



Bases Datos NoSQL

Grupo 2

Betty Arrobo
Jonathan Castro
Víctor Javier Cedeño
Juan Carlos Jaramillo



Características o atributos generales de las bases de datos NOSQL

- **Esquemas flexibles:** se pierde la estructura en tabla, pudiendo establecer diferentes niveles jerárquicos, e incluso pudiendo guardarse diferentes tipos de datos en un mismo campo.
- **Sin esquemas predefinidos:** no es necesario predefinir el esquema de la base de datos, sino que se pueden añadir nuevos campos a posteriori.
- **Escalabilidad horizontal:** Permiten el funcionamiento en clusters. Ya no es necesario tener todos los datos en un mismo equipo, por lo que cuando se requiere más espacio o capacidad de procesamiento, solamente hay que añadir una nueva máquina.
- **Replicabilidad y alta disponibilidad:** Al funcionar en clusters, es posible tener varias copias de un mismo documento en varios equipos, de modo que si uno de los equipos falla, automáticamente los otros toman el mando sin pérdidas de datos ni rendimiento.
- **Particionado:** La posibilidad de particionar se basa en que un documento puede distribuirse en varios equipos, eliminando la necesidad de que un solo disco tenga la capacidad de guardar todo un conjunto de datos.

Características de las bases de datos NoSQL

- **Velocidad de consultas:** Cuando las bases de datos crecen en tamaño, los tiempos de consulta de una SQL crecen, sobre todo cuando se requieren joins. Diferentes estructuras de datos y las arquitecturas en clusters reducen los tiempos de consulta cuando la cantidad de datos que maneja es elevada.
- **Velocidad de procesamiento:** Cuando se quieren hacer operaciones sobre toda la base de datos, las BBDD NoSQL llevan incorporados motores de procesamiento en paralelo, que reducen los tiempos de lectura y escritura sobre grandes tablas.
- **Alto rendimiento:** La base de datos NoSQL está optimizada para modelos de datos específicos y patrones de acceso que permiten un mayor rendimiento que el intento de lograr una funcionalidad similar con bases de datos relacionales.
- **Altamente funcional:** Las bases de datos NoSQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.

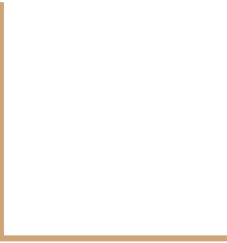
What is NoSQL Database



Tipos Bases Datos NoSQL



En general se considera que existen cuatro tipos diferentes

- **BD Orientadas a Documentos**
 - **BD Orientadas a Columnas**
 - **BD Clave Valor**
 - **BD en Grafo**
- 

Tipos Bases Datos NoSQL

Base de datos de documentos: Las bases de datos de documentos suelen almacenar documentos JSON, XML y BSON. Son similares a las bdd de valores clave, pero en este caso, un valor es un solo documento que almacena todos los datos relacionados con una clave específica. Los campos populares del documento pueden indexarse para proporcionar una recuperación rápida sin conocer la clave. Cada documento puede tener la misma estructura o una estructura diferente.

- **MongoDB:** probablemente la base de datos NoSQL más famosa del momento. MongoDB conseguía 150 millones de dólares en financiación, convirtiéndose en una de las startups más prometedoras. Algunas compañías que actualmente utilizan MongoDB son Foursquare o eBay.
- **CouchDB:** es la base de datos orientada a documentos de Apache. Una de sus interesantes características es que los datos son accesibles a través de una API Rest. Este sistema es utilizado por compañías como Credit Suisse y la BBC.

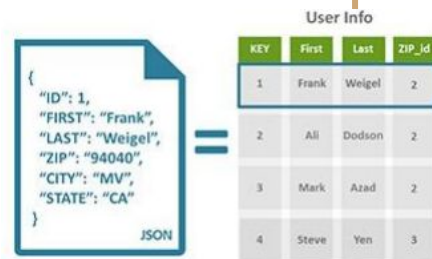


Diagram illustrating the mapping between a JSON document and a table structure. The JSON document on the left contains user information. The table on the right, titled "User Info", has columns for KEY, First, Last, and ZIP_Id, with four rows of data.

KEY	First	Last	ZIP_Id
1	Frank	Weigel	2
2	All	Dodson	2
3	Mark	Azad	2
4	Steve	Yen	3

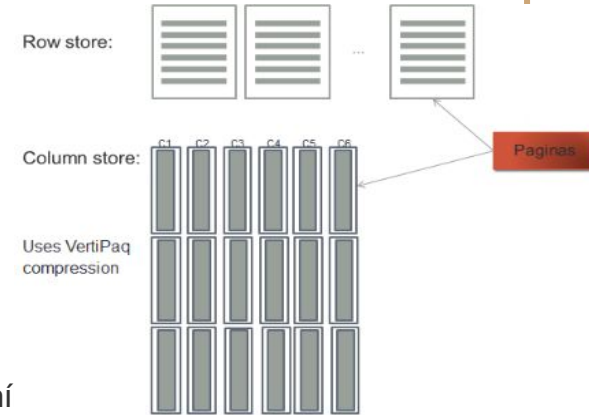
```
{
  Name: "Genbeta Dev",
  Tipo: "Blogging",
  Categorías:
  [
    {
      Título: "Desarrollo",
      Artículos: 89
    },
    {
      Título: "Formación",
      Artículos: 45
    }
  ]
}
```

Tipos Bases Datos NoSQL

Bases de datos de columnas: Las bases de datos NoSQL de columnas anchas almacenan datos en tablas con filas y columnas similares a las BBDD relacionales, pero los nombres y los formatos de las columnas pueden variar de fila a fila en la tabla. Las columnas de columnas anchas agrupan columnas de datos relacionados juntos. Una consulta puede recuperar datos relacionados en una sola operación porque sólo se recuperan las columnas asociadas con la consulta. En una BBDD, los datos estarían en diferentes filas almacenadas en diferentes lugares del disco, requiriendo múltiples operaciones de disco para su recuperación.

