

Actividad 2: Conceptos y comandos básicos de la replicación en bases de datos NoSQL

Neyder Alexis Hernández Carrillo
Víctor Alfonso Méndez Palacios

Facultad de Ingeniería, Corporación Universitaria Iberoamericana

Ing. William Ruiz

29 de noviembre del 2023

Actividad 2: Conceptos y comandos básicos de la replicación en bases de datos NoSQL

Enlace del video: <https://drive.google.com/file/d/1xhJ1wE-MTgYBL92mI4oyc4uFsuF1oGgP/view?usp=sharing>

1. Requerimientos no funcionales para el proceso de replicación

Este documento detalla los criterios de calidad no funcionales relacionados con la redundancia y la disponibilidad 24x7 para el sistema que implica la replicación en MongoDB en el contexto del proyecto “Jankenpon”:

ID: RNF - 1	Nombre: Respuesta de las transacciones	Categoría: RENDIMIENTO
Descripción	El sistema debe garantizar, el ingreso, la consulta o la modificación de la información sin contratiempos de forma ágil.	
Criterio de medición	Los tiempos de respuesta promedio de una consulta en Mongo DB se encuentra entre los 112 y 3600 milisegundos por lo que no se deberá superar los 3600 milisegundos.	

ID: RNF - 2	Nombre: Disponibilidad del sistema	Categoría: DISPONIBILIDAD
Descripción	Mantener el funcionamiento permanente de la base de datos “Jankenpon” y sus colecciones (árbitros, deportistas y entrenadores).	
Criterio de medición	El sistema deberá proporcionar una disponibilidad del (99,9%) del tiempo	

ID: RNF - 3	Nombre: Respaldo continuo	Categoría: RESPALDO Y RECUPERACIÓN
Descripción	Se mantendrá la replicación de la base de datos como mínimo en tres nodos (un primario y dos secundarios), los cuales representan a tres servidores con el almacenamiento de la misma información para garantizar la redundancia y tolerancia a fallos.	
Criterio de medición	Todos los nodos involucrados en la replicación deberán replicar la misma información	

ID: RNF - 4	Nombre: Configuración de permisos	Categoría: SEGURIDAD
Descripción	Cada servidor manejará los permisos necesarios para acceder a la información de la base de datos, replicarla.	
Criterio de medición	Los servidores secundarios solo tendrán acceso a lectura y no a escritura por parte del cliente	

ID: RNF - 5	Nombre: Recuperación	Categoría: TOLERANCIA A FALLOS / DISPONIBILIDAD
Descripción	En caso de caída del servidor principal uno de los nodos secundarios deberá tomar su posición y permitirá el acceso a la información sin generar trauma alguno.	
Criterio de medición	El sistema delegará a un nodo secundario cuando el principal no responde.	

2. Casos de pruebas en replicación con MongoDB

Tipo de prueba	Objetivo
Disponibilidad	Ingresar dos o más documentos en las colecciones propuestas en el documento de requerimientos en el nodo primario y verificar que todas las instancias tienen una réplica de los registros insertados.
Replicación	Verificar que se haya creado el nodo primario y los secundarios.
Tolerancia a fallos	Prueba de desconexión del nodo primario y promoción de alguno de los nodos secundarios a primario.
Disponibilidad	Verificar cuál de los nodos es el primario después de una caída del nodo principal.

3. Consultas para replicar la base de datos del evento deportivo

- **Paso 1 -Creación del Replicaset**

ReplicaSet = new ReplSetTest ({name: "MireplicaSetJankenpon", nodes: 3})

```
        return master;
    },
    "name" : "MireplicaSetJankenpon",
    "useHostName" : true,
    "host" : "DESKTOP-4VE9656",
    "oplogSize" : 40,
    "useSeedList" : false,
    "keyFile" : undefined,
    "protocolVersion" : undefined,
    "waitForKeys" : undefined,
    "seedRandomNumberGenerator" : true,
    "isConfigServer" : undefined,
    "nodeOptions" : {
        "n0" : undefined,
        "n1" : undefined,
        "n2" : undefined
    },
    "nodes" : [ ],
    "ports" : [
        20000,
        20001,
        20002
    ]
}
```




- **Paso 2 -Arrancar los procesos mongod de la replica**

ReplicaSet.startSet()

```

ReplSetTest startSet, nodes: [
    connection to DESKTOP-4VE9656:20000,
    connection to DESKTOP-4VE9656:20001,
    connection to DESKTOP-4VE9656:20002
]
ReplSetTest startSet took 7059ms for 3 nodes.
[
    connection to DESKTOP-4VE9656:20000,
    connection to DESKTOP-4VE9656:20001,
    connection to DESKTOP-4VE9656:20002
]

