



# Fundamentos de MongoDB Enterprise 4.4.1

Instructor: Carlos Carreño  
Email: [ccarrenovi@gmail.com](mailto:ccarrenovi@gmail.com)



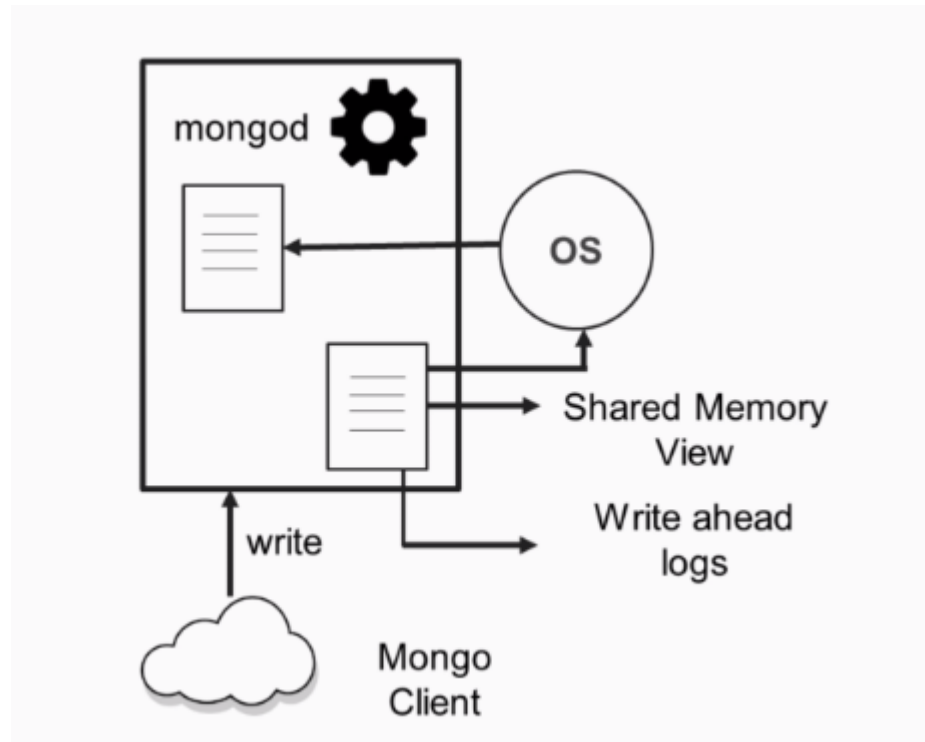
mongoDB

## Unidad 8 Gestión del Control de Acceso, Backup y Restore

Fundamentos de MongoDB Enterprise 4.4.0

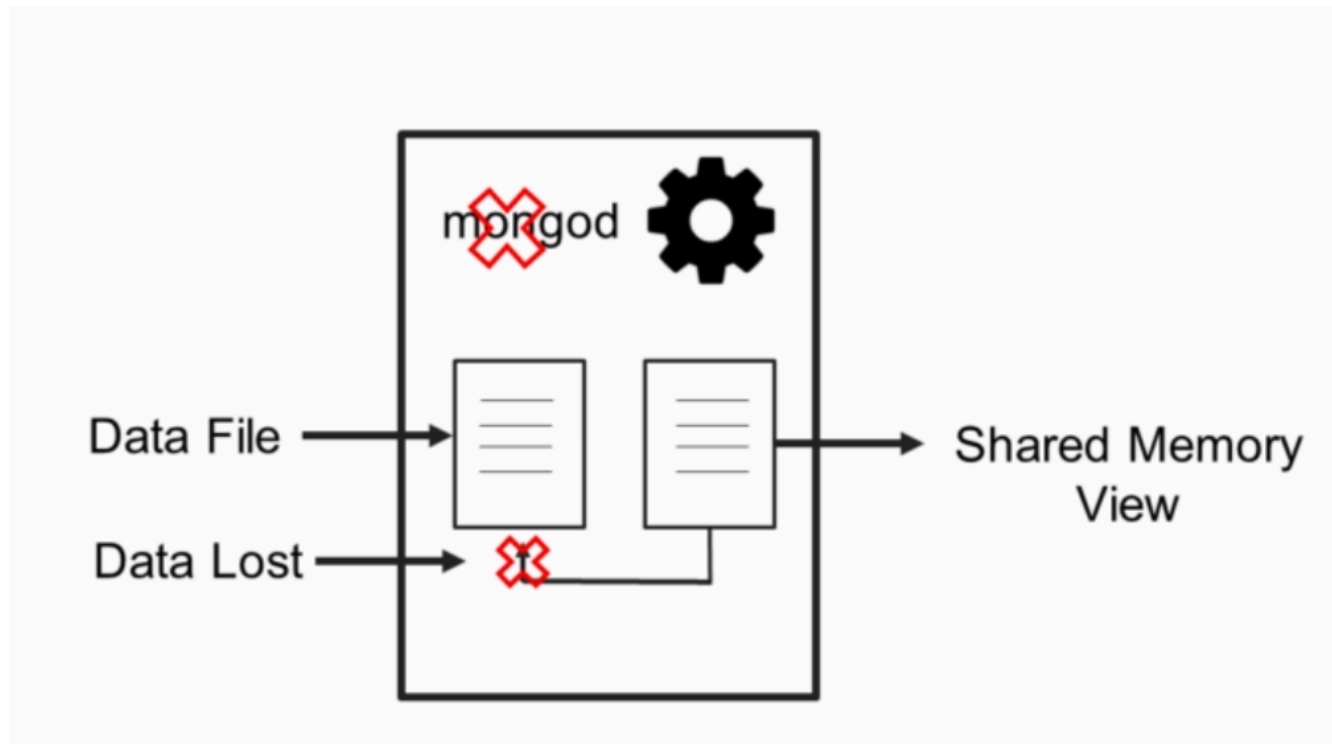
