

Laboratorio #3 Mongo Replication

Abrimos consola en modo admin y ejecutamos los comandos “mongo” y “show dbs” esto nos mostrara las bases que hay actualmente corriendo en nuestro mongo principal

```
Administrador: Símbolo del sistema - mongo
Microsoft Windows [Versión 10.0.19045.2130]
(c) Microsoft Corporation. Todos los derechos reservados.

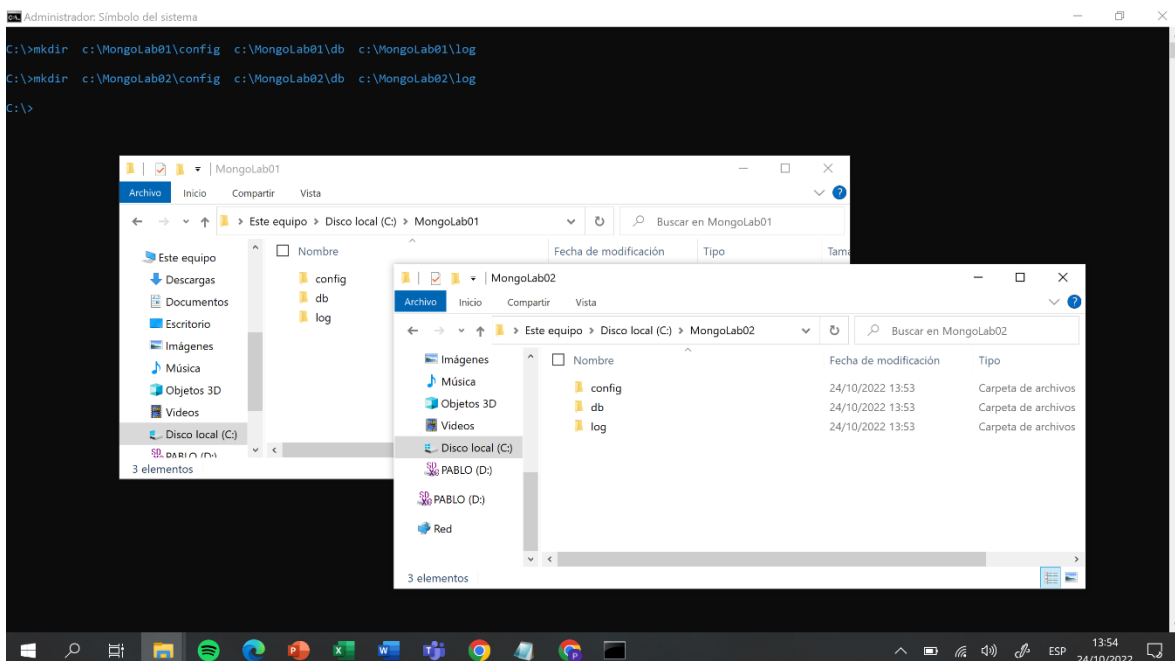
C:\Windows\system32>mongo
MongoDB shell version v5.0.13
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongod
Implicit session: session { "id" : UUID("f179f95b-3aa1-40d1-89ee-7f0cf7dec88c") }
MongoDB server version: 5.0.13

Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/

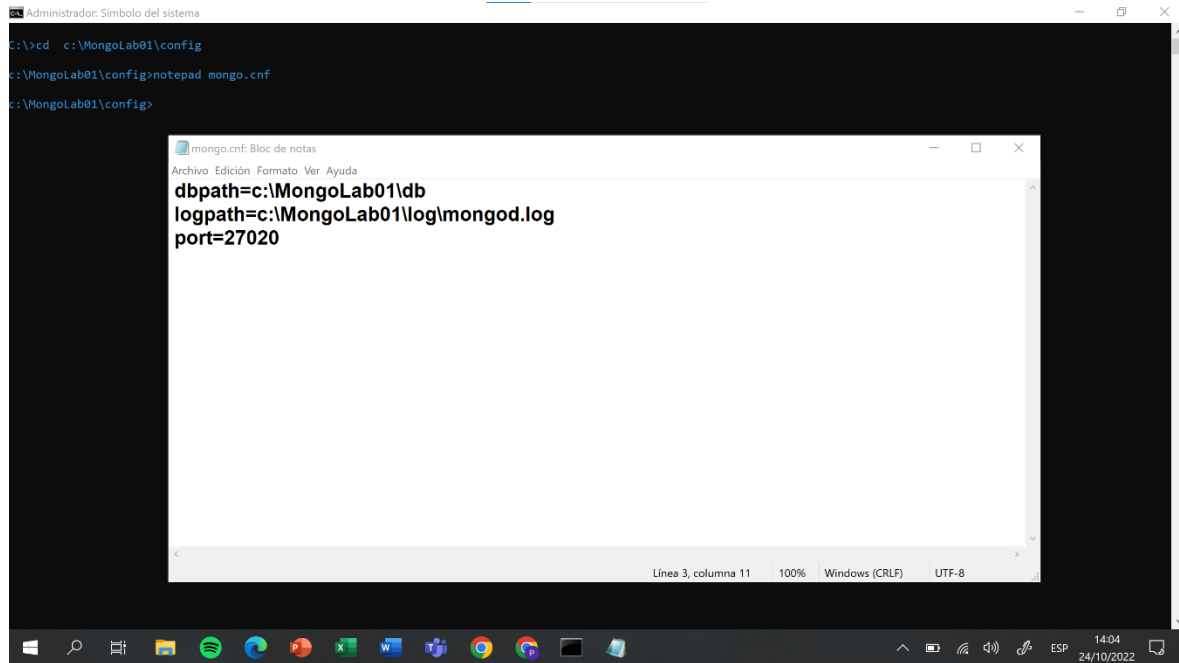
The server generated these startup warnings when booting:
2022-10-21T03:10:29.107-06:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

> show dbs;
admin    0.000GB
config  0.000GB
local   0.000GB
```

Debemos crear 2 carpetas en “C:” con las siguientes subcarpetas



Entramos a la subcarpeta 'config' de la carpeta 'MongoLab01' creada anteriormente en "C:" y creamos el archivo "mongo.cnf" con los siguientes atributos



```
C:\>cd c:\MongoLab01\config
C:\MongoLab01\config>notepad mongo.cnf
C:\MongoLab01\config>
```

