



# **Diseño y Programación de Software Multiplataforma**

Taller 1

## **ELABORADO POR:**

Martínez Montoya, Marlon David MM210981

Ventura Alvarenga, Roberto Alejandro VA210756

## **Docente:**

Karens Medrano

Facultad de Ingeniería / Escuela de Ingeniería Computación

Soyapango, 28/Julio 2024

Links repositorios ejercicios:

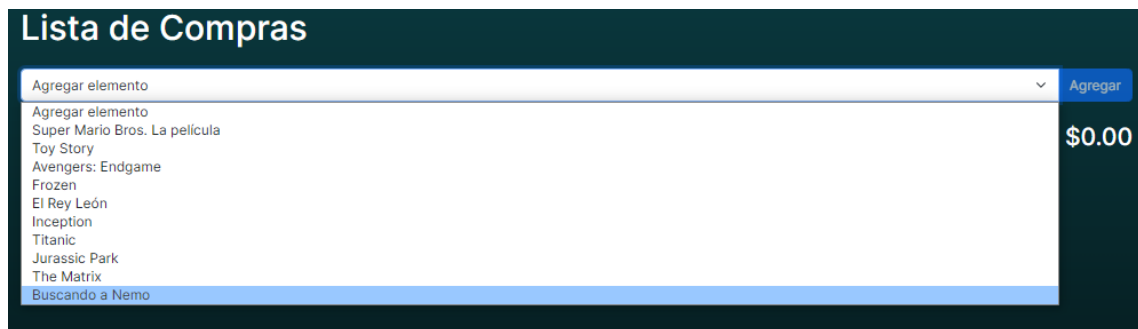
<https://github.com/VAALEJANDRO/Ejercicio1>

<https://github.com/VAALEJANDRO/Ejercicio2>

## Ejercicio 1:

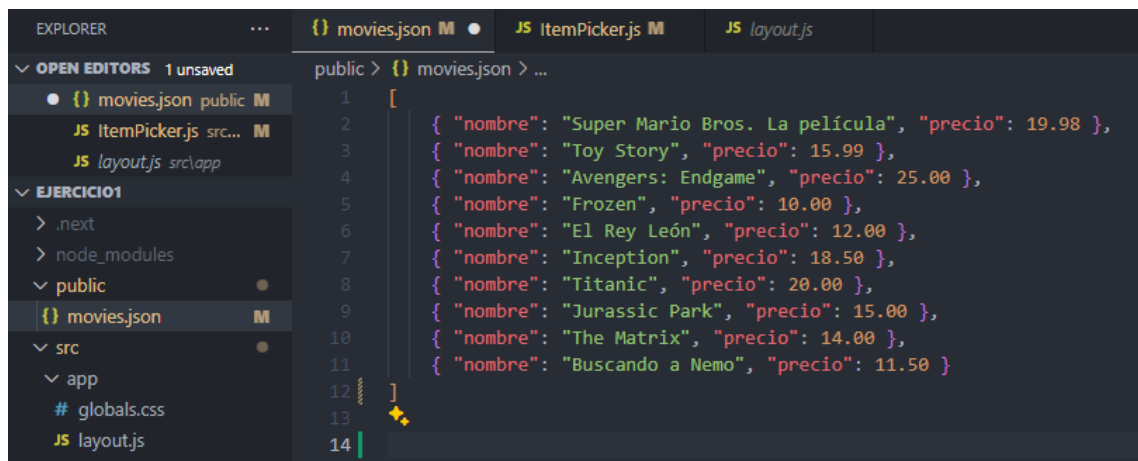
Elaborar una aplicación web para lista de compra de películas, para la aplicación web debe de considerar lo siguiente:

1. Para agregar elementos dentro de la lista estos deberán ser seleccionados desde el picker en la aplicación.



2. Los datos que mostrará el picker debe venir desde una lista json que contendrá 10 elementos y puede poseer la siguiente estructura de ejemplo {nombre:' Super Mario Bros. La película', precio:19.98}

Estructura del JSON:



Componente donde se extrae el contenido:

```
src > components > JS ItemPicker.js > ItemPicker
1  "use client"
2
3  import { useState, useEffect } from 'react';
4
5  export default function ItemPicker({ onAddItem }) {
6    const [items, setItems] = useState([]);
7    const [selectedItem, setSelectedItem] = useState('');
8
9    useEffect(() => {
10      fetch('/movies.json')
11        .then(response => response.json())
12        .then(data => setItems(data))
13        .catch(error => console.error('Error al cargar el JSON:', error));
14    }, []);
15
16    const handleAddItem = () => {
17      const item = items.find(i => i.nombre === selectedItem);
18      if (item) {
19        onAddItem(item);
20        setSelectedItem(''); // Limpia select luego de agregar item
21      }
22    };
23  };
24
```

