Composición, dividiendo componente en subcomponentes II - ListView

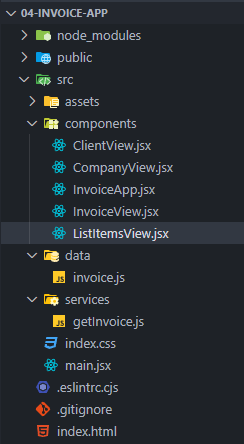
En esta sección se continuará con la refactorización del componente **InvoiceApp**, se procederá a extraer la sección de la tabla de productos de la factura en un nuevo componente llamado **ListItemsView**; con la finalidad de mantener el código más organizado y ordenado modularmente.

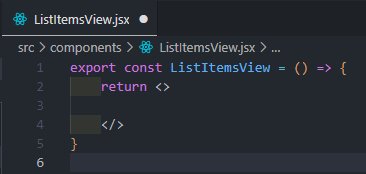
Pasos 40 al 43

Sigue estos pasos:

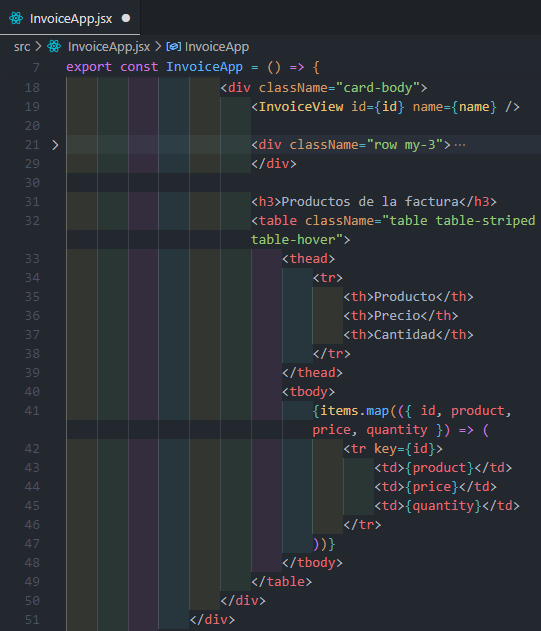
Paso 40: Crear el componente para la lista de items

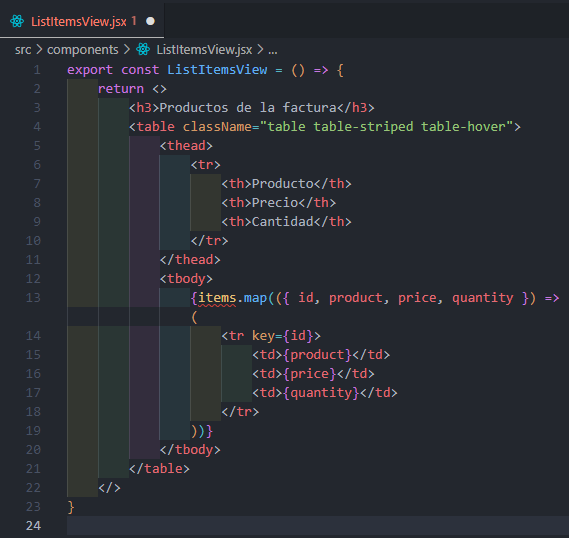
Crea creando un nuevo archivo en la carpeta components llamado **ListItemsView.jsx** o **TableItemView.jsx** según tu preferencia. En este archivo, se define una función exportable para el componente:





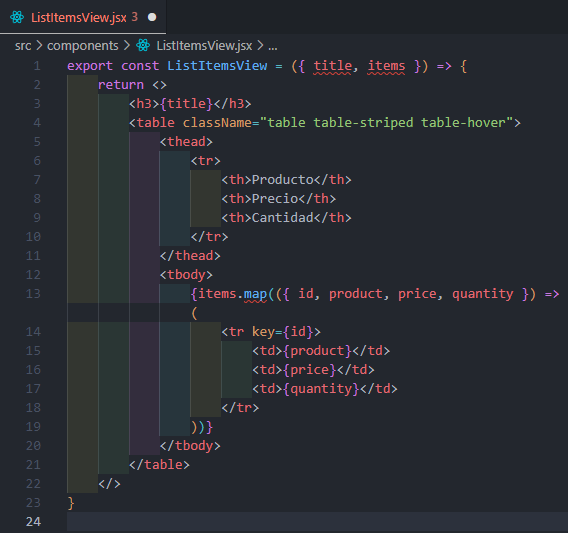
Luego, ve al archivo **InvoiceApp.jsx** y corta todo el código HTML que contiene el título **"Productos de la factura"** y la tabla definida con el elemento **<table>**, pegalo dentro del componente **ListItemsView** recién creado:





Paso 41: Integrar el componente ListItemsView en InvoiceApp

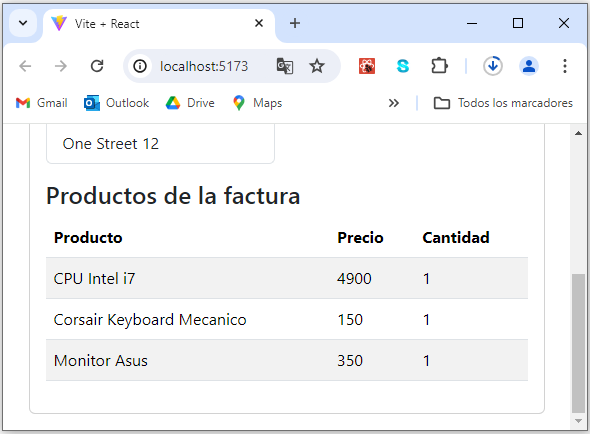
El componente **ListItemsView** va a aceptar dos props: **title** para el título de la sección e **items** para la lista de productos de la factura. Dentro de las etiquetas **<h3>**, se va a mostrar el título dinámicamente usando la prop **title**.



