

Desarrollo Responsivo

El primer elemento que se debe utilizar para convertir un sitio a versión responsiva se relaciona con la etiqueta *view-port* la cual indica el comportamiento de la página al cargarse.

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

En este caso la propiedad de ***width*** con el valor ***device-width*** indica el ancho del dispositivo, la escala indica que será de 1.

Para convertir un sitio a responsivo, se deben considerar los siguientes escenarios:

- Cambio de elementos de estructura general
- Cambios de tamaño de caracteres
- Cambio de elementos particulares que no desbordan
- Cambio de elementos particulares que se desbordan

Los tamaños más comunes para versiones responsivas incluyen resoluciones que van de: 320 – 400px, 400 – 700px y mayores de 800px, lo que se consideraría una PC de escritorio.

Cambio de Elementos de Estructura General

Lo primero a realizar es cambiar los anchos de los elementos generales para que no se reflejen en píxeles, sino en porcentaje.

Al contenedor general se le especifica un tamaño máximo y un tamaño mínimo, para no permitir que se reduzca más allá de eso.

general.css

```
#envase{  
max-width:1000px;  
min-width:320px;  
background-color:#FFFFFF;  
text-align:left;  
margin:auto;  
}
```

En este caso se maneja un ancho máximo de 1000px y uno mínimo de 320px. De momento, esto solo limita el tamaño mínimo que tendrá la pantalla al reducirla.

Manejando Porcentajes

En los elementos que forman la estructura general, se comienza a cambiar el ancho del elemento de píxeles a porcentajes. En este caso se cambiará el ancho del *banner* y del *footer*. Como ambos

elementos miden el ancho del contenedor, su ancho en porcentaje es de 100%

general.css

```
header{
background-color:#640923;
height:100px;
margin-bottom:3px;
width:100%;
}
footer{
background-color:#640923;
height:60px;
text-align:center;
padding-top:20px;
margin-top:5px;
width:100%;
}
```

Cambio en los Elementos dentro del Contenedor

Para cambiar el ancho de los elementos a porcentaje cuando están dentro de un contenedor, hay que considerar el ancho del contenedor en el que se encuentran. Por ejemplo, para cambiar el ancho de píxeles a porcentaje del *cuerpo_principal*, hay que considerar su ancho actual (640px) y el máximo de su contenedor (1000px), por lo que el porcentaje sería $(640/1000)*100\% = 64\%$.

```
#cuerpo_principal{
width:64%;
margin-left:5px;
float:left;
border: 1px solid #ddd9d9;
padding:5px;
min-height:800px;
}
```

Lo mismo se hará con los elementos *barra_lateral_principal* (170px) y *menu_principal* (150px), que quedan de la siguiente manera:

```
#barra_lateral_principal{
width:17%;
margin-left:5px;
float:left;
border: 2px solid #640923;
border-radius:5px;
padding:2px;
padding-bottom:150px;
}
```

```
#menu_principal{
width:15%;
margin-left:5px;
padding-bottom:150px;
float:left;
border: 2px solid #640923;
border-radius:5px;
}
```

Cambio en el Tamaño de Caracteres

Para manejar de manera adecuada el tamaño de caracteres, conviene no expresarlo en píxeles, sino en unidades **em** que es una expresión de un tamaño relativo a un tamaño base. Se debe tener un tamaño base en el documento, por ejemplo en el *body* o especificarlo por defecto con ***, en este caso se tiene un tamaño de 20px como base.

```
{
font-size:20px;
color:#000000;
margin:0px;
}
```

Para cambiar el tamaño de las letras de **px** a **em**, se toma en consideración el valor base del padre inmediato y se realiza la siguiente operación:

$$em = destino/base$$

Si se quiere cambiar un tamaño que no es hijo directo del elemento que es la base, se debe considerar como base el tamaño en **px** que tendría el padre inmediato.

Esto se puede evitar indicando que se tienen **rem** en lugar de **em**, pero algunos navegadores todavía no soportan la implementación de **rem**.

