

## **Actividad 3 - Conceptos y Comandos básicos del particionamiento en bases de datos NoSQL**

PRESENTADO AL PROFESOR:  
WILLIAM RUIZ.

PRESENTADO POR:  
GERMAN YOHANE JEREZ SIERRA

Diciembre 2023

Corporación Universitaria Iberoamericana  
Tumaco – Nariño  
Bases de datos avanzadas

## Introducción

El particionamiento, conocido como fragmentación o sharding, es esencial en las bases de datos NoSQL para distribuir conjuntos de datos complejos entre varios servidores, mejorando el rendimiento y la escalabilidad. Este método implica dividir los datos en fragmentos llamados shards y distribuirlos entre diferentes nodos o servidores. La clave de shard es crucial, ya que define cómo se segmentan los datos de manera equitativa y eficiente. Cada shard representa una parte específica de los datos, y es común tener múltiples réplicas para alta disponibilidad y tolerancia a fallos.

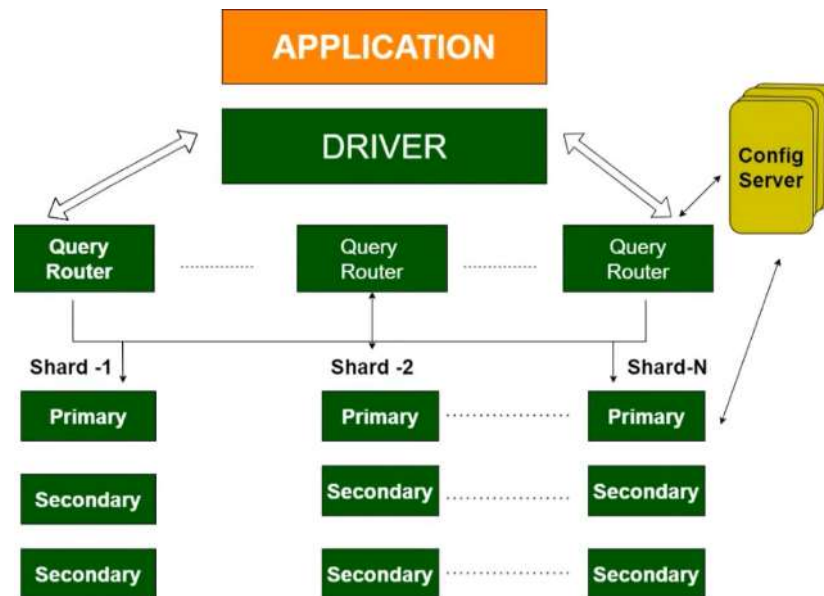
Los comandos básicos para el particionamiento en bases de datos NoSQL varían según la plataforma. Entre ellos están configurar servidores para gestionar metadatos (como en MongoDB), habilitar el sharding en una base de datos, definir la clave de shard, configurar colecciones o tablas para ser particionadas y añadir o eliminar nodos de shard para balancear la carga. Determinar el número de réplicas por shard es esencial para asegurar la redundancia y disponibilidad. Factores importantes incluyen la elección cuidadosa de la clave de shard para un particionamiento eficiente, monitorear y balancear la carga entre los shards para un óptimo rendimiento, y estrategias de respaldo para la integridad de los datos.

Estos conceptos y comandos proveen una base para comprender la distribución y gestión de datos a gran escala en entornos particionados, si bien su aplicación puede variar según la plataforma y tecnología específica de la base de datos utilizada.

## Tabla de Contenidos

Requerimientos no funcionales .....	4
Criterios para Implementar el Particionamiento .....	5
Desarrollar los comandos necesarios para el particionamiento horizontal .....	6
Diseño básico y funcionalidad .....	7
Link del video .....	11
Bibliografía .....	12

## Requerimientos no funcionales.



## Escalabilidad Horizontal

La base de datos "Mundial-2014" debe ser altamente escalable horizontalmente para afrontar eficazmente un crecimiento exponencial de datos y la creciente demanda de operaciones. Este crecimiento puede derivar de la adición de nuevos eventos deportivos, temporadas adicionales, y la inclusión de más detalles sobre equipos, jugadores y encuentros.