En esta categoría encontramos:

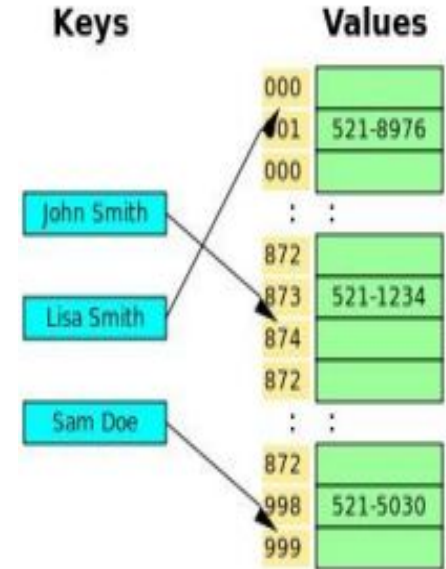
- **Cassandra:** incluida en esta sección, aunque en realidad sigue un modelo hí orientada a columnas y clave-valor. Es utilizada por Facebook y Twitter (aunque dejaron de usarla para almacenar tweets).
- **HBase.** Escrita en Java y mantenida por el Proyecto Hadoop de Apache, se utiliza para procesar grandes cantidades de datos. La utilizan Facebook, Twitter o Yahoo.



Tipos Bases Datos NoSQL

Bases de datos de valores clave: las bases de datos NoSQL de valor-clave hacen hincapié en la simplicidad y son muy útiles para acelerar una aplicación que admita el procesamiento de alta velocidad de lectura y escritura de datos no transaccionales. Los valores almacenados pueden ser cualquier tipo de objeto binario (texto, video, documento JSON, etc.) y se accede a través de una clave. La aplicación tiene control total sobre lo que se almacena en el valor, convirtiéndolo en el modelo NoSQL más flexible. Los datos se comparten y se replican en un clúster para obtener escalabilidad y disponibilidad. Por esta razón, las bases de datos de valores clave a menudo no admiten transacciones. Sin embargo, son muy eficaces en aplicaciones de escala que se ocupan de datos de alta velocidad y no transaccionales.

- Algunos ejemplos de este tipo son **Cassandra, BigTable o HBase**.

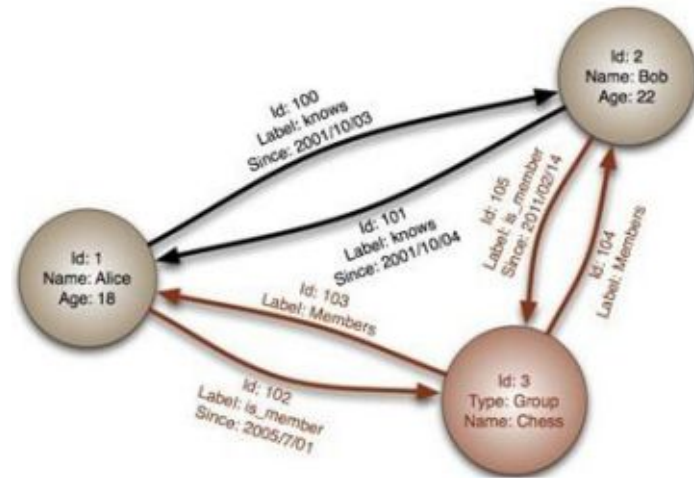


Tipos Bases Datos NoSQL

Bases de datos de grafos: una base de datos de grafos utiliza estructuras de grafos para almacenar, correlacionar y consultar relaciones. Proporcionan una adyacencia libre de índice, de modo que los elementos adyacentes se unen entre sí sin usar un índice.

Las bases de datos multimodales aprovechan alguna combinación de los cuatro tipos descritos anteriormente y, por lo tanto, pueden soportar una gama más amplia de aplicaciones

- Algunos ejemplos de este tipo son **Neo4j**, **InfoGrid** o **Virtuoso**



Tipos Bases Datos NoSQL



Son muchas las empresas grandes que hacen uso de este tipo de bases de datos no relacionales, como:

Cassandra: Facebook, Twitter, Apple, Netflix y Soundcloud...

HBase: Yahoo, Adobe...

Redis: Flickr, Instagram, Github...

Neo4j: Infojobs...

MongoDB: Verizon, Ebay, Sega, Google y Adobe, FourSquare, SourceForge, CERN...

Dynamo: AWS...

Modelos de datos NoSQL - Características

Modelo y Meta-modelo conceptual: Permitirá entender la relación y la organización de los posibles conceptos a alto nivel describiéndolos en más de un dominio, por lo tanto, vamos a utilizar el punto de vista de Información de *ArchiMate* (*Information Archimate Viewpoint*).

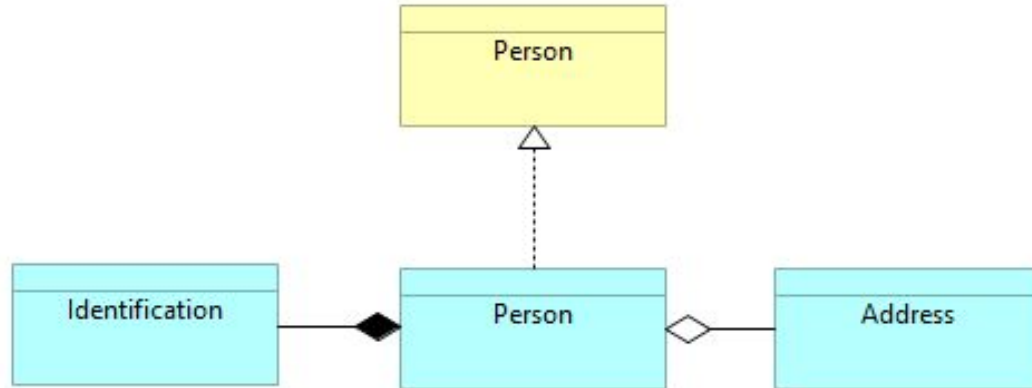
Metamodelo lógico: Permite visualizar todos los campos susceptibles a ser utilizados en los conceptos. Para esto, vamos a utilizar un diagrama de clases de UML (*UML Class diagram*). A cada concepto le corresponde una clase.

Modelo y Meta-modelo físico: Este modelo extiende el meta-modelo lógico y en él se describen las distintas formas en las que un concepto o esquema puede ser almacenado físicamente. Esta visualización permite verificar los modelos/meta-modelos conceptual y lógico para, mas tarde, visualizar las bases de datos en funcionamiento. Para esto vamos a utilizar un diagrama de clases de UML (también puede ser un diagrama de objetos UML).

Modelos de datos NoSQL - Características

Meta-modelo conceptual.

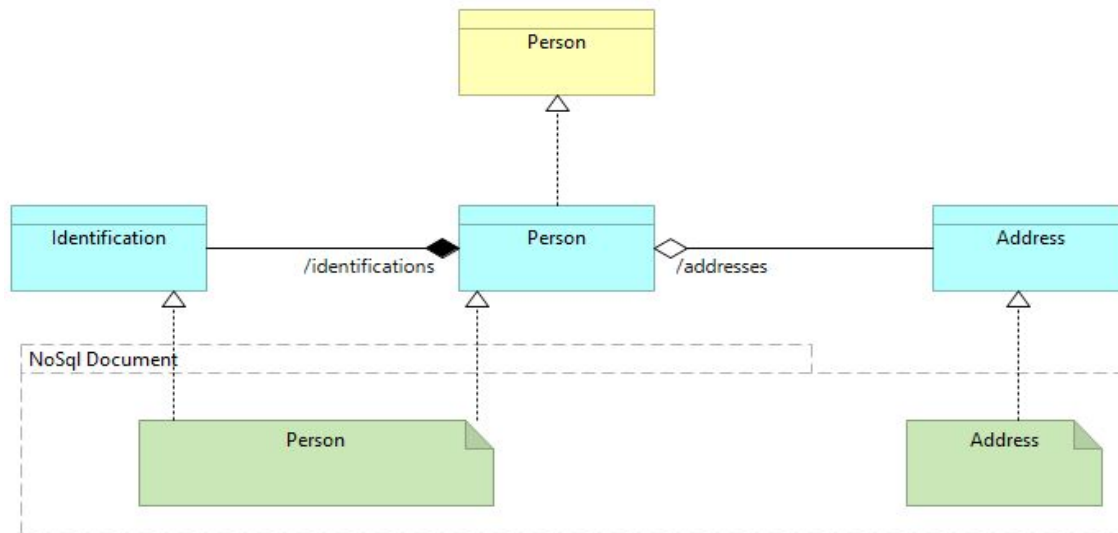
A continuación vemos un modelo en 2 dimensiones. Por un lado, en la dimensión de negocio se describe el concepto *Person* o persona. Por otro lado, en la dimensión de aplicación, el concepto *Person* se desglosa en tres entidades: *Person* se compone de *Identification*, y se le agrega *Address*, es decir, una persona debe tener al menos una identificación y puede tener al menos una dirección.



Modelos de datos NoSQL - Características

Meta-modelo conceptual.

Si nuestro modelo representa una base de datos documental. Agregamos al modelo conceptual el dominio de infraestructura, en el que se indican los tipos de documentos físicos que se van a crear, además, se indicará qué conceptos van a almacenar, como se puede ver a continuación:



Modelos de datos NoSQL - Características

Meta-modelo conceptual.

