

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

Julian David Triana Celis

Juan Diego Ruiz Gutiérrez

Jorge Rodríguez



Ingeniería De Software, Facultad De Ingeniería

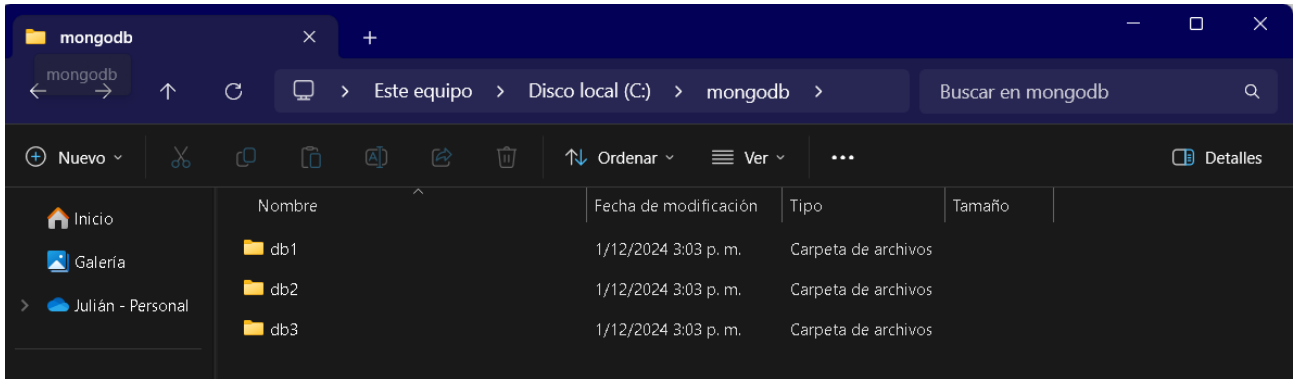
Universidad Iberoamericana

Bogotá D.C

2024

Resultados de la realización de los casos de prueba de replicación propuestos

1. Creación de las carpetas para el almacenamiento de las réplicas.



2. Abrimos 3 consolas cmd para cada una de las instancias, en este caso los nodos con puertos 27018, 27019 y 27020, donde cada puerto corresponde a cada una de las carpetas como se observa.

```
Símbolo del sistema - mongo x + v
Microsoft Windows [Versión 10.0.22631.4460]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\USUARIO>mongod --dbpath "C:\mongodb\db1" --port 27018 --replSet rs0
```

```
Símbolo del sistema - mongo x + v
Microsoft Windows [Versión 10.0.22631.4460]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\USUARIO>mongod --dbpath "C:\mongodb\db2" --port 27019 --replSet rs0
```

```
Símbolo del sistema - mongo x + v
Microsoft Windows [Versión 10.0.22631.4460]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\USUARIO>mongod --dbpath "C:\mongodb\db3" --port 27020 --replSet rs0
```

3. En una nueva consola cmd abrimos el mongo Shell en el puerto que tendremos como primario en este caso el 27018

```

Microsoft Windows [Versión 10.0.22631.4460]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO>mongosh --port 27018
Current Mongosh Log ID: 674cc01362d7a79d67893bf7
Connecting to:      mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.4
Using MongoDB:      8.0.3
Using Mongosh:      2.3.4

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2024-12-01T14:57:25.629-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-12-01T14:57:25.629-05:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
-----

```

4. Iniciamos la ejecución de la replica set en el nodo 1, el nodo primario, en el puerto

27018

```

test> rs.initiate()
{
  info2: 'no configuration specified. Using a default configuration for the set',
  me: 'localhost:27018',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1733083169, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1733083169, i: 1 })
}

```

5. Agregamos los 2 nodos que tendremos como secundarios, en los puertos 27019 y

27020

```

rs0 [direct: secondary] test> rs.add("localhost:27019")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1733083222, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1733083222, i: 1 })
}

```

```
rs0 [direct: primary] test> rs.add("localhost:27020")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1733083243, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1733083243, i: 1 })
}
```

