

Pedro Sandoval Martínez



## Bootstrap 5



## Contenido

Introducción .....	1
Navbar .....	2
Body.....	3
Footer .....	4
Componentes de “index” .....	4
Componentes de “etapas” .....	5
Componentes de “instalaciones” .....	7
Componentes de “servicios” .....	8
Componentes de “contacto” .....	9

## Introducción

Para el desarrollo de esta práctica he usado Bootstrap en su versión 5, esta vez cargando su librería dentro del proyecto, para evitar los problemas de la anterior práctica con cierto tipo de clases. Primero paso a explicar los elementos comunes del sitio web, como son el *nav*, clases dentro del *body* y el *footer*.

## Navbar

Utilizo la clase *navbar* que hemos visto en clase junto al *navbar toggler* para que se adapte correctamente en dispositivos móviles. Contiene:

- Logo “grifo” en la parte izquierda con un efecto *hover* y un link que abre una pestaña con la página web original del Vega Media. Debajo de él situamos el nombre con una clase de css propio.

```
.logoLetra{
  font-size: 0.7em;
  color: #9dbe15;
  border-bottom: 1.5px solid #9dbe15;
  transition: 0.5s;
}
```

- Anclas a cada una de las secciones de la página (con iconos de Bootstrap). Cada una incluye un efecto *hover* que cambia su color y tamaño.

```
.nav-enlaces: hover{
  transform: scale(1.03);
  color: green;
  font-weight: bold;
}
```

- *Navbar-toggler* o *hamburguesa* a partir de tamaños “sm”.

La frase “Cooperativa de Enseñanza” Solo es visible a partir de tamaño “lg”. El Nombre del centro también cambia. En tamaños “md” se visualiza otro elemento distinto, con mayor tamaño y más grande.

```
<a href="https://ces-vegamedia.es" target="_blank" class="navbar-brand ms-3 text-center">
  
  <span class="efectoGrande logoLetra d-none d-md-grid">C.E.S Vegamedia</span>
</a>
<div class="d-grid d-md-none me-4">
  <span id="colorVerde" class="letraNav">C.E.S Vegamedia</span>
</div>
```

## Body

He creado dos clases propias para introducir una imagen al “background”. La clase *cuerpoIndex* la usamos en el resto de archivos excepto en *etapas* que cambio la imagen.

La clase lleva atributos que cambian el posicionamiento, para que sea fija y ocupe sólo su tamaño. Además, incluye un efecto oscuro, para que tenga menos brillo.

```
.cuerpoIndex, .cuerpoEtapas{
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  background-color: rgba(0, 0, 0, 0.5);
  /* Filtro oscuro */
  background-blend-mode: darken;
  background-attachment: fixed;
}

.cuerpoIndex{
  background-image: url('images/vegamedia01.jpg');
}

.cuerpoEtapas{
  background-image: url('images/niños.jpg');
}
```

Dentro del body encontramos un “div .container”, y dentro del mismo he creado “infoContainer” con el fondo blanco, relleno, y con los bordes redondeados.

```
.infoContainer{
  background-color: white;
  border-radius: 4px;
  padding: 15px;
}
```

También he creado una clase para alinear al centro del contenedor los textos, para que al variar el tamaño de columnas o de dispositivo, siempre se mantenga centrado verticalmente y no deje huecos (como en el portafolio).

Su uso lo vemos por ejemplo en el inicio, con el texto de “Presentación”:

```
.centrarAltura{
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 70%;
}
```

```
<div class="col-12 col-md-10 col-lg-4 me-5 ms-md-5 ms-lg-0">
  <h1 class="mt-4 text-success mb-3">Presentación</h1>

  <div class="centrarAltura mt-1">
    <p>El CES Vega Media un centro concertado donde la e
    <p>Somos un centro dinámico, siempre buscando
  </div>
</div>
```

## Footer

El *footer* tiene modificado su aspecto en el archivo *css* también. Cambiamos aquí su color de fondo, el de las letras y añadimos relleno. Además, a los enlaces de redes sociales les aumentamos el tamaño.

```

footer{
  background-color: black;
  color: white;
  padding: 5px ;
}

footer i{
  font-size: 1.4em;
}

```

El contenido está distribuido en 3 columnas con espaciado entre ellas. La primera, que contiene el logo del centro, solo es visible en tamaños *lg* o superior, para que no sobrecargue el pie de página.

