



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Culiacán

Administración de Base de Datos

Replicación

ALUMNO:

Jesús Omar Langarica Ornelas
10-11

MAESTRO:

Leopoldo Zenaido Zepeda Sanchez

CARRERA:

Ingeniería en Sistemas Computacionales

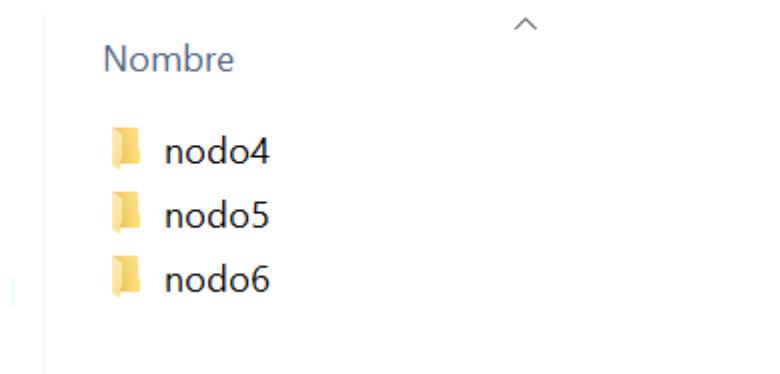
Culiacán, Sinaloa

23 de Octubre del 2025

Paso 1: Cree las carpetas donde MongoDB guardara los datos.

```
C:\Users\Omar>mkdir C:\datos  
C:\Users\Omar>mkdir C:\datos\nodo4  
C:\Users\Omar>mkdir C:\datos\nodo5  
C:\Users\Omar>mkdir C:\datos\nodo6
```

Este equipo > Local Disk (C:) > datos



Paso 2: Iniciamos los tres nodos.

Utilice estos comandos, uno en cada cmd.

```
cd "C:\Program Files\MongoDB\Server\8.2\bin"
```

```
mongod --replSet rs0 --port 27017 --bind_ip 127.0.0.1 --dbpath C:\datos\nodo4 --oplogSize 120
```

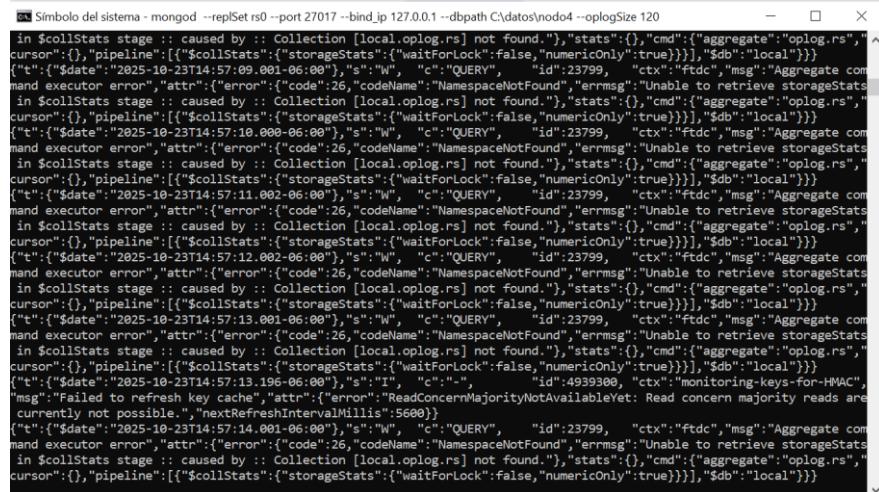
```
cd "C:\Program Files\MongoDB\Server\8.2\bin"
```

```
mongod --replSet rs0 --port 27018 --bind_ip 127.0.0.1 --dbpath C:\datos\nodo5 --oplogSize 120
```

```
cd "C:\Program Files\MongoDB\Server\8.2\bin"
```

```
mongod --replSet rs0 --port 27019 --bind_ip 127.0.0.1 --dbpath C:\datos\nodo6 --oplogSize 120
```

