

Clase 9: No relacionales - MongoDB

Sitio: [Centro de E-Learning - UTN.BA](#)
Curso: Curso de Backend Developer - Turno
Noche
Libro: Clase 9: No relacionales - MongoDB

Imprimido
por:

Nelson Brian Avila Solano

Día:

Tuesday, 9 de December de 2025,
19:47

Descripción

Objetivos

- Introducir a los estudiantes en el mundo de las bases de datos no relacionales.
- Comprender la arquitectura de MongoDB y su instalación.
- Realizar la instalación y configuración de MongoDB en sus equipos.
- Construir bases de datos y colecciones utilizando MongoDB Compass.
- Realizar consultas básicas en MongoDB utilizando Compass.

Tabla de contenidos

1. Introducción
2. Bases de datos NoSQL
3. MONGO instalación
4. ¡A practicar!
5. Mongodb compass

1. Introducción

Las bases de datos no relacionales han revolucionado la forma en que se almacena y gestiona la información en diversos entornos tecnológicos. A diferencia de los sistemas relacionales, estas bases de datos permiten una estructura más flexible y adaptativa, lo que resulta especialmente útil para manejar grandes volúmenes de datos con esquemas dinámicos. En esta clase, los participantes explorarán el concepto de bases de datos NoSQL, con un enfoque particular en MongoDB, una de las soluciones más utilizadas en la actualidad por su capacidad de escalabilidad y facilidad de uso.

El objetivo de esta clase es brindar a los participantes los conocimientos fundamentales sobre la instalación, configuración y uso de MongoDB, tanto a través de la línea de comandos como con la interfaz gráfica MongoDB Compass.

2. Bases de datos NoSQL

Bases de datos NoSQL

Las bases de datos no relacionales son un sistema de almacenamiento de información que se caracteriza por no usar el lenguaje SQL para las consultas.

Están orientadas a los documentos y le permiten almacenar y recuperar datos en formatos que no sean tablas. Algunas de las plataformas NoSQL más conocidas son MongoDB, Elasticsearch® y Redis®.

Las bases de datos no relacionales usan un modelo de almacenamiento que está optimizado para los requisitos específicos del tipo de datos que se almacena. Por ejemplo, los datos se pueden almacenar como pares clave/valor simple, como documentos JSON o como un grafo que consta de bordes y vértices.

¿Qué es JSON?

La notación de objetos JavaScript, más comúnmente conocida como JSON, fue definida como parte del lenguaje JavaScript a principios de la década de 2000 por el creador de JavaScript, Douglas Crockford, aunque no fue hasta 2013 cuando se especificó oficialmente el formato.

Los objetos de JavaScript son contenedores asociativos simples, en los que una clave de cadena se asigna a un valor (que puede ser un número, una cadena, una función o incluso otro objeto). Este rasgo de lenguaje simple permitió que los objetos de JavaScript se representarán de manera notablemente simple.

```
{
  "_id": 1,
  "name": { "first": "John", "last": "Backus" },
  "contribs": [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],
  "awards": [
    {
      "award": "W.W. McDowell Award",
      "year": 1967,
      "by": "IEEE Computer Society"
    }, {
      "award": "Draper Prize",
      "year": 1993,
      "by": "National Academy of Engineering"
    }
  ]
}
```

MongoDB es un sistema de base de datos NoSQL, orientado a documentos y de código abierto.

En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.



MongoDB y Colecciones

Una colección en MongoDB es muy similar a una Tabla de una base de datos. La tabla almacena registros (filas) mientras que las colecciones almacenan documentos.

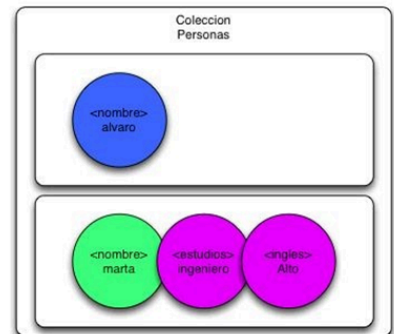
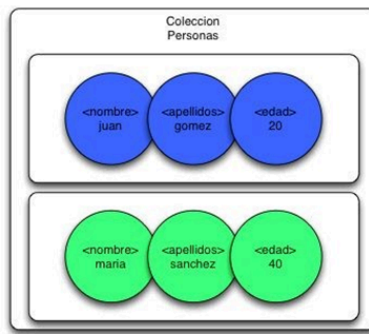
Base de datos y colecciones



MongoDB y Documentos

Un documento está compuesto por claves y valores (key,value) y cada documento puede tener variaciones importantes con el anterior dentro de una colección.

