BD2: Trabajo Práctico 3

Indice

Parte 1: Bases de Datos NoSQL y Relacionales

1. ¿Cuáles de los siguientes conceptos de RDBMS existen en MongoDB? En caso de no existir, ¿hay alguna alternativa? ¿Cuál es?

- **Base de Datos**
- **Tabla / Relación**
- **Fila / Tupla**
- **Columna**

Base de datos: Si existe. Una instancia de MongoDB tiene 0 o más bases de datos.

Tabla / Relación: Las Tablas/Relaciones no existen como tal en MongoDB. Pero existen las colecciones, que en cierta forma se corresponden con las mismas. En una Base de datos MongoDB puede haber 0 o más colecciones, a su vez cada colección puede contener otro tipo de entidades identificadas como documentos.

Fila / Tupla: Similar al caso anterior, en MongoDB no existe como tal el concepto de Fila o Tupla. Pero como alternativa a los mismos existen los documentos. Los cuales cumplen la función de almacenar grupos de datos relacionados y se almacenan en formato BSON (JSON binario).

Columna: Como no existen las tablas entonces tampoco se tiene el concepto de columna. Teniendo en cuenta el formato de almacenamiento en documentos, los cuales se corresponden de cierta forma con los JSON, entonces la alternativa en MongoDB para hablar de un atributo particular son los “campos”.

2. MongoDB tiene soporte para transacciones, pero no es igual que el de los RDBMS. ¿Cuál es el alcance de una transacción en MongoDB?

En lo que abarca fuera del alcance, las características **ACID** no son garantizadas por las bases de datos que utilizan NoSQL. Si bien el concepto ACID es meramente para bases de datos relacionales, debemos considerar diversos conceptos asociados al mismo pero para base de datos NoSQL, Existe un acrónimo análogo a ACID para NoSQL que prima disponibilidad frente a consistencia. Dicho concepto es **BASE** (Basically Available, Soft state, Eventual consistency).

- Basically Available (Disponibilidad Básica): el sistema responderá aún con datos viejos.
- Soft State (Estado de Soft): el estado puede que no sea estable y puede ser corregido.
- Eventually Consistency (Consistencia Eventual): se garantiza que al cabo de un tiempo el sistema quedará en un estado consistente, es decir, que eventualmente se conseguirá un estado consistente.

BASE tiene sus raíces en el teorema de consistencia, disponibilidad y tolerancia a la partición (**CAP**) de Eric Brewer , y la consistencia eventual es la base de cualquier sistema distribuido que tenga como objetivo proporcionar alta disponibilidad y tolerancia a la partición.

BASE esencialmente acepta el hecho de que la verdadera consistencia no se puede lograr en el mundo real y, como tal, no se puede modelar en sistemas distribuidos altamente escalables. El teorema establece que es imposible para un sistema distribuido garantizar simultáneamente estas tres características, es decir, no puede tener más de dos de estas tres características simultáneamente. Por eso se dividen sistemas distribuidos de acuerdo a estos términos:

- AP (Disponibilidad y Tolerancia a fallos): Cassandra y CouchDB.
- CP (Consistencia y Tolerancia a Fallos): HBase, Paxos, **MongoDB**.
- CA (Tolerancia a fallos y consistencia): RDBMS.

3. Para acelerar las consultas, MongoDB tiene soporte para índices. ¿Qué tipos de índices soporta?

En MongoDB los índices se definen a nivel de las “collections”. Se soportan índices por cualquier campo o subcampo del documento.
Como primer índice tenemos que MongoDB crea un índice sobre el campo _id durante la creación de una colección. Este índice controla la unicidad del valor del _id y no puede ser eliminado.
Aparte de este, se sabe que MongoDB soportan los siguientes tipos de índices:

- Single field: Como el nombre indica son índices que se aplican a un solo campo de una colección. Puede estar ordenado de forma ascendente o descendente.
- Compound index: Este índice acepta varios campos, ordenando la colección según cada campo en el orden que estos se hayan especificado, cada campo a su vez se puede especificar si se lo quiere de forma ascendente o descendente.
- Multikey index: Como también indica el nombre, este tipo de índice permite asignar a un campo más de una clave.
- Geospatial index: Este tipo de índice permite asignar a los campos datos tanto en formato de par de coordenadas como de GeoJSON.
- Text index: Este tipo de índice permite admitir consultas de búsqueda de texto en contenido de cadena. Los índices pueden incluir cualquier campo cuyo valor sea una cadena o una matriz de elementos de cadena. Una colección solo puede tener un índice de búsqueda de texto, pero ese índice puede cubrir varios campos.
- Hashed index: Utilizan una función hash para realizar la búsqueda sobre los documentos. Pero no acepta índices de varias claves.

4. ¿Existen claves foráneas en MongoDB?

No, no existen. Ya que al ser una base de datos no relacional no se admite el uso de claves foráneas. Sin embargo MongoDB admite el uso de referencias, que sirven para relacionar varios documentos, sin embargo no pareciera que puedan llegar a restringir operaciones de borrado o actualización como si lo pueden hacer las claves foráneas.

Parte 2: Primeros pasos con MongoDB

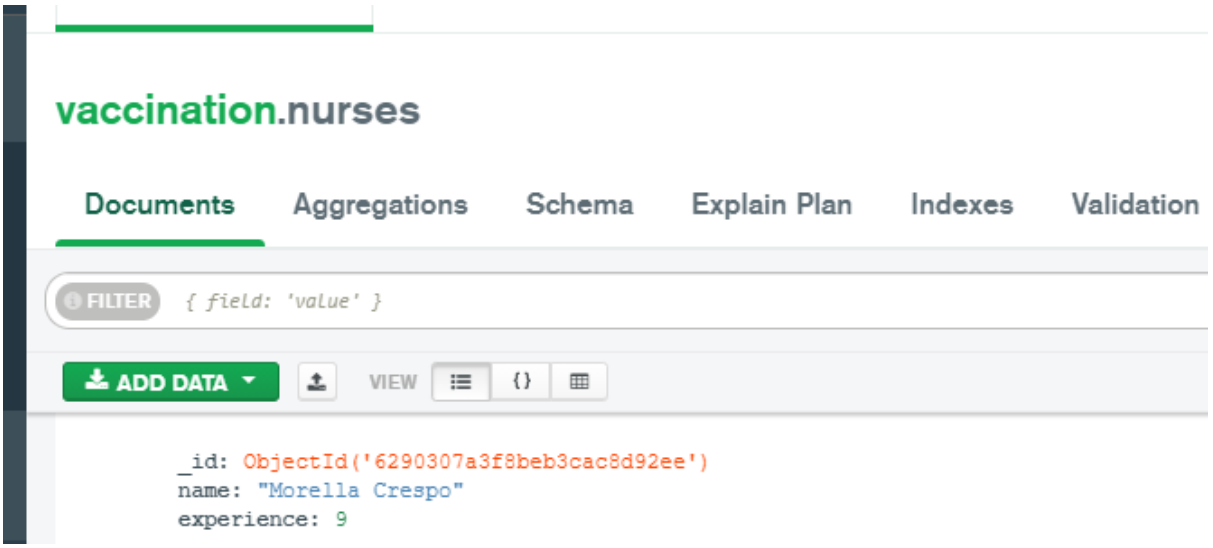
5. Cree una nueva base de datos llamada vaccination, y una colección llamada nurses. En esa colección inserte un nuevo documento (una enfermera) con los siguientes atributos:

{name:'Morella Crespo', experience:9}

recupere la información de la enfermera usando el comando db.nurses.find() (puede agregar la función .pretty() al final de la expresión para ver los datos indentados). Notará que no se encuentran exactamente los atributos que insertó. ¿Cuál es la diferencia? Se refiere al atributo id.

