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?

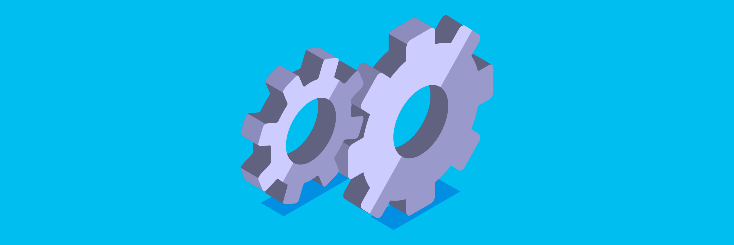


MongoDB es una base de datos NoSQL orientada a documentos. Esta utiliza un formato llamado BSON, que es una representación en binario de un documento JSON.  
Mongo es un motor de bases de datos multiplataforma de esquema libre, es decir, que cada registro puede contener una estructura diferente y esto 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

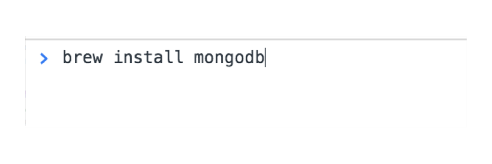
MongoDB: qué es y cómo instalarlo



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:



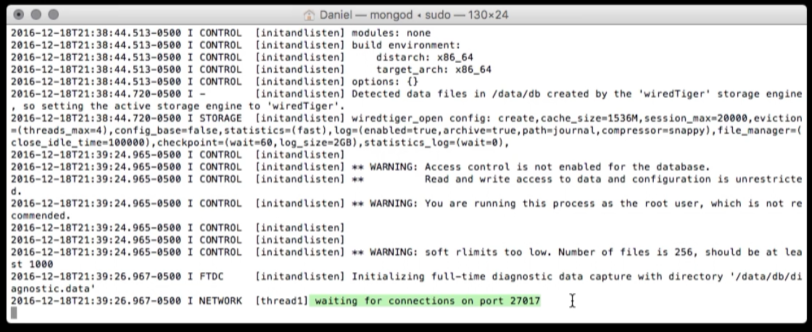
Los que están usando Linux o Windows les recomiendo ingresar a la página oficial de MongoDB y realizar los pasos de instalación.

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

Servicios o demonios: 13 servicios

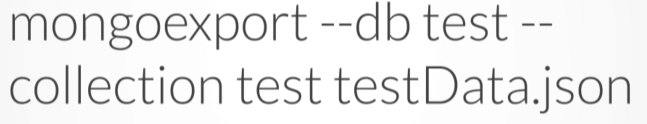
Más importantes:

* 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.

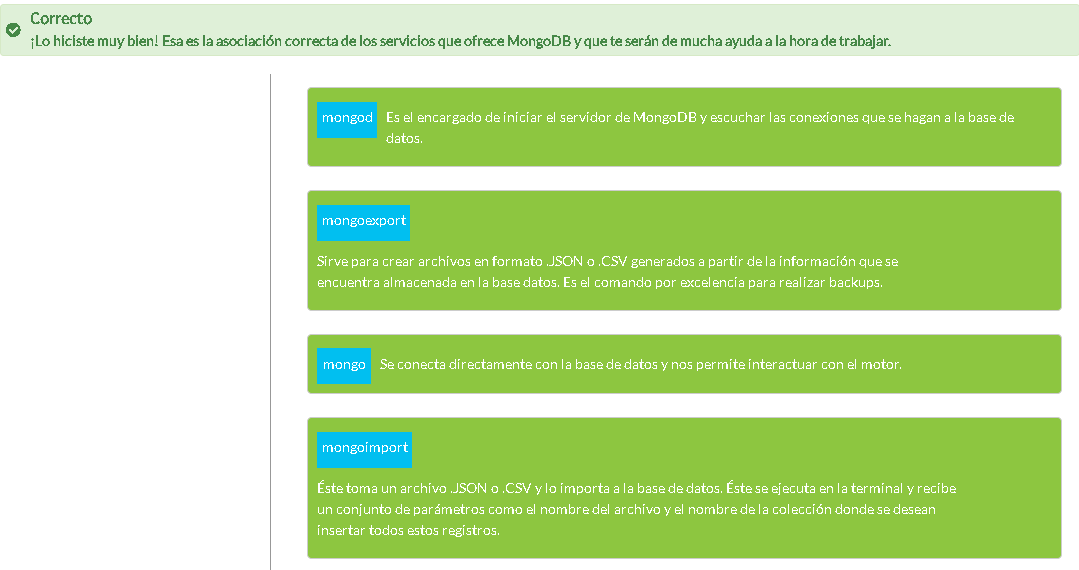
Con el comando:



* 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()

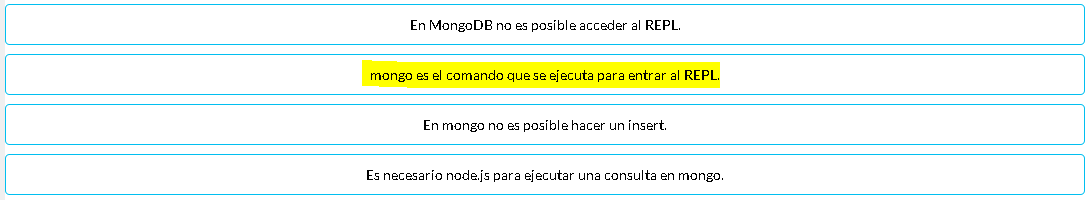
{ "\_id" : ObjectId("5f8f71e0d7c8b87267fcd4a0"), "nombre" : "Daniel", "edad" : 23, "peso" : 70 }

>

3.2 Actividad

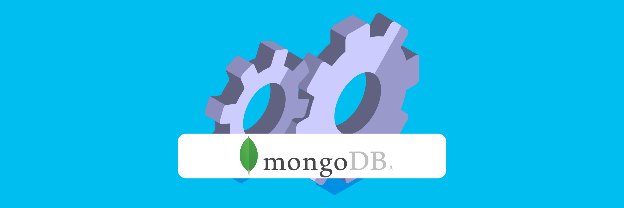
Tiempo de pensar

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

\*/

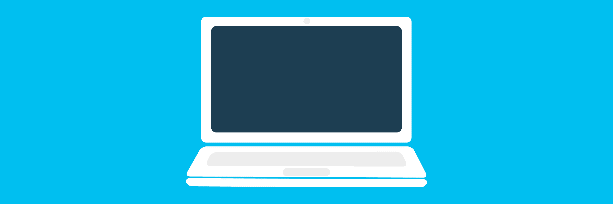
db.close()

})

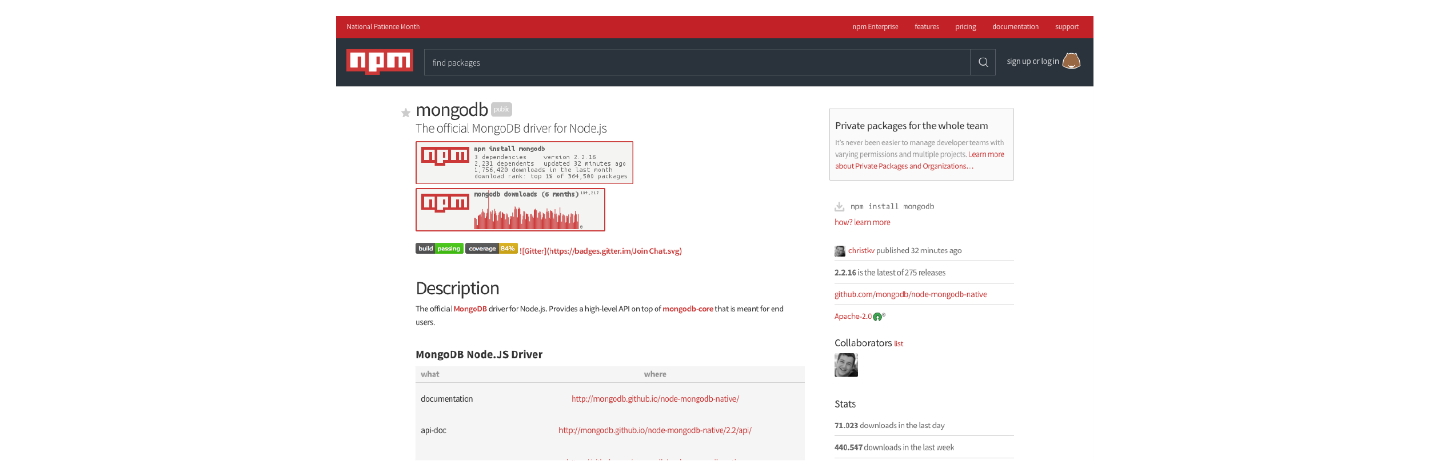
Esto es una introducción al driver nativo de node.js, en el próximo video veremos qué otras opciones existen para realizar una conexión entre mongodb y node.js.

