

Gestión de bases de datos NoSQL con MongoDB

Breve descripción:

En este componente formativo, se aborda el tema de las bases de datos no relacionales, en el que se presentan sus diferentes tipos y la forma en la que sus características se acoplan más fácilmente a ciertos contextos. Adicionalmente, se introduce la plataforma MongoDB como ejemplo para la gestión de bases de datos NoSQL de tipo documental.

Tabla de contenido

Introducción	1
1. Bases de datos NoSQL	2
2. MongoDB.....	15
2.1. Documentos	26
2.2. Comandos básicos	30
2.3. Operadores.....	30
Síntesis	37
Material complementario	38
Glosario	39
Referencias bibliográficas	40

Introducción

Cordial bienvenida a este componente formativo en el que trataremos la gestión de bases de datos NoSQL con MongoDB; para conocer la importancia del tema y los diferentes temas que se incluirán, le invitamos a revisar el siguiente video:

Video 1. Introducción



Introducción

Síntesis del video: Introducción

Hola estimado aprendiz, el día de hoy quiero hablarle del componente formativo que vamos a trabajar que es: Gestión de bases de datos NoSQL con MongoDB.

Si estas involucrado en la industria del “software” o has leído o haces un documento acerca de esto sabes que la gestión de los datos es vital para cualquier aplicación independientemente si hablamos de aplicaciones web de móvil o de escritorio, las bases de datos siempre es un elemento fundamental es transversal a

todo el proceso prácticamente, vamos a abordar bases de datos NOSQL, vamos a abordar sus características porque hoy en día esta en auge el uso de este tipo de aplicaciones de bases de datos; vamos a mirar las diferencias que tienen con las bases de datos tradicionales llamadas relacionales o SQL y vamos a abordar también el tema de MongoDB.

MongoDB es el sistema gestor de base de datos documental mas usado en la industria, vamos a mirar sus características, vamos a configurarlo, vamos a conectarlo y vamos a empezar a hacer algunas consultas básicas que nos van a permitir hacer un “crub” sobre una base datos de este tipo.

1. Bases de datos NoSQL

El concepto de bases de datos NoSQL está ampliamente divulgado en el entorno del desarrollo de software para identificar aquellos sistemas de bases de datos que no son gestionados bajo el estándar de SQL y/o no se pueden clasificar dentro de las bases de datos relacionales.

Estas bases de datos NoSQL son una forma de solucionar los problemas de diseño de bases de datos que no podían ser resueltos mediante los sistemas de bases de datos relacionales o que, en caso de ser implementados (los sistemas de bases de datos relacionales), representaban un manejo de estructuras demasiado complejas que sacrificaban el rendimiento de las aplicaciones que hacían uso de estas. En este sentido, las bases de datos NoSQL permiten resolver problemas en contextos específicos.

Entonces, analice las características de los sistemas de datos relacionales (SQL) frente a los no relacionales (NoSQL):

SQL

- I. Conjunto de reglas preestablecidas para la estandarización o normalización de los conceptos y relaciones existentes.
- II. Potencia las consultas y reportes de datos que se pueden obtener bajo el estándar de SQL.
- III. Información relacionada con un problema particular.

NoSQL

- I. Base de datos flexible, que ofrece estructuras de almacenamiento que no requieren ser normalizadas.
- II. No tiene un conjunto de reglas que preestablezcan los tipos de relaciones que definirán su comportamiento.
- III. Alto nivel de escalamiento y evolución superior.

Es importante tener en cuenta que, con la globalización de las aplicaciones, el uso masivo del Internet y las telecomunicaciones, la forma de procesar los datos se encuentra en constante evolución, lo que requiere de esquemas menos rígidos que permitan cambios de sus estructuras de una manera natural y que sean más fáciles de manejar y escalar a grandes volúmenes de datos; en estas nuevas circunstancias, muchos de los sistemas actuales necesitan tener un alto nivel de rendimiento y una alta escalabilidad, con necesidades de despliegue inmediato, por lo que se acoplan mejor a los sistemas de almacenamiento NoSQL.

El concepto NoSQL agrupa dentro de sí diferentes tipos de bases de datos, cada una de ellas con características muy diferentes y particulares, lo que brinda un abanico adicional de opciones que pueden adaptarse mejor a soluciones requeridas en diferentes entornos.

Según AWS (2021), las bases de datos NoSQL más representativas son las siguientes:

Bases de dato clave-valor

Este tipo es una de las bases de datos NoSQL más simples que existen. Su estructura, generalmente, se representa como una tabla de dos columnas:

Clave: en esta columna se realiza un único registro.

Valor: almacena los valores, que pueden ser de tipos de datos primitivos o complejos.

A continuación, se presenta un ejemplo de la estructura de este tipo de base de datos:

Figura 1. Representación de una base de datos Clave-Valor

Key1	AAA, BBB, CC
Key2	AAA, 05/05/2022
Key3	{ <u>name</u> : "abc", age:34}
Key4	188800051424
Key5	AAA, 6799, true
Key6	NAME

Este tipo de estructura de almacenamiento es similar a otros tipos de estructura que se utilizan en programación para el procesamiento de colecciones en memoria temporal, como son los diccionarios y las tablas Hash.

Como hemos indicado, son altamente flexibles, divisibles y permiten el escalado horizontal a niveles que otros tipos de bases de datos no pueden alcanzar. Algunos ejemplos de aplicaciones que usan este tipo de base de datos:

Aplicaciones para juegos.

Aplicaciones para tecnología publicitaria.

Aplicaciones de IoT.

En la siguiente tabla, se muestran algunos de los sistemas gestores de bases de datos NoSQL más famosos en el mercado, que permiten el uso de la estructura de almacenamiento clave-valor:

Tabla 1. Ejemplos de gestores de bases de datos tipo clave-valor

Nombres y logo	URL sitio oficial
 redis	https://redis.io
 CouchDB	https://couchdb.apache.org
 Project Voldemort	https://www.project-voldemort.com/voldemort/
 amazon DynamoDB	https://aws.amazon.com/es/dynamodb/
 riak	https://riak.com
 ORACLE[®] BERKELEY DB	https://www.oracle.com/co/database/technologies/related/berkeleydb.html
 AEROSPIKE	https://aerospike.com

Bases de datos documentales

El nombre documental hace referencia al tipo de estructura principal usada para representar la información, el cual se denomina documento.

