# ACTIVIDAD 3 - CONCEPTOS Y COMANDOS BÁSICOS DE LA REPLICACIÓN EN BASES DE DATOS NOSQL

**BRANDON SNEIDER FRANCO QUINTERO**

**UNIVERSIDAD IBEROAMERICANA**

**BASE DE DATOS AVANZADAS**

**OCTUBRE 2022**

**BOGOTA**

# ACTIVIDAD 3 - CONCEPTOS Y COMANDOS BÁSICOS DE LA REPLICACIÓN EN BASES DE DATOS NOSQL

**DOCENTE**

**WILLIAM RUIZ MARTINEZ**

**ESTUDIANTE**

**BRANDON SNEIDER FRANCO QUINTERO**

**UNIVERSIDAD IBEROAMERICANA**

**BASE DE DATOS AVANZADAS**

**OCTUBRE 2022**

**BOGOTA**

# REPLICACION EN BD

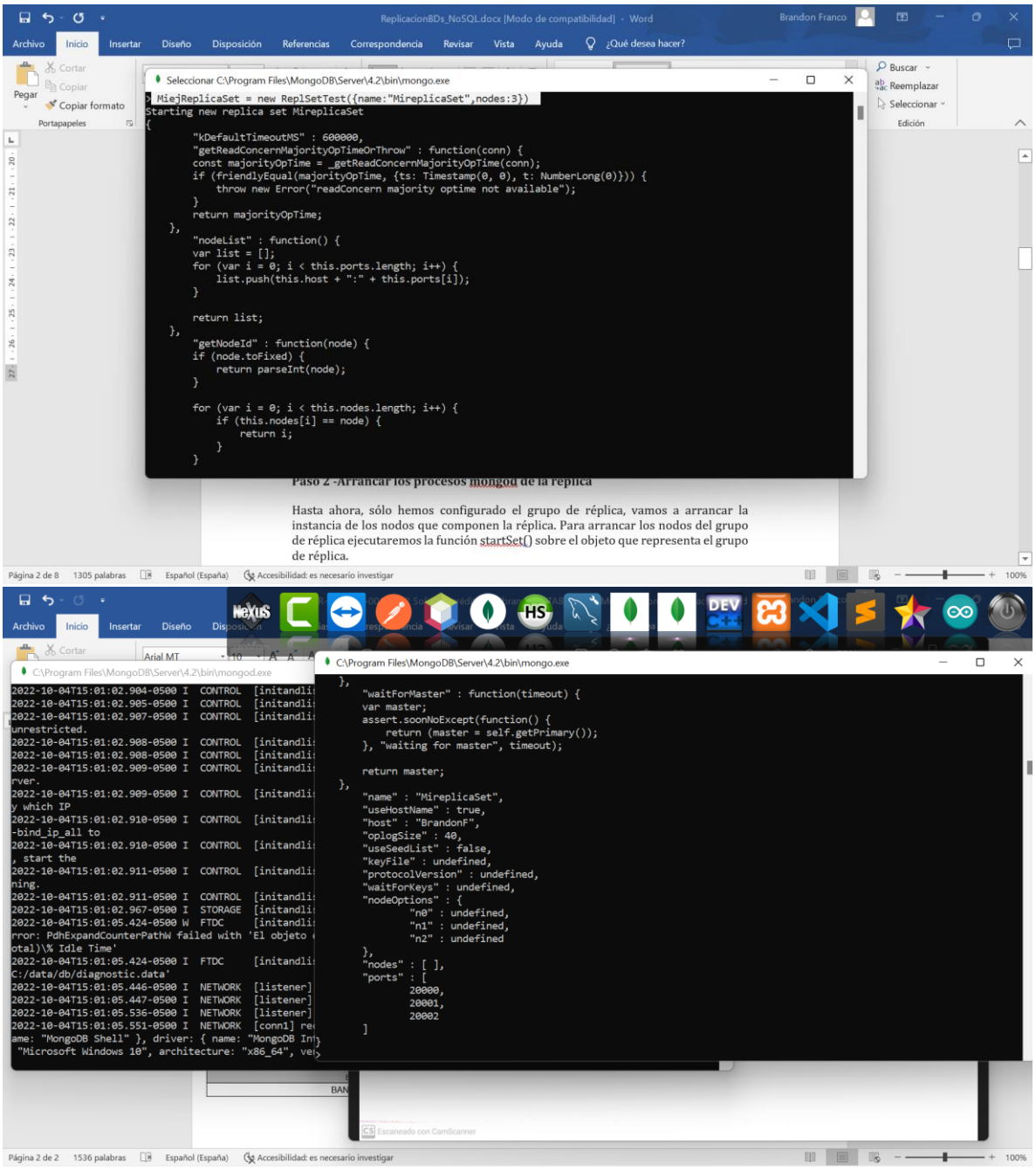
La replicación de datos es cuando los mismos datos se almacenan intencionalmente en más de un sitio o servidor. Hay varias razones por las que las empresas replican datos. Permite que los datos estén disponibles sin problemas en el caso de un tiempo de inactividad del servidor o mucho tráfico hacia el servidor. Los datos se vuelven accesibles para los usuarios de manera constante sin interferir ni ralentizar el acceso de otros usuarios. Para las aplicaciones en la nube, la replicación de datos le permite acceder a una copia de los datos en una base de datos local con un rendimiento mucho mayor que acceder a los datos a través de la API de la aplicación en la nube, que es especialmente útil para análisis y la ciencia de datos. La replicación de datos también puede permitirle evitar los límites de transacción de la API y la aceleración que tienen algunas aplicaciones en la nube.

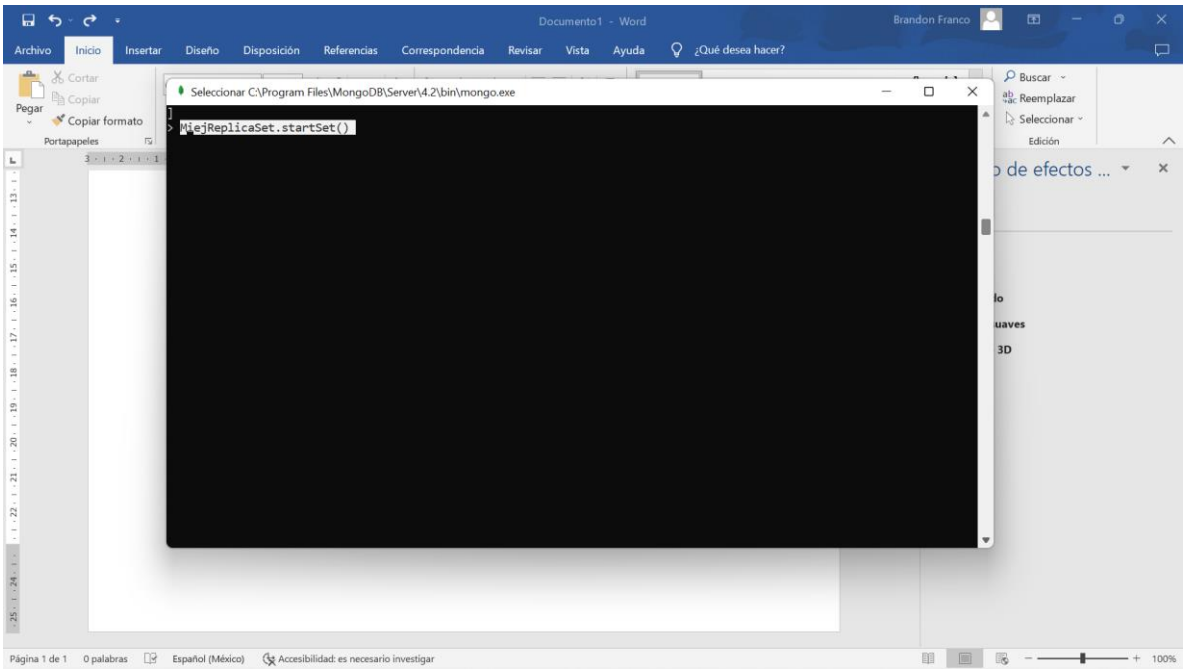
- **Aumento de la fiabilidad:** mediante la replicación de base de datos a través de múltiples servidores, te aseguras que los datos van a estar disponibles incluso en el caso de que una de las máquinas tenga un fallo grave de hardware. El sistema distribuido de gestión de bases de datos debe ser capaz de enrutar a los usuarios afectados a otro de los nodos disponibles.
- **Mejora en el rendimiento:** al estar los datos distribuidos en diferentes servidores, los múltiples accesos no saturan los servidores. Esto es importante sobre todo en el caso de aplicaciones que pueden tener miles o cientos de miles de peticiones simultáneas. El rendimiento de las aplicaciones aumenta notablemente.
- **Mejora en la seguridad de los datos:** en un sistema transaccional tradicional, todas las actualizaciones de una base de datos se guardan en un mismo disco. La seguridad de tus datos queda entonces en manos de la estrategia de copias de seguridad que tengas implementada en ese servidor. Con la replicación de base de datos aumentas la seguridad de los datos ya que las actualizaciones están siendo escritas en varios servidores. Es decir, varios discos, varias fuentes de alimentación, CPU’s, etc. son utilizadas para asegurar que tus datos estarán a salvo en algunos servidores, aunque pueda ocurrir un desastre en otros.

