

# Herramientas HTML y CSS II

## PEC 2

Profesor/a: **César Pablo Córcoles Briongos**  
Profesor/a colaborador/a: **Jordi Tarrida Checa**  
Trabajo realizado por: **Raimon Espasa Bou**

Fecha: **06.05.22**

# Contenido

<b>0. Información general del proyecto</b>	<b>3</b>
<b>1. Documentación del proceso de desarrollo</b>	<b>3</b>
1.1 Instalación del boilerplate y dependencias	3
1.1.1 Instalación de Bootstrap	3
1.2 Desarrollo de las páginas	4
1.2.1 Homepage	4
1.2.2 Bands	4
1.2.3 Página de información	5
1.2.4 Formulario de contacto	5
<b>2. Justificación de las decisiones tomadas en el desarrollo, dependencias instaladas y personalización, problemas resueltos...</b>	<b>5</b>
2.1 Diseño de la web	5
2.1.1 Diseño del poster de portada	5
2.2 Uso de Bootstrap y componentes	6
2.3 Diseño atómico y componentes	7
<b>3. Justificación de la elección de la metodología y/o guía de estilo y descripción de la aplicación de los criterios elegidos</b>	<b>7</b>
3.1 Uso de contenedores flex y grid.	7
3.2 Uso de Bootstrap y sus clases	7

## 0. Información general del proyecto

El proyecto, tanto el código como la demo desplegada se pueden encontrar en los siguientes enlaces:

URL pública: <https://herramientashtmlcss2-pec2-raiesbo.netlify.app/>

Repositorio: [https://github.com/raiesbo/UOC\\_herramientasHTMLyCSS2\\_PEC2](https://github.com/raiesbo/UOC_herramientasHTMLyCSS2_PEC2)

## 1. Documentación del proceso de desarrollo

### 1.1 Instalación del boilerplate y dependencias

Este segundo proyecto parte de un boilerplate UOC nuevo, aunque añadiendo todas aquellas configuraciones que son pertinentes y relevantes del proyecto anterior.

Entre las dependencias y configuraciones que se importan en este nuevo proyecto encontramos Stylelint y su configuración, además del sistema de integración continua con Netlify.

Adicionalmente, en base al feedback recibido en la misma PEC, se han realizado dos cambios de metodología. En primer lugar, se ha instalado font-awesome a modo de dependencia del proyecto mediante NPM y no mediante el uso del enlace a su CDN y, en segundo lugar, aunque se mantiene el método BEM de estructura y nomenclatura de las clases CSS, la estrategia de anidado se ha reducido a casos puntuales para permitir una mayor flexibilidad.

#### 1.1.1 Instalación de Bootstrap

Como se indica en el módulo 4.1, para implementar bootstraps en nuestro proyecto tenemos que, en primer lugar, instalar la dependencia mediante NPM y posteriormente importar los módulos instalados en nuestro proyecto, tanto en los archivos SCSS como JavaScript.

```
npm install @popperjs/core bootstrap --save
```

Importación del módulo en el documento `src/assets/styles/main.scss`.

```
@import "bootstrap/scss/bootstrap";
```

Importación en `src/assets/scripts/main.js`:

```
import * as bootstrap from 'bootstrap';
```

Una vez disponemos de la configuración previa ya podemos utilizar todo el potencial de bootstrap en el proyecto.

## 1.2 Desarrollo de las páginas

Para el desarrollo de las distintas páginas, desde el punto de vista del diseño atómico en base a componentes y teniendo en cuenta que prácticamente ningún elemento se repite más allá del *header*, *footer* y algún componente puntual, se ha tomado la decisión de implementar tanto el *header* como el *footer* como componentes flexibles que se acomodan en todas las páginas e implementar el resto de elementos directamente en cada uno de los documentos correspondientes.

### 1.2.1 Homepage

La página principal se compone de dos partes principales. En su sección superior encontramos el llamado “póster” que contiene información básica del festival como es el nombre y localización, además de una cuenta atrás que termina el supuesto día de inicio del evento. Por otro lado, en su parte inferior encontramos el listado de conciertos clasificados en base los días del evento.

