# Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

# Bases de datos avanzadas

# Ingeniería de Software

# Instructor: Jorge Castañeda

# Presentado por: Leanira Valentina Hernandez Cristancho

# Diciembre 2024

**Documento con la Descripción del Escenario y los Requerimientos No Funcionales**

**Descripción del Escenario:**

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

Imagina un sistema de gestión de torneos de fútbol que almacena información sobre equipos, jugadores, partidos y estadísticas. Durante los periodos de alta actividad, como finales de temporada o torneos internacionales, el sistema debe manejar un alto volumen de solicitudes y datos en tiempo real. Esto incluye:

* Consultas frecuentes de usuarios para revisar resultados y clasificaciones.
* Actualizaciones constantes de datos sobre partidos en vivo.
* Generación de informes de rendimiento para equipos y jugadores.

A medida que el sistema crece, la carga en el servidor aumenta significativamente, lo que genera cuellos de botella en el rendimiento. Para resolver este problema, se decide implementar **particionamiento (sharding)** en MongoDB, distribuyendo los datos entre varios nodos para garantizar la escalabilidad y la alta disponibilidad del sistema.

**Requerimientos No Funcionales:**

1. **Escalabilidad:**
   * El sistema debe manejar el crecimiento continuo de datos y usuarios sin afectar el rendimiento. El particionamiento permite distribuir los datos en diferentes shards y escalar horizontalmente.
2. **Alta Disponibilidad:**
   * En caso de fallos en un nodo, el sistema debe seguir funcionando sin interrupciones. Esto se logra mediante la configuración de replica sets en los shards.
3. **Rendimiento:**
   * Las consultas deben ejecutarse rápidamente, incluso con grandes volúmenes de datos. El sharding optimiza las consultas, dirigiéndolas solo a los shards relevantes.
4. **Mantenimiento:**
   * El sistema debe permitir la adición de nuevos shards sin interrumpir las operaciones. Esto asegura que el sistema pueda adaptarse rápidamente a las necesidades futuras.
5. **Seguridad:**
   * Se deben implementar mecanismos de autenticación y autorización para proteger los datos sensibles, así como cifrado en tránsito y en reposo.

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

**1. Scripts para el Particionamiento en MongoDB**

**Configurar el Config Server Replica Set:**

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

**Configurar los Shards:**

**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:**

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

**Conectar a mongos:**

mongosh --host localhost --port 27017

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

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

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

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

use torneo\_futboll

sh.enableSharding("torneo\_futboll")

**2. Resultados Obtenidos**

1. **Config Server Activo:**
   * El Config Server se ejecuta correctamente en el puerto 27019 y está listo para manejar el metadata del sharding.
2. **Shards Configurados:**
   * Los shards están activos en los puertos 27018 y 27020, y cada uno está configurado como un replica set.
3. **Shard Router (mongos) en Ejecución:**
   * El proceso mongos está funcionando en el puerto 27017, dirigiendo las consultas a los shards correspondientes.
4. **Base de Datos Habilitada para Sharding:**
   * La base de datos **torneo\_futboll** está habilitada para sharding, lo que permite distribuir los datos eficientemente entre los shards.

**LINK REPOSITORIO GIT:**

https://github.com/Valentina1296/Actividad-4.git

**LINK VIDEO:**

https://drive.google.com/file/d/17dq6MDSlud9ocb8RlMNxS6ITIK4nTVIS/view?usp=sharing