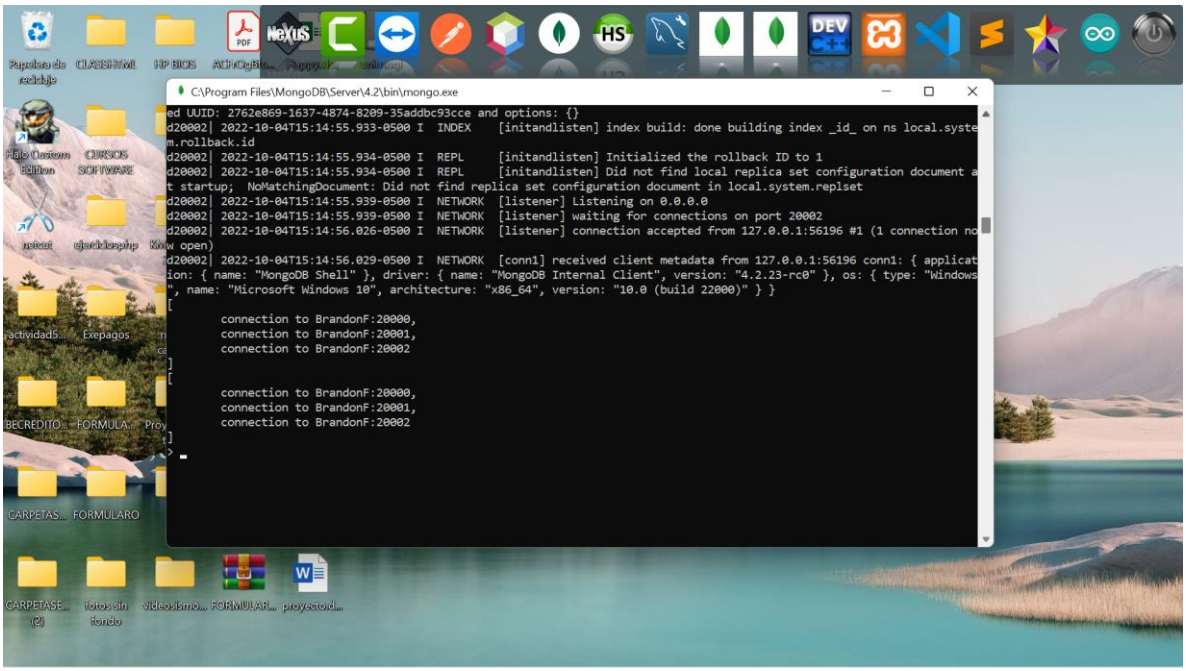
# PROCESO

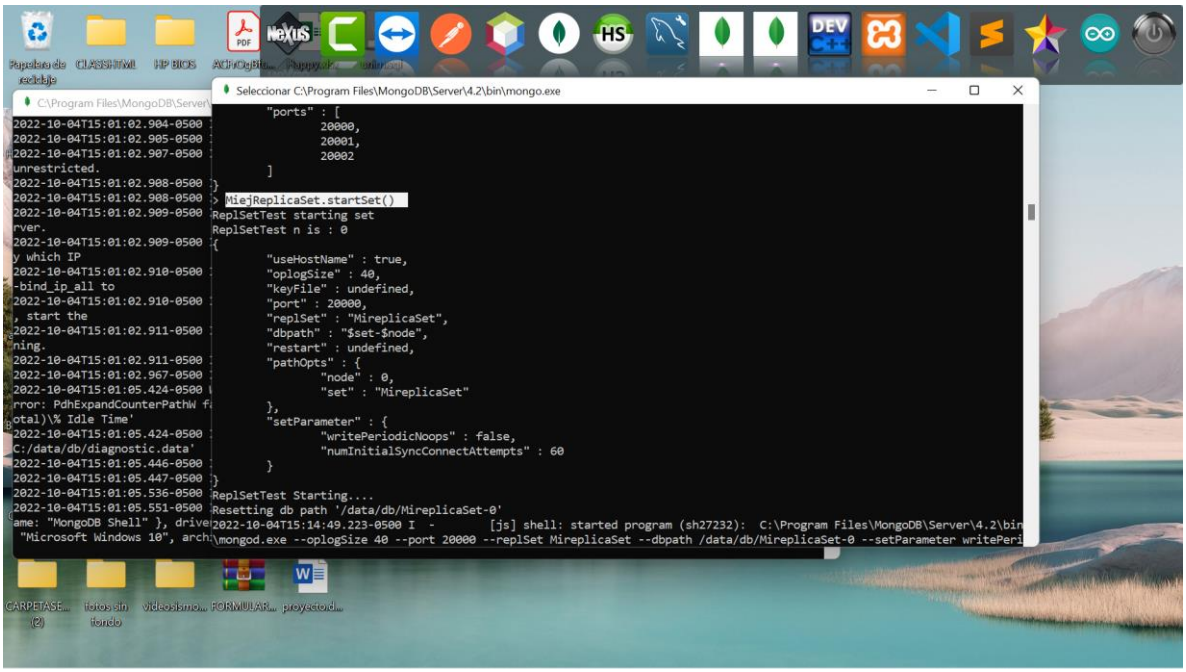
## Requerimientos no funcionales

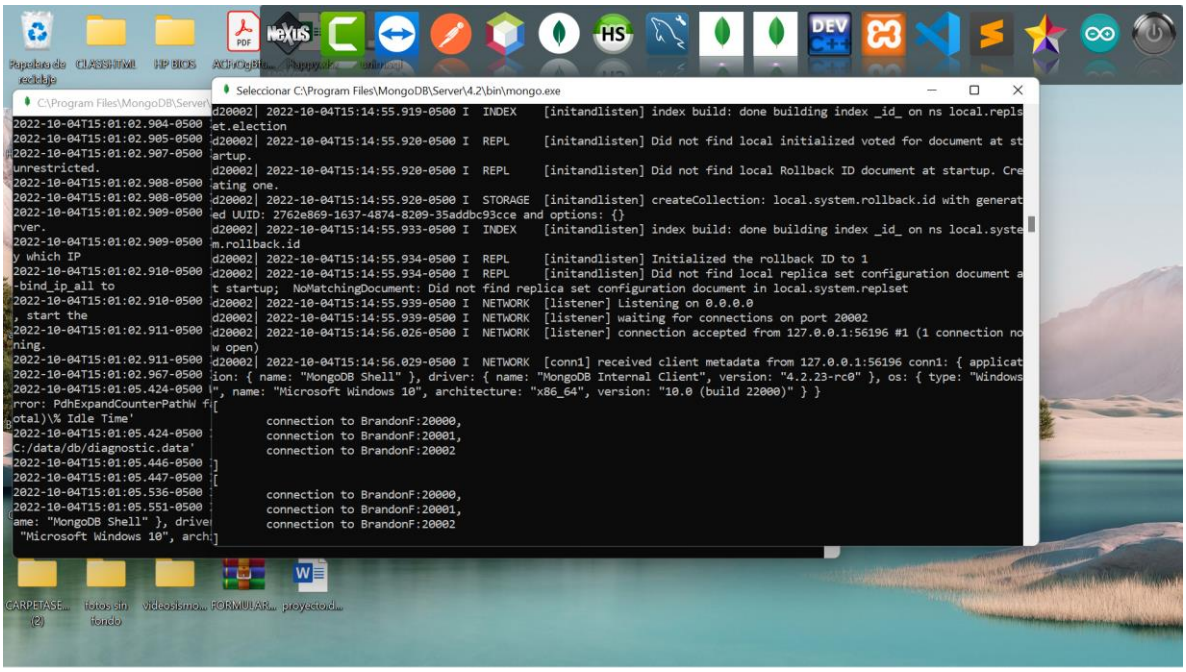
- La base de datos debe permitir realizar todas las operaciones de consulta necesarias.
- La base de datos debe estar disponible 24/7
- Se debe garantizar la seguridad de la información.
- Debe permitir el acceso de varios usuarios al mismo tiempo (concurrencia)
- Se debe garantizar la fiabilidad de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

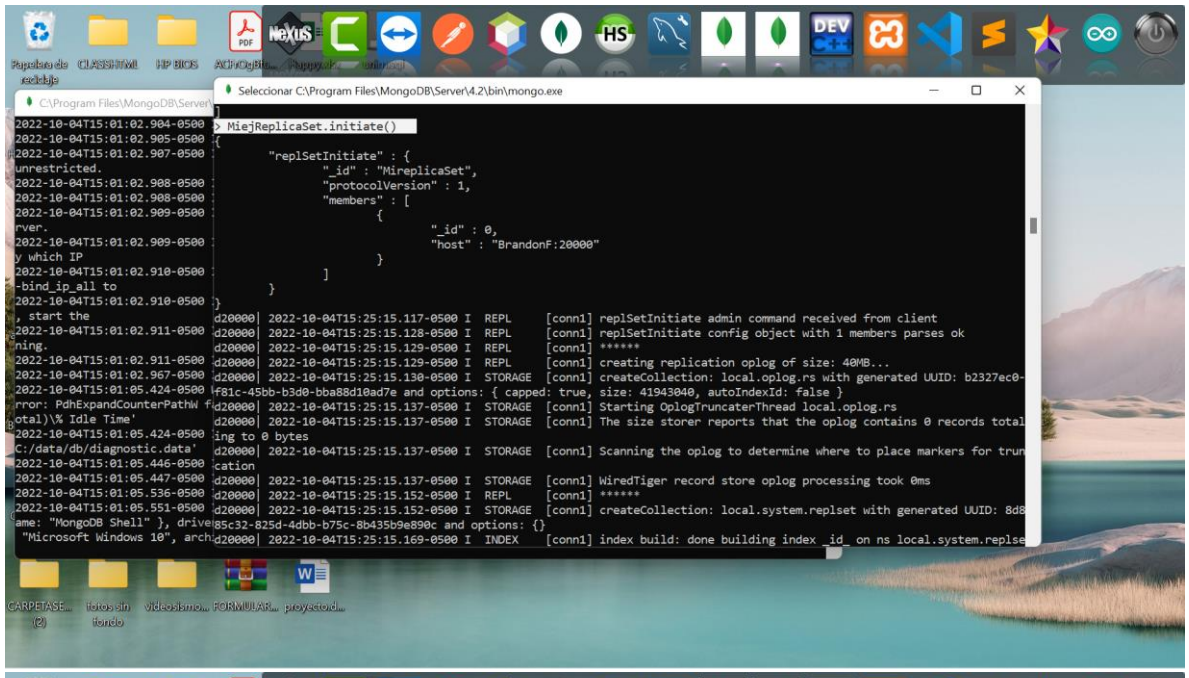

ReplicacionBDs_NoSQL.docx [Modo de compatibilidad] - Word
Brandon Franco
Archivo Inicio Insertar Diseño Disposición Referencias Correspondencia Revisar Vista Ayuda ¿Qué desea hacer?
Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
MiejReplicaSet = new ReplSetTest({name:"MireplicaSet",nodes:3})
Starting new replica set MireplicaSet
{
"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;
}
}
Paso 2 -Arrancar los procesos mongod de la réplica
Hasta ahora, sólo hemos configurado el grupo de réplica, vamos a arrancar la instancia de los nodos que componen la réplica. Para arrancar los nodos del grupo de réplica ejecutaremos la función startSet() sobre el objeto que representa el grupo de réplica.
Página 2 de 8 1305 palabras Español (España) Accesibilidad: es necesario investigar 100%
C:\Program Files\MongoDB\Server\4.2\bin\mongod.exe
2022-10-04T15:01:02.904-0500 I CONTROL [initandli
2022-10-04T15:01:02.905-0500 I CONTROL [initandli
2022-10-04T15:01:02.907-0500 I CONTROL [initandli
unrestricted.
2022-10-04T15:01:02.908-0500 I CONTROL [initandli
2022-10-04T15:01:02.908-0500 I CONTROL [initandli
2022-10-04T15:01:02.909-0500 I CONTROL [initandli
rver.
2022-10-04T15:01:02.909-0500 I CONTROL [initandli
y which IP
2022-10-04T15:01:02.910-0500 I CONTROL [initandli
-bind_ip_all to
2022-10-04T15:01:02.910-0500 I CONTROL [initandli
, start the
2022-10-04T15:01:02.911-0500 I CONTROL [initandli
ning.
2022-10-04T15:01:02.911-0500 I CONTROL [initandli
2022-10-04T15:01:02.967-0500 I STORAGE [initandli
2022-10-04T15:01:05.424-0500 W FTDC [initandli
rror: PdhExpandCounterPathW failed with 'El objeto
otal)\% Idle Time'
2022-10-04T15:01:05.424-0500 I FTDC [initandli
C:/data/db/diagnostic.data'
2022-10-04T15:01:05.446-0500 I NETWORK [listener]
2022-10-04T15:01:05.447-0500 I NETWORK [listener]
2022-10-04T15:01:05.536-0500 I NETWORK [listener]
2022-10-04T15:01:05.551-0500 I NETWORK [conn1] re
ame: "MongoDB Shell" }, driver: { name: "MongoDB In
"Microsoft Windows 10", architecture: "x86_64", ve
C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
},
"waitForMaster" : function(timeout) {
var master;
assert.soonNoExcept(function() {
return (master = self.getPrimary());
}, "waiting for master", timeout);
return master;
},
"name" : "MireplicaSet",
"useHostName" : true,
"host" : "BrandonF",
"oplogSize" : 40,
"useSeedList" : false,
"keyFile" : undefined,
"protocolVersion" : undefined,
"waitForKeys" : undefined,
"nodeOptions" : {
"n0" : undefined,
"n1" : undefined,
"n2" : undefined
},
"nodes" : [ ],
"ports" : [
20000,
20001,
20002
]
Escaneado con CamScanner
Página 2 de 2 1536 palabras Español (España) Accesibilidad: es necesario investigar 100%

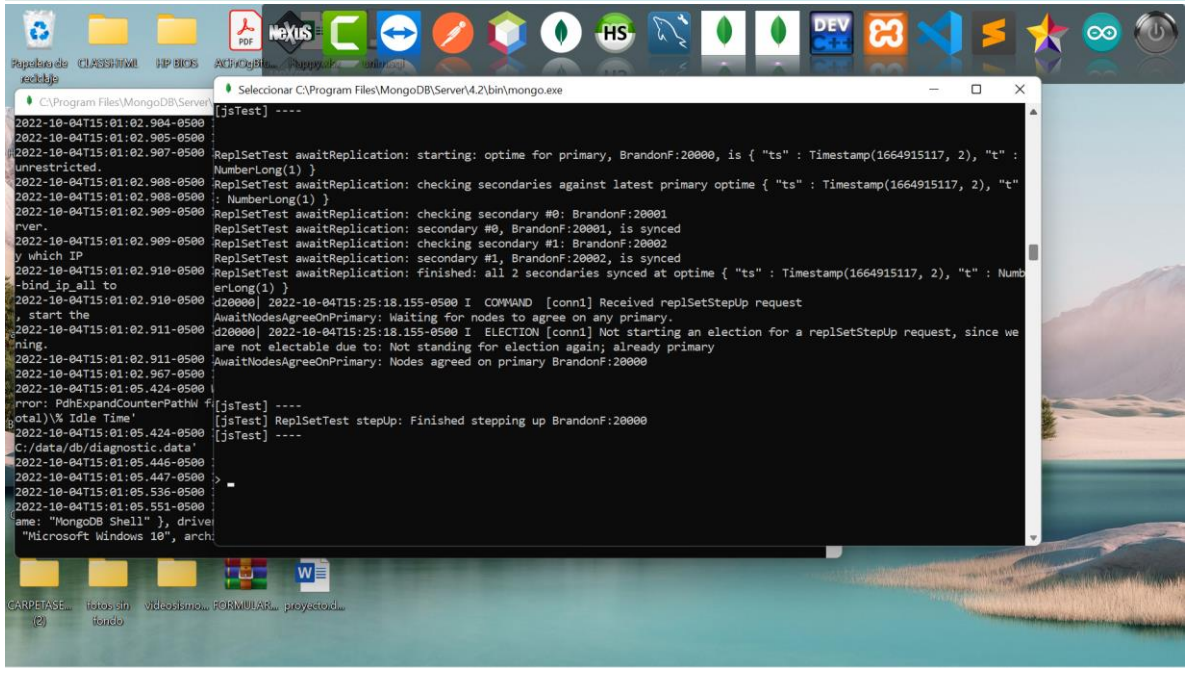

Documento1 - Word
Brandon Franco
Archivo
Inicio
Insertar
Diseño
Disposición
Referencias
Correspondencia
Revisar
Vista
Ayuda
¿Qué desea hacer?
Pegar
Cortar
Copiar
Copiar formato
Portapapeles
Buscar
Reemplazar
Seleccionar
Edición
Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> MiejReplicaSet.startSet()
Página 1 de 1
0 palabras
Español (México)
Accesibilidad: es necesario investigar
100%

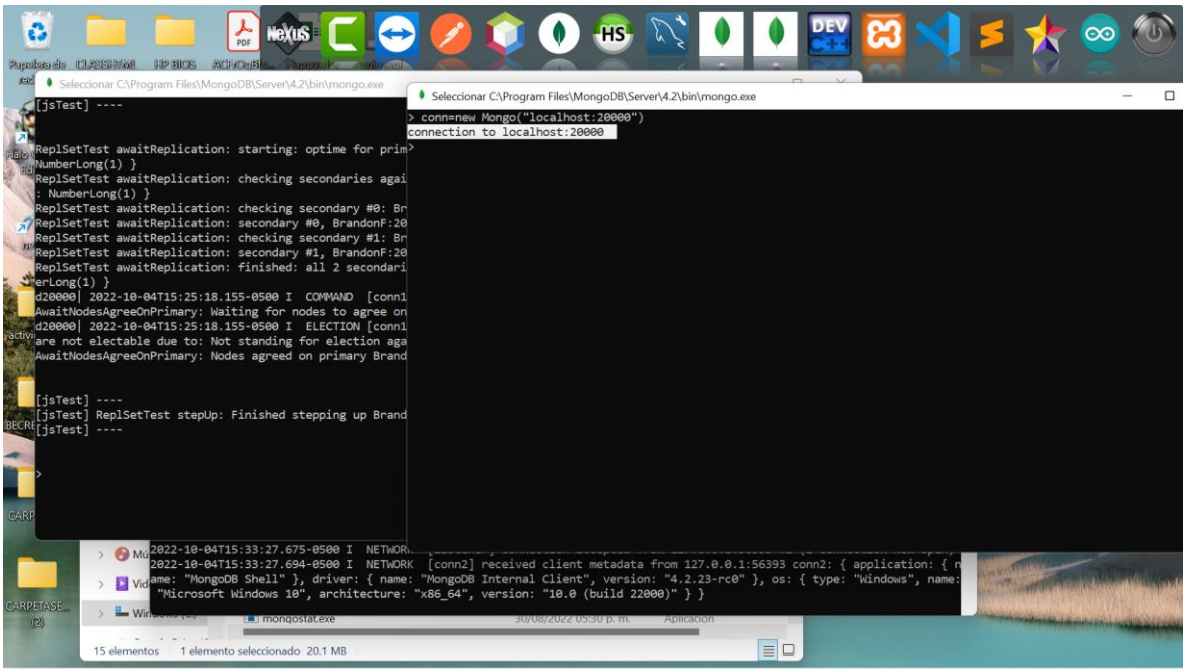

C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
ed UUID: 2762e869-1637-4874-8209-35addbc93cce and options: {}
d20002| 2022-10-04T15:14:55.933-0500 I INDEX [initandlisten] index build: done building index _id_ on ns local.syste
m.rollback.id
d20002| 2022-10-04T15:14:55.934-0500 I REPL [initandlisten] Initialized the rollback ID to 1
d20002| 2022-10-04T15:14:55.934-0500 I REPL [initandlisten] Did not find local replica set configuration document a
t startup; NoMatchingDocument: Did not find replica set configuration document in local.system.replset
d20002| 2022-10-04T15:14:55.939-0500 I NETWORK [listener] Listening on 0.0.0.0
d20002| 2022-10-04T15:14:55.939-0500 I NETWORK [listener] waiting for connections on port 20002
d20002| 2022-10-04T15:14:56.026-0500 I NETWORK [listener] connection accepted from 127.0.0.1:56196 #1 (1 connection no
w open)
d20002| 2022-10-04T15:14:56.029-0500 I NETWORK [conn1] received client metadata from 127.0.0.1:56196 conn1: { applicat
ion: { name: "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "4.2.23-rc0" }, os: { type: "Windows
", name: "Microsoft Windows 10", architecture: "x86_64", version: "10.0 (build 22000)" } }
[
connection to BrandonF:20000,
connection to BrandonF:20001,
connection to BrandonF:20002
]
[
connection to BrandonF:20000,
connection to BrandonF:20001,
connection to BrandonF:20002
]
>