```

> Este equipo > Disco local (C:) > data > db				
Nombre	Fecha de modificación	Tipo	Tamaño	
 MireplicaSetJankenpon-0	30/11/2023 3:51 p. m.	Carpeta de archivos		
 MireplicaSetJankenpon-1	30/11/2023 3:51 p. m.	Carpeta de archivos		
 MireplicaSetJankenpon-2	30/11/2023 3:51 p. m.	Carpeta de archivos		

- **Paso 3 -Arrancar el proceso de replica**

```
ReplicaSet.initiate()
```

- **Paso 4 -Prueba del grupo de replica**

```
conn=new Mongo("DESKTOP-4VE9656:20000")
```

```

> conn=new Mongo("DESKTOP-4VE9656:20000")
connection to DESKTOP-4VE9656:20000

```

```
testDB=conn.getDB("jankenpon")
```

```

> testDB=conn.getDB("jankenpon")
jankenpon

```

testDB.isMaster()

```
> testDB.isMaster()
{
  "topologyVersion" : {
    "processId" : ObjectId("6569294d306e9c30b095ad1c"),
    "counter" : NumberLong(8)
  },
  "hosts" : [
    "DESKTOP-4VE9656:20000",
    "DESKTOP-4VE9656:20001",
    "DESKTOP-4VE9656:20002"
  ],
  "setName" : "MireplicaSetJankenpon",
  "setVersion" : 3,
  "ismaster" : true,
  "secondary" : false,
  "primary" : "DESKTOP-4VE9656:20000",
  "me" : "DESKTOP-4VE9656:20000",
  "electionId" : ObjectId("7fffffff0000000000000001"),
```

- Paso 5 -Insertar un conjunto de datos en el nodo primario

Árbitros

```
testDB.arbitros.insert(
{
  "nombre": "Mateo García",
  "edad": 23,
  "nacionalidad": "Peruano"
});
```

```
testDB.arbitros.insert(
{
  "nombre": "Rodolfo Hernandez",
  "edad": 27,
  "nacionalidad": "Colombiano"
});
```

```
testDB.arbitros.insert(
```

```
{
```

```
  "nombre": "Julio Morales",
```

```
  "edad": 28,
```

```
  "nacionalidad": "Mexicano"
```

```
});
```

```
testDB.arbitros.insert(
```

```
{
```

```
  "nombre": "Maria López",
```

```
  "edad": 30,
```

```
  "nacionalidad": "Chilena"
```

```
});
```

```
testDB.arbitros.insert(
```

```
{
```

```
  "nombre": "Camila rojas",
```

```
  "edad": 34,
```

```
  "nacionalidad": "Ecuatoriana"
```

```
})
```

```

> testDB.arbitros.insert(
... {
...   "nombre": "Mateo García",
...   "edad": 23,
...   "nacionalidad": "Peruano"
... });
WriteResult({ "nInserted" : 1 })
> testDB.arbitros.insert(
... {
...   "nombre": "Rodolfo Hernandez",
...   "edad": 27,
...   "nacionalidad": "Colombiano"
... });
WriteResult({ "nInserted" : 1 })
> testDB.arbitros.insert(
... {
...   "nombre": "Julio Morales",
...   "edad": 28,
...   "nacionalidad": "Mexicano"
... });
WriteResult({ "nInserted" : 1 })
>
> testDB.arbitros.insert(
... {
...   "nombre": "Maria López",
...   "edad": 30,
...   "nacionalidad": "Chilena"
... });
WriteResult({ "nInserted" : 1 })
>
> testDB.arbitros.insert(
... {
...   "nombre": "Camila rojas",
...   "edad": 34,
...   "nacionalidad": "Ecuatoriana"
... });
WriteResult({ "nInserted" : 1 })

```

Comprobación:

```
testDB.arbitros.count();
```

```

WriteResult({ "nInserted" : 1 })
> testDB.arbitros.count();
5

