

P1

# Desarrollo de una web

Autor: **Néstor Bethencourt Santana**  
Asignatura: **Herramientas HTML y CSS**  
Fecha: **21/04/2024**

|   |          |
|---|----------|
| <b>Boilerplate.....</b>                               | <b>3</b> |
| <b>Gestión de dependencias.....</b>                   | <b>4</b> |
| <b>Repositorio Git.....</b>                           | <b>6</b> |
| <b>Adecuación y estructura.....</b>                   | <b>6</b> |
| <b>Diseño responsive, complejidad y estética.....</b> | <b>7</b> |
| <b>Semántica y accesibilidad.....</b>                 | <b>7</b> |
| <b>Publicación a internet.....</b>                    | <b>8</b> |

# Boilerplate

Se ha creado una plantilla (boilerplate) de estructura de ficheros y conjunto de herramientas para comenzar fácilmente un desarrollo.

Como gestor de paquetes se utiliza npm, y como module bundle, Parcel.

Se crea *package.json* con `npm init -yes`

Para instalar Parcel `npm install --save-dev parcel`

Se han configurado los scripts para hacer el *start* y el *build* de tal forma que se haga un *clean* primero para poder generar los ficheros nuevamente y no vaya creciendo las copias indefinidamente.

He añadido como punto de entrada *home.html*. Además, he definido los navegadores soportados, para que Parcel a la hora de transpilar asegure la compatibilidad con dichos navegadores. En concreto, los navegadores soportados serán los utilizados que tienen más de un 0.5% de uso global, las dos últimas versiones de los navegadores principales y los que tengan aún soporte.

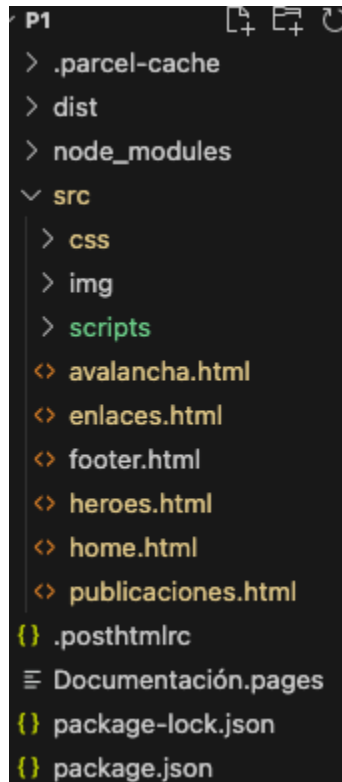
```
"source": "/src/home.html",
"browserslist": "> 0.5%, last 2 versions, not dead",
"scripts": {
  "parcel:dev": "parcel",
  "parcel:build": "parcel build",
  "clean": "rimraf dist .parcel-cache",
  "start": "npm-run-all clean parcel:dev",
  "build": "npm-run-all clean parcel:build",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

Para que los comandos start y build se ejecuten correctamente he instalado *rimraf* y *npm-run-all*.

```
npm install --save-dev rimraf npm-run-all
```

El paquete *rimraf* permite borrar directorios de forma recursiva y *npm-run-all* permite ejecutar múltiples scripts *npm* desde el archivo *package.json*.

En el directorio principal *src* se encuentran los ficheros *html* y las carpetas *css*, *img* y *scripts*, que contienen las hojas de estilo, las imágenes y el código de script respectivamente.



Para el entorno de desarrollo se utiliza *npm run start*. En *http://localhost:1234* podemos ver los cambios sin tener que refrescar la página.

Para producción se ejecuta *npm run build*, lo que crea la carpeta *dist* con todo el código de producción optimizado.

## Gestión de dependencias

He instalado ***posthtml-include*** para poder incluir código HTML de un fichero en otro fichero HTML.

*npm i -D posthtml-include*

He configurado *posthtml-include* en el fichero *.posthtmlrc* indicando la ruta raíz para el proyecto.

```
"plugins": {
  "posthtml-include": {
    "root": "src"
  }
}
```

La dependencia anterior me ha permitido, por ejemplo, utilizar un mismo *footer* para todas las páginas creadas, facilitando de esta manera el mantenimiento del código.

Con el siguiente include se inserta el código de footer.html

```
<footer>
  <include src="footer.html"></include>
</footer>
```

He utilizado **SASS** como preprocesador CSS, ya que permite utilizar variables, anidamiento, etc., que ayuda a escribir un CSS más mantenible. En este caso, *Parcel* detecta los ficheros .sass e instala automáticamente el plugin *@parcel/transformer-sass*.

En la siguiente figura se puede ver el código en SASS y a la derecha ya transformado a CSS (en este caso en la versión de desarrollo para que se pueda entender más fácilmente mientras se desarrolla).



