



La mejor forma de Aprender Programación online y en español
www.campusmvp.es

FRIKADAS: Cómo funcionan los monitores ópticos de pulsaciones (pulsómetros) como el del Apple Watch |
JavaScript: Variables y funciones privadas

Qué es el Responsive Web Design y por qué debería importarme aunque sea programador

Por [José Manuel Alarcón](#). Publicado el 11 de mayo de 2015 a las 10:00

[**Certifica que sabes HTML5, CSS3 y JS:** Prepara el examen 70-480 con nosotros. ¡Haz clic aquí!](#)

¿Con qué dispositivo estás leyendo este artículo? Si lo piensas, dependiendo del momento del día y otros factores, seguramente podrías responder varias cosas diferentes: con mi *smartphone* táctil personal de 4.5 pulgadas, un *tablet* de 10" con resolución de 2048x1536, el *phablet* de 6" que me ha dado la empresa o mi monitor de 24" con resolución Full-HD y usando un teclado y un ratón.

Y es que hoy en día **la información y las aplicaciones se consumen en infinidad de formatos**, y una misma persona utiliza diversos dispositivos con capacidades de pantalla, factores de forma, tamaños y resoluciones diferentes, y usándolos de manera táctil o de la manera tradicional: con teclado y ratón.

Hasta hace menos de una década las cosas eran mucho más sencillas. Si queríamos crear una aplicación o un sitio web "universal" tan solo teníamos que preocuparnos de unas cuantas variaciones. Simplificando, lo que hacíamos **era detectar si los usuarios se estaban conectando usando un móvil o no** (¿alguien recuerda [WAP](#) ?) y luego se mostraba una versión diferente de la misma web o aplicación según el caso. Incluso existían subdominios (ej: m.miempresa.com) o dominios de primer nivel tan infames

como .mobi (que ni siquiera era fácil de escribir en un tecladito móvil de aquellos) que se usaban de manera especial para la versión móvil.

Este, aunque cueste creerlo ahora, era el [método recomendado de hacer las cosas](#). De hecho, por ejemplo, en ASP.NET existe [una clase especializada](#) solamente en eso, que se basaba en la detección del agente de usuario para averiguar cuál era el navegador y así poder indicar sus capacidades. Esto ya [no sirve de nada](#).

En la actualidad la variedad es tal que **es imposible seguir esta estrategia** para tener una **web o una aplicación que funcione en todos los dispositivos**. Tendríamos que crear cientos de versiones diferentes de la aplicación para que funcionase correctamente casi siempre, y aún así nos quedarían casos sin cubrir.

¿Cuál es entonces la solución?: pues **crear una única versión** de nuestra página o aplicación pero consiguiendo que se adapte a todos los posibles dispositivos de manera automática.



Esto, tan complicado a primera vista, se consigue utilizando lo que se conoce como **Responsive Web Design**.

Diseño Web Responsivo - RWD

El diseño web "responsive" o "responsivo" es un **conjunto de técnicas que permiten que un determinado HTML se adapte de manera automática y flexible al tamaño, relación de aspecto y resolución de la pantalla del dispositivo** con el que se está visualizando. Cualquiera que sea ésta. El usuario verá el contenido de forma correcta siempre y podrá utilizar la funcionalidad independientemente de estos factores y también del método de interacción que utilice (ratón o táctil).

Esto implica no solo la redistribución de los elementos, sino también el uso de **diferentes tamaños para todos los elementos** (letras, espaciados, botones...), la optimización y uso

de **diversas versiones de imágenes** apropiadas para cada caso, el **reflujo automático** del contenido, la **ocultación y conversión** de elementos, etc...

Para ello, el RWD se basa en diversos pilares fundamentales, entre los que cabe destacar:

- El concepto de cuadrícula fluida
- El uso de unidades de medida relativas
- Las imágenes flexibles
- Las limitaciones en tamaño máximo y mínimo de elementos
- Las media-queries
- Meta-configuración del viewport
- Tipografía adaptable
- Polyfills para soportar navegadores más antiguos o modernos pero menos capaces

Además, existen **varios conceptos relacionados** que se suelen utilizar en paralelo a las técnicas propias de RWD.

Tradicionalmente se han hecho los diseños para el escritorio, buscando luego que estos se adaptasen a móviles y otros tamaños de pantalla, y forzando de paso que toda la funcionalidad encajase en esa versión reducida. A esta orientación se le suele denominar "degradación elegante" (*graceful degradation* en inglés), aunque en no pocas ocasiones de "elegante" tenía poco y acababa con una versión poco usable, fea y saturada de la aplicación. Mala cosa.

En la actualidad se hace justo al revés. Los diseños se plantean en primera instancia para los dispositivos móviles, tratando de extraer su esencia, lo verdaderamente importante que debe funcionar. Posteriormente se construye desde esta versión para añadir mejoras o incluso funcionalidades en pantallas más grandes...

