

Preparar el entorno de desarrollo

En primer lugar descargué UOC Boilerplate del enlace <https://github.com/uoc-advanced-html-css/uoc-boilerplate> en formato ZIP. Lo descomprimí en el directorio en el que quería tener el proyecto. Como la versión de **npm** que tenía instalada estaba obsoleta, descargué la última versión de la web <https://nodejs.org/en/> que en ese momento era la versión 14.16.0 LTS. Una vez ejecutado el fichero msi descargado la versión de **node.js** y **npm** estaban actualizadas. En el proceso de instalación acepté la instalación de las herramientas adicionales.

Después accedí al directorio desde el terminal PowerShell de Windows 10 y ejecuté el comando: `npm install`. Esto instaló todas las dependencias descritas en el archivo **package.json** y todas las dependencias que necesitan. Al finalizar el proceso aparece la carpeta **node_modules**.

Siguiendo la guía facilitada en el aula, procedí a instalar la dependencia **FontAwesome** mediante el instrucción: `npm install --save @fortawesome/fontawesome-free`.

En este momento ya podía empezar con el desarrollo, para ello ejecuté en el terminal la instrucción: `npm run dev`. Esta instrucción crea un servidor local en la dirección: <http://localhost:8123> y abre una ventana del navegador que se actualiza cada vez que modificamos algún fichero del proyecto. El editor de texto que yo he usado es **Visual Studio Code**.

Desarrollo

Diseño

Para la realización de la página web me decidí por un diseño sencillo:

Un **header** que constaba de dos bloques:

- El primero incluye: Nombre + Profesión + pequeña descripción.
- El segundo: Fotografía + datos de contacto.

Después viene una **sección** con 2 bloques:

- Experiencia, en la que se muestra los últimos trabajos realizados.
- Formación, en la que se muestra los títulos de más alto nivel adquiridos.

Por último una **sección** en la que se muestran el nivel que tengo sobre algunas habilidades.

Las buenas prácticas recomiendan empezar el diseño de una página web primero por la versión “mobile”, pero en este caso he decidido empezar por la versión de escritorio ya que tenía muy claro que la versión “mobile” iba a tener cada uno de estos bloques unos encima de los otros en una sola columna, así que el verdadero reto de diseño estaba en la versión de escritorio.

Para hacer la página “**responsive**”, he decidido usar **flex-box** para organizar los bloques horizontalmente tanto en el header como en la sección con la Experiencia y la Formación. Para la versión “mobile” he añadido unos **media query** que para pantallas menores de 700px de anchura, organizan los bloques verticalmente. También he tenido que hacer pequeños cambios en el tamaño de la fotografía para la versión “mobile”.

Metodología y guía de estilo

Al ver que mi diseño constaba básicamente de 5 bloques, he decidido usar la **metodología BEM** ya que se basa en localizar en html bloques, elementos que los componen y modificadores, y hacer código CSS que se pueda reutilizar si en cualquier momento se vuelven a usar estos bloques.

Esta metodología tiene unas reglas de nombrado de clases muy estrictas, que he intentado seguir. He configurado **Stylelint** para que compruebe que los nombres de mis clases siguen el patrón con esta regla en el fichero **.stylelintrc**:

"selector-class-pattern": "^.[a-z]([a-z0-9-]+)?(__([a-z0-9-]+-?)+)?(--([a-z0-9-]+-?)+){0,2}\$"

Al comprobar por primera vez si mi código seguía la guía de estilo me salieron errores en clases que había escrito con letras mayúsculas. La instrucción para comprobar esto es: **npm run stylelint**

Al modificar estos nombres con letras minúsculas Stylelint ya no mostraba más errores.

Otra característica de esta guía de estilo es que en vez de usar tabulaciones para anidar elementos en HTML usa dos espacios. Yo estaba acostumbrado a usar las tabulaciones, pero la verdad que creo que con 2 espacios el código se interpreta igual de bien y no corres el riesgo de desplazarte demasiado a la derecha si hay muchos elementos anidados unos dentro de los otros.

Compilación para producción y publicación

La instrucción para crear la versión de producción es: **npm run build**

Esta instrucción genera una nueva carpeta (llamada dist) dentro del proyecto que incluye estos ficheros a destacar:

- index.html: es el mismo fichero que hay en entorno de desarrollo.
- un fichero CSS: es el resultado de compilar todos los ficheros **scss** que tenemos en desarrollo.
- un fichero JS: es el resultado de compilar todos los ficheros **js** que tenemos en desarrollo.

En cuanto la versión de producción se creó correctamente subí el proyecto a **GitHub** usando el programa **GitHub Desktop**. Una vez el proyecto estaba en el repositorio fui a mi cuenta de **Netlify** y la sincronice con mi cuenta de GitHub. Una vez sincronizada, me dispuse a hacer el despliegue del proyecto, creando una versión de producción y publicándola en Internet. Este paso falló, debido a que una imagen tenía caracteres en mayúscula. Una vez modificado el

nombre del archivo y actualizados todos los ficheros implicados volví a hacer el despliegue, esta vez con resultado positivo.

Enlace a GitHub: <https://github.com/ariolm/CV>

Enlace a Netlify: <https://pedantic-benz-2da15b.netlify.app/>