# Working with out Journaling

- Arquitectura por defecto de mongoDB



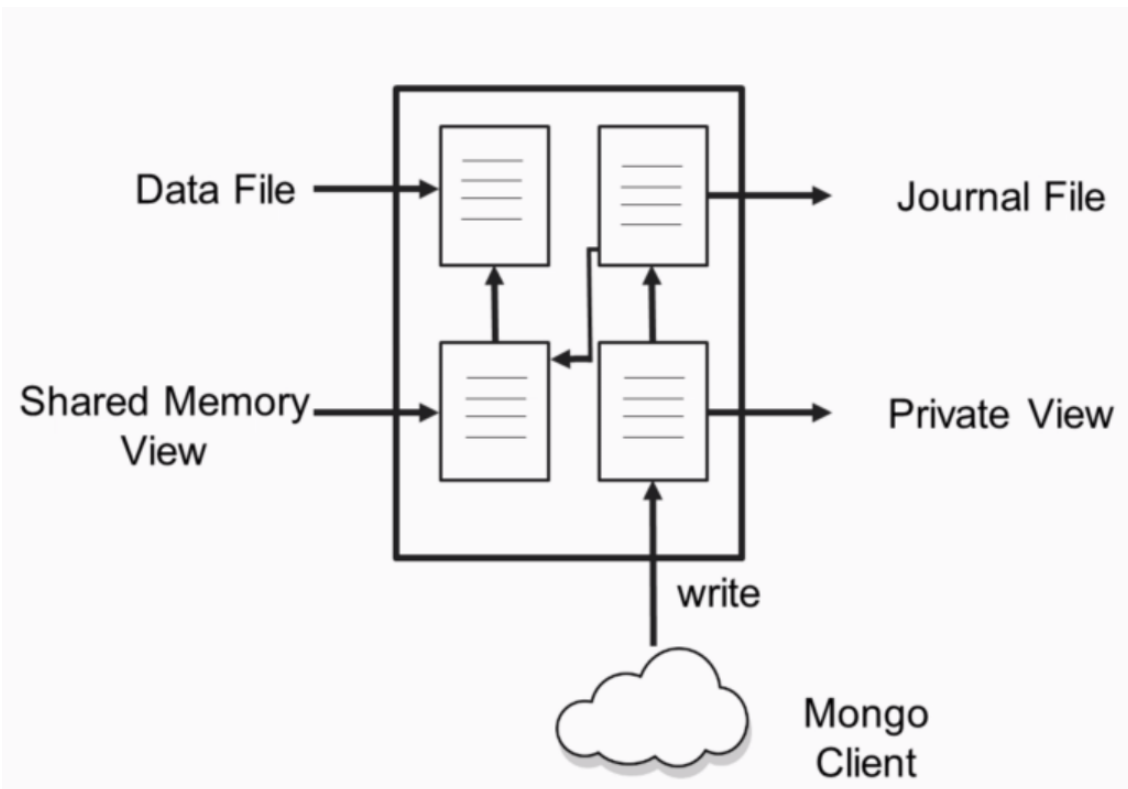
# Working with out Journaling

- En una caída del servidor se podría perder datos



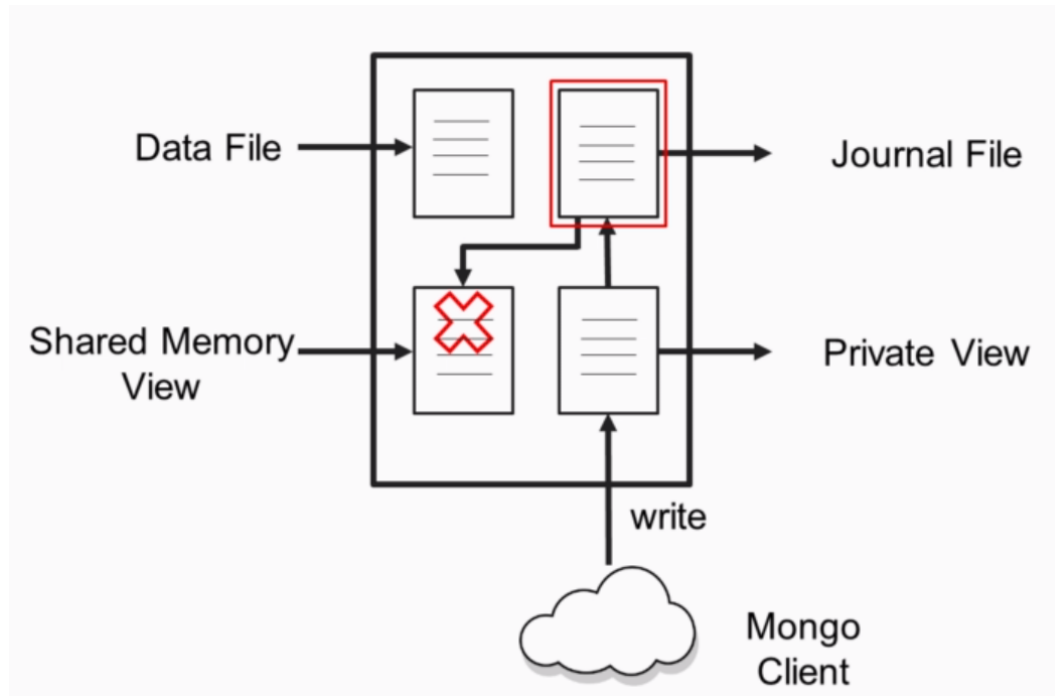
# Working with Journaling

- En este caso se crean journal files que contienen una copia del log, utilizando un private view