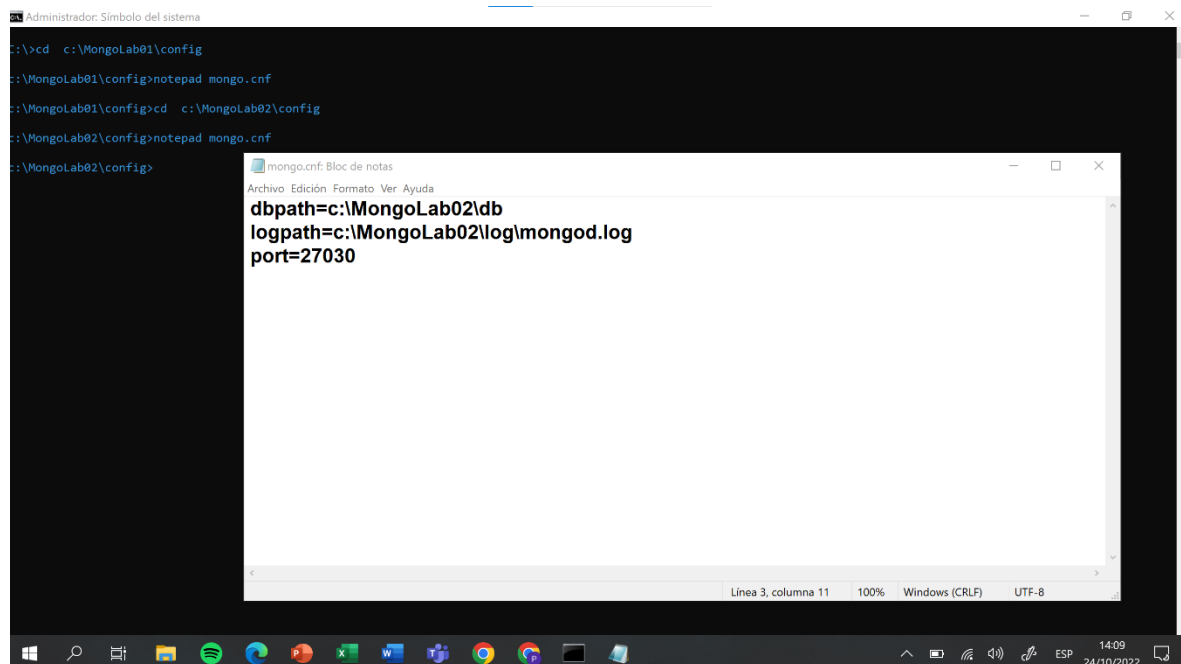
mongo.cnf: Bloc de notas

Archivo Edición Formato Ver Ayuda

dbpath=c:\MongoLab01\db
logpath=c:\MongoLab01\log\mongod.log
port=27020

Línea 3, columna 11 100% Windows (CRLF) UTF-8

Aplicamos lo mismo pero ahora en segunda carpeta que creamos



```
C:\>cd c:\MongoLab01\config
C:\MongoLab01\config>notepad mongo.cnf
C:\MongoLab01\config>cd c:\MongoLab02\config
C:\MongoLab02\config>notepad mongo.cnf
C:\MongoLab02\config>
```

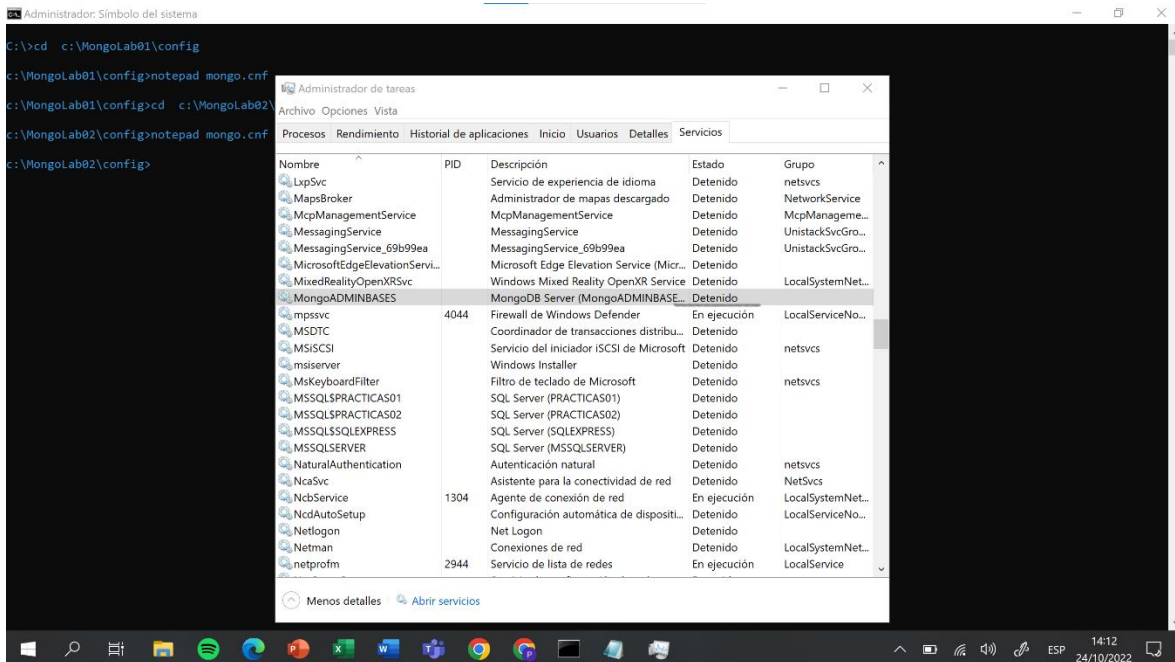
mongo.cnf: Bloc de notas

Archivo Edición Formato Ver Ayuda

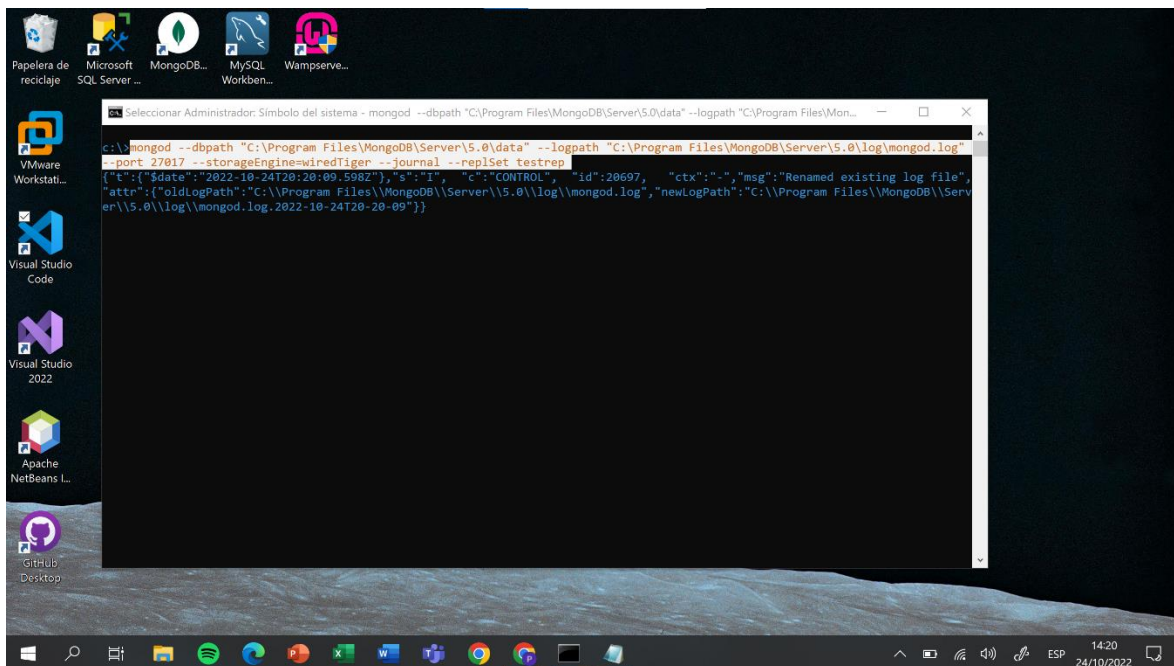
dbpath=c:\MongoLab02\db
logpath=c:\MongoLab02\log\mongod.log
port=27030

