

# UNIVERSIDAD DON BOSCO



Materia: Diseño de Software Multiplataforma

Grupo Teórico: 01L

Tema Para Evaluar: Desafío Práctico #1

Docente: Karens Medrano

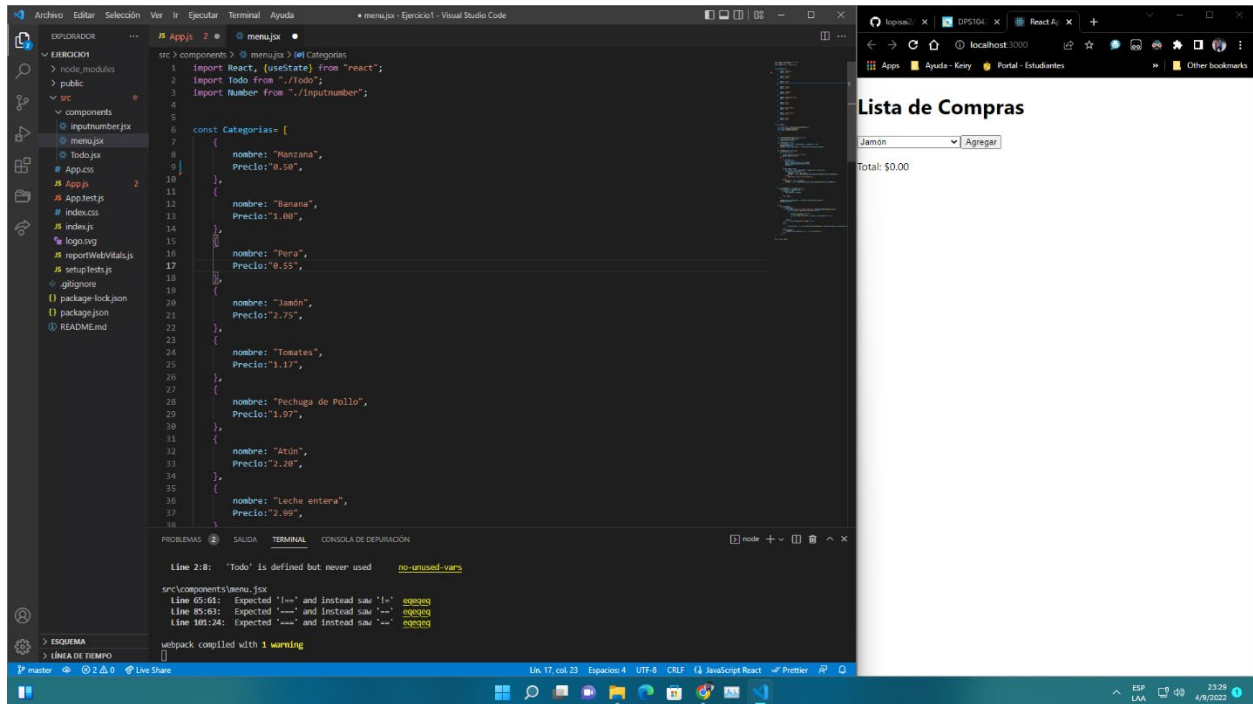
Integrantes:

- David Isaí Alfaro López AL201498
- Keila Jael Rivas Jiménez RJ202336

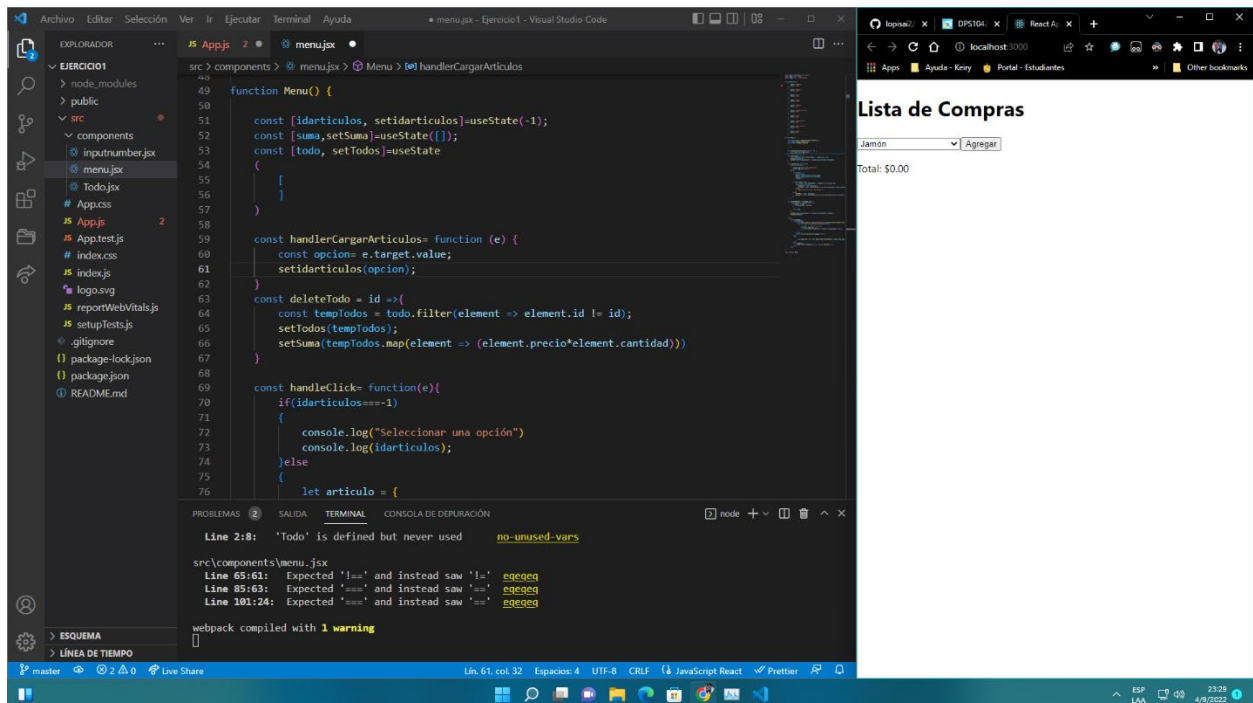
San Salvador, 04 de septiembre de 2022

## Proceso: Ejercicio #1

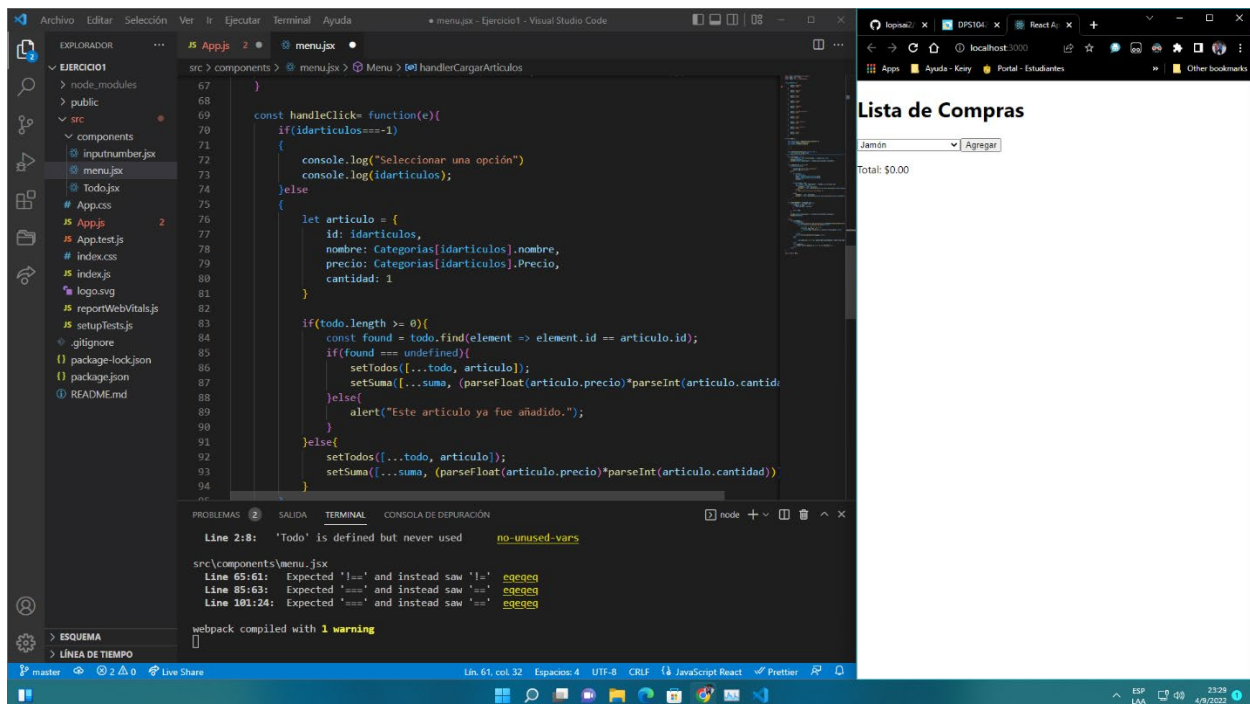
Paso 1: Se crea una constante que contenga los datos que serán cargados en el picker (Select) de forma dinámica.



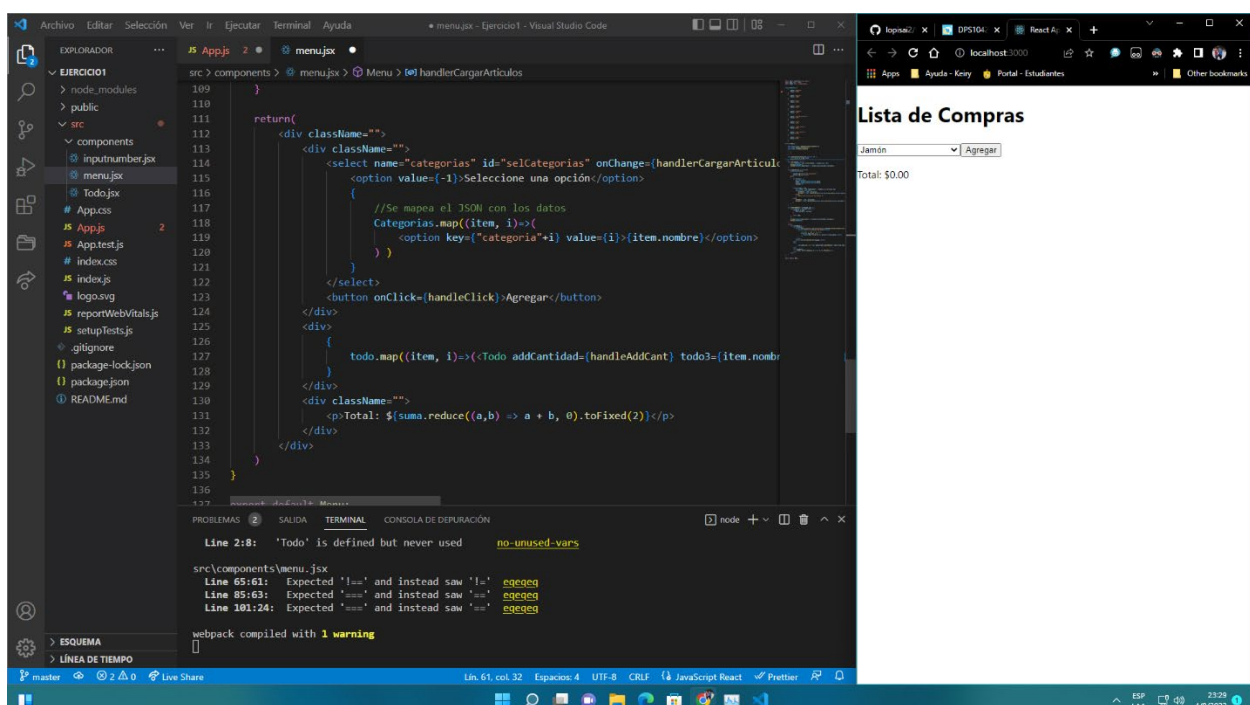
Paso 2: Se declaran los hooks que van a permitir actualizar las variables en tiempo real, así como también los llamados “handler” que servirán para cuando ocurran eventos en el select, en el botón de agregar o cuando cambie el input number.



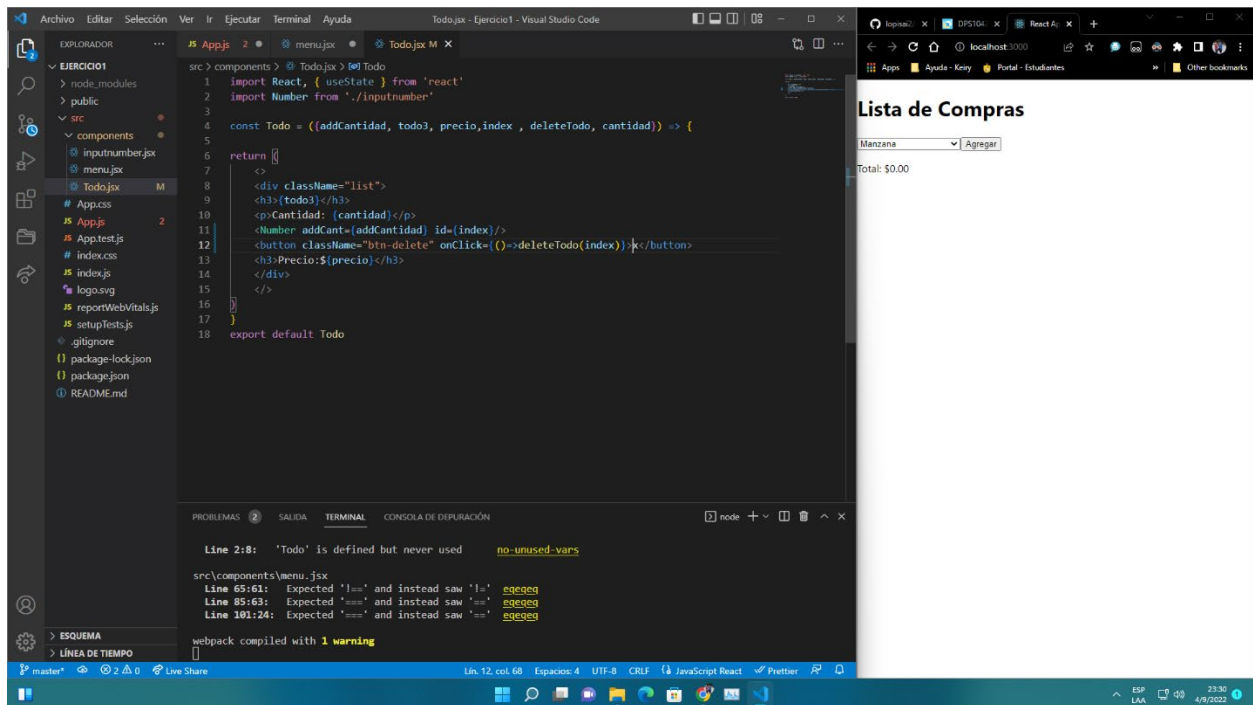
Paso 3: Se especifica los detalles de los handlers, los cuales permitirán crear la lista de compras cuando el usuario pulse el botón de agregar, el cual contendrá el nombre, el precio, el input number y el botón de eliminar.



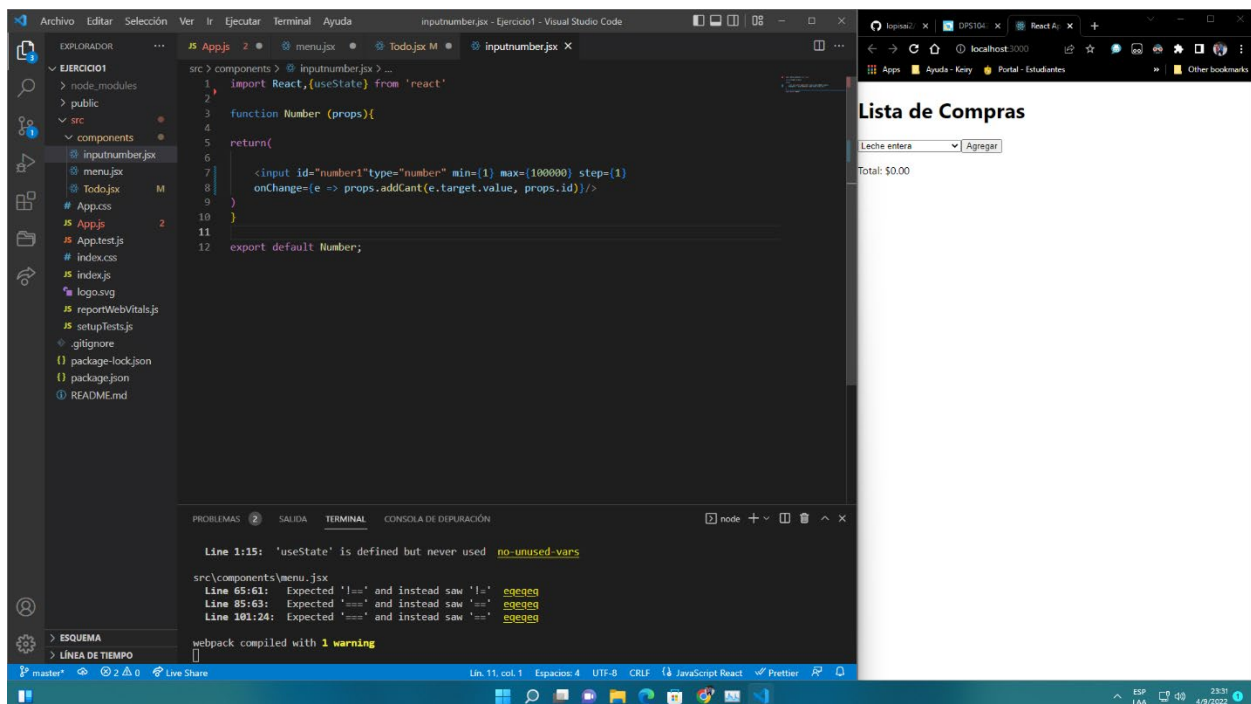
Paso 4: Se hace el return de la información, que contiene varios arreglos.map que se recorren para obtener información de las listas y el select, así como también muestra el total de la suma de la cantidad de productos por el precio.



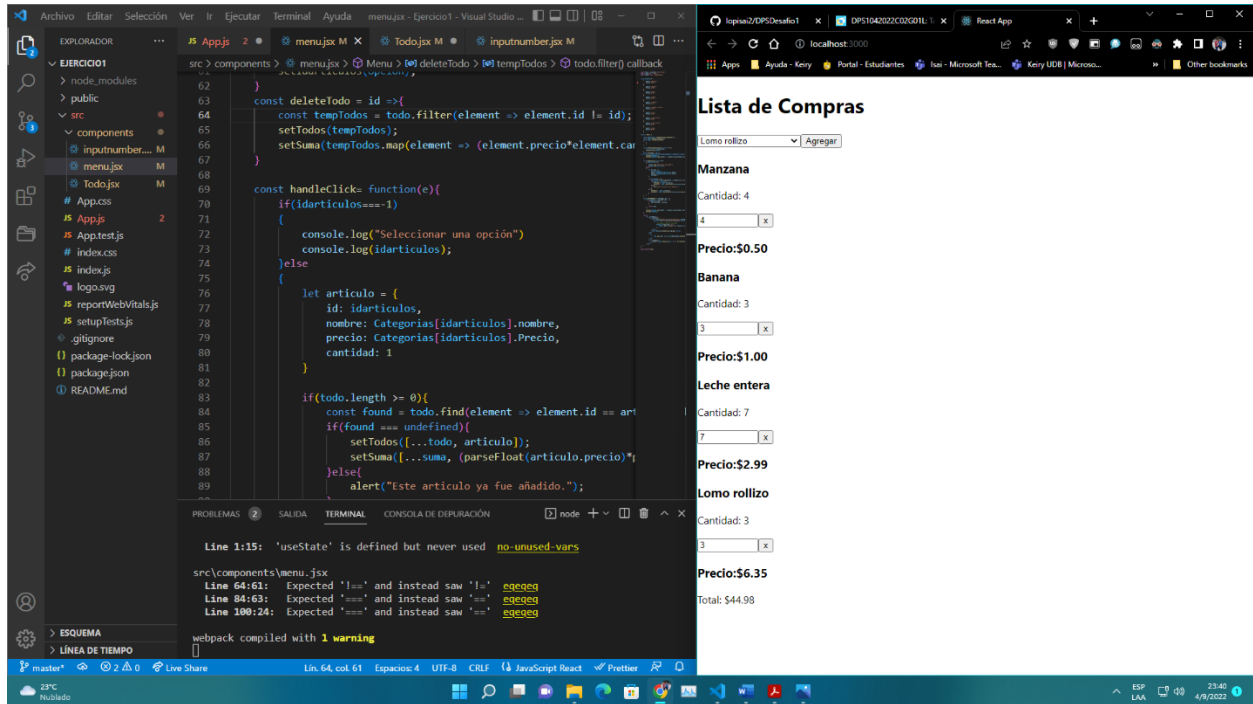
Paso 5: Se utiliza otro componente llamado “Todo” que se utiliza para forma la lista de compras dinámica



Paso 6: También se crea el componente inputnumber que se utiliza para cargar el input de forma dinámica junto con la lista



## Paso 7: Demostración de la lista de compras creada



The screenshot displays a web application running in a browser, showing a shopping list titled "Lista de Compras". The list includes items like Manzana, Banana, Leche entera, and Lomo rollizo, each with a quantity and price. The total is \$44.98. The background shows the VS Code editor with the menu.js file open, displaying the logic for adding and deleting items from the shopping list.

**Lista de Compras**

Lomo rollizo  Agregar

**Manzana**  
Cantidad: 4  
4   
Precio:\$0.50

**Banana**  
Cantidad: 3  
3   
Precio:\$1.00

**Leche entera**  
Cantidad: 7  
7   
Precio:\$2.99

**Lomo rollizo**  
Cantidad: 3  
3   
Precio:\$6.35

Total: \$44.98

**VS Code Editor (menu.js):**

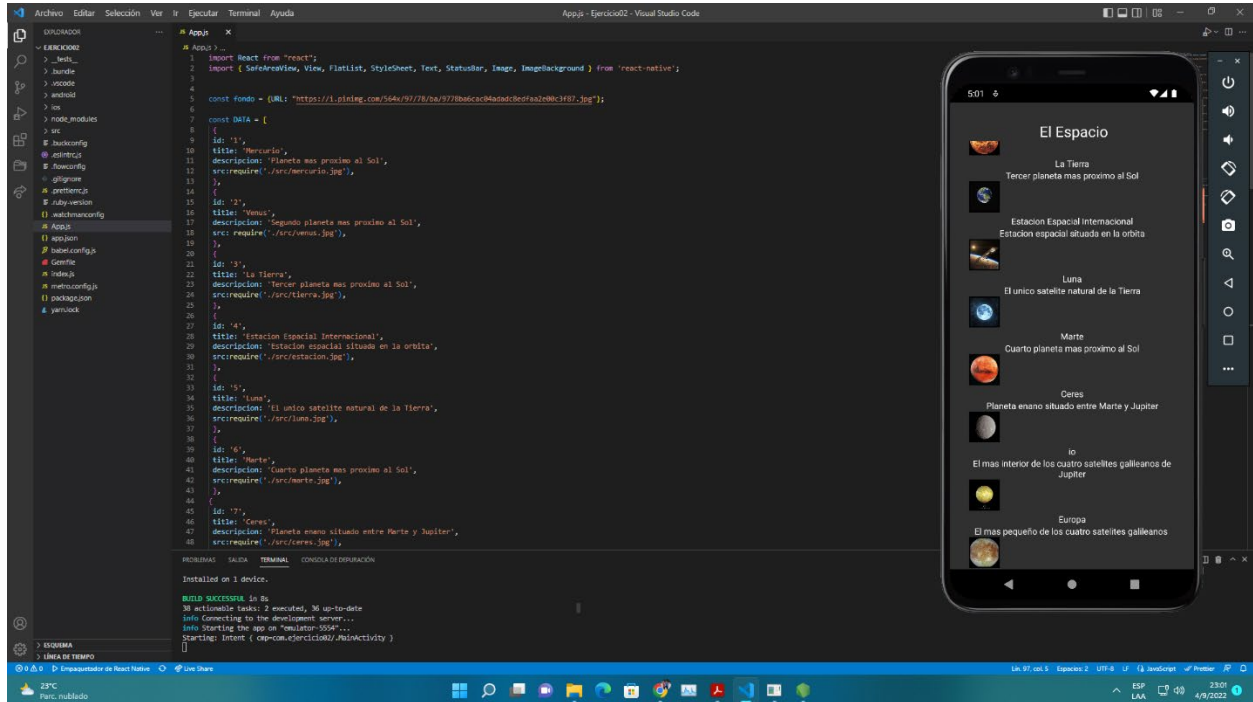
```
src > components > menu.js > Menu > deleteTodo > tempTodos > todo.filter() callback
62 }
63 const deleteTodo = id => {
64   const tempTodos = todo.filter(element => element.id !== id);
65   setTodos(tempTodos);
66   setSuma(tempTodos.map(element => (element.precio*element.cantidad)));
67 }
68
69 const handleClick = function(e) {
70   if(idarticulos===-1)
71   {
72     console.log("Seleccionar una opción")
73     console.log(idarticulos);
74   }else
75   {
76     let articulo = {
77       id: idarticulos,
78       nombre: Categorias[idarticulos].nombre,
79       precio: Categorias[idarticulos].Precio,
80       cantidad: 1
81     }
82
83     if(tempTodos.length >= 0){
84       const found = tempTodos.find(element => element.id === articulo.id);
85       if(found === undefined){
86         setTodos([...tempTodos, articulo]);
87         setSuma([...suma, (parseFloat(articulo.precio)*1)]);
88       }else{
89         alert("Este articulo ya fue añadido.");
90       }
91     }
92   }
93 }
```