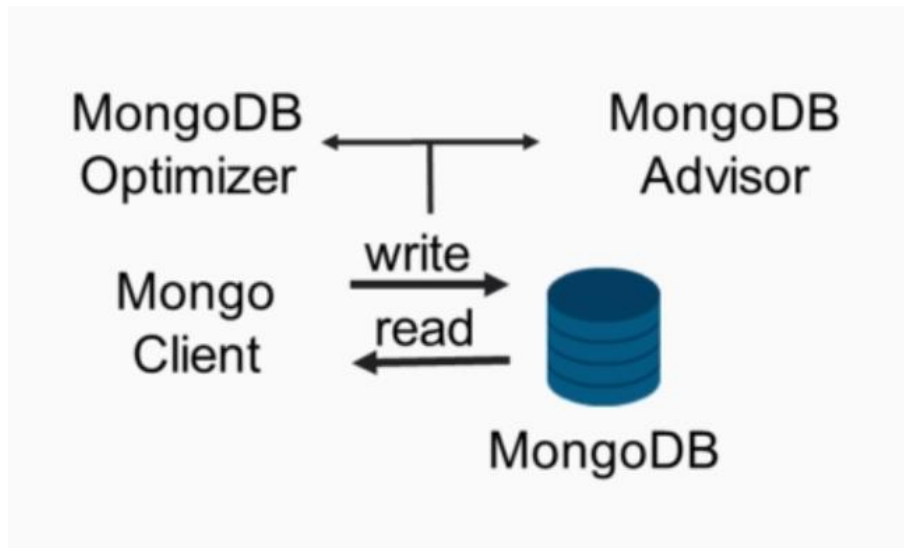
# Working with Journaling

- En caso de una caída el journal file contiene una copia y puede actualizar el shared memory view



# Discovering Importance of Profiler

- Por defecto MongoDB no muestra información adicional acerca de una consulta a los datos



# Profiles and Level - RESUME

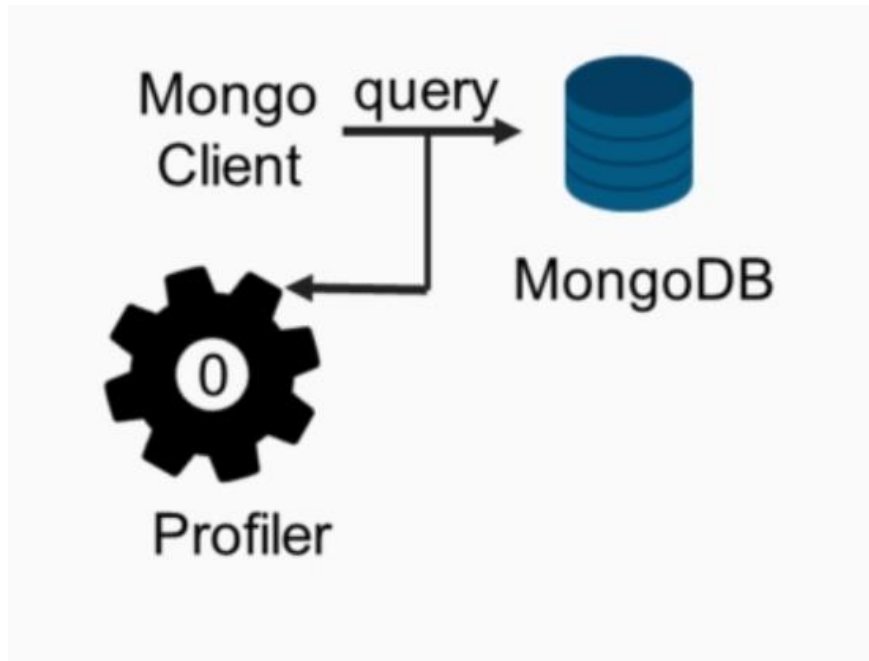
- Estos son los niveles del profile en MongoDB

Level	Description
0	The profiler is off and does not collect any data. This is the default profiler level.
1	The profiler collects data for operations that take longer than the value of <code>slowms</code> .
2	The profiler collects data for all operations.



