Wiki app Guidelines

Este pequeño documento tiene como finalidad dejar una pequeña documentación acerca de todas las decisiones técnicas tomadas para el proyecto y dar una visión sencilla de todo lo que se implementó basados en los requerimientos necesarios.

Wiki App es una pequeña wiki fan page de la exitosa serie Rick & Morty, donde podemos consultar los datos de todos los personajes y episodios que participó cada uno, también esta la opción de encontrar de forma personalizada por nombre a cualquier personaje que exista en la serie, y así cualquier true fan podrá saber todas las versiones de Rick o Morty que han existido a lo largo de todas las temporadas.

La wiki es 100% multidevices (puedes verlo desde un computador, tablets y celulares) y esta disponible desde esta url https://wiki-app-lsalazar85.vercel.app/

Todas la información es consumida de una API REST gratuita de la serie
https://rickandmortyapi.com/

Desiciones técnicas Tech Stack

Para construir todo el proyecto, me base en todas las siguientes tecnologías:

- **Next.js:** Elegí este framework basado en react, configurable tanto en javascript y typescript, simplemente por el hecho de apuntar a un buen performance tomando en cuenta la cantidad de data manejada, ayudado por el SSR (Server Side Rendering) para dar una mejor experiencia fluida al usuario que consuma todo lo que ofrece la wiki.

- **Typescript:** Para dar mas solidez al proyecto, el main language configurado es Typescript, al ser fuertemente tipado, me da la garantía de trabajar con data más real y evitar malas interpretaciones de funcionalidades que trabajen con data diversa.

- **React Query:** Este package nos ofrece podemos manejar de forma sencilla poder almacenar en caché y actualizar datos sin la necesidad de configurar un estado global como Redux o Context. Me pareció muy buena idea implementarlo para el consumo de toda la api, me dio las opciones de crear hooks para obtener la data y poder consumirla desde cualquier componente y asi establecer la base para los filtros de ***Search by Name, Characters & Episodes***, evitar las llamadas innecesarias a la api y optimizar el perfomance ya que es una implementación sencilla y no caemos en la clásica s**obre ingeniería** para la obtención de datos y administración de los mismos.

- **Styled Components:** Todos los componentes están basados en styled components, inyectar estilos desde js siempre es una opción optima para trabajar de forma modulada.

- **Jest & RTL (React Testing Library):** Configurado para ts, para dar más soporte a todo el código trabajado bajo test unitarios de los principales componentes UI del proyecto.

- **Arquitectura:** Todo el proyecto esta basado es responsabilidades únicas bien identificadas, apelando al buen uso de la reutilizaron de componentes y funciones pensando en una escalabilidad modulada del proyecto. Cada componente tiene su capeta única con todos los archivos que necesita para su correcto funcionamiento *(Componente > Estilos > Test File > Mocks)*

- EsLint & Prettier: Cualquier desarrollador que se descargue este proyecto, tiene a su disposición la implementación de es Eslint para trabajar de forma estricta todo el código del proyecto y así evitar código con características extrañas. Y Prettier para darle un formato más genérico a todo el código de componentes y funciones.

- **Paginación**: Se implementó un infinity scroll para cada listado tanto de personajes y capítulos, esto ayuda plenamente al consumo racional de la api y mediante un hook de React Query (**useInfiniteQuery)** va consumiendo cada una de las paginas que contiene toda la información de la misma forma que el usuario vaya haciendo scroll y no hace un *re-fetch* si consultamos de nuevo la sección ya que toda esta almacenado en la caché que nos sirve de “estado global”.