

## UD 4\_1: FRAMEWORKS DE DISEÑO WEB RESPONSIVE - BOOTSTRAP

### FRAMEWORKS DE DISEÑO WEB

Los frameworks de diseño ofrecen un esqueleto basado en columnas que facilita mucho la tarea de construir una web que se adapte a cualquier dispositivo. Además, ofrecen una serie de clases CSS y funciones de JavaScript que se pueden utilizar directamente y que nos ahorran tiempo a la hora de crear código.

Hoy en día existe en el mercado una amplia variedad de **frameworks de diseño web *responsive*** que nos ayudan a implementar nuestras interfaces gráficas. Algunos de los **frameworks más utilizados** son los siguientes:

- **Bootstrap** (<http://getbootstrap.com/>): Este framework, desarrollado por **Twitter**, es uno de los más populares actualmente. Bootstrap ha sido creado para ofrecer la mejor experiencia de usuario tanto en versión escritorio, como en tabletas y smartphones. Utiliza un *grid responsive* de 12 columnas y trae integrado decenas de complementos, plugins de JavaScript, tipografías, controladores de formularios, etc. Además, utiliza el [preprocesador de CSS](#) Less.
- **Materialize CSS** (<https://materializecss.com/>): Es un framework inspirado en Material Design de Google, lo que permite dar un aspecto muy similar a lo que podemos ver en Android o en las páginas creadas por Google. No es tan completo como Bootstrap pero visualmente es muy atractivo y está muy orientado al diseño web *responsive*.
- **Zurb Foundation** (<http://foundation.zurb.com/>): Junto con Bootstrap es uno de los frameworks más avanzados que existen en la actualidad. Ha sido desarrollado con Sass, un potente preprocesador de CSS que hace de Foundation un framework personalizable.
- **Skeleton** (<http://getskeleton.com/>): Ofrece un *grid responsive* basado en una resolución de 960px que se ajusta al tamaño de los dispositivos móviles. Tiene poco peso e incluye una colección de archivos CSS y JS para facilitarnos el diseño de nuestra web.
- **HTML5 Boilerplate** (<https://html5boilerplate.com/>): Tal y como los anteriores, ofrece un set de utilidades para construir nuestra web *responsive* de forma rápida y sencilla, con la ventaja de ser uno de los frameworks que menos ocupan.

Fuente: <https://www.eniun.com/frameworks-diseno-web-responsive/>

### BOOTSTRAP - DESCRIPCIÓN

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “**responsive**”

**design”** o diseño adaptativo.

Ventajas:

- Utiliza componentes y servicios creados por la comunidad web, tales como: HTML5 shim, Normalize.css, OOCSS (CSS orientado a objetos), jQuery UI, LESS y GitHub.
- Es un conjunto de buenas prácticas.
- El famoso Grid system, que por defecto incluye 12 columnas fijas o fluidas, dependiendo de si tu diseño será Responsive o no. Está orientado siguiente el concepto Mobile First.
- Soporte: Hay una enorme comunidad que soporta este desarrollo y cuenta con implementaciones externas como WordPress, Drupal, SASS o jQuery UI.
- Comodidad y rapidez: Herramienta sencilla y ágil para construir sitios web e interfaces.
- Componentes: Mucha variedad de plantillas y temas.

Desventajas:

- Aprendizaje: Es necesario adaptarse a su forma de trabajo, si bien su curva de aprendizaje es liviana, deberás comprender y familiarizarte con su estructura y nomenclatura.
- Adaptación: Debes adaptar tu diseño a un grid de 12 columnas, que se modifican según el dispositivo. Aquí empiezan los problemas, Bootstrap por defecto te trae anchos, márgenes y altos de línea, y realizar cambios específicos es por decir, un poco tedioso.
- Mantenimiento: Es complicado, cambiar de versión si has realizado modificaciones profundas sobre el core.
- Ampliar componentes: Si necesitas añadir componentes que no existen, debes hacerlos tú mismo en CSS y cuidar de que mantenga coherencia con tu diseño y cuidando el responsive.
- Pesado: No es ligero, y además, para algunas funcionalidades, será necesario tener que usar JavaScript y jQuery.