Primero creamos una la base de datos en mongoDB e insertamos el documento indicado:

```
> use vaccination
switched to db vaccination
> db.nurses.insert({name : "Morella Crespo", experience : 9})
WriteResult({ "nInserted" : 1 })
```



Luego con el cliente ejecutamos los comandos indicados:

```
> db.nurses.find()
{ "_id" : ObjectId("6290307a3f8beb3cac8d92ee"), "name" : "Morella Crespo", "experience" :
9 }
> db.nurses.find().pretty()
{
        "_id" : ObjectId("6290307a3f8beb3cac8d92ee"),
        "name" : "Morella Crespo",
        "experience" : 9
}
```

En el documento que devuelve el comando find, además de los campos que hemos creado, aparece uno denominado id. Este campo lo genera MongoDB automáticamente, siempre que no lo especifiquemos en la inserción. Digamos que es como la clave principal que todo documento debe tener. Todos los documentos tienen este campo y tiene que ser único. El campo id, como cualquier campo de MongoDB, puede ser de cualquier tipo. Podemos insertar números, texto o como hace MongoDB un ObjectId. Una vez insertado, el campo id no se puede modificar.

6. Agregue los siguientes documentos a la colección de enfermeros:

{name:'Gale Molina', experience:8, vaccines: ['AZ', 'Moderna']}
{name:'Honoria Fernández', experience:5, vaccines: ['Pfizer', 'Moderna', 'Sputnik V']}
{name:'Gonzalo Gallardo', experience:3}
{name:'Altea Parra', experience:6, vaccines: ['Pfizer']}

Y busque los enfermeros:

- **de 5 años de experiencia o menos**
- **que hayan aplicado la vacuna "Pfizer"**
- **que no hayan aplicado vacunas (es decir, que el atributo vaccines esté ausente)**
- **de apellido 'Fernández'**
- **con 6 o más años de experiencia y que hayan aplicado la vacuna 'Moderna'**
- **vuelva a realizar la última consulta pero proyecte sólo el nombre del enfermero/a en los resultados, omitiendo incluso el atributo _id de la proyección.**

Primero agregamos los documentos indicados:

```
> db.nurses.insert({name : "Gale Molina", experience : 8, vaccines : ["AZ", "Moderna"]})
WriteResult({ "nInserted" : 1 })
> db.nurses.insert({name : "Honoria Fernández", experience : 5, vaccines : ["Pfizer",
"Moderna", "Sputnik V"]})
WriteResult({ "nInserted" : 1 })
> db.nurses.insert({name : "Gonzalo Gallardo", experience : 3})
WriteResult({ "nInserted" : 1 })
> db.nurses.insert({name : "Altea Parra", experience : 6, vaccines : ["Pfizer"]})
WriteResult({ "nInserted" : 1 })
```

(tener en cuenta que aparte de los enfermeros que acabamos de insertar, también tenemos a "Morella Crespo" que añadimos en el punto 5)

Ahora hacemos las búsquedas solicitadas:

- **de 5 años de experiencia o menos**

```
> db.nurses.find({"experience": { $lte: 5 }}).pretty()
{
        "_id" : ObjectId("628ffb759ae37401713de563"),
```

```
        "name" : "Honoria Fernández",
        "experience" : 5,
        "vaccines" : [
                "Pfizer",
                "Moderna",
                "Sputnik V"
        ]
}
{
        "_id" : ObjectId("628ffbc49ae37401713de564"),
        "name" : "Gonzalo Gallardo",
        "experience" : 3
}
```

- **que hayan aplicado la vacuna “Pfizer”**

```
> db.nurses.find({"vaccines":"Pfizer"}).pretty()
{
        "_id" : ObjectId("628ffb759ae37401713de563"),
        "name" : "Honoria Fernández",
        "experience" : 5,
        "vaccines" : [
                "Pfizer",
                "Moderna",
                "Sputnik V"
        ]
}
{
        "_id" : ObjectId("628ffbe89ae37401713de565"),
        "name" : "Altea Parra",
        "experience" : 6,
        "vaccines" : [
                "Pfizer"
        ]
}
```

- **que no hayan aplicado vacunas (es decir, que el atributo vaccines esté ausente)**

```
> db.nurses.find({vaccines: null})
{ "_id" : ObjectId("6290307a3f8beb3cac8d92ee"), "name" : "Morella Crespo", "experience" :
9 }
{ "_id" : ObjectId("6290346d5ef4e7e8c912d0fa"), "name" : "Gonzalo Gallardo", "experience" :
3 }
```

- **de apellido ‘Fernández’**

```
> db.nurses.find({"name": /.*Fernandez.*/}).pretty()
{
	"_id" : ObjectId("628f1fe0cd9ae917a650d7b4"),
	"name" : "Honoria Fernandez",
	"experience" : 5,
	"vaccines" : [
		'"Pfizer"',
		'"Moderna"',
		"Sputnik V"
	]
}
```

- **con 6 o más años de experiencia y que hayan aplicado la vacuna ‘Moderna’ vuelva a realizar la última consulta pero proyecte sólo el nombre del enfermero/a en los resultados, omitiendo incluso el atributo _id de la proyección.**

```
> db.nurses.find({"experience":{"$gte":6},"vaccines":"Moderna"}).pretty()
{
	"_id" : ObjectId("628f1e65cd9ae917a650d7b3"),
	"name" : "Gale Molina",
	"experience" : 8,
	"vaccines" : [
		"Pfizer",
		"Moderna"
	]
}
```

7. Actualice a “Gale Molina” cambiándole la experiencia a 9 años.

```
> db.nurses.update({name: "Gale Molina"}, {$set: {experience : 9}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.nurses.find({name: "Gale Molina"}).pretty()
{
	"_id" : ObjectId("62903fdd487bdba936979ce6"),
	"name" : "Gale Molina",
	"experience" : 9,
	"vaccines" : [
		"AZ",
		"Moderna"
	]
}
```

8. Cree el array de vacunas (vaccines) para ”Gonzalo Gallardo".

```
> db.nurses.update({name: "Gonzalo Gallardo"}, {$set: {vaccines : []}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.nurses.find({"name": "Gonzalo Gallardo"}).pretty()
{
	"_id" : ObjectId("628ffbc49ae37401713de564"),
	"name" : "Gonzalo Gallardo",
	"experience" : 3,
	"vaccines" : [ ]
}
```

9. Agregue “AZ” a las vacunas de “Altea Parra”.

```
> db.nurses.update({name:"Altea Parra"}, {$push: {vaccines : "AZ"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.nurses.find({name: "Altea Parra"}).pretty()
{
	"_id" : ObjectId("629034c35ef4e7e8c912d0fd"),
	"name" : "Altea Parra",
	"experience" : 6,
	"vaccines" : [
		"Pfizer",
		"AZ"
	]
}
```

