

UD3.T3 Operaciones CRUD. BD Marcador

Consultas

- 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 }
]
```

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

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

- 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 } ]
```

- 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({ puntos : { $gt : 20, $lte: 50 } })
[
  { _id: 1, usuario: 'pepegrillo', puntos: 50, finalizado: true },
  { _id: 3, usuario: 'paulina', puntos: 30 }
]
```

- 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 } ]
```

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

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

- 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 }
]
```

- Contar el número de documentos de la colección.

```
marcador> db.puntuaciones.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
3
```

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

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

Trabajando con estructuras anidadas y arrays

- Consultar por coincidencias en un array: Listamos los documentos que tengan como favoritos “postres”.

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

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

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

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

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

- 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" }).pretty()
[
  {
    _id: 4,
    usuario: 'cocinero',
    favoritos: [ 'cocinar', 'postres' ],
    direccion: { pais: 'es', ciudad: 'madrid' }
  }
]
```

- 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 }
]
```

- Ordenar de mayor a menor puntuación.

```
marcador> db.puntuaciones.find().sort({puntos:-1}).pretty()
[
  { _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' ],
    direccion: { pais: 'es', ciudad: 'madrid' }
  },
  {
    _id: 5,
    usuario: 'deportista',
    favoritos: [ 'running' ],
    direccion: { pais: 'es', ciudad: 'toledo' }
  }
]
```

Actualización de documentos en MongoDB

- 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
}
```

- 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
}
```

- 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
}
```