



Bootstrap es un framework de CSS que nos facilita y estandariza el desarrollo de sitios web.

Esta pensando para que se adapte tanto a las pantallas de equipos de escritorio como a móviles y tablets.

Bootstrap es desarrollado y mantenido por la empresa Twitter y la ha liberado como un producto Open Source.

Tiene una filosofía muy intuitiva para el maquetado de sitios web que puede ser rápidamente aprendida por desarrolladores que no vienen del mundo del diseño web.

1. Emplear un CDN para cargar Bootstrap 4
2. Sistema de grillas de Bootstrap 4
3. Múltiples reglas a las columnas de una fila
4. Contenedor: clases container y container-fluid
5. Columnas de diseño automático
6. Alineamiento vertical de las columnas
7. Alineamiento horizontal de las columnas
8. Varias líneas en una misma fila "row"
9. Reordenar las columnas en una fila
10. Anidamiento de columnas
11. Offset (desplazamiento) de columnas
12. Table
13. Form
14. Button
15. Tipografía
16. Imágenes
17. Componente : jumbotron
18. Componente : card
19. Componente : alert
20. Componente : modal
21. Componente : navbar
22. Componente : carousel
23. Componente : pagination
24. Componente : progress
25. Componente : dropdown (lista desplegable)
26. Componente : collapse
27. Componente : accordion
28. Componente : breadcrumb
29. Componente : badge
30. Componente : nav
31. Componente : Scrollspy
32. Componente : Tooltips
33. Componente : Popover

Emplear un CDN para cargar Bootstrap 4

Si vamos a utilizar un CDN (Content Delivery Network - red de entrega de contenidos) luego nuestro esqueleto de página html tendrá una estructura similar a esta:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
      integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
      crossorigin="anonymous">
```

Las ventajas de utilizar esta forma de cargar el framework Bootstrap 4 es:

- Reduce la carga de nuestros servidores (importante si utilizamos un hosting compartido con recursos limitado)
- Facilita que Bootstrap 4 quede en caché del navegador y se vuelva a cargar cada vez que se visita una web que lo utilice.
- Reduce la latencia.
- Reducción de costos de nuestros servidores por requerir menor ancho de banda y tráfico.

Más adelante veremos componentes de Bootstrap 4 que si las utilizamos debemos importar una serie de archivos Javascript.

1. jquery-3.2.1.slim.min.js
2. popper.min.js
3. bootstrap.min.js

Sistema de grillas de Bootstrap 4

Lo más importante de entender en un principio es el sistema de grillas que plantea la librería Bootstrap.

Debemos pensar para armar nuestro esquema de la página que tenemos la posibilidad de definir filas y en cada fila definir de 1 hasta 12 columnas. Cada columna con un ancho relativo a ese número 12.

Luego Bootstrap se encarga de colapsar las columnas cuando se accede al sitio desde un dispositivo con una capacidad limitada en cuanto al ancho en píxeles (esto permite una experiencia más placentera al visitante que accede a nuestra página desde un celular con capacidades limitadas)

Debemos utilizar la clase "row" para indicar el comienzo de una fila. Dentro de la fila dispondremos tantos div como columnas tenga dicha fila. Para indicar cada columna debemos utilizar la siguiente sintaxis para cada columna:

```
col-xl-*
```

Donde aparece el asterisco lo remplazamos por un valor entre 1 y 12 (en nuestro ejemplo hemos dispuesto un 4, esto quiere decir que quedan 8 unidades de columna para repartir entre las otras columnas)

```
<div class="row">
  <div class="col-xl-4"
style="background-color:#aaa">
    <h1>Columna 1</h1>
    <p>Esto es una prueba de
bootstrap.</p>
  </div>
  <div class="col-xl-4"
style="background-color:#bbb">
    <h1>Columna 2</h1>
    <p>Esto es una prueba de
bootstrap.</p>
  </div>
  <div class="col-xl-4"
style="background-color:#ccc">
    <h1>Columna 3</h1>
    <p>Esto es una prueba de
bootstrap.</p>
  </div>
</div>
```

Pero ahora nos preguntamos porque la segunda fila que hemos definido no colapsa cuando el ancho del navegador baja de 1200px, esto es debido que hemos definido cada columna con la siguiente sintaxis:

```
<div class="row">
  <div class="col-lg-4"
style="background-color:#aaa">
    <h1>Columna 1</h1>
    <p>Esto es una prueba de
bootstrap.</p>
  </div>
  <div class="col-lg-4"
style="background-color:#bbb">
    <h1>Columna 2</h1>
    <p>Esto es una prueba de
bootstrap.</p>
  </div>
  <div class="col-lg-4"
style="background-color:#ccc">
    <h1>Columna 3</h1>
    <p>Esto es una prueba de
bootstrap.</p>
  </div>
</div>
```

Cuando utilizamos las clases "col-lg-*" luego la fila colapsará cuando el ancho del navegador se reduzca a menos de 992px, podemos probarlo fácilmente reduciendo más el ancho del navegador y ver que ahora colapsan tanto la primera fila como la segunda fila:

col-lg-4

Lo mismo sucederá cuando utilicemos "col-md-*" y reduzcamos los 720px:

La cuarta fila utilizamos "col-sm-*" y colapsa al llegar a un valor menor de 540px.

Por último hay que decir que cuando utilizamos las clases "col-*" nunca colapsarán las columnas.

Es bueno analizar de que palabras vienen los nombres de reglas de estilo que provee Bootstrap:

col-*	
col-sm-*	(s)m(a)ll
col-md-*	(m)e(d)i(u)m
col-lg-*	(l)a(r)g(e)
col-xl-*	(e)xtra (l)a(r)ge

Múltiples reglas a las columnas de una fila

Si queremos que una página se muestre de forma diferente si la accedemos desde un dispositivo con pantalla muy grande, grande, mediana, pequeña o muy pequeña podemos asignar múltiples reglas a cada columna.

Implementemos una página que muestre los datos en dos columnas y el ancho de las columnas dependa de la pantalla del dispositivo:

	1er columna	2da columna
Pantalla muy grande:	10 unidades	2 unidades
Pantalla grande:	9 unidades	2 unidades
Pantalla mediana:	8 unidades	4 unidades
Pantalla chica:	7 unidades	5 unidades
Pantalla muy chica:	6 unidades	6 unidades

Es decir podemos asignar múltiples reglas a un div de una columna, actuará el que corresponde dependiendo del ancho del dispositivo. Por ejemplo si ejecutamos y comenzamos a reducir el ancho del navegador podremos ver cómo se redimensionan los anchos de las columnas a medida que reducimos el ancho del navegador.

En un dispositivo muy grande ≥ 1200 :

En un dispositivo grande ≥ 992 :

En un dispositivo mediano ≥ 768 :

En un dispositivo pequeño ≥ 576 :

Y en un dispositivo muy pequeño ≤ 576 :

En este caso tengamos en cuenta que nunca colapsan las columnas ya que hemos definido la regla `col-*`

La definición de múltiples reglas es una de las herramientas más poderosas de Bootstrap, nos permite definir distintos diseños de página según las capacidades del dispositivo del visitante a nuestro sitio web.

Contenedor: clases container y container-fluid

Cuando implementamos un contenedor podemos utilizar un:

Ancho fijo centrado (Siempre hemos utilizado este estilo hasta ahora)

Ancho variable que se adapte al ancho del navegador.

Vamos a implementar un página que muestre un contenedor de ancho fijo con una fila de tres columnas, y otro contenedor pero fluido que muestre el mismo contenido.

Columnas de diseño automático

Vimos en conceptos anteriores que en Bootstrap disponemos de 12 unidades a ser repartidas en una o más columnas, además de indicar a que tamaño de dispositivo está orientado.

Si todas las columnas tendrán el mismo ancho podemos evitar disponer el valor que indicamos donde hay un asterisco.

