



INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:
Educación

TALLER MONGODB NOSQL

Presentado por:

STIVEN YESID DIAS LOPEZ
YULY CAROLINA CUADROS ORTEGA

Presentado a:
ING. BRAYAN ARCOS

Materia:
BD_ALMACENAMIENTO_MASIVO

Semestre:
OCTAVO A



El **Saber** como **Arma** de **Vida**



INSTITUTO TECNOLÓGICO DEL PUTUMAYO
MOCO-A-PUTUMAYO
2024

ÍNDICE

1. Resumen Ejecutivo
2. Introducción
 - 2.1 Contexto y Motivación
 - 2.2 Alcance del Informe
 - 2.3 Objetivos
3. Metodología
 - 3.1 Herramientas Utilizadas
 - 3.2 Procedimientos
4. Desarrollo del Informe
 - 4.1 Descripción de la Base de Datos
 - 4.2 Consultas NoSQL
 - 4.2.1 Consultas Realizadas
 - 4.2.2 Resultados de Consultas
 - 4.2.3 Explicación de Consultas
 - 4.3 Diseño de Base de Datos
5. Análisis y Discusión
 - 5.1 Interpretación de Resultados
6. Conclusiones
7. Recomendaciones
8. Referencias

1. Resumen Ejecutivo

Este informe detalla el proceso de creación, manipulación y exportación de una base de datos NoSQL utilizando MongoDB. A lo largo del proyecto, se trabajó con varias colecciones que representan empleados, departamentos, proyectos y tareas dentro de una empresa ficticia. Se implementaron operaciones CRUD (Crear, Leer, Actualizar y Eliminar) y se realizaron consultas avanzadas para analizar los datos almacenados.

El diseño de la base de datos se centró en la flexibilidad y escalabilidad, características clave de las bases de datos NoSQL¹. Se consideraron varios factores durante la construcción del esquema, incluyendo la estructura de los datos, las relaciones entre las colecciones y la eficiencia de las consultas. Los resultados obtenidos a través de las consultas proporcionaron información valiosa sobre la organización y gestión de los datos dentro de la empresa ficticia.

El **Saber** como Arma de Vida

2. Introducción.

2.1 Contexto y Motivación

La popularidad de las bases de datos NoSQL ha crecido exponencialmente en los últimos años debido a su capacidad para manejar grandes volúmenes de datos estructurados y no estructurados. A diferencia de las bases de datos relacionales tradicionales, las bases de datos NoSQL, como MongoDB, ofrecen una mayor flexibilidad en el diseño del esquema y permiten una escalabilidad horizontal más eficiente. Esto es particularmente útil en aplicaciones modernas que requieren un manejo dinámico de datos, como las redes sociales, el comercio electrónico y los sistemas de análisis en tiempo real.

2.2 Alcance del Informe

Este informe se centra en la implementación de una base de datos NoSQL utilizando MongoDB. Se abordan las siguientes áreas clave:

- **Creación de la base de datos:** Configuración inicial y creación de colecciones que representan diferentes entidades dentro de una empresa ficticia.
- **Operaciones CRUD:** Implementación de operaciones básicas de Crear, Leer, Actualizar y Eliminar en las colecciones.
- **Consultas avanzadas:** Realización de consultas complejas para extraer y analizar datos específicos.
- **Diseño de la base de datos:** Consideraciones y mejores prácticas para el diseño del esquema de la base de datos.
- **Exportación de datos:** Métodos para exportar la base de datos para su uso en otros sistemas o para fines de respaldo.

2.3 Objetivos

- **Demostrar la creación y administración de bases de datos NoSQL en MongoDB:** Proporcionar una guía paso a paso sobre cómo configurar y gestionar una base de datos MongoDB.
- **Explorar la flexibilidad de MongoDB:** Mostrar cómo MongoDB puede manejar estructuras de datos complejas y cambiantes.
- **Resaltar la importancia del diseño eficiente:** Subrayar las mejores prácticas en el diseño de esquemas de bases de datos NoSQL para optimizar el rendimiento y la escalabilidad.
- **Proporcionar ejemplos prácticos:** Incluir ejemplos de consultas y operaciones que se pueden realizar en MongoDB para ilustrar su funcionalidad y utilidad en escenarios del mundo real.

El **Saber** como Arma de Vida

3. Metodología

3.1 Herramientas Utilizadas

- **MongoDB:** Para la creación y administración de la base de datos.
- **MongoDB Shell:** Para la ejecución de comandos y consultas.
- **MongoDB Compass:** Para la visualización gráfica y gestión de la base de datos.
- **Git:** Para la clonación y gestión de repositorios.

