**Universidad Mariano Gálvez de Guatemala**

**Facultad de Ingeniería**

**Centro Universitario Jalapa**



**Carrera**

Ingeniería en Sistemas de Información y Ciencias de la Computación

**Curso**

Introducción a los Sistemas de Cómputo

**Catedrático**

Ing. Marco Tulio Valdez

**Dulce Marielena Monterroso Reyes**

**0907-24-21707**

**“Trabajo de Investigación”**

**Jalapa, 13 de abril de 2024**

**Introducción**

Con los avances de la tecnología han aparecido aplicaciones cada vez más grandes que necesitan mucha capacidad, flexibilidad, rapidez, y la respuesta a esas necesidades son las bases de datos no relacionales y en este documento se hablará del porqué de esto y se brinda información valiosa acerca de estas bases de datos, sus características, ventajas, desventajas, su diferencia con las bases de datos relacionales y algunos motores de estas bases de datos.

**Bases de datos no relacionales**

**Definición**

Las bases de datos no relacionales son un sistema de almacenamiento de información que se caracteriza por no usar el lenguaje SQL para las consultas. Esto no significa que no puedan usar el lenguaje SQL, pero no lo hacen como herramienta de consulta, sino como apoyo. Por ello también se les suele llamar NoSQL o «no solo SQL».

Otra de sus principales características es que no trabajan con estructuras definidas. Es decir, los datos no se almacenan en tablas, y la información tampoco se organiza en registros o campos.

Tienen una gran escalabilidad y están pensadas para la gestión de grandes volúmenes de datos. Por otro lado, a diferencia de las bases de datos relacionales no cumple con el estándar ACID de atomicidad, consistencia, aislamiento y durabilidad.

Las bases de datos no relacionales son más actuales que las relacionales, y su desarrollo se ha basado en la necesidad de crear sistemas de gestión capaces de trabajar con datos no estructurados o semi-estructurados.

**Características y funciones a tener en cuenta**

Las principales características de una base de datos no relacional son las siguientes:

La información no se almacena en tablas sino a través de documentos.

Son bases de datos muy útiles para organizar y gestionar información no estructurada, o cuando no se tiene una noción clara de los datos a almacenar.

Son bases de datos con alto grado de escalabilidad y están diseñadas para soportar grandes volúmenes de datos.

No utilizan el lenguaje SQL para consultas, aunque sí lo pueden usar como herramienta de apoyo.

Es un sistema de almacenamiento de datos relativamente nuevo, y como tal, todavía no posee un sistema estandarizado.

A diferencia de las no relacionales, no garantizan el cumplimiento de las cualidades ACID, esto es, atomicidad, consistencia, integridad y durabilidad.

**Ventajas y desventajas**

Una base de datos relacional tiene ciertas particularidades que las distinguen del resto, lo cual se traduce en una serie de ventajas y desventajas.

**Ventajas**

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.

**Desventajas**

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.

**Tipos**

Existen diferentes tipos de bases de datos no relacionales, en función del método que emplean para almacenar la información. Veamos algunos de ellos.

**Clave-valor**

Se trata de bases de datos no relacionales que almacenan la información en base a pares de clave valor. Es decir, cada clave sirve como un identificador único, y a cada una de ellas se le aplica un valor. Son especialmente usadas a la hora de almacenar datos de juegos, aplicaciones o aparatos que funcionan mediante el internet de las cosas (IoT).

**Documentos**

En una base de datos relacional basada en documentos la información se representa como objetos o documentos JSON. Su principal ventaja es que los documentos son de naturaleza flexible, semiestructurada y jerárquica, lo que facilita a los desarrolladores las tareas de almacenamiento, gestión y consulta de datos. Es un modelo usado habitualmente en sistemas de administración de contenidos o para gestionar perfiles de usuarios.

**Gráficos**

Las bases de datos no relacionales basadas en gráficos están pensadas para crear relaciones y navegar por ellas. Las entidades de datos se almacenan mediante nodos y los bordes son los que crear las relaciones entre entidades. Con frecuencia las bases de datos gráficas se emplean en redes sociales, sistemas de detección o prevención de fraudes o sistemas de recomendaciones.

**En memoria**

Son bases de datos no relacionales diseñadas para ofrecer respuestas en milisegundos y soportar grandes picos de tráfico. Un ejemplo de bases de datos en memoria son las empleadas en tablas de clasificaciones de juegos o en herramientas para hacer análisis en tiempo real.

