

TEST EXAM - 1 -

Ejercicio: Gestión avanzada del carrito de Pokémon

En este ejercicio, trabajarás con el archivo `db.json` para implementar una lógica de carrito de compras. Además, te enfocarás en filtrar, gestionar y actualizar el carrito, así como en emplear Promesas, `Promise.allSettled`, `Map`, `Set`, y CRUD.

Objetivos**1. Obtener todas las cartas de Pokémon:**

- Desarrolla una función que recupere todas las cartas de Pokémon de `http://localhost:3000/pokemons` utilizando `async/await` y almacénalas en un `Map` para un acceso rápido por `nombre`.

2. Agregar y quitar del carrito:

- Implementa funciones para agregar o eliminar cartas en el carrito usando su `id`.
- Asegúrate de que al agregar una carta ya existente, se actualice la cantidad en lugar de duplicarla. Utiliza un `Set` para evitar duplicados.

3. Filtrar Pokémon por habilidades:

- Implementa una función que filtre los Pokémon por habilidades específicas y devuelva aquellos que las posean.
- Utiliza `Set` para asegurar que no haya habilidades repetidas y usa `Map` para almacenar y devolver los resultados.

4. Actualizar precios en base al multiplicador:

- Crea una función para actualizar los precios de cada carta cuando cambie el multiplicador en `db.json`.
- Usa `Promise.allSettled` para manejar las actualizaciones de cada carta, asegurando que si alguna promesa falla, puedas ver el error específico de esa carta.

5. CRUD del carrito:

- Implementa funciones para realizar operaciones CRUD en el carrito:
 - **Create:** Añadir una nueva carta.
 - **Read:** Leer la lista actual de cartas en el carrito.
 - **Update:** Actualizar la cantidad de una carta específica.
 - **Delete:** Eliminar una carta del carrito.

Información sobre db.JSON

1. **Pokemons:** Lista de 13 cartas de Pokémon con habilidades, puntuaciones y precios calculados. El precio de cada carta es `puntuacionTotal * multiplicador`.
2. **Multiplicador:** Usado para calcular el precio total de cada carta.
3. **Carrito:**
 - `items`: Array de objetos que representa cada Pokémon en el carrito con su `id`, `nombre`, `cantidad`, `precioUnitario`, y `precioTotal`.
 - `totalCarrito`: Total acumulado del carrito, calculado como la suma de todos los `precioTotal` de los ítems en `items`.