Actividad 4 - Pruebas de particionamiento de bases de datos NoSQL

Realizado por:

Joe Alejandro Sierra Ocassal

Ingrid Johana Rojas Gómez

Corporación Universitaria Iberoamericana

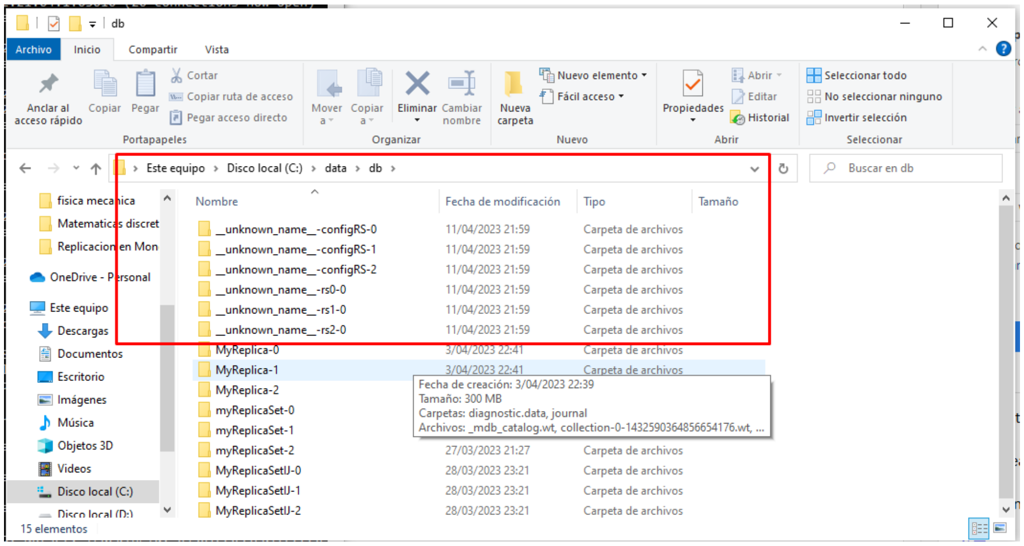
Bases de datos avanzadas

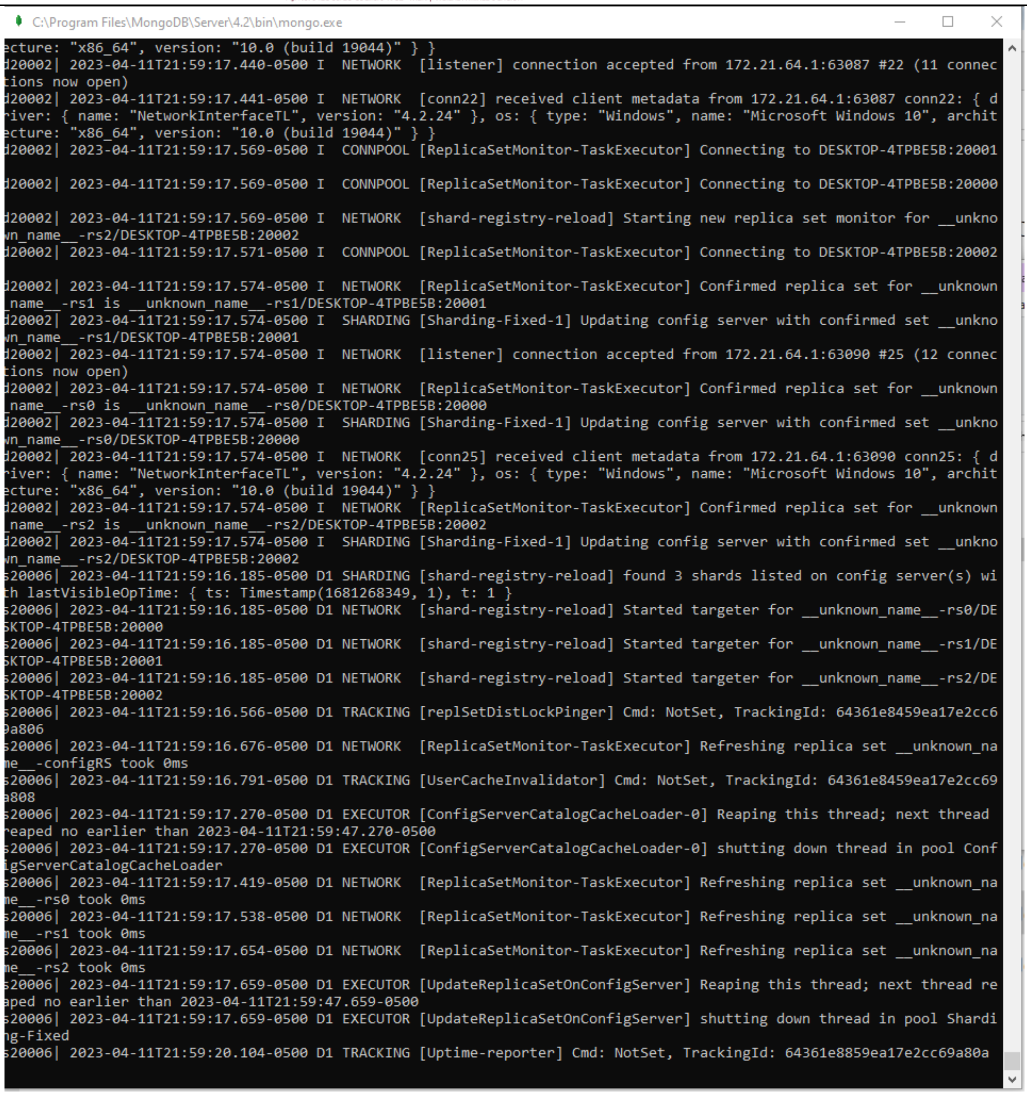
WILLIAM RUIZ

Abril 2023

Casos de prueba

Disponemos de los siguientes casos de prueba, en los cuales establecemos el objetivo de cada prueba a realizar, detallando el resultado obtenido, con el cual garantizamos la calidad y el funcionamiento del particionamiento dando cumplimiento a los requerimientos no funcionales documentados en la actividad #3.

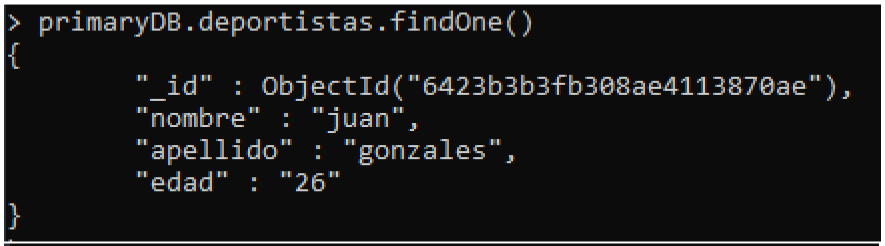
#	Descripción	Resultado
1	Particionamiento Se debe verificar que se crearon las particiones físicas correspondientes donde la información quede almacenada y a su vez se replica la data en cada shard como se especificó previamente en el esquema centralizado de replicas.	