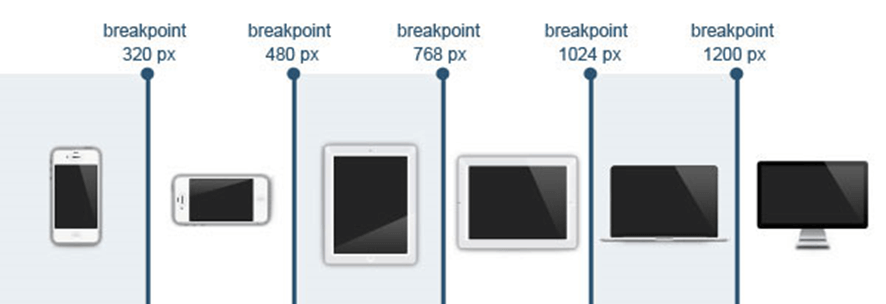
**Media Queries**

Las**Media Queries**son las **herramientas fundamentales que se encargan de aplicar diferentes estilos para diferentes dispositivos**, y proporcionan la mejor experiencia para cada tipo de usuario que se encuentra navegando en tu sitio web. Nacen de la necesidad de **crear *breakpoints* o puntos de ruptura** en la hoja de estilos CSS que tengas predefinida. Permite que tu sitio Web sea manejable desde diferentes dispositivos.



Si no te ha quedado muy claro, las **Media Queries** son un**módulo de CSS que sirve para detectar el tipo de dispositivo por el que se está navegando;** de esa manera el contenido consigue adaptarse al dispositivo concreto a través de las distintas condiciones que tú mismo asignas, como pueden ser ancho y alto de la ventana del navegador, ancho y alto del dispositivo, la resolución del dispositivo o la orientación de la pantalla. Son declaraciones lógicas que actúan dependiendo de las condiciones específicas que tú mismo declaras en la hoja de estilos. Si la premisa se cumple, se aplicaran los estilos definidos; si no, los omitirá por completo.

Hay dos formas de implementarlas:

La **primera opción** para poner en funcionamiento las **Media Queries** es a través del atributo *media* de la etiqueta *<link>*. Como sabemos, esta etiqueta es la que se usa para enlazar una hoja de estilo con un documento HTML. En ese enlace podemos **especificar condiciones que deben cumplirse** para que los estilos enlazados se apliquen. Debería ir dentro del*<head>* de nuestro HTML.

Recuerda que la etiqueta *<link>* tiene esta forma:

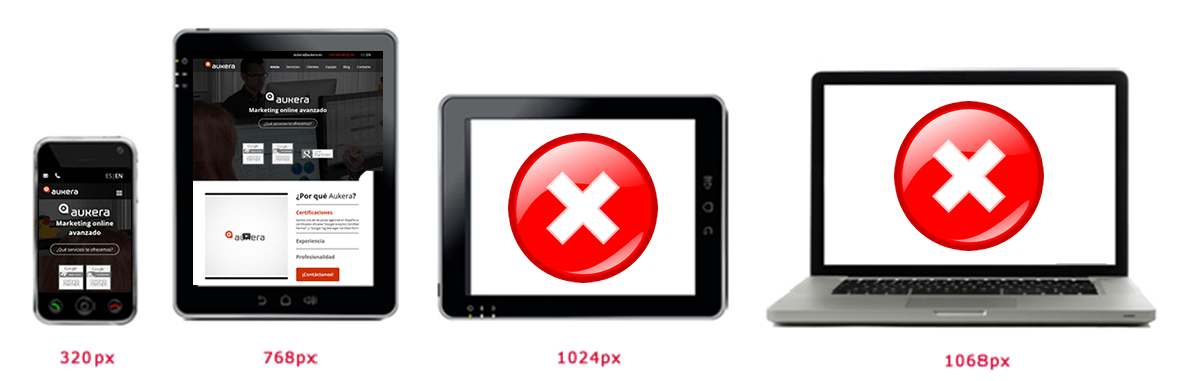
<**link** rel="stylesheet" href="estilos.css">

Pues ahora simplemente **agregamos el atributo *media* indicando la condición** que se debe cumplir para que estos estilos se apliquen:

<**link** rel="stylesheet" media="only screen and (max-width: 768px)" href="estilos.css">

Lo que concretamente l**e estamos indicando es que cargue la hoja de estilos “estilos.css”  si se cumple que el dispositivo de salida es una pantalla**, no una impresora u otro dispositivo (only screen), **y si la anchura de la ventana del navegador tiene de máximo 768 píxeles** (max-width: 768px).

