Facultad de Informática – Universidad Nacional de La Plata

Cátedra de Bases de Datos II

Bases de Datos 2 2022 - TP3 - Grupo 3

Bases de Datos NoSQL / Práctica con MongoDB

entrega: 13/6

Grupo 3 - Integrantes

Gerónimo Boza	13134/5
Julián Casaburi	14471/6
Facundo Tomas Fabbi	15875/7

Parte 1: Bases de Datos NoSQL y Relacionales

▶ Si bien las BBDD NoSQL tienen diferencias fundamentales con los sistemas de BBDD Relacionales o RDBMS, algunos conceptos comunes se pueden relacionar. Responda las siguientes preguntas, considerando MongoDB en particular como Base de Datos NoSQL.

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

   - **Base de Datos:** El concepto existe. Una instancia de MongoDB tiene 0 o más bases de datos y en una base de datos de MongoDB puede haber 0 o más colecciones.
   - **Tabla / Relación:** El concepto de tabla en sí no existe, pero es reemplazado por el de **colecciones**. Las relaciones pueden ser modeladas como **documentos embebidos** o a través de **referencias**.
   - **Fila / Tupla:** A lo que se conoce como fila en los RDBMS, en MongoDB se lo conoce como **documento**, el cual se almacena en formato BSON (JSON binario).
   - **Columna:** En MongoDB, no existe el concepto de columnas, ya que las colecciones de MongoDB no tienen un esquema por defecto, es decir, **los documentos de una colección no necesitan tener el mismo conjunto de campos** y el tipo de datos para un campo puede diferir entre los documentos dentro de una colección. La alternativa en MongoDB para hablar de un atributo particular es un “campo”.

Fuente:
https://www.mongodb.com/docs/manual/core/databases-and-collections/

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 MongoDB las operaciones realizadas sobre un solo documento son atómicas, aún si la operación involucra múltiples documentos embebidos dentro del mismo.
Por otra parte si se realizara una modificación sobre varios documentos (ej: db.collection.updateMany()), la modificación de cada uno de los documentos es atómica, pero no será atómica la operación como un todo.
Por otra parte, MongoDB tiene soporte para transacciones atómicas a nivel multi-documento a partir de la versión 4.0. Desafortunadamente las transacciones de este tipo conllevan un gran costo en cuanto al rendimiento de las operaciones realizadas.

Fuentes:
https://docs.mongodb.com/manual/core/transactions/
https://www.mongodb.com/docs/manual/core/data-modeling-introduction/

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 colecciones. Se soportan índices por cualquier campo o subcampo del documento.
De forma predeterminada, todas las colecciones tienen un índice de tipo **"Single Field Index"** en el campo **_id**. Este índice controla la unicidad del valor del _id y no puede ser eliminado.

- **Single Field Index**: 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:** Permite la creación de un índice basado en múltiples campos del documento. El orden de los campos de un índice compuesto tiene importancia. Por ejemplo, si un índice compuesto consiste en **{ userid: 1, score: -1 }**, el índice ordena primero por userid y luego, dentro de cada valor de userid, ordena por score. A su vez se puede especificar por cada campo el orden (forma ascendente o descendente).
- **Multikey index**: Son índices compuestos por matrices, MongoDB crea además índices para cada uno de los elementos de la matriz. Los índices multiclave se pueden construir sobre matrices que contienen valores escalares (por ejemplo, strings, números) y documentos anidados.
- **Geospatial index:** Este tipo de índices se utiliza para indexar coordenadas geoespaciales (formato de par de coordenadas o GeoJSON).
- **Text index:** Este tipo de índice permite consultas de búsqueda de texto. 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:** Son índices formados por el valor del hash de un dato en un campo.

Fuente:

https://www.mongodb.com/docs/manual/indexes/

4. ¿Existen claves foráneas en MongoDB?

No, en MongoDB no existen claves foráneas. Dado esto nunca habrá restricciones al borrado o a la modificación, o una operación "en cascada" como resultado.

Sin embargo, en MongoDB se pueden utilizar los documentos embebidos para representar las relaciones entre los datos, pero también se pueden utilizar mecanismos de multi-documentos para obtener modelos normalizados. En este último caso, las relaciones se representan mediante referencias entre los documentos, parecidas al sistema de claves foráneas de las bases de datos relacionales.

Fuentes:

https://docs.mongodb.com/manual/core/data-model-design/#data-modeling-embedding
https://docs.mongodb.com/manual/core/data-model-design/#data-modeling-referencing
https://docs.mongodb.com/manual/reference/database-references/

Parte 2: Primeros pasos con MongoDB

▶ Descargue la última versión de MongoDB desde el sitio oficial. Ingrese al cliente de línea de comando para realizar los siguientes ejercicios.