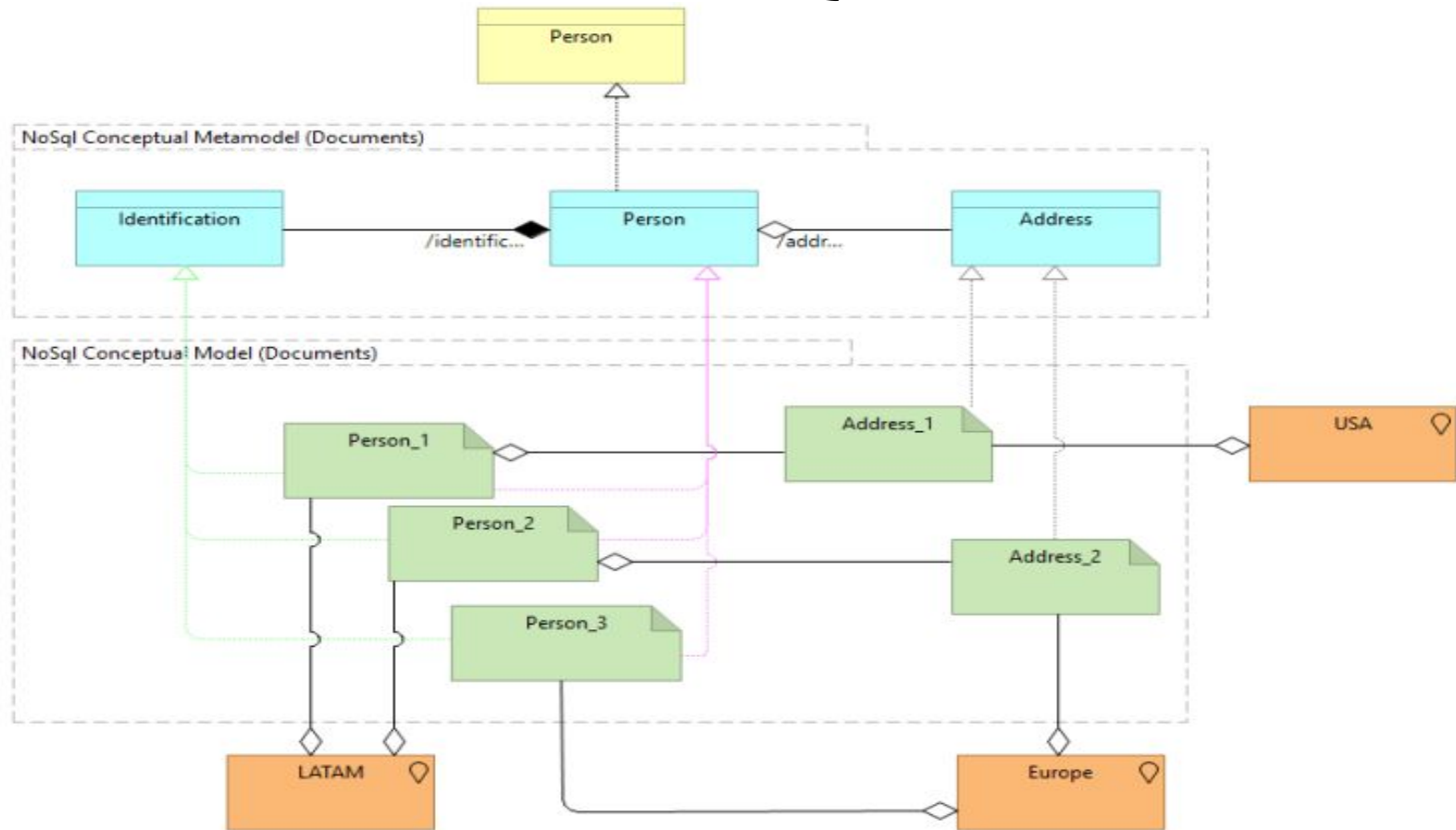
El artefacto *Person* (*ArchiMate Artifact*) representa a un tipo de documento NoSql que internamente describe a los conceptos *Person* e *Identification*, paralelamente un artefacto *Address* almacenará sólo un concepto de su mismo nombre. Por lo tanto, el artefacto *Person* en JSON,

Cabe resaltar que con el modelo anterior no se quiere decir que existen solo dos documentos, lo que se muestra es una representación de los tipos de documentos que pueden existir en la base de datos. Si se desea ser más explícito con el diseño, se pueden agregar más artefactos en el dominio de infraestructura, que puede servir para describir cómo escalar horizontalmente las bases de datos, a modo de ejemplo.

A continuación vemos el diseño completo. En él que se distinguen dos partes:

- El dominio de aplicación se concebirá como meta-modelo conceptual de una base de datos documental.
- Y el dominio de infraestructura como el Modelo Conceptual

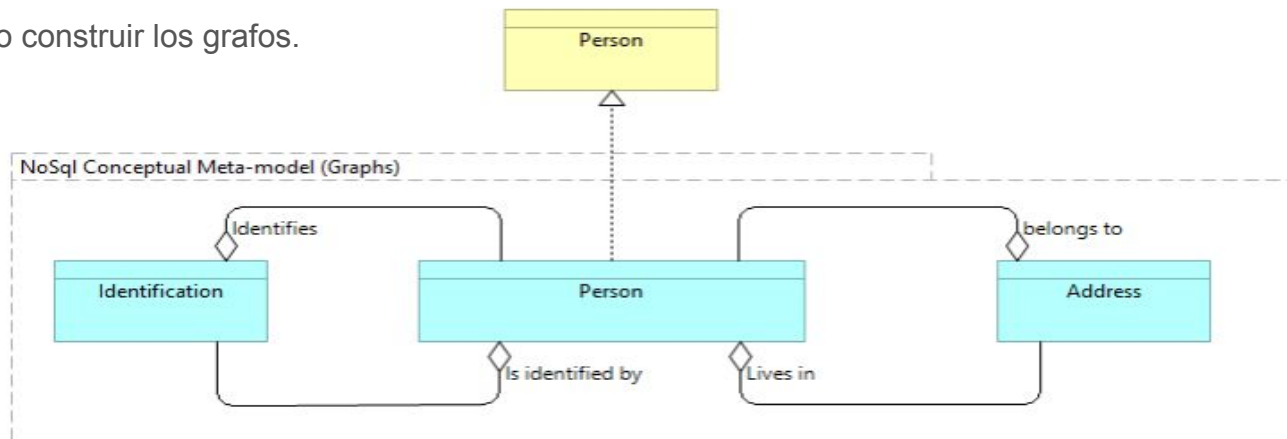
Modelos de datos NoSQL - Características



Modelos de datos NoSQL - Características

Meta-modelo conceptual.

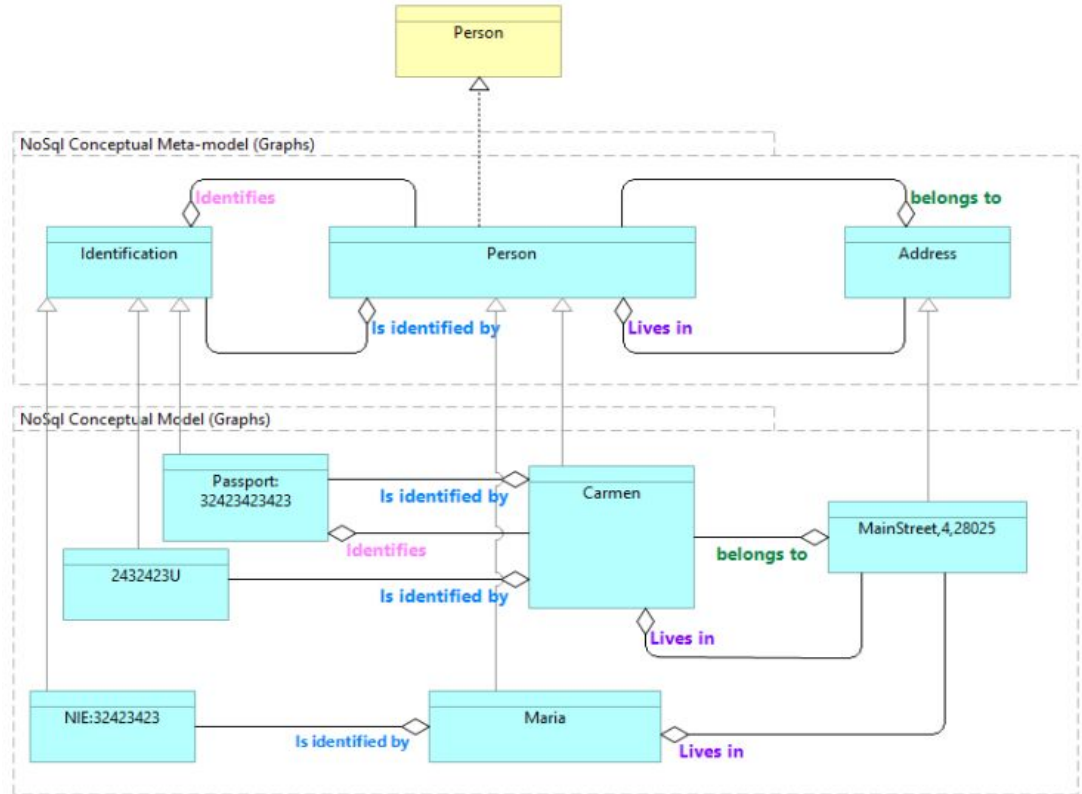
En el caso de que de bases de datos de grafos, el meta-modelo conceptual tiene la apariencia que se muestra a continuación. En el meta-modelo se observan las tres entidades descritas anteriormente, pero en este caso, las relaciones son de ida y vuelta entre entidades, es decir: *Person se identifica con un Identification*, un *Identification identifica a un Person*, un *Person vive en un Address* y un *Address pertenece a una o varias Person*. Este modelo es la base para luego construir los grafos.



Modelos de datos NoSQL - Características

Meta-modelo conceptual.

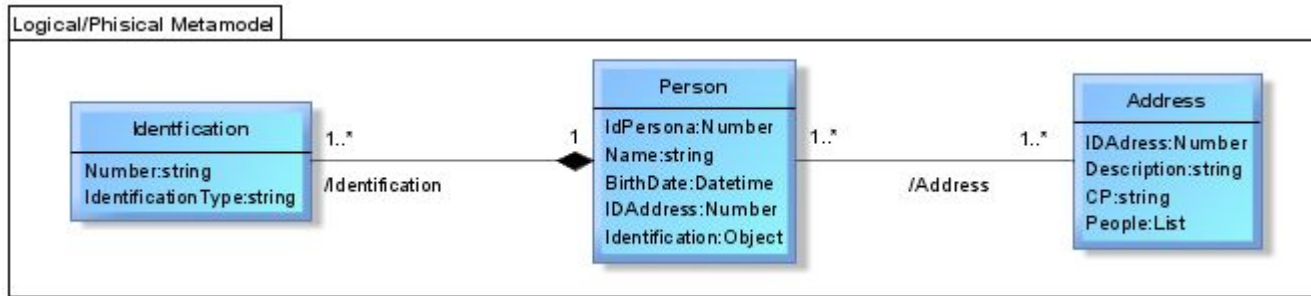
De esta manera, el meta-modelo describe un mapa de todas las relaciones base sobre el que luego se construye el grafo físicamente. A continuación, agregamos un ejemplo de Modelo de grafos que permite validar el meta-modelo. Se heredan los conceptos del meta-modelo para probar si el modelo de grafos se ajusta a nuestras necesidades de representación.



Modelos de datos NoSQL - Características

Meta-modelo Lógico.

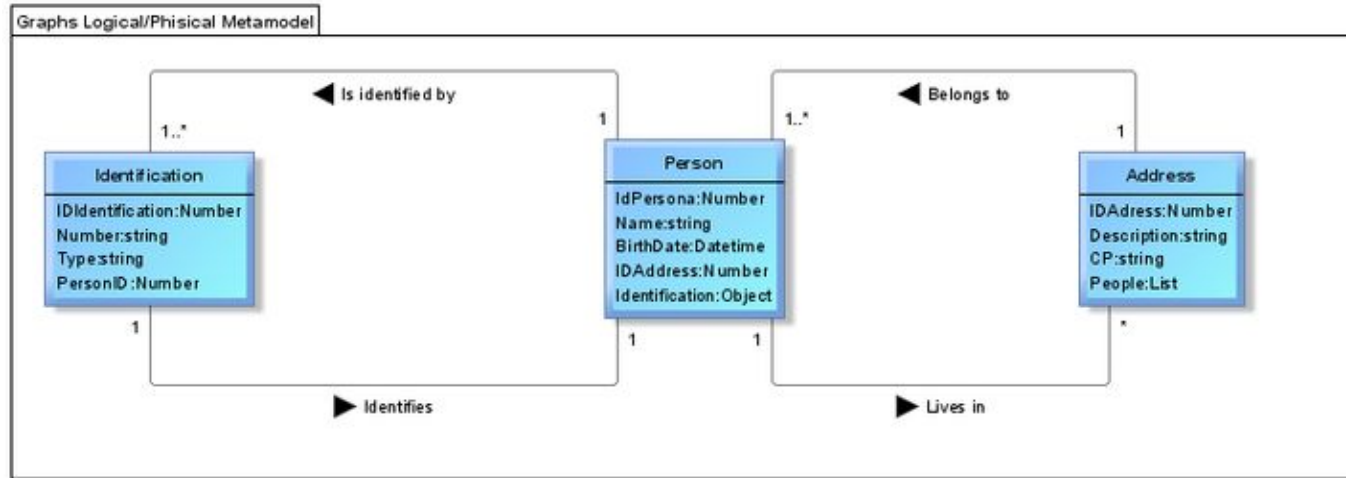
Se describe el meta-modelo lógico para base de datos documental. Para cada concepto identificado en el meta-modelo conceptual, se describen las propiedades susceptibles a ser utilizadas, sus tipos (cadenas, numéricos, etc), la cardinalidad entre clases. Las relaciones entre conceptos, dependen de las que se permita dentro de un documento NoSQL..



Si se trata de una base de datos de grafos, al igual en el el diagrama documental, describimos las posibles propiedades y sus tipos, en el meta-modelo lógico se respetan las relaciones de ida y vuelta del meta -modelo conceptual.

Modelos de datos NoSQL - Características

Meta-modelo Lógico.



Se define un meta-modelo lógico y no como modelo lógico, porque en lugar de describir un futuro modelo lógico, se describe un patrón de como debe crecer y evolucionar una base de datos NoSQL. Es decir, **sirve de guía para saber como relacionar un nuevo concepto, agregar correctamente una propiedad a un concepto adecuado y tener una referencia del tipo que debería tener**. Por otro lado, nos ayuda a medir el impacto, cuando es necesario evolucionar el esquema.

Modelos de datos NoSQL - Características

Meta-modelo físico.

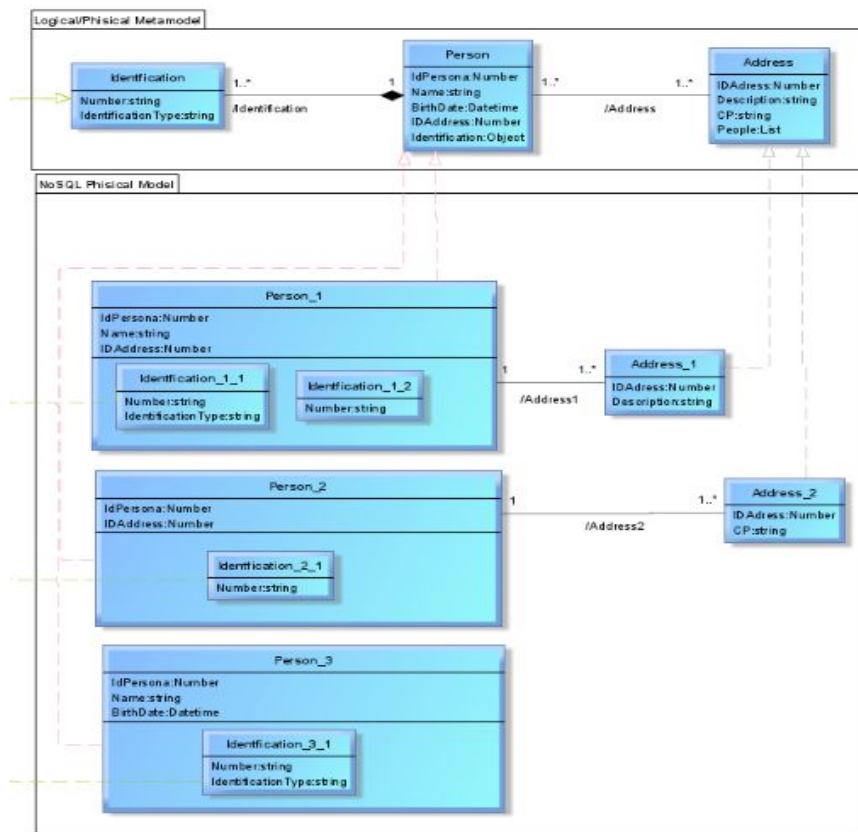
El meta-modelo físico puede llegar a ser similar o el mismo que el modelo lógico, dependerá de si la notación de los campos y tipos de datos corresponden físicamente con algún motor de base de datos específico. El propósito del meta-modelo físico, es el mismo que el del meta-modelo lógico, es decir, describir un patrón de cómo debe crecer la base de datos. La utilización de un solo meta-modelo lógico/físico o separarlos en dos distintos, dependerá de las necesidades de visualización que tengan, para el ejemplo el meta-modelo es el mismo.

