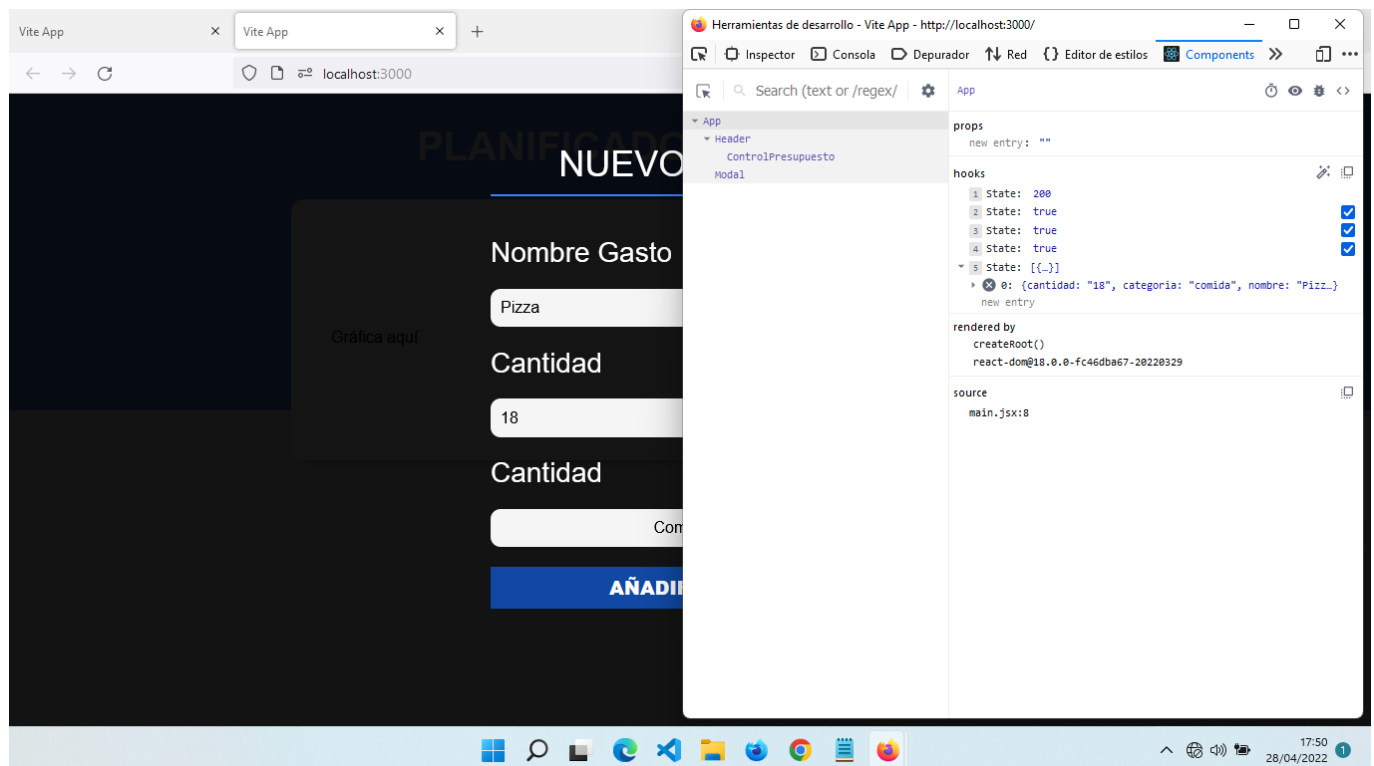


# Creando función para generar un id único

- Coloco la función de setGastos en guardarGasto y uso el spread para traerme al arreglo el state anterior y le agrego el objeto gasto que viene desde el Modal.

```
const guardarGasto = gasto =>{
  setGastos([...gastos, gasto])
}
```



- Ahora ocurren dos cosas: el objeto en el state no tiene id y el modal no se cierra ni tiene la transición de cerrar cuando agrego el gasto
- Primero lo del id.
- Puedo crear una carpeta de helpers con varias funciones que utilizaré en mi proyecto
- Llamo al archivo index.js así cuando lo importo no hay que poner el nombre.
- La función queda así
  - Math.random para el numero aleatorio de 0 a 1
  - toString para convertirlo a string con el valor 36 ?
  - substring con el valor de 2 para quitarle los dos primeros dígitos
  - Date now para tener la fecha actual y pasarla a string con toString

```
export const generarId = ()=>{
  const random = Math.random().toString(36).substring(2)
```

```
const fecha = Date.now().toString(36)
return random + fecha
}
```

- Importo la función en App.jsx, no necesito poner el nombre porque está cómo index
- Lleva llaves porque no es una export por default

```
import {generarId} from './helpers'
```

- Lo agrego a gasto en la función guardarGasto

```
const guardarGasto = gasto =>{
  gasto.id = generarId();
  setGastos([...gastos, gasto])
}
```

- Ahora puedo ver en components de devtools que en el state hay un id único dentro del objeto
- El modal se abre y se cierra con setModal y setAnimarModal.
- Para cerrarlo hay que llamar al setAnimarModal con un timeout y luego al setModal en guardarGastos como en ocultarModal

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

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