

# Un Estudio de Procesos de Diseño de Bases de Datos NoSQL

Luciano Marrero<sup>1</sup> , Verena Olsowy<sup>1</sup> , Fernando Tesone<sup>1</sup> , Pablo Thomas<sup>1</sup> , Leandro Corbalán<sup>1</sup> , Juan Fernández Sosa<sup>1</sup> , Patricia Pesado<sup>1</sup> 

<sup>1</sup> Instituto de Investigación en Informática LIDI  
Facultad de Informática - Universidad Nacional de La Plata – Argentina  
Centro Asociado Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

{lmarrero, volsowy, ftesone, pthomas, corbalan, jfernandez  
ppesado}@lidi.info.unlp.edu.ar

**Resumen:** Los Sistemas de Gestión de Bases de Datos (SGBDs) NoSQL no poseen un proceso tradicional para el modelado de datos. Para los SGBDs relacionales, generalmente, existe un proceso de diseño que se inicia con un modelo conceptual, luego es derivado a un modelo lógico relacional y finaliza con un modelo físico de la Base de Datos. La ausencia de documentación en un proceso de diseño de Bases de Datos puede llevar a una interpretación errónea sobre la semántica de los datos que se encuentran almacenados. En este trabajo se plantea una revisión bibliográfica sobre los procesos de diseño para obtener un esquema de los datos en Bases de Datos NoSQL. El principal objetivo es el de realizar un análisis comparativo a partir de las propuestas existentes.

**Keywords:** Proceso de Diseño de Bases de Datos, Modelado de Datos en Bases de Datos NoSQL, Almacenamiento no estructurado de datos.

## 1. Introducción

Las metodologías tradicionales de diseño y construcción de Bases de Datos relacionales han sido ampliamente estudiadas, aplicadas y refinadas por décadas. Inicialmente, en base a una especificación de requerimientos, se presenta una descripción conceptual de alto nivel. Posteriormente, se realiza un mapeo del modelo conceptual a algún tipo de modelo lógico aplicando las reglas adecuadas para un determinado tipo de Base de Datos, siendo el más popular, el modelo lógico relacional [26]. Finalmente, se obtiene una representación del esquema físico de la Base de Datos. Los principios o reglas que se aplican a un modelo de datos relacional no resultan apropiados para una Base de Datos NoSQL. Esto se debe a que NoSQL no representa un tipo de Base de Datos, sino que son un conjunto de tipos de Bases de Datos y cada una de ellas posee su propia forma de estructurar y almacenar sus datos. En general, NoSQL, se basa en la redundancia, la desnormalización y las consultas que se impactarán sobre el sistema. Las Bases de Datos relacionales y NoSQL

proponen modelos diferentes, por lo tanto, su proceso de diseño también tendrá que ser diferente [27, 28].

En este trabajo se propone realizar una revisión bibliográfica sobre procesos y/o metodologías de diseño para Base de Datos NoSQL. Además, se realiza un análisis comparativo con el objetivo de brindar una guía rápida que muestre las distintas características de cada enfoque propuesto.

A partir de la sección 2, el trabajo se organiza del siguiente modo: se detallan las características de NoSQL, en la sección 3 se presentan los trabajos relacionados, en la sección 4 se realiza el estudio propuesto, en la sección 5 se presenta un análisis comparativo, y finalmente, en la sección 6 y 7 se expresan las conclusiones generales y trabajos futuros.

## 2. Características de las Bases de Datos NoSQL

Las Bases de Datos NoSQL representan un conjunto de tipos de Bases de Datos que poseen sus propias implementaciones para el almacenamiento de datos. NoSQL se distingue de los tradicionales sistemas de gestión de Bases de Datos relacionales en diversos aspectos; no poseen un lenguaje de consulta estructurado (SQL) como lenguaje principal, no requieren de una estructura fija y tabular, no soportan operaciones JOIN, no garantizan por completo las propiedades de ACID (atomicidad, consistencia, aislamiento y durabilidad), y en general, su estructura se basa en la escalabilidad horizontal [27, 29, 30].

