**2.1: Información del curso paso 6**

// COMPONENTE PRINCIPAL

const App = () => {

const course = 'Half Stack application development'

const parts = [

{

name: 'Fundamentals of React',

exercises: 10

},

{

name: 'Using props to pass data',

exercises: 7

},

{

name: 'State of a component',

exercises: 14

}

]

return <Course course={course} parts={parts} />

}

// COMPONENTE QUE FORMATEA UN CURSO A LA VEZ

const Course = ({ course, parts }) => {

return (

<>

<Header course={course} />

<Content parts={parts} />

</>

)

}

// COMPONENTE HEADER (ENCABEZADO)

const Header = ({ course }) => <h1>{course}</h1>

// COMPONENTE QUE DEFINE UNA PARTE

const Part = ({ part }) => (

<p>

{part.name} {part.exercises}

</p>

)

// COMPONENTE CONTENT (CONTENIDO)

const Content = ({ parts }) => (

<>

{parts.map((part, index) => (

<Part key={index} part={part} />

))}

</>

)

export default App

**1. Componente App (Componente principal)**

Este es el componente principal de tu aplicación:

const App = () => {

const course = 'Half Stack application development'

const parts = [

{ name: 'Fundamentals of React', exercises: 10 },

{ name: 'Using props to pass data', exercises: 7 },

{ name: 'State of a component', exercises: 14 }

]

return <Course course={course} parts={parts} />

}

**Explicación**:

* Aquí defines dos variables:
  + course: es una cadena de texto que almacena el nombre del curso ('Half Stack application development').
  + parts: es un array de objetos, donde cada objeto tiene dos propiedades: name y exercises. Cada objeto representa una parte del curso y cuántos ejercicios tiene.
* Luego, en el return, el componente Course es llamado, pasándole dos props:
  + course: el nombre del curso.
  + parts: las partes del curso, que incluyen el nombre de cada parte y los ejercicios.

**2. Componente Course**

Este componente es responsable de renderizar el curso y sus partes.

const Course = ({ course, parts }) => {

return (

<>

<Header course={course} />

<Content parts={parts} />

</>

)

}

**Explicación**:

* Course recibe dos props: course (nombre del curso) y parts (array de partes del curso).
* Dentro del componente Course, se renderizan dos componentes:
  1. **Header**: se le pasa el prop course para mostrar el nombre del curso.
  2. **Content**: se le pasa el prop parts, que contiene las partes del curso, y este componente será responsable de mostrar los detalles de cada parte.

**3. Componente Header**

Este componente es simple y solo muestra el título del curso:

const Header = ({ course }) => <h1>{course}</h1>

**Explicación**:

* Header recibe el prop course que contiene el nombre del curso.
* Muestra el nombre del curso dentro de una etiqueta HTML <h1>. Es simplemente el encabezado.

**4. Componente Part**

Este componente se encarga de renderizar una parte específica del curso (nombre de la parte y cuántos ejercicios tiene).

const Part = ({ part }) => (

<p>

{part.name} {part.exercises}

</p>

)

**Explicación**:

* Part recibe un objeto part como prop, que contiene dos propiedades: name (nombre de la parte) y exercises (número de ejercicios).
* Luego, dentro de un párrafo (<p>), muestra el nombre de la parte y el número de ejercicios.

**5. Componente Content**

Este componente es responsable de renderizar todas las partes del curso, usando el componente Part para cada una.

const Content = ({ parts }) => (

<>

{parts.map((part, index) => (

<Part key={index} part={part} />

))}

</>

)

**Explicación**:

* Content recibe un prop parts, que es un array de objetos (cada objeto representa una parte del curso).
* Usa el método map() para iterar sobre cada parte en el array parts. Por cada parte, se crea un componente Part pasándole el objeto part como prop.
* **key={index}**: key es una propiedad especial en React que ayuda a identificar cada elemento cuando se renderizan listas de elementos. Usamos index como clave única para cada parte en el array.

**6. Resultado final**

Cuando todo esto se combina, el flujo de la aplicación es el siguiente:

1. App pasa el nombre del curso y las partes al componente Course.
2. Course pasa el nombre del curso al componente Header (para que lo muestre en un encabezado) y pasa las partes al componente Content.
3. Content toma el array parts y, usando map(), renderiza un componente Part para cada una, mostrando tanto el nombre de la parte como el número de ejercicios.

**Visualización del flujo:**

1. App -> Course -> Header (muestra el nombre del curso).
2. App -> Course -> Content -> Part (muestra cada parte con su número de ejercicios).

Con este enfoque, tu aplicación es modular, cada parte de la interfaz está bien organizada en componentes y se hace un uso correcto de las props para pasar datos entre componentes.