



INFORME DE TALLER

I. PORTADA

Tema:	Almacenamiento distribuido mediante sharding
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	5 – A
Alumnos participantes:	Ases Tiban Jeremy Damian Palate Moreta Kevin Damian Poveda Gómez William Alberto Pullupaxi Chango Daniel
Asignatura:	Sistema de Base de Datos Distribuidas
Docente:	Ing. Jose Caiza, Mg.

II. INFORME DE GUÍA PRÁCTICA

2.1 Objetivos

General:

Comprender cómo funciona la distribución de datos en MongoDB mediante la creación de un entorno con varios servidores que trabajan juntos para guardar la información de forma ordenada y eficiente.

Específicos:

- Configurar paso a paso los diferentes componentes necesarios para dividir y organizar los datos en varias partes.
- Comprobar cómo se guardan y reparten los registros en los distintos servidores al realizar consultas o insertar información.

2.2 Modalidad

Presencial.

2.3 Tiempo de duración

Presenciales: 3 horas.

No presenciales: N/A.

2.4 Instrucciones

1. Instalar MongoDB
2. Ejecutar instancias de MongoDB para shards
3. Inicializa replica sets
4. Ejecutar el servidor de configuración
6. Configurar los shards desde mongos
7. Crear base de datos y colección shardeada
8. Insertar datos y verificar distribución

2.5 Listado de equipos, materiales y recursos

- Computadora.

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:

- ☐ Plataformas educativas
- ☒ Simuladores y laboratorios virtuales
- ☐ Aplicaciones educativas
- ☐ Recursos audiovisuales
- ☐ Gamificación
- ☐ Inteligencia Artificial



Otros (Especifique): _____

2.6 Actividades desarrolladas

Se instalará el Mongo Community desde la pagina principal de Mongo como se ve en la figura 1

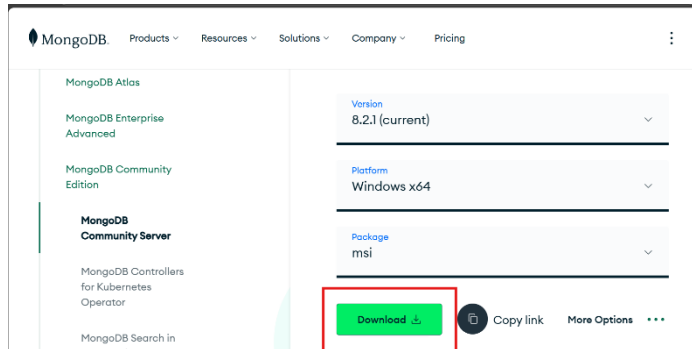


Fig 1 Descarga de Mongo Community desde la página oficial de Mongo

En la Figura 2 crearemos los directorios necesarios, tanto para el shard1, shard2 y para la configuración

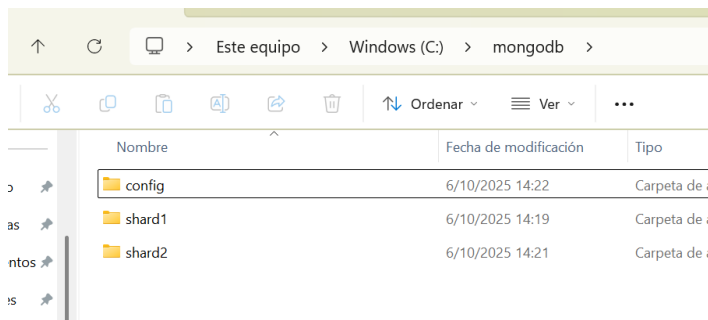


Fig 2 Creación de directorios necesarios

Ejecutaremos las instancias de MongoDB para los shards, tanto para shard1 como para shard2 (Figura 3)



Fig 3 Ejecucion de instancia tanto para shard1 y shard2



Iniciaremos las replicas sets para shard1 y shard2 con los puertos 27018 para shard1 (Figura 4) y 27019 para shard2 (Figura 5)

