

PEC 2

WEB FESTIVAL

Herramientas HTML y CSS II
Angel Megido Garcia.

Contents

Creación de la web.....	2
Boilerplate.....	2
Desarrollo HTML	5
Desarrollo CSS.....	6
Subida a producción.....	7
Conclusiones.....	8



Universitat Oberta
de Catalunya



Creación de la web.

Boilerplate

Clonar Boilerplate.

El primer paso para crear la web sobre un festival de música consiste en adecuar el Boilerplate a las características requeridas.

Se requiere el uso de la plantilla UOC Boilerplate¹, que está diseñada para correr en Parcel² y ya incluye el compilador Sass. Para ello nos descargaremos la plantilla en el directorio de trabajo con el comando:

```
git clone https://github.com/uoc-advanced-html-css/uoc-boilerplate.git
```

Esto clonará el repositorio en el directorio en el que nos encontremos, la sintaxis del comando es sencilla:

```
git clone <repositorio> <directorio>
```

Instalar Parcel

Una vez hecho esto se tendrá que instalar Parcel y las dependencias que tiene el repositorio. El siguiente comando la instalará:

```
npm install|
```

¹ <https://github.com/uoc-advanced-html-css/uoc-boilerplate/>

² <https://parceljs.org/getting-started/webapp/>

Se solicita instalar alguna dependencia más, yo he optado por usar la recomendada en los módulos previos, fontawesome-free³. Con ella podremos usar los iconos de fontawesome⁴ de manera sencilla en nuestro proyecto. Si deseáramos usar alguna otra podríamos usar el buscador de npm para encontrarla.

Ahora que tenemos el Boilerplate de la OUC instalado con Parcel y sus dependencias, deberemos revisar el entorno de desarrollo y su correcto uso. En el archivo package.json podemos ver las dependencias de Parcel así como su configuración. En especial debemos prestar atención a la parte de scripts, por ahí se configurarán los parámetros para trabajar en el entorno de desarrollo y para compilar para producción.

Instalar Stylelint

Otro de los requerimientos de esta PEC es el uso de Stylelint, el cual debemos de añadir a los scripts de desarrollo y producción para que se ejecute la comprobación de las reglas de stylelint cuando lanzamos el script. Pero previamente debemos instalarlo en nuestro entorno, el siguiente comando se encargará de ello:

```
npm install --save-dev stylelint
```

Como vamos a usar SCSS, tendremos que instalar las reglas⁵ para Stylelint, así como las recomendadas⁶ para el trabajo con SCSS:

```
npm install --save-dev stylelint-scss stylelint-config-recommended-scss
```

También se ha de crear el fichero de configuración de Stylelint, un archivo de texto con el nombre: “.stylelintrc”. En él pondremos las reglas, la configuración para usar SCSS y las reglas⁶ de estilo que queremos aplicar. En nuestro caso hemos usado la guía de estilo correspondiente a la metodología BEM⁷, y nuestro fichero quedaría como sigue:

³ <https://www.npmjs.com/package/@fortawesome/fontawesome-free>

⁴ <https://fontawesome.com/icons>

⁵ <https://github.com/stylelint-scss/stylelint-scss>

⁶ <https://github.com/stylelint-scss/stylelint-config-recommended-scss>

⁷ <http://getbem.com/introduction/>

```
{
  "plugins": [
    "stylelint-scss"
  ],
  "extends": ["stylelint-config-recommended-scss"],
  "rules": {
    "selector-class-pattern": "^[a-z]([a-z0-9-]+)?(__([a-z0-9]+-?)+)?(--([a-z0-9]+-?)+){0,2}$",
  }
}
```

Ahora que ya está instalado y configurado, con el comando:

```
npm run stylelint
```

Podemos comprobar que nuestro código cumple con las reglas de estilo seleccionadas. Además, si hemos añadido Stylelint a nuestros comandos de desarrollo y producción (*"npm run dev"* o *"npm run build"*), se ejecutarán las comprobaciones en estos entornos.

Instalar Bootstrap

En esta PEC se requiere hacer uso parcialmente del marco de trabajo Bootstrap⁸, por lo que procederemos a instalarlo con el siguiente comando:

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

Una vez instalado el framework con npm se configuran las dependencias en el fichero *main.js* haciendo la importación con:

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

Así mismo incluiremos los estilos de Bootstrap y las dependencias en nuestro fichero de principal de Sass, *main.scss*, con el comando `@import`.

⁸ <https://getbootstrap.com/>

Desarrollo HTML

Para afrontar nuestro desarrollo HTML tenemos 4 wireframes dados, que condicionaran el diseño al prototipo representado en ellos.

En nuestro caso hemos adaptado estos prototipos a nuestro proyecto, teniendo las siguientes partes en nuestra web:

Una página principal, **index.html**, donde se presenta el festival de manera gráfica, con información esencial y descriptiva.

Una pagina donde se presentan los grupos, **grupos.html**, por medio de tarjetas con imágenes e información, en una retícula predefinida en los wireframes.

Otra pagina descriptiva del festival, **festival.html**, en la que a modo de articulo se presenta información tanto grafica como en texto.

Una ultima pagina dedicada a la compra de entradas, **entradas.html**, en la que se publicita el festival mediante imágenes y texto y con un formulario de compra final.

Todas las paginas tienen dos elementos de navegación comunes, el menú y el pie. Se ha optado por diseñarlos de forma separada, añadiéndolos a cada pagina con `<include>`, para facilitar su rediseño en caso necesario. Se pueden encontrar en la carpeta `/views` bajo los títulos **footer.html** y **navbar.html**.

