

Máster en Big Data

Tecnologías de Almacenamiento

12. Hands-On: MongoDB

Presentado por: José David Angulo y Albert Ripoll

Índice

1. Introducción	3
2. Mongo DB	3
2.1. Instalación de MongoDB Compass	3
2.2. Creación de base de datos y colección	3
2.3. Inserción y consulta de datos	4
2.4. Actualización y eliminación de datos	13
2.5. Manejo colecciones bd.....	23

1. Introducción

El objetivo de este Hands-On es familiarizarse con la utilización de Mongo, tanto en la creación de documentos como en la realización de consultas.

2. Mongo DB

2.1. Instalación de MongoDB Compass

<https://www.mongodb.com/try/download/community>

2.2. Creación de base de datos y colección

- Muestra todas las bases de datos en MongoDB.

```
> show databases
< admin    40.00 KiB
  config   60.00 KiB
  local    40.00 KiB
  prueba    8.00 KiB
```

- Crea una nueva base de datos y llámala "p12".

```
> use p12
< switched to db p12
p12>
```

- Mira en qué base de datos estás.

```
> db
< p12
p12>
```

- Crea una colección en la base de datos llamada "books".

```

> use p12
< switched to db p12
> db.createCollection("books")
< { ok: 1 }
p12>

```

2.3. Inserción y consulta de datos

- Inserta un libro con los siguientes campos: nombre, autor, editorial, precio y cantidad.

```

> db.books.insertOne({
  "nombre": "El viejo y el mar",
  "autor": "Rafael Pombo",
  "editorial": "Cuervana",
  "precio": 12,
  "cantidad": 850
})
< {
  acknowledged: true,
  insertedId: ObjectId('66426944816fe65e46dbdee0')
}

```

- Muestra el resultado de la inserción y explica los dos campos y su significado.

```

> db.books.findOne({ "nombre": "El viejo y el mar" })
< {
  _id: ObjectId('66426944816fe65e46dbdee0'),
  nombre: 'El viejo y el mar',
  autor: 'Rafael Pombo',
  editorial: 'Cuervana',
  precio: 12,
  cantidad: 850
}
p12>

```

Tenemos el nombre del libro, el autor del libro, la editorial, el precio del libro, la cantidad de ejemplares que se tienen para la venta.

- Verifica que se haya creado la partición en MongoDB Compass.

```
> show collections
< books
> db.books.find()
< {
  _id: ObjectId('66426944816fe65e46dbdee0'),
  nombre: 'El viejo y el mar',
  autor: 'Rafael Pombo',
  editorial: 'Cuervana',
  precio: 12,
  cantidad: 850
}
p12>
```

- Intenta insertar otro documento con el mismo `_id` que se le ha asignado al primer libro. Explica qué pasa y por qué.

```
> db.books.insertOne({
  "_id": ObjectId("66426944816fe65e46dbdee0"),
  "nombre": "El viejo y el mar",
  "autor": "Rafael Pombo",
  "editorial": "Cuervana",
  "precio": 12,
  "cantidad": 850
})
MongoServerError: E11000 duplicate key error collection: p12.books index: _id_ dup key: { _id: ObjectId('66426944816fe65e46dbdee0') }
p12>
```

