Hook useReducer - despachando acciones

En esta sección, se realizará la migración de la gestión del estado del carrito de compras, pasando de utilizar **useState** a **useReducer**. Esto implica centralizar todas las acciones definidas en el reducer **itemsReducer** en un único punto de la aplicación.

Pasos 64 al 67

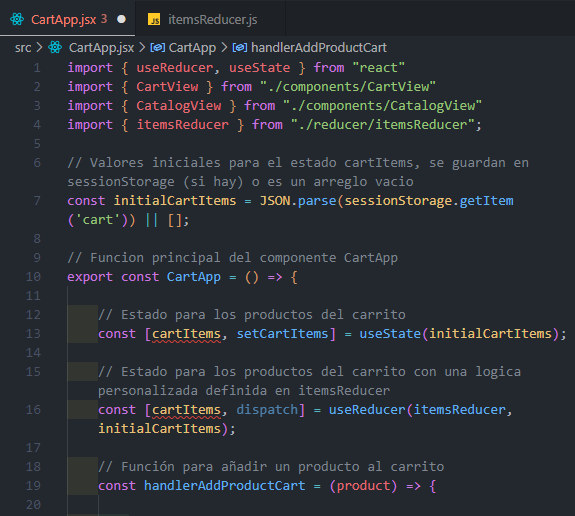
Sigue estos pasos:

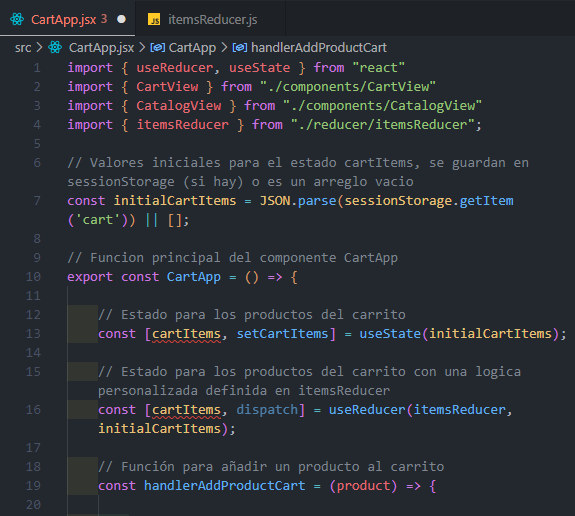
Paso 64: Importar useReducer en el componente principal

En el componente **CartApp**, se utilizará **useReducer** para manejar el estado de los productos en el carrito. Primero importa **useReducer** de React, junto con la función reductora **itemsReducer** que fue definida previamente.

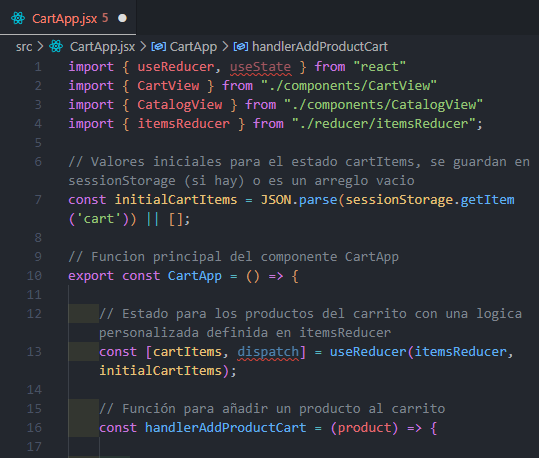
Recuerda que los valores iniciales para el estado de **cartItems** se obtienen del **sessionStorage** o, si no hay nada en el **sessionStorage**, se inicializa como un arreglo vacío.

A continuación, se utiliza **useReducer** para manejar el estado de **cartItems**. Esto va a permitir modificar el estado utilizando un **dispatch** con una acción. Asimismo, se pasa **itemsReducer**, la función reductora, como el primer argumento, e **initialCartItems** como el estado inicial.



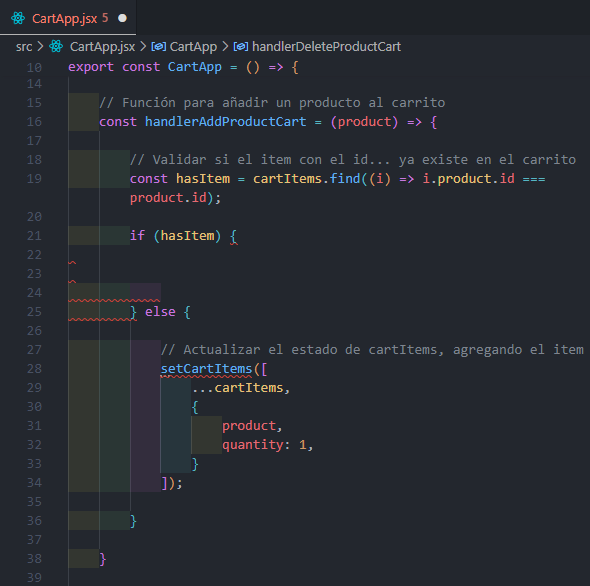


Se elimina la declaración de **cartItems** utilizando **useState**, ya que ahora se maneja exclusivamente con **useReducer**.