NoSQL propone un sistema llamado “BASE (Básicamente Disponible, Estado Suave, Consistencia Eventual)”. A través de estas propiedades se logra disponibilidad básica (Base Availability), esto significa que el sistema se encontrará disponible la mayoría del tiempo. Con el estado débil (Soft State) el sistema se vuelve más flexible en cuanto a consistencia y con la consistencia eventual (Eventual Consistency) se garantiza que el sistema eventualmente se volverá consistente [28, 29, 30].

Dependiendo de la forma en que se almacene la información, existen cuatro categorías principales de almacenamiento para Bases de Datos NoSQL.

**Almacenamiento Clave/Valor:** simples en cuanto a su implementación, almacenan datos como un conjunto de pares “clave/valor” (key-value). La clave representa un identificador único que puede retornar un objeto complejo y arbitrario de información, denominado valor (value). Por ejemplo, Redis y Amazon DynamoDB, entre otros, implementan este tipo de almacenamiento [38, 39].

**Almacenamiento Documental:** el concepto central de este tipo de almacenamiento es el documento. Una Base de Datos NoSQL Documental, almacena, recupera y gestiona datos de documentos. Estos documentos encapsulan y codifican datos o información bajo algún formato estándar (XML, YAML, JSON, BSON). Por ejemplo, MongoDB y Apache CouchDB, entre otros, son implementaciones de Bases de Datos Documentales [34, 35].

**Almacenamiento de Familia de Columnas:** en este tipo de almacenamiento los datos se encuentran organizados por columnas, en lugar de filas. Las Bases de Datos que utilizan esta forma de almacenamiento tienden a ser un híbrido entre las Bases de

Datos Relacionales y la tecnología orientada a columna. Por ejemplo, Cassandra y Apache HBase, entre otros, utilizan este tipo de almacenamiento [40, 41].

**Almacenamiento de Grafos:** en este tipo de almacenamiento se representa la Base de Datos bajo el concepto de un grafo, permitiendo almacenar la información como nodos y sus respectivas relaciones, con otros nodos, mediante aristas. Se aplica la teoría de grafos para recorrer la estructura. Son útiles para almacenar información en modelos que poseen numerosas relaciones entre sus datos. Neo4j y OrientDB, entre otros, implementan este tipo de almacenamiento [36, 37].

### **3. Trabajos relacionados**

Se han recopilado tres trabajos relacionados que realizan revisiones bibliográficas sobre el diseño de Bases de Datos NoSQL.

En [24] se realiza una revisión sistemática de la literatura para el diseño de Bases de Datos no Relacional. El análisis realizado en este trabajo es un aporte importante para futuras investigaciones y la evolución de los métodos de diseño.

En [25] se realiza una revisión de los enfoques existentes para el diseño de Bases de Datos en general con el objetivo de detectar consideraciones y necesidades en comparación al diseño tradicional de Base de Datos (Relacional).

En [42] se realiza un estudio exhaustivo sobre decisiones de diseño para el almacenamiento no estructurado de información; además se incluyen otras temáticas como modelo de consistencia, partición de datos y teorema de CAP.

En este trabajo, a diferencia de [24], [25] y [42], se presenta una revisión bibliográfica con el aporte de una grilla que permite realizar una comparación rápida y precisa al momento de tener que seleccionar o analizar las distintas técnicas o procesos para el diseño de Bases de Datos NoSQL.

### **4. Enfoques de Diseño para Bases de Datos NoSQL**

En esta sección se presenta un estudio realizado sobre 23 propuestas de diseño de Bases de Datos que utilizan almacenamiento no estructurado de datos, con el objetivo de obtener el panorama actual para el modelado y el diseño de este tipo de Base de Datos.

Cuando se piensa en el diseño de una Base de Datos NoSQL, generalmente, se presenta directamente su estructura a nivel físico teniendo en cuenta el problema a resolver. Realizar un modelo de datos en un nivel conceptual para NoSQL es algo que aún es tema en discusión. Generalmente, un modelo de datos conceptual describe sus componentes en términos de entidades y cómo éstas se relacionan, independientemente de la tecnología a utilizar. Existen aspectos como la redundancia, la desnormalización y las consultas del sistema que en una etapa conceptual aún no son consideradas.

A continuación, se presenta un breve resumen sobre distintas publicaciones consultadas que han abordado la temática de procesos de diseño para Bases de Datos NoSQL desde diferentes perspectivas. Para la búsqueda de los artículos, se han consultado diversas fuentes, por ejemplo, Google Scholar [33], IEEE Xplore [31], Springer [32], entre otros. Además, se tuvieron en cuenta las referencias que expone cada uno con el objetivo de seguir un lineamiento temporal del trabajo.

Los autores de “*ToNoSQLModel Process / NoSQLToUML*” [1, 14], proponen un enfoque automático para obtener un modelo físico (en formato JSON) a partir del motor de Base de Datos No Relacional MongoDB [34]. Este proceso se lleva a cabo a través de una secuencia de transformaciones formales utilizando QVT (Query/View/Transformation) y siguiendo un conjunto de reglas bien definidas.

Los autores de “*ToConceptualModel*” [4], amplían [1] proponiendo un enfoque de Ingeniería Inversa para Bases de Datos NoSQL orientadas a documentos realizando una transformación del modelo físico en un modelo conceptual mediante un diagrama de clases UML (Unified Modeling Language). Para su implementación utilizan Eclipse Modeling Framework (EMF, por sus siglas en Inglés) y QVT (Query/View/Transformation).

Los autores en [2] presentan “*JSON Discoverer*”, una herramienta que permite mapear un documento JSON y obtener su esquema implícito a través de un diagrama UML. La herramienta generada se puede ver en la siguiente URL: <http://som-research.uoc.edu/tools/jsonDiscoverer/#/>. La herramienta tiene como objetivo asistir a los desarrolladores en la realización de tareas que involucren inferir y visualizar el esquema implícito de los datos que poseen un documento JSON.

En [3] se presenta un nuevo enfoque de extracción de esquemas de documentos JSON mediante la introducción de un algoritmo para la extracción de esquemas que operan fuera de un motor de Bases de Datos NoSQL. En lugar de diseñar el esquema de forma adelantada, se realiza una extracción, algo que puede verse como un proceso de Ingeniería Inversa.

En [5] se presenta un enfoque basado en una Arquitectura Dirigida por Modelos (Model-Driven Architecture o MDA) que permite lograr una Ingeniería Inversa de las Bases de Datos orientadas a Grafos [36, 37]. Parte de un esquema físico descrito mediante el lenguaje Cypher propuesto por Neo4j [36]. Luego, se deduce un grafo lógico y finalmente se mapea el grafo lógico obtenido a un esquema conceptual a través de un Diagrama de Entidad-Relación Extendido (EERD).

En [6] se propone un método para el diseño no relacional denominado “*NoAM (NoSQL Abstract Model)*”. Se plantea un proceso que incluye una fase inicial conceptual (expresada en UML), seguida de una fase de diseño lógico NoAM, propuesto por los autores e independiente del sistema, y una fase final en la que se tiene en cuenta las características de los sistemas individuales. NoAM explota los puntos comunes existentes en distintos sistemas NoSQL e introduce abstracciones para equilibrar diferencias y variaciones.

En [7, 8] se presenta una solución de Ingeniería Inversa para inferir esquemas de Bases de Datos NoSQL teniendo en cuenta el versionado de los datos. El esquema obtenido se expresa en UML y la solución se ha implementado con MDE (Model-Driven Engineering) para conseguir independencia del tipo de Base de Datos.

Los autores en [9] proponen un marco en que, dado un cambio en el modelo conceptual, identifica lo que se debe modificar en un esquema de Base de Datos NoSQL y los datos subyacentes. El trabajo se centra en el estudio de siete tipos de cambios del modelo conceptual y para cada cambio se describe la transformación requerida en el esquema de la Base de Datos para mantener la consistencia entre el esquema y el modelo.

En [10] se plantean nuevas directrices de modelado para Bases de Datos NoSQL Documental. Estas directrices abarcan tanto las etapas lógicas como las físicas de los diseños. Cada una de ellas se desarrolla sobre conocimiento empírico desarrollando un enfoque exploratorio mediante consultas a expertos.

Los autores en [11] proponen “*Schema Design Advisor Model (SDAM)*” y un algoritmo automático para el problema de diseñar un esquema de Base de Datos Documental. Se logra un modelo de asesoramiento para el diseño de esquemas documentales, a partir de un esquema relacional teniendo en cuenta las consultas del sistema.

En [12] se presenta “*Mortadelo*”, un proceso de diseño de Base de Datos NoSQL basado en modelos, en donde, a partir de un esquema conceptual se puede generar de forma automática una implementación para una Base de Datos NoSQL. Este proceso se puede personalizar, de modo que algunas compensaciones de diseño se pueden gestionar de manera diferente según las necesidades de cada contexto. Con “*Mortadelo*” se generaron implementaciones para Cassandra [40] y MongoDB [34] a partir de un mismo modelo conceptual.

En [13] se presenta el diseño de un método estructurado para la selección y el diseño del modelo de Base de Datos basado en una variedad de factores (relaciones entre los datos, requisitos funcionales y requisitos no funcionales). El método recomendará qué modelos de Base de Datos son los más apropiados para una aplicación y sugerirá un diseño para los modelos recomendados.

Los autores en [15] proponen una metodología de modelado basada en consulta para el motor de base de datos NOSQL Apache Cassandra [40]. Definen importantes principios de modelado, reglas de mapeo y patrones de mapeo, para lograr un modelo lógico de datos. Además, diseñaron e implementaron una herramienta de modelado de datos basada en la web denominada KDM ([www.cs.wayne.edu/andrey/kdm](http://www.cs.wayne.edu/andrey/kdm)).

En [16] se propone un sistema para recomendar esquemas de Bases de Datos no relacionales a partir del modelo conceptual de datos de la aplicación. Además, se implementó un prototipo basado en este enfoque para el motor de Base de Datos Apache Cassandra [40]. Este prototipo denominado “*NoSQL Schema Evaluator (NoSE)*” produce esquemas eficientes y permite examinar más alternativas de las que serían posibles con un enfoque manual.

Los autores en [17] proponen un modelo conceptual común para diversos tipos de Bases de Datos NoSQL. También han ideado un lenguaje de especificación de datos NoSQL para representar un modelo de datos de nivel lógico equivalente e independiente de cualquier representación de nivel físico. Además, se han formalizado e ilustrado distintas reglas de validación relativas al modelo conceptual propuesto, utilizando un caso de estudio adecuado.

En [18] se propone un mecanismo para transformar un modelo conceptual en un modelo lógico para Bases de Datos NoSQL. El modelo lógico se representa en notación JSON. La representación lógica propuesta es capaz de representar datos estructurados y semiestructurados, y es capaz de una mayor transformación hacia tipos heterogéneos de Bases de Datos NoSQL.

Los autores de [19] abordan el problema de diseño de Bases de Datos NoSQL. Proporcionan métodos que asignan un modelo de datos estándar a un modelo de datos admitido por los servicios de datos NoSQL, utilizando “AWS *DynamoDB*” (Clave-Valor) como Base de Datos específica [39].

En [20] se muestra una estrategia de diseño en esquemas para el motor de Base de Datos “DynamoDB” (Clave-Valor) [39], teniendo en cuenta los patrones de acceso a los datos. Además, se presenta una simulación del enfoque.

En [21] se presenta una metodología de apoyo al diseño para Bases de Datos NoSQL Clave-Valor. Esta metodología propone un modelo lógico que considera los patrones de acceso y/o consulta a una Base de Datos Clave-Valor [38, 39].

Los autores en [22] proponen un método de modelización lógica adecuado para las Bases de Datos NoSQL Documentales, incorporando un nuevo tipo de diagrama denominado Diagrama de Interrelación de Documentos (DID). Este trabajo se basa en la modelización conceptual (DER) y suman patrones de consulta que surgen del análisis de requisitos.

En [23] se presenta un método de mapeo para el modelado de Bases de Datos no relacionales. Mediante la aplicación de reglas, se unifica la forma de mapear esquemas Documentales, Familia de Columnas y Grafos.

## 5. Diseño de Bases de Datos NoSQL. Análisis Comparativo

Luego de realizar el análisis de cada uno de los trabajos recopilados, se presenta una grilla comparativa para visualizar las distintas características que poseen los diferentes enfoques de procesos y/o métodos de diseño para motores de Bases de Datos NoSQL (clave-valor, documental, familia de columnas y grafos). En este trabajo se han definido los siguientes criterios:

- **Aplicable a:** estrategias tratadas (clave-valor, documental, familia de columnas o grafos) y/o motor de Bases de Datos NoSQL específico.
- **Enfoque:** define formalmente un proceso o es una metodología propuesta sobre el tema.
- **Herramienta de automatización:** Si implementan una herramienta para automatizar alguna parte del proceso.
- **Pautas bien definidas:** Si define todos los pasos necesarios para el proceso o técnica que describe.
- **Diseño completo:** si parte con la realización de modelo inicial o utiliza algún modelo conocido.
- **Diagramas:** diagramas que se utilizan en el proceso descripto.
- **Referencia:** artículo y/o trabajo analizado.

A continuación, en las tablas 1, 2, 3, 4 y 5 se presenta el resultado del análisis realizado.

**Tabla 1 - Enfoques aplicables a almacenamiento *Documental***

Aplicable a	Enfoque	Herramienta de automatización	Pautas definidas	Diseño completo	Diagramas	Referencia
MongoDB	Ingeniería inversa	No	Si	No	Físico definido por los autores y UML	[1, 4, 14]
General	Herramienta web	Si	No	No	UML y Físico JSON	[2]
General	ingeniería inversa	No	Si	No	XML y JSON	[3]
General	Grupo de normas	No	Si	Si	UML	[10]
MongoDB	Relacional a documental	No	Si	No	Físico	[11]
General	Modelado de datos	No	Si	No	DID (Diagrama de interrelación de documentos)	[22]

**Tabla 2 - Enfoques aplicables a almacenamiento *Clave-Valor***

Aplicable a	Enfoque	Herramienta de automatización	Pautas definidas	Diseño completo	Diagramas	Referencia
General	Modelo de datos	No	No	Si	DER y UML	[21]

**Tabla 3 - Enfoques aplicables a almacenamiento de *Grafos***

Aplicable a	Enfoque	Herramienta de automatización	Pautas definidas	Diseño completo	Diagramas	Referencia
Neo4j	Ingeniería inversa	No	Si	No	Esquema físico, Grafo lógico y Modelo conceptual	[5]

**Tabla 4 - Enfoques aplicables a almacenamiento de *Familia de Columnas***

Aplicable a	Enfoque	Herramienta de automatización	Pautas definidas	Diseño completo	Diagramas	Referencia
Cassandra	Actualización	No	Si	No	No aplica	[9]
Cassandra	Modelado de datos	Si	Si	Si	Modelo conceptual y Diagrama Chebotko	[15]
General	Modelado de datos	Solo en Cassandra	Si	Si	Modelo conceptual y físico	[16]

