# Conceptos y Comandos básicos del particionamiento en bases de datos NoSQL

Elian Fernando Mujica Armero
Diciembre 2023

Corporación Universitaria Iberoamericana
Facultad de Ingeniería
Bases de datos Avanzadas

## Tabla de Contenidos

## Introducción

Esta actividad tiene como objetivo apropiar los conceptos de particionamiento en bases de datos, entendiendo sus ventajas y usos. Para esto, se realizará un particionamiento para la base de datos ATP_Finals_2023, con el objetivo de cumplir los requerimientos de desempeño planteados en el documento RequerimientosNoFuncionales_IEEE.pdf.

## Pasos del particionamiento de la base de datos

A continuación, enumero y adjunto evidencia de los pasos realizados en el particionamiento de la base de datos ATP_Finals_2023. Estos pasos son realizados a partir de la guía "Tutorial de operaciones básicas de particionamiento en Mongodb".

### 1. Creación de directorios

Se crean los directorios donde se almacenará la información de las particiones y los servidores de configuración, así como sus réplicas

```
:: Creación directorios partición 1 con sus réplicas
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard1\data1"
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard1\data2"
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard1\data3"

:: Creación directorios partición 2 con sus réplicas
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard2\data1"
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard2\data2"
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard2\data3"

:: Creación directorios partición 3 con sus réplicas
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard3\data1"
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard3\data2"
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard3\data3"

:: Creación directorios servidor de configuración con sus réplicas
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\config_server\data1"
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\config_server\data2"
mkdir "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\config_server\data3"
```

## 2. Iniciación de los fragmentos

```
mongod --shardsvr --port 26017 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard1\data1" --replSet replica_atp_fragmento1

mongod --shardsvr --port 26117 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard1\data2" --replSet replica_atp_fragmento1

mongod --shardsvr --port 26217 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard1\data3" --replSet replica_atp_fragmento1


mongod --shardsvr --port 28017 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard2\data1" --replSet replica_atp_fragmento2

mongod --shardsvr --port 28117 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard2\data2" --replSet replica_atp_fragmento2

mongod --shardsvr --port 28217 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard2\data3" --replSet replica_atp_fragmento2


mongod --shardsvr --port 29017 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard3\data1" --replSet replica_atp_fragmento3

mongod --shardsvr --port 29117 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard3\data2" --replSet replica_atp_fragmento3

mongod --shardsvr --port 29217 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\shard3\data3" --replSet replica_atp_fragmento3


mongod --configsvr --port 47017 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\config_server\data1" --replSet
replica_atp_configserver1

mongod --configsvr --port 47117 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\config_server\data2" --replSet
replica_atp_configserver1

mongod --configsvr --port 47217 --dbpath "C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to
\BDA\Actividades_BDA\Actividad 3 & 4\shard_data\config_server\data3" --replSet
replica_atp_configserver1
```

3. **Se conecta a los puertos de cada fragmento**

```
mongosh --host localhost:26017
rs.initiate({_id: 'replica_atp_fragmento1', members:
[{_id:0,host:'localhost:26017'},
{_id:1,host:'localhost:26117'},
{_id:2,host:'localhost:26217'}]})

mongosh --host localhost:28017
rs.initiate({_id: 'replica_atp_fragmento2', members:
[{_id:0,host:'localhost:28017'},
{_id:1,host:'localhost:28117'},
{_id:2,host:'localhost:28217'}]})

mongosh --host localhost:29017
rs.initiate({_id: 'replica_atp_fragmento3', members:
[{_id:0,host:'localhost:29017'},
{_id:1,host:'localhost:29117'},
{_id:2,host:'localhost:29217'}]})

mongosh --host localhost:47017
rs.initiate({_id: "replica_atp_configserver1", configsvr:true,
members: [{_id:0, host:'localhost:47017'},{_id:1,
host:'localhost:47117'},{_id:2, host:'localhost:47217'}]})
```

