

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



Presentado por:

Carlos Andrés Poveda

FACULTAD DE INGENIERIA DE SOFTWARE

Bases de datos Avanzado

Docente

William Ruiz

Diciembre del 2023

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

Requerimientos No Funcionales

- El proyecto elaborado requiere tener acceso a la base de datos con el nombre de Evento Deportivo las 24 horas los 7 días de la semana, con el fin de garantizar dicho acceso y disponibilidad completa se tendrán en cuenta los siguientes requerimientos:
- Como requerimiento principal es mantener el funcionamiento permanente de la base de datos Evento Deportivo y sus colecciones.
- Se debe replicar la base de datos, en ella deben contener tres nodos, los cuales representaran a tres servidores con el almacenamiento de la misma información.
- Cada uno de los servidores o nodos debe manejar los permisos necesarios para acceder a la información de la base de datos y replicarla.
- En caso de que el servidor o nodo maestro no se pueda acceder el sistema nos permitirá que algún nodo secundario nos de acceso a la información si generar trauma alguno.
- El sistema debe garantizar que el ingreso, la consulta o la modificación de información se realice con contratiempos y de la forma más ágil posible.

Entregable

Entregables: un documento PDF con el link a los siguientes entregables que estará alojado en el repositorio GitHub o Git

que el tutor les indicará.

1. Los comandos o directivas para replicar las bases de datos del evento deportivo en mínimo 3 nodos, planteado aco

documento de requerimientos no funcionales.

2. Los casos de pruebas relacionados con disponibilidad y redundancia requeridos.

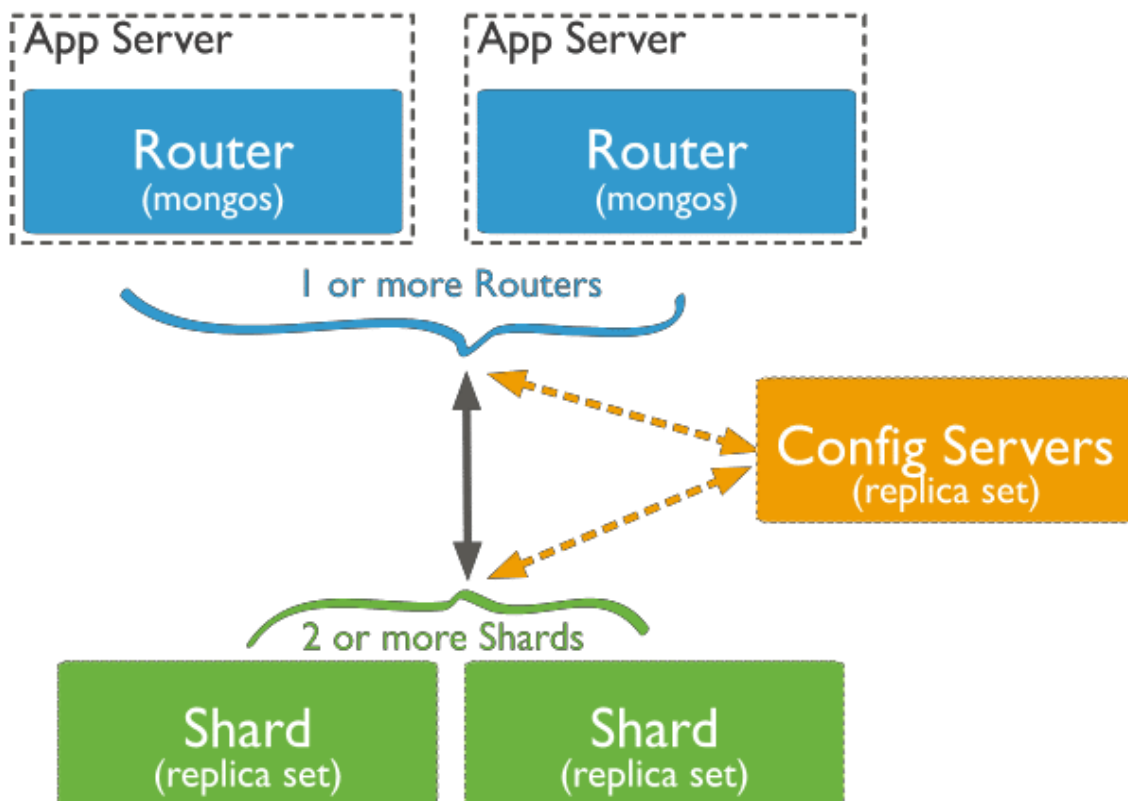
3. Resultados de la realización de los casos de prueba de replicación propuestos.

4. Video (máximo 5 minutos) explicando el proyecto y los comandos realizados y los resultados obtenidos. El video claro y preciso, adicionalmente deben participar todos los integrantes del equipo.

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

Link de video : <https://youtu.be/ZZkra4Njnb0>

Diagrama del particionamiento usado en forma horizontal



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

- Para iniciar necesitamos que cada uno de los servidores de configuración le sean creados directorios de almacenamiento o carpetas, esto gracias al modo de que cada uno tendrá el objetivo de guardar y organizar los datos.

Creación de Nodos con el método ReplicaSet Sentencia

> `MiejReplicaset = new ReplSetTest ({name: "NodosTaller", nodes: 3})`

```
> MiejReplicaset = new ReplSetTest ({name: "NodosTaller", nodes: 3})
Starting new replica set NodosTaller
{
  "kDefaultTimeoutMS" : 600000,
  "getReadConcernMajorityOpTimeOrThrow" : function(conn) {
    const majorityOpTime = _getReadConcernMajorityOpTime(conn);
    if (friendlyEqual(majorityOpTime, {ts: Timestamp(0, 0), t: NumberLong(0)})) {
      throw new Error("readConcern majority optime not available");
    }
    return majorityOpTime;
  },
  "nodeList" : function() {
    var list = [];
    for (var i = 0; i < this.ports.length; i++) {
      list.push(this.host + ":" + this.ports[i]);
    }
    return list;
  },
  "getNodeId" : function(node) {
    if (node.toFixed) {
      return parseInt(node);
    }
  },
  for (var i = 0; i < this.nodes.length; i++) {
    if (this.nodes[i] == node) {
      return i;
    }
  }
}
```

Se valida la creación de los tres nodos y el nombre del equipo

```
{
  "waitForMaster" : function(timeout) {
    var master;
    assert.soonNoExcept(function() {
      return (master = self.getPrimary());
    }, "waiting for master", timeout);
    return master;
  },
  "name" : "NodosTaller",
  "useHostName" : true,
  "host" : "William",
  "oplogSize" : 40,
  "useSeedList" : false,
  "keyFile" : undefined,
  "protocolVersion" : undefined,
  "waitForKeys" : undefined,
  "nodeOptions" : {
    "n0" : undefined,
    "n1" : undefined,
    "n2" : undefined
  },
  "nodes" : [ ],
  "ports" : [
    20000,
    20001,
    20002
  ]
}
```

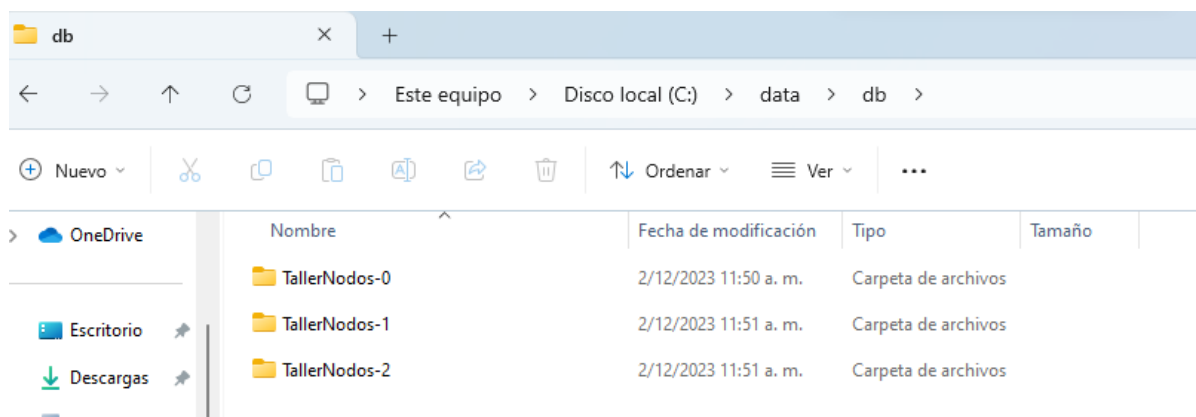
1. Actividad 2 - Conceptos y comandos básicos de la replicación en bases de datos NoSQL

