Unidad 3: NODE.js – MongoDB

1. UNIDAD 3: Introducción

Bienvenido a Interactuando con Bases de Datos

En esta Unidad conocerás MongoDB, una base de datos NoSQL orientada a documentos. Ésta utiliza un formato llamado BSON, que es una representación en binario de un documento JSON. Es un motor de bases de datos multiplataforma de esquema libre. Luego conocer cuáles son los principales servicios que ofrece MongoDB junto con su instalación, verás cómo interactúa con node.js y cómo establecer una conexión entre ambos utilizando el driver nativo.

Puntos de aprendizaje

Unidad 3: NODE.js – MongoDb  
¡Aquí comienza la Unidad 3: NODE.js – MongoDb!

* Los objetivos del Programa que corresponden a esta Unidad son:
  + Crear ambientes de programación y a gestionar recursos.
* Los objetivos del Curso 7 que corresponden a esta Unidad son:  
  + Entender, diseñar y construir bases de datos NoSQL.
  + Administrar información con bases de datos NoSQL.

1. Lección 1: ¿Qué es MongoDB? Descarga e instalación
   1. Lectura ¿Qué es mongoDB?

Qué es MongoDB?

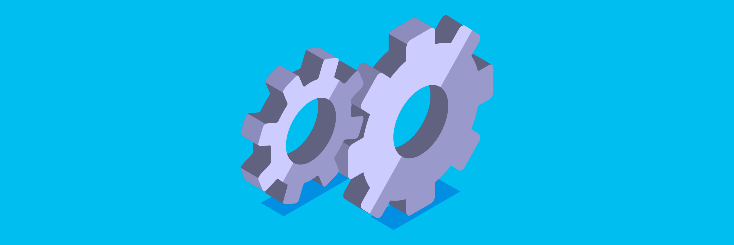


MongoDB es una base de datos NoSQL orientada a documentos. Esta utiliza un formato llamado BSON, que es una representación en binario de un documento JSON.  
Mongo es un motor de bases de datos multiplataforma de esquema libre, es decir, que cada registro puede contener una estructura diferente y esto no generará ningún problema a la hora de hacer la inserción.  
Esta base de datos posee una interfaz de línea de comandos (REPL) con la que podemos interactuar directamente con el servidor, también nos permite instalarla de forma distribuida y balancear el peso de la base de datos en diferentes servidores.



* 1. Lectura: Como funciona MongoDb – Instalación

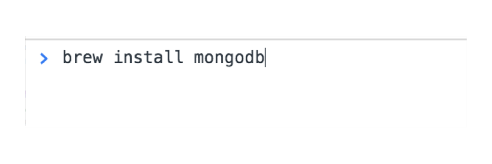
MongoDB: qué es y cómo instalarlo



En un par de ocasiones ya hemos mencionado que mongo es una base de datos orientada a documentos de tipo JSON con un esquema dinámico y totalmente flexible.  
MongoDB está escrita en C++ y contiene una capa bastante especial en la que convierte los datos en un formato bastante ligero conocido como BSON, este formato es el que le brinda el alto nivel de escalabilidad, pero no solo eso, además existen otras características que son importantes de Mongo:  
  
- Soporte de índices para mejorar el rendimiento en las consultas que interactúan con tablas de muchos registros.  
- Replicación y alta disponibilidad.  
- Consultas basadas en la sintaxis de los objetos JSON. - Auto-Sharding, permitiendo escalar horizontalmente la base de datos.  
  
Recordemos los tipos de datos que puede contener un documento JSON:  
  
- String  
- Number  
- Boolean (true o false)  
- Object  
- Array  
- Null

Instalación:

MacOS:

  
NOTA: Cualquier instalación de MongoDB, independiente del sistema operativo, necesita crear una carpeta en la raíz del sistema llamada /data/db. Este directorio es donde se almacena toda la información de la base de datos.  
  
Crear directorio:



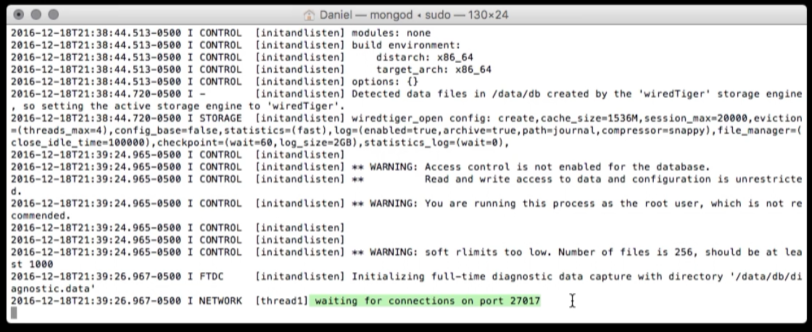
Los que están usando Linux o Windows les recomiendo ingresar a la página oficial de MongoDB y realizar los pasos de instalación.

* 1. Servicios (Mongod, Mongo, MongoExport, MongoImport)

Servicios o demonios: 13 servicios

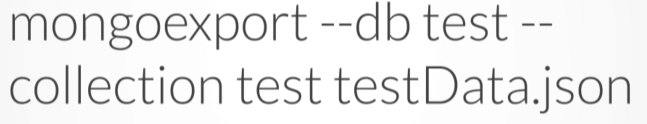
Más importantes:

* Mongod – Encargado de inicar el servidor mongodb y escuchar las conexiones que se hagan a la DB. Puerto por defecto 27017. Este es el servicio más importante, debe correr siempre que se trabaja con la DB



* Mongoexport – sirve para crear archivos en formato JSON o CSV -> Para Backups

Con el comando:



* Mongoimport – todo lo contrario a mongoExport y lo que hace es importarlo a la base de datos.

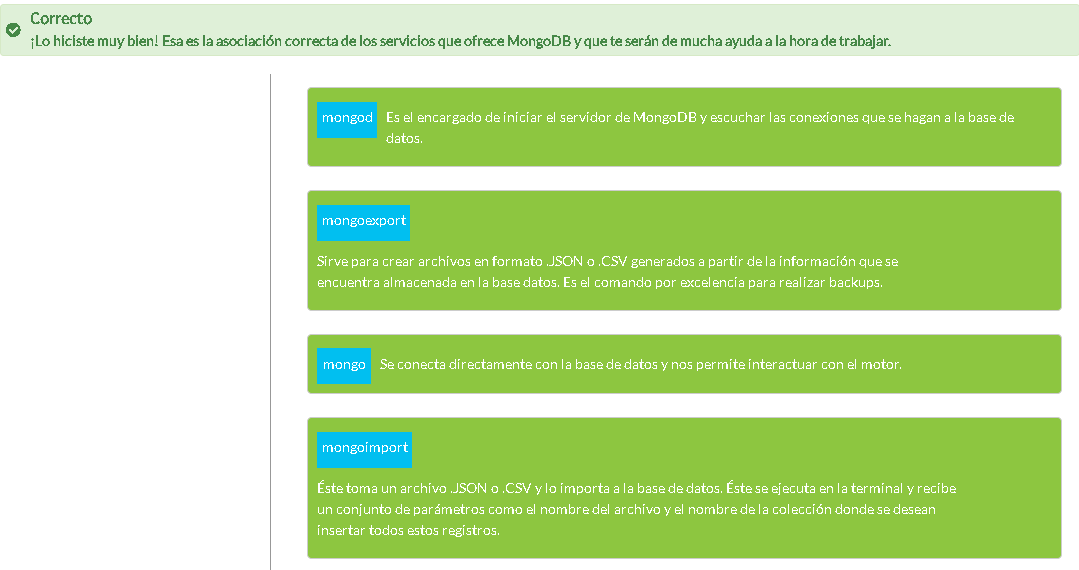
Con el comando:



* Servicio mongo – lo que hace es conectarse directamente con la base de datos y nos permite interactuar con el motor. Es el REPL de la base de datos.
  1. Actividad

Tiempo de pensar

Hemos estudiado qué es MongoDB, su proceso de descarga e instalación, y cuáles son los servicios que ofrece. En esta actividad afianzaremos el conocimiento adquirido a través de la exploración que realizamos de los archivos que hacen funcionar a MongoDB.  
  
Recuerda que vimos dos de los servicios más importantes e hicimos referencia a otros dos que son de mucha utilidad a la hora de trabajar.  
  
Instrucciones:  
En esta oportunidad, tu trabajo será relacionar cada servicio con su respectiva definición. ¡Adelante!