Las otras dos, que contienen la información de contacto y las redes sociales, cambian de tamaño 2 veces según el dispositivo. El código es el siguiente:

```

<div class="row text-center justify-content-between">
  <div class="d-none d-lg-grid col-lg-4">
    
  </div>
  <div class="col-5 col-lg-4 mb-3 mb-md-0">
    <p><i class="bi bi-send"></i> info@ces-vegamedia.es </p>
    <p class="mt-5 mt-md-0"><i class="bi bi-telephone-fill"></i> 968 620913 </p>
  </div>
  <div class="col-5 col-lg-4 mb-4 mb-md-0">
    <h6 class="text-uppercase mt-1">Accede a nuestras redes</h6>
    <a href="https://www.instagram.com/cesvegamedia" target="_blank" class="col-4 col-md-2 col-lg-auto btn text-white">
      <i class="bi bi-instagram"></i>
    </a>
  </div>
</div>

```

## Componentes de “index”

Pasamos ahora a hablar de los componentes únicos de cada archivo. Usamos un vídeo dentro de la clase “*infoContainer*” que se reproduce automáticamente cuando carga la página, en bucle.

```

.videoIndex{
  border-radius: 4px;
  width: 80%;
  height: auto;
}

```

Tiene una clase propia donde modificamos el borde y el ancho, además de escalar su altura.

Este primer apartado cambia el tamaño de sus columnas 3 veces, para que concuerde con el texto que inicialmente se encuentra debajo y posteriormente se ajusta a su lado.

```
<div class="infoContainer row justify-content-between mt-3">
  <div class="col-12 col-md-10 col-lg-7 ms-md-5 mt-md-3 mt-lg-0 ms-lg-0">
    <video class="videoIndex w-100 mt-lg-3 mb-md-1" autoplay muted loop>
      <source src="images/presentacionrecorte.mp4" type="video/mp4">
    </video>
  </div>

  <div class="col-12 col-md-10 col-lg-4 me-5 ms-md-5 ms-lg-0">
    <h1 class="mt-4 text-success mb-3">Presentación</h1>

    <div class="centrarAltura mt-1">
      <p>El CES Vega Media un centro concertado donde la enseñanza es totalmente <b>gratuita</b>,<br>Somos un centro <b>dinámico</b>, siempre buscando nuestro espacio, tenemos nuestra propia<br>
    </div>
  </div>
</div>
```

Más abajo nos encontramos con el primer acordeón, para no sobrecargar de texto a primera vista. Introducimos un breve párrafo en cada “item” del mismo. La estructura del código es similar al ejemplo visto en clase, salvo los atributos y clases propias que introduzco.

Está introducido dentro del “infoContainer”, los bordes son blancos excepto el de abajo, y además tiene un efecto “hover”.

```
#acordeon {
  text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.2);
  color: #657a0e;
  border-bottom: 1.5px solid #9dbe15;
  font-size: 0.7em;
  transition-duration: 0.7s ease;
}

#acordeon:hover{
  background-color: rgb(207, 207, 207);
  color: green;
  border-radius: 4px;
}
```

## Componentes de “etapas”

A modo de “header” encontramos el título “Etapas Educativas” y debajo una *blockquote*. Para no hacer repetitiva la página, las etapas desde infantil hasta la fp básica son *cards* con modales, grado medio tiene un *acordeon* y los superiores un *collapse*.

Ambas partes están dentro del mismo contenedor y la misma fila. Están separadas por el título “Ciclos Formativos”.

Las primeras etapas, los *cards*, cambian el tamaño de sus columnas 3 veces. En dispositivos móviles se muestran 2 y 2, en tabletas hay una fila de 3 y el cuarto se sitúa debajo en el medio, y finalmente en pantallas grandes los 4 elementos en línea.

Cada elemento contiene el mismo código. Aquí incluimos otro elemento obligatorio para la práctica, los *badgets*, que cambian su color en cada etapa.

```
<div class="card col-6 col-md-4 col-lg-3">
  <div class="card-body text-center">
    <h5 class="card-title">E. Infantil</h5>
    <span class="badge bg-info mt-2">3 a 6 años</span>
    <p class="card-text d-none d-lg-grid mt-3">La mejor educación para los más pequeños</p>

    <button type="button" class="btn btn-outline-success mt-2" data-bs-toggle="modal" data-bs-target="#modalInfantil">
      Más información
    </button>
  </div>
</div>
```

Al pulsar sobre el botón se despliega un modal en el centro de la pantalla con información y una imagen.

```
<!-- Modal Educación Infantil -->
<div class="modal fade" id="modalInfantil" tabindex="-1" aria-labelledby="modalInfantillabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="modalInfantillabel">Educación Infantil</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body p-4">
        <p class="text-start">Esta etapa está dedicada al fomento y el desarrollo integral de <b>personalidad individual</b> básicas.</p>
        <p class="mt-2">
          Nuestro proyecto propio es <u>adaptable</u> a las características de cada grupo y alumno, siendo el niño el que marca la dinámica según sus necesidades, intereses y capacidades.</p>
        
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>
      </div>
    </div>
  </div>
</div>
```

El apartado de “Grados medios” contiene el mismo código que el *accordion* del *index*. La distribución de los bloques (medio y superior), cambia sus columnas. En pantallas pequeñas ocupa cada uno el ancho del contenedor, y a partir de *md* se muestran en la misma fila.

