Buenas y malas prácticas de responsive design

El responsive design se ha consolidado como la respuesta al problema de la visualización de webs en dispositivos móviles. Sin embargo existe una diferencia entre implementar técnicamente este sistema y el diseñar de forma correcta.

Hay varias prácticas en responsive web design sobre las que deberíamos reflexionar. Este artículo se basa en mi estudio y opinión personal, lo que no indica que sean buenas o malas prácticas de diseño, pero si ofrece una perspectiva que pretende hacerte pensar antes de adaptar tus diseños.

La cabecera

Podemos considerar que reservar el mismo espacio vertical para los contenidos de la cabecera es un error de diseño o de usabilidad.

Por ejemplo Webdesignerwall usa una cabecera tan alta que cuando, por ejemplo, realizamos una búsqueda, la web se recarga mostrando la misma cabecera ocupando el 100% del alto del dispositivo.

La cuestión no es que el usuario no vaya a hacer scroll siguiendo sus intereses, sino que le obligamos a hacerlo cuando él busca una respuesta inmediata a una acción.

La navegación

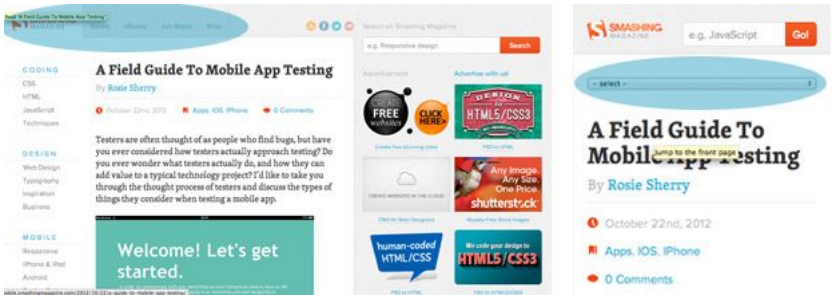
La mayoría de las webs usan un menú de navegación basado en listas. Si bien la disposición horizontal o vertical de estos menús resulta cómoda en un monitor, cuando la pantalla de nuestros dispositivos es de un tamaño pequeño (p. ej. un smartphone) usar este sistema creo que no es adecuado.

Por ejemplo en UXLondon 2011 el diseño se adaptaba mal, ocupando alto innecesario (punto anterior) y descuadrando el menú. Sólo un año después solventan el problema reescalando el tamaño de fuente del menú.

UXLondon de 2011

Existen varias opciones pero mis dos favoritas son:

- Conversión de listas (
) a un combobox (<select><option>). Un buen ejemplo es Smashing Magazine
- Reducir el menú a un icono y mostrarlo bajo demanda. Ejemplos claros son Bootstrap y CSSTricks



El menú de Smashing magazine se convierte en un select

Por ejemplo en 40horse.com toman otra solución elegante: bajan el menú al final de la página y dejan en cabecera sólo un enlace ancla apuntando al mismo.

Ocultar contenido

Por comodidad o por desconocimiento, hay quien oculta contenidos de la web en sus adaptaciones móviles usando *display:none*.

Aunque por motivos estéticos puedan obviarse elementos que aporten sólo atractivo visual, eliminar contenidos (texto, imagen, vídeo…) es negar al usuario la posibilidad acceder a la misma información que encontraría en la versión de sobremesa.

Además **ocultar elementos mediante CSS no evita la carga de los mismos**, por lo que a nivel técnico y de rendimiento tampoco hay excusas.

"Mobile first" no es "Mobile only"

Hacer *mobile first* significa hacer un diseño pensando **primero** en la versión adaptada a una pantalla smartphone e ir incrementando a los siguientes tamaños. Sin embargo en ocasiones esta conversión a monitores de mayor resolución se hace mal, dejando líneas de texto exageradamente largas o espacios enormes que afectan a la composición.

