Bases de datos

Gabriel Rodríguez Flores

January 15, 2026

ADA ITS
INSTITUTO TECNOLÓGICO SUPERIOR

  - Teoría sobre tipos de bases de datos
- Robo3t/Studio3t/MongoDBCompass/VSCodeExt como herramienta técnica
- MongoDB Atlas
- Queries básicas

Contents

1 Teoría

- BSON Data Types: ObjectId

1.1 Instalación

Docker

1.2 Básico

1.2.1 Base de datos

```
use mydb
```

1.2.2 Colecciones

```
db.createCollection('collection_name');
db.collection_name.insert(document); // Implicit*
```

*Si no existe la colección, se crea automáticamente.

1.2.3 CRUD

1.2.3.1 Find

```
db.collection_name.find();
db.collection_name.find(query);
db.collection_name.findOne(query);
```

1.2.3.2 Insert

```
db.collection_name.insert(document);
db.collection_name.insertMany([documents]);
```

> **Nota**: Si no se especifica el `_id`, se genera automáticamente.

1.2.3.3 Update

```
db.collection_name.update(query, update, options);
db.collection_name.updateMany(query, update, options);
```

1.2.3.4 Remove

```
db.collection_name.remove(query);
db.collection_name.deleteOne(query);
db.collection_name.deleteMany(query);
```

1.3 Query

1.3.1 $eq

```
db.collection_name.find({ key: 'value'});
db.collection_name.find({ { $eq: key: 'value' } });
```

> **Nota**: Si no se especifica el operador, se asume que es `$eq`.

1.3.2 $neq

```
db.collection_name.find({ key: { $neq: 'value'}});
```

1.3.3 $gt / $gte / $lt / $lte

```
db.collection_name.find({ key: { $gt: 'value'}});
db.collection_name.find({ key: { $gte: 'value'}});
db.collection_name.find({ key: { $lt: 'value'}});
db.collection_name.find({ key: { $lte: 'value'}});
db.collection_name.find({ key: { $gt: 'value', $lt: 'value'}});
db.collection_name.find({ key: { $gte: 'value', $lte: 'value'}});
```

1.3.4 $in / $nin

```
db.collection_name.find({ key: { $in: ['value1', 'value2']}});
db.collection_name.find({ key: { $nin: ['value1', 'value2']}});
```

> **Nota**: `$in` y `$nin` son equivalentes a `OR` y `NOT IN` en SQL.

1.3.5 $exists

```
db.collection_name.find({ key: { $exists: true}});
db.collection_name.find({ key: { $exists: false}});
```

> **Tip**: Se puede usar para comprobar la longitud de un array.

```
db.collection_name.find({ 'array.1': { $exists: true}}); // array.length >
    1
```

1.3.6 $type

```
db.collection_name.find({ key: { $type: 'string'}});
db.collection_name.find({ key: { $type: 'number'}});
db.collection_name.find({ key: { $type: 'array'}});
db.collection_name.find({ key: { $type: 'object'}});
```

1.4 Advanced Query

1.4.1 $regex

```
db.collection_name.find({ key: { $regex: /value/}});
db.collection_name.find({ key: { $regex: /value/i}});
db.collection_name.find({ key: { $regex: /value/, $options: 'i'}});
```

1.4.2 $and / $or / $not

```
db.collection_name.find({ $and: [{ key1: 'value1'}, { key2: 'value2'}]});
db.collection_name.find({ $or: [{ key1: 'value1'}, { key2: 'value2'}]});
db.collection_name.find({ $not: { key: 'value'}});
```

1.4.3 $all / $elemMatch

```
db.collection_name.find({ key: { $all: ['value1', 'value2']}});
db.collection_name.find({ key: { $elemMatch: { key: 'value'}}});
```

> elemMatch buscará en el array el documento que cumpla con la query, de lo contrario

1.4.4 $size

```
db.collection_name.find({ key: { $size: 2}});
```

1.5 Métodos básicos

1.5.1 count

```
db.collection_name.count();
db.collection_name.count(query);
```

1.5.2 sort

```
db.collection_name.find().sort({ key: 1 }); // ASC
db.collection_name.find().sort({ key: -1 }); // DESC
```

1.5.3 limit

```
db.collection_name.find().limit(10);
```

1.5.4 skip

```
db.collection_name.find().skip(10);
```

1.5.5 distinct

Sirve para obtener valores únicos de un campo.

```
db.collection_name.distinct('key');
```

1.5.6 aggregate

> Siguiente clase

2 Ejemplos

- Ejemplos de queries

3 Ejercicios

- Búsqueda de elementos en array in vs all
- Búsqueda de arrays con elementos (>0)
- Contar todos los documentos que le falten el parámetro `name` dentro del parámetro `user`.
  - Solución: `db.getCollection('documents').count({ 'user.name': { $exists: false}})`

4 Entregables

4.1 En clase

4.2 Tarea