

# PROYECTO FINAL DAWEC CON REACT ROUTER

**Autores:** José Antonio, Daniel Marín

**Asignatura:** Desarrollo de Aplicaciones Web en Entorno Cliente

**Curso:** S2DAW

---

## 1. Descripción de la práctica

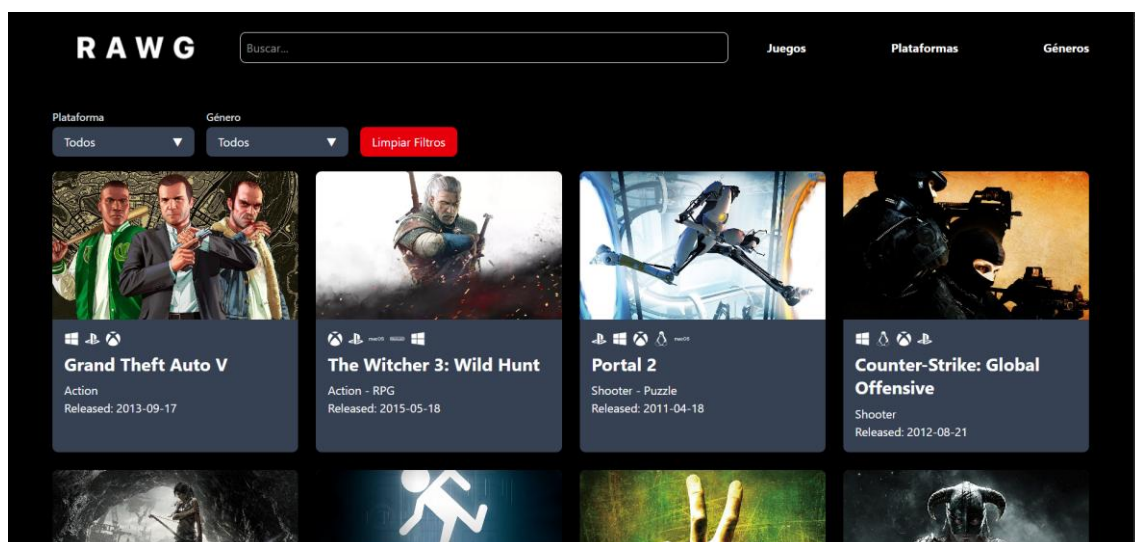
El objetivo de esta práctica consiste en acceder a los datos de una API para mostrarlos mediante una aplicación de página única utilizando **React Router**, **TypeScript** y **TailwindCSS**.

Nuestra página obtiene los datos de una API de videojuegos llamada [RAWG](#) y en nuestro proyecto listamos, filtramos y mostramos información de videojuegos, plataformas y géneros.