Shard1

```
C:\Windows\System32>mongosh --port 27018
Current Mongosh Log ID: 68e41a2d7a8542a75ccea3
Connecting to:      mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB:      8.0.13
Using Mongosh:       2.5.8

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

-----
The server generated these startup warnings when booting
2025-10-06T14:27:40.709-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> rs.initiate({
...   _id: "shard1",
...   members: [{ _id: 0, host: "localhost:27018" }]
... })
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1759779399, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    },
  },
  operationTime: Timestamp({ t: 1759779399, i: 1 })
}
shard1 [direct: secondary] test>
```

Fig 4 Inicialización de réplica set en shard1 con puerto 27018

Shard2

```
mongosh mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000

C:\Windows\System32>mongosh --port 27019
Current Mongosh Log ID: 68e41afba09b714b8ecea3
Connecting to:      mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB:      8.0.13
Using Mongosh:       2.5.8

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

-----
The server generated these startup warnings when booting
2025-10-06T14:33:22.941-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> rs.initiate({
...   _id: "shard2",
...   members: [{ _id: 0, host: "localhost:27019" }]
... })
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1759779586, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    },
  },
  operationTime: Timestamp({ t: 1759779586, i: 1 })
}
shard2 [direct: secondary] test>
```

Fig 5 Inicialización de réplica set en shard2 con puerto 27019



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

CICLO ACADÉMICO: AGOSTO 2025 – ENERO 2026



Ahora en la Figura 6 se ejecutará el servidor de configuración

```
Administrador: Símbolo del sistema - mongod --config --replSet configReplSet --port 27020 --dbpath "C:\mongodb\config" --bind_ip localhost
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found.", "stats": {}, "cm
["aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}]]
{"t": {"date": "2025-10-06T14:43:19.004-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [config.transactions] not found.", "stats": {}
["aggregate": "transactions", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "config"}]]
{"t": {"date": "2025-10-06T14:43:19.004-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [config.image.collection] not found.", "stats": {}
["aggregate": "image_collection", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "config"}]]
{"t": {"date": "2025-10-06T14:43:19.297-05:00"}, "s": "I", "c": "I", "id": 4939308, "svc": "S", "ctx": "monitoring-keys-for-IMAC", "msg": "Failed to refresh key cache", "attr
{"error": "ReadConcernMajorityNotAvailableRet Read concern majority reads are currently not possible.", "errorCode": 13000}}
{"t": {"date": "2025-10-06T14:43:20.003-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [local.oplog.rs] not found.", "stats": {}
["aggregate": "oplog.rs", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "local"}]]
{"t": {"date": "2025-10-06T14:43:20.003-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [config.transactions] not found.", "stats": {}
["aggregate": "transactions", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "config"}]]
{"t": {"date": "2025-10-06T14:43:20.003-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [config.image.collection] not found.", "stats": {}
["aggregate": "image_collection", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "config"}]]
{"t": {"date": "2025-10-06T14:43:21.001-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [config.transactions] not found.", "stats": {}
["aggregate": "transactions", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "config"}]]
{"t": {"date": "2025-10-06T14:43:21.001-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [config.transactions] not found.", "stats": {}
["aggregate": "transactions", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "config"}]]
{"t": {"date": "2025-10-06T14:43:21.001-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
de:26, "codeName": "NamespaceNotFound", "errmsg": "Unable to retrieve storageStats in $collStats stage :: caused by :: Collection [config.image.collection] not found.", "stats": {}
["aggregate": "image_collection", "cursor": {}, "pipeline": [{"$collStats": {"storageStats": {"waitForLock": false, "numericOnly": true}}}], "$db": "config"}]]
{"t": {"date": "2025-10-06T14:43:22.001-05:00"}, "s": "W", "c": "QUERY", "id": 23799, "svc": "S", "ctx": "ftdc", "msg": "Aggregate command executor error", "attr": {"error": {"co
```

