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

# Bases de datos avanzadas

# Ingeniería de Software

# Instructor: Jorge Castañeda

# Presentado por: Leanira Valentina Hernandez Cristancho

# Diciembre 2024

**Descripción del Escenario:**

**Escenario: Sistema de Gestión de Torneos de Fútbol**

Imagina que tienes un sistema de gestión para torneos de fútbol. Este sistema maneja datos sobre partidos, equipos, jugadores y estadísticas, y debe ser capaz de procesar grandes cantidades de información, especialmente durante las temporadas de alta actividad. A medida que los datos aumentan, el sistema comienza a experimentar problemas de rendimiento debido a la limitación de un solo servidor.

Para resolver este problema, decidimos implementar **sharding** (particionamiento) en MongoDB, lo que permite distribuir los datos entre varios servidores y mejorar la escalabilidad del sistema.

El sistema maneja millones de registros, y el particionamiento es esencial para:

* **Distribuir los datos entre varios servidores**.
* **Optimizar las consultas** para mejorar el rendimiento.
* **Garantizar la alta disponibilidad** en caso de fallos en algún nodo.
* **Escalar fácilmente** añadiendo nuevos shards sin afectar la operación del sistema.

Los datos se distribuyen en diferentes shards basados en una clave de partición, como el nombre del torneo o la temporada de fútbol.

**Requerimientos No Funcionales:**

1. **Escalabilidad:**
   * El sistema debe ser capaz de manejar un aumento significativo en el volumen de datos sin afectar el rendimiento. A medida que el número de torneos y partidos crece, los datos se distribuyen en diferentes shards, lo que permite que el sistema escale horizontalmente.
2. **Alta Disponibilidad:**
   * Dado que los datos están distribuidos, la arquitectura debe garantizar que, en caso de un fallo de algún servidor, el sistema siga funcionando. Esto se logra mediante el uso de replica sets y particionamiento, que aseguran que los datos se sincronicen en varios nodos.
3. **Rendimiento:**
   * Las consultas deben ejecutarse de manera eficiente, incluso cuando el sistema maneja grandes volúmenes de datos. Gracias a los shards, las consultas pueden ser dirigidas a nodos específicos, lo que mejora la velocidad de acceso.
4. **Mantenimiento:**
   * El sistema debe permitir la adición de nuevos shards sin interrumpir el funcionamiento del clúster. Además, las operaciones de mantenimiento, como el reemplazo de hardware o la actualización de servidores, deben ser simples y sin impacto en la disponibilidad.
5. **Seguridad:**
   * La plataforma debe ser segura, garantizando que solo los usuarios autorizados puedan acceder a datos sensibles. Esto implica el uso de autenticación y autorización adecuadas, así como el cifrado de datos tanto en reposo como en tránsito.

**2. Documento con los Scripts de Particionamiento y los Resultados Obtenidos**

**1. Scripts para la Configuración de Particionamiento**

A continuación, se detallan los scripts necesarios para implementar el particionamiento en MongoDB.

**Iniciar Config Server Replica Set**

Primero, debes iniciar los **Config Servers** en tu máquina. Este paso es necesario para que MongoDB pueda manejar el sharding de manera efectiva.

mongod --configsvr --dbpath C:\data\configdb --port 27019 --bind\_ip localhost --replSet configReplicaSet

**Iniciar los Shards (Replica Sets)**

Cada shard debe ser configurado como un replica set. Inicia dos shards en los siguientes puertos.

**Shard 1:**

**mongod --shardsvr --dbpath C:\data\shard1 --port 27018 --bind\_ip localhost --replSet shardReplicaSet1**

**Shard 2:**

**mongod --shardsvr --dbpath C:\data\shard2 --port 27020 --bind\_ip localhost --replSet shardReplicaSet2**

**Iniciar el proceso mongos (Shard Router)**

Una vez que los shards estén en funcionamiento, debes iniciar **mongos**, que actuará como un router de consultas, dirigiéndolas a los shards adecuados.