Si se cumplen las condiciones, los estilos se mostraran correctamente,**en caso contrario, los estilos se omiten por completo**, y el contenido se muestra sin estilos definidos:



Cargar de esta manera las Media Queries **supone un problema**, y es que cada vez que queramos cargar diferentes estilos dependiendo de ciertas condiciones que queramos aplicar para distintos dispositivos, **tendríamos que cargar una hoja de estilos nueva**. Esto conlleva una carga más lenta de tu sitio web, ya que se hacen solicitudes HTTP adicionales, que se podrían evitar.

Hay **otro sistema más recomendable para aplicar las** **Media Queries:**basta con**incluir todas las condiciones necesarias dentro de un único archivo CSS.** Así, **incorporaríamos la construcción @media** seguido de las condiciones que queremos definir para cada tipo de dispositivo y donde se apliquen entre llaves { } los estilos concretos para cada uno de ellos. Es la manera más aconsejable, ya que **la carga es de un único archivo CSS**.

La forma de incluir Media Queries dentro de la hoja de estilos CSS es la siguiente:

@**media** (max-width:320px){

  <!—**-** **Aqu**í **van** **todos** **los** **estilos** **CSS** **--**>

}

Esta Media Query **se ejecutará sólo cuando la anchura de la ventana del navegador sea menor de 320 píxeles**.

@**media** (min-width:768px){

  <!—**-** **Aqu**í **van** **todos** **los** **estilos** **CSS** **--**>

}

Esta Media Query **se ejecutará sólo cuando la anchura de la ventana del navegador sea mayor de 768 píxeles.**

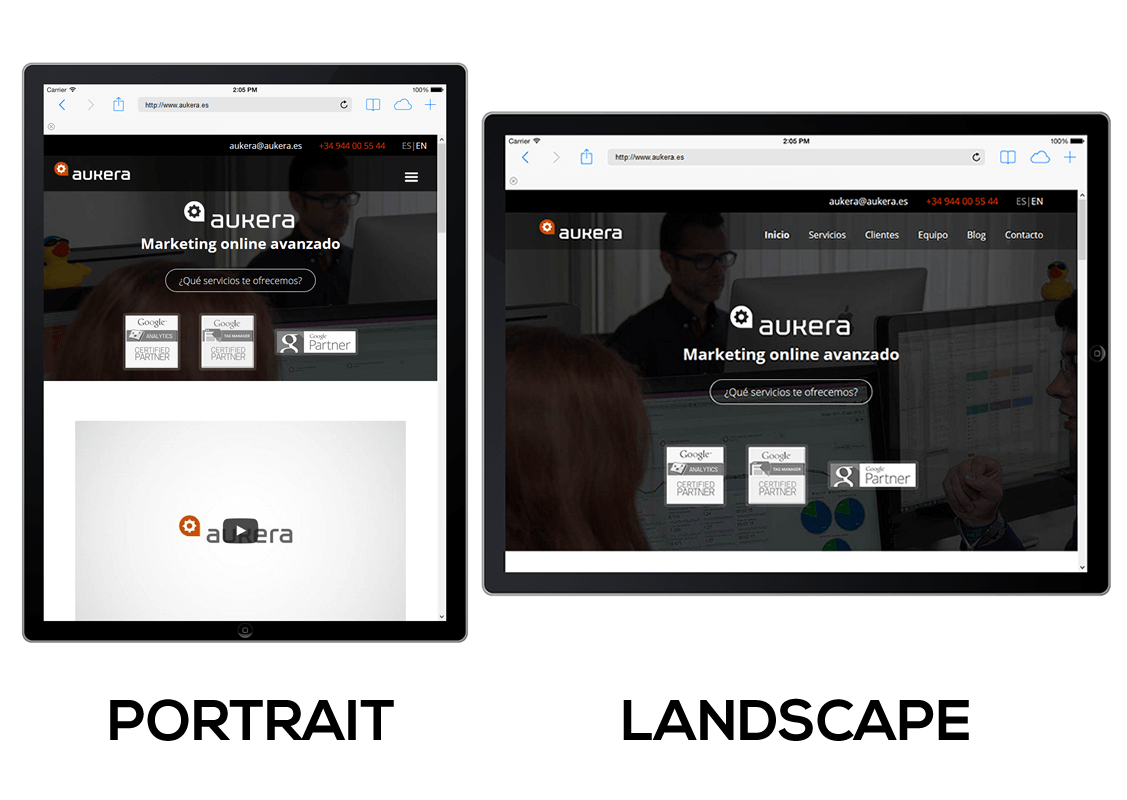
Además de las características para determinar las resoluciones y anchos de pantalla, **podemos determinar otros parámetros**, como por ejemplo la **orientación del dispositivo**, importante en dispositivos móviles:

@media (orientation: landscape) {

<!-- Aquí van todos los estilos CSS -->

}

* **Portrait:** orientación vertical
* **Landscape:** orientación horizontal



**Operadores lógicos para las Media Queries**

También **se pueden combinar más de una condición en la misma Media Query** para concretar todavía más un rango de resolución, mediante los operadores lógicos:

* Operador**and:** las dos condiciones deben cumplirse para que se apliquen los estilos.
* Operador**not:** es una negación de una condición. Cuando esta condición no se cumpla, se aplicarán las media queries definidas.
* Operador**only:** se aplican las reglas solo en el caso de que se cumpla.
* Operador**or:** se pueden poner varias condiciones separadas por comas y en el momento que se cumpla cualquiera de ellas, se aplicarán los estilos.

@media only screen **and** (**min**-width:320px) **and** (**max**-width:480px){

<!—- Aquí van todos los estilos CSS -->

}

Para esta Media Querie **se mostrarán los estilos CSS cuando la anchura de la ventana del navegador sea entre 320 pixeles y 480 pixeles,** ambos incluidos.

Estos son algunos de los **parámetros generales que se pueden emplear a la hora de construir las condiciones** en las Media Queries:

* **width**: anchura de la ventana del navegador.
* **height**: altura de la ventana del navegador.
* **device-width**: anchura de la resolución de pantalla.
* **device-height**: altura de la resolución de pantalla.
* **orientation** (portrait/landscape): dispositivo en horizontal o en vertical.
* **resolution**: densidad de píxeles.

Excepto la orientación, **el resto de parámetros admiten los valores “max” y “min”.**

* **max**-width: La anchura será **menor** que la indicada.
* **min**-width: La anchura será **mayor** que la indicada.

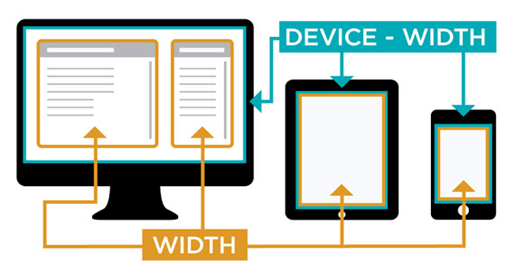
Una cosa muy importante que debemos tener en cuenta a la hora de utilizar las**Media Queries**, es **diferenciar entre el ancho de ventana del navegador y la resolución de la pantalla del dispositivo**, es decir:

@**media** only screen and (min-device-width: 960px){

  /\* Aquí van todos los estilos CSS \*/

}

En esta Media Querie que hemos definido, el atributo ***min-device-width* se refiere a la resolución de la pantalla del dispositivo** a la hora de cargar la hoja de estilos definida.



Esto quiere decir que **si reducimos el ancho del navegador, seguirá mostrándose de la misma manera,** ya que la resolución de la pantalla seguirá siendo la misma y no se adaptara al nuevo ancho de la ventana del navegador (es decir, si la pantalla de nuestro móvil tiene 450 px y el navegador detecta que lo óptimo sería visualizarla con 600 px así lo hará si no usamos la  meta-etiqueta **Viewport**).

