

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

Noviembre 2023.

### Desarrollo de la actividad

Link repositorio: <https://github.com/GjerezSierra/Bases-de-datos-NoSQL>

Link del Video: <https://youtu.be/PtjF6eTQQJk>

La actividad No. 1 la realicé con base en la información que encontré relacionada con el desarrollo de la Copa Mundial de la FIFA 2014 en Brasil, para dar continuidad a la actividad realizare los requerimientos de la actividad No. 2 usando la base de datos anterior:

#### 1. Requerimientos no funcionales:

En el contexto de la primera actividad, los estándares de calidad enfatizan la redundancia y la disponibilidad constante (24x7). La replicación en MongoDB es un procedimiento que implica mantener réplicas exactas de los datos en diversos servidores, con el propósito de asegurar la continuidad de los datos, brindar redundancia y fortalecer la resistencia ante posibles fallos.

Para llevar a cabo la configuración de la replicación en MongoDB, se emplea lo que se conoce como un conjunto de réplicas (replica set). Este conjunto consta de múltiples nodos de MongoDB que contienen datos idénticos. Un nodo primario es responsable de gestionar las operaciones de escritura, mientras que los nodos secundarios mantienen copias precisas de los datos alojados en el nodo primario.

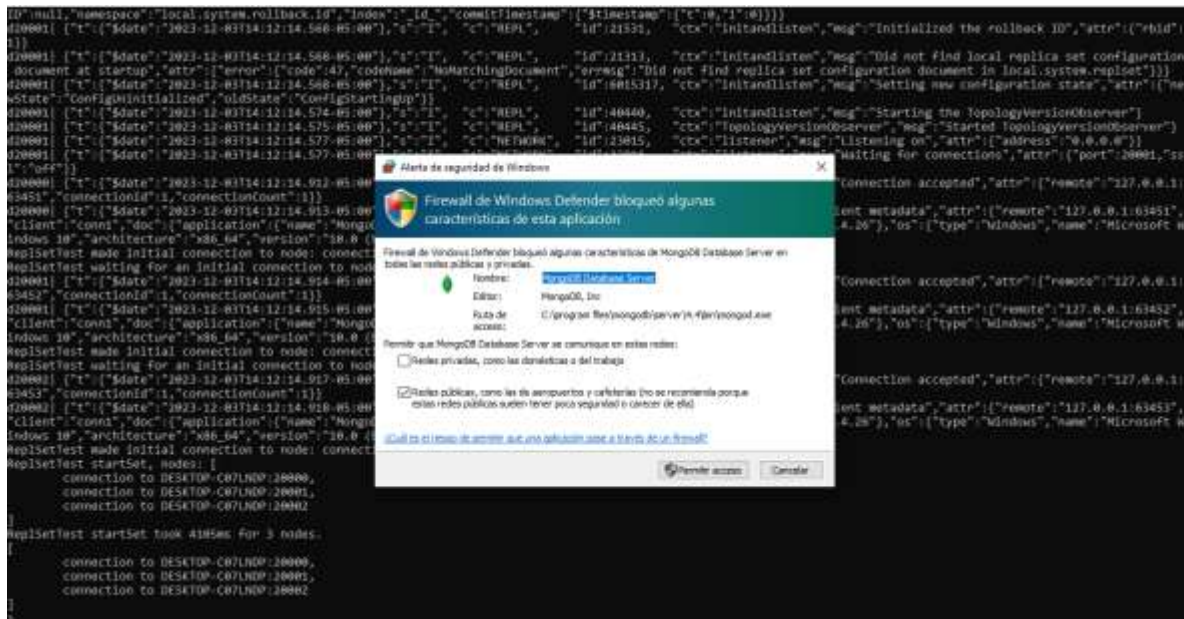
#### 2. Replicación de la base de datos Mundial 2014:




##### Configurar los nodos del conjunto de réplicas

- Inicia varios procesos mongod (instancias de MongoDB), uno para cada nodo. Cada nodo debe tener una configuración similar y estar conectado en una red que les permita comunicarse entre sí.

```
},
  "name" : "MireplicaSet",
  "useHostName" : true,
  "host" : "DESKTOP-C07LNDP",
  "oplogSize" : 40,
  "useSeedList" : false,
  "keyFile" : undefined,
  "protocolVersion" : undefined,
  "waitForKeys" : undefined,
  "seedRandomNumberGenerator" : true,
  "isConfigServer" : undefined,
  "nodeOptions" : {
    "n0" : undefined,
    "n1" : undefined,
    "n2" : undefined
  },
  "nodes" : [ ],
  "ports" : [
    20000,
    20001,
    20002
  ]
}
```

- Inicia el nodo primario. Este nodo será el primer miembro del conjunto de réplicas.



	MireplicaSet-0	3/12/2023 2:13 p. m.	Carpeta de archivos
	MireplicaSet-1	3/12/2023 2:13 p. m.	Carpeta de archivos
	MireplicaSet-2	3/12/2023 2:13 p. m.	Carpeta de archivos

- Agrega nodos secundarios al conjunto de réplicas. Para cada nodo secundario, conéctalo al nodo primario y haz que se sincronice con los datos.

## Configurar el conjunto de réplicas

- Inicia el shell de MongoDB y conéctate a uno de los nodos.

```
MongoDB shell version v4.4.26
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("63a17e60-cd94-4094-a003-a8d97f55fc18") }
MongoDB server version: 4.4.26
---
The server generated these startup warnings when booting:
  2023-12-03T13:59:28.665-05:00: Access control is not enabled for the database. Read and write access to data and
  configuration is unrestricted
---
> conn-new Mongo("DESKTOP-C07LNDP:20000")
connection to DESKTOP-C07LNDP:20000
> testDB=conn.getDB("Mundial-2014")
Mundial-2014
> testDB.isMaster()
uncaught exception: ReferenceError: testDB is not defined :
@ (shell):1:1
> testDB.isMasterIO
uncaught exception: ReferenceError: testDB is not defined :
@ (shell):1:1
> testDB.isMaster()
uncaught exception: ReferenceError: testDB is not defined :
@ (shell):1:1
> testDB=conn.getDB("Mundial-2014")
Mundial-2014
> testDB.isMaster()
{
  "topologyVersion" : {
    "processId" : ObjectId("656cd30d3bb94cbd7f01a3e1"),
    "counter" : NumberLong(8)
  }
}
```

- Inicia el conjunto de réplicas. Para hacer esto, utiliza el método `rs.initiate()` para iniciar el conjunto de réplicas y especificar el nodo primario.

```
> testDB.isMaster()
{
  "topologyVersion" : {
    "processId" : ObjectId("656cd30d3bb94cbd7f01a3e3"),
    "counter" : NumberLong(8)
  },
  "hosts" : [
    "DESKTOP-C07LNDP:20000",
    "DESKTOP-C07LNDP:20001",
    "DESKTOP-C07LNDP:20002"
  ],
  "setName" : "MireplicaSet",
  "setVersion" : 3,
  "ismaster" : true,
  "secondary" : false,
  "primary" : "DESKTOP-C07LNDP:20000",
  "me" : "DESKTOP-C07LNDP:20000",
  "electionId" : ObjectId("7fffffff0000000000000001"),
  "lastWrite" : {
    "opTime" : {
      "ts" : Timestamp(1701643334, 1),
      "t" : NumberLong(1)
    },
    "lastWriteDate" : ISODate("2023-12-03T22:42:14Z"),
    "majorityOpTime" : {
      "ts" : Timestamp(1701643334, 1),
      "t" : NumberLong(1)
    }
  }
}
```

- Agrega los nodos secundarios al conjunto de réplicas utilizando el método `rs.add()`. Asegúrate de que los nodos secundarios se sincronicen correctamente con el nodo primario.

### Monitorear y administrar el conjunto de réplicas

- Verifica el estado del conjunto de réplicas utilizando el método `rs.status()` desde el shell de MongoDB. Esto te mostrará el estado de cada miembro del conjunto de réplicas.
- Realiza tareas de administración como agregar o eliminar nodos, realizar failover manualmente (si es necesario), y monitorear el estado de los miembros del conjunto de réplicas.

### Evaluación del Proceso de Réplica del Conjunto de Nodos

Este paso involucra una evaluación exhaustiva del proceso de replicación entre los nodos que componen el conjunto en MongoDB. La apertura de una nueva consola de MongoDB marca el inicio de este proceso de verificación detallada, que sigue los siguientes pasos precisos:

**Ingreso al Entorno del Mongo Shell:** El primer paso consiste en acceder al entorno del Mongo Shell desde la consola recién abierta, permitiendo así la interacción directa con la base de datos.

**Análisis del Estado de la Réplica:** Mediante la ejecución del comando `rs.status()`, se obtiene un informe detallado sobre el estado actual del conjunto de réplicas. Este análisis revela información vital, incluyendo la configuración, salud y roles de los nodos dentro del conjunto.

**Exploración de Métricas de Réplica:** Además, se pueden utilizar comandos como `rs.printReplicationInfo()` para examinar métricas específicas de replicación, tales como el retraso de replicación, tamaño de los datos replicados y otros aspectos cruciales para evaluar la sincronización y consistencia entre los nodos.

```

    "majorityWriteDate" : ISODate("2023-12-03T22:42:14Z")
  },
  "maxBsonObjectSize" : 16777216,
  "maxMessageSizeBytes" : 48000000,
  "maxWriteBatchSize" : 100000,
  "localTime" : ISODate("2023-12-03T22:47:35.818Z"),
  "logicalSessionTimeoutMinutes" : 30,
  "connectionId" : 20,
  "minWireVersion" : 0,
  "maxWireVersion" : 9,
  "readOnly" : false,
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1701643334, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA-"),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1701643334, 1)
}
> tesDB.Arbitros.insert({id:"5060", Nombres:"Miguel Angel", Apellido:"Sanchez florez", Edad:"45", Cedula:"34"});
writeResult({ "nInserted" : 1 })
> testDB.Arbitros.count();
1

```

## Verificación de la Tolerancia a Fallos

**Descripción:** Esta prueba evalúa la capacidad del conjunto de réplicas para resistir la falla del nodo primario y el proceso de selección automática de un nodo secundario como nuevo nodo primario.

**Precondiciones:** Se requiere que el conjunto de réplicas esté en línea y ejecutando MongoDB.

### Pasos a seguir:

- Detener la instancia de MongoDB que actúa como nodo primario.
- Iniciar una sesión en el Mongo shell desde cualquier nodo secundario.
- Ejecutar el comando `rs.status()` para obtener el estado actual del conjunto de réplicas.

## **Resultados esperados:**

Se espera que uno de los nodos secundarios sea seleccionado automáticamente para asumir el rol de nuevo nodo primario. Esta acción subraya la capacidad del conjunto de réplicas para continuar operando de manera fluida y automática incluso después de la falla del nodo primario, manteniendo así la integridad y disponibilidad de los datos.

## **Evaluación del Proceso de Réplica del Conjunto de Nodos**

Este paso involucra una evaluación exhaustiva del proceso de replicación entre los nodos que componen el conjunto en MongoDB. La apertura de una nueva consola de MongoDB marca el inicio de este proceso de verificación detallada, que sigue los siguientes pasos precisos:

**Ingreso al Entorno del Mongo Shell:** El primer paso consiste en acceder al entorno del Mongo Shell desde la consola recién abierta, permitiendo así la interacción directa con la base de datos.

**Análisis del Estado de la Réplica:** Mediante la ejecución del comando `rs.status()`, se obtiene un informe detallado sobre el estado actual del conjunto de réplicas. Este análisis revela información vital, incluyendo la configuración, salud y roles de los nodos dentro del conjunto.

**Exploración de Métricas de Réplica:** Además, se pueden utilizar comandos como `rs.printReplicationInfo()` para examinar métricas específicas de replicación, tales como el retraso de replicación, tamaño de los datos replicados y otros aspectos cruciales para evaluar la sincronización y consistencia entre los nodos.

## **Verificación del Nodo Maestro**

En este paso, se realiza una comprobación específica para determinar si el nodo actual es el nodo maestro (master) en un conjunto de réplicas. Para llevar a cabo esta verificación, se utiliza el comando `TestBD.isMaster()` en el entorno del shell de MongoDB. El resultado de este comando proporciona una serie de atributos esenciales que confirman la identidad y el rol del nodo en el conjunto de réplicas:

"hosts": Muestra los nombres de los hosts involucrados en el conjunto de réplicas, identificando la topología del conjunto.

"setName": Indica el nombre del conjunto de réplicas al que pertenece el nodo.

"isMaster": Este atributo determina si el nodo es el nodo maestro (true) o no (false).

"secondary": Confirma si el nodo es secundario (true) o no (false), lo que ayuda a identificar su función en el conjunto.

"primary" y "me": Estos atributos revelan la identidad del nodo primario, confirmando si el nodo actual es el nodo maestro.