```

```
testDB.arbitros.find();
```

```

> testDB.arbitros.find();
{ "_id" : ObjectId("656904f63e1c92f57f101e4e"), "nombre" : "Mateo García", "edad" : 23, "nacionalidad" : "Peruano" }
{ "_id" : ObjectId("6569087e3e1c92f57f101e4f"), "nombre" : "Rodolfo Hernandez", "edad" : 27, "nacionalidad" : "Colombiano" }
{ "_id" : ObjectId("656908993e1c92f57f101e50"), "nombre" : "Julio Morales", "edad" : 28, "nacionalidad" : "Mexicano" }
{ "_id" : ObjectId("656908993e1c92f57f101e51"), "nombre" : "Maria López", "edad" : 30, "nacionalidad" : "Chilena" }
{ "_id" : ObjectId("656908993e1c92f57f101e52"), "nombre" : "Camila rojas", "edad" : 34, "nacionalidad" : "Ecuatoriana" }

```


Deportistas

```
testDB.deportistas.insert(  
  {  
    "numero_jugador": 1,  
    "nombre": "Michael Jordan",  
    "edad": 40,  
    "entrenador": "Gabriela Martinez",  
    "nacionalidad": "Usa"  
  });
```

```
testDB.deportistas.insert(  
  {  
    "numero_jugador": 2,  
    "nombre": "Leonel Messi",  
    "edad": 37,  
    "entrenador": "Luis Gonzales",  
    "nacionalidad": "Argentina"  
  });
```

```
testDB.deportistas.insert(  
  {  
    "numero_jugador": 3,  
    "nombre": "Lewis Hamilton",  
    "edad": 35,  
    "entrenador": "Toto Wolf",  
    "nacionalidad": "UK"  
  });
```

```
testDB.deportistas.insert(  
{  
  "numero_jugador": 4,  
  "nombre": "Lebron James",  
  "edad": 34,  
  "entrenador": "Ricardo Gómez",  
  "nacionalidad": "USA"  
});
```

```
testDB.deportistas.insert(  
{  
  "numero_jugador": 5,  
  "nombre": "Sergio Pérez",  
  "edad": 34,  
  "entrenador": "Alejandro Morales",  
  "nacionalidad": "México"  
});
```

```
testDB.deportistas.insert(  
{  
  "numero_jugador": 6,  
  "nombre": "Rafael Nadal",  
  "edad": 28,  
  "entrenador": "Carlos Fernandez",  
  "nacionalidad": "España"  
});
```

```
testDB.deportistas.insert(  
{  
  "numero_jugador": 7,  
  "nombre": "Cristiano Ronaldo",  
  "edad": 38,  
  "entrenador": "Daniela Maecha",  
  "nacionalidad": "Portugal"  
});
```

```
testDB.deportistas.insert(  
{  
  "numero_jugador": 8,  
  "nombre": "Kylian Mbape",  
  "edad": 25,  
  "entrenador": "Nicol Sanchez",  
  "nacionalidad": "Francia"  
});
```

```
> testDB.deportistas.insert(
... {
...   "numero_jugador": 1,
...   "nombre": "Michael Jordan",
...   "edad": 40,
...   "entrenador": "Gabriela Martinez",
...   "nacionalidad": "Usa"
... });
WriteResult({ "nInserted" : 1 })
> testDB.deportistas.insert(
... {
...   "numero_jugador": 2,
...   "nombre": "Leonel Messi",
...   "edad": 37,
...   "entrenador": "Luis Gonzales",
...   "nacionalidad": "Argentina"
... });
WriteResult({ "nInserted" : 1 })
>
> testDB.deportistas.insert(
... {
...   "numero_jugador": 3,
...   "nombre": "Lewis Hamilton",
...   "edad": 35,
...   "entrenador": "Toto Wolf",
...   "nacionalidad": "UK"
... });
WriteResult({ "nInserted" : 1 })
>
> testDB.deportistas.insert(
... {
...   "numero_jugador": 4,
...   "nombre": "Lebron James",
...   "edad": 34,
...   "entrenador": "Ricardo Gómez",
...   "nacionalidad": "USA"
... });
WriteResult({ "nInserted" : 1 })
```

```

> testDB.deportistas.insert(
... {
...   "numero_jugador": 5,
...   "nombre": "Sergio Pérez",
...   "edad": 34,
...   "entrenador": "Alejandro Morales",
...   "nacionalidad": "México"
... });
WriteResult({ "nInserted" : 1 })
>
> testDB.deportistas.insert(
... {
...   "numero_jugador": 6,
...   "nombre": "Rafael Nadal",
...   "edad": 28,
...   "entrenador": "Carlos Fernandez",
...   "nacionalidad": "España"
... });
WriteResult({ "nInserted" : 1 })
>
>
>
> testDB.deportistas.insert(
... {
...   "numero_jugador": 7,
...   "nombre": "Cristiano Ronaldo",
...   "edad": 38,
...   "entrenador": "Daniela Maecha",
...   "nacionalidad": "Portugal"
... });
WriteResult({ "nInserted" : 1 })
>
> testDB.deportistas.insert(
... {
...   "numero_jugador": 8,
...   "nombre": "Kylian Mbape",
...   "edad": 25,
...   "entrenador": "Nicol Sanchez",
...   "nacionalidad": "Francia"
... });
WriteResult({ "nInserted" : 1 })

