

Trabajo Práctico grupal de presentación obligatoria.

En esta parte se debe implementar el uso del hook **useEffect** para controlar los efectos secundarios en la aplicación, sincronizando los cambios en el estado de la lista de proyectos con un componente de registro de actividad en tiempo real.

Tareas a Desarrollar

1. Crear el Componente de Registro (<RegistroActividad />)

Diseñar un componente de presentación (UI) encargado de mostrar de forma clara al usuario cuándo ocurrió la última modificación en el arreglo de proyectos.

- **Requisito:** Este componente debe recibir por **props** un dato (String o Date) que represente la fecha y hora exacta.
- **Uso:** Se debe renderizar e integrarse visualmente al final o en la parte inferior de la vista ListaProyectos.jsx.

2. Implementación de useEffect en ListaProyectos.jsx

El foco de esta semana está en controlar con precisión **cuándo** se dispara un efecto secundario.

- **Consigna:** Configurar un **useEffect** en el componente principal que escuche atentamente los cambios en el estado **proyectos**.
- **Arreglo de Dependencias:** El hook debe configurarse de tal manera que **solo** se ejecute cuando la lista de proyectos sufra una modificación (es decir, cuando se agregue con éxito un nuevo proyecto mediante el formulario, o cuando se elimine un proyecto de la lista).
- **Lógica a ejecutar:** Cada vez que el efecto se dispare, deberá capturar la fecha y hora actual del sistema mediante el objeto Date de JavaScript y actualizar un nuevo estado local destinado a almacenar este registro.

3. Formato del Mensaje

El texto que se visualice en la pantalla a través del componente de registro debe respetar de manera estricta el siguiente formato dinámico:

"Última actualización de la lista: DD/MM/AAAA a las HH:MM hs."

(Donde DD/MM/AAAA corresponde al día, mes y año actual, y HH:MM a la hora y minutos exactos del evento).

4. Agregando complejidad.

Una vez que hayan logrado que el componente `<RegistroActividad />` muestre la fecha y hora utilizando `useEffect`, notarán dos comportamientos característicos del ciclo de vida de React que debemos corregir para profesionalizar nuestra aplicación:

1. **Renderizado Inicial:** El mensaje de actualización aparece apenas se carga o monta la página por primera vez.
2. **Efectos Secundarios por Filtro:** Al escribir en la barra de búsqueda, el componente se re-renderiza y vuelve a capturar la hora actual, aunque no hayamos agregado ni eliminado ningún proyecto real.

Deberán refactorizar el código del componente `ListaProyectos.jsx` para cumplir de manera estricta con los siguientes dos requisitos de optimización:

- **Evitar la carga inicial:** El componente `<RegistroActividad />` **no debe mostrarse** cuando la página se carga por primera vez. Solo debe aparecer a partir del momento en que el usuario realice la primera acción de alta (agregar) o baja (eliminar).
 - Investiguen el uso del hook **useRef** para crear una bandera o "flag" que les permita saltar la primera ejecución automática del **useEffect**.
- **Aislamiento del Filtro de Búsqueda:** Escribir en el campo de texto de búsqueda **no debe** alterar la fecha y hora de la última actualización. El efecto secundario debe quedar completamente aislado. Implementar una estrategia para evitar que se (actualice/muestre) el componente `RegistroActividad` cuando se utilice el filtro.

5. Componentización del Formulario (`<FormularioProyecto />`)

Para finalizar, deberán extraer el formulario de alta de proyectos que se encuentra actualmente en `ListaProyectos.jsx` y transformarlo en un componente independiente llamado `FormularioProyecto.jsx`.

- **Estado Local:** El estado del objeto formulario y su función `handleChange` deben pasar a ser gestionados internamente por el nuevo componente hijo.
- **Elevación del Estado:** Al realizar el envío del formulario (`submit`), el componente hijo deberá pasarle el objeto con los datos del nuevo proyecto al componente padre mediante una **prop de tipo función** (callback). El padre recibirá el objeto y se encargará de pasarlo al servicio y actualizar la lista general para disparar el registro de actividad.