## Rendimiento

La capacidad de respuesta de la base de datos es crucial. Debe garantizar tiempos de consulta rápidos y consistentes, incluso con grandes volúmenes de datos, picos de demanda durante eventos importantes como partidos o la realización de consultas complejas que involucren múltiples colecciones.

## Disponibilidad

La base de datos debe permanecer disponible en todo momento para proporcionar acceso ininterrumpido a la información vital relacionada con el Mundial-2014. Se requiere una alta disponibilidad para garantizar que los usuarios puedan acceder a los datos de manera continua.

## Tolerancia a Fallos

Es crucial que la base de datos sea altamente resistente a fallos para preservar la integridad de los datos. La implementación de mecanismos de respaldo y recuperación debe

minimizar cualquier impacto en el acceso a los datos en caso de fallos en hardware o software.

## **Criterios de Calidad en el Desempeño**

### **El particionamiento se torna crítico cuando**

El tamaño total de datos supera la capacidad de un solo servidor, lo que ocasiona una disminución en el rendimiento de consultas y operaciones debido a los cuellos de botella. Existe un incremento notable en la cantidad de usuarios concurrentes o una demanda significativa de operaciones y consultas, ejerciendo una carga considerable en el sistema. Se identifica una distribución desigual de datos, donde ciertos equipos, jugadores o eventos tienen una cantidad desproporcionada de información asociada, lo que puede generar desequilibrios en la carga y afectar el rendimiento. Se espera un crecimiento continuo en la cantidad de datos debido a la inclusión de más temporadas, eventos deportivos adicionales o la ampliación de los detalles registrados.

### **Criterios para Implementar el Particionamiento**

- Escalabilidad, la estrategia de particionamiento debe admitir un escalado horizontal, permitiendo la adición de más nodos o shards a medida que aumenta la carga de datos y operaciones.
- Balanceo de Carga, el particionamiento debe distribuir de manera equitativa la carga de trabajo entre los shards para evitar la sobrecarga en servidores individuales y garantizar un rendimiento óptimo.
- Redundancia y Disponibilidad la implementación de shards y réplicas debe asegurar la redundancia de datos para mantener la disponibilidad en caso de fallos, ofreciendo la posibilidad de recuperación sin pérdida de información.
- Selección de Clave de Shard la elección adecuada del campo de shard es esencial para asegurar una distribución uniforme de datos entre los shards y para mejorar la eficiencia en la ejecución de consultas.
- Monitoreo y Mantenimiento herramientas de monitoreo robustas deben implementarse para supervisar continuamente el desempeño de los shards y tomar acciones correctivas según sea necesario, garantizando un sistema siempre optimizado.

**Desarrollar los comandos necesarios para el particionamiento horizontal.**

**Crear las carpetas de datos.**

```
mkdir P1_1 P1_2 P1_3 P2_1 P2_2 P2_3
```

**Crear carpetas de datos para los servidores de configuración.**

```
Mkdir cfg1 cfg2 cfg3
```

**Creación de los servidores de configuración**

```
mongod --configsvr --replSet configrs --dbpath cfg2 --port 26051 --logpath cfg2/log --logappend
```



```

Microsoft Windows [Versión 10.0.19045.3883]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jerez>mongod --configsvr --replSet configrs --dbpath cfg2 --port 26051 --logpath cfg2/log --logappend

```

```
mongod --configsvr --replSet configrs --dbpath cfg3 --port 26052 --logpath cfg3/log --logappend
```



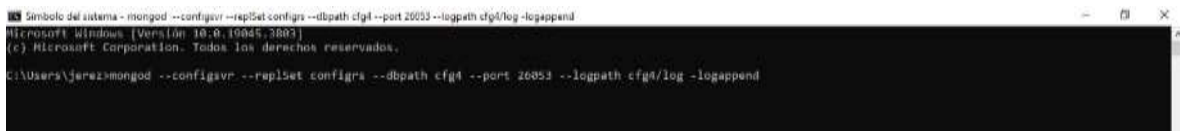
```

Microsoft Windows [Versión 10.0.19045.3883]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jerez>mongod --configsvr --replSet configrs --dbpath cfg3 --port 26052 --logpath cfg3/log --logappend

```

```
mongod --configsvr --replSet configrs --dbpath cfg4 --port 26053 --logpath cfg4/log --logappend
```



```

Microsoft Windows [Versión 10.0.19045.3883]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jerez>mongod --configsvr --replSet configrs --dbpath cfg4 --port 26053 --logpath cfg4/log --logappend

```

**Creamos los 3 servidores como miembros del mismo replica set...**

```
mongo -port 26051
```

```

conf = { _id: "configrs",
  members: [ { _id: 0, host: "127.0.0.1:26051" },
    { _id: 1, host: "127.0.0.1:26052" },
    { _id: 2, host: "127.0.0.1:26053" }, ] }

```

```

Simbolo del sistema - mongo -port 26051
Microsoft Windows [Versión 10.0.19045.3803]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jerez>mongo --port 26051
MongoDB shell version v4.4.26
connecting to: mongodb://127.0.0.1:26051/?compressors=disabled&gssapiServiceName=mongodb
implicit session: session { "id" : UUID("60e970d1-e0f4-4610-8551-d55f63119f50") }
MongoDB server version: 4.4.26

---
The server generated these startup warnings when booting:
  2023-12-16T08:57:56.750-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2023-12-16T08:57:56.750-05:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <addr>
  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 disable this warning
---
> conf = { _id: "configs",
...       members: [ { _id: 0, host: "127.0.0.1:26051" },
...                  { _id: 1, host: "127.0.0.1:26052" },
...                  { _id: 2, host: "127.0.0.1:26053" }, ] }
{
  "_id" : "configs",
  "members" : [
    {
      "_id" : 0,
      "host" : "127.0.0.1:26051"
    },
    {
      "_id" : 1,
      "host" : "127.0.0.1:26052"
    },
    {
      "_id" : 2,
      "host" : "127.0.0.1:26053"
    }
  ]
}

```

Para corroborar se inicializa

`rs.initiate(config)`

```

Simbolo del sistema - mongo -port 26051
> rs.initiate(config)
{
  "ok" : 1,
  "$gleStats" : {
    "lastOpTime" : Timestamp(1702738306, 1),
    "electionId" : ObjectId("00000000000000000000000000000000"),
    "lastCommittedOpTime" : Timestamp(0, 0)
  },
  "configs:SECONDARY" : 1
}

```

Como podemos verificar los tres servidores se encuentran asociados a un solo replica set

**Configuramos los servidores de sharding, creamos 2 conjuntos replica con 3 miembros cada uno.**

```

mongod --shardsvr --replSet P1 --dbpath "C:\data\replica\n1" --logpath "C:\data\replica\n1\log" --port 27100 --logappend --oplogSize 50
mongod --shardsvr --replSet P1 --dbpath "C:\data\replica\n2" --logpath "C:\data\replica\n2\log" --port 27101 --logappend --oplogSize 50
mongod --shardsvr --replSet P1 --dbpath "C:\data\replica\n3" --logpath "C:\data\replica\n3\log" --port 27102 --logappend --oplogSize 50
mongod --shardsvr --replSet P2 --dbpath "C:\data\replica\n4" --logpath "C:\data\replica\n4\log" --port 27200 --logappend --oplogSize 50
mongod --shardsvr --replSet P2 --dbpath "C:\data\replica\n5" --logpath "C:\data\replica\n5\log" --port 27201 --logappend --oplogSize 50
mongod --shardsvr --replSet P2 --dbpath "C:\data\replica\n6" --logpath "C:\data\replica\n6\log" --port 27202 --logappend --oplogSize 50

```

**mongod:** Es el ejecutable del servidor MongoDB.

**--shardsvr:** Especifica que este servidor será un servidor de shards, lo que significa que será parte de un conjunto de servidores que almacenan y manejan los datos distribuidos de una base de datos particionada en MongoDB (usado en configuraciones de sharding).

**--replSet P1:** Indica que este servidor será parte del conjunto de réplicas llamado "P1". Este conjunto de réplicas permitirá la replicación de datos entre los distintos servidores que formen parte de este conjunto, proporcionando redundancia y alta disponibilidad.

**--dbpath P1\_1:** Especifica la ruta del directorio donde se almacenarán los datos para este servidor de MongoDB. En este caso, los datos se almacenarán en la ruta "P1\_1".

**--logpath P1\_1/log:** Indica la ruta del archivo de registro donde se registrarán los mensajes y eventos del servidor. En este caso, se establece la ruta como "P1\_1/log".

**--port 27100:** Define el número de puerto en el que escuchará este servidor de MongoDB. En este caso, se ha configurado para escuchar en el puerto 27100.

**--logappend:** Esta opción indica al servidor que agregue nuevos registros al archivo de registro existente en lugar de sobrescribirlo si ya existe.

**--oplogSize 50:** Establece el tamaño máximo del oplog (log de operaciones) para este servidor de réplica. El oplog es una característica clave en la replicación de MongoDB que registra todas las operaciones de escritura realizadas en la base de datos.

The image displays six screenshots of Windows command prompts, each showing the configuration of a MongoDB replica set. The prompts are titled 'Símbolo del sistema - mongod' and show the following commands:

- First screenshot:** Configures replica set P1 with three members (P1\_1, P1\_2, P1\_3) on ports 27100, 27101, and 27102. The database path is P1\_1, the log path is P1\_1/log, and the oplog size is 50.
- Second screenshot:** Configures replica set P2 with three members (P2\_1, P2\_2, P2\_3) on ports 27200, 27201, and 27202. The database path is P2\_1, the log path is P2\_1/log, and the oplog size is 50.

Each screenshot shows the command being entered at the prompt 'C:\Users\jerez>' and the output of the command, which includes the MongoDB version (10.0.19045.3803) and the copyright notice for Microsoft Corporation.

## Integración de la base de datos.

use Mundial-2014



## Creación de los enrutadores de consulta o procesos mongos

```
mongos --configdb configrs/127.0.0.1:26051,127.0.0.1:26052,127.0.0.1:26053 --logappend --logpath mongos1
mongos --configdb configrs/127.0.0.1:26050,127.0.0.1:26051,127.0.0.1:26052 --logappend --logpath mongos1 --port 26601
```



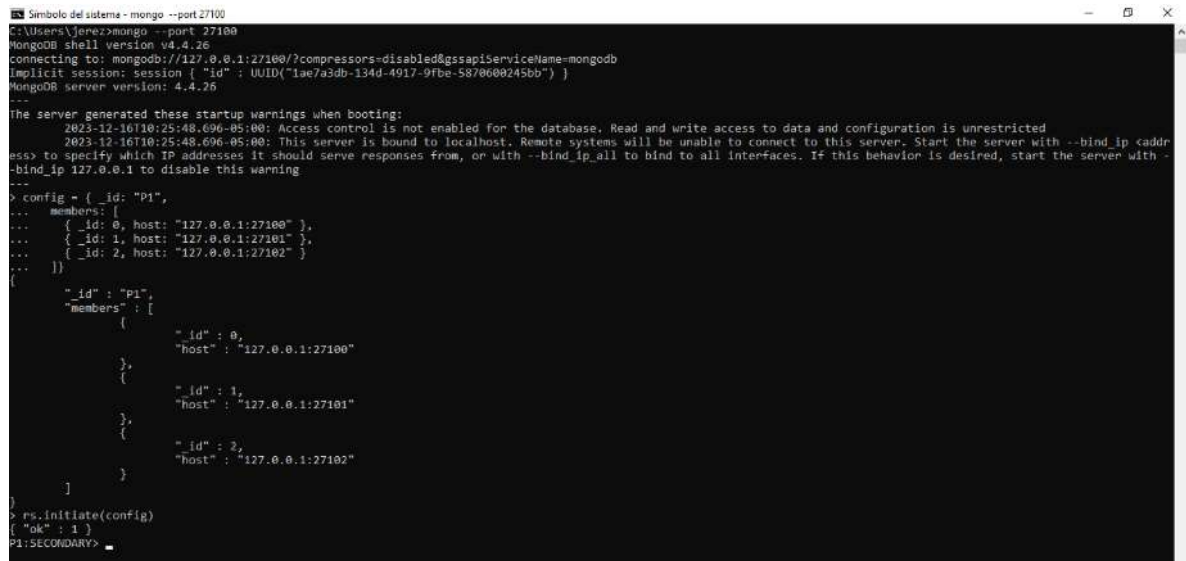
```
Simbolo del sistema - mongos --configdb configrs/127.0.0.1:26051,127.0.0.1:26052,127.0.0.1:26053 --logappend --logpath mongos1
Microsoft Windows [Versión 10.0.19045.3803]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\jerez>mongos --configdb configrs/127.0.0.1:26051,127.0.0.1:26052,127.0.0.1:26053 --logappend --logpath mongos1
```

## Activar los conjuntos replica

mongo --port 27100

```
config = { _id: "P1",
  members: [
    { _id: 0, host: "127.0.0.1:27100" },
    { _id: 1, host: "127.0.0.1:27101" },
    { _id: 2, host: "127.0.0.1:27102" }
  ]
}
```

```
rs.initiate(config)
rs.status()
```



```
Simbolo del sistema - mongo --port 27100
C:\Users\jerez>mongo --port 27100
MongoDB shell version v4.4.26
connecting to: mongodb://127.0.0.1:27100/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("1ae7a3db-134d-4917-9fbc-5870600245bb") }
MongoDB server version: 4.4.26

The server generated these startup warnings when booting:
2023-12-16T10:25:48.696-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2023-12-16T10:25:48.696-05:00: This server is bound to localhost. Remote systems will be unable to connect to this server. 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 disable this warning

> config = { _id: "P1",
...   members: [
...     { _id: 0, host: "127.0.0.1:27100" },
...     { _id: 1, host: "127.0.0.1:27101" },
...     { _id: 2, host: "127.0.0.1:27102" }
...   ]
... }
> rs.initiate(config)
{ "ok" : 1 }
P1:SECONDARY>
```

mongo --port 27200

```
config = { _id: "P2",
  members: [
    { _id: 0, host: "127.0.0.1:27200" },
    { _id: 1, host: "127.0.0.1:27201" },
    { _id: 2, host: "127.0.0.1:27202" }
  ]
}
```

```
rs.initiate(config)
rs.status()
```

```

Simbolo del sistema - mongo --port 27200
{
  "configTerm" : -1
},
{
  "_id" : 2,
  "name" : "127.0.0.1:27202",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 17,
  "optime" : {
    "ts" : Timestamp(1702744346, 5),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1702744346, 5),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2023-12-16T16:32:26Z"),
  "optimeDurableDate" : ISODate("2023-12-16T16:32:26Z"),
  "lastAppliedWallTime" : ISODate("2023-12-16T16:32:26.423Z"),
  "lastDurableWallTime" : ISODate("2023-12-16T16:32:26.423Z"),
  "lastHeartbeat" : ISODate("2023-12-16T16:32:32.387Z"),
  "lastHeartbeatRecv" : ISODate("2023-12-16T16:32:31.302Z"),
  "pingMs" : NumberLong(0),
  "lastHeartbeatMessage" : "",
  "syncSourceHost" : "127.0.0.1:27200",
  "syncSourceId" : 0,
  "infoMessage" : "",
  "configVersion" : 1,
  "configTerm" : -1
}
},
{
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1702744346, 5),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1702744346, 5)
}
P2:PRIMARY>

```

## Activar las particiones

```
sh.addShard("P1/127.0.0.1:27100")
```

```
sh.addShard("P2/127.0.0.1:27200")
```

```
sh.status()
```

```

C:\Windows\system32\cmd.exe - mongo
mongos> sh.addShard("p1/127.0.0.1:27100")
{
  "shardAdded" : "p1",
  "ok" : 1,
  "operationTime" : Timestamp(1623096931, 5),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1623096932, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}
mongos> sh.addShard("p2/127.0.0.1:27200")
{
  "shardAdded" : "p2",
  "ok" : 1,
  "operationTime" : Timestamp(1623096951, 4),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1623096952, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}
mongos>

```

**Link video** <https://youtu.be/FKLvcruULGU>

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

## **Bibliografía**

Sarasa, A. (2016). Introducción a las bases de datos NoSQL usando MongoDB. Editorial UOC. Capítulo VIII: Sharding.