

Xavier Ledesma Pons

Documentación del proceso

Para desarrollar esta práctica, primeramente, he clonado el repositorio de UOC Boilerplate de Github. A continuación, ejecutando el comando **npm install**, he instalado todas las dependencias necesarias para el desarrollo.

Una vez analizada la estructura del boilerplate, he decidido utilizar el IDE **Webstorm**, ya que estoy acostumbrado al entorno y se me hace más sencillo trabajar con el.

A continuación, he añadido la dependencia de **fontawesome-free** con el comando **npm install --save @fortawesome/fontawesome-free** para tener disponible los iconos que posteriormente utilizaré para el desarrollo del sitio web.

Para instalar **bootstrap 5** y sus dependencias he ejecutado **npm install @popperjs/core bootstrap@next --save**. A continuación, he ido al fichero **main.scss** y he añadido la línea **@import "bootstrap/scss/bootstrap.scss"**. De esta manera, todas las paginas que incluyan este fichero del boilerplate podrán hacer uso de los componentes de **bootstrap**.

Antes de empezar a desarrollar el sitio web, he analizado las diferentes opciones presentadas en los módulos de teoría para ver que metodología de estilo es la que más se asemeja a mis criterios de programación. En mi caso, me he decantado por la metodología BEM, así que antes de empezar he creado un script para analizar con stylelint (previamente instalado con **npm install --save stylelint**) mis hojas de estilo y lo he añadido al comando de build para así, ejecutar automáticamente stylelint al compilar la web para subir a producción.

Para configurar stylelint, he procedido a crear el fichero **.stylelintrc.json** y he extendido de los estilos recomendados stylelint-config-recommended-scss (**npm install --save stylelint-config-recommended-scss**). A continuación, he creado una regla para las clases, haciendo que cumplan una cierta expresión regular. De esta manera, puedo asegurar que todas las clases utilizan la nomenclatura específica del estilo BEM.

Este último apartado de decisión de estilo es heredado de la PEC anterior donde también use esta metodología.

Una vez configurado todo el entorno, he procedido a iniciar el desarrollo del sitio web, copiando las imágenes necesarias en la carpeta **assets/images**. Como el sitio web va a tener más de una página, he estructurado la carpeta **styles** de la siguiente manera:

- Una carpeta **layouts** que contiene una carpeta llamada **shared**. En esta carpeta se encuentra un partial llamado **_shared.scss** con los estilos comunes que se usarán en las diferentes páginas. De esta manera, tengo concentrado todo lo que es común en un fichero y es más fácil editar un estilo.

- Dentro de **layouts**, se encuentran los partials **_biography.scss**, **_home.scss**, **_members.scss**. Estos ficheros contienen los estilos característicos de cada una de las páginas que su nombre indica.
- Un fichero llamado **_variables.scss** que contiene la definición de variables de nuestro sitio web y donde se encuentran las variables modificadas de **bootstrap**.

En el caso de este proyecto, he modificado el color primario del tema mediante la variable **\$theme-colors** -> **primary**, la variable **\$card-border-radius** para modificar el redondo de las cards que he usado en la página **biography** y **\$navbar-light-active-color** para modificar el color de los enlaces activos de la barra de navegación.

Para estructurar las páginas, he creado un fichero HTML para cada una de las secciones solicitadas.