Se crea sin inconvenientes las particiones físicas: Este equipo › Disco local (C:) › data › db › Nombre / Fecha de modificación / Tipo / Tamaño __unknown_name__-configRS-0 11/04/2023 21:59 Carpeta de archivos __unknown_name__-configRS-1 11/04/2023 21:59 Carpeta de archivos __unknown_name__-configRS-2 11/04/2023 21:59 Carpeta de archivos __unknown_name__-rs0-0 11/04/2023 21:59 Carpeta de archivos __unknown_name__-rs1-0 11/04/2023 21:59 Carpeta de archivos __unknown_name__-rs2-0 11/04/2023 21:59 Carpeta de archivos MyReplica-0 3/04/2023 22:41 Carpeta de archivos MyReplica-1 3/04/2023 22:41 Carpeta de archivos MyReplica-2 myReplicaSet-0 myReplicaSet-1 myReplicaSet-2 27/03/2023 21:27 Carpeta de archivos MyReplicaSetIJ-0 28/03/2023 23:21 Carpeta de archivos MyReplicaSetIJ-1 28/03/2023 23:21 Carpeta de archivos MyReplicaSetIJ-2 28/03/2023 23:21 Carpeta de archivos Fecha de creación: 3/04/2023 22:39 Tamaño: 300 MB Carpetas: diagnostic.data, journal Archivos: _mdb_catalog.wt, collection-0-1432590364856654176.wt, ... 15 elementos Creación de los 3 shards:



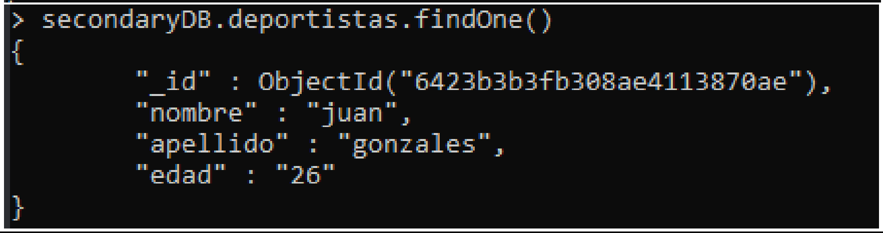
C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe

```
ecture: "x86_64", version: "10.0 (build 19044)" } }
d20002| 2023-04-11T21:59:17.440-0500 I  NETWORK  [listener] connection accepted from 172.21.64.1:63087 #22 (11 connec
tions now open)
d20002| 2023-04-11T21:59:17.441-0500 I  NETWORK  [conn22] received client metadata from 172.21.64.1:63087 conn22: { d
river: { name: "NetworkInterfaceTL", version: "4.2.24" }, os: { type: "Windows", name: "Microsoft Windows 10", archit
ecture: "x86_64", version: "10.0 (build 19044)" } }
d20002| 2023-04-11T21:59:17.569-0500 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to DESKTOP-4TPBE5B:20001

d20002| 2023-04-11T21:59:17.569-0500 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to DESKTOP-4TPBE5B:20000

d20002| 2023-04-11T21:59:17.569-0500 I  NETWORK  [shard-registry-reload] Starting new replica set monitor for __unkno
wn_name__-rs2/DESKTOP-4TPBE5B:20002
d20002| 2023-04-11T21:59:17.571-0500 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to DESKTOP-4TPBE5B:20002

d20002| 2023-04-11T21:59:17.574-0500 I  NETWORK  [ReplicaSetMonitor-TaskExecutor] Confirmed replica set for __unknown
_name__-rs1 is __unknown_name__-rs1/DESKTOP-4TPBE5B:20001
d20002| 2023-04-11T21:59:17.574-0500 I  SHARDING [Sharding-Fixed-1] Updating config server with confirmed set __unkno
wn_name__-rs1/DESKTOP-4TPBE5B:20001
d20002| 2023-04-11T21:59:17.574-0500 I  NETWORK  [listener] connection accepted from 172.21.64.1:63090 #25 (12 connec
tions now open)
d20002| 2023-04-11T21:59:17.574-0500 I  NETWORK  [ReplicaSetMonitor-TaskExecutor] Confirmed replica set for __unknown
_name__-rs0 is __unknown_name__-rs0/DESKTOP-4TPBE5B:20000
d20002| 2023-04-11T21:59:17.574-0500 I  SHARDING [Sharding-Fixed-1] Updating config server with confirmed set __unkno
wn_name__-rs0/DESKTOP-4TPBE5B:20000
d20002| 2023-04-11T21:59:17.574-0500 I  NETWORK  [conn25] received client metadata from 172.21.64.1:63090 conn25: { d
river: { name: "NetworkInterfaceTL", version: "4.2.24" }, os: { type: "Windows", name: "Microsoft Windows 10", archit
ecture: "x86_64", version: "10.0 (build 19044)" } }
d20002| 2023-04-11T21:59:17.574-0500 I  NETWORK  [ReplicaSetMonitor-TaskExecutor] Confirmed replica set for __unknown
