

TEXT CLASS REVIEW

TEMAS A TRATAR EN LA CUE

- Layout, contenidos y componentes de Bootstrap.
- Utilizar componentes JavaScript para dar interactividad.

LAYOUT

Como se mencionó en el CUE anterior, Bootstrap utiliza un sistema de 12 grillas, el cual permite ordenar y posicionar de manera sencilla y eficaz, los elementos HTML en nuestros proyectos web. Además, dicho sistema está diseñado utilizando la filosofía mobile-first, y construido sobre flexbox, dando la posibilidad de desarrollar páginas y aplicaciones web responsivas de manera rápida, con estilos dependientes del tamaño de la pantalla en el que se visualice el contenido.

Además, es importante señalar que, al utilizarse Sass en el desarrollo del código fuente de este framework, los estilos siguen y reconocen todas las directrices y propiedades de CSS que hemos revisado durante este DRILL, como son las **media queries** y el **z-index**, los cuales podemos personalizar sin problemas desde los archivos parciales de Bootstrap.

CONTENEDORES

O **containers**, corresponden al elemento más básico en el layout de Bootstrap, y se necesitan al utilizar el sistema de grillas. Recordemos que éstos se pueden anidar (posicionar uno dentro de otro), existiendo 12 grillas dentro de cada uno. Es decir, si un contenedor B se encuentra dentro de un contenedor A, utilizando 6 grillas de éste (A), tendremos 6 grillas del contenedor A disponibles y, a su vez, 12 grillas disponibles dentro del contenedor B, las cuales, al mismo tiempo, son equivalentes en ancho a 6 grillas de A.

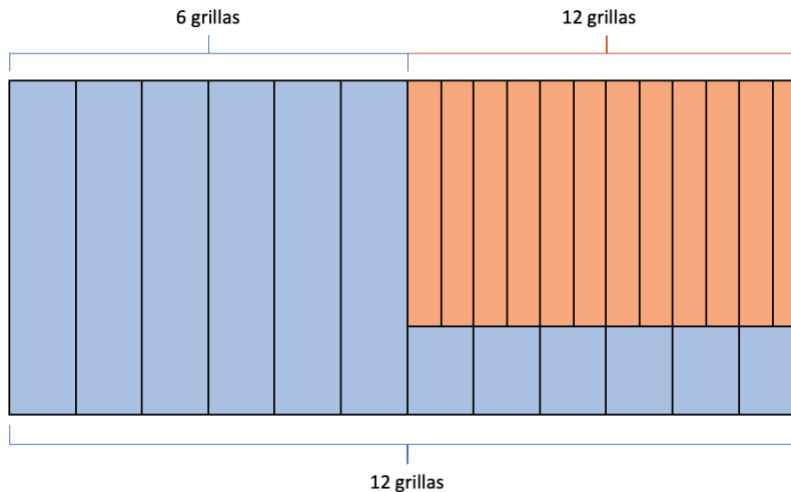


Imagen 1. Representación del sistema de grillas en contenedores anidados.

CONTENIDO

Como ya hemos podido revisar, Bootstrap es un framework muy completo, pues cubre a cabalidad todo lo relacionado a estética de los elementos de HTML, que usamos comúnmente en el desarrollo de nuestros proyectos web.

Algunas propiedades, como la familia de fuente (**font-family**), utilizada en los elementos de texto, están optimizadas para los distintos navegadores y dispositivos, lo cual significa un gran beneficio para la eficiencia de despliegue, y la amigabilidad visual de nuestros proyectos con el usuario. También, existen muchas otras propiedades que están cubiertas en las clases definidas por Bootstrap, como es: el tamaño de la fuente de los encabezados, el color de la fuente, y el alineamiento de los textos. Si desear profundizar más al respecto, te recomendamos visitar la [documentación de Bootstrap](#).

Otros elementos HTML, como: listas, segmentos de código, imágenes, tablas y figuras, también están cubiertas por este framework, lo que constituye una gran ayuda al momento de realizar páginas web responsivas, permitiendo el ajuste de tamaño de los elementos según el **viewport**, alinear los elementos en el espacio, y modificar otras variables, como por ejemplo, el color de fondo de una tabla, con tan solo agregar una clase a la etiqueta correspondiente.

COMPONENTES

Una de las propiedades más distintivas de Bootstrap, es la cantidad de componentes predefinidos que nos ofrece. Muchos de éstos requieren de JavaScript y jQuery para funcionar, por lo que es muy importante agregar los scripts correspondientes, desde el código de fuente que descargamos en el CUE anterior, directamente de la página de Bootstrap.

La información más extensa y completa sobre cada uno de los componentes de Bootstrap, se encuentra en detalle en la [documentación](#) del framework.

COMPONENTES JAVASCRIPT

Como se indicó en el punto anterior, varios de los componentes de Bootstrap, como los modales, las alertas, los carruseles, entre otros, requieren de JavaScript y jQuery, los cuales le entregan funcionalidad a dichos componentes.

Existen dos maneras de configurarlos:

1. Utilizando atributos de datos. Este método no requiere modificar directamente los scripts de nuestro proyecto, sino que se debe agregar la propiedad a configurar, y su valor directamente en la etiqueta HTML.

Revisemos un ejemplo, en el siguiente fragmento de código se muestra un carrusel (**carousel**), ahora, nos enfocaremos en las flechas que permitirán volver a la imagen previa y avanzar a la siguiente. Si revisamos el contenido de las etiquetas **<a>**, notaremos que ambas tienen el atributo de datos **"data-slide"**, el cual da la posibilidad de configurar si al hacer clic sobre este elemento el carrusel volverá a la **slide** previa, o avanzará a la siguiente.

```
1 <div id="carouselExampleIndicators" class="carousel slide"
2 data-ride="carousel">
3   <ol class="carousel-indicators">
4     <li data-target="#carouselExampleIndicators" data-slide-
5 to="0" class="active"></li>
6     <li data-target="#carouselExampleIndicators" data-slide-
7 to="1"></li>
8     <li data-target="#carouselExampleIndicators" data-slide-
9 to="2"></li>
10  </ol>
11  <div class="carousel-inner">
12    <div class="carousel-item active">
13      
15    </div>
```

```

16     <div class="carousel-item">
17         
19     </div>
20     <div class="carousel-item">
21         
23     </div>
24 </div>
25 <a class="carousel-control-prev"
26 href="#carouselExampleIndicators" role="button" data-
27 slide="prev">
28     <span class="carousel-control-prev-icon" aria-
29 hidden="true"></span>
30     <span class="sr-only">Previous</span>
31 </a>
32 <a class="carousel-control-next"
33 href="#carouselExampleIndicators" role="button" data-
34 slide="next">
35     <span class="carousel-control-next-icon" aria-
36 hidden="true"></span>
37     <span class="sr-only">Next</span>
38 </a>
39 </div>
  
```

- Utilizando jQuery. La otra manera de configurar los componentes es utilizando sentencias de jQuery, en este caso, es necesario emplear el selector correspondiente, y un objeto que contenga las opciones indicadas para dichos componentes.

Volvamos a revisar el ejemplo anterior, pero esta vez estableceremos que el carrusel no reaccionará a eventos del teclado, como presionar la flecha izquierda o derecha. Para esto, solo es necesario agregar el siguiente código de jQuery.

```

1 $('.carousel').carousel({
2     keyboard: false
3 })
  
```