

Tema 3: CSS avanzado



Ciclo Superior DAW

Asignatura: Diseño de interfaces

Curso 20/21

Introducción



- En este capítulo veremos los siguientes conceptos:
 - Bootstrap
 - Utilizar frameworks de estilos
 - Utilizar preprocesadores de hojas de estilos

Bootstrap



- **Framework HTML, CSS y JavaScript que permite aplicar diseño a páginas de manera simple, elegante y responsiva**
- Lograr un buen diseño y que además funcione en todas los dispositivos y navegadores existentes es una ardua tarea por mucho que los lenguajes sean estándar.



Bootstrap

- Liberado y desarrollado por Twitter
- Permite ahorrar muchas horas de trabajo ajustando estilos
- Proporciona una hoja de estilos cuyas clases CSS se puede aplicar en una página.
- También se pueden incluir componentes JavaScript desarrollados sobre jQuery listos para ser utilizados.

Bootstrap



- Si se trabaja con Bootstrap se deben tratar de utilizar las propias clases y recursos que ofrece, en lugar de tratar de duplicar o crear nuevos estilos para poder beneficiarse al máximo de esta herramienta.
- En cualquier caso, siempre se podrán adaptar o extender según las necesidades específicas.



Bootstrap. Primeros pasos

Para trabajar con Bootstrap tenemos varias opciones:

- Descargar el código fuente CSS y meter los ficheros de Bootstrap en una carpeta del proyecto
- Utilizar **BootstrapCDN**



Bootstrap. Primeros pasos

Descargar el código CSS y JavaScript compilado

- Es la forma más sencilla de empezar a utilizar Bootstrap.
- La desventaja es que esta versión no incluye ni los archivos originales ni la documentación.



Bootstrap. Primeros pasos

Descargar el código CSS y JavaScript compilado

- Debemos descargar un archivo CSS del sitio <http://getbootstrap.com>
- Debemos enlazar a nuestra página con el archivo **bootstrap.min.css** y ya podemos utilizar esta librería.



Bootstrap. Primeros pasos

Descargar el código CSS y JavaScript compilado

```
<link href="css/bootstrap.min.css" rel="stylesheet">
```



Bootstrap. Primeros pasos

Utilizar BootstrapCDN

- Consiste en utilizar un servidor donde se alojan todos los archivos de Bootstrap.
- En lugar de cargar el archivo "bootstrap.min.css" de nuestro servidor, lo hacemos del servidor "**<https://maxcdn.bootstrapcdn.com>**"



Bootstrap. Primeros pasos

Utilizar BootstrapCDN

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
```



Bootstrap. Primeros pasos

Utilizar BootstrapCDN

- Ventajas:
 - Reduce la carga de los servidores
 - Facilita que Bootstrap 4 quede en caché del navegador y se vuelva a cargar cada vez que se visita una web que lo utilice



Bootstrap. Primeros pasos

Utilizar BootstrapCDN

- Ventajas:
 - Reduce la latencia.
 - Reducción de costos de nuestros servidores por requerir menor ancho de banda y tráfico.



Bootstrap. Primeros pasos

Utilizar BootstrapCDN

- Desventajas:
 - El servidor donde se aloja Bootstrap 4 se puede caer.
 - **IMPORTANTE:** No podemos realizar pruebas de forma local si no hay conexión a Internet.

Actividad 1



- 1. Crea una web descargando el código fuente CSS*
- 2. Crea una web utilizando BootstrapCDN*



Bootstrap. Fichero mínimo

- Veamos un ejemplo de página web simple:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Plantilla básica Bootstrap</title>
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hola Mundo!</h1>
    <p> Mi primera página en Bootstrap </p>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```




Bootstrap. Estructura

- La **estructura en Bootstrap** consiste en un **conjunto de bloques que empiezan por un contenedor**, que es una clase que se aplica a un bloque que contiene todo el documento.
- El contenedor puede ser de tipo:
 - **container**: sitúa todo el contenido en el centro de la página y siempre tendrá el mismo ancho
 - **container-fluid**: hace que el contenido siempre ocupe el 100% de la página



Bootstrap. Estructura

- Las **columnas** son la clave de la maquetación con Bootstrap.
- Organizar la página en columnas, de forma variable o anidada según zonas de la misma.
- Todo eso permite maquetar sin utilizar tablas o utilizar complejos divs flotantes.

Bootstrap. Estructura



- La página puede tener un total de 12 columnas. A partir de ahí se pueden utilizar columnas que ocupen 1, 2, 3 o crear distintas combinaciones de columnas, que siempre sumen 12.



Bootstrap. Estructura

- Hay que tener en cuenta los tamaños de pantallas que Bootstrap clasifica en:

Prefijo	Nombre	Tipo de dispositivo	Píxeles
xs	Extra-small	Móviles	< 768px
sm	Small	Tablets	> 768px y < 992px
md	Medium	Pantallas pequeñas	> 992px y < 1200px
lg	Large	Pantallas grandes	> 1200px

- Al crear las columnas previamente se indica el tipo de pantalla y, de esa manera, las columnas se ajustarán de acuerdo a la misma.



Bootstrap. Estructura

- En resumen:
 - Clase “**col-lg-***”: La columna colapsará cuando el ancho del navegador < 992px
 - Clase “**col-md-***”: La columna colapsará cuando el ancho del navegador < 720px
 - Clase “**col-sm-***”: La columna colapsará cuando el ancho del navegador < 540px
 - Clase “**col-***”: La columna nunca colapsa



Bootstrap. Estructura. Ejemplos

Página a dos columnas

- Para lograr este efecto, se define una fila, y dentro de ella se mete 2 bloques
- **HTML:**

```
<h1>
  Ejemplo Bootstrap 2 columnas
</h1>
<div class="container">
  <div class="row">
    <div class="col-md-6">
      <span>Columna<br>1 (tamaño 6)</span>
    </div>
    <div class="col-md-6">
      <span>Columna<br>2 (tamaño 6)</span>
    </div>
  </div>
</div>
```



Bootstrap. Estructura. Ejemplos

Página a dos columnas

- CSS:

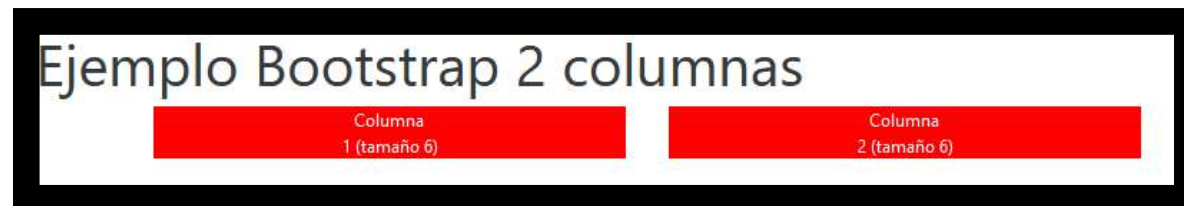
```
span {  
  display: block;  
  width: 100%;  
  font-size: 12px;  
  color: #fff;  
  text-align: center;  
  background-color: #f00;  
  margin-bottom: 5px;  
}
```



Bootstrap. Estructura. Ejemplos

Página a dos columnas

- **Resultado:**





Bootstrap. Estructura. Ejemplos

Filas y columnas variadas

- En este caso, se crean tres filas en las que las columnas son diferentes.
- La primera fila tiene una columna, la siguiente tiene dos y la última tiene tres.



Bootstrap. Estructura. Ejemplos

- HTML:

```
<h1>
  Ejemplo Bootstrap filas y columnas variadas
</h1>
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <span>Columna<br>1 (tamaño 12)</span>
    </div>
  </div>
  <div class="row">
    <div class="col-md-9">
      <span>Columna<br>1 (tamaño 9)</span>
    </div>
    <div class="col-md-3">
      <span>Columna<br>2 (tamaño 3)</span>
    </div>
  </div>
  <div class="row">
    <div class="col-md-4">
      <span>Columna<br>1 (tamaño 4)</span>
    </div>
    <div class="col-md-4">
      <span>Columna<br>2 (tamaño 4)</span>
    </div>
    <div class="col-md-4">
      <span>Columna<br>3 (tamaño 4)</span>
    </div>
  </div>
</div>
```



Bootstrap. Estructura. Ejemplos

Filas y columnas variadas

- **CSS:**

```
span {  
  display: block;  
  width: 100%;  
  font-size: 12px;  
  color: #fff;  
  text-align: center;  
  background-color: #f00;  
  margin-bottom: 5px;  
}
```



Bootstrap. Estructura. Ejemplos

Filas y columnas variadas

- **Resultado:**



Actividad 2



Reconstruye el ejemplo anterior para que, en vez de 2 columnas, sean 3 las columnas utilizadas



Bootstrap. Componentes

- Algunos de los componentes de Bootstrap son:
 - Paneles
 - Wells
 - Button
 - Glyphicons
 - Barra de navegación
 - Tables
 - Badges
 - Paginación



Bootstrap. Componentes. Paneles

- **Paneles:**

- **NO DISPONIBLE EN BOOTSTRAP 4**
- En ocasiones es necesario encerrar cierto contenido dentro de una caja. En estos casos se debería considerar el uso del componente para paneles. Por ejemplo:

```
<div class="panel panel-default">  
  <div class="panel-body">  
    Ejemplo de panel muy sencillo  
  </div>  
</div>
```

Ejemplo de panel muy sencillo



Bootstrap. Componentes. Wells

- **Wells:**
 - **NO DISPONIBLE EN BOOTSTRAP 4**
 - El estilo por defecto de este componente simplemente aplica un efecto de tipo hundido (en inglés, "inset") al elemento. Ejemplo:

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

¡Este texto está dentro de un "pozo" y por eso parece que está hundido!



Bootstrap. Componentes. Button

- **Button:**
 - Existen distintos botones según tamaño:

```
<button type="submit" class="btn btn-primary btn-sm">Pequeño</button>  
<button type="submit" class="btn btn-primary btn-lg">Grande</button>  
<button type="submit" class="btn btn-primary">Normal</button>
```





Bootstrap. Componentes. Button

- **Button:**

- Existen distintos botones según tamaño:

```
<button type="submit" class="btn btn-primary btn-sm btn-block">En bloque: Pequeño</button>  
<button type="submit" class="btn btn-primary btn-lg btn-block">En bloque: Grande</button>  
<button type="submit" class="btn btn-primary btn-block">En bloque: Normal</button>
```

En bloque: Pequeño

En bloque: Grande

En bloque: Normal



Bootstrap. Componentes. Button

- **Button:**

- Algunas de sus opciones de personalización son:
 - **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
 - **btn-info** : Es un botón para información.
 - **btn-warning** : Precaución con la acción que tiene asociada el botón



Bootstrap. Componentes. Button

- **Button:**
 - Algunas de sus opciones de personalización son:

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.

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.



Bootstrap. Componentes. Button

- **Button:**

- Algunas de sus opciones de personalización son:
 - **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.



Bootstrap. Componentes. Button

- **Button:**
 - Algunas de sus opciones de personalización son:

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.



Bootstrap. Componentes. Button

- **Grupos de botones:**

- Bootstrap permite agrupar varios botones relacionados entre sí para mostrarlos en una única

línea.. Ejemplo:

```
<div class="btn-group">  
  <button type="button" class="btn btn-outline-secondary">Suspenso</button>  
  <button type="button" class="btn btn-outline-secondary">Aprobado</button>  
  <button type="button" class="btn btn-outline-secondary">Notable</button>  
  <button type="button" class="btn btn-outline-secondary">Sobresaliente</button>  
</div>
```





Bootstrap. Componentes. Glyphicons

- **Glyphicons:**
 - **NO DISPONIBLE EN BOOTSTRAP 4**
 - Bootstrap incluye 180 iconos creados mediante una fuente especial llamada **Glyphicon Halflings**.
 - Esta fuente normalmente no es gratuita, pero su creador permite utilizar estos iconos gratuitamente dentro de Bootstrap.
 - Todos los iconos en: <https://glyphicons.com/>



Bootstrap. Componentes. Glyphicons

- Glyphicons:

```
<div class="btn-toolbar" role="toolbar">
  <div class="btn-group">
    <button type="button" class="btn btn-default">
      <span class="glyphicon glyphicon-align-left"></span>
    </button>

    <button type="button" class="btn btn-default">
      <span class="glyphicon glyphicon-align-center"></span>
    </button>

    <button type="button" class="btn btn-default">
      <span class="glyphicon glyphicon-align-right"></span>
    </button>

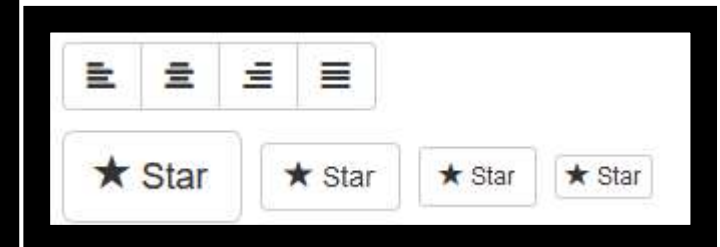
    <button type="button" class="btn btn-default">
      <span class="glyphicon glyphicon-align-justify"></span>
    </button>
  </div>
</div>

<div class="btn-toolbar" role="toolbar">
  <button type="button" class="btn btn-default btn-lg">
    <span class="glyphicon glyphicon-star"></span> Star
  </button>

  <button type="button" class="btn btn-default">
    <span class="glyphicon glyphicon-star"></span> Star
  </button>

  <button type="button" class="btn btn-default btn-sm">
    <span class="glyphicon glyphicon-star"></span> Star
  </button>

  <button type="button" class="btn btn-default btn-xs">
    <span class="glyphicon glyphicon-star"></span> Star
  </button>
</div>
```



Bootstrap. Componentes. Barra de navegación



- **Barra de navegación:**

- Son **componentes adaptados al diseño web responsive** y que se utilizan como **elemento principal de navegación** tanto en las aplicaciones como en los sitios web.
- En los dispositivos móviles **se muestran inicialmente minimizadas** y al pulsar sobre ellas, se despliegan todas sus opciones.
- Dependiendo del tamaño, puede ser necesario modificar el punto a partir del cual la barra se muestra horizontal en vez de minimizada. Lo podemos conseguir con una media-query o modificando la variable **@grid-float-breakpoint** en el archivo LESS o SASS

Bootstrap. Componentes. Barra de navegación



- Barra de navegación:

```
<nav class="navbar navbar-expand-lg">
  <a class="navbar-brand" href="#">Ciclos Montecastelo</a>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Personal</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Instalaciones</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Contacto</a>
      </li>
    </ul>
  </div>
</nav>
```

Ciclos Montecastelo Home Personal Instalaciones Contacto

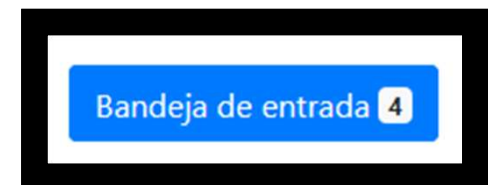


Bootstrap. Componentes. Badges

- **Badges:**

- Se utilizan para destacar elementos nuevos o que no han sido leídos.
- Se pueden añadir a los enlaces, los elementos de navegación de Bootstrap y muchos otros tipos de elementos mediante la etiqueta ``.