_name__-rs2 is __unknown_name__-rs2/DESKTOP-4TPBE5B:20002
d20002| 2023-04-11T21:59:17.574-0500 I  SHARDING [Sharding-Fixed-1] Updating config server with confirmed set __unkno
wn_name__-rs2/DESKTOP-4TPBE5B:20002
s20006| 2023-04-11T21:59:16.185-0500 D1 SHARDING [shard-registry-reload] found 3 shards listed on config server(s) wi
th lastVisibleOpTime: { ts: Timestamp(1681268349, 1), t: 1 }
s20006| 2023-04-11T21:59:16.185-0500 D1 NETWORK  [shard-registry-reload] Started targeter for __unknown_name__-rs0/DE
SKTOP-4TPBE5B:20000
s20006| 2023-04-11T21:59:16.185-0500 D1 NETWORK  [shard-registry-reload] Started targeter for __unknown_name__-rs1/DE
SKTOP-4TPBE5B:20001
s20006| 2023-04-11T21:59:16.185-0500 D1 NETWORK  [shard-registry-reload] Started targeter for __unknown_name__-rs2/DE
SKTOP-4TPBE5B:20002
s20006| 2023-04-11T21:59:16.566-0500 D1 TRACKING [replSetDistLockPinger] Cmd: NotSet, TrackingId: 64361e8459ea17e2cc6
9a806
s20006| 2023-04-11T21:59:16.676-0500 D1 NETWORK  [ReplicaSetMonitor-TaskExecutor] Refreshing replica set __unknown_na
me__-configRS took 0ms
s20006| 2023-04-11T21:59:16.791-0500 D1 TRACKING [UserCacheInvalidator] Cmd: NotSet, TrackingId: 64361e8459ea17e2cc69
a808
s20006| 2023-04-11T21:59:17.270-0500 D1 EXECUTOR [ConfigServerCatalogCacheLoader-0] Reaping this thread; next thread
reaped no earlier than 2023-04-11T21:59:47.270-0500
s20006| 2023-04-11T21:59:17.270-0500 D1 EXECUTOR [ConfigServerCatalogCacheLoader-0] shutting down thread in pool Conf
igServerCatalogCacheLoader
s20006| 2023-04-11T21:59:17.419-0500 D1 NETWORK  [ReplicaSetMonitor-TaskExecutor] Refreshing replica set __unknown_na
me__-rs0 took 0ms
s20006| 2023-04-11T21:59:17.538-0500 D1 NETWORK  [ReplicaSetMonitor-TaskExecutor] Refreshing replica set __unknown_na
me__-rs1 took 0ms
s20006| 2023-04-11T21:59:17.654-0500 D1 NETWORK  [ReplicaSetMonitor-TaskExecutor] Refreshing replica set __unknown_na
me__-rs2 took 0ms
s20006| 2023-04-11T21:59:17.659-0500 D1 EXECUTOR [UpdateReplicaSetOnConfigServer] Reaping this thread; next thread re
aped no earlier than 2023-04-11T21:59:47.659-0500
s20006| 2023-04-11T21:59:17.659-0500 D1 EXECUTOR [UpdateReplicaSetOnConfigServer] shutting down thread in pool Shardi
ng-Fixed
s20006| 2023-04-11T21:59:20.104-0500 D1 TRACKING [Uptime-reporter] Cmd: NotSet, TrackingId: 64361e8859ea17e2cc69a80a
```