1. Lección 2: Tipos de conexión a la base de datos

3.1 Mongo (REPL)

Comando: mongo 🡪 Para entrar al REPL

C:\Program Files\MongoDB\Server\4.4\bin>mongo.exe

{"t":{"$date":"2020-10-20T23:20:32.698Z"},"s":"W", "c":"-", "id":23707, "ctx":"main","msg":"Failed to adjust token privilege for privilege '{toUtf8String\_name}'","attr":{"toUtf8String\_name":"SeIncreaseWorkingSetPrivilege"}}

MongoDB shell version v4.4.1

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("76863399-42c3-4b03-a620-a8f840f24b50") }

MongoDB server version: 4.4.1

---

The server generated these startup warnings when booting:

2020-10-19T16:04:20.913-06:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2020-10-19T16:04:20.913-06:00: SeIncreaseWorkingSetPrivilege privilege is not granted to the process. Secure memory allocation for SCRAM and/or Encrypted Storage Engine may fail.

---

---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display

metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you

and anyone you share the URL with. MongoDB may use this information to make product

improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()

To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

---

>

Mostrar colecciones: show collections

Comportamiento peculiar a la hora de crear una conexión, crea una colección cuando intentamos guardar un registro y esta no existe o cuando intentamos usarla por medio del REPL

Ejemplo:

> use nuevaDB

switched to db nuevaDB

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

> show collections

> db.users.insert({nombre:'Daniel',edad:23,peso:70})

WriteResult({ "nInserted" : 1 })

> db.users.find()

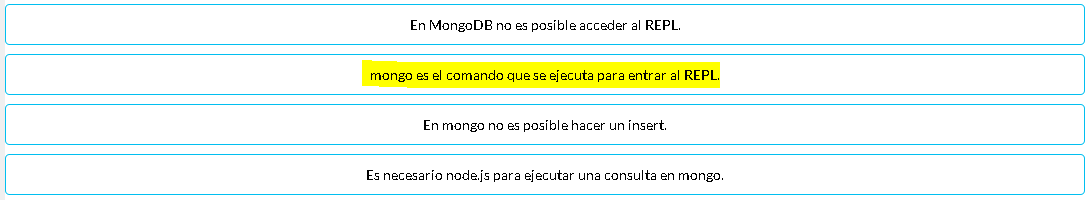
{ "\_id" : ObjectId("5f8f71e0d7c8b87267fcd4a0"), "nombre" : "Daniel", "edad" : 23, "peso" : 70 }

>

3.2 Actividad

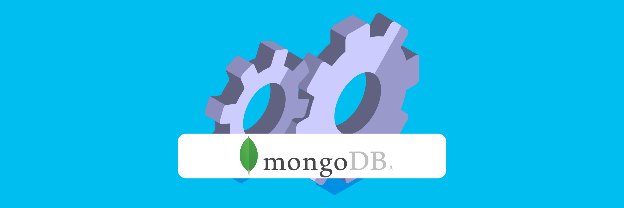
Tiempo de pensar

Hemos conocido los tipos de conexión a la base de datos que existen y, adicionalmente, aprendimos cómo usar el REPL de MongoDB y a explorar muchas de sus funcionalidades a través de este pequeño programa. Te invitamos a que realices esta actividad para que pongas en práctica lo que acabas de aprender.  
  
Instrucciones:  
En la siguiente actividad encontrarás un enunciado con varias opciones de respuesta. Lee con atención y determina, de acuerdo con lo aprendido en esta lección, cuál es la respuesta correcta.  
  
Enunciado:  
Después de haber visto el video, podemos concluir que:



3.3 Lectura: Driver Nativo Node.js

Driver nativo para node.js



Luego de ver el REPL en funcionamiento y conocer cuáles son los principales servicios que ofrece MongoDB junto con su instalación, vamos a ver cómo interactúa con node.js y cómo establecer una conexión entre ambos utilizando el driver nativo.  
  
Lo primero que haremos es iniciar un proyecto vacío con NPM y luego instalar el paquete mongodb:

npm init

Ahora vamos a instalar la librería:

npm install --save mongodb

Este paquete se llama tal cual como la base de datos, y contiene un conjunto de métodos y propiedades que nos permiten realizar una conexión e interactuar entre mongodb y node.js

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/test";

MongoClient.connect(url, function(error, db) {

console.log("Se establecio conexion con el servidor correctamente");

/\*

\* Hacer algo con la base de datos

\*/

db.close();

})

Referenciar una colección:

MongoClient.connect(url, function(error, db){

console.log("se establecio la conexion con el servidor correctamente")

//Referenciar una coleccion

let users = db.colletion("users")

/\*

\* Hacer algo con la base de datos

\*/

db.close()

})

Insertar registros en la colección:

MongoClient.connect(url, function(error, db){

console.log("se establecio la conexion con el servidor correctamente")

let users = db.colletion("users")

users.insertMany([name: "Estiven gonzales", age: 24}, {name: "Radamel Falcao", age:24}], function(err, res){

})

/\*

\*Hacer algo con la base de datos

\*/

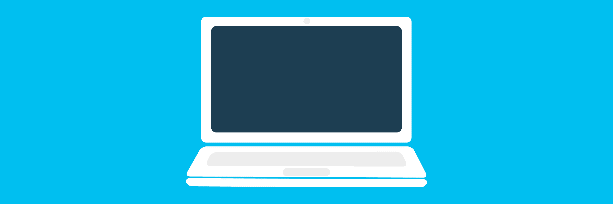
db.close()

})

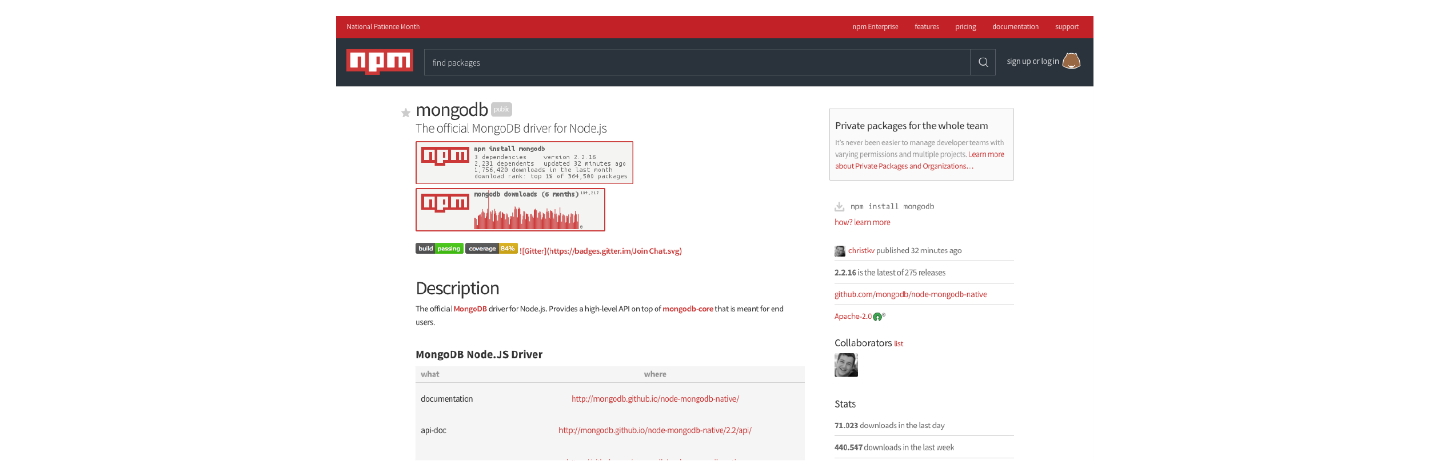
Esto es una introducción al driver nativo de node.js, en el próximo video veremos qué otras opciones existen para realizar una conexión entre mongodb y node.js.

