



Universidad del
Rosario

Educación
Continua

MongoDB Parte 1

2019

Introducción a MongoDB

Contenido

¿Qué es MongoDB y cómo funciona?

Algunas formas de usar MongoDB (Instalación)

Fundamentos de MongoDB

- Modelo de documento
- Importar datos
- CRUD
- Aggregation Framework

¿Qué es MongoDB y cómo funciona?

¿Qué es? MongoDB (de *Humongous*)

*“MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En lugar de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en **documentos** similares a **JSON** con un esquema dinámico (MongoDB utiliza una especificación llamada **BSON**), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida”.*

Wikipedia

Cómo usar MongoDB

- Para guardar datos y usarlos
- Para hacer Data Science
- Para hacer Analytics
- Para hacer aplicaciones Web/Móviles

¿Qué es MongoDB y cómo funciona?

Nuevo tipo de BD

- NO tablas
- NO SQL
- NO filas
- NO transacciones
- NO joins
- NO foreign keys
- SI **documentos**



¿Qué es MongoDB y cómo funciona?

Ventajas

- Rápida
- Escalable
- Fácil de usar
- Perfecta para guardar estructuras de datos complejas
- Analítica Real Time

Desventajas

- Deja algunas cosas atrás
- No se puede usar en un sistema de contabilidad (falta de soporte de transacciones)

Tip: Se pueden tener una solución híbrida (RDBMS + MongoDB)

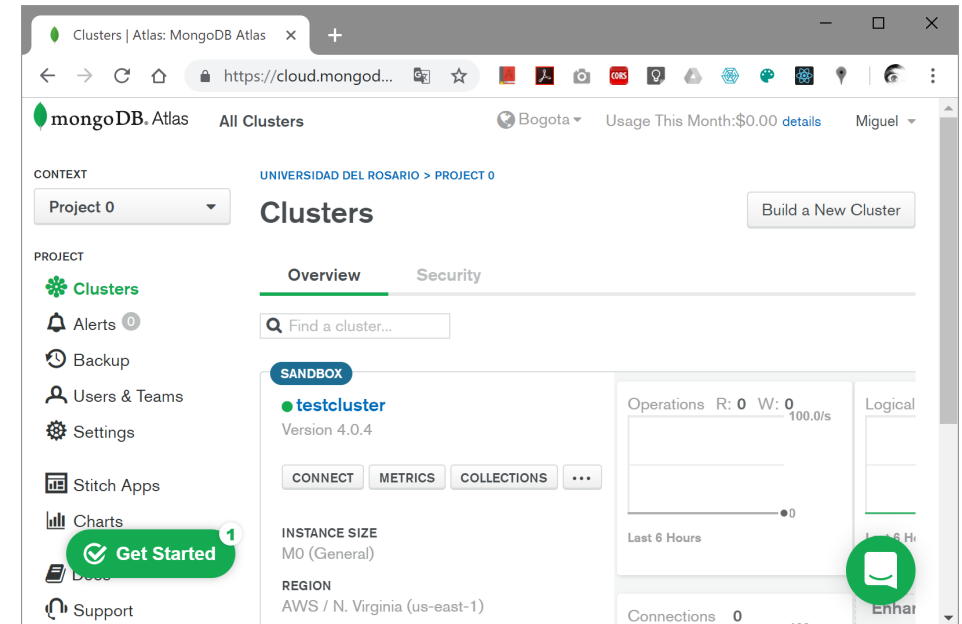
Algunas formas de usar MongoDB

En infraestructura propia



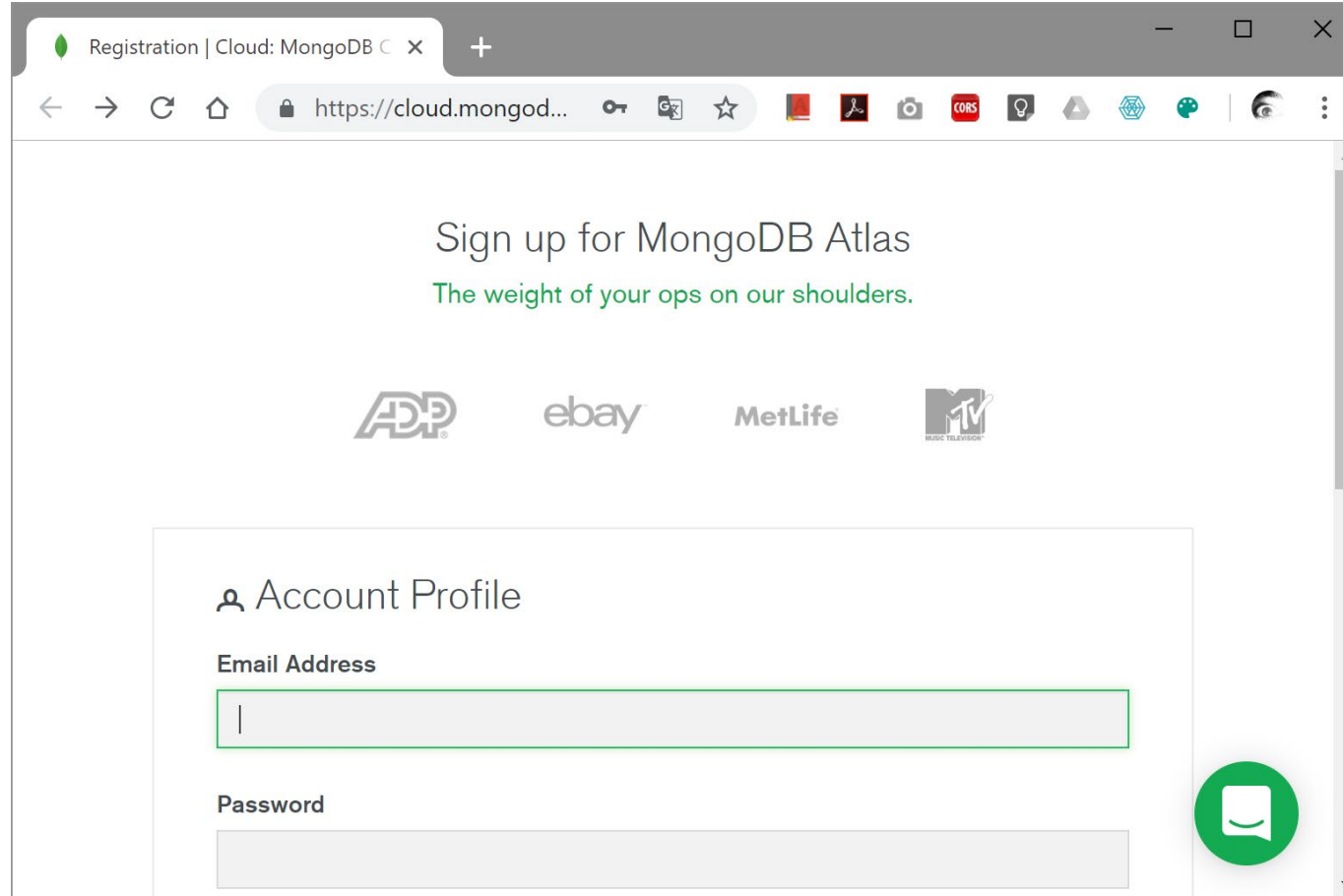
```
Administrator Command Prompt - "C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe"
> db.inventory.find( {} )
{ "_id" : ObjectId("5c1726019e763fcl83c5357"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5c1726019e763fcl83c5358"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5c1726019e763fcl83c5359"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5c1726019e763fcl83c535a"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5c1726019e763fcl83c535b"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5c1726139e763fcl83c535c"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5c1726139e763fcl83c535d"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5c1726139e763fcl83c535e"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5c1726139e763fcl83c535f"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5c1726139e763fcl83c5360"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5c1726239e763fcl83c5361"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5c1726239e763fcl83c5362"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5c1726239e763fcl83c5363"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5c1726239e763fcl83c5364"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5c1726239e763fcl83c5365"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5c1727639e763fcl83c5366"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5c1727639e763fcl83c5367"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5c1727639e763fcl83c5368"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5c1727639e763fcl83c5369"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5c1727639e763fcl83c536a"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
```

Como servicio en la nube (Cloud)



Algunas formas de usar MongoDB

Como servicio en la nube



The screenshot shows a web browser window with the title "Registration | Cloud: MongoDB C x". The address bar displays "https://cloud.mongod...". The main heading is "Sign up for MongoDB Atlas" with the tagline "The weight of your ops on our shoulders." Below this, there are logos for ADP, ebay, MetLife, and MTV. The registration form is titled "Account Profile" and includes two input fields: "Email Address" and "Password". The "Email Address" field is currently active, indicated by a green border and a cursor. A green chat bubble icon is visible in the bottom right corner of the page.