**¿Cómo funcionan y para qué sirven?**

**¿Para qué sirve una base de datos no relacional?** Principalmente, las bases de datos no relacionales se emplean para almacenar datos no estructurados, o de los que no se tienen ningún esquema o información predefinida. Sin embargo, también pueden ser empleados para el almacenamiento y gestión de datos estructurados. Veamos un ejemplo de cómo funcionaría el almacenamiento de datos en una base de datos no relacional y en una base de datos relacional.

Tomemos como ejemplo una base de datos en la que se van a almacenar DVDs de películas. Estos podrían almacenarse según el título de la película, fecha de estreno, género, director, etc.

En una base de datos relacional, cada película sería un registro y ocuparía una de las filas de la tabla. Por su parte, los diferentes datos almacenados sobre cada película (título, fecha de estreno, género o director) se organizarían en campos, que corresponderían a las columnas de la tabla.

Sin embargo, en una base de datos NoSQL o no relacional cada registro de una película se almacena como un único documento ISON. Todos los atributos o características de cada película (título, fecha de estreno, género y director) se almacenan en un solo documento, Es decir, este almacenamiento de datos está preparado para ofrecer una mayor escalabilidad horizontal y un desarrollo más intuitivo.

Por tanto, el funcionamiento de una base de datos no relacional deja de lado el uso de tablas, de forma que cada elemento se guarda en documentos que ya incluyen todos sus atributos.

**Estructura**

Como ya hemos señalado a lo largo del artículo, una base de datos no relacional no requiere de tablas para el almacenamiento de la información.

Normalmente la estructura de una base de datos no relacional se basa la organización de la información a través de documentos. Este tipo de databases están pensadas para ofrecer mayor escalabilidad horizontal y no tienen identificadores que permitan establecer relaciones entre diferentes conjuntos de datos.

Cada uno de los documentos almacenados en la base de datos incluye todos los atributos del elemento, por lo que resultan muy útiles a la hora de guardar información poco estructurada o de la que no se tiene un esquema claro de inicio.

**Ejemplo**

Imagina que se han enviado varios robots a la luna con el objetivo de encontrar nuevos materiales o para elaborar informes de sus descubrimientos durante su misión espacial. Evidentemente, no se tiene ninguna certeza sobre lo que se puedan encontrar. Cada uno de estos robots tiene incorporados unos sensores para reconocer aquello que se van encontrando, y envían informes cada 24 horas indicando qué han encontrado, además de la fecha o localización del descubrimiento.

Entonces, uno de estos robots podría almacenar la información en una base de datos, la cual se podría organizar de la siguiente manera,

{  
«robot\_id»:1, (la máquina o robot que ha hecho el descubrimiento)  
«timestamp»:149992693000, (la fecha y hora en la que ocurrió determinado evento)  
«coordenadas»:»75988823.567, 55375867.098, 12676444.311″, (los lugares donde se realizaron las búsquedas)  
«encontrado»:[  
«mineral»,  
«mineral»,  
«agua»,  
], (los elementos u objetos encontrados por el robot)  
«temperatura»:{  
«min»:-50,  
«max»:-49  
}, (temperatura recogida por los sensores)  
«ruido»:{  
«min»:72,  
«max»:4549  
} (ruido recogido por los sensores)  
}

Para terminar, cabe decir que las bases de datos no estructurales son cada vez más utilizadas ya que permiten almacenar datos de los cuáles no se tiene una estructura clara y que serían casi imposibles de organizar en una base relacional.

**MongoDB**

MongoDB (enlace externo a IBM) es un sistema de gestión de bases de datos (DBMS, por sus siglas en inglés) no relacionales y de código abierto, que utiliza documentos flexibles en lugar de tablas y filas para procesar y almacenar varias formas de datos. Al ser una solución de base de datos NoSQL, MongoDB no requiere un sistema de gestión de bases de datos relacionales (RDBMS), por lo que proporciona un modelo de almacenamiento de datos elástico que permite a los usuarios almacenar y consultar fácilmente tipos de datos multivariados. Esto no solo simplifica la gestión de la base de datos para los desarrolladores, sino que también crea un entorno altamente escalable para aplicaciones y servicios multiplaLos documentos o colecciones de documentos de MongoDB son las unidades básicas de datos. Estos documentos, con formato JSON (Java Script Object Notation) binario, pueden almacenar varios tipos de datos y distribuirse en varios sistemas. El diseño de esquema dinámico de MongoDB brinda a los usuarios una flexibilidad sin igual para crear registros de datos, consultar colecciones de documentos a través de la agregación de MongoDB y analizar grandes cantidades de información.taforma.

