



# Bases de datos orientadas a grafos

## Graph-Oriented Databases

Claudia Pinilla<sup>1</sup> Mauricio Bello<sup>2</sup> Cristian Peña<sup>3</sup>

**Para citar este artículo:** Pinilla, C.; Bello, M.; Peña, C. (2017). Bases de datos orientadas a grafos. *TIA*, 5(2), pp. 153-160.

### Resumen

Este artículo presenta una breve descripción de lo que significan las bases de datos orientadas a grafos, su modelamiento, una comparación con la base de datos relacional, sus usos en algunos contextos actuales y una propuesta de su aplicación en Scrum.

**Palabras clave:** bases de datos orientadas a grafos, nodos, NoSQL, Scrum, relaciones.

### Abstract

This article presents a brief description of what graph-oriented databases mean, their modeling, a comparison with the relational database, their uses in some current contexts, and a proposal for their application in Scrum.

**Keywords:** graphs database, nodes, NoSQL, Scrum, relationships.

### ARTÍCULO DE INVESTIGACIÓN

Fecha de recepción:  
31-05-2015

Fecha de aceptación:  
11-07-2017

ISSN: 2344-8288

Vol. 5 No. 2

Julio - diciembre 2017

Bogotá-Colombia

## INTRODUCCIÓN

Las bases de datos relacionales tienen una amplia aceptación en el mundo del desarrollo de *software* y han demostrado su efectividad en los distintos contextos en los que se ha aplicado a la hora de realizar el almacenamiento de información; sin embargo, los tiempos han cambiado y la información ha evolucionado a tal punto que los sistemas de información se tienen que encontrar con la manipulación de grandes cantidades de datos, lo cual ha resultado difícil y se han implementado otras estrategias para manejar la información.

Una de esas estrategias es un concepto alternativo para la gestión de bases de datos conocido como NoSql, el cual no solo utilizan el lenguaje SQL, este es un sistema de gestión enfocado a solucionar los problemas de escalabilidad y rendimiento de las bases de datos relacionales, en contextos donde el volumen de datos es alto y los sistemas sean bastante concurridos por los usuarios.

Dependiendo de la forma en qué se almacena la información en un sistema de gestión NoSql, se pueden encontrar varios tipos diferentes de bases de datos, uno de ellos son las bases de datos orientadas a grafos.

## BASE DE DATOS ORIENTADA A GRAFOS (BDOG)

Antes de iniciar la definición formal de una base de datos orientada a grafos, es necesario refrescar el concepto de grafo, este básicamente es un conjunto de puntos (vértices) en el espacio que están conectados por un conjunto de líneas (aristas).

Los grafos, dentro del contexto de *software*, representan las entidades como nodos y las aristas por la forma como estos nodos se pueden encontrar relacionados, ya sean entre sí o no.

## Definición

Una base de datos orientada a grafos es aquella que permite almacenar la información como nodos de un grafo y sus respectivas relaciones con otros nodos, permitiendo así aplicar la teoría de grafos para recorrer la base de datos; son muy útiles para guardar información en modelos con muchas relaciones como redes y conexiones sociales. Cada nodo consta de un grado que indica el número de aristas que tiene, a su vez un grafo puede ser dirigido o no dirigido, dependiendo de si las aristas tienen nodos origen y nodos destino.

El uso de este tipo de bases de datos depende altamente de la lógica de negocio donde se encuentre involucrada la información a almacenar, ya que no puede aplicar en todos los escenarios, o tal vez no se podría aprovechar su potencial en unos u otros contextos.

## Modelamiento

El modelamiento consiste en tratar de trasladar la mayoría de elementos claves contenidos en una realidad a un espacio concreto que lo quiere describir, aproximándose a dicha realidad en un ámbito específico y limitado.

“En general un modelo de datos es una colección de herramientas conceptuales usadas para modelar representaciones de entidades del mundo real y las relaciones de estas” [1].

El modelamiento de datos mediante grafos es un proceso que se centra en describir un dominio arbitrario como un grafo conectado de nodos y relaciones, las cuales permitan fácilmente expresar semánticamente las preguntas que se quieren resolver [2].

Las unidades fundamentales que forman un grafo son los nodos y las relaciones, ambos pueden contener propiedades; los nodos son frecuentemente usados para representar entidades y, dependiendo del dominio las relaciones, pueden usarse para cualquier propósito.

En este artículo se trata del modelo para grafos basado en nodos y relaciones con etiquetas y con atributos. Considere la siguiente descripción entre dos personas: “María y Juan son amigos. Ambos han leído un libro, *Cien años de soledad*”.

