

Universidad Autónoma del Estado de Hidalgo

Escuela Superior de Tizayuca

Licenciatura en Tecnologías de la Información

Chavez Ramirez Brandon German

Ortega Jimenez Denilson

Torres Armas Mey-Lee Mayela

Semestre: 5 Grupo: 1

Desarrollo de aplicaciones móviles

Docente: Ing. José Alejandro Mondragón Arvi

Reporte de proyecto

Fecha de entrega: 07/Octubre/2022

# Técnicas de desarrollo de aplicaciones web responsivas en diversas plataformas.

---



## Introducción

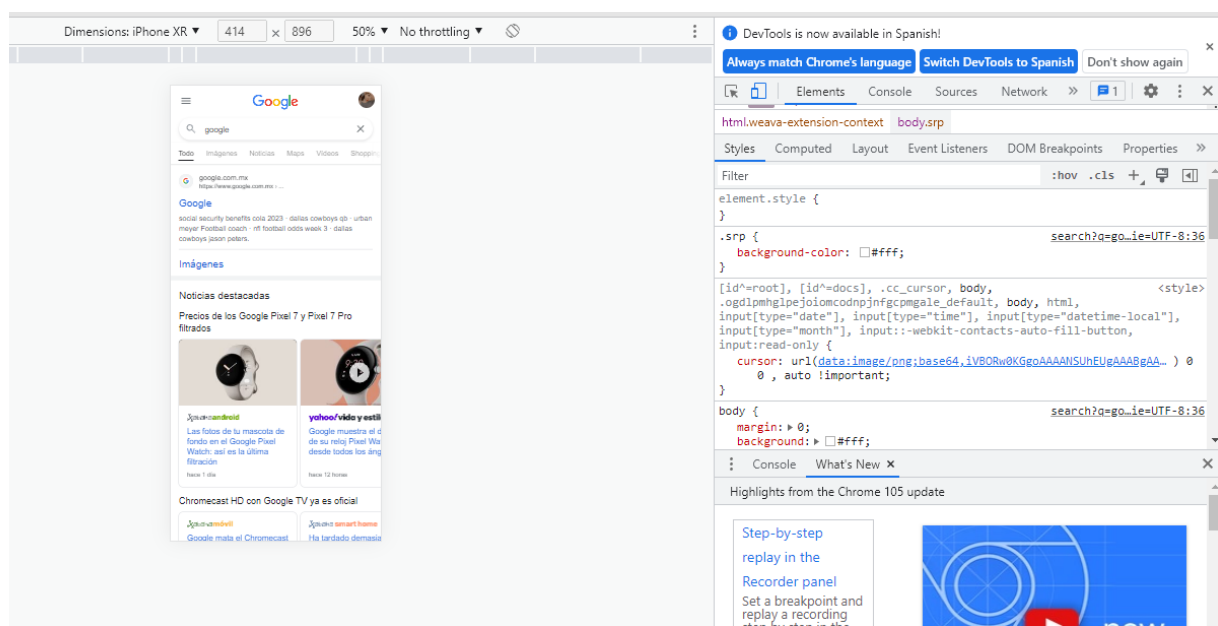
El diseño web responsive o adaptativo (Responsive Web Design) es una configuración en la que el servidor siempre envía el mismo código HTML a todos los dispositivos y se usa CSS para modificar el procesamiento de la página en el dispositivo.

En ocasiones, se confunde el responsive con las webs para móviles, pero no se trata de lo mismo. En el caso del diseño responsive, tenemos un solo sitio web que puede adaptarse para dispositivos de todo tipo, desde ordenadores de escritorio hasta móviles, pasando por tablets y otros. En cambio, crear un sitio móvil implica diseñar desde cero una web independiente, cuyos contenidos y formato están especialmente adaptados para funcionar mejor en dispositivos móviles.

## ¿Por qué Google prefiere el diseño responsive?

Google es a día de hoy un perfecto equilibrio entre eficiencia y calidad, que se traduce en eficacia. El hecho de tener una web adaptativa permite a Google 2 cosas. Por un lado, ofrecer en sus buscadores lugares que se preocupan en ofrecer la máxima comodidad a sus usuarios, y por otro lado tener que indexar un solo lugar.

A veces, es fácil decantarse por elegir un diseño para ordenador, o incluso crear una web para móvil y volcar allí todos los esfuerzos. Pero es un error. La carga de trabajo que se genera es doble, no sólo por repetir el proceso dos o más veces, sino por intentar posicionar doblemente una web. En este caso, lo más acertado es inclinarse por un diseño responsive, que nos permita la conexión y visibilidad desde cualquier dispositivo indexando una única web. A continuación se muestra el apartado de herramientas para desarrolladores el cual nos sirve para conocer las dimensiones de la página en cuestión.



## Diseño responsive multidispositivo

Cuando hablamos de diseño responsive nos referimos a la adaptación de un diseño para su correcta visualización en todo tipo de pantallas, es decir, no sólo pantallas de ordenador y portátil, sino también dispositivos móviles como smartphones y tablets. Esto es importante si tenemos en cuenta que cada vez

más gente accede a internet a través de sus dispositivos móviles y que éstos, les permiten estar conectados cada vez más tiempo y desde cualquier lugar.

El diseño responsive te permite llegar a un público más amplio con el que podrás satisfacer a los clientes que buscan una experiencia móvil atractiva. De hecho, la investigación muestra que puedes perder hasta el 90% de tus clientes potenciales debido a una mala experiencia optimizada para dispositivos móviles.