Se toma el conjunto de replicas y se replican con el siguiente comando a cada uno de los nodos

MiejReplicaSet.startSet()

```
> MieiReplicaSet.startSet()
ure":"x86_64","version":"10.0 (build 22631)}"}]}
ReplSetTest made initial connection to node: connection to Cpovedar:20000
ReplSetTest waiting for an initial connection to node 1
d20001| {"t":{"$date":"2023-11-30T20:47:53.422-05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"
Connection accepted","attr":{"remote":"127.0.0.1:178 31","connectionId":1,"connectionCount":1}}
d20001| {"t":{"$date":"2023-11-30T20:47:53.433-05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn1","msg":"cli
ent metadata","attr":{"remote":"127.0.0.1:178 31","client":"conn1","doc":{"application":{"name":"MongoDB Shell"},"driver
":{"name":"MongoDB Internal Client","version":"4.4.25"},"os":{"type":"Windows","name":"Microsoft Windows 11","architect
ure":"x86_64","version":"10.0 (build 22631)}"}}}}
ReplSetTest made initial connection to node: connection to Cpovedar:20001
ReplSetTest waiting for an initial connection to node 2
d20002| {"t":{"$date":"2023-11-30T20:25:53.435-05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"
Connection accepted","attr":{"remote":"127.0.0.1:56179","connectionId":1,"connectionCount":1}}
d20002| {"t":{"$date":"2023-11-30T20:25:53.435-05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn1","msg":"cli
ent metadata","attr":{"remote":"127.0.0.1:56179","client":"conn1","doc":{"application":{"name":"MongoDB Shell"},"driver
":{"name":"MongoDB Internal Client","version":"4.4.25"},"os":{"type":"Windows","name":"Microsoft Windows 10","architect
ure":"x86_64","version":"10.0 (build 22631)}"}}}}
ReplSetTest made initial connection to node: connection to Cpovedar:20002
ReplSetTest startSet, nodes: [
  connection to Cpovedar:20000,
  connection to Cpovedar:20001,
  connection to Cpovedar:20002
]
ReplSetTest startSet took 3821ms for 3 nodes.
[
  connection to Cpovedar:20000,
  connection to Cpovedar:20001,
  connection to Cpovedar:20002
]
```

Se valida la creación de los nodos



En esta instancia ya se van a validar los nodos en donde se realizan las replicas

Primero se establece conexión con el nodo primario

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

```
Nodos Taller x + v
MongoDB shell version v4.4.25
connecting to: mongodb://127.0.0.1:27014/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("717bd71f-7499-479e-9488-7f235289743c") }
MongoDB server version: 4.4.25
---
The server generated these startup warnings when booting:
  2023-11-30T08:00:24.844-05:00: Access control is not enabled for the database. Read and write access to data and
  configuration is unrestricted
---
> conn = new Mongo ("Cpovedar:20000")
connecting to Cpovedar:20000
> testDB=conn.getDB("Taller")
Taller
> |
```

Se validan las Colecciones Dentro de la Base de Datos Taller

```
> show collections
arbitros
deportistas
encuentros
entrenadores
equipos
tablaposiciones
> |
```

Se realiza las validaciones para verificar el nodo primario

Comando: TestDB.isMaster

```
Nodos Taller x + v
{
  "hosts" : [
    "Cpovedar:20000",
    "Cpovedar:20001",
    "Cpovedar:20002"
  ],
  "setName" : "TallerNodes",
  "setVersion" : 3,
  "ismaster" : true,
  "secondary" : false,
  "primary" : "Cpovedar:20000",
  "me" : "Cpovedar:20000",
  "electionId" : ObjectId("7fffffff0000000000000001"),
  "lastWrite" : {
    "upTime" : {
      "ts" : Timestamp(1701222746, 1),
      "t" : NumberLong(1)
    },
    "lastWriteDate" : ISODate("2023-11-30T01:22:26Z"),
    "majorityOpTime" : {
      "ts" : Timestamp(1701222746, 1),
      "t" : NumberLong(1)
    },
    "majorityWriteDate" : ISODate("2023-11-29T01:52:26Z")
  },
  "maxBsonObjectSize" : 17777216,
  "maxMessageSizeBytes" : 48000000,
  "maxWriteBatchSize" : 100000,
  "localTime" : ISODate("2023-11-30T01:56:05.815Z"),
  "logicalSessionTimeoutMinutes" : 48,
}
```

Se realiza pruebas con inserción de datos a la colección Deportistas

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

Sentencia: testDB.deportistas.insert

```
> testDB.deportistas.insert(
... .. { "Nombres": "Julian",
... ..   "Apellidos": "Alvarez",
... ..   "Edad": 23,
... ..   "Cedula": 1061741241,
... ..   "Celular": {
... ..     "$numberLong": "3154117224"}});
WriteResult({ "nInserted" : 1 })
```

Se realiza pruebas con inserción de datos a la colección Equipos

```
> testDB.equipo.insert(
... .. {
... ..   "Nombre": "SantaFe",
... ..   "Ciudad": "Bogota",    41241,
... ..   "Año": 1912
... .. }
WriteResult({ "nInserted" : 1 })
```

Se valida consultando los datos ingresados deportistas

testDB.deportistas.find()

```
> testDB.deportistas.count()
2
> testDB.deportistas.find()
{ "_id" : ObjectId("656ba07cc1c42873bfc6f3af"), "Nombres" : "Julian", "Apellidos" : "Alvarez", "Edad" : 23, "Cedula" : 1061741241, "Celular" : "312412741" }
{ "_id" : ObjectId("656ba0a7c1c42873bfc6f3b1"), "Nombres" : "Gabriel", "Apellidos" : "Jesus", "Edad" : 25, "Cedula" : 1061741241, "Celular" : "312412741" }
```

Se valida consultando datos ingresado a la colección equipos

```
> testDB.equipo.count()
2
> db.equipo.find()
> testDB.equipo.find()
{ "_id" : ObjectId("656b9fab1c42873bfc6f3aa"), "Nombre" : "SantaFe", "Ciudad" : "Bogota", "Año" : 1912 }
{ "_id" : ObjectId("656b9fe5c1c42873bfc6f3ac"), "Nombre" : "Barcelona", "Ciudad" : "Madrid", "Año" : 1900 }
```

Se valida conectándose al nodo 20001 para verificar que si se este replicando la

información del nodo maestro

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

Primero se valida conexión

```
> connSecondary = new Mongo("Cpovedar:20001")
connection to Cpovedar :20001
> secondaryTestDB = connSecondary.getDB("Taller.")
```

Luego de conectarnos le consultamos al nodo si es o no el nodo maestro lo cual la respuesta fue (False) es decir que no

SecondaryTestDB.isMaster()

```
"setName" : "TallerNodos",
"setVersion" : 3,
"ismaster" : false,
"secondary" : true,
"primary" : "Cpovedar:20000",
"me" : "Cpovedar:20001",
"lastWrite" : {
  "opTime" : {
```