3.4 Lectura: Drivers para trabajar MongoDb con NodeJS

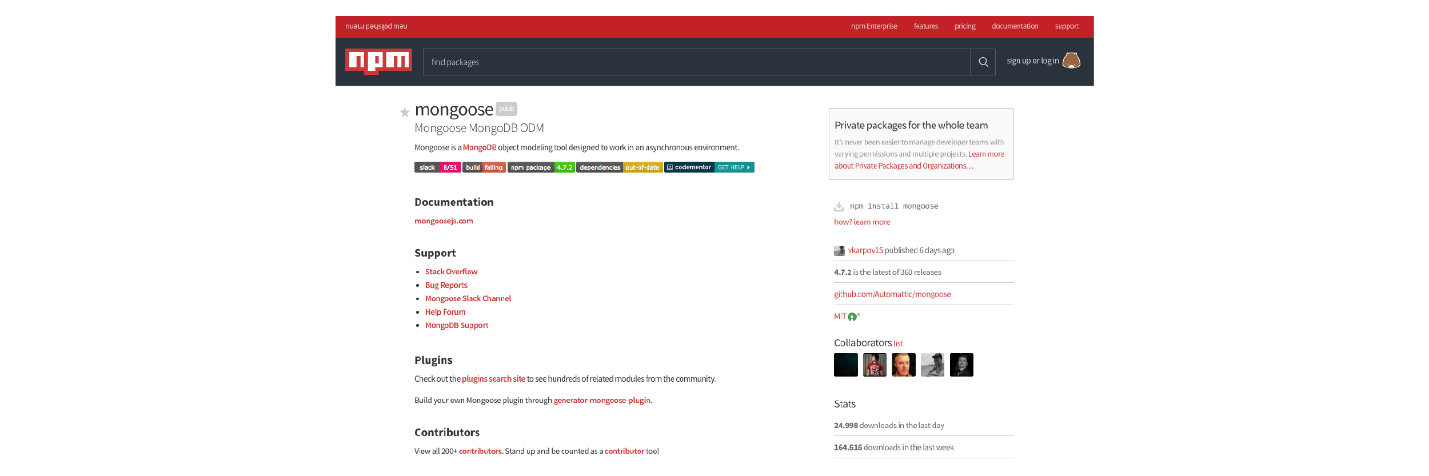
Drivers para trabajar MongoDB con NodeJS

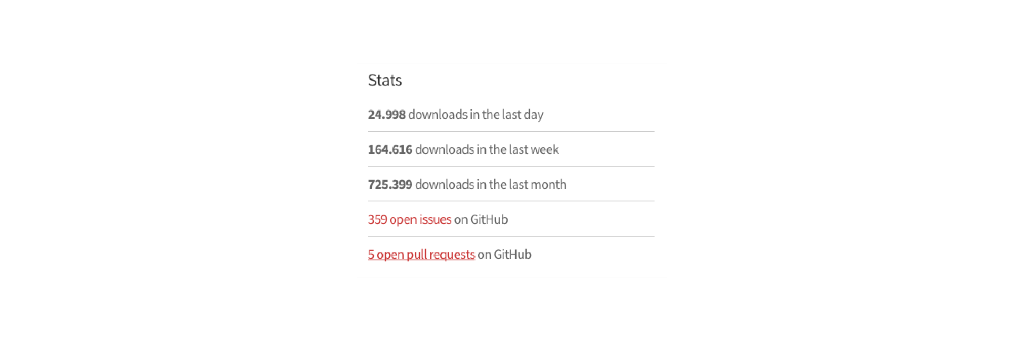


En la actualidad tenemos dos opciones para trabajar MongoDB junto con Node.js y tener un flujo de trabajo amigable para el desarrollador.  
  
En primer lugar se encuentra el driver nativo de Node.js (mongodb). Este driver nos provee una interfaz para realizar conexiones, manipular la información e incluso configurar programáticamente las funciones de replicación y auto-sharding.  
  
En npm podemos ver toda la descripción completa del driver.



Por otro lado, existe una alternativa bastante utilizada por los desarrolladores alrededor del mundo. Su nombre es **Mongoose** y sobresale sobre el driver nativo, ya que, además de realizar todas las funcionalidades que realiza el driver, contiene una capa para el modelado de los datos, es decir, Mongoose me obliga a crear un esquema y a mantener una consistencia en los registros que ingresan a la base de datos. Esto es una gran ventaja, porque la información que tengamos almacenada siempre tendrá la misma estructura, facilitando el trabajo de los desarrolladores a la hora de manipular estos datos.  
  
Repositorio de Mongoose en npm:

  
Aquí podemos ver que es bastante utilizado en la sección de descargas:



1. Lección 3: Interacción con la base de datos

[4.1. Insertar registros](https://learn.nextu.com/mod/lesson/view.php?id=3570&pid=P_WEBDEV_V2)

Operaciones CRUD (crear actualizar consultar y eliminar) con MongoDB y NodeJS

4.2. Actividad 1

4.3. Eliminar registros

4.4. Actividad 2

4.5. Consultar y actualizar registros

4.6. Actividad 3

1. Lección 4: Mongoose
2. UNIDAD 3: Prueba