

Parte 1 de 4: Bootstrap

INTRODUCCIÓN A BOOTSTRAP INSTALACIÓN CREACIÓN DE UNA PÁGINA BÁSICA SISTEMA DE REJILLA MEDIA QUERY

Apuntes de: **Rosa María Medina Gómez**

Adaptados a Power Point por José Jesús Torregrosa García

Curso de Formación del Profesorado a distancia
Cefire Específico de FP de Chestre

Generalitat Valenciana Curso 2018 - 2019



Bootstrap

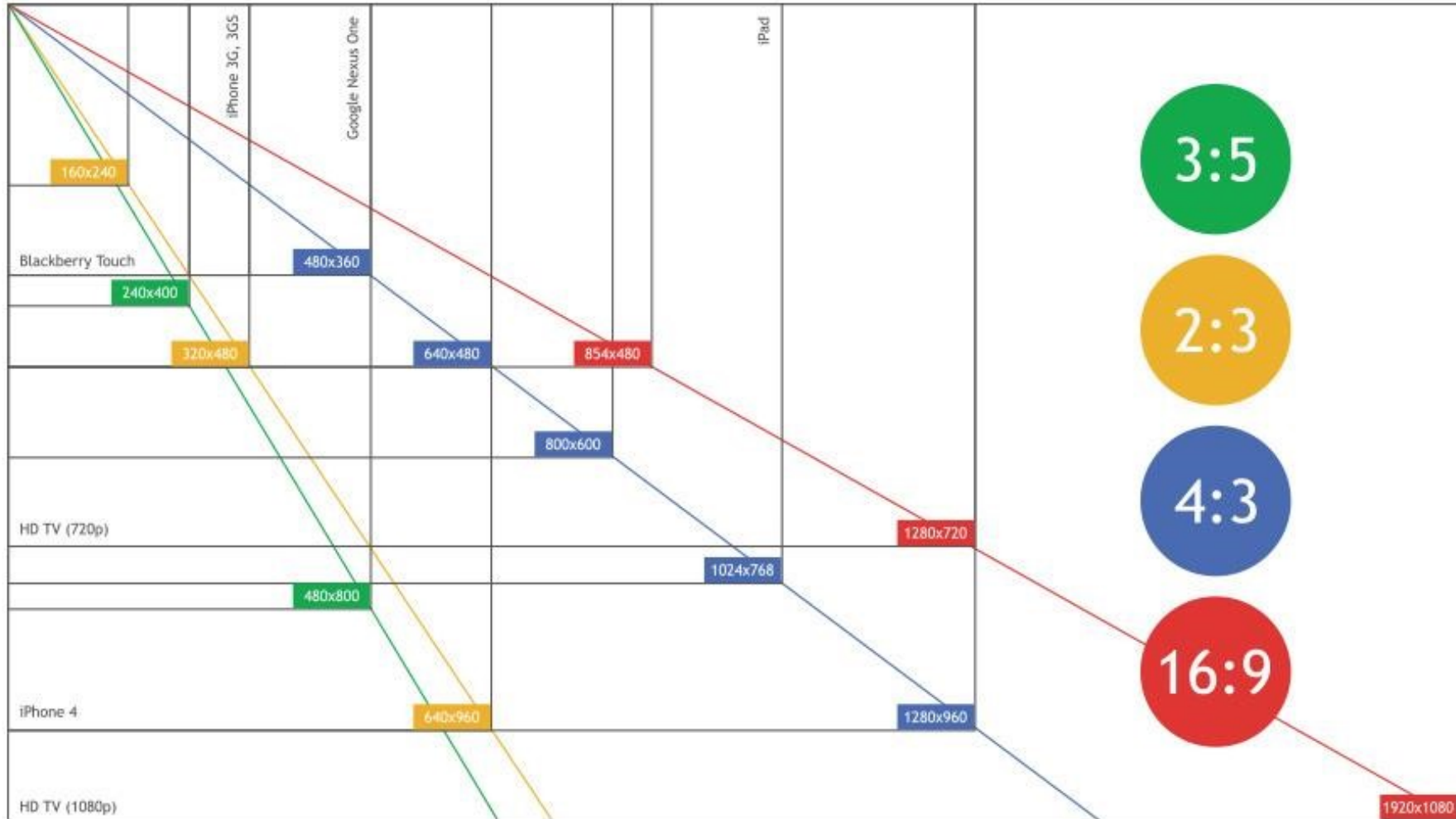
Introducción

El *Diseño Web Responsive* (**RWD** - Responsive Web Design) es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo (tablets, teléfonos, ebooks, portátiles, PCs, etc.) que se esté utilizando para visualizarla.

