Documentación

# **Autor:** Albert Andrés Vidueira

# **Contacto:** aandresvi@uoc.edu

# **Enlace al repositorio GITHUB:** <https://github.com/Anvii8/Eines-HTML-i-CSS.git>

# **URL pública de la web:** <https://calm-salamander-3519c3.netlify.app/>

# Creación del boilerplate basado en Parcel. Como se ha definido el entorno de desarrollo y producción, y cómo se configura el soporte para navegadores antiguos.

La creación del boilerplate consiste en tener una estructura y unos archivos que nos sirven de punto de partida de nuestro desarrollo de la página web. En este caso, para realizar este boilerplate seguí los pasos proporcionados en el Modulo 2, es decir, dispongo de un fichero package.json que comentaré más adelante, y dos carpetas, src y dist, donde en src encontramos los archivos que contienen el código de nuestra página web y en dist, los archivos resultantes los cuales han sido optimizados, compilados y transpilados para que se puedan implementar en un entorno de producción.

Dentro del fichero package.json vamos a centrarnos en 4 aspectos importantes:

* En primer lugar, en “source” indicamos cuál va a ser la página principal de nuestra web.
* En segundo lugar, la configuración indicada para el soporte a navegadores antiguos ha sido la siguiente: "browserslist": "> 0.5%, last 2 versions, not dead". Esto consiste en que estamos indicando que nuestro código ha de ser compatible con aquellos navegadores los cuales tengan una cuota de mercado de mas del 0.5%, se admiten las dos últimas versiones de cada navegador y se excluyen aquellos navegadores que entendemos que están muertos, es decir, que ya no reciben soporte ni mantenimiento.
* En tercer lugar, se han configurado dos accesos más rápidos para poder realizar tanto el build como el run de nuestros ficheros.
* Y, por último, vemos las dependencias, en las cuáles nos encontramos con npm, parcel y posthtml-includes, el cual será comentado en el siguiente apartado.

# Gestión de dependéncias: pre- o postprocesadores y dependencias adicionales.

Este apartado consiste en uso de herramientas externas las cuáles nuestra página web pueda necesitar durante el desarrollo.

En este caso se ha utilizado el pre-postprocesador de HTML PostHtml, ya que como había partes, como el header y el footer, las cuáles eran comunes para las diferentes páginas de la web.

El beneficio de su uso ha sido el tener un código más limpio y evitar repetir lo mismo en todos los ficheros, de esta manera antes de que el navegador procesase el HTML, se realizaba esta transformación de código y se añadían las partes del header y del footer.

Por lo tanto, en esta web, su uso ha sido como Preprocesador de HTML.

# Creación del repositorio Git

Este paso es esencial para poder desarrollar y disponer de nuestro repositorio para el código de la web. De esta manera podemos tener nuestro código accesible para cualquier persona. La principal razón en utilizar GitHub es por su comodidad y facilidad de uso que tiene, además de que es la herramienta más usada para este tipo de gestiones.

En mi caso, he utilizado el IDE Visual Studio Code desde el cual se puede sincronizar con el repositorio y trabajar sin problemas y dispone de una consola para realizar los comandos necesarios para mantener el repositorio actualizado.

Al tener el repositorio creado, ya se puede trabajar desde cualquier sitio y poderlo compartir con cualquier persona. En la 1ª página de este documento se encuentra el enlace al repositorio.

# Adecuación a la temática y estructura de la práctica.

La temática y estructura de la práctica estaba muy bien detallada en el enunciado. Por un lado, la temática de la práctica tenía que ser un portal sobre una serie que nos hubiese parecido interesante- Por otro lado, la estructura estaba muy bien especificada, indicando que apartados debía tener (Portada, Categoría, Detalle, Presentación y Enlaces) con una pequeña explicación sobre que debía de tener cada uno y, además, unos pequeños requisitos de navegación.

En mi caso, la serie que escogí fue Two Broke Girls, es una serie del estilo Big Bang Theory la cuál me pareció muy divertida. Referente a la estructura, he seguido los consejos y puntos indicados.