6. Usamos el rs. Config para ver los 3 nodos creados y el entorno de replicación.

```
rs0 [direct: primary] test> rs.conf()
{
  _id: 'rs0',
  version: 5,
  term: 1,
  members: [
    {
      _id: 0,
      host: 'localhost:27018',
      arbiterOnly: false,
      buildIndexes: true,
      hidden: false,
      priority: 1,
      tags: {},
      secondaryDelaySecs: Long('0'),
      votes: 1
    },
    {
      _id: 1,
      host: 'localhost:27019',
      arbiterOnly: false,
      buildIndexes: true,
      hidden: false,
      priority: 1,
      tags: {},
      secondaryDelaySecs: Long('0'),
      votes: 1
    },
    {
      _id: 2,
      host: 'localhost:27020',
      arbiterOnly: false,
      buildIndexes: true,
      hidden: false,
      priority: 1,
      tags: {},
      secondaryDelaySecs: Long('0'),
      votes: 1
    }
  ],
  protocolVersion: Long('1'),
  writeConcernMajorityJournalDefault: true,
  settings: {
    chainingAllowed: true,
    heartbeatIntervalMillis: 2000,
    heartbeatTimeoutSecs: 10,
    electionTimeoutMillis: 10000,
    catchUpTimeoutMillis: -1,
    catchUpTakeoverDelayMillis: 30000,
    getLastErrorModes: {},
    getLastErrorDefaults: { w: 1, wtimeout: 0 },
    replicaSetId: ObjectId('674cc0215c7d592f3551f4b8')
  }
}
```

Ejecución de casos de pruebas:

- Como primer caso de prueba tenemos la verificación de lectura desde los nodos secundarios, es decir que la información insertada dentro del nodo primario la podamos ver en los demás nodos.
7. Primero creamos la base de datos db_torneo

```
rs0 [direct: primary] test> use bd_torneo
switched to db bd_torneo
```

8. Ahora insertamos las colecciones de nuestra base de datos, como equipos, entrenadores, jugadores, encuentros y posiciones.

```
rs0 [direct: primary] bd_torneo> db.Equipos.insertMany([{"_id": 1, "ciudad": "Bogotá", "estadio": "El Campín", "equipo": "Millonarios"}, {"_id": 2, "ciudad": "Medellín", "estadio": "Atanasio Girardot", "equipo": "Atlético Nacional"}, {"_id": 3, "ciudad": "Bogotá", "estadio": "El Campín", "equipo": "Independiente Santa Fe"}, {"_id": 4, "ciudad": "Cali", "estadio": "Pascual Guerrero", "equipo": "América de Cali"}, {"_id": 5, "ciudad": "Cali", "estadio": "Estadio Deportivo Cali", "equipo": "Deportivo Cali"}, {"_id": 6, "ciudad": "Barranquilla", "estadio": "Metropolitano Roberto Melendez", "equipo": "Junior"}])
{
  acknowledged: true,
  insertedIds: { '_id': 1, '1': 2, '2': 3, '3': 4, '4': 5, '5': 6 }
}
```

```

rs0 [direct: primary] bd_torneo> db.entrenadores.insertMany([
...   "_id": 1,
...   "equipo": "Millonarios",
...   "entrenador": "Alberto Gamero"
... },
... {
...   "_id": 2,
...   "equipo": "Atlético Nacional",
...   "entrenador": "Efraín Juárez"
... },
... {
...   "_id": 3,
...   "equipo": "Independiente Santa Fe",
...   "entrenador": "Pablo Peirano"
... },
... {
...   "_id": 4,
...   "equipo": "America De Cali",
...   "entrenador": "Jorge da Silva"
... },
... {
...   "_id": 5,
...   "equipo": "Deportivo Cali",
...   "entrenador": "Alfredo Arias"
... },
... {
...   "_id": 6,
...   "equipo": "Junior de barranquilla",
...   "entrenador": "Cesar Farias"
... }
... ])
{
  acknowledged: true,
  insertedIds: { '0': 1, '1': 2, '2': 3, '3': 4, '4': 5, '5': 6 }
}
rs0 [direct: primary] bd_torneo> |


