	PRUEBA DE CONOCIMIENTO REACT		
Samtel	**Nombre:**		
	Identificación:		
	Teléfono:		**Fecha:**
	Título Profesional:		

1. **¿Cuál es la principal diferencia entre los componentes de clase y los componentes funcionales en React?**

   a. Los componentes de clase pueden manejar estados y ciclos de vida, mientras que los funcionales no
   X. Los componentes funcionales son más eficientes y permiten el uso de hooks
   c. Los componentes de clase son recomendados para proyectos nuevos, mientras que los funcionales no
   d. Los componentes funcionales no pueden usar Redux

2. **¿Qué hook se utiliza para manejar efectos secundarios en componentes funcionales?**

   a. useState
   X. useEffect
   c. useContext
   d. useReducer

3. **¿Cómo puedes evitar que un componente hijo se vuelva a renderizar innecesariamente?**

   X. Usando React.memo
   b. Usando useState
   c. Usando useEffect
   d. Usando useReducer

4. **¿Qué técnica usarías para manejar el estado global en una aplicación React grande?**

   X. Context API o Redux
   b. Local Storage
   c. Componentes de clase
   d. Prop Drilling

5. **¿Qué hook se debe utilizar para actualizar el estado basándose en el estado anterior?**

   X. useState
   b. useEffect
   c. useContext
   d. useReducer

Samtel

6. **¿Cómo puedes optimizar la carga de una aplicación React dividiendo el código en partes más pequeñas?**

X a. Code Splitting
b. Lazy Loading
c. Prop Drilling
d. Memoization

7. **¿Qué método del ciclo de vida de un componente de clase se utiliza para ejecutar código después de que el componente se ha renderizado?**

X a. componentDidMount
b. componentDidUpdate
c. componentWillUnmount
d. componentWillMount

8. **¿Cuál es el propósito de useReducer en React?**

X a. Manejar el estado complejo en componentes funcionales
b. Reducir el tamaño de la aplicación
c. Optimizar la performance de la renderización
d. Simplificar la estructura del componente

9. **¿Qué es el Context API y cuándo deberías usarlo?**

X a. Es una forma de pasar datos a través del árbol de componentes sin tener que pasar props manualmente
b. Es una herramienta para optimizar la performance
c. Es un hook para manejar estados locales
d. Es una API para manejar efectos secundarios

10. **¿Cómo puedes manejar formularios en React de manera eficiente?**

a. Usando componentes controlados y no controlados
X b. Usando useState para cada campo de formulario
c. Usando useEffect para validar campos
d. Usando useContext para manejar el estado del formulario

11. **¿Qué es JSX y cómo se diferencia de HTML?**

X a. Es una extensión de JavaScript que permite escribir HTML dentro de JavaScript
b. Es un lenguaje de marcado similar a HTML, pero con sintaxis diferente
c. Es un preprocesador que convierte JavaScript en HTML
d. Es una librería para manejar estados en React.

12. ¿Cuál es la función de useEffect en React?

a. Manejar estados locales
X b. Realizar efectos secundarios en componentes funcionales
c. Crear contextos globales
d. Optimizar la performance de los componentes

13. ¿Qué es un "key" en React y por qué es importante?

X a. Un identificador único para cada elemento en un array
b. Una variable global en React
c. Un hook especial para manejar arrays
d. Un evento especial que se dispara en un componente

14. ¿Cómo se puede manejar el estado local en un componente funcional?

X a. Con el hook useState
b. Con el hook useContext
c. Con el método setState
d. Con el método componentDidMount

15. ¿Qué es React Router y para qué se utiliza?

X a. Una librería para manejar la navegación en aplicaciones React
b. Un hook para manejar estados en React
c. Un método para optimizar la carga de componentes
d. Un contexto para compartir datos globales

16. ¿Qué es Redux y cuándo deberías usarlo?

X a. Una librería para manejar el estado global de una aplicación
b. Una librería para optimizar componentes
c. Un hook para manejar efectos secundarios
d. Un preprocesador de CSS

17. ¿Qué es un Hook y por qué son importantes en React?

X a. Funciones especiales que permiten usar estados y otras características de React en componentes funcionales
b. Métodos de ciclo de vida para componentes de clase
c. Propiedades especiales que se pasan a componentes
d. Eventos especiales que se disparan en un componente

18. ¿Cómo puedes evitar que un componente se vuelva a renderizar innecesariamente?

X a. Usando React.memo y useCallback
b. Usando useState
c. Usando useEffect con dependencias vacías
d. Usando componentDidUpdate

19. ¿Qué técnica usarías para cargar componentes de manera diferida?

X a. Lazy Loading con React.lazy y Suspense
b. Usar useState
c. Usar useEffect con dependencias
d. Usar useReducer