Actividad 3 - Conceptos y comandos básicos de la replicación en bases de datos NoSQL

Brayan Steven Bonilla Castellanos

Juan Carlos Monsalve Gómez

Corporación Universitaria Iberoamericana

Ingeniería de Software

Bases de datos avanzadas

**Requerimientos no funcionales**

* La base de datos debe permitir realizar todas las operaciones de consulta necesarias.
* La base de datos debe estar disponible 24/7
* Se debe garantizar la seguridad de la información.
* Debe permitir el acceso de varios usuarios al mismo tiempo (concurrencia)
* Se debe garantizar la fiabilidad de la información.
* Debe permitir la escalabilidad horizontal para el crecimiento del negocio.

**Enlace Repositorio GIT**

<https://github.com/jcmonsalveg/Actividad-3---ReplicacionNoSQL>

**Enlace Video**

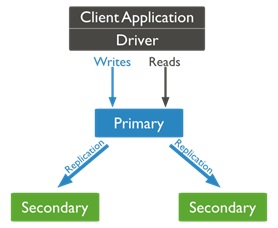
[**https://youtu.be/B2pZj4jVjT0**](https://youtu.be/B2pZj4jVjT0)

**Réplicas DB**

La replicación es la acción de sincronizar la información de una base de datos a través de servidores múltiples, de tal manera que pueda garantizar la redundancia y la disponibilidad de los datos.

En el conjunto de réplicas existen varios nodos, de los cuales hay uno primario o maestro y varios esclavos o secundarios. Dentro del conjunto de nodos se puede tener un árbitro, este es un nodo que participa en las elecciones para la elección del nodo primario frente a una posible caída del nodo primario. la configuración mínima de una réplica establecida pide, entonces, un nodo primario y dos secundarios.

Entre los nodos creados existe una comunicación constante en la cual se verifica la existencia de los mismos (heartbeat). Frente a una caída de un servidor (donde reposa uno de los nodos), los demás nodos garantizan que no existe pérdida de la información, por ejemplo si se cae el nodo primario sucede una elección entre los nodos existentes y uno de los que quedan vivos asumir el nodo de primaria. Cuando el nodo caído se levante, se levantará como nodo secundario, ocurrirá una resincronización y se estabilizará el servicio.