Fig 6 Ejecución del servidor de configuración en CMD

En la figura 7 iniciaremos la replica set en el servidor de configuración mediante el puerto 27020

```
C:\Windows\System32>mongosh --port 27020
Current Mongosh Log ID: 68e41c1f297d987d1ceba3
Connecting to:
  mongodb://127.0.0.1:27020/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB:
  8.0.13
Using Mongosh:
  2.5.8

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

-----
The server generated these startup warnings when booting
2025-10-06T14:42:57.697-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> rs.initiate({
...
  _id: "configReplSet",
...
  members: [{ _id: 0, host: "localhost:27020" }]
...
})
ok: 1,
  clusterTime: {
    clusterTime: Timestamp({ t: 1759779917, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA', 0),
      keyId: Long('0')
    },
    operationTime: Timestamp({ t: 1759779917, i: 1 })
  }
configReplSet [direct: secondary] test>
```

Fig 7 Inicializacion de la replica set de configuración en el puerto 27020

En la Figura 8 ejecutaremos el router de MongoDB

```
Administrador: Símbolo del sistema - mongos --configdb configReplSet/localhost:27020 --port 27017 --bind_ip localhost
{"t": {"date": "2025-10-06T14:47:48.918-05:00"}, "s": "I", "c": "SHARDING", "id": 22842, "svc": "R", "ctx": "mongosMain", "msg": "Waiting for signing keys, sleeping before checki
ng again", "attr": {"signingKeysCheckIntervalSeconds": 1}}
{"t": {"date": "2025-10-06T14:47:49.932-05:00"}, "s": "W", "c": "FTDC", "id": 23911, "svc": "R", "ctx": "mongosMain", "msg": "FTDC is disabled because neither --logpath nor
set parameter 'diagnosticDataCollectionDirectoryPath' are specified."}
{"t": {"date": "2025-10-06T14:47:49.935-05:00"}, "s": "I", "c": "CONNPPOOL", "id": 22576, "svc": "R", "ctx": "TaskExecutorPool-0", "msg": "Connecting", "attr": {"hostAndPort": "local
host:27020"}}
{"t": {"date": "2025-10-06T14:47:49.953-05:00"}, "s": "I", "c": "TXN", "id": 8918609, "svc": "R", "ctx": "ClusterServerParameterRefresher", "msg": "Close internal transaction
error handling step", "attr": {"nextStep": "do not retry", "txnInfo": {"execContext": "own session", "sessionId": {"_id": {"$uid": "s604869c-7c4e-4d93-9c75-6274165c17f9"}, "
uid": {"binary": "base64":"A70D0p048Sa+TImH53CeuQERk5Np3WZG3H5uFuH", "subType": "0"}}, "txnNumber": 0, "autoCommit": false, "state": "started commit", "lastOperationTime": {"t
s": Timestamp(1759780069, 1)}, "latestResponseHasTransientTransactionErrorLabel": false, "deadline": "none", "writeConcern": {"w": "majority", "wtimeout": 0}, "readConcern": {"level
": "snapshot"}, "APIParameters": {}, "canceled": false, "attempts": 1, "commitError": "InvalidOptions: commitTransaction can only be run within a session", "commitWCErrors": "OK", "err
orHandler": "runCommit"}}
{"t": {"date": "2025-10-06T14:47:49.953-05:00"}, "s": "W", "c": "CONTROL", "id": 7410719, "svc": "R", "ctx": "ClusterServerParameterRefresher", "msg": "Could not refresh cluster s
erver parameters from config servers due to failure in getCVAndClusterParametersFromConfigServer. Will retry after refresh interval", "attr": {"ex": {"code": 72, "codeName": "In
validOptions", "errmsg": "Command error committing internal transaction :: caused by :: commitTransaction can only be run within a session"}}}