Registration | Cloud: MongoDB C x

https://cloud.mongod...

Sign up for MongoDB Atlas

The weight of your ops on our shoulders.

ADP ebay MetLife MTV

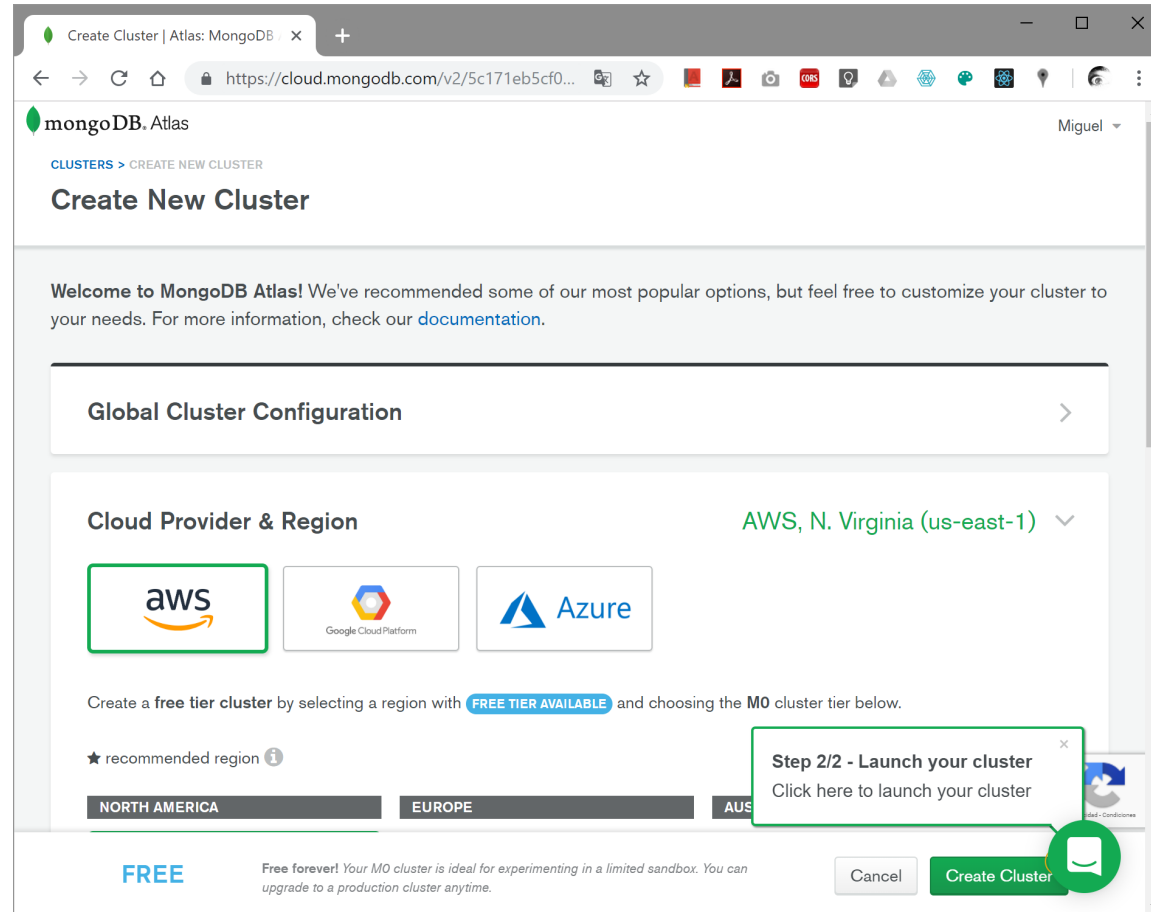
Account Profile

Email Address

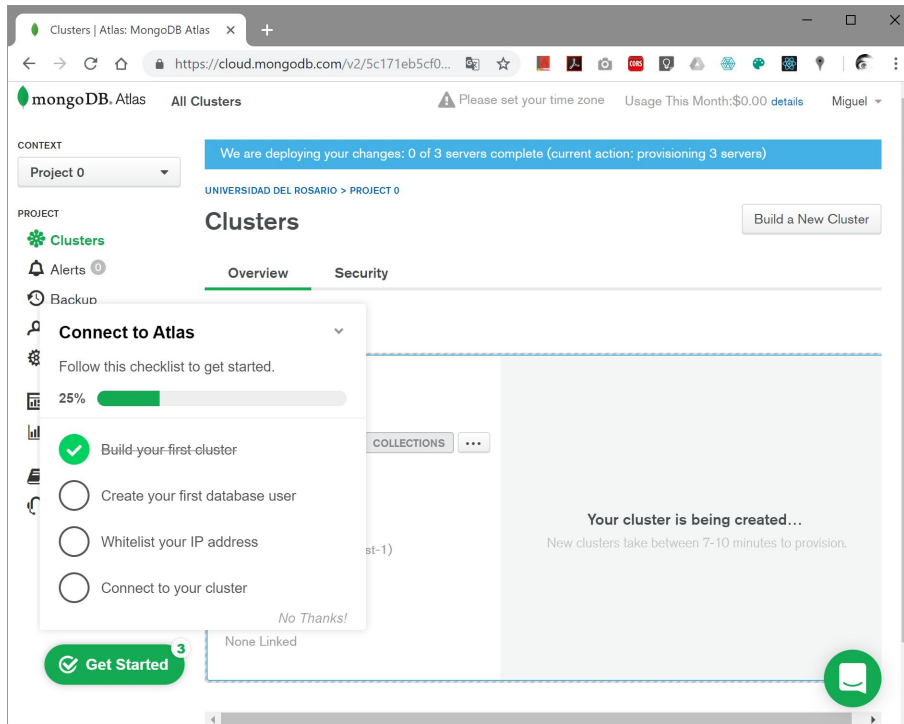
Password

Algunas formas de usar MongoDB

Como servicio en la nube



Algunas formas de usar MongoDB Como servicio en la nube



Connect to testcluster

✓ Setup connection security ✓ Choose a connection method Connect

- 1 Copy the connection string compatible with your driver version:
Check which MongoDB versions your driver version is compatible with
[See documentation on how to check the version of your driver](#)

Short SRV connection string (For drivers compatible with MongoDB 3.6+)

Standard connection string (For drivers compatible with MongoDB 3.4+)

Copy the URI connection string:

```
mongodb://miguelmongo:<PASSWORD>@testcluster-shard-00-00-ibdw9.mongodb.net:27017,testcluster-shard-00-01-ibdw9.mongodb.net:27017,testcluster-shard-00-02-ibdw9.mongodb.net:27017/test?ssl=true&replicaSet=testcluster-shard-00&authSource=admin&retryWrites=true
```

COPY

Note: If using the node.js driver make sure you specify the name of your database after making your connection ([example](#)), otherwise your collections will all appear in a database called "test". Alternatively you can replace "test" in the connection string with a different default database name.