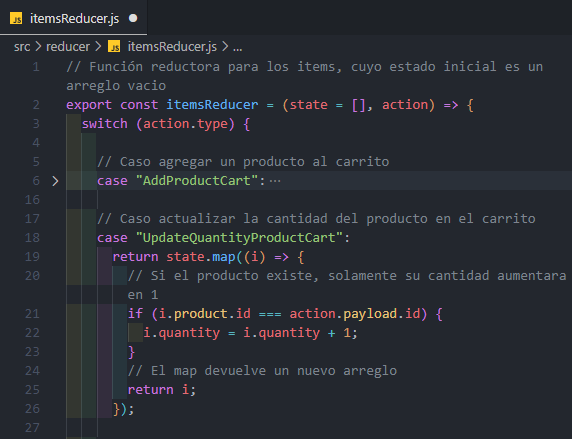


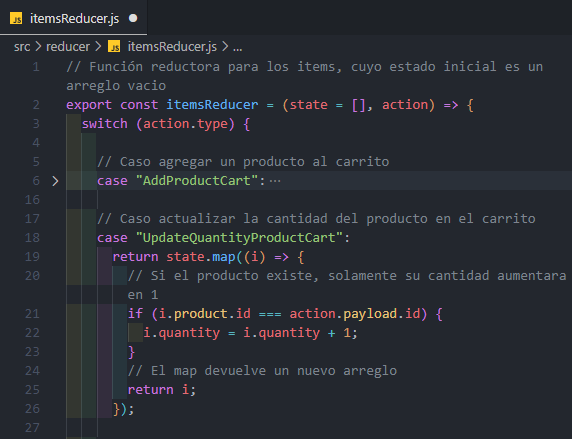
Paso 65: Actualización de la cantidad del producto en el carrito

En la función **handlerAddProductCart**, dentro de la condición que validaba si el ítem existía en el carrito (**hasItem**), elimina los comentarios y la lógica que utilizaba **setCartItems** para actualizar el estado del carrito. En su lugar, se utilizará un **dispatch** para enviar una acción que indique la actualización de la cantidad del producto en **1**.

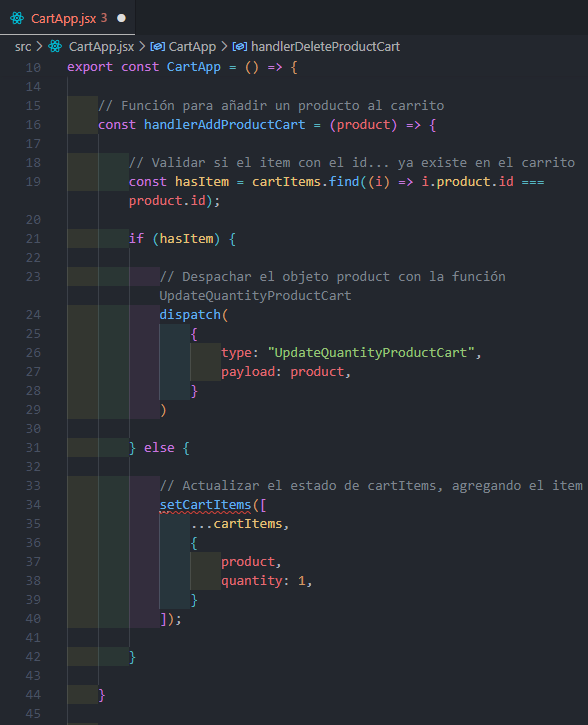


Recuerda que en **itemsReducer**, se define la acción **UpdateQuantityProductCart** para incrementar la cantidad del producto en **1**. Se recibe la acción que contiene el tipo (**type**) y el **payload**, que en este caso es el objeto producto (**product**). Dentro del **switch**, se identifica la acción por su tipo y se utiliza **action.payload** para obtener el objeto producto que se va a actualizar.