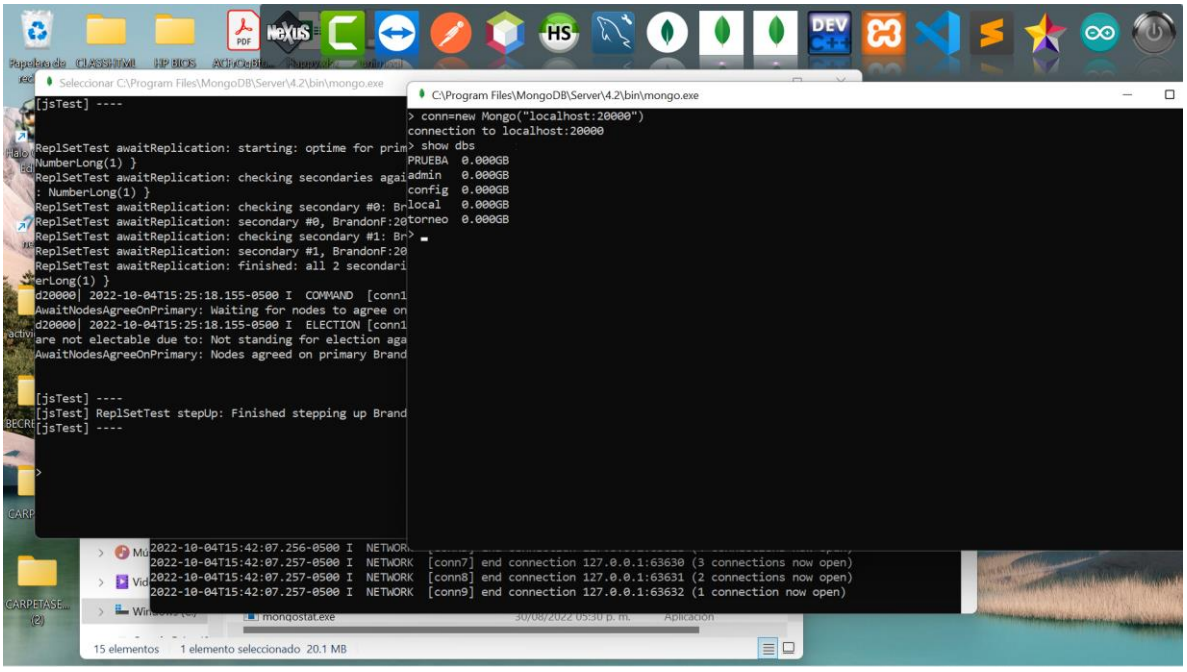

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
"ports" : [
20000,
20001,
20002
]
MiejReplicaSet.startSet()
ReplSetTest starting set
ReplSetTest n is : 0
{
"useHostName" : true,
"oplogSize" : 40,
"keyFile" : undefined,
"port" : 20000,
"replSet" : "MireplicaSet",
"dbpath" : "$set-$node",
"restart" : undefined,
"pathOpts" : {
"node" : 0,
"set" : "MireplicaSet"
},
"setParameter" : {
"writePeriodicNoops" : false,
"numInitialSyncConnectAttempts" : 60
}
}
ReplSetTest Starting....
Resetting db path '/data/db/MireplicaSet-0'
2022-10-04T15:14:49.223-0500 I - [js] shell: started program (sh27232): C:\Program Files\MongoDB\Server\4.2\bin\mongod.exe --oplogSize 40 --port 20000 --replSet MireplicaSet --dbpath /data/db/MireplicaSet-0 --setParameter writePeri

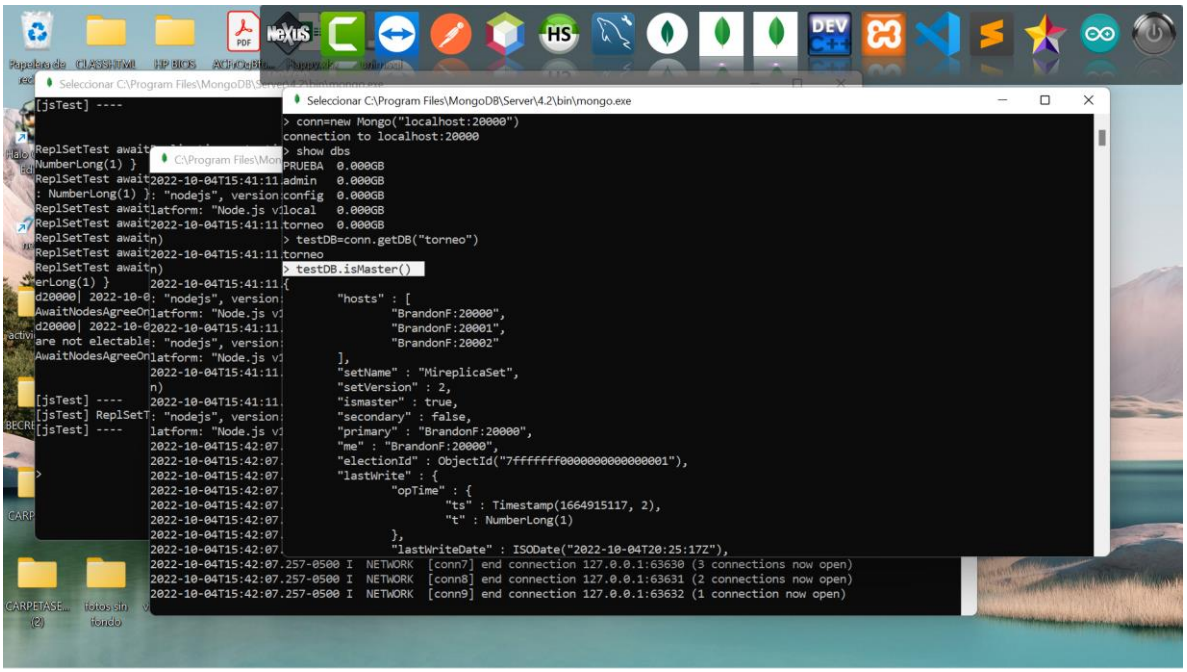

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
d20002| 2022-10-04T15:14:55.919-0500 I INDEX [initandlisten] index build: done building index _id_ on ns local.replset.election
d20002| 2022-10-04T15:14:55.920-0500 I REPL [initandlisten] Did not find local initialized voted for document at startup.
d20002| 2022-10-04T15:14:55.920-0500 I REPL [initandlisten] Did not find local Rollback ID document at startup. Creating one.
d20002| 2022-10-04T15:14:55.920-0500 I STORAGE [initandlisten] createCollection: local.system.rollback.id with generated UUID: 2762e869-1637-4874-8209-35addbc93cce and options: {}
d20002| 2022-10-04T15:14:55.933-0500 I INDEX [initandlisten] index build: done building index _id_ on ns local.system.rollback.id
d20002| 2022-10-04T15:14:55.934-0500 I REPL [initandlisten] Initialized the rollback ID to 1
d20002| 2022-10-04T15:14:55.934-0500 I REPL [initandlisten] Did not find local replica set configuration document at startup; NoMatchingDocument: Did not find replica set configuration document in local.system.replset
d20002| 2022-10-04T15:14:55.939-0500 I NETWORK [listener] Listening on 0.0.0.0
d20002| 2022-10-04T15:14:55.939-0500 I NETWORK [listener] waiting for connections on port 20002
d20002| 2022-10-04T15:14:56.026-0500 I NETWORK [listener] connection accepted from 127.0.0.1:56196 #1 (1 connection now open)
d20002| 2022-10-04T15:14:56.029-0500 I NETWORK [conn1] received client metadata from 127.0.0.1:56196 conn1: { application: { name: "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "4.2.23-rc0" }, os: { type: "Windows", name: "Microsoft Windows 10", architecture: "x86_64", version: "10.0 (build 22000)" } }
[
connection to BrandonF:20000,
connection to BrandonF:20001,
connection to BrandonF:20002
]
[
connection to BrandonF:20000,
connection to BrandonF:20001,
connection to BrandonF:20002
]

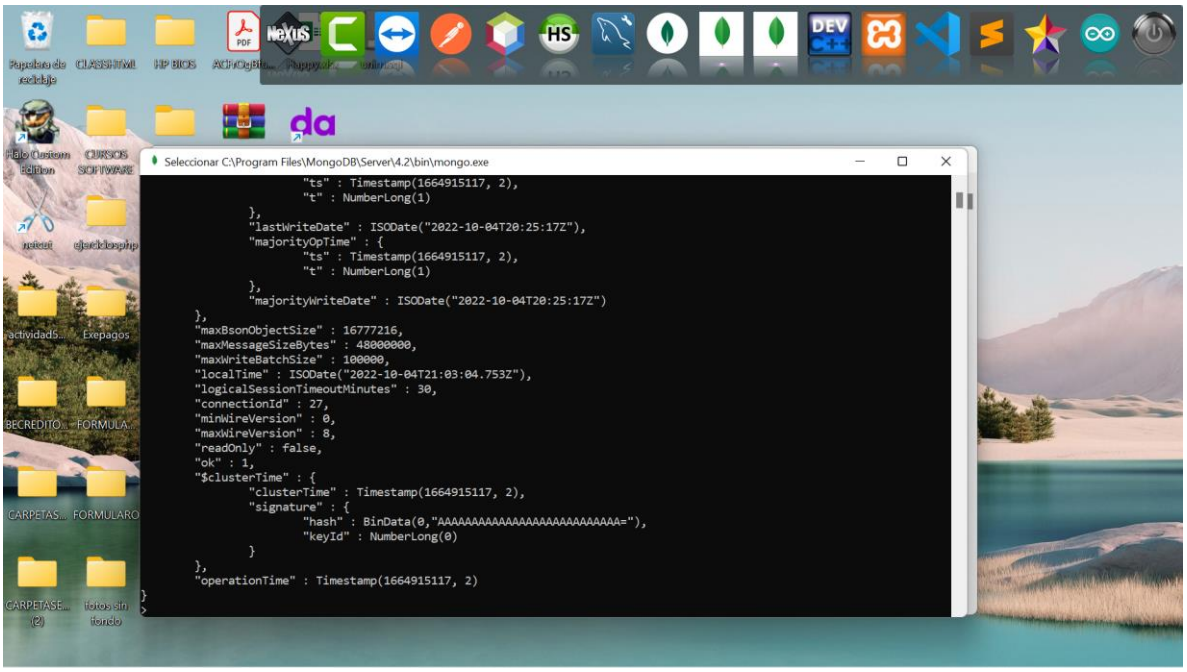

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> MiejReplicaSet.initiate()
{
"replSetInitiate" : {
"_id" : "MireplicaSet",
"protocolVersion" : 1,
"members" : [
{
"_id" : 0,
"host" : "BrandonF:20000"
}
]
}
}
d20000| 2022-10-04T15:25:15.117-0500 I REPL [conn1] replSetInitiate admin command received from client
d20000| 2022-10-04T15:25:15.128-0500 I REPL [conn1] replSetInitiate config object with 1 members parses ok
d20000| 2022-10-04T15:25:15.129-0500 I REPL [conn1] ******
d20000| 2022-10-04T15:25:15.129-0500 I REPL [conn1] creating replication oplog of size: 40MB...
d20000| 2022-10-04T15:25:15.130-0500 I STORAGE [conn1] createCollection: local.oplog.rs with generated UUID: b2327ec0-f81c-45bb-b3d0-bba88d10ad7e and options: { capped: true, size: 41943040, autoIndexId: false }
d20000| 2022-10-04T15:25:15.137-0500 I STORAGE [conn1] Starting OplogTruncaterThread local.oplog.rs
d20000| 2022-10-04T15:25:15.137-0500 I STORAGE [conn1] The size storer reports that the oplog contains 0 records totaling to 0 bytes
d20000| 2022-10-04T15:25:15.137-0500 I STORAGE [conn1] Scanning the oplog to determine where to place markers for truncation
d20000| 2022-10-04T15:25:15.137-0500 I STORAGE [conn1] WiredTiger record store oplog processing took 0ms
d20000| 2022-10-04T15:25:15.152-0500 I REPL [conn1] ******
d20000| 2022-10-04T15:25:15.152-0500 I STORAGE [conn1] createCollection: local.system.replset with generated UUID: 8d885c32-825d-4dbb-b75c-8b435b9e890c and options: {}
d20000| 2022-10-04T15:25:15.169-0500 I INDEX [conn1] index build: done building index _id_ on ns local.system.replse

