# Actividad 2 - Conceptos y comandos básicos de la replicación en bases de datos NoSQL

# Bases de datos avanzadas

# Instructor: Jorge Castañeda

# Estudiante: Leanira Valentina Hernandez Cristancho

# Diciembre 2024

# Configuración y Pruebas de Replicación en MongoDB

## Introducción

Este documento describe la configuración de un conjunto de réplicas en MongoDB para la base de datos 'torneo\_futboll', con el objetivo de garantizar la redundancia y disponibilidad 24x7. Además, se detallan los casos de prueba realizados para verificar el correcto funcionamiento del sistema.

## Requerimientos No Funcionales

1. Redundancia:

- Configuración de tres nodos para garantizar la replicación de datos.

- En caso de fallo de un nodo, el sistema debe continuar funcionando sin interrupciones.

2. Disponibilidad 24x7:

- Tiempo de recuperación ante fallos: máximo 5 segundos.

- Acceso continuo a los datos durante mantenimientos.

3. Consistencia:

- Garantizar la sincronización en tiempo real entre los nodos del Replica Set.

4. Escalabilidad:

- Permitir la adición de nuevos nodos al sistema sin afectar el rendimiento.

## Estrategia de Replicación

Para cumplir con los requerimientos, se implementó un conjunto de réplicas (Replica Set) en MongoDB con la siguiente configuración:

1. Un nodo primario: Responsable de todas las operaciones de escritura.

2. Dos nodos secundarios: Responsables de replicar los datos y permitir consultas en caso de fallo.

Pasos para configurar el Replica Set:

1. Crear carpetas separadas para los nodos: nodo1, nodo2, nodo3.

2. Iniciar tres instancias de MongoDB con la opción --replSet habilitada.

3. Inicializar el Replica Set desde el nodo primario con los nodos configurados.

## Casos de Prueba

\*\*Caso de Prueba 1: Sincronización de Datos\*\*

- Objetivo: Verificar que los datos insertados en el nodo primario se replican correctamente en los secundarios.

- Pasos:

1. Insertar un documento en el nodo primario.

2. Consultar el documento en uno de los nodos secundarios.

- Criterio de Éxito: El documento debe estar disponible en el nodo secundario.

\*\*Caso de Prueba 2: Failover Automático\*\*

- Objetivo: Verificar que, al detener el nodo primario, uno de los secundarios asume el rol de primario.

- Pasos:

1. Detener el nodo primario.

2. Verificar que uno de los secundarios es promovido a primario.

- Criterio de Éxito: El failover debe completarse automáticamente en menos de 5 segundos.

\*\*Caso de Prueba 3: Disponibilidad 24x7\*\*

- Objetivo: Asegurar que las consultas siguen funcionando durante un failover.

- Pasos:

1. Realizar consultas continuas en el nodo primario.

2. Durante las consultas, detener el nodo primario.

3. Verificar que las consultas son atendidas por el nuevo primario.

- Criterio de Éxito: Las consultas no deben fallar.

## Reporte de Resultados

\*\*Resultados de los Casos de Prueba:\*\*

|  |  |  |
| --- | --- | --- |
| Caso de Prueba | Resultado | Observaciones |
| Sincronización de Datos | Éxito | Los datos fueron replicados correctamente. |
| Failover Automático | Éxito | El failover se completó en 3 segundos. |
| Disponibilidad 24x7 | Éxito | Las consultas continuaron funcionando sin interrupciones. |

## Conclusión

El sistema de replicación en MongoDB para la base de datos 'torneo\_futboll' garantiza alta disponibilidad, redundancia y consistencia de los datos. Las pruebas realizadas confirmaron que los datos se replican correctamente entre los nodos, y que el sistema puede manejar fallos sin interrupciones.

# LINK VIDEO: https://drive.google.com/file/d/13rW4tefsh3yejlFKH-m94G91XkxdU8E1/view?usp=sharing