

Modelo NoSQL de Propuestas Académicas de Grado

Vizcaino Corsetti Fanny¹, Smail Ana², Pompei Sabrina³

Instituto de Investigación y Transferencia de Tecnología (ITT)⁴

Comisión de Investigaciones Científicas (CIC)

Escuela de Tecnología (ET)

Universidad Nacional del Noroeste de la Provincia de Buenos Aires (UNNOBA)

Sarmiento Nro 1119 3er Piso, Junín (B), Argentina

fbvizcainocorsetti@comunidad.unnoba.edu.ar ; {ana.smail,
sabrina.pompei}@itt.unnoba.edu.ar

Abstract. La sobre-duración de los estudios universitarios es una de las problemáticas actuales en la educación superior argentina que nos plantea la necesidad de identificar factores determinantes a fin de tomar decisiones en pos de disminuir la duración real de las carreras. Distintos factores provocan esta sobre-duración ya sea por causas extrínsecas a los estudiantes como intrínsecas a los mismos. La UNNOBA no es ajena a esta problemática nacional.

El presente artículo muestra un avance del trabajo realizado en pos de obtener un modelo NoSQL de grafos para las propuestas académicas de grado de la universidad. Esto implica el diseño de grafos tanto para los planes de estudio como para los trayectos académicos de los egresados. Se espera que este modelo permita a futuro mediante el análisis de los grafos resultantes buscar evidencias sobre problemáticas propias del currículo que inciden en la sobre-duración de las propuestas.

Keywords: base de datos noSQL, base de datos orientada a grafos, sobre-duración de propuestas académicas

1 Introducción

Actualmente los modelos relacionales luchan y se debaten en dominios altamente conectados frente a alternativas NoSQL, sistemas de bases de datos que a diferencia del modelo relacional no usan SQL como lenguaje de consultas. En el mundo relacional el incremento de relaciones y conectividad se traduce en incremento de “join” lo que reduce el rendimiento y la adaptabilidad en respuesta a los cambios de negocio. Frente a esto, la creciente demanda de análisis basado en relaciones ha puesto en escena a los

¹ Becario ITT

² Docente Investigador - ITT

³ Docente Investigador - ITT

⁴ ITT - Centro Asociado CIC

grafos como modelo para representar los datos y sus relaciones a fin de optimizar el descubrimiento de conocimiento, implementados en bases de datos NoSQL orientadas a grafos (BDOG).

Por definición un grafo es un conjunto de nodos y aristas. Cuando modelamos con grafos las entidades son representadas por los nodos, y el modo en que estas entidades se relacionan mediante las aristas. Esta estructura de propósito general nos permite modelar todo tipo de escenarios.

En las BDOG los datos conectados se almacenan como datos conectados, es decir, donde hay conexiones en el dominio, hay conexiones en los datos. Por esta razón el modelo de grafos es particularmente útil cuando los datos a almacenar tienen multitud de interrelaciones entre sí, y cuando la importancia recae más en estas interrelaciones que en los propios datos. Es el caso de nuestro objeto de estudio, dado que un plan de una propuesta de grado está determinado por las relaciones entre las actividades y conjuntos de actividades que lo componen, siendo este el trayecto académico óptimo de un egresado.

En el modelado de las bases de datos orientadas a grafos se reduce la brecha entre el modelado lógico y físico. Esta disonancia semántica entre nuestra conceptualización del mundo (modelo lógico) y la instanciación de la base de datos (modelo físico) de ese modelo, se reduce en el modelado de la BDOG. El hecho de no requerir una definición explícita del esquema de la base de datos y que las propiedades de los nodos no sean estrictas facilita la extensión ad hoc de la base de datos, de manera que nuevos tipos de nodos y relaciones, así como nuevas propiedades, pueden ser añadidas al grafo actual directamente, sin necesidad de redefinir el esquema de la base de datos.

Otra característica a destacar de los modelos de grafos, es el hecho de que los grafos no solo comunican cómo pensamos que las cosas están relacionadas, sino que también el tipo de preguntas que queremos hacerle a nuestro dominio. Las relaciones en los grafos naturalmente crean caminos por lo que consultar el grafo implica recorrer caminos. Esta naturaleza orientada a caminos del modelo de datos implica que la mayoría de las operaciones de la BDOG están altamente alineadas con el modo en que los datos se disponen, haciendo esto extremadamente eficiente. Esto incide además en la escalabilidad del sistema dado que el tiempo de ejecución de las consultas no depende directamente del tamaño del grafo sino de la porción del grafo recorrido.

2 Objetivos

El objetivo general de nuestro trabajo se centra en alcanzar un modelo de grafo aplicable a las propuestas de grado de la universidad lo que implica el diseño de grafos tanto para los planes de estudio como para los trayectos académicos de los egresados. Este modelo permitirá a futuro mediante el análisis de los grafos resultantes buscar evidencias sobre problemáticas propias del currículo que inciden en la sobre-duración de las propuestas.

Para ello se definen los siguientes objetivos específicos:

- Identificar las características del dominio que lo hacen idóneo para ser almacenado en una base de datos orientada a grafos.

- Realizar el proceso de diseño de una base de datos orientada a grafos, desde la descripción de los datos hasta la implementación, pasando por la creación del modelo conceptual de datos y el diseño lógico.
- Determinar las consultas posibles a realizar sobre el dominio de los datos que validan el modelo.

3 Metodología

Los planes de estudios académicos de una propuesta en la Universidad, tienen una estructura naturalmente orientada a grafos; donde cada actividad en el mismo será correlativa o tendrá por correlativa a otra. Así mismo dichos planes pasan por distintas versiones, de las cuales existe una o varias activas, y una única vigente.

Las actividades que forman parte del mismo tienen un nombre y una carga horaria semanal y total en el período electivo o módulo del que forman parte, además de pertenecer a un área específica de conocimiento.

Teniendo claro las características de los datos y relaciones en el dominio en cuestión, nos encontramos en condiciones de realizar el diseño conceptual, el cual será independiente de la tecnología elegida para crear la base de datos. Definimos además las consultas que nos permitirán a posterior validar la transformación al modelo lógico elegido.

Dada la naturaleza de la estructura de grafo del dominio, un diseño lógico de almacenamiento de los datos en una base de datos orientada a grafos parece lo más apropiado para lograr la abstracción del mismo.

Una vez que se optado por un modelo lógico basado en grafos, el problema es: ¿cómo modelamos el grafo?

Podemos elegir entre tres tipos de modelos de diseño lógico de una base de datos orientada a grafos:

- Modelo grafo de propiedades: formado por nodos que pueden tener aristas de entrada, salida, propiedades y etiquetas; relaciones que tienen etiqueta, son dirigidas (un nodo de origen y uno de destino) y también pueden tener propiedades. Las propiedades son pares clave-valor.
- Modelo de hipergrafos: son grafos en el que una relación (denominada hiperarco) puede conectar cualquier número de nodos dados, y pueden ser útiles cuando sus datos incluyen un gran número de relaciones de muchos a muchos.
- Modelo de tripletas: almacenan los datos en un formato conocido como triple. Los triples consisten en una estructura de datos sujeto-predicado-objeto.

En nuestro caso, hemos elegido el modelo de grafo de propiedades, por lo que para transformar el modelo conceptual a este tipo de grafo, representamos los objetos de las entidades como nodos, y etiquetamos estos nodos con los nombres de dichas entidades.

Después de obtener el grafo inicial que se adapta a nuestro modelo conceptual, procedimos a ajustar el modelo lógico añadiendo relaciones y nodos, teniendo en cuenta que este debería responder a las consultas planteadas en el dominio. Posteriormente se

procedió al diseño físico que consiste en crear el grafo planteado en un Sistema de Gestión de BDOG, en este caso, Neo4j.

Neo4j es un Sistema de Gestión de Base de Datos (DBMS) multiplataforma, puramente orientado a grafos. Es nativo en almacenamiento y procesamiento de grafos. Actualmente es el sistema más expandido y mejor valorado por la comunidad de usuarios. EBay, Wal-Mart, Cisco y Hewlett-Packard, entre otras, han utilizado las cualidades de Neo4j para mejorar sus servicios.

El lenguaje declarativo con el que Neo4j trabaja es Cypher. Es potente y fácil de usar. Está basado en SQL y proporciona una forma visual y lógica para emplear nodos y relaciones. Muestra los resultados usando ASCII-Art. Algunos puntos importantes de su sintaxis:

- Cláusula create: para crear nodos y relaciones.
- Propiedades: se indica el conjunto, y sus valores en formato JSON: {clave: valor}.
- Etiquetas: clasifican los nodos dentro de la base de datos y en las consultas. Su sintaxis: (:Etiqueta)
- Cláusula match: busca en la base de datos uno o varios nodos, usando o no relaciones. Un ejemplo de su utilización:
- MATCH (nodo) – [:RELACION] -> (listaDeNodos)
- Cláusula return: retorna, mediante el uso de variables, el gráfico que contiene los nodos que éstas indican. Si dichos nodos tienen relaciones entre ellos, también se grafican.

4 Resultados

En una primera etapa con la información extraída de la base de datos de la universidad, se diseñó un modelo conceptual orientado a la implementación de los Planes de Estudio de Grado, recorrido óptimo de un egresado. En la figura 1 podemos ver el grafo correspondiente a este modelo.

