

WeakMap para torpes 🤪

¿Qué es?

Un `WeakMap` es como un diccionario que relaciona objetos con valores, pero con superpoderes de limpieza automática.

Analogía simple

Imagina un guardarropa de un teatro:

- **Entrada normal (objeto clave):** Tu abrigo 🧥
- **Ticket (valor asociado):** Información sobre tu abrigo
- **Magia:** Cuando te vas del teatro con tu abrigo, el ticket se destruye automáticamente (no queda basura)

Diferencia con Map normal

```
// Map normal - guarda referencias fuertes
const mapNormal = new Map();
let objeto = { nombre: 'Juan' };
mapNormal.set(objeto, 'algún dato');
objeto = null; // ❌ El objeto NO se borra de memoria (Map lo retiene)

// WeakMap - referencias débiles
const weakMap = new WeakMap();
let objeto2 = { nombre: 'Ana' };
weakMap.set(objeto2, 'algún dato');
objeto2 = null; // ✅ El objeto SÍ puede ser borrado (WeakMap no lo retiene)
```

En tu código

```
const estudianteMap = new WeakMap();

// Al crear el botón/elemento
const boton = document.createElement('li');
const estudiante = { nombre: 'Pedro', nota: 8 };

// Guardas: "este botón pertenece a este estudiante"
estudianteMap.set(boton, estudiante);

// Al hacer click
boton.addEventListener('click', (event) => {
  // Recuperas: "¿qué estudiante pertenece a este botón?"
  const est = estudianteMap.get(event.currentTarget);
  console.log(est.nombre); // 'Pedro'
});

// Si eliminas el botón del DOM
boton.remove();
// 🧹 El WeakMap automáticamente limpia su entrada
```

Ventajas en tu caso

1. **No usas arrays:** No necesitas `dataset.index` ni buscar en el array
2. **Una sola función:** Todos los botones comparten el mismo `handleClick`
3. **Limpieza automática:** Si borras un estudiante del DOM, no queda basura en memoria

Restricciones

- Solo acepta **objetos** como claves (no strings, números)
- No puedes listar todas las claves (no tiene `.keys()`, `.forEach()`)
- Solo tiene: `.set()`, `.get()`, `.has()`, `.delete()`

Regla de oro

Usa WeakMap cuando asocies **elementos DOM** ↔ **datos** y quieras que la memoria se limpie sola. 📝 ✨