Xavier Serrano Salvador

Eines HTML i CSS

# PEC 1

## BOILERPLATE BASADO EN PARCEL

Versión de node:

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

Creamos el package json por defecto:

Texto

Descripción generada automáticamente

Y se borra el campo main para evitar interferencias con la configuración de Parcel:

Texto

Descripción generada automáticamente

### Instalación Parcel



Se crea un documento index.html i se empieza el servidor Parcel apuntando a este mismo documento. Vemos como se crea la carpeta *dist* en la que Parcel hará los cambios en tiempo real y los optimizará.

Texto

Descripción generada automáticamente



localhost:1234:

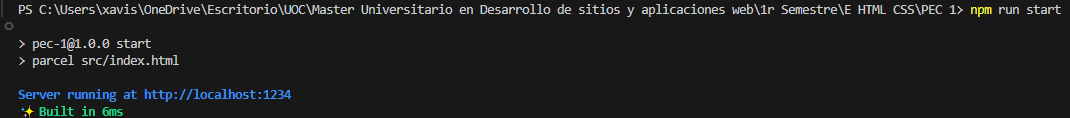
Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Configuración del package.json con dos scripts:

Texto

Descripción generada automáticamente

* *“start”* -> entorno de desarrollo. Se crea el servidor web en local en el navegador. 

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

* *“build”* -> entorno de producción (archivos que se subirán a internet lo más optimizados y ligeros posibles).

Texto

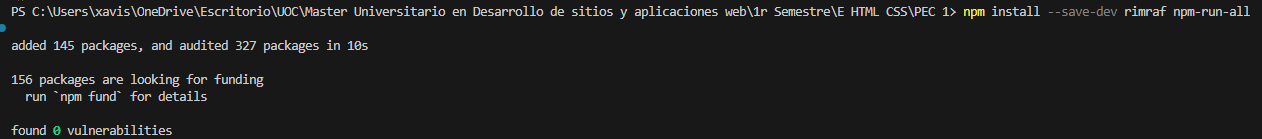
Descripción generada automáticamente

Imagen en blanco y negro

Descripción generada automáticamente con confianza baja

### Instalación rimraf

Para optimizar el desarrollo y evitar que se dupliquen las carpetas cada vez que se ejecuta npm run build, se instalará rimraf y se configurará el package.json de la siguiente manera:



Antes:

Texto

Descripción generada automáticamente

Después de la instalación de *rimraf* y utilizar sus funcionalidades en los scripts de *“start”* y *“build”*:

Texto

Descripción generada automáticamente

### Soporte para navegadores antiguos

Declaramos el *“browserlist”* cuya funcionalidad viene por defecto en Parcel para que transpile los archivos según su compatibilidad con los navegadores desde los que se abra la web.

Texto

Descripción generada automáticamente

## CONFIGURACIÓN DEL ARCHIVO Y REPOSITORIO EN GITHUB

Para poder trabajar sin miedo a perder los cambios, es recomendable utilizar un controlador de versiones. En este caso se ha decidido utilizar GitHub.

Para ello, se ha creado un repositorio en la página de [GitHub](https://github.com/Ivax7/PEC1-E-HTML-i-CSS) y se ha enlazado con la carpeta con los archivos correspondientes a la PEC1.

Captura de pantalla de un teléfono celular

Descripción generada automáticamente

## ADECUACIÓN A LA TEMÁTICA Y ESTRUCTURA DE LA PRÁCTICA

### Metodología

Para la nomenclatura de las clases utilizaremos la metodología [BEM](https://blog.ida.cl/desarrollo/metodologia-bem-desarrollo-front-end/) (Bloque, Elemento, Modificador). El método propone dividir la interfaz del usuario por bloques independientes para crear componentes fáciles de escalar, mantener o reutilizar en nuestras hojas de estilo. Se ha decidido por esta metodología pues es con la que más experiencia se ha trabajado.

### Dependencias

***npm install swiper:***

La instalación de la librería ***Swiper*** a través de npm install swiper es debida a su versatilidad y facilidad de uso en la creación de *sliders* de imágenes y galerías. Esta herramienta es particularmente útil para presentar visualmente lugares turísticos, eventos y paisajes, permitiendo a los usuarios interactuar con el contenido de manera dinámica y atractiva. En este proyecto se utilizará para mostrar distintas imágenes de las zonas que se recomiendan visitar. Pese a que será necesario el uso de JavaScript, el número de líneas se reducirá mucho y quedará mucho más visible, limpio y fácil de entender.

Entre sus características más utilizadas son las transiciones fluidas, opciones de personalización y soporte para dispositivos móviles, garantizando una experiencia de usuario optimizada en todas las plataformas. Además, su comunidad activa y documentación extensa facilitan la implementación y resolución de problemas, aspecto muy importante pues no se conocía su funcionamiento antes de desarrollar el proyecto y con vídeos de YouTube, documentación oficial y ChatGPT se ha aprendido a utilizar.