```

Comprobación:

```
testDB.deportistas.count();
```

```

> testDB.deportistas.count();
8

```

```
testDB.deportistas.find();
```

```
> testDB.deportistas.find();
{ "_id" : ObjectId("656908ba3e1c92f57f101e53"), "numero_jugador" : 1, "nombre" : "Michael Jordan", "edad" : 40, "entrenador" : "Gabriela Martinez", "nacionalidad" : "USA" }
{ "_id" : ObjectId("656908d03e1c92f57f101e54"), "numero_jugador" : 2, "nombre" : "Leonel Messi", "edad" : 37, "entrenador" : "Luis Gonzales", "nacionalidad" : "Argentina" }
{ "_id" : ObjectId("656908d03e1c92f57f101e55"), "numero_jugador" : 3, "nombre" : "Lewis Hamilton", "edad" : 35, "entrenador" : "Toto Wolf", "nacionalidad" : "UK" }
{ "_id" : ObjectId("656908d03e1c92f57f101e56"), "numero_jugador" : 4, "nombre" : "Lebron James", "edad" : 34, "entrenador" : "Ricardo Gómez", "nacionalidad" : "USA" }
{ "_id" : ObjectId("656908d03e1c92f57f101e57"), "numero_jugador" : 5, "nombre" : "Sergio Pérez", "edad" : 34, "entrenador" : "Alejandro Morales", "nacionalidad" : "México" }
{ "_id" : ObjectId("656908d03e1c92f57f101e58"), "numero_jugador" : 6, "nombre" : "Rafael Nadal", "edad" : 28, "entrenador" : "Carlos Fernandez", "nacionalidad" : "España" }
{ "_id" : ObjectId("656908d13e1c92f57f101e59"), "numero_jugador" : 7, "nombre" : "Cristiano Ronaldo", "edad" : 38, "entrenador" : "Daniela Maecha", "nacionalidad" : "Portugal" }
{ "_id" : ObjectId("656908d13e1c92f57f101e5a"), "numero_jugador" : 8, "nombre" : "Kylian Mbape", "edad" : 25, "entrenador" : "Nicol Sanchez", "nacionalidad" : "Francia" }
}
```

Entrenadores

```
testDB.entrenadores.insert(
{
  "nombre": "Gabriela Martinez",
  "edad": 40,
  "nacionalidad": "USA"
});
```

```
testDB.entrenadores.insert(
{
  "nombre": "Luis Gonzales",
  "edad": 37,
  "nacionalidad": "Argentina"
});
```

```
testDB.entrenadores.insert(
{
  "nombre": "Toto Wolf",
  "edad": 36,
  "nacionalidad": "UK"
});
```

```
testDB.entrenadores.insert(  
  {  
    "nombre": "Ricardo Gómez",  
    "edad": 31,  
    "nacionalidad": "USA"  
  });
```

```
testDB.entrenadores.insert(  
  {  
    "nombre": "Alejandro Morales",  
    "edad": 32,  
    "nacionalidad": "México"  
  });
```

```
testDB.entrenadores.insert(  
  {  
    "nombre": "Carlos Fernandez",  
    "edad": 34,  
    "nacionalidad": "España"  
  });
```

```
testDB.entrenadores.insert(  
  {  
    "nombre": "Daniela Maecha",  
    "edad": 46,  
    "nacionalidad": "Portugal"  
  });
```

```
testDB.entrenadores.insert(  
{  
  "nombre": "Nicol Sanchez",  
  "edad": 49,  
  "nacionalidad": "Francia"  
});
```

```
> testDB.entrenadores.insert(  
... {  
...   "nombre": "Toto Wolf",  
...   "edad": 36,  
...   "nacionalidad": "UK"  
... });  
WriteResult({ "nInserted" : 1 })  
>  
> testDB.entrenadores.insert(  
... {  
...   "nombre": "Ricardo Gómez",  
...   "edad": 31,  
...   "nacionalidad": "USA"  
... });  
WriteResult({ "nInserted" : 1 })  
>  
> testDB.entrenadores.insert(  
... {  
...   "nombre": "Alejandro Morales",  
...   "edad": 32,  
...   "nacionalidad": "México"  
... });  
WriteResult({ "nInserted" : 1 })  
>  
> testDB.entrenadores.insert(  
... {  
...   "nombre": "Carlos Fernandez",  
...   "edad": 34,  
...   "nacionalidad": "España"  
... });  
WriteResult({ "nInserted" : 1 })  
>  
> testDB.entrenadores.insert(  
... {  
...   "nombre": "Daniela Maecha",  
...   "edad": 46,  
...   "nacionalidad": "Portugal"  
... });  
WriteResult({ "nInserted" : 1 })
```


