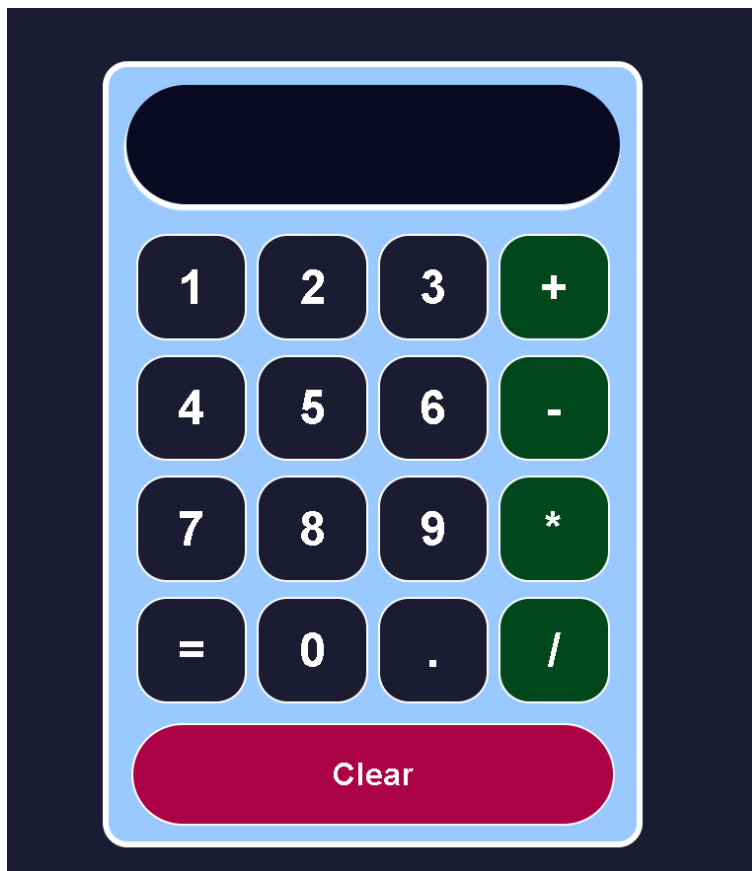


CALCULADORA

Actividad 3.....	1
PREPARACIÓN DEL PROYECTO	2
CREAMOS CARPETAS NUEVAS	2
AÑADIMOS/SUSTITUIMOS LAS HOJAS DE ESTILO.....	2
CREAMOS LA CALCULADORA EN APP.JS	3
CREAMOS EL COMPONENTE Pantalla	3
CREAMOS EL COMPONENTE Boton	3
CREAMOS EL COMPONENTE BotonClear	4
PROGRAMAMOS EN App.js	4

Actividad 3: CALCULADORA



PREPARACIÓN DEL PROYECTO

Eliminamos archivos que no vamos a utilizar

Vamos a trabajar en la carpeta **src**, pero antes eliminamos archivos que no vamos a necesitar, por ejemplo, setupTests.js, reportWebVitals.js y App.test.js.

Limpiamos el archivo App.js

Queremos que aparezca en principio una página en blanco así que modificamos *App.js* eliminando todo lo que hay dentro del div principal, sin borrar ese div principal.

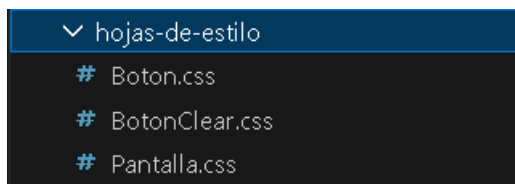
Eliminamos también la primera línea del archivo en la que se importa el logo.

Eliminamos líneas que no vamos a usar de index.js

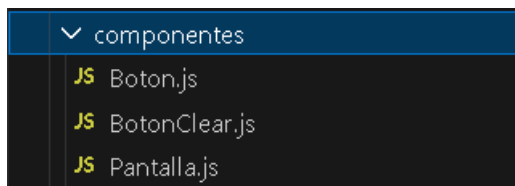
Éstas son todas las que hagan referencia a "reportwebVitals" que hemos eliminado anteriormente. Eliminamos también los comentarios.