**npm install fancybox**

### Temática del proyecto

El proyecto se enfoca en el "Triángulo Friki" de Barcelona, una zona emblemática que se ha consolidado como un refugio para los amantes de la cultura geek y alternativa. Situado entre el Arco de Triunfo y el barrio del Born, este espacio alberga una variedad de tiendas especializadas que ofrecen productos como cómics, mangas, juegos de rol, figuras de colección y merchandising de ciencia ficción y fantasía.

**Lugares Destacados:**

1. **Calle Bailén:** Aquí se encuentran **Norma Cómics**, conocida por su extensa colección de cómics y su compromiso con la promoción de la cultura gráfica, y **Gigamesh**, que se especializa en literatura fantástica y ofrece un espacio para eventos culturales
2. **Calle Ali Bei y Zona del Arco de Triunfo:** En esta área destaca, así como **Otamashi Merchandising**, que cuenta con una variada gama de productos de series y videojuegos. **Kaburi**, que combina tienda y cafetería, siendo un punto de encuentro para jugadores de mesa y rol.
3. **El Born y Passeig de Sant Joan:** **Continuará Cómics**, con su catálogo de ediciones raras, y **Arkham Comics**, que celebra la pasión por el cómic y la cultura pop, añaden al encanto de esta zona.

**Eventos y Actividades:** Además de las tiendas, el "Triángulo Friki" es un punto dinámico de eventos como presentaciones de libros, firmas de autores y actividades relacionadas con el Salón del Manga y Comic Barcelona, creando un entorno vibrante y atractivo para los entusiastas de la cultura pop.

Este proyecto no solo será una guía temática de lugares geek en Barcelona, sino también una celebración de la historia, curiosidades y el ambiente que rodea a estas tiendas emblemáticas.

### Esbozo

Se crea un esbozo utilizando [Excalidraw](https://excalidraw.com/) para tener una idea general de la estructura de la web y sus componentes. Más adelante viendo como queda cada elemento se decide si cambiarlo, eliminarlo o reestructurarlo.

### Texto Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Enlaces** Una página con documentación y enlaces a las fuentes de las que se ha sacado el contenido y atribuciones a los derechos copyleft cuando sea necesario. En caso de haber utilizado herramientas de IA generativas **para la obtención del contenido**, tenéis que hacerlo constar en este apartado (recordad que no se admite el uso de estas herramientas para el desarrollo de la práctica).

**DISEÑO RESPONSIVE**

El proceso de decisión para implementar un diseño web responsive implica una serie de etapas que garantizan que el sitio se adapte adecuadamente a dispositivos móviles y tablets.

La mejora de la experiencia del usuario, la optimización para motores de búsqueda y la reducción de costos a largo plazo al mantener un solo sitio en lugar de versiones separadas para dispositivos móviles son motivos más que suficientes para diseñar la página web responsive.

Se crean esbozos que representan cómo se verá la página en diferentes dispositivos (con [Excalidraw](https://excalidraw.com/), por ejemplo), asegurando que los elementos clave, como menús, imágenes y botones, se adapten a distintas resoluciones.

Después, se implementa el diseño responsive en un entorno de prueba. Se realizan pruebas exhaustivas en diferentes dispositivos y navegadores para garantizar que la experiencia del usuario sea fluida y que la página cargue rápidamente. Las herramientas de desarrollo son útiles en esta fase para identificar y solucionar problemas antes del lanzamiento.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

En este proyecto se han fijado ciertas resoluciones de pantalla por píxeles según los dispositivos:

* **Tablets:** 1196px
* **Movil:** 768px

**Semántica y accesibilidad**

**Publicación a internet**

Se inicia sesión en Netflify:

Captura de pantalla de un celular

Descripción generada automáticamente

## BIBLIOGRAFIA

Creating a package.json file | npm Docs. (s. f.). <https://docs.npmjs.com/creating-a-package-json-file#creating-a-default-packagejson-file>

Resolving EACCES permissions errors when installing packages globally | npm Docs. (s. f.). https://docs.npmjs.com/resolving-eacces-permissions-errors-when-installing-packages-globally

Building a web app with Parcel. (s. f.). https://parceljs.org/getting-started/webapp/

Building a web app with Parcel. (s. f.-b). <https://parceljs.org/getting-started/webapp/#declaring-browser-targets>

▷ El Triángulo Friki de Barcelona 【Todo lo Que Debes Saber】. (2019, 28 agosto). World Experience. https://weareworldexperience.com/es/whats-on/el-triangulo-friki-de-barcelona-todo-lo-que-debes-saber/

Swiper - the most modern mobile touch slider. (s. f.). Swiper. <https://swiperjs.com/>

Adrian Twarog. (2020, 25 abril). Swiper JS Tutorial | Carousel Slider with SwiperJS [Vídeo]. YouTube. https://www.youtube.com/watch?v=4oyj\_smPAjc

Triángulo Friki de Barcelona – Google My Maps. (s. f.). Google My Maps. https://www.google.com/maps/d/u/0/viewer?mid=1gpeXUUgSvGyrjwx2vMovJeoah\_S9CtBY&femb=1&ll=41.391533478776%2C2.1788917423484033&z=18