Muestro el código de otro elemento obligatorio de la práctica, el *collapse*:

```
<!-- superior -->
<div class="col-12 col-md-6">
  <h4 class="text-center mt-1" id="colorVerde">Grados Superiores</h4>
  <div class="row justify-content-center text-center">
    <div class="ciclos col-11 mt-md-2 mt-lg-0 bg-white p-2">
      <h5 class="color2">D. de Aplicaciones Web</h5>
      <!-- <p>Haz click para más info</p> -->
      <button type="button" class="col-3 btn btn-success" data-bs-toggle="collapse" data-bs-target="#plegable1">Ampliar
      <div id="plegable1" class="collapse">
        2000 horas de formación. Desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.
      </div>
    </div>
    <div class="ciclos col-11 mt-1 bg-white p-2">
      <h5 class="color2">Educación Infantil</h5>
      <!-- <p>Haz click para más info</p> -->
      <button type="button" class="col-3 btn btn-success" data-bs-toggle="collapse" data-bs-target="#plegable2">Ampliar
      <div id="plegable2" class="collapse">
        2000 horas de formación. Para programar, diseñar y llevar a cabo actividades de intervención educativa y de atención social con la infancia y sus familias.
      </div>
    </div>
  </div>
```

## Componentes de “instalaciones”

Integro un *carousel* con la clase *carousel slide* compuesto por 2 imágenes y un vídeo.

En el archivo css ajusto el máximo de altura y añadimos la propiedad *cover*, haciendo que la imagen se ajuste al tamaño del contenedor y se recorte si es necesario. Esto me ayuda a mantener la relación de aspecto con el resto de ellas.

```
.carousel-item img, .carousel-item video {
    max-height: 600px;
    object-fit: cover;
}
```

```
<div id="ejemplo-carousel" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <video class="w-100 mt-3 mb-2" autoplay muted loop>
        <source src="images/presentacionrecorte.mp4" type="video/mp4">
      </video>
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>

  <button class="carousel-control-prev" type="button" data-bs-target="#ejemplo-carousel" data-bs-slide="prev">
    <span class="carousel-control-prev-icon"></span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#ejemplo-carousel" data-bs-slide="next">
    <span class="carousel-control-next-icon"></span>
  </button>

  <div class="carousel-indicators">
    <button type="button" data-bs-target="#ejemplo-carousel" data-bs-slide-to="0" class="active"></button>
    <button type="button" data-bs-target="#ejemplo-carousel" data-bs-slide-to="1"></button>
    <button type="button" data-bs-target="#ejemplo-carousel" data-bs-slide-to="2"></button>
  </div>
</div>
```

En el *collage* de imágenes nos encontramos el primer elemento novedoso, el *lightbox*. Cada imagen, si pulsamos sobre ella se muestra en pantalla completa.

```
<h1 class="mt-1 text-success text-center mb-3">Nuestras instalaciones</h1>

<div class="row justify-content-between ms-xl-2">
  <a href="images/eso.jpg" class="collage col-6 col-md-5 col-lg-4 " data-toggle="lightbox">
    
  </a>

  <a href="images/enfermeria.jpg" class="col-6 col-md-5 col-lg-4">
    
  </a>
  <a href="images/daw.jpg" class="collage col-6 col-md-5 col-lg-4 mt-1">
    
  </a>

  <a href="images/patio.jpg" class="col-6 col-md-5 col-lg-4 mt-lg-1 mt-1">
    
  </a>
  <a href="images/pista.jpg" class="col-6 col-md-5 col-lg-4 mt-lg-1 mt-1">
    
  </a>
  <a href="images/primaria2.jpg" class="collage col-6 col-md-5 col-lg-4 mt-lg-1">
    
  </a>
</div>
```



La distribución de las imágenes cambia según el tamaño. En móviles se muestran de dos en dos. He tenido que editar algunas de ellas con la clase “collage”, debido a que eran más altas que el resto:

```
.collage img {
  max-height: 228px;
  object-fit: cover;
}
```

La sección de “Videos inmersivos” incluye 3 videos de youtube igualmente distribuidos, usando el *iframe* y jugando con la altura para que cuadre bien en las columnas. Se muestra uno seguido de otro en dispositivos móviles, y a partir de tamaño mediano, en línea.

```
<h2 class="text-center mt-3 mb-4" id="colorVerde">Videos inmersivos</h2>
<div class="text-center col-12 col-lg-4 col-xl-3">
  <iframe height="260px" src="https://www.youtube.com/embed/ORD2Nj8L7Z0?si=J5b-If5si0PiK2V"
</div>
<div class="text-center col-12 col-lg-4 col-xl-3">
  <iframe height="260px" src="https://www.youtube.com/embed/QbebYd1HMkY?si=2aToAtf-2FEqLV"
</div>
<div class="text-center col-12 col-lg-4 col-xl-3 me-xl-5">
  <iframe height="260px" src="https://www.youtube.com/embed/QSUEoG2z2Ug?si=svhgvsSkkblaZN_
</div>
```

## Componentes de “servicios”

Incluyo aquí otro elemento obligatorio, un *alert* con botón para cerrarlo. También introduzco el segundo elemento novedoso, el *list group*, que paso a explicar a continuación.