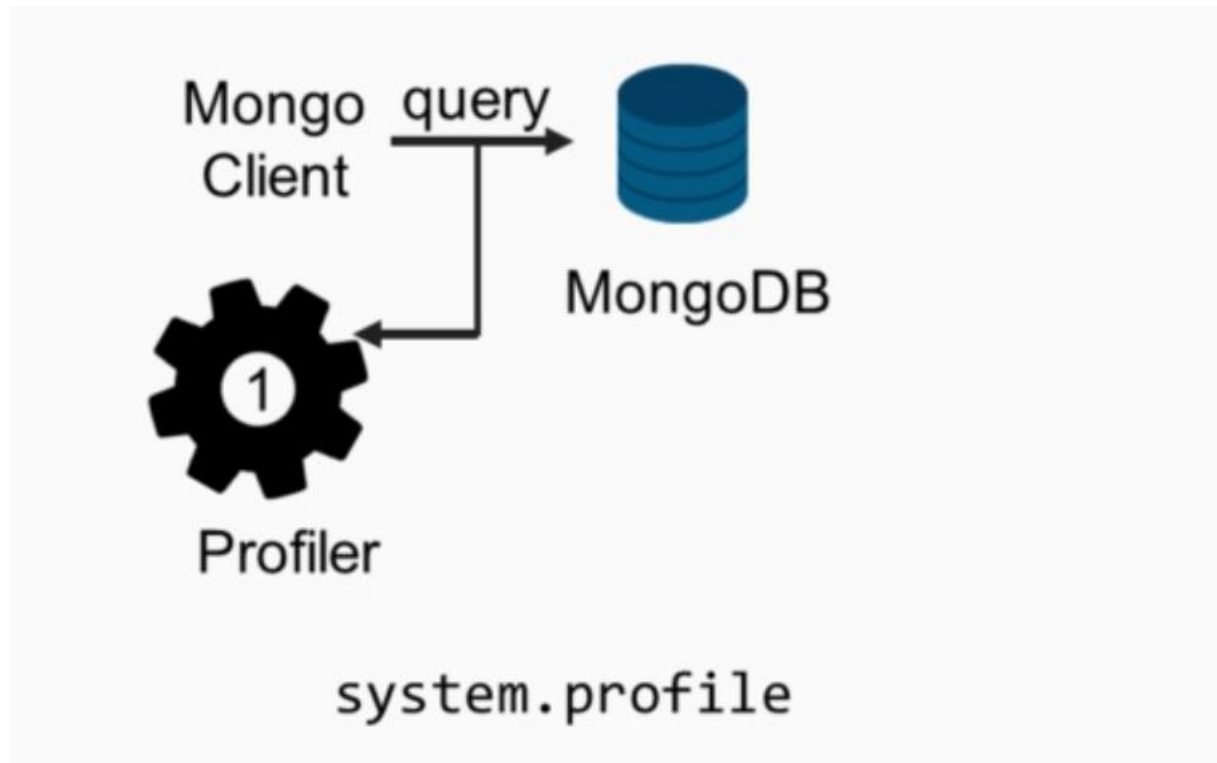
# Profiles and Levels – Level CERO

- El profiler en el nivel CERO no obtiene datos adicionales de la consulta



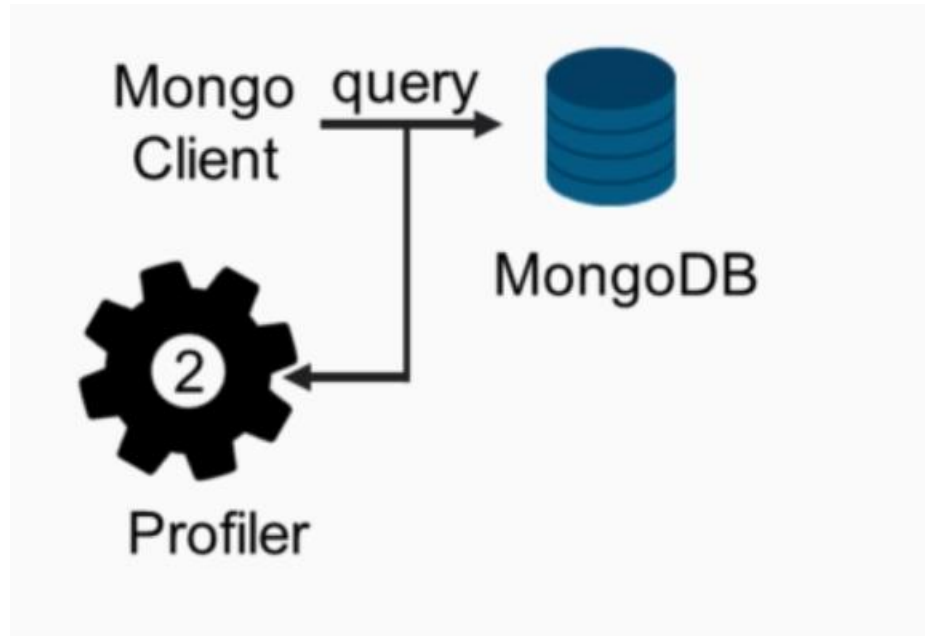
# Profiles and Levels – Level ONE

- En el nivel UNO el profile trae datos adicionales de la consulta dentro de un limite dado, estos se almacenan en la colección **system.profile**



# Profiles and Levels – Level TWO

- En el nivel DOS el profiler obtiene datos de la performance de la consulta ejecutándose en ambiente de cluster, estos datos se guardan en la colección system.profile



# Database Profiler

- Activando el profiler, por defecto slows tiene el valor de 100 millis  
`db.setProfilingLevel(2)`
- Definiendo un valor de slows  
`db.setProfilingLevel(1, { slowms: 20 })`
- Conociendo el estado y nivel del profiler  
`db.getProfilingStatus()`  
`db.getProfilingLevel()`
- Deshabilitando el profiler  
• `db.setProfilingLevel(0)`

# Query Profiler

- Consultando el profiler

```
db.system.profile.find().limit(10).sort( { ts : -1 } ).pretty()
```

```
db.system.profile.find( { ns : 'mydb.test' } ).pretty()
```

```
db.system.profile.find( { millis : { $gt : 5 } } ).pretty()
```

# Profiler Size

- Por defecto el profiler tiene un tamaño de 1 M, puedes cambiar el tamaño a 4M por ejemplo, con el siguiente procedimiento:

```
db.setProfilingLevel(0)
```

```
db.system.profile.drop()
```

```
db.createCollection( "system.profile", { capped: true, size:4000000 } )
```

```
db.setProfilingLevel(1)
```

# Practicing MongoDB Backup and Restore Techniques

- **mongodump** es una utilidad para crear una exportación binaria de los contenidos de una base de datos.
- **mongodump** puede exportar datos de instancias mongod o mongos; es decir, puede exportar datos desde implementaciones de sharding clúster y standalone servers.

