UD5 Bootstrap y SASS.

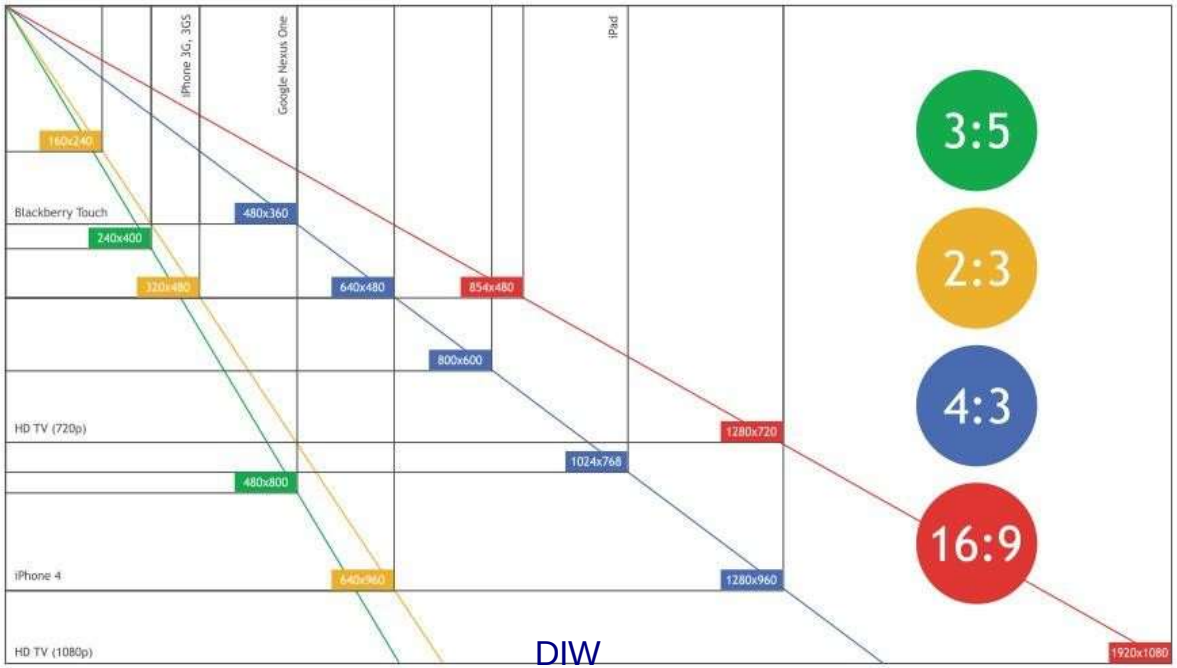
DISEÑO DE INTERFACES WEB

UNIDAD 5.1

Introducción a Bootstrap
Instalación
Creación de una página básica
Sistema de rejilla
Media Query

Introducción

El diseño web responsive (RWD Responsive Web Design) es **adaptar la apariencia de las páginas web al dispositivo** que se esté utilizando para visualizarla. Hoy en día las páginas web se visualizan en multitud de tipos de dispositivos como tablets, teléfonos, portátiles, etc. Esta tecnología pretende que con **un solo diseño** web tengamos una web adecuada para cualquier dispositivo.



Introducción

A la hora de crear una web, debemos tener en cuenta a qué usuarios va destinados para poder analizar su **usabilidad**, ya que ya no podemos centrarnos en desarrollar una web pensando que los usuarios van a tener probablemente una única resolución de pantalla.

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

En la actualidad 1024x768 ya no es la resolución más utilizada, sino 1366x768 y resoluciones superiores a 1920x1080. Es fundamental tener en cuenta que en el diseño responsive, al variar tanto las posibles resoluciones debemos mostrar primero los contenidos más importantes.

