

EJERCICIO

1. Creamos una BD marcador.

```
test> use marcador
switched to db marcador
marcador> db.createCollection("puntuaciones")
{ ok: 1 }
marcador>
```

2. Vamos a insertar algunas puntuaciones de prueba para trabajar con ellas.

```
marcador> db.puntuaciones.find()
[
  { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true },
  { _id: 2, usuario: 'pablito', puntos: 80, finalizado: false },
  { _id: 3, usuario: 'paulina', puntos: 30 }
]
marcador>
```

3. Consultas

- a. Listar todos los documentos de la colección, que la salida se muestre estructurada para facilitar la lectura.

```
marcador> db.puntuaciones.find().pretty()
[
  { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true },
  { _id: 2, usuario: 'pablito', puntos: 80, finalizado: false },
  { _id: 3, usuario: 'paulina', puntos: 30 }
]
marcador>
```

- b. Listar alguno de los documentos de la colección, para mostrar la estructura de los documentos que contiene.

```
marcador> db.puntuaciones.find({"_id":1}).pretty()
[ { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true } ]
```

- c. Listar los elementos por valor de sus atributos: filtrar por el nombre de usuario "pepegrillo".

```
marcador> db.puntuaciones.find({"usuario":"pepegrillo"})
[ { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true } ]
marcador>
```

- d. Listar por rango de valores: en este caso documentos donde el puntos sea mayor que 20 y menor o igual a 50.

```

marcador> db.puntuaciones.find({$and:[{puntos:{$gt:20}},{puntos:{$lte:50}}]})
[
  { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true },
  { _id: 3, usuario: 'paulina', puntos: 30 }
]
marcador>

```

- e. Listar elementos que no incluyen (o que incluyen) un atributo. En este caso listamos el documento que no tiene el atributo finalizado.

```

marcador> db.puntuaciones.find({finalizado:{$exists:false}})
[ { _id: 3, usuario: 'paulina', puntos: 30 } ]
marcador>

```

- f. Listar elementos usando expresiones regulares: cuyo usuario contenga la cadena uli

```

marcador> db.puntuaciones.find({"usuario":/uli/})
[ { _id: 3, usuario: 'paulina', puntos: 30 } ]
marcador>

```

- g. Hacer una agrupación de condiciones, en este caso elementos cuyo _id sea 1 o 2

```

marcador> db.puntuaciones.find({$or:[{"_id":1},{"_id":2}]})
[
  { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true },
  { _id: 2, usuario: 'pablito', puntos: 80, finalizado: false }
]
marcador>

```

- h. Contar el número de documentos de la colección.

```

marcador> db.puntuaciones.find().count()
3
marcador>

```

- i. Contar el número de documentos de la colección con el filtro que esté finalizado.

```

marcador> db.puntuaciones.find({"finalizado":true}).count()
1
marcador>

```

4. Consultar por coincidencias en un array:

- a. Insertar un par de documentos un poco más ricos en elementos

```

marcador> db.puntuaciones.insertMany([{"_id":4,usuario:"cocinero",favoritos:["cocinar","postres"],direccion:{pais:"es",ciudad:"madrid"}},{_id:5,usuario:"deportista",favoritos:["deportes","viajes"],direccion:{pais:"es",ciudad:"madrid"}}])
marcador>
{ acknowledged: true, insertedIds: { '0': 4, '1': 5 } }

```

- b. listamos los documentos que tengan como favoritos "postres".

```

marcador> db.puntuaciones.find({"favoritos":"postres"})
[
  {
    _id: 4,
    usuario: 'cocinero',
    favoritos: [ 'cocinar', 'postres' ],
    direccion: { pais: 'es', ciudad: 'madrid' }
  }
]
marcador>