Por ejemplo, para cambiar el valor del tamaño del elemento `<h1>` del header (30px), se considera:

$$30/20 = 1.5 em$$

```
header h1{
font-size:1.5em;
color:#FFFFFF;
margin:0px;
text-align:center;
padding-top:20px;
}
```

En caso del **footer**, en particular para las ligas, como se tenía un tamaño por defecto, se consideraba el base, por lo que el valor en **em** sería de 1em

```
footer a{
display:inline;
padding-left:60px;
text-decoration:none;
color:#e098ac;
font-size:1em;
}

footer a:hover{
color:#FFFFFF;
font-size:1em;
}
```

De esta manera se cambiarán los tamaños de letra de todos los elementos en las diferentes hojas de estilo.

Cambios de Tamaño en los Elementos Específicos

Es posible cambiar el tamaño de los elementos de píxeles a porcentaje sin tener que realizar cambios mayores en la estructura.

principal.css

Se especifica un ancho de 100% de la lista de ligas en la página principal respecto a su contenedor cuyo tamaño original era de 640px.

```
#lista_index{
text-align:center;
margin:auto;
width:100%;
}
```

```
#lista_index a{
padding:5px;
border: 1px solid #BFBFBF;
background-color:#FFFFFF;
border-radius:5px;
display:block; /*Hace que la liga afecte a todo lo que está dentro*/
text-decoration:none;
width:25%;
```

```
height:210px;
list-style:none;
float:left;
margin-left:10px;
margin-bottom:10px;
}
```

Dentro de cada liga se tiene una imagen con un ancho de 150px, en este caso para cambiar el ancho, se toma como base el de su contenedor que era de 160px.

```
#lista_index a li figure img{
width:93.7%;
height:150px;
}
```

En la plantilla de información, el texto correspondiente a la introducción no se ajusta ya que tiene un ancho fijo de 500px, por lo que hay que cambiarlo respecto al de su contenedor, que era de 640px. En este caso se tendría:

informacion.css

```
#introduccion{
padding:5px;
width:78.125%;
text-align:center;
margin:auto;
}
```

Sucede lo mismo con el Título del Artículo que se encuentra en el *header* del *cuerpo_principal*, considerar que originalmente no ocupa el 100%, ya que *cuerpo_principal* tiene un *padding* de 5px por lado, adicionalmente se consideran 5px del redondeo, por lo que el tamaño real sería 625 px. En este caso se tiene:

$$(625/640)*100 = 97.656\%$$

```
#cuerpo_principal header{
padding:5px;
height:30px;
background-color:#640923;
font-size:1em;
border-radius:5px;
color:#FFFFFF;
width:97.656%;
}
```

La imagen también tenía un tamaño establecido, cuando se quiere cambiar el tamaño de la imagen, conviene hacerlo en su contenedor y luego la imagen ocupará el 100% del contenedor. El contenedor tenía un ancho de 300px por lo que ahora queda de la siguiente manera, considerando que el contenedor mide 640px.

```
#cuerpo_articulo figure{  
margin:5px;  
float:left;  
width:46.8%;  
}
```

La imagen ocupa el 100% del tamaño del contenedor.

```
#cuerpo_articulo figure img{  
width:100%;  
height:300px;  
}
```

De manera similar ocurre con los bloques de comentarios que tenían un ancho de 500px

```
.comentario_usuario{  
width:78.125%;  
font-size:1em;  
border: 1px solid #640923;  
margin-bottom:10px;  
margin-top:5px;  
border-radius:5px;  
color:#FFFFFF;  
}
```

Media Queries

Un manejo más avanzado del comportamiento de elementos se realiza a través de las *media queries*.

Las *media queries* son sentencias de CSS que se aplican al diseño, pero solo si se cumplen ciertas condiciones, se pueden aplicar a nivel de HTML o a nivel de CSS.

En este caso solo se presentará una introducción a las mismas dependiendo de las condiciones de tamaño de la resolución.

La especificación es de la siguiente manera:

```
@media screen and ([CONDICION]) {  
/* reglas de diseño para el ancho o resolución */  
}
```

En este caso si se quiere utilizar una serie de sentencias dependiendo del ancho, se tienen tres elementos :

Sentencias en resoluciones de menos de X píxeles de ancho:

```
@media screen and (max-width: ANCHOpX) {  
/* reglas de diseño para el ancho ANCHO o menos */  
}
```

Sentencias en resoluciones de más de X píxeles de ancho:

```
@media screen and (min-width: ANCHOpX) {  
/* reglas de diseño para el ancho ANCHO o más */  
}
```

Sentencias en resoluciones entre X y Y píxeles de ancho:

```
@media screen and (min-width: ANCHO_Xpx) and (max-width: ANCHO_Ypx){  
/* reglas de diseño para el ancho entre ANCHO_X y ANCHO_Y */  
}
```

Se recomienda colocar las *media queries* al final del documento CSS para no confundirse.

