

Desarrollo Web

Diseño web responsive



Índice

Introducción	3
¿Qué es el diseño responsive?	4
¿En qué consiste el diseño adaptativo para el teléfono móvil y resto de dispositivos?	4
¿Cuáles son las ventajas de tener un sitio web responsive?	4
Media queries	6
Sintaxis de una media query. Cómo funciona	6
Cómo funciona una media query	7
Mobile first: cómo usar las media queries de forma correcta	7

Introducción

El diseño web responsive o diseño adaptativo es una de las mejores prácticas que existe para lograr una excelente experiencia de usuario. Se trata de readaptar el diseño de una web para que pueda verse desde cualquier teléfono móvil o dispositivo, es decir, todos y cada uno de los elementos de un sitio web se adaptan a todo tipo de tamaños de pantallas. Tablet, ordenadores, smartphones... cualquiera que sea el dispositivo que utilice el usuario, no será un problema si tu web tiene un responsive design.

Para ser conscientes de la importancia del acceso desde móviles a Internet, vamos a recordar ciertas estadísticas clave. Un 79% de los españoles ha comprado alguna vez a través de su smartphone. Entre los menores de 35 años, el porcentaje aumenta hasta el 87%, según datos del Estudio Anual de Mobile & Connected Devices 2019 de IAB Spain. El mismo estudio afirma que 2 de cada 3 minutos que navegamos por internet lo hacemos utilizando dispositivos móviles y que el smartphone se utiliza más entre semana y en horario no laboral.

¿Qué es el diseño responsive?

Todos hemos oído hablar más de una vez de este concepto y sin embargo quizás no sepamos en concreto qué significa y cuál es su importancia dentro del diseño web y del branding corporativo. El diseño web responsive, también llamado adaptativo, es un método de diseño web que tiene como objetivo principal lograr una **correcta visualización de una misma página** en varios tipos de dispositivos, ya sean ordenadores de mesa, portátiles, tabletas, teléfonos móviles o cualquier otro dispositivo inteligente.

A día de hoy los usuarios acceden a los contenidos de las páginas webs desde diversos dispositivos, por lo que es necesario que una misma página pueda verse correctamente desde cada uno de estos terminales y según el tamaño de la pantalla del mismo.

En líneas anteriores, ya vimos varios datos estadísticos que reflejan la importancia de tener un **sitio web responsive**. Aplicándolo y ajustando nuestra site en función del dispositivo, el tráfico web no se verá afectado y la tasa de rebote será mínima.

¿En qué consiste el diseño adaptativo para el teléfono móvil y resto de dispositivos?

La necesidad del diseño responsive o adaptativo surge del hecho de que una web, la nuestra, por ejemplo, se adapte a los diferentes tamaños de pantalla de los diferentes dispositivos que hay, como móviles, tabletas, portátiles u ordenadores de escritorio.

Cada dispositivo y cada terminal tiene un tamaño de pantalla distinto, por lo que el diseño responsive debe encargarse de **redimensionar y ubicar todos los elementos de una página web** de forma que su visualización sea la adecuada, garantizando, por lo tanto, una **mejor usabilidad al usuario**. Los contenidos ([layouts](#)), las imágenes, las tipografías, los vídeos, los pluggins y la usabilidad en general se vuelven fluidos gracias al código [media-queries](#).

El diseño web responsive no debe confundirse con **webs adaptadas para usuarios móviles**.

Cuando hablamos del diseño responsive hablamos de un único sitio web que puede verse adecuadamente desde dispositivos de todo tipo. Cuando mencionamos de **webs adaptadas para usuarios móviles** nos referimos a un sitio móvil en el que se diseña desde cero una web independiente, en este caso el contenido y la imagen están diseñados especialmente para figurar en teléfonos móviles.

¿Cuáles son las ventajas de tener un sitio web responsive?

La primera y más importante, la que hemos mencionado en el punto anterior, el usuario logra una mejor y más optimizada experiencia lo que hace eco en beneficios para la marca. Según los datos de **Google Think Insights**, un usuario con una experiencia positiva en un sitio web se traduce en un 67% más de posibilidades de que repita visita. Pero aún hay más ventajas a tener en cuenta:

Es importante para el SEO. Así nos lo hace saber desde el 2015 el algoritmo de Google. El posicionamiento del principal buscador premia una experiencia de uso cómoda y fluida como es la que se logra con el diseño responsive.

Nuevamente, se obtienen más conversiones y leads.

