

Métodos del Ciclo de Vida en React

Los métodos del ciclo de vida en React son funciones que se llaman automáticamente en diferentes etapas del ciclo de vida de un componente. Estas etapas son:

1. **Montaje (Mounting):** Estas etapas ocurren cuando se crea e inserta el componente en el DOM.
 - **Constructor:** Se ejecuta al crear una instancia del componente. Se utiliza para inicializar el estado y enlazar métodos.
 - **Render:** Retorna los elementos JSX para renderizar el componente en el DOM.
 - **ComponentDidMount:** Se ejecuta después de que el componente se haya montado en el DOM. Se utiliza para hacer solicitudes de datos o interactuar con APIs externas.

Métodos del Ciclo de Vida en React

2. **Actualización (Updating):** Estas etapas ocurren cuando el componente está siendo actualizado debido a cambios en el estado o en las props.
 - **Render:** Se llama nuevamente para actualizar la interfaz de usuario.
 - **ComponentDidUpdate:** Se ejecuta después de que se haya realizado la actualización del componente y se haya renderizado en el DOM. Se utiliza para realizar acciones adicionales después de una actualización, como solicitar nuevos datos.
3. **Desmontaje (Unmounting):** Esta etapa ocurre cuando el componente se elimina del DOM.
 - **ComponentWillUnmount:** Se ejecuta justo antes de que el componente sea eliminado del DOM. Se utiliza para limpiar los recursos o cancelar suscripciones.

Estos son algunos de los métodos del ciclo de vida más utilizados en React. Cada uno de ellos te permite manejar el comportamiento de tu componente en diferentes momentos. A partir de la versión 16.3 de React, algunos están en desuso y se recomienda usar en su lugar componentes funcionales con Hooks, pero en esta clase los veremos para que sepas cómo usarlos en caso de encarar algún proyecto en el cuál se haga uso de ellos.

Métodos del Ciclo de Vida en React

Para poder trabajar con los métodos del ciclo de vida, tenemos que hacer uso de los componentes basados en clase, por lo cuál, en nuestro index, comentaremos el componente "ContadorFuncional" y activaremos el componente "ContadorClass".

```
16 <Aula/>
17 <ContadorClass incrementar={1} disminuir={1}/>
18 {/*<ContadorFuncional incrementar={1} disminuir={1}/>*/}
```

Ahora dentro del "ContadorClass" vamos a ver un ejemplo del uso de los métodos del ciclo de vida, comenzando con el **ComponentDidMount** que permitirá ejecutar código cuándo el componente sea montado en la app colocando el siguiente código:

```
7   componentDidMount(){
8     console.log('El componente se cargó en el DOM');
9   }
```

Seguidamente para verlo en acción en el navegador cerramos sesión, colocamos en el formulario de inicio el usuario y clave "clase", limpiamos la consola e iniciamos sesión. De esta forma en consola saldrá el mensaje indicado en el código anteriormente:

Errores Advertencias Registros Información Depurar

El componente se cargó en el DOM

Métodos del Ciclo de Vida en React

En secuencia a lo anterior, trabajaremos con **ComponentDidUpdate** que permitirá ejecutar código cuándo el componente haya vuelto a renderizarse gracias a una actualización que se ejecuta porque su estado cambia. En este método podemos conocer qué propiedades y qué estado tenía antes del cambio, colocando dos (2) variables dentro de los paréntesis del método al momento de su creación:

```
10  componentDidUpdate(PropsAnteriores, EstadoAnterior){  
11      console.log("El componente ha sido actualizado");  
12      console.log("Las propiedades anteriores del componente son:", PropsAnteriores);  
13      console.log("El estado anterior del componente es: ", EstadoAnterior);  
14  }
```

Una vez hecho lo anterior, cada vez que demos un clic en el botón de aumentar o disminuir, se ejecutará este método del ciclo de vida y se imprimirán en la consola el mensaje, las propiedades y el estado anterior del componente:

```
El componente ha sido actualizado  
Las propiedades anteriores del componente son: ▶ Object { incrementar: 1, disminuir: 1 }  
El estado anterior del componente es: ▶ Object { contador: 0 }  
  
El componente ha sido actualizado  
Las propiedades anteriores del componente son: ▶ Object { incrementar: 1, disminuir: 1 }  
El estado anterior del componente es: ▶ Object { contador: 1 }
```

Métodos del Ciclo de Vida en React

Sucesivamente vamos a ver cómo trabaja **ComponentWillUnmount** que permitirá ejecutar código antes de quitar el componente del DOM:

```
15   componentWillUnmount(){  
16     |   console.log("El componente se dejará de mostrar en el DOM");  
17   }
```

Seguidamente para verlo en acción en el navegador cerramos sesión. De esta forma en consola saldrá el mensaje indicado anteriormente en el código:

