



UD17 – MongoDB

Diferencia DB SQL / no SQL

SQL	NoSQL
Permite unir tablas con JOIN	No permite el uso de JOINS o están muy limitados
Dificultad para distribuir los datos	Facilidad para distribuir los datos
Escalabilidad vertical (mejorar potencia del servidor para obtener mejores resultados)	Escalabilidad horizontal (repartir / distribuir base de datos en diferentes servidores)
Los datos se estructuran siguiendo los esquemas de las tablas	Los datos no siguen ningún esquema
Posibilidad de crear restricciones, triggers, claves foráneas... sobre los datos	Estas utilidades no suelen estar disponibles



SQL

Cuando el volumen de mis datos no crece o lo hace poco a poco.

Cuando las necesidades de proceso se pueden asumir en un sólo servidor.

Cuando no tenemos picos de uso del sistema por parte de los usuarios más allá de los previstos.



NoSQL

Cuando el volumen de mis datos crece muy rápidamente en momentos puntuales.

Cuando las necesidades de proceso no se pueden preveer.

Cuando tenemos picos de uso del sistema por parte de los usuarios en múltiples ocasiones.



Introducción

MongoDB es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo código abierto.

Básicamente en lugar de guardar los datos en tablas como se hace en las base de datos, se almacenan documentos similares a JSON pero utiliza una especificación denominada [BSON](#).

La primera versión se lanzó en el año 2009.



Instalación MongoDB - Linux

Para Instalar en Linux seguiremos los siguientes pasos o haremos la instalación directa desde su web:

[Descargar MongoDB](#)

1º Instalamos la key

```
$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv  
EA312927
```

2º Instalamos la lista para el repositorio

```
$ echo "deb http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2  
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list
```

Instalación MongoDB - Linux

A continuación actualizaremos los paquetes.

```
$ sudo apt-get update
```

Procedemos a instalar Mongo

```
$ sudo apt-get install -y mongodb-org
```

Instalación MongoDB - Linux

A continuación deberemos arrancar el servicio de MongoDB

```
$ sudo systemctl start mongod  
// Versiones anteriores.  
$ sudo service mongod start
```

Instalación MongoDB - Linux

Para comprobar que el servicio se encuentra levantado simplemente debemos ejecutar lo siguiente

```
$ sudo systemctl status mongod
```

Para habilitar automáticamente el arranque de mongod debemos ejecutar

```
$ sudo systemctl enable mongod
```

Instalación MongoDB - Linux

Otras instrucciones a tener en cuenta:

Para reiniciar el servicio de MongoDB

```
$ sudo systemctl restart mongod
```

Para parar el servicio completamente.

```
$ sudo systemctl stop mongod
```


Instalación MongoDB - Windows

MongoDB Community Server

MongoDB offers both an Enterprise and Community version of its powerful distributed document database. The community version offers the flexible document model along with ad hoc queries, indexing, and real time aggregation to provide powerful ways to access and analyze your data. As a distributed system you get high availability through built-in replication and failover along with horizontal scalability with native sharding.

The MongoDB Enterprise Server gives you all of this and more. Review the Enterprise Server tab to learn what else is available.

Available Downloads

Version

4.4.4 (current) 

Platform

Windows 

Package

msi 



Download

[Copy Link](#)

[Current releases & packages](#)

[Development releases](#)

[Archived releases](#)

[Changelog](#)

[Release Notes](#)

Instalación RoboMongo


Robomongo o Robo 3T es una herramienta multiplataforma con la que podemos administrar gráficamente nuestras bases de datos para MongoDB.

Actualmente nos encontramos en la versión 3T



Instalación RoboMongo

Para instalarnos la aplicación sólo debemos dirigirnos a la web de [RoboMongo](#) y descargarnos la aplicación para el sistema que queremos utilizar, en el directorio /etc en linux y en windows en el directorio por defecto..

 Robo 3T Download Blog Account

Download the latest version of Robo 3T

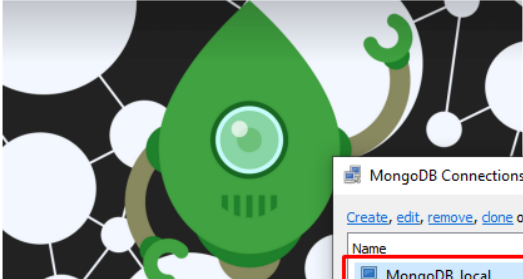
A free 30-day trial of the full access edition of Studio 3T is included with Robo 3T.