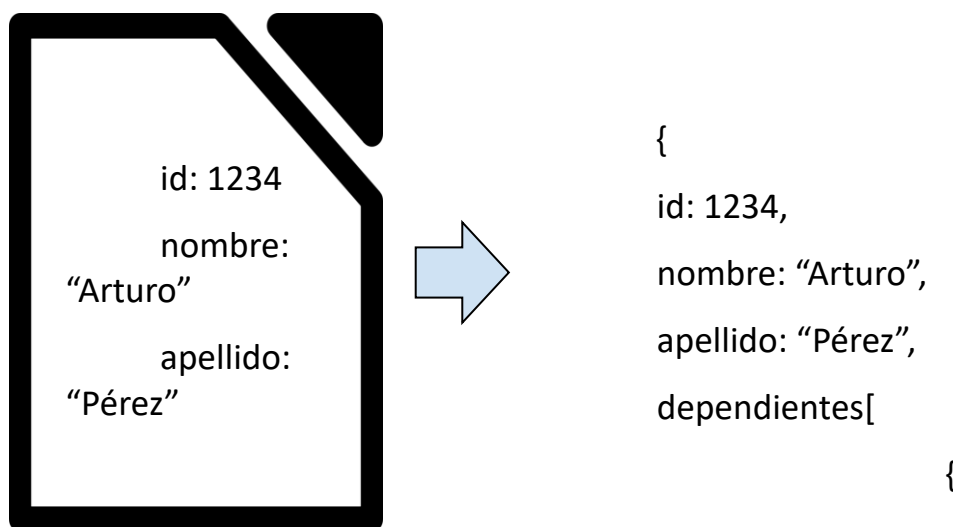
En este tipo de base de datos, la información no está contenida en tablas, sino que se compone de una estructura de datos semiestructurada, agrupada por colecciones de objetos o documentos de tipo JSON, XML, YAML o archivos de texto, los cuales, a su vez, pueden estar estructurados por diferentes tipos de datos primitivos o compuestos. Este es un modelo eficiente e intuitivo para los desarrolladores de “software”.

Estas bases de datos semiestructuradas facilitan la versatilidad, por lo que pueden evolucionar según las necesidades de las aplicaciones, y, entre sus ventajas, ofrece la posibilidad de gestión de información semiestructurada, que puede albergar gran cantidad de información con diversos tipos de datos; ya que no tiene un esquema formal o rígido, simplifican las tareas de inserción y actualización de datos.

Las aplicaciones web y móviles actuales están sometidas a gran cantidad de cambios en las estructuras de información que gestionan, por lo cual este tipo de bases de datos NoSQL se adecúan muy bien a este tipo de contextos.

A continuación, se observa un ejemplo de una posible estructura para una base de datos tipo documental.

Figura 2. Representación gráfica de bases de datos tipo documental








De acuerdo con la información ya proporcionada,

¿dónde es posible encontrar este tipo de base de datos?

Las aplicaciones que comúnmente utilizan estas representaciones son los catálogos, perfiles de usuario y los sistemas de administración de contenido.

En la siguiente tabla 2, se muestran algunos de los sistemas gestores de bases de datos NoSQL más famosos en el mercado, que permiten el uso de la estructura de almacenamiento documental. Recuerde que los vistos en la anterior tabla fueron los de clave-valor:

Tabla 2. Ejemplos de gestores de bases de datos tipo documental

Nombres y logo	URL sitio oficial
	https://www.mongodb.com
	https://ravendb.net
	https://www.marklogic.com
	https://azure.microsoft.com/es-es/services/cosmos-db/
	https://orientdb.org

Bases de datos de tipo grafos

Esta base de datos se caracteriza por representar la información por medio de vértices y aristas, lo que la asocia directamente con la teoría de grafos, ampliamente estudiada en la matemática y las ciencias de la computación.

Entonces,

¿qué se entiende por nodo o vértice?

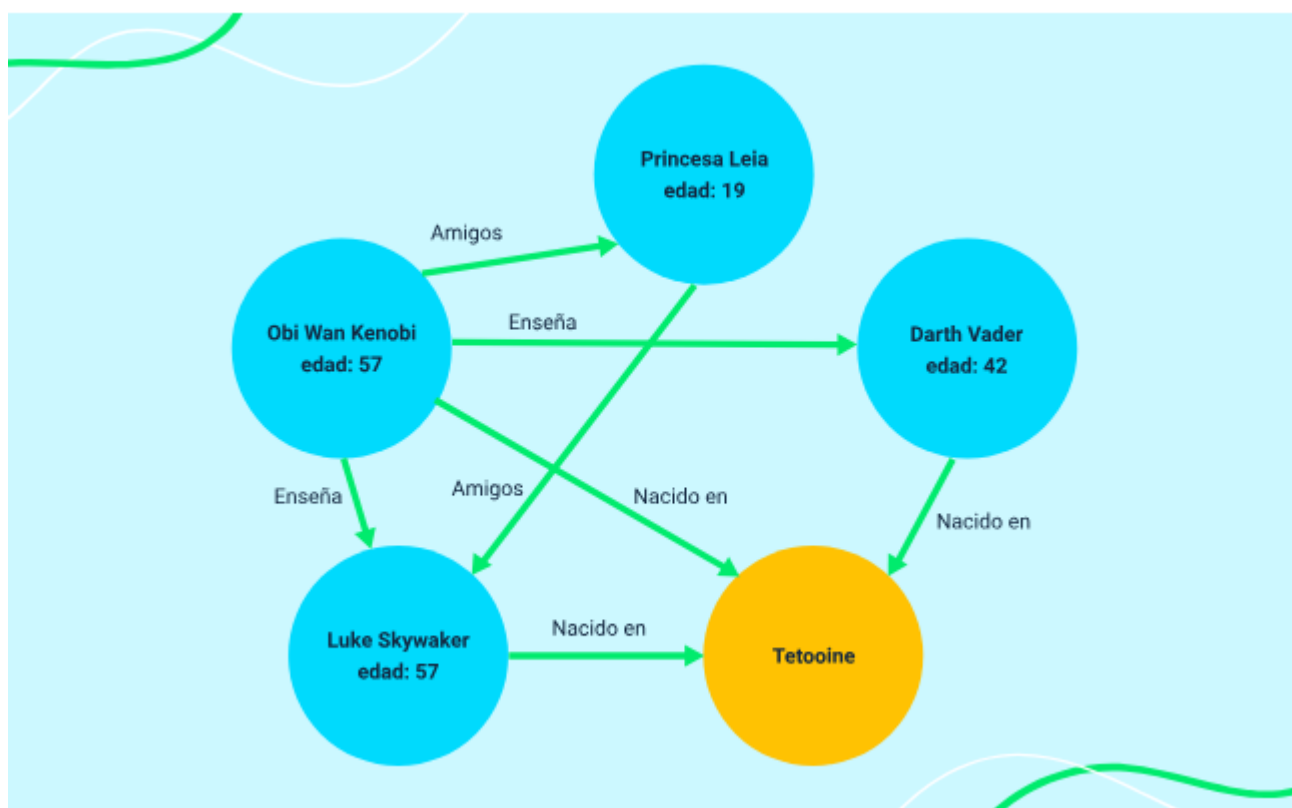
Nodo o vértice: en esencia, un vértice o nodo representa una entidad donde se almacena un conjunto de atributos que, normalmente, van representados con la estructura de clave – valor, y las aristas representan las relaciones que permiten representar la conexión y asociación entre los diferentes nodos.