Algunas formas de usar MongoDB

En infraestructura propia

The screenshot shows the MongoDB Download Center website. The browser address bar displays <https://www.mongodb.com/download-cente...>. The navigation bar includes links for DOCS, LEARN, WHAT'S MONGODB?, BLOG, LOGIN, and a 'Get MongoDB' button. The main content area has tabs for Cloud, Server (selected), and Tools. Below the tabs, it says 'Select the server you would like to run:'. There are two options: 'MongoDB Community Server' (FEATURE RICH. DEVELOPER READY.) and 'MongoDB Enterprise Server' (ADVANCED FEATURES. PERFORMANCE GRADE.). Under the Community Server tab, there are dropdown menus for 'Version' (4.0.4 (current release)) and 'OS' (Windows 64-bit x64). Below these is a 'Package' dropdown (MSI) and a green 'Download' button. At the bottom, a URL is provided: https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.4-signed.msi. On the right side, there is a list of links: Release notes, Changelog, All version binaries, Installation instructions, Download source (tgz), and Download source (zip).

MongoDB Download Center | Mc x

← → ↻ 🏠 🔒 <https://www.mongodb.com/download-cente...> ☆

DOCS LEARN WHAT'S MONGODB? BLOG LOGIN

📞 🔍 [Get MongoDB](#) 📄

mongoDB. | FOR GIANT IDEAS SOLUTIONS CLOUD CUSTOMERS RESOURCES ABOUT US

Cloud **Server** Tools

Select the server you would like to run:

MongoDB Community Server
FEATURE RICH. DEVELOPER READY.

MongoDB Enterprise Server
ADVANCED FEATURES. PERFORMANCE GRADE.

Version
4.0.4 (current release) ▾

OS
Windows 64-bit x64 ▾

Package
MSI ▾

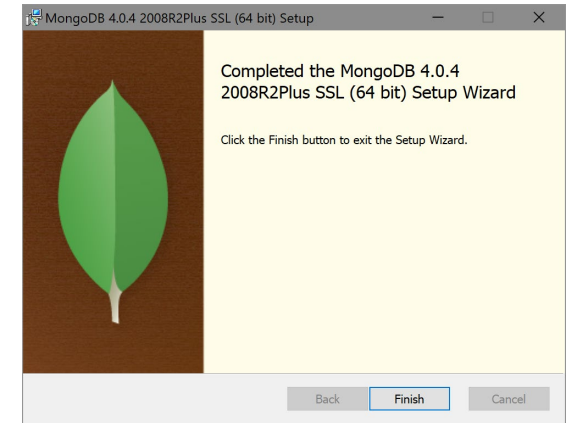
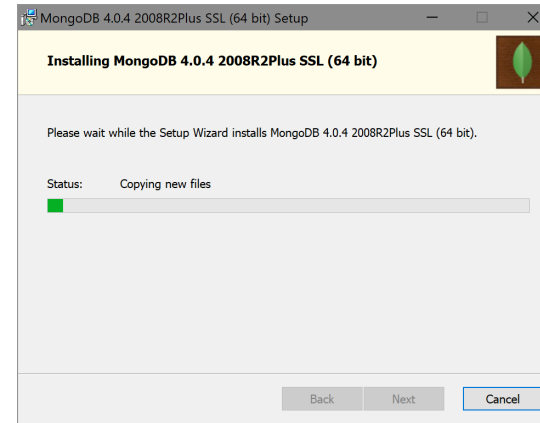
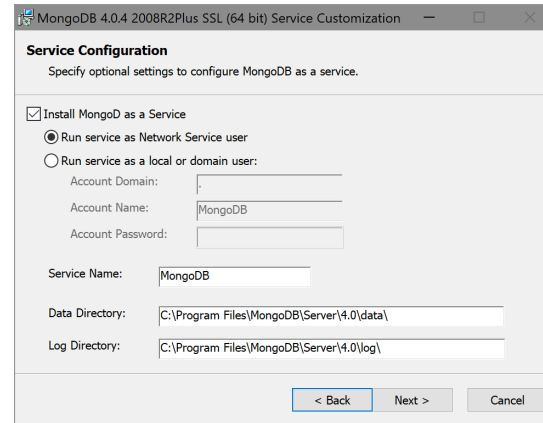
[Download](#)

https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.4-signed.msi

- Release notes
- Changelog
- All version binaries
- Installation instructions
- Download source (tgz)
- Download source (zip)

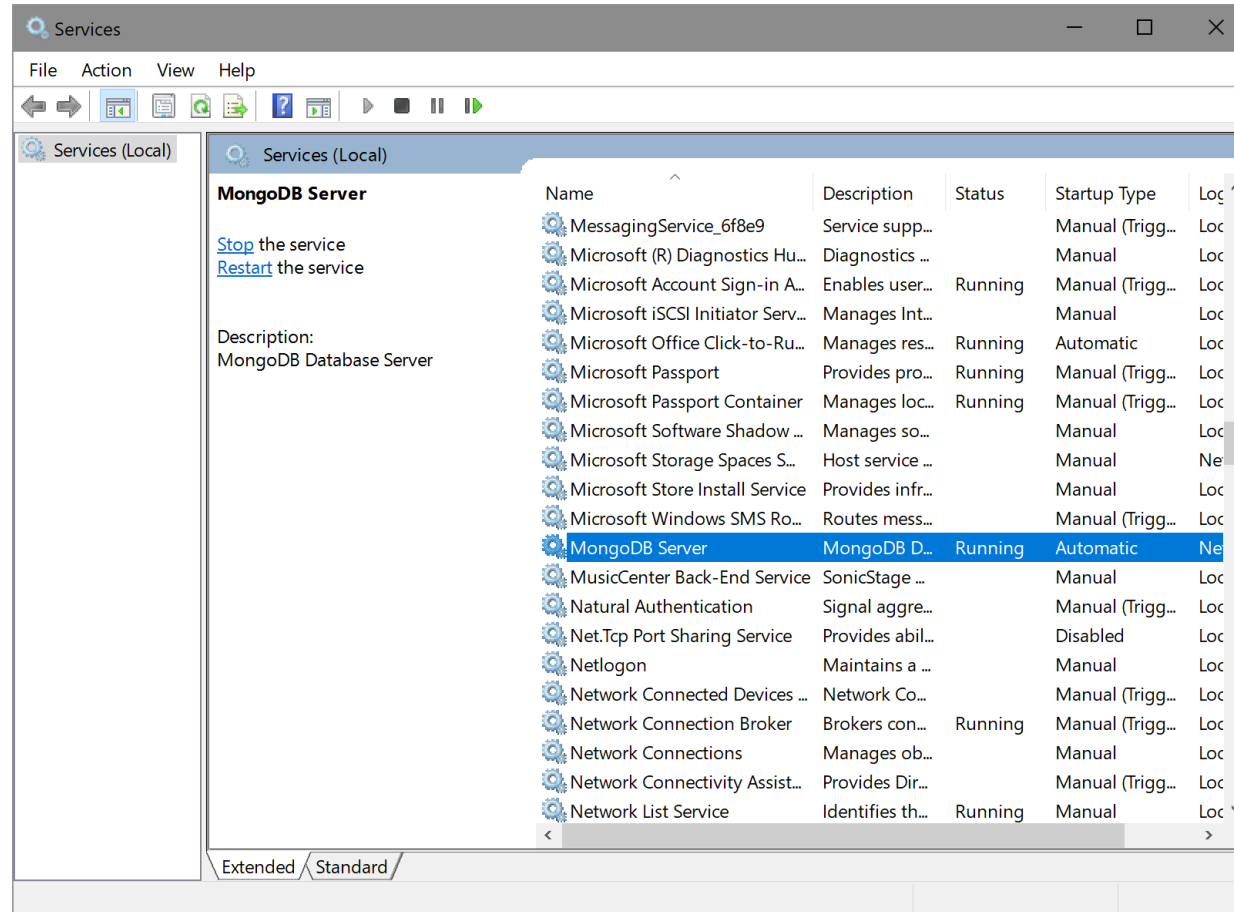
Algunas formas de usar MongoDB

En infraestructura propia



Algunas formas de usar MongoDB

En infraestructura propia



Algunas formas de usar MongoDB

En infraestructura propia

```
C:\Program Files\MongoDB\Server\4.0\bin>mongo.exe
MongoDB shell version v4.0.4
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("9bc16cae-b29a-4d58-93e7-f4276c3e23da") }
MongoDB server version: 4.0.4
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2018-12-16T23:07:11.073-0500 I CONTROL [initandlisten]
2018-12-16T23:07:11.073-0500 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-12-16T23:07:11.073-0500 I CONTROL [initandlisten] **           Read and write access to data and configuration is unrestricted.
2018-12-16T23:07:11.073-0500 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

Fundamentos de MongoDB

- Modelo de **documento**
- Importar datos
- CRUD
- Framework de agregación

Dos formas de almacenar información

Apellido	Nombre	Telefono
Mendez	Pedro	+852 1234 5678
Torres	Walter	+81 1234 5678

CSV (Comma Separated Value)

Apellido, Nombre, Telefono

Mendez, Pedro, +852 1234 5678

Torres, Walter, +81 1234 5678

JSON (JavaScript Object Notation)

```
{  
  "apellido": "Mendez",  
  "nombre": "Pedro",  
  "telefono": "+852 1234 5678"  
}
```

llave: valor

Documentos


Un **documento** consiste en parejas **llave-valor**

llave: valor

```
{  
  "apellido": "Mendez",  
  "nombre": "Pedro",  
  "telefonos": ["+852 1234 5678", "+852 4321 0987"]  
}
```

Documentos

```
{  
  "apellido": "Mendez",  
  "nombre": "Pedro",  
  "telefono": "+852 1234 5678"  
}
```



```
{  
  "apellido": "Mendez",  
  "nombre": "Pedro",  
  "telefonos": [  
    {  
      "casa": "+852 1234 5678"  
    },  
    {  
      "oficina": "+852 4321 0987"  
    }  
  ]  
}
```

Fundamentos de MongoDB – Modelo de documento

Documentos

```
{
  "Type": "CD",
  "Artist": "Nirvana",
  "Title": "Nevermind",
  "Genre": "Grunge",
  "Releasedate": "1991.09.24",
  "Tracklist": [
    {
      "Track": "1",
      "Title": "Smells Like Teen Spirit",
      "Length": "5:02"
    },
    {
      "Track": "2",
      "Title": "In Bloom",
      "Length": "4:15"
    }
  ]
}
```

```
{
  "type": "Book",
  "Title": "Definitive Guide to MongoDB 3rd ed",
  "ISBN": "978-1-4842-1183-0",
  "Publisher": "Apress",
  "Author": [
    "Hows, David",
    "Plugge, Eelco",
    "Membrey, Peter",
    "Hawkins, Tim"
  ]
}
```

Pregunta: Estos dos elementos conviven en la misma **colección** llamada Media, ¿por qué?

Documentos

- Este tipo de información se guarda de forma muy eficiente
- El funcionamiento de los documentos de MongoDB se hace en un formato llamado BSON (Binary JSON)
- Podemos hacer estructuras complejas y consultarlas mediante el lenguaje de consulta de MongoDB, índices secundarios y herramientas de analítica como MongoDB Integration Framework.

Modelo de base de datos

Relacional

Base de datos



Tablas



Filas



MongoDB

Base de datos



Colecciones



Documentos



Fundamentos de MongoDB – Modelo de documento

Tipos de datos

- String: Valores de texto `{"Pais": "Colombia"}`
- Integer: Valores enteros `{"Rank": 1 }`
- Boolean: Valores TRUE o FALSE `{"Alergico": TRUE }`
- Double: Valores punto flotante `{"Precio": 6.4 }`
- Arrays: Arreglos de elementos `["Magnus Carlsen", "Garri Kasparov", "Bobby Fischer", "Alphazero"]`
- Null: Valores nulos `{"Precio": Null }`
- Object: Documentos embebidos
- Timestamps
- Symbol
- Date
- Object ID
- Binary data
- Expresión regular
- Código Javascript

```
{
  item: "journal",
  qty: 25,
  size: {
    h: 14,
    w: 21,
    uom: "cm"
  },
  status: "A"
}
```

subdocumento
embebido

Fundamentos de MongoDB – Modelo de documento

Documentos

```
{
  "Type": "CD",
  "Artist": "Nirvana",
  "Title": "Nevermind",
  "Genre": "Grunge",
  "Releasedate": "1991.09.24",
  "Tracklist": [
    {
      "Track": "1",
      "Title": "Smells Like Teen Spirit",
      "Length": "5:02"
    },
    {
      "Track": "2",
      "Title": "In Bloom",
      "Length": "4:15"
    }
  ]
}
```

```
{
  "type": "Book",
  "Title": "Definitive Guide to MongoDB 3rd ed",
  "ISBN": "978-1-4842-1183-0",
  "Publisher": "Apress",
  "Author": [
    "Hows, David",
    "Plugge, Eelco",
    "Membrey, Peter",
    "Hawkins, Tim"
  ]
}
```

String: Valores de texto
Integer: Valores enteros
Boolean: Valores TRUE o FALSE
Double: Valores punto flotante
Arrays: Arreglos de elementos
Null: Valores nulos
Object: Documentos embebidos

Pregunta: ¿Qué tipos de datos identifica en los documentos anteriores?

Colecciones

- Contenedor de documentos
- Cada colección debe tener un nombre único
 - Inicia con una letra o underscore (_)
 - Puede tener números y letras
 - El símbolo \$ está reservado
 - No puede iniciar con system.

Fundamentos de MongoDB – Importar datos

Conexión + Crear colección

Connect to testcluster

✓ Setup connection security > ✓ Choose a connection method > Connect

- 1 Copy the connection string compatible with your driver version:
[Check which MongoDB versions your driver version is compatible with](#)
[See documentation on how to check the version of your driver](#)

Short SRV connection string (For drivers compatible with MongoDB 3.6+)

Standard connection string (For drivers compatible with MongoDB 3.4+)

Copy the URI connection string:

```
mongodb://miguelmongo:<PASSWORD>@testcluster-shard-00-00-ibdw9.mongodb.net:27017,testcluster-shard-00-01-ibdw9.mongodb.net:27017,testcluster-shard-00-02-ibdw9.mongodb.net:27017/test?ssl=true&replicaSet=testcluster-shard-00&authSource=admin&retryWrites=true
```

COPY

Note: If using the node.js driver make sure you specify the name of your database after making your connection ([example](#)), otherwise your collections will all appear in a database called "test". Alternatively you can replace "test" in the connection string with a different default database name.