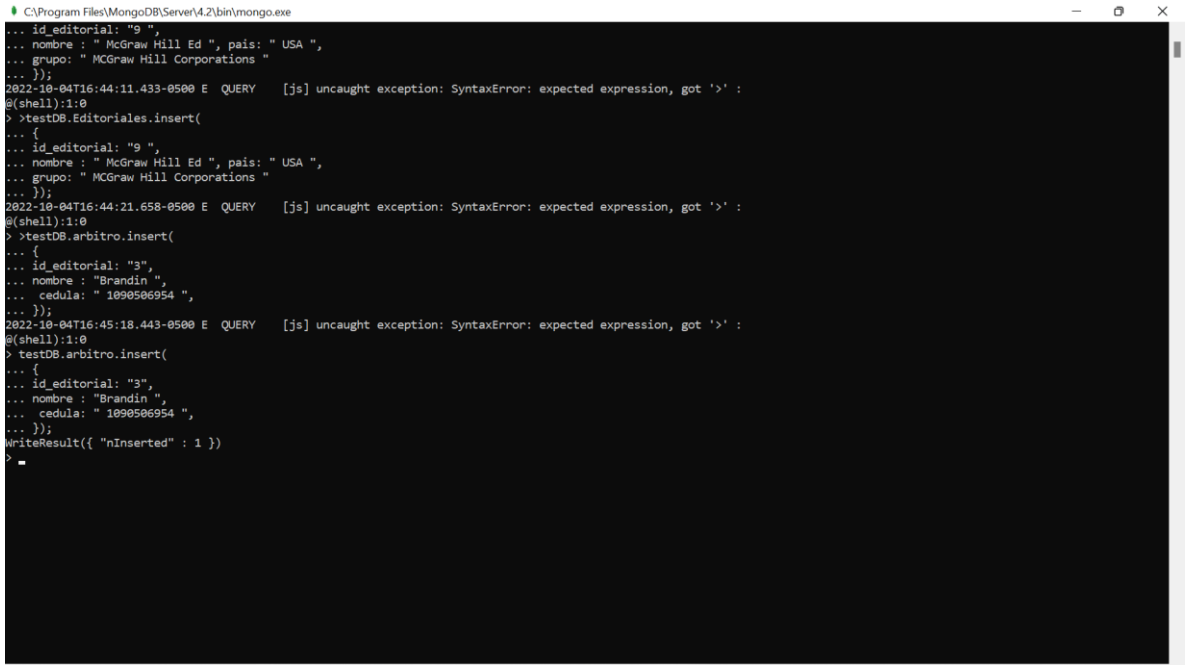

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
[jsTest] ----
ReplSetTest awaitReplication: starting: optime for primary, BrandonF:20000, is { "ts" : Timestamp(1664915117, 2), "t" : NumberLong(1) }
ReplSetTest awaitReplication: checking secondaries against latest primary optime { "ts" : Timestamp(1664915117, 2), "t" : NumberLong(1) }
ReplSetTest awaitReplication: checking secondary #0: BrandonF:20001
ReplSetTest awaitReplication: secondary #0, BrandonF:20001, is synced
ReplSetTest awaitReplication: checking secondary #1: BrandonF:20002
ReplSetTest awaitReplication: secondary #1, BrandonF:20002, is synced
ReplSetTest awaitReplication: finished: all 2 secondaries synced at optime { "ts" : Timestamp(1664915117, 2), "t" : NumberLong(1) }
d20000| 2022-10-04T15:25:18.155-0500 I COMMAND [conn1] Received replSetStepUp request
AwaitNodesAgreeOnPrimary: Waiting for nodes to agree on any primary.
d20000| 2022-10-04T15:25:18.155-0500 I ELECTION [conn1] Not starting an election for a replSetStepUp request, since we are not electable due to: Not standing for election again; already primary
AwaitNodesAgreeOnPrimary: Nodes agreed on primary BrandonF:20000
[jsTest] ----
[jsTest] ReplSetTest stepUp: Finished stepping up BrandonF:20000
[jsTest] ----
>

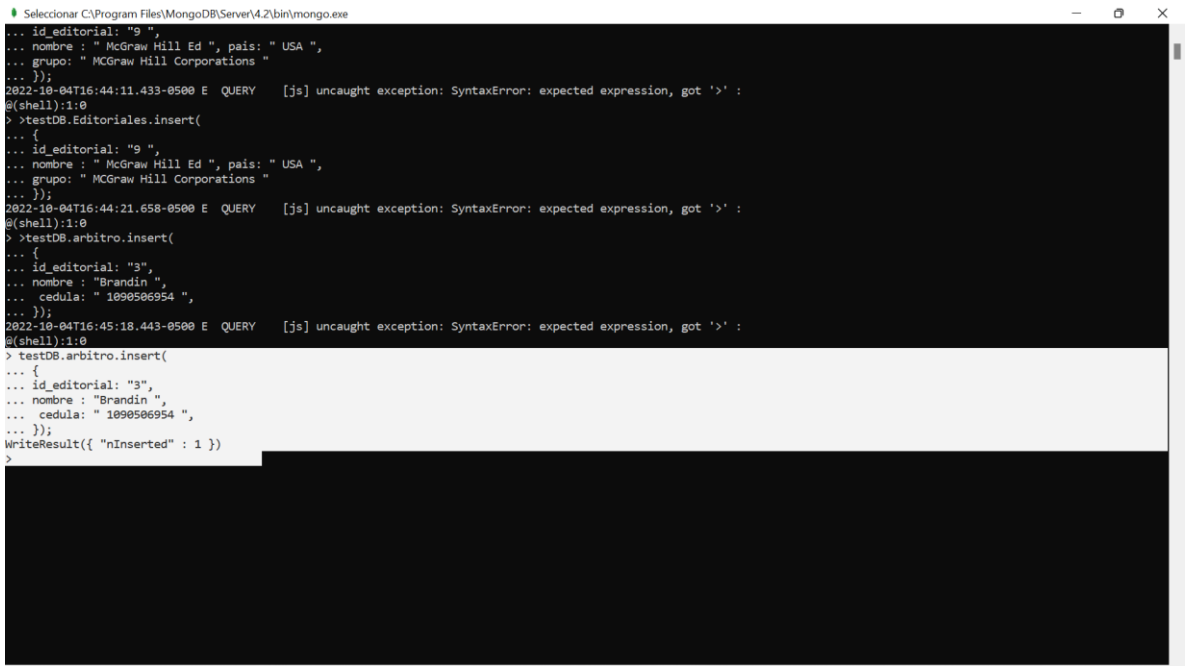

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> conn=new Mongo("localhost:20000")
connection to localhost:20000
[jsTest] ----
[jsTest] ReplSetTest stepUp: Finished stepping up Brand
[jsTest] ----
15 elementos
1 elemento seleccionado 20.1 MB

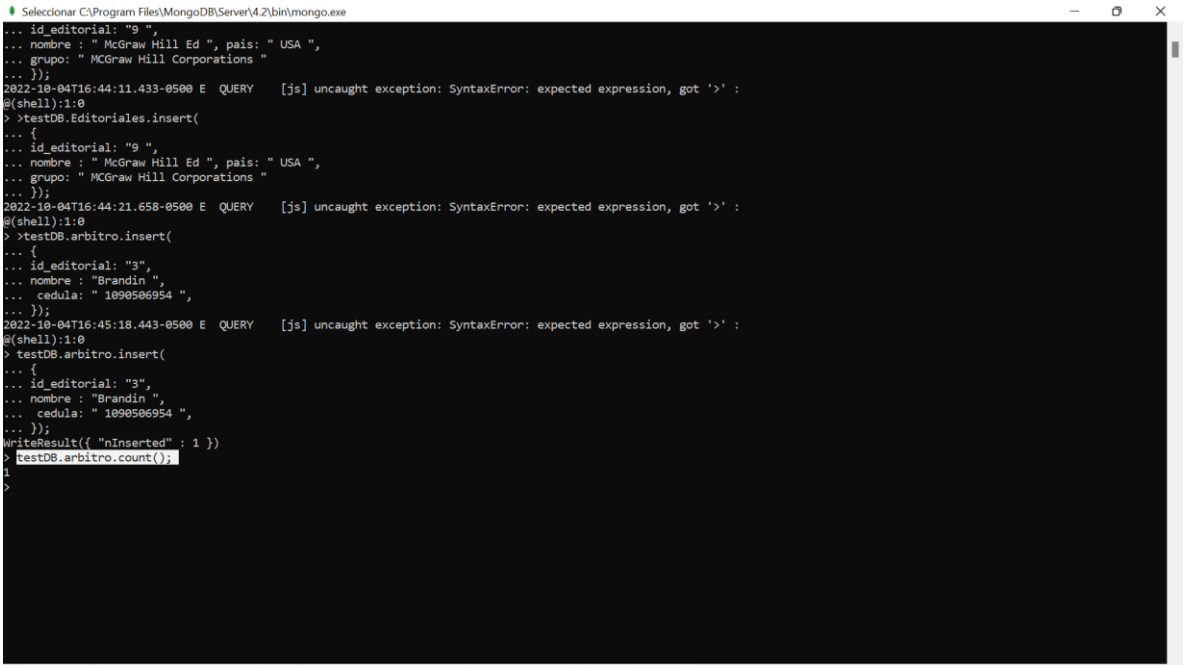

C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> conn=new Mongo("localhost:20000")
connection to localhost:20000
> show dbs
PRUEBA 0.000GB
admin 0.000GB
config 0.000GB
local 0.000GB
torneo 0.000GB
[jsTest] ----
[jsTest] ReplSetTest stepUp: Finished stepping up Brand
[jsTest] ----
15 elementos
1 elemento seleccionado 20.1 MB

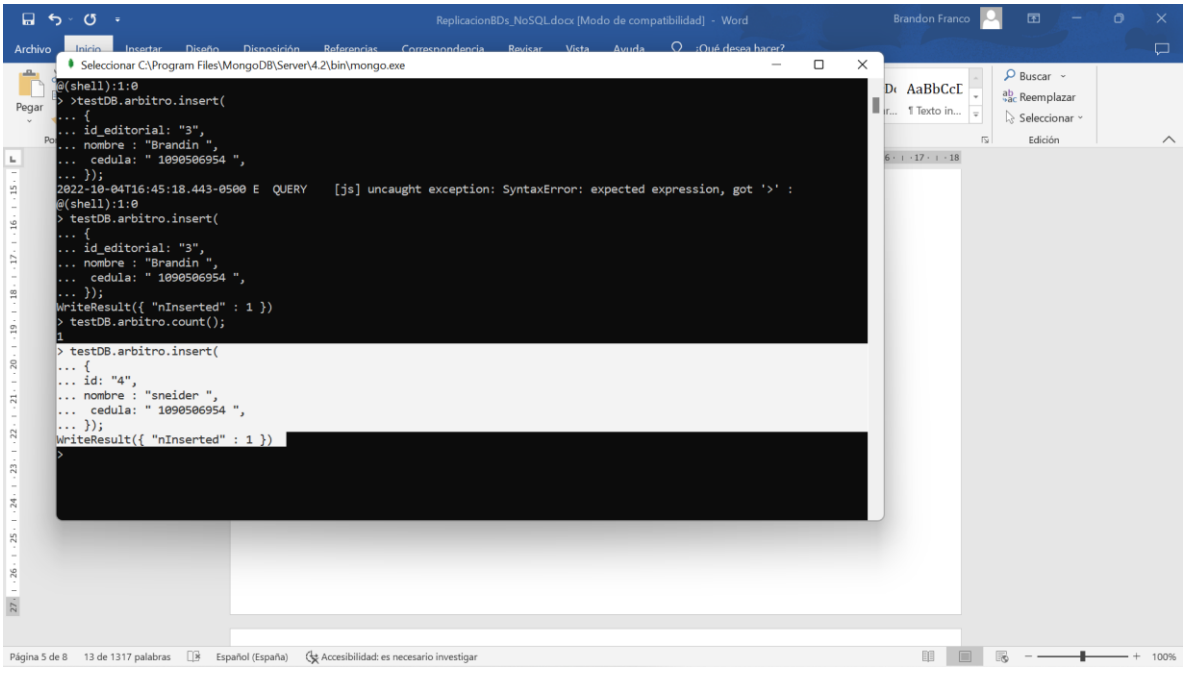

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> conn=new Mongo("localhost:20000")
connection to localhost:20000
> show dbs
PRUEBA  0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
torneo  0.000GB
> testDB=conn.getDB("torneo")
torneo
> testDB.isMaster()
{
"hosts" : [
"BrandonF:20000",
"BrandonF:20001",
"BrandonF:20002"
],
"setName" : "MireplicaSet",
"setVersion" : 2,
"ismaster" : true,
"secondary" : false,
"primary" : "BrandonF:20000",
"me" : "BrandonF:20000",
"electionId" : ObjectId("7fffffff0000000000000001"),
"lastWrite" : {
"opTime" : {
"ts" : Timestamp(1664915117, 2),
"t" : NumberLong(1)
},
"lastWriteDate" : ISODate("2022-10-04T20:25:17Z"),
2022-10-04T15:42:07.257-0500 I NETWORK [conn7] end connection 127.0.0.1:63630 (3 connections now open)
2022-10-04T15:42:07.257-0500 I NETWORK [conn8] end connection 127.0.0.1:63631 (2 connections now open)
2022-10-04T15:42:07.257-0500 I NETWORK [conn9] end connection 127.0.0.1:63632 (1 connection now open)

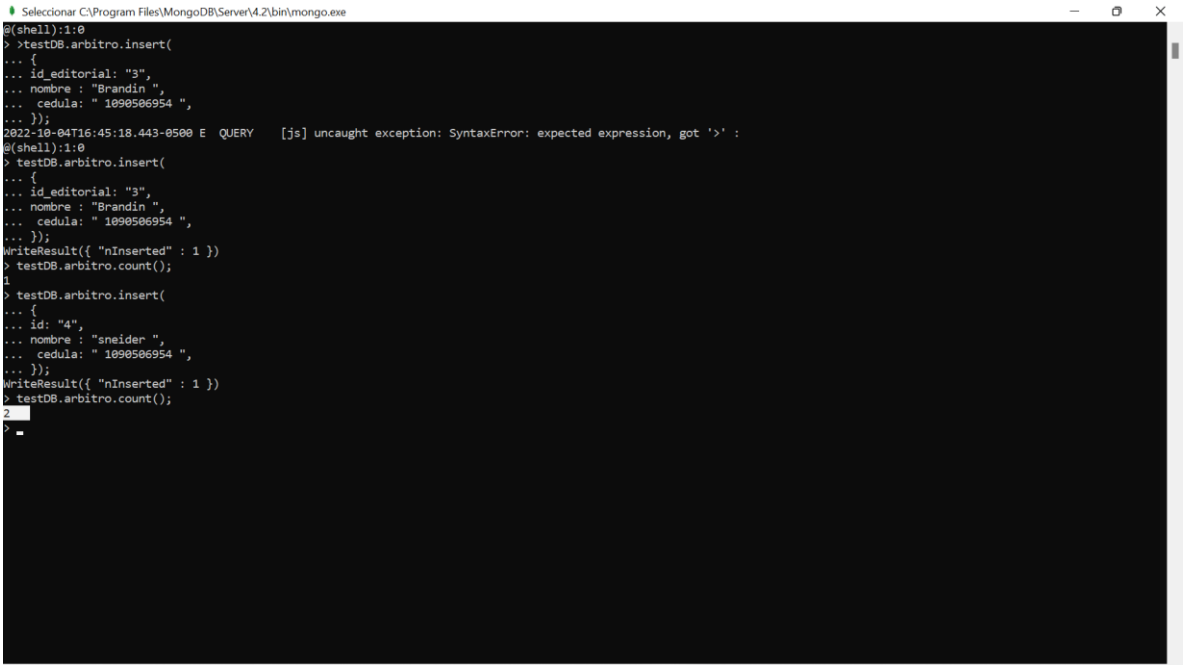

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
"ts" : Timestamp(1664915117, 2),
"t" : NumberLong(1)
},
"lastWriteDate" : ISODate("2022-10-04T20:25:17Z"),
"majorityOpTime" : {
"ts" : Timestamp(1664915117, 2),
"t" : NumberLong(1)
},
"majorityWriteDate" : ISODate("2022-10-04T20:25:17Z")
},
"maxBsonObjectSize" : 16777216,
"maxMessageSizeBytes" : 48000000,
"maxWriteBatchSize" : 100000,
"localTime" : ISODate("2022-10-04T21:03:04.753Z"),
"logicalSessionTimeoutMinutes" : 30,
"connectionId" : 27,
"minWireVersion" : 0,
"maxWireVersion" : 8,
"readOnly" : false,
"ok" : 1,
"$clusterTime" : {
"clusterTime" : Timestamp(1664915117, 2),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
},
"operationTime" : Timestamp(1664915117, 2)
}

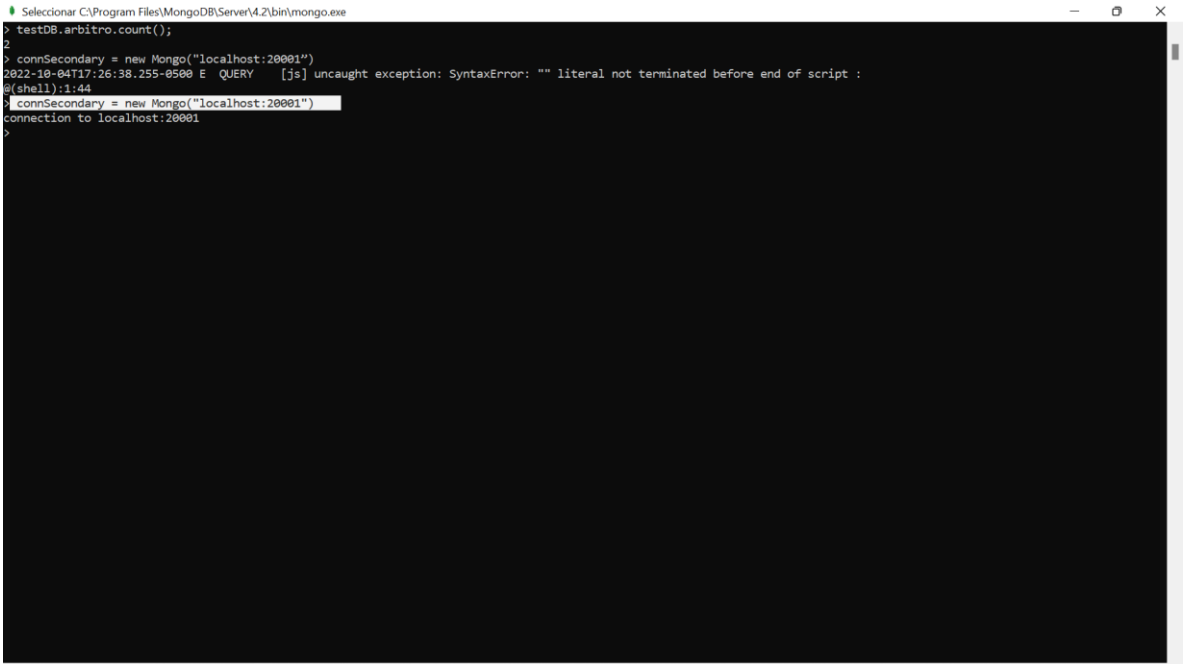

C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
... id_editorial: "9 ",
... nombre : " McGraw Hill Ed ", pais: " USA ",
... grupo: " MCGraw Hill Corporations "
... });
2022-10-04T16:44:11.433-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> >testDB.Editoriales.insert(
... {
... id_editorial: "9 ",
... nombre : " McGraw Hill Ed ", pais: " USA ",
... grupo: " MCGraw Hill Corporations "
... });
2022-10-04T16:44:21.658-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> >testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
... cedula: " 1090506954 ",
... });
2022-10-04T16:45:18.443-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
... cedula: " 1090506954 ",
... });
WriteResult({ "nInserted" : 1 })
>


Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
... id_editorial: "9 ",
... nombre : " McGraw Hill Ed ", pais: " USA ",
... grupo: " MCGraw Hill Corporations "
... });
2022-10-04T16:44:11.433-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> >testDB.Editoriales.insert(
... {
... id_editorial: "9 ",
... nombre : " McGraw Hill Ed ", pais: " USA ",
... grupo: " MCGraw Hill Corporations "
... });
2022-10-04T16:44:21.658-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> >testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
... cedula: " 1090506954 ",
... });
2022-10-04T16:45:18.443-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
... cedula: " 1090506954 ",
... });
WriteResult({ "nInserted" : 1 })
>

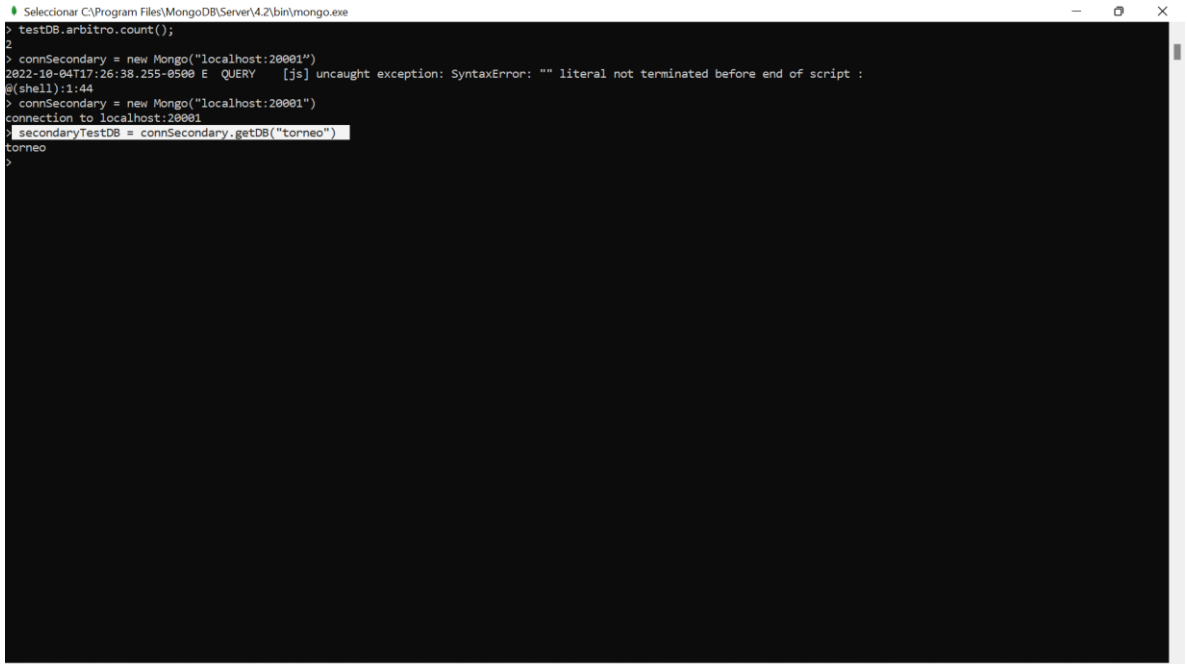

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
... id_editorial: "9 ",
... nombre : " McGraw Hill Ed ", pais: " USA ",
... grupo: " MCGraw Hill Corporations "
... });
2022-10-04T16:44:11.433-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> >testDB.Editoriales.insert(
... {
... id_editorial: "9 ",
... nombre : " McGraw Hill Ed ", pais: " USA ",
... grupo: " MCGraw Hill Corporations "
... });
2022-10-04T16:44:21.658-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> >testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
... cedula: " 1090506954 ",
... });
2022-10-04T16:45:18.443-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
... cedula: " 1090506954 ",
... });
WriteResult({ "nInserted" : 1 })
> testDB.arbitro.count();
1
>


ReplicacionBDs_NoSQL.docx [Modo de compatibilidad] - Word
Brandon Franco
Archivo
Inicio
Insertar
Diseño
Disposición
Referencias
Correspondencia
Revisar
Vista
Ayuda
¿Qué desea hacer?
Pegar
Buscar
Reemplazar
Seleccionar
Edición
Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
@(shell):1:0
> >testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
... cedula: " 1090506954 ",
... });
2022-10-04T16:45:18.443-0500 E QUERY [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
... cedula: " 1090506954 ",
... });
WriteResult({ "nInserted" : 1 })
> testDB.arbitro.count();
1
> testDB.arbitro.insert(
... {
... id: "4",
... nombre : "sneider ",
... cedula: " 1090506954 ",
... });
WriteResult({ "nInserted" : 1 })
>
Página 5 de 8
13 de 1317 palabras
Español (España)
Accesibilidad: es necesario investigar
100%

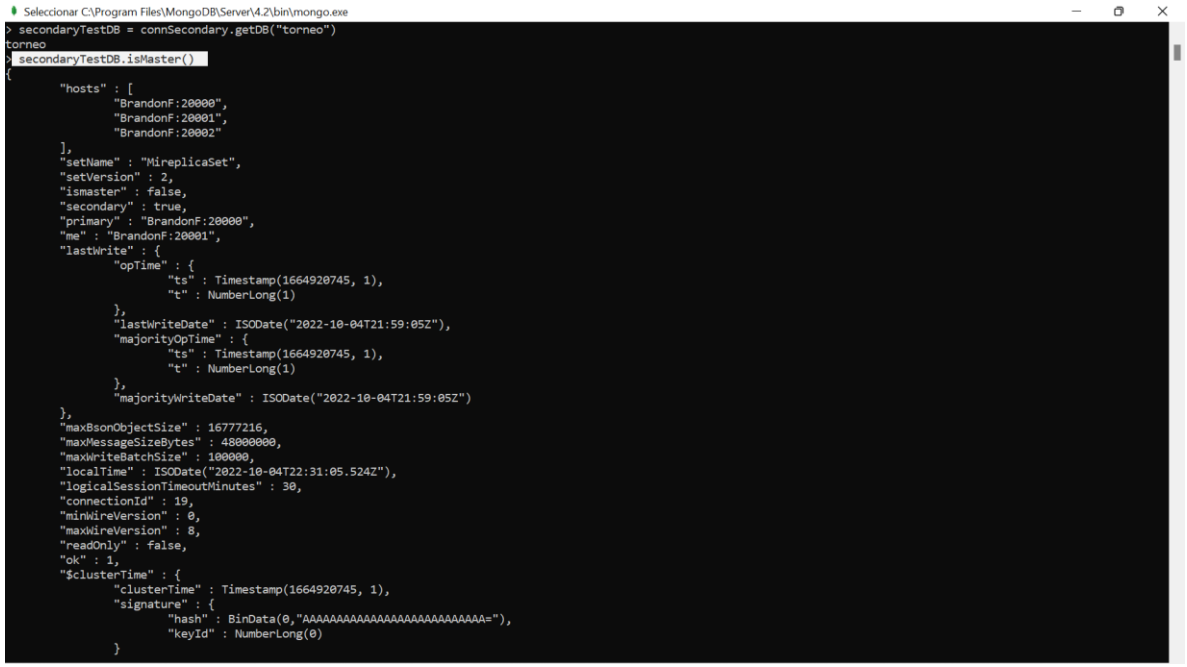

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
@(shell):1:0
> >testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
...  cedula: " 1090506954 ",
... });
2022-10-04T16:45:18.443-0500 E  QUERY    [js] uncaught exception: SyntaxError: expected expression, got '>' :
@(shell):1:0
> testDB.arbitro.insert(
... {
... id_editorial: "3",
... nombre : "Brandin ",
...  cedula: " 1090506954 ",
... });
WriteResult({ "nInserted" : 1 })
> testDB.arbitro.count();
1
> testDB.arbitro.insert(
... {
... id: "4",
... nombre : "sneider ",
...  cedula: " 1090506954 ",
... });
WriteResult({ "nInserted" : 1 })
> testDB.arbitro.count();
2
>

