INSTITUTO SUPERIOR TECNOLÓGICO QUITO

Formamos tu PROPÓSITO DE VIDA

GUÍA GENERAL DE

BASE DE DATOS NO RELACIONALES

AUTOR: MÓNICA RAMIREZ

ISBN: 978-9942-562-33-3

9 789942 562333

ITQ

WWW.ITQ.EDU.EC

INVESTIGACIÓN

DOXA EDITION

INSTITUTO SUPERIOR TECNOLÓGICO QUITO
Excelencia en Educación Superior

GUÍA GENERAL DE BASE DE DATOS NO RELACIONALES

AUTOR: MÓNICA ALEXANDRA RAMÍREZ VELASTEGUI

EDICIÓN: PRIMERA EDICIÓN

AÑO: 2025

TRABAJO EN EDICIÓN:

DOXA EDITION

EQUIPO EDITORIAL:

MGS. CAMILA DÍAZ DÍAZ

MGS. KEYERMAN TOAPANTA CISNEROS

ISBN: 978-9942-562-33-3

9 789942 562333

SOBRE LA AUTORA

Mónica Ramírez es una profesional con más de 10 años de experiencia en el apasionante mundo de las Tecnologías de la Información (TI). A lo largo de su carrera, se ha destacado por liderar proyectos innovadores y fomentar una comunicación efectiva en equipos multidisciplinarios. Su amor por la educación la llevó a obtener una Maestría en Gerencia y Liderazgo Educativo, y la Maestría en Inteligencia Artificial Aplicada, lo que refleja su compromiso con el aprendizaje continuo.

Con más de 8 años de experiencia como docente universitaria, Mónica ha dejado una huella significativa en el desarrollo académico y personal de sus estudiantes. Su enfoque pedagógico busca formar profesionales íntegros, combinando conocimientos técnicos con habilidades interpersonales esenciales para el ámbito laboral.

La sólida trayectoria de Mónica en TI, junto con su interés en áreas emergentes como la Inteligencia Artificial, le permite ofrecer una educación actualizada y relevante. Ella cree firmemente que la educación tiene el poder de transformar vidas, preparándolos para enfrentar los desafíos del futuro con confianza, conocimiento y creatividad.

Su carrera comenzó como Ingeniera de Software, donde adquirió habilidades técnicas esenciales. Luego, se desempeñó como Analista de Software y Administradora de Base de Datos, así como Analista de Ciencia de Datos, aplicando técnicas de análisis para extraer información valiosa que respalda la toma de decisiones estratégicas. Su capacidad de liderazgo la llevó a convertirse en Gerente de Desarrollo de Software, donde impulsó proyectos innovadores y mejoró la eficiencia de los equipos. Además, ha realizado importantes contribuciones en el ámbito educativo, trabajando como Analista Técnica en evaluación educativa a nivel nacional y como Docente Investigador.

Gracias a su experiencia en el sector público y privado, así como a sus valiosas contribuciones a la educación y su liderazgo académico, Mónica se ha consolidado como una referente en el ámbito de las Tecnologías de la Información y la Educación. Su compromiso con la excelencia y su anhelo de influir positivamente en las futuras generaciones de profesionales en tecnología son palpables en su trayectoria y en los logros que ha conseguido a lo largo de su carrera.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

CONTENIDO

GUÍA GENERAL DE BASE DE DATOS NO RELACIONALES 7
DESCRIPCIÓN DE LA ASIGNATURA 7
1. BIBLIOGRAFÍA 8
1.1. Básica 8
1.2. Complementaria 8
2. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS 9
3. OBJETIVO GENERAL 10
4. FORMACIÓN CIUDADANA, VALORES Y HABILIDADES BLANDAS 11
5. NORMAS DE CLASE 11
6. SISTEMA DE EVALUACIÓN 12
7. UNIDADES 12
7.1 UNIDAD 1: FUNDAMENTOS DE BASE DE DATOS NO RELACIONAL Y MODELOS DE DATOS NoSQL 13
7.2 Historia y Evolución de las Bases de Datos 13
7.3 Surgimiento de NoSQL 14
7.4 ¿Qué es NoSQL? 14
7.5 Introducción a Bases de datos no relacionales 15
7.6 Comparación entre bases de datos relacionales y NoSQL. 15
7.7 Bases de datos móviles 17
7.8 Manejo de repositorio de datos. 18
7.9 Diseño de base de datos no relacionales 19
7.10 Tipos de bases de datos NoSQL. 20
7.11 Base de datos de documentos: 20
7.12 Base de datos Clave - Valor 20
7.13 Base de datos de Columnas anchas 21
7.14 Base de datos de Grafos 21
7.15 MongoDB 21
7.16 Manejo de datos en MongoDB 23
7.16.1 Base de Datos 23
7.16.2 Colecciones 24
7.16.3 Documentos 24
7.17 Campos 24
7.18 Herramientas de MongoDB 25
7.18.1 Mongo Shell (mongosh): 25

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

7.18.2 MongoDB Compass: 26
7.18.3 MongoDB Atlas: 26
7.19 Instalación de MongoDB 27
7.20 Tipos de Datos en MongoDB 28
7.20.1 Operadores en Mongo DB 29
7.20.2 Operadores de Comparación 30
7.20.3 Operadores Lógicos 31
7.20.4 Operadores de Elementos 31
7.21 Autoevaluación 1 33
7.22 Resumen de la Unidad 1 36
8 UNIDAD 2 DISEÑO Y MANEJO DE BASES DE DATOS NoSQL 38
8.1 Comandos básicos en MongoDB 38
8.2 Operaciones CRUD en MongoDB 39
8.3 Agregaciones en MongoDB 44
8.4 Pipeline de Agregación 44
8.5 Etapas Comunes en un Pipeline 45
8.6 Autoevaluación 2 48
8.7 Resumen de la Unidad 2 51
9 UNIDAD 3 ADMINISTRACIÓN Y SEGURIDAD DE BASE DE DATOS NoSQL 52
9.1 La indexación en MongoDB 52
9.2 Tipos de índices 53
9.3 Procesos de Backup y Restauración en MongoDB 54
9.4 Backup (Copia de Seguridad) 54
9.5 Restore (Restauración) 54
9.5.1 Scripts para-Backup usando mongodump 54
9.5.2 Scripts para-Restauracion usando mongorestore 55
9.5.3 Gestión de Usuarios y Roles en MongoDB 55
9.6 Crear un Usuario 57
9.6.1 El método db.createUser() permite crear un nuevo usuario dentro del sistema de MongoDB 57
9.6.2 Autenticación como Usuario 57
9.6.3 Crear un rol personalizado 57
9.7 Mongo en la Nube 59
9.7.1 MongoDB Atlas 60
9.8 Autoevaluación 3 61
9.9 Resumen de la Unidad 3 65

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

10. REFERENCIAS 66
11. SOLUCIONARIO 67

ÍNDICE DE FIGURAS

Figura 1 Evolución de bases de datos no relacionales 14
Figura 2 Modelado de datos de bases de datos no relacionales 21
Figura 3 Estructura relacional vs Estructura Json 23
Figura 4 Comparación entre estructura de MongoDB y un Gestor de base de datos relacional 24
Figura 5 Pantalla de MONGOSH 25
Figura 6 Pantalla de MONGO COMPASS 26
Figura 7 Pantalla de MONGODB ATLAS 27
Figura 8 Comparación de Operadores en MongoDB con relación a SQL 31
Figura 9 Comando use 38
Figura 10 Comando show 39
Figura 11 Comando dropDatabase 39
Figura 12 Comando insertOne 40
Figura 13 Comando insertMany 40
Figura 14 Comando find() 41
Figura 15 Comando updateOne 42
Figura 16 Comando updateMany 42
Figura 17 Comando replaceOne 43
Figura 18 Comando deleteOne 43
Figura 19 Comando deleteMany 44
Figura 20 Framework de agregación 44
Figura 21 Ejemplo de uso de Agregaciones -$match y $group 45
Figura 22 Ejemplo de uso de Agregaciones - $sort y $limit 46
Figura 23 Ejemplo de uso de Agregaciones - $project 46
Figura 24 Ejemplo de uso de Agregaciones - $unwind 47
Figura 25 Comando createIndex 53
Figura 26 Script mongodump 54
Figura 27 Script mongorestore 55
Figura 28 Comando createUser 57
Figura 29 Comando auth 57

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 30 Comando createRole 58
Figura 31 Comando grantRolesToUser 58
Figura 32 Comando dropUser 59
Figura 33 Comando dropRole 59
Figura 34 Pantalla de MongoDB Atlas 60

ÍNDICE DE TABLAS

Tabla 1 Comparativa entre Base de datos relacionales y bases de datos no relacionales 16
Tabla 2 Pasos para instalar MongoDB en diferentes Sistemas Operativos 27
Tabla 3 Diferentes usos del comando find 41
Tabla 4 Tipos de índices 53

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

GUÍA GENERAL DE BASE DE DATOS NO RELACIONALES

DESCRIPCIÓN DE LA ASIGNATURA

La asignatura de Bases de Datos No Relacionales es esencial para los estudiantes de la carrera de Desarrollo de Software, ya que les brinda un conocimiento profundo sobre la gestión de datos que trasciende el modelo relacional tradicional. Esta disciplina abarca tanto el estudio como la implementación de bases de datos que emplean modelos de datos como clave-valor, documentos, columnares y grafos, siendo MongoDB una de las herramientas más representativas y utilizadas en este ámbito. En el contexto actual, las bases de datos no relacionales desempeñan un papel crucial debido a su capacidad para manejar grandes volúmenes de datos dispersos y su escalabilidad horizontal, características que son cada vez más requeridas en la industria tecnológica. Las tendencias hacia el manejo de big data, la computación en la nube y el desarrollo de aplicaciones en tiempo real destacan aún más la relevancia de este módulo. El estudio de bases de datos no relacionales no solo enriquece el perfil técnico del estudiante, sino que también lo prepara estratégicamente para enfrentar los desafíos emergentes en el campo del desarrollo de software. Al integrar este conocimiento, los futuros profesionales estarán en condiciones de diseñar soluciones más efectivas y adaptativas que respondan a las necesidades del mercado actual y futuro, convirtiéndose en un pilar fundamental para la formación integral del desarrollador de software y dotándolos de habilidades críticas que fomentan tanto la innovación tecnológica como la competitividad profesional.

Los resultados de su formación personal y profesional dependen de cuánto esté dispuesto a invertir en esfuerzo y responsabilidad. Le invitamos a que, juntos, avancemos en el estudio de la asignatura. Tanto el equipo físico como el humano del ITQ estamos aquí para acompañarlo en su proceso de enseñanza y aprendizaje.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

1. BIBLIOGRAFÍA

1.1. Básica

- Sarasa, A. (2016). *Introducción a las bases de datos NoSQL usando MongoDB*. Editorial UOC.

Este texto abarca conceptos sobre bases de datos NoSQL, enfocándose en MongoDB como una base de datos documental ampliamente adoptada como alternativa a las bases de datos relacionales tradicionales. Introduce a los conceptos y modelos propios de NoSQL mediante un enfoque práctico, también analiza las funcionalidades de MongoDB. Incluyendo ejercicios prácticos con sus soluciones.

- Díaz, J. (2016). *Utilización de las bases de datos relaciones en el sistema de gestión y almacenamiento de datos*. Editorial Tutor Formación.

Este libro es una guía integral sobre informática básica, redes, bases de datos y automatización de tareas. A lo largo de cinco capítulos, se abordan temas como los componentes de un ordenador, redes locales e inalámbricas, instalación y mantenimiento de equipos, así como el uso de bases de datos. También se explican conceptos clave de los sistemas gestores de bases de datos, su estructura, administración y diseño. Se profundiza en la búsqueda de información mediante consultas básicas y avanzadas y se enseña a automatizar tareas repetitivas mediante macros, además de importar y exportar documentos.

1.2. Complementaria

- Marqués, M. (2009). *Base de datos.* Universitat Jaume I. Servei de Comunicació i Publicacions

Este libro ofrece una visión integral del mundo de las bases de datos, abarcando desde los fundamentos teóricos hasta las prácticas avanzadas de diseño y consulta.

- Conesa, J. Rodríguez, M. (2017). *¿Cómo usar una base de datos en grafo?.* Editorial UOC.

Este libro se basa en el modelo H2PAC, un enfoque pedagógico que organiza el aprendizaje a partir de actividades prácticas centradas en retos. Cada unidad comienza con un desafío concreto que el lector debe resolver. Para afrontarlo, se proporciona el conocimiento

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

imprescindible, es decir, el contenido teórico necesario para comprender los conceptos fundamentales. Posteriormente, se presentan soluciones propuestas, que orientan al lector en el proceso de resolución del reto.

Es dirigido a quienes desean iniciarse en el mundo de las bases de datos NoSQL en grafo, el libro ofrece una introducción clara y práctica a los conceptos básicos del modelo en grafo. Además, enseña a utilizar Neo4j, uno de los sistemas gestores de bases de datos en grafo más populares en la actualidad. Para ello, se trabaja con un caso real: una base de datos compuesta por más de 3 millones de registros extraídos de Twitter, lo que permite al lector aplicar los conocimientos adquiridos en un contexto realista y significativo.

- Piñeiro, J. (2013). *Cuaderno del alumno: gestión de bases de datos.* Editorial CEP, S.L..

Este cuaderno aborda temas clave como la estructura y diseño de bases de datos, la utilización del lenguaje SQL para realizar consultas y manipular datos, así como aspectos esenciales de administración de sistemas de bases de datos, incluyendo la creación de usuarios, copias de seguridad y gestión de permisos.Además, el libro incluye ejercicios prácticos y actividades guiadas

- Sanz, I. Aramburu, M. (2012). *Bases de datos avanzadas*. Universitat Jaume I. Servei de Comunicació i Publicacions.