**Tabla 5 - Enfoques aplicables a más de un tipo de almacenamiento**

Aplicable a	Enfoque	Herramienta de automatización	Pautas definidas	Diseño completo	Diagramas	Referencia
Documental, Clave-Valor y Familia de Columnas	Ingeniería inversa	No	Si	No	JSON y UML	[7, 8]
Documental, Clave-Valor y Familia de Columnas	Proceso de diseño	No	Si	Si	UML, Lógico NoSQL y Físico	[6]
General	Proceso de diseño	No	Si	Si	Generic Data Metamodel (GDM), Metamodelo para Familia de Columnas y Metamodelo para Documental	[12]
General	Asesor de base de datos	No	No	Si	Ninguno	[13]
General	Proceso de diseño	No	Si	Si	Conceptual NoSQL y Lógico JSON	[17, 18]
DynamoDB	Modelado de datos	No	Si	Si	Modelo de datos estándar y Físico DynamoDB	[19]
DynamoDB	Modelado de datos	No	Si	Si	Hipergrafo de modelo conceptual, Esquema jerárquico y Esquema Físico	[20]

Documental, Familia de Columnas y Grafos	Modelado de datos	No	Si	No	TPC-Hbenchmark	[23]
---	----------------------	----	----	----	----------------	------

## 6. Conclusiones

Este trabajo se centra en una revisión de la bibliografía existente para el estudio y análisis sobre distintos procesos y/o metodologías de diseño para Bases de Datos NoSQL. Para ello, se consultaron diversas fuentes, como Google Scholar [33], IEEE Xplore [31], Springer [32] entre otros. Los criterios de exploración y selección se basaron principalmente en la identificación de artículos que traten sobre diseño Bases de Datos NoSQL para alguna de las 4 categorías principales de almacenamiento no estructurado de información (documental, clave-valor, familia de columnas y grafos). En este contexto, se analizaron más de 30 artículos. De este grupo, se han seleccionado unos 23 trabajos que proponen o se aproximan al desarrollo de un proceso o metodología para el diseño de Bases de Datos NoSQL con líneas diferenciadas entre sí, abarcando distintas alternativas y características. Se pretende obtener un panorama tentativo sobre cómo ha evolucionado el diseño de Bases de Datos NoSQL en los últimos años.

A través de este estudio, se puede observar que existe una gran variedad de enfoques y para un mismo tipo de almacenamiento, se tienen distintas variantes según la visión del autor. No contar con la posibilidad de tener un proceso de diseño unificado, como sucede para las Bases de Datos Relacionales, probablemente implique tener que realizar grandes modificaciones al modelo conceptual (si es que ha sido realizado) ante un cambio en el modelo físico. Además, algunos de los enfoques, son específicos para un motor de Base de Datos NoSQL, o el proceso de diseño propuesto, involucra numerosas pautas o reglas que los diseñadores deben aplicar de forma manual, algo que puede ser propenso a cometer errores.

En resumen, si bien desde hace varios años existen diversos estudios sobre el diseño de Bases de Datos NoSQL, aún no se ha encontrado un proceso de diseño adoptado formalmente que contenga una serie de pasos establecidos y que sea genérico a casi todas las formas de almacenamiento no estructurado de datos, similar a lo que sucede en Bases de Datos Relacionales (Modelado Conceptual, Modelado Lógico y Modelado Físico). Además, existen propiedades que las Bases de Datos NoSQL poseen, que son muy poco consideradas, como la disponibilidad, la consistencia eventual y la tolerancia a particiones.

## 7. Trabajo Futuro

Como trabajo futuro, se prevé extender la revisión bibliográfica realizada con el objetivo de tener un panorama más amplio sobre el diseño de Bases de Datos NoSQL.

Además, se pretende seleccionar aquellos enfoques y/o procesos que abarcan todas las etapas del diseño (Conceptual, Lógico y Físico) e incorporarlos a proyectos reales con el objetivo evaluar su aplicabilidad y aportar mejoras o nuevas características que aún no se tienen en cuenta durante el diseño de la Base de Datos, por ejemplo, la escalabilidad horizontal.