En caso de usar los atributos para la**resolución de la pantalla, la etiqueta Viewport es necesaria.**

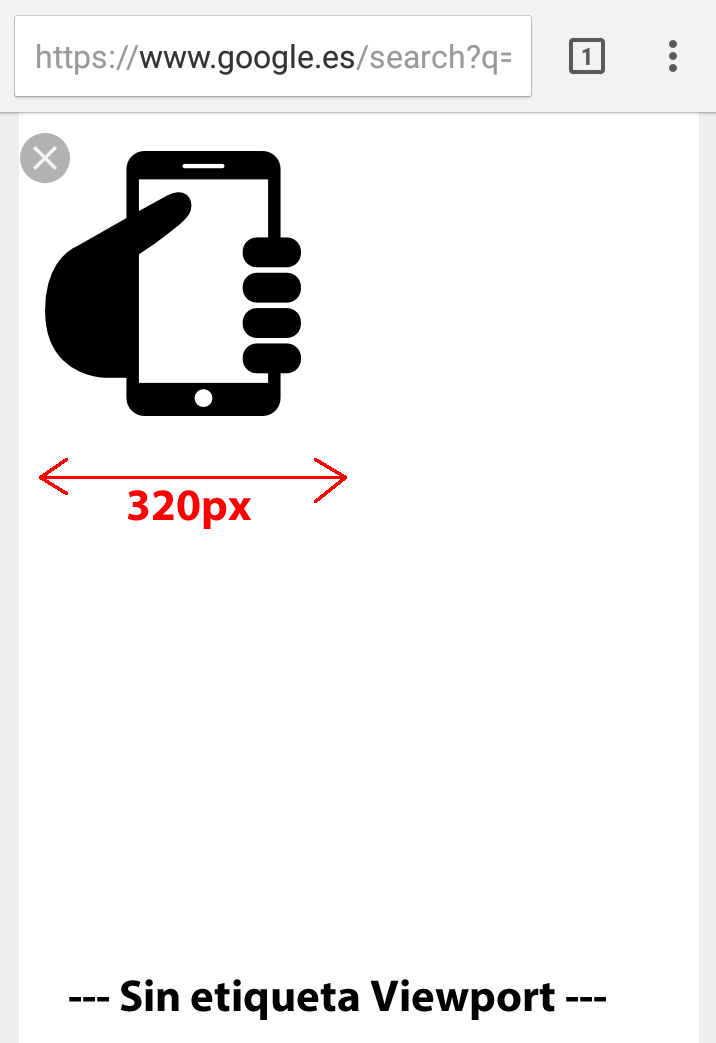
**¿Qué hace la meta-etiqueta Viewport?**

El **Viewport es el área visual donde se plasma el contenido del documento HTML** de tu sitio web. Se podría traducir  como vista o ventana y nos sirve para definir **qué área de pantalla está disponible al renderizar un documento, la escala/zoom que debe mostrar inicialmente**. Todo ello, con parámetros que le damos a la propia etiqueta *meta*, separados por comas en caso de utilizar más de uno:

|  |  |  |
| --- | --- | --- |
| ATRIBUTO | VALORES | DESCRIPCIÓN |
| *width* | Valor integral (en pixels) o constante device-width | Define el ancho de la página. |
| *height* | Valor integral (en pixels) o constante device-height | Define el alto de la página. |
| *Initial-scale* | Cualquier número real de 0.1 en adelante.  1 representa no escalable  (1.0 para no tener zoom o 2.5 para tener un zoom del *2,5* de aumento, por ejemplo). | La escala / zoom inicial del sitio Web. |
| *User-scale* | “yes” / “no” | Define los permisos para que el usuario pueda o no hacer zoom. |
| *Minimun-scale* | Cualquier número real de 0.1 en adelante. 1 representa no escalable | Define la escala / zoom mínimo que podemos hacer en la página. |
| *Maximun-scale* | Cualquier número real de 0.1 en adelante. 1 representa no escalable | Define la escala / zoom máxima que podemos hacer en la página. |

Prácticamente**todos los navegadores de smartphones al entrar a un sitio analizan el tamaño total y lo escalan para que se muestre completo en la pantalla**, este procedimiento genera muchas veces resultados inapropiados.

