Respuestas a ejercicios:

JavaScript:

1. Lo que se mostrara en la pantalla es el resultado de lala suma de las variables +2 \* a\*b;

* Y una cuarta alerta que intenta mostrar el tipo de la variable "squareOfA", que no está definida fuera de la función y, por lo tanto, muestra un mensaje de error.

React:

1. ¿Qué son los accesorios en un componente?

* Respuesta Nº3: Es un mecanismo para poner datos dentro del componente.

1. ¿Qué es un componente?

* Respuesta Nº2: Una función o clase que devuelve un elemento.

1. ¿Cuáles de estos tipos de componentes son válidos?

* Respuesta Nº2 y 3: Componente de clase.

Función componente.

1. ¿A qué se refiere el término JSX?

* Respuesta Nº2: Similar en apariencia a XML y HTML, JSX se usa para crear elementos usando una sintaxis familiar.

1. ¿Cómo paso un valor de un componente principal a un componente secundario?

* Respuesta Nº1 y 4: Usando accesorios.

Referir el componente hijo a los valores del componente padre.

1. ¿Cuál de estas implementaciones cambia el estado de un componente?

* Respuesta Nº2: setValue(“mi valor”)

1. ¿Cuántos y qué argumentos recibe useEffect Hook?

* Respuesta Nº3: Dos argumentos, que son los estados de las variables y el

dependencias

1. ¿Cómo se importan los módulos de Javascript instalados con NPM o YARN?

* Respuesta Nº3: importar el nombre de '../../node\_modules/name/index.js';

1. ¿Qué funcionalidad proporciona el uso de Context API?

* Respuesta Nº1: Es un mecanismo que proporciona una forma de pasar datos a través del componente.árbol sin pasar accesorios manualmente en todos los niveles.

1. ¿Cómo importo el siguiente componente, considerando que está dentro de un

archivo en el mismo directorio, llamado MyComponent.js?

* Respuesta Nº4: importar { MiComponente } desde './MiComponente.js';

1. ¿Cuál es la diferencia entre componentes de clase y componentes funcionales?

* Respuesta: Un componente de clase se diferencía de un componente funcional en dos principales cosas: La capacidad de administrar un estado propio del componente y la exposición de métodos que pueden ejecutar funcionalidad personalizada en distintas etapas del ciclo de vida de un componente.

1. ¿Cómo establecería el estado de un componente?

* Respuesta: Definiéndolo en el constructor del componente mediante la asignación directa a la propiedad this. state o establecerlo dinámicamente en cualquier función o evento, mediante la función setState() .

1. ¿Qué sucede cuando cambiamos el estado de un componente?

* Respuesta: El componente responde volviendo a renderizar.

16. ¿Cuál es el error en el siguiente componente?

* Respuesta:El error en el componente es que falta la palabra clave "return" antes de la definición del elemento <div>. El componente no devuelve nada y aparece como si estuviera vacío en la interfaz de usuario.

17. Este componente es un ejemplo sencillo de cómo usar el gancho useState en React. Dentro de la función de componente Component, se declara una variable llamada "state" utilizando la función useState, y se le asigna un valor de 1. Luego la funcion “click” utiliza el método "setState" que viene con el estado para actualizar el valor de la variable "state". La actualización del valor es sumar 1 al valor del estado.En el cuerpo del componente, se muestra el valor actual del estado en un elemento h1 que cuando se llama a la función click vuelva a renderizar los componentes con el nuevo valor actualizado del estado.

19. ¿Cuál es el uso del gancho useEffect?

* Respuesta: permite que los componentes funcionales hagan cosas cuando se monta un componente o cuando cambian algunas propiedades o estados.