## 8. Bibliografía

1. Brahim, A.; Ferhat, R. and Zurfluh, G. (2019). Model Driven Extraction of NoSQL Databases Schema: Case of MongoDB. In Proceedings of the 11th International Joint Conference on Knowledge Discovery. V.1: KDIR, ISBN 978-989-758-382-7, pages 145-154.
2. Izquierdo, J. L. C., & Cabot, J. (2016). JSONDiscoverer: Visualizing the schema lurking behind JSON documents. Knowledge-Based Systems, 103, 52-55.
3. M. Klettke, U. Störl, S. Scherzinger, Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores, in: BTW conf., 2015, pp. 425–444.
4. Fatma ABDELHEDI, Amal AIT BRAHI, Rabah TIGHILT FERHAT, Gilles ZURFLUH. Reverse engineering approach for NoSQL databases. 22nd International Conference, DaWaK 2020, Bratislava, Slovakia, September 14–17, 2020.
5. Isabelle Comyn-Wattiau; Jacky Akoka. Model driven reverse engineering of NoSQL property graph databases: The case of Neo4j. 2017 IEEE International Conference on Big Data..
6. Paolo Atzeni, Francesca Bugiotti, Luca Cabibbo, Riccardo Torlone. Data Modeling in the NoSQL World. HAL open science. <https://hal.archives-ouvertes.fr/hal-01611628>.
7. Severino Feliciano Morales, Jesús García Molina, Diego Sevilla Ruiz. Inferencia del esquema en bases de datos NoSQL a través de un enfoque MDE.
8. Diego Sevilla Ruiz, Severino Feliciano Morales, Jesús García Molina. Inferring Versioned Schemas from NoSQL Databases and Its Applications. [https://link.springer.com/chapter/10.1007/978-3-319-25264-3\\_35](https://link.springer.com/chapter/10.1007/978-3-319-25264-3_35) (Springer).
9. Pablo Suárez Otero, Michael J. Mior, Maria José Suárez Cabal, Javier Tuya. Maintaining NoSQL Database Quality During Conceptual Model Evolution.
10. Paolo Atzenia, Francesca Bugiotti, Luca Cabibbo, Riccardo Torlone. Data Modeling Guidelines for NoSQL Document-Store Databases. International Journal of Advanced Computer Science and Applications.
11. Basant Namdeo, Ugrasen Suman. Schema Design Advisor Model for RDBMS to NoSQL Database Migration.
12. Alfonso de la Vega, Diego García, Saiz Carlos Blanco, Marta Zorrilla, Pablo Sánchez. Mortadelo: Automatic generation of NoSQL stores from platform-independent data models. Future Generation Computer Systems. Volume 105, April 2020, Pages 455-474.
13. Noa Roy-Hubara. The Quest for a Database Selection and Design Method. CAiSE 2019 (31st International Conference on Advanced Information Systems Engineering). 3-7 June 2019, Rome, Italy.
14. Amal AIT BRAHIM; Rabah TIGHILT FERHAT; Gilles ZURFLUH. Extraction process of conceptual model from a document-oriented NoSQL database. 2019 11th International Conference on Knowledge and Systems Engineering (KSE).
15. Artem Chebotko, Andrey Kashlev, Shiyong Lu. A Big Data Modeling Methodology for Apache Cassandra. 2015 IEEE International Congress on Big Data.
16. Michael Joseph Mior, Kenneth Salem; Ashraf Abounaga, Rui Liu. NoSE: Schema Design for NoSQL Applications. IEEE Transactions on Knowledge and Data Engineering.
17. Shreya Banerjee, Anirban Sarkar. Modeling NoSQL Databases: From Conceptual to Logical Level Design. 3rd International Conference on Applications and Innovations in Mobile Computing (AIMOC – 2016) At: Kolkata, India.

