

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

Liliana Pérez Castiblanco
Yeniffer Alvares Caceres
Julian Garcia Arias

Universidad Iberoamericana
Facultad de ingenieria de software
Materia: Bases de datos avanzadas
Profesor: Willian Ruiz

Bogota D.C

24 de mayo de 2024

REQUERIMIENTOS NO FUNCIONALES DE LA BASE DE DATOS TORNEO DE TENIS.

Requerimientos No Funcionales

El proyecto elaborado requiere tener acceso a la base de datos con el nombre de TorneoDeTenis las 24 horas los 7 días de la semana, con el fin de garantizar dicho acceso y disponibilidad completa se tendrán en cuenta los siguientes requerimientos:

- Como requerimiento principal es mantener el funcionamiento permanente de la base de datos TorneoDeTenis y sus colecciones.
- Se debe replicar la base de datos, en ella deben contener tres nodos, los cuales representaran a tres servidores con el almacenamiento de la misma información.
- Cada uno de los servidores o nodos debe manejar los permisos necesarios para acceder a la información de la base de datos y replicarla.
- En caso de que el servidor o nodo maestro no se pueda acceder el sistema nos permitirá que algún nodo secundario nos de acceso a la información sin generar trauma alguno.
- El sistema debe garantizar que el ingreso, la consulta o la modificación de información se realice sin contratiempos y de la forma más ágil posible.

CREACION DEL REPLICASET

```
> MijReplicaSet = new ReplSetTest ({name: "MireplicaSet", nodes: 3})
```

```
> MijReplicaSet.startSet()
```

```
    }, "waiting for master", timeout);  
  
    return master;  
  },  
  "name" : "MireplicaSet",  
  "useHostName" : true,  
  "host" : "DESKTOP-5KQ6VFL",  
  "oplogSize" : 40,  
  "useSeedList" : false,  
  "keyFile" : undefined,  
  "protocolVersion" : undefined,  
  "waitForKeys" : undefined,  
  "nodeOptions" : {  
    "n0" : undefined,  
    "n1" : undefined,  
    "n2" : undefined  
  },  
  "nodes" : [ ],  
  "ports" : [  
    20000,  
    20001,  
    20002  
  ]  
}
```

Paso 2 -Arrancar los procesos mongod de la replica

```
> MieiReplicaSet.startSet()
ReplSetTest starting set
ReplSetTest n is : 0
{
  "useHostName" : true,
  "oplogSize" : 40,
  "keyFile" : undefined,
  "port" : 20003,
  "replSet" : "MireplicaSet",
  "dbpath" : "$set-$node",
  "restart" : undefined,
  "pathOpts" : {
    "node" : 0,
    "set" : "MireplicaSet"
  },
  "setParameter" : {
    "writePeriodicNoops" : false,
    "numInitialSyncConnectAttempts" : 60
  }
}
```

```
ReplSetTest n is : 1
{
  "useHostName" : true,
  "oplogSize" : 40,
  "keyFile" : undefined,
  "port" : 20004,
  "replSet" : "MireplicaSet",
  "dbpath" : "$set-$node",
  "restart" : undefined,
  "pathOpts" : {
    "node" : 1,
    "set" : "MireplicaSet"
  },
  "setParameter" : {
    "writePeriodicNoops" : false,
    "numInitialSyncConnectAttempts" : 60
  }
}
```

```
ReplSetTest n is : 2
{
  "useHostName" : true,
  "oplogSize" : 40,
  "keyFile" : undefined,
  "port" : 20005,
  "replSet" : "MireplicaSet",
  "dbpath" : "$set-$node",
  "restart" : undefined,
  "pathOpts" : {
    "node" : 2,
    "set" : "MireplicaSet"
  },
  "setParameter" : {
    "writePeriodicNoops" : false,
    "numInitialSyncConnectAttempts" : 60
  }
}
ReplSetTest Starting
```

Puertos:

```
connection to DESKTOP-5KQ6VFL:20003,
connection to DESKTOP-5KQ6VFL:20004,
connection to DESKTOP-5KQ6VFL:20005

connection to DESKTOP-5KQ6VFL:20003,
connection to DESKTOP-5KQ6VFL:20004,
connection to DESKTOP-5KQ6VFL:20005
```

Paso 3 -Arrancar el proceso de replica

> `MiejReplicaSet.initiate()`

```
reconfiguring replica set to add in other nodes
{
  "replSetReconfig" : {
    "_id" : "MireplicaSet",
    "protocolVersion" : 1,
    "members" : [
      {
        "_id" : 0,
        "host" : "DESKTOP-5KQ6VFL:20003"
      },
      {
        "_id" : 1,
        "host" : "DESKTOP-5KQ6VFL:20004"
      },
      {
        "_id" : 2,
        "host" : "DESKTOP-5KQ6VFL:20005"
      }
    ],
    "version" : 2
  }
}
```

| Portapaneles | | Organizar | | Nuevo | Abrir |
|--|--|-----------------------|---------------------|--------|-------|
| Este equipo > Disco local (C:) > DATA > db > | | | | | |
| Nombre | | Fecha de modificación | Tipo | Tamaño | |
| MireplicaSet-0 | | 24/05/2024 2:30 p. m. | Carpeta de archivos | | |
| MireplicaSet-1 | | 24/05/2024 2:30 p. m. | Carpeta de archivos | | |
| MireplicaSet-2 | | 24/05/2024 2:30 p. m. | Carpeta de archivos | | |