Variabilidad en las resoluciones de pantalla

Actualmente el desarrollo web se basaba en una amplia variedad de resoluciones (no solo para 1024 x 768, como antiguamente), y no solo en ordenadores de escritorio sino también para tablets y móviles.

Introducción



Introducción

Estadísticas de las resoluciones de pantalla más utilizadas (Enero del 2017). En la actualidad ya no es 1024x768 la resolución más utilizada, sino que es **1366x768** y resoluciones **superiores a 1920x1080**:

Resolución	% de utilización
> 1920x1080	35%
1920x1080	17%
1366x768	35%
1280x1024	5%
1280x800	4%
1024x768	3%
800x600	0.5%
< 800x600	0.5%

Introducción

Es fundamental tener en cuenta que en el diseño responsive, al variar tanto las posibles resoluciones deberemos mostrar en primer lugar los **contenidos** más importantes e **imprescindibles**. Como ejemplo tenemos: **Food Sense**



Introducción

Matt Kersley



Frameworks responsive

Hoy en día existen en el mercado una amplia variedad de frameworks responsive (para evitar inventar la rueda), algunos de los más utilizados son:

- **Bootstrap** (<http://getbootstrap.com/>): desarrollado por el equipo de *Twitter*. Bootstrap ha sido creado para ofrecer la mejor experiencia de usuario tanto a usuarios de PC como a smartphones y tablets. Utiliza el *grid responsive de 12 columnas* y trae integrado decenas de *complementos, plugins de JavaScript, tipografía, controladores de formularios, etc.* Además, utiliza el *preprocesador de CSS less*.

Frameworks responsive

- **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.

Frameworks responsive

- **HTML5 Boilerplate** (<https://html5boilerplate.com/>): Al igual que los demás nos ofrece un set de utilidades para construir nuestra web responsive de forma rápida y sencilla, con la ventaja de ser uno de los que menos ocupan.

En este curso, nos centraremos en Bootstrap por ser uno de los más completos, utilizados y que mejor funciona.

Framework Bootstrap

- **Navegadores soportados:** Chrome, Firefox, Opera, Safari e Internet Explorer (a partir de la **versión 8**).
- Preparado para que funcione tanto en navegadores de PC, portátiles, así como tablets y Smartphone
- Sistema de rejilla dividido en columnas para el posicionamiento de los elementos web.
- Utilidades y complementos (formularios, botones, barras de navegación, etc.) para simplificar el desarrollo de una **web responsive**.

Diseño adaptable

- Adaptar de forma dinámica el diseño web en función de la resolución de la pantalla del cliente incluido móviles sin tener dos sitios separados.



Diseño adaptable

- **Responsive Web Design** nos permite hacer interfaces adaptadas al entorno del usuario mediante estructuras, bloques, columnas e imágenes fluidas gracias a *media-queries* de CSS
- A partir de la versión de **CSS 2.1** las hojas de estilo han incluido los *media types*, lo cual nos ha facilitado, por ejemplo, proveer un estilo distinto para imprimir.

```
<link rel="stylesheet" href="core.css" type="text/css" media="screen">  
<link rel="stylesheet" href="print.css" type="text/css" media="print">
```

Diseño adaptable

A partir de **CSS3**, el **W3C** creó las *media queries*. Una media query nos permite inspeccionar las características del dispositivo que está renderizando nuestra web. Para utilizarlas podemos incorporar una *query* al atributo media de un link a una hoja de estilos:

```
<link rel="stylesheet" href="shetland.css" type="text/css"  
      media="screen and (max-device-width: 480px)">
```

Diseño adaptable

La query contiene dos componentes:

→ **type:screen, print o all:** una consulta entre paréntesis conteniendo una *característica a inspeccionar* (**max-device-width** o **min-device-width**) seguida por el valor al que apuntamos.