Considerar que las *media queries* tienen un comportamiento en cascada, si ya se especificó un valor para una resolución menor a 800, esta se mantendrá para resoluciones menores a menos que en esa resolución menor se le cambie el efecto.

Cambiando Comportamientos y Tamaños Dependiendo la Resolución

Modificando el Tamaño de la Fuente

Lo primero que se realizará será cambiar el tamaño de las fuentes dependiendo de qué resolución se esté trabajando, para esto se puede realizar de dos maneras:

- Cambiar el tamaño de la letra “base” (la de *) y eso afectará al resto de las fuentes que ya están expresadas en *em*.
- Cambiar el tamaño de la letra (en píxeles) de un elemento en particulares

Se manejarán dos resoluciones, menores de 700px y menores de 400px

general.css

```
@media screen and (max-width: 700px){  
  
*{  
font-size:18px;  
}  
  
}  
  
@media screen and (max-width: 400px){  
}
```

Sin embargo, el tamaño de la fuente del encabezado es muy grande cuando se tiene una resolución de 320px, si se quiere cambiar solo ese tamaño, sin afectar al resto se modifica la sentencia de la siguiente manera:

```
@media screen and (max-width: 400px){  
header h1{  
font-size:16px;  
margin:auto;  
max-width:80%;  
}  
}
```

Cambiando el Espaciado y Alineación

En el *footer*, cuando la resolución disminuye, los elementos no se distribuyen bien, esto se debe a la alineación y al espaciado entre ellos, por esta razón se cambiará la alineación de los mismos y el espacio.

Cuando la interfaz está entre 400 y 700px, se desea que solo se reduzca un poco el espacio entre elementos.

general.css

```
@media screen and (max-width: 700px){
*{
font-size:18px;
}
footer a{
padding-left:20px;
font-size:0.8em;
height:auto;
}
footer a:hover{
font-size:0.8em;
}
}
@media screen and (max-width: 400px){ {
header h1{
font-size:16px;
margin:auto;
max-width:80%;
}
}
footer a{
display:block;
}
}
```

Cambiando el Alto de un Elemento

En la plantilla de la información genérica, el título se desborda cuando el tamaño de la interfaz ya es pequeño, por lo que es necesario ajustarlo. Para esto se podría reducir más el tamaño de la fuente o dejar el alto del título en automático.

generica.css

```
@media screen and (max-width: 400px){
#cuerpo_principal header{
height:auto;
font-size:1em;
}
}
```

En la plantilla del artículo, si el título es muy largo, comienza a desbordar al contenedor, en este caso hay que manejar que éste contenedor no tenga un alto predefinido cuando se alcanza cierta resolución.

Se desea que si es menor a 800px, el alto del contenedor del título se ajuste de manera automática, para esto se utiliza la siguiente secuencia de instrucciones.

informacion.css

```
@media screen and (max-width: 800px){
#cuerpo_principal header{
height:auto;
}
}
```

La imagen sigue flotada sin importar la resolución, lo cual no le da una buena vista al elemento, es necesario eliminar el efecto de flotado, pero solo para resoluciones menores a 400px.

informacion.css

```
@media screen and (max-width: 400px){
#cuerpo_articulo figure{
width:100%;
}
}
```

Eliminando Imágenes de una Lista

En las listas que contienen imágenes (*index* y *seleccion*), al disminuir el tamaño de la interfaz, las imágenes ya no se aprecian bien porque se reducen, en este caso lo mejor es indicar que no aparecerán.

seleccion.css

```
@media screen and (max-width: 400px){
.lista_imagen li a figure{
display:none;
}
}
```

