Introducción a la sección

En React, la composición de componentes es fundamental para construir aplicaciones modulares y reutilizables. Esta característica nos permite descomponer la interfaz de usuario en pequeñas piezas independientes, cada una encargada de una tarea específica.

Composición de componentes

El siguiente proyecto que se va a construir a partir de la sección **3.3** hasta la sección **3.10**, consiste en lo siguiente:

Vamos a crear un componente **HelloWorldApp**, que a su vez va a estar compuesto por tres subcomponentes:

* **Title:** Este componente recibe el título como una prop (propiedad) y lo muestra en un elemento **<h1>**. El título se define en el archivo **main.jsx** y se pasa como una prop al componente **HelloWorldApp**, que luego se transmite al componente **Title**.
* **UserDetails:** Este componente muestra detalles sobre un usuario, como su nombre y apellido. El nombre se pasa como una prop al componente **UserDetails** desde el componente **HelloWorldApp**. El apellido se fija internamente en el componente, pero podría ser pasado como una prop desde un componente padre si fuera necesario.
* **Book:** Este componente muestra detalles sobre un libro, como su nombre. Similar a **UserDetails**, el nombre del libro se pasa como una prop al componente **Book** desde el componente **HelloWorldApp**.

En el archivo **main.jsx**, definimos los valores para el título, el nombre del usuario y el nombre del libro. Luego, pasamos estos valores como props a **HelloWorldApp**, que a su vez los transmite a sus subcomponentes.

Al ejecutar la aplicación, vamos a ver cómo se compone **HelloWorldApp** con sus subcomponentes **Title**, **UserDetails** y **Book**, y cómo los valores se pasan de un componente a otro para crear una interfaz de usuario coherente y dinámica.

Al comprender estos conceptos de composición y descomposición de componentes, podemos construir aplicaciones más estructuradas, escalables y fáciles de mantener en React.