Estados avanzados
clase n°16

revolución* digital_

San Nicolás de los Arroyos


REC
recordá
poner a grabar la clase

introducción a los estados avanzados

En React, el **estado compartido** se refiere al manejo de datos que necesitan ser utilizados por varios componentes. Este proceso permite **sincronizar información entre componentes** y mantener una única fuente de verdad.

Existen varias formas de compartir estados en React, como:

- **Lifting State Up:** Mover el estado a un componente padre.
- **Context API:** Usado cuando necesitamos compartir estado en un árbol profundo.
- **Estados derivados:** Cálculo de valores a partir de un estado existente.

En esta clase, nos enfocaremos en el uso de Lifting State Up.

cuando compartir estado

El estado debe compartirse cuando:

- Dos o más componentes necesitan leer y modificar los mismos datos.
- Los cambios en un componente deben reflejarse en otro.
- Deseamos mantener una única fuente de verdad para evitar inconsistencias.

estrategias para compartir estado

El proceso de **Lifting State Up** es el siguiente:

- Crear el estado en el componente padre.
- Pasar el estado y las funciones para modificarlo a los componentes hijos mediante props.
- Los hijos pueden invocar estas funciones para modificar el estado en el padre.

A continuación, veremos ejemplos simples.

Un pequeño repaso: Compartir estado de Hijo a Padre

En este ejemplo, un componente hijo enviará información al padre utilizando una función recibida por props.

```jsx
//$ src/components/App.js
import { useState } from "react";
import Child from "./Child"; // Importar el componente hijo
function App() {
  const [message, setMessage] = useState("");  // Estado en el padre
  // Función para actualizar el estado
  const handleMessage = (newMessage) => {
    setMessage(newMessage);
  };
  return (
    <div>
      <h1>Mensaje desde el Hijo:</ h1>
      <p>{message}</ p> {/* Mostrar el mensaje */ }
      <Child onMessageChange={handleMessage} /> { /* Pasar la función al hijo */ }
    </div>
  );
}
export default App;
```

```jsx
//$ src/components/Child.js
import React from "react";
function Child({ onMessageChange }) {
  const sendMessage = () => {
    onMessageChange( "Hola desde el Hijo!" ); // Llamar a la función del padre
  };
  return (
    <div>
      <button onClick={sendMessage}>Enviar Mensaje</ button>
    </div>
  );
}
export default Child;
```

paso a paso

- El **estado message** está en el componente padre (App)
- La función **handleMessage** permite modificar el estado
- El **componente hijo** (Child) recibe esta función como prop y la llama
- Esto actualiza el estado del padre y **se refleja en la interfaz**

Ejemplo: Compartir estado de Hijo a Padre y a otros Hijos

En este caso, el estado enviado por un hijo se compartirá con otros hijos a través del padre.

```
//$ src/components/App.js
import React, { useState } from "react";
import ChildA from "./ChildA";
import ChildB from "./ChildB";
function App() {
  const [data, setData] = useState("");  // Estado en el padre
  return (
    <div>
      <h1>Compartir Estado entre Hijos</ h1>
      <ChildA onDataChange={setData} /> { /* Hijo que actualiza el estado */ }
      <ChildB data={data} /> { /* Hijo que lee el estado */ }
    </div>
  );
}
export default App;
```

```
//$ src/components/ChildA.js
import React from "react";
function ChildA({ onDataChange }) {
  const sendData = () => {
    onDataChange( "Dato compartido desde Hijo A" ); // Enviar dato al padre
  };
  return (
    <div>
      <button onClick={sendData}>Enviar Dato al Padre</ button>
    </div>
  );
}
export default ChildA;
```

```
//$ src/components/ChildB.js
import React from "react";
function ChildB({ data }) {
  return (
    <div>
      <h2>Hijo B:</h2>
      <p>{data ? data :  "No hay datos compartidos" }</p> {/* Mostrar el dato */ }
    </div>
  );
}
export default ChildB;
```

paso a paso

- El **estado data** vive en el componente padre (App).
- El componente **ChildA** recibe la función **setData** como prop y la utiliza
- El componente **ChildB** recibe el **estado data** como prop y lo muestra en la
- Los cambios realizados en **ChildA** se reflejan inmediatamente en **ChildB**.
- **Conclusión:** Compartir estado mediante **Lifting State Up** es una técnica clave en React para **sincronizar datos entre componentes.** Este enfoque asegura que todos los componentes involucrados se mantengan sincronizados al usar una única fuente de verdad.

Próximos pasos:
Experimentar con estados compartidos más complejos.
Explorar el uso de Context API para casos donde el árbol de componentes sea profundo.

actividad

Trabajan en una empresa que desarrolla herramientas de encuestas. Les han asignado un proyecto para refactorizar un componente donde los usuarios seleccionan una respuesta a una pregunta. Actualmente, cada componente hijo gestiona su propio estado, lo que dificulta que el formulario recopile todas las respuestas en un único lugar. Su tarea es centralizar el estado en el componente padre utilizando elevación del estado.
El formulario tiene dos preguntas, cada una en su propio componente hijo. El estado debe ser centralizado para permitir el envío de las respuestas.
Crea el componente “Encuesta.jsx” copia el siguiente código y pegalo en el archivo creado.

```
//$ src/Encuesta.js
import { useState } from "react";
function Encuesta() {
  return (
    <div>
      <h1>Encuesta</h1>
      <Pregunta1 />
      <Pregunta2 />
      <BotonEnviar />
    </div>
  );
}
function Pregunta1() {
  const [respuesta, setRespuesta] = useState("");
  return (
    <div>
      <h2>¿Cuál es tu color favorito?</ h2>
      <select value={respuesta} onChange={(e)  => setRespuesta(e.target.value)}>
        <option value="">Selecciona una opción</ option>
        <option value="Rojo">Rojo</option>
        <option value="Azul">Azul</option>
        <option value="Verde">Verde</option>
      </select>
    </div>
  );
```

```
}
function Pregunta2() {
  const [respuesta, setRespuesta] = useState("");
  return (
    <div>
      <h2>¿Cuál es tu animal favorito?</ h2>
      <select value={respuesta} onChange={(e) => setRespuesta(e.target.value)}>
        <option value="">Selecciona una opción</ option>
        <option value="Perro">Perro</option>
        <option value="Gato">Gato</option>
        <option value="Ave">Ave</option>
      </select>
    </div>
  );
}
function BotonEnviar() {
  const enviarEncuesta = ()  => {
    alert("Encuesta enviada.");
  };
  return (
    <div>
      <button onClick={enviarEncuesta}>Enviar Encuesta</ button>
    </div>
  );
}
export default Encuesta;

```

revøluciøn*
digital_