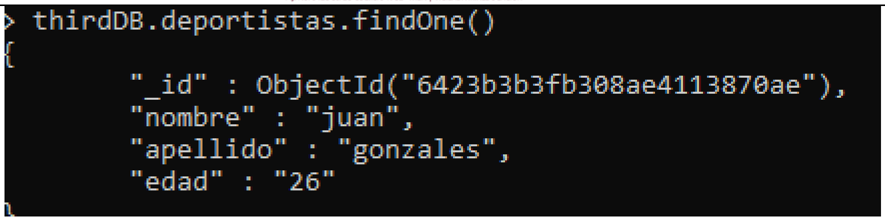
Comprobación de registros insertados, donde se evidencia la replicación funcionando:



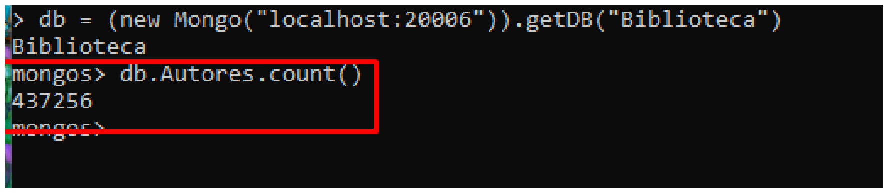
```
> primaryDB.deportistas.findOne()
{
        "_id" : ObjectId("6423b3b3fb308ae4113870ae"),
        "nombre" : "juan",
        "apellido" : "gonzales",
        "edad" : "26"
}
```



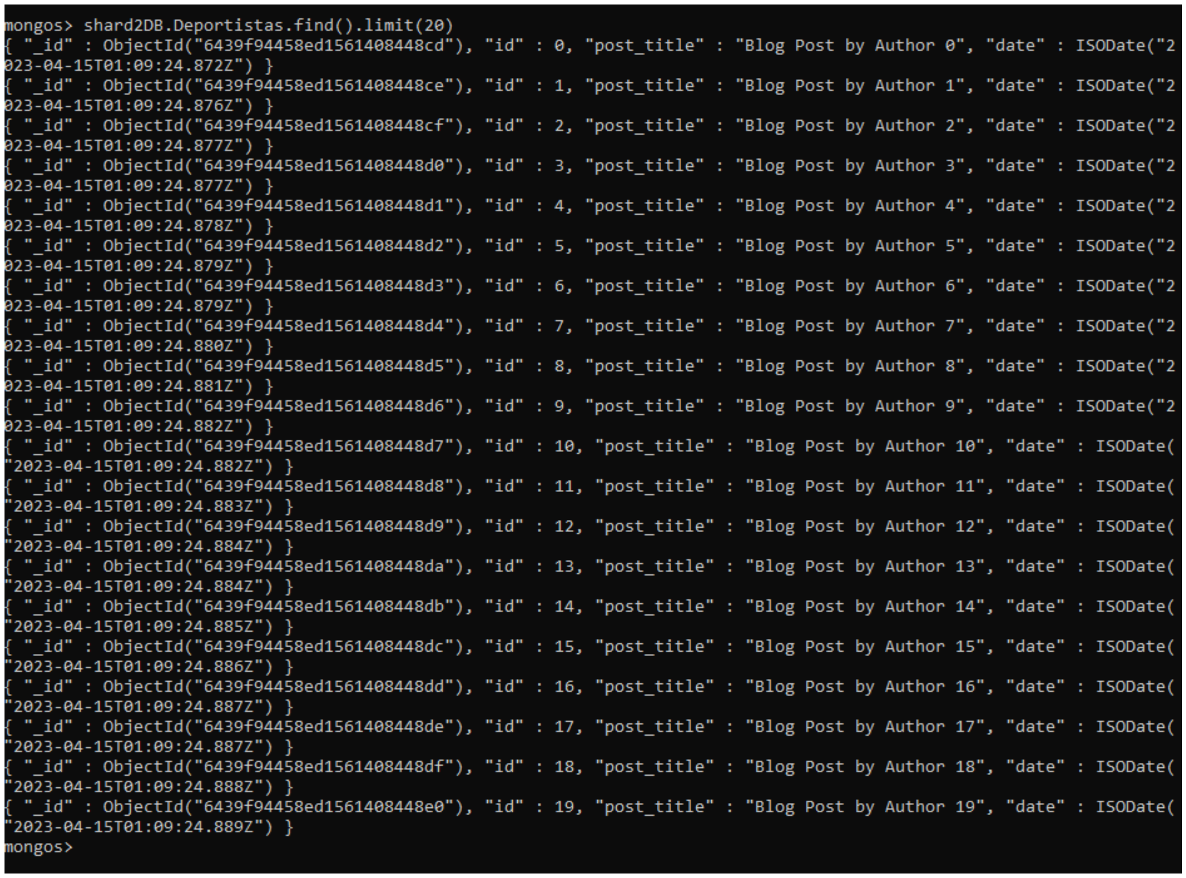
```
> secondaryDB.deportistas.findOne()
{
        "_id" : ObjectId("6423b3b3fb308ae4113870ae"),
        "nombre" : "juan",
        "apellido" : "gonzales",
        "edad" : "26"
}
```

```
> thirdDB.deportistas.findOne()
{
        "_id" : ObjectId("6423b3b3fb308ae4113870ae"),
        "nombre" : "juan",
        "apellido" : "gonzales",
        "edad" : "26"
}
```