# Mongodump exporting data

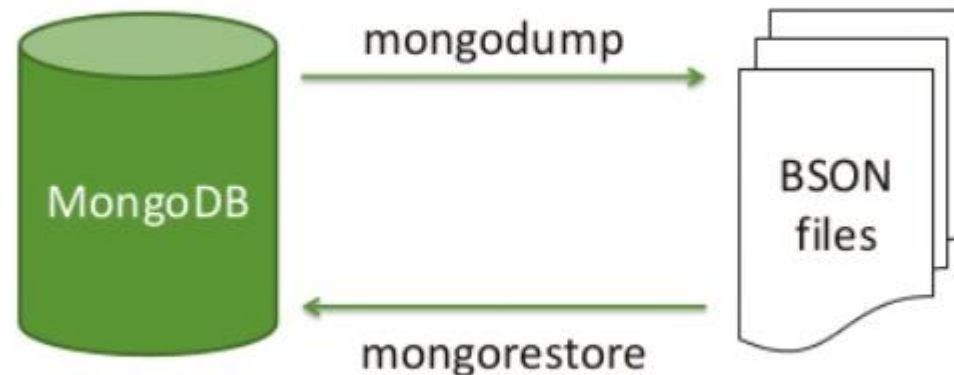
- Realizando un backup con mongodump

```
[admin@odiseo ~]$ mongodump --host "odiseo.example.com" --port 27020 --out backup
2019-11-03T08:15:18.842-0500   writing admin.system.version to
2019-11-03T08:15:18.843-0500   done dumping admin.system.version (1 document)
2019-11-03T08:15:18.843-0500   writing devdb.cars to
2019-11-03T08:15:18.843-0500   writing proddb.employees to
2019-11-03T08:15:18.843-0500   writing test.cars to
2019-11-03T08:15:18.845-0500   done dumping devdb.cars (2 documents)
2019-11-03T08:15:18.846-0500   done dumping test.cars (1 document)
2019-11-03T08:15:18.846-0500   done dumping proddb.employees (2 documents)
[admin@odiseo ~]$ ls -l backup/
total 0
drwxrwxr-x 2 admin admin 69 Nov  3 08:15 admin
drwxrwxr-x 2 admin admin 49 Nov  3 08:15 devdb
drwxrwxr-x 2 admin admin 59 Nov  3 08:15 proddb
drwxrwxr-x 2 admin admin 49 Nov  3 08:15 test
[admin@odiseo ~]$
```



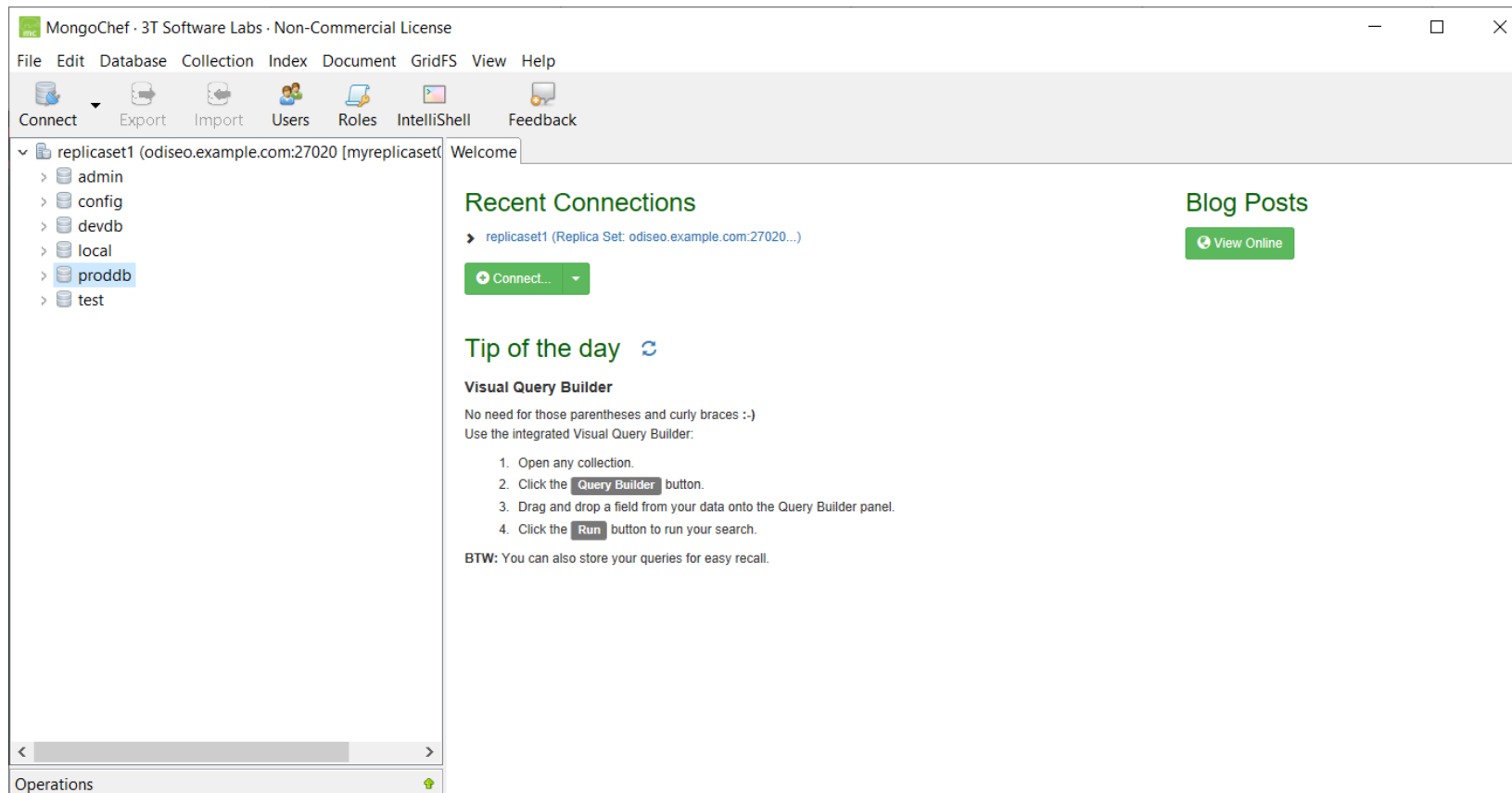
# Practicing MongoDB Backup and Restore Techniques

- El programa **mongorestore** carga datos de un volcado de base de datos binario creado por mongodump o la entrada estándar (a partir de la versión 3.0.0) en una instancia de mongod o mongos.