El modelo físico en cambio, tanto en la base de datos documental, como en la base de datos de grafos, heredarán los conceptos definidos en el meta-modelo. Este modelo puede ayudar a verificar el meta-modelo lógico físico y prever reglas de validación de datos.

Para el modelo documental, se describen las clases que representan los documentos, se pueden mostrar los conceptos anidados con clases anidadas. Los campos pueden variar entre clase y clase, pero siempre se ajustan al meta-modelo físico. Una regla de ejemplo, puede ser que por cada *Identification que exista*, ésta debe tener al menos una propiedad *Number* aunque no se pueda determinar el *IdentificationType*.

Modelos de datos NoSQL - Características

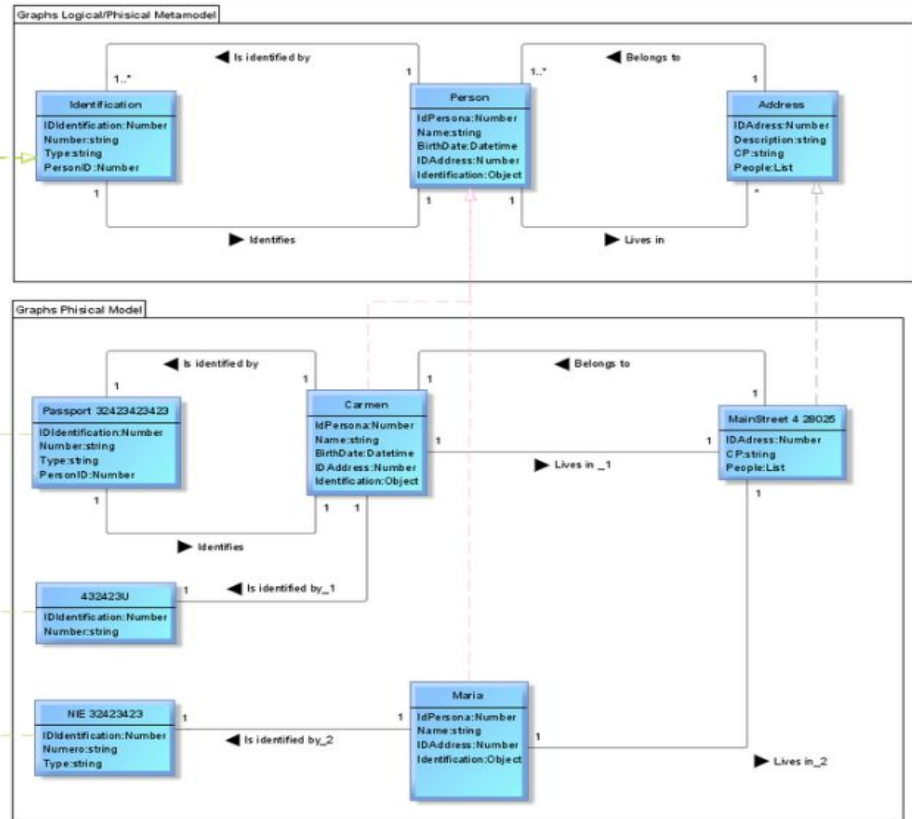
Meta-modelo físico.



Modelos de datos NoSQL - Características

Meta-modelo físico.

En el caso del modelo de grafos, cada clase representa un nodo del grafo y cada nodo heredaría un concepto del meta-modelo de físico y respeta sus relaciones. Las propiedades de las clases nodo, pueden contener una, o varias, o todas las de las clases del meta-modelo físico. El siguiente modelo muestra lo descrito anteriormente, el ejemplo del grafo producido por el meta-modelo. Si se utiliza un diagrama de objetos de UML(UML object diagram) los objetos del diagrama son instancias de las clases del meta-modelo físico.

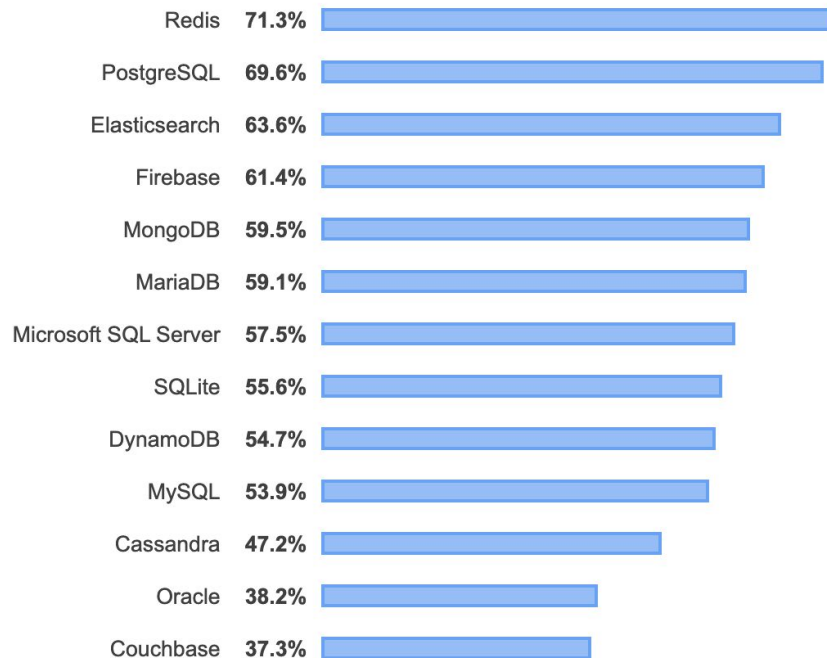


ESTADÍSTICAS

Loved

Dreaded

Wanted

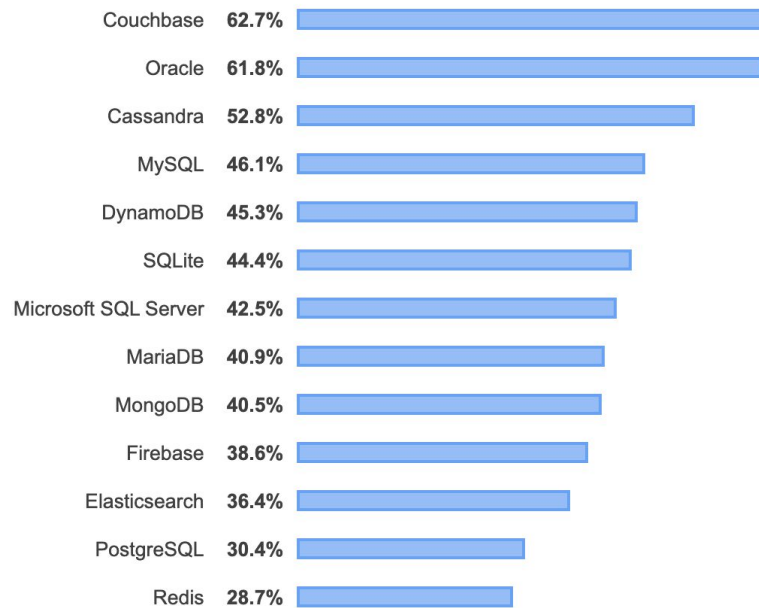


ESTADÍSTICAS

Loved

Dreaded

Wanted

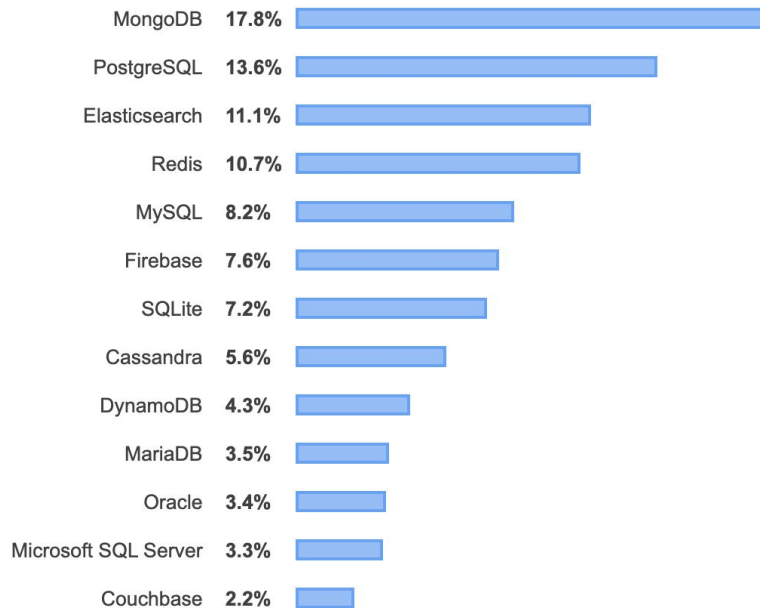


ESTADÍSTICAS

Loved

Dreaded

Wanted



TECNOLOGÍA

Lookup

Enter a URL



Price per lookup: 1 credit. Get 50 credits per month on a free plan.

6 TECHNOLOGIES IDENTIFIED



[RxJS](#)



[HTTP/2](#)



[Facebook](#)



[PHP](#)



[React](#)



[Google Analytics](#)

CONCLUSIÓN

CARACTERÍSTICAS	NOSQL	RDBMS
RENDIMIENTO	ALTO	BAJO
FIABILIDAD	POBRE	BUENO
DISPONIBILIDAD	BUENO	BUENO
CONSISTENCIA	POBRE	BUENO
ALMACENAMIENTO	GRANDES CANT. DATOS	MEDIANAS Y LARGAS
ESCALABILIDAD	ALTO	ALTO (MAS CARO)