→ También es posible utilizarlas directamente en nuestro código CSS como parte de una regla:

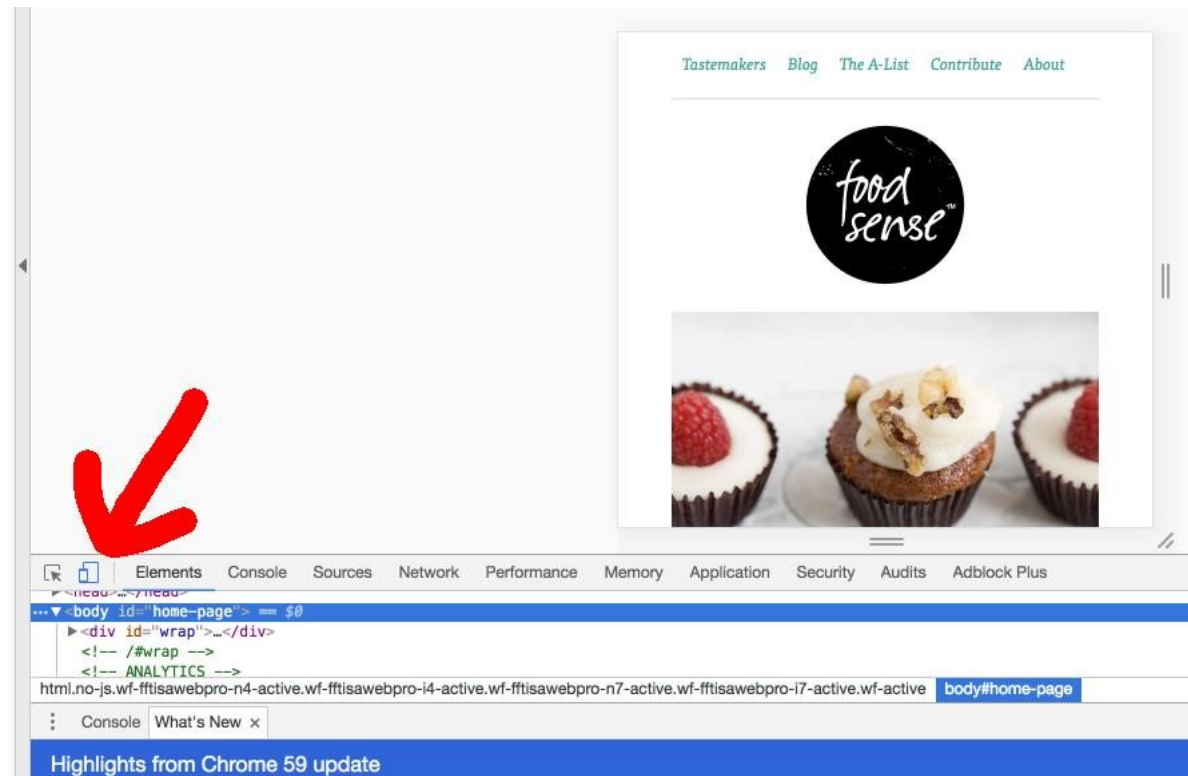
```
@media only screen and (max-width: 500px)
{
  body {
    background-color: lightblue;
  }
}
```

Para más información sobre las media queries podéis consultarlo aquí:

<https://www.w3.org/TR/css3-mediaqueries/>

Herramientas del desarrollador

Tanto [Firefox](#) como [Chrome](#), tiene instalado por defecto una serie de herramientas que nos permiten entre otras cosas ver la consola de mensajes, inspeccionar el código o ver la secuencia de llamadas al servidor.



Herramientas del desarrollador

Podemos probar nuestra página en local y además inspeccionar el código, así como modificar los estilos en tiempo real. Usando el inspector de estas herramientas nos podemos ahorrar mucho tiempo a la hora de realizar pruebas sobre la propia página cargada, ya que de otra forma tendríamos que modificar el código directamente, recargarla y volver a probarla.

Descarga e instalación

Para descargar **Bootstrap** (no es un instalador), deberemos descargarnos el zip que se encuentra en la web oficial. Descargaremos bien la última versión disponible, o bien la que se ha utilizado para realizar el curso que está [aquí](#) (**v4.1.2**).

Descarga e instalación

Una vez descargado el archivo, lo descomprimos y lo guardamos en el directorio que queramos y/o nos sea fácil acceder a él.

Para poder acceder desde nuestro **HTML** a él, lo que haremos será enlazar el **CSS**, y si queremos usar las funcionalidades de **javascript**, también deberemos enlazar este, ojo **el bundle incluye la librería Popper, pero en ningún caso incluye jQuery**, por tanto, **se os recomienda descargarlas**.

