

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3089 Base de Datos 2

Sección 20

Daniela Mesalles



## Laboratorio 2

Linda Jiménez, 21169

GUATEMALA, 7 de febrero de 2024

# Ejercicio 1

## 1.1. Script data random y carga de los datos a la base de datos.

```
for (let i = 1; i < 100001; i++) {  
  let usuarios = {  
    nombre: "Usuario " + i,  
    email: "usuario" + i + "@example.com",  
    fecha_registro: getRandomDate(new Date(2017, 0, 1), new Date()),  
    activo: i % 2 === 0,  
    puntos: getRandomInt(10, 10000),  
    historial_compras: [{ producto: "Producto " + getRandomInt(0, 20), fecha: getRandomDate(new Date(2017, 0, 1), new Date()) }],  
    dirección: {  
      calle: "Calle " + i,  
      ciudad: "Ciudad " + i,  
      codigo_postal: getRandomInt(1000, 100000)  
    },  
    tags: ["tag" + getRandomInt(6,10), "tag" + getRandomInt(1, 5), "tag" + getRandomInt(11,15), "tag" + getRandomInt(16,20)],  
    archivo: i % 2 === 0,  
    notas: "Notas del usuario " + i,  
    visitas: getRandomInt(0, 10000),  
    amigos: Array.from({ length: getRandomInt(0, 9000) }, (_, j) => j),  
    preferencias: {  
      color: getRandomElement(colores),  
      idioma: getRandomElement(idioma),  
      tema: getRandomElement(tema)  
    }  
  };  
  db.usuarios.insertOne(usuarios);  
}
```

```
Microsoft Windows [Version 10.0.22631.3085]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\lijv1>"C:\Program Files\MongoDB\Server\7.0\bin\mongosh.exe"  
"C:\Program Files\MongoDB\Server\7.0\bin\mongosh.exe" is not recognized as an internal or external command,  
operable program or batch file.  
  
C:\Users\lijv1>"C:\Users\lijv1\AppData\Local\Programs\mongosh\bin\mongosh.exe"  
Current Mongosh Log ID: 65c1e0a950f3bee4f8004161  
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2  
.1.3  
Using MongoDB:      7.0.5  
Using Mongosh:      2.1.3  
  
For mongosh info see: https://docs.mongodb.com/mongodb-shell/  
  
-----  
The server generated these startup warnings when booting  
2024-02-06T01:27:25.457-06:00: Access control is not enabled for the database. Read and write access to data and conf  
iguration is unrestricted  
-----  
  
test> use lab02_1  
switched to db lab02_1  
lab02_1> load("C:/Users/lijv1/OneDrive/Desktop/usuarios.js")  
|
```

```
lab02_1> load("C:/Users/lijv1/OneDrive/Desktop/usuarios.js")  
load("C:/Users/lijv1/OneDrive/Desktop/usuarios.js")  
true  
1 of 100000 documents loaded.
```

## 1.2.

a.

```
4
5 ✓ db.usuarios.find({activo: true, puntos: {$gt: 500}}).sort({puntos: 1})
6
```

	nombre	notas	preferencias	puntos	tags
1	Usuario 934	Notas del usuario 934	{ "color": "Violeta", "idioma": "Francés", "tema": "C	501	["tag10", "tag5", "tag13", "t
2	Usuario 12774	Notas del usuario 12774	{ "color": "Rojo", "idioma": "Alemán", "tema": "Claro	501	["tag9", "tag4", "tag15", "ta
3	Usuario 25646	Notas del usuario 25646	{ "color": "Amarillo", "idioma": "Español", "tema": "	501	["tag9", "tag4", "tag12", "ta
4	Usuario 67252	Notas del usuario 67252	{ "color": "Azul", "idioma": "Alemán", "tema": "Claro	501	["tag7", "tag1", "tag11", "ta
5	Usuario 80618	Notas del usuario 80618	{ "color": "Verde", "idioma": "Italiano", "tema": "Os	501	["tag8", "tag3", "tag13", "ta
6	Usuario 84750	Notas del usuario 84750	{ "color": "Naranja", "idioma": "Alemán", "tema": "Os	501	["tag7", "tag3", "tag14", "ta
7	Usuario 10516	Notas del usuario 10516	{ "color": "Verde", "idioma": "Alemán", "tema": "Oscu	502	["tag10", "tag3", "tag13", "ta
8	Usuario 24046	Notas del usuario 24046	{ "color": "Verde", "idioma": "Francés", "tema": "Osc	502	["tag8", "tag1", "tag15", "ta
9	Usuario 40788	Notas del usuario 40788	{ "color": "Amarillo", "idioma": "Francés", "tema": "	502	["tag10", "tag3", "tag12", "t
10	Usuario 57052	Notas del usuario 57052	{ "color": "Rojo", "idioma": "Español", "tema": "Oscu	502	["tag10", "tag4", "tag11", "t
11	Usuario 93970	Notas del usuario 93970	{ "color": "Violeta", "idioma": "Inglés", "tema": "Cl	502	["tag8", "tag2", "tag13", "ta
12	Usuario 33576	Notas del usuario 33576	{ "color": "Amarillo", "idioma": "Francés", "tema": "	503	["tag6", "tag1", "tag14", "ta
13	Usuario 61182	Notas del usuario 61182	{ "color": "Naranja", "idioma": "Alemán", "tema": "Os	503	["tag7", "tag4", "tag11", "ta
14	Usuario 61254	Notas del usuario 61254	{ "color": "Verde", "idioma": "Español", "tema": "Osc	503	["tag6", "tag2", "tag12", "ta
15	Usuario 72264	Notas del usuario 72264	{ "color": "Naranja", "idioma": "Español", "tema": "O	503	["tag6", "tag3", "tag13", "ta
16	Usuario 24778	Notas del usuario 24778	{ "color": "Amarillo", "idioma": "Español", "tema": "	504	["tag6", "tag4", "tag13", "ta
17	Usuario 31040	Notas del usuario 31040	{ "color": "Violeta", "idioma": "Italiano", "tema": "Os	504	["tag6", "tag6", "tag13", "ta