Este tipo de base de datos carece de un esquema fijo y los nodos, a su vez, pueden tener diferentes tipos y cantidades de atributos, por lo que se dice que son multidimensionales.

Otra característica clave es que las relaciones pueden almacenar información adicional y no existen restricciones sobre la forma en la que se establecen dichas relaciones, es decir, pueden ser sin dirección, unidireccionales o bidireccionales. Esta característica le permite tener un altísimo rendimiento en la búsqueda de resultados y el descubrimiento de nueva información.

A continuación, un ejemplo gráfico de la representación de una base de datos de tipo grafo:

Figura 3. Representación gráfica de bases de datos tipo grafo








Nota. Tomado de Almagro (2017).

Este tipo de base de datos facilita la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos son los que incluyen redes sociales, motores de recomendaciones, detección de fraude y gráficos de conocimiento.

Ya vimos algunos de los sistemas gestores de bases de datos NoSQL más famosos en el mercado, que permiten el uso de la estructura tipo clave-valor y de almacenamiento documental; ahora, es momento de ver los de tipo grafo:

Tabla 3. Ejemplos de gestores de bases de datos tipo grafo

Nombres y logo	URL sitio oficial
	https://neo4j.com/
	https://www.tigergraph.com/
	https://www.graphengine.io/
	https://www.arangodb.com/
	https://titan.thinkaurelius.com/

Paralelo entre SQL y NoSQL

Las bases de datos SQL y NoSQL cumplen con la labor de gestionar información y existe gran cantidad de sistemas gestores de bases de datos en el mercado, que facilitan este proceso con altos estándares de calidad. Sencillamente, estos dos modelos utilizan estructuras de almacenamiento y estándares de acceso a la información que son transversalmente diferentes. El uso o no de un modelo SQL o NoSQL dependerá del contexto de la solución que se busca resolver y de las necesidades de infraestructura requeridas por el cliente.

A continuación, se hace un paralelo de algunas de las características más importantes que diferencian un sistema SQL de un sistema NoSQL (AWS, 2021).

Modelos de datos

SQL

Presenta un esquema rígido conformado por un conjunto de tablas y relaciones; estas tablas se descomponen en columnas y filas, en las que se utilizan índices para agilizar los procesos de recuperación de información.

Se siguen procesos preestablecidos de normalización que permiten imponer restricciones de integridad referencial de forma muy exitosa.

NoSQL

Existe gran variedad de sistemas de representación, entre los que se destacan los clave-valor, documentales y grafos, que no se ciñen a un esquema rígido, por lo cual se garantiza la integridad referencial (esta labor recae sobre los desarrolladores).

Permiten la optimización para el rendimiento y el crecimiento a gran escala.

Propiedades ACID (Atomicidad, Coherencia, Aislamiento y Durabilidad)

SQL

Garantizan completamente estas cuatro propiedades:

1. La atomicidad de las transacciones tiene garantía de ejecución al 100% o no se ejecutan en absoluto.
2. La coherencia se garantiza a través de sus esquemas preestablecidos, por ello, los datos deben cumplir las reglas definidas, de lo contrario, no serán aceptados.
3. El aislamiento se mantiene, ya que tiene sistemas que permiten la ejecución simultánea de transacciones de forma aislada.
4. La durabilidad se genera con sistemas complementarios que permiten recuperarse frente a casos inesperados de errores.

NoSQL

Las características de atomicidad, coherencia, aislamiento y durabilidad se flexibilizan para ofrecer el escalamiento horizontal, lo cual hace que estos sistemas sean la mejor opción para casos de baja latencia y alto rendimiento.

Cargas óptimas de trabajo

SQL

Presentan un alto procesamiento de transacción online (OLTP) y de procesamiento analítico online (OLAP).

NoSQL

Este tipo de datos es efectivo para sistemas de baja latencia y para el análisis de datos semiestructurados.

API

SQL

Utilizan el lenguaje de consulta SQL, que se ejecuta del lado del servidor de base de datos, para almacenar y recuperar la información.

NoSQL

Son óptimas para ser procesadas por APIS construidas por los desarrolladores.

2. MongoDB

MongoDB es uno de los sistemas gestores de bases de datos NoSQL de tipo documental más famosos del mercado; su nombre viene del término en inglés “humongous”, que significa gigantesco. Ofrece un conjunto de soluciones, entre las que se incluye la posibilidad de trabajar con servidores que se pueden descargar y configurar en el equipo físico o servicios alojados en la nube por medio del gestor Atlas.