**Casos de uso de MongoDB**

**Aplicaciones móviles**

El modelo de documento JSON de MongoDB le permite almacenar datos de aplicaciones de fondo donde los necesite, incluso en dispositivos Apple iOS y Android, así como soluciones de almacenamiento basado en cloud .Esta flexibilidad le permite agregar datos en múltiples entornos con indexación geoespacial y secundaria, brindando a los desarrolladores la capacidad de escalar fácilmente sus aplicaciones móviles.

**Analítica en tiempo real**

A medida que las empresas escalan sus operaciones, es fundamental obtener acceso a métricas clave e información de la empresa extraída de grandes conjuntos de datos. MongoDB maneja la conversión de documentos JSON y similares a JSON, como BSON, en objetos Java con facilidad, lo que hace que la lectura y grabación de datos en MongoDB sea rápida e increíblemente eficiente cuando se analiza información en tiempo real en varios entornos de desarrollo. Esto ha demostrado ser beneficioso para varios sectores empresariales, incluidos el Gobierno, los servicios financieros y el comercio minorista.

**Sistemas de gestión de contenido**

Los sistemas de gestión de contenido (CMS, por sus siglas en inglés) son herramientas potentes que juegan un papel importante para garantizar una buena experiencia del usuario al acceder a sitios de comercio electrónico, publicaciones en línea, plataformas de gestión de documentos y otras aplicaciones y servicios. MongoDB le permite agregar fácilmente nuevas funciones y atributos a sus aplicaciones en línea y sitios web mediante una única base de datos y con alta disponibilidad.

**Almacén de datos empresariales**

El marco de trabajo de Apache Hadoop es una colección de módulos de código abierto, incluidos Hadoop Distributed File System y Hadoop MapReduce, que funcionan con MongoDB para almacenar, procesar y analizar grandes cantidades de datos. Las organizaciones pueden utilizar MongoDB y Hadoop para realizar modelado de riesgos, análisis predictivos y procesamiento de datos en tiempo real.

**Ventajas de MongoDB**

A lo largo de los años, MongoDB se ha afianzado como solución para muchas empresas que buscan una base de datos NoSQL potente y altamente escalable. Pero MongoDB es mucho más que una base de datos tradicional basada en documentos y cuenta con unas prestaciones excelentes que la hacen destacar entre otras DBMS.

**Equilibrio de carga**

El auge de las aplicaciones en cloud y el incremento de la demanda de recursos para las empresas complican la disponibilidad y la fiabilidad de los servicios. El proceso de uso compartido del equilibrio de carga de MongoDB distribuye grandes conjuntos de datos a través de varias máquinas virtuales a la vez manteniendo rendimientos aceptables de lectura y grabación. Esta escalada horizontal se llama sharding (fragmentación) y ayuda a las organizaciones a evitar el coste de la escalada vertical de hardware al mismo tiempo que expande la capacidad de los despliegues basados en cloud.

**Consultas de base de datos ad hoc**

Una de las mayores ventajas de MongoDB sobre otras bases de datos es su capacidad para manejar consultas ad hoc que no requieren esquemas predefinidos. Las bases de datos MongoDB utilizan un lenguaje de consulta similar a las bases de datos SQL que es extremadamente accesible, tanto para desarrolladores principiantes como avanzados. Esta accesibilidad facilita el envío, la consulta, la clasificación, la actualización y la exportación de sus datos con métodos de ayuda comunes y mandatos shell simples.

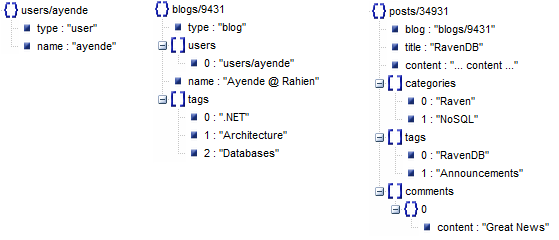
**Soporte a múltiples lenguajes**

Uno de los mejores aspectos de MongoDB es su soporte a múltiples lenguajes. Se han publicado varias versiones de MongoDB y están en continuo desarrollo con soporte de controlador para los lenguajes de programación más populares, incluidos Python, PHP, Ruby, Node.js, C++, Scala, JavaScript y muchos más.

**Raven DB**

Raven es un proyecto de código abierto, pero con una opción comercial para base de datos para plataformas .NET/Windows. Raven ofrece un diseño de modelo de datos flexible para adaptarse a las necesidades de los sistemas del mundo real. Este sistema permite utilizar las consultas Linq que se basan en la baja latencia y el obtienen un alto rendimiento.  
  
