Actividad 2 - Conceptos y comandos básicos de la replicación en bases de datos NoSQL

Realizado por:

Joe Alejandro Sierra Ocassal

Ingrid Johana Rojas Gómez

Corporación Universitaria Iberoamericana

Bases de datos avanzadas

WILLIAM RUIZ

Marzo 2023

IBEROAMERICANA
CORPORACIÓN UNIVERSITARIA
P.J. No. 0428 del 28 de Enero 1982 - MEN | VIGILADA MINEDUCACIÓN

Contenido

Requerimientos no funcionales

De acuerdo con nuestra anterior actividad de crear un diseño para una base de datos del evento deportivo, especificamos los criterios de calidad para evitar la redundancia y una disponibilidad de 24x7 al acceso de la información:

1. El Ing. de Arquitectura debe garantizar que el sistema tenga disponibilidad al acceso de los datos, aunque haya ocurrido una falla en el motor de la base de datos de cualquier índole.
2. En dado caso de que se caiga el nodo principal debe existir una replica para permitir el acceso a la información en un 100%.
3. El servidor secundario debe quedar como principal para seguir manejando la persistencia de los datos.
4. El sistema de replica debe contener configurado al menos 3 nodos.
5. Los nodos deben estar conectados a la base de datos en cuestión, para este caso, la del evento de deportes “TorneoDeportivo”.
6. Cada nodo debe de ejecutarse en un puerto distinto para asegurar la disponibilidad.

Video del proceso realizado

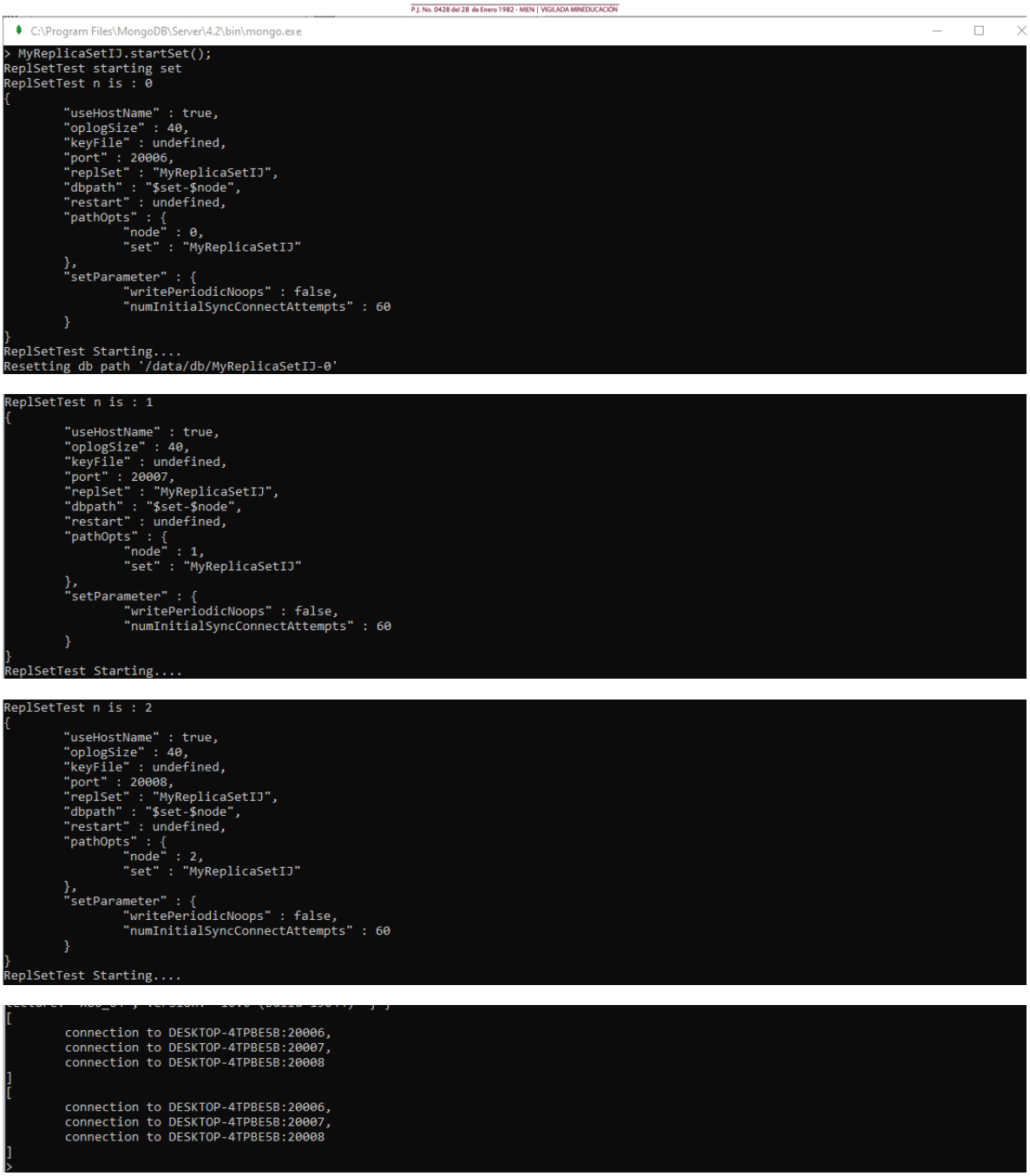
https://laiberocol-my.sharepoint.com/:v:/g/personal/jsierrao_ibero_edu_co/EXyNK4QVqwNCoWaVwLRK43YBbjKWvEHVTM4WpvJlxbJGhg?e=e8bvOd

Paso a paso comandos realizados

1. Crear replica

```
> MyReplicaSetIJ = new ReplSetTest({ name: "MyReplicaSetIJ", nodes: 3 }); print("Replica creada");
Starting new replica set MyReplicaSetIJ
Replica creada
>
```

2. Iniciar replica



3. Inicializar replica configurada en los 3 nodos

```
Reconfiguring replica set to add in other nodes
{
        "replSetReconfig" : {
                "_id" : "MyReplicaSetIJ",
                "protocolVersion" : 1,
                "members" : [
                        {
                                "_id" : 0,
                                "host" : "DESKTOP-4TPBE5B:20006"
                        },
                        {
                                "_id" : 1,
                                "host" : "DESKTOP-4TPBE5B:20007"
                        },
                        {
                                "_id" : 2,
                                "host" : "DESKTOP-4TPBE5B:20008"
                        }
                ],
                "version" : 2
        }
}
```

4. Crear una conexión al nodo principal

```
> conn = new Mongo("DESKTOP-4TPBE5B:20006");
d20006| 2023-03-28T19:41:41.428-0500 I  NETWORK  [listener] connection accepted from 192.168.56.1:58460 #26 (8 connections now open)
d20006| 2023-03-28T19:41:41.438-0500 I  NETWORK  [conn26] received client metadata from 192.168.56.1:58460 conn26: { application: { name:
 "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "4.2.24" }, os: { type: "Windows", name: "Microsoft Windows 10",
architecture: "x86_64", version: "10.0 (build 19044)" } }
connection to DESKTOP-4TPBE5B:20006
>
```

```
> testDB = conn.getDB("TorneoDeportivo");
TorneoDeportivo
```

```
> testDB.isMaster()
{
        "hosts" : [
                "DESKTOP-4TPBE5B:20006",
                "DESKTOP-4TPBE5B:20007",
                "DESKTOP-4TPBE5B:20008"
        ],
        "setName" : "MyReplicaSetIJ",
        "setVersion" : 2,
        "ismaster" : true,
        "secondary" : false,
        "primary" : "DESKTOP-4TPBE5B:20006",
        "me" : "DESKTOP-4TPBE5B:20006",
        "electionId" : ObjectId("7fffffff0000000000000001"),
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1680050420, 1),
                        "t" : NumberLong(1)
                },
                "lastWriteDate" : ISODate("2023-03-29T00:40:20Z"),
                "majorityOpTime" : {
                        "ts" : Timestamp(1680050420, 1),
                        "t" : NumberLong(1)
                },
                "majorityWriteDate" : ISODate("2023-03-29T00:40:20Z")
        },
        "maxBsonObjectSize" : 16777216,
        "maxMessageSizeBytes" : 48000000,
        "maxWriteBatchSize" : 100000,
        "localTime" : ISODate("2023-03-29T00:44:00.215Z"),
        "logicalSessionTimeoutMinutes" : 30,
        "connectionId" : 26,
        "minWireVersion" : 0,
        "maxWireVersion" : 8,
        "readOnly" : false,
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1680050420, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1680050420, 1)
}
```

5. Insertar un registro

```
> testDB.deportistas.insert({nombre:"juan", apellido:"gonzales", edad:"36"})
d20006| 2023-03-28T21:11:05.430-0500 I  STORAGE  [conn26] createCollection: TorneoDeportivo.deportistas with generated UUID: 301a556c-0407-4b46-8218-7b39abc4d99b and options: {}
d20006| 2023-03-28T21:11:05.441-0500 I  INDEX    [conn26] index build: done building index _id_ on ns TorneoDeportivo.deportistas
WriteResult({ "nInserted" : 1 })
> d20007| 2023-03-28T21:11:05.443-0500 I  STORAGE  [repl-writer-worker-1] createCollection: TorneoDeportivo.deportistas with provided UUID: 301a556c-0407-4b46-8218-7b39abc4d99b and options:
{ uuid: UUID("301a556c-0407-4b46-8218-7b39abc4d99b") }
d20008| 2023-03-28T21:11:05.443-0500 I  STORAGE  [repl-writer-worker-1] createCollection: TorneoDeportivo.deportistas with provided UUID: 301a556c-0407-4b46-8218-7b39abc4d99b and options: {
uuid: UUID("301a556c-0407-4b46-8218-7b39abc4d99b") }
d20007| 2023-03-28T21:11:05.452-0500 I  INDEX    [repl-writer-worker-1] index build: done building index _id_ on ns TorneoDeportivo.deportistas
d20008| 2023-03-28T21:11:05.454-0500 I  INDEX    [repl-writer-worker-1] index build: done building index _id_ on ns TorneoDeportivo.deportistas

>
```

6. Verificar registros

```
> testDB.deportistas.count();
1
```

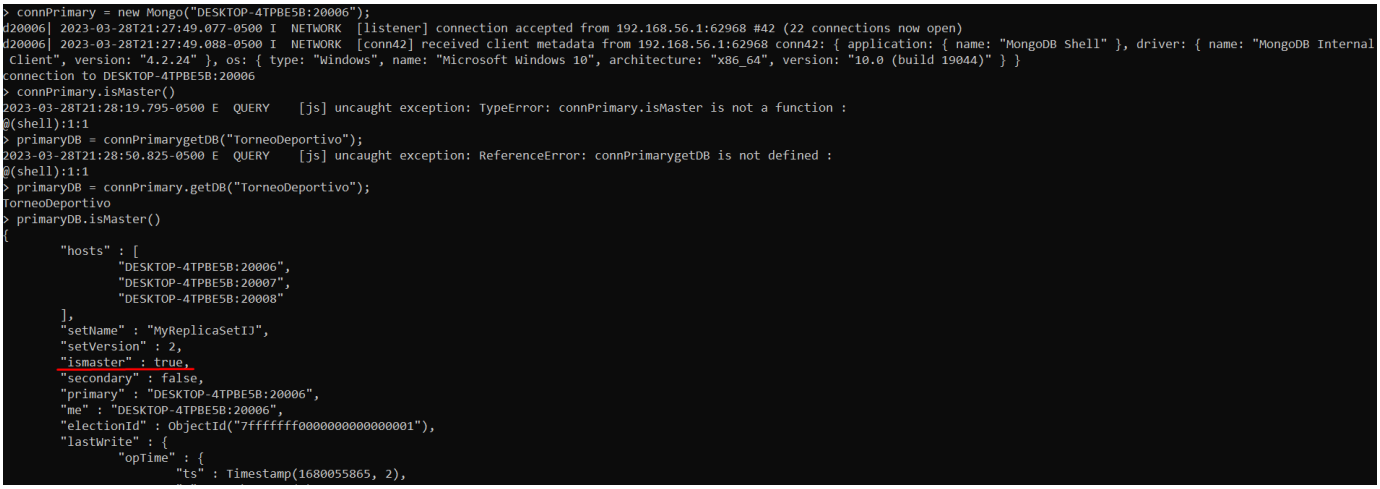
7. Nos conectamos al segundo nodo:

```
> connSecondary = new Mongo("DESKTOP-4TPBE5B:20007");
d20007| 2023-03-28T21:14:44.297-0500 I  NETWORK  [listener] connection accepted from 192.168.56.1:62534 #20 (6 connections now open)
d20007| 2023-03-28T21:14:44.297-0500 I  NETWORK  [conn20] received client metadata from 192.168.56.1:62534 conn20: { application: { name: "MongoDB Shell" }, driver: { name: "MongoDB Internal
 Client", version: "4.2.24" }, os: { type: "Windows", name: "Microsoft Windows 10", architecture: "x86_64", version: "10.0 (build 19044)" } }
connection to DESKTOP-4TPBE5B:20007
> secondaryTestDB = connSecondary.getDB("tecnicos");
tecnicos
```

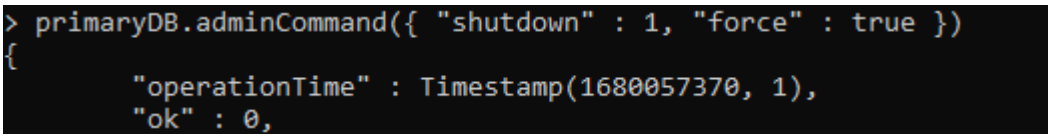
8. Verificamos si es MÁSTER:

```
> secondaryTestDB.isMaster()
{
        "hosts" : [
                "DESKTOP-4TPBE5B:20006",
                "DESKTOP-4TPBE5B:20007",
                "DESKTOP-4TPBE5B:20008"
        ],
        "setName" : "MyReplicaSetIJ",
        "setVersion" : 2,
        "ismaster" : false,
        "secondary" : true,
        "primary" : "DESKTOP-4TPBE5B:20006",
        "me" : "DESKTOP-4TPBE5B:20007",
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1680055865, 2),
                        "t" : NumberLong(1)
                },
                "lastWriteDate" : ISODate("2023-03-29T02:11:05Z"),
                "majorityOpTime" : {
                        "ts" : Timestamp(1680055865, 2),
                        "t" : NumberLong(1)
                },
                "majorityWriteDate" : ISODate("2023-03-29T02:11:05Z")
        },
        "maxBsonObjectSize" : 16777216,
        "maxMessageSizeBytes" : 48000000,
        "maxWriteBatchSize" : 100000,
        "localTime" : ISODate("2023-03-29T02:15:21.458Z"),
        "logicalSessionTimeoutMinutes" : 30,
        "connectionId" : 20,
        "minWireVersion" : 0,
        "maxWireVersion" : 8,
        "readOnly" : false,
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1680055865, 2),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
```

9. Verificamos nuevamente nodo principal:



10. Detenemos el nodo



Casos de prueba

Casos de pruebas en replicación bajo MongoDB

# de caso	Tipo de prueba	Objetivo
1	Replicación	Verificar que se hayan creado el nodo primario y los secundarios.
2	Disponibilidad	Ingresar 2 o más documentos em las colecciones propuestas en el documento de requerimientos en el nodo maestro y verificar que todas las instancias tienen una réplica de los registros insertados.
3	Tolerancia a fallos	Prueba de desconexión del nodo primario y promoción de algunos de los nodos secundarios a primario.
4	Disponibilidad	Verificar cuál de los nodos secundarios es ahora el nodo primario.

Resultado casos de prueba

Como resultado para los casos de pruebas vamos a relacionar cada uno como corresponde:

Caso 1 Replicación:

Verificamos la existencia de los nodos creados:

```
Reconfiguring replica set to add in other nodes
{
        "replSetReconfig" : {
                "_id" : "MyReplicaSetIJ",
                "protocolVersion" : 1,
                "members" : [
                        {
                                "_id" : 0,
                                "host" : "DESKTOP-4TPBE5B:20006"
                        },
                        {
                                "_id" : 1,
                                "host" : "DESKTOP-4TPBE5B:20007"
                        },
                        {
                                "_id" : 2,
                                "host" : "DESKTOP-4TPBE5B:20008"
                        }
                ],
                "version" : 2
        }
}
```

Conexión primer nodo:

```
> primaryConn = new Mongo("DESKTOP-4TPBE5B:20000");
d20000| 2023-03-28T21:58:17.850-0500 I  NETWORK  [listener] connection accepted from 192.168.56.1:64897 #39 (19 connecti
ons now open)
d20000| 2023-03-28T21:58:17.850-0500 I  NETWORK  [conn39] received client metadata from 192.168.56.1:64897 conn39: { app
lication: { name: "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "4.2.24" }, os: { type: "Window
s", name: "Microsoft Windows 10", architecture: "x86_64", version: "10.0 (build 19044)" } }
connection to DESKTOP-4TPBE5B:20000
```

Conexión segundo nodo:

```
> secondaryConn = new Mongo("DESKTOP-4TPBE5B:20001");
d20001| 2023-03-28T22:03:35.379-0500 I  NETWORK  [listener] connection accepted from 192.168.56.1:65057 #20 (6 connectio
ns now open)
d20001| 2023-03-28T22:03:35.379-0500 I  NETWORK  [conn20] received client metadata from 192.168.56.1:65057 conn20: { app
lication: { name: "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "4.2.24" }, os: { type: "Window
s", name: "Microsoft Windows 10", architecture: "x86_64", version: "10.0 (build 19044)" } }
connection to DESKTOP-4TPBE5B:20001
```

Conexión tercer nodo:

```
> thirdConn = new Mongo("DESKTOP-4TPBE5B:20002");
d20002| 2023-03-28T22:07:20.994-0500 I  NETWORK  [listener] connection accepted from 192.168.56.1:65188 #20 (6 connectio
ns now open)
d20002| 2023-03-28T22:07:20.994-0500 I  NETWORK  [conn20] received client metadata from 192.168.56.1:65188 conn20: { app
lication: { name: "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "4.2.24" }, os: { type: "Window
s", name: "Microsoft Windows 10", architecture: "x86_64", version: "10.0 (build 19044)" } }
connection to DESKTOP-4TPBE5B:20002
```

Caso 2 disponibilidad:

Se insertaron registros en el primer nodo:

```
> primaryDB = primaryConn.getDB("TorneoDeportivo")
TorneoDeportivo
> primaryDB.deportistas.insert({ nombre:"juan", apellido: "gonzales", edad: "26" })
```

y se solo visualizar en los siguientes:

Registro en primer nodo:

```
> primaryDB.deportistas.findOne()
{
        "_id" : ObjectId("6423b3b3fb308ae4113870ae"),
        "nombre" : "juan",
        "apellido" : "gonzales",
        "edad" : "26"
}
```

Registro en segundo nodo:

```
> secondaryDB.deportistas.findOne()
{
        "_id" : ObjectId("6423b3b3fb308ae4113870ae"),
        "nombre" : "juan",
        "apellido" : "gonzales",
        "edad" : "26"
}
```

Registro en tercer nodo:

```
> thirdDB.deportistas.findOne()
{
        "_id" : ObjectId("6423b3b3fb308ae4113870ae"),
        "nombre" : "juan",
        "apellido" : "gonzales",
        "edad" : "26"
}
```

Caso 3 y 4 tolerancia a fallos - disponibilidad:

Realizamos el apagado de el primer nodo y verificamos el estado del segundo:

```
> secondaryDB.isMaster()
{
        "hosts" : [
                "DESKTOP-4TPBE5B:20000",
                "DESKTOP-4TPBE5B:20001",
                "DESKTOP-4TPBE5B:20002"
        ],
        "setName" : "MyReplicaSetIJ",
        "setVersion" : 2,
        "ismaster" : true,
        "secondary" : false,
```

IBEROAMERICANA
CORPORACIÓN UNIVERSITARIA
P.J. No. 0428 del 28 de Enero 1982 - MEN | VIGILADA MINEDUCACIÓN

Repositorio GitHub

https://github.com/SrZombra/act1-base-datos-nosql.git

Bibliografía:

https://aulavirtual.ibero.edu.co/mod/resource/view.php?id=310119

Actividad practica unidad 2

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