El libro profundiza en temas más allá del modelo relacional tradicional, abordando conceptos y tecnologías emergentes en la gestión de datos. Incluye contenidos sobre bases de datos orientadas a objetos, bases de datos deductivas, XML y bases de datos semiestructuradas, así como almacenamiento y procesamiento de datos distribuidos.

2. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS

Las competencias genéricas y específicas de la materia de Base de datos no relacionales se resumen de la siguiente manera:

- Aplicar pensamiento lógico y analítico para modelar, estructurar y consultar datos en sistemas de bases de datos no relacionales, considerando las necesidades de almacenamiento flexible y escalabilidad.
- Utilizar herramientas tecnológicas actuales, como MongoDB u otras bases de datos NoSQL, para diseñar, implementar y mantener soluciones de almacenamiento de datos orientadas a documentos, grafos, columnas o claves-valor.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- Interpretar y adaptar modelos de datos no relacionales a contextos reales de desarrollo de software, seleccionando el tipo de base de datos más adecuado según los requerimientos del sistema.
- Trabajar de forma colaborativa en entornos de desarrollo ágil, integrando bases de datos NoSQL en aplicaciones modernas y distribuidas.
- Demostrar autonomía en el aprendizaje de nuevas tecnologías y enfoques relacionados con el manejo de grandes volúmenes de datos y arquitecturas orientadas a servicios.
- Comunicar de forma efectiva los resultados de consultas, estructuras de datos y decisiones de diseño, utilizando terminología técnica adecuada.

Estas competencias permiten a los profesionales en desarrollo de software diseñar, implementar y gestionar soluciones de almacenamiento de datos utilizando tecnologías NoSQL, como MongoDB, que ofrecen flexibilidad, escalabilidad y alto rendimiento. Gracias a estas habilidades, los estudiantes pueden estructurar y consultar grandes volúmenes de datos no estructurados o semiestructurados, integrándolos eficazmente en aplicaciones modernas y distribuidas, lo que respalda el desarrollo de sistemas ágiles, escalables y orientados a las necesidades actuales del mercado tecnológico.

3. OBJETIVO GENERAL

Ofrecer a los estudiantes un conocimiento práctico y profundo sobre el diseño, la implementación y la gestión de estas bases de datos, con un enfoque particular en los sistemas MongoDB. A través de esta materia, se busca que los alumnos desarrollen habilidades clave para aplicar estos sistemas de manera efectiva en entornos de desarrollo de software contemporáneos, donde la escalabilidad, la flexibilidad y el rendimiento son esenciales. Los estudiantes aprenderán a evaluar las necesidades de almacenamiento de datos y a elegir la solución de base de datos más adecuada para cada situación, entendiendo cómo estos sistemas pueden optimizar el rendimiento y la funcionalidad de las aplicaciones, tanto las que existen actualmente como las que se desarrollarán en el futuro. De esta manera, estarán alineados con las demandas y tendencias del mercado tecnológico, fortaleciendo su perfil profesional y preparándolos para los retos que enfrentarán en su carrera.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

4. FORMACIÓN CIUDADANA, VALORES Y HABILIDADES BLANDAS

La asignatura desempeña un papel crucial en el desarrollo integral de los alumnos al proporcionarles herramientas esenciales para su crecimiento personal y social, enfocándose en:

1. Educación ambiental: biodiversidad
2. Valores & habilidades blandas: amor: comunicación asertiva y escucha activa
3. Valores & habilidades blandas: compromiso social: adaptabilidad
4. Valores & habilidades blandas: cultura: creatividad
5. Valores & habilidades blandas: lealtad: liderazgo
6. Valores & habilidades blandas: optimismo: planificación y gestión del tiempo
7. Valores & habilidades blandas: orgullo nacional: pensamiento crítico
8. Valores & habilidades blandas: respeto: inteligencia emocional
9. Valores & habilidades blandas: solidaridad: trabajo en equipo
10. Valores & habilidades blandas: tolerancia: flexibilidad
11. Valores & habilidades blandas: verdad y honradez: proactividad

5. NORMAS DE CLASE

Con relación a las normas de clase, es importante destacar que la evaluación de los componentes de gestión académica se compone de tres notas sumativas, cada una con una puntuación máxima de 6.60/6.60, así como un proyecto práctico, como evaluación formativa que se valora con 3.40/3.40, lo que da un total de 10/10 para la calificación del módulo.

Los parciales se califican en una escala de hasta 6.60 puntos, representando cada uno el 2.22 de la calificación total de 6.6 puntos.

Para presentarse al proyecto final, el estudiante debe haber obtenido al menos 4.50 puntos sumando las tres primeras notas. En caso de no alcanzar este mínimo en el proyecto, se otorga una oportunidad de recuperación dentro de las 48 horas laborables siguientes, según el calendario académico oficial.

La nota mínima acumulada requerida para aprobar la asignatura es 7/10, y es esencial mantener al menos un 70% de asistencia a las clases. Los docentes deben informar a los estudiantes sobre sus notas individuales antes de registrarlas en el sistema, y se espera que los

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

alumnos confirmen su aceptación y conformidad con estas calificaciones. Además, los docentes deben entregar un reporte de notas y asistencia a través del SGA y notificado a la coordinación de carrera y registrar las calificaciones en el sistema en un plazo máximo de 5 días posteriores a la recepción del proyecto final.

6. SISTEMA DE EVALUACIÓN

- PARCIAL I - 22%
- PARCIAL II - 22%
- PARCIAL III - 22%
- PROYECTO FINAL - 34%

7. UNIDADES

1. FUNDAMENTOS DE BASE DE DATOS NO RELACIONAL Y MODELOS DE DATOS NoSQL.
2. DISEÑO Y MANEJO DE BASES DE DATOS NoSQL.
3. ADMINISTRACIÓN Y SEGURIDAD DE BASE DE DATOS NoSQL.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

7.1 UNIDAD 1: FUNDAMENTOS DE BASE DE DATOS NO RELACIONAL Y MODELOS DE DATOS NoSQL

"La vida es una elección constante,
solo el tiempo dirá si fue la correcta o no"
D.O

- **Historia y Evolución de las Bases de Datos**
- **Introducción a Bases de datos no relacionales**
- **Comparación entre bases de datos relacionales y NoSQL.**
- **Tipos de bases de datos NoSQL.**
- **Base de datos móviles**
- **Manejo de repositorio de datos**
- **MongoDB: Estructuras en MongoDB**
- **MongoDB: Operadores en Mongo DB**

7.2 Historia y Evolución de las Bases de Datos

La historia de las bases de datos se remonta a la década de 1960, con el desarrollo de los primeros sistemas de gestión de bases de datos (DBMS). (Codd, 1970) sentó las bases para las bases de datos relacionales. Propuso un modelo de datos basado en relaciones matemáticas, donde los datos se organizan en tablas (o relaciones) compuestas por filas y columnas.

Luego en la década de 1980, Seún (Díaz Salvo, 2016), las bases de datos relacionales se volvieron dominantes con la introducción de sistemas como Oracle, IBM DB2 y Microsoft SQL Server. Estos sistemas utilizaban el lenguaje SQL para la manipulación y consulta de datos, y se convirtieron en la piedra angular de la gestión de datos empresariales.

https://youtu.be/yx0dOelDRxE?si=CUtLVWS5GwgPdGcB

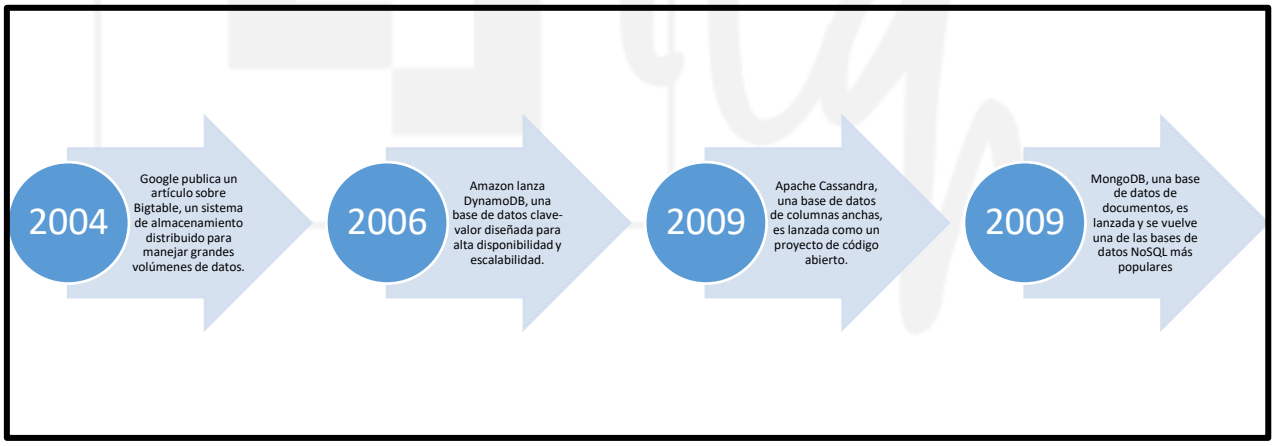
(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

7.3 Surgimiento de NoSQL

A medida que la web crecía y las aplicaciones se volvían más complejas, las limitaciones de las bases de datos relacionales comenzaron a hacerse evidentes. Las bases de datos relacionales estaban diseñadas para manejar datos estructurados y transacciones ACID, pero tenían dificultades para escalar horizontalmente y manejar grandes volúmenes de datos no estructurados, es por eso que, en la década del 2000, con el auge de las redes sociales, el big data y las aplicaciones en tiempo real, surgió la necesidad de sistemas de gestión de bases de datos que pudieran manejar estos nuevos desafíos. Fue en este contexto que las bases de datos NoSQL comenzaron a ganar popularidad.

Según (Moniruzzaman, 2013), las bases de datos NoSQL ofrecen esquemas flexibles, lo que permite almacenar datos no estructurados o semiestructurados sin la necesidad de un esquema fijo. Hoy en día, las bases de datos NoSQL son una parte integral del ecosistema de gestión de datos y se utilizan en una amplia variedad de aplicaciones, desde sistemas de gestión de contenido hasta análisis de big data, sistemas de recomendación y aplicaciones en tiempo real.

Figura 1
Evolución de bases de datos no relacionales



***Nota*.** Figura que representa los hitos en la evolución de NoSQL. **Fuente:** Investigación propia.

7.4 ¿Qué es NoSQL?

NoSQL, que significa "Not Only SQL" (No Solo SQL), es un término que se refiere a una amplia clase de sistemas de gestión de bases de datos que difieren del modelo relacional tradicional utilizado en las bases de datos SQL. A diferencia de las bases de datos relacionales, que almacenan datos en tablas y utilizan esquemas fijos, las bases de datos NoSQL están diseñadas para manejar datos no estructurados o semiestructurados y ofrecen esquemas flexibles.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Las bases de datos NoSQL son especialmente útiles en entornos donde se requiere escalabilidad horizontal, alta disponibilidad y la capacidad de manejar grandes volúmenes de datos. Estas características las hacen ideales para aplicaciones modernas como redes sociales, sistemas de recomendación, análisis de big data y aplicaciones en tiempo real.


https://youtu.be/1kMfcigoFmA?si=4EsDvmo-lIo0WfHn

7.5 Introducción a Bases de datos no relacionales

En la era del Big Data, la cantidad, variedad y velocidad con la que se generan los datos ha superado las capacidades de los sistemas tradicionales. Las bases de datos relacionales, aunque eficientes en muchos contextos, presentan limitaciones cuando se trata de manejar datos a gran escala, no estructurados o con esquemas flexibles. Ante esta necesidad, surgieron las bases de datos no relacionales, comúnmente conocidas como NoSQL.

Las bases de datos NoSQL permiten trabajar con grandes volúmenes de datos heterogéneos, lo que las hace ideales para aplicaciones modernas como redes sociales, análisis en tiempo real, sistemas de recomendación, IoT y almacenamiento de contenidos. Además, muchas de estas bases están diseñadas para operar en infraestructuras distribuidas en la nube, lo que facilita su escalado horizontal y su disponibilidad continua.

En este contexto, el conocimiento y la aplicación de bases de datos NoSQL se han vuelto fundamentales para desarrolladores, analistas de datos y arquitectos de sistemas. Comprender su funcionamiento, ventajas y casos de uso permitirá seleccionar la tecnología más adecuada según las necesidades de cada proyecto.

7.6 Comparación entre bases de datos relacionales y NoSQL.

Las bases de datos constituyen un pilar fundamental en los sistemas de información modernos. Tradicionalmente, las bases de datos relacionales (RDBMS) han dominado el panorama, sustentadas en el modelo relacional propuesto por Codd en el año de 1970. Estas bases organizan los datos en tablas estructuradas y normalizadas organizadas en filas y columnas, con relaciones claramente definidas entre ellas, es de ahí su nombre.

Según (Silberschatz A. H., 2019) las bases de datos relacionales utilizan el lenguaje SQL (Structured Query Language) para manipular los datos y son ampliamente reconocidas por su integridad, consistencia y robustez en el manejo y control de las transacciones.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Por otro lado, con el auge del Big Data, los sistemas distribuidos y las aplicaciones en tiempo real han dado lugar al desarrollo de las bases de datos no relacionales, que ofrecen esquemas más flexibles, alto rendimiento y escalabilidad horizontal. A diferencia de las relacionales, las NoSQL no requieren esquemas fijos y son capaces de manejar datos semi-estructurados o no estructurados. Además, permiten el almacenamiento distribuido, facilitando el trabajo en entornos cloud y sistemas con altas demandas de disponibilidad y rendimiento.

Según (Li, 2013) una de las principales diferencias entre las dos tecnologías radica en la estructura de los datos. Mientras las bases relacionales emplean un modelo tabular, las NoSQL pueden adoptar diferentes modelos como documentos por ejemplo como lo maneja, MongoDB), clave-valor por ejemplo como lo maneja Redis, columnas como lo maneja Cassandra o grafos como lo maneja Neo4j, dependiendo de la naturaleza de la aplicación. Esta diversidad de modelos permite una mayor adaptabilidad a distintos tipos de datos y casos de uso.

