



IndexedDB

Rafael Bernal Romero
José Luis Narváez Salazar



¿Para qué sirve?

IndexedDB es una base de datos que nos permite almacenar información del lado del cliente en un navegador, y además de manera persistente.

- Se trata de una base de datos clave/valor.
- No es una base de datos relacional.
- Trabaja de manera asíncrona.

Compatibilidad con navegadores

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			62		10.2				
		57	63		10.3				4
11	16	58	64	11	11.2	all	64	11.8	6.2
	17	59	65	11.1	11.3				
		60	66	TP					
		61	67						



¿Es una tecnología nueva o sustituye a alguna anterior?

Se plantea como posible sustituto de WebSQL.

WebSQL fue marcada como obsoleta por W3C en 2010. Los dos son soluciones para el almacenamiento, pero no ofrecen las mismas funcionalidades.

WebSQL tiene un límite de almacenamiento, mientras que IndexedDB tiene un almacenamiento ilimitado.



¿Cómo funciona?

IndexedDB nos permite almacenar y obtener objetos indizados a partir de una *clave*.

Todos los cambios realizados en la base de datos ocurren dentro de transacciones.

Las búsquedas en la base de datos se hacen a través de la clave, o a través de índices para buscar por otras “columnas”.



Principales métodos y atributos

Para comenzar a usar IndexedDB debemos crear un objeto que implemente la interfaz IDBFactory.

```
var indexedDB = window.indexedDB || window.mozIndexedDB ||  
                window.webkitIndexedDB || window.msIndexedDB ||  
                window.shimIndexedDB;
```



Principales métodos y atributos

IDBFactory

- `.open(nombre, [versión])`
- `.deleteDatabase(nombre)`

```
var open = indexedDB.open('Instituto', 1);
```



Principales métodos y atributos

IDBOpenDBRequest

Eventos

- upgradeneeded
- success

Propiedades

- .result



Principales métodos y atributos

IDBOpenDBRequest

```
open.onupgradeneeded = function() {  
    // Se ejecuta la primera vez que se crea la BD, y cada vez  
    // que se cambia la versión en el método .open()  
    var db = open.result; // Devuelve una conexión a la BD  
};  
  
open.onsuccess = function() {  
    // Se ejecuta cuando se ha realizado la conexión con  
    // la base de datos correctamente.  
};
```



Principales métodos y atributos

IDBDataBase

- `.createObjectStore(nombre, [opciones])`

```
var alum = db.createObjectStore('Alumnos', {keyPath: 'id', autoIncrement: true});
```

- `.transaction(nombre, modo)` [readonly, readwrite]

```
var tx = db.transaction('Alumnos', 'readwrite');
```



Principales métodos y atributos

IDBObjectStore

- `.createIndex(nombreIndice, columna)`

```
alum.createIndex('idxNombre', ['nombre.segundo', 'nombre.primerio']);
```

- `.index(nombreIndice)`

```
var index = store.index('idxNombre');
```



Principales métodos y atributos

IDBObjectStore

- `.put(objeto)`

```
store.put({  
  id: 1,  
  nombre: {  
    primero: 'Juan',  
    segundo: 'Ramírez'  
  },  
  genero: 'Hombre',  
  edad: 22  
});
```



Principales métodos y atributos

IDBObjectStore

- `.delete(id)`

```
store.delete(3);
```

- `.get(id)`

```
var query1 = store.get(1);
```

- `.add()`



Principales métodos y atributos

IDBTransaction

- `.objectStore(nombre)`

```
var store = tx.objectStore('Alumnos');
```

- `complete`

```
tx.oncomplete = function() {  
    db.close();  
};
```



Principales métodos y atributos

IDBIndex

- `.get(valor)`

```
var query2 = index.get(['González', 'Claudia']);
```



Principales métodos y atributos

IDBRequest

- success

```
query1.onsuccess = function() {  
    if (this.result !== undefined) {  
        console.log(this.result.nombre.segundo);  
    }  
};
```




Actividad

Crea una base de datos llamada 'Liga', donde se recogerán 'Equipos' y tendrán como datos *'id'*, *'nombre'*, *'liga'* y *'trofeos'*.

- Guardar nuevo equipo.
- Consultar equipo.
- Eliminar equipo.

ANEXOS





Webgrafía

- <http://blog.koalite.com/2015/10/introduccion-a-indexed-db/>
- <https://developer.mozilla.org/es/docs/IndexedDB-840092-dup>



Plugin de jQuery

```
/* before */
var request = window.indexedDB.open("BookShop-1");
request.onsuccess = function(event){
    var db = request.result;
    var transaction = db.transaction(["BookList"], IDBTransaction.READ_WRITE);
    var objectStore = transaction.objectStore("BookList");
    var request = DAO.objectStore.openCursor();
    request.onsuccess = function(event){
        var cursor = request.result;
        if (cursor) {
            write(cursor.key + " " + cursor.value);
            cursor["continue"]();
        }
    };
};

/* after */
$.indexeddb("BookShop-1").objectStore("BookList").openCursor().each(write);
```

[Enlace al plugin](#)