b.

```
7 const fechaActual = new Date();
8 const milisegundosEnUnaSemana = 7 * 24 * 60 * 60 * 1000; // 7 días en milisegundos
9 const fechaLimite = new Date(fechaActual.getTime() - milisegundosEnUnaSemana);
10
11 ✓ db.usuarios.find({
12   "historial_compras.producto": "Producto 1",
13   "historial_compras.fecha": {$gt: fechaLimite}
14 }).sort({"historial_compras.fecha": -1})
15
```

	historial_compras	nombre	notas	preferencias
1	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 99444	Notas del usuario 99444	{ "color": "Azul", "idioma": "Italian
2	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 84403	Notas del usuario 84403	{ "color": "Violeta", "idioma": "Ital
3	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 81836	Notas del usuario 81836	{ "color": "Azul", "idioma": "Italian
4	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 27221	Notas del usuario 27221	{ "color": "Naranja", "idioma": "Ingl
5	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 14985	Notas del usuario 14985	{ "color": "Violeta", "idioma": "Espa
6	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 38658	Notas del usuario 38658	{ "color": "Rojo", "idioma": "Español
7	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 52436	Notas del usuario 52436	{ "color": "Verde", "idioma": "Italia
8	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 99051	Notas del usuario 99051	{ "color": "Naranja", "idioma": "Espa
9	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 62915	Notas del usuario 62915	{ "color": "Naranja", "idioma": "Ingl
10	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 90475	Notas del usuario 90475	{ "color": "Naranja", "idioma": "Espa
11	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 6975	Notas del usuario 6975	{ "color": "Violeta", "idioma": "Fran
12	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 46647	Notas del usuario 46647	{ "color": "Verde", "idioma": "Francé
13	{ "producto": "Producto 1", "fecha": new ISODate("20	Usuario 57442	Notas del usuario 57442	{ "color": "Verde", "idioma": "Alemán

c.

```
15
16 ✓ db.usuarios.find({
17     tags: "tag2",
18     visitas: { $gt: 100 }
19 }).sort({visitas: 1})
20
21 //Usuarios con preferencias de color "azul" y que tienen entre 1000 y 2000 amigos.
```

Output lab02\_1.usuarios lab02\_1.usuarios 2 ×

	visitas	preferencias	puntos	tags	visitas
1	del usuario 34651	{"color": "Violeta", "idioma": "Italiano", "tema": "Claro"}	7746	["tag7", "tag2", "tag12", "tag18"]	101
2	del usuario 55384	{"color": "Azul", "idioma": "Francés", "tema": "Claro"}	5707	["tag7", "tag2", "tag12", "tag18"]	101
3	del usuario 75631	{"color": "Azul", "idioma": "Alemán", "tema": "Oscuro"}	3141	["tag6", "tag2", "tag14", "tag16"]	101
4	del usuario 76963	{"color": "Verde", "idioma": "Español", "tema": "Oscuro"}	206	["tag10", "tag2", "tag13", "tag20"]	102
5	del usuario 13187	{"color": "Azul", "idioma": "Inglés", "tema": "Oscuro"}	6998	["tag7", "tag2", "tag13", "tag20"]	103
6	del usuario 22925	{"color": "Naranja", "idioma": "Inglés", "tema": "Claro"}	9464	["tag6", "tag2", "tag14", "tag20"]	103
7	del usuario 5084	{"color": "Violeta", "idioma": "Francés", "tema": "Oscuro"}	9617	["tag8", "tag2", "tag11", "tag18"]	104
8	del usuario 64046	{"color": "Rojo", "idioma": "Italiano", "tema": "Oscuro"}	2647	["tag10", "tag2", "tag11", "tag16"]	104
9	del usuario 69862	{"color": "Azul", "idioma": "Francés", "tema": "Claro"}	4854	["tag8", "tag2", "tag13", "tag16"]	104
10	del usuario 91445	{"color": "Amarillo", "idioma": "Alemán", "tema": "Oscuro"}	3824	["tag6", "tag2", "tag12", "tag20"]	104
11	del usuario 39742	{"color": "Violeta", "idioma": "Inglés", "tema": "Oscuro"}	7984	["tag8", "tag2", "tag14", "tag19"]	105
12	del usuario 71164	{"color": "Azul", "idioma": "Inglés", "tema": "Oscuro"}	4557	["tag8", "tag2", "tag11", "tag19"]	105
13	del usuario 83642	{"color": "Verde", "idioma": "Español", "tema": "Oscuro"}	6703	["tag6", "tag2", "tag11", "tag19"]	105

d.

```
23 ✓ db.usuarios.find({
24     "preferencias.color": "Azul",
25     $expr: {
26         $and: [
27             { $gte: [{ $max: "$amigos" }, 1000] },
28             { $lte: [{ $max: "$amigos" }, 2000] }
29         ]
30     }
31 }).sort({amigos: -1})
```