Descarga e instalación

```
<!-- Bootstrap CSS -->
Ejemplo:
<link rel="stylesheet" href="../../bootstrap-4.1.2/dist/css/bootstrap.min.css"
type="text/css">
<script src="../../bootstrap-4.1.2/dist/js/bootstrap.min.js"
type="text/javascript"></script>
```

Si no queremos tener descargado el archivo, y preferimos que la web se conecte al servidor de Bootstrap y lo cargue en caché usaremos las etiquetas que lo enlaza y deberemos poner antes de nuestros CSS:

```
<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/css/bootstrap.min.css"
integrity="sha384-Smlep5jCw/wG7hdkwQ/Z5nLiefveQRIY9nfy6xoR1uRYBtpZgI6339F5dgvm/e9B"
crossorigin="anonymous">
```

Descarga e instalación

Luego incluimos los JS antes de cerrar el body, y ponemos primero jQuery, luego Popper.js y por último Bootstrap:

```
<!-- Optional JavaScript -->
<!-- Optional JS: jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbbJiSnjAK/l8WvCWPIpM49"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/js/bootstrap.min.js"
integrity="sha384-o+RDsa0aLu++PJvFqy8fFScvbHFLtbvScb8AjopnFD+iEQ7wo/CG0xlczd+20/em"
crossorigin="anonymous"></script>
```

Contenido directorio Bootstrap

Una vez que hemos descargado y descomprimido, si ejecutáis en linux el comando **tree -d bootstrap-4.1.2**, tenemos dentro algo parecido a:

```
bootstrap/  
├── dist/  
│   ├── css/  
│   └── js/  
├── docs/  
│   └── examples/  
├── js/  
└── scss/
```

Contenido directorio

Bootstrap

- **scss/** y **js/** está el código de Bootstrap CSS y JavaScript.
- **dist/** incluye todo lo que aparece en la sección precompilada.
- **docs/** tiene el código fuente de nuestra documentación
- **examples/** ejemplos de cómo usar Bootstrap.

Aparte de estos directorios, cualquier otro directorio incluido proporciona soporte para paquetes, información de licencias y desarrollo.

Si queremos ver en qué navegadores/plataformas está soportado Bootstrap, podemos consultarlo [aquí.](#)

Página básica

Bootstrap utiliza elementos HTML y propiedades CSS que requieren el uso del **doctype de HTML5** para que funcionen, por lo tanto, es **importante añadirlo en TODAS** las páginas:

```
1 <!doctype html>
2 <html lang="es">
3   <head>
4     <title>Ejercicio 1</title>
5     <!-- Required meta tags -->
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

Además, para asegurarnos que tse está visualizando de forma correcta y que podemos utilizar el zoom en los móviles, debemos añadir la etiqueta anterior **meta** con name “**viewport**” dentro de **<head>**

Página básica

Si queremos deshabilitar el zoom para los móviles, tendríamos que añadir la etiqueta meta del viewport, con el valor de **user-scalable=no**.