# Drop database

- Elimina la base de datos “proddb”



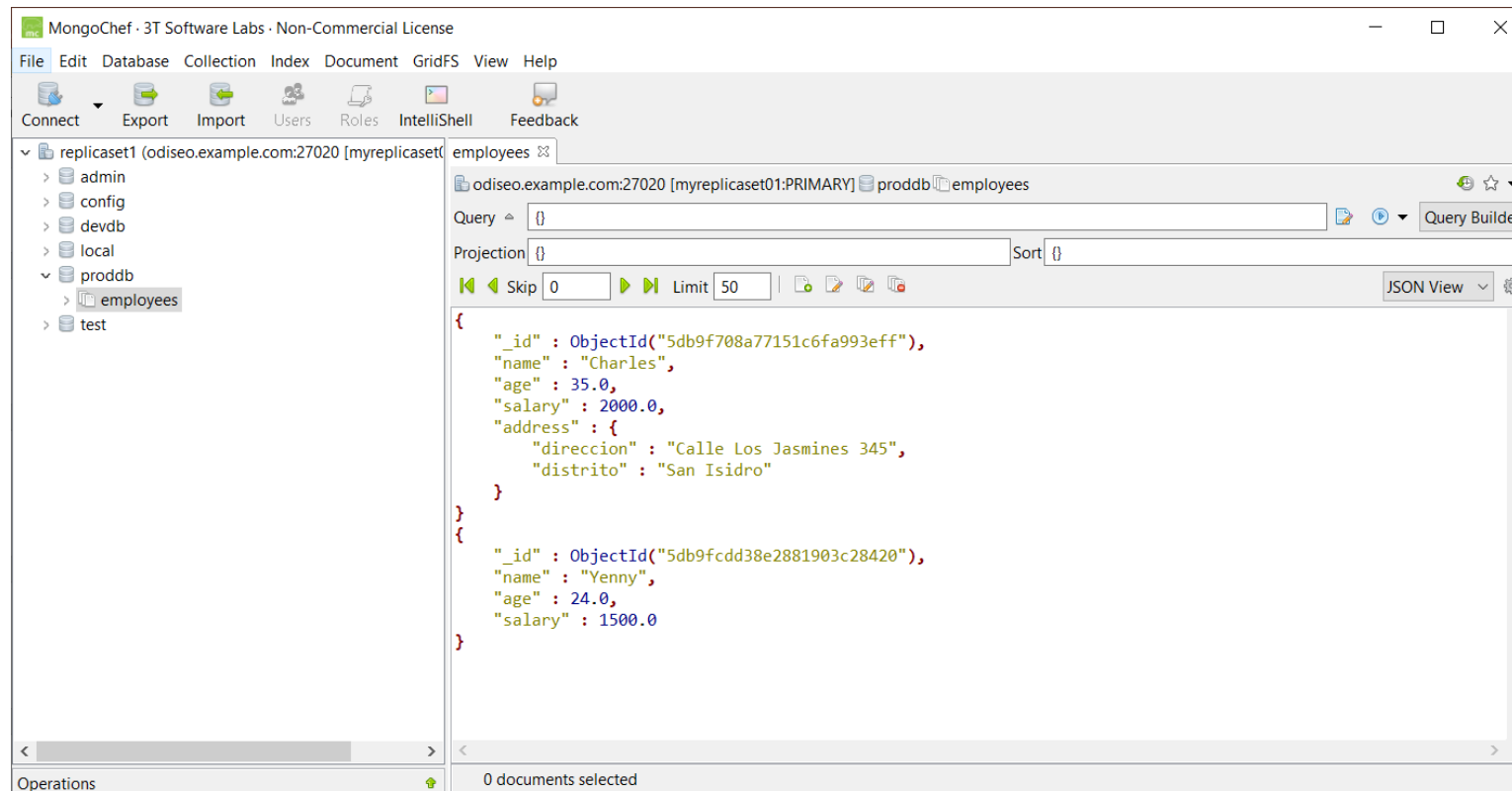
# Restore Database

- Restaura la base de datos “proddb” con mongorestore

```
[admin@odiseo ~]$ mongorestore --host "odiseo.example.com" --port 27020 backup
2019-11-03T08:24:48.492-0500 preparing collections to restore from
2019-11-03T08:24:48.493-0500 restoring to existing collection proddb.employees without dropping
2019-11-03T08:24:48.493-0500 reading metadata for proddb.employees from backup/proddb/employees.metadata.json
2019-11-03T08:24:48.493-0500 restoring proddb.employees from backup/proddb/employees.bson
2019-11-03T08:24:48.496-0500 restoring to existing collection devdb.cars without dropping
2019-11-03T08:24:48.496-0500 reading metadata for devdb.cars from backup/devdb/cars.metadata.json
2019-11-03T08:24:48.497-0500 restoring devdb.cars from backup/devdb/cars.bson
2019-11-03T08:24:48.499-0500 restoring to existing collection test.cars without dropping
2019-11-03T08:24:48.499-0500 reading metadata for test.cars from backup/test/cars.metadata.json
2019-11-03T08:24:48.499-0500 restoring test.cars from backup/test/cars.bson
2019-11-03T08:24:48.500-0500 continuing through error: E11000 duplicate key error collection: proddb.employees index:
2019-11-03T08:24:48.500-0500 continuing through error: E11000 duplicate key error collection: proddb.employees index:
2019-11-03T08:24:48.500-0500 no indexes to restore
2019-11-03T08:24:48.500-0500 finished restoring proddb.employees (0 documents, 2 failures)
2019-11-03T08:24:48.503-0500 continuing through error: E11000 duplicate key error collection: devdb.cars index: _id_
2019-11-03T08:24:48.503-0500 continuing through error: E11000 duplicate key error collection: devdb.cars index: _id_
2019-11-03T08:24:48.503-0500 no indexes to restore
2019-11-03T08:24:48.503-0500 finished restoring devdb.cars (0 documents, 2 failures)
2019-11-03T08:24:48.503-0500 continuing through error: E11000 duplicate key error collection: test.cars index: _id_ c
2019-11-03T08:24:48.503-0500 no indexes to restore
2019-11-03T08:24:48.503-0500 finished restoring test.cars (0 documents, 1 failure)
2019-11-03T08:24:48.503-0500 0 document(s) restored successfully. 5 document(s) failed to restore.
[admin@odiseo ~]$
```