Línea 3, columna 11 100% Windows (CRLF) UTF-8

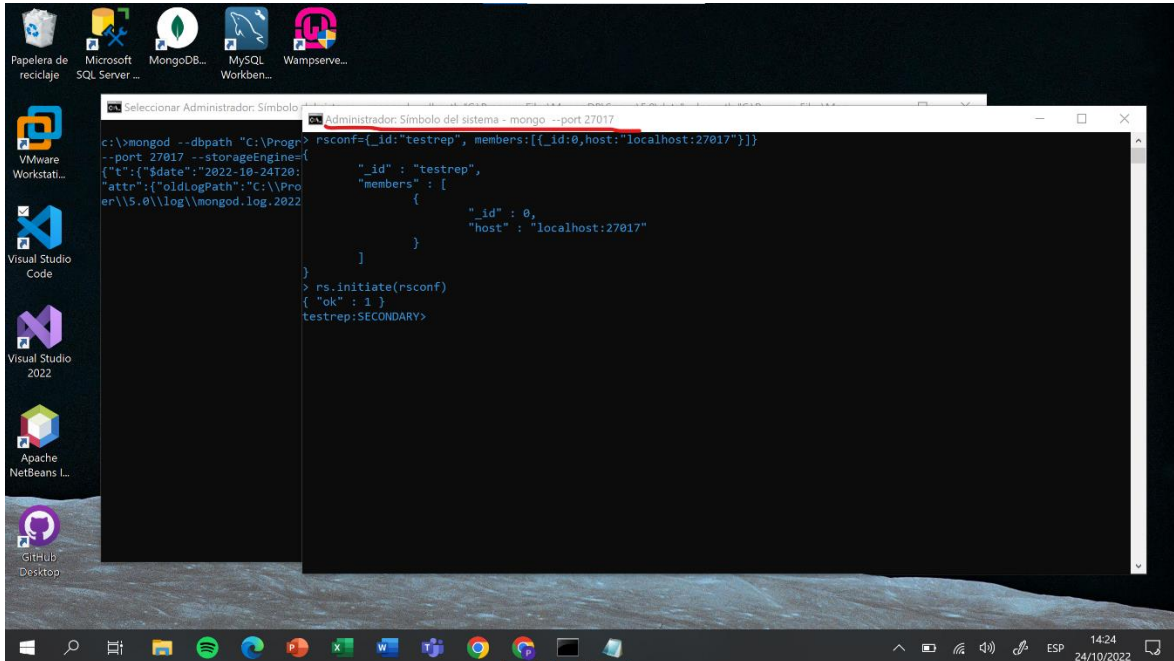
Debemos detener el servicio de Mongo para poder continuar con la replicación en mongo



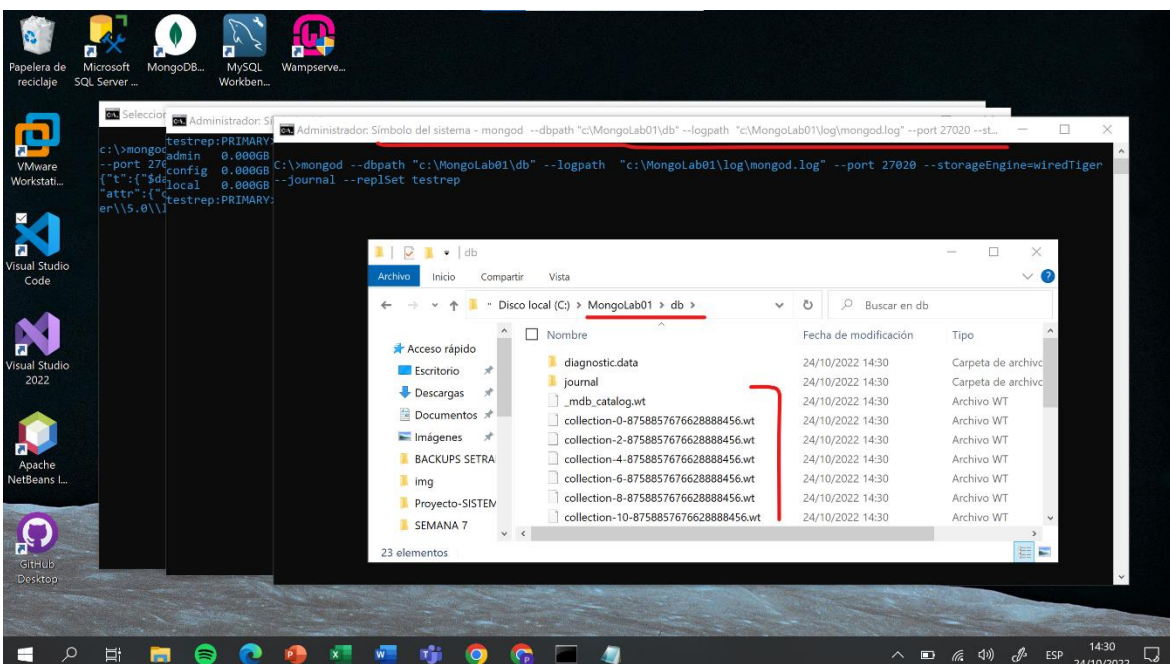
En el CMD modo admin escribimos el comando resaltado en la imagen, lo que va ser es crear la replica en la ruta de la instancia principal en este caso el nombre de la replica es “testrep”



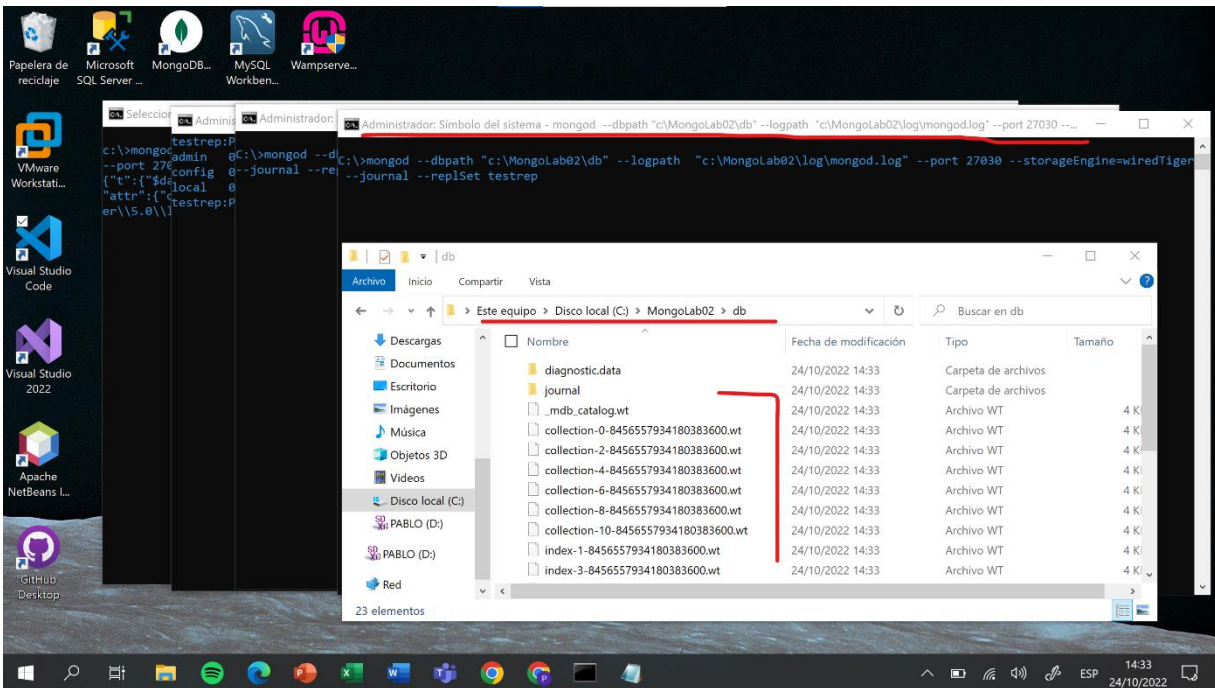
Ahora abrimos otra terminal con la instancia principal en este caso se ubica en el puerto 27017, escribimos los comandos en pantalla para la configuración de la replica y la iniciación de dicha configuración



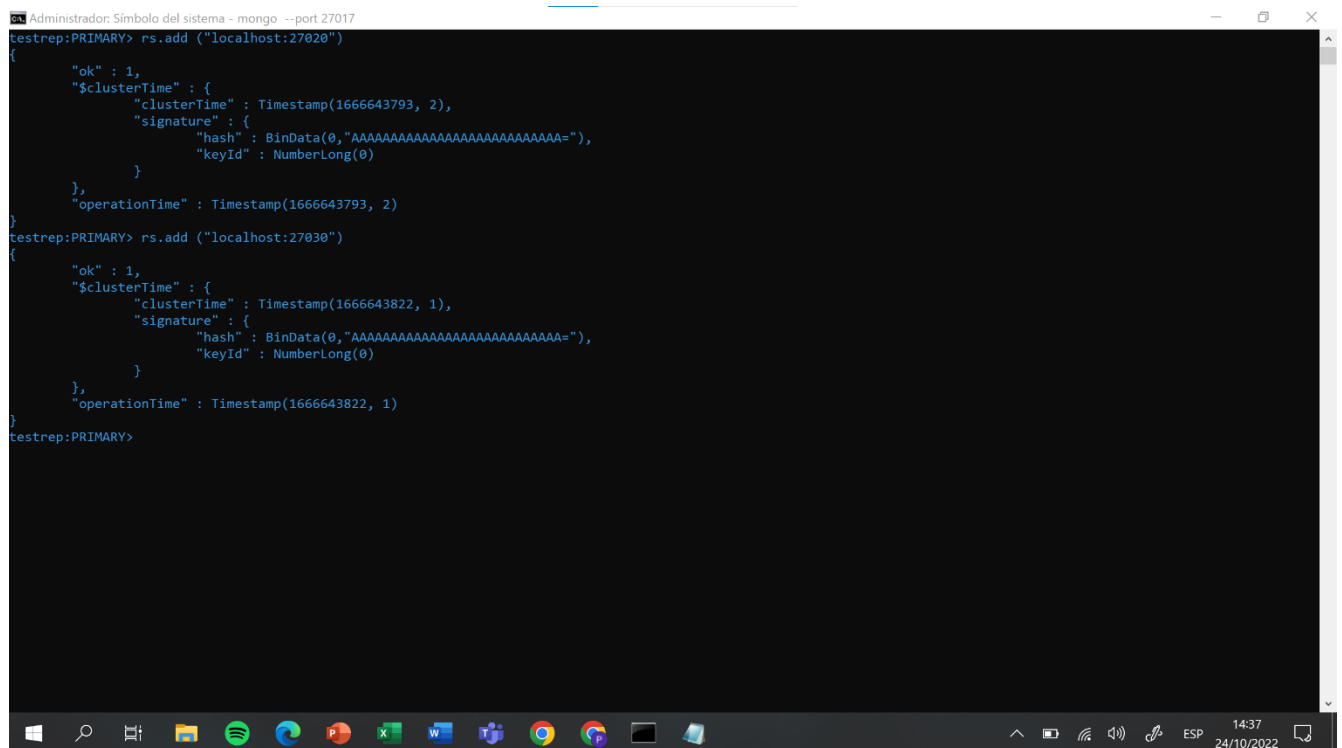
Ahora debemos aplicar el mismo comando de la replica solo que en la ruta 1 creada anteriormente en “C:” y con otro puerto diferente al de la instancia principal, en este caso el 27020, además observamos como se empiezan a crear los archivos de replicación en la carpeta que estaba vacía antes



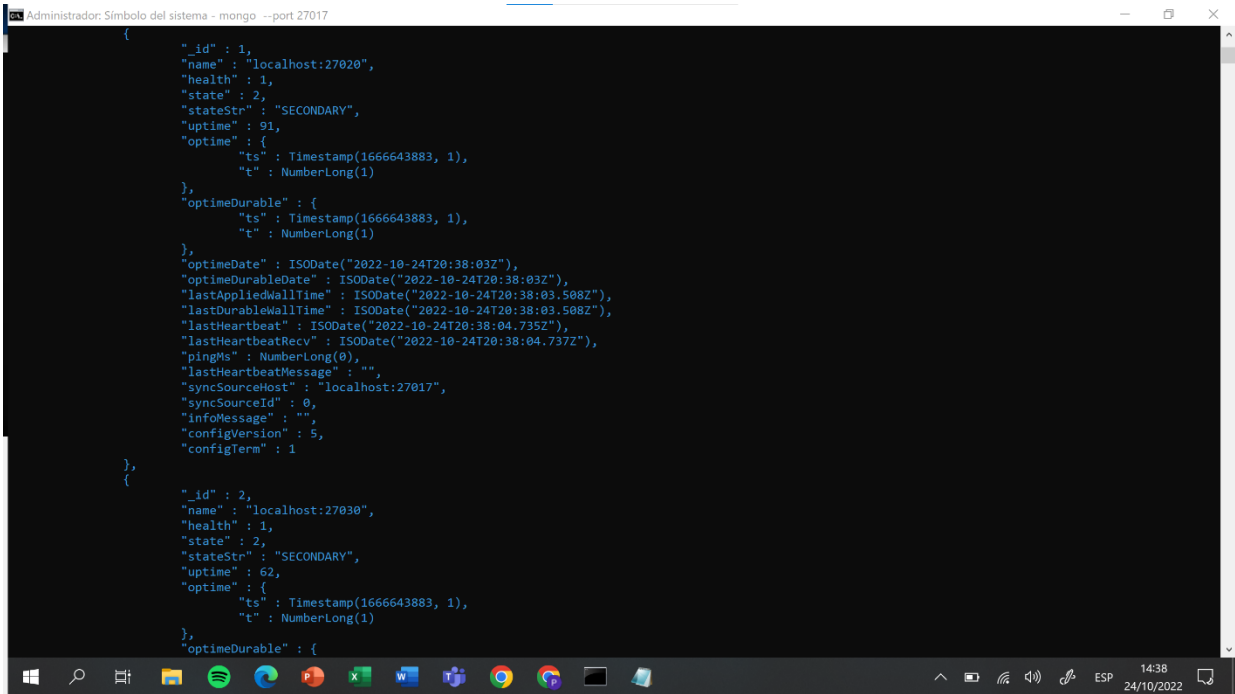
Aplicamos lo mismo solo que en la ruta 2 creada anteriormente en "C:" y le asignamos el puerto 27030



