

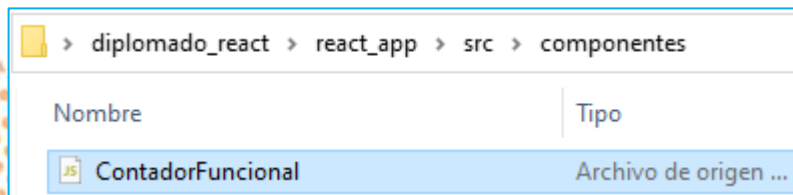
# De Componentes en Clases A Componentes en Funciones

## Actividad:

**Convertir el componente  
"ContadorClass.js" basado en clases a  
un componente basado en funciones**

# De Componentes en Clases A Componentes en Funciones

A continuación, vamos a realizar la transformación de nuestro componente basado en clases a un componente basado en funciones. Para ello entonces lo primero que haremos es crear un archivo llamado "ComponenteFuncional.js" y dentro de él colocaremos la estructura básica de nuestro componente:



En esta estructura, lo primero que colocaremos es la importación de React y del Hook de "useState" para poder trabajar con el estado. Seguidamente en la parte interior de nuestro componente colocaremos el encabezado y los botones para incrementar y disminuir el número de clases:

```
1  import React, {useState} from 'react';
2  const ContadorFuncional = () => {
3      return (
4          <div>
5              <h2>Esta es la clase número </h2>
6              <button>Incrementar</button>
7              <button>Disminuir</button>
8          </div>
9      );
10 }
11 export default ContadorFuncional;
```

# De Componentes en Clases A Componentes en Funciones

Seguidamente vamos al index, importamos el componente funcional, lo mostramos en pantalla, comentamos el componente basado en clases y le pasamos a nuestro componente funcional cómo propiedades la cantidad a incrementar y a disminuir:

```
8 import ContadorFuncional from './componentes/ContadorFuncional';

16 <Aula/>
17 /*<ContadorClass incrementar={1} disminuir={1}/>*/
18 <ContadorFuncional incrementar={1} disminuir={1}/>
19 <Boton rojo onClick={() => cambiarEstadoSesion(false)}>Cerrar Sesión</Boton>
```

Luego dentro de nuestro componente funcional, recibiremos las propiedades al momento de crear el componente:

```
2 const ContadorFuncional = (props) => {
```

Adicionalmente creamos el estado contador, la función para modificarlo y las funciones para incrementar y disminuir el valor del mismo, según la cantidad enviada desde el index:

```
3 const [contador, cambiarContador] = useState(0);
4 const incrementar = (cantidad) => {
5   |   cambiarContador(contador + cantidad);
6   | }
```

```
7 const disminuir = (cantidad) => {
8   |   cambiarContador(contador - cantidad);
9   | }
```

# De Componentes en Clases A Componentes en Funciones

Hecho esto, dentro de nuestro botones colocamos el evento "onClick" y llamamos a la función correspondiente de incrementar o disminuir pasando a las mismas la cantidad enviada desde el index a través de las propiedades:

```
10     return (  
11         <div>  
12             <h2>Esta es la clase número </h2>  
13             <button onClick={() => incrementar(props.incrementar)}>Incrementar</button>  
14             <button onClick={() => disminuir(props.disminuir)}>Disminuir</button>  
15         </div>  
16     );
```

Para finalizar esta conversión, colocamos el estado contador en nuestro encabezado <h2> para que así comience a mostrarse en nuestra app:

```
12     <h2>Esta es la clase número {contador}</h2>
```

# De Componentes en Clases A Componentes en Funciones

Tras la realización del proceso anterior, lograremos ver en pantalla nuestro contador de funciones funcionando: Cada vez que demos clic en el botón "Incrementar" sumaremos un (1) dígito a la cuenta y cuándo demos clic en "Disminuir" restaremos un (1) dígito:

