

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA,
CENTRO UNIVERSITARIO DE JALAPA
FACULTAD DE INGENIERIA EN SISTEMAS
INTRODUCCIÓN A LOS SISTEMAS DE COMPUTO
MARCO TULIO VALDEZ ALBIZUREZ



BASE DE DATOS NO RELACIONALES

NOMBRE: Danny Josue Ignacio Tax
CARNÉ: 0907-24-2011
SECCION: B

INTRODUCCIÓN

En el ámbito de la programación, las **bases de datos no relacionales** desempeñan un papel fundamental. En este trabajo de investigación, exploraremos sobre estas bases de datos, comprendiendo sus ventajas, desventajas y su estructura. Además, analizaremos otros elementos esenciales para su uso eficiente.

Estos son los temas importantes en la investigación:

1. Tipos de Bases de Datos No Relacionales:

- MongoDB.
- RavenDB.
- CouchDB.
- Firebase.

2. Comparación entre Bases de Datos Relacionales y No Relacionales:

- Analizaremos las diferencias clave entre estos dos tipos de bases de datos.

3. Formatos de Datos Utilizados por los Motores Investigados:

- Investigaremos los formatos de datos específicos utilizados por cada motor de base de datos.
- Comprenderemos cómo se almacenan y consultan los datos en estos sistemas.

BASE DE DATOS NO RELACIONALES

Una **base de datos no relacional** es aquella que no usa el esquema tabular de filas y columnas que se encuentra en la mayoría de los sistemas de base de datos más tradicionales. En su lugar, las bases de datos no relacionales usan un modelo de almacenamiento que está optimizado para los requisitos específicos del tipo de datos que se almacena. Por ejemplo, los datos se pueden almacenar como pares clave/valor simple, como documentos JSON o como un grafo que consta de bordes y vértices.

Lo que todos lo que estos almacenes de datos tienen en común están que no utilizan un modelo relacional. Además, tienden a ser más específicos en el tipo de datos que admiten y en cómo se pueden consultar los datos. Por ejemplo, los almacenes de datos de serie temporal están optimizados para consultas en secuencias de datos basadas en el tiempo. Sin embargo, los almacenes de datos de grafos están optimizados para explorar relaciones ponderadas entre entidades. Ninguno de los formatos se generalizaría bien en la tarea de administrar datos transaccionales.

El término **NoSQL** hace referencia a almacenes de datos que no usan SQL para las consultas. En su lugar, los almacenes de datos usan otros lenguajes de programación y otras construcciones para consultar los datos. En la práctica, "NoSQL" significa "base de datos no relacional", aunque muchas de estas bases de datos admiten consultas compatibles con SQL. Sin embargo, la estrategia de ejecución de consultas subyacente normalmente es de la forma en que una RDBMS tradicional ejecutaría la misma consulta SQL.

Estas son las principales ventajas de una base de datos no relacional:

- Son mucho más flexibles a la hora de crear esquemas de información, lo que las convierte en una solución ideal para el almacenamiento y gestión de datos no estructurados o semiestructurados.
- Ofrecen una mayor escalabilidad. Pueden soportar mayores volúmenes de datos y añadir mayor capacidad añadiendo nuevos módulos de software, sin necesidad de añadir nuevos servidores.
- Garantizan un alto rendimiento, ya que están diseñadas para trabajar con modelos de datos concretos y patrones de acceso específicos.
- Son muy funcionales, ya que cuentan con API exclusivas y proporcionan modelos de datos para trabajar con cada tipo de datos presentes en la base.

Por su parte, las principales desventajas de una base de datos no relacional son las siguientes:

- No cumplen igual que las relacionales con las propiedades de atomicidad, consistencia, integridad y durabilidad.
- No son compatibles con determinadas consultas en lenguaje SQL.

- Carecen de un sistema estandarizado ya que todavía son bases de datos relativamente nuevas.
- Muchos sistemas de gestión de bases de datos relacionales son de código abierto y tienen una gran comunidad detrás programando soluciones y nuevas funcionalidades. En el caso de las bases de datos no relacionales este soporte es mucho más limitado.

MONGODB: Es una base de datos de documentos diseñada para facilitar el desarrollo y escalado de aplicaciones y flexibilidad, y un modelo de consultas e indexación avanzado. Almacena datos en documentos flexibles similares a JSON, por lo que los campos pueden variar entre documentos y la estructura de datos puede cambiarse con el tiempo. El modelo de documento se asigna a los objetos en el código de su aplicación para facilitar el trabajo con los datos

Las consultas ad hoc, la indexación y la agregación en tiempo real ofrecen maneras potentes de acceder a los datos y analizarlos. Esta base de datos es distribuida en su núcleo, por lo que la alta disponibilidad, la escalabilidad horizontal y la distribución geográfica están integradas y son fáciles de usar.

RAVENDB: Es una base de datos documental de código abierto con soporte transaccional nativo escrita íntegramente en C# y desarrollada por Hibernating Rhinos Ltd. Es una base de datos multiplataforma compatible con Windows, Linux y MacOS. RavenDB almacena datos como documentos JSON y soporta clústeres distribuidos con replicación maestro-maestro. Los datos se almacenan como documentos sin esquema en formato JSON. A nivel de almacenamiento, los documentos se escriben en un formato binario nativo llamado "blittable". Los documentos se agrupan en colecciones, y cada documento pertenece exactamente a una colección.

