Eliminar ítems

El objetivo de esta sección es finalizar el desarrollo de la aplicación de tipo factura: **InvoiceApp**, se busca implementar un botón que permita eliminar un ítem de la lista de ítems al hacer clic en él y a su vez va a modificar el precio total.

Pasos 116 al 120

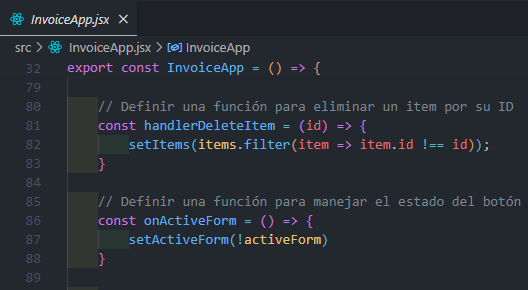
Sigue estos pasos:

Paso 116: Definir una función para eliminar un ítem

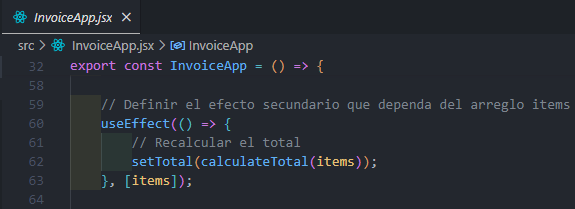
Procede a definir una función para eliminar un ítem en el componente principal **InvoiceApp**, similar a cómo se define la función para agregar un ítem. La función de eliminación cambiará el estado de **items** de manera segura.

Define la función **handlerDeleteItem** utilizando una función de flecha, que acepta un argumento **id** para identificar el ítem a eliminar. La función actualiza el estado **items** (con **setItems**) utilizando **filter** para crear un nuevo arreglo que excluya el ítem con el **id** especificado.

Recuerda que el método **filter** recorre cada item y solo incluye aquellos cuyo **id** no coincide con el ID proporcionado.

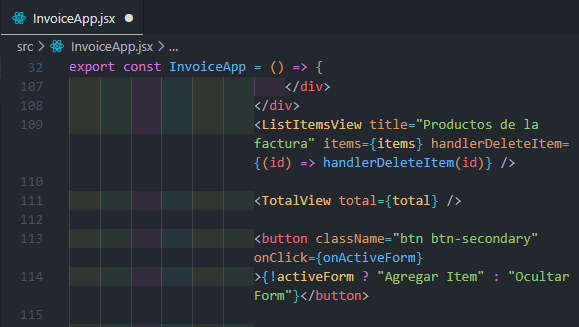


La lógica para calcular el total se mantiene dentro de un **useEffect** que depende del estado **items** (se ha definido un arreglo vacío como su valor por defecto).



Paso 117: Definir una propiedad en el renderizado del componente ListItemsView

En el renderizado del componente **ListItemsView**, se define la propiedad **handlerDeleteItem** y se le asigna la función **handlerDeleteItem**. La función se define con una función de flecha y va a contener un parámetro **id** para asegurar que el **id** del item que se va a eliminar se pase correctamente argumento.



Opcionalmente se puede pasar la función **handlerDeleteItem** directamente como una propiedad (prop) al componente **ListItemsView** sin envolverla en una función de flecha si no se necesita modificar los argumentos que recibe. Esto es posible porque la función ya está correctamente definida y su uso no requiere transformación adicional. Si la función solo toma un argumento y no necesitamos modificarlo, podemos pasarla directamente de esta manera: **handlerDeleteItem = {handlerDeleteItem}**.

Paso 118: Pasar el id del item hacia el componente ListItemsView

En el componente **ListItemsView**, se define la propiedad **handlerDeleteItem**. En el renderizado del componente **RowItemView**, se define una nueva propiedad llamada **handlerDeleteItem**, que se define como una función que tomará el valor de la propiedad que se pasa a **ListItemsView**, la función **handlerDeleteItem**.

Por otro lado, procede a crear una nueva fila cabecera de la tabla con **<th>** que va a contener el texto **"Eliminar"**. Y también se le añade una nueva propiedad al renderizado del componente **RowItemView** llamada **id**, cuyo valor será igual al **id** desestructurado definido en el método **map** de **items** (propiedad desestructurada de **ListItemsView**).



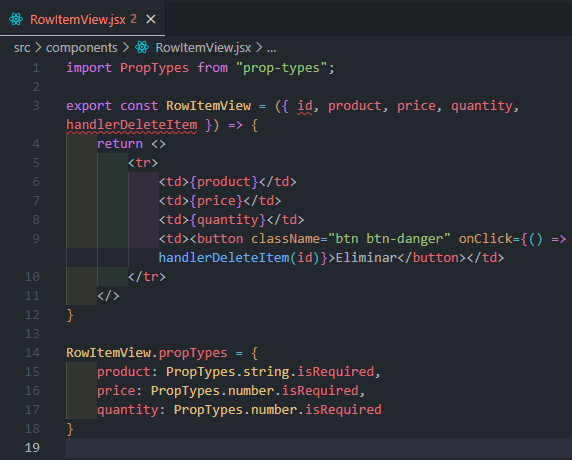


Paso 119: Pasar el id del item hacia el componente RowItemView

En el componente **RowItemView**, se pasa la propiedad **handlerDeleteItem** y se añaden unas nuevas etiquetas **<td>** dentro de las etiquetas **<tr>**. Las etiquetas **<td>** van a contener un botón definido con **<button></button>**. Luego se define un evento **onClick** que invoca la prop **handlerDeleteItem** como una función con su argumento **id**.