****

**Réplica MongoDB**

**Creamos el Nodo 1 (Debe ser el principal, puerto 27017)**



**Creamos el Nodo 2**

**Creamos el Nodo 3**



**Cerramos las conexiones**



**Nos conectamos al Nodo principal desde la terminal de Mongosh**



**Ejecutamos el comando**  
rs.initiate()

**Agregamos el nodo 2 al nodo primario**



**Agregamos el nodo 3 al nodo primario**

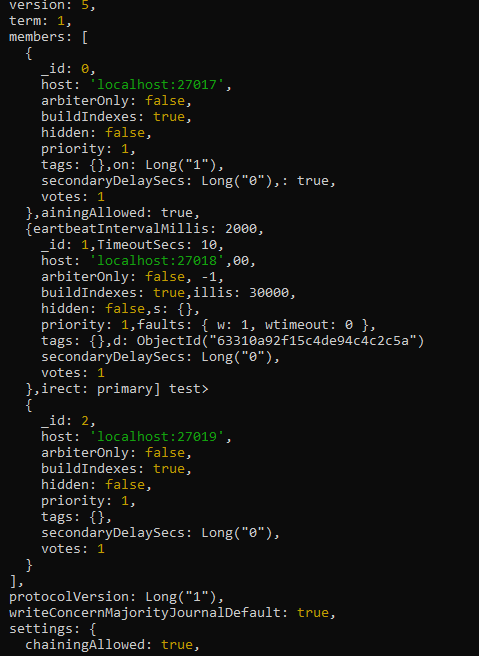


**Agregamos el nodo 4 al nodo primario**



**Verificamos la configuración**





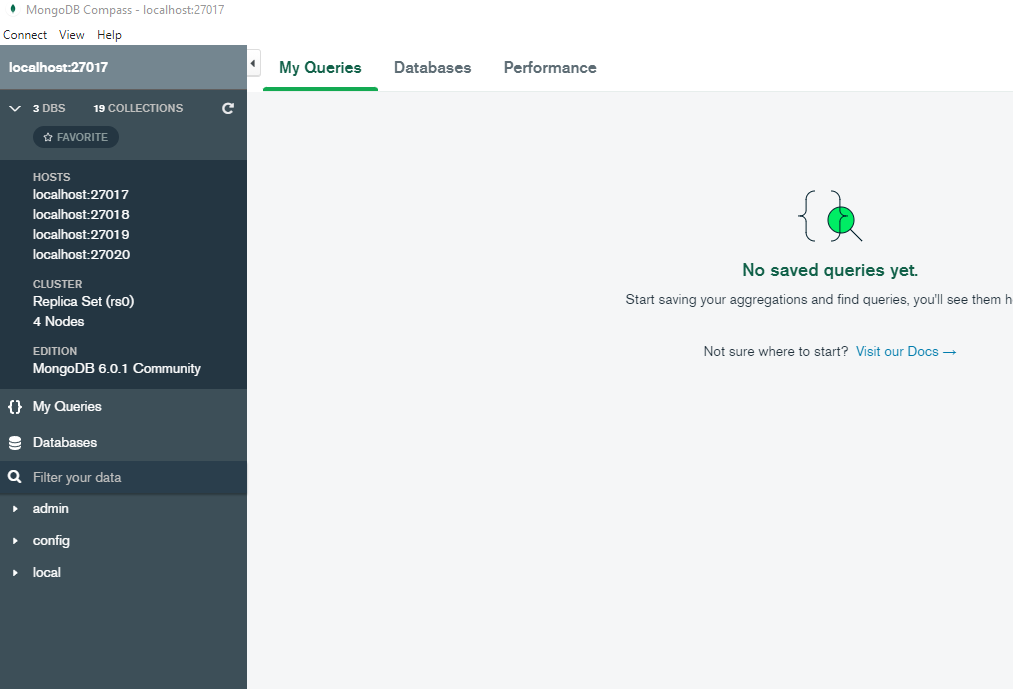
**Verificamos el estado**







**Verificamos la creación correcta de los nodos en la instancia principal**



**Asignamos la configuración a una variable**



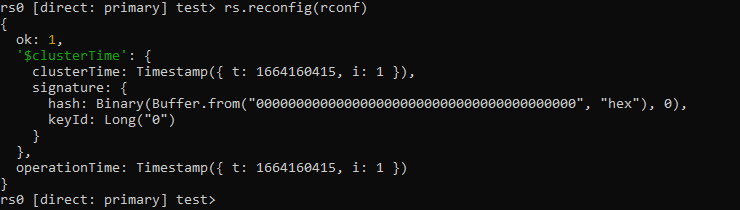
**Cambiamos la prioridad al nodo principal para tolerancia a fallos pueda ser instanciado de nuevo como principal**

