REACT HOOKS

useEffect

Ejecuta algo en los siguientes casos:

1. Cuando se carga o se actualiza un componente. Queda en pausa

useEffect(() => {

  setCount(0);

}, []);

1. Cuando se carga el componente y se actualiza lo que esta dentro del array.

useEffect(() => {

  console.log(first);

}, [count]);

1. Cuando se descarga o desmonta un componente

useEffect(() => {

  return () => { console.log('El componente ha sido desmontado');}

}, []);

useEffect(() => {

   console.log('El componente ha sido cargado');

}, []);

useEffect(() => {

  return () => { console.log('El componente ha sido descargado');}

}, []);

useEffect(() => {

  return () => { console.log('El componente ha sido cargado y actualizado');}

}, []);

**CREAR UNA REFERENCIA A UN HTML COMO UN BOTON -> useRef**

**Vimos:**

**useState**

**useEffect**

**useRef**

**useMemo**

**useCallback**

**y hook personalizado, ej useGonzalo… siempre empienzan con use**

**PARA REPASAR PARA EL CHECKPOINT:**

**COMPONENTES DE TIPO CLASE**

1. **COMPONENTDIDMOUNT, COMPONENDIDUPDATE, COMPONENTWILLUNMOUNT**
2. **STATE Y SETSTATE**
3. **TODO ESTO COMPONENTES DE TIPO CLASE**

**COMPONENTES FUNCIONALES**

1. **STATE Y USESTATE**
2. **PASAR PROPS Y EVENTOS**
3. **APLICARLE CLASES A UN ELEMENTO Y APLICARLE CLASES DE MANERA CONDICIONAL (EJ CLASE DANGER DE LOS FORMULARIOS).**
4. **FORMULARIOS CONTROLADOS, ONCHANCHE, VALUE.**
5. **VAN A PEDIR ALGUNOS INPUTS Q REQUIEREN UN TYPE ESPECIFICO Y POSIBLEMENTE NO PIDE UN INPUT SINO OTRA COSA, HAY Q LEER BIEN EL ENUNCIADO.**

**OTRAS LIBRERIAS**

1. **REDUX THUNK**
2. **SELECT, TEXARIA, SELECT, PASSWORD, DATE??**
3. **REDUX, ACTIONS, REDUCER, STORE, REDUX THUNK, CONNECT, USEDISPATCH O USESELECT.**
4. **REACT.USEFFECT REACT.USESTATE**
5. **ROUTING…**