

Comportamiento Responsivo con Bootstrap

Bootstrap ofrece como parte de sus características un comportamiento responsivo que ofrece buenos resultados, sin embargo en ocasiones es necesario realizar una personalización del comportamiento de diversos elementos.

Al cambiar el tamaño de la pantalla, la mayoría de los elementos se adaptan automáticamente, pero otros si necesitan ser ajustados.

Para el ajuste se utilizarán los puntos de ruptura disponibles para Bootstrap.

- LG Mayor a 1200px
- MD Entre 992 y 1200px
- SM Entre 768 y 991px
- XS Menor a 768px

Corrigiendo el Banner

Dentro de los elementos que no se acomodan de manera correcta está el banner, el cuál tiene el logo de la UAM, y dos textos. Al reducir la pantalla, los textos desaparecen. Esto sucede por que Bootstrap por defecto hace que todas las columnas ocupen un 100% (12 unidades) al reducir el tamaño de la pantalla.

La estructura del banner es la siguiente:

index.html

```
<header class="row banner bg-fondo_guinda">
  <section class="col-lg-12">

    <section class="col-lg-3 contenedor_imagen">
      <figure>
        
      </figure>
    </section>

    <section class="col-lg-9 contenedor_info_header">

      <section class="col-lg-12 bloque_header_A">
        <h1>Trabajando con Bootstrap</h1>
      </section>

      <section class="col-lg-12 bloque_header_B">
        <h2>Creando la Primera Plantilla</h2>
      </section>

    </section>
  </section>
</header>
```

La cual tiene dos secciones dentro de otra sección. El tamaño en columnas de estas dos secciones es de 3 para la de la imagen **contenedor_imagen** y de 9 para el texto **contenedor_info_header**. Cuando la resolución es menor a su punto de ruptura (**lg**), cada una ocupa 12 columnas y al tener el banner un alto fijo, éstas desaparecen.

Se agregará un nuevo punto de ruptura para que esto no suceda, se desea que para tamaños medianos (**MD**) y pequeños (**SM**), el comportamiento sea el mismo. Notar que si se coloca un nuevo punto de ruptura para elementos relacionados entre sí, la suma también debe ser de 12 columnas.

index.html

```
<header class="row banner bg-fondo_guinda">
  <section class="col-lg-12">

    <section class="col-lg-3 col-md-3 col-sm-3 contenedor_imagen">
      <figure>
        
      </figure>
    </section>

    <section class="col-lg-9 col-md-9 col-sm-9 contenedor_info_header">

      <section class="col-lg-12 bloque_header_A">
        <h1>Trabajando con Bootstrap</h1>
      </section>

      <section class="col-lg-12 bloque_header_B">
        <h2>Creando la Primera Plantilla</h2>
      </section>

    </section>

  </section>
</header>
```

Aunque no es notorio por el menú, el contenido del banner se desborda. Lo mejor será ocultar el contenido de texto al llegar a un tamaño pequeño.

index.html

```
<section class="col-lg-9 col-md-9 col-sm-9 hidden-xs contenedor_info_header">

  <section class="col-lg-12 bloque_header_A">
    <h1>Trabajando con Bootstrap</h1>
  </section>

  <section class="col-lg-12 bloque_header_B">
    <h2>Creando la Primera Plantilla</h2>
  </section>

</section>
```

Corrigiendo el Footer

El Footer tiene un comportamiento similar, por lo que se realizarán modificaciones para que en todo momento sean visibles los datos.

En particular se trabajará con los tres bloques de cuatro columnas.

index.html

```
<section class="col-lg-4 bloque_footer_A">
  BLOQUE FOOTER A
</section>
<section class="col-lg-4 bloque_footer_A">
  BLOQUE FOOTER B
</section>
<section class="col-lg-4 bloque_footer_A">
  BLOQUE FOOTER C
</section>
```

Al agregar el manejo de los puntos de ruptura se tiene

index.html

```
<footer class="row footer_sitio bg-fondo_guinda">
  <section class="col-lg-12 titulo_footer">
    INFORMACION CENTRADA
  </section>
  <section class="col-lg-4 col-md-4 col-sm-4 col-xs-4 bloque_footer_A">
    BLOQUE FOOTER A
  </section>
  <section class="col-lg-4 col-md-4 col-sm-4 col-xs-4 bloque_footer_A">
    BLOQUE FOOTER B
  </section>
  <section class="col-lg-4 col-md-4 col-sm-4 col-xs-4 bloque_footer_A">
    BLOQUE FOOTER C
  </section>
</footer>
```

