
Herramientas HTML y CSS - PEC 2

4 de diciembre del 2020

Proceso de desarrollo

Primeros pasos y configuración

La configuración de esta práctica ha sido más sencilla que la de la anterior, no solamente porque ya lo había hecho antes, sino porque me he basado en la primera y la he replicado. El único cambio que ha habido ha sido a la hora de la integración continua. En la primera práctica cometí un error (no entendí bien como tenía que ir) y publiqué la web subiendo la carpeta dist al repositorio. Yo sabía que en dist era donde se guardaba la build de producción, por lo que la subí e hice deploy desde ahí.

En esta ocasión, lo he mantenido en el gitignore y en cambio he utilizado el CI/CD de Netlify usando como comando “npm run build” y como directorio de output “dist” para hacer la build de producción desde ahí, manteniendo en el repositorio solo los archivos de configuración y los src.

Sobre metodología y estilos

Al igual que en la práctica anterior, he decidido utilizar BEM. Me parece una nomenclatura muy útil y clara y debo decir que no solo la he aplicado aquí sino que también la he introducido en mi trabajo y ahora mismo la estamos aplicando. En la anterior práctica comenté que me parecía algo problemático el tema de los elementos anidados, pero tras usarlo un poco más, creo que la técnica de aplanarlos (no escribirlos anidados sino utilizar solo el nombre del bloque, __ y el nombre del elemento independientemente de su nivel) es bastante clara.

He utilizado stylelint de la misma manera que en la anterior práctica. También he empezado a aplicarlo en mi trabajo para validar mi CSS. El archivo de configuración lo adjunto de nuevo aquí:

```

1  module.exports = {
2    "extends": "stylelint-config-standard",
3    "plugins": [
4      "stylelint-scss",
5      "stylelint-selector-bem-pattern"
6    ],
7    "rules": {
8      "plugin/selector-bem-pattern": [
9        "componentName": "[A-Z]+",
10       "componentSelectors": {
11         "initial": "^\\.{componentName}(?:-[a-z]+)?$",
12         "combined": "^\\..combined-{componentName}-[a-z]+$"
13       },
14       "utilitySelectors": "^\\.util-[a-z]+$"
15     ],
16     "selector-nested-pattern": "^&",
17     "indentation": 2,
18     "no-descending-specificity": null,
19     "no-eol-whitespace": null,
20     "declaration-empty-line-before": null,
21     "value-keyword-case": null,
22     "at-rule-no-unknown": [
23       true,
24       {
25         "ignoreAtRules": [
26           "tailwind",
27           "apply",
28           "responsive",
29           "variants",
30           "screen",
31           "use",
32           "include",
33           "mixin"
34         ]
35       }
36     ]
37   }
38 }

```

Sobre el diseño

Para la página he decidido basarme en lo que se conoce como la música Emo. Se trata de una música por lo general punk o rock, a veces algo más melódica (pop-punk) y de característica oscura y rebelde. Estereotípicamente, sería música de adolescentes rebeldes. A pesar de haber pasado hace muchos años esa época, muchos de esos grupos siguen estando entre mis favoritos y de mis amigos. Así pues, de todo esto ha nacido el EmoSound 2021. Se trata de un festival musical ficticio en el que se reúnen las bandas más míticas de este género, algunas de ellas que se deshicieron hace tiempo han vuelto a reunirse sólo para este momento. He intentado darle a la página un toque humorístico con algunos de sus textos y reírme así un poco del estereotipo.

En cuanto a los colores, al tratarse de un festival Emo, debían predominar los colores oscuros. Por tanto, se trata de una paleta de negros y grises con blanco para los textos. Habría quedado muy sosa, así que necesitaba un color vivo que acentuase las distintas partes. Aprovechando que la música emo suele asociarse a sentimientos tristes (en inglés, *feeling blue*), he decidido optar por un color azul para resaltar ciertas partes y representar todo esto.

Sobre la metodología o enfoque, he decidido utilizar un enfoque desktop-first. El motivo son en realidad dos. El primero, que los blueprints estaban hechos en desktop y por tanto tenía sentido partir de esa base y adaptarlo luego a móvil. El segundo es que, al ser una página de un festival, era especialmente importante que la homepage fuese visual y ahí siempre va a tener más impacto el desktop.

Aún así, el diseño funciona en ambos.