```
base_a_guardar <- mtcars
View(base_a_guardar)
nrow(base_a_guardar)
```

```
library(mongolite)

# Haciendo conexión al servidor
con <- mongo(collection = "mtcars", url = "mongodb://miguelmongo:<PASSWORD>@testc...")

con$count()
if(con$count() > 0) {
  con$drop()
}
con$count()
con$insert(base_a_guardar)
con$count()

# Consultando los datos
mydata <- con$find()

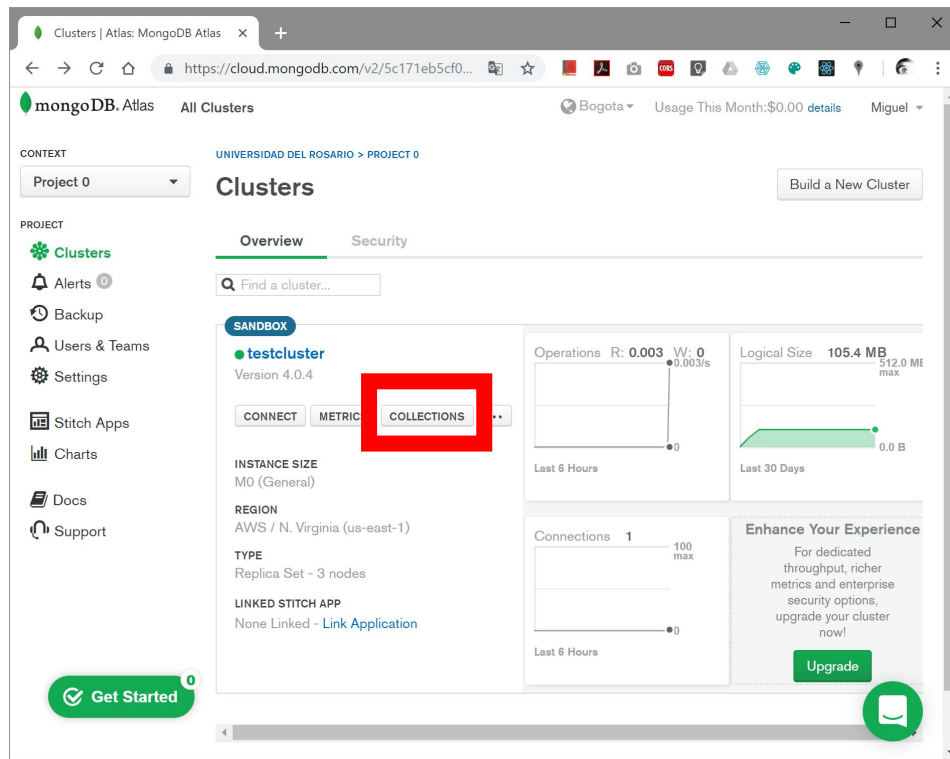
# Borrando los datos
con$drop()

# Se desconecta de la base de datos cuando se remueve la conexión
rm(con)
gc()
```

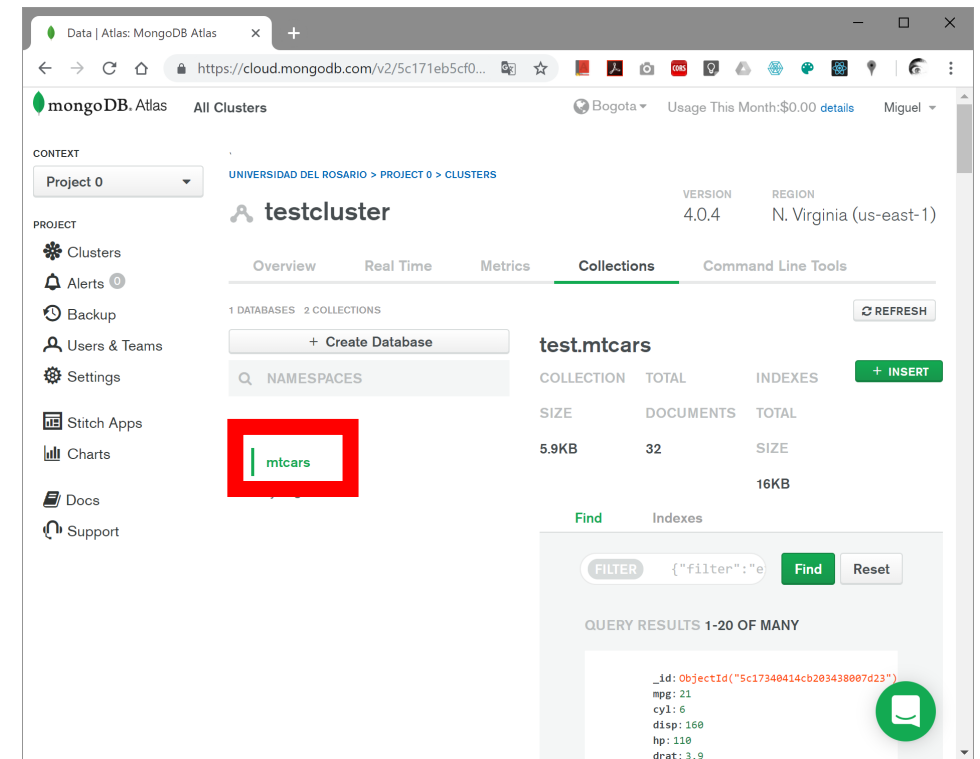
1_cadenadeconexion.R

Fundamentos de MongoDB – Importar datos

Conexión + Crear colección



The screenshot shows the 'Clusters' page for a project named 'UNIVERSIDAD DEL ROSARIO > PROJECT 0'. The cluster 'testcluster' is highlighted. The 'COLLECTIONS' tab is selected and highlighted with a red box. The cluster details include: Version 4.0.4, Instance Size M0 (General), Region AWS / N. Virginia (us-east-1), Type Replica Set - 3 nodes, and Linked Stitch App None Linked. The 'COLLECTIONS' tab is highlighted with a red box.



The screenshot shows the 'Collections' page for the 'testcluster' project. The 'mtcars' collection is highlighted with a red box. The collection details include: 1 DATABASES, 2 COLLECTIONS, and a table showing the collection 'mtcars' with 5.9KB size, 32 documents, and 16KB total size. The 'mtcars' collection is highlighted with a red box.

COLLECTION	TOTAL	INDEXES
mtcars	5.9KB	32

QUERY RESULTS 1-20 OF MANY

```
{ "_id": ObjectId("5c17340414cb203438007023"), "mpg": 21, "cyl": 6, "displ": 160, "hp": 110, "drat": 3.9 }
```

Fundamentos de MongoDB – Importar datos

Consultar colección

A data frame with 32 observations on 11 variables.

```
[, 1] mpg Miles/(US) gallon  
[, 2] cyl Number of cylinders  
[, 3] disp Displacement (cu.in.)  
[, 4] hp Gross horsepower  
[, 5] drat Rear axle ratio  
[, 6] wt Weight (1000 lbs)  
[, 7] qsec 1/4 mile time  
[, 8] vs V/S  
[, 9] am Transmission (0 = automatic, 1 = manual)  
[,10] gear Number of forward gears  
[,11] carb Number of carburetors
```

The screenshot shows the MongoDB Atlas web interface. The browser tab is 'Data | Atlas: MongoDB Atlas'. The URL is 'https://cloud.mongodb.com/v2/5c171eb5cf09a20f2b3da114#metrics/re...'. The page title is 'mongoDB Atlas All Clusters'. The user is 'Bogota' and the usage for this month is '\$0.00'. The left sidebar shows the 'PROJECT' menu with options: Clusters, Alerts, Backup, Users & Teams, Settings, Stitch Apps, Charts, Docs, and Support. The main content area has tabs for 'Overview', 'Real Time', 'Metrics', 'Collections', and 'Command Line Tools'. The 'Collections' tab is active, showing '1 DATABASES 2 COLLECTIONS'. A 'test' database is expanded, showing 'mtcars' and 'nycflights' collections. The 'mtcars' collection is selected, showing a 'COLLECTION SIZE' of 5.9KB, 32 'DOCUMENTS', and 16KB 'INDEXES TOTAL'. A '+ INSERT' button is visible. Below this, a 'Find' section shows a filter '{ am: 0 }' and a 'Find' button. The 'QUERY RESULTS 1-19 OF 19' section displays a JSON document for a Lincoln Continental with automatic transmission.