3.4 Lectura: Drivers para trabajar MongoDb con NodeJS

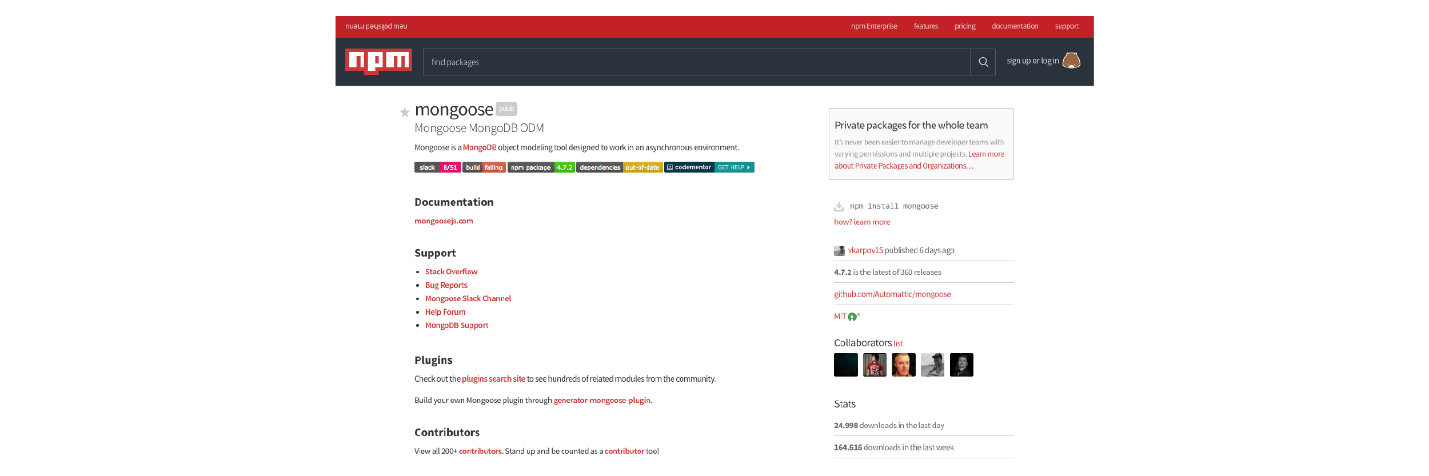
Drivers para trabajar MongoDB con NodeJS

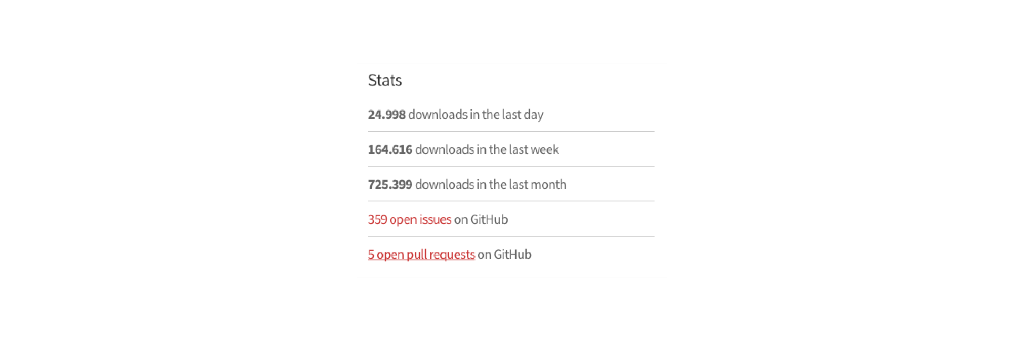


En la actualidad tenemos dos opciones para trabajar MongoDB junto con Node.js y tener un flujo de trabajo amigable para el desarrollador.  
  