```
<button type="button" class="btn btn-primary">  
|   Bandeja de entrada <span class="badge badge-light">4</span>  
</button>
```





Bootstrap. Componentes. Table

- **Table:**

- En Bootstrap las tablas tienen un papel muy importante, y tiene muchas opciones de personalización.
- Se compone de:
 - **Cabecera:** `<thead>` y `<th>`
 - **Filas:** `<tbody>` y `<tr>`
 - **Celdas:** `<tbody>` y `<td>`



Bootstrap. Componentes. Table

- **Table:**

- Ejemplo:

Titulo 1	Titulo 2
Contenido 1	Contenido 2
Contenido 1	Contenido 2

```
<table class="table">
  <thead>
    <tr>
      <th>Titulo 1</th>
      <th>Titulo 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Contenido 1</td>
      <td>Contenido 2</td>
    </tr>
    <tr>
      <td>Contenido 1</td>
      <td>Contenido 2</td>
    </tr>
  </tbody>
</table>
```



Bootstrap. Componentes. Table

- **Table:**

- Algunas de sus opciones de personalización son:

- **Modo oscuro:**

```
<table class="table table-dark">
```

- **Colores alternos:**

```
<table class="table table-striped">
```

- **Celdas bordeadas:**

```
<table class="table table-bordered">
```

- **Celdas con hover:**

```
<table class="table table-hover">
```

- **Formato a la cabecera:**

```
<thead class="table-dark">
```



Bootstrap. Componentes. Table

- **Table:**

- Algunas de sus opciones de personalización son:
 - Aplicar clases a los colores de fondo de una fila o celda de una tabla:
 - **table-danger** (Representa una acción peligrosa)
 - **table-warning** (Representa una advertencia)
 - **table-info** (Representa una información)
 - **table-success** (Representa algo exitoso)



Bootstrap. Componentes. Table

- **Table:**

- Algunas de sus opciones de personalización son:
 - Aplicar clases a los colores de fondo de una fila o celda de una tabla:
 - **table-primary** (Representa algo importante)
 - **table-secondary** (Representa algo no tan importante)
 - **table-dark**
 - **table-light**
 - **table-active**



Bootstrap. Componentes. Imagen

- En Bootstrap tenemos las siguientes clases para aplicar al elemento ``:
 - **rounded**: Define las esquinas redondeadas en la imagen.
 - **rounded-circle**: Convierte la imagen en un círculo, para ello Bootstrap 4 inicializa la propiedad **border-radius: 50%**
 - **img-thumbnail**: Define un recuadro y muestra la imagen cuando se carga la página en el navegador



Bootstrap. Componentes. Imagen

- En Bootstrap tenemos las siguientes clases para aplicar al elemento ``:
 - **img-fluid**: Permite escalar correctamente la imagen (Esto lo logra Bootstrap 4 definiendo la propiedad `max-width: 100%;` y `height: auto;`)
 - **float-left**: flota la imagen a la izquierda.
 - **float-right**: flota la imagen a la derecha.



Bootstrap. Componentes. Imagen

- Por ejemplo:

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





Bootstrap. Componentes. Jumbotron

- El componente Jumbotron nos sirve para presentar un título y un contenido de forma destacada en una página
- En Bootstrap, podemos aplicarlo a un **<div>** con la **clase jumbotron**
- Si queremos que ocupe todo el ancho de la página, **debemos utilizar la clase jumbotron-fluid**



Bootstrap. Componentes. Jumbotron

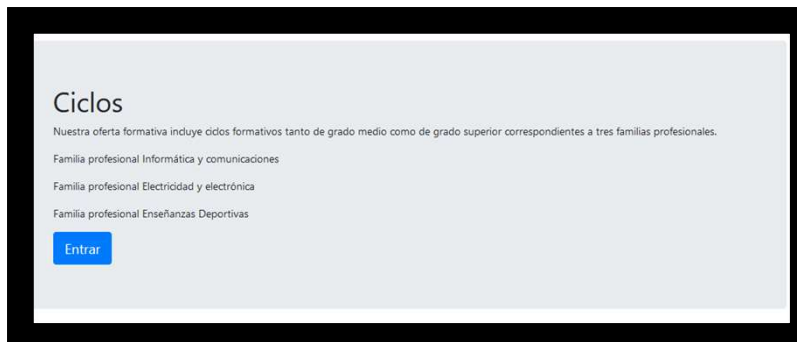
- Por ejemplo:

```
<div class="jumbotron">
  <h1>Ciclos</h1>

  <p>Nuestra oferta formativa incluye ciclos formativos tanto de grado medio
  como de grado superior correspondientes a tres familias profesionales.</p>

  <p>Familia profesional Informática y comunicaciones</p>
  <p>Familia profesional Electricidad y electrónica</p>
  <p>Familia profesional Enseñanzas Deportivas</p>

  <a href="#" class="btn btn-primary btn-lg">Entrar</a>
</div>
```





Bootstrap. Componentes. Tarjeta

- El componente Card es un contenedor de información flexible.
- Podemos aplicarlo a un **<div>** para representar varios tipos de formatos.
- Tenéis todas las opciones en el siguiente link:

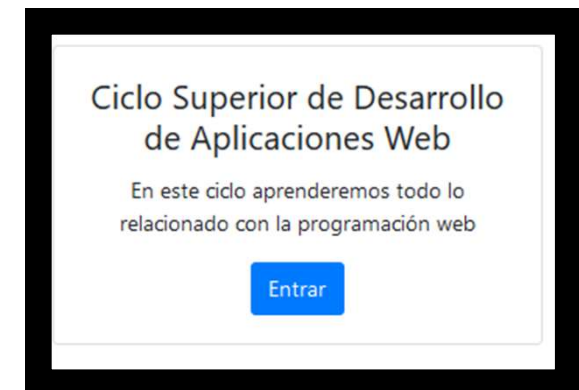
<https://getbootstrap.com/docs/4.0/components/card/>



Bootstrap. Componentes. Tarjeta

- Algunos de las clases con las que vamos a trabajar son:
 - **Tarjeta básica:**

```
<div class="col-md">
  <div class="card text-center">
    <div class="card-body">
      <h4 class="card-title">Ciclo Superior de Desarrollo de Aplicaciones Web</h4>
      <p class="card-text">
        En este ciclo aprenderemos todo lo relacionado con la programación web
      </p>
      <a href="#" class="btn btn-primary">Entrar</a>
    </div>
  </div>
</div>
```





Bootstrap. Componentes. Tarjeta

- Algunos de las clases con las que vamos a trabajar son:
 - **Tarjeta con cabecera y pie:**

```
<div class="col-md">
  <div class="card text-center">
    <div class="card-header">
      Ciclo Medio
    </div>
    <div class="card-body">
      <h4 class="card-title">Ciclo Medio de Sistemas Microinformáticos y Redes</h4>
      <p class="card-text">
        En este ciclo aprenderemos todo lo relacionado con sistemas microinformáticos
      </p>
      <a href="#" class="btn btn-primary">Entrar</a>
    </div>
    <div class="card-footer">
      Curso 20/21
    </div>
  </div>
</div>
```

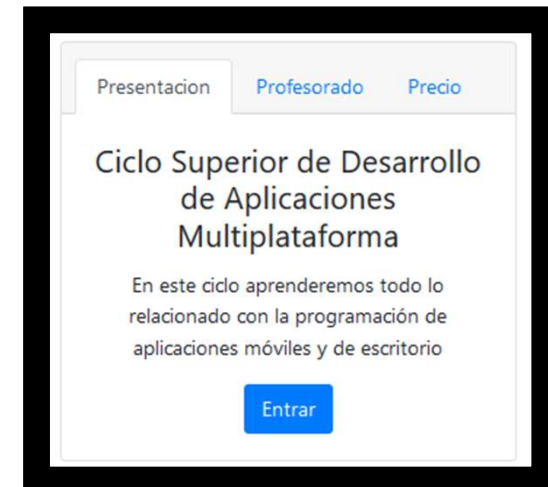




Bootstrap. Componentes. Tarjeta

- Algunos de las clases con las que vamos a trabajar son:
 - **Tarjeta con pestañas:**

