



Daremos continuación al proyecto de la práctica anterior con el funcionamiento del ejercicio, para ello crearemos un archivo llamado script.js que contendrá el siguiente código.

Primero invocamos todos los elementos

```
1  const contentFilas = document.querySelector("#filas-content")
2  const templateFilas = document.querySelector("#filas-template")
3
4  const contentDetalles = document.querySelector("#detalles-content")
5  const templateDetalles = document.querySelector("#detalles-template")
6
7  const btnGuardar = document.querySelector('#btn-guardar')
8  const btnCancelar = document.querySelector('#btn-cancelar')
9  const txtNombre = document.getElementById("txtNombre")
10 const txtCantidad = document.getElementById("txtCantidad")
11 const txtPrecio = document.getElementById("txtPrecio")
12 const fragment = document.createDocumentFragment()
```

```
14 // array que almacenara los productos
15 let productos = []
16
17 // evento para guardar los productos
18 btnGuardar.addEventListener('click', () => {
19     productos.push({
20         id: productos.length + 1,
21         name: txtNombre.value,
22         stock: txtCantidad.value,
23         price: txtPrecio.value
24     })
25
26     filasTabla()
27     mostrarDetalles()
28     vaciarControles()
29 });
30
31 btnCancelar.addEventListener('click', () => {
32     vaciarControles()
33 });
34
```



Nótese que estamos invocando funciones que aun no existen, pero las crearemos posteriormente.

```
35 function filasTabla(){
36     contentFilas.textContent = ''
37
38     //recorremos la lista de productos
39     productos.forEach(item => {
40         //clonamos la estructura de los nodos
41         const clone = templateFilas.content.cloneNode(true)
42         clone.querySelector('.id').textContent = item.id
43         clone.querySelector('.name').textContent = item.name
44         clone.querySelector('.stock').textContent = item.stock
45         clone.querySelector('.price').textContent = `$ ${item.price}`
46         clone.querySelector('.btn-remove').dataset.id=item.id
47         //agregamos el clon a nuestro fragmento
48         fragment.appendChild(clone)
49     })
50
51     //agregamos el fragmento al contenedor principal
52     contentFilas.appendChild(fragment)
53     agregarEventoRemoveProducto()
54 }
```

```
56 function mostrarDetalles(){
57     contentDetalles.textContent = ''
58
59     productos.forEach(item => {
60         const clone = templateDetalles.content.cloneNode(true)
61         clone.querySelector('#producto').textContent = item.name
62         clone.querySelector('#sumatoria').textContent = `$ ${parseInt(item.stock) * parseFloat(item.price)}`
63
64         fragment.appendChild(clone)
65     })
66
67     contentDetalles.appendChild(fragment)
68 }
69 }
```



```
70 // evento para remover el producto
71 function agregarEventoRemoverProducto(){
72     const buttons = document.querySelectorAll('.btn-remover')
73
74     buttons.forEach(btn => {
75         btn.addEventListener('click', (e) => {
76             let id = btn.dataset.id
77             removerProducto(id)
78             filasTabla()
79             mostrarDetalles()
80         })
81     })
82 }
```

```
84 // remover producto del array
85 function removerProducto(id){
86     productos = productos.filter(
87         pro => parseInt(pro.id) !== parseInt(id)
88     )
89 }
90
91 // vaciar los controles del modal
92 function vaciarControles(){
93     txtNombre.value = ''
94     txtCantidad.value = ''
95     txtPrecio.value = ''
96 }
97
98 // mostramos el resultado
99 filasTabla()
100 mostrarDetalles()
```