4. **Verificar el estado de replicación de cada fragmento:**

```
replica_atp_fragmento1 [direct: primary] test> db.serverStatus().repl
{
  topologyVersion: {
    processId: ObjectId("657e1f54c27426a5b17e135b"),
    counter: Long("7")
  },
  hosts: [ 'localhost:26017', 'localhost:26117', 'localhost:26217' ],
  setName: 'replica_atp_fragmento1',
  setVersion: 1,
  isWritablePrimary: true,
  secondary: false,
  primary: 'localhost:26017',
  me: 'localhost:26017',
```

```
replica_atp_fragmento2 [direct: other] test> db.serverStatus().repl
{
  topologyVersion: {
    processId: ObjectId("657e1f50ce2745055641aca8"),
    counter: Long("7")
  },
  hosts: [ 'localhost:28017', 'localhost:28117', 'localhost:28217' ],
  setName: 'replica_atp_fragmento2',
  setVersion: 1,
  isWritablePrimary: true,
  secondary: false,
  primary: 'localhost:28017',
  me: 'localhost:28017',
```

```
replica_atp_fragmento3 [direct: other] test> db.serverStatus().repl
{
  topologyVersion: {
    processId: ObjectId("657e1f51d7e9f7d9eb3766a8"),
    counter: Long("7")
  },
  hosts: [ 'localhost:29017', 'localhost:29117', 'localhost:29217' ],
  setName: 'replica_atp_fragmento3',
  setVersion: 1,
  isWritablePrimary: true,
  secondary: false,
  primary: 'localhost:29017',
  me: 'localhost:29017',
```

```
replica_atp_configserver1 [direct: primary] test> db.serverStatus().rep
{
  topologyVersion: {
    processId: ObjectId("657e4d46b41c52951517c9c1"),
    counter: Long("6")
  },
  hosts: [ 'localhost:47017', 'localhost:47117', 'localhost:47217' ],
  setName: 'replica_atp_configserver1',
  setVersion: 1,
  isWritablePrimary: true,
  secondary: false,
  primary: 'localhost:47017',
  me: 'localhost:47017',
```

5. **Se Inicia el conjunto de servidores de Configuración con mongos (Enrutador)**

```
start mongos --configdb replica_atp_configserver1/localhost:47017,localhost:47117,localhost:47217 --port 1000
```

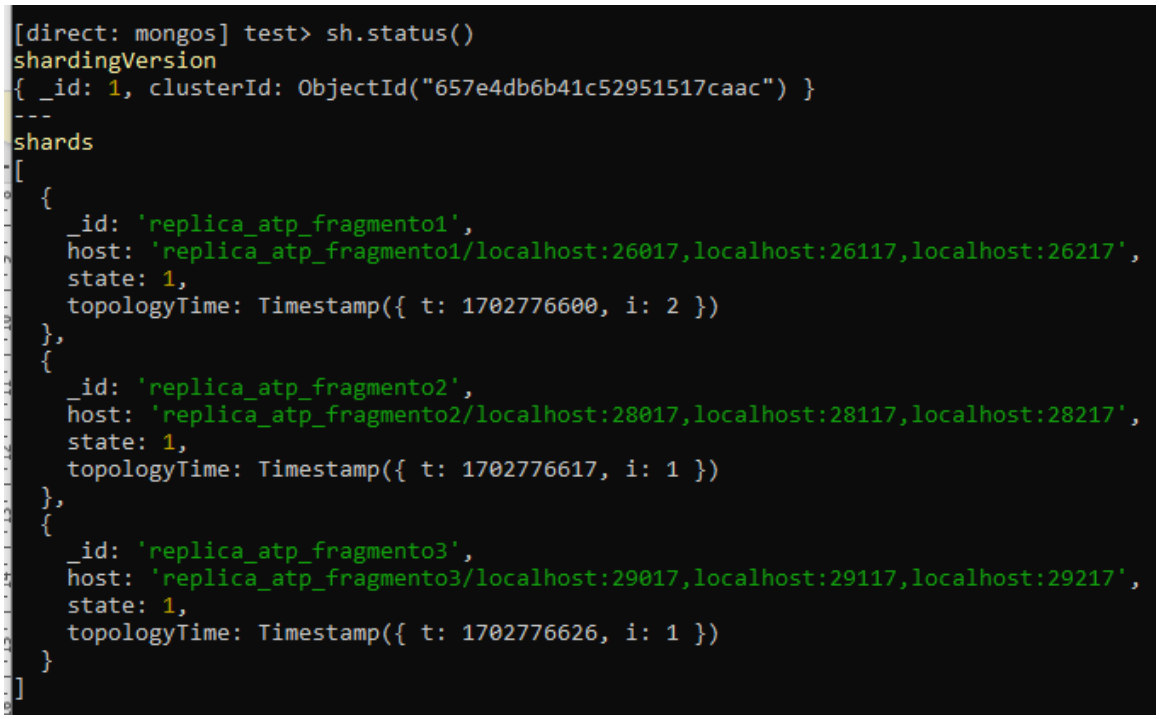
6. **Se conecta al enrutador por el puerto 1000**

```
mongosh --host localhost:1000
```

