

Calculando el Presupuesto Gastado

- Para hacer los cálculos de Gastado y Disponible, he de pasar el arreglo de gastos a ControlPresupuesto.
- Por ello se lo paso via props en el header y del header a Control Presupuesto
- Importo useEffect en ControlPresupuesto para que esté escuchando por los cambios que sucedan en gastos
- Si yo le pongo un console.log, en el momento de que un presupuesto es valido ya se monta el componente y ya puedo visualizarlo en consola
- En ese momento ya puedo definir lo disponible y lo gastado
- Para ello voy a utilizar dos piezas de state

```
const [disponible, setDisponible] =useState(0)
const [gastado, setGastado] =useState(0)

useEffect(() => {

}, [gastos])
```

Agrego los states

```
return (
  <div className="contenedor-presupuesto contenedor sombra dos-columnas">
    <div>
      <p>Gráfica aquí</p>
    </div>
    <div className="contenido-presupuesto">
      <p>
        <span>Presupuesto: </span> {formatearCantidad(presupuesto)}
      </p>
      <p>
        <span>Disponible: </span> {formatearCantidad(disponible)}
      </p>
      <p>
        <span>Gastado: </span> {formatearCantidad(gastado)}
      </p>
    </div>
  </div>
)
```

- Calculo primero lo gastado por que de eso depende lo disponible
- Si tienes un arreglo con objetos, el método más útil para esto es .reduce
- Va a acumular una gran cantidad de datos en una sola variable.

- Le vamos a decir que los gastos, que va a tener un arreglo de objetos con todos los gastos y la info, en gasto tenemos la cantidad y podemos leer esa cantidad y acumularlo en una variable que llamaré total gastado

NOTA: es importante en este caso que el arreglo de gastos lo inicié como un arreglo vacío y no como un objeto, porque por eso tengo acceso al método `.reduce`

- `.reduce` toma dos argumentos, uno es el acumulador y el otro es la instancia del objeto donde va a ir iterando
- Entonces con un callback le digo que va a ir acumulando de `gasto.cantidad` en el total y el valor del que va a iniciar que es 0

```
import {useEffect, useState} from 'react'

const ControlPresupuesto = ({presupuesto, gastos}) => {

  const [disponible, setDisponible] =useState(0)
  const [gastado, setGastado] =useState(0)

  useEffect(() => {
    const totalGastado = gastos.reduce((total,gasto)=> gasto.cantidad + total, 0)
  }, [gastos])

  const formatearCantidad =(cantidad) =>{
    return cantidad.toLocaleString('en-US', {
      style: 'currency',
      currency: 'USD'
    })
  }

  return (
    <div className="contenedor-presupuesto contenedor sombra dos-columnas">
```

- Si pongo un `console.log` con `totalGastado` en el `useEffect`, puedo comprobar si funciona en la consola
- Veo que si, puedo usar `setGastado(totalGastado)` para escribir el state que antes coloqué



Lo siguiente es calcular lo disponible

```
useEffect(() => {  
  const totalGastado = gastos.reduce((total,gasto)=> gasto.cantidad + total, 0)  
  const totalDisponible= presupuesto - totalGastado;  
  setDisponible(totalDisponible)  
  setGastado(totalGastado)  
}, [gastos])
```