

@MongoDB + @javascript = @mongoosejs

Jesse Javier Cogollo Alvarez

Developer by passion

twitter: @jessecogollo

March 18, 2015

Contenido

MongoDB

Javascript

Mongoose

Que es @MongoDB

'MongoDB (from "humongous") is an open-source document database, and the leading NoSQL database. Written in C++.'

<https://www.mongodb.org/>

'MongoDB was not designed in a lab. We built MongoDB from our own experiences building large-scale, high availability, robust systems...' [Eliot Horowitz, CTO and Co-Founder](#)

Que es @MongoDB

'MongoDB (from "humongous") is an open-source document database, and the leading NoSQL database. Written in C++.'

<https://www.mongodb.org/>

'MongoDB was not designed in a lab. We built MongoDB from our own experiences building large-scale,high availability, robust systems...' [Eliot Horowitz, CTO and Co-Founder](#)

NOSQL

En informática, NoSQL (a veces llamado 'no sólo SQL') es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico del sistema de gestión de bases de datos relacionales (RDBMS) en aspectos importantes, el más destacado que no usan SQL como el principal lenguaje de consultas.

<http://es.wikipedia.org/wiki/NoSQL/>

NOSQL

Las características comunes de las bases de datos NoSQL son:

- No utilizan el modelo relacional.
- Corren bien en clusters.
- Open-source.
- sin esquemas.
- El resultado mas importante del aumento de las bases de datos NoSQL es la **Persistencia Polígloa**.

[http:](http://martinfowler.com/articles/nosqlKeyPoints.html)

[//martinfowler.com/articles/nosqlKeyPoints.html](http://martinfowler.com/articles/nosqlKeyPoints.html)

NOSQL

Las características comunes de las bases de datos NoSQL son:

- No utilizan el modelo relacional.
- Corren bien en clusters.
- Open-source.
- sin esquemas.
- El resultado mas importante del aumento de las bases de datos NoSQL es la **Persistencia Polígloa**.

<http://martinfowler.com/articles/nosqlKeyPoints.html>

NOSQL

Las características comunes de las bases de datos NoSQL son:

- No utilizan el modelo relacional.
- Corren bien en clusters.
- Open-source.
- sin esquemas.
- El resultado mas importante del aumento de las bases de datos NoSQL es la **Persistencia Polígloa**.

<http://martinfowler.com/articles/nosqlKeyPoints.html>

NOSQL

Las características comunes de las bases de datos NoSQL son:

- No utilizan el modelo relacional.
- Corren bien en clusters.
- Open-source.
- sin esquemas.
- El resultado mas importante del aumento de las bases de datos NoSQL es la **Persistencia Polígloa**.

<http://martinfowler.com/articles/nosqlKeyPoints.html>

NOSQL

Las características comunes de las bases de datos NoSQL son:

- No utilizan el modelo relacional.
- Corren bien en clusters.
- Open-source.
- sin esquemas.
- El resultado mas importante del aumento de las bases de datos NoSQL es la **Persistencia Polígloa**.

[http:](http://martinfowler.com/articles/nosqlKeyPoints.html)

[//martinfowler.com/articles/nosqlKeyPoints.html](http://martinfowler.com/articles/nosqlKeyPoints.html)

Persistencia políglota



Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
4. Auto Sharding
 - Partner with MongoDB.
 - Multiples drivers.
5. Querying
6. Map Reduce
7. GridFS
8. **Other more...**

Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
4. Auto Sharding
 - Partner with MongoDB.
5. Querying
 - Multiples drivers.
6. Map Reduce
7. GridFS
8. **Other more...**

Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
4. Auto Sharding
 - Partner with MongoDB.
5. Querying
 - Multiples drivers.
6. Map Reduce
7. GridFS
8. **Other more...**

Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
 - Partner with MongoDB.
 - Multiples drivers.
4. Auto Sharding
5. Querying
6. Map Reduce
7. GridFS
8. **Other more...**

Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
4. Auto Sharding
 - Partner with MongoDB.
 - Multiples drivers.
5. Querying
6. Map Reduce
7. GridFS
8. **Other more...**

Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
4. Auto Sharding
 - Partner with MongoDB.
 - Multiples drivers.
6. Map Reduce
7. GridFS
8. **Other more...**

Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
4. Auto Sharding
 - Partner with MongoDB.
5. Querying
 - Multiples drivers.
6. Map Reduce
7. GridFS
8. **Other more...**

Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
4. Auto Sharding
 - Partner with MongoDB.
5. Querying
 - Multiples drivers.
6. Map Reduce
7. GridFS
8. Other more...