7. **Se agregan los fragmentos al enrutador**

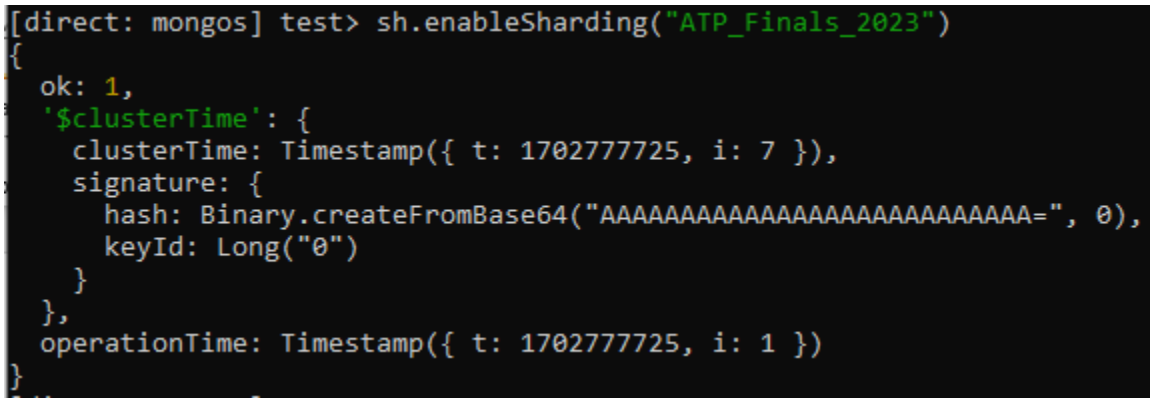
```
direct: mongos] test> sh.addShard("replica_atp_fragmento1/localhost:26017,localhost:26117,localhost:26217")

 shardAdded: 'replica_atp_fragmento1',
 ok: 1,
 '$clusterTime': {
   clusterTime: Timestamp({ t: 1702776600, i: 5 }),
   signature: {
     hash: Binary.createFromBase64("AAAAAAAAAAAAAAAAAAAAAAAAAAA=", 0),
     keyId: Long("0")
   }
 },
 operationTime: Timestamp({ t: 1702776600, i: 5 })

direct: mongos] test> sh.addShard("replica_atp_fragmento2/localhost:28017,localhost:28117,localhost:28217")

 shardAdded: 'replica_atp_fragmento2',
 ok: 1,
 '$clusterTime': {
   clusterTime: Timestamp({ t: 1702776617, i: 13 }),
   signature: {
     hash: Binary.createFromBase64("AAAAAAAAAAAAAAAAAAAAAAAAAAA=", 0),
     keyId: Long("0")
   }
 },
direct: mongos] test> sh.addShard("replica_atp_fragmento3/localhost:29017,localhost:29117,localhost:29217")

 shardAdded: 'replica_atp_fragmento3',
 ok: 1,
 '$clusterTime': {
   clusterTime: Timestamp({ t: 1702776653, i: 1 }),
   signature: {
     hash: Binary.createFromBase64("AAAAAAAAAAAAAAAAAAAAAAAAAAA=", 0),
     keyId: Long("0")
   }
 },
 operationTime: Timestamp({ t: 1702776653, i: 1 })

direct: mongos] test>
```

8. **Se verifica el estado de la conexión entre los fragmentos y el enrutador**



```
[direct: mongos] test> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId("657e4db6b41c52951517caac") }
---
shards
[
  {
    _id: 'replica_atp_fragmento1',
    host: 'replica_atp_fragmento1/localhost:26017,localhost:26117,localhost:26217',
    state: 1,
    topologyTime: Timestamp({ t: 1702776600, i: 2 })
  },
  {
    _id: 'replica_atp_fragmento2',
    host: 'replica_atp_fragmento2/localhost:28017,localhost:28117,localhost:28217',
    state: 1,
    topologyTime: Timestamp({ t: 1702776617, i: 1 })
  },
  {
    _id: 'replica_atp_fragmento3',
    host: 'replica_atp_fragmento3/localhost:29017,localhost:29117,localhost:29217',
    state: 1,
    topologyTime: Timestamp({ t: 1702776626, i: 1 })
  }
]
```

9. **Se habilita el Sharding para la base de datos**



```
[direct: mongos] test> sh.enableSharding("ATP_Finals_2023")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1702777725, i: 7 }),
    signature: {
      hash: Binary.createFromBase64("AAAAAAAAAAAAAAAAAAAAAAAAAAA=", 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1702777725, i: 1 })
}
```

10. **Se habilita el Sharding para las colecciones “deportistas” y “partidos”**

Se utiliza el _id como clave del shard, sin embargo, no es una práctica recomendada y se realiza aquí únicamente para fines académicos.

```
[direct: mongos] test> sh.shardCollection("ATP_Finals_2023.deportistas",{"_id":1})
{
  collectionsharded: 'ATP_Finals_2023.deportistas',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1702789149, i: 24 }),
    signature: {
      hash: Binary.createFromBase64("AAAAAAAAAAAAAAAAAAAAAAAAAAA=", 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1702789149, i: 24 })
}
[direct: mongos] test> sh.shardCollection("ATP_Finals_2023.partidos",{"_id":1})
{
  collectionsharded: 'ATP_Finals_2023.partidos',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1702789155, i: 1 }),
    signature: {: {
      hash: Binary.createFromBase64("AAAAAAAAAAAAAAAAAAAAAAAAAAA=", 0),
      keyId: Long("0")
    } hash: Binary.createFromBase64("AAAAAAAAAAAAAAAAAAAAAAAAAAA=", 0),
  },  keyId: Long("0")
  operationTime: Timestamp({ t: 1702789155, i: 1 })
} },
```

### 11. Se configura el ChunkSize

Como la base de datos que se esta manejando contiene poca información el sharding no va a funcionar correctamente (toda la información se almacena en un solo chunk y no es posible evidenciar la distribución de datos). Para solucionar esto, únicamente con fines académicos, se modifica el chunkSize de 64mb a 1mb de la siguiente manera:

```
[direct: mongos] demos> use config
switched to db config
[direct: mongos] config> db.settings.updateOne( { _id: "chunksize" }, { $set: { value: 1 } }, {upsert:true});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
```

**12. Se importa la BD “ATP_Finals_2023”**

```
C:\Users\USUARIO>mongorestore --host localhost:1000 --db ATP_Finals_2023 C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\
BDA\Actividades_BDA\BD\ATP_Finals_2023
2023-12-16T21:01:59.240-0500    The --db and --collection flags are deprecated for this use-case; please use --nsInclude
 instead, i.e. with --nsInclude=${DATABASE}.${COLLECTION}
2023-12-16T21:01:59.497-0500    building a list of collections to restore from C:\Users\USUARIO\OneDrive\Documentos\Iber
o\6to\BDA\Actividades_BDA\BD\ATP_Finals_2023 dir
2023-12-16T21:01:59.498-0500    reading metadata for ATP_Finals_2023.deportistas from C:\Users\USUARIO\OneDrive\Document
os\Ibero\6to\BDA\Actividades_BDA\BD\ATP_Finals_2023\deportistas.metadata.json
2023-12-16T21:01:59.499-0500    reading metadata for ATP_Finals_2023.partidos from C:\Users\USUARIO\OneDrive\Documentos\
Ibero\6to\BDA\Actividades_BDA\BD\ATP_Finals_2023\partidos.metadata.json
2023-12-16T21:01:59.507-0500    restoring to existing collection ATP_Finals_2023.deportistas without dropping
2023-12-16T21:01:59.507-0500    restoring to existing collection ATP_Finals_2023.partidos without dropping
2023-12-16T21:01:59.507-0500    restoring ATP_Finals_2023.deportistas from C:\Users\USUARIO\OneDrive\Documentos\Ibero\6t
o\BDA\Actividades_BDA\BD\ATP_Finals_2023\deportistas.bson
2023-12-16T21:01:59.507-0500    restoring ATP_Finals_2023.partidos from C:\Users\USUARIO\OneDrive\Documentos\Ibero\6to\B
DA\Actividades_BDA\BD\ATP_Finals_2023\partidos.bson
2023-12-16T21:01:59.527-0500    finished restoring ATP_Finals_2023.partidos (15 documents, 0 failures)
2023-12-16T21:01:59.542-0500    finished restoring ATP_Finals_2023.deportistas (9 documents, 0 failures)
2023-12-16T21:01:59.542-0500    no indexes to restore for collection ATP_Finals_2023.deportistas
2023-12-16T21:01:59.542-0500    no indexes to restore for collection ATP_Finals_2023.partidos
2023-12-16T21:01:59.542-0500    24 document(s) restored successfully. 0 document(s) failed to restore.
```