**Entre las características de este sistema podemos destacar:**

* Configuración sencilla en Windows. Es fácil de instalar y se ejecuta en ventanas como un servicio normal o un sitio web de IIS7.
* Definir fácilmente los índices con consultas LINQ.
* API .NET. Raven viene con una api que implementa la unidad de trabajo.
* Infraestructura escalable.

Raven está publicado bajo licencia de código abierto y una licencia comercial, lo que significa que está libremente disponible, pero si se quiere utilizar con un software propietario, habría que adquirir una licencia comercial.  
  
  
Los documentos en Raven son esquemas JSON. Los documentos son entidades independientes no relacionadas entre sí. La siguiente imagen es un ejemplo de documentos.  
  
  


Tiene algunas características interesantes que la hacen adecuadas para aplicaciones de alto rendimiento y baja latencia:

* Transaccional: esta característica no la comparte con MongoDB
* Escalable: soportando sharding, replicación y multitenancy out-of-the-box
* Sin esquemas
* Rápida
* Alto rendimiento
* Caché incorporado
* Búsqueda Full-text incorporada
* APIs en varios lenguajes
* API HTTP
* Soporte geospacial
* UI Web para la gestión de la base de datos
* [Extensible](http://ravendb.net/docs/server/extending/bundles)
* Embebible
* Backups sencillos
* Multitenancy
* Versionado
* Expiración

**CouchDB**

CouchDB es una base de datos NoSQL capaz de replicarse en una amplia gama de entornos cliente y servidor. Por sus respectivas características, el Cloud Computing es el ambiente de funcionamiento más natural de CouchDB, otra de las aplicaciones Open Source que podemos instalar y desplegar sobre los [Servidores Cloud](https://www.arsys.es/servidores/cloud)de Arsys en sólo unos minutos.

**¿Qué es CouchDB?**

CouchDB es un sistema gestor de bases de datos no relacional, basado completamente en tecnologías sobre las que está implementada la web, como HTTP, JSON o Javascript. Es una base de datos distribuida, capaz de adaptarse tanto a ambientes de servidor como a clientes de diversos tipos. Se trata de un proyecto Open Source de la Fundación Apache pensado especialmente para usarse como infraestructura en aplicaciones web. Su principal característica es la definición del Couch Replication Protocol, un sistema que permite que una variedad de productos pueda comunicarse entre sí, compartiendo datos y realizando una sincronización de los mismos de manera automática.

Existe una gama de tecnologías que usan el protocolo de replicación de CouchDB, desde servidores en cluster hasta teléfonos móviles o navegadores de todo tipo. Este protocolo permite, entre otras cosas, que los datos de CouchDB puedan alojarse en varios sistemas, replicados de forma distribuida. En la práctica, los sistemas podrían gestionar los datos del sistema remoto y tenerlos disponibles incluso cuando están offline. CouchDB se encarga de gestionar las inconsistencias, realizando la sincronización de la información cuando se disponga de una conexión a Internet.

Para ello, puede apoyarse en las funcionalidades de diversos sistemas emergentes, como son PouchDB o Couchbase Lite. PouchDB es un sistema gestor de bases de datos que funciona sobre navegadores de ordenadores y móviles, capaz de sincronizar automáticamente con el nodo de CouchDB remoto. Por su parte, Couchbase Lite es básicamente lo mismo, pero para lenguajes de desarrollo nativo de apps.

**Características de CouchDB**

**Orientada a documentos.**

Igual que otros sistemas gestores de bases de datos NoSQL, CouchDB está orientado a documentos, que son expresados mediante el lenguaje JSON.

**Soporte a ACID.**

CouchDB implementa la semántica ACID, lo que permite que diversos sistemas puedan escribir y leer en múltiples instancias de la base de datos al mismo tiempo.

**Usa REST sobre HTTP.**

Al estar basado en web, usa los mismos verbos del HTTP para las operaciones CRUD, del mismo modo que las API REST. Por ello, es capaz de responder a acciones simplemente con cualquier cliente capaz de utilizar el protocolo HTTP, como navegadores o lenguajes de todo tipo.

**Offline-first**

Gracias a su sistema de replicación, los sistemas que mantienen datos de CouchDB pueden trabajar offline, de manera autónoma, enviando la información a la base de datos remota cuando disponen de una conexión a Internet. Esto es muy interesante para el desarrollo de apps para dispositivos o [Progressive Web Apps](https://www.arsys.es/blog/progressive-web-apps" \t "_blank), ya que permite seguir trabajando con las aplicaciones en situaciones de nula o escasa conectividad.