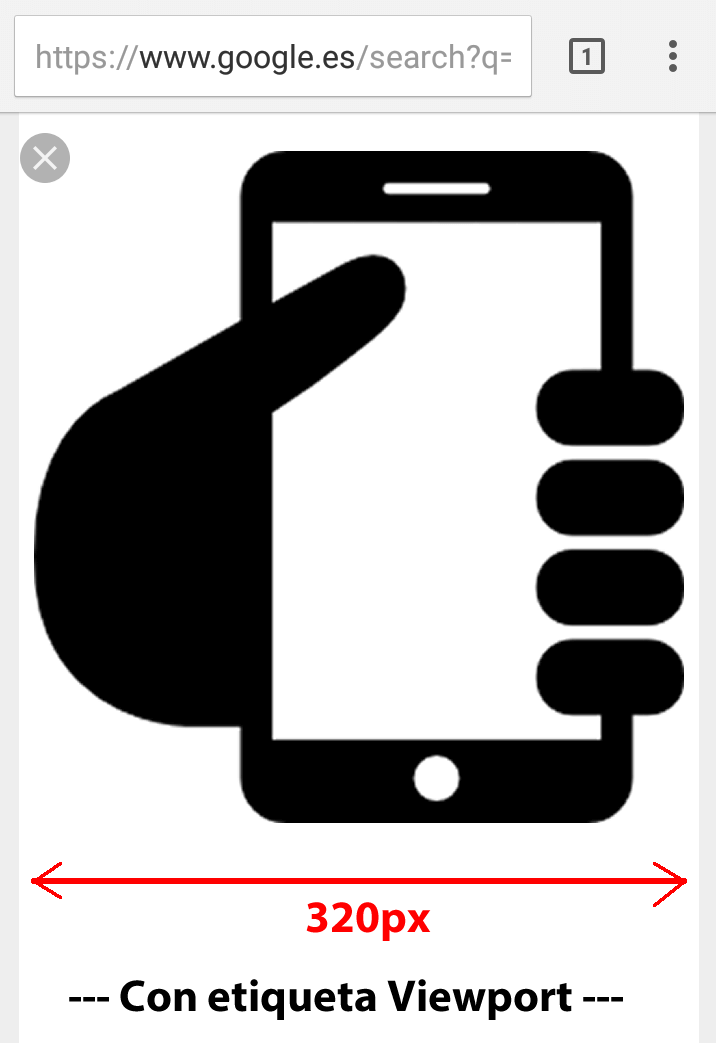
Por ejemplo, esta imagen mide 320 píxeles al igual que la pantalla del dispositivo, ahora bien, **la imagen aparece con un tamaño inferior a causa del efecto de la  escala automática**.



**La escala automática se puede evitar y controlar muy fácil con el uso de este atributo** **Viewport**: es un atributo del tag *<meta>* que  debe incluirse entre las etiquetas *<head>* del documento HTML de tu sitio web:

<**meta** name="**viewport**" content="width=device-width"/>

Con solo agregar estas líneas de código, **la imagen se adaptará al dispositivo**:



Es posible definir un tamaño específico para el  área visible del documento;**muchos sitios web ajustan directamente el Viewport a 320 px** para ajustar la apariencia al display vertical de un smartphone, usando un código similar a este:

<**meta** name="viewport" content="width=320"/>

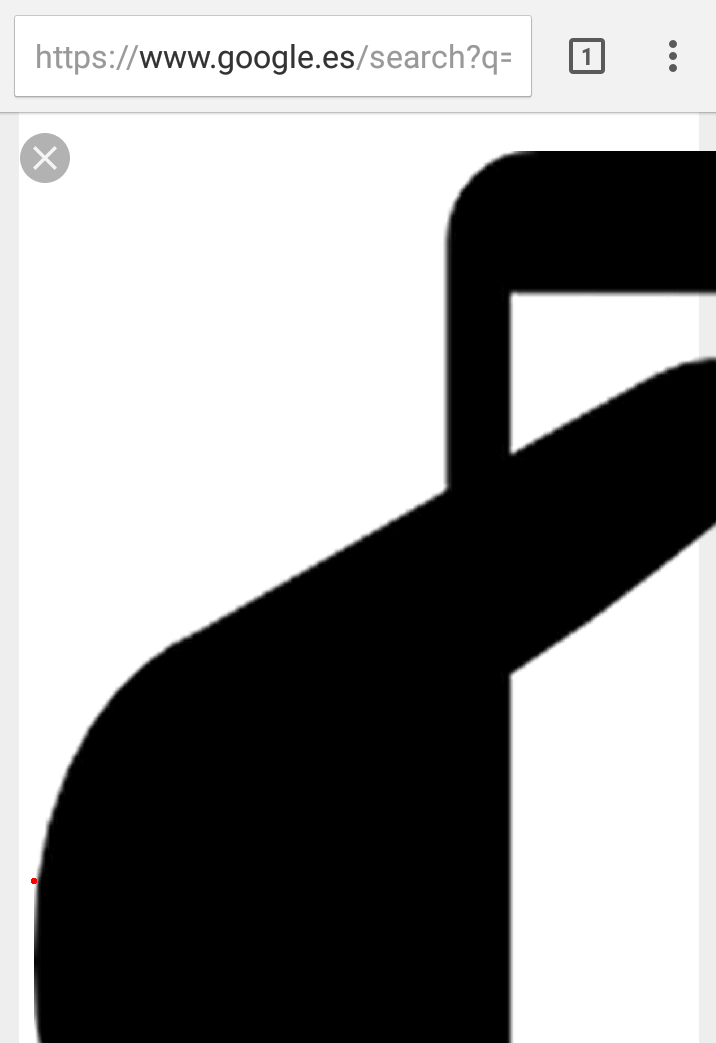
Pero, con los diferentes  equipos, dispositivos  y tamaños de pantalla, definir un tamaño específico puede ser una mala práctica que puede mostrar resultados erróneos en algunos equipos o cuando el dispositivo cambia de posición. Afortunadamente**podemos configurar el viewport para ajustarse dinámicamente al tamaño de cada dispositivo** usando el atributo “*device-width*”, que es**equivalente al 100% del ancho de la pantalla del  dispositivo**, independiente de su tamaño, posición o resolución:

<**meta** name="viewport" content="width=device-width"/>

**El alto de la pantalla  también es configurable** con las mismas propiedades a través del atributo “*height*” , aunque –salvo condiciones muy específicas– no es necesario definirlo. Esta propiedad se asignará automáticamente a través del scroll.

También podemos**controlar la escala de la vista con el atributo “*initial-scale*“**. El sitio se mostrará al doble de su tamaño original:

<**meta** name=”viewport” content=”width=device-width; **initial-scale=2**“/>



Es posible además, **limitar el tamaño al que se puede escalar el sitio** con el atributo “*maximum-scale*” . El siguiente ejemplo muestra el documento en escala correcta y permite ampliar (zoom)  hasta al doble de su tamaño.

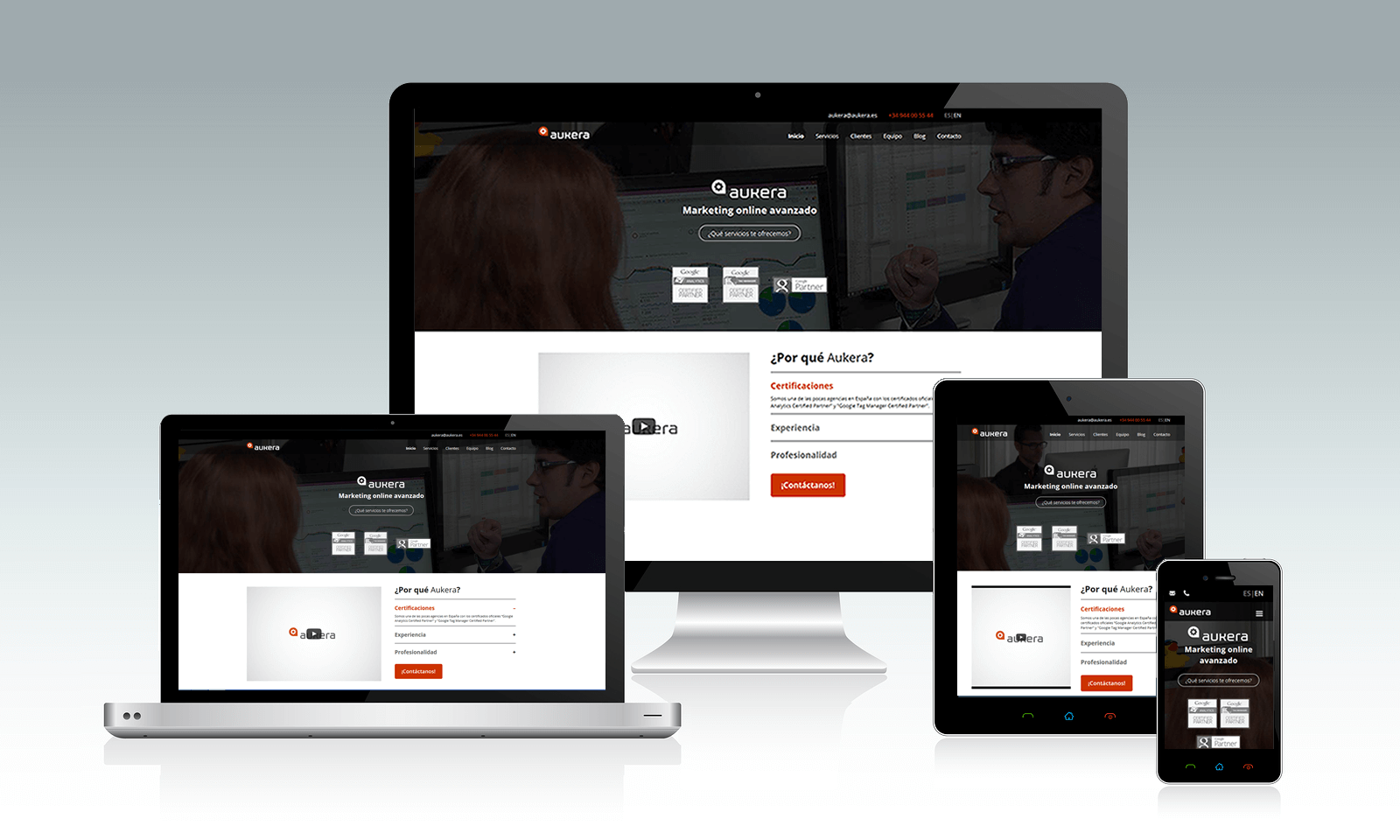
<**meta** name="viewport" content="width=device-width, **maximum-scale=2**"/>