```
<div class="card text-center">
  <div class="card-header">
    <ul class="nav nav-tabs card-header-tabs">
      <li class="nav-item">
        <a class="nav-link active" href="#">Presentacion</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Profesorado</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Precio</a>
      </li>
    </ul>
  </div>
  <div class="card-body">
    <h4 class="card-title">Ciclo Superior de Desarrollo de Aplicaciones Multiplataforma</h4>
    <p class="card-text">
      En este ciclo aprenderemos todo lo relacionado con
      la programación de aplicaciones móviles y de escritorio
    </p>
    <a href="#" class="btn btn-primary">Entrar</a>
  </div>
</div>
```





Bootstrap. Componentes. Pagination

- El componente Pagination nos **sirve para mostrar un paginador de datos**
- Utilizaremos una clase llamada "**pagination**", que **debemos asignársela a una lista no ordenada de HTML**
- Las clases que utilizaremos son:
 - **pagination** (se aplica a un "ul")
 - **page-item** (se aplica a cada "li")
 - **page-link** (se aplica a cada "a")



Bootstrap. Componentes. Pagination

- Por ejemplo:

```
<div class="container">
  <ul class="pagination justify-content-center">
    <li class="page-item"><a class="page-link" href="#"><</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">></a></li>
  </ul>
</div>
```





Bootstrap. Componentes. Progress

- El componente Progress nos **sirve para crear una barra de progreso**
- Para crear una barra de progreso debemos **definir un div con la clase "progress" y un div interno al anterior con la clase "progress-bar"**
- Podéis ver todas las opciones en el siguiente link:

<https://getbootstrap.com/docs/4.0/components/progress/>



Bootstrap. Componentes. Progress

- **Por ejemplo:**

```
<div class="container">
  <div class="progress">
    <div class="progress-bar" style="width:50%">
      50%
    </div>
  </div>
</div>
```





Bootstrap. Componentes. Accordion

- El componente Accordion nos **sirve para mostrar una serie de enlaces que ocultan contenido.**
- Cuando se presiona un hipervínculo se muestra el contenido para dicho enlace, luego cuando se presiona otro enlace se oculta el actual y se abre el nuevo



Bootstrap. Componentes. Accordion

ALERTA:

- Para utilizar este componente, necesitaremos importar las clases de JavaScript de Bootstrap.

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
```




Bootstrap. Componentes. Accordion

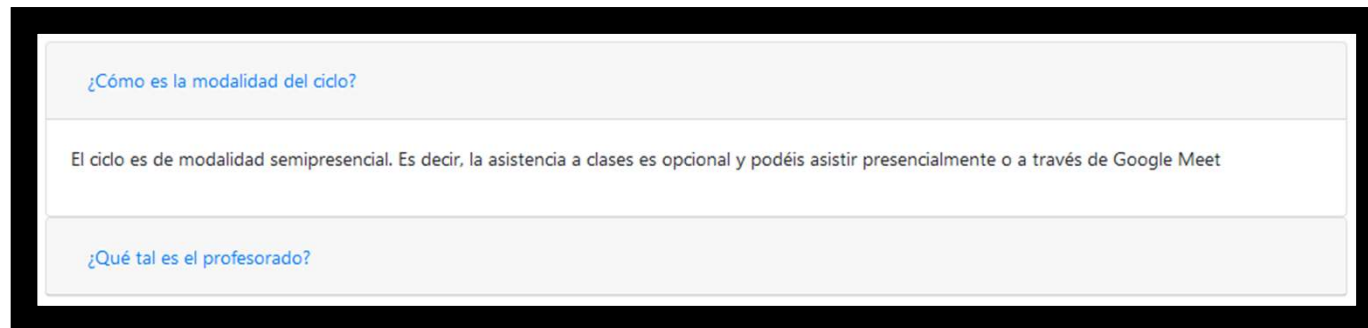
- Por ejemplo:

```
<div id="accordion1">
  <div class="card">
    <div class="card-header" id="cabeceral">
      <h5 class="mb-0">
        <button class="btn btn-link" data-toggle="collapse" data-target="#respuestal" aria-expanded="true" aria-controls="respuestal">
          ¿Cómo es la modalidad del ciclo?
        </button>
      </h5>
    </div>
    <div id="respuestal" class="collapse show" aria-labelledby="cabeceral" data-parent="#accordion1">
      <div class="card-body">
        <p>El ciclo es de modalidad semipresencial. Es decir, la asistencia a clases es opcional
          y podéis asistir presencialmente o a través de Google Meet
        </p>
      </div>
    </div>
  </div>
</div>
```



Bootstrap. Componentes. Accordion

- **Por ejemplo:**





Bootstrap. Componentes. Carousel

- El componente Carousel nos **sirve para mostrar un conjunto de diapositivas**
- Avanza en forma automática cada diapositiva y luego de la última pasa a mostrar la primera.



Bootstrap. Componentes. Carousel

- Permite activar botones a izquierda y derecha para que el usuario pueda retroceder o avanzar diapositivas.
- Además, permite activar unos pequeños indicadores para elegir cualquiera de las imágenes



Bootstrap. Componentes. Carousel

ALERTA:

- Para utilizar este componente, necesitaremos importar las clases de JavaScript de Bootstrap.

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
```



Bootstrap. Componentes. Carousel

- Por ejemplo:

```
<div id="carousel" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carousel" data-slide-to="0" class="active"></li>
    <li data-target="#carousel" data-slide-to="1"></li>
    <li data-target="#carousel" data-slide-to="2"></li>
  </ol>
  <!-- diapositivas -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption">
        <h3>Estadio Balaídos </h3>
        <p>Grada Marcador</p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption">
        <h3>Estadio Balaídos </h3>
        <p>Grada Gol</p>
      </div>
    </div>
  </div>
  <!-- botones de desplazamiento a izquierda y derecha -->
  <a class="carousel-control-prev" href="#carousel" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carousel" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```



Bootstrap. Componentes. Carousel

- **Por ejemplo:**





Bootstrap. Componentes. Formularios

- El formulario es de los componentes más utilizados en la web
- Esto es debido a que se utiliza para recoger información y transferirla
- Bootstrap nos permite darle un diseño adaptable y responsive
- A continuación, veremos algunos ejemplos de personalización de formularios:



Bootstrap. Componentes. Formularios

- Formulario común:

```
<form>
  <div class="form-group">
    <label for="nombre">Ingrese su nombre:</label>
    <input type="text" class="form-control" id="nombre" name="nombre">
  </div>
  <div class="form-group">
    <label for="clave">Ingrese su clave:</label>
    <input type="password" class="form-control" id="clave" name="clave">
  </div>
  <button type="submit" class="btn btn-default">Enviar</button>
</form>
```

Formulario común

Ingrese su nombre:

Ingrese su clave:



Bootstrap. Componentes. Formularios

- **Formulario horizontal:**

```
<form action="#" method="POST">
  <div class="form-group row">
    <label for="nombre" class="col-lg-3 col-form-label">Ingrese su nombre:</label>
    <div class="col-lg-9">
      <input type="text" class="form-control" id="nombre" name="nombre">
    </div>
  </div>
  <div class="form-group row">
    <label for="usuario" class="col-lg-3 col-form-label">Ingrese su clave:</label>
    <div class="col-lg-9">
      <input type="password" class="form-control" id="clave" name="clave">
    </div>
  </div>
  <button type="submit" class="btn btn-default">Enviar</button>
</form>
```