En cuanto a las transacciones, las bases relacionales implementan el modelo ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), garantizando operaciones confiables, lo que las hace adecuadas para sistemas bancarios o financieros. En contraste, muchas bases NoSQL priorizan la disponibilidad y escalabilidad sobre la consistencia inmediata, adoptando en su lugar el modelo BASE (Básicamente Disponible, Estado Suave, Eventualmente Consistente).

Finalmente, la escalabilidad representa otra diferencia clave. Las RDBMS generalmente escalan verticalmente donde se agregan más recursos al mismo servidor, mientras que las NoSQL están diseñadas para escalar horizontalmente es decir agregar más servidores, lo cual es esencial en aplicaciones con grandes volúmenes de datos y usuarios concurrentes.

Tabla 1
Comparativa entre Base de datos relacionales y bases de datos no relacionales

Criterio	Bases de Datos Relacionales (RDBMS)	Bases de Datos No Relacionales (NoSQL)
Modelo de Datos	Tablas con filas y columnas, modelo relacional.	Documentos, clave-valor, columnas, grafos.
Lenguaje de Consulta	SQL (Structured Query Language).	Lenguajes específicos según el tipo de NoSQL (ej. MongoDB Query Language).

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Esquema	Fijo y predefinido, requiere normalización.	Esquema flexible, permite datos semi-estructurados o no estructurados.
Escalabilidad	Vertical (mayor capacidad al mismo servidor).	Horizontal (añadiendo más nodos o servidores).
Transacciones	Soporta ACID: consistencia estricta.	BASE: eventualmente consistente, alta disponibilidad.
Integridad de Datos	Alta integridad mediante claves primarias y foráneas.	Menor control de integridad, depende de la lógica de aplicación.
Rendimiento	Óptimo para operaciones complejas y relaciones.	Mejor en consultas rápidas y grandes volúmenes de datos.
Uso Típico	Aplicaciones bancarias, ERP, sistemas transaccionales.	Big Data, redes sociales, IoT, análisis en tiempo real.
Ejemplos de Sistemas	MySQL, PostgreSQL, Oracle, SQL Server.	MongoDB, Cassandra, Redis, Neo4j.
Manejo de Datos	Exclusivamente usa datos estructurados.	Estructurados, semi-estructurados y no estructurados.

Nota. Tabla que muestra la comparativa entre Base de datos relacionales y bases de datos no relacionales según varios criterios. **Fuente:** Investigación propia.

7.7 Bases de datos móviles

Las bases de datos móviles representan una evolución significativa en el ámbito de las bases de datos no relacionales, diseñadas específicamente para operar en dispositivos móviles como smartphones y tablets. Estas bases de datos están optimizadas para funcionar en entornos con recursos limitados, como menor capacidad de almacenamiento y procesamiento, así como conectividad intermitente.

Según (Sadalage, 2013), una de las principales ventajas de las bases de datos móviles es su eficiencia en el uso de recursos. Están diseñadas con un tamaño reducido, lo que permite una instalación ligera y un menor uso de almacenamiento interno, lo cual es fundamental en dispositivos con capacidad limitada. Además, estas bases consumen baja cantidad de CPU y memoria RAM, lo que contribuye a una mejor autonomía energética y rendimiento general del dispositivo.

También la sincronización de datos es una función esencial en entornos móviles. Muchas bases de datos móviles ofrecen sincronización offline, lo que permite a los usuarios

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

continuar trabajando sin conexión a Internet y sincronizar los cambios una vez que se restablece la conectividad. Este proceso requiere un manejo eficiente de conflictos de sincronización, especialmente cuando los mismos datos han sido modificados desde distintos dispositivos o usuarios. Una base de datos móvil efectiva debe tener mecanismos robustos para resolver estos conflictos sin pérdida de información.

Y otro factor principal dado que los dispositivos móviles están expuestos a pérdidas, robos o accesos no autorizados, la seguridad de los datos es un aspecto crítico. Por ello, estas bases de datos suelen incorporar cifrado en el almacenamiento, asegurando que los datos estén protegidos incluso si el dispositivo cae en manos equivocadas. Además, implementan sistemas de autenticación y autorización para controlar el acceso, permitiendo que solo los usuarios legítimos puedan interactuar con la base de datos.

Entre los ejemplos más destacados de bases de datos móviles se encuentra Realm, una base de datos orientada a objetos diseñada especialmente para aplicaciones móviles. Se caracteriza por ofrecer sincronización en tiempo real, soporte multiplataforma y un rendimiento altamente optimizado, lo que la convierte en una opción ideal para aplicaciones que requieren rapidez y eficiencia. Además, cuenta con cifrado nativo y una API intuitiva para facilitar el trabajo de los desarrolladores.

Por otro lado, SQLite, aunque es una base de datos relacional, es ampliamente utilizada en dispositivos móviles debido a su ligereza y simplicidad. Se trata de una base de datos embebida que no requiere servidor ni configuración adicional, lo cual permite ejecutar operaciones SQL estándar de forma rápida y eficiente. Es, de hecho, una de las bases de datos predeterminadas en sistemas operativos como Android e iOS.

Finalmente, Couchbase Mobile es una base de datos NoSQL que destaca por su capacidad de sincronización bidireccional, incluso en escenarios offline. Está orientada al manejo de datos en formato JSON y ofrece una arquitectura flexible y escalable, adecuada para entornos distribuidos. Su capacidad de operar tanto localmente como en la nube la hace especialmente útil en aplicaciones que requieren acceso constante a la información.

7.8 Manejo de repositorio de datos.

Un repositorio de datos es un sistema que permite almacenar, gestionar y recuperar datos de manera eficiente. En el contexto de las bases de datos NoSQL, los repositorios de datos están diseñados para manejar grandes volúmenes de información no estructurada o semiestructurada. Estos repositorios pueden incluir documentos, gráficos, columnas anchas y pares clave-valor, entre otros formatos.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Los repositorios de datos NoSQL han cobrado relevancia en los últimos años debido a la necesidad creciente de manejar datos con estructuras variables, distribuir grandes volúmenes de información y responder con rapidez a los cambios en las aplicaciones modernas. Una de las principales características de los sistemas NoSQL es su capacidad de operar con esquemas flexibles. A diferencia de las bases de datos relacionales, que requieren una estructura fija de tablas y campos previamente definida, los repositorios NoSQL permiten almacenar datos con diferentes estructuras sin necesidad de realizar migraciones complejas. Esto brinda una ventaja significativa en entornos de desarrollo ágil, donde los requerimientos cambian con frecuencia y es fundamental adaptar el modelo de datos sin perder tiempo en reestructuraciones.

Otra característica importante es la escalabilidad horizontal, lo que significa que estos sistemas están diseñados para distribuir la carga de trabajo y los datos a través de múltiples servidores o nodos. Este enfoque permite aumentar la capacidad del sistema simplemente añadiendo más servidores, en lugar de depender únicamente de un servidor central más potente.

Existen diferentes tipos de repositorios de datos NoSQL, cada uno diseñado para satisfacer necesidades específicas, por ejemplo, los repositorios de documentos, clave-valor, columnas anchas y repositorios gráficos.

Las bases de datos NoSQL ofrecen una variedad de enfoques para el almacenamiento y gestión de datos, permitiendo elegir el tipo más adecuado según las necesidades de la aplicación. Su flexibilidad, escalabilidad y alto rendimiento las convierten en una herramienta fundamental en el desarrollo de sistemas modernos que demandan agilidad, disponibilidad continua y eficiencia a gran escala.

7.9 Diseño de base de datos no relacionales

Según (Hecht, 2011), el diseño de bases de datos no relacionales, o NoSQL, se centra en la creación de estructuras de datos flexibles y escalables que pueden manejar grandes volúmenes de información no estructurada o semiestructurada. A diferencia de las bases de datos relacionales, que utilizan tablas con esquemas fijos, las bases de datos NoSQL permiten esquemas dinámicos, lo que facilita la adaptación a cambios en los requisitos de datos sin necesidad de rediseños complejos.

En el diseño de bases de datos NoSQL, es crucial seleccionar el tipo adecuado de base de datos según las necesidades específicas de la aplicación.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

A continuación, se presentan los diferentes tipos de bases de datos NoSQL, cada uno con sus características y casos de uso específicos, lo que permitirá una mejor comprensión de cómo seleccionar la base de datos más adecuada según el tipo de datos y los requisitos de la aplicación.

7.10 Tipos de bases de datos NoSQL.

 https://youtu.be/LEX16Cmvark?si=vo9H0XoJuy5HM4Jw

Los tipos de repositorio de datos NoSQL está estrechamente vinculado a un tipo específico de base de datos, diseñado para resolver problemas concretos y optimizar el rendimiento según la estructura de los datos y las necesidades del sistema. Esta diversidad permite a los desarrolladores y arquitectos de software seleccionar la herramienta más adecuada para cada caso de uso. Las bases de datos NoSQL son sistemas de gestión de bases de datos que no siguen el modelo relacional tradicional. Según (Grolinger, 2013) existen algunos tipos comunes de bases de datos NoSQL entre los que se puede citar:

7.11 Base de datos de documentos:

Las bases de datos de documentos almacenan la información en documentos, generalmente en formato JSON, BSON o XML. Cada documento contiene pares de clave-valor y puede tener una estructura flexible, lo que permite que los documentos dentro de una misma colección tengan campos diferentes. Este tipo de bases de datos es ideal para aplicaciones que requieren esquemas flexibles y la capacidad de manejar datos jerárquicos. Un ejemplo de un repositorio de datos muy usado es MongoDB que es una de las bases de datos de documentos más populares. Se utiliza en una variedad de aplicaciones, desde sistemas de gestión de contenido hasta aplicaciones móviles, debido a su flexibilidad y escalabilidad.

7.12 Base de datos Clave - Valor

Las bases de datos clave-valor son las más simples entre los tipos de bases de datos NoSQL. Almacenan datos como un diccionario, donde cada valor se asocia con una clave única. Este modelo es extremadamente eficiente para operaciones de lectura y escritura, lo que lo hace ideal para aplicaciones que requieren alto rendimiento y baja latencia. Redis es un repositorio de datos que maneja base de datos clave-valor en memoria que se utiliza comúnmente para caching y sesiones de usuario debido a su alta velocidad y eficiencia.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

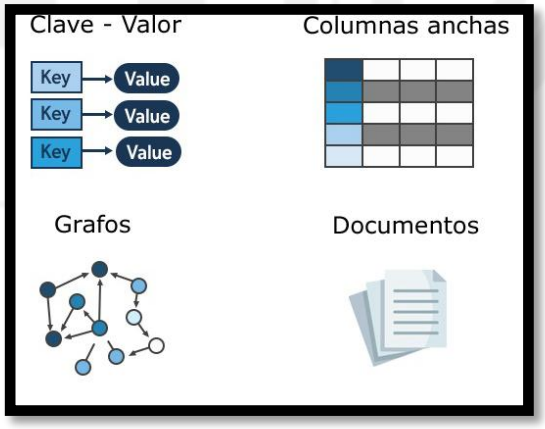
7.13 Base de datos de Columnas anchas

Las bases de datos de columnas anchas almacenan datos en columnas en lugar de filas, lo que permite una mayor flexibilidad y escalabilidad. Este modelo es especialmente útil para aplicaciones que requieren el manejo de grandes volúmenes de datos y consultas complejas sobre grandes conjuntos de datos. Por ejemplo, el repositorio de datos Apache Cassandra es una base de datos de columnas anchas diseñada para manejar grandes cantidades de datos distribuidos en múltiples nodos. Es conocida por su alta disponibilidad y tolerancia a fallos

7.14 Base de datos de Grafos

Las bases de datos gráficas o de grafos almacenan datos en nodos y aristas, lo que permite representar relaciones complejas entre los datos. Este modelo es ideal para aplicaciones que requieren el análisis de redes y relaciones, como redes sociales, sistemas de recomendación y detección de fraudes. Por ejemplo Neo4j es una de las bases de datos gráficas más populares, se utiliza en aplicaciones que requieren el análisis de relaciones complejas, como redes sociales y sistemas de recomendación.

Figura 2
Modelado de datos de bases de datos no relacionales



Nota. Figura que representa gráficamente el modelado de datos de las bases de datos NoSQL. **Fuente:** Investigación propia.

7.15 MongoDB

MongoDB es una base de datos NoSQL orientada a documentos, diseñada para ofrecer alta escalabilidad y flexibilidad en el manejo de datos. A diferencia de las bases de datos relacionales tradicionales, que almacenan datos en tablas con esquemas fijos, MongoDB almacena datos en documentos similares a JSON, lo que permite una estructura de datos más

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

dinámica y adaptable. Esta característica hace que MongoDB sea especialmente adecuado para aplicaciones que requieren manejar grandes volúmenes de datos no estructurados o semiestructurados, así como para entornos donde los requisitos de datos pueden cambiar con frecuencia.

Según (Silberschatz A. K., 2019), uno de los conceptos fundamentales de MongoDB es el documento, que es la unidad básica de almacenamiento. Los documentos en MongoDB son similares a los objetos JSON y consisten en pares clave-valor. Cada documento puede tener una estructura diferente, lo que permite una gran flexibilidad en el modelado de datos. Los documentos se agrupan en colecciones, que son análogas a las tablas en las bases de datos relacionales, pero con la ventaja de que no requieren un esquema predefinido.

Según (Sarasa, 2016) enfatiza que otro concepto clave en MongoDB es el de las consultas ad hoc, que permiten a los usuarios realizar búsquedas y recuperaciones de datos de manera eficiente y flexible. MongoDB soporta un rico lenguaje de consultas que incluye operaciones de filtrado, proyección, ordenación, agregación y más. Esto facilita la manipulación y análisis de datos de manera rápida y eficiente, incluso en conjuntos de datos muy grandes.

