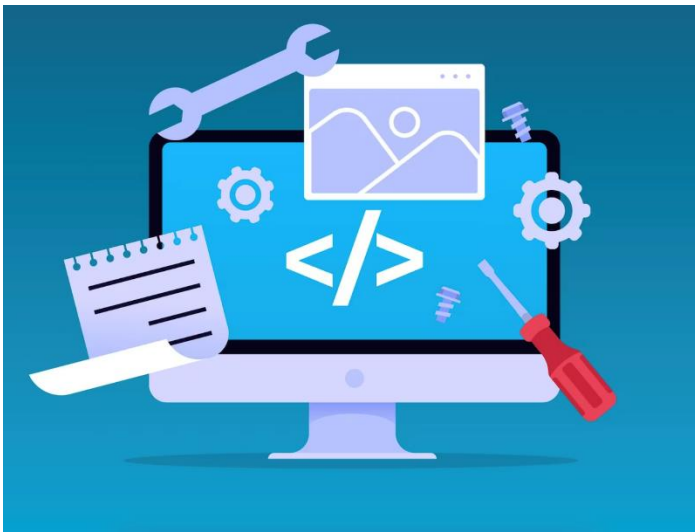


---

# Herramientas HTML y CSS

---

*P1 – Desarrollo de una web*



MARÍA CAÑAS ENCINAS

Enlace al directorio Github: <https://github.com/mariacanas/Herramientas-HTML-y-CSS---Master-Dw-UOC>

Enlace a la página principal del proyecto:

<https://mariacanas.github.io/Herramientas-HTML-y-CSS---Master-Dw-UOC/src/index.html>

## **ÍNDICE**

1. Creación del Boilerplate
2. Gestión de dependencias
3. Creación del repositorio de GitHub
4. Temática y estructura
5. Diseño del proyecto
6. Semántica y accesibilidad
7. Publicación en Internet

## **1. 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**

Para la creación del boilerplate se han seguido los siguientes pasos:

-Lo primero de todo se tiene que instalar Parcel, para ello desde la terminal vamos a ejecutar el comando ***npm install -g parcel-bundler***

-Después, se ha creado un nuevo directorio que va a ser donde se desarrolle nuestro proyecto. Desde consola vamos a navegar hasta el y vamos a inicializar un nuevo proyecto npm, para ello vamos a ejecutar el comando ***npm init -y*** este comando lo que va a hacer es crearnos el archivo “**package.json**”, en este fichero se van a guardar la configuración y las dependencias de nuestro proyecto

-Una vez instalado y creado el directorio del proyecto, se ha desarrollado el proyecto utilizando los lenguajes de HTML, CSS y JS. En nuestro caso es un proyecto sobre un grupo de música, para ello se han creado diferentes páginas HTML, un css para dar estilos a ese código y un Js. La página principal que será la portada de nuestro proyecto se llamará ***index.html*** y desde ella se van a poder acceder a los demás html.

-Después de crear el proyecto, con el código necesario y los estilos, se debe de ejecutar el proyecto por consola. Mediante el comando ***parcel index.html*** para así iniciar el servidor de desarrollo y poder abrir nuestro proyecto en el navegador. Al utilizar Parcel, se va a actualizar automáticamente cuando realicemos cambios en nuestros archivos del proyecto

Configuración soporte para navegadores antiguos

Para realizarlo, se va a utilizar la configuración de Babel y Autoprefixer, debido a que Parcel va a integrar de forma automática estas herramientas para manejar la compatibilidad con navegadores antiguos. Los pasos a realizar son:

-Instalación de las dependencias necesarias: Para ello tenemos que utilizar el siguiente comando ***npm install @babel/core @babel/preset-env babel-***

***loader autoprefixer --save-dev*** este comando lo que va a hacer es instalar Babel y autoprefixer en nuestro proyecto

-Una vez instalado, vamos a configurar nuestro parcel para construir el proyecto. En nuestro archivo ***package.json*** tenemos definidos los scripts que van a iniciar y ejecutar nuestro proyecto. Tenemos que ejecutar por consola ***npm start*** y ***npm run build*** para compilar nuestro proyecto y así Parcel se encarga de forma automática del soporte de nuestro proyecto para navegadores antiguos sin tener que preocuparnos por la compatibilidad con estos navegadores.