Output lab02\_1.usuarios Usuarios con prefer...e 1000 y 2000 amigos. ×

	vo	amigos	archivo	preferencias
1		[96), new NumberInt("1997"), new NumberInt("1998"), new NumberInt("1999")]		{"color": "Azul", "idioma": "Italiano", "tema": "Claro"}
2	e	[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	false	{"color": "Azul", "idioma": "Alemán", "tema": "Claro"}
3	e	[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	false	{"color": "Azul", "idioma": "Inglés", "tema": "Oscuro"}
4	e	[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	false	{"color": "Azul", "idioma": "Italiano", "tema": "Claro"}
5		[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	true	{"color": "Azul", "idioma": "Francés", "tema": "Claro"}
6	e	[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	false	{"color": "Azul", "idioma": "Francés", "tema": "Oscuro"}
7	e	[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	false	{"color": "Azul", "idioma": "Italiano", "tema": "Claro"}
8		[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	true	{"color": "Azul", "idioma": "Español", "tema": "Claro"}
9		[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	true	{"color": "Azul", "idioma": "Español", "tema": "Oscuro"}
10		[new NumberInt("0"), new NumberInt("1"), new NumberInt("1")]	true	{"color": "Azul", "idioma": "Español", "tema": "Claro"}

### 1.3.

- a. Para la consulta a del inicio 1.1 podemos ver que realizó un escaneo de un total de 100,000 documentos y de estos con la consulta ya retorno 47,571 documentos. El tiempo de ejecución fue de 27,159 ms.

```
4
5 ✓ db.usuarios.find({activo: true, puntos: {$gt: 500}}).sort({puntos: 1}).explain("executionStats")

Output lab02_1.usuarios Result 90 x
1 row
16 "executionStats": {
17   "executionSuccess": true,
18   "nReturned": 47571,
19   "executionTimeMillis": 27159,
20   "totalKeysExamined": 0,
21   "totalDocsExamined": 100000,
22   "executionStages": {
23     "stage": "SORT",
24     "nReturned": 47571,
25     "executionTimeMillisEstimate": 26627,
26     "works": 147573,
27     "advanced": 47571,
28     "needTime": 100001,
29     "needYield": 0,
30     "saveState": 841,
31     "restoreState": 841,
32     "isEOF": 1,
33     "sortPattern": {
34       "puntos": 1
35     },
36     "memLimit": 104857600,
37     "type": "simple"
```

- b. Para la consulta b del inicio 1.1 podemos ver que realizó un escaneo de un total de 100,000 documentos y de estos con la consulta ya retorno 15 documentos. El tiempo de ejecución fue de 321 ms.

```
11 ✓ db.usuarios.find({
12   "historial_compras.producto": "Producto 1",
13   "historial_compras.fecha": {$gt: fechaLimite}
14 }) sort({"historial_compras.fecha": -1}).explain("executionStats")

Output lab02_1.usuarios Result 92 x
1 row
16 "executionStats": {
17   "executionSuccess": true,
18   "nReturned": 15,
19   "executionTimeMillis": 321,
20   "totalKeysExamined": 0,
21   "totalDocsExamined": 100000,
22   "executionStages": {
23     "stage": "SORT",
24     "nReturned": 15,
25     "executionTimeMillisEstimate": 67,
26     "works": 100017,
27     "advanced": 15,
28     "needTime": 100001,
29     "needYield": 0,
30     "saveState": 100,
31     "restoreState": 100,
32     "isEOF": 1,
33     "sortPattern": {
34       "historial_compras.fecha": -1
35     },
36     "memLimit": 104857600,
37     "type": "simple"
```

- c. Para la consulta c del inciso 1.1 podemos ver que realizó un escaneo de un total de 100,000 documentos y de estos con la consulta ya retorno 20,052 documentos. El tiempo de ejecución fue de 11,367 ms.

```
16 ✓ db.usuarios.find({
17   tags: "tag2",
18   visitas: { $gt: 100 }
19 }).sort({visitas: 1}).explain("executionStats")
```

Output lab02\_1.usuarios Result 93 ×

```
16   "executionStats": {
17     "executionSuccess": true,
18     "nReturned": 20052,
19     "executionTimeMillis": 11367,
20     "totalKeysExamined": 0,
21     "totalDocsExamined": 100000,
22     "executionStages": {
23       "stage": "SORT",
24       "nReturned": 20052,
25       "executionTimeMillisEstimate": 11090,
26       "works": 120054,
27       "advanced": 20052,
28       "needTime": 100001,
29       "needYield": 0,
30       "saveState": 414,
31       "restoreState": 414,
32       "isEOF": 1,
33       "sortPattern": {
34         "visitas": 1
35       },
36     },
37   },
38 }
```