CREAMOS CARPETAS NUEVAS

Para comenzar creamos la carpeta que almacenará el css de los elementos Boton, BotonClear y Pantalla creado por nosotros:



Y por último creamos otra carpeta que albergará nuestros archivos react con nuestros componentes Boton, BotonClear y Pantalla:



AÑADIMOS/SUSTITUIMOS LAS HOJAS DE ESTILO

Tendremos que actualizar las hojas de estilo App.css e index.css.

Añadimos la nueva hoja de estilo de nuestro elemento Boton con Boton.css.

Añadimos la nueva hoja de estilo de nuestro elemento BotonClear con BotónClear.css.

Añadimos la nueva hoja de estilo de nuestro elemento Pantalla con Pantalla.css.

CREAMOS LA CALCULADORA EN APP.JS

En primer lugar creamos un div con la clase **contenedor-calculadora** que contendrá todo lo demás.

La calculadora estará compuesta por:

- Un componente **Pantalla**
- Cinco filas de botones, cada una con su div correspondiente con la clase **fila**.
 - o Cuatro componentes **Botón** en cada una de las cuatro filas primeras.
 - o Un componente **BotonClear** en la última fila.

CREAMOS EL COMPONENTE Pantalla

El componente **Pantalla** va a recibir un props con lo que se ha escrito por pantalla y lo va a mostrar dentro de un div con la clase **input**.

Estos serán los números u operaciones pulsadas o el resultado final. Lo haremos más adelante. Por ahora que simplemente muestra el props enviado al que puedes ponerle el nombre **input**, por ejemplo.

Es decir, un ejemplo de lo que devolverá este componente puede ser:

```
<div class="input">  
  4 + 6 / 3  
</div>
```

CREAMOS EL COMPONENTE Boton

El componente **Boton** va a recibir un props genérico y va a estar formado por un div con la clase **botón-contenedor**.

Además si es un botón de operaciones (+ , - , * o /) va a tener otra clase **operador** el div anterior.

Para poder obtener el número pulsado o la operación pulsada en nuestra calculadora haremos uso de la propiedad **props.children**, la cual devuelve lo que hay dentro del componente Boton de App.js.

<botón> Lo que hay dentro </botón>

Cuando hagamos click en el botón (*onClick*) llamaremos a una función denominada **manejarClic()** que estará definida en App.js.

Es decir, un ejemplo de lo que devolverá este componente en la tecla 3 de la calculadora puede ser:

```
<div class="botón-contenedor" onClick='manejarClic(3) '>  
  1  
</div>
```

Y un ejemplo de lo que devolverá este componente en la tecla + de la calculadora puede ser:

```
<div class="botón-contenedor operador" onClick='manejarClic(+)'>  
  +  
</div>
```

CREAMOS EL COMPONENTE **BotonClear**

El componente **BotonClear** opera también con la palabra contenida dentro del componente **BotonClear** (ver apartado CREAMOS EL COMPONENTE Boton).

Va a tener un div de la clase botón-clear y cuando hagamos click en él llamará a la función manejarClear. Devolverá el texto contenido en el componente BotonClear en el App.js.

Es decir, este componente devolverá:

```
<div class="botón-clear" onClick='manejarClear'>  
  Clear  
</div>
```

PROGRAMAMOS EN App.js

Vamos a utilizar **useState** para tener en cuenta el estado del componente Pantalla. Para ello declaramos una variable input y una función setInput que la actualice.

Al componente pantalla le vamos a pasar directamente ese valor {input} cómo propiedad (props).

A continuación vamos a tener que implementar tres funciones:

- **agregarInput()**: Añade a lo que ya haya mostrado en la pantalla el número pulsado o la operación pulsada, siempre que sea +, -, * o /.
- **calcularResultado()**: Actualiza la pantalla al cálculo del resultado de la expresión que aparece en la pantalla. Para ello hacer uso de la función matemática **evaluate**(expresión). Y no olvides importarla desde **mathjs**. La expresión para importarla es igual que la utilizada con useState.

- ***manejarClear()***: Actualiza la pantalla a pantalla limpia ("").