{"t": {"date": "2025-10-06T14:47:50.447-05:00"}, "s": "W", "c": "FTDC", "id": 23718, "svc": "R", "ctx": "mongosMain", "msg": "Failed to initialize Performance Counters for FT
DC", "attr": {"error": {"code": 179, "codeName": "WindowsPdhError", "errmsg": "PdhAddEnglishCounter failed with 'El objeto especificado no se encontró en el equipo.'}}}
{"t": {"date": "2025-10-06T14:47:50.447-05:00"}, "s": "I", "c": "FTDC", "id": 20625, "svc": "R", "ctx": "mongosMain", "msg": "Initializing full-time diagnostic data capture",
"attr": {"dataDirectory": ""}}
{"t": {"date": "2025-10-06T14:47:50.458-05:00"}, "s": "I", "c": "HEALTH", "id": 5936511, "svc": "R", "ctx": "mongosMain", "msg": "No active health observers are configured."}
{"t": {"date": "2025-10-06T14:47:50.458-05:00"}, "s": "I", "c": "HEALTH", "id": 5936502, "svc": "R", "ctx": "mongosMain", "msg": "The fault manager initial health checks have com
pleted", "attr": {"state": "OK"}}
{"t": {"date": "2025-10-06T14:47:50.458-05:00"}, "s": "I", "c": "HEALTH", "id": 5936503, "svc": "R", "ctx": "mongosMain", "msg": "Fault manager changed state", "attr": {"state": "O
K"}}
{"t": {"date": "2025-10-06T14:47:50.451-05:00"}, "s": "I", "c": "NETWORK", "id": 23815, "svc": "R", "ctx": "listener", "msg": "Listening on", "attr": {"address": "127.0.0.1:27017"}}
{"t": {"date": "2025-10-06T14:47:50.451-05:00"}, "s": "I", "c": "NETWORK", "id": 23816, "svc": "R", "ctx": "listener", "msg": "Waiting for connections", "attr": {"port": "27017", "ssl
": "off"}}
{"t": {"date": "2025-10-06T14:47:50.451-05:00"}, "s": "I", "c": "SHARDING", "id": 8423485, "svc": "R", "ctx": "mongosMain", "msg": "Mongos startup complete", "attr": {"Summary of tim
e elapsed": {"Statistics": {"Set up periodic runner": "0 ms", "Set up online certificate status protocol manager": "0 ms", "Set up transport layer listener": "0 ms", "Initialize glo
bal sharding state": "5 ms", "Reset the shard registry config connection string": "0 ms", "Load global settings from config server": "11 ms", "Wait for signing keys": "1000 ms", "
Pre-cache mongos routing info": "3 ms", "Warm up connections to shards": "2 ms", "Refresh the balancer configuration": "4 ms", "Update read write concern defaults": "4 ms", "Start
mongos FTDC": "516 ms", "Set up mongos engine": "1 ms", "Build user and roles graph": "0 ms", "Run mongosServer topology elapsed time": "1150 ms"}}}}
{"t": {"date": "2025-10-06T14:47:50.455-05:00"}, "s": "I", "c": "SH_REFRESH", "id": 4619902, "svc": "R", "ctx": "CatalogCache-0", "msg": "Collection has found to be unsharded after r
efresh", "attr": {"namespace": "config.system.sessions", "durationMillis": 3}}
{"t": {"date": "2025-10-06T14:47:50.455-05:00"}, "s": "I", "c": "CONTROL", "id": 20712, "svc": "R", "ctx": "LogicalSessionCacheReap", "msg": "Sessions collection is not set up;
waiting until next sessions reap interval", "attr": {"error": "NamespaceNotFound: Expected collection config.system.sessions to be sharded"}}
{"t": {"date": "2025-10-06T14:47:50.455-05:00"}, "s": "I", "c": "CONTROL", "id": 20719, "svc": "R", "ctx": "LogicalSessionCacheRefresh", "msg": "Failed to refresh session cache,
will try again at the next refresh interval", "attr": {"error": "NamespaceNotFound: Expected collection config.system.sessions to be sharded"}}
```