```

9. Insertamos los entrenadores de los equipos, y así sucesivamente para lo jugadores, los encuentros y las posiciones.
10. Hacemos un show collections para mostrar las colecciones insertadas en la base de datos

```

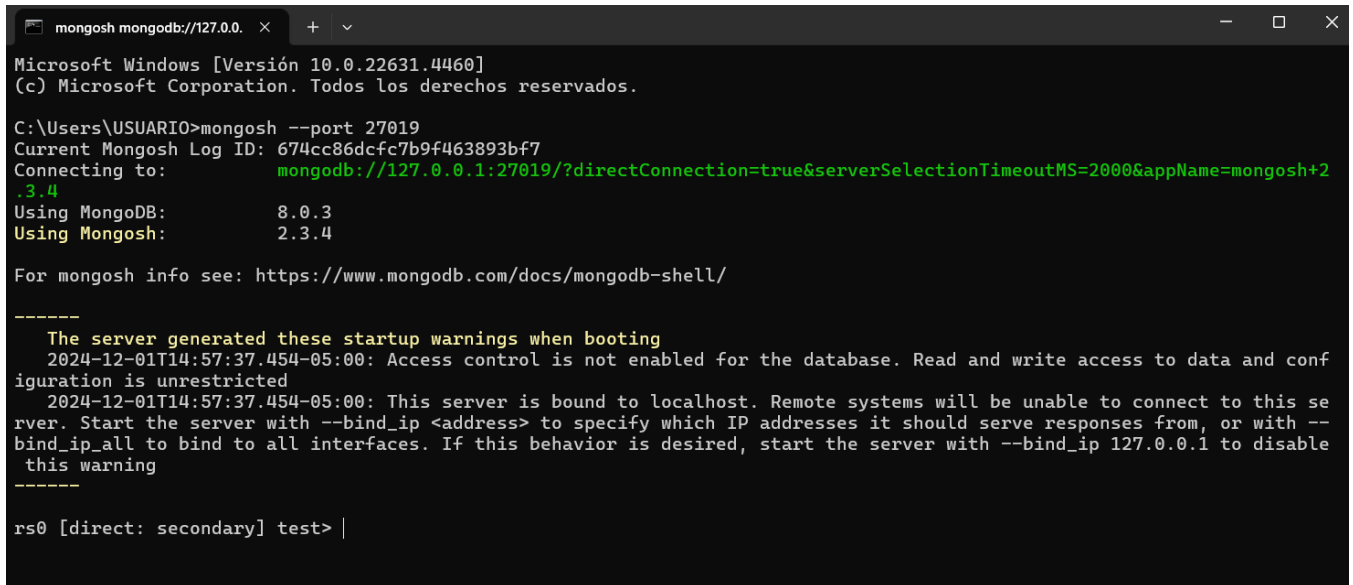
rs0 [direct: primary] bd_torneo> show collections
encuentros
entrenadores
Equipos
jugadores
posiciones

```



- Como podemos observar lo estamos mirando en el nodo primario, lo que queremos hacer es que también se refleje en los nodos secundarios.

11. Para lo cual abro una nueva consola cmd y la abriré mongoshell con el puerto 27019



```
mongosh mongodb://127.0.0.1:27019
Microsoft Windows [Versión 10.0.22631.4460]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO>mongosh --port 27019
Current Mongosh Log ID: 674cc86dcfc7b9f463893bf7
Connecting to:      mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.4
Using MongoDB:      8.0.3
Using Mongosh:       2.3.4

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2024-12-01T14:57:37.454-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2024-12-01T14:57:37.454-05:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
  -----

rs0 [direct: secondary] test> |
```

Si hago el uso de la base de datos bd_torneo, creada en el nodo primario 27018 debería permitirme conectar a la base de datos y si hago la consulta de las colecciones que también se crearon en el nodo primario, también deberían poder verse.


```
mongosh mongodb://127.0.0.1:27019
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO>mongosh --port 27019
Current Mongosh Log ID: 674cc86dcfc7b9f463893bf7
Connecting to:      mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=30000
Using MongoDB:      8.0.3
Using Mongosh:       2.3.4

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2024-12-01T14:57:37.454-05:00: Access control is not enabled for the database.
2024-12-01T14:57:37.454-05:00: This server is bound to localhost. Remote system access is disabled.
2024-12-01T14:57:37.454-05:00: Remote system access is disabled.
-----

rs0 [direct: secondary] test> use bd_torneo
switched to db bd_torneo
rs0 [direct: secondary] bd_torneo> show collections
encuentros
entrenadores
Equipos
jugadores
posiciones
rs0 [direct: secondary] bd_torneo> |
```