Las bases de datos se pueden implementar en un clúster distribuido de servidores (llamados "nodos") mediante replicación multimaestro. Algunas operaciones a nivel de clúster requieren el consenso de la mayoría de los nodos; el consenso se determina mediante una implementación del algoritmo Raft llamado Rachis. Las tareas se distribuyen a los diferentes nodos de forma equilibrada. RavenDB utilizó originalmente el motor de almacenamiento Esent. La versión 3.0 lo reemplazó con un nuevo motor de almacenamiento de código abierto llamado Voron.

Los clientes son compatibles con C#, C++, Java, Node.js, Python, Ruby y Go.

COUCH DB: Es una base de datos de documentos NoSQL de código abierto que recopila y almacena datos en formatos de documentos basados en JSON. A diferencia de las bases de datos relacionales, CouchDB utiliza un modelo de datos sin esquema, que simplifica la gestión de registros en varios dispositivos informáticos, teléfonos móviles y navegadores web.

A diferencia del software privado que puede facilitar la dependencia del proveedor, CouchDB es de código abierto, de uso gratuito y se integra fácilmente en su infraestructura de gestión de datos actual. Debido a que usted tiene más control sobre

el software, también tiene más flexibilidad para adaptarlo a las necesidades únicas de su negocio. Ya sea que requiera un almacén de documentos de propósito general, que permita una sincronización de datos eficiente o que adopte una mentalidad **"Sin conexión primero"**, CouchDB brinda a las empresas la flexibilidad que necesitan para crear infraestructuras duraderas, confiables y escalables.

FIREBASE: Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización.

Esta plataforma proporcionó a los desarrolladores una API que permitía la integración de sistemas de chat en sus páginas web. Estaba siendo utilizado en gran escala por los desarrolladores para pasar paquetes de información de sus aplicaciones, como el estado de las partidas en el caso de los juegos. Fue entonces cuando decidieron separar ambas funcionalidades, el sistema de chat y el sistema de arquitectura en tiempo real que lo propulsaba, dando como resultado la fundación de Firebase en abril de 2012

BASE DE DATOS RELACIONAL	BASE DE DATOS NO RELACIONAL
Basada en datos.	Basada en documentos.
Mantiene una estructura definida previamente en el diseño.	Su estructura puede ser dinámica (para bases orientadas a documentos.
Se diseñan para tener la menor redundancia de datos posibles.	Permite tener redundancia de datos.
Para aplicaciones que requieren datos consistentes sin dar posibilidades al error (Ej. Datos financieros.	Para aplicaciones que requieren grandes cantidades de datos a consultar.
Escalables verticalmente: Aumentar las prestaciones del servidor.	Escalable horizontalmente: Agregar más servidores.

MongoDB usa un formato de almacenamiento de documentos llamado BSON, que es una forma binaria de JSON (JavaScript Object Notation o notación de objetos de JavaScript) que puede acomodar más tipos de datos.

RavenDB se almacena como documentos JSON y puede ser obtenida y modificada mediante colas Linq o utilizando herramientas API

CouchDB usa varios formatos y protocolos para almacenar, transmitir y procesar datos. El formato principal para el almacenamiento de datos es JSON. También, usa JavaScript como lenguaje de consulta y MapReduce para el procesamiento. Además, utiliza el protocolo HTTP para las comunicaciones.

Firebase nos proporciona cuatro tipos de soluciones para almacenar nuestros datos en la nube. ¿Cuáles son las funcionalidades que nos brinda cada uno en función de las necesidades de nuestro proyecto?

- **Cloud Storage** permite almacenar y descargar archivos como imágenes o vídeos.
- **Cloud Firestore** es una base de datos flexible y escalable.
- **Firebase Realtime Database** almacena datos de aplicación JSON.
- **Firebase Remote Config** almacena pares clave-valor.

CONCLUSION

En resumen, este trabajo de investigación nos ha permitido explorar las bases de datos no relacionales y comprender su relevancia en el ámbito de la programación. Hemos analizado ventajas y desventajas, así como los formatos de datos utilizados por motores específicos como MongoDB, RavenDB, CouchDB y Firebase. Además, su capacidad para adaptarse a diversos tipos de datos, su flexibilidad y su enfoque en la escalabilidad las convierten en la elección ideal para aplicaciones modernas que manejan grandes cantidades de información no estructurada y la comparación en la base de datos relacional y la base de datos no relacional.

Esto nos hace entender de mejor manera como funcionan las bases de datos no relacionales y sus funciones y así poderlos aplicar de manera correcta en el ámbito de la programación.

E-GRAFIAS

- <https://learn.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data>
- <https://www.mongodb.com/es/company/what-is-mongodb>
- <https://es.wikipedia.org/wiki/RavenDB>
- <https://www.ibm.com/mx-es/topics/couchdb>
- <https://www.purestorage.com/es/knowledge/what-is-mongodb.html>
- <https://aprenderbigdata.com/couchdb>
- <https://www.paradigmadigital.com/dev/que-es-firebase-cloud-storage/>