18. Shreya Banerjee, Anirban Sarkar. Logical level design of NoSQL databases.2016 IEEE Region 10 Conference (TENCON).
19. Sudarshan S. Chawathe. Data Modeling for a NoSQL Database Service. 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference 2019 (New York).
20. W. Y. Mok.A Feasible Schema Design Strategy for Amazon DynamoDB: A Nested Normal Form Approach. Industrial Engineering and Engineering Management (IEEM 2020).
21. Gerardo ROSSEL, Andrea MANNA. A Modeling methodology for NoSQL Key-Value databases.Database Systems Journal, Bucharest, Romania. 12-18, August.
22. Diseño de Bases de Datos basadas en Documentos: Modelo de Interrelación de Documentos. Rossel Gerardo, Manna Andrea. XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016). Páginas 662-671. <http://sedici.unlp.edu.ar/handle/10915/56746>.
23. Afef Gueidi, Hamza Gharsellaoui, Samir Ben Ahmed.Towards Unified Modeling for NoSQL Solution Based on Mapping Approach. 25th KES-2021. <http://kes2021.kesinternational.org/>
24. Noa Roy-Hubara, Arnon Sturm.Design methods for the new database era: a systematic literature review. Software and Systems Modeling | Issue 2/2020 (<https://www.springerprofessional.de/en/software-systems-modeling/5337962>).
25. Noa Roy-Hubara, Arnon Sturm.Exploring the Design Needs for the New Database Era. Springer ([https://link.springer.com/chapter/10.1007/978-3-319-91704-7\\_18](https://link.springer.com/chapter/10.1007/978-3-319-91704-7_18)).
26. Carlo Batini, Stefano Ceri, Shamkant B. Navathe. Diseño Conceptual de Bases de Datos, un enfoque de entidades-interrelaciones. ISBN 0-201-60120-6 (1994).
27. Análisis de performance en Bases de Datos NoSQL y Bases de Datos Relacionales. Pesado P., Thomas P., Delía L., Marrero L., Olsowy V., Tesone F.. XXVI Congreso Argentino de Ciencias de la Computación (CACIC 2020). ISBN 978-987-4417-90-9. <http://sedici.unlp.edu.ar/handle/10915/114202>.
28. Un estudio comparativo de bases de datos relacionales y bases de datos NoSQL. Pesado P., Thomas P., Delía L., Marrero L., Olsowy V., Tesone F., Fernandez S. J. XXV Congreso Argentino de Ciencias de la Computación (CACIC 2019). ISBN 978-987-688-377-1. <http://sedici.unlp.edu.ar/handle/10915/91403>.
29. Aspectos de Ingeniería de Software, Bases de Datos Relacionales, y Bases de Datos No Relacionales y Bases de Datos Como Servicios en la Nube para el desarrollo de Software Híbrido. XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021). ISBN 978-987-24611-3-3; 978-987-24611-4-0. <http://sedici.unlp.edu.ar/handle/10915/120139>.
30. NoSQL: modelos de datos y sistemas de gestión de bases de datos. XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste). <http://sedici.unlp.edu.ar/handle/10915/67258>.
31. IEEE Xplore.<https://ieeexplore.ieee.org/Xplore/home.jsp>. Mayo de 2022.
32. Springer. <https://link.springer.com/>. Mayo de 2022.
33. Google Scholar. <https://scholar.google.com/>. Mayo de 2022.
34. MongoDB. <https://www.mongodb.com/es>. Mayo de 2022.
35. CouchDB. <https://couchdb.apache.org/>. Mayo de 2022.
36. Neo4j. <https://neo4j.com/>. Mayo de 2022.
37. OrientDB. <https://orientdb.org/>. Mayo de 2022.
38. Redis. <https://redis.io/>. Mayo de 2022.
39. Amazon DynamoDB. <https://aws.amazon.com/es/dynamodb/>. Mayo de 2022.
40. Apache Cassandra. [https://cassandra.apache.org/\\_/index.html](https://cassandra.apache.org/_/index.html). Mayo de 2022.
41. Apache HBase. <https://hbase.apache.org/>. Mayo de 2022.
42. Davoudian, Ali and Chen, Liu and Liu, Mengchi. A survey on NoSQL stores. ACM Computing Surveys (CSUR), volume=51, number=2, pages=1--43, year=2018, publisher=ACM New York, NY, USA.