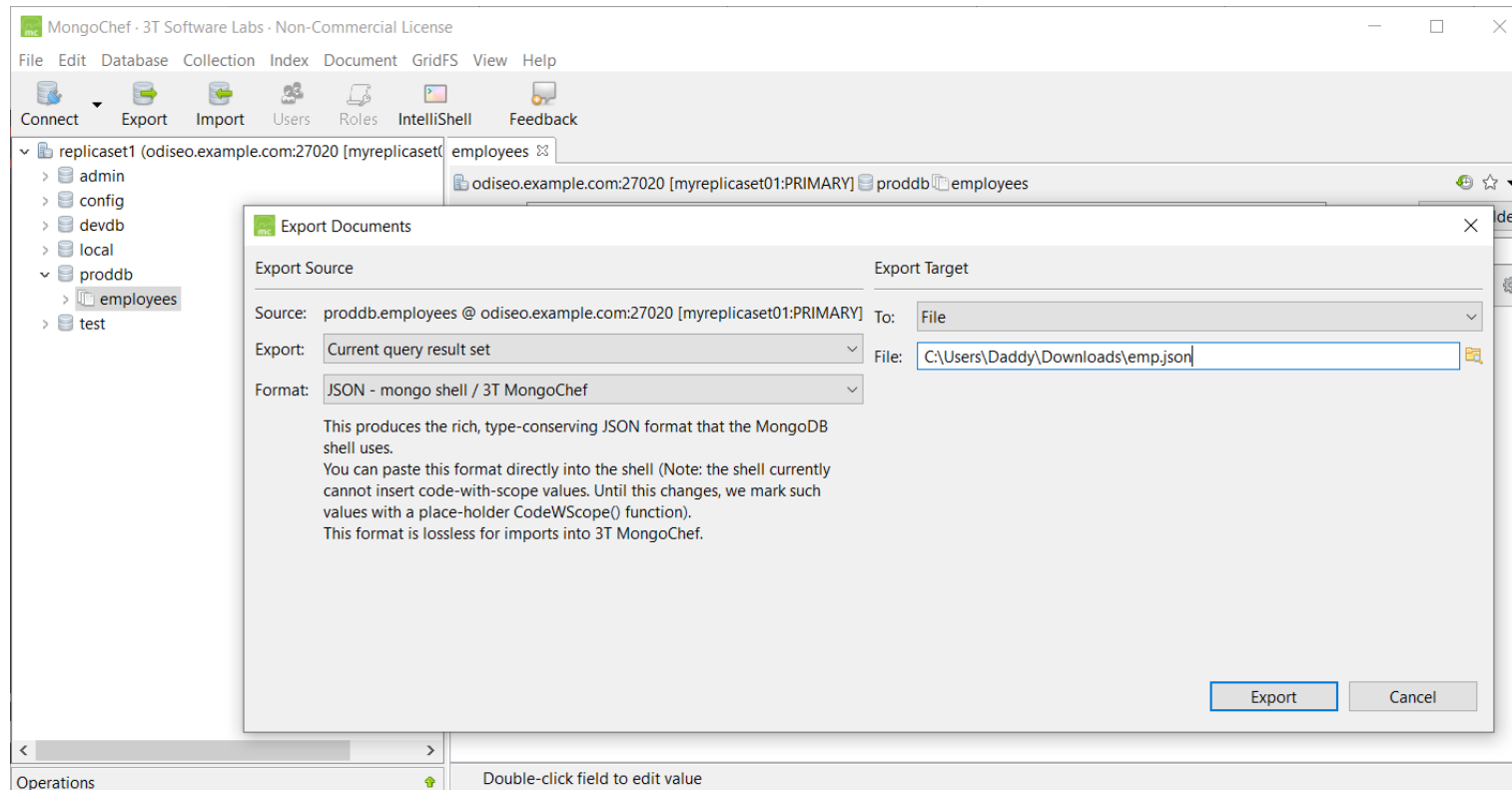
# Check Restore Database

- Verifica que la base de datos “proddb”, se haya recuperado



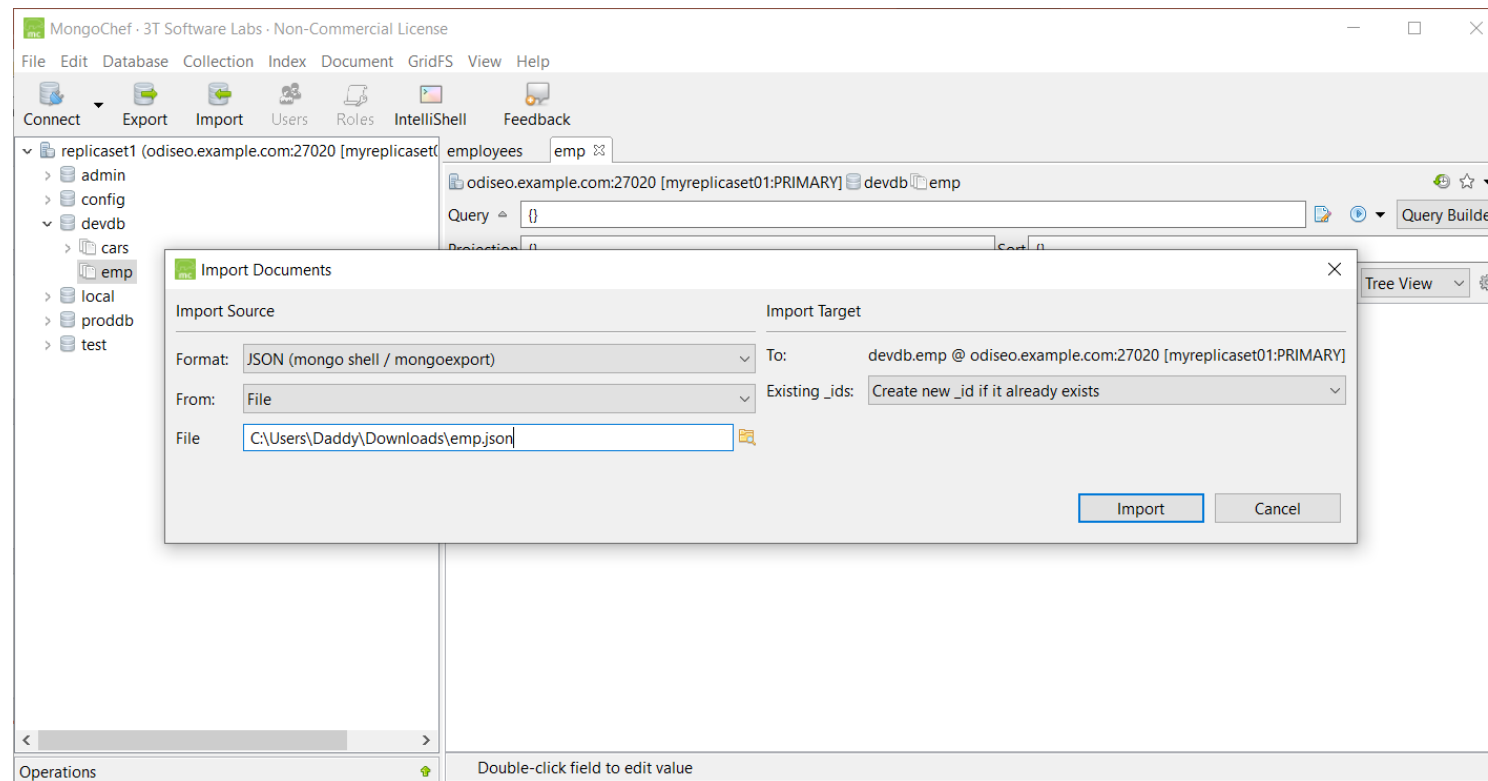
# Exporting Data

- Usando Mongochef puedes exportar colecciones de documentos



# Import Data

- Crea la colección e importa datos usando Mongochef



# Enable Access Control

- Conéctate a una instancia sin autenticación
- Crea el usuario administrador

```
use admin
db.createUser(
{
  user: "myUserAdmin",
  pwd: passwordPrompt(), // o texto claro
  roles: [ { role: "userAdminAnyDatabase", db: "admin" },
"readWriteAnyDatabase" ]
}
)
```

# Enable Access Control

- Detén la instancia

```
db.adminCommand( { shutdown: 1 } )
```

- Inicia la instancia con autenticación

```
mongod --auth --port 27017 --dbpath /var/lib/mongoddb
```

- Conéctate y authenticate con el usuario

```
mongo --port 27017 --authenticationDatabase "admin" -u  
"myUserAdmin" -p
```



# Authenticate User with admin database

- Conectate a la instancia

```
$mongo -port 27017
```

- Cambia a la base de datos admin y authenticate

```
use admin
```

```
db.auth("myUserAdmin","<password>");
```

# Enable Access Control with Configuration File

- Edita el archivo de configuración de la instancia `/etc/mongod.conf`

```
root@odiseo:~  
storage:  
  dbPath: /var/lib/mongo  
  journal:  
    enabled: true  
  # engine:  
  # wiredTiger:  
  
# how the process runs  
processManagement:  
  fork: true # fork and run in background  
  pidFilePath: /var/run/mongodb/mongod.pid # location of pidfile  
  timeZoneInfo: /usr/share/zoneinfo  
  
# network interfaces  
net:  
  port: 27017  
  bindIp: 0.0.0.0 # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternative  
ly, use the net.bindIpAll setting.  
  
security:  
  authorization: enabled  
#operationProfiling:  
  
