Taller de MongoDB

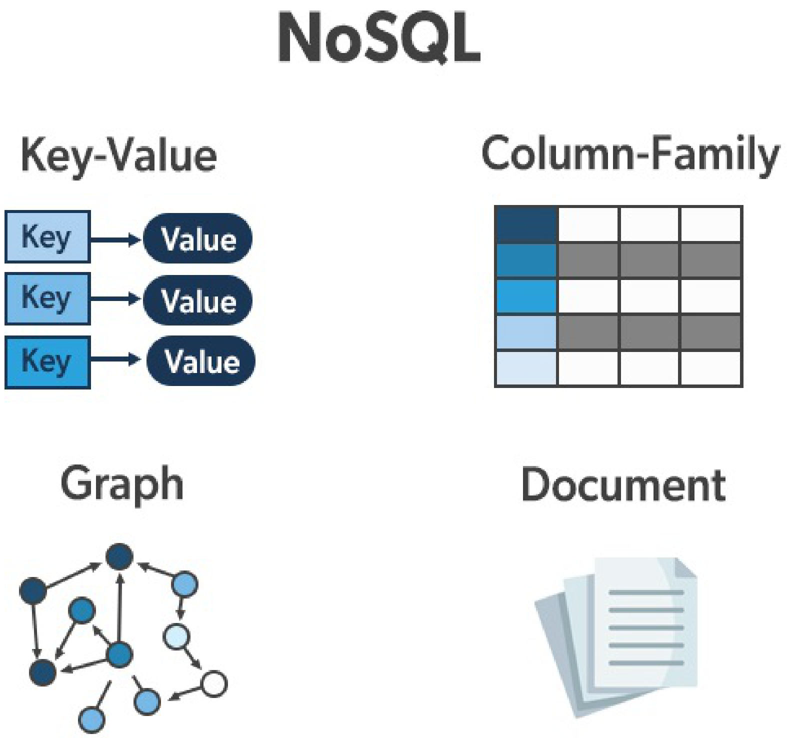
Sesión 1

Nixvoid

Contenido

- Introducción
- Configuración ATLAS y Compass
- JSON/BSON
- Operaciones CRUD
- Tipos de data
- Relaciones y Modelado
- Validación de esquema
- Introducción al aggregation framework
- Siguientes pasos

Introducción



Introducción

Configuración de Atlas Cluster

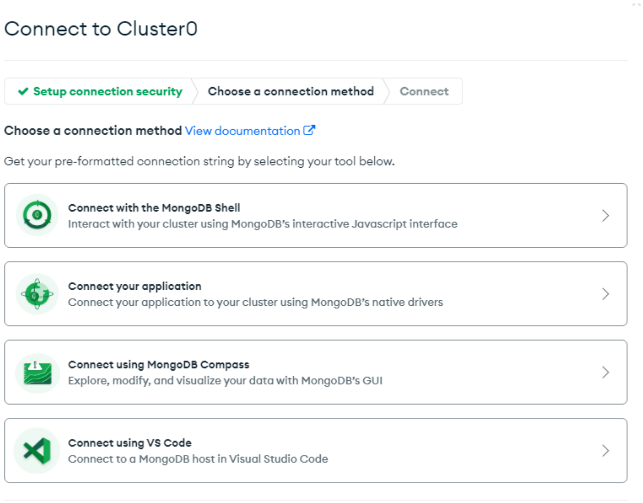
Registro

- Atlas Sign-in

Conexiones a la base de datos de MongoDB

```
$ mongosh "mongodb+srv://cluster0.lp9dt.mongodb.net/myFirstDatabase" --apiVersion 1 --username <username>
```

```
mongodb+srv://<username>:<password>@cluster0.lp9dt.mongodb.net/test
```



Connect to Cluster0

Choose a connection method View documentation

Get your pre-formatted connection string by selecting your tool below.

Connect with the MongoDB Shell
Interact with your cluster using MongoDB's interactive Javascript interface

Connect your application
Connect your application to your cluster using MongoDB's native drivers

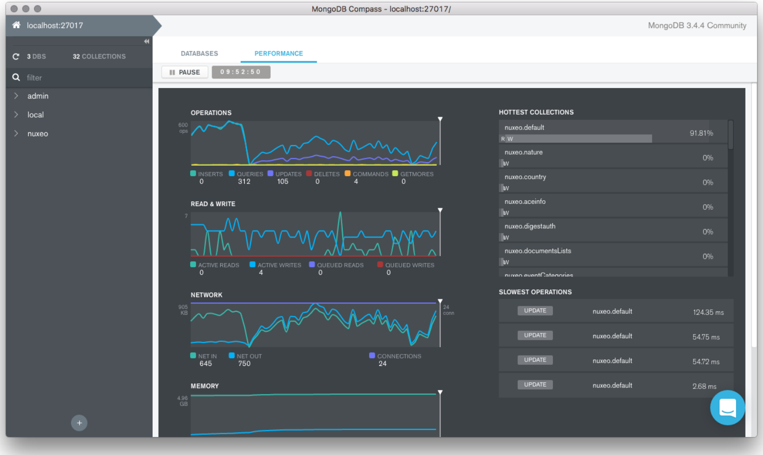
Connect using MongoDB Compass
Explore, modify, and visualize your data with MongoDB's GUI