Modificación del Menú

Para crear un menú responsivo se deben realizar los siguientes cambios:

```
<section class="row">
  <nav class="navbar menu_principal">
    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
    data-target=".menu_p" aria-expanded="false" aria-controls="navbar">
      <span class="glyphicon glyphicon-menu-hamburger logo_menu"></span>
    </button>
    <div class="navbar-collapse collapse menu_p">
      <section class="navbar-header encabezado_menu">
        <a class="navbar-brand" href="#">Esto es Importante</a>
      </section>
      <ul class="nav navbar-nav elementos_menu">
        <li class="active"><a href="#">Inicio</a></li>
        <li><a href="plantillas/textos.html">Texto e Imágenes</a></li>
        <li><a href="plantillas/listas.html">Listas</a></li>
        <li><a href="plantillas/tablas.html">Tablas</a></li>
        <li><a href="plantillas/elementos_control.html">Elementos de Control</a></li>
        <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown"
        href="#">Adornos<span class="caret"></span></a>
```

```

        <ul class="dropdown-menu emergente">
          <li><a href="#">Glyphicons</a></li>
          <li><a href="#">Jumbotron</a></li>
          <li><a href="#">Carrusel</a></li>
          <li><a href="#">Otros</a></li>
        </ul>
      </li>
    </ul>
  </div>
</nav>
</section>

```

Se crea un botón que aparecerá cuando se colapse el menú y se lo coloca un glyphicon con la imagen representativa de menú responsivo.

Se encierra el contenido del menú dentro de las clases

```
navbar-collapse collapse menu_p
```

En donde menu_p está referenciado dentro del botón para mostrarse u ocultarse.

```
data-target=".menu_p
```

Finalmente se da estilo al glyphicon para que resalte.

general.css

```

.logo_menu{
  color:#FFFFFF;
  font-size:25px;
}

```

Listas

El manejo de listas de forma responsiva es muy simple, ya que se adaptan a su contenedor de manera automática.

Elementos de Control

Lo mismo sucede con los elementos de control, al estar colocados dentro de contenedores especificados por Bootstrap, el comportamiento responsivo se logra de manera automática.

Tablas

El manejo que *Bootstrap* ofrece para un comportamiento responsivo de las tablas consiste en colocar un *scroll* horizontal dentro de la tabla para poder recorrerla. Para esto, se debe encerrar a la tabla completo dentro de un contenedor de la clase **table-responsive**

tablas.html

```
<div class="table-responsive">

<table class="table tabla_sencilla">
  <thead>
    <tr>
      <th>Columna 1</th>
      <th>Columna 2</th>
      <th>Columna 3</th>
      <th>Columna 4</th>
      <th>Columna 5</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>DATO 1_1</td>
      <td>DATO 1_2</td>
      <td>DATO 1_3</td>
      <td>DATO 1_4</td>
      <td>DATO 1_5</td>
    </tr>
    <tr>
      <td>DATO 2_1</td>
      <td>DATO 2_2</td>
      <td>DATO 2_3</td>
      <td>DATO 2_4</td>
      <td>DATO 2_5</td>
    </tr>
    <tr>
      <td>DATO 3_1</td>
      <td>DATO 3_2</td>
      <td>DATO 3_3</td>
      <td>DATO 3_4</td>
      <td>DATO 3_5</td>
    </tr>
    <tr>
      <td>DATO 4_1</td>
      <td>DATO 4_2</td>
      <td>DATO 4_3</td>
      <td>DATO 4_4</td>
      <td>DATO 4_5</td>
    </tr>
    <tr>
      <td>DATO 5_1</td>
      <td>DATO 5_2</td>
      <td>DATO 5_3</td>
      <td>DATO 5_4</td>
      <td>DATO 5_5</td>
    </tr>
  </tbody>
</table>

</div>
```

Textos e Imágenes

El manejo responsivo de los textos es muy simple, se adaptan a su contenedor por lo que el control que ofrece Bootstrap sobre ellos es más que suficiente.

Las imágenes tienen el problema de que si tienen un tamaño fijo, puede llegar el momento en que se desborden.

Bootstrap ofrece la clase **img-responsive** la cuál se agrega al elemento `` para tener un comportamiento responsivo. Para que funcione correctamente se debe eliminar cualquier tamaño especificado en estilos de la imagen o de su contenedor, en este caso `<figure>`.

textos.html

```
<figure class="flotada_izquierda pull-left">
  
  <figcaption>Una imagen flotada a la izquierda con Bootstrap</figcaption>
</figure>
```

Mientras que del archivo de estilos se eliminó la propiedad de ancho

textos_imagenes.html

```
.bloque_izquierdo_2 figure{
  margin-bottom: 10px;
}
```

Sin embargo esta modificación no funciona cuando los elementos están flotados, por lo que será necesario hacer cambios usando otro manejo de propiedades responsivas. Al quitar el ancho del contenedor `<figure>` también se pierde la propiedad de centrado.

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.

Corrigiendo el Footer

Si bien el Footer ya tiene un comportamiento responsivo, cuando se reduce la resolución, llega el momento en que los elementos se desbordan, esto se debe a que se especificó un alto fijo. Se creará una sentencia con *media query* para ajustar este ancho a ciertas resoluciones.

Si bien las *media queries* se pueden declarar por separado, por orden es conveniente que se coloquen en la misma hoja de estilo en donde están los elementos a los que se quiere afectar.