2	Carga de datos Se debe validar el tiempo de inserción de datos, colección de Torneo Deportivo sea inferior a un segundo.	Para realizar la prueba a este caso, hemos intentado realizar la inserción de 800.000 datos a la colección de prueba Autores, en la que se insertaron 437.256 datos en 60 segundos:  Por lo cual se encontró un rendimiento efectivo ya que por segundo se registraron 7.287 filas en la base de datos por segundo.

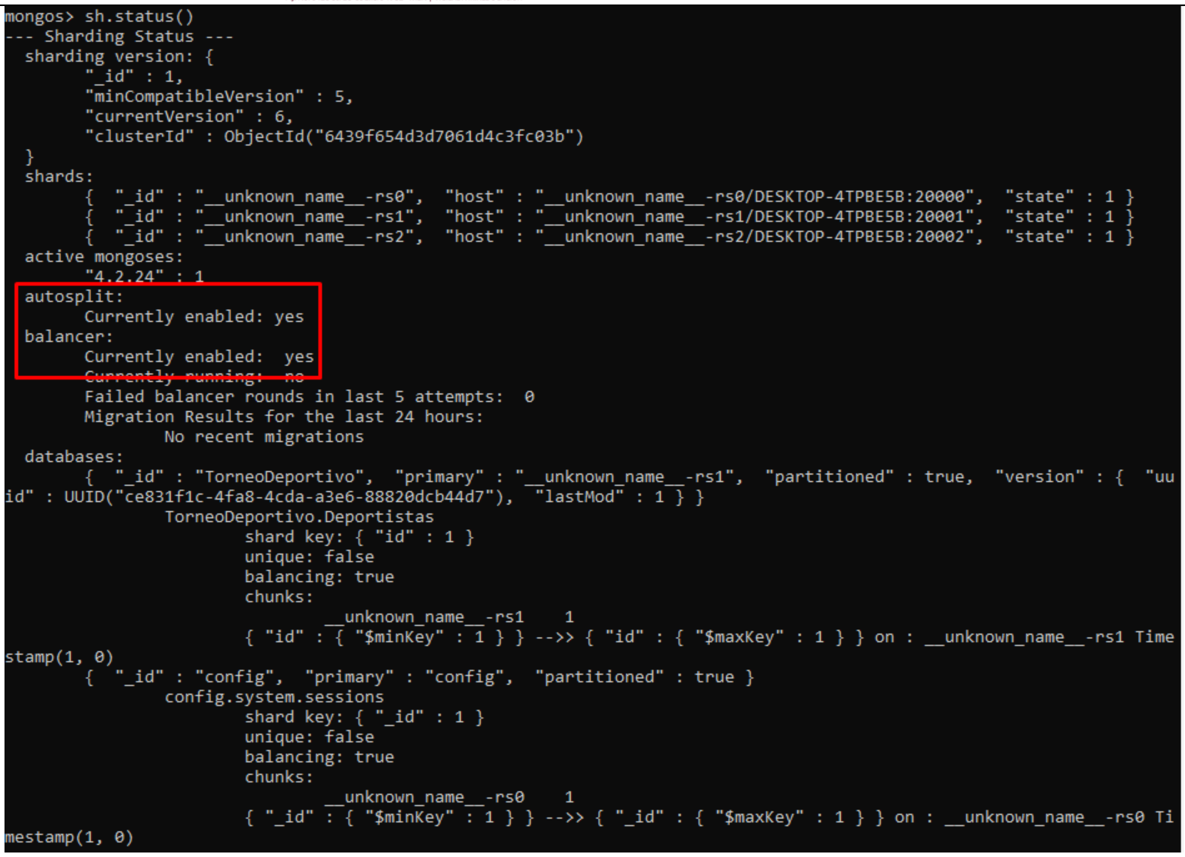
```
> db = (new Mongo("localhost:20006")).getDB("Biblioteca")
Biblioteca
mongos> db.Autores.count()
437256
mongos>
```