Nota
Para los ejercicios, realizamos los siguientes pasos:

1. Iniciamos el servicio de MongoDB:

```
$ sudo systemctl start mongodb
```

2. Accedemos al shell de MongoDB:

```
$ mongo
```

3. Ingresamos los comandos para realizar las operaciones correspondientes.

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?

```
> use vaccination
switched to db vaccination

> db.nurses.insertOne({name:"Morella Crespo", experience:9})
{
   "acknowledged" : true,
   "insertedId" : ObjectId("628c7ff29d3be00896eb442a")
}


> db.nurses.find().pretty()
{
      "_id" : ObjectId("628c7ff29d3be00896eb442a"),
      "name" : "Morella Crespo",
      "experience" : 9
}
```

Nota:
Las consultas pueden ser ejecutadas ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ5/EJ5-queries.js
```

Luego de crear el documento, se agregó el campo “_id”. Esto se debe a que cada documento almacenado en una colección requiere un campo _id, que debe ser único para la colección, que actúa como clave principal. Si se omite el campo _id al insertar el documento, MongoDB genera automáticamente un ObjectId para el campo _id.

▶ Una característica fundamental de MongoDB y otras bases NoSQL es que los documentos no tienen una estructura definida, como puede ser una tabla en un RDBMS. En una misma colección pueden convivir documentos con diferentes atributos, e incluso atributos de múltiples valores y documentos embebidos.

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.

Ejecutamos la siguiente **consulta** para **insertar los documentos** en la colección **nurses**:

```
use vaccination
db.nurses.insertMany([
    {
        "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"
        ]
    }
])
```

Nota:
La consulta de inserción puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ6/EJ6-insertMany.js
```

Como **resultado** obtuvimos:

```
{
	"acknowledged" : true,
	"insertedIds" : [
		ObjectId("628ef073afb9e45858ab869e"),
		ObjectId("628ef073afb9e45858ab869f"),
		ObjectId("628ef073afb9e45858ab86a0"),
		ObjectId("628ef073afb9e45858ab86a1")
	]
}
```

Nota:
Para los siguientes ejercicios, se deberá tener en cuenta que además de los documentos que acabamos de insertar, la colección nurses también cuenta con el documento insertado en el ejercicio 5, correspondiente al nurse “Morella Crespo”.

A. Enfermeros de 5 años de experiencia o menos

Consulta

```
use vaccination
db.nurses.find( { experience: { $lte: 5} })
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ6/EJ6-queryA.js
```

Resultado

```
{ "_id" : ObjectId("628ef073afb9e45858ab869f"), "name" : "Honoria Fernández",
```

```
"experience" : 5, "vaccines" : [ "Pfizer", "Moderna", "Sputnik V" ] }
{ "_id" : ObjectId("628ef073afb9e45858ab86a0"), "name" : "Gonzalo Gallardo",
"experience" : 3 }
```

B. Enfermeros Que hayan aplicado la vacuna “Pfizer”

Consulta

```
use vaccination
db.nurses.find( { vaccines: "Pfizer" } )
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ6/EJ6-queryB.js
```

Resultado

```
{ "_id" : ObjectId("628ef073afb9e45858ab869f"), "name" : "Honoria Fernández",
"experience" : 5, "vaccines" : [ "Pfizer", "Moderna", "Sputnik V" ] }
{ "_id" : ObjectId("628ef073afb9e45858ab86a1"), "name" : "Altea Parra",
"experience" : 6, "vaccines" : [ "Pfizer" ] }
```

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

Consulta

```
use vaccination
db.nurses.find( { vaccines: { $exists: false } } )
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ6/EJ6-queryC.js
```

Resultado

```
{ "_id" : ObjectId("628c7ff29d3be00896eb442a"), "name" : "Morella Crespo",
"experience" : 9 }
{ "_id" : ObjectId("628ef073afb9e45858ab86a0"), "name" : "Gonzalo Gallardo",
"experience" : 3 }
```

D. Enfermeros de apellido ‘Fernández’

Consulta

```
use vaccination
db.nurses.find( { name: { $regex: '\s*Fernández' } } )
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ6/EJ6-queryD.js
```

Resultado

```
{ "_id" : ObjectId("628ef073afb9e45858ab869f"), "name" : "Honoria Fernández",
"experience" : 5, "vaccines" : [ "Pfizer", "Moderna", "Sputnik V" ] }
```

E. Enfermeros con 6 o más años de experiencia y que hayan aplicado la vacuna ‘Moderna’

Consulta

```
use vaccination
db.nurses.find( { $and: [ { experience: { $gte: 6} }, { vaccines: "Moderna" }
] } )
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ6/EJ6-queryE.js
```

Resultado

```
{ "_id" : ObjectId("628ef073afb9e45858ab869e"), "name" : "Gale Molina",
"experience" : 8, "vaccines" : [ "AZ", "Moderna" ] }
```

F. 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.

Consulta

```
use vaccination
db.nurses.find( { $and: [ { experience: { $gte: 6} }, { vaccines: "Moderna" }
] }, {name:1, _id:0} )
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ6/EJ6-queryF.js
```

Resultado

```
{ "name" : "Gale Molina" }
```

▶ En MongoDB hay diferentes maneras de realizar actualizaciones, de acuerdo a las necesidades del esquema flexible de documentos.