Nodo 4 – Puerto 27017



```
Simbolo del sistema - mongod --replSet rs0 --port 27017 --bind_ip 127.0.0.1 --dbpath C:\datos\nodo4 --oplogSize 120
in $collStats stage :: caused by :: Collection [local.oplog.rs] not found.", "stats": {}, "cmd": {"aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}}
("t": {"$date": "2025-10-23T14:57:09.001-06:00"}, "s": "W", "c": "QUERY", "id": 23799, "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"code": 26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}, "stats": {}, "cmd": {"aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}}
("t": {"$date": "2025-10-23T14:57:10.000-06:00"}, "s": "W", "c": "QUERY", "id": 23799, "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"code": 26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}, "stats": {}, "cmd": {"aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}}
("t": {"$date": "2025-10-23T14:57:11.002-06:00"}, "s": "W", "c": "QUERY", "id": 23799, "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"code": 26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}, "stats": {}, "cmd": {"aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}}
("t": {"$date": "2025-10-23T14:57:12.002-06:00"}, "s": "W", "c": "QUERY", "id": 23799, "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"code": 26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}, "stats": {}, "cmd": {"aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}}
("t": {"$date": "2025-10-23T14:57:13.196-06:00"}, "s": "I", "c": "", "id": 4939300, "ctx": "monitoring-keys-for-HMAC", "msg": "Failed to refresh key cache", "attr": {"error": {"ReadConcernMajorityNotAvailableYet: Read concern majority reads are currently not possible."}, "nextRefreshIntervalMillis": 5000}}
("t": {"$date": "2025-10-23T14:57:14.001-06:00"}, "s": "W", "c": "QUERY", "id": 23799, "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"code": 26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}, "stats": {}, "cmd": {"aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}}
("t": {"$date": "2025-10-23T14:57:14.001-06:00"}, "s": "W", "c": "QUERY", "id": 23799, "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"code": 26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}, "stats": {}, "cmd": {"aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}}
```

Nodo 5 – Puerto 27018

Symbolo del sistema - mongod --repSet rs0 --port 27018 --bind_ip 127.0.0.1 --dbpath C:\data\mnode5 --oplogSize 120

```
mand executor error","attr":{"error":{"code":26,"codeName":"NamespaceNotFound","errmsg":"Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}},"stats":{},"cmd":{"aggregate":["oplog.rs","cursor":{}],"pipeline":[{"$collStats":{"$storageStats":{"$waitForLock":false,"$numericOnly":true}}}],"$db":"local"}}]}},{"t":{"$date":2015-10-23T14:57:27.001-06:00},"s":"W","c":"QUERY","id":23799,"ctx":"ftdc","msg":"Aggregate command executor error","attr":{"error":{"code":26,"codeName":"NamespaceNotFound","errmsg":"Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}},"stats":{},"cmd":{"aggregate":["oplog.rs","cursor":{}],"pipeline":[{"$collStats":{"$storageStats":{"$waitForLock":false,"$numericOnly":true}}}],"$db":"local"}}]}]}},{"t":{"$date":2015-10-23T14:57:28.001-06:00},"s":"W","c":"QUERY","id":23799,"ctx":"ftdc","msg":"Aggregate command executor error","attr":{"error":{"code":26,"codeName":"NamespaceNotFound","errmsg":"Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}},"stats":{},"cmd":{"aggregate":["oplog.rs","cursor":{}],"pipeline":[{"$collStats":{"$storageStats":{"$waitForLock":false,"$numericOnly":true}}}],"$db":"local"}}]}]}},{"t":{"$date":2015-10-23T14:57:28.000-06:00},"s":"W","c":"QUERY","id":23799,"ctx":"ftdc","msg":"Aggregate command executor error","attr":{"error":{"code":26,"codeName":"NamespaceNotFound","errmsg":"Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}},"stats":{},"cmd":{"aggregate":["oplog.rs","cursor":{}],"pipeline":[{"$collStats":{"$storageStats":{"$waitForLock":false,"$numericOnly":true}}}],"$db":"local"}}]}]}},{"t":{"$date":2015-10-23T14:57:29.000-06:00},"s":"W","c":"QUERY","id":23799,"ctx":"ftdc","msg":"Aggregate command executor error","attr":{"error":{"code":26,"codeName":"NamespaceNotFound","errmsg":"Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}},"stats":{},"cmd":{"aggregate":["oplog.rs","cursor":{}],"pipeline":[{"$collStats":{"$storageStats":{"$waitForLock":false,"$numericOnly":true}}}],"$db":"local"}}]}]} {"t":{"$date":2015-10-23T14:57:29.653-06:00}, "s": "I", "c": "-", "id": 4989300, "ctx": "monitoring-keys-for-HMAC", "msg": "Failed to refresh key cache", "attr": {"error": "ReadConcernMajorityNotAvailableYet: Read concern majority reads are currently not possible.", "nextRefreshIntervalMillis": 46000}}]} {"t":{"$date":2015-10-23T14:57:30.002-06:00}, "s": "W", "c": "QUERY", "id": 23799, "ctx": "ftdc", "msg": "Aggregate command executor error","attr":{"error":{"code":26,"codeName":"NamespaceNotFound","errmsg":"Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}},"stats":{},"cmd":{"aggregate":["oplog.rs","cursor":{}],"pipeline":[{"$collStats":{"$storageStats":{"$waitForLock":false,"$numericOnly":true}}}],"$db":"local"}}]}]} {"t":{"$date":2015-10-23T14:57:31.002-06:00}, "s": "W", "c": "QUERY", "id": 23799, "ctx": "ftdc", "msg": "Aggregate command executor error","attr":{"error":{"code":26,"codeName":"NamespaceNotFound","errmsg":"Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found."}},"stats":{},"cmd":{"aggregate":["oplog.rs","cursor":{}],"pipeline":[{"$collStats":{"$storageStats":{"$waitForLock":false,"$numericOnly":true}}}],"$db":"local"}})}]
```

Nodo 6 – Puerto 27019

Paso 3: Iniciamos el Replica Set

- **Me conecte al primer nodo con el MongoDB Shell, con el siguiente comando:**

```
cd "C:\Program Files\MongoDB\mongosh-2.5.8-win32-x64\bin"
```

```
mongosh --port 27017
```

```
C:\Windows\system32>cd "C:\Program Files\MongoDB\mongosh-2.5.8-win32-x64\bin"
C:\Program Files\MongoDB\mongosh-2.5.8-win32-x64\bin>mongosh --port 27017
Current Mongosh Log ID: 68faa15f911ce98bcebea3
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB:     8.2.1
Using Mongosh:    2.5.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-10-23T15:42:12.747-06:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
ed
-----
```

- **Ejecuté el siguiente comando para iniciar el Replica Set:**

```
rs.initiate({
  _id: "rs0",
  members: [
    { _id: 0, host: "127.0.0.1:27017" },
    { _id: 1, host: "127.0.0.1:27018" },
    { _id: 2, host: "127.0.0.1:27019" }
  ]
})
```

```
test> rs.initiate({ _id: "rs0", members: [{ _id: 0, host: "127.0.0.1:27017" }, { _id: 1, host: "127.0.0.1:27018" }, { _id: 2, host: "127.0.0.1:27019" }])
{
  ok: 1,
  "$clusterTime": {
    clusterTime: Timestamp({ t: 1761255778, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1761255778, i: 1 })
}
:rs0 [direct: secondary] test>
:
```