```
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
```


Sistema de rejilla: elemento contenedor

- El **sistema de rejilla de Bootstrap** se basa en rejillas flexibles, las cuales se escalan al tamaño y posición adecuadas dependiendo de la pantalla en la que se esté visualizando la web.
- El sistema de rejilla **tiene que ser utilizado dentro de uno de los dos elementos contenedores** que tiene Bootstrap: **.container** o **.container-fluid**. Es importante tener en cuenta que estos elementos se utilizan como raíz de la rejilla y no se podrán anidar unos dentro de otros.

Sistema de rejilla: funcionamiento

- El sistema de rejilla está pensado para ayudar en la disposición del contenido y adaptación a las diferentes pantallas.
- El contenido irá dentro de celdas o columnas que irán dentro de filas.
- Cada fila se puede dividir hasta en 12 columnas.
- La persona que diseña es el que define las columnas para cada tamaño de pantalla.

Sistema de rejilla: funcionamiento

- Las columnas deben ir agrupadas dentro de filas (**.row**)
- Las filas (**.row**) se deben colocar dentro de una etiqueta contenedora: **.container** (para ancho fijo) o **.container-fluid** (para poder ocupar todo el ancho), esto permitirá alinear las celdas y asignarles el espacio correcto.
- El contenido se debe disponer dentro de columnas o celdas, las cuales deben de ser el único hijo posible de las filas (**.row**), que a su vez será el único hijo posible del contenedor (**.container** ó **.container-fluid**)

Sistema de rejilla: funcionamiento

- Al seguir este orden el sistema de rejilla funcionará creando el espaciado interior y los márgenes apropiados dependiendo de las dimensiones de la pantalla.
- Cada fila se puede dividir hasta un máximo de 12 columnas, pero somos nosotros los que tendremos que definir el número de columnas en el que queremos dividir cada fila y su ancho para cada tamaño de pantalla.
- Si el tamaño total de las columnas de una fila excede de 12, el sobrante se colocará en la siguiente fila.

Sistema de rejilla: funcionamiento

- El **tamaño de las columnas** se especificará con **clases CSS** que Bootstrap define para cada tamaño de pantalla, por ejemplo **.col-md-XX** donde XX es el tamaño de la columna, que podrá tomar valores entre 1 y 12. Si quieres que todas las columnas tengan el mismo tamaño se usaría: **.col-4** por ejemplo.

Sistema de rejilla: funcionamiento

- En la siguiente tabla se recoge un resumen del funcionamiento de la **rejilla de Bootstrap** dependiendo del **tamaño de la pantalla** y las **clases CSS**:

Pantalla (tamaño)	Prefijo de la clase	Ancho del contenedor
Extra pequeño (<576px)	.col-	Ninguno (automático)
Pequeño (≥576px)	.col-sm-	540px
Mediano (≥768px)	.col-md-	720px
Largo (≥992px)	.col-lg-	960px
Extra largo (≥1200px)	.col-xl-	1140px

Ejemplos: mismo ancho

- Es importante tener en cuenta que, al definir estas clases, no sólo se aplican para ese tamaño de pantalla, sino que para las superiores también. Por ejemplo, **si no** aplicamos clase para tamaños de pantalla medianos y grandes **pero sí para pequeños** (**.col-sm-**), se usará esta, por ejemplo en [rejillas3.html](#) (dentro de la carpeta [./Ejemplos sesion 5/Rejillas](#)) son dos filas con un número diferente de celdas, y en todos los dispositivos se ve igual. Pero si los defines se sobrescriben y pasan a utilizar los definidos para dichos tamaños.

1 of 2		2 of 2	
1 of 3	2 of 3		3 of 3

Ejemplos: solo para escritorio

Ejemplo de 3 filas, la primera de 2 columnas de tamaño desigual, la segunda 3 columnas del mismo tamaño y la tercera fila 2 columnas del mismo tamaño ([ver rejillas4.html](#)).



.col-md-8		.col-md-4
.col-md-4	.col-md-4	.col-md-4
.col-md-6		.col-md-6

Ejemplos: solo para escritorio

Como las columnas se han especificado únicamente mediante las clases **.col-md-*** esto se aplicará tanto en pantallas de escritorio medianas y grandes, pero no para los tamaños pequeños (tablets y móviles). En estos dos últimos casos, las columnas se ampliarán utilizando todo el ancho por lo que se mostrarán de la forma de la derecha



.col-md-8
.col-md-4
.col-md-4
.col-md-4
.col-md-4
.col-md-6
.col-md-6

Ejemplos: solo para móvil y escritorio

Si no queremos que las columnas se apilen en tamaños pequeños, podemos indicar también la disposición que queremos para esos casos usando las clases: **.col-xs-***, además de las que ya teníamos con **.col-md-***. Ejemplo ([ver rejillas5.html](#)).

- **Resultado** para pantallas md y lg (medianas y grandes respectivamente).