Siguiente, nos dirigimos a la instancia principal y añadimos las dos replicas que acabamos de configurar, deberá verse como la imagen, con los respectivos puertos asignados



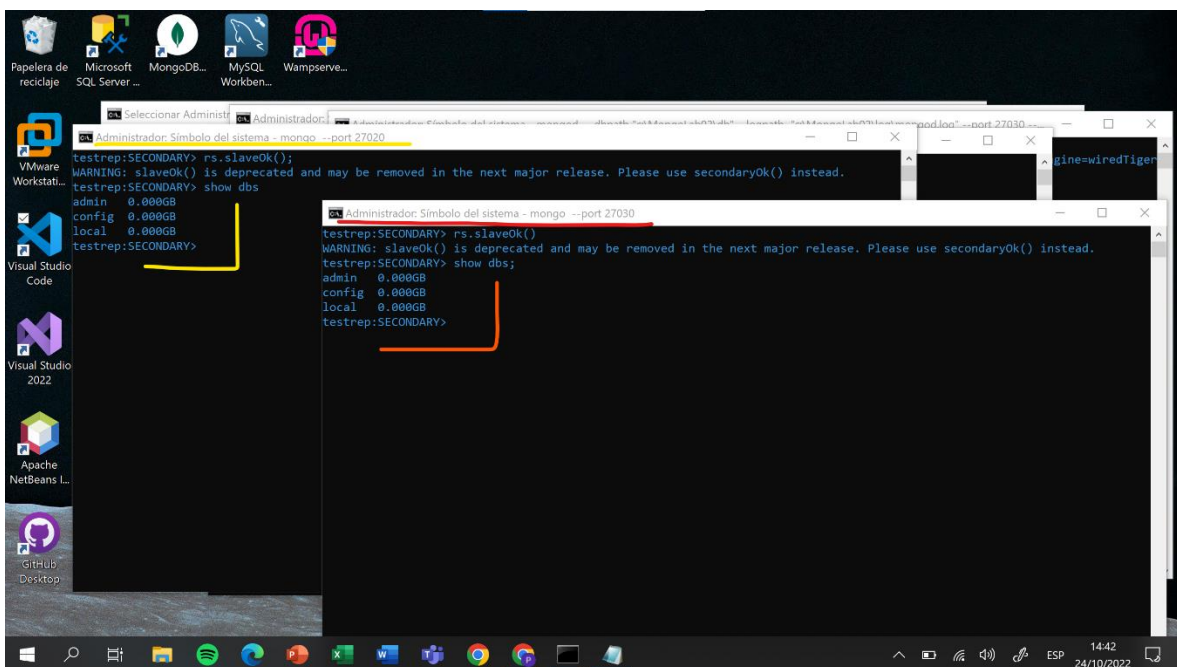
Con el comando “rs.status()” en la terminal de la instancia principal (puerto 27017) deberá salirnos lo siguiente, que indica que ya están añadidos a la lista las otras dos instancias además están como “SECUNDARIAS”



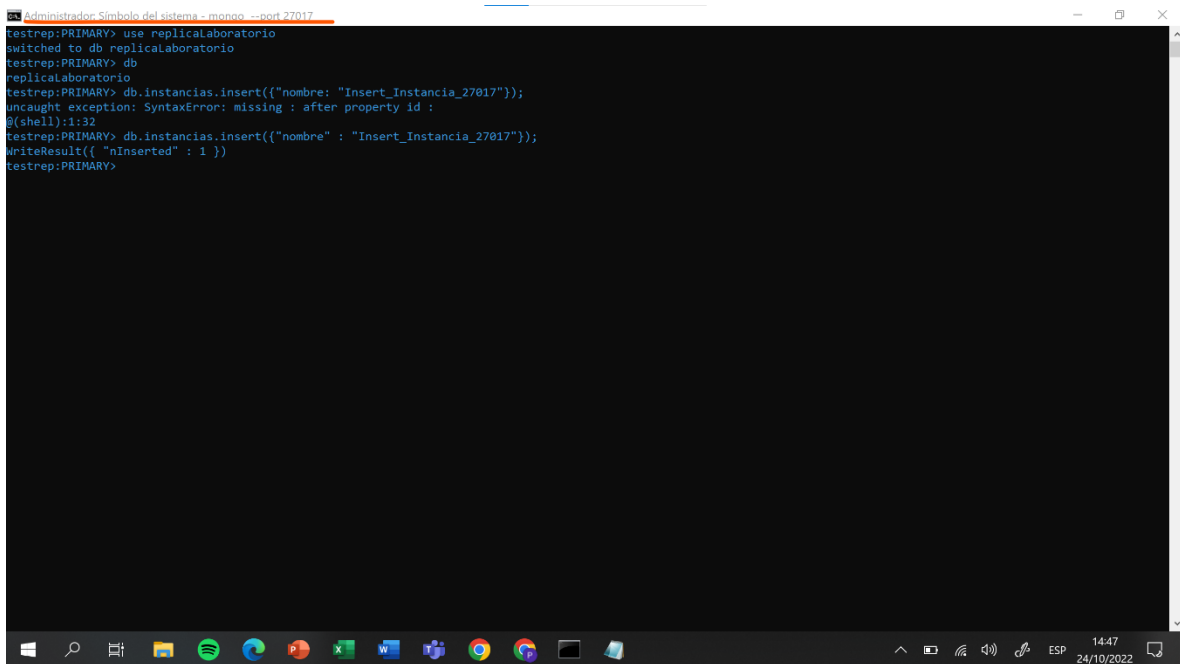
```
Administrator: Símbolo del sistema - mongo --port 27017
{
  "set": "rs0",
  "term": 1,
  "primary": "localhost:27020",
  "members": [
    {
      "_id": 1,
      "name": "localhost:27020",
      "health": 1,
      "state": 2,
      "stateStr": "SECONDARY",
      "uptime": 91,
      "optime": {
        "ts": Timestamp(1666643883, 1),
        "t": NumberLong(1)
      },
      "optimeDurable": {
        "ts": Timestamp(1666643883, 1),
        "t": NumberLong(1)
      },
      "optimeDate": ISODate("2022-10-24T20:38:03Z"),
      "optimeDurableDate": ISODate("2022-10-24T20:38:03Z"),
      "lastAppliedWallTime": ISODate("2022-10-24T20:38:03.508Z"),
      "lastDurableWallTime": ISODate("2022-10-24T20:38:03.508Z"),
      "lastHeartbeat": ISODate("2022-10-24T20:38:04.735Z"),
      "lastHeartbeatRecv": ISODate("2022-10-24T20:38:04.737Z"),
      "pingMs": NumberLong(0),
      "lastHeartbeatMessage": "",
      "syncSourceHost": "localhost:27017",
      "syncSourceId": 0,
      "infoMessage": "",
      "configVersion": 5,
      "configTerm": 1
    },
    {
      "_id": 2,
      "name": "localhost:27030",
      "health": 1,
      "state": 2,
      "stateStr": "SECONDARY",
      "uptime": 62,
      "optime": {
        "ts": Timestamp(1666643883, 1),
        "t": NumberLong(1)
      },
      "optimeDurable": {