Luego, importa el componente **ListItemsView** en el componente padre **InvoiceApp**. Para la prop **title** se define un texto y la prop **items** contendrá la lista de productos de la factura, por lo cual se asigna el arreglo de objetos **items**.

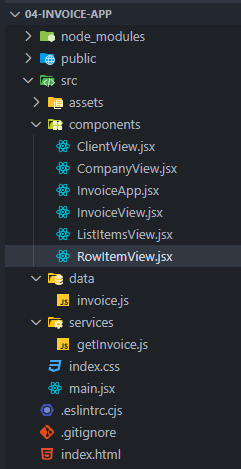


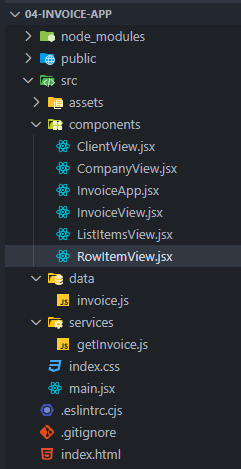
Esta modularización permite mantener un código más limpio y reutilizable. Cada componente tiene la responsabilidad de representar una parte específica de la factura, lo que facilita su mantenimiento y modificación. El navegador debe seguir funcionando correctamente después de esta refactorización. La reutilización se puede evidenciar al duplicar el componente **ListItemsView** varias veces (puedes intentar esto).

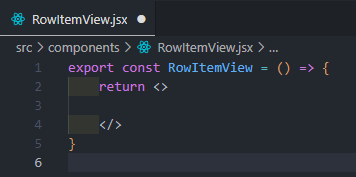


Paso 42: Crear el componente RowItemView

En la carpeta **components**, crea un nuevo archivo con el nombre **RowItemView.jsx**. Defina una función de flecha que va a retornar un fragmento vacío para inicializar el componente **RowItemView**.





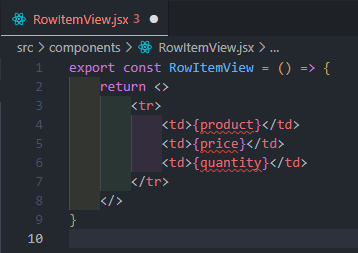


Este componente contendrá la estructura de una fila de la tabla de productos, esto va a permitir su reutilización en otros contextos, como por ejemplo, un carrito de compras.

En el componente **ListItemsView**, corta la fila dinámica de la tabla (se encuentra dentro de unas etiquetas **<tr>**) y pegalo en el componente **RowItemView**. Además se elimina el atributo **key** y su valor **id**.



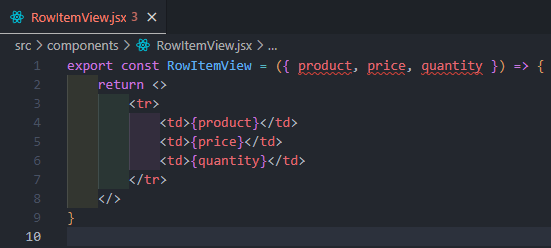




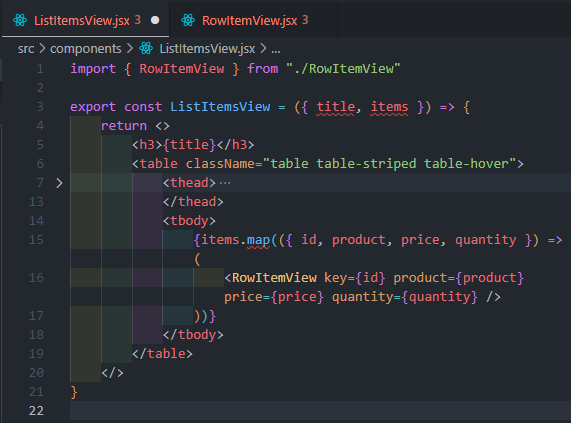
El código fuente de la fila de la tabla se transfiere al componente **RowItemView**.

Paso 43: Refactorizar el componente ListItemsView

En el componente **RowItemView**, se pasan los datos **product**, **price**, y **quantity** como props. Asimismo se excluye el **id** ya que se usará solamente como **key** en la iteración.



En el componente **ListItemsView**, importa el componente **RowItemView** y procede a utilizar sus props **product**, **price**, y **quantity** en la iteración de los elementos de la tabla. También se asigna la prop **id** como valor para el atributo **key** para cada **RowItemView**.



El componente **ListItemsView** ahora tiene un código más limpio y modular. **RowItemView** se encarga de renderizar cada fila de la tabla de productos, facilitando la reutilización del código.

El resultado en la consola y en el navegador debe permanecer sin cambios visibles, reflejando la correcta funcionalidad de la refactorización.