- Inserta 10 libros con `insertMany`, con precios entre 5 y 30 euros. Y verifica el resultado.

```
> db.books.insertMany([
  { nombre: "Reconstrucción casas", autor: "Manuel Garcia", editorial: "Benuventure", precio : 45 , cantidad : 230 },
  { nombre: "Reconstrucción casas 2", autor: "Manuel Garcia 2", editorial: "Benuventure 2", precio : 23 , cantidad : 100 },
  { nombre: "Reconstrucción casas 3", autor: "Manuel Garcia 3", editorial: "Benuventure 3", precio : 65 , cantidad : 340 },
  { nombre: "Reconstrucción casas 4", autor: "Manuel Garcia 4", editorial: "Benuventure 4", precio : 74 , cantidad : 633 },
  { nombre: "Reconstrucción casas 5", autor: "Manuel Garcia 5", editorial: "Benuventure 5", precio : 41 , cantidad : 740 },
  { nombre: "Reconstrucción casas 6", autor: "Manuel Garcia 6", editorial: "Benuventure 6", precio : 17 , cantidad : 60 },
  { nombre: "Reconstrucción casas 7", autor: "Manuel Garcia 7", editorial: "Benuventure 7", precio : 25 , cantidad : 23},
  { nombre: "Reconstrucción casas 8", autor: "Manuel Garcia 8", editorial: "Benuventure 8", precio : 5 , cantidad : 20 },
  { nombre: "Reconstrucción casas 9", autor: "Manuel Garcia 9", editorial: "Benuventure 9", precio : 7 , cantidad : 241},
  { nombre: "Reconstrucción casas 10", autor: "Manuel Garcia 10", editorial: "Benuventure 10", precio : 54 , cantidad : 300 }
]);
```

```
< {  
  acknowledged: true,  
  insertedIds: {  
    '0': ObjectId('66427102816fe65e46dbdee1'),  
    '1': ObjectId('66427102816fe65e46dbdee2'),  
    '2': ObjectId('66427102816fe65e46dbdee3'),  
    '3': ObjectId('66427102816fe65e46dbdee4'),  
    '4': ObjectId('66427102816fe65e46dbdee5'),  
    '5': ObjectId('66427102816fe65e46dbdee6'),  
    '6': ObjectId('66427102816fe65e46dbdee7'),  
    '7': ObjectId('66427102816fe65e46dbdee8'),  
    '8': ObjectId('66427102816fe65e46dbdee9'),  
    '9': ObjectId('66427102816fe65e46dbdeea')  
  }  
}  
p12> |
```

VERIFICANDO RESULTADOS

```

> show collections
< books
> db.books.find()
< {
  _id: ObjectId('66426944816fe65e46dbdee0'),
  nombre: 'El viejo y el mar',
  autor: 'Rafael Pombo',
  editorial: 'Cuervana',
  precio: 12,
  cantidad: 850
}
{
  _id: ObjectId('66427102816fe65e46dbdee1'),
  nombre: 'Reconstrucción casas',
  autor: 'Manuel Garcia',
  editorial: 'Benuventure',
  precio: 45,

```

```

  editorial: 'Benuventure',
  precio: 45,
  cantidad: 230
}
{
  _id: ObjectId('66427102816fe65e46dbdee2'),
  nombre: 'Reconstrucción casas 2',
  autor: 'Manuel Garcia 2',
  editorial: 'Benuventure 2',
  precio: 23,
  cantidad: 100
}
{
  _id: ObjectId('66427102816fe65e46dbdee3'),
  nombre: 'Reconstrucción casas 3',
  autor: 'Manuel Garcia 3',
  editorial: 'Benuventure 3',
  precio: 65,
  cantidad: 340

```

```

    cantidad: 20
  }
  {
    _id: ObjectId('66427102816fe65e46dbdee9'),
    nombre: 'Reconstrucción casas 9',
    autor: 'Manuel Garcia 9',
    editorial: 'Benuventure 9',
    precio: 7,
    cantidad: 241
  }
  {
    _id: ObjectId('66427102816fe65e46dbdeea'),
    nombre: 'Reconstrucción casas 10',
    autor: 'Manuel Garcia 10',
    editorial: 'Benuventure 10',
    precio: 54,
    cantidad: 300
  }
}
p12> |

```

- Muestra solo el primer libro que has insertado (utiliza el campo `_id`).

```

> db.books.find({_id:ObjectId('66426944816fe65e46dbdee0')})
< {
  _id: ObjectId('66426944816fe65e46dbdee0'),
  nombre: 'El viejo y el mar',
  autor: 'Rafael Pombo',
  editorial: 'Cuervana',
  precio: 12,
  cantidad: 850
}
p12>

```

- Muestra los libros que cuestan 10 euros y tenemos 5 unidades.


```
> db.books.find({precio: 10, cantidad: 5})  
<  
p12>
```

