Preentrega 3: versión estática

En este documento, se detalla el proceso de desarrollo de una página web de reseñas de películas utilizando React. Se describe la estructura de archivos creada, los componentes principales y los desafíos que tuvimos durante el desarrollo, especialmente en la integración de datos a través de APIs de terceros como TMDb y YouTube.

Distribución de Archivos

Se creó una estructura de archivos organizada para gestionar el proyecto de manera eficiente. Los archivos principales para este proyecto son:

- App.js: El componente principal de la aplicación.
- Header.js: Componente que contiene la barra de navegación superior.
- Main.js: Componente principal que aloja el contenido principal de la página.
- Footer.js: Componente que contiene el pie de página de la aplicación.
- MovieCarousel.js: Componente para mostrar un carrusel de películas destacadas.
- MovieList.js: Componente para mostrar la lista de películas.
- SearchBar.js: Componente para la barra de búsqueda de películas.
- moviesData.js: Archivo para almacenar datos de películas, incluyendo calificación, elenco, resumen, etc.
- movies.json: Archivo que podría contener datos de películas en formato JSON.
- setupTests.js: Archivo para configurar pruebas en la aplicación.
- index.js: Archivo de entrada principal de la aplicación.
- index.css, App.css, Menu.css, MovieList.css: Archivos de estilo para diferentes partes de la aplicación.
- reportWebVitals.js: Archivo para reportar métricas de rendimiento web.

Detalles adicionales sobre los componentes principales:

Header, Main y Footer: Estos componentes son muy importantes para la estructura básica de la página web. El encabezado contiene la navegación, el área principal aloja el contenido principal de la página, y el pie de página proporciona información adicional y enlaces útiles para el funcionamiento de la pagina web.

MovieList: Este componente es importante ya que es el que se encarga de mostrar la lista de películas. Enfrentamos la dificultad de integrar datos de manera eficiente desde la API de TMDb. Para mejorar la experiencia del usuario, se buscó automatizar la obtención de datos como calificación, elenco y resumen de cada película, minimizando la necesidad de búsquedas manuales. Sin embargo, tuvimos obstáculos, como los problemas de CORS, que ocasionalmente afectaron la conectividad con la API.

Integración de APIs Externas

Se utilizó la API de TMDb para obtener información detallada sobre películas, incluyendo calificación, elenco, resumen y otros detalles que le pueden interesar al usario. También se integró la API de YouTube para mostrar tráilers de películas de esta manera buscamos mejorar la experiencia del usuario. La información obtenida se organizó según la saga a la que pertenece cada película y su categoría para así tener la información de una manera clara y organizada.

Automatización de la Obtención de Datos

Se buscó automatizar la obtención de datos de películas para mejorar la comodidad del usuario final. Sin embargo, debido a la complejidad y los problemas de CORS, esta tarea nos resulto bastante complicada.

Problemas de CORS

Se experimentaron problemas intermitentes de CORS (Cross-Origin Resource Sharing) al conectarse con la API de TMDb. Esto afectó la disponibilidad de datos y requirió una gestión más complicada y detallada del problema con protocolos.

próximos pasos

se buscará mejorar la automatización de la obtención de datos y gestionar de manera más efectiva los problemas de CORS para garantizar una experiencia de usuario fluida.