La escalabilidad es otro aspecto crucial de MongoDB. Está diseñado para escalar horizontalmente, lo que significa que puede manejar grandes volúmenes de datos y tráfico mediante la distribución de la carga entre múltiples servidores. Esto se logra a través de la fragmentación (sharding), que divide los datos en partes más pequeñas y los distribuye entre diferentes nodos en un clúster. Además, MongoDB soporta la réplica de datos, lo que mejora la disponibilidad y la tolerancia a fallos.

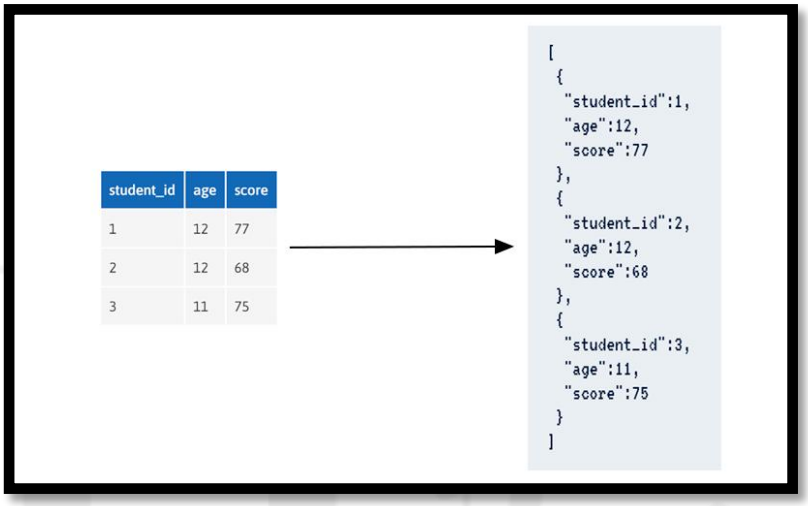
La flexibilidad y escalabilidad de MongoDB lo hacen ideal para una amplia variedad de aplicaciones, desde pequeñas aplicaciones web hasta grandes sistemas empresariales. Es utilizado por numerosas organizaciones en diferentes industrias para manejar datos en tiempo real, análisis de big data, gestión de contenido, y más.

En resumen, MongoDB es una poderosa base de datos NoSQL que ofrece una combinación única de flexibilidad, escalabilidad y rendimiento. Su capacidad para manejar datos no estructurados y semiestructurados, junto con su potente lenguaje de consultas y características de escalabilidad, lo convierten en una opción popular para desarrolladores y empresas que buscan soluciones de gestión de datos modernas y eficientes.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

https://youtu.be/2GU938abfQg?si=zlUtpHrlZOs5VC51

Figura 3
Estructura relacional vs Estructura Json



Nota. Figura que muestra una representación de una tabla que contiene la información de id del estudiante, la edad y la puntuación en estructura relacional y como se vería en misma información en estructura Json. **Fuente:** Investigación propia.

7.16 Manejo de datos en MongoDB

MongoDB organiza los datos en una estructura jerárquica que comienza con la base de datos, que contiene colecciones, las cuales a su vez contienen documentos. Cada documento está compuesto por campos que almacenan los datos reales. Esta estructura permite una gran flexibilidad y escalabilidad, ya que los documentos dentro de una colección no necesitan tener la misma estructura y pueden evolucionar con el tiempo sin requerir cambios en el esquema de la base de datos.

https://youtu.be/dydU7RqbpFQ?si=HYXBXP4unjRD4M-S

7.16.1 Base de Datos

En MongoDB, una base de datos es un contenedor físico para las colecciones. Cada base de datos tiene su propio conjunto de archivos en el sistema de archivos. MongoDB puede

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

alojar múltiples bases de datos, cada una con sus propias colecciones y datos. Las bases de datos en MongoDB se utilizan para organizar y agrupar colecciones de manera lógica, facilitando la gestión y el acceso a los datos.

7.16.2 Colecciones

Las colecciones en MongoDB son grupos de documentos. Son análogas a las tablas en las bases de datos relacionales, pero a diferencia de estas, no requieren un esquema fijo. Esto significa que los documentos dentro de una colección pueden tener diferentes estructuras. Las colecciones permiten organizar los documentos de manera lógica y facilitan la realización de consultas y operaciones sobre conjuntos de datos relacionados.

7.16.3 Documentos

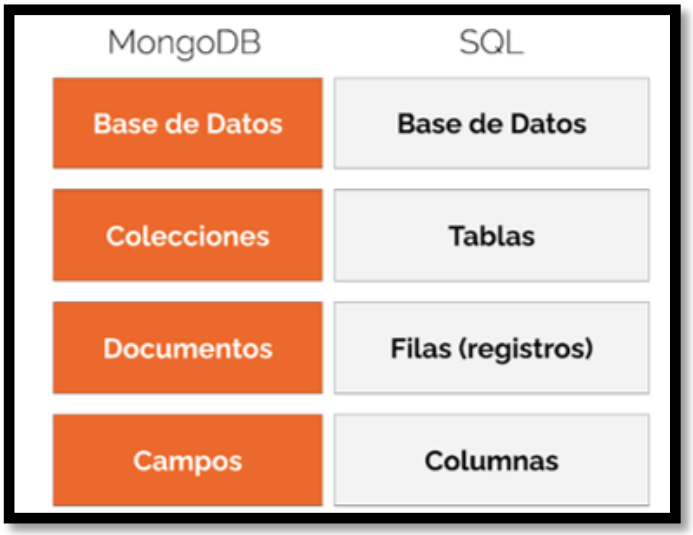
Los documentos son la unidad básica de almacenamiento en MongoDB y son análogos a las filas en las bases de datos relacionales. Sin embargo, a diferencia de las filas, los documentos en MongoDB son estructuras de datos similares a JSON (JavaScript Object Notation) que consisten en pares clave-valor. Cada documento en una colección puede tener una estructura diferente, lo que permite una gran flexibilidad en el modelado de datos. Los documentos pueden contener datos anidados y arrays, lo que facilita la representación de estructuras de datos complejas y jerárquicas.

7.17 Campos

Los campos son los elementos básicos de un documento en MongoDB y son análogos a las columnas en las bases de datos relacionales. Cada campo en un documento es un par clave-valor, donde la clave es una cadena que identifica el campo y el valor puede ser de varios tipos de datos, como cadenas, números, booleanos, arrays y otros documentos. Los campos permiten almacenar y organizar los datos dentro de un documento, facilitando la representación de información estructurada y semiestructurada.

Figura 4
Comparación entre estructura de MongoDB y un Gestor de base de datos relacional

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC



Nota. Figura que muestra una comparativa entre estructura como maneja los datos MongoDB y un Gestor de base de datos relacional SQL. **Fuente:** Investigación propia.

7.18 Herramientas de MongoDB

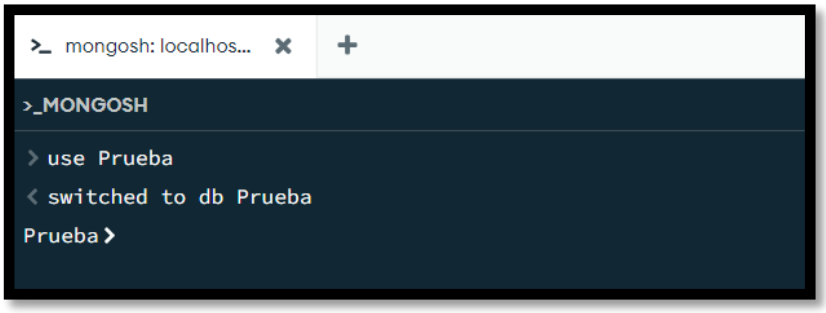
MongoDB almacena datos en formato BSON, lo que permite una estructura de datos más dinámica y adaptable. Entre sus herramientas más destacadas se encuentran Mongo Shell, una interfaz de línea de comandos que facilita la interacción con la base de datos mediante comandos JavaScript; MongoDB Compass, una interfaz gráfica intuitiva que simplifica la exploración y gestión de datos; y MongoDB Atlas, un servicio en la nube completamente gestionado que ofrece soluciones escalables y seguras para alojar bases de datos.

7.18.1 Mongo Shell (mongosh):

Mongo Shell es una interfaz de línea de comandos (CLI) para interactuar con MongoDB. Permite a los usuarios ejecutar comandos JavaScript para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar), administrar bases de datos y realizar consultas complejas. Es especialmente útil para administradores de bases de datos y desarrolladores que necesitan realizar tareas rápidas y automatizadas.

Figura 5
Pantalla de MONGOSH

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC



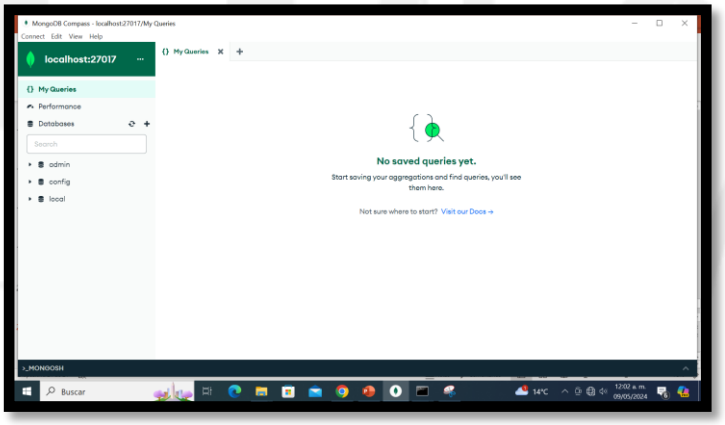
Nota. Figura que muestra la pantalla de Mongo Shell. **Fuente:** Investigación propia.

7.18.2 MongoDB Compass:

MongoDB Compass es una interfaz gráfica de usuario (GUI) para MongoDB. Proporciona una manera visual y fácil de interactuar con los datos almacenados en MongoDB.

Incluye funciones como la exploración de datos, la creación y gestión de índices, la visualización de esquemas y la ejecución de consultas ad hoc. Es ideal para desarrolladores y analistas de datos que prefieren una interfaz visual.

Figura 6
Pantalla de MONGO COMPASS



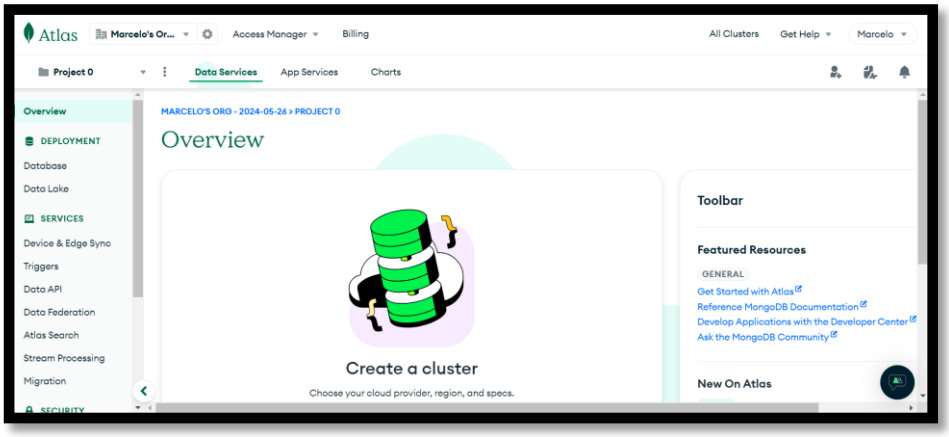
Nota. Figura que muestra la pantalla de Mongo Compass. **Fuente:** Investigación propia.

7.18.3 MongoDB Atlas:

MongoDB Atlas es un servicio de base de datos en la nube completamente gestionado por MongoDB. Ofrece una solución escalable y segura para alojar bases de datos MongoDB en la nube.

Incluye funciones como la replicación automática, el sharding, la monitorización, las copias de seguridad y la seguridad avanzada. Es ideal para empresas que buscan una solución de base de datos en la nube sin la necesidad de gestionar la infraestructura subyacente.

Figura 7
Pantalla de MONGODB ATLAS



***Nota*.** Figura que muestra la pantalla de Mongo Atlas. **Fuente:** Investigación propia.

7.19 Instalación de MongoDB

La instalación de MongoDB puede variar dependiendo del sistema operativo que se este utilizando. A continuación, se proporcionan instrucciones detalladas para instalar MongoDB en Windows, macOS y Linux.

https://youtu.be/jWGeQeegdws?si=BlJflB3KXIpkM2xw

Tabla 2
Pasos para instalar MongoDB en diferentes Sistemas Operativos

Paso	Windows	macOS	Linux (Ubuntu)

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

1	Descargar MongoDB Community Server desde el sitio oficial: https://www.mongodb.com/try/download/community	Instalar Homebrew: /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"	Importar la clave pública: sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4
2	Ejecutar el instalador MSI y seguir las instrucciones	Instalar MongoDB: brew tap mongodb/brew y brew install mongodb-community	Agregar el repositorio: `echo "deb [arch=amd64] https://repo.mongodb.org/apt/ubuntu $(lsb_release -sc)/mongodb-org/4.4 multiverse"
3	Configurar MongoDB como servicio	Crear directorio de datos: sudo mkdir -p /data/db	Actualizar lista de paquetes: sudo apt-get update
4	Finalizar la instalación	Establecer permisos: sudo chown -R $(whoami) /data/db	Instalar MongoDB: sudo apt-get install -y mongodb-org
5	Verificar instalación: mongod --version	Iniciar MongoDB: mongod	Iniciar MongoDB: sudo systemctl start mongod
6		Verificar instalación: mongo --version	Habilitar inicio automático: sudo systemctl enable mongod
7			Verificar instalación: mongod --version

Nota. Tabla que muestra los pasos a seguir para instalar MongoDB en Windows, MacOS y Linux. **Fuente:** Investigación propia.