En primer lugar se encuentra el driver nativo de Node.js (mongodb). Este driver nos provee una interfaz para realizar conexiones, manipular la información e incluso 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)

Para conectarse a una DB mongo:

var MongoClient = require('mongodb').MongoClient

var url = "mongodb://localhost/"

var Operaciones = require('./CRUD.js')

MongoClient.connect(url, { useNewUrlParser: true, useUnifiedTopology: true } , function(err,client){

const db = client.db('nodeDriver');

// Conexión

if(err) console.log(err);

console.log("Conexión establecida con la base de datos");

Operaciones.insertarRegistro(db, (error, result) =>{

if(error) console.log("Error insertando los registros: "+error);

})

})

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

NodeDriverExmple>npm init

Par instalar el driver nativo:

NodeDriverExmple>npm install --save mongodb

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

nodeDriver 0.000GB

> use nodeDriver

switched to db nodeDriver

> show collections

users

> db.users.find()

{ "\_id" : ObjectId("5f91c86a6fce9b4d185e6be0"), "nombre" : "David", "edad" : 25, "peso" : 70 }

{ "\_id" : ObjectId("5f91c86a6fce9b4d185e6be1"), "nombre" : "Steven", "edad" : 35, "peso" : 80 }

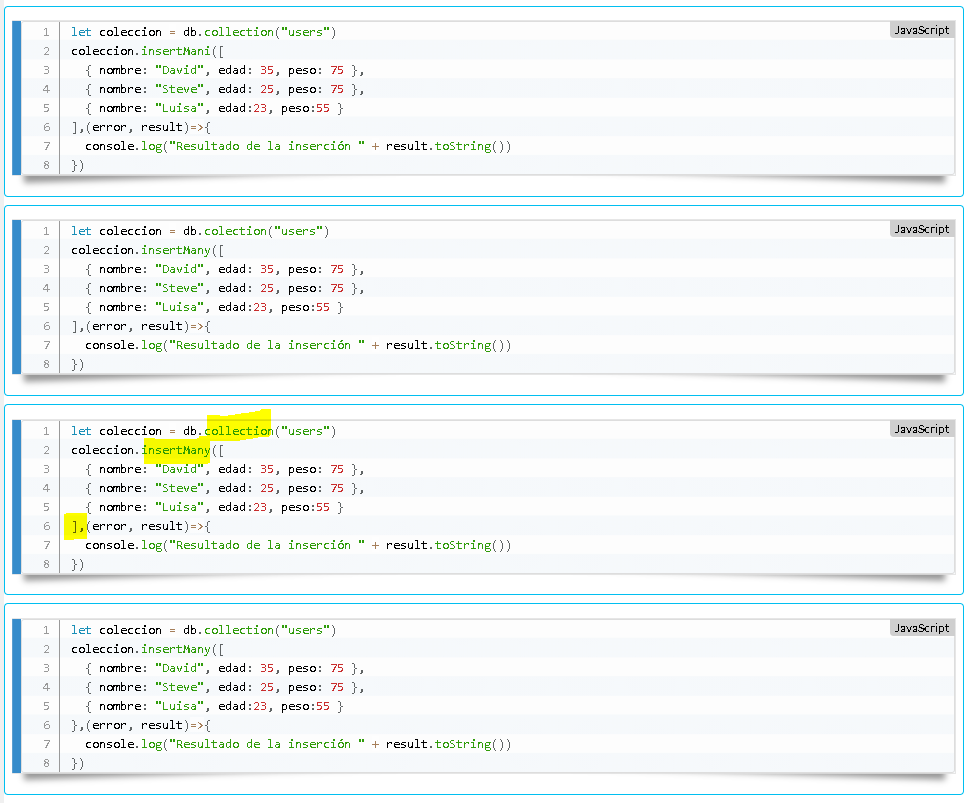
{ "\_id" : ObjectId("5f91c86a6fce9b4d185e6be2"), "nombre" : "Fernando", "edad" : 40, "peso" : 68 }

>

4.2. Actividad 1

Tiempo de pensar

Hemos estudiado las bases para realizar operaciones CRUD en MongoDB, pero se hizo énfasis en una de ellas que es la inserción de datos dentro de una base de datos Mongo. Te invitamos a que afiances todo el conocimiento que adquiriste haciendo esta actividad.  
  