Este gestor de bases de datos NoSQL fue escrito en lenguaje C ++, pero todas las consultas usadas para la gestión de la información utilizan el formato BSON, que es similar al formato JSON, ampliamente utilizado por los desarrolladores web y móviles, para los procesos de transferencia de información a través de Internet.

Se presenta, entonces, la forma de hacer su instalación; luego, su configuración; y finalmente, su administración:

Instalación de MongoDB

A continuación, en el video, se muestra la forma en la que se realiza una instalación exitosa de una instancia gratuita de MongoDB y el servicio de gestión Cloud Atlas, accediendo a la dirección <https://www.mongodb.com/es>

Video 2. Instalación MongoDB



Instalación Mongo DB

Síntesis del video: Instalación Mongo DB

Hola nuevamente, el día de hoy vamos a mostrar como hacer el proceso de instalación de una instancia de Mongo DB o más bien el registro de un servicio de Mongo DB, entonces podemos ir al buscador cualquiera, podemos usar cualquiera de los navegadores que tengamos disponibles y vamos a entrar a MongoDB.com, estando en la interfaz de MongoDB automáticamente nos saludan con el servicio de atlas, nos habla de que es un servicio que esta alojado en la nube y que tiene unos servicio provistos por algunas plataformas reconocidas como AWS, Azure, y Google cloud. bueno vamos a iniciar de manera gratuita acá nos pide iniciar con un correo

electrónico, podemos diligenciar la información, comienza por un usuario, un nombre, un apellido y vamos a definir un password, esto que estamos creando acá simplemente son las credenciales de acceso general al servicio; aceptamos las condiciones, una vez hacemos el proceso de registro del usuario, es muy importante que este correo electrónico sea un correo válido, un correo que exista ya que la información nos va a llegar directamente al correo electrónico y la plataforma me dice que al usuario del correo electrónico que yo registre va a llegar un link de verificación que va a expirar en dos horas, entonces abrimos nuestro correo electrónico voy a actualizar acá, acá esta MongoDB cloud aquí verifico que es un correo que viene de MongoDB doy verificación, esto expira en dos horas voy a verificar el e-mail y listo, satisfactoriamente verificado el e-mail, damos continuar, aquí voy a iniciar con mi correo electrónico registrado, aquí voy a colocar las credenciales de acceso que ya registré en el formulario de registro que acabamos de hacer.

Listo entonces aquí la plataforma de Mongo me da la bienvenida, inicialmente el lo que va a hacer es preguntarme inicialmente para que tipo de proyectos es que yo requiero crear este servicio de MongoDB, entonces en primer lugar nos da las instancias, esto es totalmente informativo esto no va a restringir el servicio, voy a decirle que voy a aprender Mongo DB, el tipo de aplicación que voy a construir, como las aplicaciones que yo construyo basadas en SQL, están pensadas a ser manipuladas y gestionadas por ápice externas entonces acá me pregunta que tipo de plataforma es la que yo deseo desarrollar utilizando este tipo de estructuras de base de datos no relacional, aquí se menciona lo que se llama afinami y finalmente el lenguaje, el lenguaje es simplemente para pre definir una vez que uno vaya a esclarecer la

conexión, cuales son las cadenas de conexiones que debería utilizar para este lenguaje, voy a utilizar JavaScript y damos en finalizar.

Antes de terminar como eso es un servicio basado en cloud, entonces me ofrecen diferentes tipos de servicios, entonces hay tres tipos de servicios, serverless dedicados, que tienen unos costos, aquí en la misma interfaz yo puedo verificar los costos asociados a cada uno de esos servicios y el servicio compartidos básicamente nos va a crear un clúster de servicios que son unas máquinas que van a compartir memoria y van a compartir recursos con otros usuarios, está la versión gratuita que es la que yo voy a utilizar. Viene la versión gratuita pues acá me permite seleccionar cual va a ser el Cloud Provider que puede ser aws, Google cloud o Azure, en este caso es aws voy a seleccionar una de las regiones, generalmente pues uno selecciona la que este más cerca a su país de origen en este caso voy a seleccionar Norteamérica me parece que son los más estables voy a dejarle la versión por defecto; también aquí me va a recordar que en la versión gratuita yo voy a tener un servicio que va a estar compartido y con un tamaño de almacenamiento de 512 MG inicialmente si yo quiero adquirir servicio adicionales posterior a esto se puede ampliar sin problemas.

Y finalmente acá si quiero yo puedo manejar el nombre del clúster, recordemos que el clúster como esto son servicios manejados en nube van a ver un conjunto de servidores que van a estar alojados en alguna parte interconectados, esos servidores conectados van a poder proveer el servicio a muchos usuarios ya que estamos en el servicio compartido, y yo puedo nombrar el clúster como quiera yo lo voy a dejar por defecto clúster cero y le digo que clúster.

Cos esto terminaría la configuración inicial automáticamente el me lleva al Daswoord de MongoDB, me loguea, aquí quedaría pendiente hacer ya lo que es la configuración de seguridad y la configuración relacionada con la conexión; acá como estamos adquiriendo unos servicios gratuitos él se demora un tiempo aproximadamente entre 3 y 5 minutos para terminar de proveer los servicios que hemos configurado hasta el momento.

Entonces resumiendo creamos una cuenta, registramos un usuario, una contraseña, verificamos el e-mail, esto automáticamente nos da acceso al servicio una vez en el servicio configuramos que tipo de servicio voy a adquirir si es dedicado, compartido o serlerves en caso compartido que es la versión gratuita, definimos el nombre del clúster, definimos la región a la cual nos vamos a conectar y quedamos listos con la configuración inicial de MongoDB.

Espero que haya sido bastante clara la información, la interfaz puede variar normalmente este servicio se esta actualizando constantemente la página es muy estable, pero en términos generales estos son los pasos que yo debo realizar para configurar el servicio.