- **Mejora el branding corporativo.** La buena usabilidad de un sitio web crea una buena impresión y experiencia para los usuarios. ¿Qué pensarías si al entrar desde tu smartphone a una página web todo apareciera descolocado? Si los elementos se cortan o no se ven bien, seguramente ese esfuerzo extra de intentar leer de forma incómoda te haga abandonar la página rápidamente, ¿no? Eso exactamente es lo que harán los usuarios de tu web si no es responsive.
- **Se reduce en gran medida el tiempo de desarrollo de la web.** Menos recursos invertidos y menos costes que crear una página web específica para ordenador y una aplicación o una versión concreta para móvil.
- Proporciona una **mayor viralidad de los contenidos y las imágenes.** Los usuarios pueden compartir contenido e imágenes de forma sencilla y rápida, más naturalmente.
- **El tiempo de carga se reduce exponencialmente.** La información se vas a instante desde el teléfono móvil o desde el dispositivo que el usuario esté utilizando.
- **Se evitan los contenidos duplicados.** Algo que además penalizan la mayoría de navegadores, como Google. Siendo negativo para el posicionamiento orgánico de la página web si no se aplica el diseño responsive.

Y en el caso de que ya tengamos un diseño responsive en nuestra web deberemos vigilar ciertos puntos clave que a veces se traducen en fallos o errores si no se ha aplicado bien el diseño responsive.

A continuación, enumeraremos algunos detalles que debes tener en cuenta para que todo funcione a la perfección:

- **Medición.** Analizar tus resultados web es fundamental para poder mejorar tus resultados y tomar las medidas necesarias en su caso. Esto siempre es importante por las razones que acabamos de comentar, pero además **nos ayuda a detectar posibles errores.** Gracias a la analítica web podemos descubrir una alta tasa de rebote en una página concreta y ver a qué se debe. Puede que el diseño no esté funcionando o que algún modulo no se visualice como debe.
- **Efectos.** Existen multitud de efectos que se pueden aplicar a una página web: animaciones al posar el ratón en cierto elemento, módulos dinámicos... Pero a veces los que funcionan a la perfección en un site, no lo hacen en formato mobile. Por eso siempre deben comprobarse las animaciones en dispositivos móviles, para adaptarlas o modificarlas si es necesario.
- **Diseño.** Al diseñar se debe tener en cuenta el diseño responsive para que todo se visualice como debe, tanto en el diseño web en general como en el diseño concreto de las imágenes y vídeos que se vayan a publicar.

Media queries

Las **media queries** son una sintaxis especial para CSS que nos permite definir unos estilos que solo se aplicarán en el caso de que se cumplan unas condiciones concretas. Podemos asimilarlos a unas líneas de código opcional, que solo se mostrarán para algunos usuarios o dispositivos.

MEDIA QUERIES



En un momento donde el tráfico móvil se multiplica cada año, ya no servían las webs estáticas diseñadas para mostrarse en una resolución concreta. Era necesario adaptar los diseños a unas dimensiones concretas para que el usuario recibiera la información web en su dispositivo sin tener de hacer zoom para leer los textos.

En este momento surgen dos modos de trabajar. La primera abogaba por tener dos versiones de la misma web, una para escritorio y otra para móvil. En la mayoría de los casos no están conectadas entre sí. Esto era costoso de mantener (dos webs) y además no siempre se mostraba el mismo contenido y diseño a ambos usuarios.

La segunda forma de trabajar es usar un mismo diseño para ambas resoluciones. Un diseño que se pudiera adaptar a unas determinadas resoluciones o ser flexible en su totalidad, independientemente del contenido y diseño que tuvieran. A esta nueva forma de trabajar se le denominó **diseño web responsive**.

Y como el encargado del diseño en la web es el CSS, surgió un nuevo elemento capaz de incorporar esta flexibilidad a la hoja de estilos: las **media queries**, que pueden abarcar porciones de código que solo se aplicarán según qué condiciones: si la resolución de pantalla es pequeña, si es muy grande, si el dispositivo está en posición horizontal...

Sintaxis de una media query. Cómo funciona

```
@media not|only mediatype
and|not|only
(media feature) {
  .my-code {...}
}
```

Las **media queries** actúan como contenedor de las reglas y selectores a aplicar, por lo tanto, todo lo que engloba empieza con la apertura de una llave y termina con el cierre de la misma. En cuanto a la sintaxis, una **media query** siempre comienza con el indicador de **@media**, seguido de un **mediatype**.

```
@media not|only mediatype
and|not|only (media feature) {
  .my-code { ... }
}
```

Este [mediatype](#) es el encargado de seleccionar que tipo de formato será objeto de estas reglas. Aquí podemos decidir entre todos, impreso, pantalla o speech, útil para los equipos adaptados y accesibles.