<http://getbootstrap.com/getting-started/>

#### BOOTSTRAP 5 NOVEDADES:

- Adiós a JQuery
- No será compatible con Internet Explorer
- Propiedades personalizadas para CSS: variables
- Actualización de los formularios: nuevas características como los "switch" (interruptor)
- Mejor compatibilidad con Escritura de derecha a izquierda (RTL)
- Nuevo breakpoint (xxl)
- Menús fuera de lienzo (off-canvas): Menús que no aparecen en la página (su aparición se realiza al pulsar un botón,...)
- SVG en HTML: Incrustar estos elementos por html (no por css)

#### APUNTES EN INGLES

Bootstrap (página de Bootstrap): <https://getbootstrap.com/docs/5.1/getting-started/introduction/>  
(W3School) <https://www.w3schools.com/bootstrap5/>

#### APUNTES EN CASTELLANO

Bootstrap 5: [https://www.youtube.com/watch?v=ZuOL\\_DoG9k](https://www.youtube.com/watch?v=ZuOL_DoG9k)

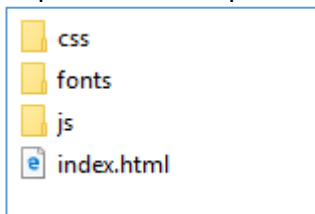
Bootstrap 4: <http://mialtoweb.es/indice-bootstrap-4/>

Bootstrap 3 (librosweb): [https://librosweb.es/libro/bootstrap\\_3/capitulo\\_2.html](https://librosweb.es/libro/bootstrap_3/capitulo_2.html)

## DESCARGA E INSTALACIÓN

### a) Opción DESCARGA:

1. Descarga: <http://getbootstrap.com/> → Download
2. Descomprimir
3. Copiar lo descomprimido en la carpeta del proyecto.



### b) Opción linkar online:

1. Entrar en <http://getbootstrap.com/>
2. Copiar el código:

### CDN via jsDelivr

Skip the download with [jsDelivr](#) to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFtdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgys0g+OMhuP+IlRH9sENB00Lrn5q+8nbTov4+1p" crossorigin="anonymous"></script>
```

### a. Pegarlo en el head del html:

```
<!-- Bootstrap CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFtdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgys0g+OMhuP+IlRH9sENB00Lrn5q+8nbTov4+1p" crossorigin="anonymous"></script>
```

## PRIMEROS PASOS

- Bootstrap escala hasta a 12 columnas según el dispositivo o el viewport. Para ello incluye clases, variables y mixins predefinidos.
- Bootstrap 5 establece varios tamaños diferentes de pantalla, con sus correspondientes puntos de corte:

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px

- Se utilizan sistemas de cuadrícula para crear distribuciones de página por medio de una serie de filas y columnas que albergan su contenido. Para eso, sólo tenemos que definir columnas en las que introducir los divisores con la siguiente pauta:
  - Resolución XXL: queremos los 4 bloques en fila. Pero que ocupen tamaños diferentes: El primero y el último 2 columnas y los del medio 4.
  - Resolución XL: queremos los 4 bloques en fila. Pero que ocupen tamaños diferentes: El primero 3 columnas, el segundo 4, el tercero 2 y el ultimo 3 (3+4+2+3= 12 columnas. Todos en la misma fila)
  - Resolución LG: queremos los 4 bloques en fila. Como hay 12 columnas→3 partes de 4 columnas
  - Resolución MD: queremos 3 bloques por fila: como hay 12 columnas, necesitaremos 4 partes de 3 columnas. Y al último le asignamos que ocupe las 12 (se irá a otra línea)
  - Resolución SM: queremos 2 bloques por fila: como hay 12 columnas, pondremos 6 partes de 2 columnas. Ocuparán 2 filas.
  - Resolución XS: queremos 1 bloque por fila al principio y al final y las 2 centrales en la misma.
- Si más de 12 columnas se colocan en una única fila, el grupo de columnas extras (del 13 en adelante) pasarán como una sola unidad a la siguiente línea.

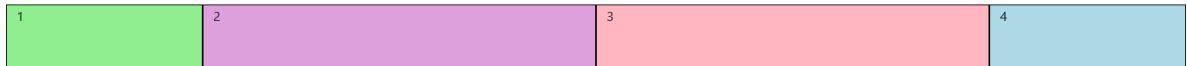


1-Ejemplo Primeros Pasos.html

```
<div class="row">
  <div class="col-xxl-2 col-xl-3 col-lg-3 col-md-4 col-sm-6 col-12 verde">1</div>
  <div class="col-xxl-4 col-xl-4 col-lg-3 col-md-4 col-sm-6 col-6 morado">2</div>
  <div class="col-xxl-4 col-xl-2 col-lg-3 col-md-4 col-sm-6 col-6 rosa">3</div>
  <div class="col-xxl-2 col-xl-3 col-lg-3 col-md-12 col-sm-6 col-12 azul">4</div>
</div>
```

- En pantallas extra extra grandes (xxl) el primer bloque ocupa 2 de las 12 columnas. El último igual. Y los 2 centrales 4.
 

