Hooks

Módulo 2 - Unidad 1

Lista de tareas con *hooks*

Objetivos

Practicar:

- useState y useEffect para gestionar estado y efectos secundarios.
- useRef para manejar elementos del DOM.
- useCallback o useMemo para optimizar cálculos.
- La creación de un ***custom hook*** para persistir datos en localStorage.

Consigna

1. **Componente principal (App.jsx)**
    - Mostrar una lista de tareas.
    - Campo de texto (input) y botón para agregar nuevas tareas.

- Botón en cada tarea para marcarla como completada o eliminarla.
- *Input* de búsqueda que filtre tareas por título.

2. **Estado y efectos**
   - Usar useState para incluir:
     - Lista de tareas.
     - Texto del *input*.
     - Texto de búsqueda.
   - Usar useEffect para guardar las tareas en localStorage cuando cambien.
   - Usar useRef para enfocar automáticamente el campo de texto al montar el componente.
3. **Optimización**
   - Usar useMemo para que el filtrado de tareas se recalule solo cuando cambie el texto de búsqueda o la lista de tareas.
4. ***Custom hook***
   - Crear el *hook* useLocalStorage(key, initialValue) que:
     - Obtenga un valor inicial desde localStorage.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

        - Guarde automáticamente el nuevo valor cuando cambie.

    - Usar este *hook* en lugar de useState para la lista de tareas.

5. **Extras opcionales**

    - Mostrar cuántas tareas están completadas y cuántas pendientes.

    - Usar useCallback para manejar funciones pasadas a componentes hijos si decides dividir la UI.

Formato de presentación

La entrega debe realizarse en un **repositorio de GitHub**, que incluya:

- Proyecto en **React (Vite recomendado)**, limpio y organizado.

- **App.jsx** con: lista de tareas, input controlado para agregar, botones de completar/eliminar y filtro por búsqueda.

- Uso de ***hooks***:

    - useState, useEffect, useRef.

    - useMemo y, opcionalmente, useCallback.

- ***Custom hook*** useLocalStorage(key, initialValue).

- **README.md** con:

    - Descripción del proyecto.

- Instrucciones de instalación y ejecución.
- Capturas de pantalla.
- Créditos del autor.
- Citación de fuentes.

Criterios de evaluación

- Implementación correcta de **useState**, **useEffect** y **useRef**.
- Persistencia de datos con **localStorage** mediante un *custom hook*.
- Uso de **useMemo** para optimizar el filtrado de tareas.
- (Opcional) Uso de **useCallback** para optimización en funciones.
- Interfaz funcional: agregar, eliminar, completar y filtrar tareas.
- Organización clara del proyecto y limpieza del código.
- Repositorio prolijo con README completo y capturas.

Bibliografía utilizada y sugerida

Libros y otros manuscritos

Abramov, D. y Clark A. *Fullstack React*. 1ª ed. Accomazzo LLC; 2017.

Banks, A. y Porcello, E. *Learning React: Modern Patterns for Developing React Apps*. 2ª ed. O'Reilly Media; 2020.

Artículos y documentación en línea

React. (s.f.-a). *useState*. https://react.dev/reference/react/useState

React. (s.f.-b). *useEffect*. https://react.dev/reference/react/useEffect

React. (s.f.-c). *useRef*. https://react.dev/reference/react/useRef

React. (s.f.-d). *useMemo*. https://react.dev/reference/react/useMemo

React. (s.f.-e). *Reusing Logic with Custom Hooks*. https://react.dev/learn/reusing-logic-with-custom-hooks