguientes trucos:

■ **Eliminar márgenes iniciales no deseados en los elementos.** Un truco muy simple y efectivo es darle valor 0 a las propiedades **margin** y **padding** de los elementos HTML que usaremos en el sitio, al

ENLACES RELACIONADOS

- Fuentes que vienen instaladas con Windows
www.autoitscript.com/autoit3/docs/appendix/fonts.htm
- Recomendaciones para el uso de alt cuando mostramos imágenes (en inglés)
www.cs.tut.fi/~jkorpela/html/alt.html
- Prácticas comunes sobre diseño web (en inglés)
www.webdesignpractices.com/index.html
- 25 errores comunes en los desarrollos web
www.minid.net/articulos/1190/25-errores-comunes-en-los-desarrollos-web
- Browsershots, servicio para capturas de pantallas gratuito <http://browsershots.org/>

comienzo del archivo CSS. Puesto que el valor inicial de estas propiedades varía entre los navegadores, utilizando este sencillo truco nos aseguramos de que todos los elementos se iniciarán obligatoriamente en 0. Más adelante podremos especificar un tamaño de **margin** o **padding** para los elementos HTML que nos hagan falta, sobrescribiendo las correspondientes reglas en CSS. Un ejemplo:

```
*{
    margin: 0;
    padding: 0;
}
```

Como vemos, utilizar el selector universal * (asterisco) nos permite indicarle al navegador que todos los elementos del sitio tendrán sus propiedades **margin** y **padding** restablecidas a 0 desde un principio.

■ **Unidades relativas manejables.** Como sabemos, la mejor unidad para representar tamaños de tipografías en pantalla es EM. Sin embargo, al ser esta medida relativa, cada navegador la interpreta a su manera, y a veces puede resultar en tamaños no esperados. Para ello, lo que podemos hacer es obligar a que un valor de EM sea igual a una unidad en píxeles, digamos, 10px, al principio del archivo CSS. Las unidades EM seguirán siendo relativas, los usuarios de Internet Explorer podrán aumentar el tamaño de las tipografías (lo que no podrían hacer si estuviéramos usando una medida como PX, por ejemplo), y nosotros podremos controlar los valores de EM como si estuviéramos trabajando en píxeles. Veamos un ejemplo:

```
body {
    font-size: 62.5%
}

p {
    font-size: 1.1em;
}
```

Lo que estamos haciendo con la primera regla es disminuir el tamaño de todos los elementos hijos del elemento <body> a un 62,5%, o sea 10 px, o 1 EM. Ahora bien, <p> es un elemento hijo de <body>, por lo tanto, su tamaño inicial será de 1 EM. Al tener la equivalencia 10 px = 1 EM, es decir, un número redondo y manejable, se nos hace fácil pensar los tamaños en términos de píxeles, pero seguir manejándolos en EM: 1,1 EM sería el equivalente a 11 px, 1,2 EM a 12px, y así sucesivamente.

■ **Especificar fuentes alternativas y familias genéricas.** Es importante recordar que no todos los usuarios tendrán las fuentes tipográficas que estamos especificando en nuestro archivo CSS. Pongamos el ejemplo de la fuente Lucida Grande, típica de Mac OS, que no viene con las instalaciones de Windows. Especificar una fuente alternativa y una familia de fuente genérica puede ayudarnos a controlar cómo se deberá mostrar nuestro sitio en estos casos. Por ejemplo, veamos el caso de esta regla relacionada con las fuentes para el elemento <body>:

```
body {
```

```
font-family: Lucida Grande;
```

```
}
```

Podríamos mejorarla de esta manera:

```
body {
```

```
font-family: Lucida Grande, Lucida Sans,
```

```
sans-serif;
```

```
}
```

Lo que hicimos fue indicar que, si no se encuentra instalada la fuente Lucida Grande, el navegador deberá optar por Lucida Sans, y de última, si no encuentra ninguna de las dos, deberá elegir cualquier fuente *sans-serif* instalada en el sistema, como Verdana. Las familias de fuentes son: *serif*, *sans-serif*, *monospace*, *cursive* y *fantasy*.

HERRAMIENTAS INDISPENSABLES

Si usamos Firefox, existe una extensión indispensable para el desarrollador web. Se trata de Web Developer (<https://addons.mozilla.org/extensions/moreinfo.php?id=60>), una extensión que permite la edición de código CSS en la misma ventana del navegador, lo que nos permite ir vien-

Contacto

Usted está aquí: [Página de Inicio](#) > [Contacto](#)

[Página d'inici](#) > [Casos d'estudi](#) > www.perits.org

PGi: [Inicio](#) > [Contactar](#) > [Mapas](#), [Teléfonos](#)



Breadcrumb Navigation aplicado en varios sitios. Vemos que en todo momento se le está informando al usuario en dónde está y de dónde vino.

do los cambios a medida que los introducimos. En Internet Explorer también contamos con una herramienta parecida, que es la Web Accessibility Toolbar (www.nils.org.au/ais/web/resources/toolbar).

CONCLUSION

Es una buena idea probar nuestros sitios en varios navegadores y en diferentes escenarios, particularmente si estamos usando CSS en forma exhaustiva. Esperamos que en este artículo hayan encontrado trucos interesantes para reducir la cantidad de problemas entre los diferentes navegadores. ■

Primera Exposición y Feria

Productos y Servicios relacionados a la Industria de Video Juegos, PC y Triple w en Sudamérica.

TORNEOS

Arcade Daytona: **Organiza Nivel X**
 The Elixir (Desarrollo Argentino): **Organiza Korpos**
 FIFA GAMES: **Organiza Fifa Games**
 PC / Painkiller: **Organiza 3DGames**
 PC / StarCraft: **Organiza 3DGames**
 PC / Warcraft: **Organiza 3DGames**
 Game Cube / Super Smash Brot Game cube: **Electronic Things**
 Winning Eleven / Play 2: **Electronic Things**
 XBOX / Halo 2 - Top Spin: **Electronic Things**

IMPORTANTES PREMIOS

CURSOS

Dictados por: Edu Hard Instituto Tecnológico Argentino.
 Convergencia Tecnológica. • Redes Wireless. • Seguridad informática para el hogar, PyMES y pequeñas redes.

CHARLAS:

Centro Metropolitano de Diseño, Entretenimiento Digital. Una oportunidad para la Ciudad.
 Mercado Libre: Cómo vender en Internet
 Gamesario: GameSario y la actualidad del desarrollo en el país.
 Gamesario: Cómo desarrollar tu propio juego.
 Korpos: Presentación del Juego The Elixir
 PsicoFxp: Las comunidades virtuales como medio de comunicación.

Diseño: JINIA NEGRERA 03752 435923

VENI DIVERTITE Y APRENDE EN EXPO VIDEO JUEGOS Y PC 2005

Del 1 al 4 de Septiembre de 2005 • El Dorrego • Zapiola y Dorrego
 Ana Juan Congresos • Tel: 54 11 4902 1017 • 4903 7072 • Pallisò diseño • Tel: 54 11 4582 2163
 Aranguren 551 P.B.B Capital Federal • Bs. As. • Argentina
 Para más información entra a: www.expovideojuegosypc.com.ar • info@expovideojuegosypc.com.ar