7.20 Tipos de Datos en MongoDB

MongoDB es una base de datos NoSQL orientada a documentos que soporta una amplia variedad de tipos de datos. Esta flexibilidad en los tipos de datos permite a los desarrolladores modelar y almacenar información de manera más cercana a cómo se utiliza en las aplicaciones. A continuación, se describen los tipos de datos más comunes soportados por MongoDB:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- String: Este es el tipo de dato más común y se utiliza para almacenar texto. En MongoDB, las cadenas se representan como UTF-8.
- Integer: Se utiliza para almacenar números enteros. MongoDB soporta enteros de 32 y 64 bits.
- Double: Este tipo de dato se utiliza para almacenar números de punto flotante. Es útil para representar valores decimales.
- Boolean: Se utiliza para almacenar valores booleanos, es decir, verdadero o true y/o falso o false.
- Array: Permite almacenar listas de valores. Los arrays en MongoDB pueden contener valores de cualquier tipo de dato, incluyendo otros documentos.
- Object: Este tipo de dato se utiliza para almacenar documentos incrustados. Los documentos incrustados permiten representar estructuras de datos jerárquicas y complejas.
- Null: Se utiliza para representar la ausencia de un valor o un valor nulo.
- Date: Este tipo de dato se utiliza para almacenar fechas y horas. MongoDB almacena las fechas como el número de milisegundos desde la época Unix (1 de enero de 1970).
- ObjectId: Es un tipo de dato especial utilizado para identificar de manera única los documentos en una colección. Cada ObjectId es un identificador de 12 bytes que se genera automáticamente.
- Binary Data: Se utiliza para almacenar datos binarios. Es útil para almacenar archivos y otros tipos de datos binarios.
- Code: Este tipo de dato se utiliza para almacenar código JavaScript.

7.20.1 Operadores en Mongo DB

MongoDB proporciona una amplia gama de operadores que permiten realizar consultas complejas y manipulaciones de datos. Estos operadores se utilizan en conjunto con los métodos de consulta para filtrar, proyectar, actualizar y manipular datos de diversas maneras. A continuación, se describen algunos de los operadores más comunes en MongoDB:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

7.20.2 Operadores de Comparación

https://youtu.be/HGcwusEvyBA?si=RUs-kI-WuBtPI9HL

- $eq: Igual a. Selecciona documentos donde el valor de un campo es igual al valor especificado.
- $ne: No igual a. Selecciona documentos donde el valor de un campo no es igual al valor especificado.
- $gt: Mayor que. Selecciona documentos donde el valor de un campo es mayor que el valor especificado.
- $gte: Mayor o igual que. Selecciona documentos donde el valor de un campo es mayor o igual que el valor especificado.
- $lt: Menor que. Selecciona documentos donde el valor de un campo es menor que el valor especificado.
- $lte: Menor o igual que. Selecciona documentos donde el valor de un campo es menor o igual que el valor especificado.
- $in: En. Selecciona documentos donde el valor de un campo es igual a cualquier valor en una lista especificada.
- $nin: No en. Selecciona documentos donde el valor de un campo no es igual a ningún valor en una lista especificada.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 8

Comparación de Operadores en MongoDB con relación a SQL

MongoDB

Tipo	Significado
$eq	Igual
$lt	Menor que
$gt	Mayor que
$lte	Menor o igual que
$gte	Mayor o igual que
$ne	Distinto
$in	Dentro de
$nin	No dentro de

SQL

Tipo	Significado
=	Igual
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
!=	Distinto
IN	Dentro de

Nota. Figura que muestra una comparativa de Operadores en MongoDB con relación a SQL y sus significados. **Fuente:** Investigación propia.

7.20.3 Operadores Lógicos

https://youtu.be/r7pTxv5SPEk?si=YTncFgkvPx4Mca_C

- $and: Y lógico. Selecciona documentos que cumplen con todas las condiciones especificadas.
- $or: O lógico. Selecciona documentos que cumplen con al menos una de las condiciones especificadas.
- $not: No lógico. Selecciona documentos que no cumplen con la condición especificada.
- $nor: Ni lógico. Selecciona documentos que no cumplen con ninguna de las condiciones especificadas.

7.20.4 Operadores de Elementos

https://youtu.be/6DoveObas9c?si=uoIXUoRox3T_Wwz7

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- $exists: Existe. Selecciona documentos que contienen un campo especificado.
- $type: Tipo. Selecciona documentos donde el valor de un campo es de un tipo especificado.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

7.21 Autoevaluación 1

Lea detenidamente la pregunta y marque la respuesta correcta entre las opciones dadas.

1. ¿Cuál fue una de las principales limitaciones de las bases de datos relacionales que motivó el surgimiento de NoSQL?

a) La necesidad de almacenar datos estructurados.

b) El uso exclusivo de SQL como lenguaje de consulta.

c) La dificultad para manejar grandes volúmenes de datos no estructurados.

d) La falta de soporte para múltiples usuarios simultáneamente.

2. ¿Qué característica distingue principalmente a las bases de datos NoSQL de las bases de datos relacionales?

a) La normalización de datos.

b) La capacidad de realizar transacciones ACID.

c) El uso de esquemas fijos.

d) La flexibilidad de esquemas y la escalabilidad horizontal.

3. ¿Cuál de los siguientes modelos de datos NoSQL utiliza nodos y relaciones como estructura principal?

a) Documentos.

b) Columnas anchas.

c) Clave-valor.

d) Grafos.

4. En el contexto de MongoDB, ¿cómo se denomina al conjunto equivalente a una tabla en bases de datos relacionales?

a) Colección.

b) Base de datos.

c) Campo.

d) Documento.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

5. ¿Cuál de las siguientes herramientas proporciona una interfaz gráfica para trabajar con MongoDB?

a) mongosh.

b) Compass.

c) Mongo Shell.

d) mongod.

6. ¿Qué operador lógico de MongoDB se utiliza para combinar múltiples condiciones y evaluar si al menos una es verdadera?

a) $eq

b) $and

c) $or

d) $not.

7. ¿Qué afirmación es verdadera respecto a las bases de datos móviles?

a) No requieren sincronización con servidores centrales.

b) Son diseñadas exclusivamente para dispositivos de escritorio.

c) Su objetivo es ejecutar consultas complejas desde el servidor.

d) Están optimizadas para operar en entornos desconectados o de red limitada.

8. ¿Qué comando en Mongo Shell permite seleccionar una base de datos existente o crear una nueva si no existe?

a) use <nombreBD>.

b) show dbs.

c) db.createDatabase()

d) connect <nombreBD>.

9. ¿Qué sistema NoSQL es especialmente adecuado para análisis de grandes volúmenes de datos por su estructura de documentos y colecciones?

a) Redis.

b) MongoDB.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

c) Cassandra.

d) Neo4j.

10. ¿Qué operador de elementos se usa en MongoDB para verificar si un campo existe en un documento?

a) $exists.

b) $type.

c) $eq.

d) $in.

Si ha concluido la autoevaluación:

Verifique sus respuestas en el solucionario que se encuentra al final de la presente Guía.

Si alcanzó un porcentaje alto de respuestas correctas en las preguntas de evaluación, es hora de continuar con la Unidad 2, caso contrario, vuelva a revisar los temas en los que ha tenido dificultad.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

7.22 Resumen de la Unidad 1

La Unidad 1 aborda los fundamentos teóricos de las bases de datos no relacionales o también llamados NoSQL, realiza una revisión de la evolución de los sistemas de gestión de bases de datos desde la década de 1960, con los primeros modelos jerárquicos y de red, hasta el modelo relacional propuesto por Edgar F. Codd en 1970. Sin embargo, con la llegada de la web, el Big Data y las aplicaciones en tiempo real, surgieron limitaciones en los sistemas relacionales, dando paso a nuevas alternativas más escalables y flexibles: las bases de datos NoSQL.

El término NoSQL que viene de Not Only SQL, abarca una variedad de tecnologías que se alejan del modelo tabular tradicional y adoptan estructuras como documentos, clave-valor, columnas y grafos. Estas permiten trabajar con datos estructurados, semiestructurados y no estructurados, sin requerir esquemas fijos. Son altamente escalables, ofrecen alta disponibilidad y están diseñadas para funcionar en infraestructuras distribuidas, siendo ideales para entornos modernos como redes sociales, sistemas de recomendación, IoT y análisis de big data.

También se analiza una comparación entre bases de datos relacionales y NoSQL, resaltando diferencias clave en cuanto a modelo de datos, lenguaje de consulta, flexibilidad de esquemas, transacciones, integridad, escalabilidad y rendimiento. Las relacionales, basadas en SQL y el modelo ACID, son ideales para transacciones complejas y datos estructurados, mientras que las NoSQL priorizan el rendimiento y la disponibilidad, adoptando el modelo BASE y soportando estructuras dinámicas.

También se realiza un análisis de los tipos de bases de datos NoSQL, cada uno asociado a un modelo específico de repositorio de datos:

Las bases de datos de documentos como MongoDB, que almacenan información en formato JSON o BSON, permiten esquemas flexibles y estructuras jerárquicas.

Las bases clave-valor como Redis, que ofrecen alta eficiencia en consultas simples y rápidas.

Las bases de columnas anchas como Cassandra, ideales para manejar grandes volúmenes de datos distribuidos.

Las bases de grafos como Neo4j, que representan relaciones complejas entre datos, usadas en redes sociales o detección de fraudes.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

En conclusión, esta unidad proporciona una visión sólida de las bases de datos NoSQL, contextualizando su origen, evolución y aplicaciones prácticas. Resalta su relevancia en el desarrollo de sistemas actuales que requieren alta flexibilidad, disponibilidad y rendimiento, y destaca la importancia de herramientas como MongoDB en la implementación exitosa de soluciones NoSQL.

¡Siga adelante! ¡Muchos éxitos!

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

8 UNIDAD 2 DISEÑO Y MANEJO DE BASES DE DATOS NoSQL

"El éxito llega para aquellos que están dispuestos a trabajar un poco más que el resto"

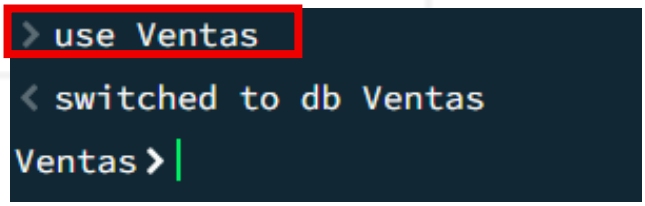
Og Mandino.

- Comandos básicos en MongoDB
- Operaciones CRUD en MongoDB
- Introducción a la Agregaciones
- Pipeline

8.1 Comandos básicos en MongoDB

Comando use: Mediante este comando se activa la base de datos en la que se desea trabajar.

Figura 9
Comando use

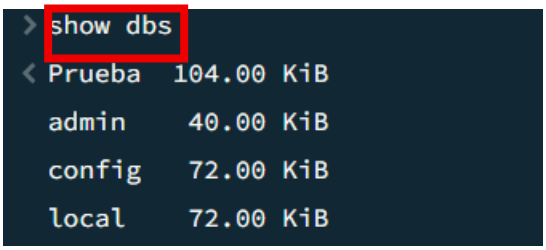


Nota. Pantalla que muestra en Mongo Shell usando el comando show dbs y enlista todas las bases de datos existentes. **Fuente:** Investigación propia.

Comando show: El comando show muestra o enlista. Si se usa show dbs muestra todas las bases de datos en forma de lista qe existan, si se usa show collection muestra todas las colecciones que tiene la base de datos activa.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

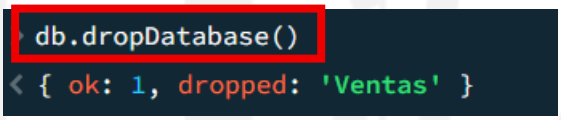
Figura 10
Comando show



Nota. Pantalla que muestra en Mongo Shell usando el comando show dbs y enlista todas las bases de datos existentes. **Fuente:** Investigación propia.

Comando dropDatabase: El comando db muestra la base de datos actualmente seleccionada. Si se usa juntamente con el comando dropDatabase se elimina la base de datos activa.

Figura 11
Comando dropDatabase



Nota. Pantalla que muestra en Mongo Shell usando el comando dropDatabase, donde se elimina la base de datos Ventas. **Fuente:** Investigación propia.

8.2 Operaciones CRUD en MongoDB

https://youtu.be/tixAeJUPoXI?si=rH8y8bOXkUCC_Hgh

Los comandos CRUD (Create, Read, Update, Delete) son fundamentales para interactuar con una base de datos MongoDB.

Comandos CREATE: Los comandos Create se utilizan para insertar nuevos documentos en una colección de MongoDB. Es importante mencionar que una colección es similar a una tabla en las bases de datos relacionales. MongoDB proporciona los siguientes métodos para insertar documentos en una colección:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- db.collection.insertOne(): Comando que permite insertar un solo documento en la colección especificada.

Figura 12
Comando insertOne

```
db.users.insertOne(
    {
        name: "sue",
        age: 26,
        status: "pending"
    }
)
```

Nota. Pantalla que muestra en Mongo Shell usando el comando para insertar un documento en la colección Users. **Fuente:** Investigación propia.

- db.collection.insertMany(): Comando que permite inserta varios documentos en la colección.

Figura 13
Comando insertMany

```
db.Producto.insertMany([
  {
    codigo: "P001",
    nombre: "Laptop Lenovo",
    precio: 850.00
  },
  {
    codigo: "P002",
    nombre: "Mouse Inalámbrico",
    precio: 25.50
  },
  {
    codigo: "P003",
    nombre: "Teclado Mecánico",
    precio: 60.00
  }
]);
```

Nota. Pantalla que muestra en Mongo Shell usando el comando para insertar tres documentos en la colección Producto. **Fuente:** Investigación propia.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Comandos READE: Las operaciones de lectura recuperan documentos de una colección; es decir, consultan documentos en una colección. MongoDB ofrece los siguientes métodos para leer documentos de una colección:

- db.collection.find(): Comando que se utiliza para consultar documentos en una colección y devuelve un cursor a los documentos que coinciden con la consulta

Figura 14
Comando find()

```
db.Producto.find();
```

```
[
  { "_id": ObjectId("..."), codigo: "P001", nombre: "Laptop Lenovo", precio: 850.00 },
  { "_id": ObjectId("..."), codigo: "P002", nombre: "Mouse Inalámbrico", precio: 25.50 },
  { "_id": ObjectId("..."), codigo: "P003", nombre: "Teclado Mecánico", precio: 60.00 }
]
```

Nota. Pantalla que muestra en Mongo Shell como se usa el comando find y el resultado donde se muestra los documentos que tiene la colección Producto en formato Json. **Fuente:** Investigación propia.