Ambas partes se han compuesto principalmente mediante el uso `display: grid`, con la peculiaridad de que, tal y como se pide en el enunciado, se ha proporcionado una versión alternativa a la maquetación del póster para aquellos navegadores que no cuenten aún con esta funcionalidad. Para ello se ha utilizado la siguiente herramienta.

```
@supports (display: grid) {  
  // ...  
}
```

Para la maquetación alternativa se ha utilizado `display: flex`, el cual dispone de una mayor compatibilidad entre navegadores.

### 1.2.2 Bands

Desde la URL `/bandas.html` nos encontramos con un listado en forma de matriz el cual contiene información sobre parte de las bandas que acudirán al evento. Esta retícula, como se pide en el enunciado, no está maquetada directamente con bootstrap, sino que se han implementado directamente los estilos *grid* necesarios que podemos observar a continuación.

```
.bands-wrapper {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));  
  gap: 30px;  
  // ...  
}
```

En la parte inferior de la página se ha añadido un botón de tal modo que aparente la opción de poder mostrar más elementos del listado aunque, en este caso, es principalmente estético.

### 1.2.3 Página de información

La página de información nos aporta información genérica del festival junto a una serie de imágenes y opiniones en formato de artículo. Para ello se han incluido múltiples párrafos, listas y figuras (elementos envueltos con la etiqueta `<figure>`). De entre estos elementos, cabe destacar la opinión de un asistente del evento a modo de *quote* de estructura similar a la mostrada a continuación y el carrusel de imágenes.

```
<figure class="quote">
  <blockquote class="quote__text">
    <!-- TEXTO -->
  </blockquote>
  <figcaption class="quote__author">
    <!-- AUTOR -->
  </figcaption>
</figure>
```

### 1.2.4 Formulario de contacto

Para este documento se han introducido dos elementos principales, por un lado, tenemos el formulario de contacto y, posteriormente, contamos con una galería de imágenes. El formulario de contacto se ha desarrollado mediante los componentes de bootstraps, los cuales permiten componerlos a medida de las necesidades. Por otro lado, la galería de imágenes se ha diseñado siguiendo el patrón del listado de bandas, es decir, mediante un conjunto de estilos *grid*.

## 2. Justificación de las decisiones tomadas en el desarrollo, dependencias instaladas y personalización, problemas resueltos...

### 2.1 Diseño de la web

En cuanto al diseño general de la web, se ha buscado principalmente seguir los patrones del *wireframe* lo más aproximado posible, a excepción de algunos añadidos y de la página de contacto de la cual no se contaba con ningún detalle relevante al ser de contenido libre.

En cuanto a la responsividad de la web, puesto que se parte de wireframes para pantallas de pantallas grandes, se ha realizado un acercamiento *desktop first* donde los cambios necesarios para acomodar la página a la web se realizan en segundo lugar.

#### 2.1.1 Diseño del poster de portada

Para el diseño de la portada se ha partido de una retícula de 7x6 donde la altura está definida por el tamaño de fuente del texto contenido y el ancho es responsivo al ancho total de pantalla, y en su diseño se basa en una representación abstracta de ondas de sonido dispuestas verticalmente.

Para su desarrollo, se ha utilizado una matriz que se ha ajustado con `grid-template-columns` para que todas las celdas tengan el mismo ancho.

```
.time-to-concert {  
  display: grid;  
  grid-template-columns: repeat(6, 1fr);  
  // ...  
}
```

Posteriormente, se han ajustado los anchos de cada uno de los elementos para que se ajusten a los ejes indicados, como en el ejemplo que se ve a continuación:

```
.days {  
  grid-column-start: 4;  
  grid-column-end: 6;  
}
```

Con todo ello se obtiene como resultado un rompe con la ortogonalidad típica de las webs que es complicado alcanzar mediante el uso de *flex*.