**Terminal:**

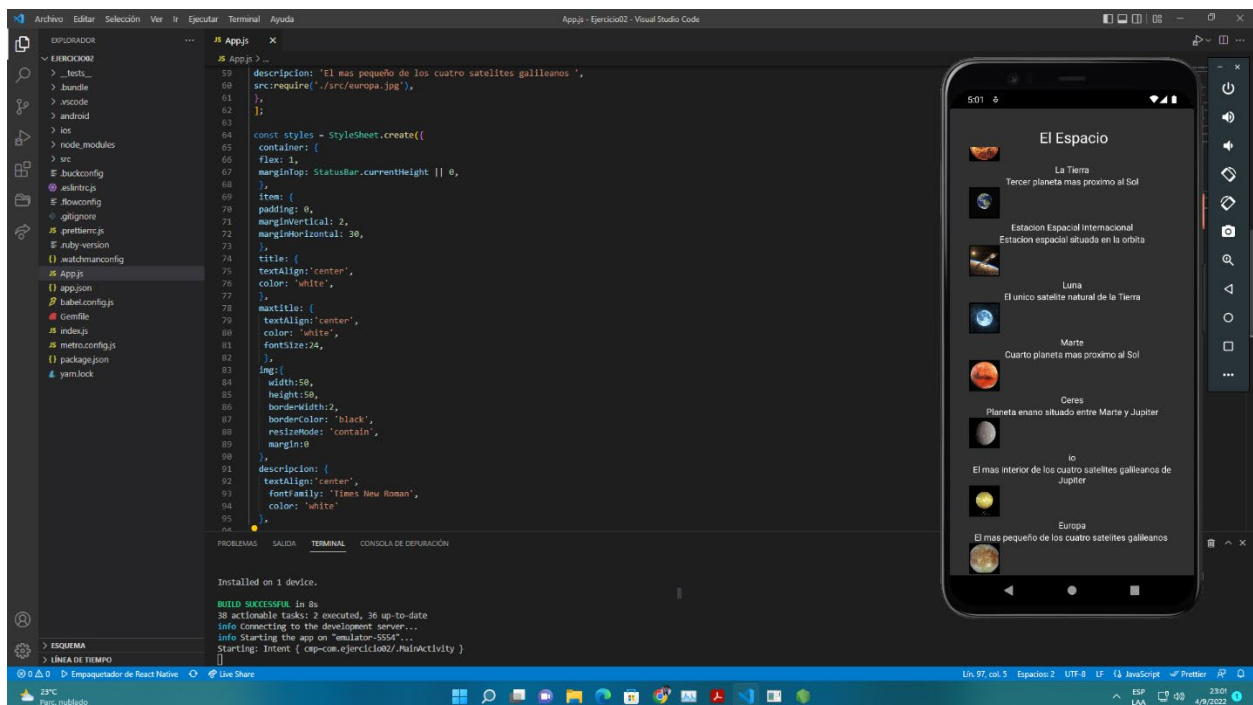
```
Line 1:15: 'useState' is defined but never used no-unused-vars
src\components\menu.js
Line 64:61: Expected '===' and instead saw '!=' eeggeg
Line 84:63: Expected '===' and instead saw '!=' eeggeg
Line 100:24: Expected '===' and instead saw '!=' eeggeg
webpack compiled with 1 warning
```

## Proceso: Ejercicio #2

Paso 1: Se crea un arreglo de datos que contendrá la información de los planetas en el espacio y su respectiva imagen



Paso 2: Se definen los estilos que tendrá la aplicación móvil





Paso 3: Se declaran constates de item el cual contiene información del arreglo DATA, y luego se realiza un renderItem que va a cargar los datos en pantalla. Y se carga finalmente la lista en con una Flatlist