En las listas de la página principal hay que realizar más modificaciones, hay que eliminar la imagen, indicar que las listas no están flotando y ajustar los textos

principal.css

```
@media screen and (max-width: 1000px){
#lista_index a{
height:auto;
}
}
@media screen and (max-width: 600px){
#lista_index{
padding:0px;
width:90%;
margin-left:0px;
}

#lista_index a{
width:90%;
}

#lista_index a li h1{
text-align:left;
}

#lista_index a li figure img{
display:none;
}
}
```

Afectando la Estructura de los Elementos

La barra de ligas laterales no se ha modificado y por esa razón se coloca debajo ocupando un espacio mucho menor a lo deseado, para corregir esto, se debe indicar de al tener una resolución menor al tamaño máximo del contenedor principal, la barra deje de flotar. Será necesario ajustar los tamaños del cuerpo principal y de la misma barra.

general.css

```
@media screen and (max-width: 1000px){
#barra_lateral_principal{
width:97%;
margin-left:5px;
margin-top:10px;
padding-bottom:10px;
}
#cuerpo_principal{
width:81%;
}
```

```
}  
}
```

Adicionalmente se cambia la orientación para que aparezcan una delante de la otra.

```
@media screen and (max-width: 1000px){  
#barra_lateral_principal{  
width:97%;  
margin-left:5px;  
margin-top:10px;  
padding-bottom:10px;  
}  
#cuerpo_principal{  
width:81%;  
}  
#ligas_laterales li{  
float:left;  
}  
}
```

Ocultando y mostrando secciones

Las tablas se muestran de manera correcta con una resolución grande, sin embargo en algún momento comienzan a desbordarse, en este caso se tienen dos secciones con dos representaciones diferentes para la información de las tablas.

Una sección muestra las tablas como tales, mientras que otra, muestra una lista. Lo primero será ocultar la lista.

tablas.css

```
#tablas_resp{  
display:none;  
}
```

Las columnas se van adaptando al tamaño del contenedor, pero en algún momento ya no se ven de manera adecuada y posteriormente se desbordan. Es necesario indicar que en cierta resolución, desaparecerán y aparecerá la sección con la representación de lista.

```
@media screen and (max-width: 600px){  
#tablas{  
display:none;  
}
```

```
#tablas_resp{  
display:block;  
}  
}
```

Al seguir reduciendo la resolución, es necesario hacer que el elemento no luzca como flotado, esto se puede lograr haciendo que el elemento flotado abarque el 100% del contenedor. Se manejará esta propiedad para una resolución menor a 500px.

```
@media screen and (max-width: 500px){  
.encabezado{  
width:100%;  
}  
  
.contenido{  
margin-left:0px;  
}  
}
```

Menú Responsivo

Cuando se tiene un menú lateral con las secciones desplegadas en bloque (una debajo de otra), el primer paso es cambiarlo a un menú horizontal en la parte superior de la página. El cambio se realizará cuando se tengan menos de 650px.

general.css

```
@media screen and (max-width: 650px){
#menu_principal{
width:97%;
padding-bottom:5px;
margin-bottom:5px;
background-color:#640923;
}
#cuerpo_principal{
width:95%;
}
#menu_principal ul li{
display:inline;
}
#menu_principal ul li a{
text-decoration:none;
color:#FFFFFF;
font-size:0.7em;
padding-left:20px;
text-align:right;
}
}
```

Este cambio de posición requerirá un ajuste en el tamaño del resto de los elementos. Sin embargo, cuando la pantalla sigue reduciéndose, comienzan a no mostrarse de manera correcta las ligas, será necesario realizar un nuevo ajuste para resoluciones menores.

Primero se cambiará el menú para cuando mida menos de 550px, en este caso se desea que las ligas del menú aparezcan una debajo de la otra y ocupen todo su contenedor.