**13. Se comprueba que la base de datos se importo en el enrutador**

```
[direct: mongos] ATP_Finals_2023> db.deportistas.find()
[
  {
    _id: ObjectId("656b883e3b6e5cc115f463ad"),
    nombre: 'Novak Djokovic',
    sencillos_ranking: 1,
    edad: 36,
    fecha_nacimiento: ISODate("1987-05-22T00:00:00.000Z"),
    profesional_desde: 2003,
    peso_lbs: 170,
    peso_kg: 77,
    altura_cm: 188,
    lugar_nacimiento: 'Belgrado (Serbia)',
    juego: 'Diestro, Dos Manos Reves',
    entrenadores: [ 'Goran Ivanisevic' ]
  },
  {
    _id: ObjectId("656b883e3b6e5cc115f463ae"),
    nombre: 'Andrey Rublev',
    sencillos_ranking: 5,
```

**14. Se comprueba la distribución de datos entre las particiones:**

```
[direct: mongos] ATP_Finals_2023> db.deportistas.getShardDistribution()
Shard replica_atp_fragmento1 at replica_atp_fragmento1/localhost:26017,localhost:26117,localhost:26217
{
  data: '2KiB',
  docs: 9,
  chunks: 1,
  'estimated data per chunk': '2KiB',
  'estimated docs per chunk': 9
}
---
Totals
{
  data: '2KiB',
  docs: 9,
  chunks: 1,
  'Shard replica_atp_fragmento1': [
    '100 % data',
    '100 % docs in cluster',
    '297B avg obj size on shard'
  ]
}
```

```
[direct: mongos] ATP_Finals_2023> db.partidos.getShardDistribution()
Shard replica_atp_fragmento1 at replica_atp_fragmento1/localhost:26017,localhost:26117,localhost:26217
{
  data: '1KiB',
  docs: 15,
  chunks: 1,
  'estimated data per chunk': '1KiB',
  'estimated docs per chunk': 15
}
---
Totals
{
  data: '1KiB',
  docs: 15,
  chunks: 1,
  'Shard replica_atp_fragmento1': [
    '100 % data',
    '100 % docs in cluster',
    '103B avg obj size on shard'
  ]
}
[direct: mongos] ATP_Finals_2023>
```

Como la información es muy poca se concentra en una única partición.

### 15. Se agrega información aleatoria

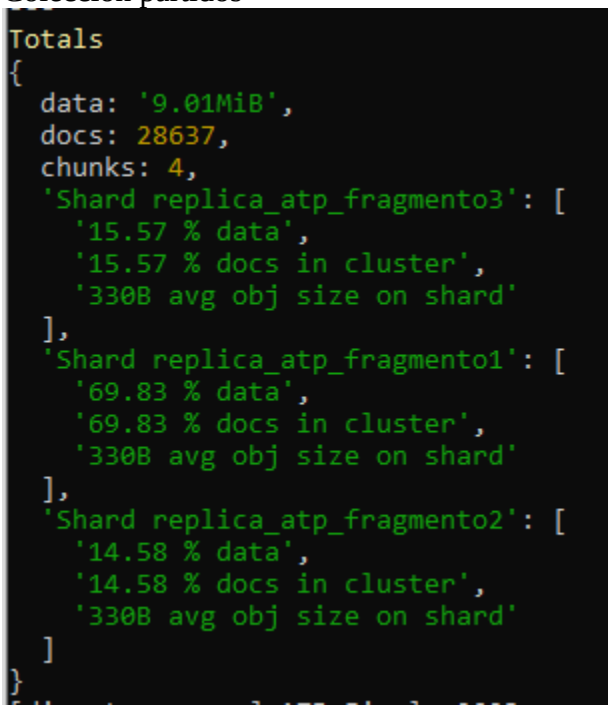
Esto con el fin de incrementar el tamaño en mb de la base de datos y comprobar que las particiones funcionan correctamente para las dos colecciones.

```
[direct: mongos] ATP_Finals_2023> for (var i = 0; i < 10000; i++) { db.partidos.insert({ campo1: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. ", campo2: "Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. ", campo3: 1234567890, campo4: ["elemento1", "elemento2", "elemento3", "elemento4"], campo5: { subcampo1: "Otra información", subcampo2: 9876543210 } }); }
```