Instrucciones:  
Selecciona el segmento de código que al ejecutarse permite realizar correctamente una inserción de múltiples datos dentro de una colección en mongodb y que su sintaxis esté correcta:



4.3. Eliminar registros

3 métodos:

Remove() 🡪 Elimina uno o todos los documentos que coincidan con un filtro en específico, el cual corresponde a un objeto enviado como parámetro

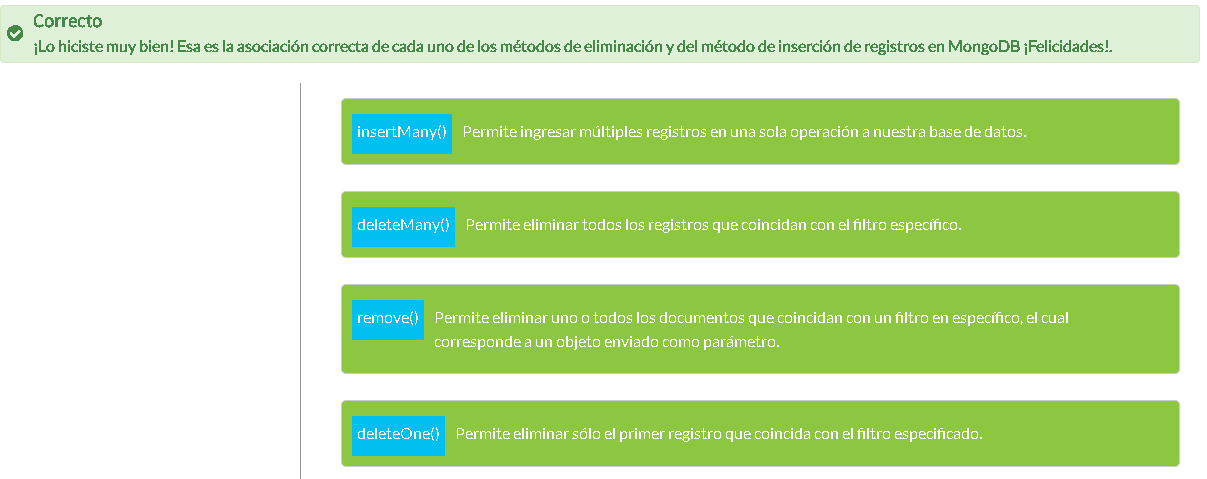
deleteOne() 🡪 Este método elimina solo el primer registro que coincida con el filtro especificado.

deleteMany() 🡪 Este método elimina todos los registros que coincida con el filtro especificado.

4.4. Actividad 2

Tiempo de pensar

Hemos estudiado los diferentes métodos de eliminación de registros de nuestra base de datos Mongo, con esto ya llevamos dos de las partes que componen el CRUD. Estudiamos a detalle cómo utilizarlos y cuáles son las condiciones para precisar cuál de ellos usar en determinada situación. Ahora es momento de afianzar el conocimiento que adquiriste, por eso te proponemos la siguiente actividad.  
  
Instrucciones:  
Asocia cada método con su respectiva función en MongoDB, teniendo presente lo visto anteriormente:

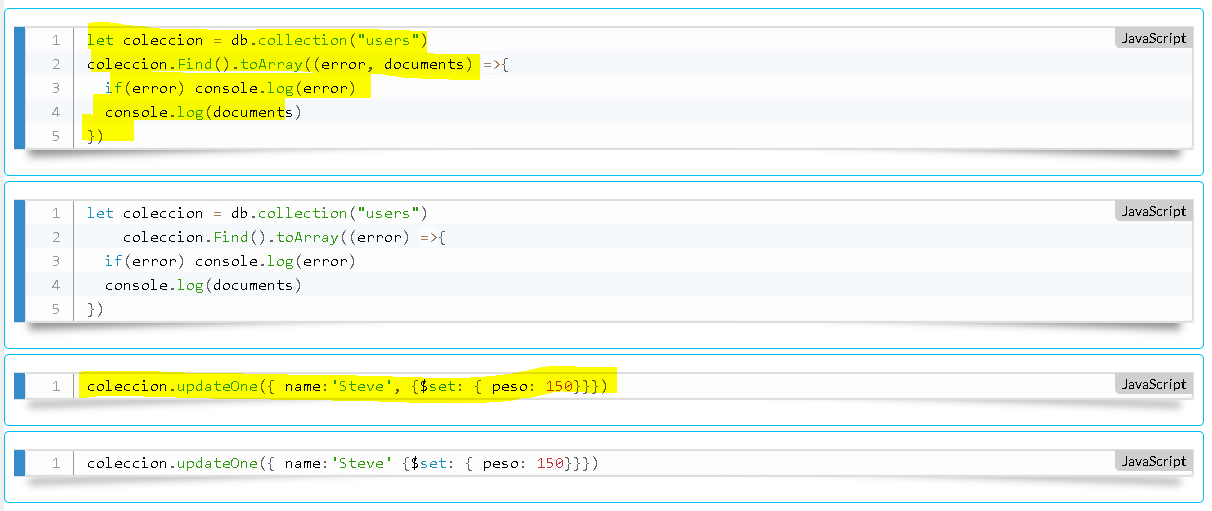


4.5. Consultar y actualizar registros

4.6. Actividad 3

Tiempo de pensar

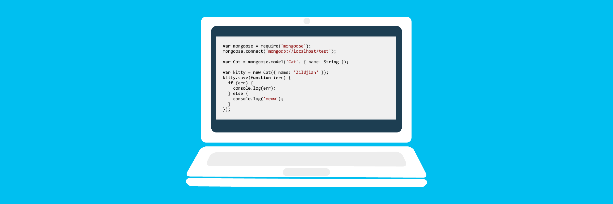
Hemos estudiado el método para consultar y actualizar un registro de la base de datos usando el driver nativo de node.js para MongoDB. Adicionalmente, también estudiamos cómo hacer inserción y eliminación de datos en MongoDB, con lo que completamos las cuatro operaciones component el CRUD. Ahora es tiempo de que pongas a prueba tus conocimientos realizando esta actividad.  
  
Instrucciones:  
Selecciona los segmentos de código que permiten realizar consulta y modificación de registros en MongoDB, teniendo presente que su sintaxis debe estar escrita correctamente.



1. Lección 4: Mongoose

[5.1. Lectura: Que es mongoose, instalación](https://learn.nextu.com/mod/lesson/view.php?id=3571&pid=P_WEBDEV_V2)

¿Qué es Mongoose? - Instalación



Mongoose.js es una librería para Node.js que nos permite mapear los documentos de la base de datos en objetos JavaScript, esto es conocido como un ODM (Object Data Mapping).  
  
Además de convertir los documentos en objetos Javascript para ser tratados con mayor facilidad dentro de nuestra aplicación, Mongoose nos provee una solución para realizar el modelado de nuestros documentos, es decir, definir un esquema consistente, manejando tipos de datos y relaciones dentro de las colecciones. Estos son conocidos como esquemas y modelos, un esquema es un objeto JavaScript donde definimos los campos y el tipo de dato que corresponde a cada campo dentro de un documento específico.  
  
Ejemplo de un esquema con Mongoose:

let UserSchema = new mongoose.Schema ({

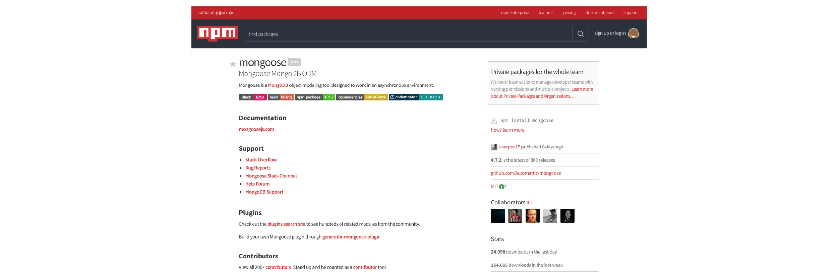
nombre: { type: String, required: true},

edad: { type: Number, required: true},

peso: { type: Number, required: true},

})

Instalación:  
  
La url oficial donde se encuentra el paquete Mongoose en NPM es la siguiente: <https://www.npmjs.com/package/mongoose>



La página oficial de Mongoose es la siguiente: <http://mongoosejs.com/>

  
Aquí podemos encontrar toda la documentación oficial y las guías de trabajo. La instalación será a través de NPM, así que en la terminal y en el directorio del proyecto escribimos el comando:

npm install --save mongoose

5.2. Schemas & Modelos

Schema: Un schema se conoce como la definición de las propiedades y los tipos de datos que almacerán estas propiedades dentro de la conexión.

Modelo: Es la representación del Schema a través de un nombre específico

Mongoose permite los siguientes tipos de datos:

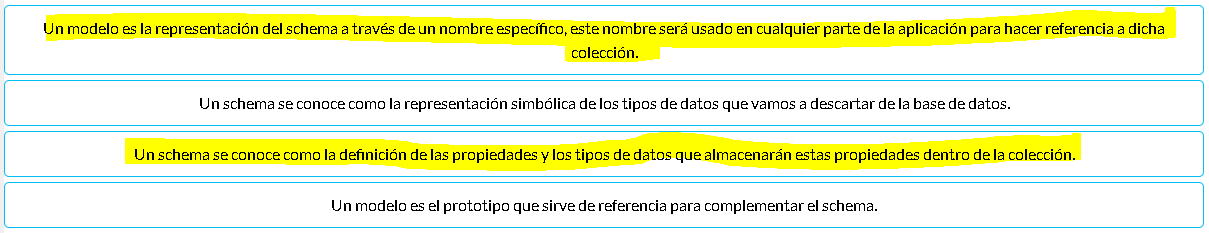


5.3. Actividad

Tiempo de pensar

En el video anterior vimos cómo reemplazar la conexión utilizando el driver nativo, por una conexión utilizando Mongoose. También aprendimos dos definiciones muy importantes: schema y modelo. Ahora es tu oportunidad de poner a prueba tus conocimientos haciendo esta actividad.  
  
Instrucciones:  
Lee con atención la pregunta con sus respectivas opciones y escoge la definición correcta de ambos conceptos.

Enunciado:  
Después de conocer qué son los schemas y los modelos, escoge de las opciones aquella que defina mejor a cada concepto.



5.4. CRUD con mongoose

5.5. Ejercicio

HORA DE  
**codificar**

El sistema de base de datos MongoDB, es uno de los más usados actualmente por su característica innovadora en el almacenamiento de los datos. Sumadas a sus características orieentadas a documentos y su interacción sencilla desde los lenguajes del lado del servidor, MongoDB trabaja junto con diversas librerías y complementos que mejoran su rendimiento. Entre ellos se encuentra Mongoose, la cual, es una librería de JavaScript que obliga la creación de un esquema en la base de datos y aporta una serie de interacciones basadas en objetos entre Node.js y MongoDB.  
  
En este ejercicio complementarás el proyecto que se ha venido desarrollando durante las lecciones de esta unidad.

Contenido

Para el desarrollo de este ejercicio debes descargar el código base en el que encontrarás el código relacionado con la base de datos de estudiantes que se desarrolló durante los videos de las pasadas lecciones.

Instrucciones

1. Analiza la estructura de archivos del código base, identificando las carpetas y archivos según su funcionalidad.
2. Corre el demonio de MongoDB para que el sistema de bases de datos escuche conexiones
3. Ingresa 10 registros a la base de datos con los siguientes datos:  
   a. nombre del estudiante  
   b. edad  
   c. género  
   d. estatura
4. Realiza una consult para obtener todos los registros ingresados en el paso anterior, verifica que hayan sido añadidos correctamente.
5. Usa la función **sort** de Mongoose para obtener los estudiantes ordenados según su edad de manera descendente.

Buenas Prácticas

Es muy recomendado hacer uso de Mongoose con el objetivo de tener una base de datos consistente que responda a una estructura determinada, esto permite la unificación de los datos y la solidez del diseño sobre el cual se realiza el proceso de almacenamiento de información.

1. UNIDAD 3: Prueba

# **Interactuando con Bases de Datos >**UNIDAD 3: Prueba

|  |  |
| --- | --- |
| **Comenzado en** | Thursday, 22 de October de 2020, 15:23 |
| **Estado** | Terminados |
| **Finalizado en** | Thursday, 22 de October de 2020, 15:26 |
| **Tiempo empleado** | 2 mins 49 segundos |
| **Puntos** | 10/10 |
| **Calificación** | **100** de un total de 100 |
| **Comentario de retroalimentación** | Buen trabajo! |

Top of Form

### **Pregunta**1

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Selecciona dos características de MongoDB.

