



MONGODB

por Christian Dario Nieves



mongoDB®



MONGODB

- Base de datos NoSQL orientada a documentos.
- Trabaja con formato JSON/BSON
- Puede almacenar grandes cantidades de información
- No tiene mucho control o chequeo sobre tipos de datos



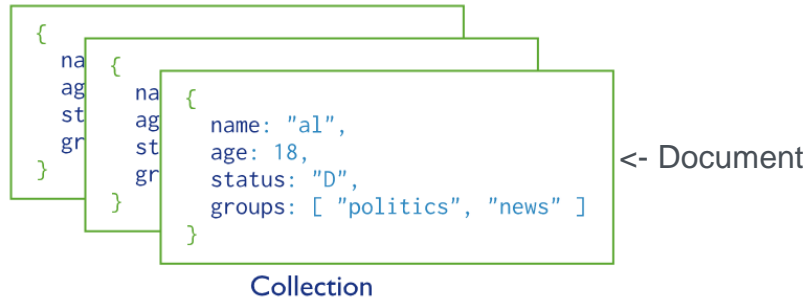
“ *Se suele usar por ejemplo para Logs, Blogs, Web-Analytics o cuando se quiere persistir información cuya estructura no está bien definida* ”



Cuadro de comparación

RDBMS	Mongo
Database Instance	MongoDB Instance
Database / Schema	Database
Table	Collection
Row	Document
Row Primary Key	_id

Documents < Collections < Database

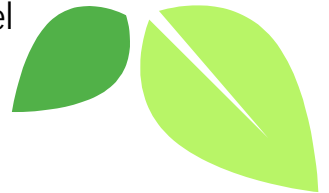


Crear/Elegir una DB - *use dbEjemplo*

Si tu DB no existe, MongoDB recién la crea cuando guardas datos en ella. Los datos en sí se manejan como documentos y los documentos pertenecen a una colección.

Crear/Elegir una Collection - *db.miColeccion.metodo*

Si la colección no existe, MongoDB recién la crea cuando guardas datos en ella. También existe el comando *db.createCollection()* que la crea de manera explícita y que permite asignarle más atributos de seguridad como tamaño máximo de documento, o reglas lógicas/control sobre el documento





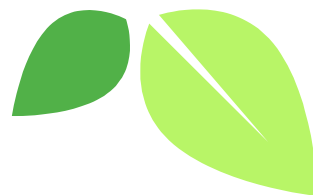
Documentos

- MongoDB acepta documentos JSON de manera básica que transforma y guarda en formato BSON (Binary Json).
- Pero al guardarlos en formato BSON nos ofrece, la ventaja de nuevos tipos de datos a usar además de los ya ofrecidos por un documento JSON.

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

← field: value
← field: value
← field: value
← field: value

```
var mydoc = {  
  _id: ObjectId("5099803df3f4948bd2f98391"),  
  name: { first: "Alan", last: "Turing" },  
  birth: new Date('Jun 23, 1912'),  
  death: new Date('Jun 07, 1954'),  
  contribs: [ "Turing machine", "Turing test", "Turingery" ],  
  views : NumberLong(1250000)  
}
```





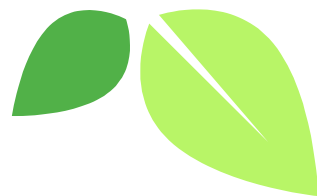
_id: ObjectId()

El nombre de campo *_id* está reservado para uso como Primary Key por ende su valor debe ser único en la colección. Si un documento insertado omite este campo el Driver de MongoDB lo autogenera con la función ObjectId().

El campo *_id* siempre debe estar primero en el documento y si no es así el propio servidor de MongoDB lo mueve/crea.

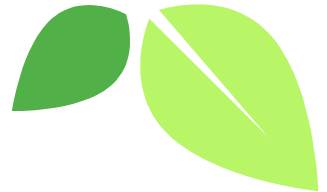
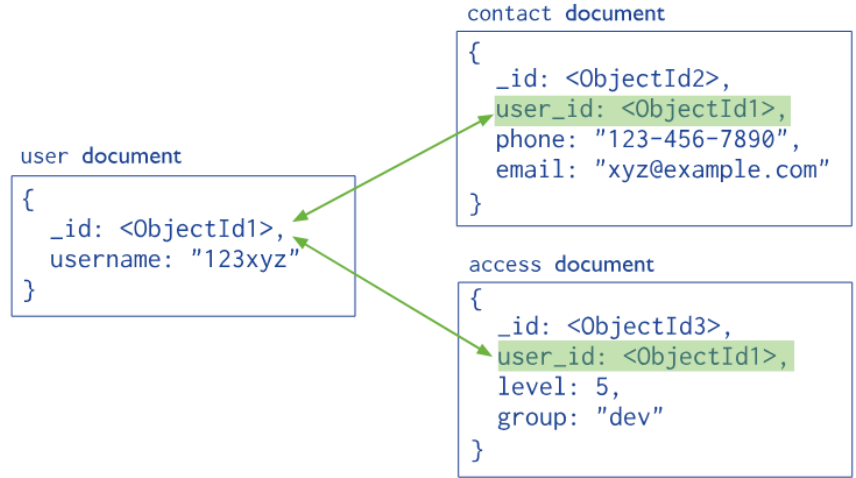
El campo *_id* puede contener un ObjectId(), un identificador unico y natural, un número autoincremental entre otros.

Pero que es el *ObjectId()* ? Son IDs de 12 bytes donde:
4 bytes representan un Timestamp de cuando el objeto fue creado
5 bytes que son un valor random
3 bytes como un contador que empieza en un valor random





Documentos embebidos vs referenciados



Operaciones CRUD

(Create, Read, Update, Delete)

Crear documento

db.miColeccion.insertOne({})
db.miColeccion.insertMany([{}])

```
db.users.insertOne( ← collection
{
  name: "sue", ← field: value
  age: 26, ← field: value
  status: "pending" ← field: value
}
) } document
```

Recuperar documento

db.miColeccion.find({})

```
db.users.find( ← collection
  { age: { $gt: 18 } }, ← query criteria
  { name: 1, address: 1 } ← projection
).limit(5) ← cursor modifier
```

Actualizar documento

db.miColeccion.updateOne({})
db.miColeccion.updateMany({})
db.miColeccion.replaceOne({})

```
db.users.updateMany( ← collection
  { age: { $lt: 18 } }, ← update filter
  { $set: { status: "reject" } } ← update action
)
```

Borrar documento

db.miColeccion.deleteOne({})
db.miColeccion.deleteMany({})

```
db.users.deleteMany( ← collection
  { status: "reject" } ← delete filter
)
```



“ *Para mas informacion*

MongoDB:

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

MongoDB Driver .NET:

<http://mongodb.github.io/mongo-csharp-driver/2.7/>