- Obtiene los libros más caros de 10 euros.

```
<  
> db.books.find({precio: {$gte:10}})  
< {  
  _id: ObjectId('66426944816fe65e46dbdee0'),  
  nombre: 'El viejo y el mar',  
  autor: 'Rafael Pombo',  
  editorial: 'Cuervana',  
  precio: 12,  
  cantidad: 850  
}  
{  
  _id: ObjectId('66427102816fe65e46dbdee1'),  
  nombre: 'Reconstrucción casas',  
  autor: 'Manuel García',  
  editorial: 'Benuventure',  
  precio: 45,  
  cantidad: 230  
}  
{
```

```
{
  _id: ObjectId('66427102816fe65e46dbdee2'),
  nombre: 'Reconstrucción casas 2',
  autor: 'Manuel García 2',
  editorial: 'Benuventure 2',
  precio: 23,
  cantidad: 100
}
{
  _id: ObjectId('66427102816fe65e46dbdee3'),
  nombre: 'Reconstrucción casas 3',
  autor: 'Manuel García 3',
  editorial: 'Benuventure 3',
  precio: 65,
  cantidad: 340
}
{
```

```
  _id: ObjectId('66427102816fe65e46dbdee4'),
  nombre: 'Reconstrucción casas 4',
  autor: 'Manuel García 4',
  editorial: 'Benuventure 4',
  precio: 74,
  cantidad: 633
}
{
  _id: ObjectId('66427102816fe65e46dbdee5'),
  nombre: 'Reconstrucción casas 5',
  autor: 'Manuel García 5',
  editorial: 'Benuventure 5',
  precio: 41,
  cantidad: 740
}
{
```

```

    _id: ObjectId('66427102816fe65e46dbdee6'),
    nombre: 'Reconstrucción casas 6',
    autor: 'Manuel García 6',
    editorial: 'Benuventure 6',
    precio: 17,
    cantidad: 60
  }
  {
    _id: ObjectId('66427102816fe65e46dbdee7'),
    nombre: 'Reconstrucción casas 7',
    autor: 'Manuel García 7',
    editorial: 'Benuventure 7',
    precio: 25,
    cantidad: 23
  }
  {

```

```

    _id: ObjectId('66427102816fe65e46dbdeea'),
    nombre: 'Reconstrucción casas 10',
    autor: 'Manuel García 10',
    editorial: 'Benuventure 10',
    precio: 54,
    cantidad: 300
  }
p12>

```

- Obtiene los libros que tengamos menos de 3 unidades.

```

> db.books.find({cantidad: {$lt:3}})
<
p12>

```

- Obtiene los libros que cuesten entre 15 y 25 euros.

```
> db.books.find({precio: {$gte:15,$lte:25}})
< {
  _id: ObjectId('66427102816fe65e46dbdee2'),
  nombre: 'Reconstrucción casas 2',
  autor: 'Manuel García 2',
  editorial: 'Benuventure 2',
  precio: 23,
  cantidad: 100
}
{
  _id: ObjectId('66427102816fe65e46dbdee6'),
  nombre: 'Reconstrucción casas 6',
  autor: 'Manuel García 6',
  editorial: 'Benuventure 6',
  precio: 17,
  cantidad: 60
}
{
```

```
  _id: ObjectId('66427102816fe65e46dbdee7'),
  nombre: 'Reconstrucción casas 7',
  autor: 'Manuel García 7',
  editorial: 'Benuventure 7',
  precio: 25,
  cantidad: 23
}
p12>
```

- Obtiene los libros que no pertenezcan a alguna de las editoriales que habéis añadido.

PRIMERO CREO UNA VARIABLE PARA ASIGNARLE LAS DISTINTAS EDITORIALES QUE TIENE CADA FILA (DOCUMENTO)

```

> var addeditorial = db.books.distinct('editorial');
> print(addeditorial);
< [
  'Benuventure',
  'Benuventure 10',
  'Benuventure 2',
  'Benuventure 3',
  'Benuventure 4',
  'Benuventure 5',
  'Benuventure 6',
  'Benuventure 7',
  'Benuventure 8',
  'Benuventure 9',
  'Cuervana'
]
p12>

```

LUEGO DEBO BUSCAR LOS LIBROS QUE NO PERTENEZCAN A NINGUNA DE LAS ANTERIORES EDITORIALES GUARDADAS EN EL ARRAY – VARIABLE LLAMADA “addeditorial”

```

> db.books.find({editorial: {$nin:addeditorial}})
<
p12>

```

2.4. Actualización y eliminación de datos

- Elimina solo el primer libro que has insertado.

```

> db.books.deleteOne({})
< {
  acknowledged: true,
  deletedCount: 1
}
p12>

```

```

> db.books.find()
< {
  _id: ObjectId('66427102816fe65e46dbdee1'),
  nombre: 'Reconstrucción casas',
  autor: 'Manuel García',
  editorial: 'Benuventure',
  precio: 45,
  cantidad: 230
}
{
  _id: ObjectId('66427102816fe65e46dbdee2'),
  nombre: 'Reconstrucción casas 2',
  autor: 'Manuel García 2',
  editorial: 'Benuventure 2',
  precio: 23,
  cantidad: 100
}

```

DE ESTA FORMA SE COMPRUEBA QUE YA NO EXISTE EL PRIMER LIBRO INSERTADO LLAMADO “El viejo y el mar” de “Rafael pombo”

- Elimina todos los libros que tengan menos de 5 unidades.

```

> db.books.deleteMany({cantidad: {$lt:5}})
< {
  acknowledged: true,
  deletedCount: 0
}
p12>

```

- Añade un campo de descripción a uno de los libros.

```

> db.books.updateOne(
  { _id: ObjectId('66427102816fe65e46dbdee6') },
  { $set: { descripcion: "Aprendiendo a añadir un registro del documento" } }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
p12>

```

VERIFICO RESULTADOS

```

> db.books.find({_id:ObjectId('66427102816fe65e46dbdee6')})
< {
  _id: ObjectId('66427102816fe65e46dbdee6'),
  nombre: 'Reconstrucción casas 6',
  autor: 'Manuel García 6',
  editorial: 'Benuventure 6',
  precio: 17,
  cantidad: 60,
  descripcion: 'Aprendiendo a añadir un registro del documento'
}
p12>

```

- Elimina este campo creado de este documento. (Pista: \$unset)

```

> db.books.updateOne({_id:ObjectId('66427102816fe65e46dbdee6')},{ $unset:{descripcion:''}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
p12>

```

- Actualiza el campo cantidad a 10 para todos los libros que cuesten más de 15 euros.

```
> db.books.updateMany(  
  { precio: { $gt: 15 } },  
  { $set: { cantidad: 10 } }  
)  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 8,  
  modifiedCount: 8,  
  upsertedCount: 0  
}  
p12>
```

- Añade un campo llamado "order" y ponlo a true solo a los libros que tengan menos de 3 unidades.

```
> db.books.updateMany(  
  { unidades: { $lt: 3 } },  
  { $set: { order: true } }  
)  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 0,  
  modifiedCount: 0,  
  upsertedCount: 0  
}  
p12>
```

- Muestra los libros que tengan un precio mayor a 15 o que tengamos que hacer una orden.

>_MONGOSH

```
> db.books.find({$or: [{ precio: { $gt: 15 } }],{ order: true }})
```

```
< {
  _id: ObjectId('66427102816fe65e46dbdee1'),
  nombre: 'Reconstrucción casas',
  autor: 'Manuel García',
  editorial: 'Benuventure',
  precio: 45,
  cantidad: 10
}
{
  _id: ObjectId('66427102816fe65e46dbdee2'),
  nombre: 'Reconstrucción casas 2',
  autor: 'Manuel García 2',
  editorial: 'Benuventure 2',
  precio: 23,
  cantidad: 10
}
{
```

```
  _id: ObjectId('66427102816fe65e46dbdee3'),
  nombre: 'Reconstrucción casas 3',
  autor: 'Manuel García 3',
  editorial: 'Benuventure 3',
  precio: 65,
  cantidad: 10
}
{
  _id: ObjectId('66427102816fe65e46dbdee4'),
  nombre: 'Reconstrucción casas 4',
  autor: 'Manuel García 4',
  editorial: 'Benuventure 4',
  precio: 74,
  cantidad: 10
}
{
```

```

    _id: ObjectId('66427102816fe65e46dbdee5'),
    nombre: 'Reconstrucción casas 5',
    autor: 'Manuel Garcia 5',
    editorial: 'Benuventure 5',
    precio: 41,
    cantidad: 10
  }
  {
    _id: ObjectId('66427102816fe65e46dbdee6'),
    nombre: 'Reconstrucción casas 6',
    autor: 'Manuel Garcia 6',
    editorial: 'Benuventure 6',
    precio: 17,
    cantidad: 10
  }
  {

```

```

    _id: ObjectId('66427102816fe65e46dbdee7'),
    nombre: 'Reconstrucción casas 7',
    autor: 'Manuel Garcia 7',
    editorial: 'Benuventure 7',
    precio: 25,
    cantidad: 10
  }
  {
    _id: ObjectId('66427102816fe65e46dbdeea'),
    nombre: 'Reconstrucción casas 10',
    autor: 'Manuel Garcia 10',
    editorial: 'Benuventure 10',
    precio: 54,
    cantidad: 10
  }
p12>

```

- Haz otra query usando \$not y otra con \$and.

```
> db.books.find({$nor:[{cantidad:{$lte:20}}]})
< {
  _id: ObjectId('66427102816fe65e46dbdee9'),
  nombre: 'Reconstrucción casas 9',
  autor: 'Manuel García 9',
  editorial: 'Benuventure 9',
  precio: 7,
  cantidad: 241
}
p12>
```

```
> db.books.find({$and:[{ cantidad:{$lte: 20}},{editorial:'Cuervana'}]})
<
p12>
```

- Después de mostrarlas, elimina todas las entradas de la consulta anterior con un solo comando.

```
> db.books.deleteMany({$or:[{cantidad:{$lt:20}},{editorial:'Cuervana'}]})
< {
  acknowledged: true,
  deletedCount: 8
}
p12>
```

- Vuelve a insertar todos los libros y muéstralos ordenados por título de forma descendente.

PARA EVITAR ERRORES Y COMO QUEDARON POCAS DOCUMENTOS, SE PROCEDE A ELIMINAR TODO Y VOLVER A INSERTAR

```
> db.books.deleteMany({})
< {
  acknowledged: true,
  deletedCount: 2
}
p12>
```

INSERTAR DE NUEVO LOS DATOS

```
> db.books.insertMany([
  { nombre: "Reconstrucción casas", autor: "Manuel Garcia", editorial: "Benuventure", precio : 45 , cantidad : 230 },
  { nombre: "Reconstrucción casas 2", autor: "Manuel Garcia 2", editorial: "Benuventure 2", precio : 23 , cantidad : 100 },
  { nombre: "Reconstrucción casas 3", autor: "Manuel Garcia 3", editorial: "Benuventure 3", precio : 65 , cantidad : 340 },
  { nombre: "Reconstrucción casas 4", autor: "Manuel Garcia 4", editorial: "Benuventure 4", precio : 74 , cantidad : 633 },
  { nombre: "Reconstrucción casas 5", autor: "Manuel Garcia 5", editorial: "Benuventure 5", precio : 41 , cantidad : 740 },
  { nombre: "Reconstrucción casas 6", autor: "Manuel Garcia 6", editorial: "Benuventure 6", precio : 17 , cantidad : 60 },
  { nombre: "Reconstrucción casas 7", autor: "Manuel Garcia 7", editorial: "Benuventure 7", precio : 25 , cantidad : 23},
  { nombre: "Reconstrucción casas 8", autor: "Manuel Garcia 8", editorial: "Benuventure 8", precio : 5 , cantidad : 20 },
  { nombre: "Reconstrucción casas 9", autor: "Manuel Garcia 9", editorial: "Benuventure 9", precio : 7 , cantidad : 241},
  { nombre: "Reconstrucción casas 10", autor: "Manuel Garcia 10", editorial: "Benuventure 10", precio : 54 , cantidad : 300 }
] );
```

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6643a7cd816fe65e46dbdeeb'),
    '1': ObjectId('6643a7cd816fe65e46dbdeec'),
    '2': ObjectId('6643a7cd816fe65e46dbdeed'),
    '3': ObjectId('6643a7cd816fe65e46dbdeee'),
    '4': ObjectId('6643a7cd816fe65e46dbdeef'),
    '5': ObjectId('6643a7cd816fe65e46dbdef0'),
    '6': ObjectId('6643a7cd816fe65e46dbdef1'),
    '7': ObjectId('6643a7cd816fe65e46dbdef2'),
    '8': ObjectId('6643a7cd816fe65e46dbdef3'),
    '9': ObjectId('6643a7cd816fe65e46dbdef4')
  }
}
p12>
```

- Muestra solo los 5 libros más baratos.