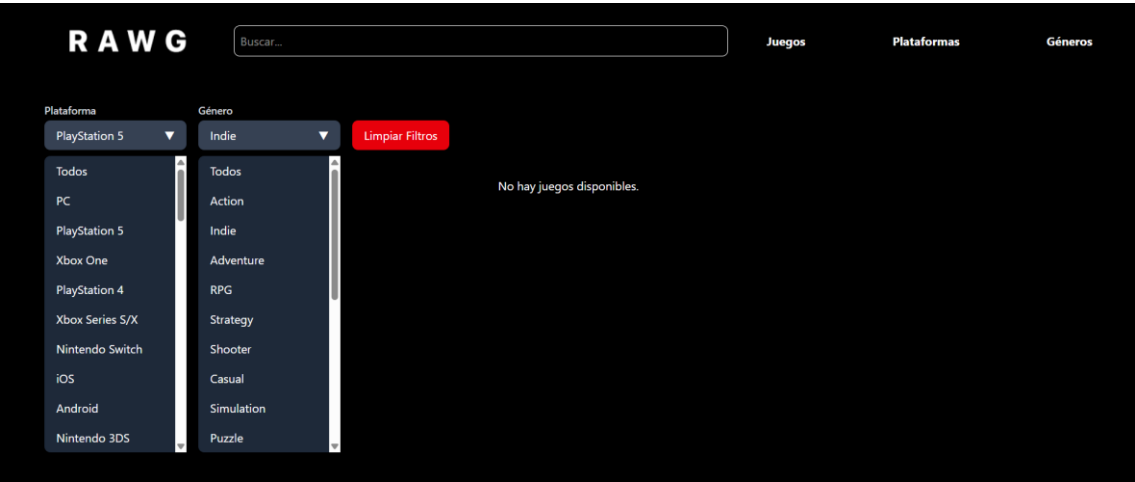
---

## 2. Explicación de la aplicación

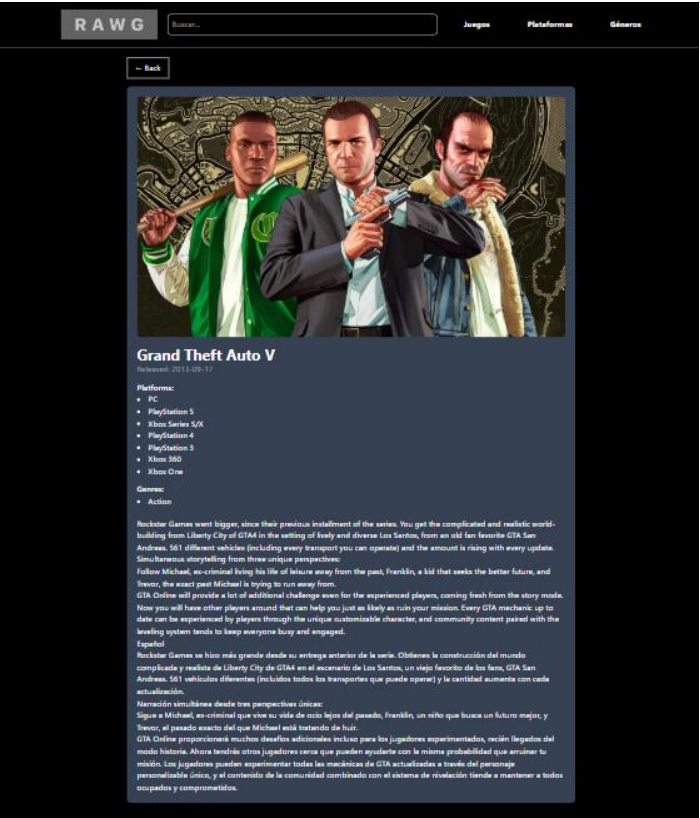
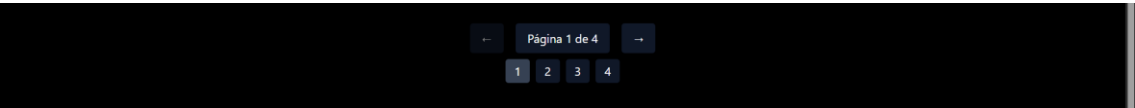
Nuestra aplicación consiste en una página web llamada **Games**, que contiene la lista de todos los videojuegos paginados de 12 en 12 y muestra información básica como su nombre, fecha de lanzamiento, género y plataformas.



En esta página se encuentran disponibles dos tipos de filtrados, que permiten clasificar los juegos por **géneros** y **plataformas**.



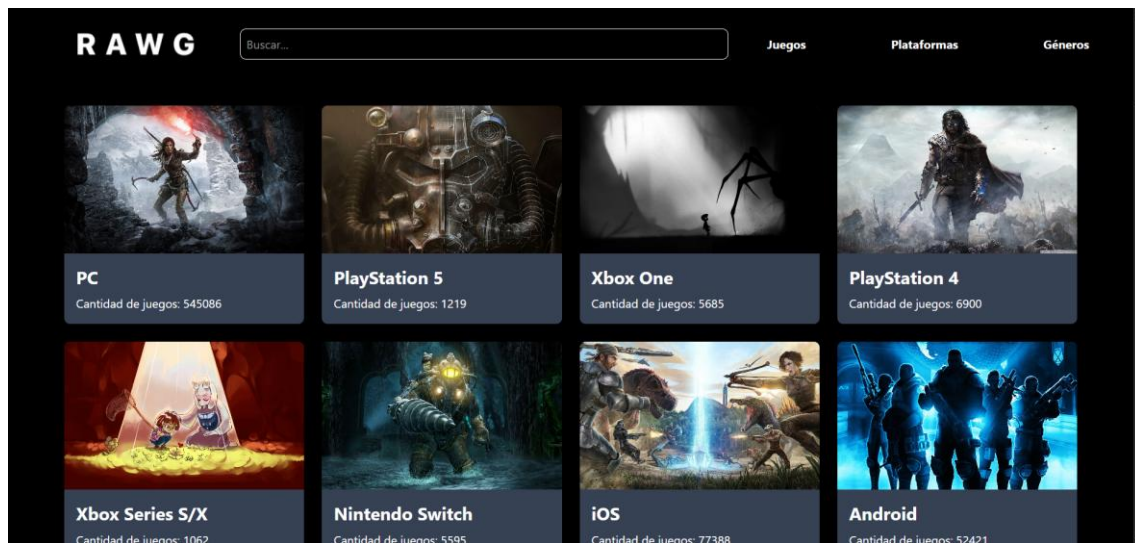
Al final de la página también encontraremos unos botones para navegar por las diferentes páginas de videojuegos generadas debido al paginado de 12 en 12.



