

Crear Componente de Gasto

- Voy a App.jsx y donde está isValidPresupuesto lo envuelvo todo en un Fragment y añado un main.
- Puedo cambiar el ternario que había por un &&. Si "esto" haz "esto otro".

```
{isValidPresupuesto && (  
  <>  
    <main>  
  
    </main>  
  
    <div className="nuevo-gasto">  
      <img src={IconoNuevoGasto} alt="icono-gasto"  
        onClick={handleNuevoGasto}/>  
    </div>  
  
  </>  
)}
```

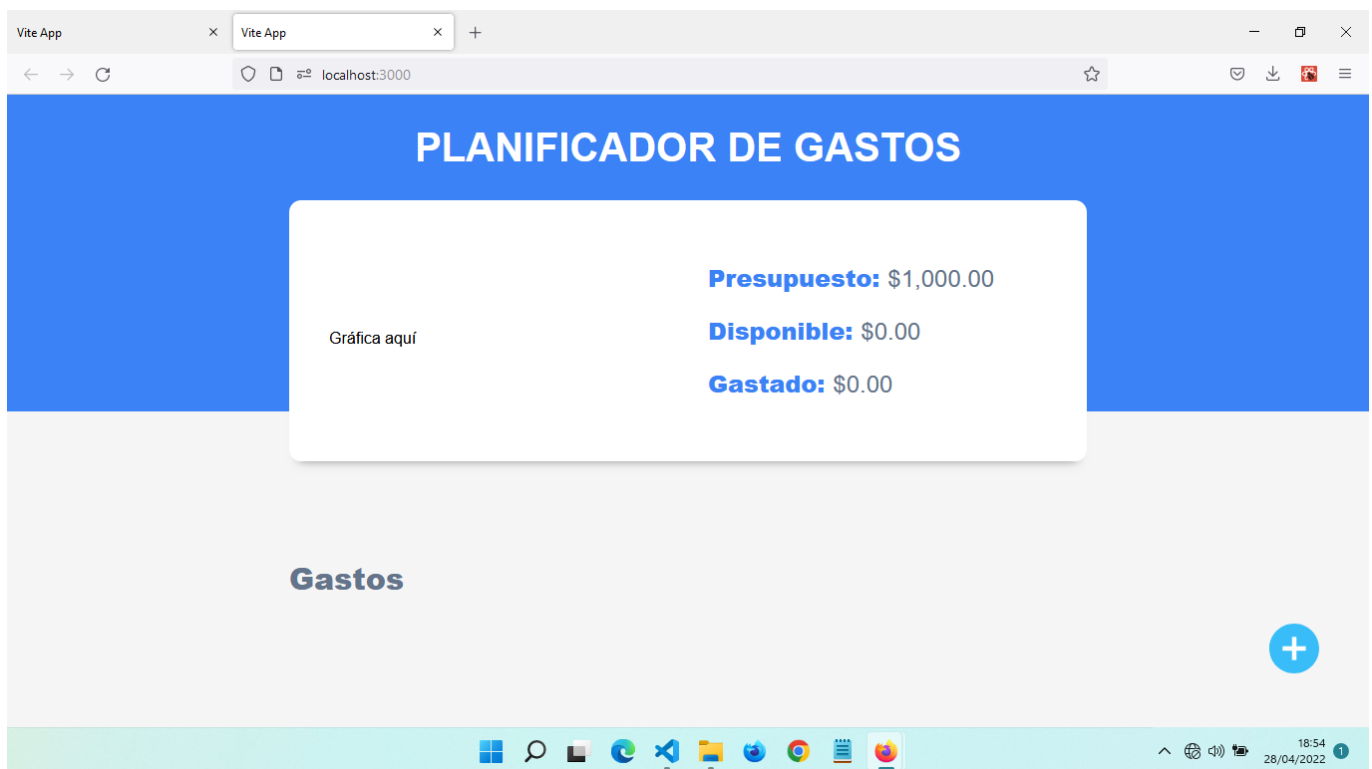
- Creo el componente ListadoGastos, le añado un párrafo para visualizarlo cuando lo importe en el main App.jsx

```
{isValidPresupuesto && (  
  <>  
    <main>  
      <ListadoGastos />  
    </main>  
    <div className="nuevo-gasto">  
      <img src={IconoNuevoGasto} alt="icono-gasto"  
        onClick={handleNuevoGasto}/>  
    </div>  
  
  </>  
)}
```

- Aparece en el lateral izquierdo



- Tiene la validación para que se muestre solo si el presupuesto es válido.
- Le falta la condición de : si no hay gastos aún que muestre no hay gastos o el gasto
- Agrego la clase al div de ListadoGastos "listado-gastos contenedor" y un h2 con Gastos
- Luce asi