#replication:  
  
#sharding:  
  
## Enterprise-Only Options  
  
#auditLog:  
  
#snmp:
```

# Exploring User Roles and Authorization

- `db.createUser(..)`
- `db.changeUserPassword(..)`
- `db.auth(..)`
- `db.dropUser(..)`
- `db.dropAllUsers`

# Built-In Roles

- Database user roles
  - read
  - readWrite
- Database Administration Roles
  - dbAdmin
  - dbOwner
  - userAdmin
- Cluster Administrator Roles
  - clusterAdmin
  - clusterManager
  - clusterMonitor
  - hostManager

# Built-In Roles

- Backup and restore roles
  - backup
  - restore
- All Database Role
  - readAnyDatabase
  - readWriteAnyDatabase
  - userAdminAnyDatabase
  - dbAdminAnyDatabase
- Super User Roles
  - root

# Creating Additional Users

```
use test
```

```
db.createUser(  
  {  
    user: "myTester",  
    pwd: passwordPrompt(), // o texto claro  
    roles: [ { role: "readWrite", db: "test" },  
             { role: "read", db: "reporting" } ]  
  }  
)
```

# Testing User

- Conectate con el nuevo usuario

```
$mongo --port 27017 -u "myTester" --authenticationDatabase "test" -p
```

- Inserta datos

```
db.foo.insert( { x: 1, y: 1 } )
```

# Practica

- Practica 8 Gestión del Control de Acceso, Backup y Restore



# Referencias

- <https://docs.mongodb.com/manual/core/journaling/>
- <https://docs.mongodb.com/manual/tutorial/manage-the-database-profiler/>
- <https://docs.mongodb.com/manual/reference/program/mongodump/>
- <https://docs.mongodb.com/manual/reference/program/mongorestore/>
- <https://www.php.net/manual/es/mongo.writeconcerns.php>
- <https://antoniofernandez.com/seguridad-mongodb-produccion/>
- <https://parzibyte.me/blog/2018/12/11/autenticacion-administrador-mongodb/>
- <https://www.guru99.com/top-20-mongodb-tools.html>
- <https://askubuntu.com/questions/420981/how-do-i-save-terminal-output-to-a-file>