

**Conceptos y Comandos básicos del particionamiento en bases de datos NoSQL.**

Actividad 3 – Unidad 3

**Nombre de los estudiantes**

Santiago Herrera Rocha

**Docente**

Jorge Isaac Castañeda Valbuena

CORPORACIÓN UNIVERSITARIA IBEROAMERICANA  
INGENIERÍA DE SOFTWARE, FACULTAD DE INGENIERÍA  
BASES DE DATOS AVANZADAS

Bogotá, D.C.

15 de diciembre de 2024

## Requerimientos no funcionales (para el particionamiento)

### 1. Escenario para Requerir Particionamiento

Para el modelo del torneo de fútbol definido en la Actividad 1, el particionamiento horizontal (sharding) de la base de datos se hace necesario cuando las siguientes condiciones están presentes:

- El torneo se expande para incluir más de 100 equipos, varias temporadas, o registros detallados de cada jugador y encuentro, donde se alcanzaría un volumen de datos significativo que podría superar la capacidad de un único nodo.
- Hay consultas frecuentes sobre tablas de posiciones, encuentros específicos por rondas, o si estadísticas completas de jugadores se vuelven lentas porque toda la información se encuentra en un solo nodo.
- Los usuarios finales acceden desde ubicaciones geográficas diferentes. Ahí el particionamiento podría ayudar a mejorar los tiempos de respuesta distribuyendo los datos entre nodos en distintas ubicaciones.

Adicionalmente, la correcta implementación de un particionamiento sobre los datos traería algunos beneficios como:

- **Mejorar el Rendimiento:** Las consultas se ejecutan más rápidamente al limitarse a fragmentos específicos en lugar de escanear la base de datos completa.
- **Escalabilidad Horizontal:** Se permite agregar más nodos al clúster para soportar un volumen de datos en aumento.

- **Alta Disponibilidad:** Si un nodo falla, el sistema puede seguir operando, utilizando los datos de otros fragmentos.

## 2. Requerimientos no funcionales

### *Desempeño*

- **Tiempo de Respuesta:** Las consultas clave como las estadísticas por equipo o resultados por ronda, entre otras, deben ejecutarse en menos de 100 ms, incluso con 1 millón de documentos.
- **Eficiencia de Almacenamiento:** Los datos deben ser distribuidos equitativamente entre los nodos.

### *Disponibilidad*

El sistema debe garantizar acceso continuo a los datos, incluso si uno de los nodos experimenta fallos.

### *Escalabilidad*

La arquitectura debe soportar el crecimiento de datos de manera horizontal al agregar más nodos al clúster.

### *Seguridad*

El acceso a los datos debe estar controlado mediante autenticación y autorización en cada nodo del sistema.