Caracteristicas

1. Document-Oriented Storage
2. Full Index Support
3. Replication
 - MMS.
4. Auto Sharding
 - Partner with MongoDB.
 - Multiples drivers.
5. Querying
6. Map Reduce
7. GridFS
8. **Other more...**

Insert Find Update Remove (CRUD)

IFUR

```
db.meetups.insert({ "name": "mongoosejs", "place": "Ruta N" })
```

IFUR

```
db.meetups.find({ "name": "mongoosejs" })
```

IFUR

```
db.meetups.update({ "name": "mongoosejs" },  
{$set: { "description": "Ruta N, piso 0." }})
```

IFUR

```
db.meetups.remove({ "name": "mongoosejs" })
```

Insert Find Update Remove (CRUD)

IFUR

```
db.meetups.insert({ "name": "mongoosejs", "place": "Ruta N" })
```

IFUR

```
db.meetups.find({ "name": "mongoosejs" })
```

IFUR

```
db.meetups.update({ "name": "mongoosejs" },  
{$set: { "description": "Ruta N, piso 0." }})
```

IFUR

```
db.meetups.remove({ "name": "mongoosejs" })
```

Insert Find Update Remove (CRUD)

IFUR

```
db.meetups.insert({ "name": "mongoosejs", "place": "Ruta N" })
```

IFUR

```
db.meetups.find({ "name": "mongoosejs" })
```

IFUR

```
db.meetups.update({ "name": "mongoosejs" },  
  { $set: { "description": "Ruta N, piso 0." } })
```

IFUR

```
db.meetups.remove({ "name": "mongoosejs" })
```

Insert Find Update Remove (CRUD)

IFUR

```
db.meetups.insert({ "name": "mongoosejs", "place": "Ruta N" })
```

IFUR

```
db.meetups.find({ "name": "mongoosejs" })
```

IFUR

```
db.meetups.update({ "name": "mongoosejs" },  
{$set: { "description": "Ruta N, piso 0." }})
```

IFUR

```
db.meetups.remove({ "name": "mongoosejs" })
```


Insert Find Update Remove (CRUD)

IFUR

```
db.meetups.insert({ "name": "mongoosejs", "place": "Ruta N" })
```

IFUR

```
db.meetups.find({ "name": "mongoosejs" })
```

IFUR

```
db.meetups.update({ "name": "mongoosejs" },  
{$set: { "description": "Ruta N, piso 0." }})
```

IFUR

```
db.meetups.remove({ "name": "mongoosejs" })
```

NodeJS - IOJS

NodeJS es una plataforma de javascript construida sobre el "motor" V8 de Chrome. <https://nodejs.org//>

IOJS es un fork de NodeJS. Implementando ES6 y desarrollado bajo un modelo de gobierno abierto. <https://iojs.org//>

NodeJS - IOJS

NodeJS es una plataforma de javascript construida sobre el "motor" V8 de Chrome. <https://nodejs.org//>

IOJS es un fork de NodeJS. Implementando ES6 y desarrollado bajo un modelo de gobierno abierto. <https://iojs.org//>

Mongoose

Es una herramienta para el modelado de objetos MongoDB.
Diseñado para trabajar en un ambiente asincronico.

instalación (En el directorio del proyecto.)

```
# npm install mongoose --save
```

Mongoose

Es una herramienta para el modelado de objetos MongoDB.
Diseñado para trabajar en un ambiente asincronico.

instalación (En el directorio del proyecto.)

```
# npm install mongoose --save
```

Entendiendo mongoosejs

Esquema

Un esquema es mapeado como una **colección** en MongoDB y define la forma de los documentos con esa colección.

Schema

```
1 var mongoose = require('mongoose');
2 var schema = mongoose.schema;
3 var userSchema = new Schema({
4   firstName:String,
5   lastName:String,
6   telefones:{
7     primary:String,
8     secondary:String
9   },
10  hobbies:Array
11 });
```

Tipos: Number, Date, Buffer, Boolean, mixed and ObjectId.

Entendiendo mongoosejs

model

Un modelo es un constructor compilado de nuestra definición de esquema. y representan los **documentos** que pueden ser guardados y recuperados de la base de datos.

