



# Cómo conectar MongoDB en Node.js

Ing. Luis Guillermo Molero Suárez

MongoDB es una de las bases de datos no relacionales más utilizadas en la actualidad, ahora veremos cómo conectar un proyecto en Node.js con este motor de base de datos.

Toda aplicación necesita un lugar donde almacenar sus datos, para esto existen múltiples opciones, entre las más populares están las bases de datos relacionales como Mysql, Postgres, Oracle, etc, por otro lado, también hay otro grupo de base de datos no relaciones o NoSQL por lo cual, hemos escogido MongoDB, y aprenderemos a como configurar un proyecto en Node.js para que pueda utilizar esta base de datos.

## Mongoose

Mongoose es un módulo de terceros que podemos encontrarlo en el repositorio de npm y que nos va a permitir realizar la conexión entre Node.js y MongoDB.

## Configuración

La configuración es relativamente sencilla, al inicio puede que sea un poco confuso y esto depende de acuerdo a como tengan organizado su proyecto en Node, para que me puedan entender voy a dejar la estructura que siempre utilizo para mis proyectos y de acuerdo a eso pueden guiarse.



Ahora empezaremos con el contenido de los archivos.

## config/mongoose/index.js

En este archivo vamos a poner la configuración general de mongoose.

```
'use strict';
const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost/auth');

module.exports = mongoose;
```

Es relativamente fácil solo debemos utilizar el método connect() y conectarlo con una base de datos, en este caso tiene por nombre auth, no se preocupen por crear la base de datos, esto se hace automáticamente.

## Esquemas

Los esquemas son la estructura de una colección, con estructura me refiero a los atributos que tenga cada colección que vayamos a crear. Por ejemplo, definamos un esquema de usuarios solo con dos atributos.

## apps/users/schemas.js

En este archivo suelo poner los esquemas que corresponden con esta aplicación, ahora veamos el código.

```
'use strict';
const mongoose = require('../../config/mongoose'),
      Schema = mongoose.Schema;

const schemas = {

  userSchema: new Schema({
    username: {type: String},
    password: {type: String},
  })

};

module.exports = schemas;
```



Con esto ya tenemos un esquema de usuarios que tienen solo 2 atributos, un username y un password, ahora debemos crear un modelo a partir de este esquema.

## Modelos

Un modelo es la abstracción que vamos a utilizar para realizar operación en la BD, un modelo está definido por un esquema, en este ejemplo vamos a crear el modelos de usuarios.

```
'use strict';
const mongoose = require('../../config/mongoose'),
    userSchema = require('./schemas').userSchema;

const models = {

User: mongoose.model('user', userSchema)

};

module.exports = models;
```

Siempre que queramos realizar operaciones (insertar, eliminar, actualizar, etc) a MongoDB, debemos hacerlo a travez del modelo User.

Para terminar el ejemplo usaremos el controlador para traer a todos los usuarios (si es que los tienes) de la colección de usuarios.



# Controlador

En el controlador es donde podemos realizar las operaciones que necesitemos a la base de datos.

## apps/users/controllers

```
'use strict';
const express = require('express'),
      router = express.Router(),
      User = require('./models').User;

router.route('/')

  .get(function (req, res) {
    User.find()
      .then(function (users) {
        res.locals.users = users;
        return res.render('users/users.html');
      });
  });
```

Primero requerimos el modelo que hemos creado, y con este ya podemos realizar consultas.