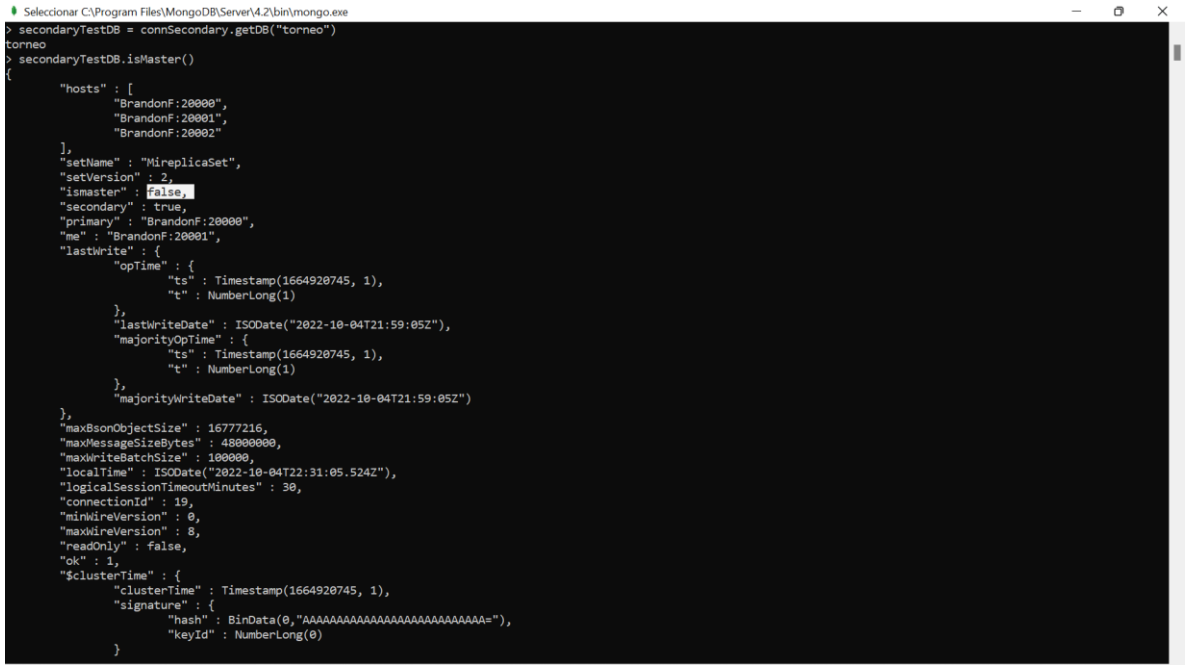

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> testDB.arbitro.count();
2
> connSecondary = new Mongo("localhost:20001”)
2022-10-04T17:26:38.255-0500 E  QUERY    [js] uncaught exception: SyntaxError: "" literal not terminated before end of script :
@(shell):1:44
> connSecondary = new Mongo("localhost:20001")
connection to localhost:20001
>

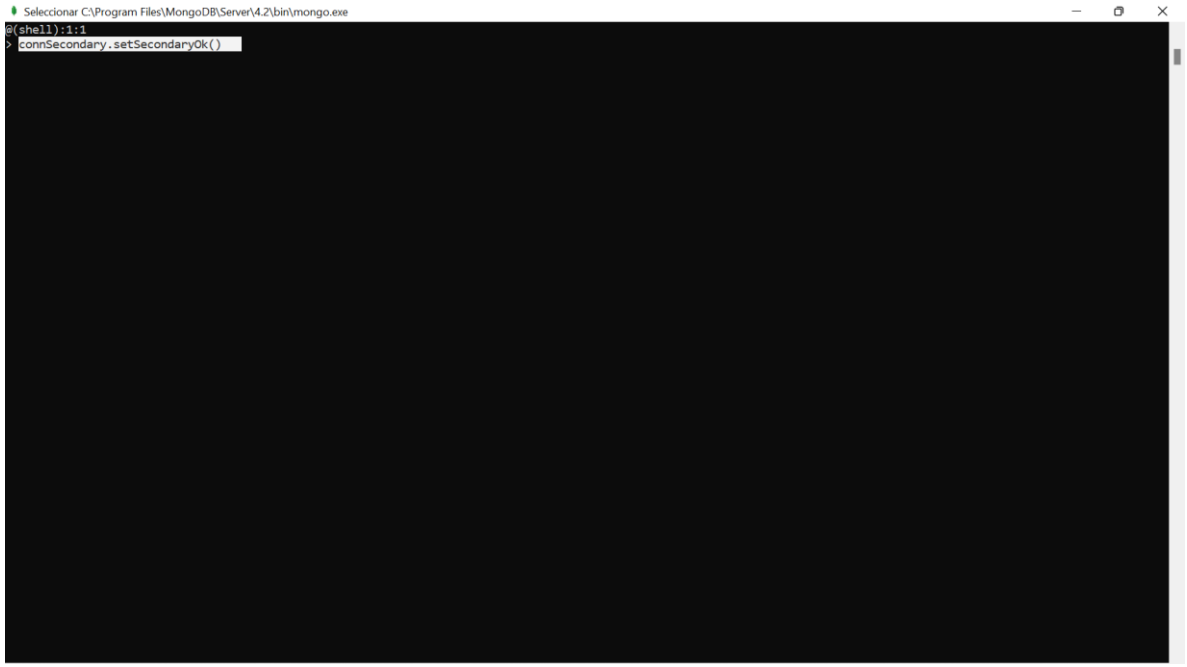

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> testDB.arbitro.count();
2
> connSecondary = new Mongo("localhost:20001")
2022-10-04T17:26:38.255-0500 E QUERY [js] uncaught exception: SyntaxError: "" literal not terminated before end of script :
@(shell):1:44
> connSecondary = new Mongo("localhost:20001")
connection to localhost:20001
>

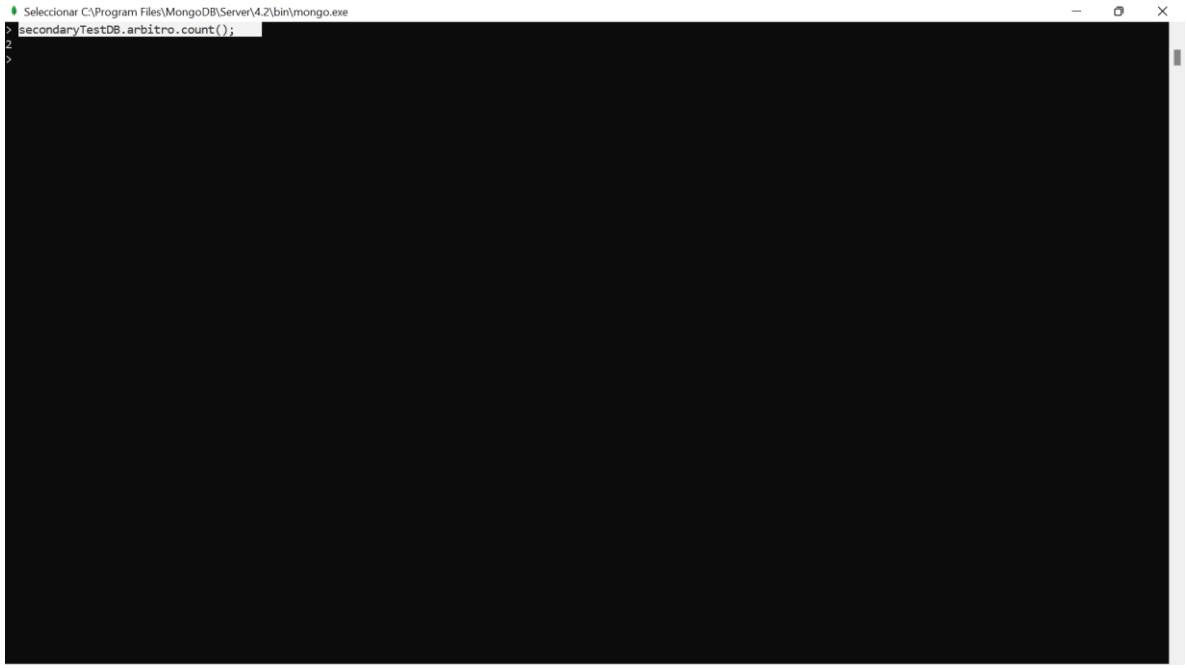

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> testDB.arbitro.count();
2
> connSecondary = new Mongo("localhost:20001")
2022-10-04T17:26:38.255-0500 E QUERY [js] uncaught exception: SyntaxError: "" literal not terminated before end of script :
@(shell):1:44
> connSecondary = new Mongo("localhost:20001")
connection to localhost:20001
> secondaryTestDB = connSecondary.getDB("torneo")
torneo
>

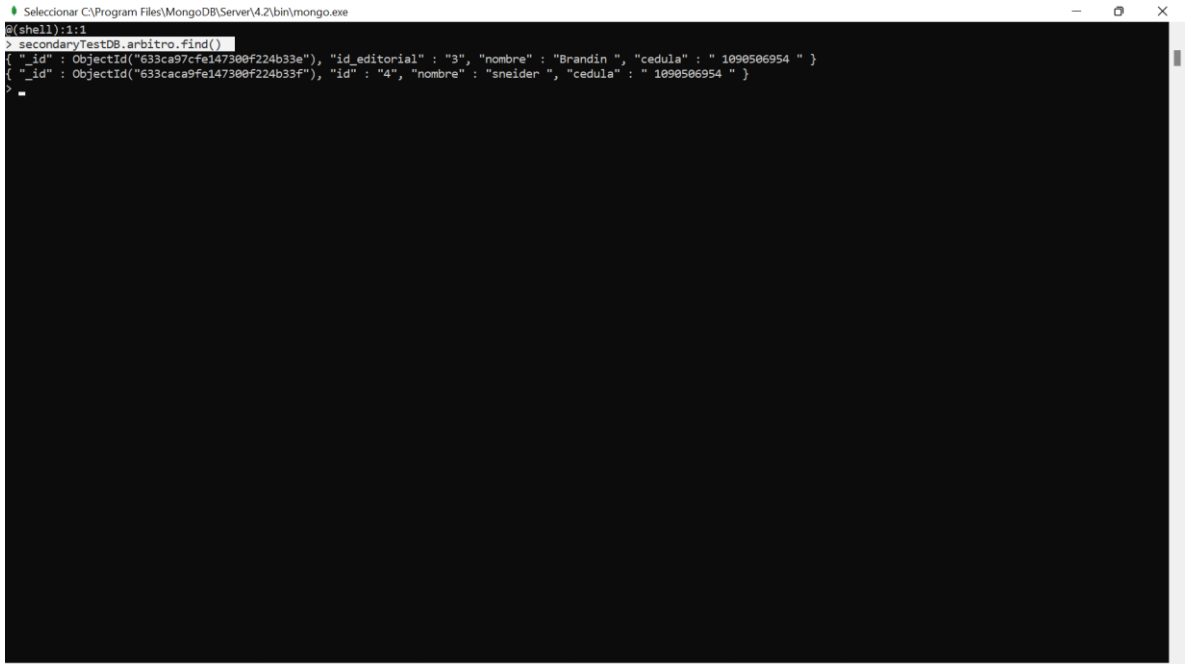

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> testDB.arbitro.count();
2
> connSecondary = new Mongo("localhost:20001")
2022-10-04T17:26:38.255-0500 E  QUERY    [js] uncaught exception: SyntaxError: "" literal not terminated before end of script :
@(shell):1:44
> connSecondary = new Mongo("localhost:20001")
connection to localhost:20001
> secondaryTestDB = connSecondary.getDB("torneo")
torneo
>

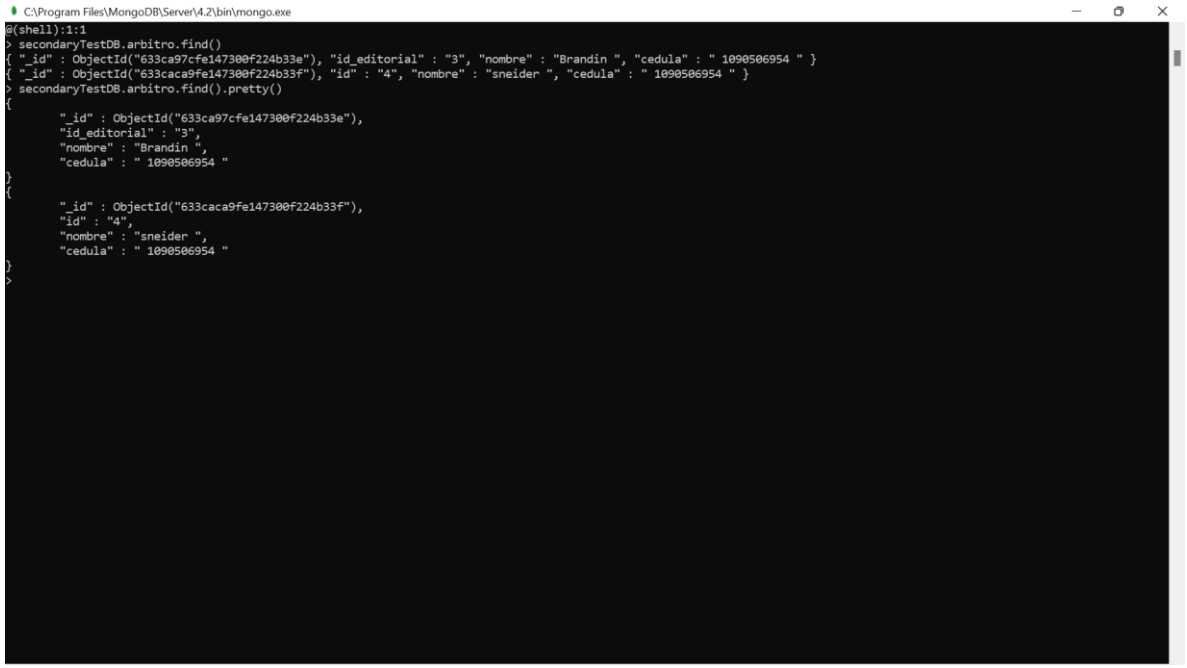

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> secondaryTestDB = connSecondary.getDB("torneo")
torneo
> secondaryTestDB.isMaster()
{
"hosts" : [
"BrandonF:20000",
"BrandonF:20001",
"BrandonF:20002"
],
"setName" : "MireplicaSet",
"setVersion" : 2,
"ismaster" : false,
"secondary" : true,
"primary" : "BrandonF:20000",
"me" : "BrandonF:20001",
"lastWrite" : {
"opTime" : {
"ts" : Timestamp(1664920745, 1),
"t" : NumberLong(1)
},
"lastWriteDate" : ISODate("2022-10-04T21:59:05Z"),
"majorityOpTime" : {
"ts" : Timestamp(1664920745, 1),
"t" : NumberLong(1)
},
"majorityWriteDate" : ISODate("2022-10-04T21:59:05Z")
},
"maxBsonObjectSize" : 16777216,
"maxMessageSizeBytes" : 48000000,
"maxWriteBatchSize" : 100000,
"localTime" : ISODate("2022-10-04T22:31:05.524Z"),
"logicalSessionTimeoutMinutes" : 30,
"connectionId" : 19,
"minWireVersion" : 0,
"maxWireVersion" : 8,
"readOnly" : false,
"ok" : 1,
"$clusterTime" : {
"clusterTime" : Timestamp(1664920745, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}

