Introducción a MongoDB

Miguel Asencio

```
{
    name: 'Miguel Angel
            Asencio Hurtado',
    email: 'maasencioh@gmail.com',
    developer: true,
    gitHub: '@maasencioh',
    twitter: '@maasencioh',
    npm: '@maasencioh'
}
```
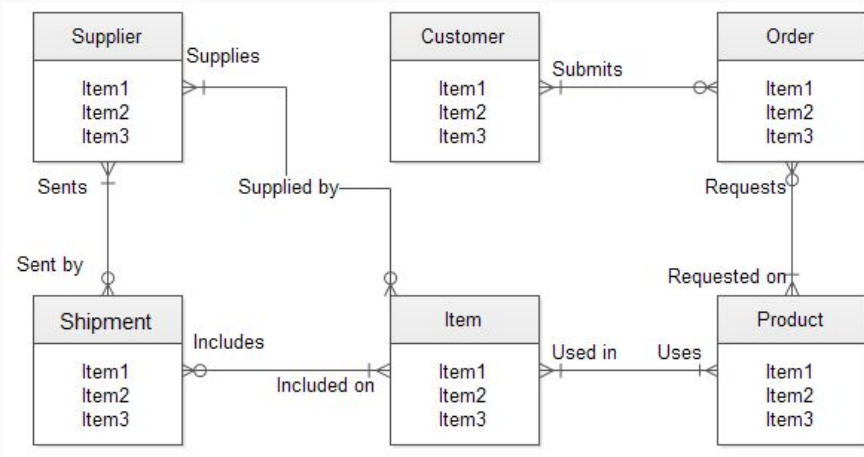
Organizador
BeerJS

Que es MongoDB?

- Sistema de gestión de base de datos NoSQL
- MongoDB Inc. (2007)

https://www.mongodb.com/

Escucharon bien, NoSQL!!!

```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021"
    },
    "phoneNumber": [
        { "type": "home", "number": "212 555-1234" },
        { "type": "fax", "number": "646 555-4567" }
    ]
}
```

Empecemos!

Instrucciones para instalación:

https://docs.mongodb.
org/manual/installation/

Recomendaciones:

- Máquinas de 64-bits
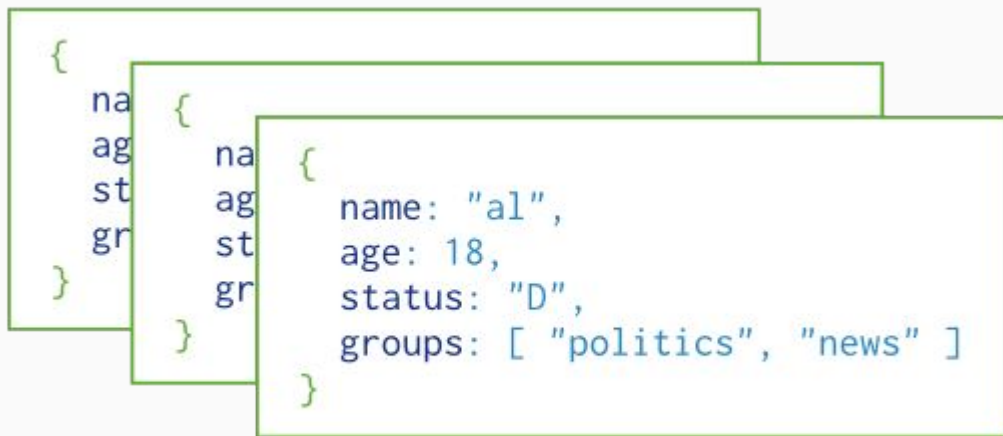- V3.2 (community)
- Agregar al PATH
- Correr como sudo