Formulario horizontal

Ingrese su nombre:

Ingrese su clave:



Bootstrap. Componentes. Formularios

- Formulario en una línea:

```
<form class="form-inline" action="#" method="POST">
  <div class="form-group row">
    <label for="nombre" class="col-lg-3 col-form-label">Ingrese su nombre:</label>
    <div class="col-lg-9">
      <input type="text" class="form-control" id="nombre" name="nombre">
    </div>
  </div>
  <div class="form-group row">
    <label for="usuario" class="col-lg-3 col-form-label">Ingrese su clave:</label>
    <div class="col-lg-9">
      <input type="password" class="form-control" id="clave" name="clave">
    </div>
  </div>
  <button type="submit" class="btn btn-default">Enviar</button>
</form>
```

Formulario en una línea

Ingrese su nombre: Ingrese su clave:



Bootstrap. Componentes. Formularios

- Formulario con texto de ayuda:

```
<form action="#" method="POST">
  <div class="form-group">
    <label for="nombre">Ingrese su nombre:</label>
    <input type="text" class="form-control" id="nombre" name="nombre">
    <small class="form-text text-muted">No ingrese espacios en blanco.</small>
  </div>
  <div class="form-group">
    <label for="usuario">Ingrese su clave:</label>
    <input type="password" class="form-control" id="clave" name="clave">
    <small class="form-text text-muted">No ingrese espacios en blanco.</small>
  </div>
  <button type="submit" class="btn btn-default">Enviar</button>
</form>
```

Formulario común

Ingrese su nombre:

No ingrese espacios en blanco.

Ingrese su clave:>

No ingrese espacios en blanco.

Enviar



Bootstrap. Componentes. Formularios

- **Checkbox:**

```
<div class="form-check">  
  <input class="form-check-input" type="checkbox" value="" id="check1" name="check1">  
  <label class="form-check-label" for="check1">  
    Alcohol  
  </label>  
</div>
```

Adicciones

- ☐ Alcohol
- ☐ Tabaco
- ☐ Ludopatía



Bootstrap. Componentes. Formularios

- Radio Button:

```
<div class="form-check">  
  <input class="form-check-input" type="radio" value="" id="radiol" name="radiol">  
  <label class="form-check-label" for="radiol">  
    Soltero  
  </label>  
</div>
```

Estado civil

- ☐ Soltero
- ☐ Casado
- ☐ Divorciado
- ☐ Viudo



Bootstrap. Componentes. Formularios

- **Select:**

```
<div class="form-group">
  <label for="control1">Selecione una opción</label>
  <select class="form-control" id="control1" name="control1">
    <option>ESO</option>
    <option>Bachillerato/FP</option>
    <option>Universidad</option>
    <option>Doctorado</option>
  </select>
</div>
```

Nivel de estudios

Selecione una opción

ESO



Bootstrap. Componentes. Formularios

- Textarea:

```
<div class="form-group">  
  <textarea class="form-control" rows="5" id="comentarios" name="comentarios"></textarea>  
</div>
```

Comentarios:

Actividad 3



Utilizando la Actividad 2, añádele todos los componentes que acabamos de ver.

Comprueba si la web resultante es responsive

En caso de que no lo sea, ¿qué tenemos que hacer para hacerla responsive?

Usando Bootstrap



Anchura total de la página con Bootstrap: ¿100% o limitada?

- Hacer RESPONSIVE una página no quiere decir hacer que ocupe el 100% de la anchura de cualquier monitor en el que se vaya a mostrar, sino a interrumpir el escalado al llegar a cierto tamaño, de manera que la visualización sea óptima.

Usando Bootstrap



Anchura total de la página con Bootstrap: ¿100% o limitada?

- Tenemos tres posibilidades:
 - **Utilizar una anchura limitada** (nada rebasa esa anchura)
 - **Utilizar el 100% de la anchura del monitor para todo el contenido**
 - **Utilizar el 100% de la anchura del monitor sólo para distribuir “bandas” de visualización** (con fondos que ocupen el ancho total) y luego dentro de esos fondos ya acotar el contenido a una anchura limitada

Usando Bootstrap



Anchura total de la página con Bootstrap: ¿100% o limitada?

- Utilizar una anchura limitada:

```
<div class="container"> aquí nuestros contenidos acotados en anchura </div>
```



Usando Bootstrap

Anchura total de la página con Bootstrap: ¿100% o limitada?

- Utilizar anchura completa para todo el contenido:

```
<div class="container-fluid"> aquí nuestros contenidos al 100% de página </div>
```



Usando Bootstrap

Anchura total de la página con Bootstrap: ¿100% o limitada?

- Ambas opciones combinadas:

```
<div class="container-fluid">  
  <div class="container"> aquí nuestros contenidos </div>  
</div>
```

Actividad 4



¿Cuál de las tres opciones es la mejor? ¿Por qué?



Usando Bootstrap

Anchuras de página para diferentes resoluciones

- En Bootstrap hay 4 tamaños diferentes de pantalla:
 - **Pantallas grandes (LG)**, a partir de 1200 pixeles
 - **Pantallas medias (MD)**, a partir de 992 pixeles
 - **Pantallas tipo tablet (SM)**, a partir de 768 pixeles
 - **Pantallas móviles (XS)**, por debajo de esa cifra.

Usando Bootstrap



Anchuras de página para diferentes resoluciones

- De tal manera, sabemos que los contenedores medirán como mucho 1200 pixeles en pantallas grandes, y el 100% de la anchura de pantalla en cualquier pantalla menor de esa resolución.

Usando Bootstrap



Modelo de rejilla de 12 columnas de Bootstrap

- Como ya vimos anteriormente, Bootstrap define un modelo de divisor horizontal (una fila) divisible en 12 columnas.
- Sin embargo, tenemos la ventaja de que podemos decir cuántas hay en cada resolución de pantalla.



Usando Bootstrap

Modelo de rejilla de 12 columnas de Bootstrap

- Es más fácil de entender con un ejemplo simple: pensemos en varias imágenes seguidas que, cuando estamos en **pantalla LG** (más de 1200 píxeles) queremos que estén **todas en fila**.

Usando Bootstrap



Modelo de rejilla de 12 columnas de Bootstrap

- Sin embargo, a medida que se vaya estrechando la pantalla, en vez de desaparecer de la vista, vayan saltando de línea: **3 por fila en resolución MD** (de más de 992 píxeles), **2 por fila en resoluciones SM** (de más de 768px), y **uno por fila en móviles** (pantalla XS)



Usando Bootstrap

Modelo de rejilla de 12 columnas de Bootstrap

- Para lograrlo, definiremos columnas tal que:
 - Resolución LG: 4 bloques en fila -> **4 partes de 3 columnas**
 - Resolución MD: 3 bloques por fila -> **3 partes de 4 columnas**
 - Resolución SM: 2 bloques por fila -> **2 partes de 6 columnas**
 - Resolución XS: 1 bloque por fila -> **1 parte de 12 columnas.**



