Componente Cart Routes

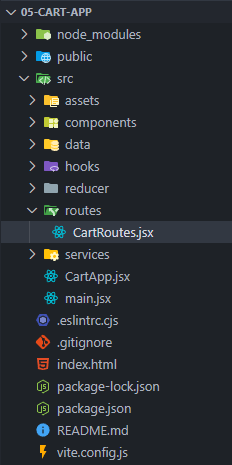
El objetivo de esta sección es finalizar el desarrollo de la aplicación de tipo carrito de compras: **CartApp**, se busca separar las rutas del componente principal en un nuevo componente independiente para facilitar la comprensión del flujo de navegación en la aplicación.

Pasos 94 al 96

Sigue estos pasos:

Paso 94: Definir un nuevo componente para las rutas

Primero, crea una nueva carpeta llamada **"routes"** dentro del directorio **"src"** para organizar las rutas. Luego, en esa carpeta, crea un nuevo archivo llamado **"CartRoutes.jsx"** donde se definirán las rutas.



En el archivo **"CartApp"**, corta todo el código relacionado con las rutas, es decir, las etiquetas **<Routes></Routes>** y todo su contenido anidado.



En el componente **CartRoutes** crea una función de flecha exportable con el mismo nombre del componente. Dentro de esta función (en el **return**), pega todo el código que ha sido cortado.

El componente **"CartRoutes"** tiene un elemento padre **<Routes>**, por lo cual se puede dejarse sin fragmentos. Además, importa los componentes necesarios como **"Routes"**, **"Route"**, **"CatalogView"**, **"CartView"** y **"Navigate"**.



Paso 95: Asignar valores a las propiedades del componente CartRoutes

Asigna valores a las propiedades del componente **"CartRoutes"** para que pueda recibir y utilizar las funciones y datos necesarios. Primero, pasa las funciones **"handlerAddProductCart"** y **"handlerDeleteProductCart"**, así como los datos **"cartItems"**, como propiedades desestructuradas al componente **"CartRoutes"**.

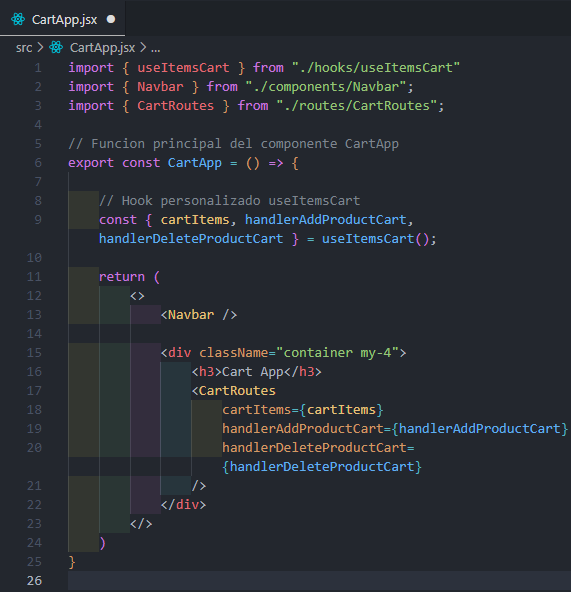


Dentro de **"CartApp"**, importa y renderiza el componente **"CartRoutes"**. Al hacerlo, se pasan las propiedades **"cartItems"**, **"handlerAddProductCart"** y **"handlerDeleteProductCart"** con los valores correspondientes que se obtienen del hook personalizado **"useItemsCart"**.



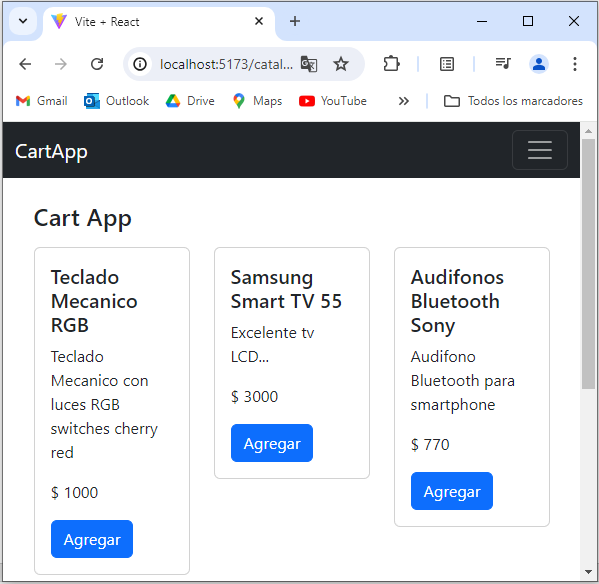


Finalmente, se eliminan todos los **imports** que no se están utilizando en **"CartApp"** para mantener el código limpio y organizado.

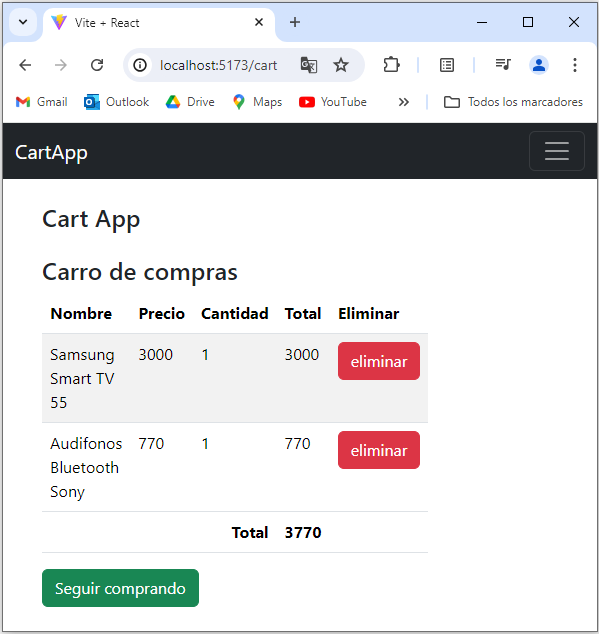


Paso 96: Realizar la prueba final

En primer lugar, el componente principal, **CartApp**, contiene el componente **Navbar**, que proporciona la barra de navegación para la aplicación. Luego, el título del contenedor principal establece el contexto general de la aplicación.



Por otro lado, **CartRoutes** se encarga de definir las rutas de navegación utilizando React Router DOM. Este componente asigna las URL a los componentes correspondientes, como **CatalogView** y **CartView**, y se asegura de cargar el componente adecuado según la URL actual.

****

En resumen, cada componente desempeña un papel crucial en la creación de una experiencia fluida de navegación y compra para el usuario. Utilizando React Router DOM, se han logrado establecer las reglas de navegación y asignar los componentes a URL específicas.

Con esto, se ha completado la primera fase del desarrollo de **CartApp**.