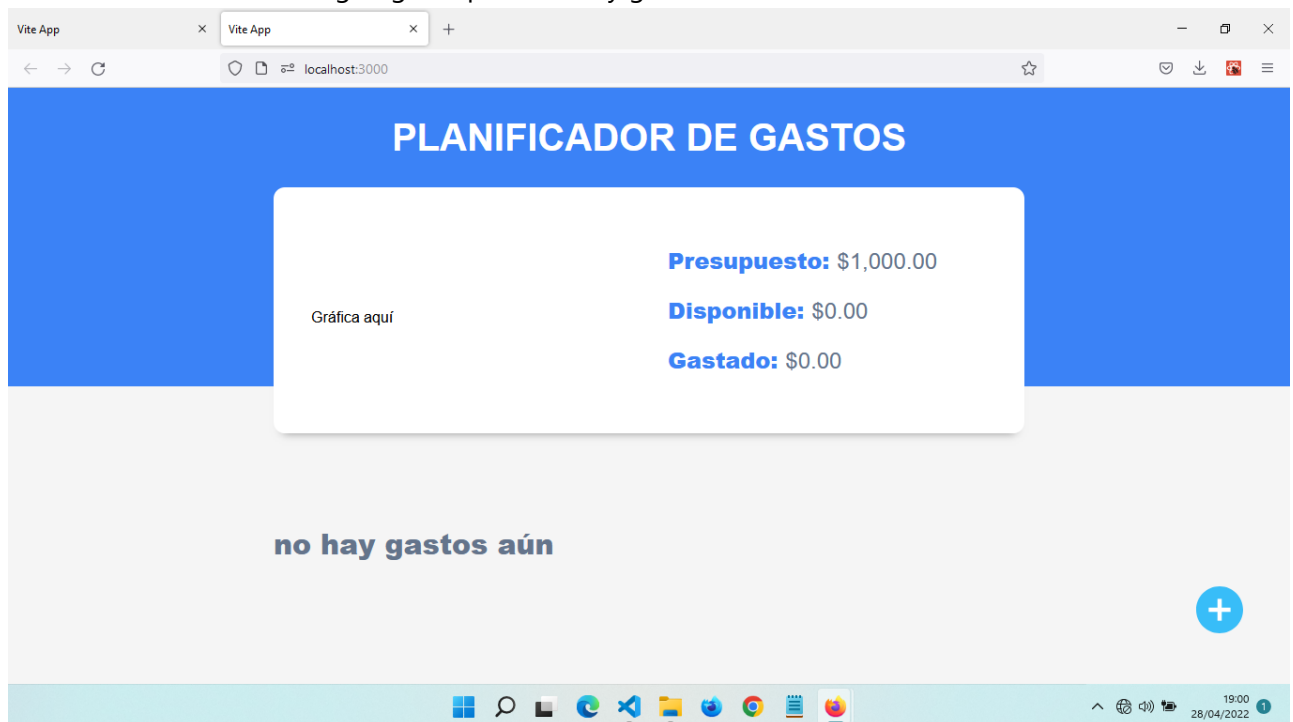


- Le paso via props el arreglo de gastos a ListadoGastos

- Le añado un ternario como validación para que muestre Gastos o no hay Gastos

```
const ListadoGastos = ({gastos}) => {  
  return (  
    <div className="listado-gastos contenedor">  
      <h2>{gastos.length ? "Gastos" : "no hay gastos aún"}</h2>  
    </div>  
  )  
}  
  
export default ListadoGastos
```


- Ahora si no introduzco ningun gasto pone no hay gastos



- Pero si introduzco un gasto aparece Gastos
- despues de este h2 con la validación voy a usar gastos.map para iterar y mostrar los diferentes componentes Gasto que todavía tengo que crear, en el caso de que los haya.

NOTA: recuerda que para que el return del callback esté implícito debe estar entre paréntesis. Si le pones llaves no renderizará el elemento

```
const ListadoGastos = ({gastos}) => {  
  return (  
    <div className="listado-gastos contenedor">  
      <h2>{gastos.length ? "Gastos" : "no hay gastos aún"}</h2>  
  
      {gastos.map(gasto=>(  
        <Gasto  
          key={gastos.id}  
          gasto={gasto}/>  
      ))}  
    </div>  
  )  
}
```

- Creo el componente Gasto con un párrafo para visualizar, lo importo en ListadoGastos
- Como mi objetivo es pasar la info del objeto de gastos, se lo paso via props y como debe tener un id único le paso el gasto.id
- Ahora si agrego un gasto lo renderiza en pantalla Captura71
- Bien, trabajando un poco el componente gasto le añado unas clases de css
- En el párrafo, cómo le pasé por via props gasto, puedo poner gasto.categoría
- Se puede usar desestructuración para no usar esta sintaxis de gasto.categoría y solo categoría

```
const Gasto = ({gasto}) => {
  const {categoria, nombre, cantidad, id} = gasto
  return (
    <div className="gasto sombra">
      <div className="contenido-gasto">
        <div className="descripcion-gasto">
          <p className="categoria">
            {categoria}
          </p>
          <p className="nombre-gasto">
            {nombre}
          </p>
        </div>
      </div>
    </div>
  )
}
```

- Creo una nueva variable en guardarGasto con la fecha que se genera al crear el objeto

```
const guardarGasto = gasto =>{
  gasto.id = generarId();
  gasto.fecha = Date.now()
  setGastos([...gastos, gasto])
  setAnimarModal(false)

  setTimeout(()=>{
    setModal(false)
  }, 500)
}
```

- El problema es que esta fecha no esta formateada, aparece en segundos desde el 1 de enero de 1970
- Le doy formato en la siguiente lección