3.2 Procedimientos

1. Clonación del Repositorio: Se utilizó el comando git clone para clonar un repositorio de NoSQL que contenía los datos base para trabajar.

git clone https://github.com/stivendias28/repositorio-NOSql.git

2. Exportación de la Base de Datos: Se exportó la base de datos usando mongodump.

mongodump --db pequena_empresa --out

"C:\Users\Stiven_Lopez\OneDrive\Documentos\BASE_SQL\repositorio-mongo\repositorio-mongo\backups"

3. Consultas y Manipulación de Datos: Se realizaron consultas básicas y avanzadas usando la Shell de MongoDB para interactuar con las colecciones.

4. Desarrollo del informe

Desarrollo del Informe

4.1 Descripción de la Base de Datos

La base de datos pequeña_empresa incluye varias colecciones, tales como:

- **Empleados:** Registra la información de los empleados, como nombre, cargo y salario.
- **Departamentos:** Almacena los datos de los departamentos de la empresa.
- **Proyectos:** Contiene los detalles de proyectos asignados a los empleados.
- **Tareas:** Incluye las actividades específicas dentro de cada proyecto.

4.2 Consultas NoSQL

4.2.1 Consultas Realizadas

Se realizaron varias consultas NoSQL para gestionar y obtener datos:

1. Actualizar Jefe de Departamento

```
db.departamentos.updateOne(  
  { "nombre": "Marketing" },  
  { "$set": { "jefe_departamento": "Carlos Mendez" } }  
)
```

2. Actualizar Salario de un Empleado

```
db.empleados.updateOne(  
  { "nombre": "María Gómez" },  
  { "$set": { "salario": 45000 } }  
)
```


3. Buscar empleado con salario mayor a 50000

```
db.empleados.find({ "salario": { "$gt": 50000 } }).pretty()
```

4. Buscar empleados por nombre

```
db.empleados.find({ "nombre": "Juan Pérez" }).pretty()
```

5. Buscar todos lo empleados

```
db.empleados.find().pretty()
```

6. Crear colección

```
db.createCollection("nombrebases")
```

7. Crear datos

```
db.tareas.insertMany([
  { "proyecto_id": ObjectId(), "descripcion": "Desarrollar la interfaz", "fecha_inicio":
    "2023-02-01", "fecha_fin": "2023-03-01", "estado": "En progreso",
    "empleado_asignado": ObjectId() },
  { "proyecto_id": ObjectId(), "descripcion": "Pruebas de integración", "fecha_inicio":
    "2023-04-01", "fecha_fin": "2023-05-01", "estado": "Pendiente",
    "empleado_asignado": ObjectId() },
  { "proyecto_id": ObjectId(), "descripcion": "Implementación del backend",
    "fecha_inicio": "2023-05-01", "fecha_fin": "2023-06-01", "estado": "Pendiente",
    "empleado_asignado": ObjectId() }
])
```

8. Crear nuevo empleado

```
db.empleados.insertOne({
  "nombre": "Luis Martínez",
  "edad": 32,
  "departamento_id": ObjectId(),
  "email": "luis.martinez@empresa.com",
  "telefono": "555-7890",
  "cargo": "Diseñador",
})
```



```
"salario": 45000,  
"fecha_contratacion": "2022-02-20",  
"proyectos_asignados": [ObjectId()]  
})
```

9. Eliminar database

```
use pequena_empresa  
db.dropDatabase()
```

10. Eliminar departamentos

```
db.departamentos.deleteOne({ "nombre": "Ventas" })
```

11. Eliminar empleados por nombre

```
db.empleados.deleteOne({ "nombre": "Luis Martínez" })
```

12. Eliminar Proyecto Especifico

```
db.proyectos.deleteOne({ "nombre": "Implementación de CRM" })
```

4.2.2 Resultados de Consultas

Los resultados de las consultas mostraron una correcta manipulación de los datos, actualizando, eliminando e insertando documentos en las colecciones de manera efectiva.

4.2.3 Explicación de Consultas

Las consultas ejecutadas se basaron en la sintaxis de MongoDB para manipular documentos dentro de colecciones. Las consultas de actualización (updateOne) modificaron campos específicos en los documentos, mientras que find se utilizó para obtener resultados legibles. El método deleteOne se usó para eliminar documentos basados en criterios específicos.

