ACTIVIDAD 2 - CONCEPTOS Y COMANDOS BÁSICOS DE LA REPLICACIÓN EN BASES DE DATOS NOSQL

# Documento de requerimientos no funcionales para torneo de deportivo de futbol

## Objetivo

Este documento tiene como objetivo definir los requerimientos no funcionales relacionados con la redundancia y la disponibilidad 24x7 para el torneo de futbol.

## Redundancia

El sistema debe tener una alta redundancia para garantizar que el evento se lleve a cabo sin interrupciones. Para lograr esto, se deben cumplir los siguientes requerimientos no funcionales:

• Disponibilidad de múltiples servidores: El sistema debe contar con múltiples servidores que puedan manejar el tráfico simultáneo de usuarios, y que estén configurados para proporcionar una redundancia adecuada. Los servidores deben estar ubicados en diferentes ubicaciones geográficas para garantizar la disponibilidad continua.

• Respaldos frecuentes: Se deben realizar respaldos frecuentes de la información del sistema en diferentes ubicaciones geográficas. Los respaldos deben ser almacenados en un lugar seguro y accesible en caso de una falla en el sistema principal.

• Pruebas de continuidad del negocio: El sistema debe contar con pruebas regulares de continuidad del negocio para garantizar que, en caso de una interrupción del servicio, se pueda restaurar el sistema en un tiempo razonable.

## Disponibilidad 24x7

El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, para garantizar la satisfacción del usuario y el éxito del torneo deportivo de fútbol. Para lograr esto, se deben cumplir los siguientes requerimientos no funcionales:

1. **Tiempo de actividad del sistema:** El sistema debe estar en funcionamiento continuo durante todo el evento, sin interrupciones, y estar disponible en todo momento para los usuarios.
2. **Monitoreo constante del sistema:** Se deben realizar monitoreos constantes del sistema para detectar cualquier problema que pueda afectar su disponibilidad, y solucionarlos en tiempo real.
3. **Personal de soporte técnico disponible:** Se debe contar con personal de soporte técnico disponible las 24 horas del día, los 7 días de la semana, para responder rápidamente a cualquier problema técnico que pueda surgir.

* El sistema debe contar con una infraestructura de alta disponibilidad y escalabilidad para soportar un alto volumen de tráfico.
* El sistema debe contar con un balanceador de carga para distribuir el tráfico de manera equilibrada entre los servidores.
* Los servidores deben estar configurados para permitir la actualización y el mantenimiento en vivo del sistema, sin interrupciones en el servicio.
* El sistema debe tener medidas de seguridad y recuperación de desastres para garantizar la continuidad del servicio en caso de incidentes.

Con estos requerimientos no funcionales, se garantiza una alta redundancia y disponibilidad 24x7 del sistema para el torneo deportivo de futbol.

## Conclusión:

Este documento de requerimientos no funcionales establece los criterios de calidad para garantizar la redundancia y disponibilidad 24x7 del sistema del torneo deportivo de fútbol. Los criterios establecidos aseguran la tolerancia a fallos, la consistencia de los datos y la disponibilidad del sistema, incluso en caso de fallos de los servidores. Además, se establecen medidas de seguridad y recuperación de desastres para garantizar la continuidad del servicio en todo momento.

Para replicar las bases de datos en MongoDB del torneo deportivo de fútbol en un mínimo de 3 nodos acorde al documento de requerimientos no funcionales presentado, se puede utilizar una estrategia de replicación de tipo "set de réplicas" o "replica set". A continuación, se muestran los comandos necesarios para crear un conjunto de réplicas con tres nodos:

1. Iniciar tres instancias de MongoDB en diferentes puertos. Por ejemplo:

mongod --replSet rs0 --port 27017

mongod --replSet rs0 --port 27018

mongod --replSet rs0 --port 27019

1. Conectarse a una de las instancias de MongoDB y configurar el conjunto de réplicas:

mongo --port 27017

> rs.initiate()

1. Agregar los otros dos nodos al conjunto de réplicas:

> rs.add("localhost:27018")

> rs.add("localhost:27019")

1. Verificar el estado del conjunto de réplicas.

> rs.status()

Con estos comandos, se crea un conjunto de réplicas de MongoDB con tres nodos, lo que cumple con los requerimientos planteados en el documento de requerimientos no funcionales en cuanto a la redundancia. Además, como cada nodo estará en una ubicación geográfica diferente, se garantiza una alta disponibilidad 24x7 del sistema.