Definimos además las consultas que nos permitirán a posterior validar la transformación al modelo lógico elegido:

- El Plan de Estudio de una propuesta académica
- Actividades correlativas de una actividad en un Plan de Estudio
- Actividades optativas de una actividad genérica de un Plan de Estudio
- Actividades en un período lectivo en un Plan de Estudio

Basados en este modelo y en las opciones expresadas y analizadas, se tomó como mejor opción el diseño de un modelo lógico orientado a un grafo de propiedades.

Para su realización se debe tener en cuenta que la versión del plan de estudios no es conocida por todo el grafo, sino que debe especificarse en cada relación para que su recorrido (desde el nodo que indica el plan hacia sus actividades) pueda llevarse a cabo. En una base de datos con múltiples planes, módulos y actividades, se debe discriminar cada recorrido para que no haya interferencias entre ellos. Para eso se crean las propiedades *Plan_versión* (Figura 2)

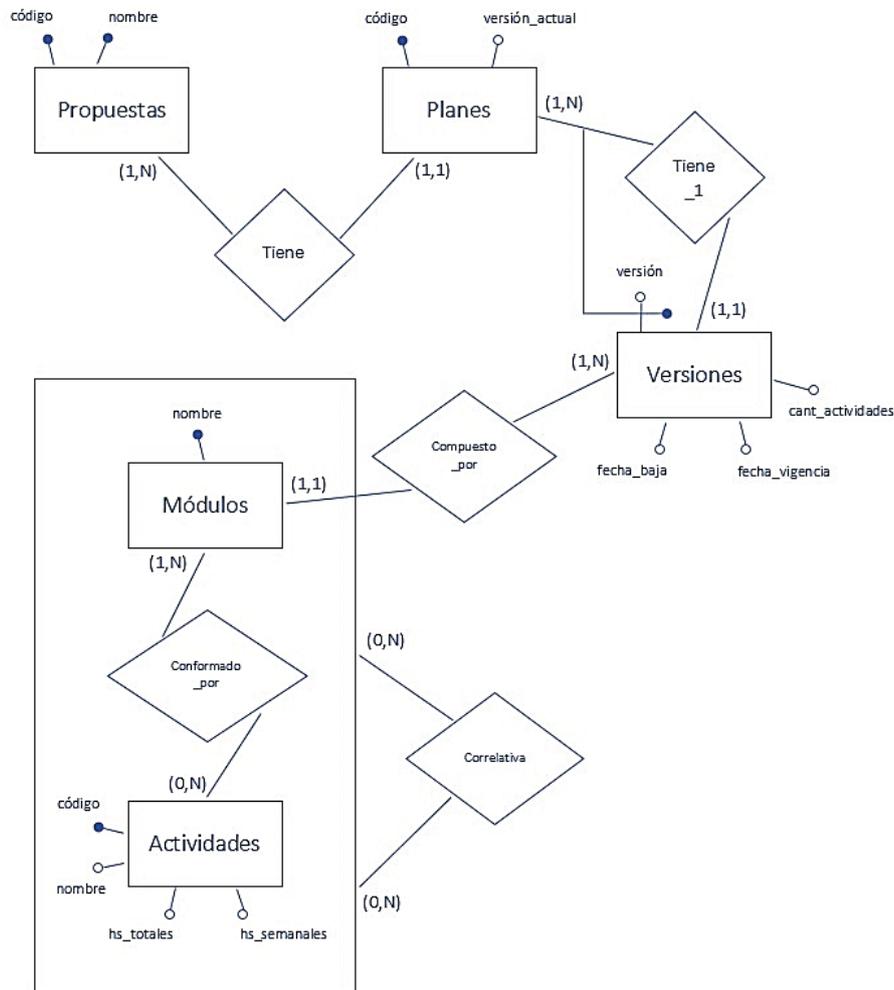


Fig. 1. Modelo Conceptual

Entre actividades se deben tener en cuenta dos tipos de correlatividades: de regularidad de cursada, y de aprobación de examen. Cada una de ellas tiene importancia en distintas etapas de la aprobación de una actividad.

Los planes de estudio cuentan con actividades genéricas: una lista de actividades optativas donde se eligen, cursan y aprueban una o varias. Se diseñó con su propia etiqueta y relaciones optativas (Figura 3).

Además de las relaciones de otros nodos con módulos, existen relaciones de correlatividad entre ellos, como por ejemplo la correlatividad n-4, donde para poder cursar un módulo cuatrimestral n, se debe tener aprobado el n-4 (Figura 4).

