

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
```

2. 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
```

3. 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"
```

4. 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" }
  ]
}
)

```

```

test> rs.initiate( { _id: "rsConfig", configsvr: true, members: [ { _id: 0, host: "localhost:27017" } ] } )
{ ok: 1 }
rsConfig [direct: other] test>

```

Configurar Shard Replica Sets:

```

rs.initiate(
{
  _id: "rsShard1",
  members: [
    { _id: 0, host: "localhost:27020" }
  ]
}
)

```

```

test> rs.initiate(
... {
...   _id: "rsShard1",
...   members: [
...     { _id: 0, host: "localhost:27020" }
...   ]
... }
... )
{ ok: 1 }
rsShard1 [direct: other] test>

```

```

rs.initiate(
{
  _id: "rsShard2",
  members: [

```

```

    { _id: 0, host: "localhost:27021" }
  ]
}
)

```

```

test> rs.initiate(
...   {
...     _id: "rsShard2",
...     members: [
...       { _id: 0, host: "localhost:27021" }
...     ]
...   }
... )
{ ok: 1 }

```

```

rs.initiate(
{
  _id: "rsShard3",
  members: [
    { _id: 0, host: "localhost:27022" }
  ]
}
)

```

```

test> rs.initiate(
...   {
...     _id: "rsShard3",
...     members: [
...       { _id: 0, host: "localhost:27022" }
...     ]
...   }
... )
{ ok: 1 }
rsShard3 [direct: other] test>

```

5. Configurar el Router (mongos): Iniciar el Router:

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

```
Administrator Command Prompt - mongos --configdb rsConfig/localhost:27017 --port 27017
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\7.0\bin

C:\Program Files\MongoDB\Server\7.0\bin>mongos --configdb rsConfig/localhost:27017 --port 27017
{"t":{"sdate":"2024-06-01T15:17:29.332Z"},"s":"W", "c":"SHARDING", "id":24132, "ctx":"thread1", "msg":"Running a shard
ed cluster with fewer than 3 config servers should only be done for testing purposes and is not recommended for producti
on."}
{"t":{"sdate":"2024-06-01T15:17:29.338-07:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1", "msg":"Initializ
d wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingIntern
alClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":21,"maxWireVersion":21},"isInternalClient
":true}}}
{"t":{"sdate":"2024-06-01T15:17:29.341-07:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1", "msg":"Automatica
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"
{"t":{"sdate":"2024-06-01T15:17:30.653-07:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit T
CP FastOpen in use."}
{"t":{"sdate":"2024-06-01T15:17:30.653-07:00"},"s":"I", "c":"HEALTH", "id":5936503, "ctx":"thread1", "msg":"Fault mana
ger changed state ", "attr":{"state":"StartupCheck"}}
{"t":{"sdate":"2024-06-01T15:17:30.656-07:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"thread1", "msg":"Access con
trol is not enabled for the database. Read and write access to data and configuration is unrestricted", "tags":["startup
Warnings"]}
{"t":{"sdate":"2024-06-01T15:17:30.656-07:00"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"thread1", "msg":"This serve
r is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <addre
ss> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If th
is behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning.", "tags":["startupWarnings"]}
[direct: mongos] test>
```

Conectar al Router:

mongosh --host localhost --port 27018

```
C:\Users\Programacion\AppData\Local\Programs\mongosh>
C:\Users\Programacion\AppData\Local\Programs\mongosh>.\\mongosh.exe --host localhost --port 27018
Current Mongosh Log ID: 665ba29eda43be40a646b798
Connecting to:      mongodb://localhost:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+
2.5
Using MongoDB:      7.0.9
Using Mongosh:      2.2.5
mongosh 2.2.6 is available for download: https://www.mongodb.com/try/download/shell
For mongosh info see: https://docs.mongodb.com/mongosh-shell/

-----
The server generated these startup warnings when booting
2024-06-01T15:36:27.542-07:00: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
2024-06-01T15:36:27.543-07:00: This server is bound to localhost. Remote systems will be unable to connect to this se
rver. 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 disab
le this warning
-----

[direct: mongos] test>
```

Agregar los Shards:

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

```
mongosh mongodb://localhost:27018/?directConnection=true&serverSelectionTimeoutMS=2000

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

-----
The server generated these startup warnings when booting
2024-06-01T15:36:27.542-07:00: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
2024-06-01T15:36:27.543-07:00: This server is bound to localhost. Remote systems will be unable to connect to this se
rver. 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 disab
le this warning
-----

[direct: mongos] test> sh.addShard("rsShard1/localhost:27020")
{
  shardAdded: 'rsShard1',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1717281468, i: 5 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    },
  },
  operationTime: Timestamp({ t: 1717281468, i: 5 })
}
[direct: mongos] test>
```

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

```
[direct: mongos] test> sh.addShard("rsShard2/localhost:27021")
{
  shardAdded: 'rsShard2',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1717281497, i: 13 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1717281497, i: 3 })
}
```

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

```
[direct: mongos] test> sh.addShard("rsShard3/localhost:27022")
{
  shardAdded: 'rsShard3',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1717281528, i: 3 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1717281528, i: 3 })
}
```

6. Habilitar Sharding en la Base de Datos y Colecciones:

sh.enableSharding("torneo_copa_america")

```
[direct: mongos] test> sh.enableSharding("torneo_copa_america")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1717281829, i: 8 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1717281829, i: 2 })
}
[direct: mongos] test>
```

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 })
```

```
[direct: mongos] torneo_copa_america>
switched to db torneo_copa_america
[direct: mongos] torneo_copa_america>

[direct: mongos] torneo_copa_america> db.deportistas.createIndex({ equipo: 1 })
equipo_1
[direct: mongos] torneo_copa_america> sh.shardCollection("torneo_copa_america.deportistas", { equipo: 1 })
{
  collectionssharded: 'torneo_copa_america.deportistas',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1717281871, i: 34 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1717281871, i: 34 })
}
[direct: mongos] torneo_copa_america> _
```

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 } }
```

```
])
```

```
[direct: mongos] torneo_copa_america> 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 } }
... ])
{
  acknowledged: true,
  insertedIds: { '0': 'deportista007', '1': 'deportista008' }
}
[direct: mongos] torneo_copa_america> _
```

2. Verificar el status del Sharding


```
[direct: mongos] test> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('665b9bdf017b8fc63c1ef119') }
---
shards
[
  {
    _id: 'rsShard1',
    host: 'rsShard1/localhost:27020',
    state: 1,
    topologyTime: Timestamp({ t: 1717281468, i: 2 })
  },
  {
    _id: 'rsShard2',
    host: 'rsShard2/localhost:27021',
    state: 1,
    topologyTime: Timestamp({ t: 1717281497, i: 1 })
  },
  {
    _id: 'rsShard3',
    host: 'rsShard3/localhost:27022',
    state: 1,
    topologyTime: Timestamp({ t: 1717281528, i: 1 })
  }
]
---
active mongoses
[ { '7.0.9': 1 } ]
---
autosplit
{ 'Currently enabled': 'yes' }
---
balancer
{
  'Currently enabled': 'yes',
  'Currently running': 'no',
  'Failed balancer rounds in last 5 attempts': 0,
  'Migration Results for the last 24 hours': 'No recent migrations'
}
---
```

```
---
databases
[
  {
    database: { _id: 'config', primary: 'config', partitioned: true },
    collections: {
      'config.system.sessions': {
        shardKey: { _id: 1 },
        unique: false,
        balancing: true,
        chunkMetadata: [ { shard: 'rsShard1', nChunks: 1 } ],
        chunks: [
          { min: { _id: MinKey() }, max: { _id: MaxKey() }, 'on shard': 'rsShard1', 'last modified': Timestamp({ t: 1, i: 0 }) }
        ],
        tags: []
      }
    }
  },
  {
    database: {
      _id: 'torneo_copa_america',
      primary: 'rsShard3',
      partitioned: false,
      version: {
        uuid: UUID('7b38d8cb-7eee-4ded-8428-97c55ad6e2c5'),
        timestamp: Timestamp({ t: 1717281828, i: 1 }),
        lastMod: 1
      }
    },
    collections: {
      'torneo_copa_america.deportistas': {
        shardKey: { equipo: 1 },
        unique: false,
        balancing: true,
        chunkMetadata: [ { shard: 'rsShard3', nChunks: 1 } ],
        chunks: [
          { min: { equipo: MinKey() }, max: { equipo: MaxKey() }, 'on shard': 'rsShard3', 'last modified': Timestamp({ t: 1, i: 0 }) }
        ],
        tags: []
      }
    }
  }
]
[direct: mongos] test>
```

3. 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()
```

```
[direct: mongos] torneo_copa_america> shard3 = new Mongo("localhost:27022").getDB("torneo_copa_america")
torneo_copa_america
[direct: mongos] torneo_copa_america> shard3.deportistas.find()
torneo_copa_america
{
  _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 }
},
{
  _id: 'deportista009',
  nombre: 'Pedro Martínez',
  edad: 24,
  nacionalidad: 'Colombia',
  posicion: 'Defensa',
  equipo: 'C',
  estadisticas: { goles: 1, asistencias: 3 }
},
{
  _id: 'deportista010',
  nombre: 'Juan Pérez',
  edad: 26,
  nacionalidad: 'México',
  posicion: 'Mediocampista',
  equipo: 'D',
  estadisticas: { goles: 2, asistencias: 1 }
},
{
  _id: 'deportista011',
  nombre: 'Miguel Santos',
  edad: 28,
  nacionalidad: 'Uruguay',
  posicion: 'Delantero',
  equipo: 'E',
  estadisticas: { goles: 7, asistencias: 4 }
},
{
  _id: 'deportista012',
  nombre: 'Diego Rivera',
  edad: 22
```