

Bootcamp de Full Stack F2

Bienvenidos a la clase N°44

- Hooks
- useState
-
- Modularización
- Modularización con Sass & grid
- Hooks
- useState
- useEffect
- useContext

React

Hooks

Hook useEffect

Este hook nos permite agregar efectos secundarios a un componente funcional dado, es decir, permite realizar modificaciones en la lógica una vez se haya realizado el renderizado. El mismo, recibe dos parámetros, el primero es un callback y el segundo es un array de dependencias. Usar un array vacío de dependencias, significa que el efecto se ejecutará sólo una vez (al montar el componente).

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

Una de las grandes ventajas de este hook es que simplifica el ciclo de vida, de forma que los tres métodos de clase (componentDidMount, componentDidUpdate y componentWillUnmount) pueden ser expresados utilizando únicamente este hook.

JavaScript

POO - Clases - Herencia

- **POO:** Programación orientada a objetos. este es un paradigma de programación que parte del concepto de "objetos" como base, los cuales contienen información en forma de propiedades y código en forma de métodos. El propósito de su creación fue acercar el manejo de las estructuras de un programa al manejo de las cosas en el mundo real, de ahí el nombre "objeto" como algo genérico, que puede representar cualquier cosa tangible. Las principales características de la POO es el encapsulamiento, herencia y polimorfismo.
- Una **clase** es un elemento de la POO que actúa como una plantilla o molde y va a definir las características y comportamientos de una entidad.
- La **herencia** es un mecanismo basado en clases, por medio del cual una clase se deriva de otra de manera que extiende su funcionalidad.

BREAK

Descansemos 15 minutos



JavaScript

LocalStorage

La propiedad **localStorage** es un objeto de almacenamiento en el lado del cliente que permite a una aplicación web almacenar datos de manera local.

Características:

- Los datos se almacenan en pares clave-valor (solo permite valores de tipo string).
- Los datos perduran cuando se cierra la pestaña o ventana.
- Se permite el acceso a los datos entre las diferentes pestañas o ventanas del navegador.
- No tiene fecha de expiración.

Comandos:

- Agregar o modificar ítem: `localStorage.setItem("marca", "Ford");`
- Obtener ítem: `localStorage.getItem("marca");`
- Quitar ítem: `localStorage.removeItem("marca");`
- Quitar todos los ítems: `localStorage.clear();`

CIERRE DE CLASE

Continuaremos en la próxima clase