Usando Bootstrap

Modelo de rejilla de 12 columnas de Bootstrap

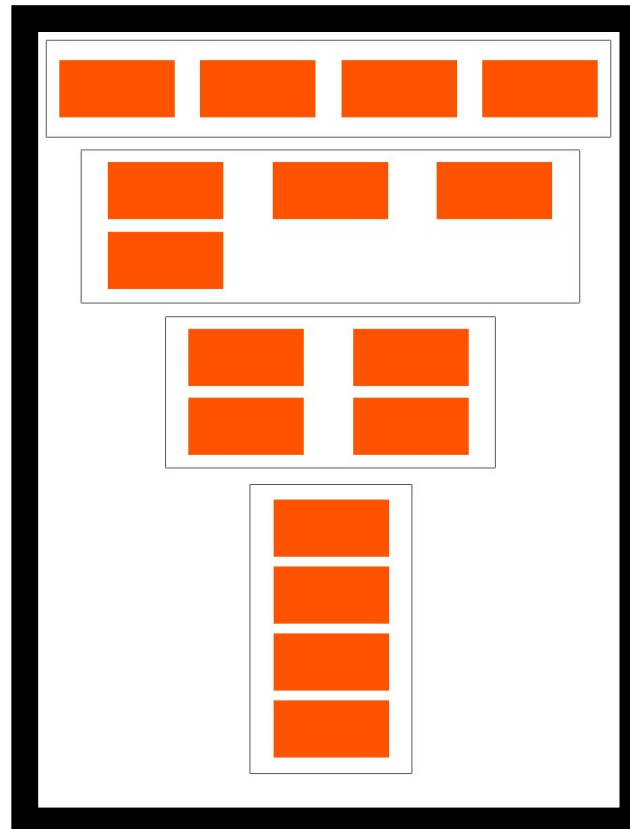
- Traducido a código:

```
<div class="row">
  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12"> Imagen </div>
  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12"> Imagen </div>
  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12"> Imagen </div>
  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12"> Imagen </div>
</div>
```

Usando Bootstrap



- Resultado final:



Actividad 5



Modifica el ejemplo anterior para lograr que se muestren:

- *6 imágenes en pantallas MD*
- *4 imágenes en pantallas SM*
- *1 imagen en pantallas XS*



Utilidad de Media Queries

- Hay una serie de casos en Bootstrap en que será muy habitual usar Media

Queries:

- **Reducir tamaños de texto en dispositivos pequeños**
- **Modificar márgenes o paddings**
- **Modificar la rejilla de Bootstrap**



Utilidad de Media Queries

- **Reducir tamaños de texto en dispositivos pequeños**
 - Si usamos pixeles para definir nuestros textos, un título de 20px de tamaño de fuente puede ser muy adecuado en un iPad. Sin embargo, en un móvil podemos encontrarlo demasiado grande.
 - Bastará con establecer para este estilo un **font-size** más pequeño para resoluciones menores que 768px (el tamaño XS en Bootstrap) para conseguir el resultado esperado



Utilidad de Media Queries

- **Modificar márgenes o paddings**

- El modelo de rejilla de Bootstrap establece 4 tamaños de pantalla.
- Eso hace que en el salto de uno a otro haya una cierta discontinuidad, es decir, un cambio de comportamiento algo brusco. Una zona que ocupando 3 de 12 columnas quedaba bien, cuando pasa a ocupar 4 de 12 puede resultar demasiado diferente en su visualización.
- Una forma de modificar esto suavizando el efecto de los cambios de resolución es utilizar los Media Queries para retocar los margins y paddings.



Utilidad de Media Queries

- **Modificar márgenes o paddings**

```
@media (max-width: 992) {  
  .bloque1 {  
    margin-left: 10px;  
  }  
}
```



Utilidad de Media Queries

- **Modificar la rejilla de Bootstrap**

- Utilizando las media queries podremos generar una rejilla de 5 columnas, que con el uso estándar de Bootstrap sería imposible (permite 1-2-3-4-6-12).
- A continuación, definiremos las columnas, creando 5 estilos específicos para ello, y dotándolos de unas cuantas características básicas (que ya tienen las columnas estándar de Bootstrap)



Utilidad de Media Queries

- **Modificar la rejilla de Bootstrap**

```
.col-xs-5ths,  
.col-sm-5ths,  
.col-md-5ths,  
.col-lg-5ths {  
    position: relative;  
    min-height: 1px;  
    padding-right: 10px;  
    padding-left: 10px;  
}
```



Utilidad de Media Queries

- **Modificar la rejilla de Bootstrap**
 - Hacemos que una columna XS ocupe un 20% de la anchura total de su contenedor:

```
.col-xs-5ths {  
  width: 20%;  
  float: left;  
}
```



Utilidad de Media Queries

- **Modificar la rejilla de Bootstrap**
 - Usando media queries, estableceremos que las columnas SM, MD y LG hagan lo mismo cuando se muestren en dispositivos con la anchura adecuada:
 - Veremos un ejemplo a continuación:

```
@media (min-width: 768px) {  
    .col-sm-5ths {  
        width: 20%;  
        float: left;  
    }  
}  
  
@media (min-width: 992px) {  
    .col-md-5ths {  
        width: 20%;  
        float: left;  
    }  
}  
  
@media (min-width: 1200px) {  
    .col-lg-5ths {  
        width: 20%;  
        float: left;  
    }  
}
```




Utilidad de Media Queries

- **Modificar la rejilla de Bootstrap**

```
<div class="row">
  <div class="col-md-5ths col-lg-5ths col-xs-6 col-sm-4">
    Contenido 1
  </div>
  <div class="col-md-5ths col-lg-5ths col-xs-6 col-sm-4">
    Contenido 2
  </div>
  <div class="col-md-5ths col-lg-5ths col-xs-6 col-sm-4">
    Contenido 3
  </div>
  <div class="col-md-5ths col-lg-5ths col-xs-6 col-sm-4">
    Contenido 4
  </div>
  <div class="col-md-5ths col-lg-5ths col-xs-6 col-sm-4">
    Contenido 5
  </div>
</div>
```

Actividad 6



Utilizando Media Queries, define los siguientes comportamientos para el <body>:

- *Comportamiento normal: Fondo negro y letras azules*
- *Es una pantalla y es menor que 992px -> Fondo azul y letras blancas*
- *Es una pantalla y es menor que 600px -> Fondo verde y letras blancas*
- *Es un móvil o es menor que 500px -> Fondo rojo y letras blancas*



Preprocesadores CSS

- CSS no es más que un lenguaje para definir y establecer las propiedades de estilo que tienen los elementos de una interfaz.
- Por ejemplo, si se necesita cambiar un color para que sea igual en muchos elementos, o se precisa una propiedad sea relativa a otro valor, un **CSS es demasiado rígido.**



Preprocesadores CSS

- Para poder desarrollar estilos que fueran más flexibles, mantenibles y, sobre todo, con una sintaxis más rica, se han desarrollado **diferentes preprocesadores de CSS**.
- Son ficheros similares a las hojas de estilos con elementos de los lenguajes de programación: variables, estructuras anidadas, operadores, etc.



Preprocesadores CSS

- Estos ficheros se compilan y se traducen a hojas de estilo CSS válidas para los navegadores.
- Los preprocesadores más conocidos son:
 - **Less**
 - **Sass**
- Son utilizados por Bootstrap y otros proyectos y permiten su personalización.



Preprocesadores CSS. Less

- Es muy sencillo de instalar **siempre que se tenga Node.js instalado previamente.**
- Basta con ir al CMD y escribir:
- Esto nos instalará el compilador Less en el sistema

```
npm install -g less
```

Preprocesadores CSS. Less



- En caso de no tener instalado el Node.js, podéis descargarlo desde aquí:

<https://nodejs.org/es/>



Preprocesadores CSS. Less

- Si se necesita únicamente para un proyecto se puede instalar así:

```
npm install -save less
```

- Una vez instalado se puede compilar un fichero Less con el siguiente comando:

```
lessc fichero.less fichero.css
```




Preprocesadores CSS. Less

- A continuación, veremos una guía de cómo instalar Less y configurarlo para su uso con el IDE Netbeans en Windows:

<https://malnuer.es/css/como-configurar-netbeans-para-usar-less-en-windows/>

Preprocesadores CSS. Less



- A continuación, veremos una guía de cómo instalar Less y configurarlo para su uso con el VS Code:

<https://marketplace.visualstudio.com/items?itemName=mrcrowl.easy-less>



Preprocesadores CSS. Less

- En VS Code es importante acordarse de colocar esta línea al principio de los

ficheros:

```
// out: ../css/estilos.css, compress: false, sourceMap: false, main: estilos.less
```

- Los parámetros son:
 - **Out:** Ruta donde se genera el fichero CSS
 - **Main:** Fichero principal que se va a convertir en CSS (se usa para ficheros parciales)

Actividad 7



Probaremos a instalar Less en nuestro equipo y a compilar un fichero de prueba



Preprocesadores CSS. Less. Variables

- Las variables en Less se identifican con el carácter especial “@”.
- Una vez definidas pueden reutilizarse a lo largo de toda la hoja de estilos.