```

Abrimos dos terminales más y iniciamos mongo en los puertos secundarios, aplicamos el comando “show dbs” y con esto se confirma que estas dos instancias están replicando a la principal

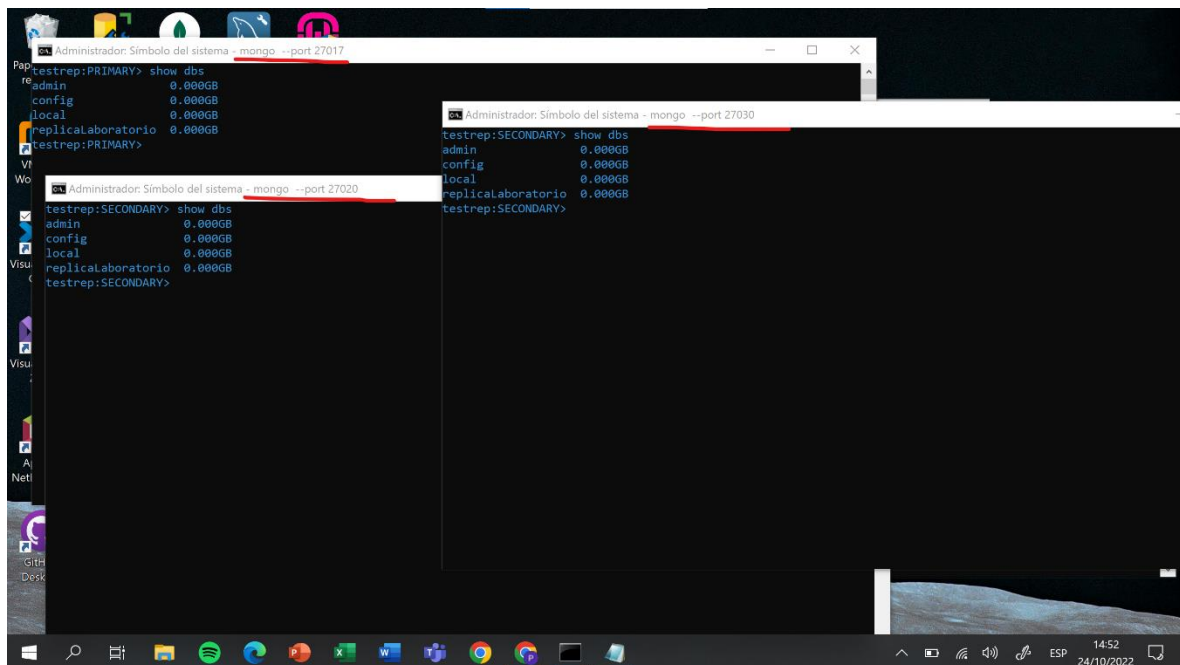


En la terminal de la instancia principal creamos una base de datos nueva y le insertamos datos como se muestra en la imagen



```
Administrador: Símbolo del sistema - mongo --port 27017
testrep:PRIMARY> use replicalaboratorio
switched to db replicalaboratorio
testrep:PRIMARY> db
replicalaboratorio
testrep:PRIMARY> db.instancias.insert({"nombre": "Insert_Instance_27017"});
uncaught exception: SyntaxError: missing : after property id :
@(:shell):1:32
testrep:PRIMARY> db.instancias.insert({"nombre": "Insert_Instance_27017"});
WriteResult({ "nInserted" : 1 })
testrep:PRIMARY>
```

En esta imagen confirmamos efectivamente que la base nueva esta replicada en las otras dos instancias secundarias del reflejo

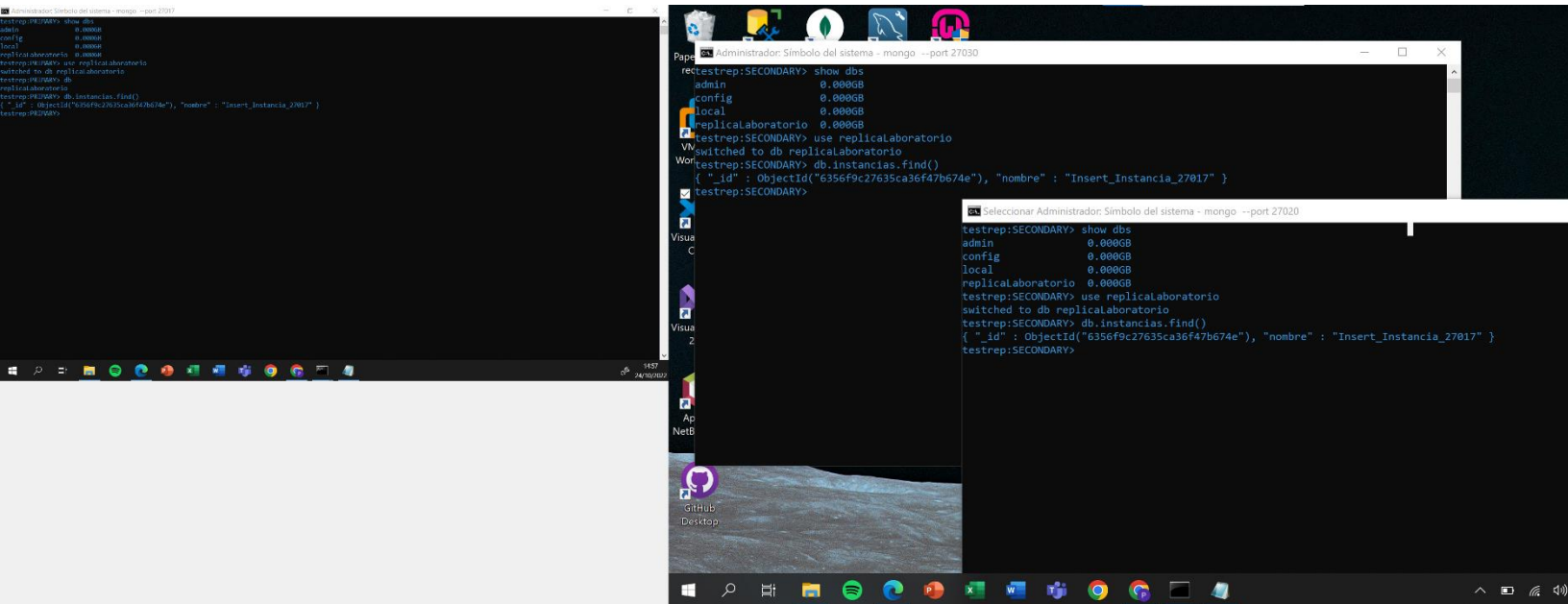


```
Administrador: Símbolo del sistema - mongo --port 27017
testrep:PRIMARY> show dbs
admin            0.000GB
config           0.000GB
local            0.000GB
replicalaboratorio 0.000GB
testrep:PRIMARY>

Administrador: Símbolo del sistema - mongo --port 27020
testrep:SECONDARY> show dbs
admin            0.000GB
config           0.000GB
local            0.000GB
replicalaboratorio 0.000GB
testrep:SECONDARY>

Administrador: Símbolo del sistema - mongo --port 27030
testrep:SECONDARY> show dbs
admin            0.000GB
config           0.000GB
local            0.000GB
replicalaboratorio 0.000GB
testrep:SECONDARY>
```