10. Duplique la experiencia de todos los enfermeros que hayan aplicado la vacuna “Pfizer”

```
> db.nurses.update({name: "Honoria Fernández"}, {$set: {experience : 5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.nurses.updateMany({"vaccines":"Pfizer"}, {$mul: {"experience": 2}})
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }
> db.nurses.find({"vaccines":"Pfizer"}).pretty()
{
	"_id" : ObjectId("628ffb759ae37401713de563"),
	"name" : "Honoria Fernández",
	"experience" : 10,
	"vaccines" : [
		"Pfizer",
		"Moderna",
		"Sputnik V"
	]
}
{
	"_id" : ObjectId("628ffbe89ae37401713de565"),
```

```
    "name" : "Altea Parra",
    "experience" : 12,
    "vaccines" : [
        "Pfizer"
    ]
}
```

Parte 3: Índices

▸ Elimine a todos los enfermeros de la colección. Guarde en un archivo llamado 'generador.js' el siguiente código JavaScript y ejecútelo con: load(). Si utiliza un cliente que lo permita (ej. Robo3T), se puede ejecutar directamente en el espacio de consultas.

```
var vaccines = ['AZ', 'Pfizer', 'Moderna', 'Sputnik V', "Johnson"];
for (var i = 1; i <= 2000; i++) {
var randomVax = vaccines.sort( function() { return 0.5 -
Math.random() } ).slice(1, Math.floor(Math.random() * 5));
var randomExperience = Math.ceil(2+(Math.random() * 20 - 2));
db.nurses.insert({
name:'Enfermero '+i,
experience:randomExperience,
tags: randomVax
});}
var patientNumber = 0;
for (var i = 1; i <= 2000; i++) {
var dosesCount = 50 + Math.ceil(Math.random() * 100);
for (var r = 1; r <= dosesCount; r++){
var randomLong = -34.56 - (Math.random() * .23);
var randomLat = -58.4 - (Math.random() * .22);
db.patients.insert({
 name:'Paciente '+patientNumber,
 address: {type: "Point",coordinates: [randomLat, randomLong]}
});
patientNumber++;
var year = 2020 + Math.round(Math.random());
var month = Math.ceil(Math.random() * 12);
var day = Math.ceil(Math.random() * 28);
db.doses.insert({
 nurse:'Enfermero '+i,
 patient:'Paciente '+patientNumber,
 vaccine: vaccines[Math.floor(Math.random()*vaccines.length)],
 date: new Date(year+'-'+month+'-'+day)
});
```

```
} }
```

Primero eliminamos los enfermeros que había en vaccination:

```
> db.nurses.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 5 }
```

Acto seguido cargamos la base de datos desde una copia del archivo dado:

```
> load("C:\\Users\\santi\\Downloads\\generador.js")
true
```

11. Busque en la colección de compras (doses) si existe algún índice definido.

```
> db.doses.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
```

12. Cree un índice para el campo nurse de la colección doses. Busque las dosis que tengan en el nombre del enfermero el string “11” y utilice el método explain("executionStats") al final de la consulta, para comparar la cantidad de documentos examinados y el tiempo en milisegundos de la consulta con y sin índice.

Buscamos las dosis sin índice:

```
> db.doses.find({"nurse": /11/}).explain("executionStats")
{
	"explainVersion" : "1",
	"queryPlanner" : {
		"namespace" : "vaccination.doses",
		"indexFilterSet" : false,
		"parsedQuery" : {
			"nurse" : {
				"$regex" : "11"
			}
		},
		"maxIndexedOrSolutionsReached" : false,
		"maxIndexedAndSolutionsReached" : false,
		"maxScansToExplodeReached" : false,
		"winningPlan" : {
			"stage" : "COLLSCAN",
			"filter" : {
				"nurse" : {
					"$regex" : "11"
				}
			},
			"direction" : "forward"
```

```
        },
        "rejectedPlans" : [ ]
    },
    "executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 12766,
        "executionTimeMillis" : 142,
        "totalKeysExamined" : 0,
        "totalDocsExamined" : 198624,
        "executionStages" : {
            "stage" : "COLLSCAN",
            "filter" : {
                "nurse" : {
                    "$regex" : "11"
                }
            },
            "nReturned" : 12766,
            "executionTimeMillisEstimate" : 4,
            "works" : 198626,
            "advanced" : 12766,
            "needTime" : 185859,
            "needYield" : 0,
            "saveState" : 198,
            "restoreState" : 198,
            "isEOF" : 1,
            "direction" : "forward",
            "docsExamined" : 198624
        }
    },
    "command" : {
        "find" : "doses",
        "filter" : {
            "nurse" : /11/
        },
        "$db" : "vaccination"
    },
    "serverInfo" : {
        "host" : "LAPTOP-LRQ8T2N7",
        "port" : 27017,
        "version" : "5.0.8",
        "gitVersion" : "c87e1c23421bf79614baf500fda6622bd90f674e"
    },
    "serverParameters" : {
        "internalQueryFacetBufferSizeBytes" : 104857600,
        "internalQueryFacetMaxOutputDocSizeBytes" : 104857600,
        "internalLookupStageIntermediateDocumentMaxSizeBytes" : 104857600,
        "internalDocumentSourceGroupMaxMemoryBytes" : 104857600,
        "internalQueryMaxBlockingSortMemoryUsageBytes" : 104857600,
        "internalQueryProhibitBlockingMergeOnMongoS" : 0,
        "internalQueryMaxAddToSetBytes" : 104857600,
        "internalDocumentSourceSetWindowFieldsMaxMemoryBytes" : 104857600
```

```
	},
	"ok" : 1
}
```

Podemos comprobar que el tiempo de ejecución sin índices fue 142 milisegundos y examinó 198624 documentos.

Creamos el índice:

```
> db.doses.createIndex({"nurse":1})
{
	"numIndexesBefore" : 1,
	"numIndexesAfter" : 2,
	"createdCollectionAutomatically" : false,
	"ok" : 1
}
```

Buscamos con índice:

```
> db.doses.find({"nurse": /11/}).explain("executionStats")
{
	"explainVersion" : "1",
	"queryPlanner" : {
		"namespace" : "vaccination.doses",
		"indexFilterSet" : false,
		"parsedQuery" : {
			"nurse" : {
				"$regex" : "11"
			}
		},
		"maxIndexedOrSolutionsReached" : false,
		"maxIndexedAndSolutionsReached" : false,
		"maxScansToExplodeReached" : false,
		"winningPlan" : {
			"stage" : "FETCH",
			"inputStage" : {
				"stage" : "IXSCAN",
				"filter" : {
					"nurse" : {
						"$regex" : "11"
					}
				},
				"keyPattern" : {
					"nurse" : 1
				},
				"indexName" : "nurse_1",
				"isMultiKey" : false,
				"multiKeyPaths" : {
					"nurse" : [ ]
				},
				"isUnique" : false,
```

```
                "isSparse" : false,
                "isPartial" : false,
                "indexVersion" : 2,
                "direction" : "forward",
                "indexBounds" : {
                    "nurse" : [
                        "[\"\", {})",
                        "[/11/, /11/]"
                    ]
                }
            }
        },
        "rejectedPlans" : [ ]
    },
    "executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 12766,
        "executionTimeMillis" : 197,
        "totalKeysExamined" : 198624,
        "totalDocsExamined" : 12766,
        "executionStages" : {
            "stage" : "FETCH",
            "nReturned" : 12766,
            "executionTimeMillisEstimate" : 9,
            "works" : 198625,
            "advanced" : 12766,
            "needTime" : 185858,
            "needYield" : 0,
            "saveState" : 198,
            "restoreState" : 198,
            "isEOF" : 1,
            "docsExamined" : 12766,
            "alreadyHasObj" : 0,
            "inputStage" : {
                "stage" : "IXSCAN",
                "filter" : {
                    "nurse" : {
                        "$regex" : "11"
                    }
                },
                "nReturned" : 12766,
                "executionTimeMillisEstimate" : 6,
                "works" : 198625,
                "advanced" : 12766,
                "needTime" : 185858,
                "needYield" : 0,
                "saveState" : 198,
                "restoreState" : 198,
                "isEOF" : 1,
                "keyPattern" : {
                    "nurse" : 1
```

```
                        },
                        "indexName" : "nurse_1",
                        "isMultiKey" : false,
                        "multiKeyPaths" : {
                              "nurse" : [ ]
                        },
                        "isUnique" : false,
                        "isSparse" : false,
                        "isPartial" : false,
                        "indexVersion" : 2,
                        "direction" : "forward",
                        "indexBounds" : {
                              "nurse" : [
                                    "[\"\", {})",
                                    "[/11/, /11/]"
                              ]
                        },
                        "keysExamined" : 198624,
                        "seeks" : 1,
                        "dupsTested" : 0,
                        "dupsDropped" : 0
                  }
            }
      },
      "command" : {
            "find" : "doses",
            "filter" : {
                  "nurse" : /11/
            },
            "$db" : "vaccination"
      },
      "serverInfo" : {
            "host" : "LAPTOP-LRQ8T2N7",
            "port" : 27017,
            "version" : "5.0.8",
            "gitVersion" : "c87e1c23421bf79614baf500fda6622bd90f674e"
      },
      "serverParameters" : {
            "internalQueryFacetBufferSizeBytes" : 104857600,
            "internalQueryFacetMaxOutputDocSizeBytes" : 104857600,
            "internalLookupStageIntermediateDocumentMaxSizeBytes" : 104857600,
            "internalDocumentSourceGroupMaxMemoryBytes" : 104857600,
            "internalQueryMaxBlockingSortMemoryUsageBytes" : 104857600,
            "internalQueryProhibitBlockingMergeOnMongoS" : 0,
            "internalQueryMaxAddToSetBytes" : 104857600,
            "internalDocumentSourceSetWindowFieldsMaxMemoryBytes" : 104857600
      },
      "ok" : 1
}
```

Podemos comprobar que el tiempo de ejecución con índices índices fue 197 milisegundos y examinó 12766 documentos.

Armamos una tabla para comparar los resultados:

Ejeccion	Tiempo en Milisegundos	Cantidad de documentos Examinados
Sin índice	142	198624
Con Índice	197	12766

El tiempo en ejecución con y sin índices no parece variar demasiado, e incluso sin índices pareciera ser un poco mejor (capaz por el indice por defecto "_id_", por el equipo que usamos o incluso por la naturaleza de la propia consulta, que busca cualquier cadena con 11, y no solo aquellas que empiezan con 11); Respecto a la cantidad de documentos examinados la diferencia si es mucho más notable, teniendo la consulta sin índice que examinar más de 10 veces la cantidad de documentos que la versión con índice.

13. Busque los pacientes que viven dentro de la ciudad de Buenos Aires. Para esto, puede definir una variable en la terminal y asignarle como valor el polígono del archivo provisto caba.geojson (copiando y pegando directamente). Cree un índice geoespacial de tipo 2dsphere para el campo location de la colección patients y, de la misma forma que en el punto 12, compare la performance de la consulta con y sin dicho índice.

```
> var buenosAires = {
...  "type":"MultiPolygon",
...  "coordinates":[[[
...    [-58.46305847167969,-34.53456089748654],
...    [-58.49979400634765,-34.54983198845187],
...    [-58.532066345214844,-34.614561581608186],
...    [-58.528633117675774,-34.6538270014492],
...    [-58.48674774169922,-34.68742794931483],
...    [-58.479881286621094,-34.68206400648744],
...    [-58.46855163574218,-34.65297974261105],
...    [-58.465118408203125,-34.64733112904415],
...    [-58.4585952758789,-34.63998735602951],
...    [-58.45344543457032,-34.63603274732642],
...    [-58.447265625,-34.63575026806082],
...    [-58.438339233398445,-34.63038297923296],
...    [-58.38100433349609,-34.62162507826766],
...    [-58.38237762451171,-34.59251960889388],
...    [-58.378944396972656,-34.5843230246475],
...    [-58.46305847167969,-34.53456089748654]
...  ]]]
... }
```