.col-xs-12 .col-md-8		.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	
.col-xs-6		.col-xs-6	

Ejemplos: solo para móvil y escritorio

- **Resultado** En el caso de pantallas pequeñas, las columnas se verían:

.col-xs-12 .col-md-8	
.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4
.col-xs-6 .col-md-4	
.col-xs-6	.col-xs-6

Ejemplos: para móvil, tablet, y escritorio

Para tener un mayor control, podemos especificar el tamaño de las columnas para las tablets, con las clases: **.col-sm-***. Ejemplo ([rejillas6.html](#)):

- **Resultado** en pantallas medianas y grandes:

.col-xs-12 .col-sm-6 .col-md-8			.col-xs-6 .col-md-4		
.col-xs-6 .col-sm-4		.col-xs-6 .col-sm-4		.col-xs-6 .col-sm-4	

Ejemplos: para móvil, tablet, y escritorio

- **Resultado** en pantallas de tipo tablet:

.col-xs-12 .col-sm-6 .col-md-8			.col-xs-6 .col-md-4		
.col-xs-6 .col-sm-4		.col-xs-6 .col-sm-4		.col-xs-6 .col-sm-4	

- **Resultado** en pantallas pequeñas:

.col-xs-12 .col-sm-6 .col-md-8					
.col-xs-6 .col-md-4					
.col-xs-6 .col-sm-4			.col-xs-6 .col-sm-4		
.col-xs-6 .col-sm-4					

Para consultar mas ejemplos y ayuda

<https://getbootstrap.com/docs/4.1/layout/grid/>

Cómo ocultar celdas u otro elemento

Con el objetivo de facilitar el desarrollo de aplicaciones web donde el contenido de páginas sea menor en función de si se están visualizando en el móvil o en el ordenador, bootstrap tiene las clases:

- **.d-none** para ocultarlo para todas las pantallas
- **.d-{sm,md,lg,xl}-none** si queremos que se oculte en alguna de esas resoluciones en concreto.
- **.d-{sm,md,lg,xl}-block** si queremos que se muestre en alguna de esas resoluciones en concreto. Por ejemplo la clase **.d-md-block** por ejemplo, donde se ocultarían en todas menos en las medias.

Cómo ocultar celdas u otro elemento

Más info: _

<https://getbootstrap.com/docs/4.1/utilities/display/#hiding-elements>

Tamaño de la pantalla	Clase
Oculto en todas	.d-none
Oculto sólo en xs	.d-none .d-sm-block
Oculto sólo en sm	.d-none .d-md-block
Oculto sólo en md	.d-none .d-lg-block
Oculto sólo en lg	.d-none .d-xl-block
Oculto sólo en xl	.d-xl-none
Visible en todas	.d-block
Visible sólo en xs	.d-block .d-sm-none
Visible sólo en sm	.d-none .d-md-block .d-md-none
Visible sólo en md	.d-none .d-md-block .d-lg-none
Visible sólo en lg	.d-none .d-lg-block .d-xl-none
Visible sólo en xl	.d-none .d-xl-block

Anidamiento de columnas (Nesting)

Bootstrap **nos permite anidar columnas**. Por ejemplo, si quisiéramos que en una columna de tamaño 9, crear una nueva fila y dividirla como queramos (funciona igual, cada fila tiene 12 celdas).

<https://getbootstrap.com/docs/4.1/layout/grid/#nesting>.

- **Ejemplo:** [cosasNuevas/rejillasPruebas.html](#)

Level 1: .col-sm-9	
Level 2: .col-xs-8 .col-sm-6	Level 2: .col-xs-4 .col-sm-6

Márgenes o espaciado entre columnas

En la versión 4 de Bootstrap, ya no se usa el offset que se empleaba en las versiones anteriores, en lugar de eso utilizamos la clase `.mr-auto` por ejemplo para forzar a las columnas a que tengan un espacio a la derecha (`ml-auto` para la izquierda).

<https://getbootstrap.com/docs/4.1/layout/grid/#margin-utilities>

Márgenes o espaciado entre columnas

- **Ejemplo:** cosasNuevas/rejillasPruebas2.html

.col-md-4

.col-md-4 .ml-auto

.col-md-3 .ml-md-auto

.col-md-3 .ml-md-auto

.col-auto .mr-auto

.col-auto

Ordenación de columnas (Reordering)

Podemos usar la clase `.order-` para controlar de forma visual el orden de nuestro contenido, teniendo en cuenta que el grid tiene 12 columnas. Si por ejemplo queremos que en una resolución md (o superior) esta columna esté la segunda usaremos la clase `.order-md-2`, y también está la posibilidad de usar `.order-first` y `.order-last`

- **Ejemplo:** [rejillas7.html](#).

First, but unordered	Third, but first	Second, but last
-------------------------	------------------	------------------

Mas documentación de reordenación:

<https://getbootstrap.com/docs/4.1/layout/grid/#reordering> y

con el grid: <https://getbootstrap.com/docs/4.1/layout/grid/>

Cómo romper una columna (salto)

Con la clase `.w-100`, lo que conseguimos es romper para que la siguiente columna baje a una línea nueva. Normalmente esto se consigue insertando una nueva `.row`, pero si por ejemplo queremos que esto ocurra en ciertas resoluciones usaremos este elemento de width: 100% (`.w-100`).

Para más información:

<https://getbootstrap.com/docs/4.1/layout/grid/#column-breaks>

.

Cómo romper una columna (salto)

- **Ejemplo:** cosasNuevas/rejillasPruebas3.html.