### ¿Cómo hacer un diseño web responsive para dispositivos móviles?

El meta tag viewport le indica al navegador cómo ajustar la página al ancho de cada dispositivo. Cuando el elemento meta viewport está ausente, los navegadores móviles mostrarán las páginas con su configuración de escritorio por defecto. Esto resulta en una experiencia que parece ser no responsive y de vista reducida.

A continuación una implementación estándar:

```
<meta name="viewport" content="width=device-width,initial-scale=1"/>
```

Las Media Queries son una parte importante del diseño web responsive comúnmente usadas para grid layouts, tamaños de texto, márgenes y padding que difieren según el tamaño de pantalla y la orientación.

A continuación un ejemplo de un caso de uso común en estilización Mobile First, en donde una columna tiene 100% de ancho para dispositivos pequeños, pero en viewports más amplios tiene un 50% de ancho.

```
.column {  
  width: 100%;  
}
```

```
@media (min-width: 600px) {
```

```
.column {  
  width: 50%;  
}
```

- El código anterior es un ejemplo simple, pero lo que realmente hace es bastante interesante. Considerando Mobile First, el elemento "column" se le definió un ancho de 100%;
- Usando una Media Query min-width, definimos reglas específicamente para viewports con un ancho mínimo de 600px (área visible más anchas que 600px). De esta manera, para viewports más anchos que 600px, nuestro elemento "column" tendrá un ancho correspondiente al 50% de su elemento padre.

### Desplazamiento horizontal con Overflow Scroll

Los usos comunes de esta técnica incluyen menús desplegables y tablas. A continuación un ejemplo de un menú desplegable.

```
<style>  
menu {  
  background: #d9d7d5;  
  padding: 0.25rem;  
  overflow-y: scroll;  
  white-space: nowrap;  
}  
span {  
  background: #767775;  
  color: #ffffff;  
  display: inline-block;  
  margin: 0.25rem;  
  padding: 0.5rem;
```



```
}  
</style>  
<menu>  
  <span>Diseño Responsive Web</span>  
  <span>RWD</span>  
  <span>menú Responsive</span>  
  <span>Ejemplo Overflow scroll</span>  
  <span>¡Esto es mucho contenido!</span>  
  <span>Si</span>  
  <span>Nosotros</span>  
  <span>tenemos</span>  
  <span>otro</span>  
  <span>elemento</span>  
</menu>
```

- *overflow-y: scroll* es el ingrediente clave de esta receta. Al definirlo, los elementos hijo se desbordarán sobre el eje horizontal con comportamiento de desplazamiento.
- Aunque pienses que *overflow-y* es suficiente, también debemos hacerle saber al navegador que no envuelva a los elementos hijos con *white-space: nowrap*.

### Imágenes responsive

Al usar atributos de tag modernos para imágenes, podemos acomodar un rango de dispositivos y resoluciones. A continuación, un ejemplo de una imagen responsive.



```
<style>
img {
  max-width: 100%;
}
</style>

<picture>
  <source type="image/webp"
srcset="https://my-image.com/my-image-100.webp 1x,
https://my-image.com/my-image-200.webp 2x">
  <source type="image/png" srcset="https://my-image.com/my-image-100.png
1x, https://my-image.com/my-image-200.png 2x">
  
</picture>
```

- Al definir *max-width: 100%* la imagen se va a ampliar o reducir basada en el ancho de su contenedor.
- Al usar la combinación de los tags *picture*, *source* e *img*, en realidad estamos renderizando solo una imagen y cargando la que mejor se adapte al dispositivo del usuario.
- WebP es un formato de imagen moderno que brinda compresión superior para imágenes en la web. Al utilizar *source* podemos hacer referencia a una imagen WebP para usar en navegadores que cuenten con el soporte

y agregar otro tag *source* para referenciar una versión PNG de las imágenes que no tienen soporte WebP.

- *srcset* es usado para decirle al navegador qué imagen utilizar basada en la resolución del dispositivo.
- Establecemos native lazy loading (carga diferida nativa) mediante el uso del par atributo / valor (attribute / value) *loading="lazy"*.

## Conclusiones:

Elegir este tipo de diseño para nuestra web es sinónimo de facilidad, suponiendo un gran acierto que nos permite las siguientes ventajas:

1. URL única, para una indexación común efectiva para un posterior posicionamiento.
2. Se realiza un solo diseño para los 3 dispositivos, ahorrándonos tiempo y costes.
3. Se evita duplicar contenido entre las diferentes versiones.
4. Diseño soportado por todos los navegadores.

Ahora, con la multitud de dispositivos diferentes, la cuestión ya no es solo esa, ahora se debe saber cómo se ve un sitio web en los diferentes tipos de tamaño de pantalla, ya sea un ordenador, un tablet o un móvil. Ahí es donde entra en juego la importancia del diseño web responsivo o Responsive Web Design, RWD por sus siglas en inglés.

## Referencias bibliográficas:

Ya, I. (2020, 8 octubre). La importancia de un diseño web responsive. INTERNET YA. Recuperado 23 de septiembre de 2022, de <https://www.internetya.co/la-importancia-de-un-diseno-web-responsive/>

Casals, C. (2019, 8 abril). Diseño Responsive: La mejor opción para una web multidispositivo. Actualidad eCommerce. Recuperado 23 de septiembre de 2022, de



<https://www.actualidadecommerce.com/disenio-responsive-web-multidispositivo/>

Boneu, M. S. (2021, 22 abril). Diseño Web Responsive — Cómo hacer que un sitio Web se vea bien en Teléfonos y Tablet. freeCodeCamp.org. Recuperado 23 de septiembre de 2022, de <https://www.freecodecamp.org/espanol/news/disenio-web-responsive-como-hacer-que-un-sitio-web-se-vea-bien-en-telefonos-y-tabletas/>