educación IT
CODE YOUR FUTURE

ReactJS Developer

Módulo 3

Estado

Estado de un componente

Se define como **estado al conjunto de variables reactivas encapsuladas dentro del componente.** En otras palabras: el estado son **variables que controlan el cambio de la interfaz** pero solo **en el componente en el que se encuentran** (y sus componentes hijos).

El estado permite implementar reactividad **focalizada** ¿Qué significa esto? Esto significa que me sirve para **implementar cambios en la interfaz que afecten solo a un elemento concreto o a sus hijos.**

Si quiero mostrar un submenú al hacer clic en un ítem, eso debería controlarse mediante el estado porque es un cambio focalizado. Si, en cambio, quiero cambiar todo el tema del sitio a modo oscuro, eso debería gestionarlo mediante props ya que es un cambio compartido por muchos componentes.

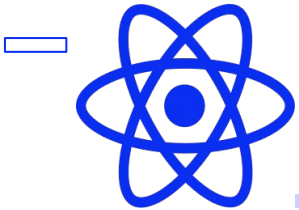
Comparación entre estado y propiedades

Estado	Propiedades
Gestiona cambios **puntuales** a un componente o sus hijos.	Gestiona cambios **globales** a varias partes de la interfaz.
Se crea **dentro del componente**.	Se pasa **por fuera del componente**.
No se comparte directamente con otros elementos.	Pueden contener **datos compartidos con otros elementos**.

Estado en componentes funcionales

- Los componentes funcionales **no tienen acceso al estado directamente**.
- Para acceder al estado desde un componente funcional se usan ***hooks***.
- Los *hooks* son una forma de **desacoplar características en React**. A diferencia de las clases, que tienen en **`React.Component`** todo lo necesario, los componentes funcionales **deben requerir las cosas que van necesitando mediante *hooks***.
- Para usar el estado vamos a usar el *hook* **useState**, importado desde **React**.
- `UseState` es una función que **recibe un parámetro** (el valor inicial de esa variable reactiva) y **devuelve un `array`**.
- El primer ítem de ese `array` es **una variable** y el segundo es **una función**. Cuando se llama a la función y se le pasa un valor, ésta **cambia el valor de la variable** y a su vez **cambia la porción de la interfaz en la que se use dicha variable**.

1. Importar la función **`useState`** de React.
2. Llamar a la función `useState`. Esta función retorna un `array` con dos elementos: el primer elemento es una variable que permite **leer** ese estado y el segundo es una función que permite **cambiar su valor**.
3. Recordemos que, al cambiar su valor, ocasiona que ese componente y sus hijos **se vuelvan a procesar**.
4. Una vez que tenemos la capacidad de leer y escribir esa porción del estado, podemos utilizarla en nuestra interfaz.



```jsx
import { useState } from "react";

function App() {
    const [nombre, setNombre] = useState("");

    return (
        <div>
            <input type="text" onInput={(e) => setNombre(e.target.value)} />
            <span>{nombre}</span>
        </div>
    );
}

export default App;
```

Estado en componentes basados en clases

- Los componentes basados en clases **tienen acceso al estado directamente.**
- Para acceder al estado se utiliza la propiedad **`this.state.`**
- Para cambiar el estado se hace únicamente a través de la función **`this.setState(nuevoEstado)`**.
- Tenemos disponibles la variable **`this.state`** y la función **`this.setState`** a lo largo de toda la clase y de todo su ciclo de vida.

1. En el constructor, asignar un valor a **`this.state`** luego de la llamada a **`super()`**. Este será **el valor inicial del estado.**
2. Llama a la función **`this.setState`** pasándole un nuevo valor para **`this.state`.**
   Si el valor previo y el nuevo son objetos, el comportamiento es el de **`Object.assing()`.**
3. Puedes leer el estado en cualquier lugar de la clase a través de la propiedad **`this.state`.**
4. Una vez que tenemos la capacidad de leer y escribir esa porción del estado, podemos utilizarla en nuestra interfaz.

```
class App {
    constructor(props) {
        super(props);

        this.state = {
            nombre: "",
        };
    }

    render() {
        return (
            <div>
                <input
                    type="text"
                    onInput={(e) => this.setState({ nombre: e.target.value })}
                />
                <span>{this.state.nombre}</span>
            </div>
        );
    }
}

export default App;

```

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!