**Cloud y CouchDB**

CouchDB puede usarse como sistema gestor de bases de datos en un único nodo, permitiendo las posibilidades de otras bases datos existentes. Sin embargo, su entorno más apropiado es el trabajo en un cluster, lo que permite ejecutar una única base de datos en diversos servidores. Gracias a las características de los Servidores Cloud y a su facilidad para el despliegue de clusters, son la plataforma perfecta para esta base de datos NoSQL, ya que cuenta con total flexibilidad y facilidad de gestión, a la par que una mayor capacidad, mejor disponibilidad y máxima tolerancia a fallos.

Además, las bases de datos gestionadas con CouchDB están pensadas para operar en la web, comenzando con la propia interfaz de manipulación de la información, basada en HTTP. De ahí, que nos interese su despliegue en máquinas que ofrezcan sus servicios en la Nube y que permitan a diversos tipos de clientes la conexión con los datos remotos para su sincronización.

**Firebase**

**¿Qué es Firebase?**

[Firebase de Google](https://firebase.google.com/?hl=es) es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.

Aunque fue creada en 2011 [pasó a ser parte de Google en 2014](https://www.muycomputerpro.com/2014/10/22/google-firebase), comenzando como una base de datos en tiempo real. Sin embargo, se añadieron más y más funciones que, en parte, permitieron agrupar los SDK de productos de Google con distintos fines, facilitando su uso.

**¿Para qué sirve Firebase?**

Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida.

Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de las mismas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis. Es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al backend, tanto en cuestiones de desarrollo como de mantenimiento.

Tal vez estés buscando en internet: tutorial Firebase, manual firebase o dev firebase, pero la mejor forma de entender en qué consiste Firebase es hacernos una idea de las herramientas que ofrece y a qué están destinadas, algo que veremos a continuación.

**Funciones de Firebase**

Firebase dispone de diferentes funcionalidades, que se pueden dividir básicamente en 3 grupos: Desarrollo (Develop), Crecimiento (Grow) y Monetización (Earn), a los que hay que sumar la Analítica (Analytics).

**Desarrollo**

El primer grupo de funciones es conocido como Desarrollo o Develop en Firebase. Como su nombre indica, incluye los servicios necesarios para el desarrollo de un proyecto de aplicación móvil o web. Estos contribuyen a que el proceso sea más rápido, puesto que se dejan determinadas actividades a mano de Firebase, mientras que otras permiten optimizar diversos aspectos para conseguir la calidad deseada.

**Crecimiento**

El segundo bloque está enfocado al proceso de crecimiento de la aplicación, que contempla tanto la gestión de aquellos que ya son usuarios de la misma, como herramientas para la captación de nuevas audiencias.

**Monetización**

La monetización en Firebase es la tercera pata contemplada. En este caso, la búsqueda de ganancias viene ligada a la publicidad que se puede insertar en las aplicaciones, consiguiendo que los usuarios de las mismas reciban anuncios relevantes en función de la segmentación que se le haya dado a la campaña.

Para integrar estos anuncios en la app, Firebase cuenta con AdMob, muy interesante para rentabilizar la aplicación.

**Ventajas y desventajas**

Como se puede concluir a partir de las funcionalidades ofrecidas, esta herramienta presenta numerosos beneficios para los desarrolladores que lo utilicen. Aunque son muchos más, algunos de ellos se recopilan brevemente. Te explicamos a continuación cuáles son los beneficios de Firebase:

Muy recomendable para aplicaciones que necesiten compartir datos en tiempo real.

Sus funcionalidades, además de ser variadas, se complementan muy bien y se pueden gestionar de forma sencilla desde un único panel. Además, no es necesario usar todas estas opciones para la aplicación, pudiendo elegir solo aquellas que más nos interesen.

Facilita el envío de notificaciones: son muy sencillas de implementar y gestionar, además de ser extremadamente útiles para mantener la atención de los usuarios.

Permite la monetización: desde el propio Firebase se puede agregar publicidad a la app, permitiendo fácilmente renatbilizarla obteniendo el ROI.

Engloba Analytics: especializado en determinadas métricas de aplicaciones móviles e integrado en el panel central de Firebase con un funcionamiento muy intuitivo. Esencial para tomar decisiones en distintas fases del proceso.