- **mongod:** (daemon) iniciar instancia, sea replica set o configuración sencilla
- **mongo:** interfaz de shell en JS ♡
- **mongos:** servicio de enrutamiento para shared clusters
- **mongodump:** hace una "copia" binaria de una base de datos y una colección
- **mongorestore:** contraria a mongodump, sirve como backup
- **mongoexport:** similar a mongodump, pero en vez de exportar a datos binarios puede usar JSON y CSV
- **mongoimport:** similar a mongorestore pero para mongoexport

CRUD

- **C**reate
- **R**ead
- **U**pdate
- **D**elete

```
{
  name: "sue",           ←——— field: value
  age: 26,               ←——— field: value
  status: "A",           ←——— field: value
  groups: [ "news", "sports" ]   ←——— field: value
}
```

```
{
  na
  ag
  st
  gr
}
    {
      na
      ag
      st
      gr
    }
        {
          name: "al",
          age: 18,
          status: "D",
          groups: [ "politics", "news" ]
        }
```

Collection

Crear - insert

- insertOne()
- insertMany()
- insert()

→ _id
→ JSON
→ BSON (16MB)

```
Collection                 Document
   ↓                          ↓
db.users.insert(
                  {
                    name: "sue",
                    age: 26,
                    status: "A",
                    groups: [ "news", "sports" ]
                  }
                )
```
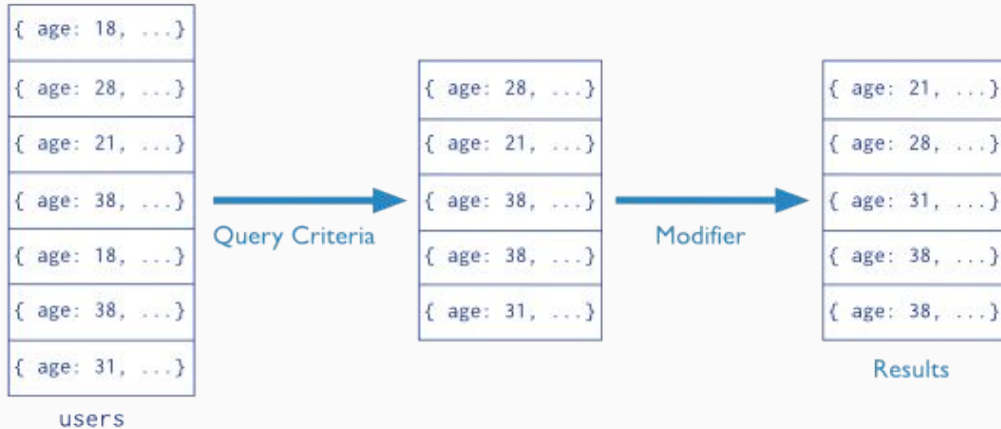
Busquedas - find

Collection
Query Criteria
Modifier
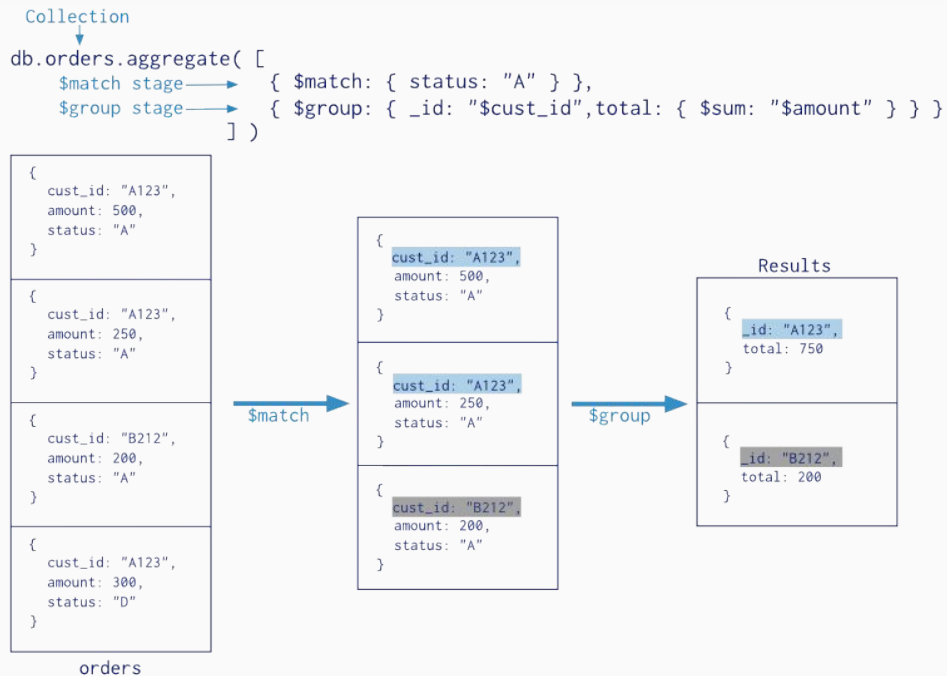
```
db.users.find( { age: { $gt: 18 } } ).sort( {age: 1 } )
```

| { age: 18, ...} |
| { age: 28, ...} |
| { age: 21, ...} |
| { age: 38, ...} |
| { age: 18, ...} |
| { age: 38, ...} |
| { age: 31, ...} |

users

Query Criteria →

| { age: 28, ...} |
| { age: 21, ...} |
| { age: 38, ...} |
| { age: 38, ...} |
| { age: 31, ...} |

Modifier →

| { age: 21, ...} |
| { age: 28, ...} |
| { age: 31, ...} |
| { age: 38, ...} |
| { age: 38, ...} |

Results

Borrar

db.inventory.remove({ type : "food" }, 1)
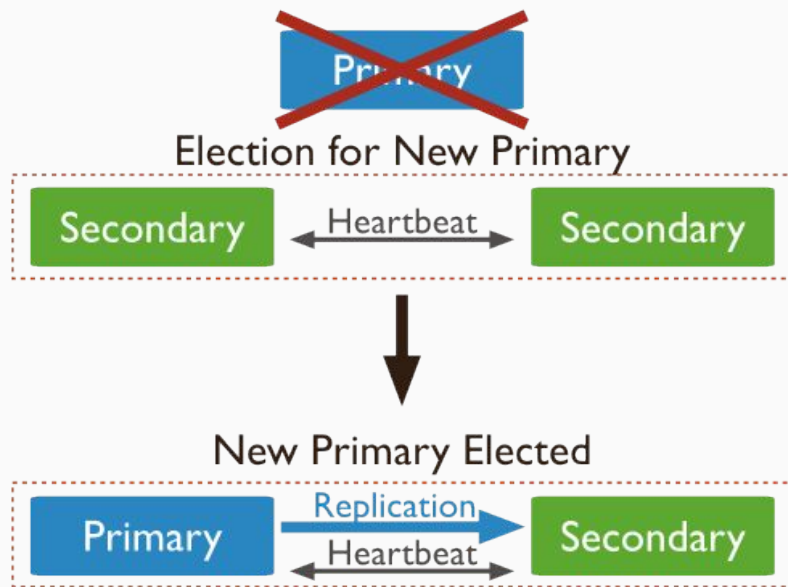
Actualizar

db.inventory.update({_id: 1},

{ $set : { type: "food" } })