Paso 4 -Prueba del grupo de replica

>conn=new Mongo(«DESKTOP-5KQ6VFL:20003»)

```
> conn=new Mongo("DESKTOP-5KQ6VFL:20003")
connection to DESKTOP-5KQ6VFL:20003
> testDB=conn.getDB("TorneoDeTenis")
TorneoDeTenis
>
```

VIDEO:

<https://youtu.be/WeOm4VmHcR8>

PRUEBAS:

1.

| Tipo de prueba | Objetivo |
|----------------|---|
| Replicación | Verificar que se hayan creado el nodo primario y los secundarios. |

Evidencias:

```
{
  "hosts" : [
    "DESKTOP-5KQ6VFL:20003",
    "DESKTOP-5KQ6VFL:20004",
    "DESKTOP-5KQ6VFL:20005"
  ],
  "setName" : "MireplicaSet",
  "setVersion" : 2,
  "ismaster" : true,
  "secondary" : false,
  "primary" : "DESKTOP-5KQ6VFL:20003",
  "me" : "DESKTOP-5KQ6VFL:20003",
  "electionId" : ObjectId("7fffffff0000000000000001"),
  "lastWrite" : {
    "opTime" : {
      "ts" : Timestamp(1716578994, 1),
      "t" : NumberLong(1)
    },
    "lastWriteDate" : ISODate("2024-05-24T19:29:54Z"),
    "majorityOpTime" : {
      "ts" : Timestamp(1716578994, 1),
      "t" : NumberLong(1)
    }
  }
}
```

2.

| Tipo de prueba | Objetivo |
|----------------|--|
| Disponibilidad | Ingresa 2 o más documentos en 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. |

Evidencias:

Colección Jugadores.

```
> testDB.Jugadores.insert(
... {
... _id:"345678",
... nombre_apellido:"Monica Lopez",
... edad:"25",
... categoría:"Segunda"
... });
WriteResult({ "nInserted" : 1 })
>
```

```
> testDB.Jugadores.insert(
... {
... _id:"456789",
... nombre_apellido:"Maria Dominguez",
... edad:"23",
... categoría:"Segunda"
... });
WriteResult({ "nInserted" : 1 })
>
```

```
> testDB.Jugadores.count();
2
>
```

Colección Jueces.

```
> testDB.Jueces.insert(
... {
... _id:"3456",
... nombre:"Manuel Caro",
... nacionalidad:"Colombiano",
... rol:"Arbitro de línea"
... });
WriteResult({ "nInserted" : 1 })
>
```

```
> testDB.Jueses.insert(
... {
...   _id:"24109",
...   nombre:"Andres Cepeda",
...   nacionalidad:"Colombiano",
...   rol:"Arbitro de linea"
... });
WriteResult({ "nInserted" : 1 })
>
```

```
> testDB.Jueses.insert(
... {
...   _id:"345609",
...   nombre:"Julio Malaver",
...   nacionalidad:"Colombiano",
...   rol:"Arbitro de linea"
... });
WriteResult({ "nInserted" : 1 })
>
>
```

```
> testDB.Jueses.insert(
... {
...   _id:"456756",
...   nombre:"Eduardo Espitia",
...   nacionalidad:"Colombiano",
...   rol:"Arbitro de linea",
... });
WriteResult({ "nInserted" : 1 })
>
```

```
> testDB.Jueses.count();
4
>
```

Colección Torneo

```
> testDB.Torneo.insert(
... {
...   "_id": {
...     "$type": "objectId"
...   },
...   "Cancha": "1, 2, 3, 4,5,6",
...   "ronda": "Cuartos",
...   "Fecha": "20/05/2024, 22/05/2024"
... });
```

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


Colección Jugadores.

```
> testDB.Jugadores.insert(
... {
... _id: "4",
... nombre_apellido:"Manuela veltran",
... edad: "25",
... categoría: "Segunda"
... });
WriteResult({ "nInserted" : 1 })
>
```

3.

| Tipo de prueba | Objetivo |
|---------------------|---|
| Tolerancia a fallos | Prueba de desconexión del nodo primario y promoción de algunos de los nodos secundarios a primario. |