3. Cuando el usuario agregue un elemento a la lista, este debe poder visualizarse en la lista con su respectivo precio.
4. Cada elemento de la lista debe poder eliminarse de la lista.
5. Cada elemento de lista se le debe poder asignar una cantidad por medio de input number

## Lista de Compras

Agregar

Avengers: Endgame  
\$25

-

1

+

X

Total: \$25.00

Generar Factura

6. El Total debe reflejar el total a pagar según la cantidad de elementos y su respectivo precio de forma global.
7. El cálculo del precio Total debe ser de forma automática sin necesidad que el usuario presione un botón.

*(Porcentaje alcanzado 100%)*

### Lista de Compras

Agregar

Avengers: Endgame \$25	- 2 + <span>X</span>
Frozen \$10	- 1 + <span>X</span>
Buscando a Nemo \$11.5	- 3 + <span>X</span>

Total: \$94.50

Generar Factura

8. Al darle click al botón de Generar Factura se genera un archivo JSON con la información

```
{} movies.json M JS ItemPicker.js M {} Factura.json X
C: > Users > USUARIO > Downloads > {} Factura.json > ...
1  {
2    "items": [
3      {
4        "nombre": "Avengers: Endgame",
5        "precio": 25,
6        "cantidad": 2,
7        "total": 50
8      },
9      {
10       "nombre": "Frozen",
11       "precio": 10,
12       "cantidad": 1,
13       "total": 10
14     },
15     {
16       "nombre": "Buscando a Nemo",
17       "precio": 11.5,
18       "cantidad": 3,
19       "total": 34.5
20     }
21   ],
22   "total": 94.5
23 }
```

## Ejercicio 2











Elaborar una aplicación web para Jaguar Sport, para la aplicación web debe de considerar lo siguiente:

1. En la página principal se debe presentar una grid con mínimos 10 productos deportivos pertenecientes a las siguientes categorías:

- Yoga y Pilates
- Futbol
- Mochilas deportivas
- Ciclismo
- Boxeo

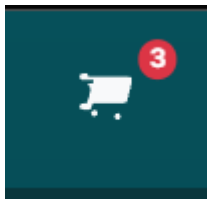
2. Cada producto presentado debe mostrar una fotografía, breve descripción y precio

### Jaguar Sport

 <p><b>Colchoneta de yoga</b> Colchoneta de yoga de alta calidad. \$35</p> <p>Registrar</p>	 <p><b>Pelota de pilates</b> Pelota de pilates para ejercicios. \$35</p> <p>Registrar</p>	 <p><b>Balon de futbol</b> Balón de futbol duradero. \$30</p> <p>Registrar</p>	 <p><b>Tacos de futbol</b> Botines de futbol cómodos. \$30</p> <p>Registrar</p>
 <p><b>Mochila deportiva</b> Mochila deportiva espaciosa. \$50</p> <p>Registrar</p>	 <p><b>Gym Bag</b> Bolso de gimnasio práctico. \$50</p> <p>Registrar</p>	 <p><b>Casco de ciclismo</b> Casco de ciclismo seguro. \$25</p> <p>Registrar</p>	 <p><b>Guantes de ciclismo</b> Guantes de ciclismo cómodos. \$25</p> <p>Registrar</p>
 <p><b>Guantes de boxeo</b> Guantes de boxeo profesionales. \$20</p> <p>Registrar</p>	 <p><b>Saco de boxeo</b> Saco de boxeo resistente. \$20</p> <p>Registrar</p>		

3. Cuando el usuario presione el botón agregar que esta junto a la imagen del producto debe de añadir el producto al carrito

Al agregar productos se almacenan en el carrito



4. Cada elemento del carrito debe poder eliminarse de la lista

5. Cada elemento del carrito se le debe poder asignar una cantidad por medio de input number.

6. El Total debe reflejar el total a pagar según la cantidad de elementos y su respectivo precio de forma global.

7. El cálculo del precio Total debe ser de forma automática sin necesidad que el usuario presione un botón

*(Porcentaje alcanzado 100%)*