```
col-xxl-2
col-xxl-4
col-xxl-4
col-xxl-2
```



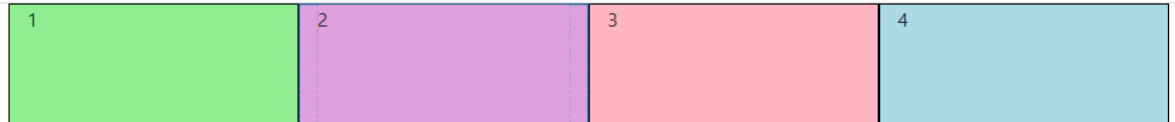
- En pantalla extra grandes (xl) el primer bloque ocupa 3 de las 12 columnas, el segundo 4 de las 12 columnas, el tercero 2 de las 12 columnas y el último 3 de las 12 columnas.

```
col-xl-3
col-xl-4
col-xl-2
col-xl-3
```



- En pantallas grandes (clase lg) ocupa 3 de las 12 columnas (25% cada div). Como hay 4 div, todos en la misma línea.

```
col-lg-3
col-lg-3
col-lg-3
col-lg-3
```



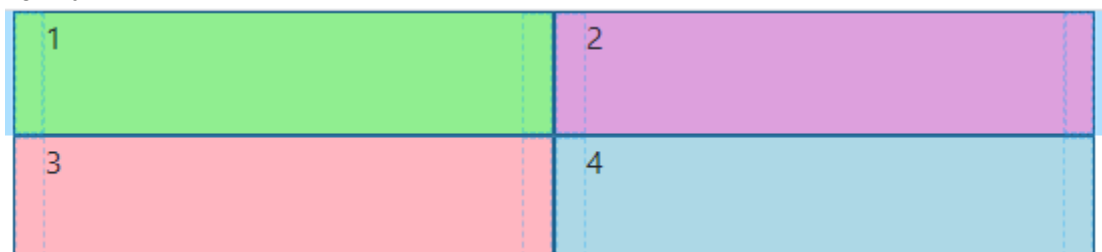
- En medianas (clase md) ocupa 4 de las 12 (33'33% cada div). Como hay 4, sólo caben 3 bloques, así que los sobrantes pasan como bloque a la siguiente línea. Y en este caso ocupa las 12 columnas (col-12)

```
col-md-4
col-md-4
col-md-4
col-md-12
```



- En Tablet (sm) ocupa 6 de las 12 (50% cada div). Como hay 4 cajas y sólo caben 2, las otras 2 pasan a la siguiente línea:

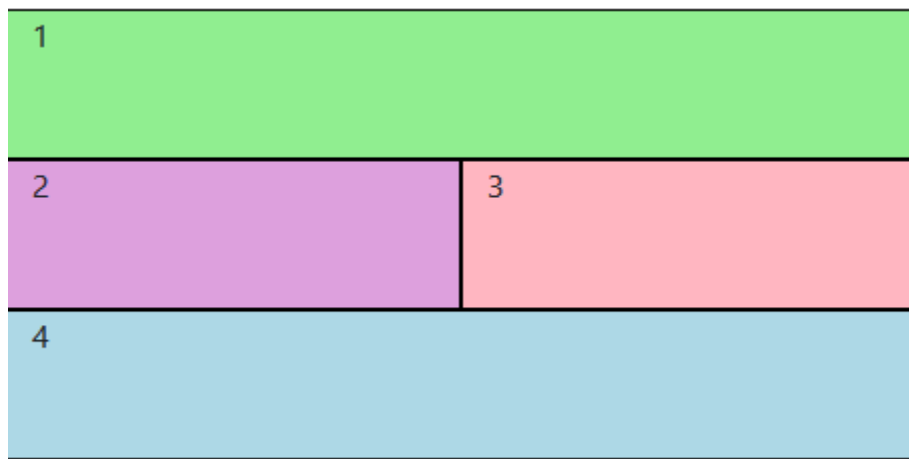
```
col-sm-6
col-sm-6
col-sm-6
col-sm-6
```



- Y en móviles (xs) ocupan la primera y la última las 12 de las 12 (100% cada div) y las del medio 6 (%50)

