



REPORTE DE TÉCNICAS DE DISEÑO RESPONSIVO



Universidad Autónoma del Estado de Hidalgo

Elaborado por:

- **García Urzua Cristopher Abiran.**
- **Quezada Eurasmo Leidy Jocelin.**
- **Zarate Rosas Antonio.**

Docente: Mondragón Arvi José Alejandro.

Asignatura: Desarrollo de Aplicaciones Móviles

Programa educativo: Licenciatura en Tecnologías de la Información

Semestre: 5

Grupo: 1

Fecha de entrega: 29 de octubre de 2021

Técnicas para hacer diseño responsivo.

Modelos de navegación



Pasar una web pensada para una pantalla de escritorio a otra más reducida no es una tarea sencilla. Tenemos menos espacio para mostrar el contenido principal y todo lo que podamos ahorrar en la navegación mejorará el impacto visual de texto o imagen. Podemos optar por varias soluciones para resolver el menú con diferentes modelos de navegación adaptativa para móviles.

Navegación por interruptor o toggle.

Esta funciona como un botón de encendido y apagado, cuando clicamos sobre él se nos muestra el menú y cuando volvemos a clicar desaparece. Es como un interruptor. De las páginas adaptativas, es una de las soluciones que más se suelen ver. De hecho, muchos frameworks de estilos, como el Bootstrap from Twitter y Foundation, ya lo llevan incorporado por defecto. Además, existen plugins muy livianos sin necesidad de jQuery para este tipo de propósitos.

El hecho de que necesitemos de un archivo externo y que la escalabilidad tenga muchos submenús es un hándicap. Por otro lado, mantiene al usuario en la parte superior de la página y reduce el espacio para mostrar el menú en los dispositivos móviles, que es algo muy importante.

Navegación al estilo app móvil.

Uno de los recursos más elegantes y que cada vez vemos más en las aplicaciones para iOS, Android y otros sistemas operativos para dispositivos móviles, es la navegación adaptativa donde el contenido del menú está oculto a un lado de la pantalla y, cuando se presiona el botón, aparece el menú desplazándose el contenido principal hacia un lado.

Es un tipo de navegación que popularizó Facebook para móviles y que muchas otras han imitado. Aplicaciones como Google +, Evernote, etc. han optado por realizarlo de este modo.

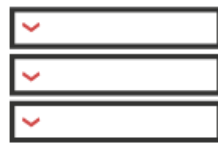
Transformar el menú en un select

Otra de las opciones para navegar dentro de un dispositivo móvil es usar el típico menú de select propio de los formularios. Dentro de lo que cabe es bastante fácil

de implementar, también existen librerías diseñadas para esta funcionalidad y contamos con la ventaja de que el usuario lo identifica y reconoce rápidamente.

Igualmente, ahorra un montón de espacio para mostrar el contenido principal y mantiene la navegación en la parte superior. Los submenús los muestra de una forma sencilla, con un guión separado de su elemento padre. TinyNav.js o este tutorial de CSS Tricks nos enseñan cómo hacerlo.

Elementos según el dispositivo



En referencia al contenido, podemos realizar diferentes presentaciones o cambiar elementos según el dispositivo, igual que no es lo mismo una redacción de una noticia para radio o TV, o una pieza de un medio offline u online. Lo primero es conocer bien el propósito de nuestra comunicación, de este modo tendremos más claro lo que debemos discriminar o mejorar para nuestra adaptación.

Existen dos estrategias básicas: eliminar el contenido o mejorar el contenido presentándolo de forma distinta (cambiando el orden, sustituyéndolo por iconos, etc.).

Eliminando contenido tendremos una representación más asequible, pero no olvidemos que las tareas y las navegaciones más básicas deben ser igual de satisfactorias en los distintos entornos. Podemos encontrarnos con sitios donde directamente no se hace nada o bien se pone un link para mostrar el contenido como si fuera una versión de escritorio. Este método es muy poco aconsejable y Jakob Nielsen ya lo comenta en su artículo *Mobile Site vs. Full Site*.

Otra posibilidad es jugar con la revelación de contenido, como hace de manera exquisita la Wikipedia, que al tener un altísimo volumen de información en algunos de sus artículos lo muestra mediante pestañas que abren el resto del contenido. Asimismo, tanto en la versión escritorio como en la versión móvil podemos ver claramente los iconos que utiliza.

Imágenes panorámicas



El formato de las imágenes debe ser lo más apaisado posible para combinar el uso de estas con el texto. Una de las principales características de la interacción móvil es el ritmo (scroll) vertical. Esto lo podemos constatar en The Next Web: hace unos años, a principios de 2011, el diseño estaba poco pensado para el consumo móvil -también es cierto que no existían tantos smartphones-, pero si nos fijamos en las imágenes observamos que el formato cuadrado no ocupa toda la pantalla del dispositivo y perdemos parte de la información visual.

Por el contrario, en las versiones actuales optan por un formato más panorámico para adaptarse mejor a todos los dispositivos.

Peso.



Disponemos de varios sistemas para optimizar las imágenes. Uno de ellos es la librería Picturefill, que dependiendo del ancho de la pantalla presentará una imagen distinta. Para ello necesitamos cambiar nuestro marcado en el html y disponer de las diferentes imágenes en el servidor.

Esta técnica nos permite usar diferentes imágenes en el ámbito de la importancia de los contenidos, porque no solo basta con tener imágenes con menor peso.