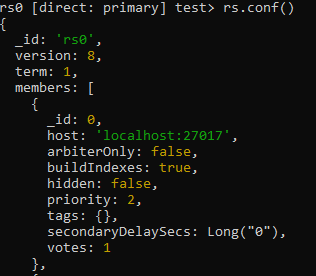


**Sobrescribimos la configuración**

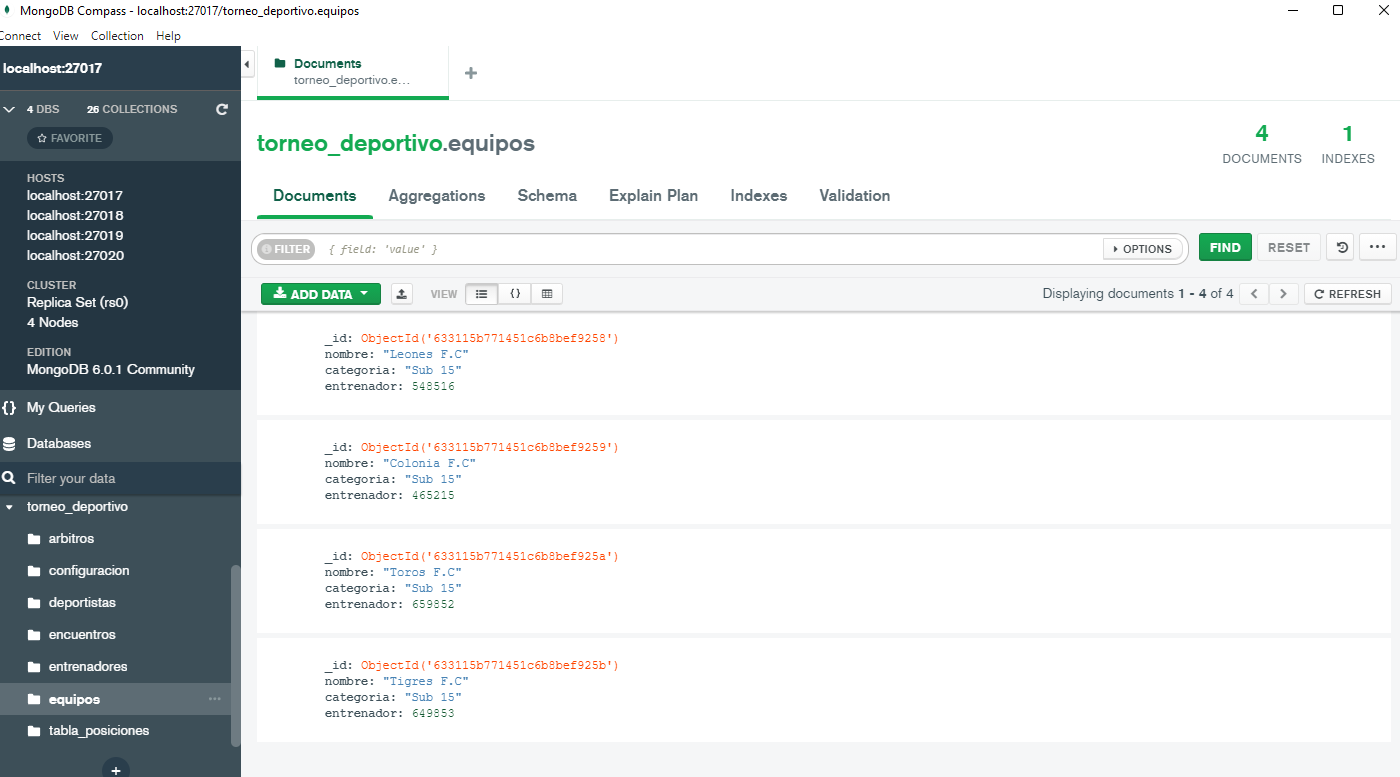


**Verificamos los cambios realizados se apliquen de forma correcta**

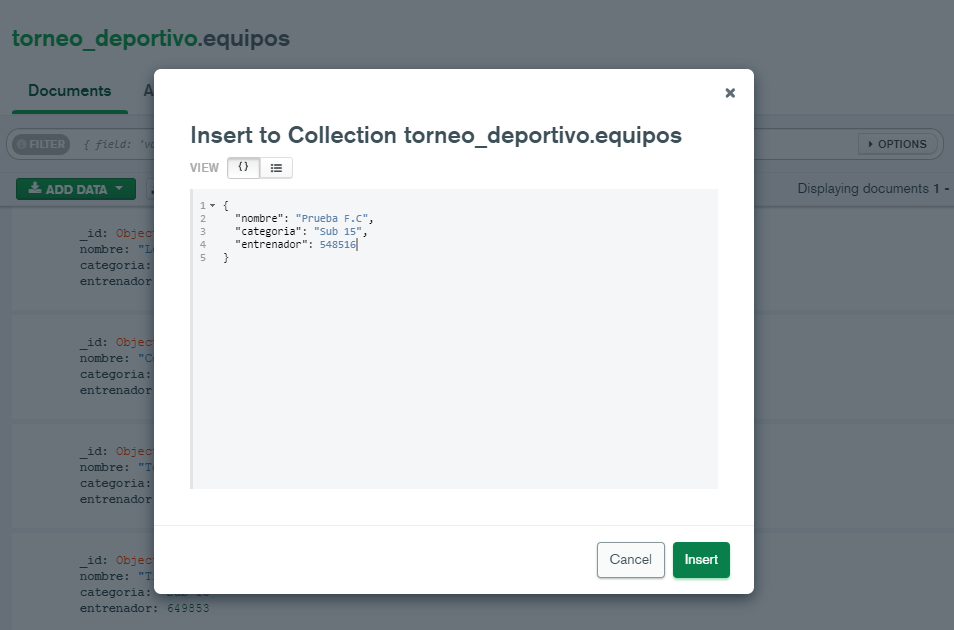




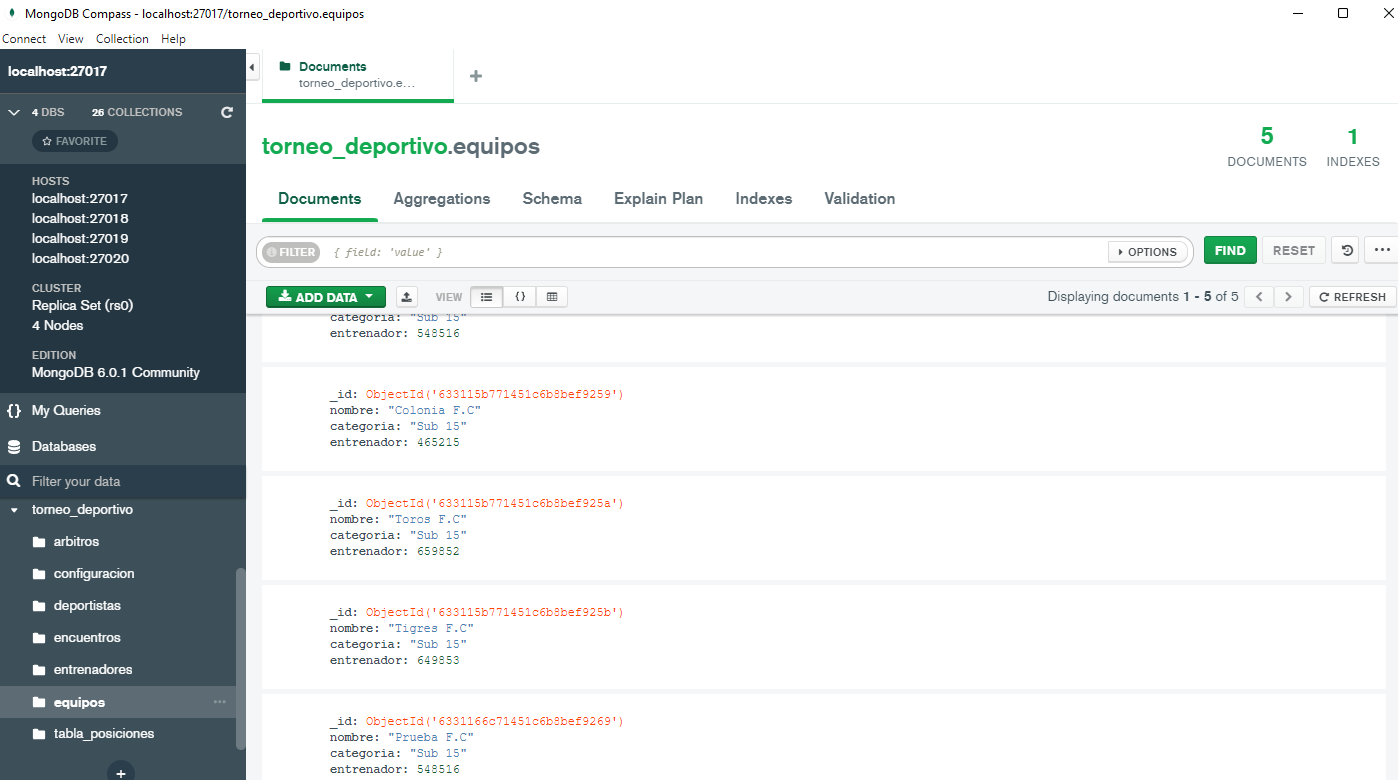