## 2.2 Uso de Bootstrap y componentes

A lo largo de la web se han utilizado una serie de componentes de bootstrap que han sido útiles para agilizar el desarrollo. Los principales elementos usados son:

1. **Carrusel.** En la página de información encontramos el componente de carrusel de bootstrap acompañado por un pie de página. Se trata de un componente responsivo que encaja en el apartado de información al permitir disponer de una serie de imágenes sin ocupar un mayor espacio.
2. **Listado.** Al igual que el componente anterior, un listado encaja en el formato de artículo que encontramos en la página de información general. Para ello se ha implementado uno de los componentes de lista disponibles en la batería de elementos de bootstrap.
3. **Botón.** A lo largo de la web se ha requerido del uso de múltiples botones y para ellos se ha utilizado la versión estándar de bootstrap.
4. **Badge.** Para poder identificar a aquellas bandas del listado que ya han confirmado su asistencia, se ha añadido el “`badge-success`” de bootstrap con el cual se consigue exactamente el resultado esperado. Se puede encontrar en el documento `bands.html`.
5. **Formulario.** Aunque técnicamente no forma parte de los elementos que se encuentran en la sección de componentes de bootstrap, el paquete dispone de un sistema de formulario muy completo y adaptativo.

## 2.3 Diseño atómico y componentes

Como ya se ha mencionado en el correspondiente apartado del proceso de desarrollo, se han desarrollado tres componentes independientes que han sido reutilizados en cada una de las páginas. Esto es debido a que, a excepción de otros elementos basados de Bootstrap, no ha habido otra sección del código que, ya sea por estilo o por contenido, se repitiese en más de una de las páginas.

Entrando en detalle en el componente *Header*, puesto que este requería de flexibilidad para adaptarse a las distintas páginas, podemos observar en sus estilos que, mediante unos selectores concretos, los estilos aplicados cambian en función de donde es cargado, lo cual era necesario al necesitar un cambio de estilo en la *homepage*.

## 3. Justificación de la elección de la metodología y/o guía de estilo y descripción de la aplicación de los criterios elegidos

A grandes rasgos, en cuanto a estilos, cabe destacar la fórmula continuista del uso de la metodología BEM como sistema principal, aunque, como ya se ha explicado anteriormente, con un acercamiento más flexible y reduciendo el recurso de anidar estilos a situaciones concretas como son modificadores o pseudo selectores.

En cuanto a la configuración de Stylelint y el conjunto de normas de sintaxis seguidas en el proyecto, por su eficacia, se ha mantenido la configuración establecida en el ejercicio anterior.

### 3.1 Uso de contenedores *flex* y *grid*.

A lo largo del proyecto se ha buscado utilizar ambos sistemas de maquetado moderno de forma complementaria en sus distintos casos de uso. Para maquetaciones más complejas o que requieran de una composición flexible en retícula se ha utilizado `display: grid`, dejando al resto de situaciones a `display: flex`, el cual es una herramienta muy útil y sencilla para centrar y ordenar elementos de forma lineal.

### 3.2 Uso de Bootstrap y sus clases

El uso de las utilidades que aporta Bootstrap, al basar sus componentes en CSS *flex*, ayuda a agilizar el desarrollo al poder estructurar las páginas de forma muy sencilla y responsiva.

Por lo que respecta a su aplicación en el proyecto, se ha buscado un uso puntual para incorporar elementos al proyecto que, de otro modo, tomaría un tiempo considerable de implementar desde cero. Como ejemplo podemos utilizar el carrusel, componente cuya funcionalidad aportada va más allá de CSS.

Con lo que respecta a la maquetación de las páginas, en este aspecto se ha primado directamente el uso de los contenedores flex y grid mencionados anteriormente y dejar a un lado las funcionalidades de Bootstrap, ya que, en muchos casos, requería de añadir etiquetas adicionales para envolver componentes o nomenclatura de clases que no acompañan completamente a la metodología preestablecida.