3	Tiempos de respuesta Validar tiempo de respuesta consulta de datos, alguna de las colecciones de la BD propuesto sea inferior a un segundo.	Realizamos la consulta de la colección especifica de deportistas en la que consultamos con un limite de 20 registro con el fin de verificar su tiempo de respuesta optimo con estos datos.  Obtenemos un resultado exitoso en el cual tiene una respuesta inferior a 1 segundo.

```
ongos> shard2DB.Deportistas.find().limit(20)
{ "_id" : ObjectId("6439f94458ed1561408448cd"), "id" : 0, "post_title" : "Blog Post by Author 0", "date" : ISODate("2
023-04-15T01:09:24.872Z") }
{ "_id" : ObjectId("6439f94458ed1561408448ce"), "id" : 1, "post_title" : "Blog Post by Author 1", "date" : ISODate("2
023-04-15T01:09:24.876Z") }
{ "_id" : ObjectId("6439f94458ed1561408448cf"), "id" : 2, "post_title" : "Blog Post by Author 2", "date" : ISODate("2
023-04-15T01:09:24.877Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d0"), "id" : 3, "post_title" : "Blog Post by Author 3", "date" : ISODate("2
023-04-15T01:09:24.877Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d1"), "id" : 4, "post_title" : "Blog Post by Author 4", "date" : ISODate("2
023-04-15T01:09:24.878Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d2"), "id" : 5, "post_title" : "Blog Post by Author 5", "date" : ISODate("2
023-04-15T01:09:24.879Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d3"), "id" : 6, "post_title" : "Blog Post by Author 6", "date" : ISODate("2
023-04-15T01:09:24.879Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d4"), "id" : 7, "post_title" : "Blog Post by Author 7", "date" : ISODate("2
023-04-15T01:09:24.880Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d5"), "id" : 8, "post_title" : "Blog Post by Author 8", "date" : ISODate("2
023-04-15T01:09:24.881Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d6"), "id" : 9, "post_title" : "Blog Post by Author 9", "date" : ISODate("2
023-04-15T01:09:24.882Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d7"), "id" : 10, "post_title" : "Blog Post by Author 10", "date" : ISODate(
"2023-04-15T01:09:24.882Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d8"), "id" : 11, "post_title" : "Blog Post by Author 11", "date" : ISODate(
"2023-04-15T01:09:24.883Z") }
{ "_id" : ObjectId("6439f94458ed1561408448d9"), "id" : 12, "post_title" : "Blog Post by Author 12", "date" : ISODate(
"2023-04-15T01:09:24.884Z") }
{ "_id" : ObjectId("6439f94458ed1561408448da"), "id" : 13, "post_title" : "Blog Post by Author 13", "date" : ISODate(
"2023-04-15T01:09:24.884Z") }
{ "_id" : ObjectId("6439f94458ed1561408448db"), "id" : 14, "post_title" : "Blog Post by Author 14", "date" : ISODate(
"2023-04-15T01:09:24.885Z") }
{ "_id" : ObjectId("6439f94458ed1561408448dc"), "id" : 15, "post_title" : "Blog Post by Author 15", "date" : ISODate(
"2023-04-15T01:09:24.886Z") }
{ "_id" : ObjectId("6439f94458ed1561408448dd"), "id" : 16, "post_title" : "Blog Post by Author 16", "date" : ISODate(
"2023-04-15T01:09:24.887Z") }
{ "_id" : ObjectId("6439f94458ed1561408448de"), "id" : 17, "post_title" : "Blog Post by Author 17", "date" : ISODate(
"2023-04-15T01:09:24.887Z") }
{ "_id" : ObjectId("6439f94458ed1561408448df"), "id" : 18, "post_title" : "Blog Post by Author 18", "date" : ISODate(
"2023-04-15T01:09:24.888Z") }
{ "_id" : ObjectId("6439f94458ed1561408448e0"), "id" : 19, "post_title" : "Blog Post by Author 19", "date" : ISODate(
"2023-04-15T01:09:24.889Z") }
mongos>
```

