

React

1. What are props in a component? Select all the options that apply.

- They are the parameters of a component.
- It is a mechanism to put data inside the component.

2. **What is a component? Mark only one option.**

- Other:

Los componentes nos permiten crear interfaces graficas, estos encapsulan codigo, estilos y estados.

Cada componente es independiente y puede ser reutilizado.

Estos se pueden definir como funciones o clases y se le pueden pasar datos a travez de un parametro llamado props, los componentes devuelven elementos de react que se muestran en pantalla.

3. **Which of these component types are valid? Select all the options that apply.**

- Class Component.
- Component Function

4. **What does the term JSX refer to?**

- Similar in appearance to XML and HTML, JSX is used to create elements using a familiar syntax.

5. **How do I pass a value from a parent component to a child component? Select all the options that apply.**

- Using props.
- Referring the child component to the values of the parent component.

6. Which of these implementations change the state of a component?
Check only one option.

```
import { useState } from "react";  
const [value, setValue] = useState("Value")
```

- setValue("my value")

7. How many and which arguments does useEffect Hook receive?

- Two arguments, which are the effect and the dependencies.

8. How do you import Javascript modules installed with NPM or YARN?
Check only one option

- import name from 'name'

9. What functionality does the use of Context API provide?

- It is a mechanism that provides a way to pass data through the component tree without manually passing props at all levels

10. How do I import the following component, considering that it is inside a file in the same directory, called MyComponent.js?

```
export const MyComponent = () => {  
  return (  
    <h1>Hello World</h1>  
  )  
}
```

- import { MyComponent } from './MyComponent';

11. What is the difference between class components and functional components?

Los componentes funcionales es una funcion pura de JavaScript que devuelve un elemento de react, mientras que los componentes de clase como dice su nombre se definen como clases y requiere que extiendan del Component de React, estos compoenentes necesitan un constructor y deben de tener un metodo Render que devuelve JSX que se mostrara en pantalla.

Inicialmente los componentes de clases permitian el manejo de estados y los compoenentes funcionales no lo permitian. En versiones recientes de React se introdujo el manejo de Hooks que nos permiten introducir estados en compoenentes funcionales, los hooks tiene diferentes sintaxis en ambos tipos de componentes.

Por otro lado los metodos de ciclo de vida de React como `componentDidUpdate()` no se pueden usar en un componente funcional y si en un componente de clases.

En versiones recientes de React se introdujo el concepto de `useEffect` que permiten hacer uso del ciclo de vida en un componente funcional.

12. How would you set the state of a component?

Componente de clase

```
constructor(props){
  super(props)
  this.state = {
    contador: 0
  }
}
```

Componente funcional

```
const [contador, setContador] = useState(0);
```

13. What happens when we change the state of a component?

Cada vez que se actualiza o cambia el estado de un componente, se vuelve a renderizar el componente y reacciona a dicho cambio.

14. Write a component that renders a list of elements, given a prop that contains an array of objects in the following form: { firstname: "demo", lastname: "demo", dni: 1234 }

```
/* Uso la destructuracion para pasar los datos */
export const ListElement = ({firstname,lastname,dni}) =>{
  return(
    <ul>
      <li>Nombre: {firstname}</li>
      <li>Apellido: {lastname}</li>
      <li>DNI: {dni}</li>
    </ul>
  )
}
```

15. Write a component that receives a prop called "data", and renders an H1 with the content of data.

```
export const Component = (data) => {
  let {name, lastName} = data
  return(
    <h1>Bienvenido {name} {lastName}</h1>
  )
}
```

16. What is the error in the following component?

```
export const Button = ({ text }) => {
  <div className="button">
    <h1 className="text-button">{text}</h1>
  </div>;
};
```

El error es que no se esta retornando ningun elemento de React que se mostrara en pantalla

17. Explain, in your own words, what this component does.

```
import { useState } from "react";

export const Component = () => {
  const [state, setState] = useState(1);

  const click = (propertyValue: number) => {
    setState(propertyValue + 1);
  };

  return (
    <>
      <h1>{state}</h1>
      <button onClick={() => click(state)}>Click here</button>
    </>
  );
};
```

Lo que hace incrementar en uno el estado del componente cada vez que se hace click.

18. How would you make a Rest API call from JavaScript?

```
/* El parametro options es un objeto de opciones que se le pasan a las peticiones,
estas comunmente son:
  method: GET, POST, UPDATE o DELETE
  Headers: Es un objeto donde se pasan cabeceras de la peticion como por ejemplo
{"content-type": "application/json"}
  body: Que son los datos que reciben la peticion, generalmete se usan en petici
ones POST o UPDATE
*/

fetch(url, options)
.then(res => res.ok
  ? res.json()
  : Promise.reject(
    {
      error: true,
      msg: "ocurrio un error en la peticion"
    }
  ))
.catch(err => err)
```

19. What is the use of the useEffect hook?

Permite hacer uso del ciclo de vida de React en un compoenente funcional, recibe como parametro una funcion que ejecuta cada vez que nuestro

compoenente se renderiza ya sea por cambio en los estados o las propiedades.

20. **How would you develop a custom hook that exposes methods to save and modify a counter? The hook must have a parameter that allows the developer to specify how much the counter should add for each call to the setter.**

```
import { useState, useEffect } from 'react';

export const useCounter = (valueInitial = 0) => {
  const [counter, setcounter] = useState(valueInitial);

  const reset = () => setcounter(valueInitial)
  const increase = num => setcounter(counter + num);
  const subtract = num => setcounter(counter - num);

  return {reset, increase ,subtract}
}
```