Comprobación:

```
testDB.entrenadores.count();
```

```
> testDB.entrenadores.count();  
8
```

```
testDB.entrenadores.find();
```

```
> testDB.entrenadores.find();  
{ "_id" : ObjectId("656908e53e1c92f57f101e5b"), "nombre" : "Gabriela Martinez", "edad" : 40, "nacionalidad" : "USA" }  
{ "_id" : ObjectId("656908f53e1c92f57f101e5c"), "nombre" : "Toto Wolf", "edad" : 36, "nacionalidad" : "UK" }  
{ "_id" : ObjectId("656908f53e1c92f57f101e5d"), "nombre" : "Ricardo Gómez", "edad" : 31, "nacionalidad" : "USA" }  
{ "_id" : ObjectId("656908f53e1c92f57f101e5e"), "nombre" : "Alejandro Morales", "edad" : 32, "nacionalidad" : "México" }  
{ "_id" : ObjectId("656908f53e1c92f57f101e5f"), "nombre" : "Carlos Fernandez", "edad" : 34, "nacionalidad" : "España" }  
{ "_id" : ObjectId("656908f53e1c92f57f101e60"), "nombre" : "Daniela Maecha", "edad" : 46, "nacionalidad" : "Portugal" }  
{ "_id" : ObjectId("656908f53e1c92f57f101e61"), "nombre" : "Nicol Sanchez", "edad" : 49, "nacionalidad" : "Francia" }  
{ "_id" : ObjectId("656909503e1c92f57f101e62"), "nombre" : "Luis Gonzales", "edad" : 37, "nacionalidad" : "Argentina" }  
>
```

- **Paso 6 -Comprobación de la réplica sobre los nodos secundarios**

```
connSecondary = new Mongo("DESKTOP-4VE9656:20001");
```

```
> connSecondary = new Mongo("DESKTOP-GE8MA9M:20001");  
connection to DESKTOP-GE8MA9M:20001
```

```
secondaryTestDB = connSecondary.getDB("jankenpon");
```

Comprobamos si este nodo es el máster o no lo es:

```
secondaryTestDB.isMaster();
```

```

> secondaryTestDB.isMaster();
{
  "topologyVersion" : {
    "processId" : ObjectId("6569294d8eba161bc798d203"),
    "counter" : NumberLong(4)
  },
  "hosts" : [
    "DESKTOP-4VE9656:20000",
    "DESKTOP-4VE9656:20001",
    "DESKTOP-4VE9656:20002"
  ],
  "setName" : "MireplicaSetJankenpon",
  "setVersion" : 3,
  "ismaster" : false,
  "secondary" : true,
  "primary" : "DESKTOP-4VE9656:20000",
  "me" : "DESKTOP-4VE9656:20001",
  "lastWrite" : {
    "opTime" : {
      "ts" : Timestamp(1701392562, 6),
      "t" : NumberLong(1)
    },
    "lastWriteDate" : ISODate("2023-12-01T01:02:42Z"),
    "majorityOpTime" : {
      "ts" : Timestamp(1701392562, 6),

```

Consultamos la colección entrenadores y notamos que no tenemos permisos de lectura

```
secondaryTestDB.entrenadores.count();
```

```

> secondaryTestDB.entrenadores.count();
uncaught exception: Error: count failed: {
  "topologyVersion" : {
    "processId" : ObjectId("6568f5a84f5664f4a3134e7f"),
    "counter" : NumberLong(4)
  },
  "operationTime" : Timestamp(1701382480, 1),
  "ok" : 0,
  "errmsg" : "not master and slaveOk=false",
  "code" : 13435,
  "codeName" : "NotPrimaryNoSecondaryOk",
  "$clusterTime" : {
    "clusterTime" : Timestamp(1701382480, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DBCollection.prototype.count@src/mongo/shell/collection.js:1401:15
@(shell):1:1
>