Explicación de cada consulta realizada:

1. Esta consulta busca en la colección departamentos un documento cuyo campo nombre sea igual a "Marketing". Una vez encontrado, actualiza (o establece) el campo jefe_departamento con el valor "Carlos Mendez". La función updateOne se asegura de que solo se actualice un documento.
2. Aquí, se busca un documento en la colección empleados donde el campo nombre sea "María Gómez". Una vez localizado, se actualiza el campo salario con el valor 45000 usando el operador \$set, que permite actualizar campos específicos sin afectar otros campos del documento.
3. Esta consulta busca todos los empleados cuyo salario sea mayor que 50000, utilizando el operador \$gt (greater than). El método pretty() se usa para formatear los resultados de manera más legible
4. Esta consulta busca documentos en la colección empleados donde el campo nombre sea "Juan Pérez". De nuevo, el método pretty() se usa para formatear los resultados de manera legible.
5. Esta consulta devuelve todos los documentos de la colección empleados. El uso de find() sin argumentos indica que no se aplica ningún filtro. El método pretty() se usa para mostrar los resultados de forma más clara.
6. Este comando crea una nueva colección llamada "nombrebase". MongoDB generalmente crea colecciones de manera implícita cuando se insertan documentos, pero este método permite hacerlo de forma explícita.

7. Esta consulta inserta múltiples documentos en la colección tareas usando el método insertMany(). Cada tarea tiene varios campos, como proyecto_id, descripcion, fecha_inicio, estado, y empleado_asignado. El campo ObjectId() se usa para generar identificadores únicos para proyecto_id y empleado_asignado.
8. Esta consulta inserta un nuevo documento en la colección empleados usando el método insertOne(). Se especifican diversos campos, como nombre, edad, departamento_id, y salario, entre otros. ObjectId() se usa para generar identificadores únicos.
9. Este comando cambia el contexto a la base de datos pequena_empresa y luego la elimina por completo usando el método dropDatabase().
10. Esta consulta elimina un documento de la colección departamentos donde el campo nombre sea "Ventas". El método deleteOne() asegura que solo se elimine un documento.
11. Esta consulta elimina un documento en la colección empleados donde el campo nombre sea "Luis Martínez".
12. Esta consulta elimina un proyecto específico de la colección proyectos cuyo campo nombre sea "Implementación de CRM".

4.3 Diseño de Base de Datos

4.3.1 Consideraciones de Diseño

El diseño de la base de datos se enfocó en la creación de colecciones estructuradas con relaciones mínimas entre ellas, utilizando claves primarias generadas automáticamente por MongoDB (ObjectId). Cada colección fue diseñada de acuerdo con su propósito, garantizando que los datos fueran accesibles y escalables.

Análisis y Discusión

5.1 Interpretación de Resultados

El análisis muestra que MongoDB es una herramienta poderosa para gestionar bases de datos NoSQL, brindando flexibilidad y facilidad para realizar operaciones CRUD. Las consultas realizadas permitieron una gestión eficiente de los datos en diferentes colecciones. Además, las operaciones avanzadas, como la actualización de salarios o la eliminación de departamentos, fueron realizadas sin contratiempos.

El **Saber** como Arma de Vida

Conclusiones

El trabajo realizado con MongoDB ha demostrado claramente que las bases de datos NoSQL son una alternativa eficiente y flexible a los sistemas de bases de datos relacionales tradicionales, especialmente en escenarios donde los datos pueden ser no estructurados o donde la escalabilidad es una prioridad. A través del uso de MongoDB, pudimos llevar a cabo de manera eficiente las operaciones básicas de CRUD (Crear, Leer, Actualizar, Eliminar), así como realizar consultas más avanzadas para el manejo de datos interrelacionados entre colecciones.

La flexibilidad de MongoDB permitió el almacenamiento de datos sin necesidad de un esquema rígido, lo que facilita la adaptabilidad a medida que cambian las necesidades de los datos. Además, el uso de ObjectId como identificadores únicos ayudó a mantener relaciones simples entre las colecciones, lo que refleja una estructura de datos menos compleja pero altamente funcional.

Sin embargo, es importante señalar que aunque MongoDB ofrece una alta flexibilidad, la planificación y diseño adecuado de la base de datos sigue siendo crucial para mantener un buen rendimiento. El uso adecuado de índices, el diseño de consultas optimizadas y la implementación de prácticas de copias de seguridad y recuperación de datos son aspectos clave para el éxito a largo plazo del manejo de una base de datos NoSQL.

Recomendaciones

- Implementar índices en las colecciones para optimizar las consultas de búsqueda.
- Realizar copias de seguridad periódicas de la base de datos para evitar la pérdida de datos importantes.
- Evaluar la escalabilidad de la base de datos a medida que crece el número de registros.





INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:
Educación

Referencias

MongoDB Documentation. (<https://docs.mongodb.com/>)

GitHub Repositorio: <https://github.com/stivendias28/repositorio-mongo.git>



El **Saber** como **Arma de Vida**