```

- c. Obtener los documentos que contengan en el array todos los elementos “postres” y “cocinar”.

```

marcador> db.puntuaciones.find({$and:[{"favoritos":"postres"}, {"favoritos":"cocinar"}]})
[
  {
    _id: 4,
    usuario: 'cocinero',
    favoritos: [ 'cocinar', 'postres' ],
    direccion: { pais: 'es', ciudad: 'madrid' }
  }
]
marcador>

```

- d. Listar los documentos que contengan en el array “cocinar”.

```

marcador> db.puntuaciones.find({"favoritos":"cocinar"})
[
  {
    _id: 4,
    usuario: 'cocinero',
    favoritos: [ 'cocinar', 'postres' ],
    direccion: { pais: 'es', ciudad: 'madrid' }': 5 } }
  }
]
marcador>

```

- e. Listando los elementos por valor de alguno de los atributos del documento embebido. En este caso, documentos cuya ciudad es madrid.

```

marcador> db.puntuaciones.find({"direccion.ciudad":"madrid"})
[
  {
    _id: 4,
    usuario: 'cocinero',
    favoritos: [ 'cocinar', 'postres' ],
    direccion: { pais: 'es', ciudad: 'madrid' }
  }
]
marcador>

```

- f. Obtener los usuarios con las dos mejores puntuaciones.

```
marcador> db.puntuaciones.find().sort({"puntos":-1}).limit(2)
[
  { _id: 2, usuario: 'pablito', puntos: 80, finalizado: false },
  { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true }
]
marcador>
```

- g. Ordenar los resultados por el valor de sus atributos. Para utilizar el orden natural, hay que utilizar el 1 positivo, y para invertir los resultados, un 1 negativo. Ordenar de mayor a menor puntuación.

```
marcador> db.puntuaciones.find().sort({"puntos":-1})
[
  { _id: 2, usuario: 'pablito', puntos: 80, finalizado: false },
  { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true },
  { _id: 3, usuario: 'paulina', puntos: 30 },
  {
    _id: 4,
    usuario: 'cocinero',
    favoritos: [ 'cocinar', 'postres' ],
  },
  {
    _id: 5,
    usuario: 'deportista',
    favoritos: [ 'running' ],
    direccion: { pais: 'es', ciudad: 'toledo' }
  }
]
marcador>
```

5. Actualización de documentos en MongoDB

- a. Actualizar los puntos de pepegrillo a 60.

```
marcador> db.puntuaciones.updateOne({"usuario":"pepegrillo"},{$set:{"puntos":60}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
marcador>

marcador> db.puntuaciones.find({"usuario":"pepegrillo"})
[ { _id: 1, usuario: 'pepegrillo', puntos: 60, finalizado: true } ]
marcador>
```

- b. Incrementa las puntuaciones de pepegrillo en 1 punto

```

marcador> db.puntuaciones.updateOne({"usuario":"pepegrillo"},{$inc:{"puntos":1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
marcador>

```

```

marcador> db.puntuaciones.find({"usuario":"pepegrillo"})
[ { _id: 1, usuario: 'pepegrillo', puntos: 61, finalizado: true } ]
marcador>

```

- c. Actualización masiva de la propiedad finalizado (true) a todos los documentos.

```

marcador> db.puntuaciones.updateMany({},{$set:{"finalizado":true}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 4,
  upsertedCount: 0
}
marcador>

```

```

marcador> db.puntuaciones.find()
[
  { _id: 1, usuario: 'pepegrillo', puntos: 61, finalizado: true },
  { _id: 2, usuario: 'pablito', puntos: 80, finalizado: true },
  { _id: 3, usuario: 'paulina', puntos: 30, finalizado: true },
  {
    _id: 4,
    usuario: 'cocinero',
    favoritos: [ 'cocinar', 'postres' ],
    direccion: { pais: 'es', ciudad: 'madrid' },
    finalizado: true
  },
  {
    _id: 5,
    usuario: 'deportista',
    favoritos: [ 'running' ],
    direccion: { pais: 'es', ciudad: 'toledo' },
    finalizado: true
  }
]
marcador>

```