Google ofrece numerosos documentos y tutoriales a modo introductorio e informativo (con gran profundidad) para que sumergirse en Firebase sea mucho más fácil.

Soporte gratuito vía email, sin importar si el desarrollador utiliza la versión gratuita o de pago.

Escalabilidad: los inicios son gratuitos, pero permite ir adaptándose a las necesidades de la aplicación con diferentes planes de pago.

Ofrece seguridad al usuario: con los certificados SSL.

Permite a los desarrolladores restarle atención al backend y a las infraestructuras complejas para centrarse completamente en otros aspectos.

La desventaja más comúnmente mencionada es el precio. Se ha hablado de la escalabilidad de Firebase, donde el inicio con el plan Spark es gratuito. Sin embargo, tiene limitaciones (principalmente de número de usuarios simultáneos y de espacio de almacenamiento), por lo que puede resultar necesario contratar una versión de pago. Los planes de pago son Flame (25 dólares al mes) y Blaze (se paga según el consumo).

Así que como puedes ver, puedes encontrar Firebase gratis, pero si necesitas trabajar con ello de manera profesional, verás que tienes que buscar cuál es el precio de Firebase. Depende de ti invertir más o menos en un proyecto.

CUADRO COMPRARATIVO

|  |  |
| --- | --- |
| **BASES DE DATOS RELACIONALES VS BASES DE DATOS NO RELACIONALES** | |
| En las bases de datos relacionales la información se organiza de forma estructurada en tablas; en las no relacionales no es así. | Normalmente la estructura de una base de datos no relacional se basa la organización de la información a través de documentos. |
| Tienen un identificador único. | Una base de datos no relacional no tiene un identificador que sirva para relacionar un conjunto de datos con otros. |
| La interfaz estándar de programa de usuario y aplicación a una base de datos relacional, es el Lenguaje de Consultas Estructuradas (SQL) | Una base de datos no relacional no usa el lenguaje SQL como lenguaje principal para sus consultas. |
| Almacena información que organiza datos en relaciones predefinidas. | Las bases de datos no relacionales se emplean sobre todo para almacenar datos no estructurados o semiestructurados. |
| Cumple con las propiedades de ACID. | Una base de datos relacional no cumple con las propiedades ACID con la misma eficacia que una base de datos relacional. |
| Es recomendable cuando el volumen de los datos no crece o lo hace poco a poco. | La escalabilidad es mayor en una base de datos no relacional, y también están preparadas para soportar mayor volumen de datos. |
| Es compleja, necesita de más recursos y procesamientos. | Las bases de datos no relacionales o NoSQL también ofrecen una mayor flexibilidad y escalabilidad horizontal. |
| SQL es el lenguaje estándar de las bases de datos relacionales. | A diferencia de las relacionales, las bases de datos no relacionales todavía no disponen de un lenguaje estandarizado (SQL). |

**Formatos de datos utilizado por los motores investigados**

**MongoDB**

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**

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

**CouchDB**

En CouchDB, los documentos son las unidades principales de datos que se utilizan en JSON, compuestas por varios campos y archivos adjuntos para un fácil almacenamiento.

**Firebase**

Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado.

**Conclusión**

Las bases de datos no relacionales no trabajan con estructuras definidas, es decir, los datos no se almacenan en tablas y la información tampoco se organiza en registros o campos, tienen una gran escalabilidad y están pensadas para la gestión de grandes volúmenes de datos, son más flexibles y útiles, recomendadas para el uso de datos con requisitos cambiantes, no requiere uniones complejas, lo que significa que las consultas de bases de datos muestran resultados más rápido, están orientadas a los documentos y le permite almacenar y recuperar dato en formatos que no sean tablas. Son adecuadas para aplicaciones web, redes sociales, análisis de datos en tiempo real y sistemas que requieren una rápida alteración y evolución del esquema.

**Bibliografías**

<https://ayudaleyprotecciondatos.es/bases-de-datos/no-relacional/>

<https://www.ibm.com/es-es/topics/mongodb>

<https://programacion.net/noticia/raven-db_1703>

<https://www.arsys.es/blog/cloud-couchdb-bbdd#:~:text=CouchDB%20es%20un%20sistema%20gestor,a%20clientes%20de%20diversos%20tipos>.

<https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>

<https://www.paradigmadigital.com/dev/crear-base-datos-firebase/>

<https://www.ibm.com/mx-es/topics/couchdb#:~:text=En%20CouchDB%2C%20los%20documentos%20son,adjuntos%20para%20un%20f%C3%A1cil%20almacenamiento>.