

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.

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.

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 solo va a tener una página, en la carpeta **styles/layouts**, he eliminado los estilos que no me hacían falta y he dejado el fichero listo para añadir los estilos que voy a usar en mi proyecto. También he creado el fichero **_functions.scss**, para introducir en él las funciones sass que necesite. En el fichero **_variables.scss**, he modificado los colores y he añadido variables de tamaño predefinidas y los colores por defecto que voy a utilizar. En **__functions.scss**, he creado una función tal que dado dos colores, devuelve el lineargradient de estos (posteriormente se utilizara para crear el fondo de la web).

Como el sitio web va a ser mi CV, he decidido utilizar la siguiente estructura:

- 1) Un primer bloque llamado **header**, donde se visualizará una foto mía y mi nombre, cada uno como un elemento del bloque principal.
- 2) Un bloque llamado **info** donde se visualizarán mis datos personales y la URL a mi perfil de LinkedIn.

- 3) Un bloque llamado **academic** donde se visualizarán mis datos académicos. Este bloque contendrá un elemento llamado **academic__table** donde estarán contenidos todos los datos.
- 4) Un bloque llamado **working** donde se visualizará mi experiencia laboral. Este bloque contendrá un elemento llamado **working__table** donde estarán contenidos todos los datos.
- 5) Un bloque llamado **languages** donde se visualizarán los idiomas que hablo y el nivel que tengo de cada uno. Este bloque contendrá un elemento llamado **languages__table** donde estarán contenidos todos los datos.
- 6) Un bloque llamado **software** donde se visualizarán mis conocimientos de software y una pequeña descripción de cada lenguaje. Este bloque contendrá un elemento llamado **software__table** donde estarán contenidos todos los datos.
- 7) Un bloque llamado **others** donde se visualizarán otros datos de interés sobre mi. Este bloque contendrá un elemento llamado **others__table** donde estarán contenidos todos los datos.

Como podemos ver, la estructura de la web se ha pensado siguiendo la metodología de estilo BEM. En cuanto a las clases definidas, se ha definido una **wildcard** para todas las clases **__table**, evitando duplicar los estilos para cada elemento.

Primero se ha pensado el sitio web para escritorio y luego se han añadido las media queries necesarias para que la fuente se adapte correctamente a cada dispositivo.

A continuación, se adjuntan unas capturas para ver el resultado final del sitio web en escritorio y en móvil.



Experiencia laboral

Community Manager (Tecnobee Innovación y Servicio S.L., Junio 2015 - Abril 2017)

Software Engineer Trainee (Idneo Technologies S.A.U, Abril 2017 - Diciembre 2017)

Software Product Engineer (Idneo Technologies S.A.U, Enero 2018 - Enero 2020)

Embedded Software Engineer (Circontrol, Enero 2020 - Actualidad)

Idiomas

Inglés B2

Castellano C1

Catalán C1

Conocimientos de Software

Python	Programación orientada a objetos
	Programación web con framework Django
C/C++/Qt	Programación de microcontroladores en C y ensamblador
	Programación sobre Kernel Linux en C y C++
	Programación C++ orientada a objetos
	Programación de aplicaciones para escritorio y embedded en Qt
Android	Programación avanzada de aplicaciones móviles Android y Java
Bases de datos	Gestión y modelado de bases de datos en SQLite, MySQL y Redis
Diseño web	Diseño frontend en HTML, CSS, Javascript y Angular
	Diseño web backend en NodeJS

Cursos

Desarrollo con NodeJS y Javascript avanzado

Git y Bitbucket avanzado

Integración continua con Jenkins

Introducción a Amazon Web Services (AWS)

Programming in C# (Certificación 70-483)

Otros datos de interés

Carné de conducir B1

Vehículo propio

Ilustración 1: Sitio web en versión escritorio



Xavier Ledesma Pons

21/04/1995

Terrassa, 08226

637 03 69 10

xaviledesma4@gmail.com

www.linkedin.com/in/xaviledesma

Preparación académica 

Máster de desarrollo de sitios y aplicaciones web (UOC, 2020-Actualidad)

Máster en desarrollo de aplicaciones para dispositivos móviles (UOC, 2018-2020)

Ingeniería en sistemas TIC (UPC, 2014-2018)

Experiencia laboral

Community Manager (Tecnobee Innovación y Servicio S.L., Junio 2015 - Abril 2017)

Software Engineer Trainee (Idneo Technologies S.A.U., Abril 2017 - Diciembre 2017)

Software Product Engineer (Idneo Technologies S.A.U., Enero 2018 - Enero 2020)

Embedded Software Engineer (Circontrol, Enero 2020 - Actualidad)

Idiomas

Inglés B2

Castellano C1

Catalán C1

Conocimientos de Software

Python

Programación orientada a objetos
Programación web con framework Django

C/C++/Qt

Programación de microcontroladores en C y ensamblador
Programación sobre Kernel Linux en C y C++
Programación C++ orientada a objetos
Programación de aplicaciones para escritorio y embedded en Qt

Android

Programación avanzada de aplicaciones móviles Android y Java

Bases de datos

Gestión y modelado de bases de datos en Sqlite, MySQL y Redis

Diseño web

Diseño frontend en HTML, CSS, Javascript y Angular
Diseño web backend en NodeJS

Cursos

Desarrollo con NodeJS y Javascript avanzado

Git y Bitbucket avanzado

Integración continua con Jenkins

Introducción a Amazon Web Services (AWS)

Programming in C# (Certificación 70-483)

Otros datos de interés

Carné de conducir B1

Vehículo propio

Ilustración 2: Sitio web en versión móvil

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 PEC1, 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://infallible-ptolemy-6d1cdf.netlify.app/>