**mongos --configdb configReplicaSet/localhost:27019 --port 27017**

**Conectar a mongos**

Conéctate al proceso mongos usando el siguiente comando:

**mongosh --host localhost --port 27017**

**Agregar Shards al Clúster**

Después de conectarte a mongos, puedes agregar los shards que has configurado a tu clúster de sharding.

sh.addShard("shardReplicaSet1/localhost:27018") sh.addShard("shardReplicaSet2/localhost:27020")

**Habilitar el Sharding en la Base de Datos**

Para habilitar el sharding en la base de datos **torneo\_futboll**, ejecuta el siguiente comando en el shell de mongosh:

use torneo\_futboll

sh.enableSharding("torneo\_futboll")

**2. Resultados Obtenidos**

* **Config Servers**: El servidor de configuración debe estar en ejecución en el puerto **27019**.
* **Shards**: Los shards deben estar en los puertos **27018** y **27020**, y ambos deben estar configurados como replica sets.
* **mongos**: El proceso mongos debe estar en funcionamiento en el puerto **27017** y conectado a los servers de configuración.
* **Resultado de sh.addShard**: Después de ejecutar sh.addShard, MongoDB confirmará que los shards han sido añadidos exitosamente al clúster.
* **Base de Datos Habilitada para Sharding**: El comando sh.enableSharding("torneo\_futboll") debe habilitar el sharding para la base de datos **torneo\_futboll**.

Texto

Descripción generada automáticamente

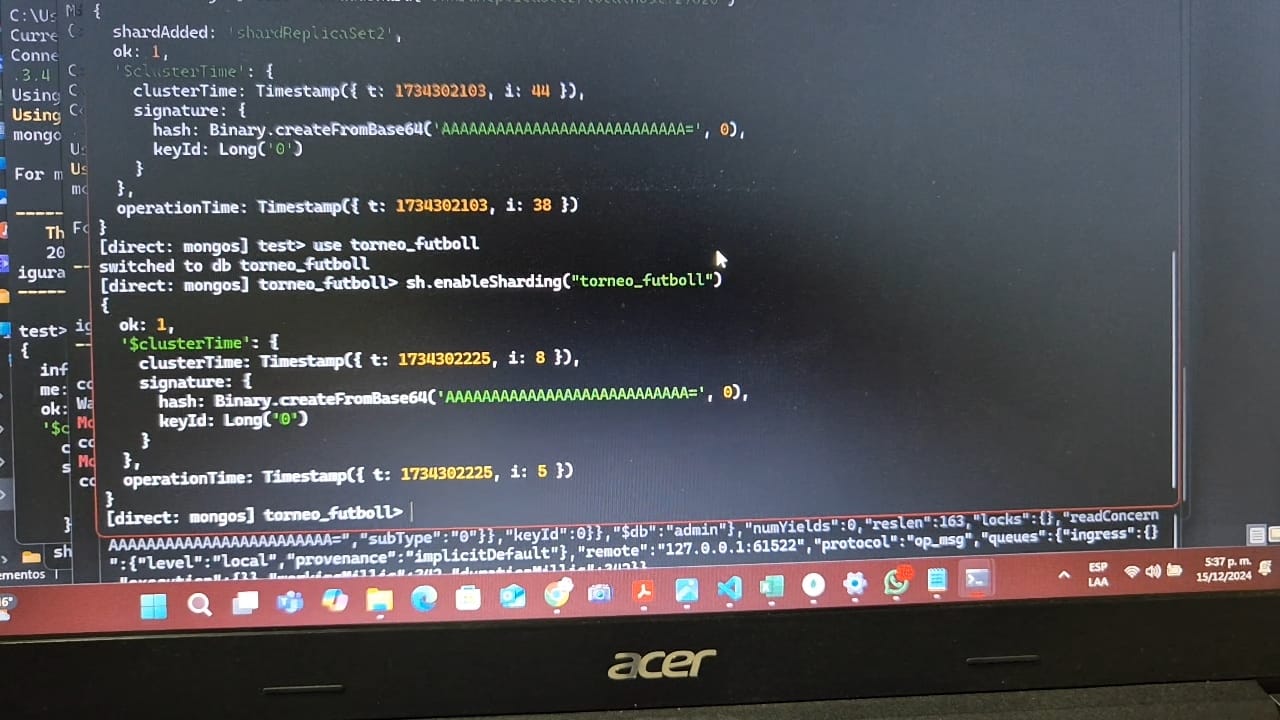


Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

**Link del repositorio git**

https://github.com/Valentina1296/Actividad-3.git

**Link del video**

https://drive.google.com/file/d/17\_spfJkAfNWspLGmtNi0oL3O6SY8jP3E/view?usp=sharing