Facultad de Informática – UCM –
NoSQL - Práctica 4

1) Usando Google Maps encuentra las coordenadas terrestres de "UCM Biblioteca María Zambrano", y completa su longitud y latitud (los valores que aparecen después de la @ en la url, pero dados la vuelta), las instrucciones de inserción siguiente (ver fichero *lugares.txt*):

```
db.lugares.drop()
db.lugares.insert({nombre:"Informática-UCM", tipo:"Facultad", pais:"España",
                   posicion: {type:"Point", coordinates:[-3.7353797,40.450305]}})
db.lugares.insert({nombre:"Metropolitano", tipo:"metro", pais:"España",
                   posicion: {type:"Point", coordinates:[-3.7202654,40.4465915]}})
db.lugares.insert({nombre:"Ciudad Universitaria", tipo:"metro", pais:"España",
                  posicion: {type:"Point", coordinates:[-3.7289768,40.4435602]}})
db.lugares.insert({nombre:"Pabellón de Plata", tipo:"templo", pais:"Japón",
                   posicion: {type:"Point", coordinates:[135.7982058,35.0270213]}})
db.lugares.insert({nombre:"María Zambrano", tipo:"bibliotecar", pais:"España",
                   posicion: {type:"Point", coordinates:[????long,????lat]}})
```

Y crear un índice tipo 2d esférico sobre el campo "posicion"

Escribir en la solución:
- Comando insert del lugar del que se han buscado las coordenadas (solo de ese)
- Instrucción para crear el índice y respuesta del shell

2.- Encontrar todos los objetos de la colección lugares que estén a más de 1500 metros de las coordenadas: lat: 40.4381963,long : -3.7251149 (museo de américa). Para cada lugar se debe mostrar solo el nombre y el tipo.

3 Ahora vamos a utilizar puntos en el plano. Para ello basta con definir documentos que tengaun una clave con las dos coodenadas {….,punto: [0,0] ….}.

Introduce en el shell de MongoDB la siguiente instrucción que añade a la colección "plano" unos cuantos puntos (fichero lugares.txt):

```
use pract2d

db.plano.drop()

for (var i=0; i<500; i++) { for (var j=0; j<=i; j++)    { db.plano.insert({desc:"("+i+","+j+")",
punto:[i,j]});}}
```

Crear un índice sobre la clave "punto" con nombre "ipunto" (un índice normal, no un geoespacial), y otro sobre la clave "desc" con nombre "idesc" (de nuevo un índice normal).

Comprobar el tamaño en bytes de los dos índices con db.plano.stats(). Copia en la solución las instrucciones para crear los índices.

Indica también por qué crees que el de mayor tamaño ocupa más que el otro (una frase debe bastar)

4) Queremos contar el número de elementos que tienen la primera coordenada mayor de 480.

Copiarlo en la solución. **Ayuda**: buscar en la parte dedicada a CRUD, dentro de “find en detalle” operadores que actúan sobre arrays.

5) Queremos probar la consulta: db.plano.find({punto:{$lt :2}},{desc:1,_id:0})

¿Crees que utilizará algún índice? ¿Cuál?

6) MongoDB permite preguntar por aquellos documentos que están dentro de la zona determinada por un polígono. Por ejemplo, podemos preguntar por los puntos que están dentro del cuadrado especificado por las coordenadas `[ 0 , 0 ], [ 3 , 0 ], [ 3 , 3 ], [0,3]`

Escribe en la solución la respuesta.

7) La consulta anterior no utiliza ningún índice. Escribe en la solución qué indice crearías para mejorar la eficiencia.

8) ¿Se puede crear el índice? ¿Por qué?

9) Supón que con los índices existentes se quiere ejecutar la consulta:

db.plano.find({punto:{$lt :2}, desc:{$gt:"(5"}},{desc:1,_id:0})

¿Te parece que podría crearse un índice que mejore el rendimiento de esta consulta?

Si la respuesta es no ¿por qué?

10) Queremos encontrar la distancia aproximada a la que se encuentra la biblioteca María Zambrano del Museo de América, con un error de más/menos 100 metros. Escribir un fragmento de código que nos indique esa distancia (sin programar, usando Mongo)