



Curso de React.js: Patrones de Render y Composición

¡No te rindas!

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

Vuelve a intentarlo en 05 horas, 54 minutos, 14 segundos

8.75

Calificación

14 / 16

Aciertos

1. ¿React.js es un framework o una librería?

Ninguna herramienta es 100% framework o librería, aunque siempre tienden más hacia un lado que al otro. React es más librería que framework. ✓

2. ¿Cuál es la mejor forma de calificar tu código?

Por legibilidad y rendimiento. ✓

3. ¿Qué son los principios de diseño?

Las reglas, normas y lineamientos generales que sigue un equipo para trabajar. ✓

4. React.js no nos exige una forma muy estricta de trabajar, pero nos provee sus principios de diseño para que al seguirlos nuestro código sea mucho mejor.

Verdadero



5. ¿Qué debemos hacer cuando se publique una nueva versión de React.js?

Actualizar, React nos dará suficiente tiempo para cambiar sin dolor si piensan obliterar alguna funcionalidad.



6. "No cambiar significa quedar obsoleto"

Amén, diosas y dioses del internet



7. ¿Qué es el código aburrido? ¿Es mejor que el código elegante?

Sí, el código aburrido es mejor que el código elegante porque es más fácil de cambiar y mejorar.



8. ¿Qué principios podemos seguir para decidir dónde colocar nuestro estado?

Máxima cercanía a la relevancia y/o stateful vs. stateless components, ambos son compatibles.



9. Tienes un componente App que por dentro tiene un componente TodoHeader que por dentro tiene un componente TodoCounter. ¿Cómo envías props desde App hasta TodoCounter sin usar React Context ni pasar esas mismas props por el componente TodoHeader?

Con React Portals.

REPASAR CLASE

10. ¿Cómo identificas a un componente que aplica composición de uno que no la aplica?

Los componentes que aplican composición no definen su contenido, sino que le delegan esa responsabilidad al componente que los llame.



11. ¿Cuál de los siguientes bloques de código es un ejemplo de render props?

```
<Componente render={() => <OtroComponente />} />
```



12. ¿Cuál de los siguientes bloques de código es un ejemplo de high order components?

```
function Componente(props) { return /* ... */; } const ComponenteWithHOC = withHoc(Componente
```



13. ¿Cuál de los siguientes bloques de código está construyendo un high order component?

```
function withHOC() { return Componente(props) { return /* ... */; } }
```



14. ¿Cuál de los siguientes bloques de código está construyendo un componente que permite usar render props?

```
function Componente(props) { return ( <React.Fragment>{props.render()}</React.Fragment> ); } ✓
```

15. ¿Cuál es una ventaja de los High Order Components?

Facilitan la inyección de props a nuestros componentes. ✓

16. ¿Cuál es una ventaja de las render props o render functions?

Facilitan la inyección de props a nuestros componentes.

REPASAR CLASE

REGRESAR