- Less

```
@maincolor: #fff;
@defaultsize: 0.9em;

body{
    background-color: @maincolor;
    font-size: @defaultsize;
}

footer{
    font-size: @defaultsize - 0.1em;
}
```

CSS

```
body{
    background-color: #fff;
    font-size: 0.9em;
}

footer{
    font-size: 0.8em;
}
```



Preprocesadores CSS. Less. Operadores

- Existen los mismos operadores que en cualquier lenguaje de programación:
suma, resta, etc.
- No existen los tipos de variables, pero sí existen los tipos implícitos en las operaciones aritméticas.



Preprocesadores CSS. Less. Operadores

- Less

```
@default-size: 1em;
@default-width: 800px;
@min-height: 10%;
@border-size: 1cm;
@default-color: #34A343;

@paragraph-size: @default-size - 0.2em;
@article-width: @default-width * 2;
@total-height: @min-height * 10;
@border-thicker: @border-size + 0.5mm;

body{
  background-color: @default-color + #ccc;
}

.content{
  width: @article-width;
  border: @border-size solid @default-color;
}
```

Preprocesadores CSS. Less. Operadores



- CSS

```
body{  
  background-color: #ffffff;  
}  
  
.content{  
  width: 1600px;  
  border: 1cm solid #34A343;  
}
```




Preprocesadores CSS. Less. Anidación

- Less permite el desarrollo de hojas de estilos mediante estructuras anidadas.

- Less

```
section {  
  background-color: gray;  
  h2 {  
    font-size: 3em;  
    text-align: center;  
  }  
  .preface{  
    font-variant: italic;  
  }  
  .content{  
    padding: 1em;  
    font-size: 0.9em;  
  }  
}
```



Preprocesadores CSS. Less. Anidación

- CSS

```
section {  
  background-color: gray;  
}  
section h2 {  
  font-size: 3em;  
  text-align: center;  
}  
section .preface{  
  font-variant: italic;  
}  
section .content{  
  padding: 1em;  
  font-size: 0.9em;  
}
```



Preprocesadores CSS. Less. Anidación con pseudoclases

- Además de anidar, en Less también se puede tener acceso a las propiedades de pseudoclase del elemento padre.
- **Para eso se usa el carácter especial &**



Preprocesadores CSS. Less. Anidación con pseudoclasas

- Less

```
tr {  
  border-bottom: 1px solid #ccc;  
  td{  
    font-size: 0.9em;  
  }  
  th{  
    font-weight: bolder;  
    text-align: center;  
  }  
  /* Se refiere al elemento padre */  
  &:hover{  
    background-color: yellow;  
  }  
  &:first{  
    background-color: #333;  
    color: #ddd;  
  }  
}
```



Preprocesadores CSS. Less. Anidación con pseudoclases

- CSS

```
tr{
  border-bottom: 1px solid #ccc;
}
tr td{
  font-size: 0.9em;
}
tr th{
  font-weight: bolder;
  text-align: center;
}
tr:hover{
  background-color: yellow;
}
tr:first{
  background-color: #333;
  color: #ddd;
}
```

Preprocesadores CSS. Less. Anidación y ámbito



- La anidación nos permite presentar el ámbito de variables en una hoja CSS
- En Less el ámbito de una variable puede variar cuando se anidan estructuras.
- Debe prestarse atención a cómo cambia el valor de la variable **@size** cuando se vuelve a definir dentro de elementos anidados

Preprocesadores CSS. Less. Anidación y ámbito



- Less

```
@size: 1em;

body{
  header{
    @size: 0.9em;
    nav{
      font-size: @size;
    }
  }
  main {
    font-size: @size + 0.1em;
  }
}
```

Preprocesadores CSS. Less. Anidación y ámbito



- CSS

```
body header nav{  
  font-size: 0.9em;  
}  
body main {  
  font-size: 1.1em;  
}
```




Preprocesadores CSS. Less. Reutilización

- De forma muy simple e intuitiva se pueden aplicar estilos definidos en otra propiedad simplemente poniendo su nombre.
- Por ejemplo, supongamos que tenemos una **clase llamada .darktext y que se quiere aplicar sus mismos estilos** en otros elementos



Preprocesadores CSS. Less. Reutilización

- Less

```
.darktext{  
  background-color: #001133;  
  color: #5c5c5c;  
  font-size: 0.8em;  
}  
  
p {  
  border: 1px solid gray;  
  .darktext;  
}  
  
.articletext {  
  border: 1px solid gray;  
  .darktext;  
}
```



Preprocesadores CSS. Less. Reutilización

- CSS

```
.darktext{  
  background-color: #001133;  
  color: #5c5c5c;  
  font-size: 0.8em;  
}  
  
p {  
  border: 1px solid gray;  
  background-color: #001133;  
  color: #5c5c5c;  
  font-size: 0.8em;  
}  
  
.articletext {  
  border: 1px solid gray;  
  background-color: #001133;  
  color: #5c5c5c;  
  font-size: 0.8em;  
}
```



Preprocesadores CSS. Less. Bubbling

- Cuando se manejan media queries es probable que se necesite aplicar distintos estilos según el tipo de medio.
- Less permite hacer una especie de anidación con la que se pueden centralizar los estilos y luego **crear variantes según la propiedad @media.**



Preprocesadores CSS. Less. Bubbling

- Less

```
.font-settings {  
  font-size: 1em;  
  @media screen {  
    background-color: black;  
    font-size: 0.9em;  
    @media (min-width: 800px) and (orientation: landscape)  
    {  
      font-size: 1.2em;  
    }  
  }  
}  
  
@media tv {  
  font-size: 1.4em;  
}
```



Preprocesadores CSS. Less. Bubbling

- CSS

```
.font-settings {  
  font-size: 1em;  
}  
  
@media screen {  
  .font-settings {  
    background-color: black;  
    font-size: 0.9em;  
  }  
}  
  
@media screen and (min-width: 800px) and (orientation: landscape) {  
  .font-settings {  
    font-size: 1.2em;  
  }  
}  
  
@media tv {  
  .font-settings {  
    font-size: 1.4em;  
  }  
}
```



Preprocesadores CSS. Less. Librerías

- Con Less es posible dividir los ficheros en distintas partes, de tal manera que se puedan organizar y reutilizar las propiedades de forma más eficaz.

- Less

```
@import "nested.css";
@import "simple.less";
@import "mixins";

header {
  font-size: @defaultsize + 0.2em;
}
```

CSS

```
@import "nested.css";

body{
  background-color: #fff;
  font-size: 0.9em;
}

footer{
  font-size: 0.8em;
}
```

Preprocesadores CSS. Less. Librerías



PISTA: Si nombramos los ficheros empezando por “_”, representan un archivo parcial, por lo cual cuando creemos un fichero que los englobe a todos, solamente se generará este fichero CSS

Actividad 8



Crea un fichero Less donde haya:

- *Tres variables*
- *2 operadores distintos*
- *2 estructuras anidadas*
- *Utilices la anidación con pseudoclases*
- *Crees una clase y la reutilices para crear otra*
- *Utilices la técnica de bubbling para aplicar distintos estilos*
- *Divide el fichero less en: “fichero_1.less” y “fichero_2.less” y accede a ellos*



Preprocesadores CSS. Sass

- Sass es otro de los preprocesadores más populares.
- La instalación es un poco más pesada, pues requiere soporte para el lenguaje Ruby.
- Instalamos Ruby on Rails desde: <https://rubyinstaller.org/>
- Instalamos Sass desde: <http://sass-lang.com/install>



Preprocesadores CSS. Sass

- Una vez instalado se puede compilar un fichero Sass con el siguiente

comando:

```
sass fichero.sass fichero.css
```

- Al igual que con Less, si no hay errores de sintaxis en el fichero Sass, se obtiene un CSS válido como resultado.