- d. Para la consulta d del inciso 1.1 podemos ver que realizó un escaneo de un total de 100,000 documentos y de estos con la consulta ya retorno 1,979 documentos. El tiempo de ejecución fue de 10,705 ms.

```
23 ✓ db.usuarios.find({
24   "preferencias.color": "Azul",
25   $expr: {
26     $and: [
27       { $gte: [{ $max: "$amigos" }, 1000] },
28       { $lte: [{ $max: "$amigos" }, 2000] }
29     ]
30   }
31 }).sort({amigos: -1}).explain("executionStats")
```

Output lab02\_1.usuarios Usuarios con prefere...e 1000 y 2

```
29   "executionStats": {
30     "executionSuccess": true,
31     "nReturned": 1979,
32     "executionTimeMillis": 10705,
33     "totalKeysExamined": 0,
34     "totalDocsExamined": 100000,
35     "executionStages": {
36       "stage": "SORT",
37       "nReturned": 1979,
38       "executionTimeMillisEstimate": 10411,
39       "works": 101981,
40       "advanced": 1979,
41       "needTime": 100001,
42       "needYield": 0,
43       "saveState": 414,
44       "restoreState": 414,
45       "isEOF": 1,
46       "sortPattern": {
47         "amigos": -1
48       },
49     },
50   },
51 }
```

1.4.

a.

```
32
33 ✓ db.usuarios.createIndex({activo: 1, puntos: 1})
34
```

Output lab02\_1.usuarios Result 95 ×

1 row 1 row > > | ↺ ⌚ ■ 📌

{ result

1	activo_1_puntos_1
---	-------------------

b.

```
34
35 ✓ db.usuarios.createIndex({"historial_compras.producto": 1, "historial_compras.fecha": 1})
36
37
```

