



## Curso de React.js

## ¡No te rindas!

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

Vuelve a intentarlo en 05 horas, 45 minutos, 32 segundos

# 8.7

Calificación

# 20 / 23

Aciertos

### Resumen

1. ¿Cómo escuchamos el evento de click en un botón con React?

`onClick`

2. ¿Qué es JSX?

Una sintaxis especial de JavaScript para escribir elementos y componentes de React que se siente como HTML.



3. ¿Cómo creamos un contexto en React?

`React.createContext`

4. ¿Para qué sirven los efectos en React?

Para ejecutar bloques de código en componentes únicamente si se cumplen ciertas condiciones en cada nuevo render. ✓

5. ¿Cuál de las siguientes es una forma o herramienta válida para trabajar proyectos con React.js?

Todas las respuestas son correctas. ✓

6. ¿Qué propiedad debemos enviarle al Provider de un contexto en React para consumirlo desde su respectivo Consumer?

value ✓

7. ¿Qué son los eventos en React?

La forma de recibir/escuchar/reaccionar ante los cambios en el estado de los componentes. [REPASAR CLASE](#)

8. ¿Cuál de las siguientes es una forma VÁLIDA de crear un estado en React?

```
const [nombreDelEstado, setNombreDelEstado] = React.useState("valor inicial de estado");
```

 ✓

9. ¿Cuál de los siguientes bloques de código ejecuta nuestro efecto únicamente la primera vez que se renderiza nuestro componente?

```
React.useEffect(() => { console.log("Efectito"); }, []);
```

 ✓

10. ¿Por qué debemos compilar nuestro proyecto con React.js antes de subirlo a GitHub Pages?

No es obligatorio compilar nuestro proyecto antes de subirlo a GitHub Pages. [REPASAR CLASE](#)

11. ¿Cómo creamos un portal en React?

ReactDOM.createPortal



12. ¿Cómo podemos enviar información de un componente "abuelo" a un componente "nieta" sin necesidad de pasar las props por el componente "hijo/padre"?

Usando React Context.



13. ¿Cómo usamos React Context con la sintaxis de React Hooks?

useContext(Contexto)



14. ¿Cuál es la diferencia entre componentes y elementos en React?

Los componentes son grupos de elementos, reciben props y pueden crear estado o efectos. Los elementos reciben atributos o eventos y (casi siempre) se transforman en etiquetas de HTML.



15. ¿Podemos crear más de un estado en nuestros componentes de React?

Verdadero



16. ¿Qué son las props en React?

La forma de comunicar componentes entre sí para transportar información. ✓

17. ¿Cómo escuchamos cuando los usuarios envíen un formulario con React?

onSubmit ✓

18. ¿Cómo escuchamos cuando un usuario escriba en un input o textarea con React?

onChange ✓

19. ¿Para qué sirve React Context?

Para comunicar componentes sin tener que pasar la información como props por cada componente intermedio. ✓

20. ¿Qué es React.js?

React es tanto una librería como una arquitectura. ✓

21. ¿Qué es el estado en React?

La forma en que React guarda información de nuestro componente para escuchar cuando tenga cambios y disparar un nuevo render. ✓

22. ¿Qué significa el "ecosistema de React"?

Todas las herramientas open-source (únicamente las oficiales) relaciona **REPASAR CLASE**

23. ¿Para qué sirven los portales en React?

Para teletransportar componentes a un nodo de HTML distinto al nodo donde hace render el resto de la aplicación. ✓

[Ver menos](#)

## Siguiente curso que podría interesarte

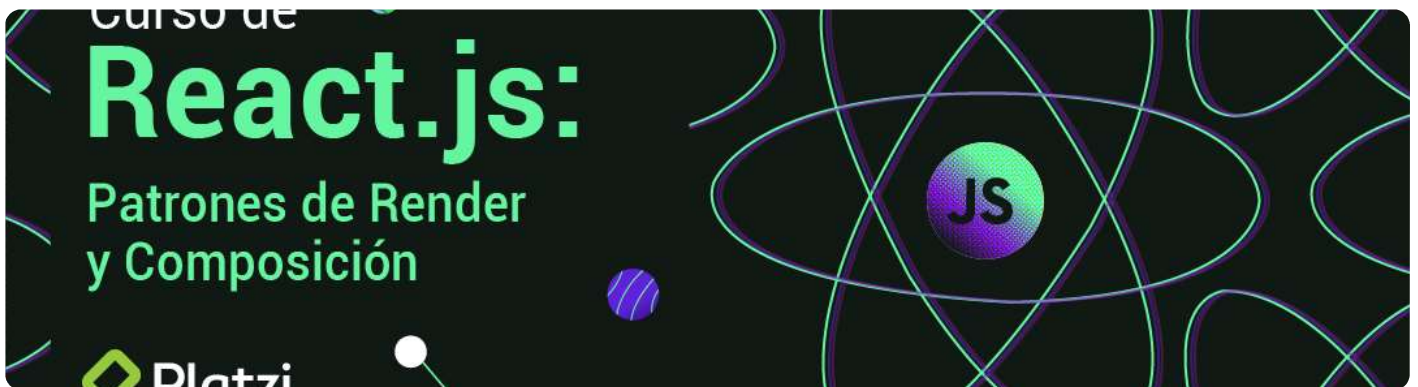
### Taller de **React.js con Vite.js y TailwindCSS**

Por Estefany Aguilar



**Curso de React.js con Vite.js y TailwindCSS**

Por: Estefany Aguilar



**Curso de React.js: Patrones de Render y Composición**

Por: Juan David Castro Gallego

**[Curso de React.js con Vite.js y TailwindCSS](#)** →

**[Volver al Home](#)**