2023

# Bases de datos avanzadas NO-SQL

ISABEL DELGADO CORREAL, LIZETH PAOLA BUITRAGO QUINTERO,
ISMAEL ALEXANDER CARVAJAL GONZALEZ

IBEROAMERICANO

**Replicas para mongo**

1. **Objetivo**
   El objetivo de este documento es establecer los criterios de calidad en cuanto a la redundancia y disponibilidad 24x7 para el sistema de gestión de participantes en un torneo deportivo. Estos criterios garantizarán un funcionamiento confiable y continuo del sistema, minimizando los tiempos de inactividad y asegurando la integridad de los datos en todo momento.

2. **Criterios de Redundancia**
   - Los datos del sistema deben estar respaldados y almacenados en múltiples ubicaciones para garantizar la redundancia.
   - Se implementará una estrategia de replicación en conjunto de réplicas con al menos 3 nodos para asegurar la disponibilidad de los datos.
   - La replicación debe ser asíncrona para minimizar la latencia en la propagación de los cambios entre los nodos.
   - Se deben realizar copias de seguridad periódicas de los datos replicados para facilitar la recuperación ante posibles fallas.

3. **Criterios de Disponibilidad 24x7**
   - El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, sin interrupciones planificadas para mantenimientos o actualizaciones.
   - Se debe implementar un monitoreo constante del sistema para detectar y resolver de manera proactiva cualquier incidencia que pueda afectar la disponibilidad.
   - Se deben establecer procedimientos de recuperación ante desastres para minimizar el tiempo de inactividad en caso de fallos del sistema o eventos imprevistos.
   - La arquitectura del sistema debe ser escalable y capaz de gestionar cargas de trabajo variables sin comprometer la disponibilidad.

4. **Consideraciones adicionales**
   - Se debe establecer un plan de contingencia para hacer frente a posibles interrupciones del servicio y garantizar una rápida recuperación.
   - Se debe realizar un monitoreo constante de los recursos del sistema, como el rendimiento de los servidores, el consumo de almacenamiento y la utilización de la red, para prevenir posibles cuellos de botella y mantener un rendimiento óptimo.

     Estos criterios de calidad en cuanto a la redundancia y disponibilidad 24x7 serán la base para el diseño e implementación del sistema de gestión de participantes en el torneo deportivo, asegurando que los datos estén protegidos, disponibles y accesibles en todo momento.

**Creación del Replicaset**

- MiejReplicaSet = new ReplSetTest ({name: "MireplicaSet", nodes: 3})
- MiejReplicaSet = new ReplSetTest ({name: "MireplicaSet", nodes: 3}); print("hecho")

**Arrancar los procesos mongod de la replica**

- MiejReplicaSet.startSet()

**Arrancar el proceso de replica**

- MiejReplicaSet.initiate()

**Prueba del grupo de replica**

- conn=new Mongo("Metal2022:20003")
- testDB=conn.getDB("ENTRENADORES")
- testDB.isMaster()


- testDB.ENTRENADORES.count()

**Comprobación de la réplica sobre los nodos secundarios**

- connSecondary = new Mongo("LAPTOP-0C1A1M9O:20007 ")
- secondaryTestDB = connSecondary.getDB("Entrenadores")
- secondaryTestDB.isMaster()
- secondaryTestDB.Entrenadores.count();
- secondaryTestDB.Entrenadores.findOne()

**Detener el nodo primario**

- connPrimary = new Mongo("localhost:20003")
- primaryDB = connPrimary.getDB("Entrenadores")
- primaryDB.isMaster()

**Comprobación del nuevo nodo primario**

- connNewPrimary = new Mongo("localhost:20003")
- newPrimaryDB = connNewPrimary.getDB("Biblioteca")
- newPrimaryDB.isMaster()

**Detener el ReplicaSet de pruebas**

- MiejReplicaSet.stopSet()

**Casos de pruebas en replicación bajo MongoDB**

1. **Caso de Prueba:** Verificar la creación exitosa de las colecciones.
    - **Descripción:** Se verifica que las colecciones se creen correctamente en la base de datos.
    - **Pasos:**
        1. Comprobar que las colecciones "Jugadores", "Entrenadores", "Árbitros", "Encuentros", "Resultados" y "TablaPosiciones" existan en la base de datos.
    - **Resultado Esperado:** Todas las colecciones deben existir en la base de datos.
2. **Caso de Prueba:** Verificar la creación exitosa de los índices.
    - **Descripción:** Se verifica que los índices se creen correctamente en las colecciones correspondientes.
    - **Pasos:**
        1. Comprobar que los índices "equipo" existan en las colecciones "Jugadores", "Entrenadores" y "TablaPosiciones".
    - **Resultado Esperado:** Los índices "equipo" deben existir en las colecciones correspondientes.
3. **Caso de Prueba:** Verificar la inserción de jugadores.
    - **Descripción:** Se verifica que los jugadores se inserten correctamente en la colección "Jugadores".
    - **Pasos:**
        1. Insertar varios documentos de jugadores en la colección "Jugadores".
        2. Comprobar que los documentos se hayan insertado correctamente.
    - **Resultado Esperado:** Los documentos de jugadores deben estar presentes en la colección "Jugadores".
4. **Caso de Prueba:** Verificar la asignación de entrenadores a equipos.
    - **Descripción:** Se verifica que los entrenadores se asignen correctamente a los equipos en la colección "Entrenadores".
    - **Pasos:**
        1. Insertar varios documentos de entrenadores en la colección "Entrenadores" con el campo "equipo" adecuado.
        2. Comprobar que los documentos se hayan insertado correctamente y contengan el campo "equipo" correspondiente.

- **Resultado Esperado:** Los documentos de entrenadores deben estar presentes en la colección "Entrenadores" y contener el campo "equipo" correctamente asignado.

**Link: https://github.com/isabeldc13008/Bases_de_datos_avanzadas-NO-SQL**