```
{  
  "_id": ObjectId("5c17340414cb203438007d32"),  
  "mpg": 10.4,  
  "cyl": 8,  
  "disp": 460,  
  "hp": 215,  
  "drat": 3,  
  "wt": 5.424,  
  "qsec": 17.82,  
  "vs": 0,  
  "am": 0,  
  "gear": 3,  
  "carb": 4,  
  "_row": "Lincoln Continental"  
}
```

Pregunta: ¿Cuántos autos tienen transmisión automática?

Operaciones CRUD



CREATE



READ



UPDATE



DELETE

C

R

U

D

Nota

Se abordarán las operaciones CRUD en MongoDB desde dos clients: la **Shell de mongo**, y el paquete **mongolite para R**.

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

```
subjects$insert('{  
  "item": "canvas",  
  "qty": "100",  
  "tags": ["cotton"],  
  "size": {  
    "h": "28",  
    "w": "35.5",  
    "uom": "cm"  
  }  
}')
```

Create

Agrega documentos nuevos a una colección. Si la colección no existe se crea.

- `db.collection.insertOne()`
 - `db.collection.insertMany()`
- } Métodos de inserción (apuntan a una sola colección)

Operaciones atómicas a nivel de documento sencillo

```
db.users.insertOne(  
  {  
    name: "sue",  
    age: 26,  
    status: "pending"  
  }  
)
```

← collection

← field: value
← field: value
← field: value } document

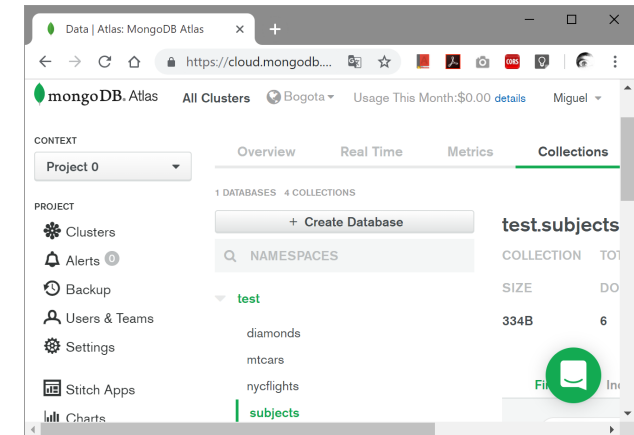
Fundamentos de MongoDB – Operaciones CRUD

Create

```
library(mongolite)
library(ggplot2)

mongourl <- "mongodb://miguelmongo:<PASSWORD>..."

subjects <- mongo("subjects", url = mongourl)
subjects$insert('{"name" : "blanca"}')
subjects$find(query = '{}', fields = '{}')
str <- c('{"name" : "jerry"}',
        '{"name": "anna", "age" : 23}',
        '{"name": "joe"}')
subjects$insert(str)
subjects$find(query = '{}', fields = '{}')
```



Cargar BD diamonds

```
library(mongolite)
library(ggplot2)

View(diamonds)

dmd <- mongo(collection = "diamonds", url = "mongodb://miguelmongo:<PASSWORD>")
dmd$insert(diamonds)

dmd$count()
```

4_creandodiamonds.R



Pregunta: ¿Qué hace el comando **insert()** en este caso?

Pregunta: ¿Qué hace el comando **count()** ?

Read

Recupera documentos de una colección. Es una consulta en una ~~tabla~~ colección que retorna ~~filas~~ documentos

- **`db.collection.find()`**

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection
← query criteria
← projection
← cursor modifier

Síntaxis de consulta

MongoDB utiliza la sintaxis basada en JSON para consultar documentos. La consulta vacía {} significa: seleccionar todos los datos. El mismo parámetro de consulta se usa para múltiples operaciones como **find ()**, **iterate ()**, **count ()**, **remove ()** y **update ()**. Necesitamos especificar la consulta JSON como una cadena en R.

Pregunta: ¿Qué hace el comando **insert()** ?

Síntaxis de consulta

SQL	MongoDB
<pre>SELECT * FROM inventory;</pre>	<pre>db.inventory.find({})</pre>
<pre>SELECT * FROM inventory WHERE status = "D";</pre>	<pre>db.inventory.find({ status: "D" })</pre>
<pre>SELECT * FROM inventory WHERE status in (" A ", " D ");</pre>	<pre>db.inventory.find({ status: { \$in: ["A", "D"] } })</pre>

Síntaxis de consulta

SQL	MongoDB
<pre>SELECT * FROM inventory WHERE status = " A " AND qty < 30;</pre>	<pre>db.inventory.find({ status: "A", qty: { \$lt: 30 } })</pre>
<pre>SELECT * FROM inventory WHERE status = " A " OR qty < 30;</pre>	<pre>db.inventory.find({ \$or: [{ status: "A" }, { qty: { \$lt: 30 } }] })</pre>
<pre>SELECT * FROM inventory WHERE status = "A" AND (qty < 30 OR item LIKE "p%")</pre>	<pre>db.inventory.find({ status: "A", \$or: [{ qty: { \$lt: 30 } }, { item: /^p/ }] })</pre>

Fundamentos de MongoDB – Operaciones CRUD

Algunos Operadores

De comparación

<code>\$eq</code>	Matches values that are equal to a specified value.
<code>\$gt</code>	Matches values that are greater than a specified value.
<code>\$gte</code>	Matches values that are greater than or equal to a specified value.
<code>\$in</code>	Matches any of the values specified in an array.
<code>\$lt</code>	Matches values that are less than a specified value.
<code>\$lte</code>	Matches values that are less than or equal to a specified value.
<code>\$ne</code>	Matches all values that are not equal to a specified value.
<code>\$nin</code>	Matches none of the values specified in an array.

Lógicos

<code>\$and</code>	Joins query clauses with a logical AND returns all documents that match the conditions of both clauses.
<code>\$not</code>	Inverts the effect of a query expression and returns documents that do <i>not</i> match the query expression.
<code>\$nor</code>	Joins query clauses with a logical NOR returns all documents that fail to match both clauses.
<code>\$or</code>	Joins query clauses with a logical OR returns all documents that match the conditions of either clause.

De elemento

<code>\$exists</code>	Matches documents that have the specified field.
<code>\$type</code>	Selects documents if a field is of the specified type.

Ejercicio 5: Algunas consultas

Toda la colección Diamonds

Los documentos donde cut es Ideal

Los documentos en los que el color es D o E o F

Los documentos donde el color es D y el precio es menor que 400

Los documentos donde el color es D o el precio es menor que 400

Los documentos donde el corte es Fair, y además el color es D o su precio es menor que 400

Algunas consultas (solución)

```
dmd$find('{}' )
```

```
dmd$find('{ "cut": "Ideal" }' )
```

```
dmd$find('{ "color": { "$in": ["D","E","F"] } }')
```

```
dmd$find('{ "color": "D", "price": { "$lt": 400 } }')
```

```
dmd$find('{ "$or": [{ "color": "D" }, { "price": { "$lt": 400 } } ] }')
```

```
dmd$find('{ "cut": "Fair" , "$or": [{ "color": "D" }, { "price": { "$lt": 400 } } ] }')
```

Consultas sobre documentos embebidos/anidados

Crearemos primero una colección de inventario

```
# Sobre documentos embebidos
library(mongolite)

mongourl <- "mongodb://..."

inventario_data <- c(
  '{ "item": "journal", "qty": "25", "size": { "h": "14", "w": "21", "uom": "cm" }, "status": "A" }',
  '{ "item": "notebook", "qty": "50", "size": { "h": "8.5", "w": "11", "uom": "in" }, "status": "A" }',
  '{ "item": "paper", "qty": "100", "size": { "h": "8.5", "w": "11", "uom": "in" }, "status": "D" }',
  '{ "item": "planner", "qty": "75", "size": { "h": "22.85", "w": "30", "uom": "cm" }, "status": "D" }',
  '{ "item": "postcard", "qty": "45", "size": { "h": "10", "w": "15.25", "uom": "cm" }, "status": "A" }'
)

inventario <- mongo("inventario", url = mongourl)
inventario$insert(inventario_data)
```


Consultas sobre documentos incrustados/anidados

```
db.inventory.find( { size: { h: 14, w: 21, uom: "cm" } } )
```

```
db.inventory.find( { size: { w: 21, h: 14, uom: "cm" } } )
```

¿retornarán el mismo documento?

NO. Las coincidencias de igualdad en todo el documento incrustado requieren una **coincidencia exacta** del documento <valor> especificado, incluido el **orden de los campos**.

Consultas sobre documentos incrustados/anidados

Seleccionando todos los documentos donde el campo **uom** en el documento **size** es igual a **"in"**

```
db.inventory.find( { "size.uom": "in" } )
```

```
inventario$find('{"size.uom": "in"}')
```

Seleccionando todos los documentos donde el campo **h** en el documento **size** es menor a **15**

```
db.inventory.find( { "size.h": { $lt: 15 } } )
```

```
inventario$find('{"size.h": { "$lt": "15" }}')
```

Actividad: Cree consulta que seleccione todos los documentos donde el campo anidado **h** es menor que **15**, el campo anidado **uom** es igual a **"in"** y el campo de estado es igual a **"D"**

Consultas sobre arreglos

Crearemos primero otra colección de inventario

```
# Sobre arreglos
library(mongolite)

mongourl <- "mongodb://miguelmongo:qwertymongo@"

inventario_data <- c(
  '{ "item": "journal", "qty": "25", "tags": ["blank", "red"], "dim_cm": [ "14", "21" ] }',
  '{ "item": "notebook", "qty": "50", "tags": ["red", "blank"], "dim_cm": [ "14", "21" ] }',
  '{ "item": "paper", "qty": "100", "tags": ["red", "blank", "plain"], "dim_cm": [ "14", "21" ] }',
  '{ "item": "planner", "qty": "75", "tags": ["blank", "red"], "dim_cm": [ "22.85", "30" ] }',
  '{ "item": "postcard", "qty": "45", "tags": ["blue"], "dim_cm": [ "10", "15.25" ] }'
)

inventario <- mongo("inventario_arrays", url = mongourl)
inventario$insert(inventario_data)
```

Consultas sobre arreglos

Consulta todos los documentos donde el campo tags es un arreglo con exactamente dos elementos, en el orden especificado

```
db.inventory.find( { tags: ["red", "blank"] } )
```

```
inventario$find('{"tags":["red","blank"]}')
```

Consulta todos los documentos donde el campo tags es un arreglo con exactamente dos elementos, en el orden especificado

```
db.inventory.find( { tags: { $all: ["red", "blank"] } } )
```

```
inventario$find('{"tags": { "$all": ["red","blank"] } }')
```

Consultas sobre arreglos

Consulta todos los documentos donde el array **tags** contiene por lo menos el elemento **"red"**

```
db.inventory.find( { tags: "red" } )
```

```
inventario$find('{"tags": "red" }')
```

Consulta todos los documentos donde el array **dim_cm** contiene por lo menos un elemento cuyo valor es mayor a **25**

```
db.inventory.find( { dim_cm: { $gt: 25 } } )
```

```
inventario$find('{"dim_cm": { "$gt": "25"} }')
```

Actividad: Cree consulta que seleccione todos los documentos donde el array **dim_cm** es menor que **25** y el array tags contiene **"blank"**

Consultas sobre arreglos

Consulta todos los documentos donde hay por lo menos un elemento que satisfaga las condiciones.

```
db.inventory.find( { dim_cm: { $gt: 15, $lt: 20 } } )
```

```
inventario$find('{"dim_cm": { "$gt" : "15", "$lt": "20" } }')
```

Si se quiere que todos los elementos cumplan las condiciones se debe emplear el operador **\$elemMatch**

```
db.inventory.find( { dim_cm: { $elemMatch: { $gt: 15, $lt: 20 } } } )
```

```
inventario$find('{"dim_cm": { "$elemMatch": { "$gt" : "15", "$lt": "20" } } }')
```

Consultas sobre arreglos

Buscar todos los documentos donde el Segundo elemento en el arreglo **dim_cm** es más grande que 25

```
db.inventory.find( { "dim_cm.1": { $gt: 25 } } )
```

```
inventario$find('{"dim_cm.1": { "$gt" : "25" } }')
```

Buscar por tamaño de arreglo con **\$size**

```
db.inventory.find( { "tags": { $size: 3 } } )
```

```
inventario$find('{"tags": { "$size": "3" } }')
```

Consulta sobre un arreglo de un documento incrustados/anidados

Creando colección

```
# Sobre documentos incrustados con arreglos
library(mongolite)

mongourl <- "mongodb://miguelmongo:qwertymon..."

inventario_data <- c(
  '{ "item": "journal", "instock": [ { "warehouse": "A", "qty": 5 }, { "warehouse": "C", "qty": 15 } ] }',
  '{ "item": "notebook", "instock": [ { "warehouse": "C", "qty": 5 } ] }',
  '{ "item": "paper", "instock": [ { "warehouse": "A", "qty": 60 }, { "warehouse": "B", "qty": 15 } ] }',
  '{ "item": "planner", "instock": [ { "warehouse": "A", "qty": 40 }, { "warehouse": "B", "qty": 5 } ] }',
  '{ "item": "postcard", "instock": [ { "warehouse": "B", "qty": 15 }, { "warehouse": "C", "qty": 35 } ] }'
)

inventario <- mongo("inventario_arraysembdoc", url = mongourl)
inventario$insert(inventario_data)
```


Consulta sobre un arreglo de un documento incrustados/anidados

Match específico de documento

```
db.inventory.find( { "instock": { warehouse: "A", qty: 5 } } )
```

```
inventario$find('{ "instock": { "warehouse": "A", "qty": 5 } }')
```

La siguiente consulta no retorna nada

```
db.inventory.find( { "instock": { qty: 5, warehouse: "A" } } )
```

```
inventario$find('{ "instock": { "qty": 5, "warehouse": "A" } }')
```

Si no conoce la posición del índice del documento anidado en el array, concatene el nombre del campo del array con un punto (.) y el nombre del campo en el documento anidado.

