Agregando productos y actualizando el estado del carrito de compras

En esta sección, se continuará implementando la funcionalidad para agregar un producto al carrito de compras, de tal manera que los datos del producto se pasen de componente hijo (**ProductCartView**) a componente padre (**CartApp**) y se actualice el estado del carrito.

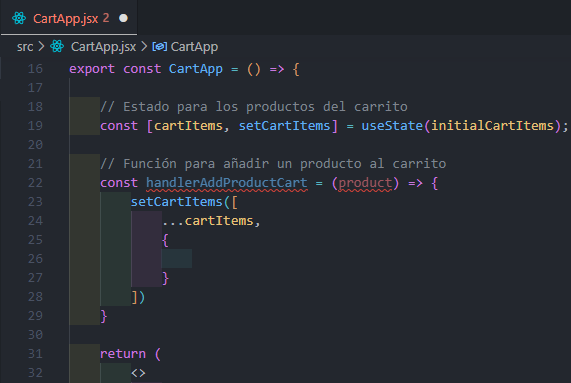
Pasos 34 al 38

Sigue estos pasos:

Paso 34: Definir una función para añadir un producto al carrito

Para manejar el evento de agregar un producto al carrito, debemos definir una función en el componente principal (**CartApp.jsx**) que recibirá el producto y actualizará el estado del carrito. Esta función se llamará **handlerAddProductCart** y será una función de flecha que recibirá el producto como argumento.

Luego se actualiza el estado **cartItems** (mediante la función **setCartItems**) con el nuevo producto. Se utiliza el operador spread (**...**) para clonar los elementos actuales del carrito y agregar el nuevo producto (por el momento como un objeto vacío) sin perder los existentes.

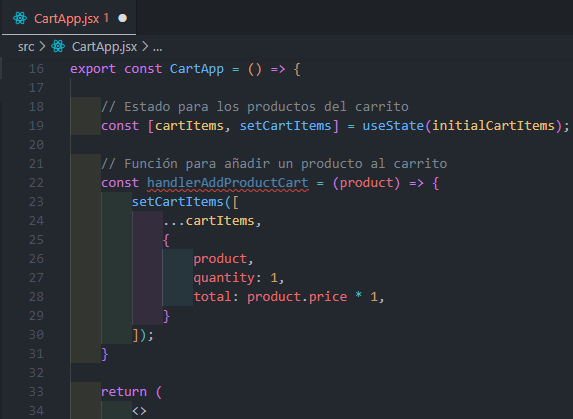


Paso 35: Agregar atributos al nuevo objeto que contendrá el nuevo ítem en el carrito

Cada ítem en el carrito de compras tiene los atributos **product**, **quantity** y **total**, similares a los que fueron definidos en el arreglo **initialCartItems**. Se procede a definir estos atributos en el nuevo objeto que se agregará al carrito.

Dentro de la función **handlerAddProductCart**, se crea un nuevo objeto con los siguientes atributos:

* **product:** el valor de **product** es el objeto del producto que se pasa como argumento.
* **quantity:** inicialmente se establece en **1**.
* **total:** es el resultado de **product.price** multiplicado por **1** (por el momento).

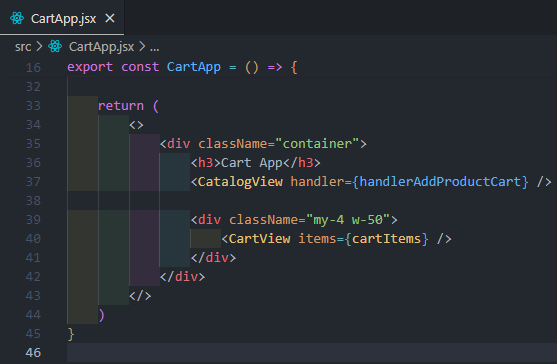


De esta forma, se está agregando un ítem de manera inmutable. Se utiliza el operador spread (**...cartItems**) para expandir los datos existentes y crear un nuevo arreglo, al que se agrega el nuevo ítem seleccionado cuando se hace clic en el botón **"Agregar"**.

Paso 36: Pasar la función handlerAddProductCart al componente hijo CatalogView

Ahora se necesita pasar la función **handlerAddProductCart** al componente hijo **CatalogView**, para que pueda ser utilizada por el componente **ProductCardView**, el cual es hijo de **CatalogView**.

En el componente principal **CartApp**, pasa la función como una prop llamada **handler** al renderizar el componente **CatalogView**. Esta prop **handler** se establece como la invocación de la función **handlerAddProductCart**, donde se emite o recibe el producto que proviene del componente hijo. Se puede simplificar la función de flecha pasada como prop al omitir el parámetro en la función de flecha y simplemente pasar **handlerAddProductCart**.



