

Tasca 6. Arregleu el component Scoreboard:

Aquest formulari té alguns errors. Feu clic al botó que augmenta la puntuació unes quantes vegades. Observeu que no augmenta. A continuació, editeu el primer nom i observeu que la puntuació s'actualitza de sobte. Finalment, editeu el cognom i observeu que la puntuació ha desaparegut completament.

La vostra tasca és solucionar tots aquests errors. A mesura que els arregles, explica per què passa cadascun d'ells.

Lo que pasa es que en `handlePlusClick()` estamos modificando `player.score` directamente, y React no detecta cambios en objetos mutados directamente. React solo vuelve a renderizar cuando se usa `setState` correctamente. Utilizaremos `setPlayer` para actualizar el estado.

```
function handlePlusClick() {  
  setPlayer(prevPlayer => ({  
    ...prevPlayer,  
    score: prevPlayer.score + 1  
  }));  
}
```

Para solucionar el error de que la puntuación se actualiza de golpe al cambiar el nombre es de la siguiente forma, cuando se cambia `firstName`, se usa `setPlayer` con un nuevo objeto basado en el estado anterior, lo cual esta bien. Sin embargo, `handleLastNameChange` solo establece `lastName`, sin preservar `score` ni `firstName`, causando una pérdida de datos.

Tendremos que modificar `handleLastNameChange` para que mantenga el estado anterior.

```
function handleFirstNameChange(e) {  
  setPlayer({  
    ...player,  
    firstName: e.target.value,  
  });  
}
```

Tasca 7. Arregleu el component ShoppingCart:

Aquest carretó de la compra té un botó "+" que funciona, però el botó "-" no fa res. Heu d'afegir-hi un handler perquè al premer-lo, disminueixi el comptador del producte corresponent. Si premeu "-" quan el recompte és 1, el producte s'hauria d'eliminar automàticament del carretó. Assegureu-vos que no mostri mai 0.

Los fallos que tiene esta aplicación son:

Primero que falta el `handleIncreaseClick`, por lo que el boton + no hace nada. Luego, no hay botón - para disminuir la cantidad de producto. El producto no se elimina cuando su

cantidad llega a 1 y se presiona -. Y por último que el estado debe actualizarse correctamente sin modificarlo directamente.

Lo que haremos será implementar el botón de incrementar y añadir el botón de disminuir el contador y eliminar el producto si su cantidad llega a 1.

```
function handleIncreaseClick(productId) {
  setProducts(prevProducts =>
    prevProducts.map(product =>
      product.id === productId
        ? { ...product, count: product.count + 1 }
        : product
    )
  );
}

function handleDecreaseClick(productId) {
  setProducts(prevProducts =>
    prevProducts
      .map(product =>
        product.id === productId
          ? { ...product, count: product.count - 1 }
          : product
      )
      .filter(product => product.count > 0) // Elimina productos con count 0
  );
}
```

Tasca 8. Arregleu el component TaskApp:

En aquest exemple, tots els controladors d'esdeveniments fan servir la mutació. Com a resultat, l'edició i la supressió de TO-DO's no funciona. Reescriu handleAddTodo, handleChangeTodo, i handleDeleteTodo perquè facin servir mètodes no mutatius.

Los problemas que he encontrado han sido que por ejemplo el handleAddTodo muta todos.push() en lugar de crear un nuevo array. También que el handleChangeTodo modifica el objeto directamente en lugar de crear una nueva lista con el todo actualizado. Y por último que handleDeleteTodo usa splice(), que modifica el array original, en lugar de crear una nueva lista filtrada.

Para solucionar cada uno he hecho lo siguiente:

- handleAddTodo: Usa setTodos([...prevTodos, nuevoTodo]).
- handleChangeTodo: Usa setTodos con map() para actualizar sin mutar.
- handleDeleteTodo: Usa filter() para eliminar sin modificar el array original.