Si nos centramos en la estructura que vemos en la web, vemos que la primera pagina que vemos es la portada y nos encontramos con un navegador el cual nos permite ir a las páginas de presentación, categoría, y enlaces. Si vamos a la página de categoría, podemos encontrar 3 publicaciones referente a los personajes que para mi son los principales. Cada personaje hace referencia a otra pagina donde podemos ver detalles sobre cada uno, además, cada personaje está enlazado con los otros para no tener que volver a la página de categorías.

# Diseño responsive, complejidad y estética

Este apartado consiste en la apariencia y funcionalidad de la web.

Si nos centramos en la parte del diseño responsive nos referimos a que la página web sea capaz de adaptarse de una manera adecuada a diferentes dispositivos y tamaños de pantalla. La técnica utilizada ha sido media queries en CSS para ajustar el diseño en pantalla pequeñas, como pantallas de teléfono móvil en orientación vertical, *max-width: 599px*.

He decidido usar esta técnica para mostrar el navegador, el cuál permite ir hacía las otras páginas de la web, en forma de columna. Primeramente, quería mostrarlo en forma horizontal y que el usuario pudiese scrollear hacía la izquierda para desplazarlo. Esta primera idea permitía tener más elementos en el navegador, pero, a la vez más dificultosa. Como en nuestro caso no existían tantos elementos, la decisión fue ponerlo en forma de columna.

Referente a la complejidad, la estructura estaba bastante detallada y el objetivo ha sido realizar un blog el cuál fuese simple para navegar, fácil de entender y de interactuar.

Por último, la estética de la web es sencilla, excluyendo la página de enlaces, las otras incluyen imágenes lo cuál hace que sea más agradable a la visualización y he tratado de tener una disposición de los elementos lo más atractivo posible evitando que se viesen las imágenes en cualquier posición o que no estuviesen alineadas con el texto.

# Semántica y accesibilidad

Este apartado es importante para personas externas al proyecto cuando se disponen a visualizar el código de la web o la propia web no tenga problemas o dificultades.

Referente a la semántica, los elementos y etiquetas HTML utilizadas son bastante intuitivas cuando vemos el diseño de la página. Se han utilizado etiquetas de título como <h1>, <h2> y <h3>. Para la creación del navegador se ha utilizado la etiqueta <nav> juntamente con la etiqueta <a> para poder realizar los correspondientes enlaces. El texto que aparece se ha realizado con la etiqueta <p> y las lista que vemos se han realizado mediante <ul>, ya que en este caso no era necesario que fuesen ordenadas. Cada elemento de la lista ha sido declarado con la etiqueta <li>.

Para las imágenes se ha utilizado la etiqueta <img> y todo el contenido ha estado realizado dentro de una etiqueta <article>.

Tanto al principio de la etiqueta <body> como al final aparecen dos etiquetas <include> las cuales se han utilizado para añadir tanto el header como el footer. Como eran partes las cuales todas las páginas compartían se ha decidido realizarlo de esta manera.

Referente a la accesibilidad, se ha asegurado la web pueda ser usada y entendida para cualquier tipo de persona independientemente de sus capacidades. Dispone de un correcto contraste entre los títulos y los textos y cada imagen tiene el atributo alt, para los casos en los que la imagen no ha podido ser cargada saber que tipo de imagen iba en ese lugar.

# Publicación a internet

Este paso al igual que el de creación del repositorio Git es muy importante, ya que es el que nos permite ver la pagina web desde un simple enlace accesible para todo el mundo.

Se ha llevado a cabo porque por un lado lo requería la práctica, pero por otro lado porque es un paso esencial cuando hablamos del desarrollo de una pagina web, ya que el objetivo es que esté en internet y que las personas puedan visualizarla.

La publicación a internet se ha realizado mediante Netlify, el proceso ha sido muy sencillo: creación de la cuenta, se vincula con GitHub, se indica el repositorio y listo.

El resultado obtenido ha sido disponer de un enlace, el cual está en la 1ª pagina de este documento, accesible para todo el mundo.