Paso 37: Pasar el valor de la propiedad handler al componente ProductCart

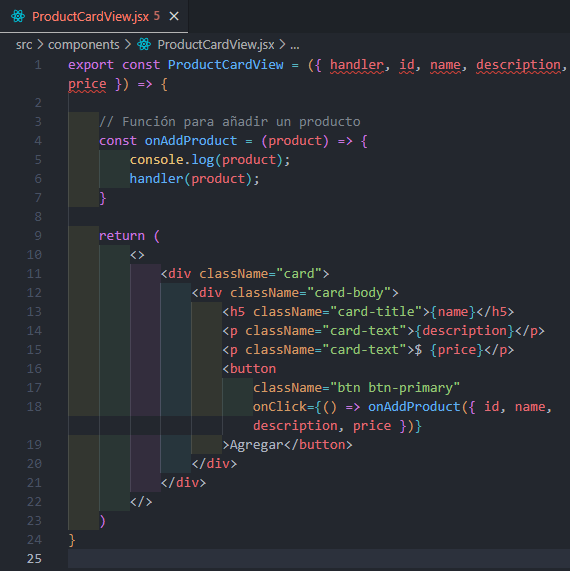
La propiedad **handler** se recibe en el componente **CatalogView** y debe ser pasada al componente hijo **ProductCardView**, donde se maneja el evento del botón para agregar productos al carrito de compras.

En el componente **CatalogView**, se define la propiedad **handler** en el renderizado de **ProductCardView**, estableciendo su valor igual al valor de la prop **handler** recibida en **CatalogView**. De esta manera, se pasa la función **handler** al componente hijo para que pueda ser utilizada en la lógica del botón de agregar producto.



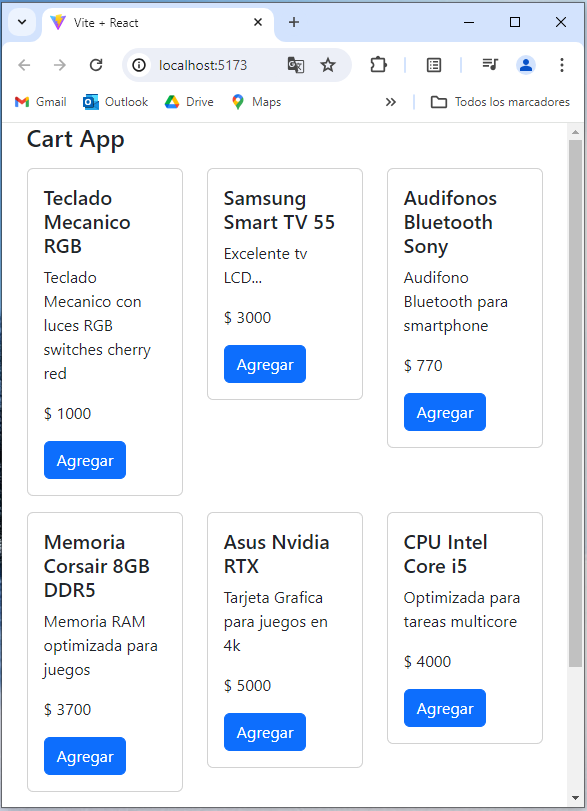
Paso 38: Invocar la función handler en el componente hijo ProductCardView y realizar una prueba en el navegador

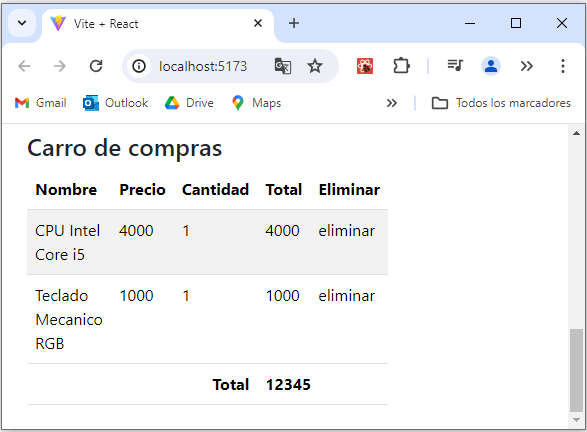
En el componente **ProductCardView**, se debe invocar la función **handler** que se recibe como propiedad (se define en las props del componente). Esta función se utilizará en la invocación de la función **onAddProduct**, donde su argumento será igual al objeto **product**.



De esta forma, al hacer clic en el botón **"Agregar"**, se invoca la función **onAddProduct**, que a su vez llama a la función **handler** pasando el objeto **product**. Esto permite que la función **handler** definida en el componente padre **CatalogView** se ejecute, actualizando así el estado del carrito de compras en el componente principal **CartApp**.

En el navegador, se puede agregar productos al carrito de compras haciendo clic en el botón **"Agregar"**.





Actualmente, al agregar un producto que ya está en el carrito, no se actualiza la cantidad, lo que resulta en una duplicación de elementos en el carrito. Esto no solo conduce a un comportamiento inesperado, sino que también genera un error en la consola debido a las claves duplicadas en el estado del carrito (atributo **key**).

