

## «Comparación de sistemas y modelos gestión de bases de datos NoSQL» por Mark Drake (versión previa O. S. Tezer)

- This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#).
- **English language:** This article is a translation from English into Spanish, published under license «[Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](#) », written by Mark Drake (previous version by O. S. Tezer [@Ostezer](#)), published on line by the company for leasing virtual machines [DigitalOcean](#). The title is «[A Comparison of NoSQL Database Management Systems and Models](#)» and [we created a copy at Wayback Machine for prevent in future a broken link](#). This work is licensed under the mentioned license but, of course, in castilian language (AKA *spanish*): «[Atribución-NoComercial-CompartirIgual 4.0 Internacional \(CC BY-NC-SA 4.0\)](#) ».
-

- Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](#).
- **En castellano:** Este artículo es una traducción del inglés al castellano, publicado bajo licencia (en idioma inglés) «[Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](#) » escrito por written by Mark Drake (previous version by O. S. Tezer [@Ostezer](#)), publicado en línea por la empresa de alojamiento de máquinas virtuales [DigitalOcean](#). El título original en idioma inglés es «[A Comparison of NoSQL Database Management Systems and Models](#)» y [hemos creado una copia en Wayback Machine](#) para prevenir un posible enlace roto a futuro.

## Introducción

Cuando la mayoría de las personas piensa en una base de datos, a menudo imaginan el modelo tradicional de base de datos relacional que involucra tablas compuestas de filas y columnas. Si bien los sistemas de administración de bases de datos relacionales aún manejan la mayor parte de los datos en Internet, los modelos de datos alternativos se han vuelto más comunes en los últimos años a medida que los desarrolladores han buscado soluciones a las limitaciones del modelo relacional. Estos modelos de bases de datos no relacionales, cada uno con sus propias ventajas, desventajas y singularidades en su utilización, han sido categorizados como *bases de datos NoSQL*.

Este artículo le presentará algunos de los modelos de bases de datos NoSQL más utilizados. Sopesará algunas de sus fortalezas y desventajas, y proporcionará algunos ejemplos de sistemas de gestión de bases de datos y los potenciales usos para cada uno de los casos.

## Bases de datos relacionales y sus limitaciones

Las bases de datos son racimos de información modelados lógicamente o, simplemente, *datos*. Un *sistema de gestión de bases de datos* (SGBD o, en idioma inglés DBMS), por su parte, es un programa informático que interactúa con una base de datos. Un SGBD le permite controlar el acceso a una base de datos, escribir datos, ejecutar consultas y realizar cualquier otra tarea relacionada con la administración de la base de datos. Aunque los sistemas de gestión de bases de datos a menudo se denominan simplemente "bases de datos", los dos términos no son exactamente intercambiables. Una base de datos puede ser cualquier colección de datos, una o más guardadas en una computadora, mientras que un SGBD es el software específico que le permite interactuar con una base de datos.

Todos los sistemas de gestión de bases de datos tienen un modelo subyacente que estructura cómo se almacenan y acceden los datos. Un *sistema de gestión de bases de datos relacionales* (SGBDR) es un SGBD que emplea el modelo de datos relacionales. En este modelo, los datos se organizan en tablas, las cuales formalmente se denominan relaciones en el contexto de los SGBDR. Los sistemas de gestión de bases de datos relacionales suelen emplear el [Lenguaje de Consulta Estructurado](#) o LCE (en idioma inglés, *SQL* el cual emplearemos frecuentemente) para gestionar y acceder a los datos contenidos en la base de datos.

El modelo relacional, históricamente, ha sido el enfoque más utilizado para administrar datos, y hasta el día de hoy [muchos de los sistemas de administración de bases de datos más populares implementan el modelo relacional](#). Sin embargo, el modelo relacional presenta varias limitaciones que pueden ser problemáticas en ciertos casos.

Por ejemplo, puede ser difícil potenciar una base de datos relacional horizontalmente. La *potenciación horizontal* o potenciado horizontal (frecuentemente traducido como "escalado horizontal" N. del T.), es la práctica de agregar más máquinas a una batería de servidores existente para distribuir la carga y permitir más tráfico y un procesamiento más rápido. Esto a menudo se contrasta con el potenciado vertical que implica la actualización del hardware de un servidor existente, generalmente agregando más RAM o CPU.

La razón por la que es difícil potenciar una base de datos relacional horizontalmente tiene que ver con el hecho de que el modelo relacional está diseñado para garantizar la *coherencia*, lo que significa que los clientes que consulten la misma base de datos siempre verán los últimos datos, con los mismos resultados. Si tuviera que potenciar una base de datos relacional de manera horizontal en varias máquinas, sería difícil garantizar la coherencia, ya que los clientes pueden escribir datos en un nodo y no en los demás, y es probable que haya un retraso entre la escritura inicial y el momento en que los otros nodos actualizado para reflejar los cambios.

Otra limitación presentada por los SGBDR es que el modelo relacional fue diseñado para

administrar datos estructurados, o datos que se alinean con un tipo de datos predefinido o que al menos están organizados de alguna manera predeterminada, lo que la hace fácilmente clasificable y por ende facilitando su búsqueda. Sin embargo, con la expansión de la informática personal y el auge de Internet a principios de la década de 1990, los datos no estructurados (mensajes de correo electrónico, fotos, videos, etcétera) se hicieron más comunes.

A medida que estas limitaciones se hicieron más restrictivas, los desarrolladores comenzaron a buscar alternativas al modelo tradicional de datos relacionales, lo que llevó al crecimiento en popularidad de las bases de datos NoSQL.

