Añadiendo PropTypes

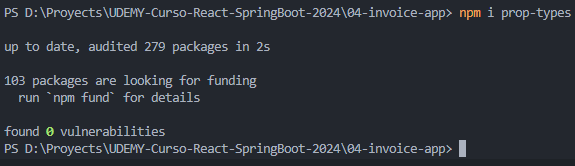
En esta sección se utilizará la biblioteca PropTypes para validar y especificar los tipos de datos de las propiedades que se están pasando en los componentes de React. Esto hará que la aplicación sea más estricta y segura, con más validaciones.

Pasos 44 al 49

Sigue estos pasos:

Paso 44: Instalar PropTypes

Primero, si la aplicación web está en ejecución, detenla pulsando las teclas **CTRL + C** dos veces. Luego, instala la dependencia **prop-types** que es parte del ecosistema de React pero debe instalarse por separado. Utiliza el siguiente comando: **npm i prop-types**



Revisa el archivo **package.json** para asegurarte de que la dependencia **prop-types** se ha añadido correctamente. Levanta la aplicación nuevamente con el comando: **npm run** **dev**



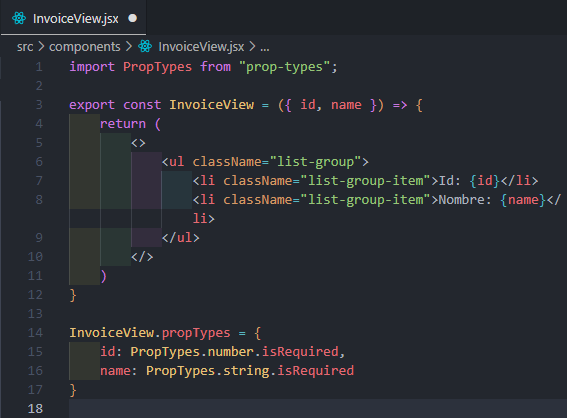


Paso 45: Añadir PropTypes a InvoiceView

En el archivo **InvoiceApp.jsx**, a plena vista se observa que no se están pasando props directamente, pero se están enviando props a los componentes hijos como **InvoiceView**.

Ve al componente **InvoiceView** para añadir las validaciones de PropTypes. Primero, importa **PropTypes** al inicio del archivo **InvoiceView.jsx**:

Luego añade las definiciones de PropTypes para el componente **InvoiceView**. Copia el nombre del componente y pégalo debajo de la función, añadiendo **.propTypes**. Dentro, define el tipo de dato para cada una de las props (**id** y **name**). Usa **isRequired** si una prop es obligatoria.



Después de añadir PropTypes, la aplicación web debe funcionar sin errores. Abre el navegador y verifica que todo se muestra correctamente. No debe haber errores en la consola.

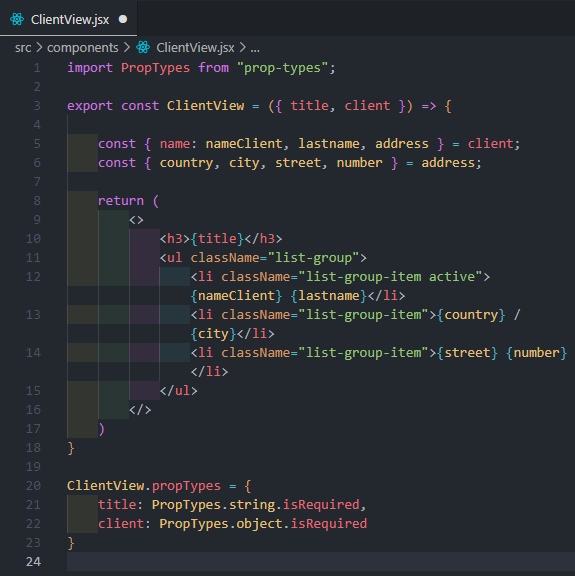
**Nota:** Para probar las validaciones de PropTypes, cambia el tipo de dato de **id** en **InvoiceView** a string en lugar de number. Esto generará un error en la consola que indicará que se esperaba un dato de tipo number para **id**. Este error se puede comprobar en el archivo **invoice.js**, donde el atributo **id** es de tipo number porque no está en comillas.

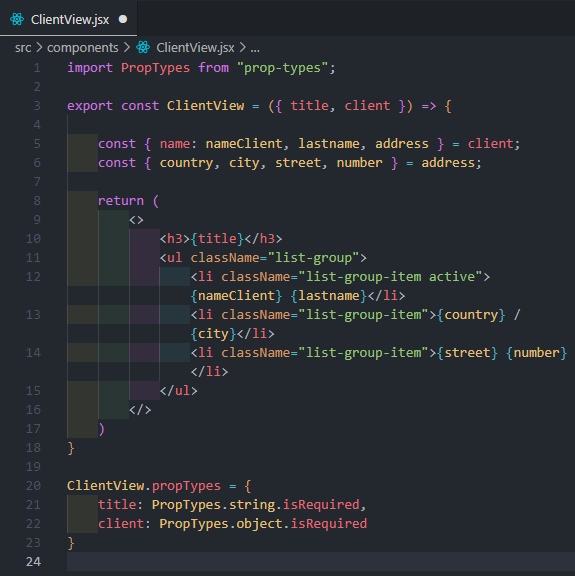
Paso 46: Añadir PropTypes a ClientView

Recuerda que con la validación de props en los componentes se garantiza que las props que se pasan sean del tipo esperado y ayuda a prevenir errores.

Importa PropTypes en el componente **ClientView**. Añade la línea de importación al inicio del archivo.

Debajo de la función del componente **ClientView**, define las propTypes para cada uno de los atributos que el componente recibe. Aquí, el atributo **client** es de tipo objeto, y el atributo **title** es de tipo string y ambos son requeridos.





Paso 47: Añadir PropTypes a CompanyView

Procede a agregar la validación de tipos para las props en el componente **CompanyView** usando PropTypes.

Importa PropTypes al inicio del archivo **CompanyView.jsx**.

Debajo de la definición de la función del componente **CompanyView**, especifica los tipos de datos esperados para cada prop. En este caso, **title** es un string y **company** es un objeto. Ambos son requeridos.



Paso 48: Añadir PropTypes a ListItemsView

Importama PropTypes al inicio del archivo **ListItemsView.jsx**.

Debajo de la definición de la función del componente **ListItemsView**, especifica los tipos de datos esperados para cada prop, **title** es un string requerido, e **items** es un arreglo de objetos. Además, se puede hacer que la prop **items** no sea obligatoriamente requerida si el objetivo es permitir facturas sin productos.



Paso 49: Añadir PropTypes a RowItemView

Para finalizar la validación de props en todos los componentes creados hasta el momento, procede a agregar PropTypes al componente **RowItemView**. Esto asegurará que todos los datos recibidos sean del tipo esperado.

Importa PropTypes al inicio del archivo **RowItemView.jsx**.

Debajo de la definición de la función del componente **RowItemView**, especifica los tipos de datos esperados para cada prop, **product** es un string, **price** es un número y **quantity** también es un número.