## ***2. Gestión de dependencias: pre o postprocesadores y dependencias adicionales***

### **Pre/post processadores utilizados:**

En el desarrollo del proyecto se ha utilizado el preprocesador Sass, debido a que facilita escribir código CSS de una forma más eficiente, además de tener funcionalidades que no están disponibles en el CSS estándar. Para ello, lo primero que he realizado ha sido instalarlo, mediante npm introduciendo el siguiente comando por consola ***npm install -g sass***. Dentro de mi proyecto al principio lo que hice fue hacer todo el código en un archivo style.css y una vez finalizado el código lo pasé a un fichero con extensión .scss y realizando modificaciones para poder aprovechar las características que tenía sass. Creando así un archivo con extensión ***.scss*** y borrando el que tenía con extensión ***.css***.

Me he decantado por utilizar este preprocesador y realizar el cambio de css a scss, ya que he querido aprovechar las características que tenía sass como el uso de variables, el anidamiento de reglas...

Ya que Sass nos ofrece diferentes funcionalidades que tienen el objetivo de mejorar la eficiencia del desarrollo web, dentro de estas funcionalidades me gustaría destacar el uso de definir variables para utilizar valores que sean reutilizables, el anidamiento de reglas para así conseguir una estructura más clara.

Dentro de los beneficios que nos aporta Sass también me gustaría destacar que nos ayuda en cuanto a mantener el código ya que nos permite tenerlo organizado de una manera más estructurada, facilitándonos así la identificación y corrección de errores como la incorporación de nuevas funcionalidad o modificación de diseño. También nos ayuda a la productividad, ya que nos la mejora porque nos proporciona herramientas como variables o mixins las cuales nos permiten escribir y modificar los estilos de una forma más rápida.

Un ejemplo del cambio de css a scss se puede ver en las siguientes imágenes:

```

33 | top: 0;
34 | }
35 |
36 | nav{
37 |   margin: auto;
38 |   text-align: center;
39 | }
40 |
41 | nav ul{
42 |   display: flex;
43 |   align-items: center;
44 |   gap: 30px;
45 |   font-size: 20px;
46 |   font-weight: bold;
47 | }
48 |
49 | nav ul li a{
50 |   color: white;
51 | }
52 |

```

```

32 | position: fixed;
33 | top: 0;
34 | }
35 |
36 | nav {
37 |   margin: auto;
38 |   text-align: center;
39 | }
40 |
41 | ul {
42 |   display: flex;
43 |   align-items: center;
44 |   gap: 30px;
45 |   font-size: 20px;
46 |   font-weight: bold;
47 | }
48 |
49 | li a {
50 |   color: white;
51 | }

```

```

60 |
61 | .redes {
62 |   display: flex;
63 |   gap: 30px;
64 |   font-size: 20px;
65 |   margin-right: 20px;
66 | }
67 |
68 | .redes a {
69 |   color: white;
70 | }
71 |
72 | .redes a:hover {
73 |   color: var(--color-letra);
74 |   cursor: pointer;
75 | }
76 |
77 | #inicio {

```

Como podemos observar, el código de la izquierda está en css y el de la derecha está en scss y este va a estar escrito de una forma más legible y estructurada ya que va a contar con reglas anidadas.

## Dependencias externas utilizadas:

### AOS

Aos es una herramienta que nos permite agregar animaciones a nuestro sitio web, las cuales se activan cuando el usuario hace scroll mientras visita la página. He optado por utilizar esta dependencia porque quería mejorar la experiencia del usuario, facilitándome así captar su atención y que su navegación fuera más fácil y dinámica.

Para instalarlo, lo que hice fue ejecutar el siguiente comando por consola ***npm i aos --save*** el cuál va a descargar la biblioteca AOS y va a añadir la dependencia en nuestro fichero package.json.