- Verifique el estado con rs.status()

```

members: [
  {
    _id: 0,
    name: '127.0.0.1:27017',
    health: 1,
    state: 1,
    stateStr: 'PRIMARY',
    uptime: 186,
    optime: { ts: Timestamp({ t: 1761255909, i: 1 }), t: Long('1') },
    optimeDate: ISODate('2025-10-23T21:45:09.000Z'),
    optimeWritten: { ts: Timestamp({ t: 1761255909, i: 1 }), t: Long('1') },
    optimeWrittenDate: ISODate('2025-10-23T21:45:09.000Z'),
    lastAppliedWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    lastDurableWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    lastWrittenWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    syncSourceHost: '',
    syncSourceId: -1,
    infoMessage: '',
    electionTime: Timestamp({ t: 1761255789, i: 1 }),
    electionDate: ISODate('2025-10-23T21:43:09.000Z'),
    configVersion: 1,
    configTerm: 1,
    self: true,
    lastHeartbeatMessage: ''
  },
  {
    _id: 1,
    name: '127.0.0.1:27018',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 140,
    optime: { ts: Timestamp({ t: 1761255909, i: 1 }), t: Long('1') },
    optimeDurability: { ts: Timestamp({ t: 1761255909, i: 1 }), t: Long('1') },
    optimeWritten: { ts: Timestamp({ t: 1761255909, i: 1 }), t: Long('1') },
    optimeWrittenDate: ISODate('2025-10-23T21:45:09.000Z'),
    optimeDurabilityDate: ISODate('2025-10-23T21:45:09.000Z'),
    optimeWrittenDate: ISODate('2025-10-23T21:45:09.000Z'),
    lastAppliedWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    lastDurableWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    lastWrittenWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    lastHeartbeat: ISODate('2025-10-23T21:45:17.412Z'),
    lastHeartbeatRecv: ISODate('2025-10-23T21:45:18.374Z'),
    pingMs: Long('0'),
    lastHeartbeatMessage: '',
    syncSourceHost: '127.0.0.1:27017',
    syncSourceId: 0,
    infoMessage: '',
    configVersion: 1,
    configTerm: 1
  },
  {
    _id: 2,
    name: '127.0.0.1:27019',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 140,
    optime: { ts: Timestamp({ t: 1761255909, i: 1 }), t: Long('1') },
    optimeDurability: { ts: Timestamp({ t: 1761255909, i: 1 }), t: Long('1') },
    optimeWritten: { ts: Timestamp({ t: 1761255909, i: 1 }), t: Long('1') },
    optimeWrittenDate: ISODate('2025-10-23T21:45:09.000Z'),
    optimeDurabilityDate: ISODate('2025-10-23T21:45:09.000Z'),
    optimeWrittenDate: ISODate('2025-10-23T21:45:09.000Z'),
    lastAppliedWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    lastDurableWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    lastWrittenWallTime: ISODate('2025-10-23T21:45:09.374Z'),
    lastHeartbeat: ISODate('2025-10-23T21:45:17.412Z'),
    lastHeartbeatRecv: ISODate('2025-10-23T21:45:18.374Z'),
    pingMs: Long('0'),
    lastHeartbeatMessage: '',
    syncSourceHost: '127.0.0.1:27017',
    syncSourceId: 0,
    infoMessage: '',
    configVersion: 1,
    configTerm: 1
  }
]

```

Se puede observar un nodo como PRIMARY y otros dos como SECONDARY.

Paso 4: Hacemos las pruebas de replicación.

- **Inserté datos en el nodo primario.**

```
use bd
```

```
db.emp.insert({id:1, name:"juaniuo2"})
```

```
db.emp.find().forEach(printjson)
```

```
rs0 [direct: primary] bd> db.emp.insert({id:1, name:"juaniuo2"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('68faa394911ce98bcecebea4') }
}
rs0 [direct: primary] bd> db.emp.find().forEach(printjson)
{
  _id: ObjectId('68faa394911ce98bcecebea4'),
  id: 1,
  name: 'juaniuo2'
}
```

- **Lectura en los nodos secundarios.**

En mi caso empecé con el nodo 5, con puerto 27018.

En el documento nos dice que utilicemos rs.slaveOk() para poder leer en nodos secundarios, pero en mi versión de mongodb la cual es la mas actualizada me dice que ya no es compatible, entonces en su lugar utilice:

```
db.getMongo().setReadPref('secondary')
```

Y después puse en uso la bd y coloqué los siguientes comandos:

```
use Bd
```

```
db.emp.find().forEach(printjson)
```

```
rs0 [direct: secondary] test> db.getMongo().setReadPref('secondary')
rs0 [direct: secondary] test> use bd
switched to db bd
rs0 [direct: secondary] bd> db.emp.find().forEach(printjson)
{
  _id: ObjectId('68faa394911ce98bcecebea4'),
  id: 1,
  name: 'juanio2'
}

rs0 [direct: secondary] bd>
```

Hacemos lo mismo en el nodo 6, con puerto 27019.

```
rs0 [direct: secondary] test> db.getMongo().setReadPref('secondary')

rs0 [direct: secondary] test> use bd
switched to db bd
rs0 [direct: secondary] bd> db.emp.find().forEach(printjson)
{
  _id: ObjectId('68faa394911ce98bcecebea4'),
  id: 1,
  name: 'juanio2'
}
```