

# Apuntes de Bootstrap

migueluisV

Realizadas: Mayo 2023

# Índice

<b>1</b>	<b>Conceptos básicos</b>	<b>6</b>
<b>2</b>	<b>Agregando Bootstrap a un proyecto web</b>	<b>6</b>
<b>3</b>	<b>Los pilares de Bootstrap</b>	<b>7</b>
3.1	Grid . . . . .	7
3.1.1	Clases para la Grid . . . . .	9
3.1.2	Breakpoints . . . . .	10
3.1.3	Estructura . . . . .	12
3.1.4	Contenedores . . . . .	12
3.1.5	Contenedores responsivos . . . . .	14
3.1.6	Trabajando con las filas y columnas . . . . .	16
3.2	Componentes . . . . .	17
3.3	Iconos . . . . .	17
3.4	Flexbox . . . . .	17
3.4.1	Propiedades del Flexbox en Bootstrap . . . . .	18

## Índice de Figuras

1	Versión de Bootstrap para agregarlo a un proyecto web . . . . .	7
2	Columnas en una fila del Grid . . . . .	8
3	Tamaño de un elemento según $n$ columnas de una fila del Grid . . . . .	8
4	Estilo de la Grid en panel uniforme . . . . .	9
5	Estilo de la Grid en filas no uniformes . . . . .	9
6	Estilo de la Grid en columnas no uniformes . . . . .	9
7	Aspecto original previo a cambio por breakpoints en sitio real . . . . .	11
8	Aspecto nuevo posterior al cambio por breakpoints en sitio real . . . . .	11
9	Uso de las clases de filas y columnas de Bootstrap . . . . .	12
10	Aspecto original de la clase container . . . . .	13
11	Aspecto original de la clase container-fluid . . . . .	13
12	Nuevo aspecto de la clase container . . . . .	14
13	Nuevo aspecto de la clase container-fluid . . . . .	14
14	Aspecto original de la clase container-lg . . . . .	15
15	Nuevo aspecto de la clase container-lg . . . . .	15
16	Caption . . . . .	16
17	Aspecto original de los contenedores responsivos . . . . .	16
18	Nuevo aspecto de los contenedores responsivos . . . . .	17

## Índice de Tablas

1	Ejemplos de clases de la Grid . . . . .	10
2	Propiedades de Flexbox . . . . .	18

Este documento se hizo con **Overleaf** y los ejemplos fueron desarrollados y probados con **Visual Studio Code**, con su extensión **Live Server** en el buscador **Microsoft Edge** o **Brave**.

La estructura HTML de los ejemplos en este documento serán omitidos, dejando únicamente lo vital para que los ejemplos funcionen y para evitar que este trabajo sea muy largo, los ejemplos CSS y Bootstrap si vendrán completos.

# 1 Conceptos básicos

## Bootstrap

Se trata de un *framework* de CSS para crear sitios y páginas web *responsivos*.

### Framework y Framework CSS

Un Framework es una base o plataforma de un cierto lenguaje de programación para ahorrar a otros desarrolladores el escribir código. Estos frameworks suelen tener contenido, módulos, clases o funciones que resuelven problemas comunes, proporcionando este código para que, quien utilice el framework, tarde menos en desarrollar una implementación.

Un framework CSS es una base con *clases*, *identificadores* y estilos predeterminados que permiten crear páginas web bien distribuidas y responsivas para los distintos tamaños de pantalla disponibles.

### Páginas y sitios web *responsivos*

La palabra responsivo viene de la palabra en inglés *responsive*, estas página o sitios web pueden ser llamados también adaptables. Estas páginas son responsivas cuando el tamaño y distribución de la página se adapta a la pantalla del dispositivo que la visualiza, siendo una tableta, celular o computadora.

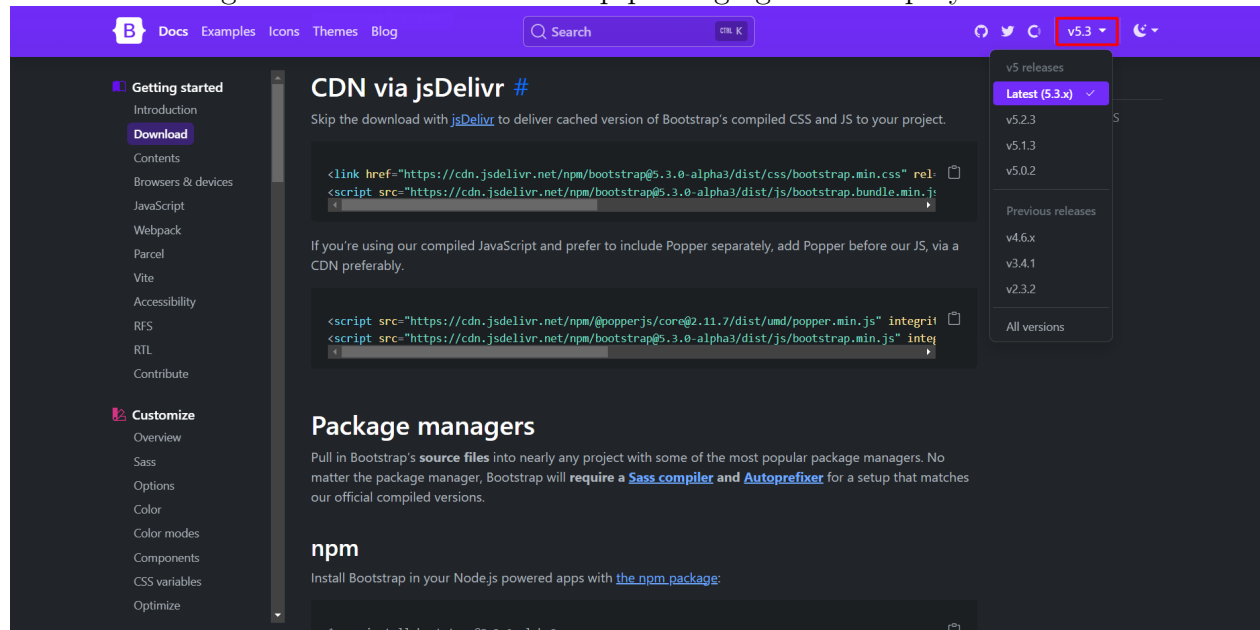
# 2 Agregando Bootstrap a un proyecto web

Esta herramienta no necesita de una instalación en particular, simplemente agregamos una etiqueta al *head* de nuestro proyecto de la siguiente manera:

```
<!DOCTYPE html>
<html lang='es'>
  <head>
    <!-- Bootstrap. -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJ0Z" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNs7NTKfAdvQSZ" crossorigin="anonymous"></script>
  </head>
  <body>
    <!-- Contenido. -->
  </body>
</html>
```

Se agrega las etiquetas *link* y *script* a la cabecera de nuestro proyecto, estas líneas fueron obtenidas de la [documentación oficial](#) de Bootstrap, en la sección de Download en la sección **CND via jsDelivr**. Debe fijarse bien qué versión va a estar implementando a su proyecto, en nuestro caso es la versión 5.3, puede revisar qué versión tomará en la esquina superior derecha de la página, como se ve en la Figura 1.

Figure 1: Versión de Bootstrap para agregarlo a un proyecto web



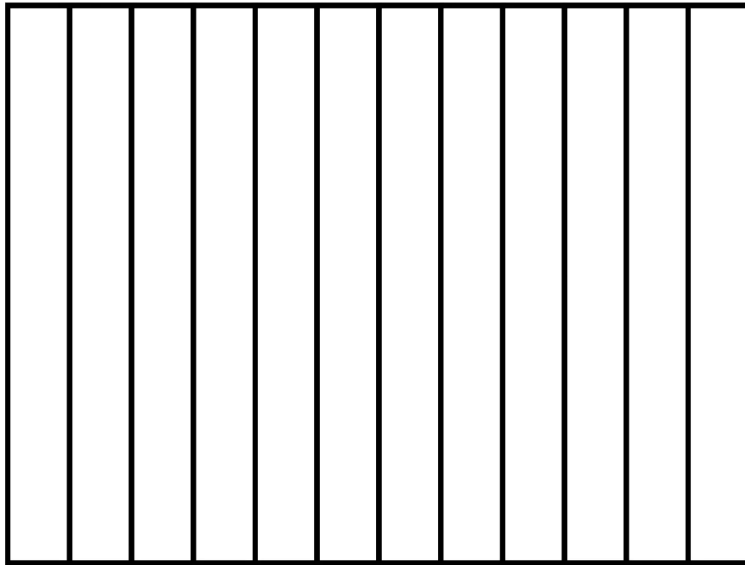
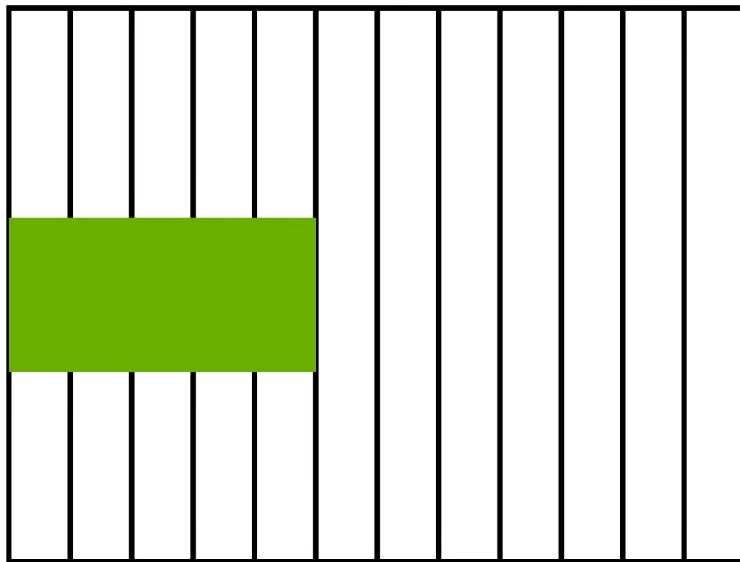
## 3 Los pilares de Bootstrap

### 3.1 Grid

La Grid (*cuadrícula* en español) define como se estructura los elementos de una página y como se adaptan a un tamaño de pantalla. El Grid se puede definir también como un conjunto de filas, columnas y contenedores que definen como se van a presentar y alinear los elementos de una página web.

Una fila del Grid se divide en 12 secciones o columnas, un elemento HTML o elemento Bootstrap puede abarcar la cantidad de secciones que este requiera, pudiendo ser 3, 5, 7 o las 12 en total, como se ve en la Figura 2 y 3.

Figure 2: Columnas en una fila del Grid

Figure 3: Tamaño de un elemento según  $n$  columnas de una fila del Grid

Además, existen estilos predeterminados para distribuir y alinear los elementos de una página, como se ven en las Figuras 4, 5 y 6.



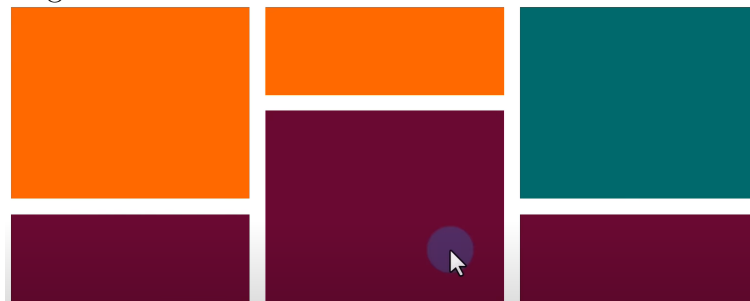
Figure 4: Estilo de la Grid en panel uniforme



Figure 5: Estilo de la Grid en filas no uniformes



Figure 6: Estilo de la Grid en columnas no uniformes



Estos estilos se adaptan a las necesidades del desarrollador por mostrar el contenido de una u otra manera.

### 3.1.1 Clases para la Grid

Entonces, para utilizar la Grid dentro del desarrollo de una página web, es necesario recurrir a las clases de la Grid, las cuales están presentes en la Tabla 1.

Table 1: Ejemplos de clases de la Grid

Nombre	Definición	Dimensión
.row	Permite que un elemento HTML o Bootstrap se acomode como una fila, pudiendo abarcar cierta cantidad de secciones, columnas o un ancho de dicha fila. Es aquí donde entran las siguientes clases.	El ancho de la ventana que lo visualiza o el contenedor que lo almacena
.col	Distribuye los elementos de una fila como columnas con un espacio definido o automático	El ancho de la ventana que lo visualiza o el contenedor que lo almacena
.col-xs	<i>xs</i> corresponde a <b>extra small</b> , es el tamaño más pequeño para una columna o elemento	< 576px
.col-sm-	<i>sm</i> corresponde a <b>small</b> , esto define un tamaño pequeño para una columna o elemento	>= 576px
.col-md-	<i>md</i> corresponde a <b>medium</b> , define un tamaño medio para una columna o elemento	>= 768px
.col-lg-	<i>lg</i> corresponde a <b>large</b> , define un tamaño grande para una columna o elemento	>= 992px
.col-xl-	<i>xl</i> corresponde a <b>extra large</b> , esto define un tamaño mayor a lg	>= 1200px
.col-xxl-	<i>xxl</i> corresponde a <b>extra extra large</b> , es el tamaño más grande para una columna o elemento	>= 1400px

Estas dimensiones representan el ancho de la ventana o pantalla del dispositivo donde se está visualizando la página, llamada también *viewport*.

### 3.1.2 Breakpoints

Un punto de quiebre o *breakpoint* es un ancho o dimensión determinado en el cual se puede cambiar la distribución, alineación o estilo de la página, es decir, si la resolución de la pantalla del dispositivo que está viendo la página web pasa de 700px a 465px, un breakpoint podrían ser los 500px, al bajar de esta resolución, los elementos se distribuyen de otra manera. Los breakpoints más comunes de Bootstrap 5 son las dimensiones en píxeles escritas en la tabla anterior, si utiliza una versión anterior, consulte los endpoints existentes en dicha versión.

Estos cambios de distribución o estilo de una página al pasar un breakpoint se puede apreciar en la Figura 7 y 8 respectivamente.

Figure 7: Aspecto original previo a cambio por breakpoints en sitio real

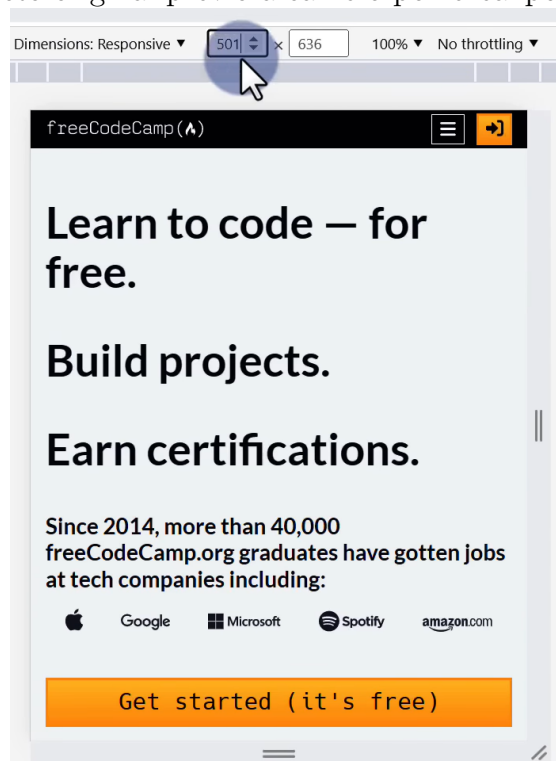
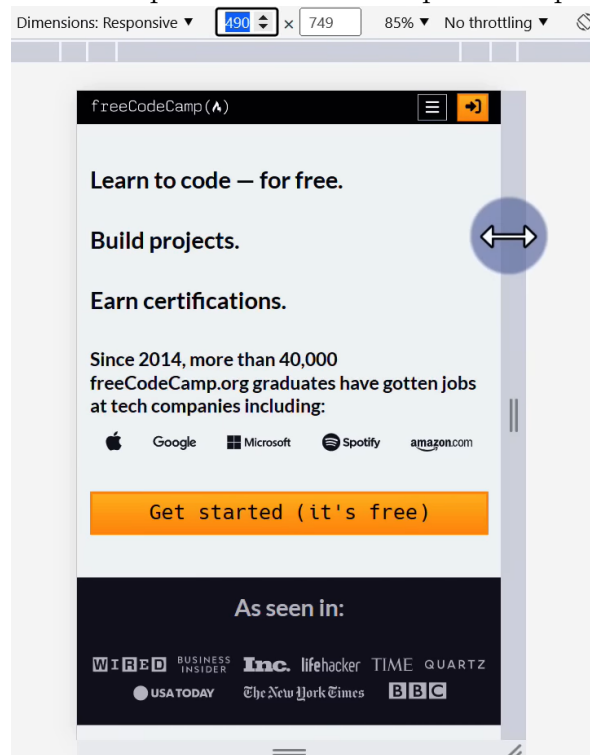


Figure 8: Aspecto nuevo posterior al cambio por breakpoints en sitio real



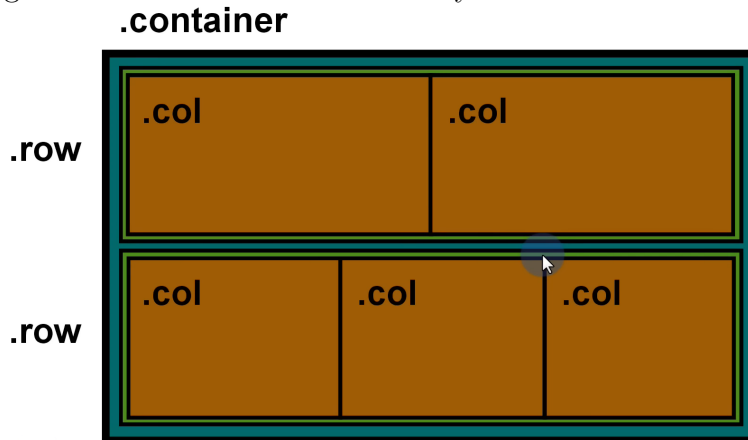
### 3.1.3 Estructura

Hablando más prácticamente de cómo será distribuido el contenido HTML en la Grid de Bootstrap, recordemos que podemos insertar la cantidad de filas que deseemos con la clase `.row` y que cada fila estará seccionada por 12 columnas, con esto podemos darle el tamaño que deseemos a los elementos, siempre y cuando esté dentro de estas 12 columnas. Un ejemplo simple y visual es el siguiente.

```
<div class='container'>
  <div class='row'>
    <div class='col'></div>
    <div class='col'></div>
  </div>
  <div class='row'>
    <div class='col'></div>
    <div class='col'></div>
    <div class='col'></div>
  </div>
</div>
```

Visualmente, tiene el aspecto de la Figura 9.

Figure 9: Uso de las clases de filas y columnas de Bootstrap



Donde el recuadro verde `.container` almacena los recuadros cafés que representan los divs con las clases `.row` y `.col`, las líneas 2 a 5 representan la primer fila y dos recuadros dentro del recuadro verde, mientras que las líneas 5 a 10 representan la segunda fila. Podemos concluir que estas filas adoptan el ancho total de su contenedor y las columnas adoptan un ancho dependiendo del tamaño de su contenedor y de la cantidad de columnas que existe, de tal manera que su ancho final es uniforme entre el número de columnas.

### 3.1.4 Contenedores

En el ejemplo de la sección anterior, vimos que todas las filas y columnas utilizadas fueron contenidas dentro de un div `.container`:

```
<div class='container'></div>
```

Un contenedor Bootstrap es aquel que almacena todos los elementos HTML y Bootstrap que serán distribuidos o utilizados por las filas y columnas del Grid de Bootstrap, es decir, un contenedor que contendrá contenedores con las clases *row* y *col*.

Bootstrap maneja dos tipos de contenedores principales, los cuales son:

- **.container**: esta clase crea un contenedor responsivo con un ancho máximo fijo que depende del ancho de la pantalla del dispositivo. Este contenedor no abarca la totalidad del ancho de pantalla, siempre deja un espacio en blanco o margen en las esquinas y se reduce en base a los breakpoints. La Figura 10 contiene un ejemplo visual de este contenedor.
- **.container-fluid**: esta clase crea un contenedor responsivo que cubre el 100% del ancho de la ventana. La Figura 11 contiene un ejemplo visual de este contenedor.

Figure 10: Aspecto original de la clase container



.container

Figure 11: Aspecto original de la clase container-fluid



.container-fluid

La Figura 12 y 13 muestran el aspecto que toman las clases container y container-fluid después de reducir su tamaño.

Figure 12: Nuevo aspecto de la clase container

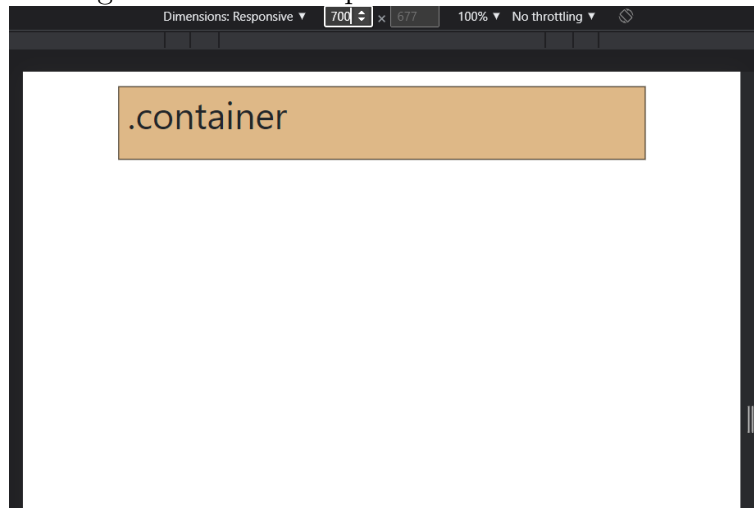
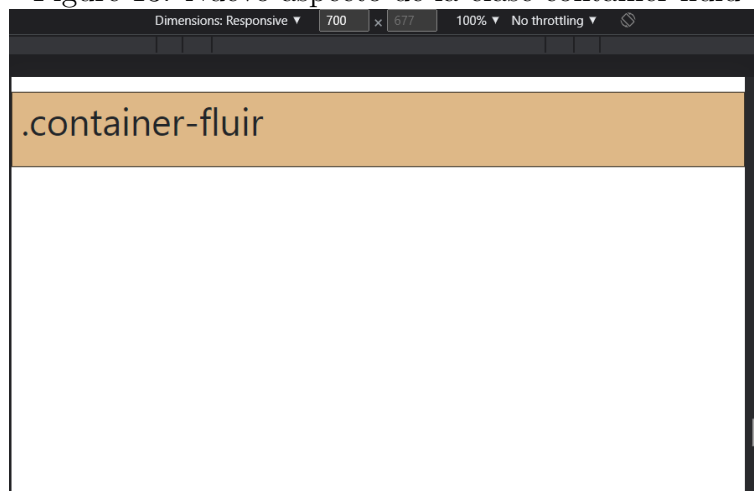


Figure 13: Nuevo aspecto de la clase container-fluid



### 3.1.5 Contenedores responsivos

Estos contenedores cambian su tamaño cuando se llega a un breakpoint determinado, recordemos que los breakpoints básicos de Bootstrap están presentes en la Tabla 1, por lo que, al sobrepasar la cantidad de píxeles de algún breakpoint, el contenedor responsivo tomará un ancho fijo con respecto al tamaño de la pantalla, si menos pasa la cantidad de píxeles, el contenedor adopta el 100% del ancho de la pantalla; dicho de otra manera, si sobre pasa el breakpoint, se comporta como la clase *container*, si lo menos pasa, adopta el comportamiento de la clase *container-fluid*. Los contenedores disponibles son:

- **.container-sm.**
- **.container-md.**
- **.container-lg.**

- **.container-xl.**
- **.container-xxl.**

Veamos un ejemplo para comprender mejor estos contenedores.

```
<div class='container'>  
  .container  
</div>  
<div class='container-fluid'>  
  .container-fluid  
</div>  
<div class='container-lg'>  
  .container-lg  
</div>
```

Añadimos las clases de los otros tipos de contenedores para que se puedan diferenciar visualmente. La Figura 14 muestra como se ven los contenedores responsivos después del breakpoint, mientras que la Figura 15 muestra su aspecto después de menos pasar el breakpoint, como podemos concluir, primer se ve como un contenedor fijo y luego como uno fluido.

Figure 14: Aspecto original de la clase container-lg

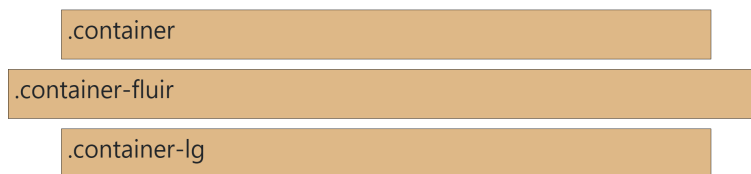
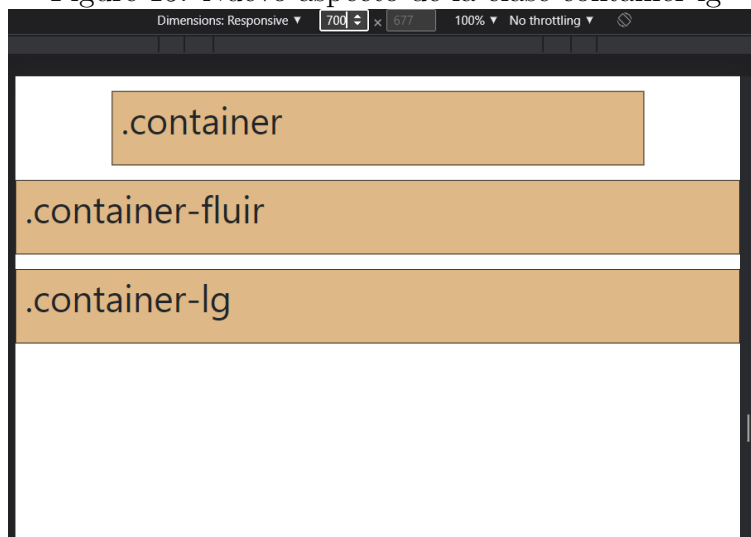


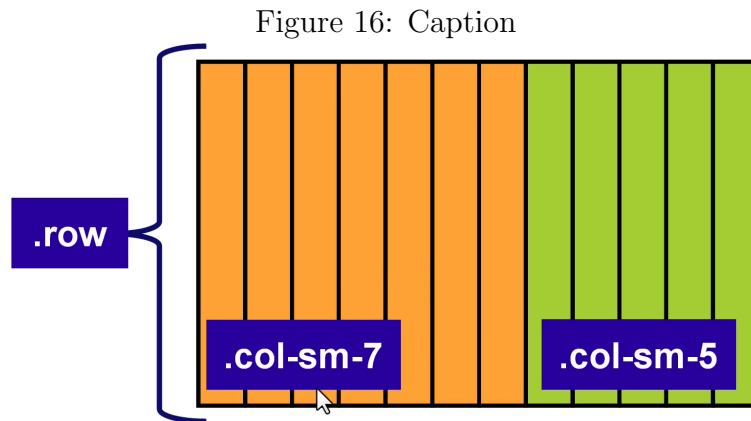
Figure 15: Nuevo aspecto de la clase container-lg



Este tipo de contenedores son efectivos cuando se pretende ajustar el contenido de la página a un tamaño específico de pantalla.

### 3.1.6 Trabajando con las filas y columnas

A las clases para la Grid se le puede agregar un valor entero positivo para que, menos pasar un breakpoint, adopte  $n$  cantidad de columnas de la fila. La Figura 16 muestra lo anterior mencionado.



Note que, si el ancho de la página es menor a la dimensión establecida por el breakpoint  $sm$ , el elemento con la clase *col-sm-7* tomará un ancho de 7 columnas con respecto a las 12 de la fila en la que está situado, mientras que el elemento con la clase *col-sm-5* adoptará 5 columnas. Debe fijarse que la suma de estos números enteros de como resultado 12, caso contrario colocará las columnas sobrantes en una nueva línea.

El siguiente ejemplo y la Figura 17 y 18 muestran un ejemplo sobre como utilizar estas filas y columnas con las clases para Grid y los contenedores.