Model

```
1 var User = mongoose.model('User',userSchema);
```

Ya tenemos a User listo para Insertar, encontrar, actualizar y eliminar. Pero además, le podemos crear nuestras propias acciones al schema.

```
1 userSchema.methods.findSimilarLastNames = function (cb
  ) {
2   return this.model('User').find({ lastName: this.
      lastName }, cb);
3 }
```

Metodos estaticos

Statics

Agregar metodos estaticos a un modelo es simple.

```
1 userSchema.statics.findByName = function (name, cb) {  
2   this.find({ firstName: new RegExp(name, 'i') }, cb);  
3 }
```

Indexes

Los índices se pueden definir a nivel de esquema o a nivel de campo.

Campo

```
1 firstName:{type:String, index:true}
```

Esquema

```
1 userSchema.index({firstName:1});
```

Virtuales

Los metodos virtuales no se pueden persistir. Estos pueden ser GET o SET.

GET: útiles para para formatear o combinar campos.

SET: útiles para descomponer un valor en múltiples valores.

Virtuals GET - SET

```
1 userSchema.virtual('fullName').get(function () {  
2   return this.firstName + ' ' + this.lastName;  
3 });
```

```
1 userSchema.virtual('fullName').set(function (fullName)  
  {  
2   var split = fullName.split(' ');  
3   this.firstName = split[0];  
4   this.lastName = split[1];  
5 });
```

Options

Son un conjunto de opciones que son configurables.

- **autoIndex**
- capped
- collection
- id
- _id
- read
- mas...

Options

Son un conjunto de opciones que son configurables.

- autoIndex
- capped
- collection
- id
- _id
- read
- mas...

Options

Son un conjunto de opciones que son configurables.

- autoIndex
- capped
- collection
- id
- _id
- read
- mas...

Options

Son un conjunto de opciones que son configurables.

- autoIndex
- capped
- collection
- id
- _id
- read
- mas...

Options

Son un conjunto de opciones que son configurables.

- autoIndex
- capped
- collection
- id
- _id
- read
- mas...

Options

Son un conjunto de opciones que son configurables.

- autoIndex
- capped
- collection
- id
- _id
- read
- mas...

Options

Son un conjunto de opciones que son configurables.

- autoIndex
- capped
- collection
- id
- _id
- read
- mas...

Mas...

- **plugins**
- validadores (validate)
- lanzadores (pre, post)

Mas...

- plugins
- validadores (validate)
- lanzadores (pre, post)

Mas...

- plugins
- validadores (validate)
- lanzadores (pre, post)

DEMO !!!

=)

Comunidad

1. Meetup: medellinjs - mongodbmedellin
2. Twitter: @medellinjs - @mongodbmedellin - @avanet
3. Facebook: /mongodbmedellin /avanet
4. gitter (chat): <https://gitter.im/coljs/medellinjs>
5. gitter (chat): <https://gitter.im/MongoDBMedellin/Meetup>

Comunidad

1. Meetup: medellinjs - mongodbmedellin
2. Twitter: @medellinjs - @mongodbmedellin - @avanet
3. Facebook: /mongodbmedellin /avanet
4. gitter (chat): <https://gitter.im/coljs/medellinjs>
5. gitter (chat): <https://gitter.im/MongoDBMedellin/Meetup>

Comunidad

1. Meetup: medellinjs - mongodbmedellin
2. Twitter: @medellinjs - @mongodbmedellin - @avanet
3. Facebook: /mongodbmedellin /avanet
4. gitter (chat): <https://gitter.im/coljs/medellinjs>
5. gitter (chat): <https://gitter.im/MongoDBMedellin/Meetup>

Comunidad

1. Meetup: medellinjs - mongodbmedellin
2. Twitter: @medellinjs - @mongodbmedellin - @avanet
3. Facebook: /mongodbmedellin /avanet
4. gitter (chat): <https://gitter.im/coljs/medellinjs>
5. gitter (chat): <https://gitter.im/MongoDBMedellin/Meetup>

Comunidad

1. Meetup: medellinjs - mongodbmedellin
2. Twitter: @medellinjs - @mongodbmedellin - @avanet
3. Facebook: /mongodbmedellin /avanet
4. gitter (chat): <https://gitter.im/coljs/medellinjs>
5. gitter (chat): <https://gitter.im/MongoDBMedellin/Meetup>

Donde aprender



javascript (nodeJS)

<http://nodeschool.io/>



MongoDB

<https://university.mongodb.com/>

Preguntas



Gracias !!! =)