```
<div class="row">

  <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
  <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

  <!-- Fuerza que la siguiente columna esté en una nueva línea -->
  <div class="w-100"></div>

  <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
  <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

</div>
```

.col-6 .col-sm-3	.col-6 .col-sm-3
.col-6 .col-sm-3	.col-6 .col-sm-3

Cómo mover columnas (offseting)

Podemos desplazar las columnas de dos formas en bootstrap: la que veremos aquí es usando las clases propias del grid **.offset-**, la otra es utilizando las [utilidades de los márgenes](#) (← [pinchar en el enlace](#)).

Para mover las columnas a la derecha usaremos **.offset-md-***, con esto moveremos la columna * posiciones cuando el tamaño de la pantalla sea media. Veamos un ejemplo en el fichero

[cosasNuevas/rejillasPruebas4.html](#)

Cómo mover columnas (offseting)

Para más información:

<https://getbootstrap.com/docs/4.1/layout/grid/#offset-classes>

`.col-md-4`

`.col-md-4 .offset-md-4`

`.col-md-3 .offset-md-3`

`.col-md-3 .offset-md-3`

`.col-md-6 .offset-md-3`

Media Queries (media object?)

En la mayoría de los casos con las clases que vienen en Bootstrap nos es suficiente para diseñar nuestra web, sin embargo, en algunas situaciones a veces queremos modificar cierto comportamiento como puede ser para aplicar colores, alineación, etc. que cambie dependiendo del tamaño de la pantalla. Será en estos casos cuando utilizemos los media queries.

<https://getbootstrap.com/docs/4.1/layout/overview/#responsive-breakpoints>

Media Queries (media object?)

Una media query se define de la siguiente forma:

```
@media (min-width: TAMAÑO-EN-PÍXELES) {  
    /* Estos estilos sólo se aplicarán a partir del  
    tamaño indicado */  
}
```

En este caso, los estilos que estén dentro de esta media query se aplicarán sólo a partir del tamaño indicado, pero también le podemos indicar el rango de tamaño al que aplicarse.



Media Queries (media object?)

```
@media (max-width: TAMAÑO-EN-PÍXELES) {  
    /* Estos estilos solo se aplicarán HASTA el tamaño  
    indicado */  
}
```

```
@media (min-width: TAMAÑO-EN-PÍXELES) and (max-  
width: TAMAÑO-EN-PÍXELES){ /* Solo se aplicarán  
ENTRE los tamaños indicados */  
}
```

Media Queries: Ejemplos de uso

Para pantallas extra pequeñas, le aplicaremos un color de fondo llamado (myBackground) que será rojo, y para el resto de tamaños el fondo será verde:

```
myBackground {  
  background-color:  green;  
}  
  
@media (max-width: 768px) {  
  .myBackground {  
    background-color:  red;  
  }  
}
```

Media Queries: Ejemplos de uso

Otro ejemplo podría ser la alineación del texto que se le aplicará cuando estemos en pantallas de escritorio:

```
.miestilo {  
  text-align: center;  
}  
@media (min-width: 992px) {  
  .miestilo {  
    text-align: left;  
  }  
}
```

Media Queries: Ejemplos de uso

Estos estilos son un ejemplo básico de lo que podemos hacer complementando el código de Bootstrap. Otros ejemplos comunes son:

- Cambiar el tamaño y posición de una imagen entre dispositivos móviles y de escritorio
- Cambiar la posición de un elemento.
- Cambiar el tamaño de letra, fuente o color dependiendo del tamaño de la pantalla.
- Aplicar combinaciones de estilos para que su apariencia se ajuste a los elementos que contenga.
- etc.

Bibliografía

Página oficial desde donde poder consultar toda la documentación y poder descargar la librería: <http://getbootstrap.com>

Blog oficial de Bootstrap donde consultar las últimas novedades: <http://blog.getbootstrap.com/>

Temas y plantillas gratuitas para Bootstrap:

o <http://startbootstrap.com> o <http://bootstrapzero.com> o <http://bootswatch.com> o <http://www.bootbundle.com>

Ejemplos y trozos de código útiles para Bootstrap: <http://bootsnipp.com>

Ejemplos de uso: <http://expo.getbootstrap.com/>

Listado de los recursos disponibles: <https://startbootstrap.com/bootstrap-resources/>