1. Introducción al almacenamiento local

¿Qué es el almacenamiento local en navegadores?

El almacenamiento local permite a las aplicaciones web guardar datos en el navegador del usuario sin necesidad de enviarlos a un servidor. Esto mejora la velocidad y permite el acceso a la información incluso sin conexión a internet.

Comparación de LocalStorage, SessionStorage e IndexedDB

Característica	LocalStorage	SessionStorage	IndexedDB
Persistencia	Permanente hasta ser eliminado	Se borra al cerrar la pestaña	Permanente hasta ser eliminado
Tamaño máximo	~5 MB	~5 MB	~50 MB o más (dependiendo del navegador)
Tipo de datos	Solo texto (strings)	Solo texto (strings)	Objetos estructurados
Asincronía	No	No	Sí
Casos de uso	Configuración de usuario, temas, preferencias	Datos temporales en una sesión	Almacenamiento de datos complejos (listas, registros)

Ventajas del almacenamiento local frente a soluciones en el servidor

- **Velocidad:** Acceso inmediato sin necesidad de realizar peticiones HTTP.
- **Offline:** Permite que algunas funcionalidades sigan disponibles sin conexión.
- **Privacidad:** No requiere enviar datos sensibles al servidor.

2. LocalStorage

Definición y características

- **Es un almacenamiento clave-valor:** Cada dato se guarda con una clave única.
- **Persiste incluso después de cerrar el navegador.**

Métodos principales

javascript Copiar Editar

```javascript
// Guardar un dato
localStorage.setItem("nombre", "Juan");

// Recuperar un dato
let nombre = localStorage.getItem("nombre");
console.log(nombre); // "Juan"

// Eliminar un dato
localStorage.removeItem("nombre");

// Limpiar todo el almacenamiento
localStorage.clear();
```

Limitaciones de LocalStorage

- **Tamaño limitado (~5 MB)**
- **Solo almacena cadenas de texto**, por lo que se debe usar `JSON.stringify` y `JSON.parse` para objetos:

javascript Copiar Editar

```javascript
let usuario = { nombre: "Juan", edad: 25 };
localStorage.setItem("usuario", JSON.stringify(usuario));

let usuarioRecuperado = JSON.parse(localStorage.getItem("usuario"));
console.log(usuarioRecuperado.edad); // 25
```

🔍 ¿Se puede borrar por nombre en IndexedDB?

No directamente. IndexedDB **sólo permite borrar objetos por su clave primaria (keyPath)**, no por otros atributos.

📌 ¿Qué es el `keyPath` en IndexedDB?

Cuando creamos un almacén de objetos en IndexedDB, definimos una **clave única** (llamada `keyPath`).

- **IndexedDB usa esta clave para acceder, modificar o eliminar registros.**
- **No permite buscar directamente por otros atributos.**

Ejemplo de definición de un `keyPath`:

```js
let request = indexedDB.open("MiBaseDeDatos", 1);
request.onupgradeneeded = function(event) {
    let db = event.target.result;

    // Se crea el almacén con 'id' como clave primaria
    db.createObjectStore("usuarios", { keyPath: "id" });
};
```

Aquí, el `id` es el `keyPath`, lo que significa que:

✅ Podemos eliminar por `id` → `store.delete(1)` (✅ Válido).

❌ No podemos eliminar por `nombre` directamente → `store.delete("Juan")` (❌ No válido).

🔥 ¿Cómo borrar por un campo que no es `keyPath`?

Para eliminar por `nombre`, necesitamos **buscar primero el objeto con un cursor** y luego eliminarlo.

Ejemplo paso a paso para borrar por nombre

js Copiar Editar

```js
let transaction = db.transaction(["usuarios"], "readwrite");
let store = transaction.objectStore("usuarios");

// Abrimos un cursor para recorrer los objetos
let request = store.openCursor();

request.onsuccess = function(event) {
    let cursor = event.target.result;

    if (cursor) {
        // Si encontramos un usuario con el nombre "Juan"
        if (cursor.value.nombre === "Juan") {
            cursor.delete();  // Eliminamos este objeto
            console.log("Usuario eliminado:", cursor.value);
        }
        cursor.continue(); // Seguimos buscando más coincidencias
    }
};
```

🔹 **¿Qué hace este código?**

1. Abre una transacción de lectura-escritura en `usuarios`.
2. Usa un cursor para recorrer todos los registros.
3. Si encuentra un usuario con `nombre === "Juan"`, lo **borra con** `cursor.delete()`.
4. Continúa buscando más coincidencias (por si hay varios "Juan").