Búsqueda sin indice:

```
> db.patients.find({address: { $geoWithin: { $geometry: buenosAires
}}}).explain("executionStats")
{
	"explainVersion" : "1",
	"queryPlanner" : {
		"namespace" : "vaccination.patients",
		"indexFilterSet" : false,
		"parsedQuery" : {
			"address" : {
				"$geoWithin" : {
					"$geometry" : {
						"type" : "MultiPolygon",
						"coordinates" : [
							[
								[
									[
-58.46305847167969,
-34.53456089748654
],
[
-58.49979400634765,
-34.54983198845187
],
[
-58.532066345214844,
-34.614561581608186
],
[
-58.528633117675774,
-34.6538270014492
],
[
-58.48674774169922,
-34.68742794931483
],
[
-58.479881286621094,
-34.68206400648744
],
[
-58.46855163574218,
-34.65297974261105
],
[
-58.465118408203125,
-34.64733112904415
],
[
-58.4585952758789,
-34.63998735602951
```

```
                                    ],
                                    [
                                        -58.45344543457032,
                                        -34.63603274732642
                                    ],
                                    [
                                        -58.447265625,
                                        -34.63575026806082
                                    ],
                                    [
                                        -58.438339233398445,
                                        -34.63038297923296
                                    ],
                                    [
                                        -58.38100433349609,
                                        -34.62162507826766
                                    ],
                                    [
                                        -58.38237762451171,
                                        -34.59251960889388
                                    ],
                                    [
                                        -58.378944396972656,
                                        -34.5843230246475
                                    ],
                                    [
                                        -58.46305847167969,
                                        -34.53456089748654
                                    ]
                                ]
                            ]
                        ]
                    }
                }
            }
        },
        "maxIndexedOrSolutionsReached" : false,
        "maxIndexedAndSolutionsReached" : false,
        "maxScansToExplodeReached" : false,
        "winningPlan" : {
            "stage" : "COLLSCAN",
            "filter" : {
                "address" : {
                    "$geoWithin" : {
                        "$geometry" : {
                            "type" : "MultiPolygon",
                            "coordinates" : [
                                [
                                    [
                                        [
                                            -58.46305847167969,
```

```
            -34.53456089748654
        ],
        [
            -58.49979400634765,
            -34.54983198845187
        ],
        [
            -58.532066345214844,
            -34.614561581608186
        ],
        [
            -58.528633117675774,
            -34.6538270014492
        ],
        [
            -58.48674774169922,
            -34.68742794931483
        ],
        [
            -58.479881286621094,
            -34.68206400648744
        ],
        [
            -58.46855163574218,
            -34.65297974261105
        ],
        [
            -58.465118408203125,
            -34.64733112904415
        ],
        [
            -58.4585952758789,
            -34.63998735602951
        ],
        [
            -58.45344543457032,
            -34.63603274732642
        ],
        [
            -58.447265625,
            -34.63575026806082
        ],
        [
            -58.438339233398445,
            -34.63038297923296
        ],
        [
            -58.38100433349609,
            -34.62162507826766
        ],
        [
```

```
                                        -58.38237762451171,
                                        -34.59251960889388
                                    ],
                                    [
                                        -58.378944396972656,
                                        -34.5843230246475
                                    ],
                                    [
                                        -58.46305847167969,
                                        -34.53456089748654
                                    ]
                                ]
                            ]
                        ]
                    }
                }
            }
        },
        "direction" : "forward"
    },
    "rejectedPlans" : [ ]
},
"executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 43543,
    "executionTimeMillis" : 746,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 198624,
    "executionStages" : {
        "stage" : "COLLSCAN",
        "filter" : {
            "address" : {
                "$geoWithin" : {
                    "$geometry" : {
                        "type" : "MultiPolygon",
                        "coordinates" : [
                            [
                                [
                                    [
                                        -58.46305847167969,
                                        -34.53456089748654
                                    ],
                                    [
                                        -58.49979400634765,
                                        -34.54983198845187
                                    ],
                                    [
                                        -58.532066345214844,
                                        -34.614561581608186
                                    ],
                                    [
```

```
                -58.528633117675774,
                -34.6538270014492
            ],
            [
                -58.48674774169922,
                -34.68742794931483
            ],
            [
                -58.479881286621094,
                -34.68206400648744
            ],
            [
                -58.46855163574218,
                -34.65297974261105
            ],
            [
                -58.465118408203125,
                -34.64733112904415
            ],
            [
                -58.4585952758789,
                -34.63998735602951
            ],
            [
                -58.45344543457032,
                -34.63603274732642
            ],
            [
                -58.447265625,
                -34.63575026806082
            ],
            [
                -58.438339233398445,
                -34.63038297923296
            ],
            [
                -58.38100433349609,
                -34.62162507826766
            ],
            [
                -58.38237762451171,
                -34.59251960889388
            ],
            [
                -58.378944396972656,
                -34.5843230246475
            ],
            [
                -58.46305847167969,
                -34.53456089748654
            ]
```

```
                                        ]
                                    ]
                                ]
                            }
                        }
                    }
                },
                "nReturned" : 43543,
                "executionTimeMillisEstimate" : 94,
                "works" : 198626,
                "advanced" : 43543,
                "needTime" : 155082,
                "needYield" : 0,
                "saveState" : 198,
                "restoreState" : 198,
                "isEOF" : 1,
                "direction" : "forward",
                "docsExamined" : 198624
            }
        },
        "command" : {
            "find" : "patients",
            "filter" : {
                "address" : {
                    "$geoWithin" : {
                        "$geometry" : {
                            "type" : "MultiPolygon",
                            "coordinates" : [
                                [
                                    [
                                        [
                                            -58.46305847167969,
                                            -34.53456089748654
                                        ],
                                        [
                                            -58.49979400634765,
                                            -34.54983198845187
                                        ],
                                        [
                                            -58.532066345214844,
                                            -34.614561581608186
                                        ],
                                        [
                                            -58.528633117675774,
                                            -34.6538270014492
                                        ],
                                        [
                                            -58.48674774169922,
                                            -34.68742794931483
                                        ],
                                        [
```

```
                                    -58.479881286621094,
                                    -34.68206400648744
                                ],
                                [
                                    -58.46855163574218,
                                    -34.65297974261105
                                ],
                                [
                                    -58.465118408203125,
                                    -34.64733112904415
                                ],
                                [
                                    -58.4585952758789,
                                    -34.63998735602951
                                ],
                                [
                                    -58.45344543457032,
                                    -34.63603274732642
                                ],
                                [
                                    -58.447265625,
                                    -34.63575026806082
                                ],
                                [
                                    -58.438339233398445,
                                    -34.63038297923296
                                ],
                                [
                                    -58.38100433349609,
                                    -34.62162507826766
                                ],
                                [
                                    -58.38237762451171,
                                    -34.59251960889388
                                ],
                                [
                                    -58.378944396972656,
                                    -34.5843230246475
                                ],
                                [
                                    -58.46305847167969,
                                    -34.53456089748654
                                ]
                            ]
                        ]
                    ]
                }
            }
        }
    },
    "$db" : "vaccination"
```

```
	},
	"serverInfo" : {
		"host" : "LAPTOP-LRQ8T2N7",
		"port" : 27017,
		"version" : "5.0.8",
		"gitVersion" : "c87e1c23421bf79614baf500fda6622bd90f674e"
	},
	"serverParameters" : {
		"internalQueryFacetBufferSizeBytes" : 104857600,
		"internalQueryFacetMaxOutputDocSizeBytes" : 104857600,
		"internalLookupStageIntermediateDocumentMaxSizeBytes" : 104857600,
		"internalDocumentSourceGroupMaxMemoryBytes" : 104857600,
		"internalQueryMaxBlockingSortMemoryUsageBytes" : 104857600,
		"internalQueryProhibitBlockingMergeOnMongoS" : 0,
		"internalQueryMaxAddToSetBytes" : 104857600,
		"internalDocumentSourceSetWindowFieldsMaxMemoryBytes" : 104857600
	},
	"ok" : 1
}
```


Podemos comprobar que el tiempo de ejecución sin índices fue 742 milisegundos y examinó 198624 documentos.

Creamos el índice:

```
> db.patients.createIndex({ address: "2dsphere"})
{
	"numIndexesBefore" : 1,
	"numIndexesAfter" : 2,
	"createdCollectionAutomatically" : false,
	"ok" : 1
}
```

Buscamos con índice:

```
> db.patients.find({address: { $geoWithin: { $geometry: buenosAires
}}}).explain("executionStats")
{
	"explainVersion" : "1",
	"queryPlanner" : {
		"namespace" : "vaccination.patients",
		"indexFilterSet" : false,
		"parsedQuery" : {
			"address" : {
				"$geoWithin" : {
					"$geometry" : {
						"type" : "MultiPolygon",
						"coordinates" : [
							[
								[
									[
										-58.46305847167969,
```

```
            -34.53456089748654
        ],
        [
            -58.49979400634765,
            -34.54983198845187
        ],
        [
            -58.532066345214844,
            -34.614561581608186
        ],
        [
            -58.528633117675774,
            -34.6538270014492
        ],
        [
            -58.48674774169922,
            -34.68742794931483
        ],
        [
            -58.479881286621094,
            -34.68206400648744
        ],
        [
            -58.46855163574218,
            -34.65297974261105
        ],
        [
            -58.465118408203125,
            -34.64733112904415
        ],
        [
            -58.4585952758789,
            -34.63998735602951
        ],
        [
            -58.45344543457032,
            -34.63603274732642
        ],
        [
            -58.447265625,
            -34.63575026806082
        ],
        [
            -58.438339233398445,
            -34.63038297923296
        ],
        [
            -58.38100433349609,
            -34.62162507826766
        ],
        [
            -58.38237762451171,
            -34.59251960889388
        ],
```

```
                                        [
                                            -58.378944396972656,
                                            -34.5843230246475
                                        ],
                                        [
                                            -58.46305847167969,
                                            -34.53456089748654
                                        ]
                                    ]
                                ]
                            ]
                        }
                    }
                }
            },
            "maxIndexedOrSolutionsReached" : false,
            "maxIndexedAndSolutionsReached" : false,
            "maxScansToExplodeReached" : false,
            "winningPlan" : {
                "stage" : "FETCH",
                "filter" : {
                    "address" : {
                        "$geoWithin" : {
                            "$geometry" : {
                                "type" : "MultiPolygon",
                                "coordinates" : [
                                    [
                                        [
                                            [
                                                -58.46305847167969,
                                                -34.53456089748654
                                            ],
                                            [
                                                -58.49979400634765,
                                                -34.54983198845187
                                            ],
                                            [
                                                -58.532066345214844,
                                                -34.614561581608186
                                            ],
                                            [
                                                -58.528633117675774,
                                                -34.6538270014492
                                            ],
                                            [
                                                -58.48674774169922,
                                                -34.68742794931483
                                            ],
                                            [
                                                -58.479881286621094,
                                                -34.68206400648744
                                            ],
                                            [
                                                -58.46855163574218,
```

```
                                        -34.65297974261105
                                    ],
                                    [
                                        -58.465118408203125,
                                        -34.64733112904415
                                    ],
                                    [
                                        -58.4585952758789,
                                        -34.63998735602951
                                    ],
                                    [
                                        -58.45344543457032,
                                        -34.63603274732642
                                    ],
                                    [
                                        -58.447265625,
                                        -34.63575026806082
                                    ],
                                    [
                                        -58.438339233398445,
                                        -34.63038297923296
                                    ],
                                    [
                                        -58.38100433349609,
                                        -34.62162507826766
                                    ],
                                    [
                                        -58.38237762451171,
                                        -34.59251960889388
                                    ],
                                    [
                                        -58.378944396972656,
                                        -34.5843230246475
                                    ],
                                    [
                                        -58.46305847167969,
                                        -34.53456089748654
                                    ]
                                ]
                            ]
                        ]
                    }
                }
            }
        },
        "inputStage" : {
            "stage" : "IXSCAN",
            "keyPattern" : {
                "address" : "2dsphere"
            },
            "indexName" : "address_2dsphere",
            "isMultiKey" : false,
            "multiKeyPaths" : {
                "address" : [ ]
```

```
                },
                "isUnique" : false,
                "isSparse" : false,
                "isPartial" : false,
                "indexVersion" : 2,
                "direction" : "forward",
                "indexBounds" : {
                        "address" : [
                                "[-7710162562058289152, -7710162562058289152]",
                                "[-7660622966157213696, -7660622966157213696]",
                                "[-7657245266436685824, -7657245266436685824]",
                                "[-7657051752390197248, -7657051752390197248]",
                                "[-7657047354343686144, -7657047354343686144]",
                                "[-7657047354343686143, -7657046804587872257]",
                                "[-7657046254832058368, -7657046254832058368]",
                                "[-7657046220472319999, -7657046186112581633]",
                                "[-7657046186112581632, -7657046186112581632]",
                                "[-7657045979954151424, -7657045979954151424]",
                                "[-7657045911234674688, -7657045911234674688]",
                                "[-7657045876874936319, -7657045842515197953]",
                                "[-7657045842515197951, -7657045705076244481]",
                                "[-7657045705076244479, -7657045155320430593]",
                                "[-7657045155320430591, -7657044605564616705]",
                                "[-7657044605564616703, -7657044055808802817]",
                                "[-7657044055808802816, -7657044055808802816]",
                                "[-7657044055808802815, -7657044021449064449]",
                                "[-7657043987089326080, -7657043987089326080]",
                                "[-7657043780930895872, -7657043780930895872]",
                                "[-7657043506052988927, -7657042956297175041]",
                                "[-7657034160204152832, -7657034160204152832]",
                                "[-7657025295391653888, -7657025295391653888]",
                                "[-7657025243852046336, -7657025243852046336]",
                                "[-7657025230967144448, -7657025230967144448]",
                                "[-7657025227745918976, -7657025227745918976]",
                                "[-7657025226940612608, -7657025226940612608]",
                                "[-7657025226739286016, -7657025226739286016]",
                                "[-7657025226688954368, -7657025226688954368]",
                                "[-7657025226680565759, -7657025226672177153]",
                                "[-7657025226672177151, -7657025089233223681]",
                                "[-7657025089233223680, -7657025089233223680]",
                                "[-7657025089233223679, -7657024951794270209]",
                                "[-7657024539477409792, -7657024539477409792]",
                                "[-7657024402038456319, -7657024264599502849]",
                                "[-7657024264599502848, -7657024264599502848]",
                                "[-7657024264599502847, -7657023714843688961]",
                                "[-7657023714843688959, -7657023165087875073]",
                                "[-7657023165087875071, -7657022615332061185]",
                                "[-7657022615332061183, -7657022065576247297]",
                                "[-7657022065576247296, -7657022065576247296]",
                                "[-7657022065576247295, -7657021928137293825]",
                                "[-7657021928137293823, -7657021790698340353]",
                                "[-7657021790698340352, -7657021790698340352]",
                                "[-7657021240942526464, -7657021240942526464]",
                                "[-7657021172223049728, -7657021172223049728]",
```

```
                                "[-7657021155043180544, -7657021155043180544]",
                                "[-7657021155043180543, -7657021146453245953]",
                                "[-7657020966064619520, -7657020966064619520]",
                                "[-7657016568018108416, -7657016568018108416]",
                                "[-7656963791459975168, -7656963791459975168]",
                                "[-7656119366529843200, -7656119366529843200]"
                        ]
                }
            }
        },
        "rejectedPlans" : [ ]
    },
    "executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 43543,
        "executionTimeMillis" : 268,
        "totalKeysExamined" : 55185,
        "totalDocsExamined" : 55166,
        "executionStages" : {
            "stage" : "FETCH",
            "filter" : {
                "address" : {
                    "$geoWithin" : {
                        "$geometry" : {
                            "type" : "MultiPolygon",
                            "coordinates" : [
                                [
                                    [
                                        [
                                            -58.46305847167969,
                                            -34.53456089748654
                                        ],
                                        [
                                            -58.49979400634765,
                                            -34.54983198845187
                                        ],
                                        [
                                            -58.532066345214844,
                                            -34.614561581608186
                                        ],
                                        [
                                            -58.528633117675774,
                                            -34.6538270014492
                                        ],
                                        [
                                            -58.48674774169922,
                                            -34.68742794931483
                                        ],
                                        [
                                            -58.479881286621094,
                                            -34.68206400648744
                                        ],
                                        [
                                            -58.46855163574218,
```

```
                                        -34.65297974261105
                                    ],
                                    [
                                        -58.465118408203125,
                                        -34.64733112904415
                                    ],
                                    [
                                        -58.4585952758789,
                                        -34.63998735602951
                                    ],
                                    [
                                        -58.45344543457032,
                                        -34.63603274732642
                                    ],
                                    [
                                        -58.447265625,
                                        -34.63575026806082
                                    ],
                                    [
                                        -58.438339233398445,
                                        -34.63038297923296
                                    ],
                                    [
                                        -58.38100433349609,
                                        -34.62162507826766
                                    ],
                                    [
                                        -58.38237762451171,
                                        -34.59251960889388
                                    ],
                                    [
                                        -58.378944396972656,
                                        -34.5843230246475
                                    ],
                                    [
                                        -58.46305847167969,
                                        -34.53456089748654
                                    ]
                                ]
                            ]
                        ]
                    }
                }
            }
        },
        "nReturned" : 43543,
        "executionTimeMillisEstimate" : 51,
        "works" : 55186,
        "advanced" : 43543,
        "needTime" : 11642,
        "needYield" : 0,
        "saveState" : 55,
        "restoreState" : 55,
        "isEOF" : 1,
```

```
"docsExamined" : 55166,
"alreadyHasObj" : 0,
"inputStage" : {
        "stage" : "IXSCAN",
        "nReturned" : 55166,
        "executionTimeMillisEstimate" : 21,
        "works" : 55186,
        "advanced" : 55166,
        "needTime" : 19,
        "needYield" : 0,
        "saveState" : 55,
        "restoreState" : 55,
        "isEOF" : 1,
        "keyPattern" : {
                "address" : "2dsphere"
        },
        "indexName" : "address_2dsphere",
        "isMultiKey" : false,
        "multiKeyPaths" : {
                "address" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
                "address" : [
                        "[-7710162562058289152, -7710162562058289152]",
                        "[-7660622966157213696, -7660622966157213696]",
                        "[-7657245266436685824, -7657245266436685824]",
                        "[-7657051752390197248, -7657051752390197248]",
                        "[-7657047354343686144, -7657047354343686144]",
                        "[-7657047354343686143, -7657046804587872257]",
                        "[-7657046254832058368, -7657046254832058368]",
                        "[-7657046220472319999, -7657046186112581633]",
                        "[-7657046186112581632, -7657046186112581632]",
                        "[-7657045979954151424, -7657045979954151424]",
                        "[-7657045911234674688, -7657045911234674688]",
                        "[-7657045876874936319, -7657045842515197953]",
                        "[-7657045842515197951, -7657045705076244481]",
                        "[-7657045705076244479, -7657045155320430593]",
                        "[-7657045155320430591, -7657044605564616705]",
                        "[-7657044605564616703, -7657044055808802817]",
                        "[-7657044055808802816, -7657044055808802816]",
                        "[-7657044055808802815, -7657044021449064449]",
                        "[-7657043987089326080, -7657043987089326080]",
                        "[-7657043780930895872, -7657043780930895872]",
                        "[-7657043506052988927, -7657042956297175041]",
                        "[-7657034160204152832, -7657034160204152832]",
                        "[-7657025295391653888, -7657025295391653888]",
                        "[-7657025243852046336, -7657025243852046336]",
                        "[-7657025230967144448, -7657025230967144448]",
                        "[-7657025227745918976, -7657025227745918976]",
```

```
                                "[-7657025226940612608, -7657025226940612608]",
                                "[-7657025226739286016, -7657025226739286016]",
                                "[-7657025226688954368, -7657025226688954368]",
                                "[-7657025226680565759, -7657025226672177153]",
                                "[-7657025226672177151, -7657025089233223681]",
                                "[-7657025089233223680, -7657025089233223680]",
                                "[-7657025089233223679, -7657024951794270209]",
                                "[-7657024539477409792, -7657024539477409792]",
                                "[-7657024402038456319, -7657024264599502849]",
                                "[-7657024264599502848, -7657024264599502848]",
                                "[-7657024264599502847, -7657023714843688961]",
                                "[-7657023714843688959, -7657023165087875073]",
                                "[-7657023165087875071, -7657022615332061185]",
                                "[-7657022615332061183, -7657022065576247297]",
                                "[-7657022065576247296, -7657022065576247296]",
                                "[-7657022065576247295, -7657021928137293825]",
                                "[-7657021928137293823, -7657021790698340353]",
                                "[-7657021790698340352, -7657021790698340352]",
                                "[-7657021240942526464, -7657021240942526464]",
                                "[-7657021172223049728, -7657021172223049728]",
                                "[-7657021155043180544, -7657021155043180544]",
                                "[-7657021155043180543, -7657021146453245953]",
                                "[-7657020966064619520, -7657020966064619520]",
                                "[-7657016568018108416, -7657016568018108416]",
                                "[-7656963791459975168, -7656963791459975168]",
                                "[-7656119366529843200, -7656119366529843200]"
                        ]
                },
                "keysExamined" : 55185,
                "seeks" : 20,
                "dupsTested" : 0,
                "dupsDropped" : 0
        }
    }
},
"command" : {
    "find" : "patients",
    "filter" : {
        "address" : {
            "$geoWithin" : {
                "$geometry" : {
                    "type" : "MultiPolygon",
                    "coordinates" : [
                        [
                            [
                                [
                                    -58.46305847167969,
                                    -34.53456089748654
                                ],
                                [
                                    -58.49979400634765,
                                    -34.54983198845187
                                ],
                                [
```

```
                -58.532066345214844,
                -34.614561581608186
            ],
            [
                -58.528633117675774,
                -34.6538270014492
            ],
            [
                -58.48674774169922,
                -34.68742794931483
            ],
            [
                -58.479881286621094,
                -34.68206400648744
            ],
            [
                -58.46855163574218,
                -34.65297974261105
            ],
            [
                -58.465118408203125,
                -34.64733112904415
            ],
            [
                -58.4585952758789,
                -34.63998735602951
            ],
            [
                -58.45344543457032,
                -34.63603274732642
            ],
            [
                -58.447265625,
                -34.63575026806082
            ],
            [
                -58.438339233398445,
                -34.63038297923296
            ],
            [
                -58.38100433349609,
                -34.62162507826766
            ],
            [
                -58.38237762451171,
                -34.59251960889388
            ],
            [
                -58.378944396972656,
                -34.5843230246475
            ],
            [
                -58.46305847167969,
                -34.53456089748654
```

```
                                        ]
                                    ]
                                ]
                            ]
                        }
                    }
                }
            },
            "$db" : "vaccination"
        },
        "serverInfo" : {
            "host" : "LAPTOP-LRQ8T2N7",
            "port" : 27017,
            "version" : "5.0.8",
            "gitVersion" : "c87e1c23421bf79614baf500fda6622bd90f674e"
        },
        "serverParameters" : {
            "internalQueryFacetBufferSizeBytes" : 104857600,
            "internalQueryFacetMaxOutputDocSizeBytes" : 104857600,
            "internalLookupStageIntermediateDocumentMaxSizeBytes" : 104857600,
            "internalDocumentSourceGroupMaxMemoryBytes" : 104857600,
            "internalQueryMaxBlockingSortMemoryUsageBytes" : 104857600,
            "internalQueryProhibitBlockingMergeOnMongoS" : 0,
            "internalQueryMaxAddToSetBytes" : 104857600,
            "internalDocumentSourceSetWindowFieldsMaxMemoryBytes" : 104857600
        },
        "ok" : 1
}
```


Podemos comprobar que el tiempo de ejecución sin índices fue 268 milisegundos y examinó 55166 documentos.

Armamos una tabla para comparar los resultados:

Ejeccion	Tiempo en Milisegundos	Cantidad de documentos Examinados
Sin índice	268	198624
Con Índice	746	55166

En esta ocasión el tiempo de en ejecución con y sin índices si varió de forma significativa, la consulta con índice nos muestra un SpeedUp de 2.7 respecto a la versión de la consulta sin índice; Respecto a la cantidad de documentos examinados la diferencia también (es notable, aunque no tanto como en el punto 12) la consulta sin índice examinó cerca de 4 veces más documentos que la versión con índice.

Parte 4: Aggregation Framework

▸MongoDB cuenta con un Aggregation Framework que brinda la posibilidad de hacer analítica en tiempo real del estilo OLAP (Online Analytical Processing), de forma similar a otros productos específicos como Hadoop o MapReduce. En los siguientes ejercicios se verán algunos ejemplos de su aplicabilidad.

14. Obtenga 5 pacientes aleatorios de la colección.

Ejecutamos la consulta sobre la DB que nos quedó de la parte 3 del TP:

```
> db.patients.aggregate([{ $sample: { size: 5 }}])
{ "_id" : ObjectId("629a85c3b120280fe954eddd"), "name" : "Paciente 182261", "address" : {
"type" : "Point", "coordinates" : [ -58.6121066872813, -34.65961644311656 ] } }
{ "_id" : ObjectId("629a8523b120280fe95026a3"), "name" : "Paciente 25688", "address" : {
"type" : "Point", "coordinates" : [ -58.467908738679604, -34.6287938700389 ] } }
{ "_id" : ObjectId("629a856fb120280fe9526ca1"), "name" : "Paciente 100183", "address" : {
"type" : "Point", "coordinates" : [ -58.43587393532566, -34.71946821951674 ] } }
{ "_id" : ObjectId("629a8507b120280fe94f600b"), "name" : "Paciente 268", "address" : {
"type" : "Point", "coordinates" : [ -58.43204258726681, -34.65273661333029 ] } }
{ "_id" : ObjectId("629a859fb120280fe953e0c5"), "name" : "Paciente 147817", "address" : {
"type" : "Point", "coordinates" : [ -58.53389285629706, -34.78978811666088 ] } }
```

15. Usando el framework de agregación, obtenga los pacientes que vivan a 1km (o menos) del centro geográfico de la ciudad de Buenos Aires ([-58.4586,-34.5968]) y guárdelos en una nueva colección.

Ejecutamos la consulta sobre la DB que nos quedó de la parte 3 del TP:

```
> db.patients.aggregate([{ $geoNear: { maxDistance: 100, distanceField: "distance", near:
{type: "Point", coordinates: [-58.4586,-34.5968]}, }}, {$out: "punto15"} ])
```

16. Obtenga una colección de las dosis aplicadas a los pacientes del punto anterior. Note que sólo es posible ligarlas por el nombre del paciente.

```
> db.punto15.aggregate([ { $lookup: { from: "doses", localField: "name", foreignField:
"patient", as: "dosis" }}, { $project: { dosis:1, _id:0 } } ])
{ "dosis" : [ { "_id" : ObjectId("629a85a1b120280fe953ebe6"), "nurse" : "Enfermero 1507",
"patient" : "Paciente 149242", "vaccine" : "Pfizer", "date" : ISODate("2020-08-17T00:00:00Z")
} ] }
{ "dosis" : [ { "_id" : ObjectId("629a858bb120280fe9534e28"), "nurse" : "Enfermero 1301",
"patient" : "Paciente 129051", "vaccine" : "Moderna", "date" :
ISODate("2021-03-20T00:00:00Z") } ] }
{ "dosis" : [ { "_id" : ObjectId("629a85bab120280fe954b09a"), "nurse" : "Enfermero 1760",
"patient" : "Paciente 174420", "vaccine" : "Johnson", "date" :
ISODate("2020-12-09T00:00:00Z") } ] }
{ "dosis" : [ { "_id" : ObjectId("629a85b2b120280fe9547864"), "nurse" : "Enfermero 1686",
"patient" : "Paciente 167225", "vaccine" : "AZ", "date" : ISODate("2020-09-23T00:00:00Z") } ]
}
```

{ "dosis" : [{ "_id" : ObjectId("629a859cb120280fe953c666"), "nurse" : "Enfermero 1460", "patient" : "Paciente 144442", "vaccine" : "Pfizer", "date" : ISODate("2020-08-16T00:00:00Z") }] }
{ "dosis" : [{ "_id" : ObjectId("629a8519b120280fe94fd880"), "nurse" : "Enfermero 156", "patient" : "Paciente 15687", "vaccine" : "Pfizer", "date" : ISODate("2020-06-04T00:00:00Z") }] }
{ "dosis" : [{ "_id" : ObjectId("629a855fb120280fe951fafc"), "nurse" : "Enfermero 863", "patient" : "Paciente 85637", "vaccine" : "Sputnik V", "date" : ISODate("2021-08-01T00:00:00Z") }] }
{ "dosis" : [{ "_id" : ObjectId("629a85c1b120280fe954e354"), "nurse" : "Enfermero 1824", "patient" : "Paciente 180913", "vaccine" : "Moderna", "date" : ISODate("2020-05-17T00:00:00Z") }] }
{ "dosis" : [{ "_id" : ObjectId("629a858bb120280fe9534aac"), "nurse" : "Enfermero 1295", "patient" : "Paciente 128605", "vaccine" : "Pfizer", "date" : ISODate("2020-01-01T00:00:00Z") }] }

▸ Si la consulta se empieza a tornar difícil de leer, se pueden ir guardando los agregadores en variables, que no son más que objetos en formato JSON.

17. Obtenga una nueva colección de nurses, cuyos nombres incluyan el string “111”. En cada documento (cada nurse) se debe agregar un atributo doses que consista en un array con todas las dosis que aplicó después del 1/5/2021

```
> db.nurses.aggregate([ {$match: { name: /111/ } }, { $lookup: {from: "doses", foreignField: "nurse", localField: "name", pipeline: [{ $match:{date: {"$gte" : ISODate("2021-05-01T00:00:00.000Z")}}}}], as: "doses"}}]).pretty()
{
	"_id" : ObjectId("62a3991c6ce7f292260bddea"),
	"name" : "Enfermero 111",
	"experience" : 10,
	"tags" : [
		"Johnson",
		"Sputnik V"
	],
	"doses" : [
		{
			"_id" : ObjectId("62a399246ce7f292260c3ba5"),
			"nurse" : "Enfermero 111",
			"patient" : "Paciente 11053",
			"vaccine" : "Sputnik V",
			"date" : ISODate("2021-06-03T00:00:00Z")
		},
		{
			"_id" : ObjectId("62a399246ce7f292260c3ba7"),
			"nurse" : "Enfermero 111",
… (no anotamos las demas lineas por que sino queda muy largo el doc)
```