




Documento de Requerimientos No Funcionales

Proyecto: Gestión de Torneos Deportivos con MongoDB



Introducción

- Este documento detalla los requerimientos no funcionales relacionados con la redundancia y disponibilidad 24x7 del sistema de gestión de torneos deportivos. El objetivo es garantizar la continuidad operativa del sistema y minimizar el impacto de fallos en los nodos.
- 



Requerimientos No Funcionales

➤ 2.1. Redundancia

➤ **Número de nodos:** El sistema contará con 3 nodos distribuidos en un replica set.

- Nodo 1: Primario (gestiona las operaciones de escritura).
- Nodo 2: Secundario (copia de seguridad en tiempo real).
- Nodo 3: Árbitro (participa en las elecciones para asegurar el quorum).

➤ **Tolerancia a fallos:**

- El sistema debe mantenerse operativo si falla un nodo.
- El nodo secundario debe sincronizarse automáticamente al reintegrarse al clúster.



Requerimientos No Funcionales

➤ 2.2. Disponibilidad 24x7

➤ Failover automático:

- Si el nodo primario falla, uno de los nodos secundarios será promovido automáticamente a primario.

➤ Sincronización entre nodos:

- El retraso máximo permitido para la sincronización entre nodos es de 5 segundos.
- Los datos deben replicarse de manera confiable en todos los nodos.



Estrategia Técnica

- Se implementará un replica set en MongoDB para garantizar redundancia y disponibilidad. Los nodos estarán configurados de la siguiente manera:
- Nodo 1: MongoDB en el puerto 27017.
- Nodo 2: MongoDB en el puerto 27018.
- Nodo 3: MongoDB en el puerto 27019.
- El sistema será probado con simulaciones de fallos para validar los criterios planteados.



Configuración de los Nodos

- **Crea directorios para las bases de datos de los tres nodos:**
- `mkdir -p /data/db1 /data/db2 /data/db3`

Configuración de los Nodos

- **Inicia los servicios de MongoDB en tres terminales diferentes:**
- `mongod --replSet "rs0" --port 27017 --dbpath /data/db1 --bind_ip_all`
- `mongod --replSet "rs0" --port 27018 --dbpath /data/db2 --bind_ip_all`
- `mongod --replSet "rs0" --port 27019 --dbpath /data/db3 --bind_ip_all`

Configuración de los Nodos

➤ Iniciar el Replica Set

➤ Conéctate al nodo primario usando el cliente de MongoDB:

➤ `mongo --port 27017`

➤ Inicializa el replica set con la siguiente configuración:

```
rs.initiate({  
  _id: "rs0",  
  members: [  
    { _id: 0, host: "localhost:27017" },  
    { _id: 1, host: "localhost:27018" },  
    { _id: 2, host: "localhost:27019", arbiterOnly: true }  
  ]  
});
```




Configuración de los Nodos

- Verifica el estado del replica set:
- `rs.status();`



Prueba Básica de Replicación

- Conéctate a uno de los nodos secundarios:
- **mongo --port 27018**
- Intenta escribir un documento en la base de datos(Debería fallar, ya que solo el nodo primario acepta escrituras.):
- **use test;**
- **db.collection.insertOne({ mensaje: "Prueba de escritura" });**
- Conéctate nuevamente al nodo primario (puerto27017) y realiza escritura
- **db.collection.insertOne({ mensaje: "Prueba de replicación" });**
- Verifica si el documento se replicó en el nodo secundario 27018
- **db.collection.find();**