**2.4: Información del curso paso 9**

//COMPONENTE PRINCIPAL

const App = () => {

  const courses = [

    {

  name: "Half Stack application development",

  id:1,

  parts: [

    {

      name: "Fundamentals of React",

      exercises: 10,

      id:1

    },

      {

        name: "Using props to pass data",

        exercises:7,

        id:2

      },

      {

        name:"State of a component",

        exercises:14,

        id:3

      },

      {

        name:"Redux",

        exercises:11,

        id:4

      }

  ]

  },

  {

    name:"Node.js",

    id:2,

    parts: [

      {

        name:"Routing",

        exercises:3,

        id:1

      },

      {

        name: "Middlewares",

        exercises: 7,

        id:2

      }

    ]

  }

]

/\*

Define el array courses, que contiene dos cursos: 'Half Stack application development' y 'Node.js'.

Usamos map() para iterar sobre courses. En cada iteración, pasamos el curso actual como prop al componente Course.

key={course.id}: React necesita una clave única para identificar cada curso, por lo que utilizamos el id de cada curso.

\*/

  return (

    <div>

      {courses.map(course=>(

        <Course key={course.id} course={course} />

      ))}

    </div>

  )

}

// COMPONENTE QUE FORMATEA UN CURSO A LA VEZ

const Course = ({ course}) => { //Course recibe dos props: course (nombre del curso) y parts (array de partes del curso).\*

  return ( //Dentro del componente Course, se renderizan dos componentes:\*\*

  /\*

  Header: se le pasa el prop course para mostrar el nombre del curso.

  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.

  \*/

    <>

      <Header course={course.name} />

      <Content parts={course.parts} />

    </>

  )

}

// COMPONENTE HEADER (ENCABEZADO)

 //Componente que muestra el título del curso

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

/\*

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.

\*/

// COMPONENTE QUE DEFINE UNA PARTE

//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>

  /\*

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.

\*/

)

// COMPONENTE CONTENT (CONTENIDO)

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

const Content = ({ parts }) => {

// sumar todos los ejercicios usando el metodo reduce

const total = parts.reduce((sum, part) => sum+part.exercises,0)

//Usa map() para iterar sobre las partes del curso y renderizar un componente Part para cada una.

return (

  <>

     {parts.map(part => (

        <Part key={part.id} part={part} />

      ))}

      <p><strong>Total of {total} exercises</strong></p>

  </>

  )

  }

  /\*

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.

\*/

/\*

const totalExercises = parts.reduce((sum, part) => sum + part.exercises, 0): Aquí estamos usando reduce para sumar todos los ejercicios:

    sum es el acumulador que comienza en 0 (el segundo argumento del reduce).

    part.exercises es el número de ejercicios de la parte actual.

    En cada iteración, se va sumando el número de ejercicios de la parte al valor acumulado.

<p><strong>Total of {totalExercises} exercises</strong></p>: Después de mostrar todas las partes, agregamos una línea que muestra el total de ejercicios calculado.

\*/

export default App

**Explicación paso a paso:**

1. **App**:
   * Define el array courses, que contiene dos cursos: 'Half Stack application development' y 'Node.js'.
   * Usamos map() para iterar sobre courses. En cada iteración, pasamos el curso actual como prop al componente Course.
   * **key={course.id}**: React necesita una clave única para identificar cada curso, por lo que utilizamos el id de cada curso.
2. **Componente Course**:
   * Recibe un solo curso como prop (course), que contiene el nombre del curso y las partes.
   * Pasa el nombre del curso al componente Header y las partes al componente Content.
3. **Componente Header**:
   * Muestra el nombre del curso dentro de una etiqueta <h1>.
4. **Componente Part**:
   * Recibe una parte individual del curso y muestra el nombre y el número de ejercicios en un párrafo.
5. **Componente Content**:
   * Usa map() para iterar sobre las partes del curso y renderizar un componente Part para cada una.
   * Además, usa reduce() para calcular el total de ejercicios y mostrarlo después de listar las partes.

**Resultado esperado:**

* La aplicación ahora puede manejar cualquier cantidad de cursos, renderizando un curso tras otro, y mostrando el total de ejercicios de cada curso al final.
* Si agregas más cursos al array courses, el componente los renderizará automáticamente.