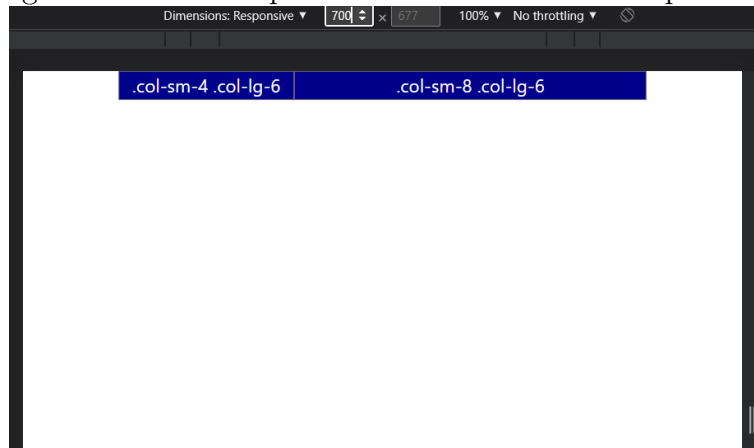
```
<div class="container text-center">
  <div class="row">
    <div class="col-sm-4 col-lg-6 columna">
      .col-sm-4 .col-lg-6
    </div>
    <div class="col-sm-8 col-lg-6 columna">
      .col-sm-8 .col-lg-6
    </div>
  </div>
</div>
```

Figure 17: Aspecto original de los contenedores responsivos





Figure 18: Nuevo aspecto de los contenedores responsivos



La primer figura muestra como se ven las columnas cuando el ancho de la pantalla es grande, adopta el tamaño de la clase *col-lg-6*, cuando este tamaño de pantalla es mejor a la resolución de *lg*, adopta el tamaño de las clases *col-sm-4* y *col-sm-8* respectivamente. Estas clases son útiles para reajustar el contenido con base a los breakpoints.

## 3.2 Componentes

Los componentes de Bootstrap son etiquetas HTML reutilizables predeterminadas. Estos controles vienen con un estilo definido y pueden llegar a ser personalizados. Visite la documentación oficial de Bootstrap en su menú lateral izquierdo para consultar todos los componentes de Bootstrap disponibles, su visualización interactiva en el sitio y el código fuente para copiar e insertar en sus propios proyectos, simplemente siga las instrucciones de uso y modificación para aplicarlo a sus trabajos.

## 3.3 Iconos

Son recursos de imágenes gratuitos disponibles a usar en Bootstrap. Utilice este [enlace](#) para explorar todos los iconos disponibles y el pequeño tutorial para implementarlos en sus proyectos.

## 3.4 Flexbox

Flexbox es la abreviación de **CSS Flexible Box Layout**, vimos previamente que todos estos conceptos se enfocan en conseguir un diseño responsivo para el cambio de tamaño entre dispositivos que visualicen las páginas de nuestro sitio web, sin embargo, se puede utilizar Flexbox para que los elementos HTML o Bootstrap reajusten su tamaño, posición alineación dentro del contenedor que lo almacena de manera automática y uniforme.

Bootstrap utiliza clases que manipulan Flexbox de manera más sencilla y eficiente. La manera tradicional sería la siguiente.

```
<style>
.cont {
```

```

        display: flex;
    }
</style>

<div class='cont'>
    Contenedor
</div>

```

Mientras que con Bootstrap sería la siguiente.

```

<div class='d-flex'>
    Contenedor
</div>

```

Los dos ejemplos anteriores realizan la misma función.

### 3.4.1 Propiedades del Flexbox en Bootstrap

La Tabla 2 muestra algunas de las propiedades básicas de Flexbox en Bootstrap.

Table 2: Propiedades de Flexbox

Propiedad	Descripción	Clases equivalentes que hay en Bootstrap
flex-direction	Establece el eje principal del contenedor, la dirección en la cual se va a colocar los elementos dentro del contenedor. Los elementos se pueden alinear horizontal o verticalmente, regular o en reversa	—
justify-content	Esta define el cómo se tienen que distribuir los elementos en el eje principal	.justify-content-start .justify-content-end .justify-content-center .justify-content-between .justify-content-around .justify-content-wvenly
align-items	Define cómo se distribuyen los elementos en el eje perpendicular (o eje secundario) al eje principal	.align-items-start .align-items-end .align-items-center .align-items-baseline .align-items-stretch
flex-wrap	Determina si los elementos deben ser ajustados para que siempre estén en una misma línea o si se les permite distribuirse en varias líneas si es necesario	.flex-nowrap .flex-wrap .flex-wrap-reverse

En esta sección debe conocer el funcionamiento de Flexbox para visualizar y trabajar esta herramienta con Bootstrap, además de consultar la documentación para conocer cómo trabajar con el resto de propiedades disponibles.