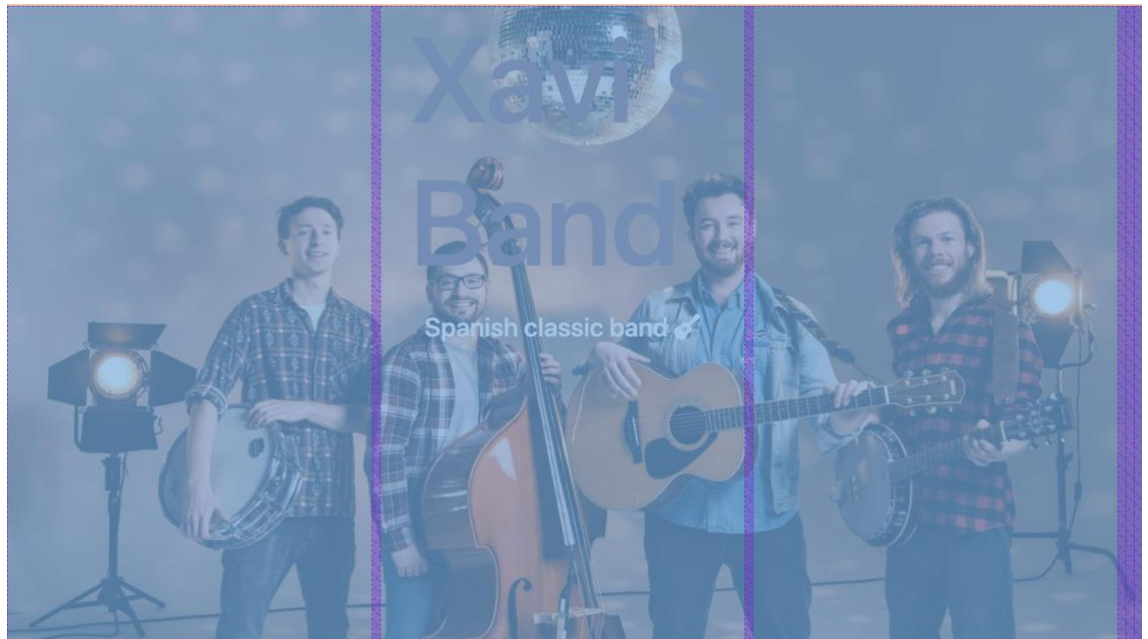
Para todas las páginas se ha seguido la siguiente estructura:

- Una clase llamada **header** que contiene una barra de navegación (usando clases de bootstrap) con el enlace de la página activa en cada caso y el logotipo de la UOC. La barra se convierte en un menú de hamburguesa cuando el tamaño de pantalla disminuye (responsive).
- Una clase llamada **footer** que contiene el logotipo de la UOC, una barra de navegación igual que la superior y unos iconos que enlazan a redes sociales. Este **footer**, se muestra siempre en la parte superior, sobrepuesto al contenido. La barra se convierte en un menú de hamburguesa cuando el tamaño de pantalla disminuye (responsive).
- La parte central de cada página es específica en relación con el contenido que se va a mostrar.

Para la página **HOME** se ha seguido la siguiente estructura:

- El contenido central es un **grid layout** que contiene una imagen, el nombre de la banda y una frase con un icono de la librería **fontsawesome**.
- Como podemos ver en su correspondiente hoja de estilos, este layout está compuesto por 3 columnas y una fila. La imagen está situada en la primera columna y fila. El título de la banda en la segunda columna y primera fila y el subtítulo en la segunda columna y primera fila.

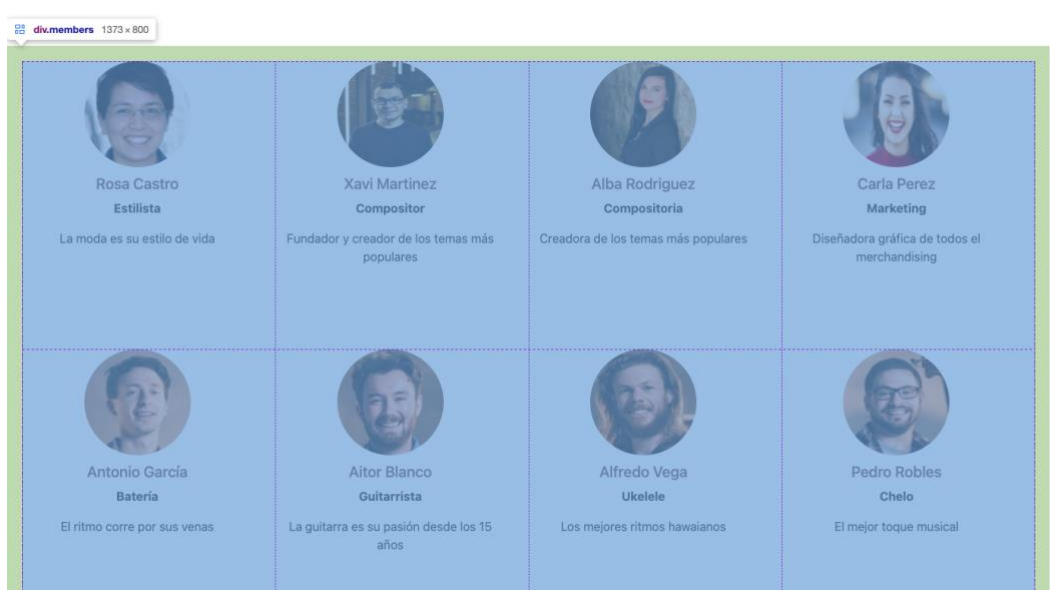
A continuación, podemos ver con el inspector de Chrome como ha quedado estructurado el **grid**.



En la página **MEMBERS** se ha seguido la siguiente estructura:

- El contenido central esta formado por un titulo y una clase **members** que como podemos ver en su correspondiente hoja de estilos crea un **flex layout**.
- Cada componente del **flex layout** esta formado por una imagen convertida en avatar, el nombre del componente, el rol en la banda y una frase característica.

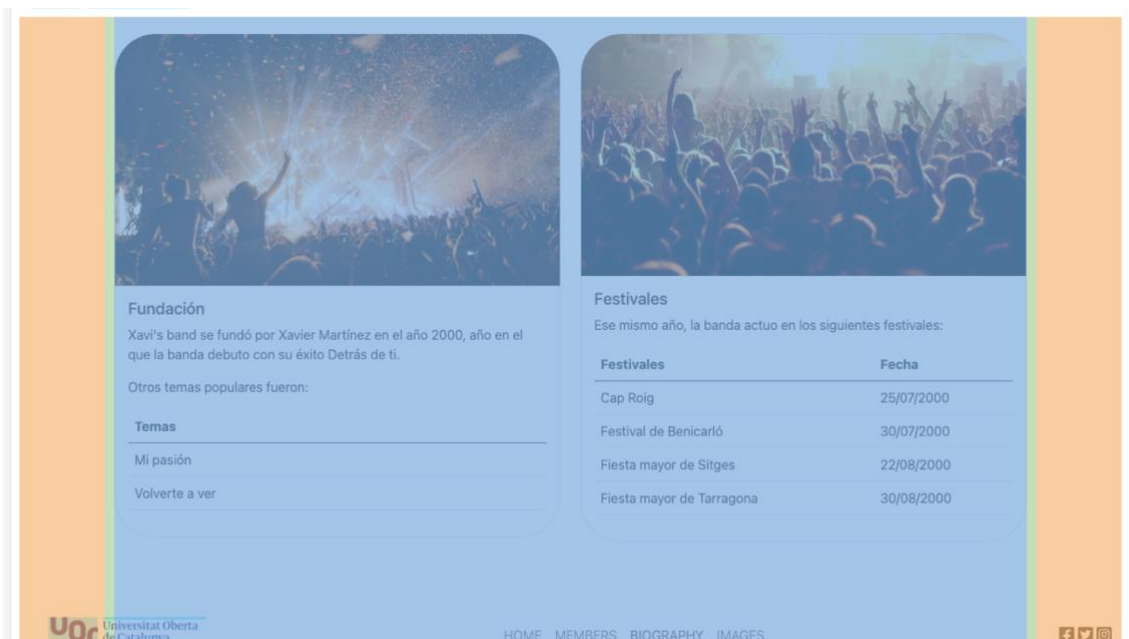
A continuación, podemos ver con el inspector de Chrome como ha quedado estructurado el **flex**.



En la página **BIOGRAPHY** se ha seguido la siguiente estructura:

- Se ha usado la clase **container** de bootstrap para luego en su interior utilizar 2 divisiones con la clase **row** de bootstrap con 2 divisiones con la clase **col** de bootstrap.
- En cada columna, se ha utilizado la clase **card** para crear 4 apartados de biografía que explican cosas sobre el grupo. Cada **card**, contiene también una imagen temática sobre el contenido.

A continuación, podemos ver con el inspector de Chrome como ha quedado estructurado el **container**.



Por último, en la página **IMAGES**, se ha seguido la siguiente estructura:

- El contenido central es un carousel de imágenes de bootstrap donde se incluyen fotos de los festivales. Este carousel va pasando de imagen automáticamente, incluye flechas laterales y un menú de navegación inferior.

A continuación, podemos ver con el inspector de Chrome como ha quedado estructurado el **carousel**.



Cabe destacar que se ha verificado y adaptado el contenido para que sea responsive y se ha seguido una metodología mobile first.

Por último, he publicado mi sitio web en Netlify siguiendo los pasos detallados a continuación:

- 1) He creado una cuenta directamente con mi **Github**.
- 2) He seleccionado mi repositorio (<https://github.com/xalepo4/HtmlAndCssII>) y he indicado que la carpeta base es PEC2, ya que he subido la práctica en esa carpeta.
- 3) He indicado que se debe ejecutar el comando **npm run build** y que la web estará contenida en la carpeta **dist**.
- 4) Se ha generado el deploy con la siguiente URL: <https://hardcore-swartz-860f3e.netlify.app>