

Documento de Casos de Prueba para Validar el Particionamiento en MongoDB

Objetivo

Validar que el particionamiento (Sharding) de la base de datos del torneo de fútbol Copa América funcione acorde al documento de requerimientos no funcionales planteado previamente.

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.

Casos de Prueba

Caso de Prueba 1: Verificación del Particionamiento de Datos

Planteamiento: Verificar que los datos se han distribuido entre los shards correctamente.

Ejecución:

1. Conectar a cada shard y verificar el conteo de documentos en la colección deportistas.

```
mongosh --port 27020
```

```
shard1DB = db.getSiblingDB('torneo_copa_america')
```

```
shard1DB.deportistas.count()
```

```
rsShard1 [direct: primary] test> shard1DB = db.getSiblingDB('torneo_copa_america')
torneo_copa_america
rsShard1 [direct: primary] test> shard1DB.deportistas.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
0
rsShard1 [direct: primary] test> █
```

```
mongosh --port 27021
```

```
shard2DB = db.getSiblingDB('torneo_copa_america')
```

```
shard2DB.deportistas.count()
```

```
rsShard2 [direct: primary] test> shard2DB = db.getSiblingDB('torneo_copa_america')
torneo_copa_america
rsShard2 [direct: primary] test> shard2DB.deportistas.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
0
rsShard2 [direct: primary] test> shard2DB.deportistas.count()
```

```
mongosh --port 27022
```

```
shard3DB = db.getSiblingDB('torneo_copa_america')
```

```
shard3DB.deportistas.count()
```

```
rsShard3 [direct: primary] test>
rsShard3 [direct: primary] test> shard3DB = db.getSiblingDB('torneo_copa_america')
torneo_copa_america
rsShard3 [direct: primary] test> shard3DB.deportistas.count()
6
```

Resultados Esperados: Los documentos en la colección deportistas deben estar distribuidos entre los tres shards de manera equilibrada.

Reporte de Resultados:

Los resultados muestran que los documentos están desbalanceados y que todos los documentos disponibles están en el Shard 3.

Análisis:

Para corregir este comportamiento, actualizaremos el chunk size a 1mb y agregaremos una cantidad considerable de deportistas.

```
[direct: mongos] config> db.settings.updateOne(
... { _id: "chunksize" },
... { $set: { _id: "chunksize", value: 1 } },
... { upsert: true }
... )
{
  acknowledged: true,
  insertedId: 'chunksize',
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
```

```
[direct: mongos] torneo_copa_america> for (let i = 10000; i < 1000000; i++) {
...   db.deportistas.insert({
...     _id: `deportista${i + 18}`,
...     nombre: `Nombre ${i}`,
...     edad: Math.floor(Math.random() * 40) + 20,
...     nacionalidad: `Pais ${i % 10}`,
...     posicion: `Posicion ${i % 5}`,
...     equipo: `Equipo ${i % 10}`,
...     estadisticas: {
...       goles: Math.floor(Math.random() * 20),
...       asistencias: Math.floor(Math.random() * 10)
...     }
...   });
... }
```

```
[direct: mongos] torneo_copa_america>
```

Iniciaremos el balanceador con

```
sh.startBalancer()
```

Ahora verificaremos de nuevo la cantidad de deportistas en cada Shard

```
rsShard1 [direct: primary] test> shard1DB.deportistas.count()
17847
rsShard1 [direct: primary] test> _
```

```
rsShard2 [direct: primary] test> shard2DB.deportistas.count()
17843
rsShard2 [direct: primary] test>
```

Podemos observar que los Shards ahora tienen repartidos los deportistas.

Caso de Prueba 2: Escalabilidad

Planteamiento: Validar que se pueden agregar nuevos shards sin afectar el rendimiento y la consistencia de los datos.

Ejecución:

Ejecución del Caso de Prueba:

1. Iniciar una nueva instancia de MongoDB para el nuevo shard.

```
Administrator: Command Prompt - mongod --shardsvr --port 27023 --dbpath "C:\data\shard4" --replSet "rsShard4"
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>mongod --shardsvr --port 27023 --dbpath "C:\data\shard4" --replSet "rsShard4"
{"t":{"$date":"2024-06-07T23:27:12.280-07:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialize
d wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingIntern
alClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient"
```

2. Configurar el nuevo Replica Set.

```
test> rs.initiate(
... {
...   _id: "rsShard4",
...   members: [
...     { _id: 0, host: "localhost:27023" }
...   ]
... }
...
rsShard4 [direct: other] test>
{ ok: 1 }
```

3. Agregar el nuevo shard al clúster de sharding.

```
[direct: mongos] test> sh.addShard("rsShard4/localhost:27023")
{
  shardAdded: 'rsShard4',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1717828147, i: 4 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1717828147, i: 4 })
}
```

4. Forzar el balanceo.

```
[direct: mongos] test> sh.startBalancer()
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1717828671, i: 2 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1717828671, i: 2 })
}
[direct: mongos] test>
```

5. Verificar la distribución de datos entre todos los shards.

Resultados Esperados: El nuevo Shard debe recibir una sección de los documentos de deportistas.

Reporte de Resultados:

```
sh.status()
```

```
collections: {
  torneo_copa_america.deportistas: {
    shardKey: { equipo: 1 },
    unique: false,
    balancing: true,
    chunkMetadata: [
      { shard: 'rsShard1', nChunks: 4 },
      { shard: 'rsShard2', nChunks: 3 },
      { shard: 'rsShard3', nChunks: 1 },
      { shard: 'rsShard4', nChunks: 1 }
    ],
    chunks: [
      { min: { equipo: MinKey() }, max: { equipo: 'Equipo 0' }, 'on shard': 'rsShard1', 'last modified': Timestamp({ t: 2, i: 0 }) },
      { min: { equipo: 'Equipo 0' }, max: { equipo: 'Equipo 1' }, 'on shard': 'rsShard2', 'last modified': Timestamp({ t: 3, i: 0 }) },
      { min: { equipo: 'Equipo 1' }, max: { equipo: 'Equipo 2' }, 'on shard': 'rsShard1', 'last modified': Timestamp({ t: 4, i: 0 }) },
      { min: { equipo: 'Equipo 2' }, max: { equipo: 'Equipo 3' }, 'on shard': 'rsShard2', 'last modified': Timestamp({ t: 5, i: 0 }) },
      { min: { equipo: 'Equipo 3' }, max: { equipo: 'Equipo 4' }, 'on shard': 'rsShard1', 'last modified': Timestamp({ t: 6, i: 0 }) },
      { min: { equipo: 'Equipo 4' }, max: { equipo: 'Equipo 5' }, 'on shard': 'rsShard2', 'last modified': Timestamp({ t: 7, i: 0 }) },
      { min: { equipo: 'Equipo 5' }, max: { equipo: 'Equipo 6' }, 'on shard': 'rsShard1', 'last modified': Timestamp({ t: 8, i: 0 }) },
      { min: { equipo: 'Equipo 6' }, max: { equipo: 'Equipo 7' }, 'on shard': 'rsShard4', 'last modified': Timestamp({ t: 9, i: 0 }) },
      { min: { equipo: 'Equipo 7' }, max: { equipo: MaxKey() }, 'on shard': 'rsShard3', 'last modified': Timestamp({ t: 9, i: 1 }) }
    ],
    tags: []
  }
}
```

```
Shard4DB = db.getSiblingDB('torneo_copa_america')
```

```
Shard4DB.deportistas.count()
```

```
rsShard4 [direct: other] test> shard4DB = db.getSiblingDB('torneo_copa_america')
torneo_copa_america
rsShard4 [direct: primary] test> shard4DB.deportistas.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
5947
rsShard4 [direct: primary] test>
```

Análisis:

El 4 Shard fue creado exitosamente y luego del balanceo, una porción de los documentos de los deportistas fueron agregados sin problemas.