```
> db.books.find().sort({precio: 1}).limit(5)
< {
  _id: ObjectId('6643a7cd816fe65e46dbdef2'),
  nombre: 'Reconstrucción casas 8',
  autor: 'Manuel García 8',
  editorial: 'Benuventure 8',
  precio: 5,
  cantidad: 20
}
{
  _id: ObjectId('6643a7cd816fe65e46dbdef3'),
  nombre: 'Reconstrucción casas 9',
  autor: 'Manuel García 9',
  editorial: 'Benuventure 9',
  precio: 7,
  cantidad: 241
}
{
```

```
  _id: ObjectId('6643a7cd816fe65e46dbdef0'),
  nombre: 'Reconstrucción casas 6',
  autor: 'Manuel García 6',
  editorial: 'Benuventure 6',
  precio: 17,
  cantidad: 60
}
{
  _id: ObjectId('6643a7cd816fe65e46dbdeec'),
  nombre: 'Reconstrucción casas 2',
  autor: 'Manuel García 2',
  editorial: 'Benuventure 2',
  precio: 23,
  cantidad: 100
}
{
```

```

    _id: ObjectId('6643a7cd816fe65e46dbdef1'),
    nombre: 'Reconstrucción casas 7',
    autor: 'Manuel Garcia 7',
    editorial: 'Benaventure 7',
    precio: 25,
    cantidad: 23
  }
p12>

```

- Muestra los libros que cuestan más de 15 euros, pero solo queremos tener el título. (Importante: no queremos el id).

```

> db.books.find({precio: {$gt: 15}}, {nombre: 1, _id: 0})
< {
  nombre: 'Reconstrucción casas'
}
{
  nombre: 'Reconstrucción casas 2'
}
{
  nombre: 'Reconstrucción casas 3'
}
{
  nombre: 'Reconstrucción casas 4'
}
{
  nombre: 'Reconstrucción casas 5'
}
{

```

```

    nombre: 'Reconstrucción casas 6'
  }
  {
    nombre: 'Reconstrucción casas 7'
  }
  {
    nombre: 'Reconstrucción casas 10'
  }
p12>

```

2.5. Manejo colecciones bd

- Inserta 5 documentos en una colección llamada "películas". Los documentos deben tener campos similares a los de los libros (nombre, director, editoriales, precio, cantidad) y añade un campo adicional con la fecha de estreno de la película usando la clase Date.

```

> db.createCollection("películas")
< { ok: 1 }
p12>

```

```

> db.películas.insertOne({
  nombre: "Picapiedras 1",
  director: "Carlos Martinez 1",
  editoriales: "films europa",
  precio: 40,
  cantidad: 10,
  f_estreno: new Date("2023-01-24")
})
< {
  acknowledged: true,
  insertedId: ObjectId('6643b131816fe65e46dbdef5')
}

```

```

> db.peliculas.insertMany([
  {nombre: "Picapiedras 2",director: "Carlos Martinez 2",editoriales: "films Europa 2",precio: 86,cantidad: 7,f_estreno: new D
  {nombre: "Picapiedras 3",director: "Carlos Martinez 3",editoriales: "films Europa 3",precio: 13,cantidad: 50,f_estreno: new
  {nombre: "Picapiedras 4",director: "Carlos Martinez 4",editoriales: "films Europa 4",precio: 8,cantidad: 6,f_estreno: new Da
  {nombre: "Picapiedras 5",director: "Carlos Martinez 5",editoriales: "films Europa 5",precio: 40,cantidad: 10,f_estreno: new

  });

< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6643b290816fe65e46dbdef6'),
    '1': ObjectId('6643b290816fe65e46dbdef7'),
    '2': ObjectId('6643b290816fe65e46dbdef8'),
    '3': ObjectId('6643b290816fe65e46dbdef9')
  }
}

```