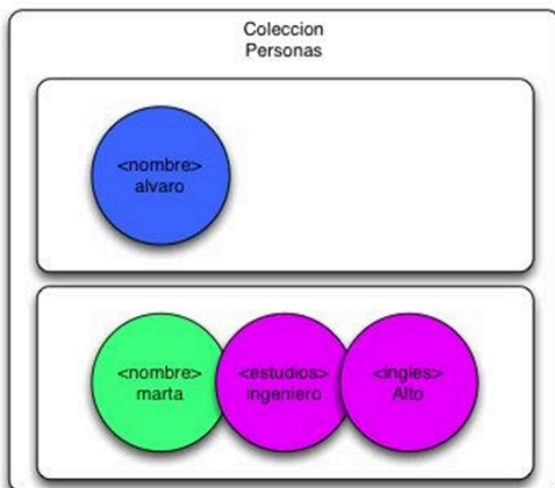
nombre	apellidos	edad
juan	gomez	20
maria	sanchez	40



Relacional

NO-Relacional

El primer documento solo dispone de un campo . Mientras el segundo dispone de tres pero estos son diferentes a los anteriores. **Esto es lo que en NoSQL se denomina Schema Free o libre y que aporta una gran flexibilidad a la hora de trabajar.** Aunque también puede acabar siendo un poco caótico



