

# Bootcamp de Full Stack

Bienvenidos a la clase N°40

- Hooks
  - useRef
  - useEffect
  - Troubleshooting
- Desafío Semanal F2-N13

# Hooks

## useRef

El hook **useRef** permite crear una **referencia mutable** que persiste durante todo el ciclo de vida del componente. No causa re-renderizados cuando su valor cambia.

Devuelve un objeto con una propiedad **.current** que puede contener cualquier valor (nodo del DOM, valor numérico, booleano, etc.).

Usos comunes:

- Referenciar elementos del DOM directamente (como inputs o contenedores).
- Guardar valores persistentes que no deben generar re-render.
- Almacenar valores intermedios (como un temporizador o una posición previa).

Sintaxis

```
const hasChangeRef = useRef(true);
```



# Hooks

## useEffect

El hook **useEffect** permite ejecutar efectos secundarios en componentes funcionales, como llamadas a APIs, suscripciones, timers, o manipulación del DOM. Se ejecuta después de que el componente se haya renderizado.

Usos comunes:

- Ejecutar lógica después del render (fetch, listeners, timers).
- Sincronizar el componente con una fuente externa de datos.
- Limpiar recursos cuando el componente se desmonta.

Reglas

- Siempre declarar la lista de dependencias (array) para controlar ejecución.
- Si se pasa un array vacío, se ejecuta solo una vez al montar.

Sintaxis

```
useEffect(() => { bloque de código }, [ dependency ]);
```



# Hooks

## Troubleshooting

### Problemas frecuentes en React

- Sincronización con useState
  - Acceso inmediato al estado: Leer el estado justo después de actualizarlo no funciona como se espera, ya que setState es asíncronico y el nuevo valor estará disponible en el siguiente render.
  - Actualización múltiple: Llamar varias veces a setState sin usar el callback de actualización puede hacer que solo se aplique la última actualización.
- Dependencias en useEffect:
  - Modificar una dependencia dentro del useEffect: Cambiar internamente una variable que está en el array de dependencias puede generar bucles infinitos.
  - Usar useRef como dependencia de useEffect: Incluir ref.current como dependencia no tiene efecto, ya que los cambios en useRef no disparan renders ni efectos.

# BREAK

Descansemos 10 minutos



# Actividad Práctica

## Resolvemos Entre Todos

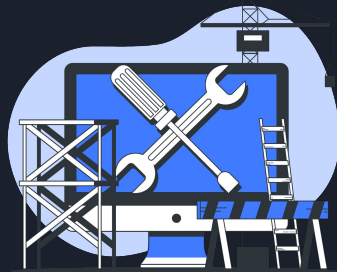
Para llevar a cabo esta actividad, accede al documento de la consigna a través del siguiente enlace:

[https://docs.google.com/document/d/1SWJ33bAyrGMBK4B2-VHnly\\_aIRf63IMDCrnTkJ3TjxJU/edit?usp=sharing](https://docs.google.com/document/d/1SWJ33bAyrGMBK4B2-VHnly_aIRf63IMDCrnTkJ3TjxJU/edit?usp=sharing)



# Desafío Semanal

F2-N13



- **Aclaración:** Esta actividad es de **carácter opcional** y se lleva a cabo de forma **individual** o en **parejas de estudiantes**, fuera del horario de clase.
- **Evaluación:** La realiza el **tutor**.
- **Fecha límite:** La **entrega** se debe realizar antes de iniciar la **siguiente clase**.
- **Consigna y recursos:** El siguiente enlace contiene todo lo necesario para el desarrollo de la actividad:  
[https://drive.google.com/drive/folders/1p-EXNV1SwOTaLgLMK4pMGE3szKF1iJvF?usp=drive\\_link](https://drive.google.com/drive/folders/1p-EXNV1SwOTaLgLMK4pMGE3szKF1iJvF?usp=drive_link)
- **Entrega:** Se hace de forma individual en formato ZIP y se sube a la carpeta personal de Google Drive.
- **Registro de entrega:** Se debe registrar en este formulario: <https://forms.gle/VeRUKYcfjCtv4afXA>

# CIERRE DE CLASE

Continuaremos en la próxima clase