Configuración de MongoDB

Posteriormente, el sistema ingresará al “dashboard” de configuración general de MongoDB, Atlas, en el que se recomienda realizar las configuraciones iniciales de seguridad. En este espacio, se establecen las credenciales de acceso necesarias para la

gestión de la base de datos. Para conocer cómo se realiza este proceso, le invitamos a revisar el siguiente video:

Video 3. Configuración MongoDB



Configuración MongoDB

Síntesis del video: Configuración Mongo DB

Hola, espero hayas podido finalizar exitosamente el registro en el servicio de MongoDB, en este video vamos a complementar lo que sería el siguiente paso una vez se ha creado el clúster para trabajar MongoDB y corresponde a la configuración de los elementos de seguridad y de accesibilidad al servicio, entonces en primer lugar estamos acá en el dashboard, quería mostrarte que aunque yo definí inicialmente un servicio que era compartido y gratuito y tiene unas limitantes, yo puedo directamente desde dashboard en la selección de bases de datos en la parte superior derecha una opción de crear, esta que aparece acá que me permite a mi ampliar el servicio que ya tengo actualmente asignado entonces si yo eventualmente estoy haciendo mis prácticas, ya me di cuenta que el servicio me sirve para mis aplicaciones, yo puedo entrar acá y configurar nuevos clúster, o cambiar la configuración de los clúster que ya

tengo definidos para cambiarlos por ejemplo a servicios dedicados o ampliar la capacidad que inicialmente tengo asignada. Entonces yo puedo hacerlo a través de este mismo espacio y si es el caso algunos servicios tendrán costos adicionales y otros no, entonces yo puedo explorar a través de esta misma opción directamente desde el dashboard.

Ahora volviendo a la configuración básica, notaremos que, en la parte inferior izquierda, hay una opción que dice Get started básicamente lo que está haciendo este recordatorio, es indicarme cuales son los pasos que he hecho hasta el momento. En este momento aparece seleccionado que ya tengo mi primer clúster, ahora vamos a configurar todas las opciones de seguridad que se necesitan para poder empezar a hacer conexiones a la base de datos.

En la parte izquierda hay un menú que esta dividido por varios fragmentos, entonces la opción de seguridad hay una opción que dice Quickstar en esta opción quickstar que es lo que vamos a hacer el día de hoy; que es configurar dos elementos en primer lugar quienes van a ser las credenciales de acceso a esta base de datos y desde donde voy a poder conectarme a esta base de datos; entonces acá lo que me esta pidiendo es las credenciales de un usuario nuevo, este usuario es diferente al que hicimos en el registro, en el registro creamos un usuario para entrar a administrar los servicios que me esta proveyendo MongoDB, todo lo que tenga que ver con clúster, manejo de usuarios, es decir, es como un super administrador de todo el servicio.

Lo que estamos registrando acá es el usuario que efectivamente se va a conectar con las bases de datos, recordemos que las bases de datos NoSQL están

pensadas para ser consumidas y administradas por ápicos externos esas APIs externas o servicios externos se van a conectar a esta base de datos a través de un usuario que no va a ser el super administrador, sino que va a ser un usuario que es el que vamos a definir en este momento.

Entonces aquí voy a colocar usuario uno, por ejemplo, y vamos a administrarle un password, estas credenciales que hemos creado acá son las que usaré en ese servicio externo API que se va a conectar a mi MongoDB.

Entonces creamos el usuario listo ya quedó registrado, puedo registrar tantos usuarios como yo quiera, puede ser que mi aplicación a través de diferentes servicios acceda y con diferentes tipos de usuarios, entonces puedo manejar múltiples usuarios con sus respectivas credenciales.

También puedo utilizar la opción de certificados si lo tenemos accesible.

Y en segundo lugar es muy importante desde donde voy a establecer la conexión con el motor de MongoDB, ahora mismo como estamos iniciando seguramente vamos a trabajar en nuestra máquina local cierto? Entonces yo voy a colocar acá la IP de mi máquina local, puedo entrar a un servicio de internet por ejemplo en Google y buscar cuál es mi IP real cierto en este momento y registrarla o puedo utilizar la opción de *Find My Current IP Address*, el me lo permite identificar y lo que estoy diciendo acá es ahora esta dirección IP desde donde estoy si se va poder conectar al servicio, hace parte de las opciones de seguridad.

Ahora acá hay algo muy importante, cuando arrancamos normalmente hacemos configuraciones siempre en local en nuestra máquina local, puede ser que más adelante cuando yo cree una aplicación un poco más robusta un servicio o una

api, eso yo lo aloje en un servicio en internet, puedo montar por ejemplo en un servidor Azure, en un GOQ, varias plataformas que me permiten a mi publicar esos servicios y de ese servicio quisiera consumir la base de datos, entonces en ese momento tendré que consultar cuales son las direcciones IP donde queda alojado el servicio y registrarlas acá, si no lo hago lo que va a pasar es que no va a poder establecer ese servicio que tengo ya construido a mi base de datos; entonces es muy importante registrar siempre acá las direcciones IP de los servicios o de las ubicaciones donde se va a acceder a la base de datos.

Ya que tengo una opción un poco más avanzada si tengo un ambiente cloud cierto, entonces también puedo especificar acá una lista de direcciones IP para hacer esa configuración de acceso.

Una vez terminamos esto damos finish and clouse, nos dice felicitaciones ya acabamos de definir las reglas de acceso ahora si ya podemos empezar a trabajar con nuestra base de datos.

En este momento no hay ninguna base de datos, pero ya tenemos nuestro clúster, tenemos definidas las reglas de seguridad, lo que queda es construir la base de datos.

Yo recomendaría que inicialmente trabajemos con la base de datos de prueba si lo desea, aquí hay una opción de base de datos en el clúster que tenemos trabajando, que si le doy clic en los punticos me aparece una opción que dice cargar datos de prueba; yo recomendaría inicialmente para irnos adecuando y conociendo las estructuras y como se maneja las conexiones y como se manejan por ejemplo las consultas, cargar estos datos de prueba.

Es una base de datos que es bastante robusta tiene bastantes elementos con los que se puede hacer bastantes prácticas, son 350 MG, acá hay diferentes tipos de estructuras yo recomendaría mucho en nuestra primera instalación hacer uno de esta base de datos, vamos a cargarla en este ejemplo recordemos que esto es algo totalmente opcional, el se demora un tiempo en cargarlo y ya quedaría pendiente solo lo que tiene que ver con lo respecto a la conexión a la base de datos.