Frameworks responsive

¿Por qué reinventar la rueda si podemos usar algunos de los frameworks que existen en el mercado para ello? Ahorremos tiempo, usando **código ya probado y unos diseños** más bonitos que los que podemos encontrar de forma nativa. :

- **Bootstrap**: Uno de los frameworks **más populares** del mercado y fue 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.

Descarga e instalación

Para **descargar Bootstrap** (no es un instalador), debemos descargarnos el zip que se encuentra en la web oficial. Una vez descargado, lo descomprimimos 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 de la siguiente manera.

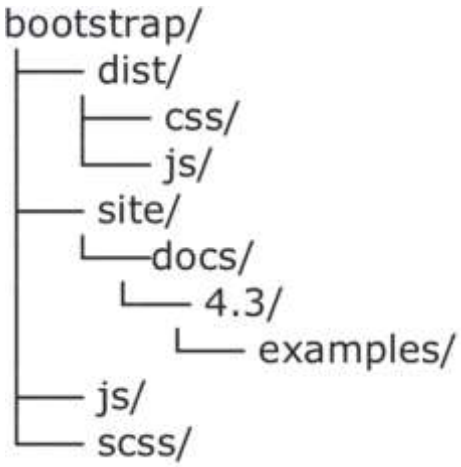
```
<link rel="stylesheet" href="../bootstrap-4.4.1/dist/css/bootstrap.min.css" type="text/css">
```

Si no queremos tener descargado el archivo, y preferimos que la web se conecte al servidor de Bootstrap deberemos poner antes de nuestros CSS:

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
```

Contenido del directorio

Una vez que hemos descargado y descomprimido el archivo tenemos dentro algo parecido a:



- scss/ y js/ contiene el código de nuestro CSS y JavaScript.
- Dist/ incluye todo lo que aparece en la sección precompilada.
- site/docs/ tiene el código fuente de la documentación
- examples/ ejemplos de cómo usar Bootstrap.

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:

```
<!DOCTYPE html>
<html>
...
</html>
```

```
<!DOCTYPE html>
<html lang="es">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title>Mi página web</title>
</head>
<body>
   ....
```

Además, para asegurarnos que todo se está visualizando de forma correcta y que podemos utilizar el zoom en los dispositivos móviles, añadimos dentro del **<head>**:

Indicamos al navegador que el ancho del viewport debe ser igual al ancho del dispositivo. La página se ajustará automáticamente al ancho de la pantalla del dispositivo. Es la parte más importante para el diseño responsivo.

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

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 de Bootstrap

Se basa en rejillas flexibles, que escalan su tamaño y posición dependiendo de la pantalla en que visualice la web.

Bootstrap comes with three different containers:

- container, which sets a max-width at each responsive breakpoint
- container-{breakpoint}, which is width: 100% until the specified breakpoint
- container-fluid, which is width: 100% at all breakpoints

Elemento contenedor

El sistema de rejilla tiene que ser utilizado dentro de uno de los dos elementos contenedores que tiene Bootstrap:

- `.container` Para que el contenido de nuestra web aparezca **centrado,** con un ancho fijo, y márgenes automáticos.
- `.container-fluid` Para que ocupe todo el ancho disponible (width: 100%)

Es importante tener en cuenta que estos elementos se utilizan como raíz de la rejilla

(Ver ejemplo rejillas1.html y rejillas2.html)

Funcionamiento del sistema rejilla

El funcionamiento del sistema rejilla es el siguiente:

- Las columnas deben ir agrupadas dentro de filas (**.row**)
- A continuación, 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 permite 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**)
- Al seguir este orden el sistema de rejilla funcionará creando el espaciado interior y los márgenes apropiados dependiendo de las dimensiones de la pantalla.

Funcionamiento del sistema rejilla

- 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.
- 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.

En la siguiente tabla se muestra un resumen del sistema de rejilla de Bootstrap, su comportamiento según el tamaño del dispositivo y las clases CSS que nos permiten controlarlo:

Funcionamiento del sistema rejilla

Pantalla	Dimensiones	Prefijo de la clase	Ancho del contenedor
Tamaño extra pequeño	< 576 px	**.col-**	Ninguno (automático)
Tamaño pequeño	≥ 576 px	.col-sm-	540px
Tamaño medio	≥ 768 px	.col-md-	720px
Tamaño grande	≥ 992 px	.col-lg-	960px
Tamaño extra grande	≥ 1200 px	.col-xl-	1140px

Nota: Al definir estas clases, no sólo se aplican para ese tamaño de pantalla, sino que para las superiores también salvo que los definas. En ese caso se se sobrescriben y pasan a utilizar los definidos para dichos tamaños.

Ejemplos:

rejillas3.html

Mismo ancho: son dos filas con un número diferente de celdas, y en todos los dispositivos se ve igual.

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

Código:

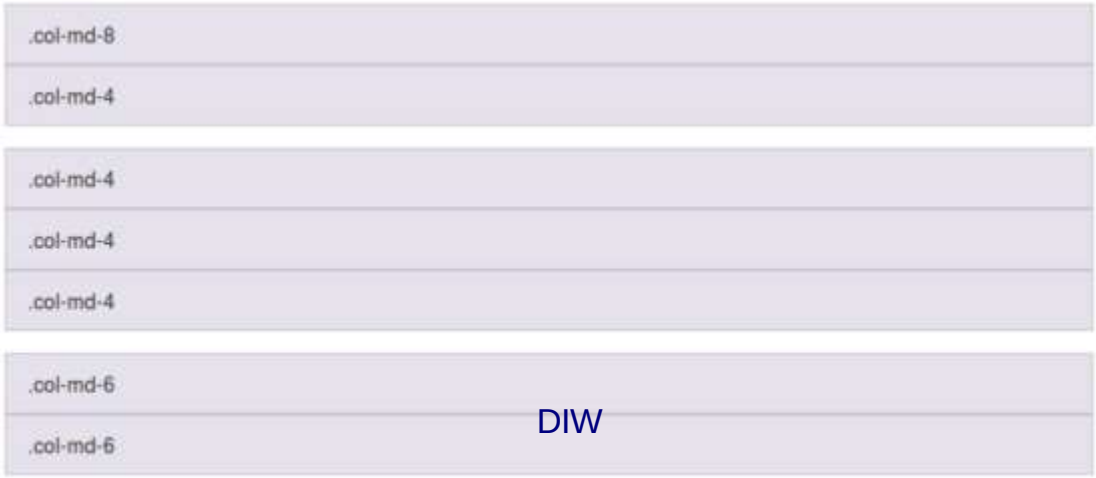
```
<div class="container">
 <div class="row">
  <div class="col">
   1 of 2
  </div>
  <div class="col">
   2 of 2
  </div>
 </div>
```