Para incorporarlo en nuestro proyecto, tenemos que importarlo a nuestro código HTML y posteriormente inicializarlo. Para ello se ha utilizado el siguiente código:

```
<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
<script>
  |   AOS.init();
</script>
```

Y luego ya sería añadirlo a la parte o elemento del código donde queramos que se ejecute esta animación. Con data-aos="Efecto que queremos que haga"

Dentro de los efectos, podremos utilizar varios valores como por ejemplo "fade", "fade-up", "zoom-in" ...

```
<section id="conciertos">
  <h3>Proximos Conciertos</h3>
  <div class="conciertos" data-aos="fade-right">
    |   <div class="concierto">
```

### Autoprefixer

Autoprefixer viene ya integrado con Parcel. Esta herramienta analiza nuestro código CSS y agregar de forma automática los prefijos de navegador necesarios que nos garanticen la compatibilidad de nuestro proyecto con diferentes navegadores. Por lo que nos va a ahorrar tiempo al no tener que escribir de forma manual los prefijos en los diferentes entornos de navegación.

### Normalize.css

Normalize es un archivo Css que nos proporciona estilos predeterminados para los elementos HTML en los diferentes navegadores, garantizándonos una apariencia más consistente en la web.

Para instalarlo, lo que hice fue ejecutar el siguiente comando por consola ***npm i normalize.css --save*** el cuál va a descargar y va a añadir la dependencia de Normalize.css en nuestro fichero package.json. Luego lo importe en mi código CSS para utilizarlo en mi proyecto:

```
@import "../node_modules/normalize.css/normalize.css";
```

Principalmente me he decido utilizar esta dependencia porque nos ofrece una base sólida para los estilos, lo cual nos garantiza que los elementos HTML se vean igual sin depender del dispositivo o navegador por el que estamos accediendo a la página web. Pudiendo así ofrecer una experiencia de usuario más uniforme y facilitando su desarrollo al proporcionarnos una base sólida en cuanto a los estilos.

### **3. Creación del repositorio Git**

Para publicar el código de la página web, se ha creado un repositorio en Github donde se subirán las diferentes entregas de la asignatura junto con las carpetas y código correspondiente, con el nombre de Herramientas HTML y CSS – Master Dw UOC.

Dirección al repositorio: <https://github.com/mariacanas/Herramientas-HTML-y-CSS---Master-Dw-UOC>

Dentro del repositorio se ha creado una carpeta **P1 Desarrollo de una web**, donde se ha subido la documentación y código correspondiente a esta entrega.

### **4. Adecuación a la temática y estructura de la práctica**

La página web que se ha desarrollado en este proyecto, tenía como temática un grupo musical y yo decidí realizarlo sobre el artista Nil Moliner.

Estructura de las diferentes páginas que componen el proyecto web.

Todas las pantallas van a estar formadas por una barra de navegación donde se podrá ver el nombre del artista junto con enlaces a las diferentes páginas que forman nuestro proyecto.

#### **Página Portada (index.html)**

La página portada es la primera impresión que van a tener los visitantes que acceden a nuestro sitio web. En ella se presenta información importante como los próximos conciertos de nil, ofreciendo fechas y ubicaciones sobre cada uno de los conciertos. También se destacan algunas de las últimas canciones de su ultimo Álbum, en el que se podrá ver el título de la canción junto con su duración. Y por último se incluye una opción para que los usuarios puedan introducir su nombre y la dirección de correo electrónico, el cual les va a permitir recibir noticias, actualizaciones y animaciones sobre Nil Moliner.

#### **Página Presentación (información.html)**

La página de información va a proporcionar una visión más profunda sobre la vida y la carrera de Nil. Ya que en ella se va a mostrar un resumen de su biografía, desde sus primeros años hasta su ascenso en la industria musical. También se destacan datos y anécdotas interesantes sobre el artista ofreciendo así una vista diferente y más personal. Además, se detallan varias de las colaboraciones musicales que ha realizado junto con otros artistas, destacando así las colaboraciones que más me han parecido interesantes y más me han gustado.