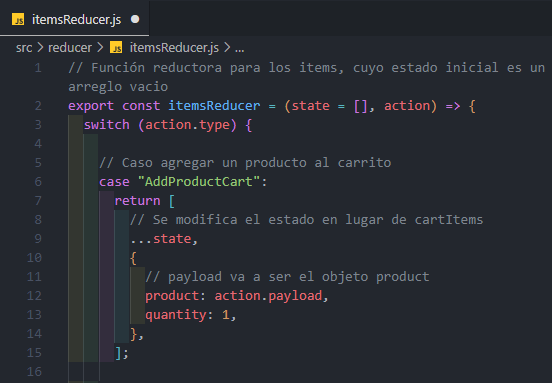


Al utilizar **dispatch** en lugar de **setCartItems**, se envía una acción que contiene el tipo **"UpdateQuantityProductCart"** y el **payload**, el objeto **product** que se va a actualizar. En **itemsReducer**, se maneja esta acción para incrementar la cantidad del producto en el carrito.



Paso 66: Agregar un producto al carrito

En **itemsReducer**, cuando se recibe la acción de tipo **"AddProductCart"**, se agrega el nuevo producto al estado del carrito. El **payload** de la acción contiene el producto que se va a agregar.



En el componente **CartApp**, la lógica para agregar un producto al carrito se encuentra en la función **handlerAddProductCart**. Si el producto ya existe en el carrito, se utiliza **dispatch** para enviar una acción de tipo **"UpdateQuantityProductCart"** y actualizar la cantidad del producto. De lo contrario, se envía una acción de tipo **"AddProductCart"** para agregar el producto al carrito.

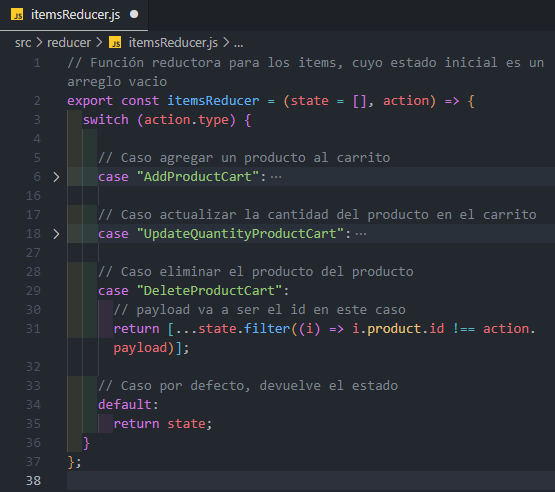


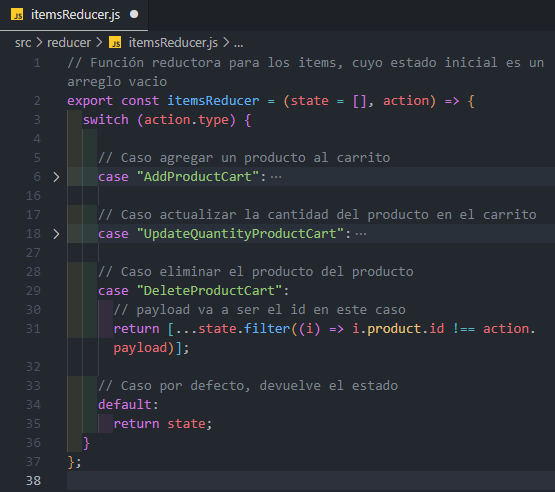


Al utilizar **dispatch** para manejar las acciones en lugar de **setCartItems**, se mantiene la lógica centralizada y coherente en **itemsReducer**, lo que facilita la gestión del estado del carrito.

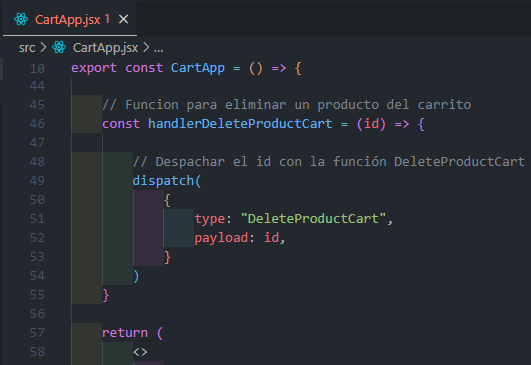
Paso 67: Eliminar un producto al carrito

En **itemsReducer**, cuando se recibe la acción de tipo **"DeleteProductCart"**, se filtra el estado del carrito para eliminar el producto con el **id** correspondiente al recibido en el **payload**.





Al utilizar **useReducer**, también se simplifica el proceso de eliminar un producto del carrito. En el caso de la función **handlerDeleteProductCart**, que se encarga de eliminar un producto, simplemente se encarga de enviar el **id** del producto al **dispatch** con la acción **"DeleteProductCart"**. Esto se realiza directamente en el componente **CartApp**.



De esta manera, al usar **useReducer**, se mantiene la lógica centralizada y por ende simplifica la gestión del estado del carrito de compras en un único punto de la aplicación.