Seleccione una o más de una:

a. Replicación y alta disponibilidad.

b. Auto-Sharding.

c. Esquemas relacionales.

d. Almacenamiento de los datos mejor organizados.

#### Retroalimentación

Entre las características de MongoDB se encuentran el auto-sharding y la replicación y alta disponibilidad.

Las respuestas correctas son: Replicación y alta disponibilidad., Auto-Sharding.

### **Pregunta**2

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Qué hace el comando mongo?

Seleccione una:

a. Permite acceder al REPL de mongo.

b. Muestra la versión instalda de mongo.

c. Instala MongoDB en la máquina.

d. No existe tal comando.

#### Retroalimentación

Para acceder al REPL de MongoDB se usa el comando mongo.

La respuesta correcta es: Permite acceder al REPL de mongo.

### **Pregunta**3

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Qué hace el comando db.users.find()?

Seleccione una:

a. Muestra los registros de la tabla users.

b. Busca un registro vacío en la tabla users.

c. Busca un registro nulo en la tabla users.

d. El comando es incorrecto.

#### Retroalimentación

Para ver todos los registros de una tabla se usa el comando find.

La respuesta correcta es: Muestra los registros de la tabla users.

### **Pregunta**4

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Qué es un schema?

Seleccione una:

a. Es la definición de las propiedades y sus tipos de datos en una colección.

b. Es la definición de un modelo en la colección.

c. Es la definición de un conjunto de tablas relacionadas.

d. Es una colección específica de la base de datos.

#### Retroalimentación

Un schema establece las propiedades de una colección indicando sus tipos de dato.

La respuesta correcta es: Es la definición de las propiedades y sus tipos de datos en una colección.

### **Pregunta**5

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Qué es MongoDB?

Seleccione una:

a. Base de Datos NoSQL orientada a documentos.

b. Base de Datos SQL orientada a documentos.

c. Base de Datos NoSQL orientada a relaciones.

d. Base de Datos SQL orientada a relaciones.

#### Retroalimentación

MongoDB es una base de datos NoSQL orientada a documentos.

La respuesta correcta es: Base de Datos NoSQL orientada a documentos.

### **Pregunta**6

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Selecciona dos drivers para trabajar con MongoDB desde Node.js.

Seleccione una o más de una:

a. mongodb.

b. Mongoose.

c. mongo-driver.

d. mongo.

#### Retroalimentación

Dos de los drivers más comunes son mongodb y Mongoose.

Las respuestas correctas son: mongodb., Mongoose.

### **Pregunta**7

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

El método para realizar una conexión a una base de datos MongoDB desde Node.js es connect( ).

Seleccione una:

a. Verdadero

b. Falso

#### Retroalimentación

El objeto MongoClient invoca el método connect para establecer una conexión a un servidor específico.

La respuesta correcta es: Verdadero

### **Pregunta**8

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cómo se crea una base de datos?

Seleccione una:

a. use nombre\_db

b. create nombre\_db

c. new nombre\_db

d. mongo -new nombre\_db

#### Retroalimentación

Para crear una base de datos se debe usar por medio del REPL con el comando use.

La respuesta correcta es: use nombre\_db

### **Pregunta**9

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Qué es un modelo?.

Seleccione una:

a. Es la representación de un schema a través de un nombre específico.

b. Es la representación de una base de datos a través de un id.

c. Es la representación de una colección a través de un nombre específico.

d. Es la representación de una tabla a través de un id.

#### Retroalimentación

Un modelo es la representación de un schema mediante un nombre específico.

La respuesta correcta es: Es la representación de un schema a través de un nombre específico.

### **Pregunta**10

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Qué función permite insertar más de un registro a la vez?

Seleccione una:

a. insertMany.

b. insert.

c. multipleInsert.

d. insertVarious.

#### Retroalimentación

El método insertMany recibe un arreglo de objetos como parámetros para insertar varios registros al tiempo.

La respuesta correcta es: insertMany.

Bottom of Form