**Consultamos los equipos**

****

**Insertamos un nuevo equipo**

****

**Ahora tenemos 5 registros**

****

**Finalizamos la conexión al nodo principal**

****

**Nos conectamos al nodo secundario**

****

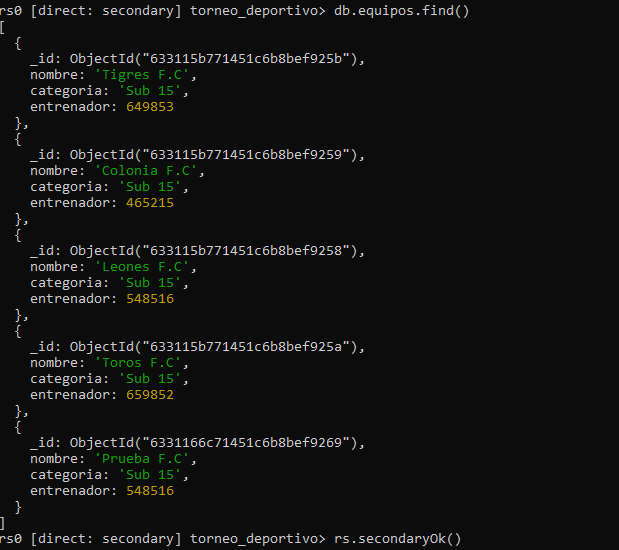