[Download Robo 3T + Studio 3T](#) [Download Studio 3T Only](#)

Conexión RoboMongo-MongoDB Server

Welcome

Robo 3T 1.4 now with MongoDB 4.2 support



[Download the new version here](#)

Find these new features and others:

- MongoDB 4.2 support
- mongo shell upgrade from 4.0 to 4.2
- Ability to manually specify visible database

[View the full blog post](#)

Robo 3T feeling a bit underpowered?

[Download Studio 3T](#)

- Double-click to edit documents (instead of right-clicking every time)

MongoDB Connections

Create, edit, remove, clone or reorder connections via drag'n'drop.

Name	Address	Attributes	Auth. Database / User
MongoDB_local	localhost:27017		

Connect Cancel

Conexión RoboMongo-MongoDB Server

Robo 3T - 1.4

File View Options Window Help

MongoDB_local (6)

- System
- Biblioteca
 - Collections (2)
 - Libro
 - personas
 - Functions
 - Users
- config
- mystore
- webstore

Terminal de interacción con BD

```
db.getCollection('Libro').find({})
```

Libro 0.005 sec.

```
/* 1 */
{
  "_id" : 1,
  "isbn" : 123456789,
  "Titulo" : "El libro negro del programador",
  "edicion" : 2017,
  "autor" : "Rafael Gomez Blanes"
}

/* 2 */
{
  "_id" : 2,
  "isbn" : 223456789,
  "Titulo" : "Código Limpio: Manual de estilo para el desarrollo ágil de software",
  "edicion" : 2012,
  "autor" : "Robert C. Martin"
}
```

Árbol de sistema

Consola de resultado.

Características de MongoDB

Características de MongoDB

En MongoDB la información se almacena en **Documentos** agrupados en colecciones que pertenecen a las base de datos, estos documentos de una colección no necesariamente deben mantener la misma estructura.

Características de MongoDB

Por ejemplo un documento podría ser este

```
{  
  "_id" : ObjectId("5ea6c1779eb4a9273c7bf09a"),  
  "email" : "d.pinilla@geekshubs.com",  
  "name":"David",  
  "lastName":"Pinilla",  
  "__v" : 0  
}
```


Características de MongoDB

Al final podemos ver que es una representación de datos como si se tratara de un objeto JSON.

Características de MongoDB

MongoDB utiliza [BSON](#) que es una extensión de Json y nos proporciona los siguientes tipos de datos, además de los habituales.

- **ObjectId:** Es un identificador de documentos que consta de 12 Bytes y que es único.
- **Numéricos:** Son enteros de 4 u 8 bytes.
- **String:** Utiliza UTF-8. Cualquier valor encerrado entre comillas es un String
- **Timestamp:** Consta de 64 bits, y guarda en este mismo formato una fecha.
- **UTC Datetime:** Date en mongo consta de 64 bits y almacena la fecha
- **BinData:** Es un array de datos binarios que puede se utilizado para guardar el contenido de ficheros.

Mongo Shell

Creando y usando una nueva DB

Para crear nuestra primera BD

```
> use prueba
```

Para comprobar en que BD nos encontramos

```
> db
```

Usando MongoDB

Para mostrar todas las base de datos usamos

```
> show dbs
```

Para eliminar una Base de datos nos tenemos que situar sobre ella y ejecutar

```
> db.dropDatabase()
```

Usando MongoDB

Podemos crear una colección de documentos mediante la instrucción.

```
> db.createCollection("Coleccion1")
```

Podemos pasarle una series de opciones a la creación de la colección

```
> db.createCollection([nombre colección],[opciones]);
```

[createCollection DOCS](#)

Usando MongoDB

Las principales opciones son:

- capped -> True o false si queremos sobrescribir la colección en el caso de que ya exista.
- autoIndexId -> Crea un índice automáticamente.
- size -> Tamaño de la colección en bytes.
- max -> Número máximo de documentos.