Sobre las dependencias

He añadido una única dependencia externa de FontAwesome, para los logos sociales del footer. La he instalado según se indica en el material proporcionado.

Y he hecho un import en el main.scss para añadirla.

Por supuesto también he utilizado Bootstrap, pero no la cuento porque era obligatoria. Respecto a Bootstrap, debo decir que no me entusiasma. En mi primer trabajo, lo utilicé mucho y todo el rato estaba abusando de containers, rows y cols para los distintos layouts. Sin embargo encontré, (seguramente debido a mi inexperiencia de entonces) que mis HTML tenían siempre contenedores extra para acomodar todo esto. Una vez descubrí Flexbox y Grid, tomé como política dejar de usar Bootstrap y desarrollar mis layouts con Flex y Grid. Esto lo explico como preferencia personal, pero veo la utilidad de Bootstrap (y otras librerías como Bulma) y pienso

que en proyectos pequeños y rápidos son la mejor opción para levantar una interfaz de usuario usable, bonita y adecuada de forma rápida.

No he encontrado en este caso ninguna otra dependencia que tuviese sentido usar en esta web.

Sobre las distintas partes

Para el header, he seguido más o menos el wireframe y para la adaptación a móvil, he optado esta vez por implementar un menú Hamburger con un mínimo de Javascript. Ya que había varios links en la barra de navegación, he creído que era la mejor opción. He añadido una pequeña animación para que el menú se despliegue desde la derecha y se esconda por el mismo lado.

Para el footer, me he basado de nuevo en los wireframes y en la paleta de colores mencionada anteriormente. Para la versión en móvil he reorganizado los componentes del footer en vertical. También he añadido una frase abajo como disclaimer sobre el uso del contenido y el fair use.

Para la página frontal, he decidido imitar algo del estilo de un póster de revista musical en versión horizontal. He aprovechado la separación natural que ofrecía la imagen de Alex Turner para crear dos secciones, una con el logo y los participantes y otra con la descripción de Arctic Monkeys. Luego he creado dos cuadros para resaltar los otros dos grupos principales del evento, con una descripción al pasar por encima y un link para descubrir más. Para maquetarlo he usado grid como pedía el enunciado y en caso de que el navegador no lo soportase, he utilizado flex y algunos ajustes para conseguir un resultado similar. El título en la versión no grid queda en la parte superior en lugar de a la izquierda. Podría haberlo situado a la izquierda con un wrapper pero no veía lógico modificar el html para un fallback. Para la versión de móvil he colocado los elementos en una única columna.

Para la página de grupos, he creado una retícula con Flex en la que aparecen los distintos grupos, su descripción y sus imágenes. Según la pantalla se hace más pequeña, el número de columnas por fila disminuye. También he añadido un catch en la parte superior que no figuraba en el wireframe.

Para el artículo, he utilizado las dos imágenes que se pedían, una como background y la otra como imagen de contenido con su pie de foto. Es aquí donde he aprovechado para aplicar más Bootstrap, utilizando estilos de lista, blockquotes, citas, container, row y col para conseguir el layout. También he modificado las variables de quote por defecto de Bootstrap (a parte del botón primario, que he utilizado en la cuarta página).

Finalmente, para la página extra, he creado un sencillo form basado en Bootstrap para aprender cómo funcionan los forms de Netlify. Me ha sorprendido lo sencillo que es tenerlos funcionando. Si ya me gustaba más el CI/CD, esto ha acabado de decantar para mí la batalla entre Github

Pages y Netlify a la hora de deployear proyectos pequeños, ya que la posibilidad de añadir un formulario de contacto tan sencillo y sin ningún tipo de backend es algo que GHPages no ofrece y a veces se echa en falta.

Como dato adicional algunas de las imágenes que no aportaban contexto (como la de los cuadros de los dos grupos principales, cuyo contexto estaba en el título y la foto) las he utilizado como background y, mirando la disposición de las mismas, he utilizado background-size y background-position para que quedasen bien dispuestas.

Sobre la compilación para producción

Para compilar para producción he utilizado npm run build, habiéndome asegurado antes de que el CSS pasaba los checks de Stylelint mediante npm run stylelint.

Repositorio: <https://github.com/CarlosRM/CV>

URL pública: <https://serene-khorana-4a715a.netlify.app/>