**Documento de Requerimientos No Funcionales y Pruebas**

**Objetivo**

Garantizar la redundancia y disponibilidad 24x7 de la base de datos del torneo de fútbol Copa América, asegurando que los datos estén siempre accesibles y consistentes incluso en caso de fallos de servidor.

**Requerimientos No Funcionales**

1. **Disponibilidad**: La base de datos debe estar disponible las 24 horas del día, los 7 días de la semana.
2. **Redundancia**: Debe existir una copia exacta de los datos en al menos tres nodos para garantizar la redundancia.
3. **Consistencia**: Los datos replicados en todos los nodos deben estar siempre consistentes con los datos del nodo primario.
4. **Tolerancia a Fallos**: En caso de fallo del nodo primario, uno de los nodos secundarios debe asumir automáticamente el rol de primario sin interrumpir el servicio.
5. **Escalabilidad**: La configuración debe permitir la fácil adición de más nodos en el futuro sin afectar la disponibilidad y consistencia de los datos.

**Estrategia de Replicación**

**Definición de la Estrategia**

La estrategia de replicación seleccionada es la configuración de un conjunto de réplicas (Replica Set) de MongoDB con tres nodos: un nodo primario y dos nodos secundarios. El nodo primario manejará todas las operaciones de escritura, mientras que los nodos secundarios replicarán los datos de forma asíncrona.

**Comandos para Crear el Entorno de Replicación**

1. Configurar Mongo en Windows para usar Replicación:
   * Se debe parar el servicio de MongoDB en el sistema de Windows

*net stop MongoDB*

* + Se debe editar el archivo mongod.cfg en la ruta C:\Program Files\MongoDB\Server\<version>\bin\mongod.cfg y agregar las siguientes líneas:

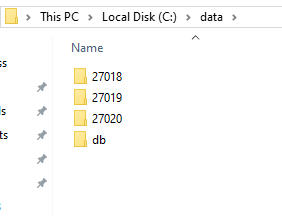
*replication:*

*replSetName: "rs0"*

* + Iniciar de nuevo el servicio de MongoDB

*net start MongoDB*

* Crear las carpetas para los datos de cada nodo



* + Inicializar los servidores de Mongod para poder conectarse con cada nodo creado en este documento

*mongod --port 27018 --dbpath "C:\data\27018" --replSet rs0*

*mongod --port 27019 --dbpath "C:\data\27019" --replSet rs0*

*mongod --port 27020 --dbpath "C:\data\27020" --replSet rs0*

* + Iniciar Mongosh para poder escribir comandos a la base de datos

*cd C:\Users\Programacion\AppData\Local\Programs\mongosh*

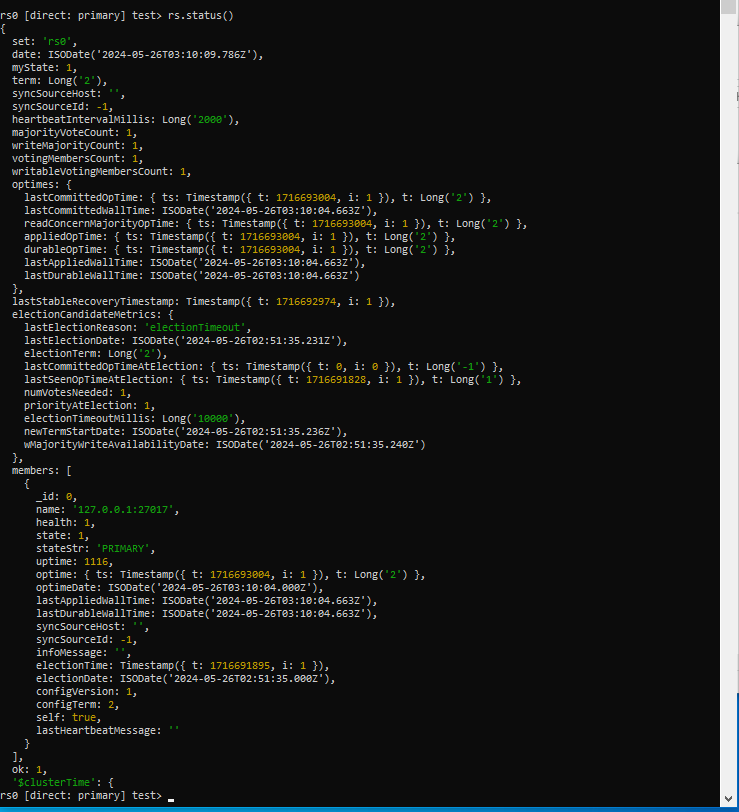
*.\mongosh.exe*

1. **Comprobar si se está ejecutando Mongos en replSet**

El comando

*rs.status()*

Muestra lo siguiente por lo que podemos concluir que ya se tiene el sistema de replicación disponible para ser utilizado:



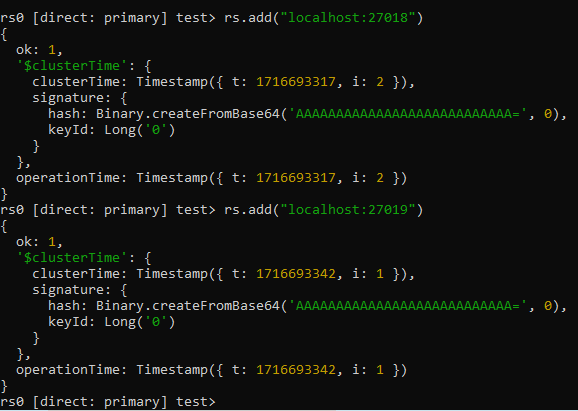
1. **Configurar los Nodos Secundarios**

Con los comandos

*rs.add("localhost:27018")*

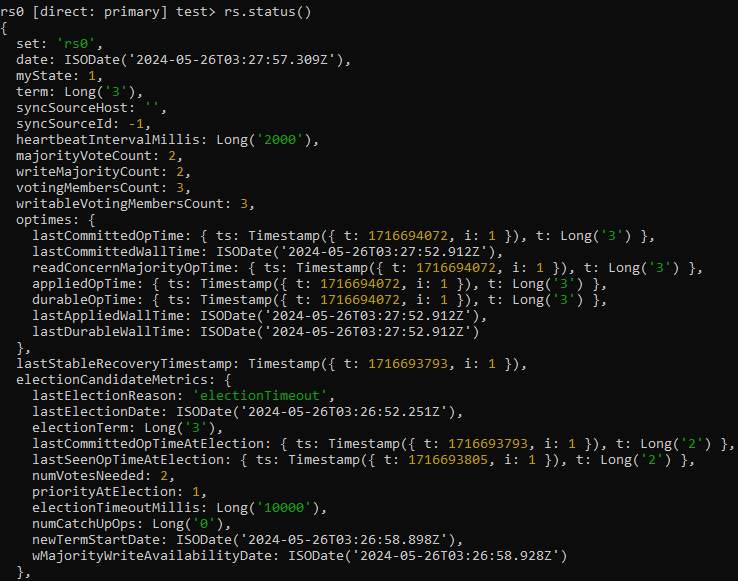
*rs.add("localhost:27019")*

Se generan 2 nuevos nodos secundarios.



1. **Verificar el status del sistema**

rs.status()







1. **Insertar Datos Iniciales en el Nodo Primario**

use torneo\_copa\_america

db.createCollection("deportistas")

db.createCollection("equipos")

db.createCollection("entrenadores")

db.createCollection("arbitros")

db.createCollection("encuentros\_deportivos")

db.createCollection("resultados")

db.createCollection("tabla\_de\_posiciones")

# usando mongo import

mongoimport --host rs0/localhost:27017 --db torneo\_copa\_america --collection deportistas --file "C:\Users\Programacion\Documents\MongoDB\deportistas.json" --jsonArray

mongoimport --host rs0/localhost:27017 --db torneo\_copa\_america --collection equipos --file "C:\Users\Programacion\Documents\MongoDB\equipos.json" --jsonArray

mongoimport --host rs0/localhost:27017 --db torneo\_copa\_america --collection entrenadores --file "C:\Users\Programacion\Documents\MongoDB\entrenadores.json" --jsonArray

mongoimport --host rs0/localhost:27017 --db torneo\_copa\_america --collection arbitros --file "C:\Users\Programacion\Documents\MongoDB\arbitros.json" --jsonArray

mongoimport --host rs0/localhost:27017 --db torneo\_copa\_america --collection encuentros\_deportivos --file "C:\Users\Programacion\Documents\MongoDB\encuentros\_deportivos.json" --jsonArray

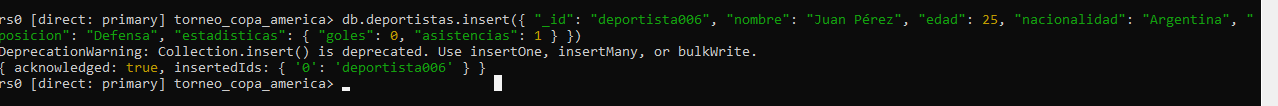
mongoimport --host rs0/localhost:27017 --db torneo\_copa\_america --collection resultados --file "C:\Users\Programacion\Documents\MongoDB\resultados.json" --jsonArray

mongoimport --host rs0/localhost:27017 --db torneo\_copa\_america --collection tabla\_de\_posiciones --file "C:\Users\Programacion\Documents\MongoDB\tabla\_de\_posiciones.json" –jsonArray

**Verificación de la Replicación**

1. **Insertar Datos en el Nodo Primario**

*db.deportistas.insert({ "\_id": "deportista006", "nombre": "Juan Pérez", "edad": 25, "nacionalidad": "Argentina", "posicion": "Defensa", "estadisticas": { "goles": 0, "asistencias": 1 } })*



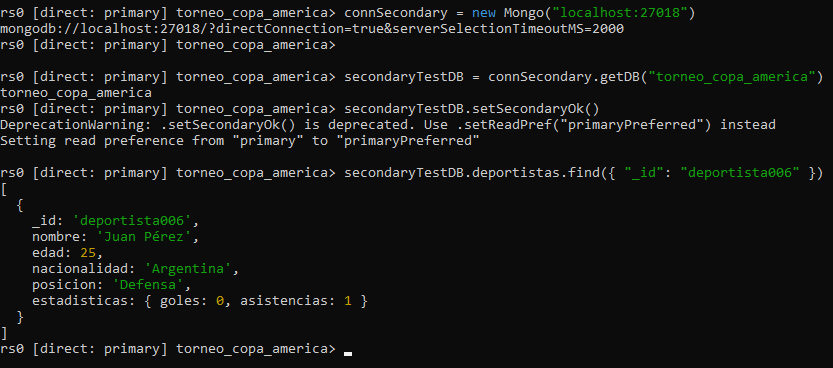
1. **Verificar la Replicación en un Nodo Secundario** Nos conectaremos al nodo en local host 27018 para verificar que los datos insertados en el nodo primario se han replicado correctamente:

connSecondary = new Mongo("localhost:27018")

secondaryTestDB = connSecondary.getDB("torneo\_copa\_america")

secondaryTestDB.setSecondaryOk()

secondaryTestDB.deportistas.find({ "\_id": "deportista006" })



**Casos de Prueba**

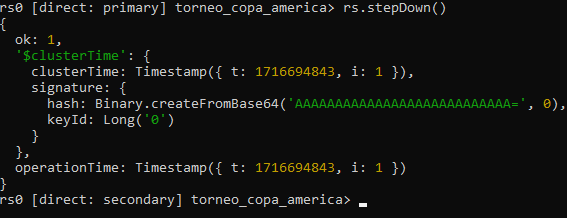
**Prueba de Disponibilidad y Redundancia**

1. **Prueba de Replicación Inicial**
   * Insertar datos en el nodo primario.
   * Verificar que los datos se replican en los nodos secundarios.
2. **Prueba de Tolerancia a Fallos**
   * Apagar el nodo primario.
   * Verificar que uno de los nodos secundarios se promueve a primario.
   * Insertar nuevos datos en el nuevo nodo primario.
   * Verificar que los datos se replican en los nodos restantes.
3. **Prueba de Escalabilidad**
   * Agregar un nuevo nodo al Replica Set.
   * Verificar que el nuevo nodo recibe todos los datos replicados y se mantiene sincronizado con el nodo primario.

**Ejecución de los Casos de Prueba**

1. **Apagar el Nodo Primario**

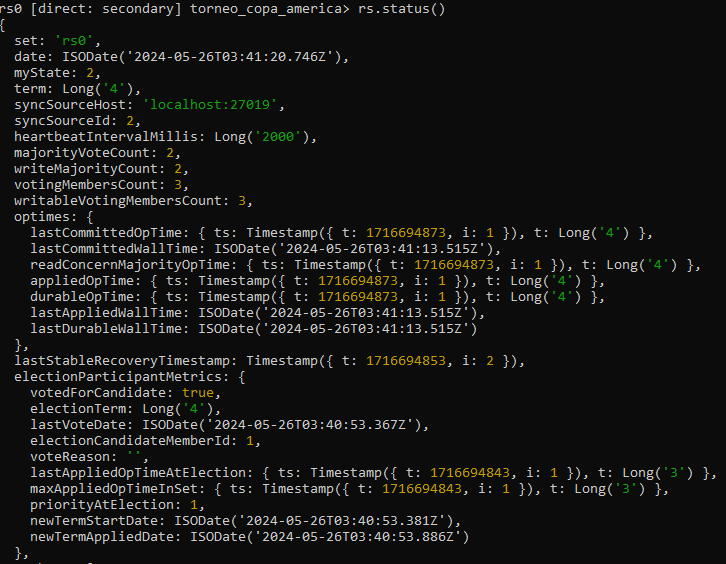
*rs.stepDown()*



1. **Verificar el Nuevo Nodo Primario**

Aqui podemos evidenciar que el nodo en local host 27018 se ha convertido en el nodo primario.

*rs.status()*







1. **Agregar un Nuevo Nodo**

*rs.add("localhost:27020")*



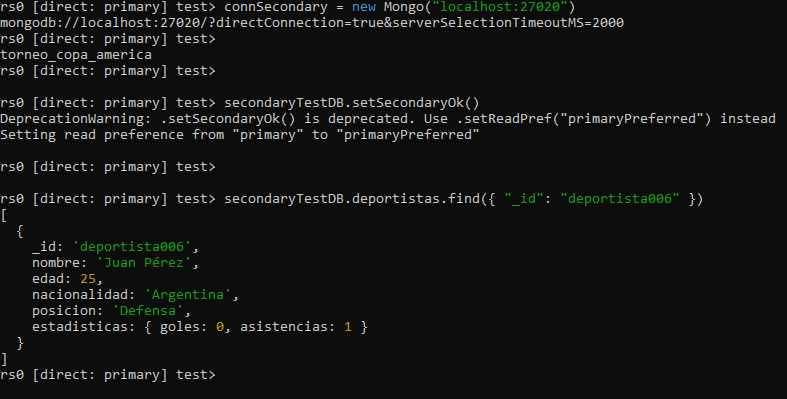
1. **Comprobar que los datos del nodo primario ya estén en el nuevo nodo creado**

connSecondary = new Mongo("localhost:27020")

secondaryTestDB = connSecondary.getDB("torneo\_copa\_america")

secondaryTestDB.setSecondaryOk()

secondaryTestDB.deportistas.find({ "\_id": "deportista006" })



**Reporte de Resultados y Análisis**

1. **Resultados de la Replicación Inicial**
   * Los datos se replicaron correctamente en los nodos secundarios.
2. **Resultados de la Prueba de Tolerancia a Fallos**
   * Un nodo secundario se promovió a primario correctamente.
   * Los datos insertados en el nuevo nodo primario se replicaron en los nodos restantes.
3. **Resultados de la Prueba de Escalabilidad**
   * El nuevo nodo se agregó y se sincronizó correctamente con los datos existentes.