...lo cual nos lleva a la **mejora progresiva**. Ésta parte de la versión móvil y va añadiendo inteligentemente capas de mejoras al diseño base, orientadas a crear mejores experiencias para el usuario a medida que se va teniendo más espacio.

A esta forma de trabajar se le denomina **mobile first**.

Esta combinación de *mobile first* y mejora progresiva tiene la ventaja de que **facilita mucho más la usabilidad y la accesibilidad** de los desarrollos/diseños, y además como hemos trabajado primero en lo más complicado, el resto suele ser mucho más fácil. Una vez que has creado una versión decente para los dispositivos menos capaces te resultará más fácil ir añadiendo más funcionalidad para el resto de dispositivos más capaces.

5 razones por las que debería importarme

A parte de todo lo expuesto hasta ahora, existen algunas **razones de peso para apostar por el RWD** como manera general de trabajo y diseño tanto si creamos aplicaciones web como si hacemos diseño HTML:

1. **Es una apuesta de futuro:** aprendiendo RWD ahora vas a tener trabajo durante muchos años, pues es la técnica predominante desde hace tiempo y nada hace prever que esto vaya a cambiar. Es más, si tu foco es el desarrollo o el diseño web es casi indispensable para el mercado laboral que conozcas *Responsive Web Design*. Hoy en día aún es una forma de diferenciar tu currículu. En los próximos años que aparezca en él será algo indispensable.
2. **Independiente del dispositivo:** las aplicaciones creadas de la manera apropiada usando RWD van a funcionar bien en todos los dispositivos del mercado, y van a seguir haciéndolo en los que haya en el futuro. Así que no tienes que preocuparte de si se va a utilizar en un móvil básico, una Smart-TV o una consola de videojuegos: simplemente funcionará bien en todos. Además, puedes convertir fácilmente tu aplicación web responsive en una aplicación móvil gracias a plataformas gratuitas y Open Source como [PhoneGap/Apache Cordova](#).
3. **Rentable:** los sitios y aplicaciones creadas con estas técnicas dan más trabajo que un sitio "normal" en el que simplemente colocas HTML sin más, pero serán utilizables durante muchos años y el hecho de existir una única versión hace que las actualizaciones y el mantenimiento sean mucho más sencillos. Eso significa un gran ahorro de tiempo y dinero a largo plazo.
4. **Google lo recomienda:** además [de manera activa](#), y eso no es cualquier cosa. Las posiciona mejor y, es más, desde abril de 2015 los sitios web responsivos tienen preferencia para el buscador en las búsquedas realizadas desde dispositivos móviles. Casi nada.
5. **Más negocio:** usando el enfoque "mobile first" se consiguen aplicaciones y webs más fáciles de usar, más optimizadas y por lo tanto más rápidas. Esto hace que gusten más a los clientes. Y clientes contentos significa negocio.

Así que, si todavía te lo estabas pensando es hora de que dejes de dudar y **te lancés a aprender Responsive Web Design ya.**

Y si ya conoces la calidad de nuestra formación on-line, no dudes en hacer **el estupendo curso que hemos preparado :-)**

[Si eres Desarrollador Web, necesitas dominar JavaScript: ¡Aprende con nosotros! »](#)



Fundador de [campusMVP](#), es ingeniero industrial y especialista en consultoría de empresa. Ha escrito diversos libros, habiendo publicado hasta la fecha cientos de artículos sobre informática e ingeniería en publicaciones especializadas. Puedes seguirlo en Twitter en [@jm_alarcon](#) o leer sus blog [técnico](#) o [personal](#).

[Ver todos los posts de José Manuel Alarcón](#)



No te pierdas ningún post

Únete gratis a nuestro canal en [Telegram](#) y te avisaremos en el momento en el que publiquemos uno nuevo.

¿Te ha gustado este artículo? ¡Compártelo!



Archivado en: [Desarrollo Web](#)

[1 comentarios](#)

Publicaciones relacionadas

[¿Cuáles son las causas más comunes de despido de un programador?](#)

¿Por qué motivos se suele despedir a un programador o a un ingeniero de software? Existen miles de r...

[Los beneficios de utilizar Docker y contenedores a la hora de programar](#)

El desarrollo con instancias de Docker está cada vez más extendido entre los equipos de desarrollo d...

[La muerte de la locura de los microservicios en 2018](#)

En los dos últimos años, sobre todo, los microservicios se han convertido en un tema muy popular. La...



Comentarios (1)



Pablo Silvoni
12/06/2016 1:29:13

Estimado José Manuel,

Tu explicación respecto de las técnicas de Responsive Web Design me han servido de mucho ya que estoy encarando un proyecto que incluye apps para smartphones y esto me indica el camino a seguir. No soy del ámbito del web design, pero en breve tomaré un curso.

Un saludo cordial,

Pablo Silvoni

[Responder](#)

Huevo, perro, abeja o viernes, ¿cuál es un día de la semana?: