



Curso de React.js: Manejo Profesional del Estado

¡No te rindas!

Necesitas una **calificación mínima de 9.0** para aprobar.

Vuelve a intentarlo en 05 horas, 58 minutos, 28 segundos

7

Calificación

7 / 10

Aciertos

1. ¿Cuál de las afirmaciones es INCORRECTA sobre los estados derivados?

Se crean con React.useState.



2. ¿Cuál es la técnica más popular para crear un reducer?

Switch



3. ¿Qué parámetros recibe un reducer?

state, actionType y actionPayload

[REPASAR CLASE](#)

4. Ningún paradigma es mejor que otro. Debemos analizar sus beneficios y costos para determinar si vale la pena implementarlos.

Verdadero



5. ¿Cuál de los siguientes métodos NO PERTENECE al ciclo de vida de los componentes creados con `React.Component`?

`componentDidUnmount`



6. ¿Qué requisitos / limitaciones tenemos para manejar el estado con reducers?

Debemos usar acciones y estados simples o independientes.

[REPASAR CLASE](#)

7. ¿Cuál es la diferencia principal al ACTUALIZAR estados compuestos en `React.Component` (con `this.setState`) vs. `React Hooks` (con `useState`)?

Con `this.setState` actualizamos propiedades del objeto `this.state` sin afectar o borrar a las demás. Con `useState` siempre debemos copiar el estado "anterior" antes de actualizar propiedades del objeto `state`.



8. Si creamos un estado patito con `React.useState`, a su actualizador por convención lo llamaremos `setPatito`. ¿Cómo se llamaría el actualizador de un estado patito si lo creamos con `React.useReducer`?

`setPatito`

[REPASAR CLASE](#)

9. ¿Para qué sirven los action creators?

Para separar el código de interfaz del código de manejo del estado.



10. ¿Cuándo debemos hacer migraciones de useState a useReducer?

Solo cuando mejore la legibilidad, agilice el mantenimiento y no duela (tanto) hacer el cambio. ✓

REGRESAR