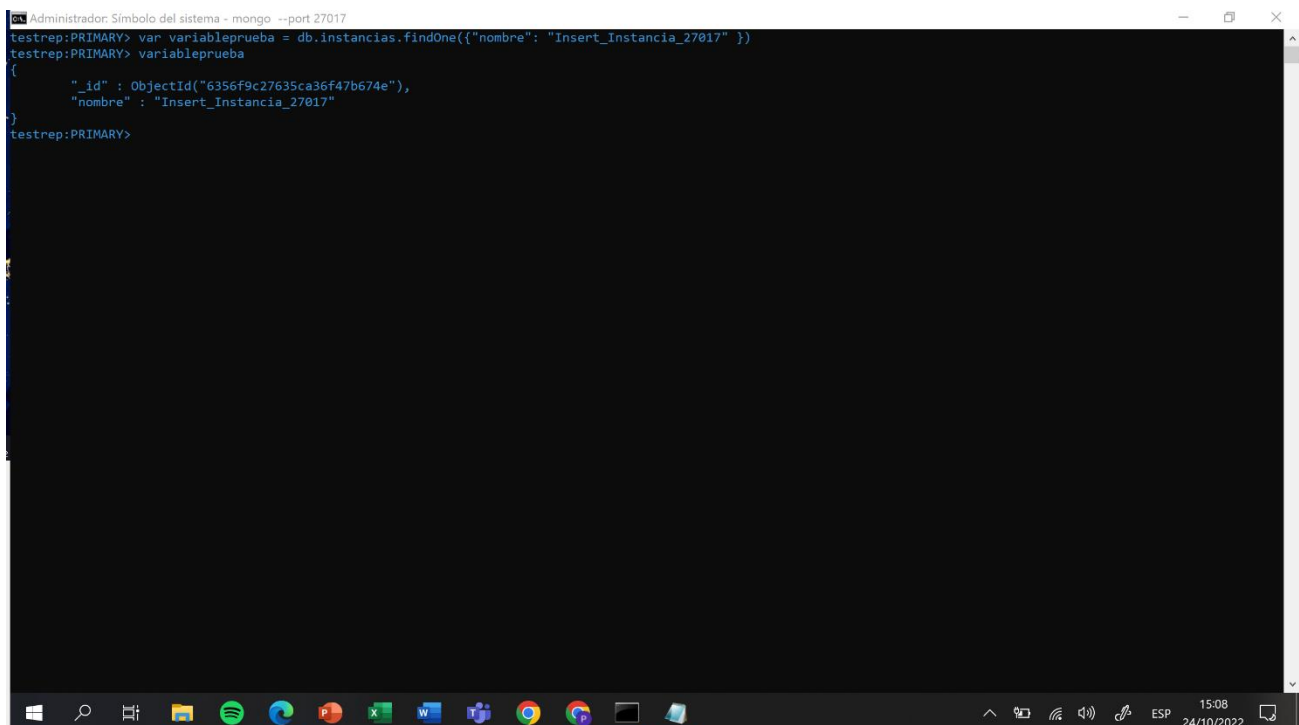
Ahora aplicamos el comando “db.instancias.find()” para demostrar que el insert hecho en la principal esta reflejado en las otras dos instancias



```
testrep:PRIMARY> show dbs
admin            0.000GB
config           0.000GB
local            0.000GB
replicaLaboratorio 0.000GB
testrep:PRIMARY> use replicaLaboratorio
switched to db replicaLaboratorio
testrep:PRIMARY> db.instancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_Instance_27017" }
testrep:PRIMARY>
```

```
testrep:SECONDARY> show dbs
admin            0.000GB
config           0.000GB
local            0.000GB
replicaLaboratorio 0.000GB
testrep:SECONDARY> use replicaLaboratorio
switched to db replicaLaboratorio
testrep:SECONDARY> db.instancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_Instance_27017" }
testrep:SECONDARY>
```

Para editar debemos crear una variable con el dato que deseamos modificar



```
Administrador: Símbolo del sistema - mongo --port 27017
testrep:PRIMARY> var variableprueba = db.instancias.findOne({"nombre": "Insert_Instance_27017" })
testrep:PRIMARY> variableprueba
{
  "_id" : ObjectId("6356f9c27635ca36f47b674e"),
  "nombre" : "Insert_Instance_27017"
}
testrep:PRIMARY>
```


Ahora con el nombre que le asignamos a la variable accedemos al atributo que deseamos cambiar, posterior colocamos de nuevo el nombre de la variable para asegurarnos que el modificar se hizo correctamente, de ser así procedemos a guardar lo hecho y realizamos un “find()” nuevamente para mostrar los datos ya cambiados y guardados

```
Administrador: Símbolo del sistema - mongo --port 27017
testrep:PRIMARY> var variableprueba = dbinstancias.findOne({"nombre": "Insert_Instance_27017" })
testrep:PRIMARY> variableprueba
{
  "_id" : ObjectId("6356f9c27635ca36f47b674e"),
  "nombre" : "Insert_Instance_27017"
}
testrep:PRIMARY> variableprueba.nombre = "Insert_MODIFICADO"
Insert_MODIFICADO
testrep:PRIMARY> variableprueba
{
  "_id" : ObjectId("6356f9c27635ca36f47b674e"),
  "nombre" : "Insert_MODIFICADO"
}
testrep:PRIMARY> dbinstancias.save(variableprueba)
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
testrep:PRIMARY> dbinstancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_MODIFICADO" }
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instance_27017_2" }
testrep:PRIMARY>
```