Los *list-group-item* únicamente se pueden mostrar uno en cada fila. Lo que hago es variar la cantidad de columnas para cuadrar los elementos que contiene cada uno.

Para los títulos, utilizo un identificador que uso en todo el sitio web .

```
#colorVerde{
  text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.2);
  color: #9dbe15;
  border-bottom: 1.5px solid #9dbe15;
}
```

El contenido de cada apartado se divide en el “header”, con el título y el *small* a la derecha, y el párrafo con la imagen en otro contenedor. Este último cambia su tamaño 3 veces, sobre todo para que la imagen ocupase un tamaño razonable respecto al texto.

```
<div class="col-12 col-md-6 list-group-item list-group-item-action align-items-start escalarPoco"
  <div class="d-flex w-100 justify-content-between ">
    <h3 id="colorVerde">Cantina</h3>
    <small class="me-3">Desayunos y Almuerzos</small>
  </div>
  <div class="row justify-content-center">
    <p class="col-10 col-md-7 col-lg-10 mb-3 mt-2" data-bs-toggle="tooltip" data-bs-placement="
    La cantina del nuestro centro es el lugar ideal para tomar un descanso mientras degustamos
    
  </div>
</div>
```

## Componentes de “contacto”

Encontramos en primer lugar el título y el logo del centro, solo visible en tamaños *md* o superior. Divido el contenido en dos bloques, uno para la información de contacto y otro para el formulario.

```
<div class="row justify-content-center">
  
  <h1 class="col-9 text-center text-md-start col-md-4 ms-md-5 mt-md-5 pt-md-4 text-success">Contacto</h1>
</div>
```

Ambos varían su tamaño según la pantalla. Cuando es pequeña, se posicionan uno encima de otro.

```
<div class="row justify-content-center mt-3 mt-md-5 ">
  <div class="ciclos col-12 col-md-5">
    <h3 id="colorVerde" class="text-center">Información</h3>
    <div class="centrarAltura mt-4">
      <h5 class="mb-4 d-none d-lg-grid">Centro Concertado CES Vega Media</h5>
      <p class="ms-1 mt-2 text-start"><i class="bi bi-send"></i> vegamedia@ces-vega
      <p class="ms-2 mt-2 "><i class="bi bi-telephone-fill"></i> 968 620 913 </p>
      <p class="ms-2 mt-2"><i class="bi bi-house-fill"></i> Ctra. De Mula, 37 3056
    </div>
  </div>
  <div class="ciclos col-12 col-md-5 offset-md-1 mt-3 mt-md-0">
```

El *h5* donde aparece “Centro Concertado” solo se muestra en *lg* o superior.

El formulario de contacto presenta el siguiente código:

```
<div class="ciclos col-12 col-md-5 offset-md-1 mt-3 mt-md-0">
  <h3 id="colorVerde" class="text-center">Envíanos un mensaje</h3>
  <div class="col-12">
    <div class="form-floating mb-1">
      <input type="text" name="" id="correo" class="form-control" placeholder="">
      <label for="correo">Correo electrónico</label>
    </div>
  </div>
  <div class="col-12 mt-2">
    <textarea class="form-control" id="comment" rows="5" placeholder="Pregúntanos lo que necesites:"></textarea>
  </div>
  <div class="col-12 text-center mt-2 mb-2">
    <button class="btn btn-success">Enviar Mensaje</button>
  </div>
</div>
```

Usamos la clase *form-floating* para el *input* del correo.

Además, en este archivo he tenido que incluir una modificación sobre la distribución de la estructura, ya que, al ser tan reducida en información, el *footer* no quedaba abajo. El siguiente código css suelo usarlo en otras páginas sin *Bootstrap*.

Lo que hago es definir el *body flex*, ajustar el mínimo de altura a 100vh (total de la pantalla) y al *footer* le damos una posición relativa.

```
body {  
  display: flex;  
  flex-direction: column;  
  min-height: 100vh;  
}  
.container {  
  flex: 1;  
}  
.footer {  
  position: relative;  
  bottom: 0;  
  width: 100%;  
}
```