El comando find() en MongoDB es una herramienta versátil que permite realizar búsquedas tanto sin condiciones como con una variedad de condiciones. A continuación, se presentan algunos ejemplos de cómo utilizar este comando en diferentes escenarios:

Tabla 3
Diferentes usos del comando find

Comando	Resultado	Interpretación
db.usuarios.find()	Este comando devuelve todos los documentos en la colección usuarios.	Realiza una búsqueda sin condiciones en la tabla usuarios.
db.usuarios.find({ nombre: "Juan" })	Este comando devuelve todos los documentos donde el campo nombre es igual a "Juan".	Realiza una búsqueda con condición de igualdad
db.usuarios.find({ edad: { $gt: 25 } })	Este comando devuelve todos los documentos donde el campo edad es mayor que 25.	Realiza una búsqueda con condición numérica
db.usuarios.find({ nombre: "Juan", edad: 30 })	Este comando devuelve todos los documentos donde el campo nombre es "Juan" y el campo edad es 30.	Realiza una búsqueda con múltiples condiciones (AND implícito)
db.usuarios.find({ $or: [{ nombre: "Juan" }, { nombre: "Ana" }] })	Este comando devuelve todos los documentos donde el campo nombre es "Juan" o "Ana".	Realiza una búsqueda usando el comando OR

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

db.usuarios.find({ edad: { $gte: 25, $lte: 35 } })	Este comando devuelve todos los documentos donde el campo edad está entre 25 y 35, inclusive.	Realiza una búsqueda usando operadores matemáticos, condicionales o lógicos

Nota. Tabla que muestra diferentes usos del comando find() usando varios operadores. **Fuente:** Investigación propia.

Comandos UPDATE: Las operaciones de actualización modifican los documentos existentes en una colección. MongoDB ofrece los siguientes métodos para actualizar los documentos de una colección:

- db.collection.updateOne() : Actualiza un solo documento dentro de la colección según el filtro.

Figura 15
Comando updateOne

```
db.Producto.updateOne(
  { codigo: "P002" },  // condición
  { $set: { precio: 29.99 } }  // actualización
);
```

Nota. Pantalla que muestra en Mongo Shell como se usa el comando updateOne que actualiza el precio del producto con código "P002" a 29.99. **Fuente:** Investigación propia.

- db.collection.updateMany(): Actualiza todos los documentos que coinciden con el filtro especificado para una colección.

Figura 16
Comando updateMany

```
db.Producto.updateMany(
  { precio: { $gt: 50 } },
  { $inc: { precio: 10 } }  // incremento del valor
);
```

Nota. Pantalla que muestra en Mongo Shell como se usa el comando updatMany, que aumentar en 10 dólares el precio de todos los productos cuyo precio sea mayor a 50. **Fuente:** Investigación propia.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- db.collection.replaceOne(): Reemplaza un solo documento dentro de la colección según el filtro.

Figura 17
Comando replaceOne

```
db.Producto.replaceOne(
  { codigo: "P003" },
  {
    codigo: "P003",
    nombre: "Audífonos Gamer",
    precio: 45.00
  }
);
```

Nota. Pantalla que muestra en Mongo Shell como se usa el comando replaceOne, que reemplaza el producto con código "P003" por otro completamente diferente. **Fuente:** Investigación propia.

En conclusión, se usa updateOne cuando se quiere modificar un solo documento, updateMany cuando se quiere modificar varios documentos a la vez y replaceOne cuando se quiere reemplazar totalmente un documento por uno nuevo.

Comandos DELETE: Las operaciones de eliminación eliminan documentos de una colección. MongoDB ofrece los siguientes métodos para eliminar documentos de una colección:

- db.collection.deleteOne(): Elimina un solo documento de una colección.

Figura 18
Comando deleteOne

```
db.Producto.deleteOne({ nombre: "Mouse Inalámbrico" });
```

Nota. Pantalla que muestra en Mongo Shell como se usa el comando deleteOne, que elimina un producto cuyo nombre sea "Mouse Inalámbrico". **Fuente:** Investigación propia.

Es importante mencionar que este comando solo se elimina uno de los productos con nombre "Mouse Inalámbrico", incluso si hay varios.

- db.collection.deleteMany():Elimina todos los documentos que coinciden con una colección.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 19
Comando deleteMany

```
db.Producto.deleteMany({ precio: { $lt: 100 } });
```

Nota. Pantalla que muestra en Mongo Shell como se usa el comando deleteMany, que elimina todos los productos cuyo precio sea menor a 100. **Fuente:** Investigación propia.

8.3 Agregaciones en MongoDB

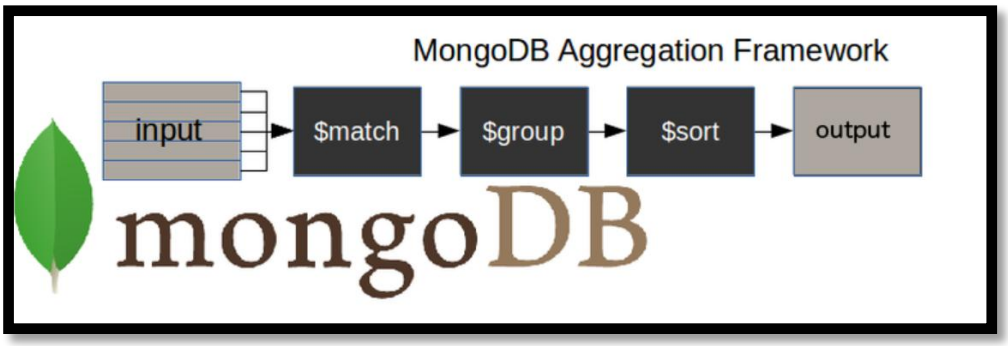
https://youtu.be/jggOZypWiBA?si=dIF-XMpir156Ac4q

Las agregaciones en MongoDB son operaciones que permiten procesar datos de una colección para obtener resultados computados o resumidos. A diferencia de las consultas simples que devuelven documentos individuales, las agregaciones permiten realizar operaciones complejas como agrupaciones, cálculos, transformaciones y más. El framework de agregación en MongoDB utiliza el concepto de "pipelines", que son una serie de etapas por las que pasan los documentos para ser procesados.

8.4 Pipeline de Agregación

Un pipeline de agregación es una secuencia de etapas (stages) donde cada etapa transforma los documentos a medida que pasan a través del pipeline. Cada etapa toma como entrada los documentos resultantes de la etapa anterior y aplica una operación específica. Los pipelines se definen como un arreglo de etapas, donde cada etapa es un objeto JSON.

Figura 20
Framework de agregación



Nota. Pantalla que muestra las etapas de un pipeline de agregacion: match, group y sort.

Fuente: *(Software Labs Ltd., 2024)*.

8.5 Etapas Comunes en un Pipeline

- $match: Filtra los documentos para pasar solo aquellos que cumplen con las condiciones especificadas.
- $group: Agrupa los documentos por un campo específico y aplica operaciones de acumulación.
- $sort: Ordena los documentos.
- $project: Selecciona y renombra campos, y puede crear nuevos campos calculados.
- $limit: Limita el número de documentos que pasan a la siguiente etapa.
- $skip: Omite un número específico de documentos.
- $unwind: Descompone un campo de tipo arreglo en múltiples documentos, uno por cada elemento del arreglo.

https://youtu.be/fX5lDf_d4FY?si=j6U9e4QDLepUdguz

Figura 21

Ejemplo de uso de Agregaciones -$match y $group

```
db.ventas.aggregate([
    {
        $match: { fecha: { $gte: ISODate("2023-01-01") } }
    },
    {
        $group: {
            _id: "$producto",
            totalVentas: { $sum: "$cantidad" }
        }
    }
])
```

Nota. Pantalla que muestra el script de agregación que permite calcular el total de ventas por producto, usando una agregacion con dos etapas:match y group. **Fuente:** Investigación propia.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 22
Ejemplo de uso de Agregaciones - $sort y $limit

```
db.ventas.aggregate([
    {
        $group: {
            _id: "$producto",
            totalVentas: { $sum: "$cantidad" }
        }
    },
    {
        $sort: { totalVentas: -1 }
    },
    {
        $limit: 5
    }
])
```

Nota. Pantalla que muestra el script de agregación que permite obtener los 5 productos más vendidos, usando una agregación con dos etapas: group sort y limit. **Fuente:** Investigación propia.

Figura 23
Ejemplo de uso de Agregaciones - $project

```
db.ventas.aggregate([
    {
        $project: {
            producto: 1,
            cantidad: 1,
            total: { $multiply: ["$precio", "$cantidad"] }
        }
    }
])
```

Nota. Pantalla que muestra el script de agregación que permite seleccionar solo ciertos campos y crear un nuevo campo calculado, usando una agregación con $project. **Fuente:** Investigación propia.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 24
Ejemplo de uso de Agregaciones - $unwind

```
db.pedidos.aggregate([
    {
        $unwind: "$productos"
    }
])
```

Nota. Pantalla que muestra el script de agregación donde se tiene una colección pedidos con documentos que contienen un arreglo de productos y se usa el comando unwind para descomponer este arreglo y tener un documento por cada producto en el pedido. **Fuente:** Investigación propia.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

8.6 Autoevaluación 2

Lea detenidamente la pregunta y marque la respuesta correcta entre las opciones dadas.

1. ¿Qué comando se utiliza para insertar un documento en una colección en MongoDB?

a) db.insertOne().

b) db.collection.insert().

c) db.collection.insertOne().

d) db.insertDocument().

2. ¿Qué comando permite visualizar todas las bases de datos existentes en un servidor MongoDB?

a) show collections

b) db.list()

c) db.getDatabases()

d) show dbs

3. ¿Cuál es la estructura básica de un documento en MongoDB?

a) Un arreglo de campos.

b) Un conjunto de columnas.

c) Un objeto JSON o BSON.

d) Una cadena de texto.

4. ¿Qué comando se utiliza para actualizar campos de un documento específico?

a) update()

b) updateOne()

c) setField()

d) modify()

5. En MongoDB, ¿qué operador se utiliza para modificar el valor de un campo existente durante una actualización?

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

a) $inc

b) $push

c) $set

d) $modify

6. ¿Qué comando elimina un solo documento que coincide con una condición dada?

a) removeOne()

b) delete()

c) deleteOne()

d) drop()

7. ¿Cuál es la función del método find() en MongoDB?

a) Insertar un nuevo documento

b) Actualizar múltiples documentos

c) Buscar documentos que cumplan una condición

d) Crear una nueva colección

8. ¿Cuál de las siguientes afirmaciones describe mejor una agregación en MongoDB?

a) Una consulta SQL tradicional

b) Una operación para eliminar documentos

c) Un operador para insertar muchos datos

d) Un proceso para transformar y analizar datos

9. ¿Qué operador de etapa de agregación se utiliza para agrupar documentos?

a) $match

b) $sort

c) $group

d) $limit

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

10. ¿Qué etapa del pipeline de agregación se utiliza para filtrar documentos al inicio del proceso?

a) $match

b) $limit

c) $project

d) $sort

Si ha concluido la autoevaluación:

> Verifique sus respuestas en el solucionario que se encuentra al final de la presente Guía.

Si alcanzó un porcentaje alto de respuestas correctas en las preguntas de evaluación, es hora de continuar con la Unidad 3, caso contrario, vuelva a revisar los temas en los que ha tenido dificultad.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

8.7 Resumen de la Unidad 2

La unidad 2 trata sobre los comandos fundamentales que permiten manipular datos en MongoDB, así como herramientas avanzadas para el procesamiento y análisis de la información. Los comandos CRUD (Crear, Leer, Actualizar y Eliminar) constituyen la base del trabajo con MongoDB, ya que permiten realizar operaciones tanto básicas como complejas sobre los documentos almacenados. A través del comando de creación, es posible insertar nuevos documentos en una colección utilizando métodos que permiten añadir uno o varios registros según se requiera. En cuanto a la lectura, MongoDB ofrece mecanismos para consultar los datos almacenados mediante búsquedas simples o condicionales que permiten recuperar la información de forma precisa y eficiente. Para la actualización, se pueden modificar documentos existentes aplicando cambios puntuales a un documento individual o a varios documentos que cumplan una condición. Finalmente, con los comandos de eliminación se pueden borrar documentos específicos o grupos completos de registros, lo que permite mantener la integridad y relevancia de los datos almacenados.

Además de los comandos CRUD, se incorporan herramientas de agregación que amplían considerablemente la capacidad de análisis de datos. El framework de agregaciones en MongoDB permite transformar y resumir grandes volúmenes de datos mediante un conjunto de etapas encadenadas conocidas como pipelines. Cada etapa realiza una operación específica sobre los documentos que recibe, y los resultados son pasados a la siguiente etapa, permitiendo construir procesos de análisis complejos de forma estructurada. Entre estas operaciones se incluye el filtrado de documentos según criterios definidos, la agrupación por campos para aplicar cálculos agregados, la ordenación de resultados, la selección y transformación de campos, la limitación y omisión de documentos en el flujo, así como la descomposición de campos tipo arreglo para tratarlos individualmente. Estas herramientas permiten realizar análisis profundos directamente sobre la base de datos, reduciendo la necesidad de procesamiento adicional en capas externas.

En conjunto, el uso de comandos CRUD junto con el framework de agregaciones proporciona un conjunto completo y flexible de herramientas para manipular, consultar y analizar datos en MongoDB. Esto convierte a MongoDB en una solución potente y adaptable para una amplia variedad de aplicaciones, especialmente aquellas que requieren manejar grandes volúmenes de datos con agilidad y eficiencia.

