Módulo 4 – Hooks y Life Cycle

# ¿Cómo crear Hooks?

Slide 2

Como muchas veces en programación, para crear algo se parte de bloque fundamentales, entonces para crear un Hook, lo hacemos a taves de otros Hooks! La idea o motivación de crear nuestors Hooks es generalmente para encapsuar código que debe mantener un estado a traves de los renders.

Por ejemplo, podemos necesitar un de Hook de paginación de datos. Con este hooks podemos enviarle todo una Array de datos, un número N que es la cantidad de items por página. Y este nos devuelve una página con N items, una función para ir a la página siguiente, atrás, principio y final.

Input debounced, a veces uno tiene un buscador, pero no desea realizar la busqueda y fetch en cada carácter que cambie. Entonces podemos crear un hook que actualice el valor final solo cuando haya pasado cierto tiempo luego del ultimo cambio realizado. Otro muy usado es un Hook para adminitrar listas, con este recibimos funciones para agregar elementos, en ciertos indices, eliminar o limpiar el state.

Otros ejemplo que he usado es “useParams” lo uso principalmente para poder invalidar fetch realizados con una librería llamada react-query. Esta librería recibe un Key y función para hacer el fetch. Muchas veces algun cambio de un fetch debe hacer que otro actualice los datos (como los forms). Entonces cone ste hooks yo adminitro el key y si este cambia marco invalidado el hook.

Mostrar ejemplos en mis repos y de librería Mantine para mas ejemplos de como usarlos

# Convenciones para Hooks

Slide 3 - 4

Las principales conversiones es que el nombre del hook debe iniciar con el prefijo “use” El resto como la firma de la función puede ser a conveniencia del usuario. Ya sea parámetros pocionales o un objecto. El return generalmente es un arreglo y en Typescript debe ser delcarado como “as const” para no tener problemas en los tipos. También se puede retornar un objecto para destructuring por nombre y no por posición como hemos visto.

# Creando Hook de paginación

1. Crear la carpeta “hooks”
2. Crear el hook de paginación con parametro en JSON, y devuelve el arreglo con items por página, función para ir adelante, atrás, total de paginas.
3. Crear el type de los parámetros.
4. Crear la funciones necesarias como “function” y explicar por que, por convencion y mejor visual al leer.

*Cotinuar con teoria y practica de life cycle components*

*La práctica/tarea será hacer un useDebouce que recibe un valor string inicial y un número en milisegundos y devuelve como valor otro string debounced. Usar este hooks para agregar un buscardor en la vista “/users/view” para buscar usuarios según su nombre.*