Con los operadores [not](#) y [only](#) podemos jugar para agrupar o excluir un tipo de [mediatype](#) concreto.

```
@media not|only mediatype
all
print
screen
speech
braile / protection / tv (deprecado)
```

El segundo parámetro de una [media query](#) en CSS es el [media feature](#), donde indicamos al navegador qué condición debe cumplir el dispositivo de salida especificado anteriormente para que la condición sea verdadera y se aplique este código contenido en nuestra [media query](#).

Los [media features](#) más comunes son los referidos a las dimensiones de pantalla del dispositivo, pudiendo establecer el alto y ancho en el que se aplicarán (con [height](#) y [width](#)), o lo que es más interesante, a partir de que ancho o alto se aplicarán (con [min/max-width](#) y [min/max-height](#))

```
@media and|not|only (media feature)
height
width
max-height / min-height
max-width / min-width
```

Aunque también las hay más complejas, como por ejemplo aquellas combinadas que la orientación del dispositivo ([landscape](#) o [portrait](#)) o incluso las que definen el ratio de píxeles de la pantalla, como [device-pixel-ratio](#), muy útil para definir estilos y CSS para las pantallas retinas, por ejemplo algunos iPads o iPhones.

Cómo funciona una media query

El funcionamiento en el navegador es muy sencillo. Simplemente aplicará el código que esté incluido en una [media query](#) sobrescribiendo el heredado. Con esto es como si estableciésemos una capa sobre otra, con cierta transparencia, ya que no eliminamos del todo el código general de la hoja de estilos CSS.

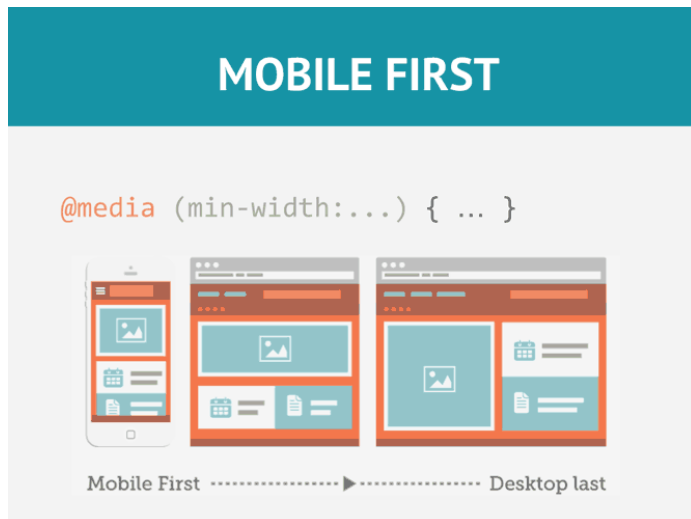
Un ejemplo, queremos que se muestre un código concreto solo en pantallas y con una resolución menor a 400px. El código resultante sería:

```
@media screen and (max-width: 400px) {
  h1{font-size: small;}
}
```

En este ejemplo hemos definido un tamaño de texto pequeño para todos los dispositivos con una resolución de hasta 400px de ancho. Para el resto de dispositivos, esta orden no tendrá efecto.

Mobile first: cómo usar las media queries de forma correcta

Una de las dudas principales que se plantean a la hora de usar las [media queries](#) es si definir el estilo general del CSS para escritorio y reservar los condicionales para los móviles o hacerlo a la inversa. Estamos hablando de las corrientes [Desktop First](#) (escritorio primero) y [Mobile First](#) (móvil primero)



Se ha discutido largo y tendido sobre esto, y poca gente duda ya que lo más interesante es trabajar con [mobile first](#), es decir, tener unos estilos para móvil como base (pequeñas resoluciones) e ir creciendo poco a poco añadiendo [media queries](#) para dispositivos más grandes gracias al [mediatype min-width](#).

Con esto conseguimos crear una versión muy liviana para los usuarios móviles (donde el navegador solo tiene que aplicar las primeras líneas de CSS) mientras que la carga de leer todo el conjunto sobrescribiendo las líneas marcadas en las [media queries](#) queda para los dispositivos de mayor resolución, que se entienden van a tener una conexión a internet más rápida y mayor velocidad de procesamiento de dichos condicionales.

Además, es menos probable tener problemas con elementos de nuestra página que se descuadran. Si diseñamos primero para navegadores de escritorio, fácilmente nos podemos encontrar con que, al pasar el diseño a dispositivos móviles, los elementos no conservan su disposición, se mezclan, se desordenan, etc.

Si diseñamos primero para dispositivos móviles, al pasar el diseño a dispositivos de escritorio estos problemas no suceden. Normalmente solo hay que reescalar los elementos.