### **Página Categorías (Categorías.html)**

En la página de Categorías, los usuarios van a poder explorar diferentes aspectos de la carrera de Nil de una manera organizada y estructurada. Van a tener una sección que está dedicada a los principales premios que ha recibido por su trabajo. Después, van a tener una sección donde van a poder ver los diferentes discos que ha publicado el artista, en el que en uno de ellos (El tercer y último disco) si acceden al botón de Ver Canciones se les va a redirigir a la página de detalles1Album (explicada a continuación). Y por último, los usuarios se van a encontrar con noticias/entrevistas que ha realizado en la prensa (Van a poder acceder a la entrevista completa al pulsar sobre el texto de Leer más)

### **Página Detalles de un Álbum (detalles1Album.html)**

**En esta página de detalles de Álbum, los usuarios van a poder acceder desde el botón de “Ver Canciones” del último Álbum situado en la página de Categorías.** Dentro de ella, los usuarios se van a encontrar con una breve explicación de este disco, junto con el listado de canciones que lo componen y una entrevista que ha realizado Nil explicando este Álbum.

### **Página Enlaces (Enlaces.html)**

Esta página sirve como fuente de referencia para los usuarios que están interesados en obtener más información sobre el trabajo de Nil Moliner. Donde se van a recopilar los enlaces a fuentes externas, en el que se van a incluir entrevistas, artículos de prensa, acceso a videoclips y sitios web oficiales relacionados con el artista. El principal objetivo es proporcionar a los usuarios información adicional a parte del que ya está explicado en nuestro proyecto.

## **5. *Diseño responsive, complejidad y estética***

En este proyecto, en el cual se ha desarrollado una página web sobre el músico Nil Moliner. El objetivo ha sido proporcionar a los seguidores y fans de este artista una experiencia donde puedan descubrir más sobre su música, conciertos, historia...

Durante el desarrollo de la página web, se ha dado prioridad a hacer que el sitio sea adaptable en diferentes dispositivos e intuitiva y fácil de usar para todos los usuarios.

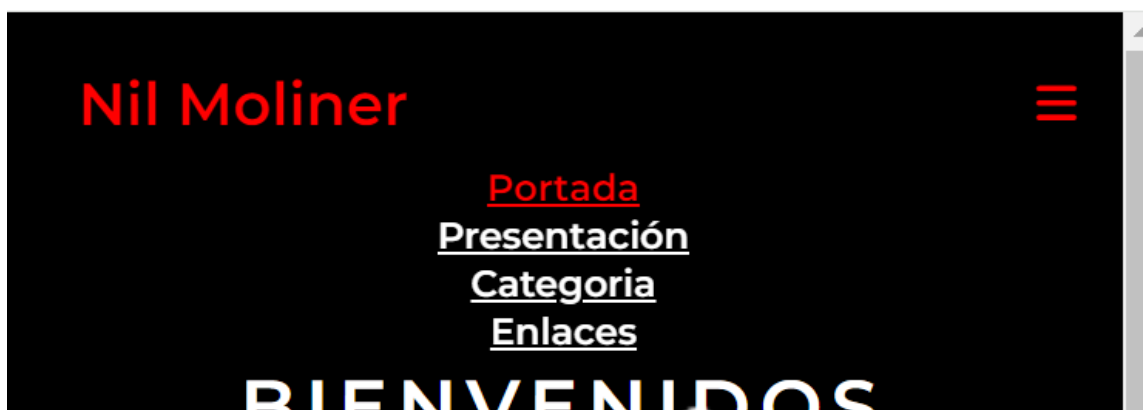
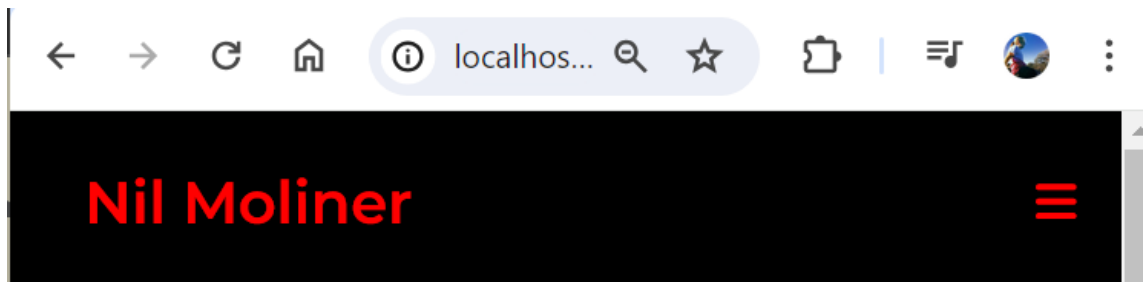
Para adaptar la página a todos los tipos de pantalla de los dispositivos por el cual accede el usuario, se ha implementado un diseño responsive con el objetivo de que nuestra aplicación funcione de una forma optima y el usuario no se encuentre con ningún problema. Un ejemplo de ello, es la barra de navegación entre las diferentes pantalla, ya que si el usuario accede desde un



