



ACTIVIDAD REFUERZO

REACT

ADSI57

ANALISIS Y DESARROLLO DE SISTEMAS DE
INFORMACION

CENTRO NACIONAL COLOMBO ALEMAN
SERVICIO NACIONAL DE APRENDIZAJE SENA

Barranquilla

2021

GC-F -005 V. 01



```
c:\>cd react  
c:\react>create-react-app actividad
```

Pautas solución:

- Importar imágenes
- Importar archivo css
- En a funcion App → definir un arreglo con los nombres de los reyes
- En return → un div contenedor con clase porejemplo contenedor
- Dentro del div contenedor tres div con clase comun porejemplo caja, en cada uno una etiqueta de imagen y un div con clase comun porejemplo “nombre” y que tome l primer nombre del array y asi sucesivamente en los otros 2 div.
- Ejecutar la aplicación → npm start
- En el css:
- Aplicar reglas correspondientes

1

Aceptar





JS App.js

Pautas:

- Etiquetas vacías o con Fragment
- Eventos(onClick/onChange)
- React a la hora de colocar eventos los coloca en la etiqueta
- Lo que es js en jsx de react se coloca dentro {}
- Todo lo que tiene que ver con creación de variables, funciones entre otras estructuras se coloca en la zona de js puro y duro y que está entre function App() y return
- Pueden trabajar las funciones de dos maneras:
 - Function habitual y tradicional
 - Funciones de flecha(algo así como un “js moderno”
- Const se utiliza mucho en React cuando un elemento no cambia en su valor
- Let cuando un elemento puede cambiar de valor el var ya está obsoleto pero no quiere decir que esto sea contagioso
- En la función del evento js recopila una serie de datos o valores de información que si quiero gestionarlos o acceder a ellos, pueden utilizar un parámetro que puede ser la e o

- la z o cualquier carácter a través de un `console.log(e)` , no funciona con `alert (e)`
- con F12 van al inspector desde el navegador(sección console)
- en nuestro caso nos interesa el `target` que nos indica a que elemento hemos hecho click
ejemplo: `div.caja` donde `div` es la caja contenedora
- utilizamos `innerHTML` para saber el contenido de la caja:
 - `console.log(e.target.innerHTML)`
- Recordar o consultar que para aplicar estilos en línea a través de `style` utilizar para sus propiedades `camelCase`, recuerden que en `css` hablamos de selectores no etiquetas
 - `P {`
 `Color:red; // declaracion=propiedad + valor de propiedad`
 `}`

- Las referencias en vez de utilizar `document.getElementById` como el que observan en `index.js`
- `useRef()` es un hook que es algo así como una función que nos permite hacer algo en nuestro caso es referenciar un elemento
- se debe importar el `useRef`
- `const refCaja=useRef()` constante para guardar la referencia de un elemento, es decir lo que almacena la constante
- siempre que utilicen una referencia en `js` hay que poner el `current`

```
function convertir(){
```

`Number(refCaja.current.innerHTML*cambio)`