```
col-12
col-6
col-6
```

col-12



Si miramos en el archivo de clases predefinidas de bootstrap.css veremos los estilos en cuanto anchura de estas clases:

```
.col-12 {width: 100%;}
.col-11 {width: 91.66666667%;}
.col-10 {width: 83.33333333%;}
.col-9 {width: 75%;}
.col-8 {width: 66.66666667%;}
.col-7 {width: 58.33333333%;}
.col-6 {width: 50%;}
.col-5 {width: 41.66666667%;}
.col-4 {width: 33.33333333%;}
.col-3 {width: 25%;}
.col-2 {width: 16.66666667%;}
.col-1 {width: 8.33333333%;}
```

El sistema de cuadrícula de Bootstrap funciona así:

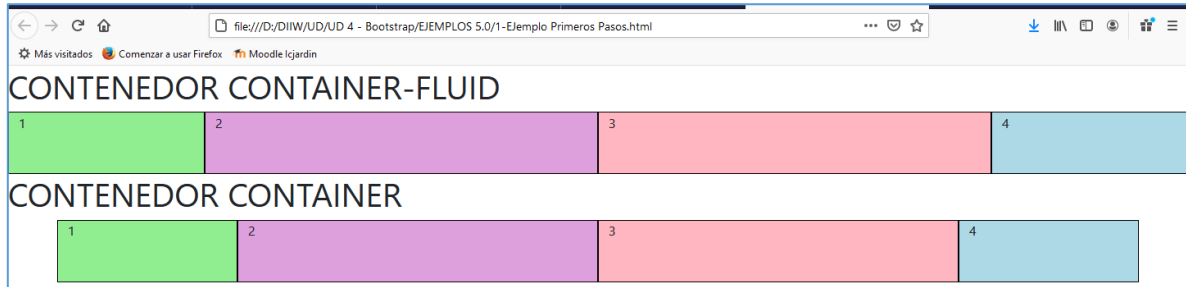
- Se deben colocar filas dentro de un `.container` (ancho fijo) o `.container-fluid` (ancho completo) para una alineación y relleno adecuados.

Si quieres centrar una página respecto a la ventana del navegador, encierra sus contenidos dentro de un elemento y aplícale la clase `.container`:

```
<div class="container">
  ...
</div>
```

<http://www.ryanwright.me/cookbook/bootstrap/container-vs-container-fluid>

- Las filas siempre se definen dentro de un contenedor de tipo `.container` (anchura fija) o de tipo `.container-fluid` (anchura variable- siempre el 100% de la pantalla). De esta forma las filas se alinean bien y muestran el padding correcto.



- Hay otros tipos de contenedores que se pueden especificar, `.container-{breakpoint}`, siendo breakpoint = `sm|md|lg|xl|xxl` teniendo en cuenta lo que ocuparán según el tamaño del viewport:

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%



2-Ejemplo Primeros Pasos -container-tamañosxs-md-lg-xl-xxl.html

[https://www.w3schools.com/bootstrap5/bootstrap\\_containers.php](https://www.w3schools.com/bootstrap5/bootstrap_containers.php)

Para saber el contenedor máximo que podrá albergar mi pantalla tengo que mirar la resolución configurada en la misma.

Ejemplo:

Resolución de pantalla

1024 × 768

En una pantalla configurada así: el contenedor mayor que podrá albergar será el lg (992px). El siguiente, xl, es de 1200px, mayor que lo que tengo configurado.

Si cuando estoy utilizando bootstrap, no puedo ver el comportamiento de los elementos en las pantallas más grandes, debo cambiar esta resolución:

Escritorio → Botón derecho → Configuración de pantalla → Resolución:

1920 × 1080	XXL
1680 × 1050	
1600 × 1200	
1600 × 1024	
1600 × 900	
1366 × 768	XL
1360 × 768	
1280 × 1024	
1280 × 960	
1280 × 800	
1280 × 768	
1280 × 720	
1152 × 864	LG
1024 × 768	
800 × 600	

- Las filas se utilizan para agrupar horizontalmente a varias columnas.
- El contenido siempre se coloca dentro de las columnas, ya que las filas sólo deberían contener como hijos elementos de tipo columna.
- Bootstrap define muchas clases CSS (como por ejemplo `.row` y `.col-4`) para crear rejillas rápidamente.
- La separación entre columnas se realiza aplicando `padding`. Para contrarrestar sus efectos en la primera y última columnas, las filas (elementos `.row`) aplican márgenes negativos.
- Las clases de cuadrícula se aplican a dispositivos con anchos de pantalla mayores o iguales a los de punto de quiebre y reemplazan a las clases de cuadrícula que se usan en los dispositivos más pequeños. Por tanto, al aplicar cualquier clase `.col-md-` a un elemento no solo que se afectará su estilo en dispositivos medianos, sino también en grandes, si no hay una clase `.col-lg-` presente.
- Se establece a blanco el color de fondo del `body` con la propiedad `background-color: white;`



- Se utiliza el valor de las variables `@font-family-base`, `@font-size-base` y `@line-height-base` definidas por LESS como atributos de las propiedades tipográficas de los elementos.
- Se establece el color de los enlaces al valor de la variable `@link-color` de LESS y sólo se muestran los enlaces subrayados en el estado `:hover`

Ejemplo estructura página web:



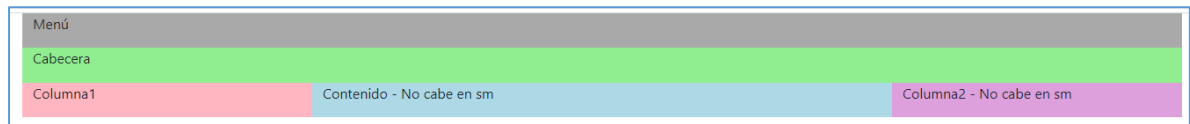
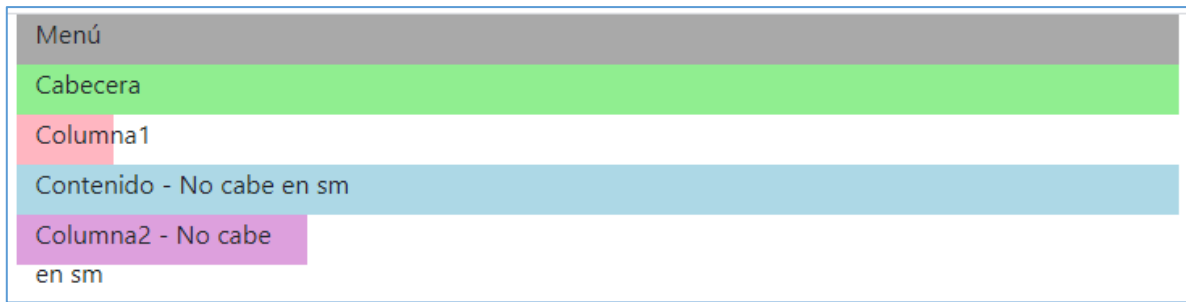
3-Ejemplo Estructura.html

```
<div class="container">
  <div class="row">
    <!-- Fila con único componente hijo. No comparte espacio. Ocupa las 12 columnas
    aunque no se lo indique-->
    <div class="gris">Menú</div>
  </div>

  <div class="row">
    <!-- Fila con único componente hijo. No comparte espacio. Ocupa las 12 columnas
    aunque no se lo indique-->
    <div class="verde">Cabecera</div>
  </div>

  <div class="row">
    <!-- Fila con 3 componentes. Cuando sea posible compartirán espacio-->
    <div class="col-12 col-sm-1 col-lg-3 rosa">Columna1</div>
    <!--No están especificados varios tamaños. Entonces recurre a la última
    configuración. Es decir:
    - md se quedará con la configuración de sm
    - xl y xxl se quedarán con la configuración de lg.-->
    <div class="col-12 col-sm-12 col-lg-6 azul">Contenido-No cabe en sm</div>
    <div class="col-12 col-sm-3 col-lg-3 morado">Columna2-No cabe en sm </div>
  </div>
</div>
```

Menú
Cabecera
Columna1
Contenido - No cabe en sm
Columna2 - No cabe en sm



## PLANTILLA INICIAL



Plantilla básica.html