Conceptos y comandos básicos de la replicación en bases de datos NoSQL

&

Pruebas e informe de replicación en Bases de Datos NoSQL

Asignatura

Bases de Datos Avanzadas

Presenta

Michael Collazos

Andes Padilla

Santiago Lopez

Docente

William Ruiz

Bogotá D.C – Colombia

2024

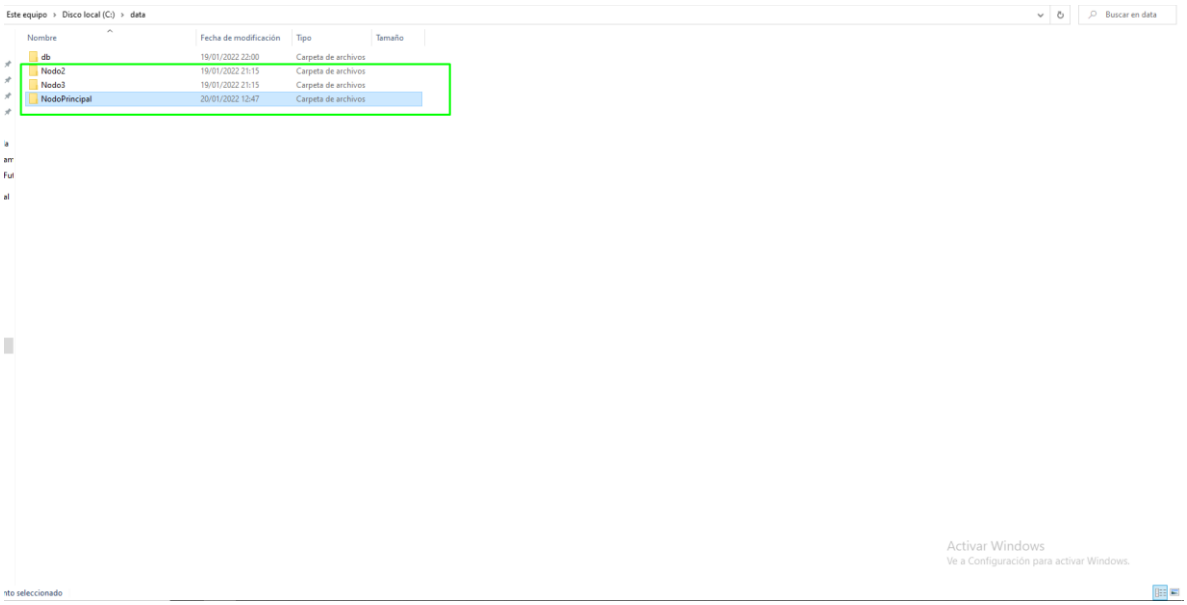
Link video: https://www.youtube.com/watch?v=NlEsrZEvhmY

GitHub: https://github.com/felipehurtado15/bases_de-datos_mongo

Requerimientos No Funcionales

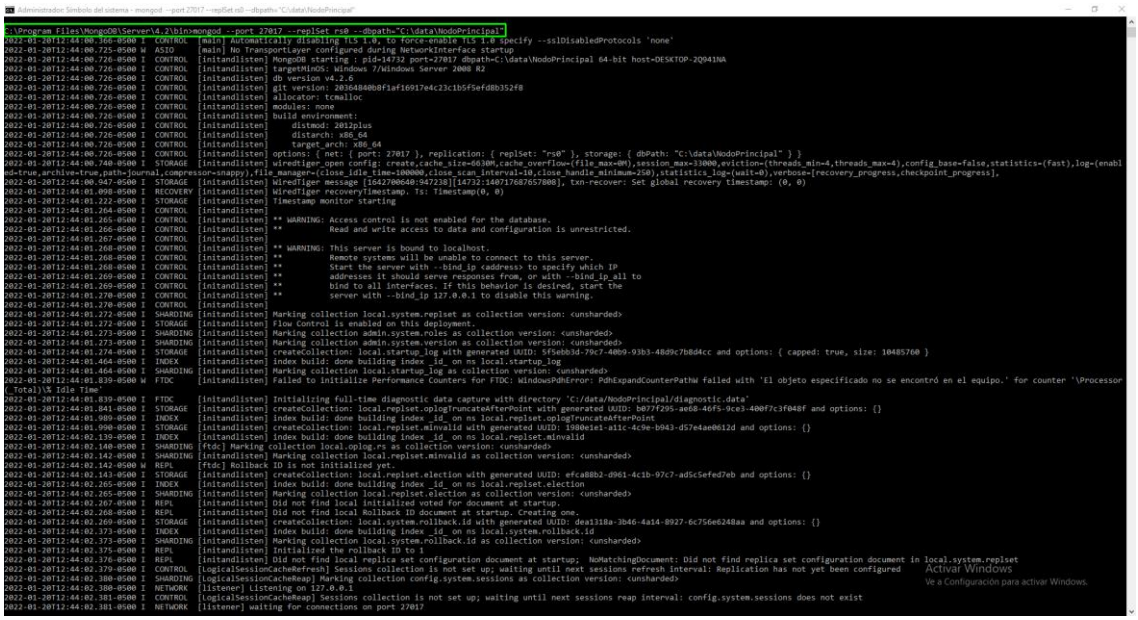
- ✓ El sistema de replicación debe poseer como mínimo 3 nodos.
- ✓ Cada uno de los nodos deben estar configurados sobre una misma replica.
- ✓ Cada uno de los nodos dispondrán de un mismo hostname, pero ocuparán diferentes puertos de conexión.
- ✓ La distribución de los nodos será maestro y esclavos, siendo así que un nodo tendrá el rol maestro, siendo este el principal y los otros nodos serán esclavos, siendo estos secundarios.
- ✓ Cada uno de los nodos deben tener acceso a la base de datos " TorneoDeportivoFutsal", junto con sus colecciones y documentos.
- ✓ Cada uno de los nodos dispondrá de carpetas de destino para el almacenamiento de los datos y la replicación de estos, y serán asignadas correspondientes a cada uno de los nodos.
- ✓ Cada uno de los nodos ocuparan una disponibilidad, para cuando el nodo maestro falle, logren reemplazarlo y sean asignados como nodos primarios.

- Se han construido tres diferentes carpetas, que serán ocupadas cada una para cada uno de los nodos, para el almacenamiento de la información correspondiente a las bases de datos.

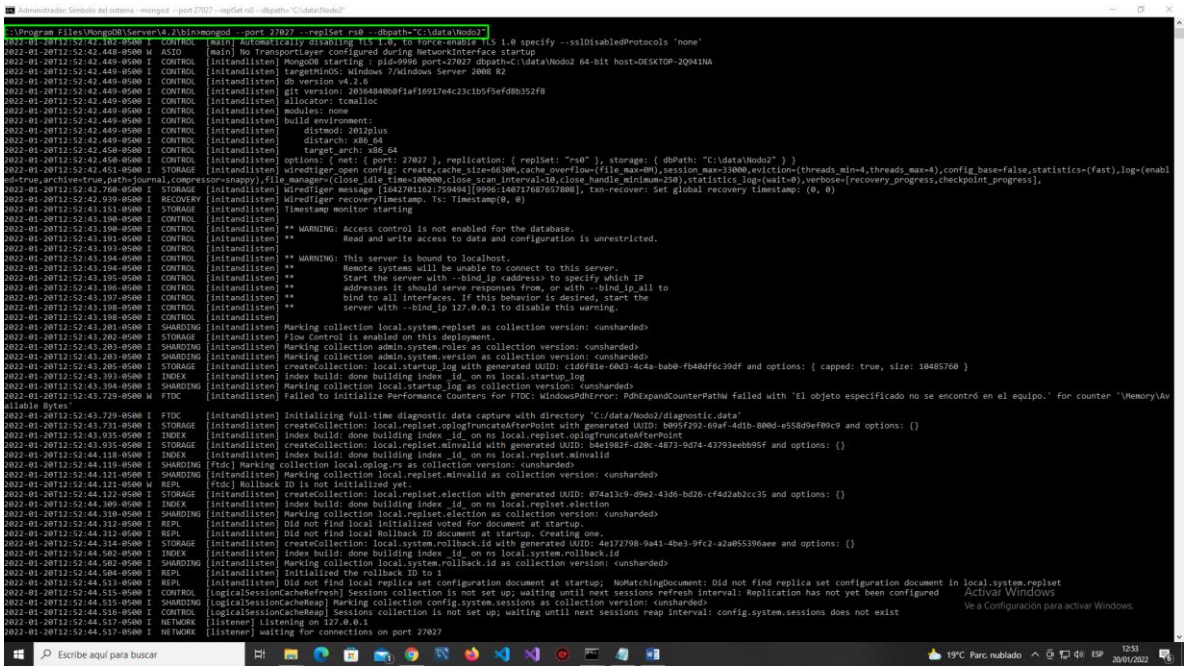


- Construimos el nodo maestro y los dos nodos esclavos, sobre la misma replica definida en el nodo maestro, y cada nodo será construido con diferentes puertos.

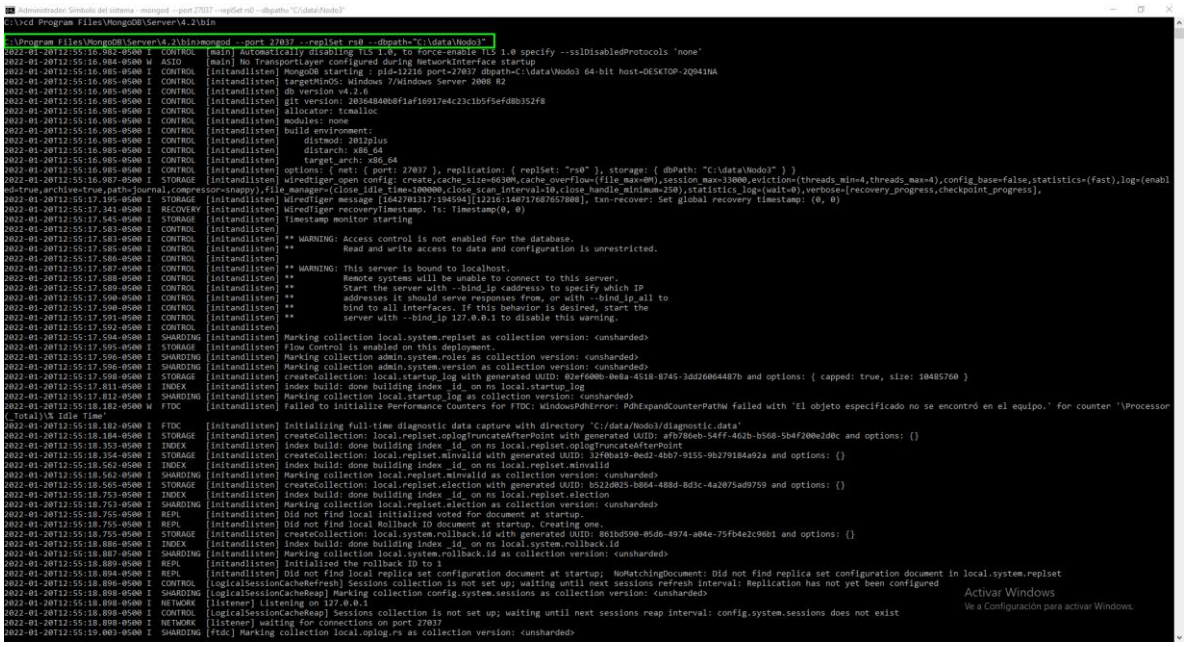
Nodo Maestro



Nodo Esclavo 2



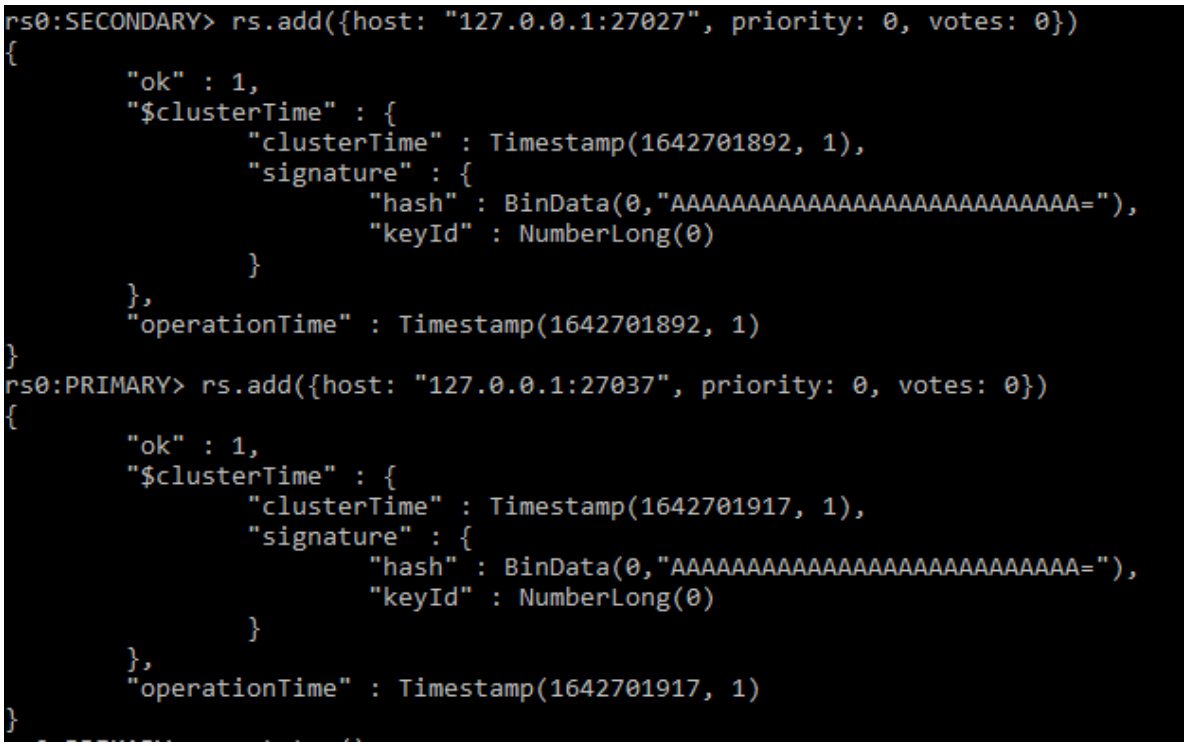
Nodo Esclavo 3



- Ahora comenzaremos inicializando la réplica construida sobre el nodo maestro, de tal manera que:

```
> rs.initiate()
{
        "info2" : "no configuration specified. Using a default configuration for the set",
        "me" : "localhost:27017",
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1642701767, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1642701767, 1)
}
```

- Aquí debemos añadir y configurar los nodos esclavos, dejándole así la prioridad mayor al nodo maestro.



- Gracias a este comando podremos conocer el estado de cada uno de los

nodos creados.

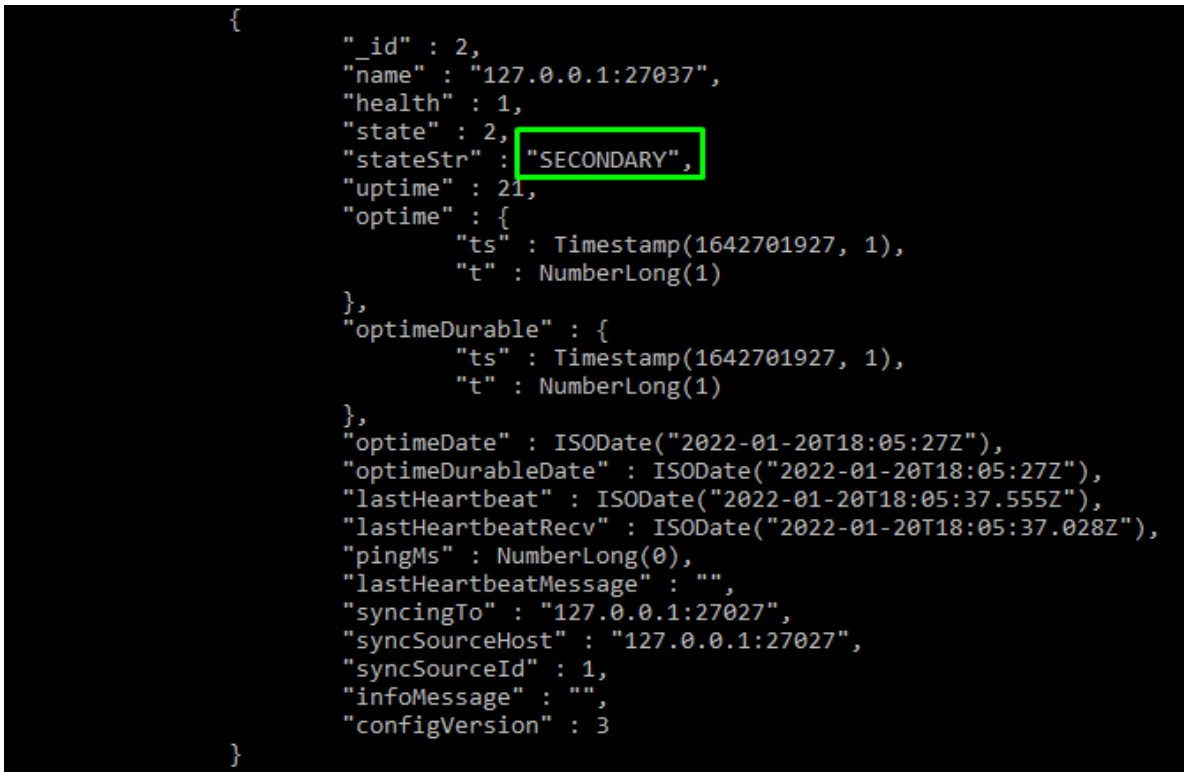
Nodo Maestro

```
"members" : [
        {
                "_id" : 0,
                "name" : "localhost:27017",
                "health" : 1,
                "state" : 1,
                "stateStr" : "PRIMARY",
                "uptime" : 1298,
                "optime" : {
                        "ts" : Timestamp(1642701937, 1),
                        "t" : NumberLong(1)
                },
                "optimeDate" : ISODate("2022-01-20T18:05:37Z"),
                "syncingTo" : "",
                "syncSourceHost" : "",
                "syncSourceId" : -1,
                "infoMessage" : "",
                "electionTime" : Timestamp(1642701767, 2),
                "electionDate" : ISODate("2022-01-20T18:02:47Z"),
                "configVersion" : 3,
                "self" : true,
                "lastHeartbeatMessage" : ""
        },
```


Nodo Esclavo 2


```
        {
                "_id" : 1,
                "name" : "127.0.0.1:27027",
                "health" : 1,
                "state" : 2,
                "stateStr" : "SECONDARY",
                "uptime" : 46,
                "optime" : {
                        "ts" : Timestamp(1642701927, 1),
                        "t" : NumberLong(1)
                },
                "optimeDurable" : {
                        "ts" : Timestamp(1642701927, 1),
                        "t" : NumberLong(1)
                },
                "optimeDate" : ISODate("2022-01-20T18:05:27Z"),
                "optimeDurableDate" : ISODate("2022-01-20T18:05:27Z"),
                "lastHeartbeat" : ISODate("2022-01-20T18:05:37.552Z"),
                "lastHeartbeatRecv" : ISODate("2022-01-20T18:05:37.556Z"),
                "pingMs" : NumberLong(0),
                "lastHeartbeatMessage" : "",
                "syncingTo" : "localhost:27017",
                "syncSourceHost" : "localhost:27017",
                "syncSourceId" : 0,
                "infoMessage" : "",
                "configVersion" : 3
        },
```


Nodo Esclavo 3



- Gracias al siguiente comando, lograremos conocer la configuración que corresponde a cada uno de los nodos.

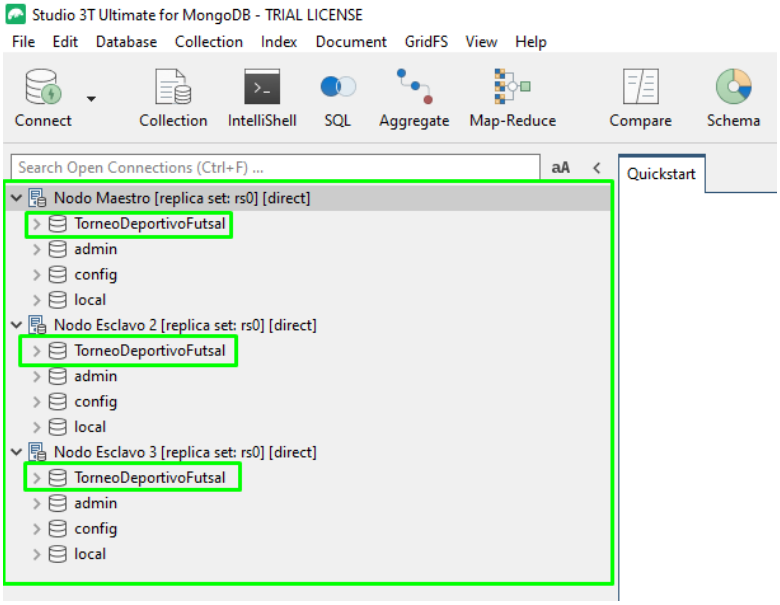
```
rs0:PRIMARY> rs.conf()
{
        "_id" : "rs0",
        "version" : 3,
        "protocolVersion" : NumberLong(1),
        "writeConcernMajorityJournalDefault" : true,
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27017",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 1,
                        "tags" : {

                        },
                        "slaveDelay" : NumberLong(0),
                        "votes" : 1
                },
                {
                        "_id" : 1,
                        "host" : "127.0.0.1:27027",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 0,
                        "tags" : {

                        },
                        "slaveDelay" : NumberLong(0),
                        "votes" : 0
                },
                {
                        "_id" : 2,
                        "host" : "127.0.0.1:27037",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 0,
                        "tags" : {

                        },
                        "slaveDelay" : NumberLong(0),
                        "votes" : 0
                }
        ],
        "settings" : {
                "chainingAllowed" : true,
                "heartbeatIntervalMillis" : 2000,
                "heartbeatTimeoutSecs" : 10,
                "electionTimeoutMillis" : 10000,
                "catchUpTimeoutMillis" : -1,
                "catchUpTakeoverDelayMillis" : 30000,
                "getLastErrorModes" : {

                },
                "getLastErrorDefaults" : {
                        "w" : 1,
                        "wtimeout" : 0
                },
                "replicaSetId" : ObjectId("61e9a3c67ffe1c9cc878b971")
```

- Ahora será necesario definir los nodos que no son maestro como esclavos, para que estos logren tener acceso a las bases de datos del nodo maestro.

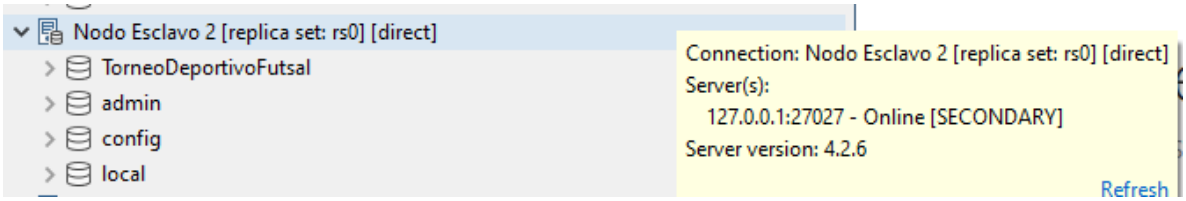
```
C:\Program Files\MongoDB\Server\4.2\bin>mongo --port 27027
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:27027/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("77a72d02-22e7-4a01-bb50-6493845843da") }
MongoDB server version: 4.2.6
Server has startup warnings:
2022-01-20T12:52:43.190-0500 I  CONTROL  [initandlisten]
2022-01-20T12:52:43.190-0500 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-01-20T12:52:43.191-0500 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-01-20T12:52:43.193-0500 I  CONTROL  [initandlisten]
2022-01-20T12:52:43.194-0500 I  CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2022-01-20T12:52:43.194-0500 I  CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2022-01-20T12:52:43.195-0500 I  CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2022-01-20T12:52:43.196-0500 I  CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2022-01-20T12:52:43.197-0500 I  CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2022-01-20T12:52:43.198-0500 I  CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2022-01-20T12:52:43.198-0500 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

rs0:SECONDARY> show dbs
2022-01-20T13:34:57.546-0500 E  QUERY    [js] uncaught exception: Error: listDatabases failed:{
        "operationTime" : Timestamp(1642703687, 1),
        "ok" : 0,
        "errmsg" : "not master and slaveOk=false",
        "code" : 13435,
        "codeName" : "NotMasterNoSlaveOk",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1642703687, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:135:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:87:12
shellHelper.show@src/mongo/shell/utils.js:906:13
shellHelper@src/mongo/shell/utils.js:790:15
@(shellhelp2):1:1
rs0:SECONDARY> rs.slaveok()
2022-01-20T13:35:31.502-0500 E  QUERY    [js] uncaught exception: TypeError: rs.slaveok is not a function :
@(shell):1:1
rs0:SECONDARY> rs.slaveOk()
rs0:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
rs0:SECONDARY>
```

```
C:\Program Files\MongoDB\Server\4.2\bin>mongo --port 27037
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:27037/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("38520718-7a2e-4f3f-80ad-20001997733b") }
MongoDB server version: 4.2.6
Server has startup warnings:
2022-01-20T12:55:17.583-0500 I  CONTROL  [initandlisten]
2022-01-20T12:55:17.583-0500 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-01-20T12:55:17.585-0500 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-01-20T12:55:17.586-0500 I  CONTROL  [initandlisten]
2022-01-20T12:55:17.587-0500 I  CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2022-01-20T12:55:17.588-0500 I  CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2022-01-20T12:55:17.589-0500 I  CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2022-01-20T12:55:17.590-0500 I  CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2022-01-20T12:55:17.590-0500 I  CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2022-01-20T12:55:17.591-0500 I  CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2022-01-20T12:55:17.592-0500 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

rs0:SECONDARY> show dbs
2022-01-20T13:36:33.798-0500 E  QUERY    [js] uncaught exception: Error: listDatabases failed:{
        "operationTime" : Timestamp(1642703787, 1),
        "ok" : 0,
        "errmsg" : "not master and slaveOk=false",
        "code" : 13435,
        "codeName" : "NotMasterNoSlaveOk",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1642703787, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:135:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:87:12
shellHelper.show@src/mongo/shell/utils.js:906:13
shellHelper@src/mongo/shell/utils.js:790:15
@(shellhelp2):1:1
rs0:SECONDARY> rs.slaveOk()
rs0:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
rs0:SECONDARY>
```

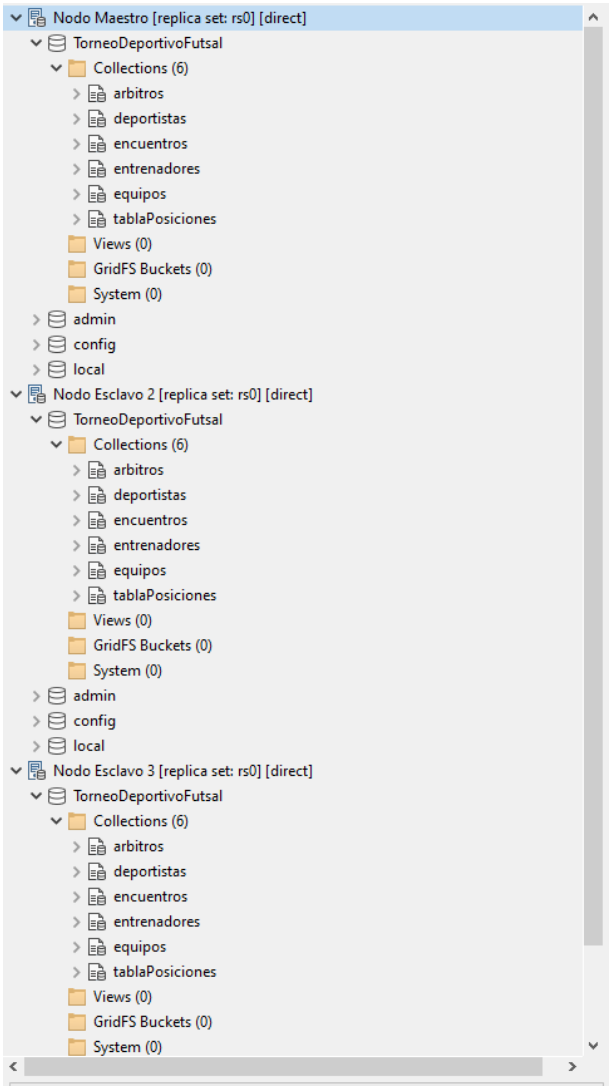
Muestra grafica de replicación y base de datos del torneo deportivo



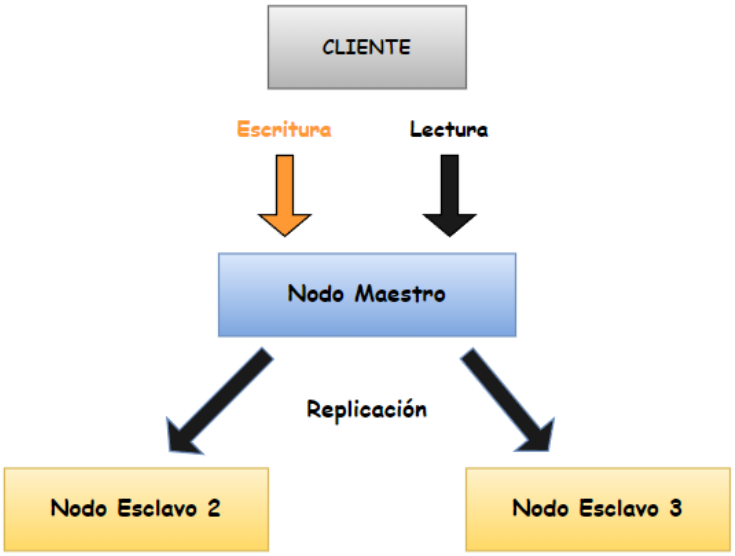

Nodo Maestro [replica set: rs0] [direct]
TorneoDeportivoFutsal
admin
config
local
Connection: Nodo Maestro [replica set: rs0] [direct]
Server(s):
localhost:27017 - Online [PRIMARY]
Server version: 4.2.6


Nodo Esclavo 2 [replica set: rs0] [direct]
TorneoDeportivoFutsal
admin
config
local
Connection: Nodo Esclavo 2 [replica set: rs0] [direct]
Server(s):
127.0.0.1:27027 - Online [SECONDARY]
Server version: 4.2.6
Refresh


Nodo Esclavo 3 [replica set: rs0] [direct]
TorneoDeportivoFutsal
admin
config
local
Connection: Nodo Esclavo 3 [replica set: rs0] [direct]
Server(s):
127.0.0.1:27037 - Online [SECONDARY]
Server version: 4.2.6
Refresh


Nodo Maestro [replica set: rs0] [direct]
TorneoDeportivoFutsal
Collections (6)
arbitros
deportistas
encuentros
entrenadores
equipos
tablaPosiciones
Views (0)
GridFS Buckets (0)
System (0)
admin
config
local
Nodo Esclavo 2 [replica set: rs0] [direct]
TorneoDeportivoFutsal
Collections (6)
arbitros
deportistas
encuentros
entrenadores
equipos
tablaPosiciones
Views (0)
GridFS Buckets (0)
System (0)
admin
config
local
Nodo Esclavo 3 [replica set: rs0] [direct]
TorneoDeportivoFutsal
Collections (6)
arbitros
deportistas
encuentros
entrenadores
equipos
tablaPosiciones
Views (0)
GridFS Buckets (0)
System (0)

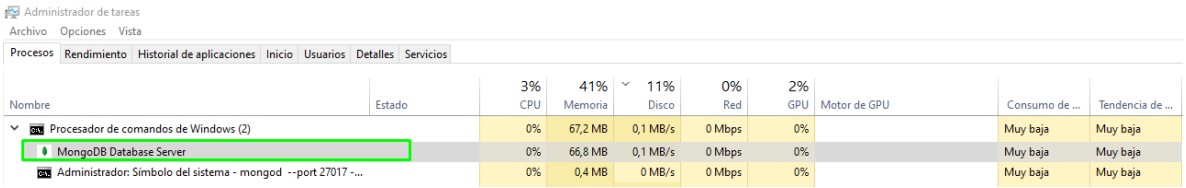
- **Diagrama de la estrategia de Maestro – Esclavo**



- Nuestra estrategia maestro-esclavo está caracterizada principalmente en poseer 3 nodos, donde uno de ellos será el nodo primario y el resto serán secundarios, nuestro nodo principal será el encargado de la escritura y lectura de los datos, para luego lograr la replicación sobre los nodos secundarios, tendiendo así la sincronización de los cambios reflejados sobre los datos. Como también cabe mencionar que nuestros nodos secundarios estarán dispuestos a ser el reemplazo del nodo principal, dado el caso que este se desconecte o falle.

Casos de prueba

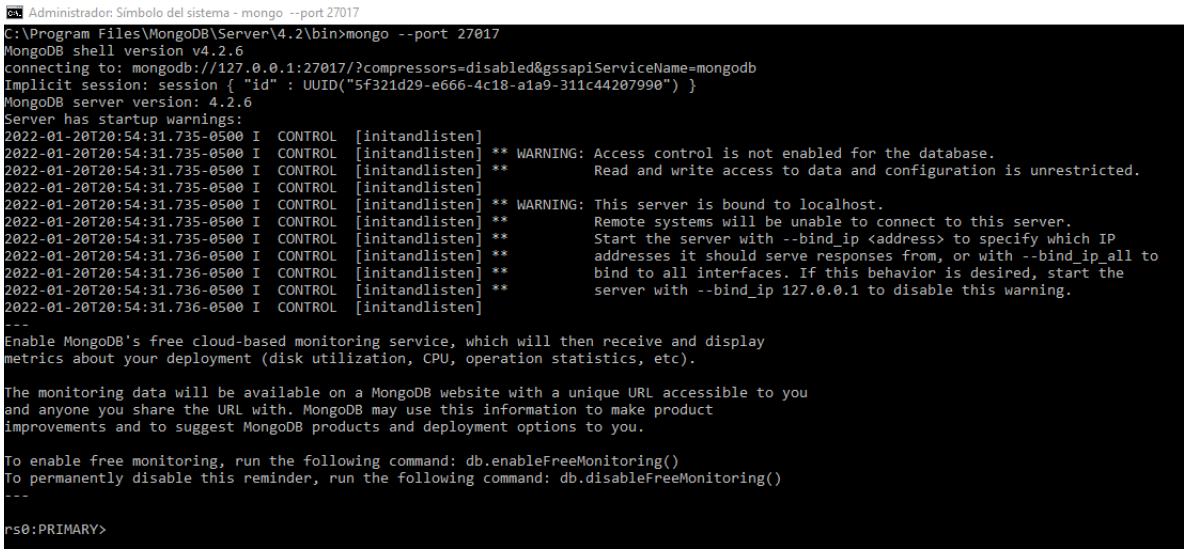
- ✓ En primer lugar, se validará la disponibilidad del orden de los nodos, y en una situación donde el nodo maestro caiga, se desconecte o falle, para que luego el nodo esclavo 2 reciba las responsabilidades del nodo maestro.



Finalizamos el servidor de mongo que está ejecutando nuestro nodo maestro, esto simulando una caída, desconexión o fallo de este.

Para luego poder visualizar que uno de los nodos esclavos, ahora hace el papel del maestro, motivo de la caída del nodo principal.

Y cuando se vuelve a reestablecer la conexión del servidor, nuevamente se asignarán los roles que han sido declarados desde un principio, de tal forma que:

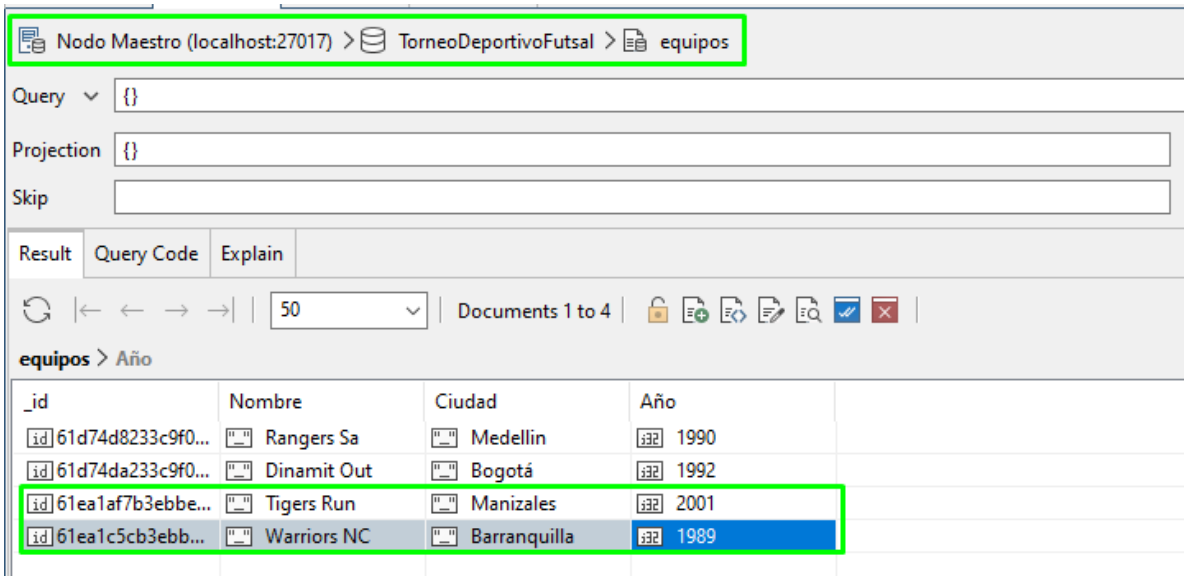


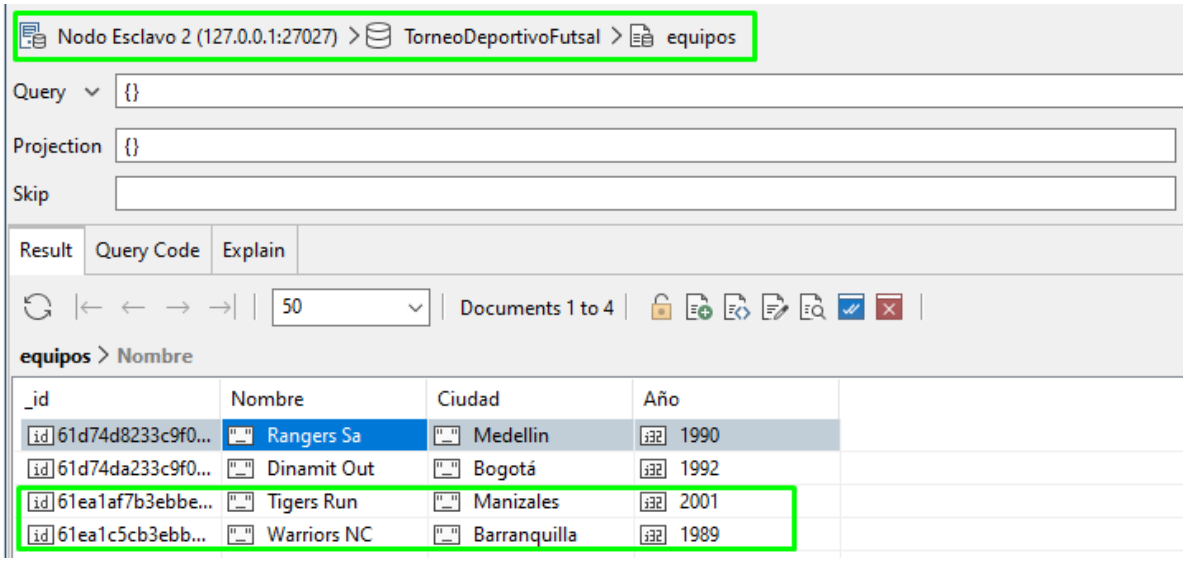
```
},
"members" : [
        {
                "_id" : 0,
                "name" : "localhost:27017",
                "health" : 1,
                "state" : 1,
                "stateStr" : "PRIMARY",
                "uptime" : 72,
                "optime" : {
                        "ts" : Timestamp(1642730142, 1),
                        "t" : NumberLong(5)
                },
                "optimeDate" : ISODate("2022-01-21T01:55:42Z"),
                "syncingTo" : "",
                "syncSourceHost" : "",
                "syncSourceId" : -1,
                "infoMessage" : "",
                "electionTime" : Timestamp(1642730082, 1),
                "electionDate" : ISODate("2022-01-21T01:54:42Z"),
                "configVersion" : 8,
                "self" : true,
                "lastHeartbeatMessage" : ""
        },
        {
                "_id" : 1,
                "name" : "127.0.0.1:27027",
                "health" : 1,
                "state" : 2,
                "stateStr" : "SECONDARY",
                "uptime" : 70,
                "optime" : {
                        "ts" : Timestamp(1642730132, 1),
                        "t" : NumberLong(5)
                },
                "optimeDurable" : {
                        "ts" : Timestamp(1642730132, 1),
                        "t" : NumberLong(5)
                },
                "optimeDate" : ISODate("2022-01-21T01:55:32Z"),
                "optimeDurableDate" : ISODate("2022-01-21T01:55:32Z"),
                "lastHeartbeat" : ISODate("2022-01-21T01:55:42.571Z"),
                "lastHeartbeatRecv" : ISODate("2022-01-21T01:55:42.423Z"),
                "pingMs" : NumberLong(0),
                "lastHeartbeatMessage" : "",
                "syncingTo" : "127.0.0.1:27037",
                "syncSourceHost" : "127.0.0.1:27037",
                "syncSourceId" : 2,
                "infoMessage" : "",
                "configVersion" : 8
        },
        {
                "_id" : 2,
                "name" : "127.0.0.1:27037",
                "health" : 1,
                "state" : 2,
                "stateStr" : "SECONDARY",
                "uptime" : 70,
                "optime" : {
                        "ts" : Timestamp(1642730132, 1),
                        "t" : NumberLong(5)
```

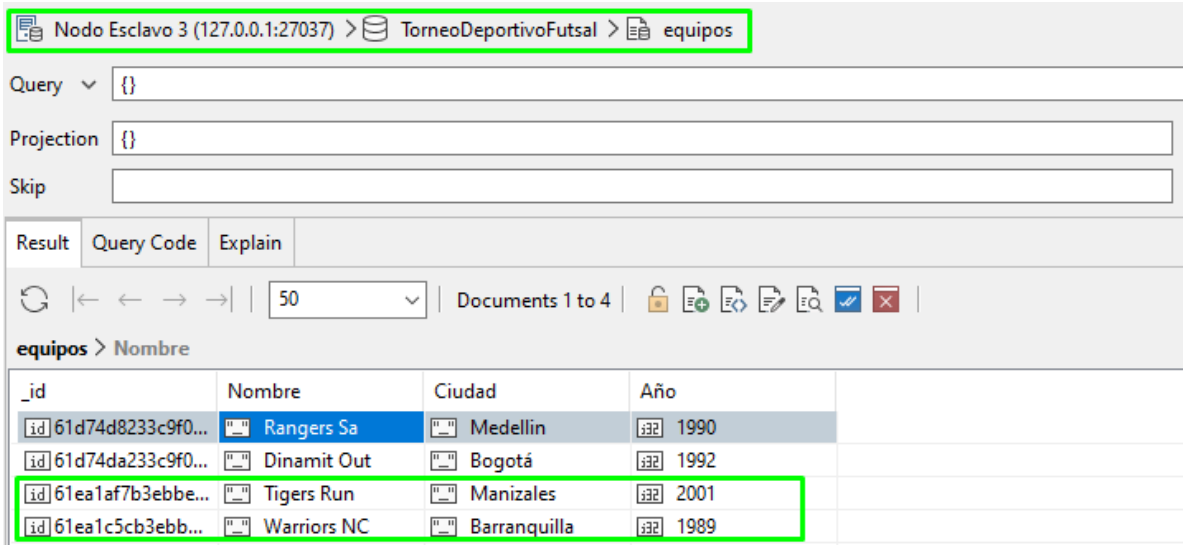

- ✓ En segundo lugar, se validará la redundancia de los datos y que, al momento de realizar registros sobre el nodo maestro, luego estos sean sincronizados con los nodos esclavos sin poseer la duplicidad de la información.

```
Administrador: Símbolo del sistema - mongo  --port 27017
rs0:PRIMARY> db
test
rs0:PRIMARY> use TorneoDeportivoFutsal
switched to db TorneoDeportivoFutsal
rs0:PRIMARY> show collections
arbitros
deportistas
encuentros
entrenadores
equipos
tablaPosiciones
rs0:PRIMARY> db.equipos.insert({"Nombre":"Tigers Run", "Ciudad":"Manizales", "Año":2001}
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.equipos.insert({"Nombre":"Warriors NC", "Ciudad":"Barranquilla", "Año":1989})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY>
```

Sabemos que hemos insertado dos equipos nuevos para la colección de nuestra base de datos, y es así que como resultado se deben visualizar sobre el nodo maestro, y haberse sincronizado sobre sus nodos esclavos.




Nodo Esclavo 2 (127.0.0.1:27027) > TorneoDeportivoFutsal > equipos
Query {}
Projection {}
Skip
Result
Query Code
Explain
50
Documents 1 to 4
equipos > Nombre
_id
Nombre
Ciudad
Año
61d74d8233c9f0...
Rangers Sa
Medellin
1990
61d74da233c9f0...
Dinamit Out
Bogotá
1992
61ea1af7b3ebbe...
Tigers Run
Manizales
2001
61ea1c5cb3ebb...
Warriors NC
Barranquilla
1989


Nodo Esclavo 3 (127.0.0.1:27037) > TorneoDeportivoFutsal > equipos
Query {}
Projection {}
Skip
Result
Query Code
Explain
50
Documents 1 to 4
equipos > Nombre
_id
Nombre
Ciudad
Año
61d74d8233c9f0...
Rangers Sa
Medellin
1990
61d74da233c9f0...
Dinamit Out
Bogotá
1992
61ea1af7b3ebbe...
Tigers Run
Manizales
2001
61ea1c5cb3ebb...
Warriors NC
Barranquilla
1989

- ✓ En conclusión, hemos logrado validar el funcionamiento de nuestra estrategia “maestro-esclavo”, tanto por parte de la disponibilidad como la redundancia de la información, encontramos que dado el momento en el que nuestro nodo maestro falle, estarán los nodos esclavos dispuesto a elegir y decidir quién lo reemplaza para seguir la operación, como también al momento de usar el comando “insert”, desde nuestro nodo maestro, se evidencia la sincronización con los otros dos nodos y se realizan la cantidad de registros correspondiente evitando la duplicidad de los mismos.

Bibliografía

- Sarasa, A. (2016). Introducción a las bases de datos NoSQL usando MongoDB. Editorial UOC. (Capitulo 7- Replicación)
- https://www.youtube.com/watch?v=gSD594Cvgy8