La propiedad box-sizing: border-box hace que el ancho total ya considere el padding y los bordes.

general.css

```
@media screen and (max-width: 550px){
#menu_principal ul li{
width:100%;
```

```

display:block;
box-sizing:border-box;
text-align:right;
height:50px;
}

#menu_principal ul li:hover{
background-color:#FFFFFF;
border-radius:5px;
}

#menu_principal ul li a{
display:block;
width:100%;
box-sizing:border-box;
height:50px;
}
}

```

Control del Menú Responsivo

Es necesario crear un elemento que sirva para manejar el menú responsivo, en este caso se creará como una **div class** llamado **control_menu** el cual se colocará dentro del <nav> del menú, pero fuera de los elementos de la lista.

index.html

```

<nav id="menu_principal">
  <div class="control_menu"> Menú</div>
<ul>
  <li><a href="">Inicio</a></li>
  <li><a href="plantillas/tablas.html">Ejemplo Tablas 1</a></li>
  <li><a href="#">Ejemplo Tablas 2</a></li>
  <li><a href="#">Ejemplo Tablas 3</a></li>
  <li><a href="#">Ejemplo Tablas 4</a></li>
</ul>
</nav>

```

Se le da estilo para que combine con el resto del menú, es estilo es independiente del tamaño, por lo que va fuera de cualquier **media query**

La propiedad **cursor:pointer** hace que el elemento se comporte como si fuera una liga, pero sin referencia a alguna dirección en otra página o en la misma.

general.css

```
.control_menu{
width:96%;
background-color:#640923;
text-align:right;
padding:15px 10px;
cursor:pointer;
color:#FFFFFF;
}
```

El problema es que ahora el elemento aparece siempre sin importar la resolución con la que se esté trabajando. Por esa razón se ocultará.

general.css

```
.control_menu{
width:96%;
background-color:#640923;
text-align:right;
padding:15px 10px;
cursor:pointer;
color:#FFFFFF;
display:none;
}
```

El menú en la resolución menor a 550px siempre está visible, lo que se desea es que desaparezca y aparezca el control del menú que se creo, para esto es necesario agregar las siguientes líneas en el media query correspondiente a los 550px.

general.css

```
#menu_principal ul{
display:none;
padding:0px;
margin-bottom:0px;
}
.control_menu{
display:block;
}
```


Ahora hay que hacer que al dar clic sobre el control, aparezca el resto del menú para poder seleccionar un elemento. Para esto se utilizará una de las funciones disponibles en jQuery. Para poder utilizar jQuery hay que indicar en donde están los archivos con las funciones, para esto se tienen dos opciones:

Indicar una ruta a partir del desarrollador en el <head> de la página:

```
<script src='http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js'  
type='text/javascript'></script>
```

Aunque para esto se necesita tener acceso a Internet.

La otra opción es incluir el archivo en el proyecto para tener acceso a él incluso sin conexión. Para eso es necesario descargar la versión más reciente de jQuery, por el momento con la versión mínima será suficiente.

La descarga se puede realizar de:

<http://jquery.com/download/>

Descargando la versión comprimida 1.11.3, una vez descargada se creará una carpeta llamada **js** en la estructura del proyecto y se copiará ahí el archivo **jquery-1.11.3.min.js**

Una vez copiado el archivo, es necesario indicar la ruta en donde se encuentra dentro del **head** de la página.

index.html

```
<script type="text/javascript" src="js/jquery-1.11.3.min.js"></script>
```

Lo siguiente es invocar a una función que realice la transición de mostrar u ocultar el menú al dar clic sobre el control.

```
<script type="text/javascript">  
$(document).ready(function() {  
$('.control_menu').click(function() {  
$('#menu_principal > ul').slideToggle(1500);  
});  
});  
</script>
```

En este caso se indica que al dar clic sobre el elemento **control_menu**, el elemento #menu_principal ul aparecerá en 1.5 segundos (1500ms).

Se recomienda colocar esto al inicio de la página después de abrir <body>, también podría colocarse en un archivo separado, pero por el momento al ser la única función a utilizar se dejará en el código.

Colocando el Ícono de Menú Responsivo

Muchas personas han asociado un determinado ícono al menú en dispositivos móviles, para colocarlo basta con indicar que se tiene una imagen en lugar de la palabra Menú en el <div> correspondiente, en este caso se utiliza la imagen **menu_responsivo.png**

index.html

```
<div class="control_menu">

</div>
```

Lo único que resta es copiar la nueva estructura del menú a todas las plantillas, así como el código para incluir el archivo de funciones de jQuery y el código del script. Es necesario realizar el ajuste en las rutas tanto para el archivo de jQuery como para la imagen del menú en el resto de las plantillas

```
<script type="text/javascript" src="!!/js/jquery-1.11.3.min.js"></script>
```

```
<div class="control_menu">

</div>
```