Evidencias:

```
> connSecondary = new Mongo("DESKTOP-5KQ6VFL:20004")
connection to DESKTOP-5KQ6VFL:20004
> secondaryTestDB = connSecondary.getDB("TorneoDeTenis")
TorneoDeTenis
>
```

```
> secondaryTestDB.isMaster()
{
  "hosts" : [
    "DESKTOP-5KQ6VFL:20003",
    "DESKTOP-5KQ6VFL:20004",
    "DESKTOP-5KQ6VFL:20005"
  ],
  "setName" : "MireplicaSet",
  "setVersion" : 2,
  "ismaster" : false,
  "secondary" : true,
  "primary" : "DESKTOP-5KQ6VFL:20003",
  "me" : "DESKTOP-5KQ6VFL:20004",
  "lastWrite" : {
    "opTime" : {
      "ts" : Timestamp(1716582192, 2),
      "t" : NumberLong(1)
    }
  }
}
```

En este caso obtenemos un error porque por defecto, tal y como comentamos en los nodos secundarios en un grupo de réplica, no admiten operaciones ni de escritura ni de lectura. Todas las operaciones deben realizarse siempre sobre el nodo principal.

```

} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DBQuery.prototype.count@src/mongo/shell/query.js:376:11
DBCollection.prototype.count@src/mongo/shell/collection.js:1401:12
@(shell):1:1
>

```

```

> secondaryTestDB.Jugadores.count()
2
> secondaryTestDB.Jugadores.findOne()
{
  "_id" : "345678",
  "nombre_apellido" : "Monica Lopez",
  "edad" : "25",
  "categoría" : "Segunda"
}
>

```

Detener el nodo primario

```

> connPrimary = new Mongo("DESKTOP-5KQ6VFL:20003")
connection to DESKTOP-5KQ6VFL:20003
> primaryDB = connPrimary.getDB("TorneoDeTenis")
TorneoDeTenis
> primaryDB.isMaster()
{
  "hosts" : [
    "DESKTOP-5KQ6VFL:20003",
    "DESKTOP-5KQ6VFL:20004",
    "DESKTOP-5KQ6VFL:20005"
  ],
  "setName" : "MireplicaSet",
  "setVersion" : 2,
  "ismaster" : true,
  "secondary" : false,
  "primary" : "DESKTOP-5KQ6VFL:20003",
  "me" : "DESKTOP-5KQ6VFL:20003",
  "electionId" : ObjectId("7fffffff0000000000000001"),
  "lastWrite" : {
    "opTime" : {
      "ts" : Timestamp(1716582192, 2),
      "t" : NumberLong(1)
    },
    "lastWriteDate" : ISODate("2024-05-24T20:23:12Z")
  }
}

```

```

    "ok" : 1,
    "$clusterTime" : {
      "clusterTime" : Timestamp(1716582655, 1),
      "signature" : {
        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
        "keyId" : NumberLong(0)
      }
    },
    "operationTime" : Timestamp(1716582192, 2)
  }
}
> primaryDB.adminCommand({shutdown : 1})
{
  "operationTime" : Timestamp(1716582192, 2),
  "ok" : 0,
  "errmsg" : "shutdown must run from localhost when running db without auth",
  "code" : 13,
  "codeName" : "Unauthorized",
  "$clusterTime" : {
    "clusterTime" : Timestamp(1716582655, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
}

```

4.

| Tipo de prueba | Objetivo |
|----------------|--|
| Disponibilidad | Verificar cuál de los nodos secundarios es ahora el nodo primario. |

Evidencia:

```

}
}
> connNewPrimary = new Mongo("localhost:20004")
connection to localhost:20004
> newPrimaryDB = connNewPrimary.getDB("TorneoDeTenis")
TorneoDeTenis
> newPrimaryDB.isMaster()
{
  "hosts" : [
    "DESKTOP-5KQ6VFL:20003",
    "DESKTOP-5KQ6VFL:20004",
    "DESKTOP-5KQ6VFL:20005"
  ]
}

```

```
],
  "setName" : "MireplicaSet",
  "setVersion" : 2,
  "ismaster" : true,
  "secondary" : false,
  "primary" : "DESKTOP-5KQ6VFL:20004",
  "me" : "DESKTOP-5KQ6VFL:20004",
  "electionId" : ObjectId("7fffffff0000000000000002"),
  "lastWrite" : {
    "opTime" : {
      "ts" : Timestamp(1716596593, 1),
      "t" : NumberLong(2)
    }
  },
}
```

Detener el ReplicaSet de pruebas

```
code:0
2024-05-24T20:14:52.536-0500 I - [js] shell: stopped mongo
005
ReplSetTest stop *** Mongod in port 20005 shutdown with code (0) *
ReplSetTest stopSet stopped all replica set nodes.
ReplSetTest stopSet deleting all dbpaths
ReplSetTest stopSet deleting dbpath: /data/db/MireplicaSet-0
ReplSetTest stopSet deleting dbpath: /data/db/MireplicaSet-1
ReplSetTest stopSet deleting dbpath: /data/db/MireplicaSet-2
ReplSetTest stopSet deleted all dbpaths
ReplSetTest stopSet *** Shut down repl set - test worked ****
>
```