De la lectura de esta descripción, fácilmente se pueden identificar nodos y relaciones; las primeras entidades que se identifican en un dominio son sus nodos, en el caso María, Juan y el libro *Cien años de soledad* (Figura 1).

Estos nodos se pueden agrupar en tipos de datos, para esto el nodo puede contener una propiedad especial que identifique ese tipo como puede ser un atributo o una etiqueta, esto permite a los motores de bases de datos orientadas a grafos indexar los nodos y optimizar las consultas. Cada nodo puede tener cero o muchas de estas etiquetas, permitiendo que ellos pertenezcan a varios tipos de datos. Siguiendo con el ejemplo los nodos María y Juan son nodos de tipo o plantilla de datos “persona” y *Cien años de soledad* es un nodo de tipo “libro” (Figura 2).

Para las relaciones se deben identificar las interacciones que ocurren entre los sujetos, retomando el ejemplo, María es amiga de Juan, Juan es amiga de María, Juan ha leído *Cien años de soledad*,

María ha leído *Cien años de soledad*. Las relaciones identificadas son “es amiga de” y “ha leído”.

Después de obtenidas las interacciones entre nodos, se pueden identificar qué tipos de nodos están unidos en una relación; en el ejemplo, un par de nodos de tipo persona están conectadas por la relación “es amiga” y un nodo persona con un nodo libro están conectado mediante la relación “ha leído” (Figura 3).

Ahora que se han identificado las relaciones y los tipos de nodos se puede representar mediante un grafo (Figura 4).

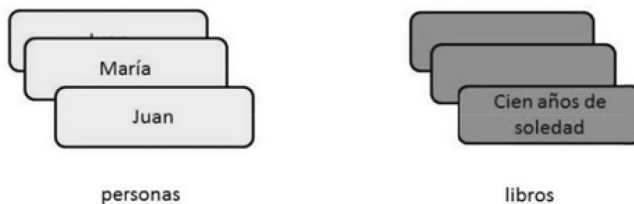
Después de obtenido los elementos nodos y relaciones, se procede a obtener sus propiedades; mediante preguntas y respuestas que quieran resolver un problema, se pueden llegar a ellas.

Supóngase que en el ejemplo se hacen las siguientes preguntas: ¿desde hace cuánto son amigos María y Juan?, ¿quién es el autor del libro *Cien años de soledad*?, ¿cuántos años tiene María?, ¿quien leyó el libro, María o Juan?, ¿cuándo terminó de leer María el libro?, ¿qué calificación le dio al libro Juan?, ¿quién es mayor en edad, María o Juan? Al resolver esas preguntas se pueden obtener atributos que se pueden tener en los nodos y relaciones, por ejemplo, de la última pregunta se puede ver “edad” como atributo de “persona”.



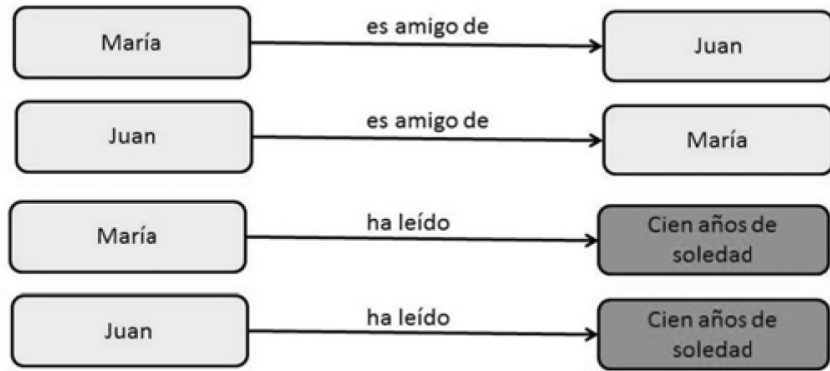
**Figura 1.** Identificación de nodos

**Fuente:** elaboración propia.



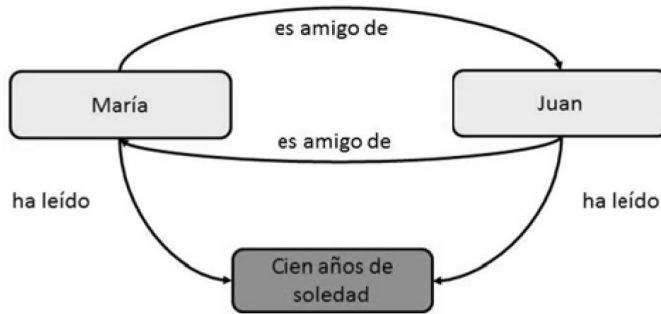
**Figura 2.** Identificación de agrupaciones o tipos de nodos

**Fuente:** elaboración propia.



**Figura 3.** Identificación de relaciones

Fuente: elaboración propia.