Output lab02\_1.usuarios Result 96 ×

1 row 1 row > > | ↺ ⌚ ■ 📌

{ result

1	historial_compras.producto_1_historial_compras.fecha_1
---	--

c.

```
36
37 ✓ db.usuarios.createIndex({tags: 1, visitas: 1})
```

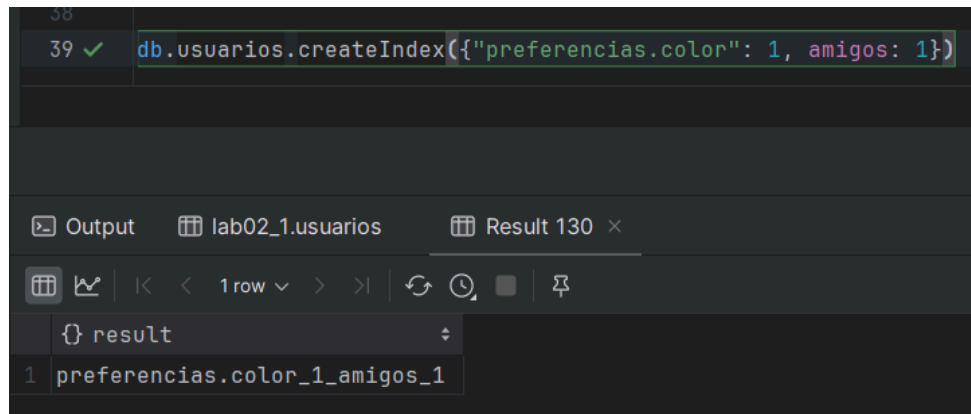
Output lab02\_1.usuarios Result 97 ×

1 row 1 row > > | ↺ ⌚ ■ 📌

{ result

1	tags_1_visitas_1
---	------------------

d.



```
38 db.usuarios.createIndex({"preferencias.color": 1, amigos: 1})
```

Output lab02\_1.usuarios Result 130 x

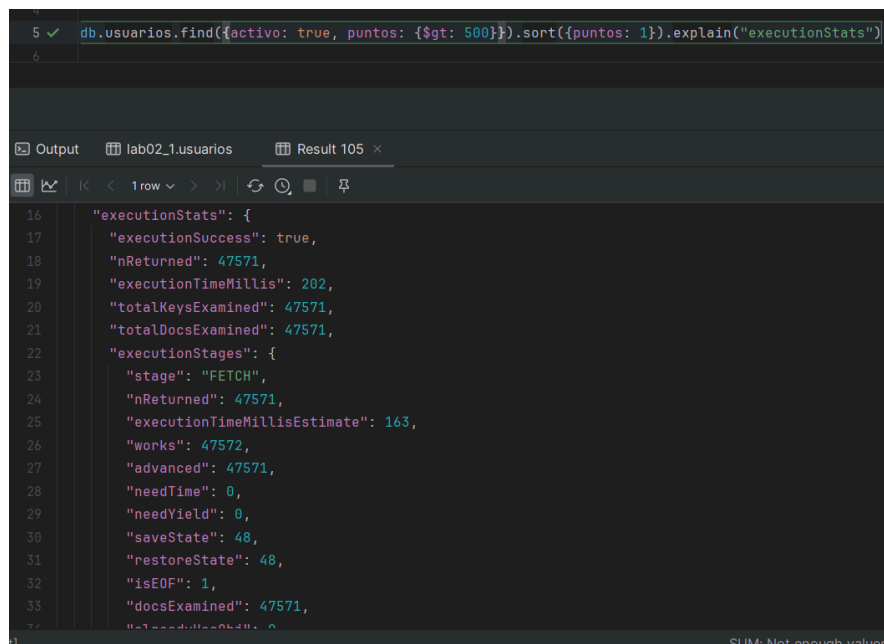
1 row

result

1 preferencias.color\_1\_amigos\_1

1.5.

- a. Para la consulta a del inciso 1.1 podemos ver que realizó un escaneo de un total de 47,571 documentos y de estos con la consulta retorno igual 47,571 documentos, vemos que ya no examino los 100,000 documentos, con la diferencia que ahora el tiempo de ejecución fue de 202 ms, una gran diferencia a comparación del tiempo anterior de 27,159 ms.



```
5 db.usuarios.find({activo: true, puntos: {$gt: 500}}).sort({puntos: 1}).explain("executionStats")
```

Output lab02\_1.usuarios Result 105 x

1 row

```
16 "executionStats": {
17   "executionSuccess": true,
18   "nReturned": 47571,
19   "executionTimeMillis": 202,
20   "totalKeysExamined": 47571,
21   "totalDocsExamined": 47571,
22   "executionStages": {
23     "stage": "FETCH",
24     "nReturned": 47571,
25     "executionTimeMillisEstimate": 163,
26     "works": 47572,
27     "advanced": 47571,
28     "needTime": 0,
29     "needYield": 0,
30     "saveState": 48,
31     "restoreState": 48,
32     "isEOF": 1,
33     "docsExamined": 47571,
34     "transform": 0
35   }
36 }
```

SUM: Not enough values



- b. Para la consulta b del inciso 1.1 podemos ver que realizó un escaneo de un total de 4,681 documentos y de estos con la consulta retornó 15 documentos, vemos que ya no examinó los 100,000 documentos, con la diferencia que ahora el tiempo de ejecución fue de 35 ms, una buena diferencia a comparación del tiempo anterior de 321 ms.

```
10
11 ✓ db.usuarios.find({
12     "historial_compras.producto": "Producto 1",
13     "historial_compras.fecha": {$gt: fechaLimite}
14 }).sort({"historial_compras.fecha": -1}).explain("executionStats")
15
```

Output lab02\_1.usuarios Result 110 x

```
16 "executionStats": {
17     "executionSuccess": true,
18     "nReturned": 15,
19     "executionTimeMillis": 35,
20     "totalKeysExamined": 4681,
21     "totalDocsExamined": 4681,
22     "executionStages": {
23         "stage": "SORT",
24         "nReturned": 15,
25         "executionTimeMillisEstimate": 10,
26         "works": 4698,
27         "advanced": 15,
28         "needTime": 4682,
29         "needYield": 0,
30         "saveState": 4,
31         "restoreState": 4,
32         "isEOF": 1,
33         "sortPattern": {
34             "historial_compras.fecha": -1
35         }
36     }
37 }
```

- c. Para la consulta c del inciso 1.1 podemos ver que realizó un escaneo de un total de 20,052 documentos y de estos con la consulta retornó igual 20,052 documentos, vemos que ya no examinó los 100,000 documentos, con la diferencia que ahora el tiempo de ejecución fue de 79 ms, una gran diferencia a comparación del tiempo anterior de 11,367 ms.

```
15
16 ✓ db.usuarios.find({
17     tags: "tag2",
18     visitas: { $gt: 100 }
19 }).sort({visitas: 1}).explain("executionStats")
20
```

Output lab02\_1.usuarios Result 111 x

```
16 "executionStats": {
17     "executionSuccess": true,
18     "nReturned": 20052,
19     "executionTimeMillis": 79,
20     "totalKeysExamined": 20052,
21     "totalDocsExamined": 20052,
22     "executionStages": {
23         "stage": "FETCH",
24         "nReturned": 20052,
25         "executionTimeMillisEstimate": 28,
26         "works": 20053,
27         "advanced": 20052,
28         "needTime": 0,
29         "needYield": 0,
30         "saveState": 20,
31         "restoreState": 20,
32         "isEOF": 1,
33         "docsExamined": 20052,
34         "alreadyHasObj": 0,
35     }
36 }
```