Si queremos hacer la consulta de las colecciones tambien podemos verlas desde el nodo secundario, como puede ser equipos

```
mongosh mongodb://127.0.0.1:27027/ bd_torneo × + v
2024-12-01T14:57:37.454-05:00: This server is bound to localhost.
rver. Start the server with --bind_ip <address> to specify which IP a
bind_ip_all to bind to all interfaces. If this behavior is desired, s
this warning
-----

rs0 [direct: secondary] test> use bd_torneo
switched to db bd_torneo
rs0 [direct: secondary] bd_torneo> show collections
encuentros
entrenadores
Equipos
jugadores
posiciones
rs0 [direct: secondary] bd_torneo> db.Equipos.find()
[
  {
    _id: 4,
    ciudad: 'Cali',
    estadio: 'Pascual Guerrero',
    equipo: 'América de Cali'
  },
  {
    _id: 3,
    ciudad: 'Bogotá',
    estadio: 'El Campín',
    equipo: 'Independiente Santa Fe'
  },
  {
    _id: 2,
    ciudad: 'Medellín',
    estadio: 'El Campesino',
    equipo: 'Deportes Pasto'
  },
  {
    _id: 1,
    ciudad: 'Barranquilla',
    estadio: 'El Campesino',
    equipo: 'Deportes Tolima'
  }
]
```

- Como segundo caso de prueba es que los cambios que se hagan en el nodo primario también se reflejen en los demás nodos o replicas, para eso nos dirigimos al nodo primario y hacemos un cambio en la colección encuentros

```
rs0 [direct: primary] bd_torneo> db.encuentros.find({ "_id": 1 })
[
  {
    _id: 1,
    equipo_local: 'Millonarios',
    equipo_visitante: 'Atlético Nacional',
    resultado: '1-1',
    puntos_local: 1,
    puntos_visitante: 1
  }
]
rs0 [direct: primary] bd_torneo> db.encuentros.updateOne({ "_id": 1 }, { $set: { "resultado": "2-2", "puntos_local": 1, "puntos_visitante": 1 } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
rs0 [direct: primary] bd_torneo> |
```

Como podemos ver, realizamos el cambio de un encuentro entre millonarios y nacional, cambio el empate de 1-1 a 2-2, que se vería reflejado en la replica de la siguiente manera:

```
mongosh mongodb://127.0.0.1:27019
{
  _id: 3,
  equipo_local: 'America de Cali',
  equipo_visitante: 'Millonarios',
  resultado: '1-4',
  puntos_local: 0,
  puntos_visitante: 3
},
{
  _id: 4,
  equipo_local: 'Atlético Nacional',
  equipo_visitante: 'Independiente Santa fe',
  resultado: '2-2',
  puntos_local: 1,
  puntos_visitante: 1
},
{
  _id: 1,
  equipo_local: 'Millonarios',
  equipo_visitante: 'Atlético Nacional',
  resultado: '2-2',
  puntos_local: 1,
  puntos_visitante: 1
},
{
  _id: 5,
  equipo_local: 'America de Cali',
  equipo_visitante: 'Millonarios',
  resultado: '1-1',
  puntos_local: 1,
  puntos_visitante: 1
}
```

Como se observa el encuentro con id = 1, tuvo su cambio en la réplica 27019

Como último caso de prueba es simular un fallo en el nodo primario, en el que si en algún momento el nodo primario llega a fallar uno de sus nodos secundarios pase a ser el nodo primario, para ello

Cerramos la instancia del nodo primario es decir la 27018 y si realizamos una consulta nos arrojará error.

```
rs0 [direct: primary] bd_torneo> show collections
Uncaught:
MongoServerError: connect ECONNREFUSED 127.0.0.1:27018
Caused by:
MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27018
Caused by:
Error: connect ECONNREFUSED 127.0.0.1:27018
bd_torneo> |
```

Entonces salimos y ejecutamos el nodo del puerto 27019 se volverá el nodo primario

```
bd_torneo> exit
C:\Users\USUARIO>mongosh --port 27019
Current Mongosh Log ID: 674ccf68c26d7651dc893bf7
Connecting to:      mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.4
Using MongoDB:      8.0.3
Using Mongosh:      2.3.4

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2024-12-01T14:57:37.454-05:00: Access control is not enabled for the database. Read and write access to data and configuration i
s unrestricted
  2024-12-01T14:57:37.454-05:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start
the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to
all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
-----
rs0 [direct: primary] test> |
```

Y sin importar que se haya caído el nodo primario, tendremos la misma información en el

nodo 27019

```
rs0 [direct: primary] test> use bd_torneo
switched to db bd_torneo
rs0 [direct: primary] bd_torneo> show collecntions
MongoshInvalidInputError: [COMMON-10001] 'collecntions' is not a valid argument for "show".
rs0 [direct: primary] bd_torneo> show collections
encuentros
entrenadores
Equipos
jugadores
posiciones
rs0 [direct: primary] bd_torneo> |
```