```

Activamos permiso para realizar operaciones de lectura y consultamos documentos

```
connSecondary.setSecondaryOk();
```

Consultas:

```
secondaryTestDB.deportistas.count();
```

```
secondaryTestDB.arbitros.count();
```

```
secondaryTestDB.entrenadores.count();
```

```
> connSecondary.setSecondaryOk();
> secondaryTestDB.deportistas.count();
8
> secondaryTestDB.arbitros.count();
5
> secondaryTestDB.entrenadores.count();
8
```

Realizamos la prueba con el nodo en el puerto 20002

```
connSecondary = new Mongo("DESKTOP-4VE9656:20002");
```

```
secondaryTestDB = connSecondary.getDB("jankenpon");
```

```
secondaryTestDB.isMaster();
```

```
> secondaryTestDB.isMaster();
{
  "topologyVersion" : {
    "processId" : ObjectId("6569294dd49c258706bcd58f"),
    "counter" : NumberLong(3)
  },
  "hosts" : [
    "DESKTOP-4VE9656:20000",
    "DESKTOP-4VE9656:20001",
    "DESKTOP-4VE9656:20002"
  ],
  "setName" : "MireplicaSetJankenpon",
  "setVersion" : 3,
  "ismaster" : false,
  "secondary" : true,
  "primary" : "DESKTOP-4VE9656:20000",
  "me" : "DESKTOP-4VE9656:20002",
  "lastWrite" : {
```

Activar permiso para realizar operaciones de lectura y consultamos documentos

```
connSecondary.setSecondaryOk();
```

Consultas:

```
secondaryTestDB.deportistas.count();  
secondaryTestDB.arbitros.count();  
secondaryTestDB.entrenadores.count();
```

```
> connSecondary.setSecondaryOk();  
> secondaryTestDB.deportistas.count();  
8  
> secondaryTestDB.entrenadores.count();  
8  
> secondaryTestDB.arbitros.count();  
5
```

Se evidencia que la réplica ha sido exitosa

- **Paso 7 -Detener el nodo primario**

```
connPrimary = new Mongo("DESKTOP-4VE9656:20000")  
primaryDB = connPrimary.getDB("jankenpon")  
primaryDB.isMaster()
```

Comando de apagado

```
primaryDB.adminCommand({ shutdown : 1 });
```

```
> primaryDB.adminCommand({shutdown : 1})  
uncaught exception: Error: error doing query: failed: network error while attempting to run command 'shutdown' on host '  
localhost:20000' :  
DB.prototype.runCommand@src/mongo/shell/db.js:169:19  
DB.prototype.adminCommand@src/mongo/shell/db.js:187:12  
@(shell):1:1  
> |
```

- **Paso 8- Comprobación del nuevo nodo primario**

```
connNewPrimary = new Mongo("DESKTOP-4VE9656:20001")  
newPrimaryDB = connNewPrimary.getDB ("jankenpon")  
newPrimaryDB.isMaster()
```

```
> newPrimaryDB.isMaster()
{
  "topologyVersion" : {
    "processId" : ObjectId("6569294d306e9c30b095ad1c"),
    "counter" : NumberLong(8)
  },
  "hosts" : [
    "DESKTOP-4VE9656:20000",
    "DESKTOP-4VE9656:20001",
    "DESKTOP-4VE9656:20002"
  ],
  "setName" : "MireplicaSetJankenpon",
  "setVersion" : 3,
  "ismaster" : true,
  "secondary" : false,
  "primary" : "DESKTOP-4VE9656:20001",
  "me" : "DESKTOP-4VE9656:20001",
}
```

- Paso 9- Detener el ReplicaSet de pruebas

ReplicaSet.StopSet();

```
> ReplicaSet.stopSet()
uncaught exception: ReferenceError: ReplicaSet is not defined :
@(shell):1:1
>
```

Conclusiones

En el marco del proyecto "Jankenpon", los requisitos no funcionales para la replicación en MongoDB establecen criterios cruciales para garantizar el rendimiento, la disponibilidad y la integridad de la base de datos, por ello para el proceso de replicación se realizó un marco de criterios para la validación de dichos requisitos no funcionales.

Referencias

Mongodb. (30 de 11 de 2023). *MongoDB community*. Obtenido de <https://www.mongodb.com/try/download/community>

Sarasa, A. (2016). *Introducció a las bases de datos NoSQL usando MongoDB*. UOC.