- d. Para la consulta d del inciso 1.1 podemos ver que realizó un escaneo de un total de 16,633 documentos y de estos con la consulta retorno los mismos 1,979 documentos, vemos que ya no examinó los 100,000 documentos, pero ahora el tiempo de ejecución fue de 60,940 ms, una gran diferencia a comparación del tiempo anterior de 10,705 ms el cual ahora es mucho más lento. En este caso el índice no favoreció al rendimiento de la consulta.

```
23 ✓ db.usuarios.find({
24   "preferencias.color": "Azul",
  }

  Output lab02_1.usuarios Usuarios con prefere...e 10
  1 row
  29 "executionStats": {
  30   "executionSuccess": true,
  31   "nReturned": 1979,
  32   "executionTimeMillis": 60940,
  33   "totalKeysExamined": 74266857,
  34   "totalDocsExamined": 16633,
  35   "executionStages": {
  36     "stage": "FETCH",
  37     "filter": {
  38       "$expr": {
  39         "$and": [
  40           {
  41             "$gte": [
  42               {
  43                 "$max": ["$amigos"]
  44               },
  45               {
  46                 "$const": 1000
  47               }
  48             ]
  49           },
  50         ]
  51       }
  52     }
  53   }
  54 }
```

## 1.6. Script con 50,000 datos random más

```
for (let i = 100001; i < 150001; i++) {  
  let usuarios = {  
    nombre: "Usuario " + i,  
    email: "usuario" + i + "@example.com",  
    fecha_registro: getRandomDate(new Date(2017, 0, 1), new Date()),  
    activo: i % 2 === 0,  
    puntos: getRandomInt(10, 10000),  
    historial_compras: [{ producto: "Producto " + getRandomInt(0, 20), fecha: getRandomDate(new Date(2017, 0, 1), new Date()) }],  
    dirección: {  
      calle: "Calle " + i,  
      ciudad: "Ciudad " + i,  
      codigo_postal: getRandomInt(1000, 100000)  
    },  
    tags: ["tag" + getRandomInt(6,10), "tag" + getRandomInt(1, 5), "tag" + getRandomInt(11,15), "tag" + getRandomInt(16,20)],  
    archivo: i % 2 === 0,  
    notas: "Notas del usuario " + i,  
    visitas: getRandomInt(0, 10000),  
    amigos: Array.from({ length: getRandomInt(0, 9000) }, (_, j) => j),  
    preferencias: {  
      color: getRandomElement(colores),  
      idioma: getRandomElement(idioma),  
      tema: getRandomElement(tema)  
    }  
  };  
  db.usuarios.insertOne(usuarios);  
}
```

- a. Para la consulta a del inciso 1.1 podemos ver que realizó un escaneo de un total de 47,571 documentos y de estos con la consulta retorno igual 47,571 documentos igual que antes,, con la diferencia que ahora el tiempo de ejecución fue de 163 ms, un poco menos a comparación del tiempo anterior de 202 ms.

```
5 ✓ db.usuarios.find({activo: true, puntos: {$gt: 500}}).sort({puntos: 1}).explain("executionStats")  
6  
  
Output lab02_1.usuarios Result 105 x  
1 row  
16 "executionStats": {  
17   "executionSuccess": true,  
18   "nReturned": 47571,  
19   "executionTimeMillis": 202,  
20   "totalKeysExamined": 47571,  
21   "totalDocsExamined": 47571,  
22   "executionStages": {  
23     "stage": "FETCH",  
24     "nReturned": 47571,  
25     "executionTimeMillisEstimate": 163,  
26     "works": 47572,  
27     "advanced": 47571,  
28     "needTime": 0,  
29     "needYield": 0,  
30     "saveState": 48,  
31     "restoreState": 48,  
32     "isEOF": 1,  
33     "docsExamined": 47571,  
34     "totalKeysExamined": 0  
35   }  
36 }
```

- b. Para la consulta b del inciso 1.1 podemos ver que realizó un escaneo de un total de 7,014 documentos y de estos con la consulta retornó 25 documentos, unos cuantos documentos más que la vez anterior. Ahora el tiempo de ejecución fue de 36 ms, casi ninguna diferencia a comparación del tiempo anterior de 35 ms.

```
10
11 ✓ db.usuarios.find({
12     "historial_compras.producto": "Producto 1",
13     "historial_compras.fecha": {$gt: fechaLimite}
14 }).sort({"historial_compras.fecha": -1}).explain("executionStats")

Output Result 12 ×
1 row
16 "executionStats": {
17     "executionSuccess": true,
18     "nReturned": 25,
19     "executionTimeMillis": 36,
20     "totalKeysExamined": 7014,
21     "totalDocsExamined": 7014,
22     "executionStages": {
23         "stage": "SORT",
24         "nReturned": 25,
25         "executionTimeMillisEstimate": 12,
26         "works": 7041,
27         "advanced": 25,
28         "needTime": 7015,
29         "needYield": 0,
30         "saveState": 7,
31         "restoreState": 7,
32         "isEOF": 1,
33         "sortPattern": {
34             "historial_compras.fecha": -1
```

- c. Para la consulta c del inciso 1.1 podemos ver que realizó un escaneo de un total de 29,863 documentos y de estos con la consulta retornó igual 29,863 documentos, unos cuantos documentos más que antes, con la diferencia que ahora el tiempo de ejecución fue de 119 ms, un poco más de tiempo con el anterior de 79 ms.

```
16 ✓ db.usuarios.find({
17     tags: "tag2",
18     visitas: { $gt: 100 }
19 }).sort({visitas: 1}).explain("executionStats")
20
```

Output Result 13 ×

1 row

```
16 "executionStats": {
17     "executionSuccess": true,
18     "nReturned": 29863,
19     "executionTimeMillis": 119,
20     "totalKeysExamined": 29863,
21     "totalDocsExamined": 29863,
22     "executionStages": {
23         "stage": "FETCH",
24         "nReturned": 29863,
25         "executionTimeMillisEstimate": 19,
26         "works": 29864,
27         "advanced": 29863,
28         "needTime": 0,
29         "needYield": 0,
30         "saveState": 29,
31         "restoreState": 29,
32         "isEOF": 1,
33         "docsExamined": 29863,
34         "alreadyHasObj": 0,
```

- d. Para la consulta d del inciso 1.1 podemos ver que realizó un escaneo de un total de 25,034 documentos y de estos con la consulta retornó 2,885 documentos, unos cuantos más documentos que la vez anterior, pero ahora el tiempo de ejecución fue de 133,499 ms, siendo un poco grande la diferencia a comparación del tiempo anterior de 60,940 ms el cual ahora es mucho más lento nuevamente.

```
23 ✓ db.usuarios.find({
24   "preferencias.color": "Azul",
25   $expr: {
    ,
    "executionStats": {
      "executionSuccess": true,
      "nReturned": 2885,
      "executionTimeMillis": 133499,
      "totalKeysExamined": 111911539,
      "totalDocsExamined": 25034,
      "executionStages": {
        "stage": "FETCH",
        "filter": {
          "$expr": {
            "$and": [
              {
                "$gte": [
                  {
                    "$max": ["$amigos"]
                  },
                  {
                    "$const": 1000
                  }
                ]
              }
            ]
          }
        }
      }
    }
  }
}
```

## Ejercicio 2

2.

2.1.

a.

```
lab02_1> use laboratorio2
use laboratorio2
switched to db laboratorio2
laboratorio2> |
```

b.

```
switched to db laboratorio2
laboratorio2> db.createCollection("vehiculos")
{ ok: 1 }
laboratorio2> |
```

```
PS C:\Users\lijv1> mongoimport --db laboratorio2 --collection vehiculos --file "C:/Users/lijv1/OneDrive/Desktop/data_lab
02/vehiculos.json" --jsonArray
2024-02-07T17:06:43.792-0600    connected to: mongodb://localhost/
2024-02-07T17:06:43.807-0600    1000 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\lijv1> |
```

c.

```
laboratorio2> db.createCollection("peliculas")
{ ok: 1 }
laboratorio2> |
```

```
PS C:\Users\lijv1> mongoimport --db laboratorio2 --collection peliculas --file "C:/Users/lijv1/OneDrive/Desktop/data_lab
02/peliculas.json" --jsonArray
2024-02-07T17:14:12.261-0600    connected to: mongodb://localhost/
2024-02-07T17:14:12.277-0600    1000 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\lijv1> |
```

d.

```
laboratorio2> db.getCollectionInfos()
[
  {
    name: 'peliculas',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: UUID('50ea29c9-a766-4082-aa68-9d16e219e416')
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' },
  },
  {
    name: 'vehiculos',
    type: 'collection',
    options: {},
    info: {
      readOnly: false,
      uuid: UUID('69121f1c-4227-4a68-975e-5d6143433af5')
    },
    idIndex: { v: 2, key: { _id: 1 }, name: '_id_' },
  }
]
laboratorio2> |
```

## 2.2.

a.

```
laboratorio2> db.vehiculos.updateMany({}, {$rename: {"Id-del-Coche": "__id", "Marca-del-Coche": "brand", "Modelo-del-Coche": "model", "Año-del-Coche": "year", "Precio-del-Coche": "price"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2000,
  modifiedCount: 1000,
  upsertedCount: 0
}
laboratorio2> |
```

b.

```
laboratorio2> db.vehiculos.aggregate([ {$match: { $and: [ { price: { $ne: "" } }], { price: { $type: "string" } } }, { $expr: { $gt: [ { $toDouble: { $substr: [ "$price", 1, -1 ] } }, 0 ] } } ] } }, { $group: { _id: { marca: "$brand", ano: "$year" }, cantidad_modelos: { $sum: 1 }, precio_promedio: { $avg: { $toDouble: { $substr: [ "$price", 1, -1 ] } } } } }, { $project: { _id: 0, marca: "$_id.marca", ano: "$_id.ano", cantidad_modelos: 1, precio_promedio: 1 } }, { $sort: { marca: 1, ano: 1 } }, { $out: "vehiculos_brand_stats" } ] )
```

```
C:\Users\lijv1>mongoexport --db laboratorio2 --collection vehiculos_brand_stats --out C:\Users\lijv1\OneDrive\Desktop\data_lab02\vehiculos_brand_stats.json --type=json --jsonArray
2024-02-08T12:45:20.760-0600 connected to: mongod://localhost/
2024-02-08T12:45:20.795-0600 exported 566 records
```