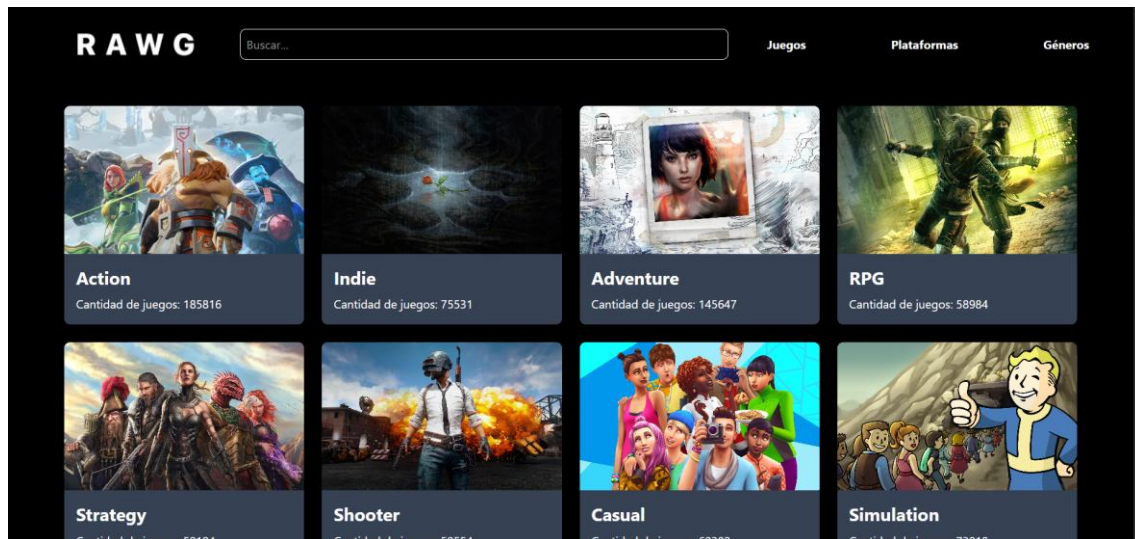
Además, si pulsamos dentro del **card** de un videojuego, nos llevará a una nueva página (**GameDetails**), donde podremos observar más información acerca del videojuego, como una breve descripción, géneros y plataformas listadas.

La aplicación también incluye:

- **Platforms:** Una segunda página que contiene un listado de plataformas.



- **Genres:** Una tercera página que contiene un listado de géneros.



- **Layout:** Una ruta especial que permite insertar el **header** y **footer** en todas las páginas de la web.

En todas las páginas se puede **filtrar por nombre**, gracias a un **SearchInput** que permite buscar videojuegos, plataformas y géneros.

### 3. Tecnologías utilizadas

Para la creación de nuestra aplicación, hemos utilizado:

- **React Router:** Para la gestión de rutas dentro de la aplicación.

- **TailwindCSS:** Para los estilos, debido a su flexibilidad y la utilidad de sus clases predefinidas.
  - **TypeScript:** En lugar de JavaScript, ya que facilita la codificación y elimina muchos errores en tiempo de ejecución gracias al tipado de variables.
  - **Vite:** Un gestor que nos permite instalar dependencias y ejecutar el proyecto de una forma sencilla y rápida.
- 

#### 4. Instrucciones de instalación

Ya que hemos utilizado **Vite** para la creación del proyecto, es necesario instalar **Node.js** para ejecutar correctamente la aplicación.

##### **Pasos para instalar y ejecutar la aplicación:**

Instalar Node.js y sus dependencias utilizando el gestor de paquetes **NPM**:

- `npm install`

Lanzar la aplicación en modo desarrollo:

- `npm run dev`

La aplicación web responderá en la dirección `http://localhost:5173`.

Opcionalmente, se puede crear una versión optimizada (build) de la aplicación:

- `npm run build`
- 

#### 5. Enlaces de interés

- Enlace a la página desplegada con Vercel: [Proyecto React RAWG](#)