

Luis Villalba Palacios

1ºASIR A

Gestión de bases de datos

Proyecto final:

Usuarios de una asociación de índole social

Índice

Valores usados	3
Find	5
Aggregate	6
Update y updateMany	7
Búsquedas geoespaciales	8

Valores usados

Mi objetivo es crear una base de datos que se aproxime a la realidad de los datos que se registran en las bases de datos de servicios sociales o asociaciones de índole social. Al tratarse de un proyecto con finalidad académica estos datos se han reducido para evitar campos con valores del mismo tipo, y así poder trabajar con diferentes tipos de búsquedas y opciones que nos ofrece MongoDB.

- String.
- Número.
- Fecha.
- Array.
- Array de documentos que incluyen un string y una fecha.
- Documento.
- Booleano.
- Documento de booleanos.
- Coordenadas.

Cada documento incluye los siguientes campos:

- **_id**: Número que le asignamos en nuestra base de datos.
- **nombre**
- **fechaNacimiento**
- **idioma**: Es un array que puede incluir varios idiomas.
- **direccion**: Incluye un documento con los campos de comunidad y provincia.
- **hijos**: Array de documentos que incluyen los campos de nombre y fecha. Este campo también puede ser 0, en caso de que el usuario no tenga hijos.
- **ingresosMensuales**
- **inscritoSEPE**: Booleano que nos indica si el usuario está registrado en el Servicio Público de Empleo Estatal.
- **discapacidad**: Incluye el porcentaje reconocido de discapacidad que tiene la persona. Este campo sólo aparece si la persona tiene discapacidad.
- **profesion**: Array que incluye la profesión/profesiones de la persona.
- **programas**: Documento de dos campos (“insercion” y “escuelaPadres” con valores booleanos).

- **visitaEducador:** Array de números que indican los días del mes en los que el educador social les visita.
- **relevante:** En este campo se incluye información relevante del usuario.
- **ayudaMensual:** Campo numérico que incluye la cantidad de ayuda económica que tiene la persona.
- **coordenadas:** Documento que nos permite utilizar las funciones de geolocalización de MongoDB, incluye el campo type: que en nuestro caso el valor es "Point", pero también puede ser un polígono formado por varias coordenadas. Y un campo "coordinates" el cual es un array que incluye la longitud y la latitud.

Find

En el documento “consultas.js” se incluyen diferentes enunciados con sus correspondientes soluciones. El objetivo es realizar consultas complejas sobre nuestra colección “usuarios.js”, utilizando operadores vistos en clase, y también la inclusión de nuevos operadores.

Operadores usados

- **\$eq:** Busca valores que sean iguales al valor que se especifique.
- **\$gt:** Busca valores que sean mayores al valor que se especifique.
- **\$gte:** Busca valores que sean iguales o mayores al valor que se especifique.
- **\$lt:** Busca valores que sean menores al valor que se especifique.
- **\$lte:** Busca valores que sean iguales o menores al valor que se especifique.
- **\$in:** Busca valores que incluyan alguno de los valores especificados en el array.
- **\$and:** Sirve para agrupar búsquedas, nos devuelve los documentos que cumplan todas.
- **\$or:** Agrupa búsquedas, nos devuelve los documentos que cumplan una u otra.
- **\$exists:** Busca aquellos documentos que tengan ese campo.
- **\$regex:** Selecciona documentos donde los valores coinciden con la expresión especificada. Se incluyen las opciones “m” (busca en cualquier posición sin importar los saltos de carro) e “i” (No importa si está en mayúsculas o minúsculas).

Aggregate

Este operador nos permite añadir campos a una colección de documentos, especificando el nombre de los campos nuevos a agregar, cada uno con una expresión que generará el valor del campo nuevo para cada documento.

Operadores usados

- **\$match:** Filtra los documentos para que solo se incluyan los documentos que coincidan.
- **\$group:** Agrupa los documentos con `_id` (obligatorio). Para cada una de las agrupaciones se genera un documento.
- **\$sum:** Calcula y devuelve la suma de los valores numéricos del campo que se indique.

Update y updateMany

Este tipo de operaciones son las encargadas de actualizar documentos en las bases de datos MongoDB. Se trata de una funcionalidad muy importante, ya que nos ahorra tener que eliminar un documento de nuestra base de datos y volver a introducirlo con la modificación correspondiente, en su lugar, utilizando el operador update actualizamos directamente la parte o partes que queramos de nuestro documento o documentos.

Operadores usados

- **\$inc:** Incrementa un campo por un valor especificado.
- **\$mul:** Multiplica un campo por un valor especificado.
- **\$unset:** Elimina un campo especificado.
- **\$addtoSet:** Añade un valor a un array, a menos que ya exista dicho valor.
- **\$rename:** Modifica el nombre del campo que indiquemos en toda nuestra colección.

Búsquedas geoespaciales

MongoDB permite trabajar con coordenadas, para ello utiliza documentos tipo GeoJSON. En este proyecto vamos a realizar una pequeña introducción a estas nuevas búsquedas.

En la colección “usuarios.js” existe el campo “coordenadas”, el cual es un documento que incluye el campo type: que en nuestro caso el valor es "Point", ya que especifica un punto en concreto en dos dimensiones, pero también puede ser un polígono formado por varias coordenadas. Y un campo “coordinates” el cual es un array que incluye la longitud y la latitud.

En primer lugar debemos de crear un Index con las coordenadas de nuestra colección, vamos a trabajar con “2dsphere” puesto que es más exacta que “2d”, ya que deforma el plano 2d teniendo en cuenta la forma esférica de la tierra.

- 1) Ejecutamos en mongo el comando:

```
db.usuarios.createIndex( { coordenadas : "2dsphere" } )
```

- 2) Reiniciamos mongo para que se actualice todo debidamente.

Operadores usados

- **\$near:** Especifica un punto y realiza una consulta geoespacial que devuelve los documentos ordenados del más cercano al más lejano.
- **\$geometry:** Especifica una geometría GeoJSON para usar con los operadores de consulta geoespacial.
- **\$maxDistance:** Especifica en metros la distancia máxima para limitar la búsqueda.
- **\$minDistance:** Especifica en metros la distancia mínima para limitar la búsqueda.