***Fuente: https://desarrolloweb.com/manuales/responsive-web-design.html***

***Responsive Design***

Consejos al momento de armar un diseño responsive:

1. No se tiene que ver igual en todos los dispositivos, lo importante es que se vea claro y que se pueda consumir el contenido
2. Usar preferentemente CSS en lugar de Javascript
3. No usar tablas, sino contenedores DIV
4. Usar unidades relativas como: %, em, rem.

Es importante que la web tenga un buen rendimiento, es decir que este optimizada desde el momento uno para que no sea muy pesada.

\*Estudios muestran que si la web tarda en cargar (3 seg) el 57% de los usuarios salen

\*el algoritmo de google toma la velocidad de carga como factor importante (SEO)

Las técnicas responsive se focalizan en diseñar para:

* Todos los navegadores y sistemas
* Todas las resoluciones y tamaños de pantalla
* Todas las velocidades de conexión

*Procedimientos habituales para desarrollo:*

Es importante que se muestre el contenido principal y dejar de lado el contenido superfluo o innecesario

Es importante racionalizar el contenido que va a tener la web

Suele haber a veces problemas de compatibilidad con CSS y los navegadores, sobre todo con los mas viejos que no soportan algunas funciones. Para estoy hay dos técnicas

Progresive Enchancement:

Es una técnica que se usa aplicando CSS mas antiguo o básico, para que se pueda ejectuar en los navegadores mas viejos, luego de esto, le vamos añadiendo capaz de mas compliejidad para los navegadores mas nuevos.

Lo importante es que de base nos queda un diseño CSS que va a ser bien interpretado por los navegadores mas viejos

## Graceful degradation:

## Se hace al revés, es decir se aplica estilos CSS para los navegadores mas actuales, y luego lo ejecuto en los navegadores mas viejos. Luego hago los cambios necesarios para que los navegadores mas viejos repliquen el contenido de forma decente

Conclusion: Los profesionales se inclinan mas por la opción Progresive Enchancement

## Mobile First:

## Esta filosofía se basa en crear primero la app para verse de manera adecuada en celulares ( 320 px aprox). Luego ir agrandando la pantalla hasta que veamos que nuestro diseño comienza a verse feo

## Una vez identificado el ancho en el cual se empieza ver feo, establecemos en el mismo un “Breakpoint”, y usamos el media querys para modificar el aspecto y adaptarlo a la nueva resolución

## La forma de aplicarlo ( mas recomendada) es usar la palabra reservada “@media” en el archivo CSS que utilicemos

## Se pueden usar operadores lógicos para especificarles distintas condiciones (AND, OR, NOT, ONLY)

@media (max-width: 600px) and (orientation: landscape) {

h1{

color: red;

}

}

## Es importante también que agregues al head del html la siguiente línea:

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

## Es importante tener en cuenta que cuando en CSS aplico estilo a algún elemento, y luego mas adelante aplico otro estilo al mismo elemento, el estilo que va a terminar impactando es el ultimo que declare

## Consideraciones a tener:

## Hay que declarar primero aquellas variables globales que quiero que se apliquen a toda la aplicación y a todo dispositivo. Por ejemplo fuentes, padding, margin, etc

## Luego ire colocando los estilos globales que quiero para cada ancho de pantalla ( en caso de que lo crea necesario)

## Como delimitar los breakpoints de mi diseño?

## Basicamente lo que conviene hacer es armar la pagina mobile first, y luego ir agrandando la ventana hasta que veamos que nuestro diseño no se ve bien, en esos puntos donde notamos eso, es donde vamos a ir marcando los breakpoints

## No ver los breakpoints como puntos para el cambio a mobiles, tables, escritorio, etc. Sino que ir midiendoloes en base al diseño mio y luego no escatimar en la cantidad de breakpoints que haya que usar.