**Seleccionamos la base de datos**

****

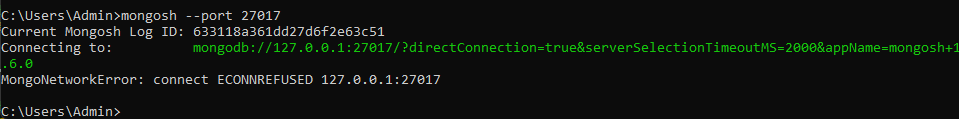
**Asignamos permisos de lectura para el usuario**

****

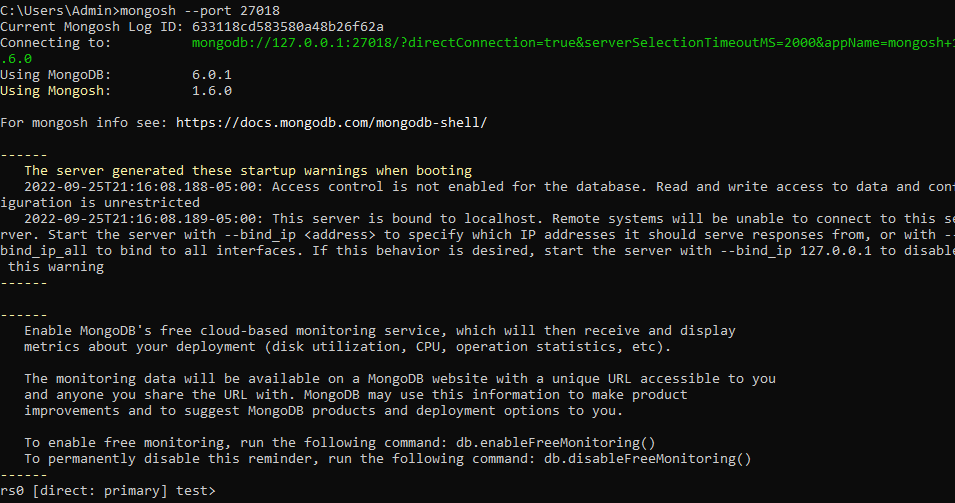
**Realizamos la consulta para verificar la réplica de la información del nodo principal**

