

1. ¿Cuál de los siguientes comandos sirve para crear una aplicación de React?

- ☐ `npx create-react-app`
- ☐ `npx create-react-app app`
- ☐ `npm create-react-app`
- ☐ `npm create-react-app app`

2. ¿Qué es JSX en React?

- ☐ Un lenguaje de consultas para acceder a los datos en los componentes de React.
- ☐ Una herramienta para formatear código de React.
- ☐ Una extensión de sintaxis que permite escribir código parecido a HTML en JavaScript.
- ☐ Un gestor de paquetes para instalar librerías y dependencias de React.

3. ¿Qué *hook* se usa en React para agregar estado a un componente funcional?

- ☐ `useEffect()`
- ☐ `useReducer()`
- ☐ `useState()`
- ☐ `useContext()`

4. ¿Cuál es el propósito de React Router?

- ☐ Manejar peticiones AJAX en aplicaciones de React.
- ☐ Permitir *server side rendering* en React.
- ☐ Gestionar el estado de componentes de React.
- ☐ Implementar navegación y *routing* en aplicaciones de React.

5. ¿Qué comando usamos para iniciar el servidor local de desarrollo?

- ☐ `npm start`
- ☐ `npm serve`
- ☐ `npm build`
- ☐ `npm run dev`

6. ¿Cuál es el puerto que usa por defecto el servidor de desarrollo?

- ☐ 8080
- ☐ 5000
- ☐ 4000
- ☐ 3000

7. ¿Qué tipo de elemento aparece en pantalla con el siguiente código?

```
const Car = () => <h1>Ford Focus</h1>;  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<Car />);
```

- ☐ `h1`
- ☐ `div`
- ☐ `ReactDOM`
- ☐ `Component`

8. ¿Para qué sirve la propiedad *children*?
- ☐ Para usar objetos como propiedades.
 - ☐ Para pasar datos a componentes que son hijos de otro componente.
 - ☐ Para agregar estado a los hijos de los componentes.
 - ☐ Para anidar componentes en otros componentes.
9. ¿Qué operador usamos en React para realizar un *render* condicional?
- ☐ ??
 - ☐ &&
 - ☐ ::
 - ☐ ||
10. Cuando mostramos una lista en React usando la función `map()`, ¿qué propiedad se debe poner en cada elemento de la lista?
- ☐ id
 - ☐ index
 - ☐ key
 - ☐ data
11. Completar los espacios en blanco.
- (a) `<p style=_____>Texto en amarillo</p>`
 - (b) `{user _____ <Profile user={user} />}`
 - (c) `<div _____="App">Mi app</div>`
 - (d) `const name = 'Juan'`
`return <Person name=_____ />;`
 - (e) `{items.map((item) => <li _____={item.id}>{item.name})}`
 - (f) `const _____ = props;`
`return ;`
 - (g) `<form _____={handleSubmit}>`
 - (h) `const [count, setCount] = useState(0);`
`return <button onClick=_____>Contar</button>;`
12. Indicar verdadero o falso.
- (a) Se necesita NodeJS para desarrollar en React. ____
 - (b) En React al pasar un estilo en línea se usa un *string* como valor del atributo `style`. ____
 - (c) Un componente de React puede devolver más de un elemento de JSX. ____
 - (d) React está pensado para realizar SPAs principalmente. ____
 - (e) Los componentes de React deben comenzar con letra mayúscula. ____
 - (f) En una aplicación web, React se encarga de las operaciones con la base de datos. ____
 - (g) Las *props* sirven para gestionar datos que pueden variar dentro de un componente. ____
 - (h) Los *hooks* sólo pueden usarse dentro de componentes funcionales. ____
 - (i) El *hook* `useEffect()` se utiliza para gestionar el estado de un componente funcional. ____
 - (j) Los componentes de React pueden pasar datos de hijos a padres usando la técnica conocida como *prop drilling*. ____

13. Crear un componente en React para mostrar un encabezado de página. Usar *props* para pasar el nombre y logo de la marca.
14. Crear un componente en React para mostrar una barra de navegación. Obtener los nombres de los links a partir de un *array*.
15. Crear un componente que muestre un h1 con un número al azar entre 1 y 1000. Al clickear el h1 elegir un nuevo número.
16. Crear un componente para mostrar un producto usando al menos un título, descripción, precio e imagen.
17. Crear un componente que use el componente anterior para mostrar una lista deslizable de productos.
18. Crear un componente para mostrar un reloj, con horas, minutos, segundos y la fecha.
19. Crear un componente para mostrar un cronómetro.
20. Combinar los dos componentes anteriores en uno que me permita cambiar entre cronómetro y reloj usando pestañas o algo similar.
21. Crear un componente para mostrar una tabla. Usar como propiedad del componente un *array* de objetos. Obtener tanto las filas de la tabla como el encabezado a partir del *array*.
22. Crear un componente para mostrar una barra de búsqueda. Usar la barra de búsqueda para filtrar una lista de equipos de fútbol. Usar un *array* para llenar la lista de equipos.
23. Crear un componente para mostrar una tarjeta de un Pokemon. Usar una barra de búsqueda para buscar un Pokemon por nombre. Usar la API en <https://pokeapi.co/api/v2/>.
24. Crear una aplicación para mostrar cuantos días faltan para el próximo feriado.
25. Crear una aplicación de lista de tareas. Guardar las tareas en `localStorage`.
26. Crear una aplicación para mostrar fotos desde <https://unsplash.com>. Incluir una barra de búsqueda para que el usuario pueda elegir la categoría de las fotos.
27. Crear un blog que muestre una lista de posts. Usar formularios para agregar nuevos posts y también comentarios en cada post. Es recomendable usar React Router para mostrar distintas páginas, como por ejemplo una por cada post.
28. Crear una aplicación para cifrar un texto usando una selección de algoritmos de cifrado. Los más simples de interpretar son los de rotación. Inspirarse en el sitio <https://cryptii.com/>.
29. Crear una aplicación para jugar trivias usando la API en <https://opentdb.com>. Mostrar el puntaje obtenido al final de cada trivia.
30. Crear una web de comercio electrónico usando React Router para tener diferentes rutas como `localhost:3000/products` o `localhost:3000/cart`. Obtener la lista de productos a la venta desde un *array* de objetos en un archivo o usar una API como <https://fakestoreapi.com/products>.

Pueden encontrar una buena colección de proyectos de React para practicar en <https://react-projects.netlify.app/>. Hay plantillas listas para usar en <https://github.com/john-smilga/react-projects>. Clonar el repositorio completo y usar el directorio `setup` para comenzar con cada proyecto. Los hay de distintos grados de dificultad. Si se defienden con el inglés hay un video del canal de freeCodeCamp con 15 de esos proyectos resueltos paso a paso en https://www.youtube.com/watch?v=a_7Z7C_JCy0.