## Mismo estos pueden ser para cambios chicos, como en imágenes o márgenes o lo que sea, cada vez que sea necesario marca run breakpoint y hacer las mejoras necesarias

## *Medidas Relativas y absolutas*

## Fijas:

## px: Píxeles in: Pulgadas (1 in es igual a 96px) pt: Puntos (1 pt es igual a 1/72 in) cm: Centímetros mm: Milímetros pc: Picas

## Relativas: Son medidas relativas al contenedor del objeto.

em: Relativa al tamaño de la fuente del elemento actual  
rem: Relativa al tamaño de la fuente del elemento raíz (HTML)  
vw: Viewport Width, relativa al tamaño del viewport, sería el 1% de la anchura del viewport  
vh: Viewport Height, relativa al tamaño del viewport, sería el 1% de la altura del viewport

Si lo usamos para usar el width, un 80%, correspondería al 80% del ancho del contenedor padre

En el caso de EM, es relativa al tamaño del texto del contenedor padre. Siempre es relativo al tamaño del texto. Ejemplo 1 em, seria en principio en cualquier navegador ( a menos que se defina otro tamaño de fuente) 16px

Que conviene para los textos?

Conviene usar la medida relativa REM (o en su defecto Viewport Width), dada que esta se compara siempre en relación con el tamaño de fuente de raíz (el tamaño declarado en el body). Siempre va a usar este como parámetro comparativo, mas alla de que el contenedor del elemento tenga otro valor de fuente asignado

El Rem tiene problemas en navegadores antiguos,como el IE8, por esto lo que se puede hacer para solucionarlo, es poner la medida en rem y en px, o sea definir el tamaño en dos unidades distintas, que valgan lo mismo. Los navegadores que no son complatibles con ciertas cosas, simplemente ignoran esa línea que no es compatible, es como si no estuviera en el código

Estilos, atributos y valores CSS claves para repsonsive:

Es importante definir un box-Sizing para todo el documento, de modo que todos los navegadores tenga el mismo criterio al momento de calcular los tamaños de las cajas. Se recomienda usar el valor border-box, junto con el\* para que sea para todos los elementos de la pagina

Tambien se recomienda usar flexbox, min-width max-widht, etc

***Imágenes con la etiqueta PICTURE (HTML5)***

Esta es una etiqueta que permite ponerle varias imágenes, y que el navegador elija cual es la que debe usar según el momento adecuado. Mismo también tiene una imagen con la etiqueta img, la cual es la que se cargara en caso de que el navegador no sea compatible con esta etiqueta PICTURE

<picture>

<source media="(min-width: 900px)" srcset="grande.png">

<source media="(min-width: 550px)" srcset="media.png">

<img src="archivo-pequeno.jpg">

</picture>

Mismo también se pueden poner mas de una imagen para una resolución, y que según la densidad de pixeles el navegador elija una u otra. Esto se indica con un x2 o x3

<source media="(min-width: 900px)" srcset="foto.jpg, foto-hd.jpg x2"

Luego si pongo alguna imagen y no le especifico el mediaquery, entonces esa va a quedar como imagen predeterminada

***Webs pesadas para móviles***

Este es otro punto a tener en cuenta al momento de hacer adaptable nuestra pagina, dado que no podemos hacer una web muy robusta porque le traería problemas de carga a los dispositivos móviles

Tecnicas para evitar esto:

## \*Usar *Progressive Enhancement*

## *\*Desarrollo mobile first*

## *\*las imágenes suelen ser el 60% del peso del sitio. Es importante dar imágenes pequeñas a sitios móviles y mas grandes a sitios de escritorio. Usar convertidores a webp*

## *Herramienta importante para el diseño responsive: Sizzy*

## [*https://www.youtube.com/watch?v=qr77J2INRik&feature=youtu.be*](https://www.youtube.com/watch?v=qr77J2INRik&feature=youtu.be) *TUTORIAL DE SIZZY*

## <https://youtu.be/qr77J2INRik> <https://youtu.be/qr77J2INRik>