Introducción a la sección

A partir de la sección **6.2** hasta la **6.23**, se construirá una aplicación web de ejemplo para gestionar un carrito de compras, compuesta por dos páginas distintas. Además, se utilizará **sessionStorage** para asegurar que los cambios realizados en la aplicación no se pierdan al actualizar la página.

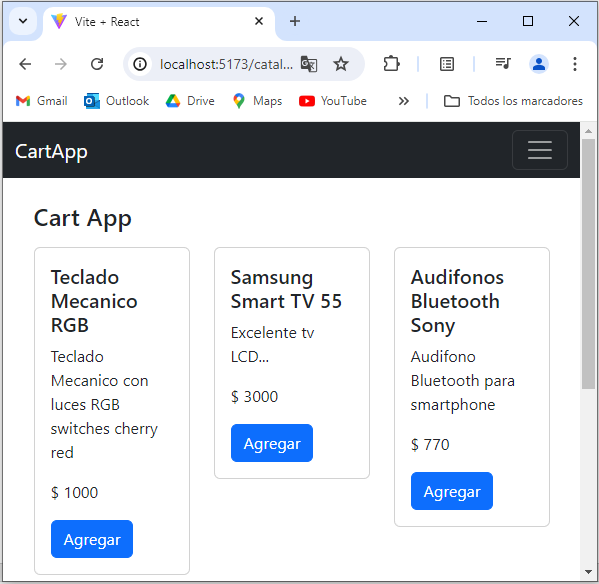
En etapas posteriores, la aplicación contará con una base de datos (MySQL) y el servidor se construirá con Spring Boot.

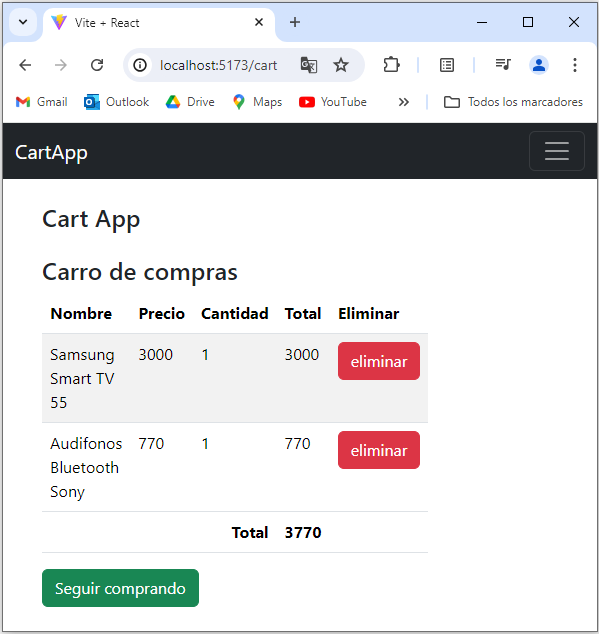
CartApp

Además del uso de Bootstrap para los estilos, las propiedades y funciones en los componentes, y la implementación de los hooks **useState** y **useEffect**; esta aplicación web va a manejar la lógica de un carrito de compras de la siguiente manera:

* **Lista de productos**
  + Aunque inicialmente se utiliza una lista de productos de ejemplo, estos datos podrían provenir de una base de datos a través del backend de Spring.
  + Los productos se pueden agregar al carrito desde esta lista pulsando un botón.
* **Visualización del carrito**
  + Al acceder al carrito por primera vez, si no hay productos, se mostrará un mensaje indicando que el carrito está vacío.
  + Al agregar un producto, lógicamente este aparecerá en el carrito.
  + La persistencia estará asegurada con **sessionStorage**, de manera que al actualizar la página, los productos en el carrito se mantendrán.
* **Funcionalidades del carrito**
  + **Agregar Productos:** Al agregar un producto ya existente en el carrito, este no se duplicará, sino que se actualizará la cantidad y se recalculará el precio total (precio por cantidad).
  + **Eliminar Productos:** Se podrá eliminar un producto del carrito pulsando un botón, lo cual recalculará automáticamente el total.
* **Navegación**
  + Se implementará un menú de navegación que permita moverse entre el catálogo de productos y el carrito de compras utilizando **react-router-dom**.
  + Desde el carrito, se podrá hacer clic en un botón para volver al catálogo de productos.
* **Manejo del estado con useReducer**
  + Se implementará el hook **useReducer** para centralizar la gestión del estado del carrito en una única función reductora. Esto va a permitir manipular el carrito de manera eficiente y organizada con diferentes acciones (agregar, eliminar, actualizar).

En resumen, esta aplicación de un carrito de compras con un catálogo completamente funcional en React, junto con la aplicación de **useReducer** y **React Router**, permitirá aprender nuevas técnicas además de las ya aplicadas en la aplicación anterior.



****