Retina display

Retina display ofrece resoluciones superiores a los dispositivos con pantallas tradiciones. Por eso debemos usar imágenes optimizadas para este tipo de pantallas. Una buena técnica es usar el formato JPG de alta resolución con una compresión del 40%. Así evitamos que as imágenes se pixelen demasiado.

Otra solución es usar Adaptive-Images, una solución que trabaja conjuntamente en el front y el servidor.

En resumen

El responsive design es una técnica relativamente joven, y aún no se ha madurado lo suficiente. Estos son algunos puntos que he estudiado y que veo mejorables en la mayoría de casos, pero no son todos ni tienen por que ser realmente malos para el usuario (yo mismo caigo en estos "errores").

Nos toca a nosotros como desarrolladores y diseñadores la búsqueda de una mejor experiencia en todos los dispositivos así que os animo a experimentar y estudiar, para que saquéis vuestras propias conclusiones.

Valora esta entrada

Rating: 9.6/**10** (5 votes cast)

Antonio Navajas

Antonio Navajas es desarrollador front-end, diseñador y formador ocasional. Apasionado por la tecnología, el diseño y en general por la cultura geek, colabora activamente en varios proyectos relacionados con la web, la formación y el diseño.

More Posts - Website

Follow Me:

Te puede interesar:

365signage 3: Starcraft (7)

365signage 16: Regreso al futuro (2)

Gif Animados con Photoshop (5)

Apple pierde un prototipo de iPhone otra vez: la verdadera historia (6)

Geolocalización HTML5 (1)

Tags: css3 , diseño , diseño web , media querie , responsive design
Publicado el: octubre 22nd, 2012 by Antonio Navajas Un comentario

 Wakkos dice: 31 octubre, 2012 a las 3:47 pm

Buena entrada.

Voy a dar mi opinón sobre ciertos puntos:

En primer lugar cuando alguien DISEÑA, no hace que una web SE VEA en diferentes resoluciones, la ADAPTA. Eso por supuesto implica que el header tiene que seguir siendo proporcional así como toda la jerarquía del sitio. (notad que no digo Responsive Design, en mi opinión DISEÑAR incluye ya responsive design, o no es diseñar).

Sabéis que? viva el background-size:content

Estoy en desacuerdo en lo del menú de listas. Una lista es un elemento que puedes estilizar al igual que un anchor, no te da ninguna ventaja una sobre otra. Puedo lograr el menú de css-tricks con listas igualmente.

De igual manera no usaría un dropdown box com el de Smashing: Cada dropdown box se muestra como le da la gana en cada dispositivo, perdiendo todo el branding tan importante para nuestra web. Me quedo con la opción de Bootstrap, aunque a ver si cambiamos ya las 3 líneas, que que empeño tiene la gente en estandarizar hasta el diseño.

Y estoy de acuerdo en que ocultar contenido no es la solución, pero no das las soluciones. Te echo un cable: En caso del texto y capas podemos definir su altura o tamaño a 0, también podemos usar un script como el de "condicional loading" que set´a en 24ways (buscad en Google). En caso de imágenes, mientras la W3C se pone de acuerdo, la mejor solución que he encontrado es PictureFill: https://github.com/scottjehl/picturefill

Funciona como pretende funcionar la W3C pero con JavaScript.

Del resto, ando vago pa continuar xD Pero muy buen aporte.

Responder

Nombre (required)

Email (no se publica) (required)

Página web

Submit Comment

Últimos tweets

@kinduff @Wakkos es lo q tienen las versiones beta :-)

@kinduff @Wakkos he eliminado el widget de suscripcion, que desde el teléfono no quiero hacer mas. Perdonad las molestias y gracias ;-)

@Wakkos en serio? Lo miro en un momento...

RT @kunfu_mkt: RT @kunfu_mkt: "Errar es de humanos. Herrar es de herreros, y E.rar es de geeks".

Seguir a @ajnavajas