```

> db.peliculas.find()
< {
  _id: ObjectId('6643b131816fe65e46dbdef5'),
  nombre: 'Picapiedras 1',
  director: 'Carlos Martinez 1',
  editoriales: 'films europa',
  precio: 40,
  cantidad: 10,
  f_estreno: 2023-01-24T00:00:00.000Z
}
{
  _id: ObjectId('6643b290816fe65e46dbdef6'),
  nombre: 'Picapiedras 2',
  director: 'Carlos Martinez 2',
  editoriales: 'films Europa 2',
  precio: 86,
  cantidad: 7,
  f_estreno: 2020-12-01T00:00:00.000Z
}

```



```

    _id: ObjectId('6643b290816fe65e46dbdef7'),
    nombre: 'Picapiedras 3',
    director: 'Carlos Martínez 3',
    editoriales: 'films Europa 3',
    precio: 13,
    cantidad: 50,
    f_estreno: 1997-05-20T00:00:00.000Z
  }
  {
    _id: ObjectId('6643b290816fe65e46dbdef8'),
    nombre: 'Picapiedras 4',
    director: 'Carlos Martínez 4',
    editoriales: 'films Europa 4',
    precio: 8,
    cantidad: 6,
    f_estreno: 2005-01-05T00:00:00.000Z
  }
  {

```

```

    _id: ObjectId('6643b290816fe65e46dbdef9'),
    nombre: 'Picapiedras 5',
    director: 'Carlos Martínez 5',
    editoriales: 'films Europa 5',
    precio: 40,
    cantidad: 10,
    f_estreno: 1996-03-14T00:00:00.000Z
  }
p12>

```

- Crea un índice único en la colección "películas" usando el campo "nombre" en orden descendente.

```

> db.peliculas.createIndex({ nombre: -1 }, { unique: true })
< nombre_-1
p12>

```

- ¿En qué ayuda haber creado un índice en la colección? Explica brevemente.

El índice único en esta colección películas, garantiza que no existan documentos (filas) con el mismo valor en el campo “nombre”, es decir prevenimos 100% los duplicados.

- Muestra los índices que tienes en la colección "películas" y explica lo que ves.

```
> db.películas.getIndexes()
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { nombre: -1 }, name: 'nombre_-1', unique: true }
]
p12>
```

La consulta indica que se tienen dos índices creados; uno que viene forma automática con el _id y el segundo índice que fue creado por mi al cual llamo automáticamente “nombre_-1”. Este ultimo lo que indica es que se creó un índice por el usuario a partir del nombre de la película y en orden descendente porque lleva un -1 y que este nombre debe ser único.

- Elimina el índice creado en la colección "películas".

```
> db.películas.dropIndex("nombre_-1")
< { nIndexesWas: 2, ok: 1 }
p12>
```

- Muestra todas colecciones de la base de datos.

```
> show collections
< books
   películas
p12>
```

- Elimina todos los documentos de la colección "películas".

```
> db.peliculas.deleteMany({})
< {
  acknowledged: true,
  deletedCount: 5
}
p12>
```

- Elimina la colección "peliculas".

```
> db.peliculas.drop()
< true
p12>
```

- Elimina la base de datos.

```
> db.dropDatabase()
< { ok: 1, dropped: 'p12' }
p12>
```

- Si no ingresamos el ID en MongoDB, ¿cómo garantiza que sea único? Explica brevemente. . (Pista: bytes)

MongoDB otorga automáticamente un identificador único para cada documento con el ObjectId.

"El ObjectId es un valor de 12 bytes que se compone de diferentes partes:

- **Timestamp (4 bytes):** Representa el número de segundos desde el 1 de enero de 1970. Esto asegura que las ObjectId creadas en un mismo segundo tienen el mismo prefijo de 4 bytes.
- **Machine Identifier (3 bytes):** Un identificador único de la máquina que genera el ObjectId. Está basado en el hostname de la máquina y una función hash.
- **Process Identifier (2 bytes):** El identificador del proceso asegura que todos los procesos en la misma máquina no generen ObjectId conflictivos.
- **Counter (3 bytes):** Valor que empieza siendo aleatorio y se incrementa cada vez que se genera un nuevo ObjectId en la misma máquina y el mismo proceso dentro de un segundo.

Ejemplo de un ObjectId: `_id: ObjectId("70c71p2f5cf1e8a76d6c4f4b")`