Aggregation

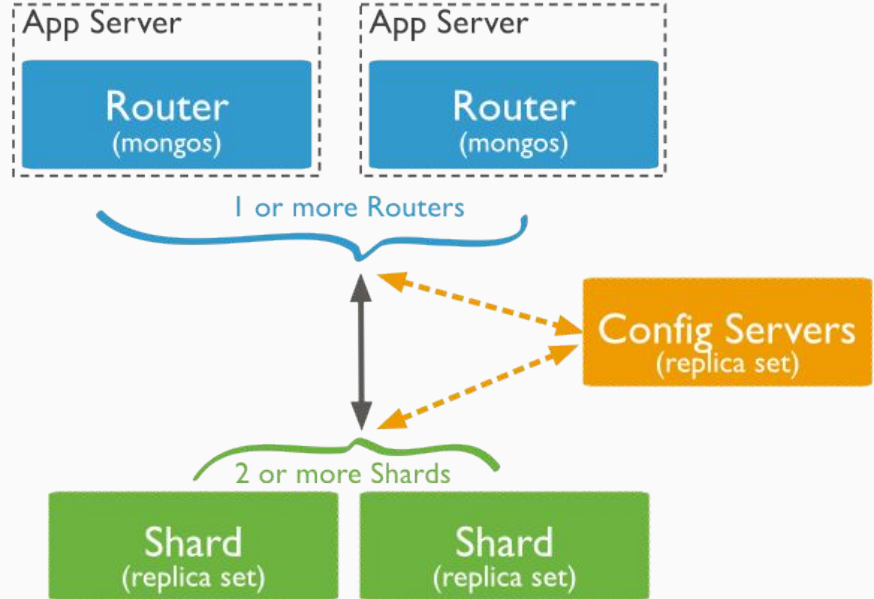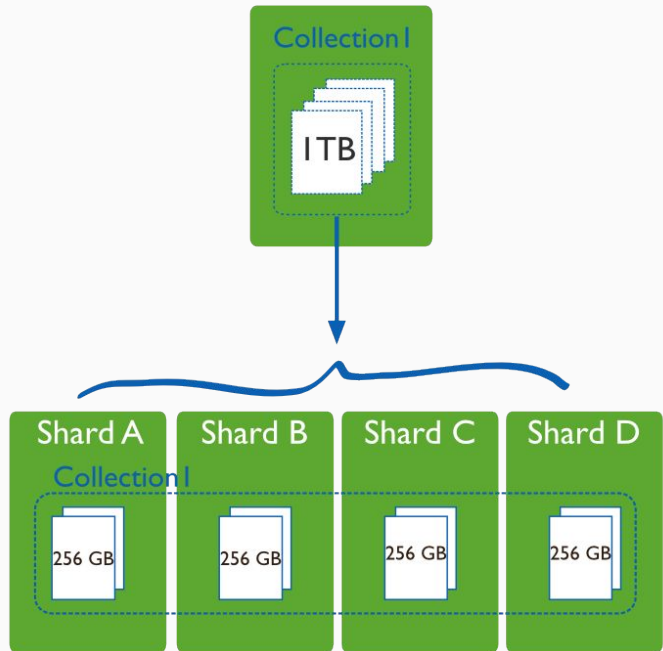
```
                        Collection
                            |
db.orders.aggregate( [
        $match stage ———→   { $match: { status: "A" } },
        $group stage ———→   { $group: { _id: "$cust_id",total: { $sum: "$amount" } } }
                        ] )
```

```
{
    cust_id: "A123",
    amount: 500,
    status: "A"
}
```

```
{
    cust_id: "A123",
    amount: 250,
    status: "A"
}
```

```
{
    cust_id: "B212",
    amount: 200,
    status: "A"
}
```

```
{
    cust_id: "A123",
    amount: 300,
    status: "D"
}
```

orders

→ $match →

```
{
    cust_id: "A123",
    amount: 500,
    status: "A"
}
```

```
{
    cust_id: "A123",
    amount: 250,
    status: "A"
}
```

```
{
    cust_id: "B212",
    amount: 200,
    status: "A"
}
```

→ $group →

Results

```
{
    _id: "A123",
    total: 750
}
```

```
{
    _id: "B212",
    total: 200
}
```

Replica set

Sharding

Gracias!

Preguntas?

- MongoDB University (https://university.mongodb.com/)
- MongoDB Docs (https://docs.mongodb.org/manual/)
- Respositorio Ejemplos (https://github.com/maasencioh/mongo-examples)