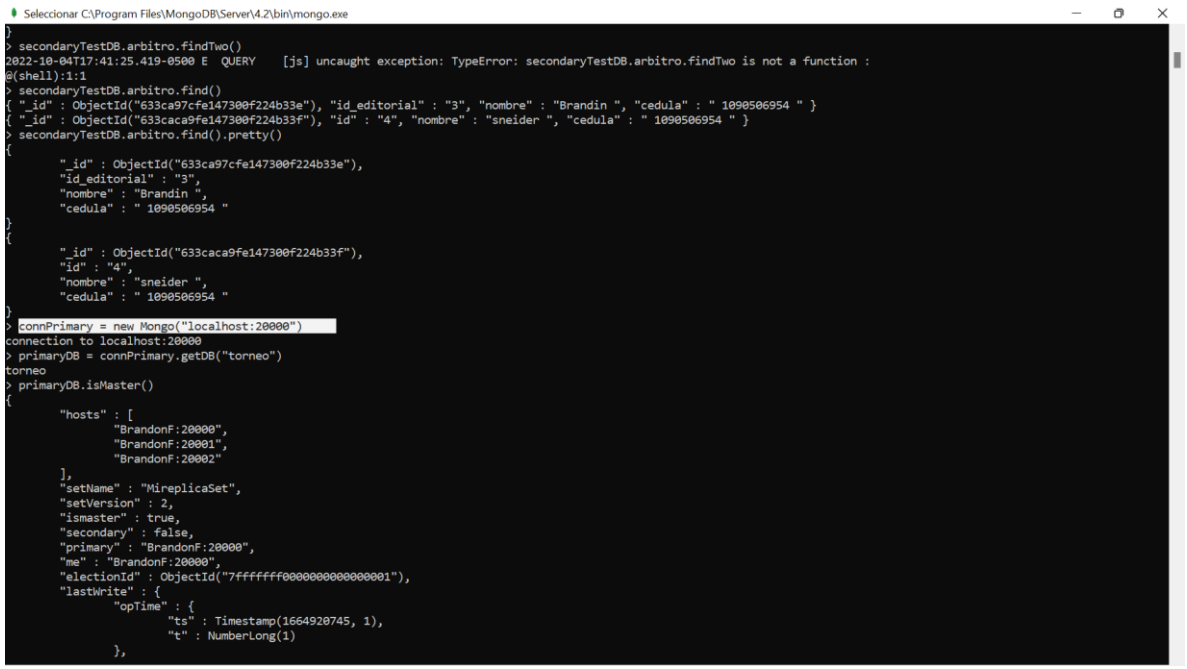

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> secondaryTestDB = connSecondary.getDB("torneo")
torneo
> secondaryTestDB.isMaster()
{
"hosts" : [
"BrandonF:20000",
"BrandonF:20001",
"BrandonF:20002"
],
"setName" : "MireplicaSet",
"setVersion" : 2,
"ismaster" : false,
"secondary" : true,
"primary" : "BrandonF:20000",
"me" : "BrandonF:20001",
"lastWrite" : {
"opTime" : {
"ts" : Timestamp(1664920745, 1),
"t" : NumberLong(1)
},
"lastWriteDate" : ISODate("2022-10-04T21:59:05Z"),
"majorityOpTime" : {
"ts" : Timestamp(1664920745, 1),
"t" : NumberLong(1)
},
"majorityWriteDate" : ISODate("2022-10-04T21:59:05Z")
},
"maxBsonObjectSize" : 16777216,
"maxMessageSizeBytes" : 48000000,
"maxWriteBatchSize" : 100000,
"localTime" : ISODate("2022-10-04T22:31:05.524Z"),
"logicalSessionTimeoutMinutes" : 30,
"connectionId" : 19,
"minWireVersion" : 0,
"maxWireVersion" : 8,
"readOnly" : false,
"ok" : 1,
"$clusterTime" : {
"clusterTime" : Timestamp(1664920745, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}

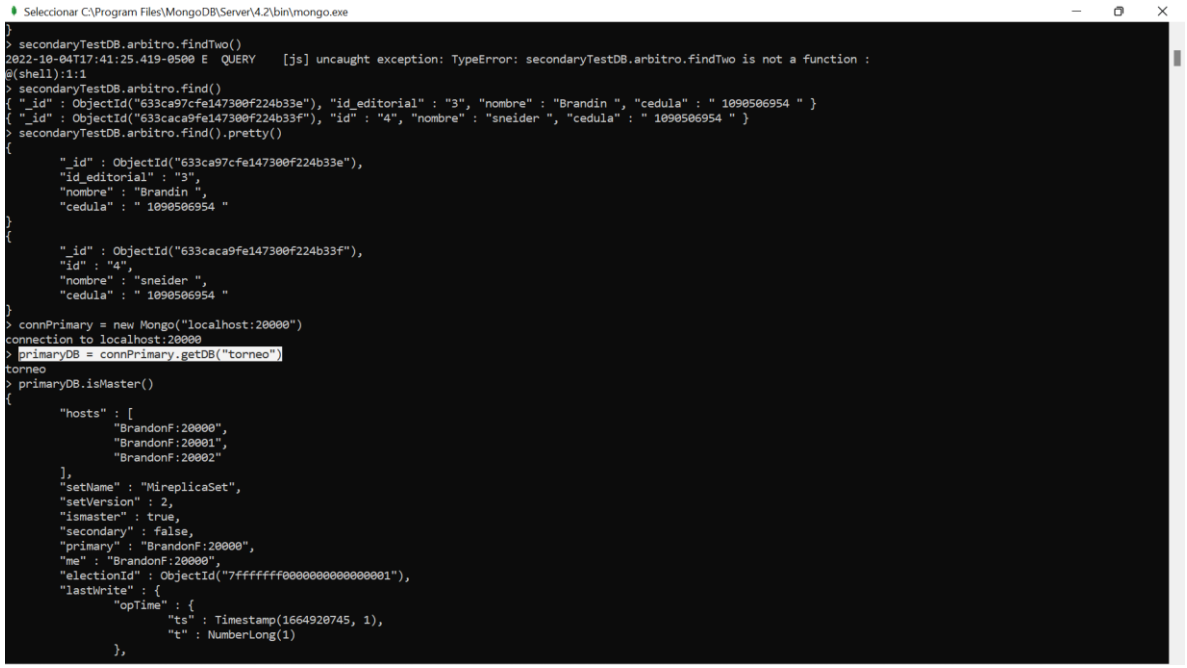

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
@(shell):1:1
> connSecondary.setSecondaryOk()

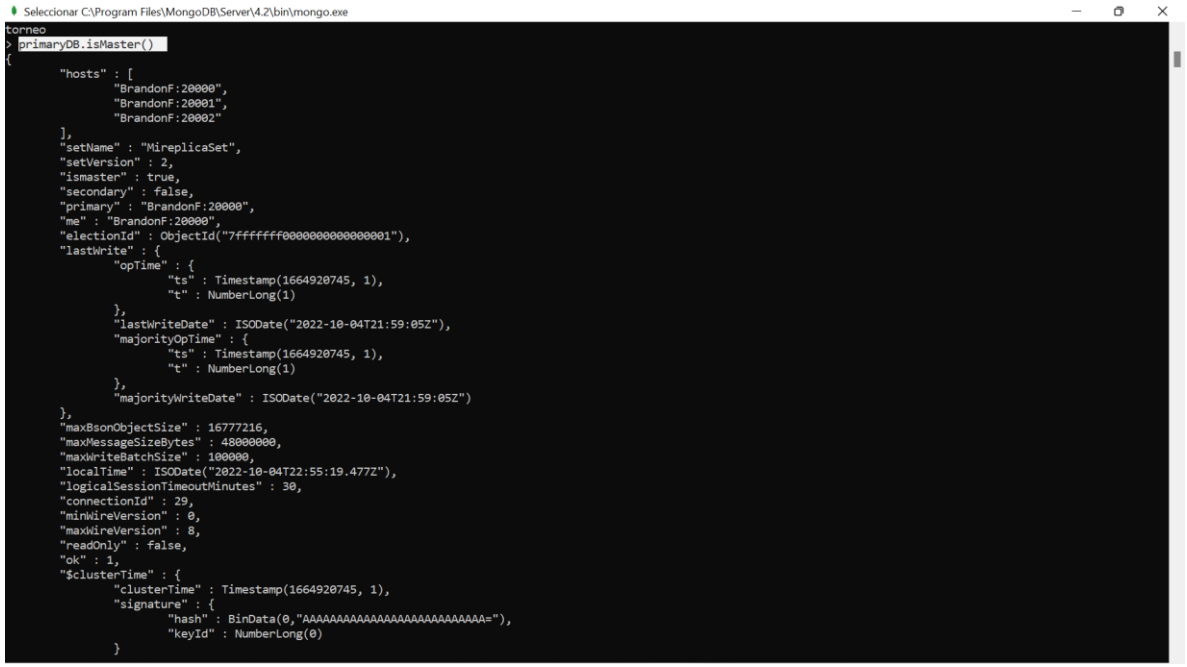

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> secondaryTestDB.arbitro.count();
2
>

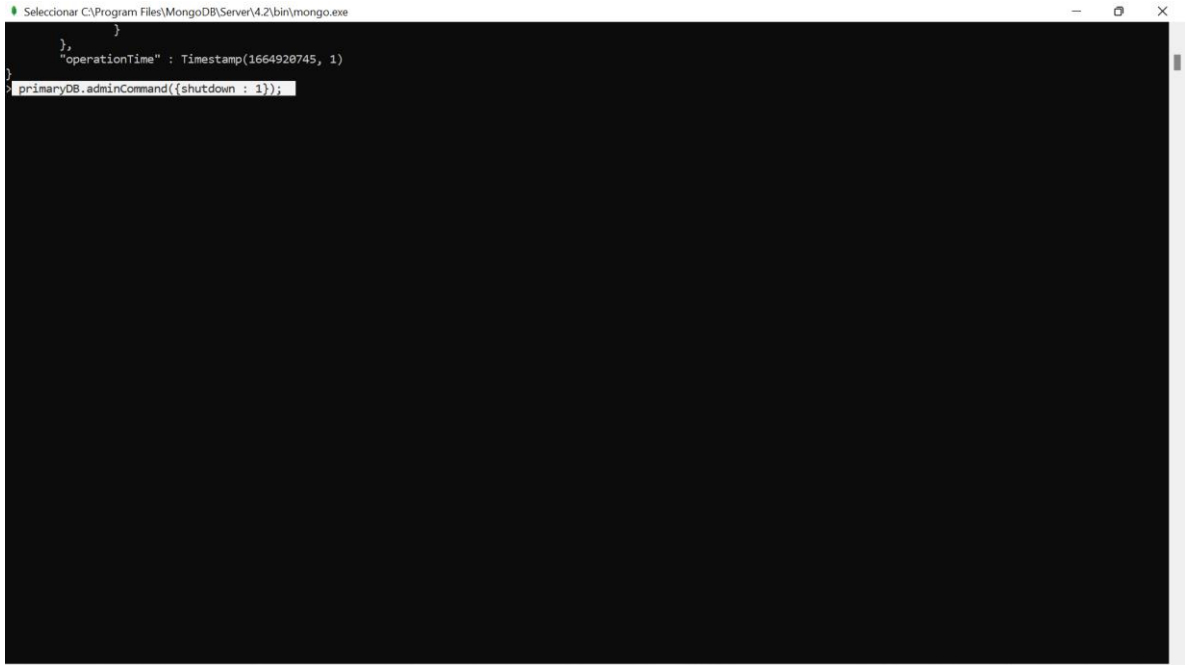

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
@(shell):1:1
> secondaryTestDB.arbitro.find()
{ "_id" : ObjectId("633ca97cfe147300f224b33e"), "id_editorial" : "3", "nombre" : "Brandin ", "cedula" : " 1090506954 " }
{ "_id" : ObjectId("633caca9fe147300f224b33f"), "id" : "4", "nombre" : "sneider ", "cedula" : " 1090506954 " }
>


C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
@(shell):1:1
> secondaryTestDB.arbitro.find()
{ "_id" : ObjectId("633ca97cfe147300f224b33e"), "id_editorial" : "3", "nombre" : "Brandin ", "cedula" : " 1090506954 " }
{ "_id" : ObjectId("633caca9fe147300f224b33f"), "id" : "4", "nombre" : "sneider ", "cedula" : " 1090506954 " }
> secondaryTestDB.arbitro.find().pretty()
{
"_id" : ObjectId("633ca97cfe147300f224b33e"),
"id_editorial" : "3",
"nombre" : "Brandin ",
"cedula" : " 1090506954 "
}
{
"_id" : ObjectId("633caca9fe147300f224b33f"),
"id" : "4",
"nombre" : "sneider ",
"cedula" : " 1090506954 "
}
>

