**Documento de Requerimientos No Funcionales y Estrategia de Particionamiento para la Actividad de Sharding**

**Objetivo**

Implementar el particionamiento horizontal (Sharding) de la base de datos del torneo de fútbol Copa América, asegurando escalabilidad, rendimiento y distribución de la carga de almacenamiento entre varios nodos de MongoDB.

**Requerimientos No Funcionales**

1. **Disponibilidad**: La base de datos debe estar disponible 24x7.
2. **Redundancia**: Cada shard debe tener réplicas para garantizar la redundancia de los datos.
3. **Consistencia**: Los datos deben estar consistentes entre los shards y sus réplicas.
4. **Tolerancia a Fallos**: En caso de fallo de un nodo, las réplicas deben asumir su rol sin interrumpir el servicio.
5. **Escalabilidad**: La configuración debe permitir la fácil adición de nuevos shards en el futuro.

**Estrategia de Particionamiento**

**Definición de la Estrategia**

La estrategia de particionamiento seleccionada incluye configurar Sharding con tres shards, cada uno con sus propias réplicas para garantizar redundancia y disponibilidad. Los datos se dividirán por la clave **equipo** para distribuir uniformemente la carga de datos.

**Comandos para Configurar el Entorno de Particionamiento**

1. **Configuración Inicial de MongoDB**

**Editar el archivo mongod.cfg para incluir Sharding y replicación**:

*# mongod.conf*

*# for documentation of all options, see:*

*# http://docs.mongodb.org/manual/reference/configuration-options/*

*# Where and how to store data.*

*storage:*

*dbPath: C:\Program Files\MongoDB\Server\7.0\data*

*# where to write logging data.*

*systemLog:*

*destination: file*

*logAppend: true*

*path: C:\Program Files\MongoDB\Server\7.0\log\mongod.log*

*# network interfaces*

*net:*

*port: 27017*

*bindIp: 127.0.0.1*

*#processManagement:*

*#security:*

*#operationProfiling:*

*#replication:*

*replication:*

*replSetName: "rsConfig"*

*#sharding:*

*sharding:*

*clusterRole: "configsvr"*

*## Enterprise-Only Options:*

*#auditLog:*

**Parar e iniciar el servicio de MongoDB**:

*net stop MongoDB*

*net start MongoDB*

1. **Crear las Carpetas para los Datos de Cada Nodo**:

*mkdir C:\data\config*

*mkdir C:\data\shard1*

*mkdir C:\data\shard2*

*mkdir C:\data\shard3*

1. **Inicializar las Instancias de mongod para Config Server y Shards**:

*mongod --configsvr --port 27017 --dbpath "C:\data\config" --replSet "rsConfig"*

*mongod --shardsvr --port 27020 --dbpath "C:\data\shard1" --replSet "rsShard1"*

*mongod --shardsvr --port 27021 --dbpath "C:\data\shard2" --replSet "rsShard2"*

*mongod --shardsvr --port 27022 --dbpath "C:\data\shard3" --replSet "rsShard3"*

1. **Iniciar mongosh y Configurar los Replica Sets**:

Es necesario iniciar mongosh.exe desde varias terminales porque toca inicializar cada Shard en una instancia de mongosh distinta.

*mongosh --port 27017*

*mongosh --port 27020*

*mongosh --port 27021*

*mongosh --port 27022*

**Configurar Config Server Replica Set**:

Esto se ejecuta en

*rs.initiate(*

*{*

*\_id: "rsConfig",*

*configsvr: true,*

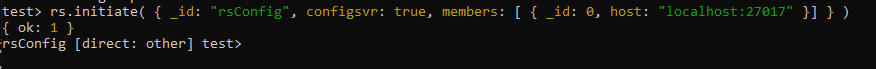
*members: [*

*{ \_id: 0, host: "localhost:27017" }*

*]*

*}*

*)*



**Configurar Shard Replica Sets**:

*rs.initiate(*

*{*

*\_id: "rsShard1",*

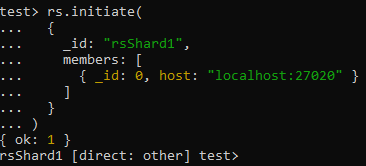
*members: [*

*{ \_id: 0, host: "localhost:27020" }*

*]*

*}*

*)*



*rs.initiate(*

*{*

*\_id: "rsShard2",*

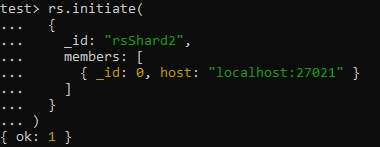
*members: [*

*{ \_id: 0, host: "localhost:27021" }*

*]*

*}*

*)*



*rs.initiate(*

*{*

*\_id: "rsShard3",*

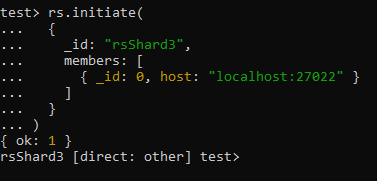
*members: [*

*{ \_id: 0, host: "localhost:27022" }*

*]*

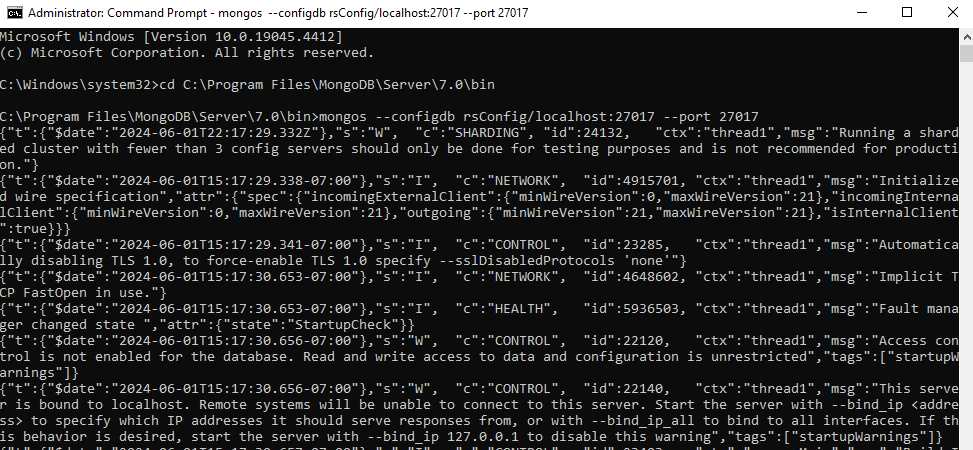
*}*

*)*



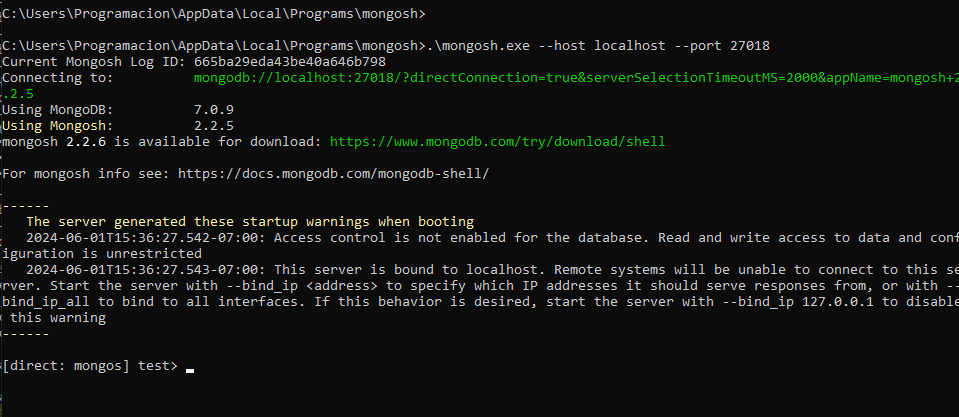
1. **Configurar el Router (mongos)**: **Iniciar el Router**:

*mongos --configdb rsConfig/localhost:27017 --port 27018*



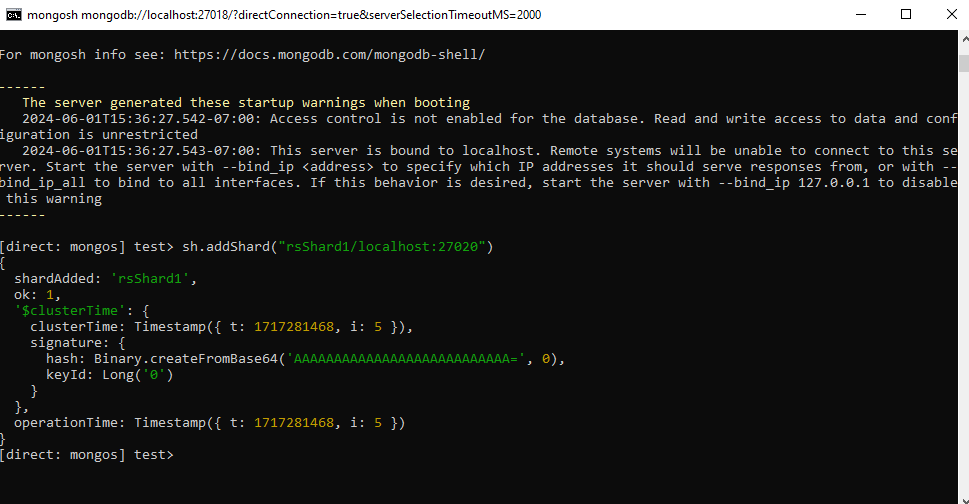
**Conectar al Router**:

*mongosh --host localhost --port 27018*

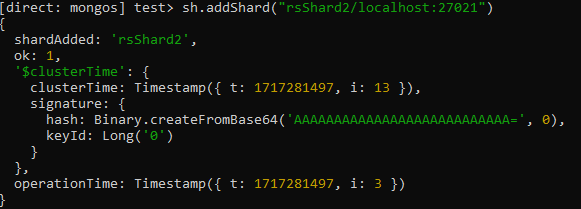


**Agregar los Shards**:

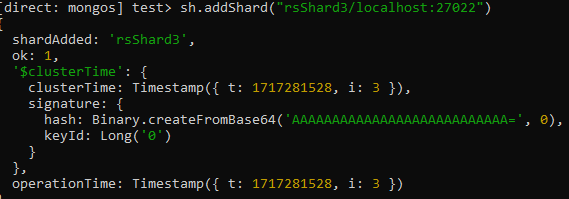
*sh.addShard("rsShard1/localhost:27020")*



*sh.addShard("rsShard2/localhost:27021")*

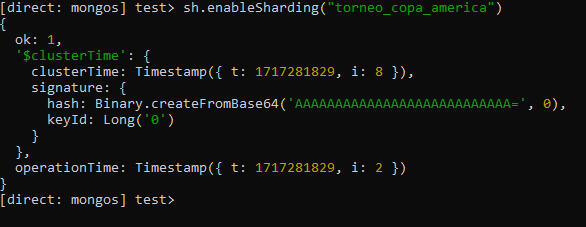


*sh.addShard("rsShard3/localhost:27022")*



1. **Habilitar Sharding en la Base de Datos y Colecciones**:

*sh.enableSharding("torneo\_copa\_america")*

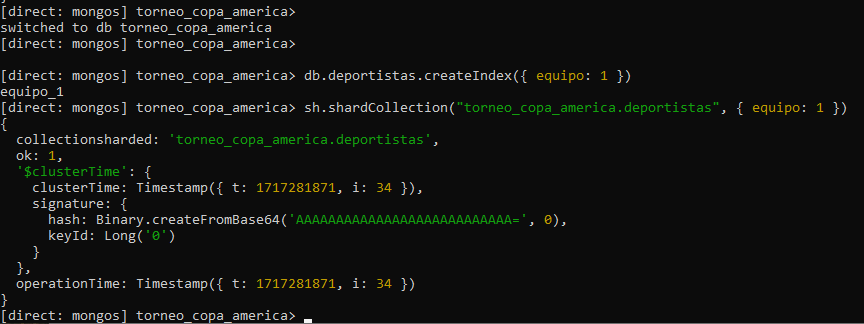


**Crear un Índice y Habilitar Sharding en la Colección**:

*use torneo\_copa\_america*

*db.deportistas.createIndex({ equipo: 1 })*

*sh.shardCollection("torneo\_copa\_america.deportistas", { equipo: 1 })*



**Verificación del Particionamiento**

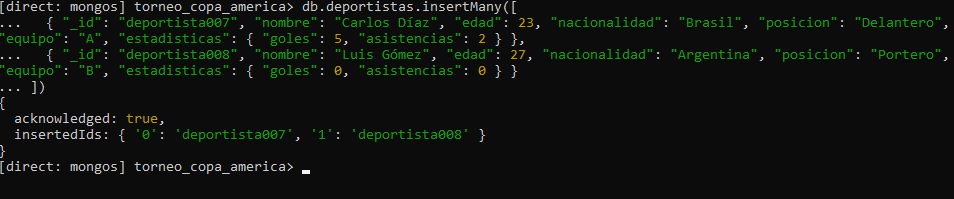
1. **Insertar Datos de Prueba**:

*db.deportistas.insertMany([*

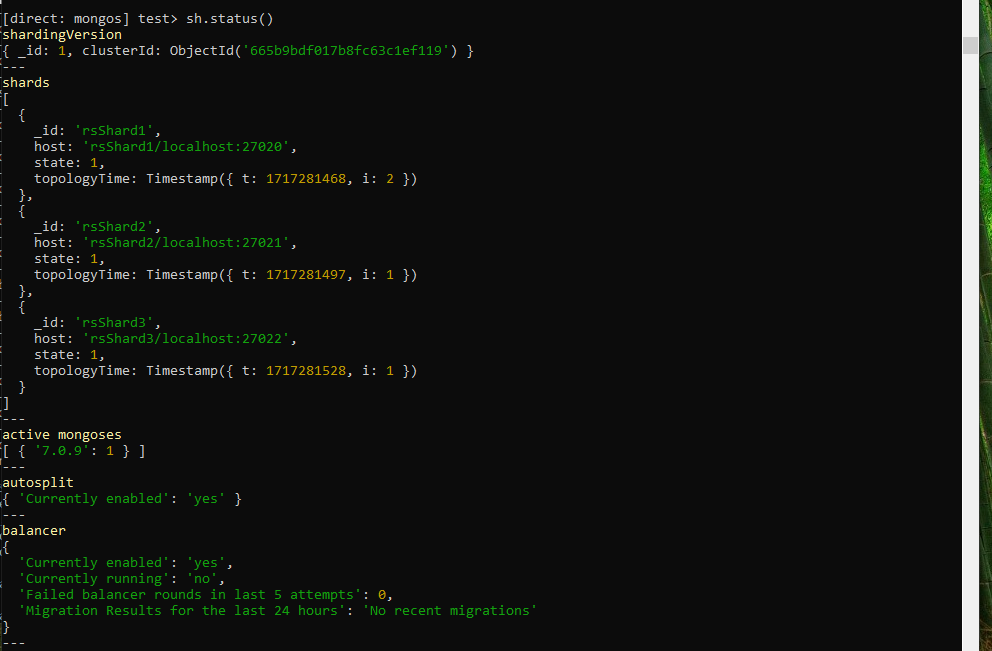
*{ "\_id": "deportista007", "nombre": "Carlos Díaz", "edad": 23, "nacionalidad": "Brasil", "posicion": "Delantero", "equipo": "A", "estadisticas": { "goles": 5, "asistencias": 2 } },*

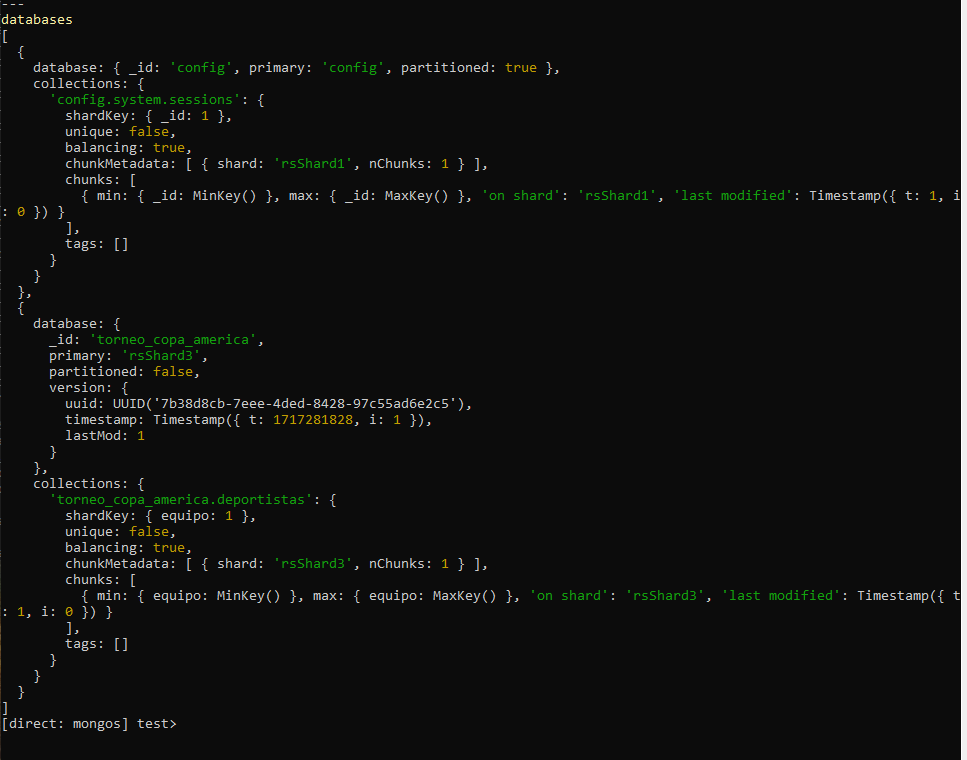
*{ "\_id": "deportista008", "nombre": "Luis Gómez", "edad": 27, "nacionalidad": "Argentina", "posicion": "Portero", "equipo": "B", "estadisticas": { "goles": 0, "asistencias": 0 } }*

*])*



1. **Verificar el status del Sharding**





1. **Verificar la Distribución de Datos en los Shards**: **Conectar a los Shards y Comprobar los Datos**:

*shard3 = new Mongo("localhost:27022").getDB("torneo\_copa\_america")*

*shard3.deportistas.find()*