Administración de MongoDB

Una vez terminada la configuración básica de seguridad, podrá acceder al “dashboard” para la administración general de la base de datos NoSQL. Tenga en cuenta que este servicio de base de datos se encuentra administrado por el proveedor Cloud seleccionado y que la base de datos se alberga en un clúster. Para conocer cómo se realizan los pasos de administración, revise lo siguiente:

Paso 1: si es la primera vez que intenta manipular bases de datos NoSQL documentales, se recomienda seguir las sugerencias de Atlas y crear una base de datos de prueba para realizar operaciones y consultas de prueba sobre la misma, a manera de entrenamiento. Esto puede tardar un par de minutos adicionales.

Data base

“Load Sample”

Paso 2: dentro de la misma información de la base de datos, podrá acceder a las opciones de conexión al clúster. Recuerde que los sistemas de bases de datos NoSQL están concebidos para ser administrados por APIS externas construidas por los

desarrolladores; por esta razón, el sistema le preguntará el medio de conexión preferido.

“Connect”

Paso 3: si desea conectarse por medio de una terminal de comandos (MongoDB “Shell”), le serán recomendados, de acuerdo al sistema operativo, los comandos o instalaciones previas a realizar, junto con la cadena de conexión que deberá ejecutar en su terminal para establecer una conexión exitosa con la base de datos.

“Connect”

“Connect”

Paso 4: una vez finalizada la instalación del “Shell” de Mongo, pruebe utilizar el comando de conexión provisto por Atlas; tenga en cuenta que esta cadena de conexión utilizará el usuario creado anteriormente en la configuración de seguridad.

“Connect”

Paso 5: si lo desea, también puede explorar y gestionar las estructuras creadas en la base de datos directamente desde el portal de MongoDB Atlas, por medio de la opción “Browse Collections”.

“Connect”

Paso 6: el sistema desplegará todas las bases de datos en un panel del lado izquierdo, en el que podrá seleccionar alguna de estas y consultar las colecciones existentes. En el panel central, tendrá un espacio de ejecución de comandos sobre la base de datos seleccionada, y en la parte inferior, se desplegará el resultado de la

ejecución de los comandos. Por defecto, desplegará todos los documentos existentes si es que tiene seleccionada una colección de la base de datos.

“Connect”

2.1. Documentos

Como ya se ha mencionado, MongoDB es un gestor de base de datos NoSQL, con una estructura basada en documentos. Por ello, es importante aclarar la jerarquía completa de esta estructuración de la información, la cual se muestra a través de la siguiente información:

- A. Clúster:** en primer lugar, se encuentra el clúster, que es una agrupación de equipos de cómputo llamados nodos, que trabajan en forma conjunta para la gestión de la información presente en la base de datos, es decir, este primer nivel corresponde primordialmente a infraestructura tecnológica.
- B. Conjunto de elementos:** las bases de datos se almacenan en los clústeres y representan el conjunto de elementos que sirven para la gestión de información correspondiente a una solución de software particular y toda la información asociada a estos elementos, incluyendo los mismos datos. En un clúster, pueden existir varias bases de datos, cada una de las cuales responderá a soluciones de software diferentes.
- C. Colecciones:** dentro de las bases de datos, se registran las colecciones que agrupan a un conjunto de documentos que representan una misma unidad de información. Si se compara con una base de datos relacional, es un documento que corresponde a una tabla.

D. Estructura documental: los documentos representan el núcleo de una estructura documental con información que corresponde a un objeto u entidad del mundo real perteneciente al contexto de la aplicación. En un documento, se estructuran, de forma dinámica, los atributos (campos) que caracterizan dichos objetos descritos estructuralmente como pares clave-valor.

Nuevamente, si se realiza una comparación con el modelo relacional, un documento equivale a una tupla o fila en una tabla.

Para un mejor entendimiento de esta jerarquía, la siguiente figura presenta gráficamente la relación de los elementos ya descritos:

Figura 4. Jerarquías de almacenamiento en MongoDB



Los documentos son abstracciones de una entidad u objeto del mundo real que expresa sus características por un conjunto de pares clave-valor. La clave o nombre del campo siempre son cadenas de texto y los valores pueden ser de cualquier tipo de dato primitivo o tipos de datos complejos.

A continuación, se presentan algunos ejemplos de documentos caracterizados por diferentes tipos de datos:

Tipo 1.

```
[  
  "nombre": "Jonathan",  
  "apellido": "Guerrero",  
  "id": "147823459",  
  "edad": 32  
]
```

Este documento representa a una persona con los atributos de nombre, apellido, id y edad. Note que cada atributo (campo) se expresa con la estructura clave-valor, donde la clave es una cadena de texto y el valor es representado por un tipo de dato primitivo ("String y Int").

Tipo 2.

```
{ "name": John,  
  "age": 30,  
  "cars": [  
    "Ford",  
    "BMW", "Fiat"  
  ]  
}
```

```
}
```

Este documento representa los datos básicos de una persona y los tipos de carros que le gustan. Note que en este ejemplo, la clave “cars” tiene asociado un tipo de dato compuesto, una colección de “Strings”. Las colecciones dentro de un documento se agrupan por medio de corchetes [,].

Tipo 3.

```
{ "name": "John", "age": 30,  
  "cars":  
  [  
    { "name": "Ford", "models": [ "Fiesta",  
      "Focus", "Mustang" ] },  
    { "name": "BMW", "models": [ "320", "X3",  
      "X5" ] },  
    { "name": "Fiat", "models": [ "500",  
      "Panda" ] }  
  ]  
}
```

Este documento, a diferencia de los anteriores, asocia como valor a la clave cars otro nuevo documento que está representado, a la vez, por los pares clave valor “name” y “models” que es una colección de “Strings”.

Note que, a diferencia de las colecciones, los tipos documentos se agrupan por los caracteres llaves { , }.

2.2. Comandos básicos

Los comandos básicos que se pueden utilizar para la manipulación y gestión de las bases de datos en MongoDB pueden ser ejecutados una vez se establece la conexión con el servidor, ya sea desde su cliente web o desde una aplicación externa.