NO-Relacional

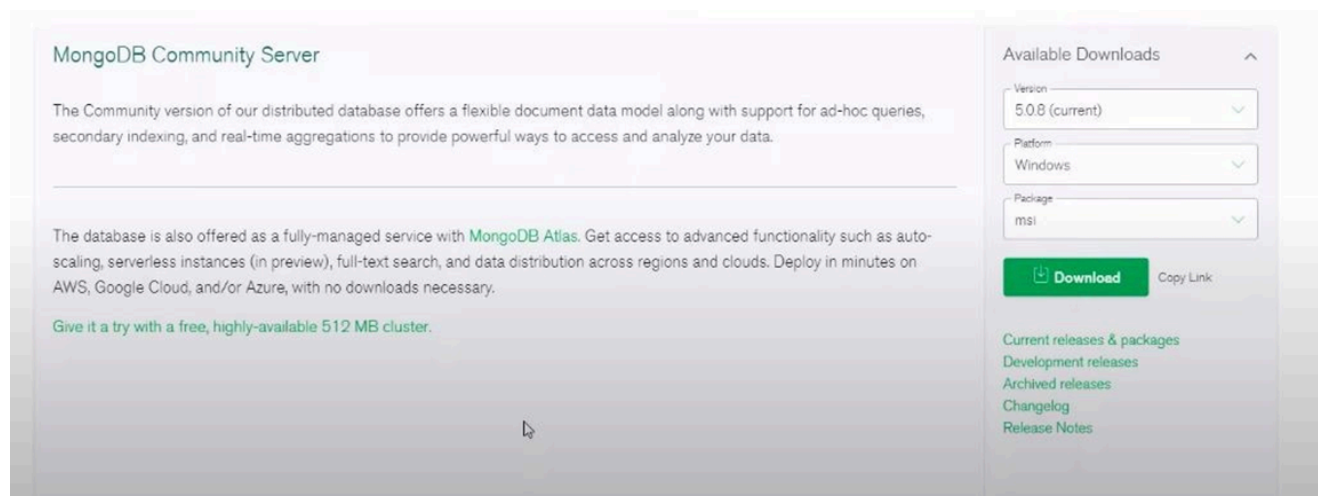
3. MONGO instalación

MONGO instalación

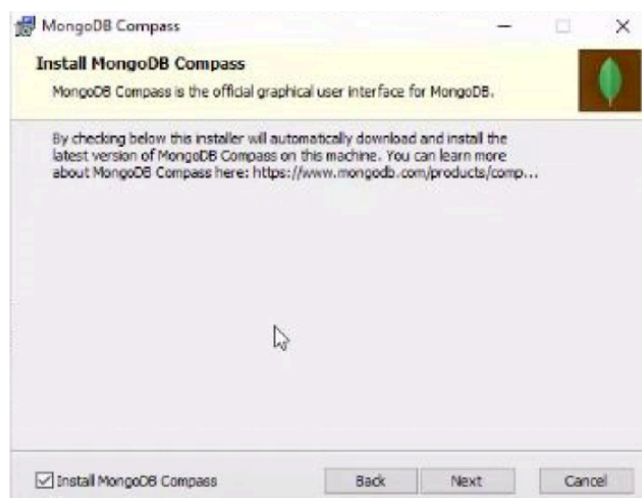
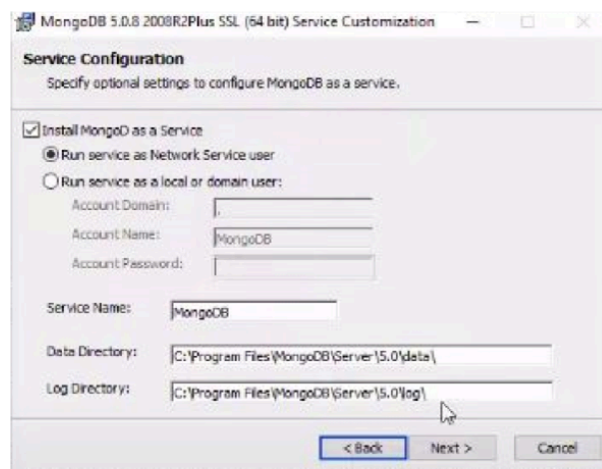
Instalación Windows

Nos dirigimos a la web oficial -> [Página oficial](https://www.mongodb.com/)

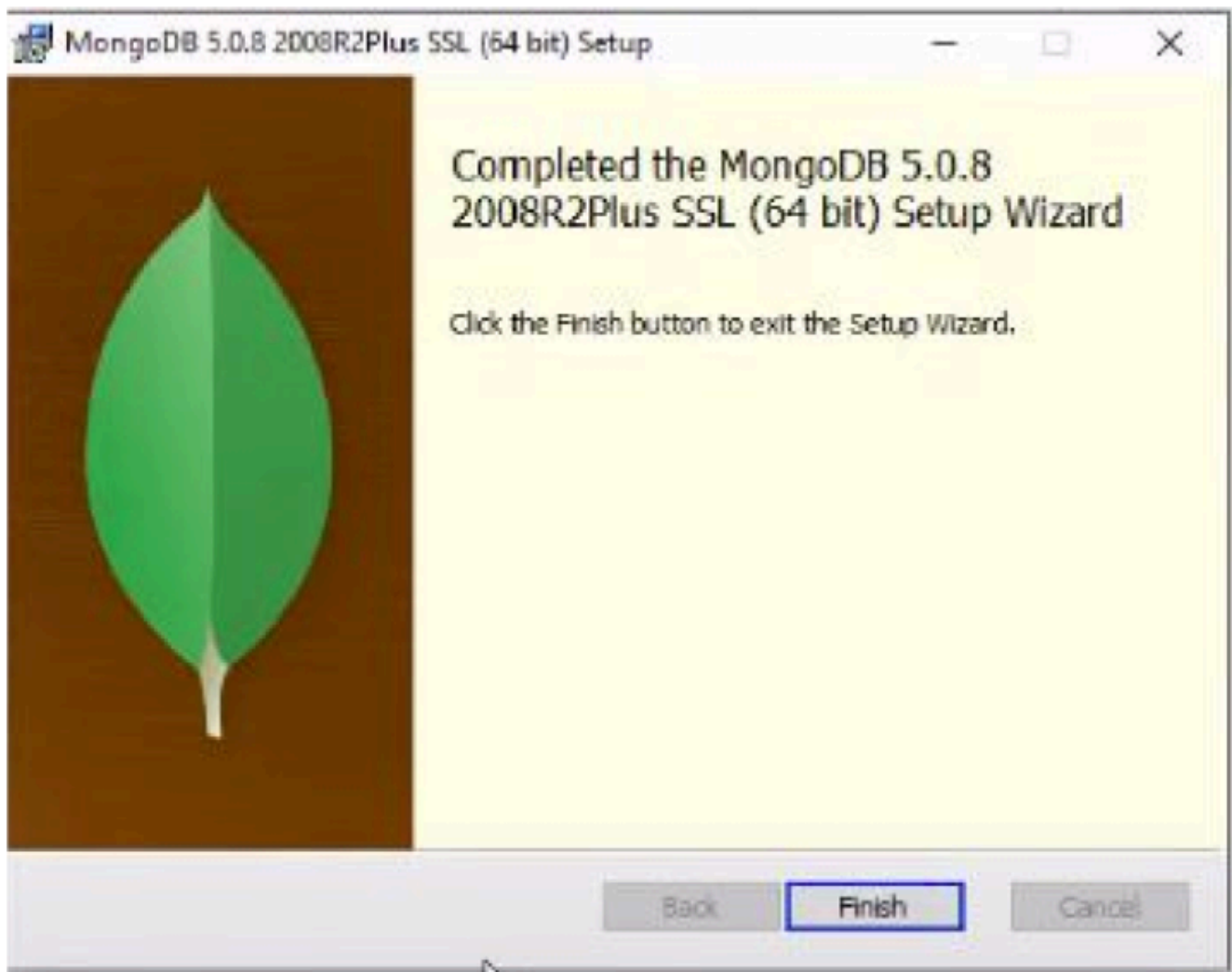
- **Sitio oficial:** <https://www.mongodb.com/>
- **Dirigirse a download:** <https://www.mongodb.com/try/download/community>



