



Prueba React Front-End Developer Grupo NEX

Se requiere la implementación del front-end de una API construida en GraphQL para la búsqueda y marcación de animes como favoritos. La página contará con los siguientes requerimientos:

- Tendrá los siguientes filtros para la búsqueda de los animes:

- Campo de búsqueda
- Género
- Año de emisión
- Estado de emisión
- Temporada de emisión

! Una vez seleccionado un filtro este valor se deberá conservar en la URL, esto para si se refresca la página o se comparte la URL sean persistentes los filtros ya seleccionados.

- Tanto para la carga inicial, como si no se encuentran filtros seleccionados se deben cargar los 6 primeros animes clasificados dentro de:

- Popular esta temporada
- Popular todos los tiempos

! Si se tienen filtros seleccionados, estas dos listas deben estar ocultas.

- Al dar click en un resultado este abrirá un modal donde se cargará la información específica de este anime con la siguiente información:

- Banner
- Título en inglés y original
- Descripción
- Cantidad de episodios
- Promedio de puntuación
- Estado de emisión
- Fecha de inicio
- Fecha de finalización, si ya finalizó
- Trailer si está disponible
- Botón de añadir a favoritos



- Se contará con una página de favoritos. Los animes marcados como favoritos por el usuario deberán persistir incluso luego de refrescar la página y podrán ser accedidos desde un botón flotante en la parte inferior derecha de la página.

Para dar solución a los requerimientos mencionados se entrega la siguiente documentación:

- Diseño: [Figma](#)
- GraphQL API: [Apollo Explorer API](#)
- Para los filtros:
 - Query para Género: GenreCollection
 - Para Estado de Emisión y Temporada de Emisión no existe query, pero si están registrados como enumeraciones:
 - Estado de Emisión enum: [Media Status](#)
 - Temporada de Emisión enum: [Media Season](#)
- Query para obtener la lista de animes: Page > media
 - Dentro de **Page** se encuentran los argumentos de paginación
 - page
 - perPage
 - Dentro de **media** se encuentran todos los argumentos y campos para la información retornada de los animes, tener en cuenta:
 - Este mismo query sirve tanto para obtener TODOS los animes como también para obtener un anime individual mediante el argumento **mediaId**
 - Este query también retorna información sobre manga, así que, para obtener solo resultados de anime, TODAS las peticiones a este query debe ir SIEMPRE con los siguientes argumentos por defecto:
 - type: ANIME
 - isAdult: false
 - Para los filtros tener en cuenta el uso de los argumentos: search, genreIn, status, seasonYear, season, y cualquier otro argumento que se considere necesario.
 - Para la carga de animes de esta temporada y de todos los tiempos hay que considerar que la season y seasonYear actual son WINTER y 2025, respectivamente. El ordenamiento de estos resultados debería ser por la popularidad de manera descendente, los valores del argumento sort se encuentra en [MediaSort](#). También considerar el uso de la paginación para obtener solo la cantidad de animes solicitados.

- Revisar los campos disponibles del query mediante Apollo explorer para identificar los campos solicitados para el modal.
- Considerar el uso de:
 - Indicadores de carga.
 - Estados de información no encontrada.
 - Manejo de errores.

STACK REQUERIDO ⚡

- React
- NextJs
- TypeScript
- TailwindCSS
- Recomendamos el uso de Redux para la persistencia de datos.

ENTREGABLES 📦

- Proyecto React en un repositorio de Github.
- Como mínimo una prueba unitaria.
- Readme con:
 - Descripción del proyecto.
 - Instrucciones claras de como se debe ejecutar el proyecto.
 - Descripción sobre las tecnologías utilizadas.

IR MÁS ALLÁ 🚀

Si crees que cubrir los entregables básicos no es suficiente para mostrar lo mejor de tus habilidades, no dudes en agregar algo más. Esto no es de ninguna manera obligatorio, pero si hay habilidades adicionales valiosas que te gustaría mostrar, hazlo. Estas son solo algunas ideas de cosas que podrías agregar:

- Expande tus pruebas unitarias, cualquier cosa que garantice que tu código sea irrompible.
- Despliega tu proyecto en la nube.
- Documenta tu código. Usa cualquier formato que conozcas para generar documentación.

⚠ El cielo es el límite, pero también lo es tu tiempo. Recuerda que cualquiera de estas tareas es completamente opcional.