A diferencia de las bases de datos relacionales, en MongoDB no existe una separación de comandos para la creación de estructuras como en el caso de las sentencias DDL de SQL y comandos para la manipulación de las estructuras creadas como las sentencias DML en SQL. Este tipo de base de datos busca tener mayor flexibilidad y adaptabilidad, por lo que una consulta sobre una estructura inexistente puede hacer que el motor de base de datos cree dicha estructura de forma automática.

Los comandos en MongoDB se pueden escribir tanto en una sola línea como ordenados por medio de espacios, saltos de línea o indentación, para hacer más legible su estructura, lo que no afecta la interpretación del motor de base de datos de MongoDB.

comandos básicos más utilizados en MongoDB

Le invitamos a conocer los [comandos básicos más utilizados](#) junto con su descripción:

2.3. Operadores

En las bases de datos relacionales, los comandos básicos pueden ampliarse para elaborar consultas mucho más complejas, además de restringir de una manera más

adecuada. En MongoDB, estos elementos se procesan por medio de operadores, que se clasifican en cuatro tipos:

Operadores de comparación

Los operadores de comparación, como su nombre lo indica, sirven para restringir las condiciones que deben ser evaluadas por el motor para determinar el conjunto de datos que puede ser afectado por el comando. Normalmente, son utilizados para ejecutar comandos de consulta o de manipulación de información en las colecciones.

En la siguiente tabla, se muestran los operadores de MongoDB, su equivalente en un ambiente relacional y su significado.

Tabla 4. Operadores de comparación en MongoDB

Operador MongoDB	Operador relacional equivalente	Significado
\$eq	=	Igual que ("equal")
\$ne	<>	Distinto de ("not equal")
\$gt	>	Valores mayores que ("greater than")
\$gte	>=	Valores mayores o iguales que ("greater than or equal")
\$lt	<	Valores menores que ("lower than")
\$lte	<=	Valores menores o iguales que ("lower than or equal")
\$in	IN	Sus valores se encuentran dentro del "array" especificado

Operador MongoDB	Operador relacional equivalente	Significado
\$nin	NOT IN	Operación contraria a \$in, obtiene los documentos cuyo valor NO estén en el “array”

Operadores de elemento

Estos operadores existen para identificar si una clave/campo se define dentro de su estructura y valida el tipo de dato asociado a un campo particular, esto teniendo en cuenta que las estructuras en MongoDB se construyen y evolucionan de forma dinámica, ya que no tienen un esquema rígido de almacenamiento.

MongoDB utiliza el operador \$exists para determinar la existencia de un campo dentro de la definición de un documento en una colección y el operador \$type para comprobar el tipo de dato asociado a un campo particular.

Observe algunos ejemplos del uso de este tipo de operadores:

Ejemplo 1:

Comando con operadores de elemento

```
db.productos.find({"nombre":{"$exists:true"}}).pretty()
```

```
db.productos.find({"nombre":{"$exists:false"}}).pretty()
```

Interpretación

En este comando, se están consultando los documentos de la colección productos, donde existe o no el campo “nombre”. Luego, pretty() formatea el resultado en una estructura tipo JSON, fácil de interpretar visualmente.

Ejemplo 2:

Comando con operadores de elemento

```
db.productos.find({"id":{"$type":"String"}}).pretty()
```

Interpretación

En este comando, se están consultando los documentos de la colección productos, donde existe o no el campo “id”, que corresponde al tipo de dato primitivo String. Luego, pretty() formatea el resultado en un estructura tipo JSON, fácil de interpretar visualmente.

Si existiera algún documento con id de tipo int o cualquier otro diferente de String, no será incluido en la consulta anterior.

Operadores lógicos

Estos operadores permiten crear condiciones más complejas en el proceso de manipulación de datos en colecciones de una base de datos MongoDB; funcionan y se interpretan igual que los operadores lógicos de cualquier lenguaje de programación y siguen la lógica proposicional con todas sus tablas de verdad.

Observe, en la siguiente tabla, algunos de los operadores lógicos más utilizados y su equivalente:

Tabla 5. Operadores lógicos en MongoDB

Operador MongoDB	Operador lógico equivalente	Ejemplo
\$or	OR	<p>Filtra de los documentos de la colección productos que cumplan alguna de las condiciones del array</p> <pre>db.productos.find({ \$or: [{"id":{"\$exists":"true"}}, {"valor":{"\$gt":10}}] }).pretty().</pre> <p>En este caso, las dos condiciones a evaluar son que el campo id exista {"id":{"\$exists":"true"}} o que el campo valor sea mayor que 10 {"valor":{"\$gt":10}}</p>
\$and	AND	<p>Similar a \$or, pero, en este caso, los documentos deben cumplir todas las condiciones del array</p> <pre>db.productos.find({\$and: [{"id":{"\$exists":"true"}}, {"valor":{"\$gt":10}}] }).pretty().</pre> <p>Su uso puede resultar superfluo, teniendo en cuenta que en un filtro normal las condiciones indicadas se juntan mediante operadores lógicos Y</p> <pre>db.productos.find({"id":{"\$exists":"true"},"valor":{"\$gt":10}}).pretty()</pre>

Operadores sobre “arrays”

Como los documentos pueden tener valores complejos que pueden ser de tipo lista, de datos primitivos o de listas de otros tipos de documentos, es muy útil usar operadores que permitan validar la información dentro de estas listas de elementos. De ahí la importancia de los operadores de “arrays” que facilitan esta labor.