**Figura 4.** Modelo de datos mediante grafos

Fuente: elaboración propia.

Las propiedades en los nodos se asocian con los atributos que representan a una entidad, y en las relaciones representan atributos asociados con las cualidades que ellas tienen, como la calidad, el peso o fuerza que son particulares en cada contexto.

### Comparación modelo relacional

El modelo relacional es ampliamente conocido, se puede comprender como un conjunto de entidades que se relacionan con otras. En el proceso de modelamiento se trasladan a tablas, por lo que en práctica se diseñan tablas con columnas para representar a cada entidad y a cada relación, para esto el experto debe seguir procesos de normalización y otros procedimientos para representar el modelo de negocio teniendo

presente los lineamientos y restricciones propias de las bases de datos relacionales.

Para una simple relación entre dos entidades, se debe modelar una tabla con los atributos que la identifican a cada entidad, y según sea el tipo de relación, pueden surgir una o varias tablas que representan la relación. Aunque la generación de las tablas para las relaciones es el resultado de aplicación de las reglas de normalización, constituyen un elemento para tener en cuenta en el modelamiento.

En el modelo de estructura de grafos se tienen dos tipos de elementos: los nodos y las relaciones, cada uno de ellos contienen sus propiedades tipo clave valor; asimismo las relaciones agregan una característica y es que pueden ser o no dirigidos.

En grafos cada elemento, entiéndase nodo y relación, tiene su espacio propio y no

necesariamente está limitado a representar un solo tipo de dato, pues en casos específicos, un nodo puede representar diferentes tipos de datos disponibles según el propósito con que se modelen; también un elemento o nodo que represente un tipo de dato es identificable además por las relaciones que tiene con otros nodos. La principal diferencia con las bases de datos relacionales es su poder de consulta y la libertad en el esquema de datos.

Las consultas en bases de datos relacionales normalizadas utilizando SQL (lenguaje de consulta estructurado) se realizan empleando la sentencia JOIN entre tablas, la cual une las diferentes tablas normalizadas para extraer los datos que se requieren. La dificultad se presenta si se desea obtener información cuando las entidades son altamente relacionadas con otras, pues implica construir consultas complejas difíciles de mantener y entender, que resultan muchas veces en la sobreutilización de recursos en los motores de bases de datos relacionales cuando se ejecutan dichas consultas. Para solucionar esto, los diseñadores de bases de datos deben encargarse de optimizar sus modelos y se ven obligados a adaptarlos a esta limitante, por eso los modelos de bases de datos muchas veces no representan claramente y de manera natural la lógica de negocio.

En su lugar, las bases de datos orientadas a grafos prometen un mayor rendimiento y facilidad en el momento de realizar consultas respecto a las realizadas en bases de datos relacionales, debido a que cuentan con un lenguaje de consulta similar a las expresiones semánticas que parecen naturales y que permiten atravesar las conexiones entre entidades altamente relacionados. Esto ayuda a evitar la creación de elementos diferentes a las entidades reales y sus relaciones identificables de forma natural.

La otra dificultad mayor asociada a lo anterior, y que se suma en la práctica de la gestión de bases de datos relacionales, es la necesidad frecuente de modificar los modelos de datos, para involucrar cambios de la lógica de negocio.

En las BDOG los datos no se adhieren estrictamente a un tipo de dato, es decir nodos y relaciones similares y que pueden ser identificados como un tipo de datos por que comparten un conjunto de atributos, también pueden tener atributos propios y únicos respecto a otros nodos similares. Esta es una característica heredada de NOSQL, la representación lógica de un esquema de datos no obliga a la BDOG a seguirla rígidamente, permitiendo la extensibilidad y redefinición de modelos de datos.

Por otro lado, la naturaleza de las BDOG libre de esquemas, se refiere a la libertad que permite de asociar la información en las diferentes formas de representarlos en nodos, propiedades o relaciones, permitiendo realizar cambios durante la evolución del modelo de la lógica de negocio, por ejemplo, una entidad asociado con un dato que puede constituir una propiedad en una entidad, también se puede modelar como una entidad completamente independiente. Para analizar otro caso de modelo libre de esquemas, supóngase esta descripción: María “es dueña de” un carro de placa XYZ, y María “es dueña de” una casa ubicada en Bogotá D.C., entendiendo que casa y carro sean dos tipos de nodos diferentes, se obtiene que la relación “es dueña de” permite conectar a “tipos” de nodos diferentes, persona-carro, persona-casa, convirtiéndolas en relaciones flexibles.