```
body
  font-family: $sans-serif-font
  font-weight: 400
h1
  font-size: 2rem
h2
  font-size: 1.7rem
h3
  font-size: 1.4rem
h4
  font-size: 1.1rem
p
  font-size: 0.6rem

body {
  font-family: Roboto Condensed, sans-serif;
  font-weight: 400;
}
h1 {
  font-size: 2rem;
}
h2 {
  font-size: 1.7rem;
}
```

*Parcel* también incluye el postprocesador **PostCSS** con **Autoprefixer**, lo que permite que se incluyan automáticamente los prefijos de proveedores a propiedades de CSS, asegurando así la compatibilidad con los navegadores especificados en el *browserlist* del *package.json* detallado anteriormente.

He añadido un validador de HTML, para poder verificar automáticamente si el HTML cumple el estándar y es válido.

```
npm install --save-dev w3c-html-validator
```

A continuación se puede ver un ejemplo de un warning:

```
Server running at http://localhost:1234
✨ Built in 2.14s
(base) nestorbs@MacBook-Pro-de-NESTOR P1 % npm run validate-html

> p1@1.0.0 validate-html
> html-validator --quiet

[11:22:33] w3c-html-validator x fail src/publicaciones.html (messages: 6)
[11:22:34] HTML warning: Article lacks heading. Consider using "h2"- "h6" elements to add identifying headings to all articles.
[11:22:34] line 144, column 13:      <article class="band-member">\n
```

## Repositorio Git

Para crear el repositorio Git, he ejecutado los siguientes comandos:

```
git init // inicializa el repositorio
git add .posthtmlrc Documentación.pages css header.html
index.html js package-lock.json package.json //añade los ficheros a ser
supervisados
git commit -m "Parcel configured" //realiza un commit
git branch -M main //se cambia a main el nombre de la rama principal
git remote add origin
git@github.com:nbethencourts/herramientas-html-P1.git //se indica
cual es el repositorio en remoto
git push -u origin main //se suben los cambios al repositorio remoto
```

## Adecuación y estructura

A continuación se detalla lo que representa cada página atendiendo a lo solicitado en el enunciado de la práctica:

| Página enunciado | Página proyecto |
|------------------|-----------------|
| Portada          | home.html       |

|              |   |
|--------------|---|
| Presentacion | heroes.html<br>Parte de la historia del grupo                                   |
| Categoria    | publicaciones.html<br>discografía, componentes del grupo,<br>últimos conciertos |
| Enlaces      | referencias.html  |
| Detalle      | avalancha.html<br>Detalle del álbum Avalancha                                   |

## Diseño responsive, complejidad y estética

He utilizado un enfoque mobile first para realizar la implementación del diseño, es decir, he partido de un diseño que se ajusta a un dispositivo móvil, y he ido modificando las páginas para ir aprovechando el espacio, adaptando el diseño a medida que el tamaño de pantalla es más grande.

Si bien he podido utilizar en algunos casos grid, he resuelto los problemas planteados utilizando flexbox porque encajaba bien con el diseño que quería implementar.

He intentado plasmar un diseño acorde al grupo de rock sobre el que se hace la página.

## Semántica y accesibilidad

Para comprobar que el contraste de los colores cumple con los estándares de accesibilidad he utilizado <https://color.adobe.com/es/create/color-contrast-analyzer>

Algunos de los landmarks utilizados, que aportan semántica a las páginas son:

- **aside**: para agrupar los artículos relacionados con el artículo principal de la página de detalle.
- **nav**: para el menú de navegación que se muestra en la cabecera de las páginas.
- **footer**: utilizado para el pie de las páginas.
- **header**: cabecera de las páginas.
- **main**: etiqueta para agrupar el contenido principal de cada página.

- **article:** para agrupar información que es independiente y autocontenida, como la información del álbum Avalancha (página correspondiente a detalle).

Para validar el HTML he utilizado el validador de W3C mediante el paquete comentado en el apartado de gestión de dependencias.

## Publicación a internet

Para realizar la publicación en Internet he utilizado Netlify. Me he creado una cuenta gratuita en esta plataforma y la he vinculado con mi cuenta de Github, de tal forma que cada vez que subo cambio a Github, se actualiza la web con el nuevo código fuente.

La url pública de la web es <https://nbethencourtsp1.netlify.app/home> y el repositorio en Github <https://github.com/nbethencourts/herramientas-html-P1/tree/main>