



CREANDO CRUD EN INDEXEDDB EN JS



¿POR QUÉ INDEXEDDB?

- ALMACENA GRANDES CANTIDADES DE DATOS DE FORMA SEGURA Y EFICAZ.
- FUNCIONA DE FORMA ASÍNCRONA, POR LO QUE ES ADECUADO PARA APLICACIONES WEB.
- ADMITE TIPOS DE DATOS COMPLEJOS, COMO ARCHIVOS Y BLOQUES.
- OFRECE AMPLIAS POSIBILIDADES DE CONSULTA.
- FUNCIONA DENTRO DEL NAVEGADOR DEL USUARIO, POR LO QUE OFRECE CAPACIDADES OFFLINE.



0

```
const idb =  
  window.indexedDB ||  
  window.mozIndexedDB ||  
  window.webkitIndexedDB ||  
  window.msIndexedDB ||  
  window.shimIndexedDB;
```

PASOS



CONFIGURACIÓN DE INDEXEDDBNT0

Para trabajar con IndexedDB, primero hay que abrir una conexión con una base de datos. Esto implica especificar el nombre de la base de datos y la versión. Si la base de datos no existe, se creará



CREACIÓN DE ALMACENES DE OBJETOS Y TRANSACCIONES

Una vez establecida la conexión, puede crear almacenes de objetos (similares a las tablas de las bases de datos relacionales) y gestionar los datos mediante transacciones.



OPERACIONES CRUD

- create
- Read
- Update
- Delete

1

```
const request = indexedDB.open('myDatabase', 1);

request.onerror = function(event) {
  console.error('Database error:', event.target.error);
};

request.onsuccess = function(event) {
  console.log('Database opened successfully');
};
```

2

```
request.onupgradeneeded = function(event) {  
  const db = event.target.result;  
  const store = db.createObjectStore('notes', {keyPath: 'id', autoIncrement: true});  
  // Create indices if needed...  
};
```

3.1

Creating Items

```
function createNote(db, note) {  
  const transaction = db.transaction(['notes'], 'readwrite');  
  const store = transaction.objectStore('notes');  
  const request = store.add(note);  
  
  request.onsuccess = function() {  
    console.log('Note added:', request.result);  
  };  
  
  request.onerror = function() {  
    console.error('Error adding note:', request.error);  
  };  
}
```

3.2

Reading Items

```
function readNote(db, id) {  
  const transaction = db.transaction(['notes'], 'readonly');  
  const store = transaction.objectStore('notes');  
  const request = store.get(id);  
  
  request.onsuccess = function() {  
    console.log('Note read:', request.result);  
  };  
}
```


3.3

Updating Items

```
function updateNote(db, note) {  
  const transaction = db.transaction(['notes'], 'readwrite');  
  const store = transaction.objectStore('notes');  
  const request = store.put(note);  
  
  request.onsuccess = function() {  
    console.log('Note updated:', request.result);  
  };  
}
```

3.4

Deleting Items

```
function deleteNote(db, id) {  
  const transaction = db.transaction(['notes'], 'readwrite');  
  const store = transaction.objectStore('notes');  
  const request = store.delete(id);  
  
  request.onsuccess = function() {  
    console.log('Note deleted');  
  };  
}
```



UNAH-CURC

Lizzy Pamela