```
col-*  
col-sm-*  
col-md-*  
col-lg-*  
col-xl-*
```

Definición con un valor fijo para una columna.

Una variante al concepto anterior es la posibilidad de definir una columna con un ancho en unidades fijo y la columna de la izquierda o derecha o ambas sin valor:

Definición con un valor automático para una columna.

Si necesitamos que se repartan los espacios teniendo en cuenta la cantidad de contenido de una columna podemos utilizar las clases col-auto, col-sm-auto, col-md-auto, col-lg-auto y col-xl-auto.

Alineamiento vertical de las columnas

Bootstrap 4 trae como ventaja con respecto a versiones anteriores (3.x, 2.x etc.) la posibilidad de alinear las columnas en forma vertical. Esta facilidad esta presente ya que Bootstrap 4 utiliza todas las reglas de estilo que provee Flexbox CSS.

Alineación de columnas individuales.

Podemos alinear solo una columna fijando alguna de estas clases: "align-self-start", "align-self-center" o "align-self-end" a un div que representa la columna.

Alineamiento horizontal de las columnas

Esta posibilidad de alinear las columnas se puede presentar si no utilizamos las 12 unidades a distribuir.

Si la suma de unidades es igual a 12 luego no podemos alinear horizontalmente las columnas ya que se encuentra toda la fila ocupada.

Varias líneas en una misma fila “row”

Con Bootstrap 4 se ha agregado la posibilidad de que en una fila definida por la clase "row" especifique varias columnas y mediante la clase "w-100" permitir hacer un salto de línea.

Anidamiento de columnas

Otro tema perfectamente resuelto por Bootstrap 4 es el anidamiento de columnas.

Vamos a Implementar una página que muestre dos columnas y dentro de la segunda columna otras dos columnas internas.

Offset (desplazamiento) de columnas

Podemos dejar columnas vacías indicando un desplazamiento en el momento de crearla.

table : básica

El elemento table de HTML nos permite mostrar datos en forma tabular. Cuando incluimos el framework Bootstrap 4 se introducen una serie de reglas a los elementos relacionados a la etiqueta "table".

form : controles input (text, password, file)

Bootstrap 4 requiere una serie de clases para la implementación de un formulario vertical.

Veamos la etiqueta input como es afectada por Bootstrap. Lo más común es encerrar el control input y label con un div que se inicializa con la clase "form-group".

form : control textarea

Bootstrap 4 requiere una serie de clases para la implementación de un formulario vertical.

Veamos la etiqueta input como es afectada por Bootstrap. Lo más común es encerrar el control input y label con un div que se inicializa con la clase “form-group”.

form : controles checkbox y radio

Veamos como trabajamos con otros dos controles muy usados en los formulario web como son los controles checkbox y radio.

Podemos hacer que los controles aparezcan uno debajo del otro o uno al lado del otro.

form : control select

Para aplicar Bootstrap 4 a un control select debemos fijar la clase "form-control" al select respectivo y también debemos encerrarlo con un div con la clase "form-group".

La sintaxis de clases aplicadas es similar a las etiquetas "input" que vimos anteriormente.

form : formulario horizontal

Otra variante de formulario que nos permite Bootstrap 4 son los formularios horizontales. En este estilo de formularios se dispone la descripción del dato a ingresar en el lado izquierdo y el control del formulario a derecha.

Para resolver este tipo de problemas debemos utilizar el sistema de grillas de Bootstrap.

form : formulario en una línea

Hasta ahora hemos visto que todos los controles se ubican uno debajo de otro (que será lo más común), pero hay situaciones que podemos necesitar que haya varios controles de distinto tipo en la misma línea, en estos casos Bootstrap 4 nos proporciona una clase que debemos asignársela a la marca form llamada "form-inline".

form : tamaño de los controles de formulario

Bootstrap 4 nos permite definir 3 tamaños de controles de formulario. Tenemos el tamaño por defecto, un tamaño grande y un tamaño pequeño.

form : texto de ayuda de un control

Una característica muy común que se agregan a los formularios HTML es un texto en la parte inferior del control informando al usuario que datos debe ingresar con la mayor precisión posible.

form : control y grupo de controles HTML deshabilitados

Para deshabilitar un único control de formulario HTML debemos insertar la propiedad disabled al control respectivo.

En el caso que necesitemos deshabilitar un conjunto de controles agrupados en una etiqueta HTML fieldset le asignamos la propiedad disabled a dicha marca.

button : estilos

Para crear botones Bootstrap 4 propone utilizar la etiqueta "button", "a" o "input" y define una serie de clases:

```
btn
btn-primary
btn-secondary
btn-success
btn-info
btn-warning
btn-danger
btn-dark
btn-light
btn-link
```

Cada una de estas clases define un estilo particular para un botón y tiene un significado que nos ayudan a definir en que caso los debemos utilizar:

- `btn-primary` : Es un botón que se destaca entre un conjunto de botones.
- `btn-secondary` : Es un botón no tan importante como `btn-primary`.
- `btn-success` : Se utiliza para indicar una acción exitosa (por ejemplo luego de registrarse a un servicio donde se muestra un botón para ir a la página principal del sitio)
- `btn-info` : Es un botón para información.
- `btn-warning` : Es un botón que nos informa que debemos tener cuidado con la acción que tiene asociado el botón.
- `btn-danger` : Indica que la acción que tiene asociado el botón es peligrosa.
- `btn-dark` : Muestra un botón oscuro.
- `btn-light` : Muestra un botón claro.
- `btn-link` : Convierte al botón como un hipervínculo, haciendo que disminuya su importancia.

button : tamaño

Hemos visto que existen distintos tipos de botones que debemos utilizar según la circunstancia. A estos estilos de botones podemos utilizar dos clases más de tamaños aparte del estándar :

- btn-sm : Define un tamaño de botón pequeño.
- btn-lg : Define un tamaño de botón grande.

Indistintamente del tamaño del botón disponemos de una clase btn-block que expande a todo el ancho del contenedor.

button : desactivo

Si queremos que un elemento HTML button se muestre y actúe como desactivo debemos inicializar la propiedad disabled con el valor disabled.

button : grupo de botones, barra de herramientas, tamaños y orientación

Bootstrap nos provee la posibilidad de agrupar un conjunto de botones mediante la clase "btn-group". Los botones se agrupan pegados en la misma fila.

En algunas situaciones podemos agrupar un conjunto de grupos de botones y crear una barra de herramientas.

FIN

Por hoy