```
db.inventory.find( { 'instock.qty': { $lte: 20 } } )
```

```
inventario$find('{ "instock.qty": { "$lte": 20 } }')
```

Consulta sobre un arreglo de un documento incrustados/anidados

El punto puede ser usado para definir la posición del arreglo a consultar

```
db.inventory.find( { 'instock.0.qty': { $lte: 20 } } )
```

```
inventario$find('{"instock.0.qty": {"$lte": 20} }')
```

```
_id: ObjectId("5c19d50314cb204604004447")
item: "journal"
✓ instock: Array
  ✓ 0: Object
    warehouse: "A"
    qty: 5
  ✓ 1: Object
    warehouse: "C"
    qty: 15
```

Consulta sobre un arreglo de un documento incrustados/anidados

Condiciones multiples en un array

```
db.inventory.find( {  
    "instock": { $elemMatch: { qty: 5, warehouse: "A" } }  
} )
```

```
inventario$find('{"instock": {"$elemMatch": {"qty": 5, "warehouse": "A"} } }')
```

```
db.inventory.find( {  
    "instock": {  
        $elemMatch: { qty: { $gt: 10, $lte: 20 } }  
    }  
} )
```

```
inventario$find('{"instock":{"$elemMatch": {"qty": {"$gt": 10, "$lt": 20} } } }')
```

Consulta sobre un arreglo de un documento incrustados/anidados

Combinación de elementos que satisfacen criterios

Busca documentos donde cualquier documento anidado en un arreglo instock tiene qty mayor que 10 y cualquier otro documento o el mismo tiene qty menor que 20

```
db.inventory.find( { "instock.qty": { $gt: 10, $lte: 20 } })
```

```
inventario$find('{ "instock.qty": { "$gt": 10, "$lte": 20 } })')
```

Cualquiera con qty mayor a 5, con cualquiera con instock.warehouse igual a A

```
db.inventory.find( {  
    "instock.qty": 5, "instock.warehouse": "A" } )
```

```
inventario$find('{ "instock.qty": 5, "instock.warehouse": "A" })')
```

Consultas sobre datos nulos

Filtro de igualdad

```
db.inventory.find( { item: null } )
```

```
nulos$find('{"item": null}')
```

Chequeo de tipo. Retorna los documentos donde el valor de item es null

```
db.inventory.find( { item : { $type: 10 } } )
```

```
nulos$find('{"item": {"$type": "null" } }', '{"_id": true}')
```

Chequeo de existencia. Retorna documentos que no contienen campo de item

```
db.inventory.find( { item : { $exists: false } } )
```

```
nulos$find('{"item": {"$exists": false } }', '{"_id": true}')
```

Fundamentos de MongoDB – Operaciones CRUD

Más sobre tipos de datos

Each BSON type has both integer and string identifiers as listed in the following table:

Type	Number	Alias	Notes
Double	1	"double"	
String	2	"string"	
Object	3	"object"	
Array	4	"array"	
Binary data	5	"binData"	
Undefined	6	"undefined"	Deprecated.
ObjectId	7	"objectId"	
Boolean	8	"bool"	
Date	9	"date"	
Null	10	"null"	

Regular Expression	11	"regex"	
DBPointer	12	"dbPointer"	Deprecated.
JavaScript	13	"javascript"	
Symbol	14	"symbol"	Deprecated.
JavaScript (with scope)	15	"javascriptWithScope"	
32-bit integer	16	"int"	
Timestamp	17	"timestamp"	
64-bit integer	18	"long"	
Decimal128	19	"decimal"	New in version 3.4.
Min key	-1	"minKey"	
Max key	127	"maxKey"	

Filtros

SQL	MongoDB
<pre>SELECT * FROM inventory;</pre>	<pre>db.inventory.find({})</pre>
<pre>SELECT * FROM inventory WHERE status = "A";</pre>	<pre>db.inventory.find({ status: "A" })</pre>
<pre>SELECT _id, item, status FROM inventory WHERE status = "A";</pre>	<pre>db.inventory.find({ status: "A" }, { item: 1, status: 1 })</pre>
<pre>SELECT item, status FROM inventory WHERE status = "A";</pre>	<pre>db.inventory.find({ status: "A" }, { item: 1, status: 1, _id: 0 })</pre>

Fundamentos de MongoDB – Operaciones CRUD

Filtros

Verificar que existe diamonds. Volver a cargar

1 DATABASES 7 COLLECTIONS

+ Create Database

Q NAMESPACES

▼ test

diamonds



inventario

inventario_arrays

inventario_arraysembdoc

mtcars

nycflights

subjects

```
library(mongolite)
library(ggplot2)
```

```
mongourl <- "mongodb://miguelmongo:qwertymo"
```

```
View(diamonds)
```

```
dmd <- mongo(collection = "diamonds", url = mongourl)
dmd$insert(diamonds)
```

13_filtros.R

Filtros

```
db.inventory.find(  
  { status: "A" },  
  { item: 1, status: 1}  
)
```

 true = 1
false = 0

```
dmd$find('{"cut" : "Premium", "price" : { "$lt" : 1000 } }')  
  
dmd$find(  
  query = '{"cut" : "Premium", "price" : { "$lt" : 1000 } }'  
)  
  
dmd$find(  
  query = '{"cut" : "Premium", "price" : { "$lt" : 1000 } }',  
  fields = '{"cut" : true, "clarity" : true}'  
)  
  
dmd$find(  
  query = '{"cut" : "Premium", "price" : { "$lt" : 1000 } }',  
  fields = '{"cut" : true, "clarity" : true}',  
  limit = 5  
)
```

Filtros

Deshabilitar el id

SQL	MongoDB
<pre>SELECT _id, item, status FROM inventory WHERE status = "A";</pre>	<pre>db.inventory.find({ status: "A" }, {item: 1, status: 1})</pre>
<pre>SELECT item, status FROM inventory WHERE status = "A";</pre>	<pre>db.inventory.find({ status: "A" }, {item: 1, status: 1, _id: 0})</pre>

```
dmd$find(
  query = '{"cut" : "Premium", "price" : { "$lt" : 1000 } }',
  fields = '{"cut" : true, "clarity" : true, "_id": false}',
  limit = 5
)
```

Ordenamiento

Ascendente

```
dmd$find('{"cut" : "Premium"}', sort = '{"price": -1}', limit = 7)
```

Descendente


```
dmd$find('{"cut" : "Premium"}', sort = '{"price": 1}', limit = 7)
```

Delete

Borran documentos de la colección.

- **`db.collection.deleteOne()`**
- **`db.collection.deleteMany()`**

```
db.users.deleteMany(  
  { status: "reject" }  
)
```



collection

delete filter

Delete

Cargar iris

```
library(mongolite)

mongourl <- "mongodb://miguelmongo:qwertym"

iriscon <- mongo("iris", url = mongourl)
iriscon$insert(iris)
```

Borrar registros

```
iriscon$remove('{"Sepal_Length" : {"$lte" : 5}}', just_one = TRUE)
```

Update

Modifica documentos existentes en una colección.

- **`db.collection.updateOne()`**
- **`db.collection.updateMany()`**
- **`db.collection.replaceOne()`**

```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "reject" } }  
)
```



← collection

← update filter

← update action

Update

Carga de datos

```
library(mongolite)

mongourl <- "mongodb://miguelmongo:qwertym"

subjects <- mongo("subjects", url = mongourl)
subjects$drop()
str <- c('{"name" : "jerry"}' , '{"name": "anna", "age" : 23}', '{"name": "joe"}')
subjects$insert(str)
```

Modificación

```
subjects$update('{"name":"jerry"}', '{"$set":{"age": 31}}')
```

Agregation Framework

SQL Terms, Functions, and Concepts MongoDB Aggregation Operators

WHERE	<code>\$match</code>
GROUP BY	<code>\$group</code>
HAVING	<code>\$match</code>
SELECT	<code>\$project</code>
ORDER BY	<code>\$sort</code>
LIMIT	<code>\$limit</code>
SUM()	<code>\$sum</code>
COUNT()	<code>\$sum</code> <code>\$sortByCount</code>
join	<code>\$lookup</code>

SQL Example

```
SELECT COUNT(*) AS count
FROM orders
```

MongoDB Example

```
db.orders.aggregate( [
  {
    $group: {
      _id: null,
      count: { $sum: 1 }
    }
  }
] )
```

Tomado de <https://docs.mongodb.com/manual/reference/sql-aggregation-comparison/>

Gracias

Para practicar

Análisis de crimen en Chicago

- <https://datascienceplus.com/using-mongodb-with-r/>