Preprocesadores CSS. Sass

- Sass permite dos tipos de ficheros fuente distintos: **Sass** y **SCSS**
- El primero no utiliza llaves, ni puntos, ni comas y se basa en tabulaciones, mientras que el segundo es como un fichero CSS.
- A continuación veremos una guía de cómo instalarlo con el Netbeans:

<https://malnuer.es/css/como-configurar-netbeans-para-usar-sass-en-windows/>

Preprocesadores CSS. Sass



- A continuación, veremos una guía de cómo instalar Sass y configurarlo para su uso con el VS Code:

<https://marketplace.visualstudio.com/items?itemName=ritwickdey.live-sass>

Preprocesadores CSS. Sass. Ejemplo básico



- Este es un ejemplo mínimo en el que se definen un par de variables y se aplican en los estilos.

- SCSS

```
$mycolor: #fff;
$pfont: Arial, Helvetica, sans-serif;

body{
  color: black;
  background-color: $mycolor;
}

p{
  color: $mycolor;
  font-family: $pfont;
}
```

Sass

```
$mycolor: #fff
$pfont: Arial, Helvetica, sans-serif

body
  color: black
  background-color: $mycolor

p
  color: $mycolor
  font-family: $pfont
```

Actividad 9



Probaremos a instalar Sass en nuestro equipo y a compilar un fichero de prueba



Preprocesadores CSS. Sass. Variables

- Las variables, en el caso de Sass, son **identificadores que vienen precedidos por el símbolo \$**.
- Se les pueden asignar valores y, por supuesto, reutilizarlos a lo largo de la hoja Sass



Preprocesadores CSS. Sass. Variables

- SCSS

```
$bgcolor: #000;  
$font-size: 1em;  
$h1-size: 4em;  
$h1-color: red;  
  
body{  
  background-color: $bgcolor;  
  font-size: $font-size;  
}  
h1{  
  color: $h1-color;  
  font-size: $h1-size;  
}
```



Preprocesadores CSS. Sass. Variables

- CSS

```
body{  
  background-color: #000;  
  font-size: 1em;  
}  
h1{  
  color: red;  
  font-size: 4em;  
}
```



Preprocesadores CSS. Sass. Operadores

- Los operadores aritméticos son los habituales en los lenguajes de programación.
- En el siguiente ejemplo, además de aplicarlos para calcular las anchuras de un bloque section, también se ve cómo CSS puede aplicar el estilo a una etiqueta a través de su atributo role.

Preprocesadores CSS. Sass. Operadores



- HTML

```
<section role="sidebar">
```



Preprocesadores CSS. Sass. Operadores

- SCSS

```
.container {width: 100%;}  
  
section[role="sidebar"]{  
  float:right;  
  width: 250px / 900px * 100%;  
}
```



Preprocesadores CSS. Sass. Operadores

- CSS

```
.container {width: 100%;}  
  
section[role="sidebar"]{  
  float:right;  
  width: 27.77%;  
}
```



Preprocesadores CSS. Sass. Anidación

- Esta posibilidad tan esencial también está presente en Sass.
- En este caso se crea estructura anidada cuya jerarquía emula a la organización de las etiquetas HTML:



Preprocesadores CSS. Sass. Anidación

- SCSS

```
main {  
  margin: 0;  
  padding: 0;  
  article {  
    font-family: Arial, Verdana, Helvetica;  
    h1.title {  
      font-size: 4em;  
      font-weight: bold;  
    }  
    p {  
      font-size: 1em;  
    }  
  }  
  .mainfoot {  
    font-size: 0.7em;  
    padding: 1em 2em;  
  }  
}
```




Preprocesadores CSS. Sass. Anidación

- CSS

```
main {  
  margin: 0;  
  padding: 0;  
}  
main article {  
  font-family: Arial, Verdana, Helvetica;  
}  
main article h1.title {  
  font-size: 4em;  
  font-weight: bold;  
}  
main article p {  
  font-size: 1em;  
}  
main .mainfoot {  
  font-size: 0.7em;  
  padding: 1em 2em;  
}
```



Preprocesadores CSS. Sass. Mixins

- Cada navegador sigue introduciendo sus propias variantes de las hojas de estilos y muchas veces hay que molestarse en introducir todos esos valores para que cada navegador aplique el que le interese.
- A través de Sass se puede, sin embargo, facilitar la tarea de mantener esas propiedades, como por ejemplo la del borde redondeado:



Preprocesadores CSS. Sass. Mixins

- SCSS

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  border-radius: $radius;  
}  
  
.box {@include border-radius(10px);}
```



Preprocesadores CSS. Sass. Mixins

- CSS

```
.box{  
  -webkit-border-radius: 10px;  
  -moz-border-radius: 10px;  
  -ms-border-radius: 10px;  
  border-radius: 10px;  
}
```



Preprocesadores CSS. Sass. Herencia

- Sass también permite aplicar el mecanismo de herencia para poder reutilizar las propiedades de un elemento en otro.
- En este caso, las propiedades de la clase **.message** se extienden en la clase **.success**



Preprocesadores CSS. Sass. Herencia

- SCSS

```
.message {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.success{  
  @extend .message;  
  border-color: green;  
}  
  
.error{  
  @extend .message;  
  border-color: red;  
}  
  
.warning{  
  @extend .message;  
  border-color: yellow;  
}
```



Preprocesadores CSS. Sass. Herencia

- CSS

```
.message , .success, .error, .warning{  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.success{  
  border-color: green;  
}  
  
.error{  
  border-color: red;  
}  
  
.warning{  
  border-color: yellow;  
}
```

Preprocesadores CSS. Sass. Importar ficheros



- Para organizar mejor un proyecto, Sass permite separar las reglas en distintos ficheros y luego importarlas o reutilizarlas en un único fichero.
- Debemos utilizar la instrucción **import**

Preprocesadores CSS. Sass. Importar ficheros



- estilos1.scss

```
code {  
  padding: .25em;  
  line-height: 0;  
}
```

- estilos2.scss

```
ul, ol {  
  text-align: left;  
  
  & & {  
    padding: {  
      bottom: 0;  
      left: 0;  
    }  
  }  
}
```

Preprocesadores CSS. Sass. Importar ficheros



- Usando la instrucción **import**

```
@import 'estilos1', 'estilos2';
```

Preprocesadores CSS. Sass. Importar ficheros



- El resultado en CSS será:

```
code {  
  padding: .25em;  
  line-height: 0;  
}  
  
ul, ol {  
  text-align: left;  
}  
  
ul ul, ol ol {  
  padding-bottom: 0;  
  padding-left: 0;  
}
```

Preprocesadores CSS. Sass. Importar ficheros



PISTA: Si nombramos los ficheros empezando por “_”, representan un archivo parcial, por lo cual cuando creemos un fichero que los englobe a todos, solamente se generará este fichero CSS



Actividad 10

Crea un fichero Sass donde haya:

- *Tres variables*
- *2 operadores distintos*
- *2 estructuras anidadas*
- *Utilices herencia*
- *Utilices mixins*
- *Importes varios ficheros en uno*

Tema 3: CSS avanzado



Ciclo Superior DAW

Asignatura: Diseño de interfaces

Curso 20/21