Por ultimo, está el atributo  “*user-scalable*”, que **controla los permisos de reducir/ampliar el documento.** Con el siguiente código, el sitio se muestra en su escala original y no es posible cambiar el  tamaño desde el dispositivo móvil (importante mencionar que no se recomienda deshabilitar la opción de escalar el contenido).

<**meta** name="viewport" content="width=device-width, **user-scalable=no**"/>

En general, el atributo viewport permite muchas configuraciones, pero **para asegurar compatibilidad con la mayor cantidad de pantallas y navegadores móviles, se recomienda utilizar este formato como base**:

<**meta** name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">



**Para conseguir que nuestro sitio web se adapte a los diferentes anchos de pantalla, estos parámetros serán muy útiles:**

* Lo primero, y lo más importante es **dejar de usar píxeles y usar porcentajes** a la hora de tomar medidas (por ejemplo: width: 60%).
* **Que el ancho de la página sea 100%** no significa que queramos que la pantalla este en una alta resolución, sino que, si queremos limitar el ancho/alto junto al máximo/mínimo del contenido, **debemos usar los diferentes parámetros apropiados para ello** (max-width o si quisiésemos establecer un alto máximo max-height; para establecer el mínimo sería min-width y min-height)
* **Las posiciones absolutas o fijas no son recomendables** usarlas para posicionar contenido (menos cuando hagan falta). Lo mejor es utilizar el atributo float (float:left/right), es una técnica muy usada.
* Hay que hacer que **las imágenes y vídeos no sobresalgan de la estructura;** si no, aparecerá un scroll lateral en los dispositivos móviles que descolocará totalmente el diseño.

En resumen, **¿cuál es la mejor opción para tu sitio web? La experiencia del usuario siempre será lo primero.**

A los usuarios no les importa que versión utilices ni como estés optimizando tu sitio web; su objetivo es **poder encontrar lo que buscan de manera más eficiente y rápida**. Por tanto, **si tu web está más centrada en el contenido**, es mejor una **Responsive Web Design.** Pero **si necesitas que el usuario interaccione mucho con la web**, es mejor una **Mobile First**, ya que cada vez se consume más información desde los dispositivos móviles.

La conclusión es sencilla, los clientes están yendo más rápido que las propias empresas y estas deben adaptarse a ellos y a sus nuevas costumbres de consumo online a través de dispositivos. Esto es una solución para **reducir la tasa de rebote, haciendo que el usuario pase más tiempo en la página por su facilidad, comodidad, y óptima visualización y lectura de los contenidos.**

A día de hoy, Google valora todas aquellas **páginas web que se adaptan perfectamente a cualquier dispositivo,**ya sea PC, smartphone, tablet…  Por ello, es necesario que optimices tu sitio web de modo que cualquier usuario pueda visualizar la página sin importar el medio por el cual acceda.