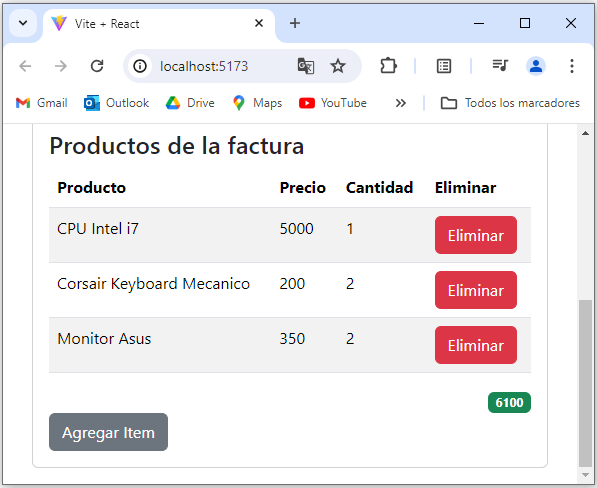
El evento **onClick** invoca una función de flecha que llama a la prop **handlerDeleteItem**, la cual viene del componente padre **ListItemsView**. Se le pasa solamente el **id** como argumento. Esta acción se propaga hacia arriba a través de las props hasta llegar al componente **InvoiceApp**.

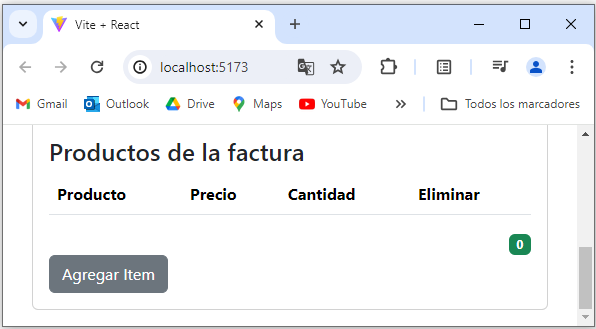
En el componente **RowItemView**, se añaden los estilos de Bootstrap al botón con la clase **btn** y **btn-danger**, que proporcionan la apariencia y el comportamiento visual apropiados para un botón de eliminación.



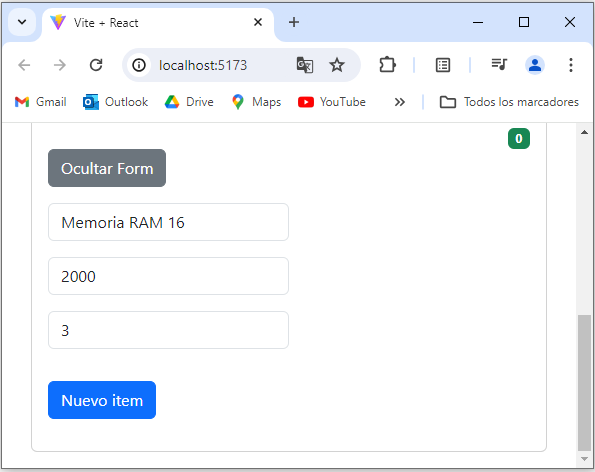
Paso 120: Realizar la prueba final

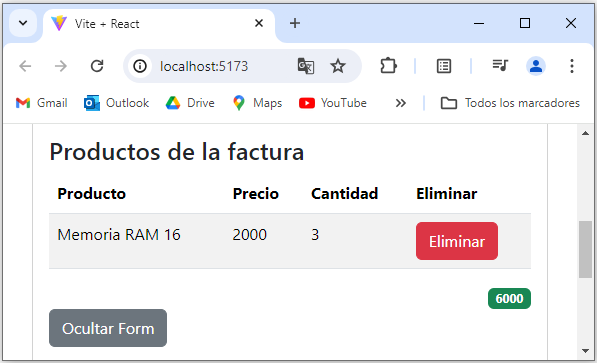
En el navegador, verifica que cada fila de la lista de ítems tenga un botón **Eliminar**. Al pulsar el botón **eliminar**, los ítems deben desaparecer uno por uno de la lista, y el total calculado debe actualizarse en consecuencia. Esto se debe a que la propiedad **handlerDeleteItem**, definida como una función, llama a la función del componente padre **ListItemsView**, pasándole el **id** del ítem correspondiente.





Al agregar un nuevo ítem, el proceso se repite, y el botón "Eliminar" estará disponible para eliminar este nuevo ítem también.





Hasta aquí se ha culminado el desarrollo de la aplicación **invoiceApp**.