Iconos.



Los iconos simbolizan información o funcionalidad mediante una representación gráfica. Según el tamaño de la pantalla van adquiriendo mayor importancia porque no se pueden representar o ayudar con texto (en el caso que usemos tooltips). Además, debemos mostrar una serie de convenciones para que el usuario lo

entienda a la primera (lápiz-editar, floppy-guardar, etc.). Los paneles de gestión, sistemas operativos y aplicaciones móviles han ayudado mucho al respecto.

Touch.



En el desarrollo no debemos olvidar los gestos dentro de una pantalla táctil. Al presentar un contenido donde predomine el texto e imagen, probablemente lo más común con lo que nos encontremos, será un slider. Una pequeña ayuda explicando cómo funciona el scroll horizontal servirá para nuestro cometido.

Existen una infinidad de librerías que nos ayudan con esto, y cada vez encontraremos más. Algunas de las más conocidas son jQuery Mobile, Hammer.js, jqMobi, Zepto.

Otra opción es utilizar los scripts para detectar el navegador del dispositivo. Los tenemos en diferentes lenguajes para poder incluirlo en nuestro proyecto.

Este proceso de adaptación de dispositivos de sobremesa a dispositivos móviles genera muchos cambios y los gestos se van introduciendo poco a poco. Las aplicaciones móviles son las que más provecho le están sacando a los gestos de la mano, pero no nos extrañará ver cómo los navegadores y websites van incorporando también estos movimientos, muy similares a los que podemos encontrar en un trackpad.

Botones grandes.



Adaptar los formularios a la pantalla del usuario. Una de las acciones que más realizará un usuario en nuestra web adaptada al móvil será interactuar mediante inputs, ya sea para comentar un contenido que hayamos publicado, registrarse o cualquier tipo de conversión o lead. Lo resolveremos como en una aplicación móvil, buscando la máxima sencillez. En este caso el input irá debajo del label del campo.

No nos olvidamos los botones. Es cierto que la tendencia en los últimos años es hacer los botones de submit y call to action con un tamaño notable. Aun así, todavía nos podemos encontrar botones minúsculos para los dispositivos móviles.

En los dispositivos móviles perdemos parte de información al rellenar los formularios al mostrarse el teclado en pantalla. Perdemos parte de contexto y el usuario no sabe realmente lo que está completando.

Carga secuenciada



Cuando hablamos de navegación móvil no hemos de desatender los puntos de corte. Éstos nos indican el ancho de pantalla del que disponemos para realizar unas modificaciones u otras. Podemos usarlos, mediante Javascript, para que haga unas funcionalidades determinadas. Existen muchos plugins como Breakpoints.js o Resonse.js. Del mismo modo, también definimos los breakpoints en los CSS con los famosos media queries.

Al tener una pantalla más reducida el contenido crece en sentido vertical de un modo que el scroll se hace casi interminable. Para esto tipo de casos nos puede interesar una carga secuenciada.

Pinterest y muchos sites al estilo aplicación (Facebook, Tumblr, Twitter...) disponen de él. Como en casi todo, podemos obtener plugins que implementen esta funcionalidad: jScroll y jQuery Waypoints serían un ejemplo de ello.

Imágenes responsive.

El siguiente código de CSS sirve para hacer que las se muestren tan grandes como sea posible. Por ejemplo, si la imagen se muestra en un contenedor que tenga 600px de ancho, la imagen se mostrará a 600px de ancho. Si, por ejemplo, cambiásemos el ancho de ese contenedor a 350px, la imagen se ajustaría automáticamente a la anchura máxima disponible de su contenedor, en este caso 350px.

```
img {  
    max-width: 100%;  
}
```

Vídeos de HTML5 responsive

Solo tienes que añadir el siguiente snippet en el código CSS y el vídeo se ajustará al ancho del contenedor.

```
video {  
    max-width: 100%;  
    height: auto;  
}
```

Menú de navegación responsive

Los menús de navegación son la vía más simple que tienen tus visitantes de encontrar el contenido que andan buscando en tu web. Pero cuando acceden a tu sitio web desde, por ejemplo, un dispositivo móvil, los menús de navegación, si tu web no es responsive, se convierten en un verdadero tormento haciendo imposible su uso. De hecho, para pantallas pequeñas, la mejor manera de utilizar un menú de navegación es transformarlo en un select en vez de mostrar el típico menú horizontal de la zona superior de la web.

HTML

```
<nav>  
  
  <ul>  
    <li><a href="/" class="active">Home</a></li>  
    <li><a href="/collections/all">Books</a></li>  
    <li><a href="/blogs/five-simple-steps-blog">Blog</a></li>  
    <li><a href="/pages/about-us">About Us</a></li>  
    <li><a href="/pages/support">Support</a></li>  
  </ul>  
  
  <select>  
    <option value="" selected="selected">Select</option>  
  
    <option value="/">Home</option>  
    <option value="/collections/all">Books</option>  
    <option value="/blogs/five-simple-steps-blog">Blog</option>  
    <option value="/pages/about-us">About Us</option>  
    <option value="/pages/support">Support</option>  
  </select>  
  
</nav>
```

CSS

```
nav select {  
    display: none;  
}  
  
@media (max-width: 960px) {
```

```
nav ul    { display: none; }  
nav select { display: inline-block; }  
}
```