Index.html

La pagina principal presenta un diseño bastante simple en su parte HTML pero muy trabajado en la aplicación de los estilos. Contiene una clase principal `<main>` con dos titulos `h1` y `h2`, un enlace, una imagen y dos párrafos de texto.

grupos.html

La segunda pagina tiene un contenedor “container” que engloba las diferentes tarjetas con las imágenes y el texto. Aunque hay más código el diseño HTML también es bastante sencillo.

festival.html

En esta pagina el contenido se engloba en la etiqueta `<article>` y como las otras tiene una clase principal “centrador” que se utiliza para centrar el contenido. El diseño es algo mas complejo, con una lista ordenada para mostrar contenido acerca del festival y un párrafo y una cita en bloque que se muestran a dos columnas haciendo uso de la clase “twocol”. Para finalizar tenemos una imagen a la que aplicamos estilo con la clase “fest” y un párrafo que la acompaña.

entradas.html

En esta pagina continuamos usando la clase centrador para adecuar el contenido. En primer lugar, tenemos un carrusel⁹ aplicado con Bootstrap, que muestra 3 diferentes imágenes. Un botón¹⁰ que llevaría a los videos de las mismas, también usando un componente de Bootstrap. Después, tras un título y un párrafo tenemos unas tarjetas¹¹ introducidas haciendo uso de Bootstrap y, por último, tras otro título y párrafo, tenemos un formulario introducido con Netlify Forms¹².

navbar.html

El menú de navegación es una lista desordenada con un logo en el primer elemento de la misma.

footer.html

El menú del pie de página está construido con Bootstrap¹³ y además hace uso de iconos con *Font Awesome*¹⁴.

Desarrollo CSS.

Para el desarrollo CSS se ha usado Sass y Bootstrap para la creación del código, que será transformado posteriormente con el preprocesador de Parcel. El diseño se ha seguido con un enfoque desktop first, ya que así venia implícito en los requisitos de la PEC.

La estructura es de un fichero principal main.scss, 6 parciales: _entradas.scss, _festival.scss, _footer.scss, _grupos.scss, _index.scss, _navbar.scss y otro parcial para las variables, _variables.scss.

main.scss

Este fichero se utiliza únicamente para la importación y no lleva ningún estilo de CSS:

_index.scss

Es el fichero principal en el que se aplican los estilos generales del proyecto y los específicos de la página index. Para el diseño del poster se ha cogido como base el diseño de Chen Hui Jing, modificándolo para adaptarlo a nuestro proyecto.

_grupos.scss

Se aplican los estilos para la página **grupos.html**

⁹ <https://getbootstrap.com/docs/5.1/components/carousel/>

¹⁰ <https://getbootstrap.com/docs/5.1/components/buttons/>

¹¹ <https://getbootstrap.com/docs/5.1/components/card/>

¹² <https://docs.netlify.com/forms/setup/>

¹³ <https://getbootstrap.com/docs/5.1/components/navbar/>

¹⁴ <https://fontawesome.com/>

_entradas.scss

Se aplican los estilos para la pagina **entradas.html**

_festival.scss

Se aplican los estilos para la página **festival.html**. La primera imagen de fondo se introduce en este fichero.

_footer.scss

Se aplican los estilos para el menú del pie de página, **footer.html**. Es un menú Bootstrap personalizado con Sass.

_navbar.scss

Se aplican los estilos para el menú de la cabecera, **navbar.html**.

Subida a producción.

Para esta PEC se requiere la puesta en producción de la web haciendo uso del servicio de Netlify¹⁵ asociado con el repositorio de GitHub¹⁶.

Se puede acceder a ambos desde aquí:

Repositorio [GitHub](#).

Web alojada en [Netlify](#).

¹⁵ <https://www.netlify.com/>

¹⁶ <https://github.com/>

Conclusiones.

Una vez finalizado el trabajo he podido reflexionar sobre las tareas llevadas a cabo y los conceptos aprendidos durante esta práctica. Puedo decir que la parte del trabajo con el CSS se me ha hecho bastante difícil, quizás por falta de tiempo y con toda seguridad por falta de soltura en la programación con este lenguaje. He perdido mucho tiempo en el diseño y me ha retrasado bastante a la hora de practicar con otras herramientas. Supongo que es algo que solo la practica y la constancia podrán resolver.

A la hora de trabajar con el boilerplate y sus diferentes dependencias me encuentro mas seguro y puedo resolver las incidencias de manera mas rápida. Durante la realización del ejercicio tuve un desafío añadido, ya que me mudé de país y en el proceso mi pc habitual para el diseño web, que tenia instalado y configurado toda la suite de diseño, se rompió, por lo que tuve que rehacer de nuevo en mi portátil toda la infraestructura para trabajar. El uso del boilerplate de la UOC ayuda mucho en esta tarea. A pesar de ello perdí algunos documentos y archivos, lo que me recuerda la necesidad de guardar copias en la nube de los documentos importantes.

En relación a las diferentes herramientas utilizadas, he podido familiarizarme un poco con el entorno Bootstrap y entrever su potencial, además de coger mayor soltura y confianza en Sass, Stylelint o el uso de guías de estilos. También con el uso de GIT y el despliegue en Netlify, que ahora apenas me lleva tiempo.

En líneas generales estoy bastante contento tanto de lo desarrollado como de lo aprendido durante el ejercicio, si acaso lamento no haber tenido mas tiempo para pulir el proyecto y el código, que creo es bastante mejorable.