Fig 8 Ejecución de MongoDB Router



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO 2025 – ENERO 2026



Abriremos el mongo con el puerto 27017 y agregaremos los shards, tanto el shard1 como el shard2 (Figura 9)

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
C:\Windows\System32>mongosh --port 27017
Current Mongosh Log ID: 68e41d3ba93d74ee5eacea3
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB: 8.0.13
Using Mongosh: 2.5.8

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

-----
The server generated these startup warnings when booting
2025-10-06T14:47:48.900-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

[direct: mongos] test> sh.addShard("shard1/localhost:27018")
{
  shardAdded: 'shard1',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1759780230, i: 21 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1759780230, i: 21 })
}
[direct: mongos] test> sh.addShard("shard2/localhost:27019")
{
  shardAdded: 'shard2',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1759780328, i: 24 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1759780328, i: 18 })
}
[direct: mongos] test>
```

Fig 9 Agregacion de shards en el puerto 27017

Creación de base de datos y conexión shardeada (Figura 10)

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
[direct: mongos] test> use universidad
switched to db universidad
[direct: mongos] universidad> sh.enableSharding("universidad")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1759780690, i: 8 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1759780690, i: 5 })
}
[direct: mongos] universidad> db.createCollection("estudiantes")
{ ok: 1 }
[direct: mongos] universidad> sh.shardCollection("universidad.estudiantes", { "codigo": 1 })
{
  collectionsharded: 'universidad.estudiantes',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1759780725, i: 45 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAAA', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1759780725, i: 45 })
}
[direct: mongos] universidad>
```

Fig 10 Creación de BD 'universidad' y colección shardeada



En la figura 11 Insertaremos datos necesarios para la verificación con un ciclo 'for'

```
[direct: mongos] universidad> use universidad
already on db universidad
[direct: mongos] universidad> for(let i=1; i<=1000; i++){
...   db.estudiantes.insertOne({codigo:i, nombre:"Estudiante_"+i})
... }
{
  acknowledged: true,
  insertedId: ObjectId('68e42024a93d74ee5ecec673')
```

Fig 11 Ciclo for para la inserción de datos a la BD

Verificaremos los datos en la BD mediante la distribución de los Shard

```
ca mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
[direct: mongos] universidad> db.estudiantes.getShardDistribution()
Shard shard2 at shard2/localhost:27019
{
  data: '60KiB',
  docs: 2000,
  chunks: 1,
  'estimated data per chunk': '60KiB',
  'estimated docs per chunk': 2000
}
---
Shard shard1 at shard1/localhost:27018
{
  data: '59KiB',
  docs: 998,
  chunks: 1,
  'estimated data per chunk': '59KiB',
  'estimated docs per chunk': 998
}
---
Totals
{
  data: '119KiB',
  docs: 2998,
  chunks: 2,
  'Shard shard2': [
    '50.5 % data',
    '66.71 % docs in cluster',
    '60B avg obj size on shard'
  ],
  'Shard shard1': [
    '49.49 % data',
    '33.28 % docs in cluster',
    '60B avg obj size on shard'
  ]
}
```

Fig 12 Distribucion de Shards

2.7 Resultados obtenidos

Se logró crear correctamente un entorno distribuido de bases de datos en MongoDB, compuesto por varios servidores conectados entre sí. La base de datos universidad fue configurada para trabajar con sharding, y la colección estudiante se dividió por el campo código. Al insertar los datos, se comprobó que la información se repartió entre los distintos servidores, mostrando cómo MongoDB distribuye la carga y mejora el rendimiento del sistema.

2.8 Habilidades blandas empleadas en la práctica

- ☐ Liderazgo
- ☒ Trabajo en equipo
- ☐ Comunicación asertiva



- ☐ La empatía
- ☐ Pensamiento crítico
- ☐ Flexibilidad
- ☐ La resolución de conflictos
- ☐ Adaptabilidad
- ☒ Responsabilidad

2.9 Conclusiones

La práctica demostró que MongoDB permite crecer de forma sencilla al agregar más servidores cuando sea necesario, lo que mejora la capacidad del sistema. Además, la distribución de los datos entre distintos nodos hace que las consultas sean más rápidas y equilibradas. Finalmente, al contar con varios servidores trabajando en conjunto, se aumenta la disponibilidad del sistema y se reducen los riesgos ante posibles fallos.

2.10 Recomendaciones

Se recomienda planificar con anticipación la estructura del sistema y el campo por el cual se dividirán los datos, para aprovechar al máximo el sharding y mantener un buen equilibrio en la distribución de la información entre los servidores.

2.11 Referencias bibliográficas

- [1] MongoDB Inc., “Sharding Introduction,” MongoDB Documentation, 2024. [En línea]. Disponible en: <https://www.mongodb.com/docs/manual/sharding/>
- [2] MongoDB Inc., “Sharded Cluster Components,” MongoDB Documentation, 2024. [En línea]. Disponible en: <https://www.mongodb.com/docs/manual/core/sharded-cluster-components/>