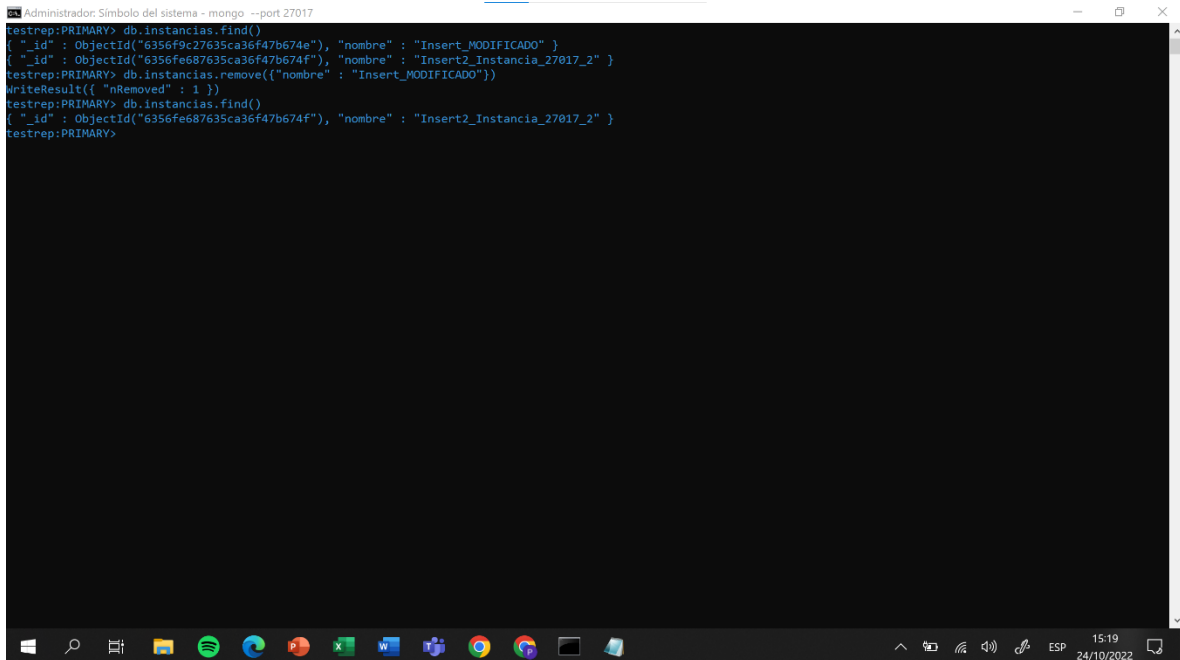
Nuevamente confirmamos que en las instancias secundarias del reflejo se aplicaron los cambios del editar

```
Administrador: Símbolo del sistema - mongo --port 27017
testrep:PRIMARY> var variableprueba = dbinstancias.findOne({"nombre": "Insert_Instance_27017" })
testrep:PRIMARY> variableprueba
{
  "_id" : ObjectId("6356f9c27635ca36f47b674e"),
  "nombre" : "Insert_Instance_27017"
}
testrep:PRIMARY> variableprueba.nombre = "Insert_MODIFICADO"
Insert_MODIFICADO
testrep:PRIMARY> variableprueba
{
  "_id" : ObjectId("6356f9c27635ca36f47b674e"),
  "nombre" : "Insert_MODIFICADO"
}
testrep:PRIMARY> dbinstancias.save(variableprueba)
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
testrep:PRIMARY> dbinstancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_MODIFICADO" }
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instance_27017_2" }
testrep:PRIMARY>
```

```
Administrador: Símbolo del sistema - mongo --port 27020
testrep:SECONDARY> dbinstancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_MODIFICADO" }
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instance_27017_2" }
testrep:SECONDARY>
```

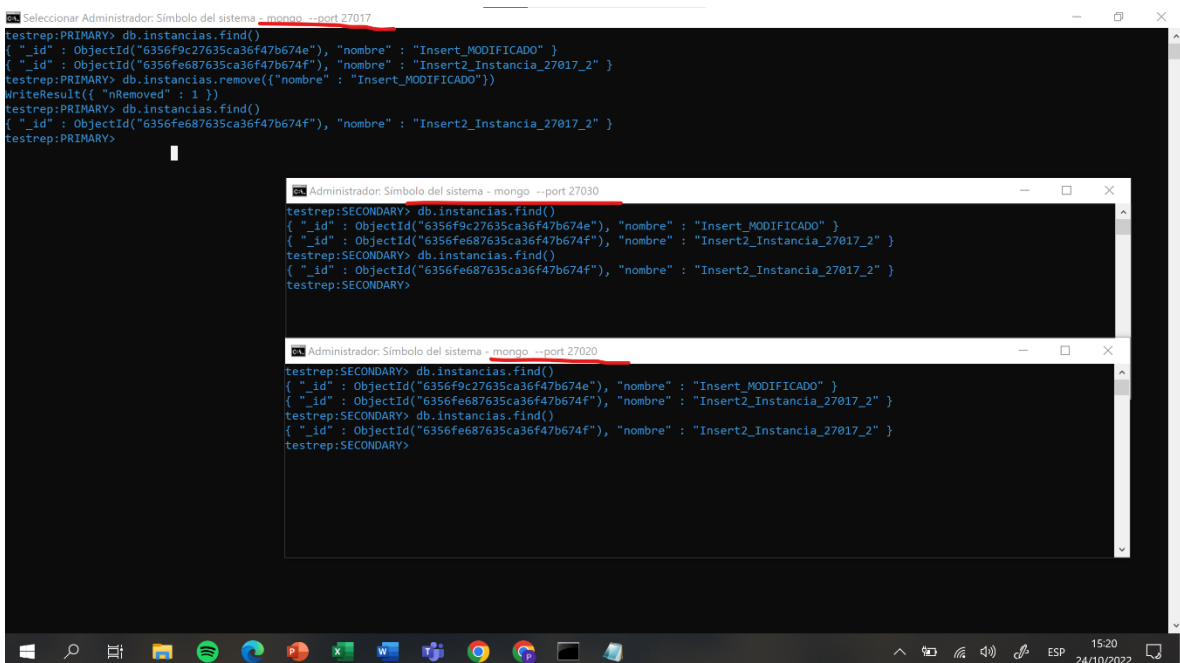
```
Administrador: Símbolo del sistema - mongo --port 27030
testrep:SECONDARY> dbinstancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_MODIFICADO" }
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instance_27017_2" }
testrep:SECONDARY>
```

Para eliminar aplicamos el comando de la imagen en la terminal de la instancia principal y de nuevo un “find()” para mostrar que se elimino correctamente



```
Administrador: Símbolo del sistema - mongo --port 27017
testrep:PRIMARY> db.instancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_MODIFICADO" }
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instancia_27017_2" }
testrep:PRIMARY> db.instancias.remove({"nombre" : "Insert_MODIFICADO"})
WriteResult({"nRemoved" : 1 })
testrep:PRIMARY> db.instancias.find()
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instancia_27017_2" }
testrep:PRIMARY>
```

Por ultimo aplicamos un “find()” en la terminal de las instancias secundarias y con esto confirmamos efectivamente que la replicación en mongo fue todo un éxito



```
Selección Administrador: Símbolo del sistema - mongo --port 27017
testrep:PRIMARY> db.instancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_MODIFICADO" }
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instancia_27017_2" }
testrep:PRIMARY> db.instancias.remove({"nombre" : "Insert_MODIFICADO"})
WriteResult({"nRemoved" : 1 })
testrep:PRIMARY> db.instancias.find()
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instancia_27017_2" }
testrep:PRIMARY>
```

```
Administrador: Símbolo del sistema - mongo --port 27030
testrep:SECONDARY> db.instancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_MODIFICADO" }
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instancia_27017_2" }
testrep:SECONDARY> db.instancias.find()
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instancia_27017_2" }
testrep:SECONDARY>
```

```
Administrador: Símbolo del sistema - mongo --port 27020
testrep:SECONDARY> db.instancias.find()
{ "_id" : ObjectId("6356f9c27635ca36f47b674e"), "nombre" : "Insert_MODIFICADO" }
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instancia_27017_2" }
testrep:SECONDARY> db.instancias.find()
{ "_id" : ObjectId("6356fe687635ca36f47b674f"), "nombre" : "Insert2_Instancia_27017_2" }
testrep:SECONDARY>
```