



### Empaquetador

Vite



### Manejador de paquete

Yarn



### Testing

Motor: vitest



Librería: react testing library



### Manejos de rutas

React-router-dom



### Librería de obtención de datos

- React query  
(Manejar la queries / API, API Nativa FETCH,)



Ó

- RTK Query  
(Manejar la queries / API, API Nativa FETCH)



### Estados globales

(Zustand ó Redux Toolkit )



### Tipado

TypeScript



## La estructura propuesta de aplicación front-end en React:

**src/:** Carpeta principal del proyecto

**config/:** Archivos de configuración de middlewares (errors, loggers, translators, etc)

**feature/:** Componentes atómicos correspondientes al proyecto

[Por cada feature debemos contar con una carpeta con los siguientes archivos:]

**component.tsx:** Lógica de negocio y renderización propio del componente.

**component.scss:** Definición de estilos para el componente.

**test/:** Carpeta principal de pruebas unitarias para el componente.

**services/:** Capa de acceso a apis, aquí se incluirán configuración y archivos correspondientes al consumo de apis.

**routes/:** archivos de rutas del proyecto.

**hooks/:** archivos de hooks personalizados.

**store/:** configuración del store

- **slices/stores:** archivos de estado de componentes.
- **models/:** Definición de modelos y entidades a utilizar: user, policy, etc.
- **utils/:** Definición de componentes utilitarios: DateManager, etc

**assets/:** contenido importante para el proyecto

- **img/**
- **audio/**

**styles/:** css globales

**i18n/:** Archivos relacionados a textos de internacionalización.

Archivos de configuración: package.json, tsconfig.json, etc.

## Comparación de las librerías de obtención de datos

React Query	RTK Query
React Query es una librería que cuenta con 2 sencillos hooks que proporcionan la obtención, el almacenamiento en caché y la actualización de datos asíncronos en las aplicaciones React.	RTK Query es una poderosa herramienta de recuperación y almacenamiento en caché de datos. Está diseñado para simplificar los casos comunes de carga de datos en una aplicación web, eliminando la necesidad de escribir manualmente la lógica de obtención y almacenamiento en caché de datos
Permite configurar el stale, la caché y el tiempo de reintento creando un objeto <code>queryClientConfig</code> .	RTK Query es un complemento opcional incluido en el paquete Redux Toolkit , y su funcionalidad se basa en las otras API en Redux Toolkit.
Permite la actualización de los datos obsoletos o stale data en segundo plano, ya que react-query hace un preajuste y Optimizar las peticiones al backend.	RTK Query se inspira en otras herramientas que han sido pioneras en soluciones para la obtención de datos, como Apollo Client, React Query, Urql y SWR, pero agrega un enfoque único a su diseño de API
Con la función <code>refetchOnWindowFocus</code> , se puede volver a buscar en segundo plano cuando el usuario cambia la pestaña del navegador o cuando vuelve a la aplicación.	Debido a que Redux Toolkit es independiente de la interfaz de usuario, la funcionalidad de RTK Query se puede usar con cualquier capa de interfaz de usuario
Los dos hooks que nos proporciona la librería son <code>useQuery</code> y <code>useMutation</code> . <code>useQuery</code> se utiliza para obtener datos y <code>useMutation</code> se utiliza para crear, borrar y actualizar los datos en el servidor. El <code>useQuery</code> toma una clave, una función para llamar a la api y un objeto para configurar esta llamada. Y todas las consultas serán almacenadas en el <code>QueryCache</code>	RTK Query también puede generar enlaces React que encapsulan todo el proceso de obtención de datos, proporcionan campos <code>isLoading</code> los componentes y administran la vida útil de los datos almacenados en caché a medida que los componentes se montan y desmontan.
React query es una gran herramienta para utilizar en aplicaciones React para gestionar el estado del servidor, lazy loading, paginación y el mantenimiento de la caché.	RTK Query proporciona opciones de "ciclo de vida de entrada de caché" que permiten casos de uso como la transmisión de actualizaciones de caché a través de mensajes websocket después de obtener los datos iniciales
Con su soporte de devtools, está claro que se puede utilizar también en aplicaciones más complejas. Después de trabajar con los proyectos que incluyen	Finalmente, RTK Query está completamente escrito en TypeScript y está diseñado para brindar una excelente experiencia de uso de TS.

## Comparación de las herramientas para estado global

Zustand	Redux Toolkit
Zustand es una solución de gestión de estados pequeña, rápida y escalable. Su gestión de estado es centralizada y basada en acciones. Puedes usar Zustand tanto en React como en alguna otra tecnología como Angular, Vue JS o incluso en JavaScript vanilla.	Redux Toolkit tiene como objetivo simplificar Redux con una mejor abstracción y mas conveniente sobre el "texto original" del que se quejaron tantos desarrolladores. Redux esta bien, pero era un poco complicado
Hay menos código repetido (comparado con Redux).	Elimina la necesidad de agregar múltiples paquetes para tener una aplicación escalable.
Su documentación fácil de entender.	Reduce el código repetitivo.
Brinda más flexibilidad	Simplifica la configuración de redux
Puedes usar Zustand de la forma simple, con TypeScript, puedes integrar immer para la inmutabilidad o incluso puedes escribir un código parecido al patrón Redux (reducers y dispatch).	los bloques de construcción fundamentales de Redux siguen siendo la acción, los reducers (reductores), el middleware y el store (tienda), y necesita un buen conocimiento de estos bloques para dominar Redux y el kit de herramientas de Redux.
No envuelve la aplicación en un proveedor como normalmente se hace en Redux. Vuelve a renderizar los componentes solo cuando hay cambios.	Nos aporta una estructura de datos consistente, ya que la crearemos en base a un modelo que cumplirá una serie de requisitos, lo que lo hará sólido.

### **¿React Query ó RTK Query?**

En pocas palabras con React Query, puede acceder y modificar fácilmente los datos que necesita, mientras que RTK Query ofrece capacidades de recuperación y almacenamiento en caché de datos basadas en la biblioteca de administración de estado de Redux. Tiene muchas características similares a React Query pero es compatible con otros componentes del kit de herramientas de Redux, como creación de middleware o reductor.

### **¿Zustand ó Redux Toolkit?**

En el caso de Zustand sin necesidad de usar proveedores como en Redux, podemos usar nuestra tienda casi en cualquier lugar ("casi" ya que sigue las reglas de los ganchos, ya que la tienda básicamente es un gancho por debajo). Sin antes mencionar que su configuración y uso no lleva ciencia, y es fácil de implementar a comparación de Redux Toolkit, además que provee la herramienta persist, la cual permite tener guardados los datos del estado en el localstorage sin necesidad de grandes configuraciones, en pocas palabras, menos código, más proporcionalidad y mismo resultado pero con más flexibilidad.