A continuación, conozca algunos de los más comunes:

I. Operador

\$all

Descripción y ejemplo

El “array” debe contener todos los valores indicados en el parámetro, que a su vez es un “array”

```
db.productos.find({"ciudad":{"$all:["cali","bogota"]}}).pretty()
```

II. Operador

\$size

Descripción y ejemplo

Especifica el tamaño del array

```
db.productos.find({"ciudad":{"$size:2}}).pretty()
```

III. Operador

\$elemMatch

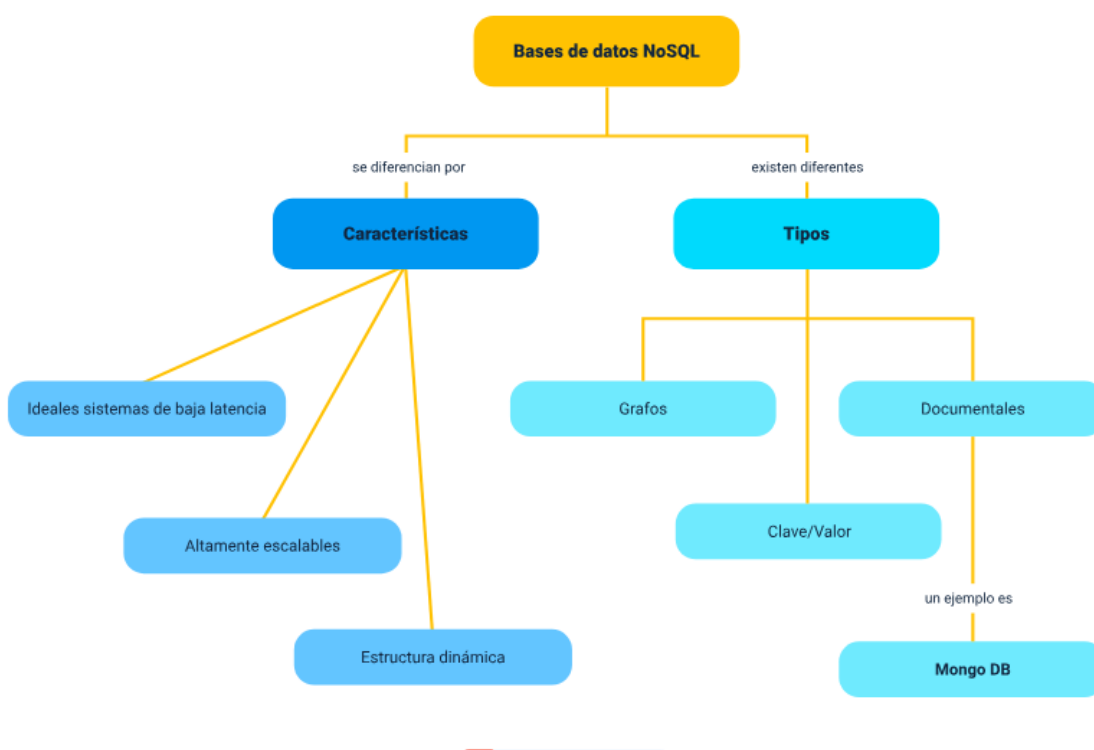
Descripción y ejemplo

Cuando los documentos principales contienen “arrays” que, a su vez, contienen documentos, \$elemMatch sirve para filtrar los documentos principales cuyo campo array contiene documentos que cumplen todos los criterios especificados en el documento pasado como parámetro.

```
{ "alumno": { $elemMatch: { "nombre" : "Maria", "edad": 23 } }}
```

Síntesis

En la actualidad, se presenta un universo muy amplio en variedad de tecnología para bases de datos, como es el caso de NoSQL que se creó con la finalidad de dar respuesta a las necesidades de desarrollo detectadas asociadas a estas aplicaciones actuales; en este sentido el presente componente formativo tiene como objetivo el abordaje de temas como las bases de datos no relacionales y la plataforma MongoDB. Estos temas se resumen a través del siguiente mapa conceptual, esto es:



Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
MongoDB	MongoDB. (s. f.). <i>Welcome to the MongoDB Documentation.</i>	Página web	https://www.mongodb.com/docs/

Glosario

Clúster: agrupación de varios equipos de cómputo especializados para ofrecer servicios.

Escalamiento horizontal: permite el crecimiento de una manera muy amplia y técnicamente sin límites. Implica infraestructura de varios servidores, conocidos como nodos, que trabajan como una única unidad y pueden equilibrar cargas y agregar más componentes a su estructura, robusteciendo su capacidad sin afectar sus servicios.

Infraestructura tecnológica: equipos de cómputo, sistemas de red y cualquier otro componente de tipo hardware requerido para el montaje y puesta en marcha de un sistema software.

Shell: ventana de comandos.

Referencias bibliográficas

Almagro-Blanco, P. y Sancho-Caparrini, F. (2017). *Generalized Graph Pattern Matching*. Cornell University. <https://arxiv.org/abs/1708.03734>

Amazon Web Services. (s. f.). *¿Qué es NoSQL?* AWS.

Graph Everywhere. (2021). *Las top 10 Bases de datos nativas de grafos*. <https://www.grapheverywhere.com/las-top-10-bases-de-datos-nativas-de-grafos> .

Créditos

Nombre	Cargo	Regional y Centro de Formación
Claudia Patricia Aristizábal	Líder del equipo	Dirección General
Liliana Victoria Morales Gualdrón	Responsable de línea de producción	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Jonathan Guerrero Astaiza	Experto temático	Centro de Teleinformática y Producción Industrial - Regional Cauca
María Fernanda Chacón Castro	Diseñadora instruccional	Centro de Gestión Industrial - Regional Distrito Capital
Andrés Felipe Velandia Espitia	Asesor metodológico	Centro de Diseño y Metrología - Regional Distrito Capital
Rafael Neftalí Lizcano Reyes	Responsable equipo desarrollo curricular	Centro Industrial del Diseño y la Manufactura - Regional Santander
Darío González	Corrector de estilo	Centro de Diseño y Metrología - Regional Distrito Capital
Alix Cecilia Chinchilla Rueda	Metodología para la formación virtual	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Yuly Andrea Rey Quiñonez	Diseñador web	Centro de Gestión de Mercados, Logística y Tecnologías de la

Nombre	Cargo	Regional y Centro de Formación
		Información - Regional Distrito Capital
Jhon Jairo Urueta Álvarez	Desarrollador fullstack	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Ernesto Navarro Jaimes	Animación y producción audiovisual	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Lady Adriana Ariza Luque	Animación y producción audiovisual	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Laura Gisselle Murcia Pardo	Animación y producción audiovisual	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Carolina Coca Salazar	Evaluación de contenidos inclusivos y accesibles	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Lina Marcela Pérez Manchego	Validación de recursos educativos	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Leyson Fabián Castaño Pérez	Validación y vinculación al LMS	Centro de Gestión De Mercados, Logística y Tecnologías de la

Nombre	Cargo	Regional y Centro de Formación
		Información - Regional Distrito Capital