OneDrive > Linda - Personal > Desktop > data\_lab02

Name	Status	Date modified	Type	Size
.DS_Store	⊖	2/1/2024 5:31 PM	DS_STORE File	7 KB
peliculas.json	⊕	2/7/2024 5:13 PM	JSON Source File	172 KB
vehiculos.json	⊕	2/7/2024 5:05 PM	JSON Source File	129 KB
vehiculos_brand_stats.json	⊕	2/8/2024 12:45 PM	JSON Source File	68 KB

```
vehiculos_brand_stats.json X
C:\Users\lijv1> OneDrive > Desktop > data_lab02 > vehiculos_b
1 Click here to ask Blackbox to help you code faster
2 {
3   "cantidad_modelos": 1,
4   "precio_promedio": 24665.86,
5   "marca": "Acura",
6   "ano": 1988
7 },
8 {
9   "cantidad_modelos": 1,
10  "precio_promedio": 29319.81,
11  "marca": "Acura",
12  "ano": 1995
13 },
14 {
15  "cantidad_modelos": 2,
16  "precio_promedio": 25802.199999999997,
17  "marca": "Acura",
18  "ano": 1996
19 },
20 {
21  "cantidad_modelos": 1,
22  "precio_promedio": 22507.42,
23  "marca": "Acura",
24  "ano": 1997
25 },
26 {
27  "cantidad_modelos": 1,
28  "precio_promedio": 21863.81,
29  "marca": "Acura",
30  "ano": 1998
31 },
32 {
33  "cantidad_modelos": 2,
34  "precio_promedio": 25543.71,
35  "marca": "Acura",
36  "ano": 2001
```










C.











```
laboratorio2> db.vehiculos.aggregate([ { $match: { "year": { $gte: 1990, $lt: 2000 } } }, { $addFields: { price_numeric: { $toDouble: { $substr: [ "$price", 1, -1 ] } } } }, { $sort: { "price_numeric": -1 } }, { $limit: 20 }, { $project: { _id: 0, Marca: "$brand", Modelo: "$model", Ano: "$year", Precio: "$price" } }, { $out: "vehiculos_top_models" } ] )
```

```
C:\Users\lijv1>mongoexport --db laboratorio2 --collection vehiculos_top_models --out C:\Users\lijv1\OneDrive\Desktop\data_lab02\vehiculos_top_models.csv --type=csv --fields=_id,brand,model,year,price
2024-02-08T12:51:30.647-0600 connected to: mongodb://localhost/
2024-02-08T12:51:30.655-0600 exported 20 records
```

Syncing > Linda - Personal > Desktop > data\_lab02 Search data



Sort  View  

Name	Status	Date modified	Type	Size
 .DS_Store		2/1/2024 5:31 PM	DS_STORE File	7 KB
 peliculas.json		2/7/2024 5:13 PM	JSON Source File	172 KB
 vehiculos.json		2/7/2024 5:05 PM	JSON Source File	129 KB
 vehiculos_brand_stats.json		2/8/2024 12:45 PM	JSON Source File	68 KB
 vehiculos_top_models.csv		2/8/2024 12:51 PM	Microsoft Excel Co...	1 KB

_id	Marca	Modelo	Ano	Precio
ObjectId(65c52538dca5f64e36b9a370)	Dodge	Grand Caravan	1997	\$29,988.13
ObjectId(65c52538dca5f64e36b9a371)	Mazda	Protege	1994	\$29,969.05
ObjectId(65c52538dca5f64e36b9a372)	Ford	Aerostar	1990	\$29,938.16
ObjectId(65c52538dca5f64e36b9a373)	Mazda	B-Series	1991	\$29,914.93
ObjectId(65c52538dca5f64e36b9a374)	Volkswagen	Cabriolet	1991	\$29,913.70
ObjectId(65c52538dca5f64e36b9a375)	Chevrolet	Malibu	1998	\$29,884.29
ObjectId(65c52538dca5f64e36b9a376)	Lincoln	Continental	1998	\$29,849.33
ObjectId(65c52538dca5f64e36b9a377)	Isuzu	Amigo	1999	\$29,842.58
ObjectId(65c52538dca5f64e36b9a378)	Toyota	MR2	1994	\$29,821.23
ObjectId(65c52538dca5f64e36b9a379)	Volkswagen	rio	1998	\$29,782.08
ObjectId(65c52538dca5f64e36b9a37a)	Saab	9000	1999	\$29,670.84
ObjectId(65c52538dca5f64e36b9a37b)	Pontiac	Grand Am	1994	\$29,599.30
ObjectId(65c52538dca5f64e36b9a37c)	Buick	Riviera	1999	\$29,586.95
ObjectId(65c52538dca5f64e36b9a37d)	Dodge	Shadow	1992	\$29,549.25
ObjectId(65c52538dca5f64e36b9a37e)	Dodge	Daytona	1992	\$29,542.29
ObjectId(65c52538dca5f64e36b9a37f)	Ford	Bronco	1991	\$29,493.53
ObjectId(65c52538dca5f64e36b9a380)	Mercury	Tracer	1994	\$29,333.25
ObjectId(65c52538dca5f64e36b9a381)	Acura	NSX	1995	\$29,319.81
ObjectId(65c52538dca5f64e36b9a382)	Land Rover	Discovery	1998	\$29,173.14
ObjectId(65c52538dca5f64e36b9a383)	Plymouth	Neon	1995	\$29,164.44