4	Validación balanceo de carga de datos en los diferentes nodos:	Para este caso, hicimos la inserción de 800.000 datos con el balanceo de carga habilitado, de acuerdo con:

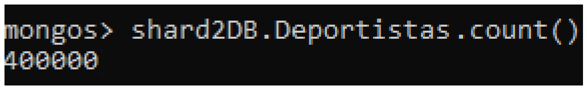
Validar el funcionamiento del balanceador de carga de datos mediante los diferentes nodos creados, al hacer inserciones masivas de datos en alguna de las colecciones de la BD propuesto.



```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("6439f654d3d7061d4c3fc03b")
  }
  shards:
        {  "_id" : "__unknown_name__-rs0",  "host" : "__unknown_name__-rs0/DESKTOP-4TPBE5B:20000",  "state" : 1 }
        {  "_id" : "__unknown_name__-rs1",  "host" : "__unknown_name__-rs1/DESKTOP-4TPBE5B:20001",  "state" : 1 }
        {  "_id" : "__unknown_name__-rs2",  "host" : "__unknown_name__-rs2/DESKTOP-4TPBE5B:20002",  "state" : 1 }
  active mongoses:
        "4.2.24" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "TorneoDeportivo",  "primary" : "__unknown_name__-rs1",  "partitioned" : true,  "version" : {  "uu
id" : UUID("ce831f1c-4fa8-4cda-a3e6-88820dcb44d7"),  "lastMod" : 1 } }
                TorneoDeportivo.Deportistas
                        shard key: { "id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                __unknown_name__-rs1    1
                        { "id" : { "$minKey" : 1 } } -->> { "id" : { "$maxKey" : 1 } } on : __unknown_name__-rs1 Time
stamp(1, 0)
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
                config.system.sessions
                        shard key: { "_id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                __unknown_name__-rs0    1
                        { "_id" : { "$minKey" : 1 } } -->> { "_id" : { "$maxKey" : 1 } } on : __unknown_name__-rs0 Ti
mestamp(1, 0)
```

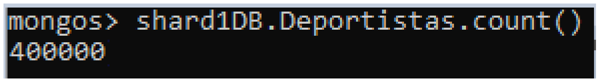
En la colección usada tenemos la siguiente verificación (después de la inserción de la cantidad de datos mencionados):

Shard2:



```
mongos> shard2DB.Deportistas.count()
400000
```

Shard1:



```
mongos> shard1DB.Deportistas.count()
400000
```

Por lo cual concluimos que se balancea la carga de datos.