## Acerca de NoSQL

La etiqueta NoSQL tiene una definición bastante difusa. "NoSQL" fue acuñado en 1998 por Carlo Strozzi como el nombre de su [nueva base de datos NoSQL](#), elegida simplemente porque no usa SQL para administrar datos.

El término adquirió un nuevo significado después de 2009 cuando Johan Oskarsson organizó una reunión para desarrolladores para discutir la difusión de "bases de datos de código abierto, distribuidas y no relacionales" como [Cassandra](#) y [Voldemort](#). Oskarsson nombró la reunión "NOSQL" y desde entonces el término se ha utilizado como un término general para cualquier base de datos que no emplea el modelo relacional. Curiosamente, la base de datos NoSQL de Strozzi de hecho emplea el modelo relacional, lo que significa que la base de datos NoSQL original no se ajusta a la definición contemporánea de NoSQL.

Debido a que "NoSQL" generalmente se refiere a cualquier SGBD que no emplea el modelo relacional, existen varios modelos de datos operativos asociados con el concepto NoSQL. La siguiente tabla incluye varios de estos modelos de datos, pero tenga en cuenta que esta no es una lista completa:

Modelo operacional BD	Ejemplo
Orientado a clave-valor	Redis, MemcacheDB
Orientado a columnas	Cassandra, Apache HBase
Orientada a documentos	MongoDB, Couchbase
Orientado a <a href="#">grafos</a>	OrientDB, Neo4j

A pesar de estos diferentes modelos de datos subyacentes, la mayoría de las bases de datos NoSQL comparten varias características. Por un lado, las bases de datos NoSQL generalmente están diseñadas para maximizar la disponibilidad a expensas de la coherencia. En este sentido, la

coherencia se refiere a la idea de que cualquier operación de lectura devolverá los datos más recientes escritos en la base de datos. En una base de datos distribuida diseñada para una fuerte consistencia, cualquier dato escrito en un nodo estará inmediatamente disponible en todos los demás nodos; de lo contrario, se producirá un error.

Por el contrario, las bases de datos NoSQL a menudo apuntan a una consistencia eventual. Esto significa que los datos recién escritos están disponibles en otros nodos en la base de datos eventualmente (generalmente en cuestión de unos pocos milisegundos), aunque no necesariamente de manera inmediata. Esto tiene el beneficio de mejorar la disponibilidad de los datos: aunque no vea los últimos datos escritos, puede ver una versión anterior en lugar de recibir un error.

Las bases de datos relacionales están diseñadas para manejar datos normalizados que se ajustan perfectamente a un esquema predefinido. En el contexto de un DBMS, los *datos normalizados* son datos que se han organizado de manera de eliminar redundancias, lo que significa que la base de datos ocupa el menor espacio de almacenamiento posible, mientras que un *esquema* es un resumen de cómo se estructuran los datos en la base de datos.

Si bien las bases de datos NoSQL están equipadas para manejar datos normalizados y pueden clasificar datos dentro de un esquema predefinido, sus respectivos modelos de datos generalmente permiten una flexibilidad mucho mayor que la estructura rígida impuesta por las bases de datos relacionales. Debido a esto, las bases de datos NoSQL tienen la reputación de ser una mejor opción para almacenar datos semiestructurados y no estructurados. Sin embargo, teniendo esto en cuenta, dado que las bases de datos NoSQL no vienen con un esquema predefinido que a menudo significa que depende del administrador de la base de datos definir cómo se deben organizar y acceder a los datos de la manera que tenga más sentido para su aplicación.

Ahora que usted ha entrado en contexto sobre qué son las bases de datos NoSQL y qué las hace diferentes de las bases de datos relacionales, echemos un vistazo más de cerca a algunos de los modelos de bases de datos NoSQL más ampliamente implementados.

## Orientado a clave-valor

Las *bases de datos de clave-valor*, también conocidas como *almacenes de valores clave*, funcionan almacenando y gestionando matrices asociativas. Una matriz asociativa, también conocida como diccionario o tabla *hash*, consiste en una colección de pares clave-valor en los que una clave sirve como un identificador único para recuperar un valor asociado. Los valores pueden ser desde objetos simples, como enteros o cadenas, hasta objetos más complejos, como estructuras JSON.

A diferencia de las bases de datos relacionales, que definen una estructura de datos compuesta por tablas de filas y columnas con tipos de datos predefinidos, las bases de datos de valores clave almacenan datos como una sola colección sin ninguna estructura o relación. Después de conectarse al servidor de la base de datos, una aplicación puede definir una clave (por ejemplo, `el_significado_de_la_vida`) y proporcionar un valor coincidente (por ejemplo, 42) que luego se puede recuperar de la misma manera al proporcionar la clave. Una base de datos de valores clave trata los datos que contiene como un **Fichero de Basura Binaria** (en idioma inglés *Opaque binary blob* u OBB); depende de la aplicación entender cómo está estructurada.

Las bases de datos de claves-valores a menudo se describen como de alto rendimiento, eficientes y potenciadas. Los casos de uso comunes para las bases de datos de valores clave son el almacenamiento en [caché](#), la [cola de mensajes](#) y la [administración de sesiones](#).

Algunos almacenes de datos clave-valor de código abierto populares son:

Base de datos

[Redis](#)

Descripción

Un almacén de datos en memoria utilizado como base de datos, caché o intermediario de mensajes. Redis admite una variedad de estructuras de datos, que van desde cadenas hasta mapas de bits, flujos e índices espaciales.