



## Curso Profesional de React.js y Redux

### ¡No te rindas!

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

Vuelve a intentarlo en 05 horas, 56 minutos, 20 segundos

# 6.11

Calificación

# 11 / 18

Aciertos

1. ¿Cuáles son los principales aspectos que necesitamos de Redux en relación al estado?

Dónde almacenarlo, un método para acceder a él y una manera de modificarlo.



2. ¿Cuáles son los 3 principios de Redux?

Los actions deben ser funciones puras, el estado es de solo lectura, el reducer d [REPASAR CLASE](#) estado inicial.

3. El principio de única fuente de verdad se refiere a que:

Podemos tener varios estados globales pero solo acceder desde un sólo store. [REPASAR CLASE](#)

---

4. El estado global de Redux:

Puede ser de cualquier tipo aunque se recomienda un objeto clave-valor.



---

5. Es una característica de una función pura:

Para una entrada siempre va haber una misma salida.



---

6. Redux solo puede ser implementado con React.js.

Falso



---

7. Desde el componente se puede modificar directamente el estado global.

Falso



---

8. Siempre es mejor usar Redux vs. React Context

Falso, depende de aspectos como el bundle size, la facilidad de depuración y curva de aprendizaje.



---

9. Redux permite prevenir renderizados innecesarios.

Verdadero



10. Cuando usamos connect, al momento de testear nuestro componente:

Debemos conectar nuestro componente a Redux en las pruebas.

[REPASAR CLASE](#)

11. Cuando usamos useSelector, al momento de testear nuestro componente:

Podemos pasar los valores como props, ya que nuestro componente "no sabe" Redux. [REPASAR CLASE](#)

12. Redux recomienda dejar de usar connect y usar en cambio useSelector.

Falso

[REPASAR CLASE](#)

13. ¿Qué es un Middleware con relación a Redux?

Es el objeto que describe el cambio que se va a hacer al estado.

[REPASAR CLASE](#)

14. Redux Thunk tiene mucho menos boilerplate que Redux Saga.

Verdadero



15. ¿Por qué es importante la inmutabilidad en Redux?

Porque Redux no indicará a la UI que debe renderizar nuevamente si su estado inicial y el estado retornado son exactamente iguales.



16. Es necesario usar librerías como immutable.js para trabajar la inmutabilidad en nuestros proyectos.

Falso



17. La curva de aprendizaje de Context es más ligera que la de Redux.

Verdadero



18. ¿Cuáles son los usos más comunes de Context?

Datos retornados de una petición asíncrona a un API.

[REPASAR CLASE](#)

**REGRESAR**