****

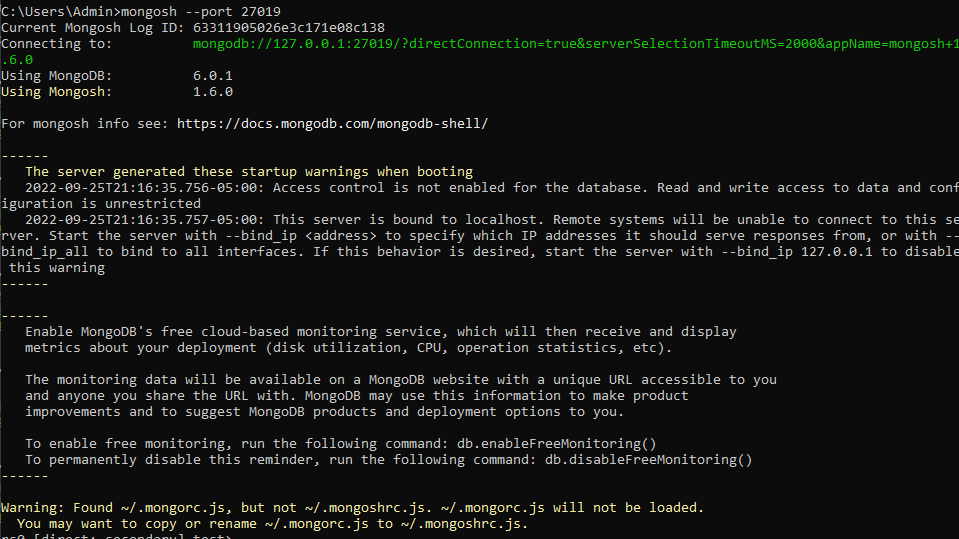
**Cerramos la conexión del nodo principal**

****

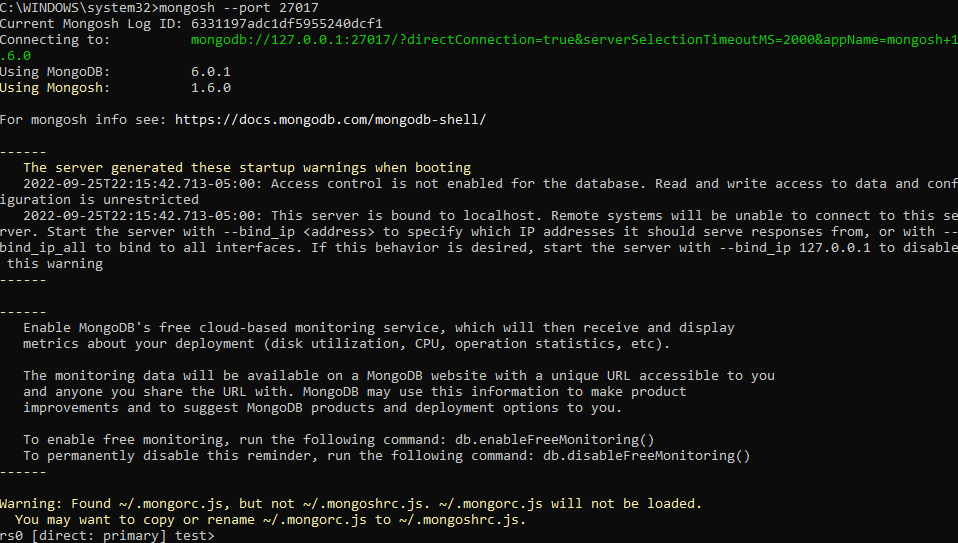
**Verificamos cual nodo tomo el lugar de primario (Tolerancia a fallos), el cual fue el siguiente**

****

**Verificamos que el nodo restante quedo como secundario**

****

**Reestablecemos la instancia del nodo principal y verificamos que es primario**

****