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

Consulta

```
use vaccination
db.nurses.updateOne( { name: "Gale Molina" }, { $set: { experience: 9 } })
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ7/EJ7-query.js
```

Resultado

```
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Find del documento actualizado

```
> db.nurses.find({name: "Gale Molina"})

{ "_id" : ObjectId("628ef073afb9e45858ab869e"), "name" : "Gale Molina",
"experience" : 9, "vaccines" : [ "AZ", "Moderna" ] }
```

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

Consulta

```
use vaccination
db.nurses.updateOne( { name: "Gonzalo Gallardo" }, { $set: { vaccines: [] } }
})
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ8/EJ8-query.js
```

Resultado

```
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Find del documento actualizado

```
> db.nurses.find({name: "Gonzalo Gallardo"})

{ "_id" : ObjectId("628ef073afb9e45858ab86a0"), "name" : "Gonzalo Gallardo",
"experience" : 3, "vaccines" : [ ] }
```

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

Consulta

```
use vaccination;
db.nurses.updateOne( { name: "Altea Parra" }, { $push: { vaccines: "AZ" } })
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ9/EJ9-query.js
```

Resultado

```
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Find del documento actualizado

```
> db.nurses.find({name: "Altea Parra"})

{ "_id" : ObjectId("628ef073afb9e45858ab86a1"), "name" : "Altea Parra",
"experience" : 6, "vaccines" : [ "Pfizer", "AZ" ] }
```

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

Consulta

```
use vaccination;
```

```
db.nurses.updateMany( { vaccines: "Pfizer" }, {  $mul: { experience: 2 } } )
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ10/EJ10-query.js
```

Resultado

```
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }
```

Find de los documentos actualizados

```
{ "_id" : ObjectId("628ef073afb9e45858ab869f"), "name" : "Honoria Fernández",
"experience" : 10, "vaccines" : [ "Pfizer", "Moderna", "Sputnik V" ] }

{ "_id" : ObjectId("628ef073afb9e45858ab86a1"), "name" : "Altea Parra",
"experience" : 12, "vaccines" : [ "Pfizer", "AZ" ] }
```

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(<ruta del archivo ‘generador.js’>)`. 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)
                  });
            }}
```

En primer lugar, eliminamos a todos los enfermeros de la colección y luego ejecutamos el script **‘generador.js’**:

```
use vaccination;
db.nurses.remove({});
load("./tp3/generador.js")
```

Obtenemos como resultado:

```
true
```

Nota:
Estos pasos pueden ser realizados ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/parte3-remove-load.js
```

Comprobamos la **cantidad de documentos** de las colecciones **nurses**, **doses** y **patients** luego de las inserciones, con el método **db.collection.count()**:

```
> db.nurses.count()
2000

> db.doses.count()
199948

> db.patients.count()
199948
```

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

Consulta

```
use vaccination;
db.doses.getIndexes()
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ11/EJ11-query.js
```

Resultado

```
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
```

Observamos que el único **índice** definido para la colección doses es el índice **_id**, de tipo **single field**, creado por defecto por MongoDB.

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.

Creación del índice

```
use vaccination;
db.doses.createIndex({"nurse":1})
```

Nota:
La creación del índice puede ser realizada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ12/EJ12-createIndex.js
```

Resultado

```
{
	"numIndexesBefore" : 1,
	"numIndexesAfter" : 2,
	"createdCollectionAutomatically" : false,
	"ok" : 1
}
```

Una vez creado el índice, procedimos a hacer el find para encontrar a los enfermeros cuyo nombre contenga el string “11”:

Consulta

```
use vaccination;
db.doses.find({nurse: /11/}).explain("executionStats")
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ12/EJ12-findNurse.js
```

Para **comparar los tiempos** de los resultados antes y después de crear el **índice** en el campo **“nurse”**, la consulta se realizó dos veces, una antes de agregar el índice y una después. Podemos ver que hay una **diferencia** en los valores de **totalKeysExamined**, **totalDocsExamamined** y de **tiempo** al momento de hacer el find.

Tabla comparativa:

Índice	executionTimeMillis	totalKeysExamined	totalDocsExamined
Sin índice	88	0	199948
Con índice	107	199948	12827

Resultado

Antes de agregar el 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" : 12827,
            "executionTimeMillis" : 88,
            "totalKeysExamined" : 0,
            "totalDocsExamined" : 199948,
            "executionStages" : {
                  "stage" : "COLLSCAN",
                  "filter" : {
                        "nurse" : {
                              "$regex" : "11"
                        }
                  },
                  "nReturned" : 12827,
                  "executionTimeMillisEstimate" : 6,
                  "works" : 199950,
                  "advanced" : 12827,
                  "needTime" : 187122,
                  "needYield" : 0,
                  "saveState" : 199,
                  "restoreState" : 199,
                  "isEOF" : 1,
                  "direction" : "forward",
                  "docsExamined" : 199948
            }
      },
      "command" : {
            "find" : "doses",
            "filter" : {
```

```
                  "nurse" : /11/
            },
            "$db" : "vaccination"
      },
      "ok" : 1
}
```

Después de agregar el 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" : 12827,
            "executionTimeMillis" : 107,
            "totalKeysExamined" : 199948,
            "totalDocsExamined" : 12827,
            "executionStages" : {
                "stage" : "FETCH",
                "nReturned" : 12827,
                "executionTimeMillisEstimate" : 9,
                "works" : 199949,
                "advanced" : 12827,
                "needTime" : 187121,
                "needYield" : 0,
                "saveState" : 199,
                "restoreState" : 199,
                "isEOF" : 1,
                "docsExamined" : 12827,
                "alreadyHasObj" : 0,
                "inputStage" : {
                    "stage" : "IXSCAN",
                    "filter" : {
                        "nurse" : {
                            "$regex" : "11"
                        }
                    },
                    "nReturned" : 12827,
                    "executionTimeMillisEstimate" : 9,
                    "works" : 199949,
                    "advanced" : 12827,
                    "needTime" : 187121,
                    "needYield" : 0,
                    "saveState" : 199,
                    "restoreState" : 199,
                    "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" : 199948,
                        "seeks" : 1,
                        "dupsTested" : 0,
                        "dupsDropped" : 0
                  }
            }
      },
      "command" : {
            "find" : "doses",
            "filter" : {
                  "nurse" : /11/
            },
            "$db" : "vaccination"
      },
      "ok" : 1
}
```


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.

Creación del índice

```
use vaccination;
db.patients.createIndex({address:"2dsphere"})
```

Nota:
La creación del índice puede ser realizada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ13/EJ13-createIndex.js
```

Resultado

```
{
      "numIndexesBefore" : 1,
      "numIndexesAfter" : 2,
      "createdCollectionAutomatically" : false,
      "ok" : 1
}
```

Para **buscar a los pacientes** que viven en la ciudad de Buenos Aires:

1. Se guardó en una variable llamada **cabageo** el valor del archivo provisto (caba.geojson)

2. Se realizó un find

```
use vaccination;
cabageo = {
 "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.patients.find({address: {$geoWithin: {$geometry: cabageo } }
}).explain("executionStats")
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ13/EJ13-findPatients.js
```

Para **comparar los tiempos** de los resultados antes y después de crear el **índice** en el campo **“address”**, la consulta se realizó dos veces, una antes de agregar el índice y una después. Podemos ver que hay una **diferencia** en los valores de **totalKeysExamined**, **totalDocsExamamined** y de **tiempo** al momento de hacer el **find**.

Tabla comparativa:

Índice	executionTimeMillis	totalKeysExamined	totalDocsExamined
Sin índice	221	0	199948
Con índice	129	55505	55485

Para esta consulta el tiempo de ejecución con y sin índices sí varió de forma significativa: se observa una disminución porcentual del 41% en el tiempo de la consulta con índices.

Antes de agregar el índice:


```
{
	"executionStats" : {
		"executionSuccess" : true,
		"nReturned" : 43878,
		"executionTimeMillis" : 221,
		"totalKeysExamined" : 0,
		"totalDocsExamined" : 199948,
		"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" : 43878,
                                            "executionTimeMillisEstimate" : 23,
                                            "works" : 199950,
                                            "advanced" : 43878,
                                            "needTime" : 156071,
                                            "needYield" : 0,
                                            "saveState" : 199,
                                            "restoreState" : 199,
                                            "isEOF" : 1,
                                            "direction" : "forward",
                                            "docsExamined" : 199948
                                        }
                                    },
                                    "ok" : 1
}
```

Después de agregar el índice:


```
{
	"executionStats" : {
		"executionSuccess" : true,
		"nReturned" : 43878,
		"executionTimeMillis" : 129,
		"totalKeysExamined" : 55505,
		"totalDocsExamined" : 55485,
		"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" : 43878,
                                        "executionTimeMillisEstimate" : 26,
                                        "works" : 55506,
                                        "advanced" : 43878,
                                        "needTime" : 11627,
                                        "needYield" : 0,
                                        "saveState" : 55,
                                        "restoreState" : 55,
                                        "isEOF" : 1,
                                        "docsExamined" : 55485,
                                        "alreadyHasObj" : 0,
                                        "inputStage" : {
                                                "stage" : "IXSCAN",
                                                "nReturned" : 55485,
                                                "executionTimeMillisEstimate" : 2,
                                                "works" : 55506,
                                                "advanced" : 55485,
                                                "needTime" : 20,
                                                "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" : 55505,
                                "seeks" : 21,
                                "dupsTested" : 0,
                                "dupsDropped" : 0
                        }
                }
        },
        "ok" : 1
}
```

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.

Consulta

```
use vaccination;
db.patients.aggregate([{ $sample: { size: 5 } }])
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ14/EJ14-query.js
```

Resultado

```
{ "_id" : ObjectId("62953cd212e74eee8bd64c2c"), "name" : "Paciente 66781",
"address" : { "type" : "Point", "coordinates" : [ -58.417993476044394,
-34.61248086978896 ] } }
{ "_id" : ObjectId("62953cc812e74eee8bd572d4"), "name" : "Paciente 38961",
"address" : { "type" : "Point", "coordinates" : [ -58.44133000087789,
-34.775222543777 ] } }
{ "_id" : ObjectId("62953cfd12e74eee8bd9f364"), "name" : "Paciente 186489",
"address" : { "type" : "Point", "coordinates" : [ -58.46465914793321,
-34.595160747246965 ] } }
{ "_id" : ObjectId("62953cef12e74eee8bd8c760"), "name" : "Paciente 148087",
"address" : { "type" : "Point", "coordinates" : [ -58.51449613315493,
-34.58237808529344 ] } }
{ "_id" : ObjectId("62953cea12e74eee8bd86288"), "name" : "Paciente 135179",
"address" : { "type" : "Point", "coordinates" : [ -58.41448656988225,
-34.649990817149614 ] } }
```

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

Consulta

```
use vaccination;
db.patients.aggregate([
  {
    $geoNear: {
       near: { type: "Point", coordinates: [-58.4586,-34.5968] },
       distanceField: "distance.meters",
       maxDistance: 1000,
       spherical: true,
    }
   },
   { $project : { "distance": 0 } },
   { $out : "cabaPatients" }
])
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ15/EJ15-query.js
```

Resultado

En este caso **no se obtiene una respuesta luego de ejecutar la query**, por lo que decidimos realizar un **find en la nueva colección** (cabaPatients), creada por el stage $out de la agregación, para mostrar los documentos creados:

```
use vaccination;
db.cabaPatients.find().limit(5).pretty()
{
      "_id" : ObjectId("62953ccb12e74eee8bd5a580"),
      "name" : "Paciente 45447",
      "address" : {
            "type" : "Point",
            "coordinates" : [
                  -58.458047171970016,
                  -34.59677606366081
            ]
```

```
	}
}
{
	"_id" : ObjectId("62953cc312e74eee8bd4f9da"),
	"name" : "Paciente 23476",
	"address" : {
		"type" : "Point",
		"coordinates" : [
			-58.458416721901386,
			-34.597271639842894
		]
	}
}
{
	"_id" : ObjectId("62953cdd12e74eee8bd73c90"),
	"name" : "Paciente 97551",
	"address" : {
		"type" : "Point",
		"coordinates" : [
			-58.45933293498666,
			-34.59675315228669
		]
	}
}
{
	"_id" : ObjectId("62953ce812e74eee8bd82586"),
	"name" : "Paciente 127370",
	"address" : {
		"type" : "Point",
		"coordinates" : [
			-58.459188686196605,
			-34.597261769295486
		]
	}
}
{
	"_id" : ObjectId("62953cc912e74eee8bd578cc"),
	"name" : "Paciente 39725",
	"address" : {
		"type" : "Point",
		"coordinates" : [
			-58.45777818047949,
			-34.59646546671366
		]
	}
}
```

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.

▸ 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.

Consulta

```
use vaccination;
db.doses.aggregate([
   {
       $lookup:
           {
               from: "cabaPatients",
               localField: "patient",
               foreignField: "name",
               as: "luPatient"
           },
   },
   {
       $match: {
           "luPatient": {$ne: []}
       }
   },
   {
       $project: {"luPatient": 0}
   },
   { $out : "ej16Doses" }
]);
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ16/EJ16-query.js
```

Resultado

En este caso **no se obtiene una respuesta luego de ejecutar la query**, por lo que decidimos realizar un **find en la nueva colección** (ej16Doses), creada por el stage $out de la agregación, para mostrar los documentos creados:

```
use vaccination;
> db.ej16Doses.find().limit(2).pretty()
{
      "_id" : ObjectId("6292e272938a9b2f9f9ed9f0"),
      "nurse" : "Enfermero 1",
      "patient" : "Paciente 22",
```

```
        "vaccine" : "Moderna",
        "date" : ISODate("2020-11-13T00:00:00Z")
}
{
        "_id" : ObjectId("6292e272938a9b2f9f9edbbc"),
        "nurse" : "Enfermero 4",
        "patient" : "Paciente 252",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-03-04T00:00:00Z")
}
```

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.

Consulta

```
use vaccination;
db.nurses.aggregate(
   [
       {
           $match: { name: { $regex: '111' } }
       },
       {
           $lookup:
               {
                   from: "doses",
                   "let": {
                       n: "$name"
                   },
                   pipeline: [
                       {
                           "$match": {
                               $expr: {
                                   $and: [
                                       {
                                           $eq: [
                                               "$$n",
                                               "$nurse"
                                           ]
                                       },
                                       {
                                           $gt: [
                                               "$date",
                                               new
```

```
ISODate("2021-05-01T00:00:00.000Z")
                                                ]
                                            }
                                        ]
                                    }
                                }
                            }
                        ],
                        as: "doses"
                    }
        },
        {
            $out : "ej17Nurses"
        }
    ]
)
```

Nota:
La consulta puede ser ejecutada ingresando el siguiente comando en la terminal:

```
mongo < ./tp3/queries/EJ17/EJ17-query.js
```

Find sobre la nueva colección

Nota:
Mostramos 2 documentos de la colección **ej17Nurses**.

```
> db.ej17Nurses.find().limit(2).pretty()
{
      "_id" : ObjectId("6292e272938a9b2f9f9ed263"),
      "name" : "Enfermero 111",
      "experience" : 18,
      "tags" : [ ],
      "doses" : [
            {
                  "_id" : ObjectId("6292e276938a9b2f9f9f2dc8"),
                  "nurse" : "Enfermero 111",
                  "patient" : "Paciente 10754",
                  "vaccine" : "Sputnik V",
                  "date" : ISODate("2021-07-08T00:00:00Z")
            },
            {
                  "_id" : ObjectId("6292e276938a9b2f9f9f2dcc"),
                  "nurse" : "Enfermero 111",
                  "patient" : "Paciente 10756",
                  "vaccine" : "Moderna",
                  "date" : ISODate("2021-12-09T00:00:00Z")
```

```
        },
        {
                "_id" : ObjectId("6292e276938a9b2f9f9f2dd4"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10760",
                "vaccine" : "AZ",
                "date" : ISODate("2021-12-23T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e276938a9b2f9f9f2de8"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10770",
                "vaccine" : "Johnson",
                "date" : ISODate("2021-06-04T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e276938a9b2f9f9f2dea"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10771",
                "vaccine" : "Johnson",
                "date" : ISODate("2021-11-09T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e276938a9b2f9f9f2df2"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10775",
                "vaccine" : "Pfizer",
                "date" : ISODate("2021-10-27T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e276938a9b2f9f9f2df6"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10777",
                "vaccine" : "Sputnik V",
                "date" : ISODate("2021-12-25T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e276938a9b2f9f9f2dfa"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10779",
                "vaccine" : "Sputnik V",
                "date" : ISODate("2021-07-23T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e276938a9b2f9f9f2e00"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10782",
```

```
        "vaccine" : "Moderna",
        "date" : ISODate("2021-12-19T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e08"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10786",
        "vaccine" : "AZ",
        "date" : ISODate("2021-09-23T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e0c"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10788",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-12-14T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e0e"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10789",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-09-13T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e14"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10792",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-10-12T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e1e"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10797",
        "vaccine" : "AZ",
        "date" : ISODate("2021-07-18T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e26"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10801",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-08-19T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e2c"),
```

```
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10804",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-11-03T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e2e"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10805",
        "vaccine" : "AZ",
        "date" : ISODate("2021-05-17T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e36"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10809",
        "vaccine" : "AZ",
        "date" : ISODate("2021-10-10T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e42"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10815",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-05-07T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e44"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10816",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-10-28T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e46"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10817",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-11-14T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e276938a9b2f9f9f2e48"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10818",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-08-27T00:00:00Z")
    },
```

```
{
        "_id" : ObjectId("6292e276938a9b2f9f9f2e50"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10822",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-07-11T00:00:00Z")
},
{
        "_id" : ObjectId("6292e276938a9b2f9f9f2e52"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10823",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-11-02T00:00:00Z")
},
{
        "_id" : ObjectId("6292e276938a9b2f9f9f2e54"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10824",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-08-10T00:00:00Z")
},
{
        "_id" : ObjectId("6292e276938a9b2f9f9f2e58"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10826",
        "vaccine" : "AZ",
        "date" : ISODate("2021-07-04T00:00:00Z")
},
{
        "_id" : ObjectId("6292e276938a9b2f9f9f2e62"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10831",
        "vaccine" : "AZ",
        "date" : ISODate("2021-10-10T00:00:00Z")
},
{
        "_id" : ObjectId("6292e276938a9b2f9f9f2e70"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10838",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-05-04T00:00:00Z")
},
{
        "_id" : ObjectId("6292e276938a9b2f9f9f2e82"),
        "nurse" : "Enfermero 111",
        "patient" : "Paciente 10847",
        "vaccine" : "AZ",
```

```
            "date" : ISODate("2021-05-17T00:00:00Z")
        },
        {
            "_id" : ObjectId("6292e276938a9b2f9f9f2e88"),
            "nurse" : "Enfermero 111",
            "patient" : "Paciente 10850",
            "vaccine" : "Sputnik V",
            "date" : ISODate("2021-11-04T00:00:00Z")
        },
        {
            "_id" : ObjectId("6292e276938a9b2f9f9f2e8a"),
            "nurse" : "Enfermero 111",
            "patient" : "Paciente 10851",
            "vaccine" : "Moderna",
            "date" : ISODate("2021-12-02T00:00:00Z")
        },
        {
            "_id" : ObjectId("6292e276938a9b2f9f9f2e90"),
            "nurse" : "Enfermero 111",
            "patient" : "Paciente 10854",
            "vaccine" : "Pfizer",
            "date" : ISODate("2021-06-20T00:00:00Z")
        },
        {
            "_id" : ObjectId("6292e276938a9b2f9f9f2e96"),
            "nurse" : "Enfermero 111",
            "patient" : "Paciente 10857",
            "vaccine" : "Pfizer",
            "date" : ISODate("2021-11-20T00:00:00Z")
        },
        {
            "_id" : ObjectId("6292e276938a9b2f9f9f2eac"),
            "nurse" : "Enfermero 111",
            "patient" : "Paciente 10868",
            "vaccine" : "Johnson",
            "date" : ISODate("2021-12-21T00:00:00Z")
        },
        {
            "_id" : ObjectId("6292e276938a9b2f9f9f2eae"),
            "nurse" : "Enfermero 111",
            "patient" : "Paciente 10869",
            "vaccine" : "Sputnik V",
            "date" : ISODate("2021-06-07T00:00:00Z")
        },
        {
            "_id" : ObjectId("6292e276938a9b2f9f9f2eb6"),
            "nurse" : "Enfermero 111",
```

```
                "patient" : "Paciente 10873",
                "vaccine" : "AZ",
                "date" : ISODate("2021-12-01T00:00:00Z")
            },
            {
                "_id" : ObjectId("6292e276938a9b2f9f9f2eba"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10875",
                "vaccine" : "AZ",
                "date" : ISODate("2021-09-18T00:00:00Z")
            },
            {
                "_id" : ObjectId("6292e276938a9b2f9f9f2ebc"),
                "nurse" : "Enfermero 111",
                "patient" : "Paciente 10876",
                "vaccine" : "Pfizer",
                "date" : ISODate("2021-07-14T00:00:00Z")
            }
        ]
}

{
        "_id" : ObjectId("6292e272938a9b2f9f9ed64a"),
        "name" : "Enfermero 1110",
        "experience" : 10,
        "tags" : [
            "Moderna",
            "Sputnik V"
        ],
        "doses" : [
            {
                "_id" : ObjectId("6292e298938a9b2f9fa23b1a"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110763",
                "vaccine" : "Pfizer",
                "date" : ISODate("2021-11-23T00:00:00Z")
            },
            {
                "_id" : ObjectId("6292e298938a9b2f9fa23b1c"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110764",
                "vaccine" : "Johnson",
                "date" : ISODate("2021-06-05T00:00:00Z")
            },
            {
                "_id" : ObjectId("6292e298938a9b2f9fa23b22"),
                "nurse" : "Enfermero 1110",
```

```
        "patient" : "Paciente 110767",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-09-10T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b2e"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110773",
        "vaccine" : "Sputnik V",
        "date" : ISODate("2021-07-06T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b44"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110784",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-11-08T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b48"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110786",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-09-07T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b50"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110790",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-10-14T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b54"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110792",
        "vaccine" : "AZ",
        "date" : ISODate("2021-07-02T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b58"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110794",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-05-10T00:00:00Z")
    },
    {
```

```
        "_id" : ObjectId("6292e298938a9b2f9fa23b5a"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110795",
        "vaccine" : "Sputnik V",
        "date" : ISODate("2021-06-01T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b6c"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110804",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-09-10T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b72"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110807",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-10-27T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b90"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110822",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-10-25T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b9a"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110827",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-12-11T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23b9c"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110828",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-12-27T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23ba4"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110832",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-11-27T00:00:00Z")
```

```
        },
        {
                "_id" : ObjectId("6292e298938a9b2f9fa23baa"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110835",
                "vaccine" : "Sputnik V",
                "date" : ISODate("2021-07-11T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e298938a9b2f9fa23bac"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110836",
                "vaccine" : "Johnson",
                "date" : ISODate("2021-09-28T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e298938a9b2f9fa23bb6"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110841",
                "vaccine" : "Pfizer",
                "date" : ISODate("2021-09-08T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e298938a9b2f9fa23bb8"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110842",
                "vaccine" : "Moderna",
                "date" : ISODate("2021-08-13T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e298938a9b2f9fa23bc6"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110849",
                "vaccine" : "Sputnik V",
                "date" : ISODate("2021-06-21T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e298938a9b2f9fa23bca"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110851",
                "vaccine" : "AZ",
                "date" : ISODate("2021-09-27T00:00:00Z")
        },
        {
                "_id" : ObjectId("6292e298938a9b2f9fa23bcc"),
                "nurse" : "Enfermero 1110",
                "patient" : "Paciente 110852",
```

```
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-09-19T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23bce"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110853",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-05-02T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23bd2"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110855",
        "vaccine" : "AZ",
        "date" : ISODate("2021-09-14T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23bdc"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110860",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-07-02T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23be0"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110862",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-05-28T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23be4"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110864",
        "vaccine" : "Sputnik V",
        "date" : ISODate("2021-05-13T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23be6"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110865",
        "vaccine" : "AZ",
        "date" : ISODate("2021-09-25T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23bec"),
```

```
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110868",
        "vaccine" : "Sputnik V",
        "date" : ISODate("2021-08-10T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23c00"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110878",
        "vaccine" : "Pfizer",
        "date" : ISODate("2021-11-03T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23c0a"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110883",
        "vaccine" : "Sputnik V",
        "date" : ISODate("2021-07-20T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23c10"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110886",
        "vaccine" : "Moderna",
        "date" : ISODate("2021-10-20T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23c1a"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110891",
        "vaccine" : "Sputnik V",
        "date" : ISODate("2021-10-18T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23c1c"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110892",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-12-11T00:00:00Z")
    },
    {
        "_id" : ObjectId("6292e298938a9b2f9fa23c22"),
        "nurse" : "Enfermero 1110",
        "patient" : "Paciente 110895",
        "vaccine" : "Johnson",
        "date" : ISODate("2021-10-17T00:00:00Z")
    },
```

```
            {
                  "_id" : ObjectId("6292e298938a9b2f9fa23c24"),
                  "nurse" : "Enfermero 1110",
                  "patient" : "Paciente 110896",
                  "vaccine" : "Moderna",
                  "date" : ISODate("2021-10-18T00:00:00Z")
            },
            {
                  "_id" : ObjectId("6292e298938a9b2f9fa23c26"),
                  "nurse" : "Enfermero 1110",
                  "patient" : "Paciente 110897",
                  "vaccine" : "Pfizer",
                  "date" : ISODate("2021-08-17T00:00:00Z")
            },
            {
                  "_id" : ObjectId("6292e298938a9b2f9fa23c2e"),
                  "nurse" : "Enfermero 1110",
                  "patient" : "Paciente 110901",
                  "vaccine" : "Johnson",
                  "date" : ISODate("2021-12-11T00:00:00Z")
            },
            {
                  "_id" : ObjectId("6292e298938a9b2f9fa23c32"),
                  "nurse" : "Enfermero 1110",
                  "patient" : "Paciente 110903",
                  "vaccine" : "AZ",
                  "date" : ISODate("2021-10-02T00:00:00Z")
            },
            {
                  "_id" : ObjectId("6292e298938a9b2f9fa23c38"),
                  "nurse" : "Enfermero 1110",
                  "patient" : "Paciente 110906",
                  "vaccine" : "AZ",
                  "date" : ISODate("2021-12-01T00:00:00Z")
            },
            {
                  "_id" : ObjectId("6292e298938a9b2f9fa23c3c"),
                  "nurse" : "Enfermero 1110",
                  "patient" : "Paciente 110908",
                  "vaccine" : "Moderna",
                  "date" : ISODate("2021-07-21T00:00:00Z")
            }
      ]
}
```