

# QUE ES MONGODB?



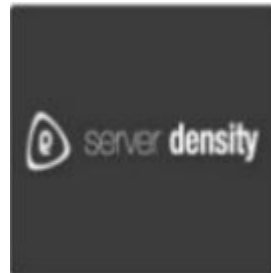
**Otra tecnología confiable de Google**

# CARACTERÍSTICAS PRINCIPALES



- Da respuesta a la necesidad de almacenamiento de todo tipo de datos: estructurados y no estructurados.
- Velocidad y sencillez a la hora de realizar consultas.
- Posee un lenguaje basado íntegramente en JS.
- Gran escalabilidad, seguridad y rápido crecimiento.
- Adaptación a las necesidades actuales de millones de usuario con miles de peticiones por segundo.
- Está orientada a documentos. Lo que quiere decir que en un único documento puede almacenar toda la información necesaria que define un producto, cliente, etc.

# QUIENES LO USAN EN LA ACTUALIDAD?



# COMO SE INSTALA Y CÓMO INICIARLO?



<https://docs.mongodb.com/manual/installation/>

## MongoDB Community Edition

### Install on Linux

Install MongoDB Community Edition and required dependencies on Linux.

### Install on OS X

Install MongoDB Community Edition on OS X systems from Homebrew packages or from MongoDB archives.

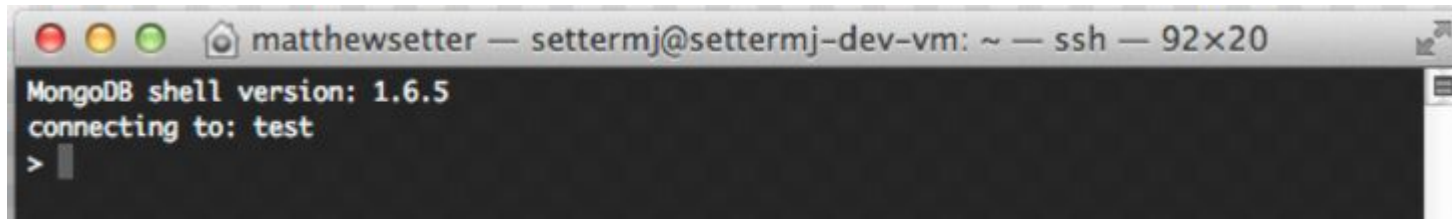
### Install on Windows

Install MongoDB Community Edition on Windows systems and optionally start MongoDB as a Windows service.

```
$sudo service mongod start (Puerto default: 27017)
```

# CÓMO CREAR Y MOSTRAR UNA BASE DE DATOS?

Ejecutar el comando **\$mongo** para abrir la interfaz:



```
matthewsetter — settermj@settermj-dev-vm: ~ — ssh — 92x20
MongoDB shell version: 1.6.5
connecting to: test
> |
```

A continuación creamos una base, simplemente usando el comando USE: **>use chechBaseMongo.**

Una vez creada, esta no aparecerá en el listado de bases (**>show databases**), si no se le crea al menos algún documento..

# COMO CREAR DOCUMENTOS? - INSERT



Los documentos se rigen por la nomenclatura de JSON  
(clave:valor)

```
> db.chechCollectionMongo.insert({curso: 'Mongo', dia: 'Viernes'});
```

O algo más elaborado, incorporando otros documento y arrays, dependiendo siempre del diseño deseado.

Un buen diseño de los documentos debe incorporar al menos los siguientes ítems: **Buena escalabilidad, relaciones y velocidad de acceso** a los datos

Nota: **chechCollectionMongo** es una colección de documentos, que se creará al insertar al menos 1

# CÓMO BUSCAR DOCUMENTOS? - FIND



```
> db.checkCollectionMongo.find({condicion},{Que mostrar})
```

Hay diferentes clases de condiciones para armar toda clase de consultas y obtener los resultados deseados:

- >..find({nombre:'Dami', edad: 22},{nombre:0}) → Simple (Solo edad)
- >..find({nombre:'Dami', edad:{&ne: 22}},{edad:0}) → Not Equal (Solo nombre)
- >..find({nombre:'Dami', edad:{&lt; 25}}) → Less Than
- >..find({nombre:'Dami', edad:{&gt; 32}}) → Greater Than
- >..find({nombre:'Dami', \$or: [{c1},{c2}]}) → Condición con OR
  
- >..find({nombre:'Dami', edad: 22}).pretty() → Se vea formateado JSON
- >..findOne({nombre:'Dami', edad: 22}) → Imprime el 1°

# COMO ACTUALIZAR DOCUMENTOS? - UPDATE



```
>db.checkCollectionMongo.update(<filtros>, <seteo>, <opciones>)
```

Existen diferentes maneras para armar toda clase de actualizaciones sobre un set de datos:

- > cambio = **..findOne**({condición}); → Asignar la búsqueda al cursor  
> cambio.elementoDoc = "Cambiar a mano"; → Cambiar el valor  
> **..save**(cambio); → Guardar el nuevo dato

---

- > cambio = **..findOne**({condición}); → Asignar la búsqueda al cursor  
> cambio.nuevoElemento = "Nuevo"; → Agregar elemento al documento  
> **..save**(cambio); → Guardar el nuevo dato

---

- > **..update**({elemento: "filtro"}, documentoACambiar);  
> **..update**({elemento: "filtro"},{\$set:{elemento="Cambio"}}); → 1°  
> **..update**({elemento: "filtro"},{\$set:{elemento="Cambio"}},{**multi**:true});  
> **..update**({elemento: "filtro"},{\$set:{elementoNuevo="Nuevo"}})



# CÓMO ELIMINAR UN DOCUMENTO?



```
> db.checkCollectionMongo.remove({condicion})
```

A diferencia del Update, cada condición que se ponga, el comando elimina todos los documentos que cumplan con la condición:

- > **..remove({})**; → Elimina todos los documentos de la colección
- > **..remove({elemento: "Filtro"})**; → Elimina los que cumplan el filtro

# CÓMO ELIMINAR UNA COLECCIÓN Y LA DB?



```
> db.checkCollectionMongo.drop();
```

```
> db.dropDatabases();
```

# COMO INDEXAR Y ACELERAR LA BÚSQUEDA?



```
> db.coleccionGrande.createIndex({elemento : 1},{elemento2: 1})
```

A continuación de detallara la primera aproximación en la creación de un campo indexado para mejorar el tiempo de búsqueda (en milisegundos):

```
> db.coleccionGrande.find(_id : ObjectId..).explain("executionStats");
```

Para la prueba se crearía un índice con otro campo, que no sea el `_id`, para luego comprobar la mejora del tiempo de búsqueda.



mongoDB