Ya estando en el nodo secundario se debe habilitar permisos de lectura y se realiza con el siguiente comando

connSecondary.setSecondaryOk()

```
> connSecondary.setSecondaryOk()
```

Luego de conceder permisos de lectura a este nodo secundario se procede a consultar las dos colecciones y los dos registros

```
> connSecondary.setSecondaryOk()
> connSecondaryTestdb.deportistas.find()
{ "_id" : ObjectId("656ba97cc1c42873bfc6f3af"), "Nombres" : "Julian", "Apellidos" : "Alvarez", "Edad" : 23, "Cedula" : 1061741241, "Celular" : "312412741" }
{ "_id" : ObjectId("656ba97cc1c42873bfc6f3b1"), "Nombres" : "Gabriel", "Apellidos" : "Jesus", "Edad" : 26, "Cedula" : 1061741241, "Celular" : "312412741" }
```

También se valida en la colección equipos

```
> connSecondaryTestdb.equipos.find()
{ "_id" : ObjectId("656b9fabc1c42873bfc6f3aa"), "Nombre" : "SantaFe", "Ciudad" : "Bogota", "Año" : 1912 }
{ "_id" : ObjectId("656b9fe5c1c42873bfc6f3ac"), "Nombre" : "Barcelona", "Ciudad" : "Madrid", "Año" : 1900 }
> |
```

Se valida conexión al nodo secundario

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

```
> connSecondary = new Mongo("Cpovedar:20002")
connection to Cpovedar:20002
> secondaryTestDB = connSecondary.getDB("Taller.")
```

Dado que el nodo secundario no tiene permiso se realiza la sentencia para los permisos de lectura

connSecondary.setSecondaryOk()

```
> connSecondary.setSecondaryOk()
```

Se valida con la sentencia que donde se esta consultando no es el nodo primario

SecondaryTestDB.isMaster()

```
"setName" : "TallerNodos",
"setVersion" : 3,
"ismaster" : false,
"secondary" : true,
"primary" : "Cpovedar:20000",
"me" : "Cpovedar:20002",
"lastWrite" : {
  "opTime" : {
```

Se consultan las colecciones de los datos

consultar las dos colecciones y los dos registros

```
> connSecondary.setSecondaryOk()
> connSecondaryTestdb.deportistas.find()
{ "_id" : ObjectId("656ba97c1c42873bfc6f3af"), "Nombres" : "Julian", "Apellidos" : "Alvarez", "Edad" : 23, "Cedula" : 1061741241, "Celular" : "3124127411" }
{ "_id" : ObjectId("656ba97c1c42873bfc6f3b1"), "Nombres" : "Gabriel", "Apellidos" : "Jesus", "Edad" : 26, "Cedula" : 1061741241, "Celular" : "3124127411" }
```

También se valida en la colección equipos

```
> connSecondaryTestdb.equipos.find()
{ "_id" : ObjectId("656b9fab1c42873bfc6f3aa"), "Nombre" : "SantaFe", "Ciudad" : "Bogota", "Año" : 1912 }
{ "_id" : ObjectId("656b9fe5c1c42873bfc6f3ac"), "Nombre" : "Barcelona", "Ciudad" : "Madrid", "Año" : 1900 }
> |
```

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

Ahora para definir un nuevo nodo secundario no conectamos al nodo 200001

```
> connPrimary = new Mongo("localhost:20001")
connection to localhost:20001
> |
```

Y con el siguiente comando le estamos definiendo un nuevo nodo primario

Con el comando `NewPrimary = connNewPrimary.getDB("Taller")`

```
> NewPrimary = connNewPrimary.getDB("Taller")
```

Volvemos a consultarle al nodo si estamos situados en el nodo primario que ya definimos para confirmar

`SecondaryTestDB.isMaster()`

```
]
"setName" : "TallerNodos",
"setVersion" : 3,
"ismaster" : true,
"secondary" : false,
"primary" : "Cpovedar:20001",
"me" : "Cpovedar:20001",
"electionId" : ObjectId("7fffffff00000000000000000002"),
"lastWrite" : {
```

Link de video : <https://youtu.be/ZZkra4NJnb0>