¡Siga adelante! ¡Muchos éxitos!

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

9 UNIDAD 3 ADMINISTRACIÓN Y SEGURIDAD DE BASE DE DATOS NoSQL

"Si no persigues lo que quieres, nunca lo tendrás. Si no vas hacia delante, siempre estarás en el mismo lugar."

Nora Roberts

- **Indexación en Mongo DB**
- **Procesos de backups y restauracion del servidor**
- **Gestión de Usuarios y Roles en MongoDB**
- **Mongo en la nube**
- **Bases de datos moviles**

9.1 La indexación en MongoDB

La indexación en MongoDB es una técnica utilizada para mejorar el rendimiento de las consultas en una colección. Los índices permiten a MongoDB encontrar y recuperar documentos de manera más eficiente, similar a cómo un índice en un libro permite encontrar rápidamente la información deseada sin tener que leer todo el libro. Sin índices, MongoDB debe realizar un escaneo completo de la colección para seleccionar los documentos que coinciden con una consulta, lo cual puede ser lento y consumir muchos recursos. Es importante mencionar que los índices en MongoDB sirven para:

- Mejorar el rendimiento de las consultas: Al reducir la cantidad de datos que MongoDB debe examinar, los índices pueden acelerar significativamente las operaciones de lectura.
- Optimizar el uso de recursos: Las consultas con índices adecuados consumen menos CPU y memoria.
- Ordenar resultados de manera eficiente: Los índices permiten ordenar los resultados de una consulta sin necesidad de realizar una operación de ordenamiento adicional.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

 https://youtu.be/fmYfdfbeX44?si=8kxrDpNX8HShpM1G

Para crear un índice en MongoDB, se utiliza el método createIndex() en una colección.

Figura 25
Comando createIndex

```
db.Producto.createIndex({ codigo: 1 });
```

Nota. Pantalla que muestra el uso del comando createIndex que crea un índice sobre el campo código, indicando que será en orden ascendente – parámetro 1. **Fuente:** Investigación propia.

9.2 Tipos de índices

Existen varios tipos de índices en MongoDB, cada uno con sus propias características y casos de uso:

Tabla 4
Tipos de índices

Indice	Comando	Interpretación
Índice de campo único	db.usuarios.createIndex({ nombre: 1 });	Índice en un solo campo de un documento.
Índice compuesto	db.usuarios.createIndex({ nombre: 1, edad: -1 });	Índice en múltiples campos de un documento.
Índice de múltiples claves	db.productos.createIndex({ etiquetas: 1 });	Índice en un campo que contiene un arreglo. MongoDB crea un índice separado para cada elemento del arreglo.
Índice de texto	db.articulos.createIndex({ contenido: "text" });	Índice que permite realizar búsquedas de texto en el contenido de los documentos.
Índice de hash	db.usuarios.createIndex({ email: "hashed" });	Índice que utiliza una función de hash para indexar el valor de un campo.
Índice TTL	db.logs.createIndex({ fecha: 1 }, { expireAfterSeconds: 3600 });	Índice que automáticamente elimina documentos después de un cierto período de tiempo.
Índice geoespacial	db.lugares.createIndex({ ubicacion: "2dsphere" });	Índice que permite realizar consultas geoespaciales en datos geográficos.

Nota. Tabla que muestra los diferentes tipos de índices en MongoDB con sus respectivos scripts.

Fuente: Investigación propia.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

9.3 Procesos de Backup y Restauración en MongoDB

Los procesos de backup y restauración son esenciales para la gestión de datos en cualquier sistema de base de datos, incluyendo MongoDB. Estos procesos aseguran que los datos puedan ser recuperados en caso de fallos del sistema, errores humanos o desastres naturales. A continuación, se presentan los conceptos clave relacionados con los procesos de backup y restauración en MongoDB.

9.4 Backup (Copia de Seguridad)

https://youtu.be/hg8OKEhjwUk?si=y2u1eUoJRU6c9CuQ

El backup en MongoDB es el proceso de crear una copia de los datos almacenados en una base de datos para que puedan ser restaurados en caso de pérdida de datos. MongoDB ofrece varias herramientas y métodos para realizar backups, incluyendo mongodump, que es una utilidad de línea de comandos para crear copias de seguridad de las bases de datos.

9.5 Restore (Restauración)

La restauración es el proceso de recuperar los datos desde una copia de seguridad. En MongoDB, la utilidad mongorestore se utiliza para restaurar datos desde un backup creado con mongodump.

9.5.1 Scripts para-Backup usando mongodump

Mongodump es una herramienta de línea de comandos que se utiliza para crear copias de seguridad de las bases de datos en MongoDB.

Figura 26
Script mongodump

```
mongodump --db Prueba1 --out C:\Respaldos\respaldo
```

Nota. Pantalla que muestra el uso del script del comando mongodump. **Fuente:** Investigación propia.

Este comando tiene los siguientes parametros:

mongodump -- Comando para realizar el backup

--db -- Indica el nombre de la base de datos de la que se va a realizar el respaldo.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

--out -- C:\Respaldos\respaldo -- Especifica la ruta de salida donde se guardará el respaldo.

9.5.2 Scripts para-Restauracion usando mongorestore

Mongorestore es la herramienta de línea de comandos que se utiliza para restaurar datos desde un backup creado con mongodump.

Figura 27
Script mongorestore

```
mongorestore --db Prueba1 C:\Respaldos\respaldo\Prueba1
```

Nota. Pantalla que muestra el uso del script del comando mongorestore. **Fuente:** Investigación propia.

Este comando tiene los siguientes parámetros:

mongorestore -- Comando para realizar la restauración

--db -- Indica el nombre de la base de datos de la que se va a realizar la restauración.

C:\Respaldos\respaldo \ Prueba1 -- Especifica la ruta del respaldo que contiene los archivos de la base de datos..

9.5.3 Gestión de Usuarios y Roles en MongoDB

https://youtu.be/ajO-iFYMUlA?si=r5bGGTxD5pYoHXRn

La gestión de usuarios y roles en MongoDB es un aspecto fundamental para asegurar la integridad y seguridad de los datos almacenados en las bases de datos. MongoDB, como una de las bases de datos NoSQL más populares, ofrece un sistema robusto y flexible para la administración de usuarios y la asignación de roles, permitiendo a los administradores controlar de manera precisa quién puede acceder a qué datos y qué operaciones pueden realizar. Este sistema no solo facilita la implementación de políticas de seguridad, sino que también ayuda a organizar y gestionar eficientemente los permisos en entornos con múltiples usuarios y bases de datos. A través de la creación de usuarios, la definición de roles personalizados y la asignación de privilegios específicos, MongoDB permite adaptar el acceso a las necesidades particulares de cada proyecto, asegurando que cada usuario tenga

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

exactamente los permisos necesarios para llevar a cabo sus tareas sin comprometer la seguridad del sistema.

A continuación, conceptos importantes sobre la Gestión de usuarios.

- Usuarios: En MongoDB, un usuario es una entidad que tiene permisos para interactuar con la base de datos. Cada usuario tiene un nombre de usuario y una contraseña, y está asociado a una o más bases de datos.
- Roles: Los roles en MongoDB definen los privilegios que un usuario tiene dentro de una base de datos. Los roles pueden ser predefinidos o personalizados.
- Autenticación: Proceso mediante el cual se verifica la identidad de un usuario. MongoDB soporta varios mecanismos de autenticación, como SCRAM (Salted Challenge Response Authentication Mechanism).
- Autorización: Proceso mediante el cual se determinan los permisos y accesos que un usuario autenticado tiene dentro de la base de datos.
- Base de Datos Administración: La base de datos admin es especial en MongoDB y se utiliza para la administración del sistema. Los usuarios creados en esta base de datos pueden tener permisos en todas las bases de datos.

Además, MongoDB proporciona una serie de roles predefinidos que cubren las necesidades más comunes de gestión de usuarios:

- read: Permite leer datos de una base de datos.
- readWrite: Permite leer y escribir datos en una base de datos.
- dbAdmin: Proporciona privilegios de administración en una base de datos, como la creación de índices.
- userAdmin: Permite gestionar usuarios en una base de datos.
- clusterAdmin: Proporciona privilegios de administración en el clúster, como la gestión de réplicas y sharding.
- readAnyDatabase: Permite leer datos de cualquier base de datos en el clúster.
- readWriteAnyDatabase: Permite leer y escribir datos en cualquier base de datos en el clúster.
- userAdminAnyDatabase: Permite gestionar usuarios en cualquier base de datos en el clúster.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- dbAdminAnyDatabase: Proporciona privilegios de administración en cualquier base de datos en el clúster.
- root: Proporciona todos los privilegios en el clúster.

9.6 Crear un Usuario

9.6.1 El método db.createUser() permite crear un nuevo usuario dentro del sistema de MongoDB

Figura 28
Comando createUser

```
use admin
db.createUser({
  user: "usuario1",
  pwd: "contraseña1",
  roles: [
    { role: "readWrite", db: "basededatos1" },
    { role: "read", db: "basededatos2" }
  ]
})
```

Nota. Pantalla que muestra el script del comando createUser crea el usuario "usuario1" con permisos de lectura y escritura. **Fuente:** Investigación propia.

9.6.2 Autenticación como Usuario

El comando db.auth() se utiliza para autenticar a un usuario proporcionando su nombre y contraseña.

Figura 29
Comando auth

```
use basededatos1
db.auth("usuario1", "contraseña1")
```

Nota. Pantalla que muestra el script del comando auth donde el usuario usuario1 se autentica en la base basededatos1. **Fuente:** Investigación propia.

9.6.3 Crear un rol personalizado

El comando createRole se utiliza para crear un rol personalizado con privilegios específicos.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 30
Comando createRole

Nota. Pantalla que muestra el script del comando createRole donde se crea el rol "rolPersonalizado" y se otorga permisos para buscar (find), insertar (insert) y actualizar (update) documentos en la colección coleccion1 de la base basededatos1. **Fuente:** Investigación propia.

Asignar un Rol a un Usuario

Con el comando grantRolesToUser, se asigna un nuevo rol a un usuario existente.

Figura 31
Comando grantRolesToUser

Nota. Pantalla que muestra como se otorga al "usuario1" el acceso definido por el rol "rolPersonalizado" sobre la base de datos basededatos1. **Fuente:** Investigación propia.

Listar Usuarios

El comando db.getUsers(), muestra una lista de todos los usuarios registrados en la base actual (en este caso, admin). Incluye información como los roles asignados y las bases de datos donde tienen acceso. Este comando es útil para hacer auditorías de seguridad y revisar los permisos existentes en el sistema.

Listar Roles

El comando db.getRoles() recupera los roles definidos en la base de datos actual. Si se usa showBuiltinRoles: true, también muestra los roles incorporados por defecto en MongoDB (como read, readWrite, dbAdmin, etc.). Este comando sirve para verificar qué roles existen y entender su alcance antes de asignarlos o modificarlos.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Eliminar un Usuario

El comando db.dropUser() elimina al usuario especificado. Este comando revoca permanentemente el acceso del usuario y se usa generalmente cuando alguien ya no forma parte del equipo o se quiere deshabilitar una cuenta por motivos de seguridad.

Figura 32
Comando dropUser

```
use admin
db.dropUser("usuario1")
```

Nota. Pantalla que muestra cómo se elimina el usuario "usuario1", de la base de datos actual. **Fuente:** Investigación propia.

Eliminar un Rol Personalizado

El comando db.dropRole() elimina un rol personalizado del sistema. Este comando se usa cuando un rol ya no se necesita o ha sido reemplazado por otro más adecuado. Solo se puede eliminar si no está en uso por algún usuario.

Figura 33
Comando dropRole

```
use admin
db.dropRole("rolPersonalizado")
```

Nota. Pantalla que muestra cómo se elimina el rol "rolPersonalizado". **Fuente:** Investigación propia.

9.7 Mongo en la Nube

MongoDB en la nube se refiere a la implementación y gestión de bases de datos MongoDB utilizando servicios en la nube. Esto permite a las organizaciones aprovechar las ventajas de la computación en la nube, como la escalabilidad, la flexibilidad y la reducción de costos operativos. MongoDB Atlas es la solución más conocida y utilizada para MongoDB en la nube, proporcionada directamente por MongoDB Inc. A continuación, se presentan los conceptos clave relacionados con MongoDB en la nube.

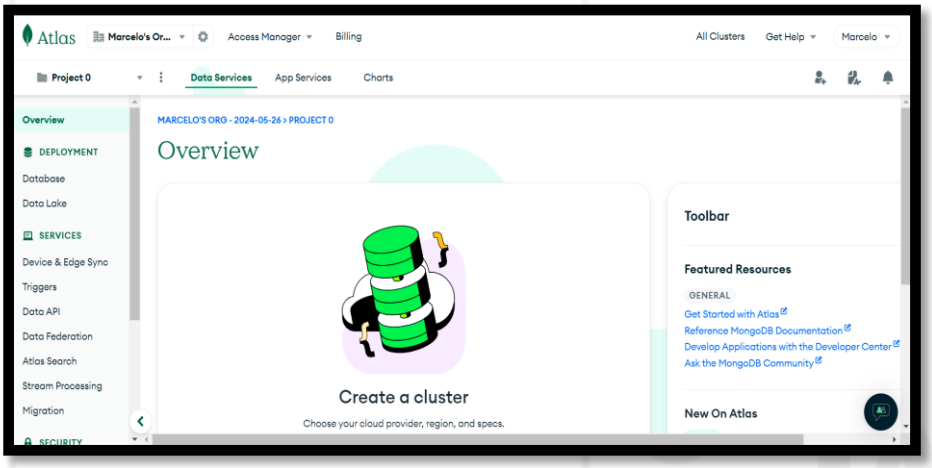
(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

9.7.1 MongoDB Atlas

https://youtu.be/s4Kn95WXnwU?si=vjJGc-kPDfXvV-Sv

Según (Strauch, 2011), MongoDB Atlas es un servicio de base de datos como servicio (DBaaS) que permite a los usuarios desplegar, gestionar y escalar bases de datos MongoDB en la nube. Atlas está disponible en múltiples proveedores de servicios en la nube, incluyendo AWS, Google Cloud Platform y Microsoft Azure.

