

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

Docente:

Ing. JORGE CASTAÑEDA

Presentado por:

WILIAN DEIBY CARRERA ANDRADE

ID: 100157745

Base de datos avanzadas  
Programa de Ingeniería de Software  
Universidad Iberoamericana  
Diciembre - 2024

### **Actividad 3: Implementación de Particionamiento Horizontal en MongoDB para un Evento Deportivo**

#### **Escenario del caso planteado**

El caso práctico se basa en la gestión de la información para un evento deportivo: la Copa América. La base de datos contiene información sobre equipos, jugadores, partidos, estadios, árbitros y otros datos relacionados. Dado el crecimiento de los datos y la necesidad de realizar consultas eficientes, se plantea el uso de particionamiento horizontal (sharding) para optimizar el rendimiento.

#### **Escenario para particionamiento:**

A medida que aumenta la cantidad de datos (por ejemplo, registros de partidos históricos o estadísticas de jugadores), las operaciones de lectura y escritura comienzan a degradarse.

La base de datos se distribuye en diferentes geografías para reducir la latencia y mejorar el acceso desde distintos países donde se realizan los partidos.

Se espera un alto volumen de usuarios consultando información en tiempo real durante el torneo.

#### **Criterio para particionamiento:**

Particionar por país del equipo anfitrión o por el año del torneo, ya que las consultas principales estarán relacionadas con estas categorías.

#### **Documento de requerimientos no funcionales**

Un sistema de base de datos NoSQL, como MongoDB, requiere cumplir ciertos criterios de calidad para garantizar un alto rendimiento, escalabilidad y confiabilidad en escenarios exigentes. Estos requerimientos son fundamentales para manejar grandes volúmenes de datos y responder a las demandas de acceso concurrente, especialmente en eventos deportivos de gran envergadura como la Copa América. A continuación, se detallan los requerimientos no funcionales para este caso:

#### **Requerimientos de calidad relacionados con el desempeño:**

##### **Escalabilidad:**

La base de datos debe escalar horizontalmente para manejar un volumen creciente de datos y consultas simultáneas.

##### **Disponibilidad:**

El sistema debe garantizar alta disponibilidad incluso si un fragmento (shard) falla.

**Tiempo de respuesta:**

Las consultas más frecuentes deben responder en menos de 500 ms.

**Balance de carga:**

Los datos deben estar distribuidos uniformemente entre los fragmentos para evitar cuellos de botella.

**Resiliencia:**

El sistema debe ser capaz de recuperarse rápidamente ante la pérdida de un nodo o fragmento.

**Mantenimiento:**

La arquitectura debe permitir agregar o remover shards sin interrumpir el servicio.

**Latencia:**

La replicación entre los shards debe ser eficiente para minimizar la latencia durante la sincronización.

**Criterios para particionamiento:**

**Clave de partición:** La clave será el campo `torneo_id`, que representa el año del torneo.

**Estrategia:** Hashing para distribuir los datos uniformemente entre los fragmentos.

---

**Comandos necesarios para el particionamiento horizontal****Configuración del entorno de sharding****1. Iniciar el servidor de configuración:**

```
mongod --configsvr --replSet configReplSet --port 27019 --dbpath /data/configdb --bind_ip localhost
```

**2. Iniciar el conjunto de réplicas de configuración:**

```
rs.initiate({  
  _id: "configReplSet",  
  configsvr: true,  
  members: [  
    { _id: 0, host: "localhost:27019" }  
  ]  
})
```

```
]
});
```

### 3. Iniciar los servidores shard:

```
mongod --shardsvr --replSet shardReplSet1 --port 27018 --dbpath /data/shard1 --bind_ip localhost
```

```
mongod --shardsvr --replSet shardReplSet2 --port 27017 --dbpath /data/shard2 --bind_ip localhost
```

### 4. Iniciar los conjuntos de réplicas para los shards:

```
rs.initiate({
  _id: "shardReplSet1",
  members: [
    { _id: 0, host: "localhost:27018" }
  ]
});
```

```
rs.initiate({
  _id: "shardReplSet2",
  members: [
    { _id: 0, host: "localhost:27017" }
  ]
});
```

### 5. Iniciar el mongod:

```
mongos --configdb configReplSet/localhost:27019 --bind_ip localhost --port 27020
```

### 6. Conectar el cliente mongo al mongos:

```
mongo --port 27020
```

## Habilitar sharding y configurar particionamiento

### 1. Habilitar sharding para la base de datos:

```
sh.enableSharding("copa_america");
```

### 2. Configurar el índice para la clave de partición:

```
db.equipos.createIndex({ "torneo_id": "hashed" });
```

### 3. **Shardear la colección:**

```
sh.shardCollection("copa_america.equipos", { "torneo_id": "hashed" })
```