



Curso Profesional de React Hooks

¡No te rindas!

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

Vuelve a intentarlo en 05 horas, 52 minutos, 21 segundos

6.96

Calificación

16 / 23

Aciertos

1. ¿Cuál de los siguientes React Hooks es un impostor (no es oficial de React.js)?

useLifecycle



2. ¿El custom hook getHomeData cumple con las reglas y convenciones de los React Hooks?

Verdadero

[REPASAR CLASE](#)

3. ¿El custom hook useHomeData cumple con las reglas y convenciones de los React Hooks?

Verdadero



4. ¿Cuáles reglas debes seguir para crear tu propio custom hook?

Empezar el nombre de todos nuestros custom hooks con use.



5. ¿A partir de cuál versión de React.js podemos crear componentes con React Hooks?

A partir de la versión 16.8.



6. ¿Es posible crear componentes con hooks y componentes con clases en un mismo proyecto?

Verdadero



7. useState nos permite:

Agregar estado a componentes creados como funciones.



8. Un ejemplo correcto de cómo usar useState es:

```
const [state, setState] = React.useState(initialState);
```



9. useEffect nos permite:

Ejecutar acciones como respuesta a un nuevo llamado o render de nuestro componente. Es muy parecido a `componentDidMount` y `componentDidUpdate`. ✓

10. ¿En cuál de los siguientes ejemplos nos aseguramos de que nuestro efecto se ejecuta solo una vez (cuando "montamos" el componente)?

`useEffect(() => { /* ... */ }, []);` ✓

11. ¿Cuál es la diferencia entre `useEffect` y `useLayoutEffect`?

`useEffect` se ejecuta antes de que el componente renderizado se "pinte" en pantalla.
`useLayoutEffect` se ejecuta después del "pintado". ✗

12. ¿Qué es memoization?

Una técnica de optimización para evitar que realicemos los mismos cálculos una y otra vez. ✓

13. ¿Cuál es el custom hook de Redux que nos permite elegir qué parte de nuestro estado leeremos en nuestro componente?

`useDispatch` ✗

14. ¿Cuál es el custom hook de React Router que funciona igual que el componente `Route`?

`useRouter` ✗

15. ¿Cuál es el custom hook de React Router nos permite modificar o "empujar" la navegación de nuestra aplicación?

useHistory



16. ¿Cuál React Hook es más eficiente para "escuchar" los cambios o efectos de nuestro componente cuando también utilizamos useRef?

useDomEffect



17. useContext nos permite:

Leer y modificar los mismos datos desde componentes en cualquier parte de la aplicación sin necesidad de props. ✓

18. useReducer nos permite:

Agregar estado y modificarlo con reducers en componentes creados como funciones. ✓

19. useMemo nos permite:

Evitar que nuestro componente haga render innecesariamente.

[REPASAR CLASE](#)

20. useCallback nos permite:

Ejecutar una función cuando el componente cumple ciertas condiciones y obtener su callback memoizado cuando no se cumplen. ✓

21. ¿Qué optimización podemos hacer con React.memo?

Evitar que nuestro componente haga render innecesariamente.



22. ¿Cuál de las siguientes herramientas nos ayudan a manejar meta-etiquetas para trabajar el SEO de nuestra aplicación?

React SEO

[REPASAR CLASE](#)

23. useRef nos permite:

Leer y modificar el valor de nuestros inputs directamente del DOM.



[REGRESAR](#)