## Fortalezas

El uso de las bases de datos orientadas a grafos puede tener ventajas en escenarios donde los sistemas a implementar requieran de una adaptación constante a los cambios de lógica de negocio, y en modelos donde existe una alta dependencia funcional entre las entidades involucradas en un sistema.

### *Rendimiento*

El rendimiento es una fortaleza clave para el uso de bases de datos con grafos; en comparación con el uso de bases de datos relacionales, donde su

rendimiento está fuertemente ligado al tamaño de los datos y las numerosas relaciones entre las entidades, implicando que el rendimiento en las consultas esté inversamente proporcional a la totalidad de los registros y relaciones envueltas entre las entidades que satisfacen la consulta, en BDOG el rendimiento tiende a permanecer relativamente constante, ya que las consultas se realizan iniciando un recorrido desde un segmento de datos del grafo, pasando por los distintos nodos y aristas necesarios para satisfacer la consulta, dando como resultante que el tiempo de ejecución es proporcional únicamente al tamaño del grafo que se tenga que recorrer para satisfacer la consulta, no el tamaño total del grafo.

En un sistema donde se presenten múltiples niveles de profundidad de los datos relacionados, se evidencia la rapidez de la ejecución de consultas en bases de datos orientadas a grafos.

## Flexibilidad

En sistemas dinámicos donde se presenten contextos necesarios de su adaptación constante, las bases de datos orientadas a grafos son totalmente aplicables debido a su propiedad de adición de nuevas aristas (relaciones), nodos y hasta subgrafos al árbol de grafos definido para un sistema, permitiendo acomodarse a las necesidades del negocio sin alterar las consultas, comportamientos de la aplicación.

## Casos de uso

El uso de una BDOG es una respuesta a la dificultad de representar los diversos sistemas complejos que caracterizan al mundo actual, y a la necesidad de tener un alto performance de un sistema que se encuentre involucrado en contextos con altos volúmenes y concurrencia de datos.

Actualmente, las bases de datos orientadas a grafos se están aplicando en los casos descritos a continuación.

- Redes sociales: las personas o grupos corresponden a nodos en una base de datos orientada a grafos,

y las formas como interactúan dichos individuos generan las distintas relaciones de los mismos permitiendo así predecir sus comportamientos.

- Recomendaciones: los algoritmos de recomendación establecen las relaciones entre individuos y los servicios a los que pueden estar sometidas las personas. Ya sea al momento de realizar una lectura de interés, la visualización de algún video, las compras que realice una persona o las diversas variedades de consumo de los individuos tienden a establecer un interés en algún tema en particular generando una conducta de la cual se pueden abstraer y almacenar múltiples relaciones para su posterior recomendación.
- Geo: las distintas operaciones geoespaciales dependen de estructuras de datos específicos, las cuales se pueden representar de una forma jerárquica; dicha representación facilita los cálculos de rutas o cualquier obtención de información entre las ubicaciones en alguna red específica tales como la de carreteras, ferroviaria o espacio aéreo. Las aplicaciones geoespaciales de las bases de datos orientadas a grafos son especialmente relevantes en las áreas de telecomunicaciones, logística, viajes, horarios y planificación de rutas.
- Controles de acceso: la autorización del uso de recursos o aplicaciones por parte de los diferentes tipos de usuarios basados en sus roles en un sistema permite que dicho flujo de información pueda ser representada mediante la utilización de grafos.

## Aplicación BDOG–Scrum

Pensar en una aplicación de las bases de datos orientadas a grafos diferentes a las normalmente conocidas vistas anteriormente no es una tarea fácil, primero, porque es cambiar un paradigma, un pensamiento que se ha inducido con el uso frecuente de las bases de datos relacionales desde las etapas de preparación universitaria hasta el ejercicio profesional de la ingeniería de sistemas; segundo, debido a que normalmente el proceso de definición y utilización de una base de datos se realiza basado



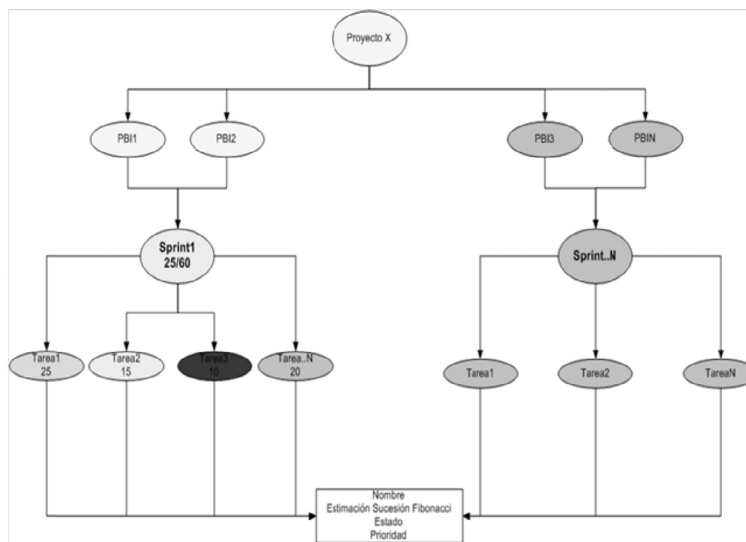
en un contexto previamente establecido y conocido, es decir, ya existe una lógica de negocio donde existe la necesidad de almacenar la información en una base de datos y el enfoque de un ingeniero es pensar en el modelado del mismo.

Para poder buscar un escenario donde se pueda aplicar el uso de BDOG, se debe identificar un contexto donde exista información altamente relacionada, la cual se pueda categorizar por ciertos niveles, un marco donde se puedan aplicar los métodos CRUD y a su vez se puedan ver reflejadas las fortalezas del uso de BDOG como su rendimiento, flexibilidad y agilidad con respecto a las bases de datos relacionales.

Bajo esta premisa, se propone un uso de BDOG aplicado en el elemento Scrum Board. Esta propuesta consiste en una alternativa diferente al seguimiento de las tareas por Sprint en un marco de trabajo Scrum, dado que los datos bajo este contexto pueden evolucionar a medida que evoluciona el negocio y se encuentran altamente relacionados; se trata de almacenar los elementos que se encuentran involucrados durante el proceso de un Sprint en un proyecto Scrum como un árbol de grafos con raíz con cuatro niveles. Los niveles que se encuentran altamente relacionados en este proceso son:

- Proyecto: identificado por una etiqueta como el nombre del sistema o producto que se está implementando.
- PBI: los diferentes ítems del *product backlog* que contienen las características funcionales a implementar del producto.
- Sprint: identificador de cada uno de los Sprints generados para cumplir con uno o varios ítems de los *product backlog*.
- Tarea: compromisos adquiridos en cada Sprint, para satisfacer una o varias características funcionales establecidas en los PBI.

El proyecto se comportaría como el nodo raíz del sistema, seguido por un nivel donde estarán representados los ítems del *product backlog* definidos previamente para todo el proyecto, comportándose como nodos hijos del nodo raíz. El siguiente nivel se relaciona con cada uno de los Sprints donde se encuentran involucrados uno o varios ítems del *product backlog* representándolos como nodos hijos de los PBIS, por último, se encuentran las tareas como un último nivel del árbol de grafos, las cuales se representan como nodos hijos de un elemento Sprint (Figura 5).



**Figura 5.** Árbol grafos proceso Sprint en Scrum

Fuente: elaboración propia.

Los distintos nodos que forman parte del árbol de grafos, tendrán una serie de propiedades como su nombre, estimación, estado, etc., los cuales facilitarán la realización del seguimiento de un Sprint en un proyecto Scrum.

## CONCLUSIONES

Las bases de datos orientadas a grafos han tenido mucho éxito en sistemas con múltiples relaciones cómo las redes o aplicaciones sociales, a su vez tienen gran adaptabilidad a modelos cambiantes.

Las BDOG facilitan la exploración de los datos gracias a su naturaleza de estructura de grafo, permitiendo hacer recorridos por caminos cortos del grafo sin necesidad de verificar la totalidad de caminos del árbol de grafos.

No es muy común el uso de base de datos orientados a grafos, ya que no cualquier negocio se puede adaptar a ello, y tampoco se debe buscar su adaptación a la fuerza porque posiblemente no se podrán explotar las fortalezas en la ejecución de consultas con alto volumen de datos.

## REFERENCIAS

- [1] Silberschatz, A., Korth, H. and Sudarshan, S. (1996). Data Models. *ACM Computing Surveys*, 28(1), 105–108.
- [2] Penchikala, S. (2014). *Data Modeling in Graph Databases: Interview with Jim Webber and Ian Robinson*. Recuperado de <http://www.infoq.com/articles/data-modeling-graph-databases>.