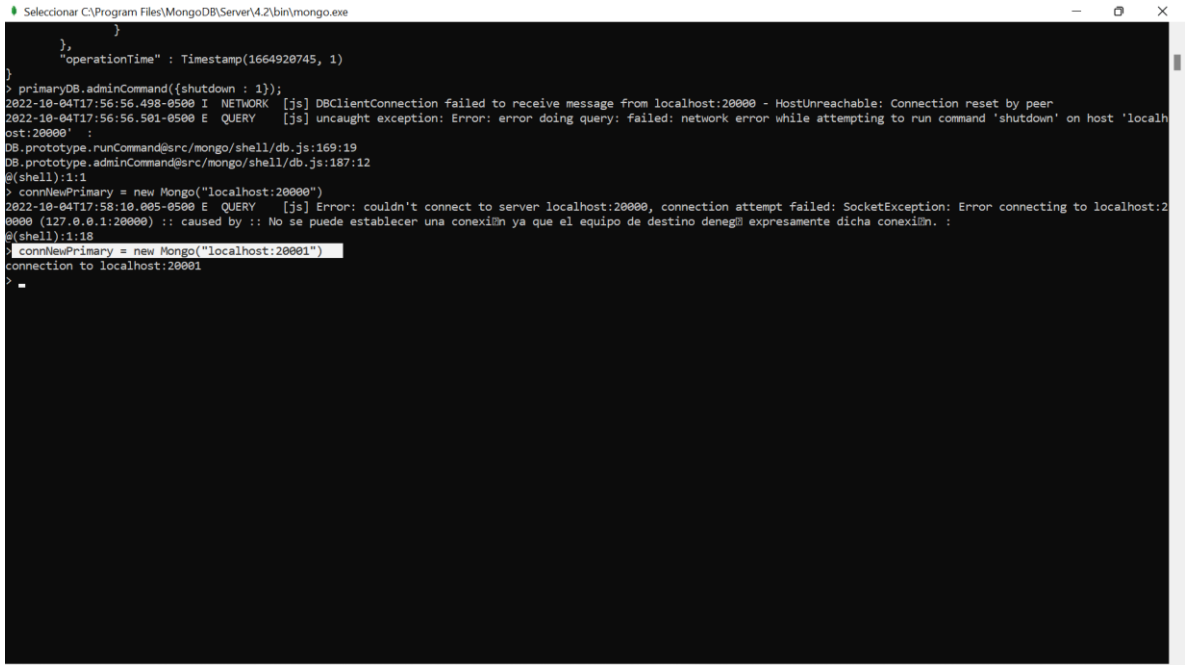

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
}
> secondaryTestDB.arbitro.findTwo()
2022-10-04T17:41:25.419-0500 E QUERY [js] uncaught exception: TypeError: secondaryTestDB.arbitro.findTwo is not a function :
@(shell):1:1
> secondaryTestDB.arbitro.find()
{ "_id" : ObjectId("633ca97cfe147300f224b33e"), "id_editorial" : "3", "nombre" : "Brandin ", "cedula" : " 1090506954 " }
{ "_id" : ObjectId("633caca9fe147300f224b33f"), "id" : "4", "nombre" : "sneider ", "cedula" : " 1090506954 " }
> secondaryTestDB.arbitro.find().pretty()
{
"_id" : ObjectId("633ca97cfe147300f224b33e"),
"id_editorial" : "3",
"nombre" : "Brandin ",
"cedula" : " 1090506954 "
}
{
"_id" : ObjectId("633caca9fe147300f224b33f"),
"id" : "4",
"nombre" : "sneider ",
"cedula" : " 1090506954 "
}
> connPrimary = new Mongo("localhost:20000")
connection to localhost:20000
> primaryDB = connPrimary.getDB("torneo")
torneo
> primaryDB.isMaster()
{
"hosts" : [
"BrandonF:20000",
"BrandonF:20001",
"BrandonF:20002"
],
"setName" : "MireplicaSet",
"setVersion" : 2,
"ismaster" : true,
"secondary" : false,
"primary" : "BrandonF:20000",
"me" : "BrandonF:20000",
"electionId" : ObjectId("7fffffff0000000000000001"),
"lastWrite" : {
"opTime" : {
"ts" : Timestamp(1664920745, 1),
"t" : NumberLong(1)
},

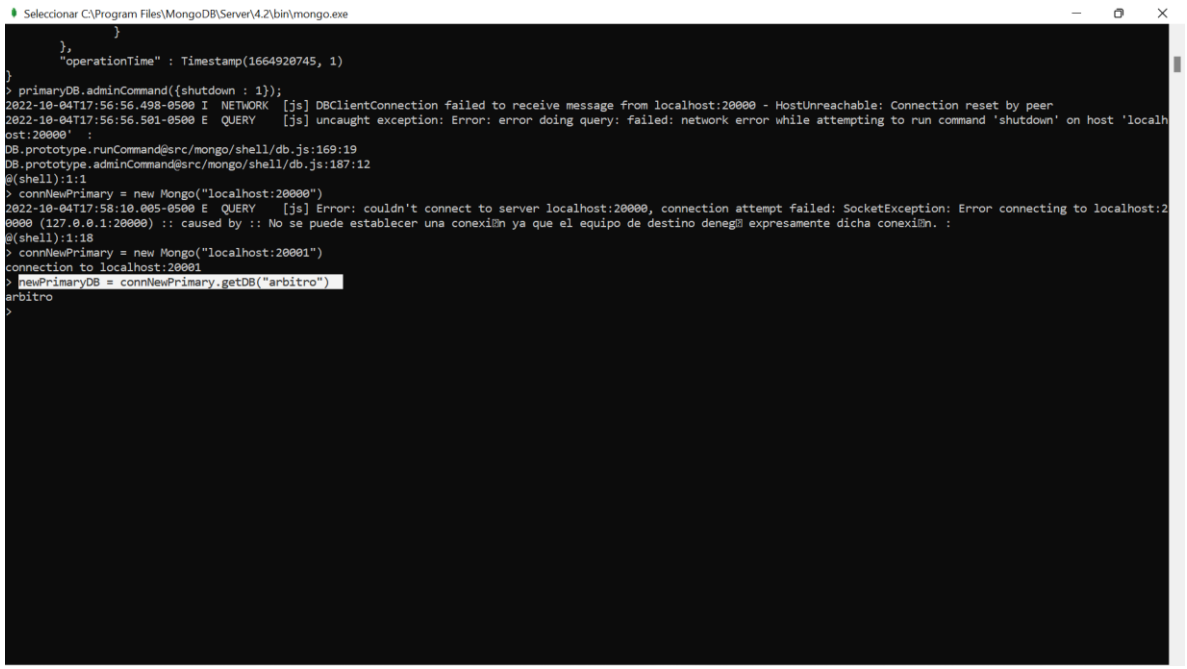

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
}
> secondaryTestDB.arbitro.findTwo()
2022-10-04T17:41:25.419-0500 E QUERY [js] uncaught exception: TypeError: secondaryTestDB.arbitro.findTwo is not a function :
@(shell):1:1
> secondaryTestDB.arbitro.find()
{ "_id" : ObjectId("633ca97cfe147300f224b33e"), "id_editorial" : "3", "nombre" : "Brandin ", "cedula" : " 1090506954 " }
{ "_id" : ObjectId("633caca9fe147300f224b33f"), "id" : "4", "nombre" : "sneider ", "cedula" : " 1090506954 " }
> secondaryTestDB.arbitro.find().pretty()
{
"_id" : ObjectId("633ca97cfe147300f224b33e"),
"id_editorial" : "3",
"nombre" : "Brandin ",
"cedula" : " 1090506954 "
}
{
"_id" : ObjectId("633caca9fe147300f224b33f"),
"id" : "4",
"nombre" : "sneider ",
"cedula" : " 1090506954 "
}
> connPrimary = new Mongo("localhost:20000")
connection to localhost:20000
> primaryDB = connPrimary.getDB("torneo")
torneo
> primaryDB.isMaster()
{
"hosts" : [
"BrandonF:20000",
"BrandonF:20001",
"BrandonF:20002"
],
"setName" : "MireplicaSet",
"setVersion" : 2,
"ismaster" : true,
"secondary" : false,
"primary" : "BrandonF:20000",
"me" : "BrandonF:20000",
"electionId" : ObjectId("7fffffff0000000000000001"),
"lastWrite" : {
"opTime" : {
"ts" : Timestamp(1664920745, 1),
"t" : NumberLong(1)
},

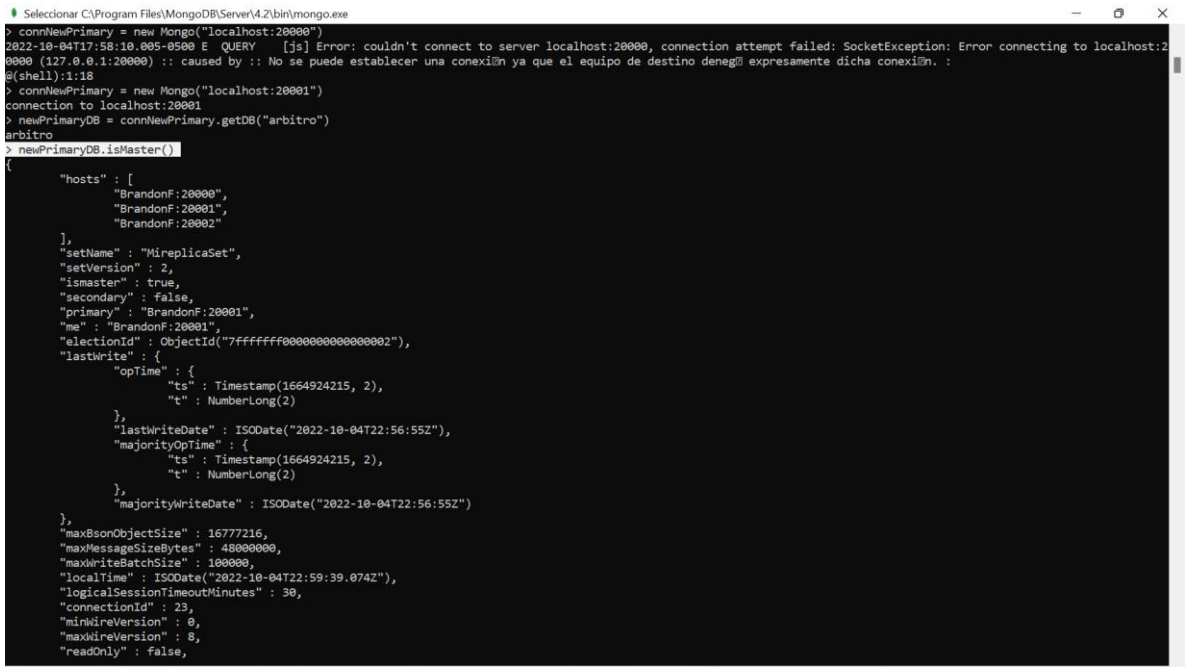

Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
torneo
> primaryDB.isMaster()
{
"hosts" : [
"BrandonF:20000",
"BrandonF:20001",
"BrandonF:20002"
],
"setName" : "MireplicaSet",
"setVersion" : 2,
"ismaster" : true,
"secondary" : false,
"primary" : "BrandonF:20000",
"me" : "BrandonF:20000",
"electionId" : ObjectId("7fffffff0000000000000001"),
"lastWrite" : {
"opTime" : {
"ts" : Timestamp(1664920745, 1),
"t" : NumberLong(1)
},
"lastWriteDate" : ISODate("2022-10-04T21:59:05Z"),
"majorityOpTime" : {
"ts" : Timestamp(1664920745, 1),
"t" : NumberLong(1)
},
"majorityWriteDate" : ISODate("2022-10-04T21:59:05Z")
},
"maxBsonObjectSize" : 16777216,
"maxMessageSizeBytes" : 48000000,
"maxWriteBatchSize" : 100000,
"localTime" : ISODate("2022-10-04T22:55:19.477Z"),
"logicalSessionTimeoutMinutes" : 30,
"connectionId" : 29,
"minWireVersion" : 0,
"maxWireVersion" : 8,
"readOnly" : false,
"ok" : 1,
"$clusterTime" : {
"clusterTime" : Timestamp(1664920745, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}


Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
}
},
"operationTime" : Timestamp(1664920745, 1)
}
> primaryDB.adminCommand({shutdown : 1});


C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
}
},
"operationTime" : Timestamp(1664920745, 1)
}
> primaryDB.adminCommand({shutdown : 1});
2022-10-04T17:56:56.498-0500 I NETWORK [js] DBClientConnection failed to receive message from localhost:20000 - HostUnreachable: Connection reset by peer
2022-10-04T17:56:56.501-0500 E QUERY [js] uncaught exception: Error: error doing query: failed: network error while attempting to run command 'shutdown' on host 'localh
ost:20000' :
DB.prototype.runCommand@src/mongo/shell/db.js:169:19
DB.prototype.adminCommand@src/mongo/shell/db.js:187:12
@(shell):1:1
>


Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
}
},
"operationTime" : Timestamp(1664920745, 1)
}
> primaryDB.adminCommand({shutdown : 1});
2022-10-04T17:56:56.498-0500 I NETWORK [js] DBClientConnection failed to receive message from localhost:20000 - HostUnreachable: Connection reset by peer
2022-10-04T17:56:56.501-0500 E QUERY [js] uncaught exception: Error: error doing query: failed: network error while attempting to run command 'shutdown' on host 'localh
ost:20000' :
DB.prototype.runCommand@src/mongo/shell/db.js:169:19
DB.prototype.adminCommand@src/mongo/shell/db.js:187:12
@(shell):1:1
> connNewPrimary = new Mongo("localhost:20000")
2022-10-04T17:58:10.005-0500 E QUERY [js] Error: couldn't connect to server localhost:20000, connection attempt failed: SocketException: Error connecting to localhost:2
0000 (127.0.0.1:20000) :: caused by :: No se puede establecer una conexi@n ya que el equipo de destino deneg@ expresamente dicha conexi@n. :
@(shell):1:18
> connNewPrimary = new Mongo("localhost:20001")
connection to localhost:20001
>


Seleccionar C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
}
},
"operationTime" : Timestamp(1664920745, 1)
}
> primaryDB.adminCommand({shutdown : 1});
2022-10-04T17:56:56.498-0500 I NETWORK [js] DBClientConnection failed to receive message from localhost:20000 - HostUnreachable: Connection reset by peer
2022-10-04T17:56:56.501-0500 E QUERY [js] uncaught exception: Error: error doing query: failed: network error while attempting to run command 'shutdown' on host 'localh
ost:20000' :
DB.prototype.runCommand@src/mongo/shell/db.js:169:19
DB.prototype.adminCommand@src/mongo/shell/db.js:187:12
@(shell):1:1
> connNewPrimary = new Mongo("localhost:20000")
2022-10-04T17:58:10.005-0500 E QUERY [js] Error: couldn't connect to server localhost:20000, connection attempt failed: SocketException: Error connecting to localhost:2
0000 (127.0.0.1:20000) :: caused by :: No se puede establecer una conexi@n ya que el equipo de destino deneg@ expresamente dicha conexi@n. :
@(shell):1:18
> connNewPrimary = new Mongo("localhost:20001")
connection to localhost:20001
> newPrimaryDB = connNewPrimary.getDB("arbitro")
arbitro
>




LINK VIDEO: https://drive.google.com/file/d/1k7KDL2XUDxk6AibmUP8rxlO2o-X8B_GG/view?usp=sharing

LINK GITHUB: https://github.com/Brandfranco/ACTIVIDAD-3-DBS

**BIBLIOGRAFIA**

Sarasa, A. (2016). Introducción a las bases de datos NoSQL usando MongoDB. Editorial UOC. https://elibro.net/es/lc/biblioibero/titulos/58524

Adicionalmente, el capítulo 7 (Replicación) del libro de Sarasa, A. (2016) Introducción a las bases de datos NoSQL usando MongoDB. Editorial UOC, disponible en: https://elibro.net/es/lc/biblioibero/titulos/58524.