**Ejemplo completo Indexed DB**

Aquí tienes un ejemplo completo de cómo abrir una base de datos con **IndexedDB** y realizar operaciones **CRUD** (Crear, Leer, Actualizar y Eliminar). Utilizaremos un almacén llamado "productos" para simular una gestión de inventario en una tienda.

**Código completo: Operaciones CRUD con IndexedDB**

// Abrir o crear la base de datos

const request = indexedDB.open("MiTienda", 1);

request.onsuccess = function (event) {

console.log("Base de datos abierta exitosamente.");

const db = event.target.result;

// Realizar operaciones CRUD

agregarProducto(db, { nombre: "Laptop", precio: 1500, cantidad: 10 });

leerProductos(db);

actualizarProducto(db, 1, { precio: 1400 }); // Cambiar el precio del producto con id = 1

eliminarProducto(db, 1); // Eliminar el producto con id = 1

};

request.onerror = function (event) {

console.error("Error al abrir la base de datos:", event.target.error);

};

request.onupgradeneeded = function (event) {

console.log("Actualización de la base de datos en curso...");

const db = event.target.result;

// Crear un almacén de objetos si no existe

if (!db.objectStoreNames.contains("productos")) {

const objectStore = db.createObjectStore("productos", { keyPath: "id", autoIncrement: true });

objectStore.createIndex("nombre", "nombre", { unique: false });

objectStore.createIndex("precio", "precio", { unique: false });

console.log("Almacén 'productos' creado.");

}

};

**Funciones CRUD**

**1. Crear (Agregar datos)**

function agregarProducto(db, producto) {

const transaction = db.transaction("productos", "readwrite");

const store = transaction.objectStore("productos");

const request = store.add(producto);

request.onsuccess = function () {

console.log("Producto agregado:", producto);

};

request.onerror = function (event) {

console.error("Error al agregar producto:", event.target.error);

};

}

**2. Leer (Obtener todos los datos)**

function leerProductos(db) {

const transaction = db.transaction("productos", "readonly");

const store = transaction.objectStore("productos");

const request = store.getAll();

request.onsuccess = function () {

console.log("Productos en la base de datos:", request.result);

};

request.onerror = function (event) {

console.error("Error al leer productos:", event.target.error);

};

}

**3. Actualizar (Modificar un dato existente)**

function actualizarProducto(db, id, cambios) {

const transaction = db.transaction("productos", "readwrite");

const store = transaction.objectStore("productos");

const request = store.get(id);

request.onsuccess = function () {

const producto = request.result;

if (!producto) {

console.error(`No se encontró el producto con id ${id}`);

return;

}

// Aplicar cambios

Object.assign(producto, cambios);

// Guardar el producto actualizado

const updateRequest = store.put(producto);

updateRequest.onsuccess = function () {

console.log("Producto actualizado:", producto);

};

};

request.onerror = function (event) {

console.error("Error al buscar el producto para actualizar:", event.target.error);

};

}

**4. Eliminar (Borrar un dato)**

function eliminarProducto(db, id) {

const transaction = db.transaction("productos", "readwrite");

const store = transaction.objectStore("productos");

const request = store.delete(id);

request.onsuccess = function () {

console.log(`Producto con id ${id} eliminado.`);

};

request.onerror = function (event) {

console.error("Error al eliminar producto:", event.target.error);

};

}

**Cómo funciona el código**

1. **Inicialización:**
   * Al abrir la base de datos por primera vez, se crea el almacén "productos" con una clave primaria (id) y dos índices (nombre y precio).
2. **CRUD:**
   * **Crear:** Agrega un nuevo producto con datos como nombre, precio y cantidad.
   * **Leer:** Obtiene todos los productos almacenados.
   * **Actualizar:** Busca un producto por su id y aplica los cambios necesarios.
   * **Eliminar:** Borra un producto del almacén por su id.

**Ejemplo de salida en la consola**

Si ejecutas este código, podrías ver algo como esto en la consola:

Base de datos abierta exitosamente.

Producto agregado: { nombre: "Laptop", precio: 1500, cantidad: 10 }

Productos en la base de datos: [{ id: 1, nombre: "Laptop", precio: 1500, cantidad: 10 }]

Producto actualizado: { id: 1, nombre: "Laptop", precio: 1400, cantidad: 10 }

Producto con id 1 eliminado.