- **Siguiente y configuración por defecto.**
- Instalar mongodb Compass



Instalación



Verificar la instalación

En terminal ejecutamos el comando `mongod -version` y obtendremos los detalles de nuestra instalación.

```
naranjas@naranjas:~$ mongod -version
db version v5.0.8
Build Info: {
  "version": "5.0.8",
  "gitVersion": "c87e1c23421bf79614baf500fda6622bd90f674e",
  "opensslVersion": "OpenSSL 1.1.1f  31 Mar 2020",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "ubuntu2004",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```

Mongo Shell

Comandos:

>mongod // mongosh

>show dbs

>use escuela → Para ir a una base de datos

>use banco → para crear una nueva base de datos (por consola sólo muestra las que tiene algún documento)

>use escuela

>show collections → mostrar colecciones

>db.estudiantes.find() → Ver la información dentro de la colección

>db.estudiantes.insertOne({"nombre" : "Harry", "apellido" : "Potter", "ciudad" : "Valle de Godric", "edad" : 30})

>db.estudiantes.find({_id:ObjectId("628d47ed55967109369cbcf5")}) → Buscar por id

4. ¡A practicar!

¡A practicar!

Por consola:

- Entrar instancia mongo -> mongo // mongosh
- Ver las bd creadas por defecto -> show dbs
- Entrar a alguna de las anteriores -> use nombredb
- Crear una bd -> use nombredb
- Ver colecciones -> show collections
- Ver documentos -> db.nombrecoleccion.find()
- Insertar un documento ->
db.nombrecoleccion.insertOne({"nombre":
"Harry", "apellido":<"Potter"})

Mongo Shell

Buscar por coincidencias:

```
> db.estudiantes.find({"nombre": "harry"})
```

Insertar fechas:

```
> db.estudiantes.insertOne({"nombre" : "Hermione", "apellido" : "Granger", "ciudad" : "Londres", "edad" :  
30, "fecha_nacimiento": new  
Date(1993,02,20)})
```

5. Mongodb compass

Mongodb compass

Es una herramienta interactiva para consultar, optimizar y analizar sus datos en MongoDB.



¡A practicar lo visto!

- Ingresar y verificar conexión en Compass.
- Crear una base de datos.
- Crear una colección.
- Insertar, modificar y eliminar documentos.

Agregar, editar y actualizar documentos

Agregar documentos

Para agregar documentos nuevos a tu colección:

Selecciona tu Base de Datos y Colección: Abre MongoDB Compass y selecciona la base de datos y la colección donde deseas agregar documentos.

Insertar un Nuevo Documento:

1. Haz clic en el botón "ADD DATA".
2. Introduce los campos y valores del nuevo documento en el editor JSON.
3. Haz clic en "Insertar" para guardar el nuevo documento en la colección.

Editar un documento

1. Selecciona y abre el Documento: haz doble clic para abrirlo en el editor.
2. Realiza los Cambios:

Edita los campos y valores según sea necesario en el editor.

Haz clic en "UPDATE" para aplicar los cambios al documento editado.

Actualizar documentos

1. Selecciona tu Colección y define el filtro:
 - Selecciona la colección donde deseas realizar la actualización.
 - Haz clic en "Agregar Filtro" para especificar qué documentos actualizarás.
2. Ejecuta la Operación de Actualización:
 - Elige una operación de actualización como \$set, \$unset, etc., para modificar los campos del documento.
 - Confirma la operación haciendo clic en "Actualizar" para aplicar los cambios a los documentos seleccionados.