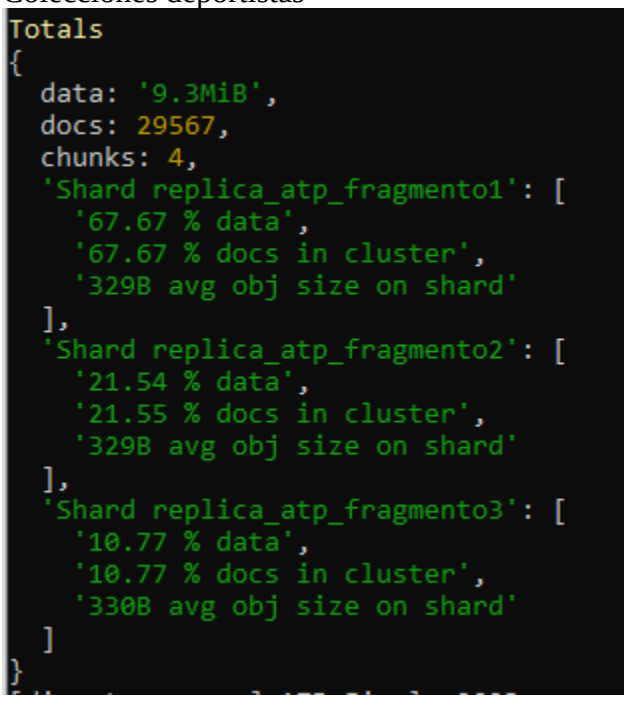
```
[direct: mongos] ATP_Finals_2023> for (var i = 0; i < 20000; i++) { db.deportistas.insert({ campo1: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. ", campo2: "Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. ", campo3: 1234567890, campo4: ["elemento1", "elemento2", "elemento3", "elemento4"], campo5: { subcampo1: "Otra información", subcampo2: 9876543210 } }); }
```

**16. Se comprueba que los datos se encuentren en distintas particiones**

Colección partidos




Colecciones deportistas



Nota: Como se puede observar, la distribución de datos no es uniforme, esto es debido a que se utilizo el _id como clave para las particiones, la cual no permite que se distribuyan los datos correctamente y por lo tanto es considerado como una mala práctica.

## Conclusiones

En esta actividad se ha desarrollado y estudiado el particionamiento de una base de datos (con manejo de réplicas por cada partición), teniendo como objetivo mejorar el desempeño a través de un escalamiento horizontal según los requerimientos no funcionales planteados en el documento RequerimientosNoFuncionales_IEEE.pdf.

## Lista de referencias


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

Mujica, E. (2023). Especificación de Requerimientos no Funcionales. https://github.com/Elianfm/actividad_mongodb/blob/d9cf02a02a9c3dd30ff9417582aee448e3a37f6d/Actividad%203%20%26%204/RequerimientosNoFuncionales_IEEEE.pdf

Corporación Universitaria Iberoamericana (2023). Tutorial de operaciones básicas de particionamiento en Mondodb. https://aulavirtual.ibero.edu.co/repositorio/Cursos-Matriz/Ingenieria/Ingenieria-sistemas/Bases-datos-avanzadas/presentacion-5/1.html

MongoDB (2023). Modify Chunk Size in a Sharded Cluster. Modify Chunk Size in a Sharded Cluster — MongoDB Manual