ordenador (La pantalla es más grande) por lo que podrá ver por completo toda la barra, como se muestra en la siguiente imagen.



Sin embargo, si el usuario accede a la página utilizando un dispositivo que tiene la pantalla más pequeña, la barra de navegación se va a “esconder” y se va a mostrar un icono que al pulsarle se le mostrará la barra de navegación con las diferentes opciones colocadas en bloque, si el usuario vuelve a pulsar sobre ese icono esta barra se esconderá.



Otra de las principales prioridades en las que me he centrado a la hora de desarrollar el proyecto, ha sido la de conseguir que la navegación sea intuitiva y accesible, para ello he querido garantizar que la página web fuera fácil de usar para todos los usuarios independientemente si contaban ya con experiencia en la navegación por páginas web. Para lograr esto, he diseñado una navegación sencilla y fácil de entender, con botones, enlaces y etiquetas que fueran descriptivas y claras, con el objetivo de que el usuario pudiera realizar y obtener la información deseada sobre el artista sin encontrar ninguna dificultad.

Sobre los colores/estilo, en todas las páginas se ha utilizado un fondo oscuro, junto con un color de letra blanca y roja, este último se ha utilizado por ejemplo en la barra de navegación para saber en que pantalla se encuentra el usuario, al pasar el cursor por encima de un enlace/Boton este también se

pondrá rojo para identificarlo como un enlace. En cuanto al contenido multimedia se han añadido imágenes / Videos donde se puede ver información sobre el artista, colaboraciones, actuaciones, cortes de prensa...

## **6. Semántica y accesibilidad**

En el proceso de desarrollo del proyecto web, he dedicado gran importancia a dos aspectos que son la semántica y la accesibilidad. Ambos, no solo tienen un gran peso a la hora de garantizarnos que nuestro sitio sea comprensible, si no que también sea accesible para todos los usuarios.

En la estructura del código HTML se han utilizado etiquetas semánticas como **<header>**, **<nav>**, **<main>**, **<section>**, **<footer>** con el objetivo de tener una mejor organización y una estructura clara para que en la navegación de los usuarios por la página web no tengan ningún problema.

En cuanto a las imágenes utilizadas en el proyecto, se ha incluido el atributo **alt**, el cual proporciona un texto alternativo para así describir el contenido de la imagen.

En cuanto al tema de los enlaces, se ha introducido un texto que sea fácil de comprender por los usuarios de que tienen la posibilidad de pulsar sobre él y acceder a más información, se han utilizado enlaces con el texto “Leer más”, “Ver la entrevista completa”...

Sobre las pruebas de accesibilidad, se han utilizado validadores de código (Aunque no son herramientas de accesibilidad) HTML y CSS, como W3C Markup Validator y W3C CSS Validator con el objetivo de identificar problemas en nuestra estructura y estilo, dado que estos podrían afectar a la accesibilidad. También al realizar el proyecto en modo responsive, se ha probado la página con diferentes tamaños de pantalla con el objetivo de garantizar que el sitio sea igualmente accesible para todos.

## **7. Publicación en internet**

Para la publicación de la página web en internet se ha utilizado Github Pages.

Las direcciones de acceso al directorio creado en Github y a la página principal (index.html) del proyecto son:

Enlace al directorio Github: <https://github.com/mariacanas/Herramientas-HTML-y-CSS---Master-Dw-UOC>

Enlace a la página principal del proyecto:

<https://mariacanas.github.io/Herramientas-HTML-y-CSS---Master-Dw-UOC/src/index.html>