Connect using VS Code
Connect to a MongoDB host in Visual Studio Code

Configuración de MongoDB Compass

Descarga

- MongoDB Compass



HOTTEST COLLECTIONS

nuxeo.default R W	91.81%
nuxeo.nature W	0%
nuxeo.country W	0%
nuxeo.aceinfo W	0%
nuxeo.digestauth W	0%
nuxeo.documentsLists W	0%
nuxeo.eventCategories	

SLOWEST OPERATIONS

UPDATE	nuxeo.default	124.35 ms
UPDATE	nuxeo.default	54.75 ms
UPDATE	nuxeo.default	54.72 ms
UPDATE	nuxeo.default	2.68 ms

JSON

```
{
  "_id": 1,
  "name" : { "first" : "John", "last" : "Backus" },
  "contribs" : [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],
  "awards" : [
    {
      "award" : "W.W. McDowell Award",
      "year" : 1967,
      "by" : "IEEE Computer Society"
    }, {
      "award" : "Draper Prize",
      "year" : 1993,
      "by" : "National Academy of Engineering"
    }
  ]
}
```

BSON

```
{"hello": "world"} →
\x16\x00\x00\x00            // total document size
\x02                        // 0x02 = type String
hello\x00                   // field name
\x06\x00\x00\x00world\x00  // field value
\x00                        // 0x00 = type EOO ('end of object')
```

JSON VS BSON

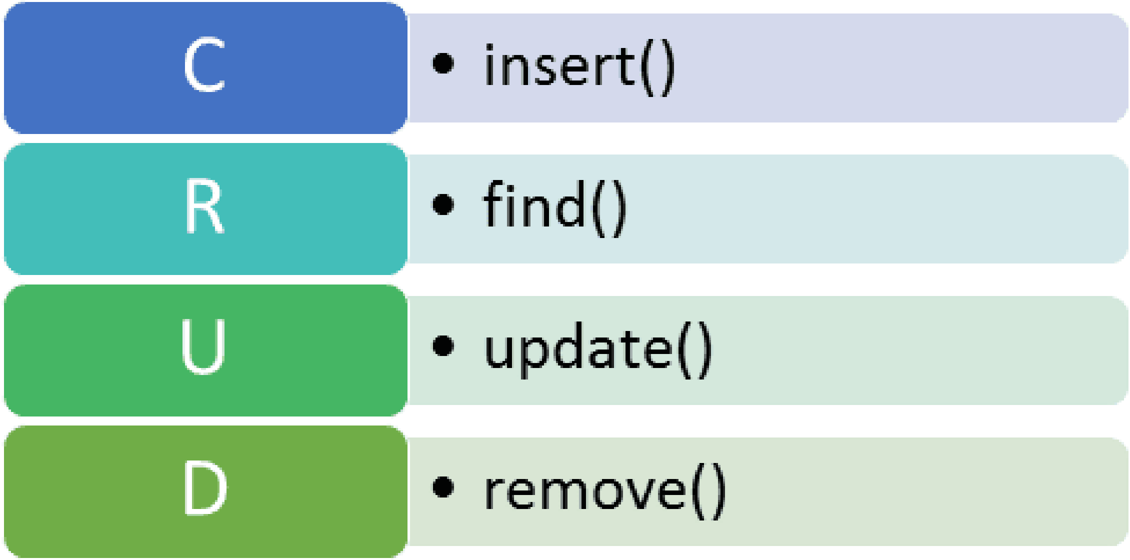
JSON

- **Encoding:** UTF-8 String
- **Data:** String, Boolean, Number, Array
- **Human readable**

BSON

- **Encoding:** Binary
- **Data:** String, Boolean, Number (Integer, Long ,…), Array, Date, Raw binary
- **Machine readable**

Operaciones CRUD



C

- insert()

R

- find()

U

- update()

D

- remove()

Insert

Operations

```
db.inventory.insertOne(
   { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } }
)
```

```
db.inventory.insertMany([
   { item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },
   { item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },
   { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }
])
```

Find

Operations

```
db.collection.find({})

SELECT * FROM inventory
```

```
db.inventory.find( { field: "value" } )

SELECT * FROM inventory WHERE field = "value"
```

```
db.inventory.find( { field: {"$in":["valor1","valor2"]} } )

SELECT * FROM inventory WHERE field in ("valor1","valor2")
```

- Query documents
- Query embedded documents
- Query Arrays
- Query an Array of embedded documents
- Projection