Figura 34
Pantalla de MongoDB Atlas



Nota. Pantalla que muestra la pantalla principal de Mongo Atlas. **Fuente:** Investigación propia.

MongoDB Atlas ofrece una plataforma de base de datos como servicio que se adapta a las necesidades dinámicas de las aplicaciones modernas. Entre sus principales características se encuentra la escalabilidad automática, que permite ajustar los recursos de almacenamiento y procesamiento según la demanda sin generar interrupciones en el servicio. Esta capacidad resulta especialmente útil en entornos con cargas de trabajo variables, ya que garantiza un rendimiento constante. Asimismo, Atlas proporciona alta disponibilidad mediante la replicación de datos en múltiples nodos y zonas geográficas, lo que asegura el acceso continuo a la información incluso en caso de fallos en parte de la infraestructura. En cuanto a la seguridad, incorpora mecanismos avanzados como el cifrado en tránsito y en reposo, la autenticación multifactor y los controles de acceso basados en roles, protegiendo los datos de accesos no autorizados. Además, cuenta con backups automáticos que permiten realizar copias de seguridad programadas y restauraciones eficientes ante cualquier incidente. El

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

sistema de monitoreo y alertas en tiempo real facilita el seguimiento del rendimiento y la estabilidad del sistema, ayudando a prevenir problemas antes de que afecten a los usuarios. También destaca por su integración con herramientas de big data y análisis, lo que permite aprovechar los datos almacenados para procesos analíticos directamente desde la nube.

El uso de MongoDB en la nube aporta beneficios significativos a las organizaciones, comenzando por la reducción de costos operativos al eliminar la necesidad de gestionar infraestructura física y de mantener equipos dedicados exclusivamente a la administración de bases de datos. Este modelo también proporciona escalabilidad y flexibilidad, ya que los recursos pueden ajustarse rápidamente según el crecimiento del negocio o los cambios en la demanda. La accesibilidad que ofrece permite trabajar con los datos desde cualquier ubicación y en cualquier momento, lo cual resulta clave para equipos distribuidos. Además, los servicios en la nube están preparados para la recuperación ante desastres, garantizando la continuidad del negocio incluso en situaciones críticas. Otro aspecto relevante es que las actualizaciones y el mantenimiento del sistema son gestionados por el proveedor, liberando a los equipos internos para enfocarse en actividades de mayor valor.

En cuanto a sus aplicaciones prácticas, MongoDB en la nube se utiliza ampliamente en el desarrollo de aplicaciones web y móviles que requieren bases de datos ágiles y escalables capaces de manejar grandes volúmenes de datos y usuarios simultáneos. También es una herramienta poderosa para el análisis de datos en tiempo real, permitiendo a las empresas obtener información inmediata a partir de grandes flujos de datos. En el ámbito del Internet de las Cosas (IoT), MongoDB se adapta perfectamente al procesamiento y almacenamiento de datos generados por sensores y dispositivos conectados. Finalmente, en plataformas de gestión de contenidos, su capacidad de escalar y adaptarse a distintos tipos de datos lo convierte en una opción ideal para manejar grandes catálogos digitales y volúmenes elevados de usuarios. MongoDB en la nube, en resumen, ofrece una solución robusta, flexible y eficiente para una amplia variedad de necesidades tecnológicas actuales.

9.8 Autoevaluación 3

Lea detenidamente la pregunta y marque la respuesta correcta entre las opciones dadas.

1. ¿Cuál es el propósito principal de los índices en MongoDB?

a) Aumentar el tamaño del documento

b) Eliminar datos duplicados

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

c) Mejorar el rendimiento de las consultas

d) Encriptar los datos de la base

2. ¿Qué tipo de índice se crea automáticamente en el campo _id de una colección?

a) Índice compuesto

b) Índice geoespacial

c) Índice de texto

d) Índice único

3. ¿Qué comando se utiliza para realizar una copia de seguridad (backup) de una base de datos en MongoDB?

a) mongobackup

b) mongoexport

c) mongodump

d) mongoback

4. ¿Qué comando se utiliza para restaurar una base de datos a partir de un respaldo en MongoDB?

a) mongorestore

b) mongoload

c) restoremongo

d) mongoimport

5. ¿Qué comando permite crear un nuevo usuario con credenciales en MongoDB?

a) createUser()

b) addUser()

c) db.createUser()

d) insertUser()

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

6. ¿ Qué rol de MongoDB permite al usuario realizar todas las operaciones administrativas en una base de datos?

a) read

b) readWrite

c) dbAdmin

d) userAdmin

7. ¿Qué comando permite ver todos los usuarios registrados en una base de datos?

a) db.listUsers()

b) show users

c) getUsers()

d) users.list()

8. ¿Cuál es el objetivo de los roles personalizados en MongoDB?

a) Cambiar el nombre de los usuarios

b) Controlar el acceso a datos específicos según funciones

c) Mejorar la velocidad de indexación

d) Comprimir los datos en la nube

9. ¿Qué comando permite eliminar un rol personalizado en MongoDB?

a) db.dropRole()

b) deleteRole()

c) dropCustomRole()

d) removeRole()

10. ¿Cuál es una ventaja principal de usar MongoDB Atlas?

a) Solo funciona sin conexión a internet

b) Requiere instalación local obligatoria

c) Permite desplegar MongoDB en la nube con alta disponibilidad

d) Solo permite consultas de lectura

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Si ha concluido la autoevaluación:

> Verifique sus respuestas en el solucionario que se encuentra al final de la presente Guía.

Si alcanzó un porcentaje alto de respuestas correctas en las preguntas de evaluación, ha finalizado este módulo de Base de datos no relacionales, caso contrario, vuelva a revisar los temas en los que ha tenido dificultad.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

9.9 Resumen de la Unidad 3

La Unidad 3 abarca aspectos fundamentales para la administración avanzada y segura de bases de datos en MongoDB, centrándose en la indexación, los procesos de respaldo y restauración, el uso de MongoDB en la nube y la gestión de usuarios y roles.

La indexación es una técnica esencial para mejorar el rendimiento de las consultas. Al crear índices adecuados, se optimiza la velocidad de lectura y se reduce el uso de recursos del sistema. MongoDB proporciona distintos tipos de índices para diferentes necesidades, lo que permite personalizar la optimización de las operaciones de consulta según el caso de uso.

Los procesos de backup y restauración son clave para garantizar la integridad y disponibilidad de los datos. Herramientas como mongodump y mongorestore, junto con las funciones integradas de MongoDB Atlas, permiten realizar copias de seguridad y restauraciones eficientes. Estos procedimientos son parte fundamental de cualquier estrategia de continuidad operativa y recuperación ante desastres.

El uso de MongoDB en la nube, especialmente a través de MongoDB Atlas, proporciona una solución completa para la gestión de bases de datos alojadas en entornos cloud. Con funcionalidades como escalabilidad automática, alta disponibilidad, seguridad avanzada y copias de seguridad automatizadas, MongoDB Atlas permite a las organizaciones centrarse en el desarrollo de sus aplicaciones mientras delegan la administración de la infraestructura.

La unidad también aborda la gestión de usuarios y roles, lo que permite controlar con precisión el acceso a las bases de datos. A través de comandos como db.createUser() y db.auth(), se puede crear y autenticar usuarios con distintos niveles de permisos. Asimismo, con db.createRole() es posible definir roles personalizados con privilegios específicos sobre bases y colecciones, los cuales pueden asignarse mediante db.grantRolesToUser(). Para la administración, se cuenta con comandos como db.getUsers(), db.getRoles(), db.dropUser() y db.dropRole().

En conjunto, los contenidos de esta unidad dotan al estudiante de herramientas prácticas y conceptos clave para mantener bases de datos MongoDB seguras, eficientes y adaptadas a los entornos actuales, tanto locales como en la nube.

¡Siga adelante! ¡Muchos éxitos!

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

10. REFERENCIAS

Codd, E. (1970). A relational model of data for large shared data bank.

Díaz Salvo, J. M. (2016). *Utilización de las bases de datos relaciones en el sistema de gestión y almacenamiento de datos.* Editorial Tutor Formación.

Grolinger, K. H. (2013). Data management in cloud environments: NoSQL. *Journal of Cloud Computing: Advances, Systems and Applications* .

Hecht, R. &. (2011). NoSQL evaluation: A use case-oriented survey.

Li, F. &. (2013). A performance comparison of SQL and NoSQL databases.

Moniruzzaman, A. B. (2013). NoSQL database: New era of databases for big data analytics – Classification, characteristics and comparison. International Journal of Database Theory and Application.

Sadalage, P. J. (2013). *NoSQL distilled: A brief guide to the emerging world of polyglot persistence. .* Addison-Wesley.

Sarasa, A. (2016). *Introducción a las bases de datos NoSQL usando MongoDB.* Editorial UOC.

Silberschatz, A. H. (2019). *Fundamentos De Bases De Datos (5a. Ed.).* McGraw-Hill.

Silberschatz, A. K. (2019). *Database System Concepts. .* McGraw-Hill.

Software Labs Ltd. (2024). Obtenido de The Beginner's Guide to MongoDB Aggregation (With Exercise): https://studio3t.com/knowledge-base/articles/mongodb-aggregation-framework/

Strauch, C. (2011). *NoSQL databases.* Stuttgart Media University.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

11. SOLUCIONARIO

AUTOEVALUACIÓN 1

PREGUNTA	Respuesta	Retroalimentación
1	c.	Las bases de datos relacionales funcionan bien con datos estructurados, pero fallan al manejar grandes volúmenes de datos no estructurados o semiestructurados como imágenes, videos o texto libre. NoSQL surgió para resolver estos desafíos, brindando flexibilidad y escalabilidad.
2	d.	NoSQL permite trabajar sin un esquema fijo, lo cual es ideal para datos cambiantes. Además, permite escalar horizontalmente (añadiendo más servidores), lo cual no es sencillo en sistemas relacionales tradicionales.
3	d.	Las bases de datos de grafos representan entidades como nodos y sus conexiones como relaciones o aristas, facilitando el análisis de redes sociales, rutas, o relaciones complejas entre datos.
4	a.	En MongoDB, una colección agrupa documentos (que serían similares a filas), y no requiere que todos los documentos tengan el mismo esquema.
5	b.	MongoDB Compass es una herramienta gráfica oficial que permite visualizar, consultar y administrar tus bases de datos sin necesidad de usar línea de comandos.
6	c.	El operador $or evalúa si al menos una de las condiciones dadas es verdadera, y devuelve los documentos que cumplan con esa lógica.
7	d.	Las bases de datos móviles están diseñadas para funcionar sin conexión, sincronizándose luego con el servidor. Son esenciales en apps móviles donde la conectividad no siempre es estable.
8	a.	El comando use permite cambiar de base de datos o crear una nueva si esta no existe. Es fundamental para empezar a trabajar con una base de datos específica en MongoDB.
9	b.	MongoDB está basado en documentos JSON (BSON internamente), lo que lo hace muy adecuado para manejar grandes volúmenes de datos semiestructurados, como catálogos, perfiles o registros.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

10	a.	$exists se utiliza para verificar si un campo específico está presente (existe) o no dentro de un documento en MongoDB.

AUTOEVALUACIÓN 2

PREGUNTA	Respuesta	Retroalimentación
1	c.	insertOne() permite agregar un solo documento a una colección específica en MongoDB. Es parte de los comandos básicos de inserción.
2	d.	Este comando muestra todas las bases de datos disponibles en el servidor. Es útil para explorar el entorno de trabajo.
3	c.	MongoDB almacena datos en documentos de tipo BSON (una extensión binaria de JSON), lo que permite gran flexibilidad.
4	b.	updateOne() permite actualizar un solo documento que coincida con un criterio. Se usa comúnmente junto a $set.
5	c.	$set se emplea para asignar un nuevo valor a un campo existente o crear uno nuevo si no existe.
6	c.	deleteOne() elimina el primer documento que cumpla con la condición especificada. Para eliminar múltiples se usa deleteMany().
7	c.	find() es utilizado para consultar documentos dentro de una colección. Puede devolver uno o varios documentos.
8	d.	Las agregaciones permiten realizar operaciones avanzadas como agrupaciones, sumas, filtros y proyecciones dentro de una colección.
9	c.	$group permite agrupar documentos por un campo común y aplicar funciones como sumas, promedios y conteos.
10	a.	$match se usa al comienzo del pipeline para filtrar los documentos que serán procesados por las etapas siguientes.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

AUTOEVALUACIÓN 3

PREGUNTA	Respuesta	Retroalimentación
1	c.	Los índices permiten que las búsquedas se realicen más rápidamente al evitar escaneos completos de colecciones.
2	d.	MongoDB crea un índice único por defecto en el campo _id para garantizar que cada documento tenga un identificador único.
3	c.	mongodump exporta los datos de una base de datos en formato BSON, útil para realizar respaldos completos.
4	a.	mongorestore permite importar archivos generados con mongodump y restaurar el estado de la base de datos.
5	c.	db.createUser() se utiliza para crear usuarios en una base de datos específica y asignarles roles de acceso.
6	d.	El rol userAdmin permite crear y gestionar usuarios dentro de la base de datos, incluyendo la asignación de roles.
7	a.	db.getUsers() devuelve un arreglo con la información de los usuarios existentes en la base de datos actual.
8	c.	Los roles personalizados permiten crear conjuntos de permisos específicos según las necesidades de la organización o el sistema.
9	a.	db.dropRole() elimina un rol personalizado creado previamente. Solo puede ejecutarlo un usuario con los permisos adecuados.
10	c.	MongoDB Atlas es un servicio gestionado en la nube que ofrece escalabilidad, respaldo automático y seguridad integrada, sin necesidad de administrar la infraestructura física.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC