**EXPLICACIÓN DE COMO MODIFICAR UN ITEM DE LA LISTA DE COMPRAS**

**COMPONENTE APP**

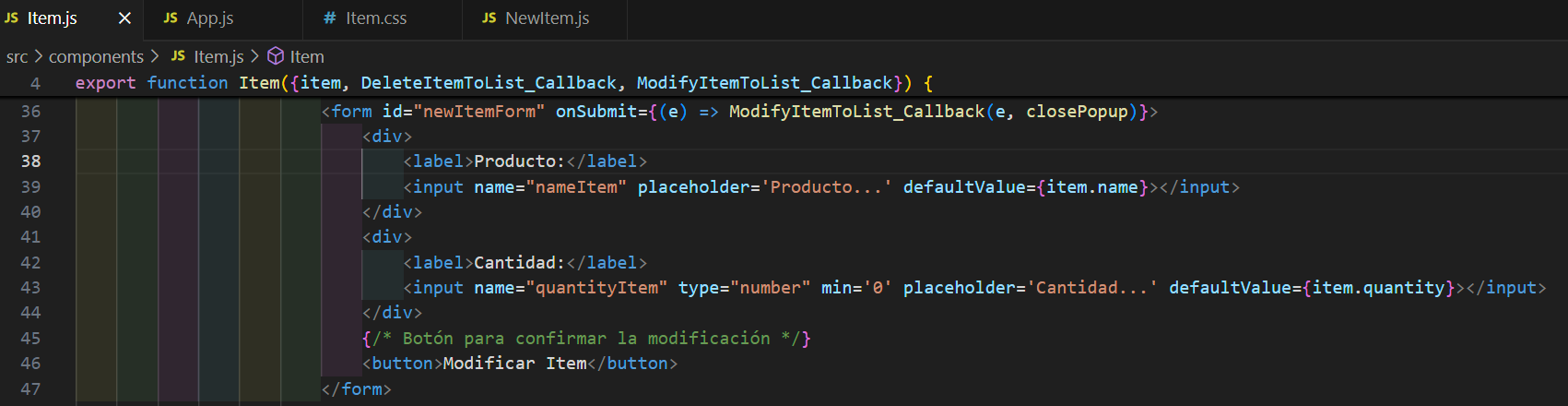
Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**COMPONENTE ITEM**



**DECLARACIÓN DE LA FUNCIÓN ANÓNIMA**: ModifyItemToList\_Callback.

**(e, closePopup) => ModifyItemToList(e, item\_map.id, closePopup)**

Esta es una **función de flecha** (arrow function) de JavaScript.

* **(e, closePopup)** son los parámetros de entrada de la función.
  + **e**: El evento del formulario (evento submit).
  + **closePopup**: La función que cierra el popup (pasada desde el componente Item).
  + **item\_map.id**: es el identificador del item específico que se está modificando.
* **ModifyItemToList(e, item\_map.id, closePopup)** es la expresión que se ejecutará cuando se llame a esta función.

Como se trata de una función de flecha con una sola línea de código (sin llaves {}), lo que sigue después de => es **implícitamente** el valor de retorno de la función. Significa que cuando esta función se ejecute, simplemente llamará a ModifyItemToList con los tres argumentos: el evento e, el id del item (item\_map.id), y la función closePopup.

Esta declaración de función permite **capturar Variables Locales** (**item\_map.id**). Aunque **item\_map.id** pertenece al contexto del método **map** de la lista, la función anónima "recuerda" esta variable cuando es creada. Este es un ejemplo de un **closure** en JavaScript, que permite que una función capture y acceda a variables en el contexto en que fue definida, incluso después de que dicho contexto haya cambiado o dejado de existir.

De esta manera, la función anónima **(e, closePopup) => ModifyItemToList(e, item\_map.id, closePopup)** permite capturar el id del item actual (**item\_map.id**) en su propio contexto. Asi, cada Item utiliza su propio id al invocar la función de modificación, asegurando que los cambios sean específicos para el item correspondiente.

**¿QUÉ OCURRE CUANDO SE MODIFICA EL ITEM Y SE DISPARA EL FORMULARIO?**

**<form id="newItemForm" onSubmit= {(e) => ModifyItemToList\_Callback(e, closePopup)} >**

* + Cuando el formulario es enviado, se llama a la función anónima **(e) => ModifyItemToList\_Callback(e, closePopup)** en el **submit** del form del componente **Item**.
  + **ModifyItemToList\_Callback**, a su vez, llama a **ModifyItemToList(e, item\_map.id, closePopup)** en el componente **App**, pasando el **evento de formulario** y la función **closePopup**. Notemos que **item\_map.id** ya tiene un valor que fue pasado en un inicio cuando se llamó al componente **Item**.
  + La función **ModifyItemToList** luego actualiza la lista de items y cierra el popup llamando a **closePopup()**.