```
<div class="row">
 <div class="col">
  1 of 3
 </div>
 <div class="col">
  2 of 3
 </div>
 <div class="col">
  3 of 3
 </div>
</div>
</div>
```

rejillas4.html

Tamaño de columnas solo para pantallas de 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

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 ampliaran utilizando todo el ancho por lo que se mostrarán de la siguiente forma:



rejillas6.html

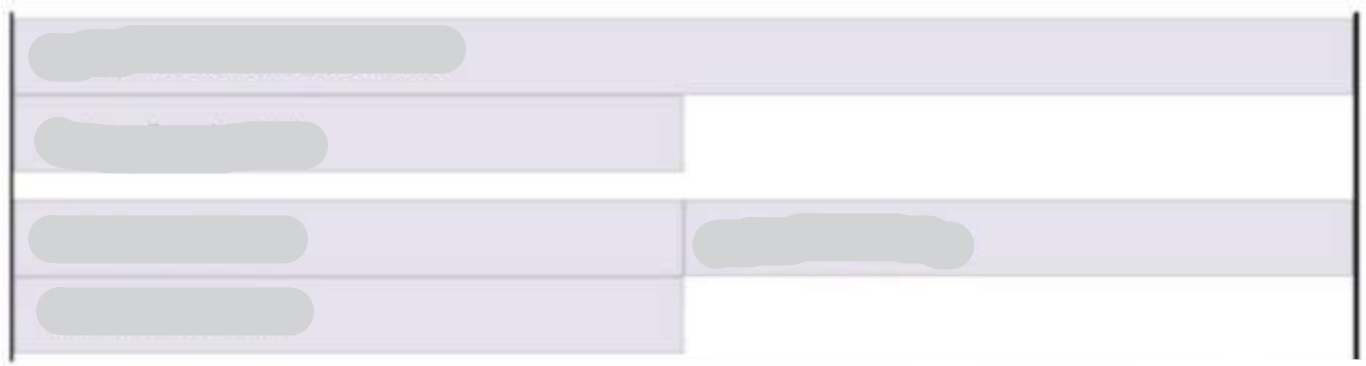
Tamaño para móvil, tablet y escritorio:

Para tener un mayor control, podemos especificar el tamaño de las columnas para las tablets, con las clases: **.col-sm-***.

Resultado en pantallas medianas y grandes:

Resultado en pantallas de tipo tablet:

Resultado en pantallas pequeñas:



Cómo ocultar celdas u otro elemento

Cuando queremos que **el contenido de la página sea menor** en función de si se visualiza en el móvil o en el ordenador, bootstrap tiene la clase `.d-none` para ocultarlo para todas las pantallas o `.d-{sm,md,lg,xl}-none` en alguna resolución en concreto.

Para **mostrar un elemento** sólo en una resolución, puedes combinar una clase `.d-none` con otra `.d-block` (muestra el elemento), por ejemplo `.d-none .d-md-block`

```
<div class="d-lg-none">hide on screens wider than lg</div>
<div class="d-none d-lg-block">hide on screens smaller than lg</div>
```

En la siguiente tabla se muestran todos los nombres de clases.

Cómo ocultar celdas u otro elemento

Usa .visually-hidden-focusable para ocultar visualmente un elemento de forma predeterminada, pero mostrarlo cuando está enfocado (por ejemplo, por un usuario que solo usa el teclado). .visually-hidden-focusable también se puede aplicar a un contenedor, gracias a :focus-within, el contenedor se mostrará cuando cualquier elemento secundario del contenedor reciba el foco.

```
<h2 class="visually-hidden">Título para lectores de pantalla</h2>
<a class="visually-hidden-focusable" href="#content">Saltar al contenido principal</a> <div class="visually-hidden-focusable">Un contenedor con un <a href="#">elemento enfocable</a>.</div>
```

Cómo ocultar celdas u otro elemento

Screen Size	Class
Hidden on all	`.d-none`
Hidden only on xs	`.d-none .d-sm-block`
Hidden only on sm	`.d-sm-none .d-md-block`
Hidden only on md	`.d-md-none .d-lg-block`
Hidden only on lg	`.d-lg-none .d-xl-block`
Hidden only on xl	`.d-xl-none`
Visible on all	`.d-block`
Visible only on xs	`.d-block .d-sm-none`
Visible only on sm	`.d-none .d-sm-block .d-md-none`
Visible only on md	`.d-none .d-md-block .d-lg-none`
Visible only on lg	`.d-none .d-lg-block .d-xl-none`
Visible only on xl	`.d-none .d-xl-block`

Anidamiento de columnas (nesting)

Con Bootstrap también podemos anidar columnas. Por ejemplo, si quisiéramos que en una columna de tamaño 9, crear una nueva fila y dividirla como queramos (cada fila máximo 12 celdas)

Código:

```
<div class="col-sm-9">
  Level 1: .col-sm-9
  <div class="row">
    <div class="col-xs-8 col-sm-6">
      Level 2: .col-xs-8 .col-sm-6
    </div>
    <div class="col-xs-4 col-sm-6">
      Level 2: .col-xs-4 .col-sm-6
    </div>
  </div>
</div>
```

Resultado:

Márgenes o espaciado entre columnas (margin utilities)

En la versión 5 de Bootstrap, Las clases se nombran usando el formato {property}{sides}-{size} para xs y {property}{sides}-{breakpoint}-{size} para sm, md, lg, xl y xxl.

Donde property es uno de:
m - para clases que establecen margin
p - para clases que establecen padding

g - gutter (ancho del espacio entre columnas)

Donde *sides* es uno de:

- t - para clases que establecen margin-top o padding-top
- b - para clases que establecen margin-bottom o padding-bottom
- s - (start) para clases que establecen margin-left o padding-left en LTR, margin-right o padding-right en RTL

left to right — right to left

- e - (end) para clases que establecen margin-right o padding-right en LTR, margin-left o padding-left en RTL
- x - para clases que establecen tanto *-left como *-right
- y - para clases que establecen tanto *-top como *-bottom
- en blanco - para clases que establecen un margin o padding en los 4 lados del elemento

Márgenes o espaciado entre columnas (margin utilities)

Donde *size* es uno de:

- 0 - para clases que eliminan el margin o padding al establecerlo en 0
- 1 - (por defecto) para las clases que establecen el margin o padding en $spacer * .25
- 2 - (por defecto) para las clases que establecen el margin o padding en $spacer * .5
- 3 - (por defecto) para las clases que establecen el margin o padding en $spacer
- 4 - (por defecto) para las clases que establecen el margin o padding en $spacer * 1.5
- 5 - (por defecto) para las clases que establecen el margin o padding en $spacer * 3
- auto - para clases que establecen el margin en automático

```
.mt-0 {
    margin-top: 0 !important;
}

.ms-1 {
   margin-left: ($spacer * .25)
important;
}
```

```
.px-2 {
  padding-left: ($spacer * .5)
!important;
  padding-right: ($spacer * .5)
!important;
 }

.p-3 {
  padding: $spacer !important;
}
```

Ordenación de columnas

Podemos usar la clase `.order-` para controlar de forma visual el orden de nuestro contenido. Estas clases son responsive, por tanto, podemos establecer el orden 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` (ver **rejillas7.html**).

Código:

```
<div class="container">
  <div class="row">
    <div class="col">
      First, but unordered
    </div>
    <div class="col order-12">
      Second, but last
    </div>
    <div class="col order-1">
      Third, but first
    </div>
  </div>
</div>
```

Resultado:

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

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`).

Código:

```
<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>
```

Resultado:

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

Cómo mover columnas (offsetting)

Una de las formas que tenemos en bootstrap para mover columnas es usando las clases propias del grid `.offset-`. Por ejemplo, con `.offset-md-*` movemos la columna * posiciones cuando el tamaño de la pantalla sea media. Veamos un ejemplo

Código: https://easyhtml5video.com/articles/bootstrap-offset-tutorial-978.html

```
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>
</div>
<div class="row">
  <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
  <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
</div>
<div class="row">
  <div class="col-md-6 offset-md-3">.col-md-6 .offset-md-3</div>
</div>
```

Resultado:

Media queries (Media object)

Como ya vimos en el apartado de CSS, a veces queremos modificar cierto comportamiento como colores, alineación, etc. **dependiendo del tamaño de la pantalla**. Será en estos casos cuando utilicemos los media queries. Por ejemplo:

REALMENTE DEBERÍA SER USANDO BOOTSTRAP MEDIA QUERY breakpoints

@include media-breakpoint-up(sm)
@include media-breakpoint-down(sm)
@include media-breakpoint-only(xs)

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

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 **(1)**.

Otro ejemplo podría ser la alineación del texto que se le aplicará cuando estemos en pantallas de escritorio **(2)**.

(1)

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

(2)

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