Se observa que para una resolución poco menor a 500px ya se presenta un desborde, por lo que la *media query* se aplicará para resoluciones menores o iguales a 500px

general.css

```
@media screen and (max-width: 500px){  
  .footer_sitio{  
    height:auto;  
  }  
}
```

Porcentajes y Desborde de Elementos

La principal causa de los desbordamientos es el ancho de los elementos, para un manejo responsivo adecuado se debe manejar su ancho en porcentaje más que en píxeles, el porcentaje de ancho se considera en base a su contenedor inmediato superior.

En Bootstrap, por el manejo de rejillas, no se manejan anchos en píxeles, por lo que no se manejarán cambios en porcentaje de ancho de los elementos.

Tamaño de las Fuentes

Si bien los textos tienen un comportamiento responsivo adecuado, el tamaño de la letra se mantiene, por lo que en resoluciones pequeñas se aprecia que es muy grande. Para modificar esto se puede realizar una sentencia como la anterior, pero solo afectará al texto indicado, también se puede modificar la letra según un tamaño base. Esto se logra expresando el tamaño en **em** y no **píxeles**.

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 `*`.

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**.

No se había especificado un tamaño base, esto se hará en el estilo por defecto:

general.css

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

Esto cambia el tamaño de la fuente de algunos elementos, por ejemplo el menú y del párrafo del bloque lateral. Para corregirlo se realizará lo siguiente:

menu_navegacion.css

```
.encabezado_menu a{
color:#FFFFFF;
font-size: 1em;
}

.elementos_menu li a{
color:#FFFFFF;
font-size: 0.8em;
}
```


El encabezado del menú tiene un tamaño de letra de **1em**, es decir de **20px** (tamaño base), mientras que el menú tiene un tamaño de letra de **0.8em** respecto al tamaño base, lo que sería equivalente a 16px (20 x 0.8). De esta manera se puede ajustar el tamaño de una fuente dependiendo la resolución.

También se cambiará el tamaño de letra del párrafo lateral.

general.css

```
.bloque_lateral_2 p{
  font-size: 0.8em;
}
```

Cambiando Tamaño de un Texto según la Resolución

Se le cambiará el tamaño a un texto dependiendo la resolución, en este caso al encabezado (h1) de la plantilla de textos

textos_imagenes.css

```
@media screen and (max-width:600px){
  .bloque_izquierdo_2 h1{
    font-size:20px;
  }
}
```

En este caso solo se agrega una propiedad (tamaño de letra), la propiedad de color se “hereda” de la declaración inicial y no se desea modificar, por lo que no es necesario incluirla.

Combinando las *media queries* y el uso de **em**, se podría cambiar el tamaño a una gran cantidad de párrafos, solo sería necesario cambiar el tamaño de fuente base dependiendo la resolución.

Modificando estilos de Centrado y Flotado

Uno de los problemas con volver responsivas las imágenes es que se quitaban algunos efectos anteriores, específicamente el de centrado. Se había eliminado la propiedad de ancho del contenedor.

textos_imagenes.css

```
.bloque_izquierdo_2 figure{
  margin-bottom: 10px;
}
```

También se agregó la clase de imagen responsiva a las imágenes.

textos_imagenes.css

```
<figure class="center-block">
  
  <figcaption>Una imagen al azar de Lorem Pixel Centrada</figcaption>
</figure>
```

Eso les daba un comportamiento responsivo, pero en el caso de la imagen centrada con una de las clases de Bootstrap, ésta dejaba de estarlo.

Para solucionarlo, se realizarán las siguientes modificaciones:

textos_imagenes.css

```
.centrada{
  text-align: center;
}
.centrada img{
  margin:auto;
}
```

Modificando la estructura del Sitio

Es posible modificar la estructura del sitio web conforme cambia la resolución.

Lo que se hará es cambiar el orden en el que aparece el cuerpo principal y el bloque lateral para resoluciones pequeñas o menores. En el código original, aparece primero el contenedor de 9 columnas (bloque principal) y después el de 3 columnas (bloque lateral).

Para cambiar el orden, éstos se invertirán (considerar que se han colapsado para un manejo más sencillo).

index.html

```
<section class="col-lg-3 bloque_lateral_2"> </section>
<section class="col-lg-9 bloque_izquierdo_2"> </section>
```

Pero esto cambia el orden en cualquier resolución, lo que se debe hacer es “empujar” y “jalar” los elementos para la resolución menor en la que se quiere mantener esta organización, en este caso la mediana MD

index.html

```
<section class="col-lg-3 col-md-3 col-md-push-9 bloque_lateral_2">
</section>
<section class="col-lg-9 col-md-9 col-md-pull-3 bloque_izquierdo_2">
</section>
```

En este caso se “empuja” 9 columnas el bloque lateral y se “jala” 3 columnas el bloque principal.

El bloque lateral todavía mantiene el borde que se especificó al inicio, por lo que se eliminará para resoluciones menores a la media (MD)

general.css

```
@media screen and (max-width: 991px){
  .bloque_lateral_2{
    border: 0px solid #640923;
  }
}
```