Usando MongoDB

Podemos eliminar una colección mediante.

```
> db.products.drop()
```


Usando MongoDB

Insertando Documentos

```
>db.products.insert(documento)
```

Una de las posibilidades de las colecciones es que podemos insertar documentos de diferentes tipos, así como tamaños.

Usando MongoDB

Si queremos buscar

```
>db.products.find({criterio de búsqueda})
```

Sí solo queremos buscar uno

```
>db.products.findOne({criterio de búsqueda})
```

Usando MongoDB

Actualizando documentos.

```
>db.products.update({query}, {actualizaciones})
```

Podemos guardar un documento directamente con `.insertOne`, pero en caso de querer reemplazar el documento debemos utilizar `replaceOne`, indicarle un `_id` a la query se convertirá en un `update` del documento ya existente sobrescribiéndolo.

Anteriormente se hacía con el metodo `.save()` (ahora deprecated).

```
>db.products.replaceOne({_id:
ObjectId("5ba0f3b9747a1aa0fedb0c19")}, product:'Samsung TV',
price:399})
```

Usando MongoDB

Si queremos eliminar los documentos.

```
>db.products.remove({criterio de busqueda})
```

Si sólo queremos eliminar uno

```
>db.products.remove({criterio de búsqueda},1)
```

Usando MongoDB

Comandos MongoDB para el find(), los criterios de búsqueda son equivalentes a las cláusulas where de SQL

```
>db.products.find({"nombre":"TV"}) ->where nombre = "TV"  
>db.products.find({"price":{"$lt:400}}) -> where price < 400  
>db.products.find({"price":{"$lte:400}})-> where price<= 400  
>db.products.find({"price":{"$gt:410}}) -> where price> 410  
>db.products.find({"price":{"$gte:410}})-> where price>=410  
>db.products.find({"price":{"$ne:500}}) -> where price!=500
```

Usando MongoDB

And o Or para el find()

```
>db.products.find({  
  $and:  
    [  
      {price :399},{name:"Samsung TV"}  
    ]  
  })
```

O simplemente:

```
>db.products.find({  
  price :399,  
  name:"Samsung TV"  
})
```

Usando MongoDB

And o Or para el find()

```
>db.products.find(  
  {  
    $or:  
      [  
        {price :400},{price:399}  
      ]  
  }  
)
```

Usando MongoDB

Podemos establecer el límite de resultados de un búsqueda.

```
db.products.find().limit(número)
```

También podemos establecer a partir de que registró queremos que nos devuelva.

```
db.products.find().limit(número).skip(número)
```


Usando MongoDB

Otra opción es ordenar los registros ASC y DESC

```
db.products.find().sort({price:1}) //Ascendente  
db.products.find().sort({price:-1}) //Descendente
```

MongoDB

Exportación simple a CSV

```
mongoexport --collection events --db reporting --out events.json
```

Antes de empezar a probar exportaciones más complejas con parámetros, **empezaremos con la exportación más básica disponible**. Esta es la configuración mínima que tendrá un export **dando por hecho que se realiza sobre la misma máquina que contiene MongoDB**. Los datos necesarios son:

- Base de datos
- Colección
- Usuario con permiso de exportación (al menos de lectura).
- Contraseña del usuario
- Base de datos donde está creado el usuario (authenticationDatabase)

```
mongoexport --db base_de_datos_a_exportar --collection coleccion_a_exportar --authenticationDatabase  
admin -u usuario_exportacion -p Pa$$ --out /ruta/export/exportacion_simple.csv
```

MongoDB

Robo 3T - 1.4

File View Options Window Help

New release available. Find out [what's new in Robo 3T](#) - [Download here](#).

▼ MongoDB_local (6)

- > System
- ▼ Biblioteca
 - ▼ Collections (2)
 - > Libro
 - > personas
 - > Functions
 - > Users
- > config
- > mystore
- > webstore

Welcome x * New Shell x * db.getCollection('Libro').find({... x

MongoDB_local localhost:27017 Biblioteca

```
db.getCollection('Libro').find({}).toArray()
```

0.002 sec.

Key	Value
▼ (1)	[4 elements]
> [0]	{ 5 fields }
> [1]	{ 5 fields }
> [2]	{ 5 fields }
> [3]	{ 5 fields }

Expand Recursively Alt+ Right

Collapse Recursively Alt+ Left

View Document...

Copy JSON