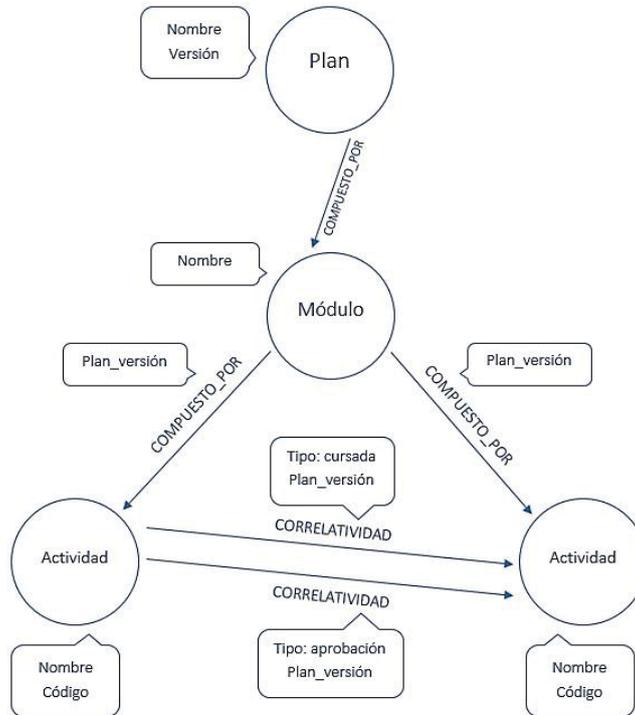


Fig. 2. Grafo de propiedades del Plan de Estudio.

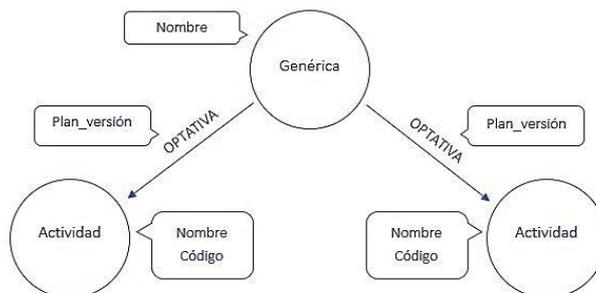


Fig. 3. Grafo de propiedades para Actividad Genérica.



Fig. 4. Grafo de propiedades del Módulo n-4

Una vez logrado este modelo lógico, se utiliza para la realización del modelo físico en la base de datos orientada a grafos elegida: Neo4j.

En su lenguaje, Cypher, no se declara la estructura y en base a ella los nodos, sino que los mismos nodos son los que dan la estructura. Es decir, para poder acceder a una generalización de la base de datos, se deben crear todos los nodos con las mismas etiquetas y propiedades.

Utilizando la sentencia Create, creamos todos los nodos y relaciones con sus propiedades. Un ejemplo reducido:

```
CREATE (lic:Plan {nombre:'Licenciatura en Sistemas',
version:415})
CREATE (cuatri6:Modulo {nombre:'Sexto Cuatrimestre'})
CREATE (cuatri10:Modulo {nombre:'Decimo Cuatrimestre'})
CREATE (bdd1:Actividad {nombre:'Base de Datos I',
codigo:867})
CREATE (bdd2:Actividad {nombre:'Base de Datos II',
codigo:387})
CREATE (cuatri6) - [:CORRELATIVIDAD {plan_version:415,
tipo:'n-4'}] -> (cuatri10), (cuatri6) - [:COMPUESTO_POR
{plan_version:415}] -> (bdd2), (bdd1) -
[:CORRELATIVIDAD {plan_version:415, tipo:' cursada'}] ->
(bdd2), (bdd1) - [:CORRELATIVIDAD {plan_version:415,
tipo:'aprobacion'}] -> (bdd2)
CREATE (optativa2:Generica {nombre:'Optativa II'})
CREATE (pweb:Actividad {nombre:'Programacion Web',
codigo:882})
CREATE (cuatri10) - [:COMPUESTO_POR {plan_version:415}]
-> (optativa2), (bdd2) - [:CORRELATIVIDAD
{plan_version:415, tipo:'aprobacion'}] -> (pweb),
(optativa2) - [:OPTATIVA {plan_version:415}] -> (pweb)
```

Luego, el modelo obtenido debería responder a las consultas iniciales planteadas a este grafo para su validación.

A continuación, presentamos algunos ejemplos detallando el código correspondiente a la consulta realizada y el grafo resultante como respuesta a la misma.

- El plan de estudio de una propuesta académica. Ejemplo: consulta sobre módulos y actividades correspondiente al Plan de Estudio versión 415

```
match (m:Modulo) - [:COMPUESTO_POR {plan_version:415}]
-> (mat) return m, mat
```

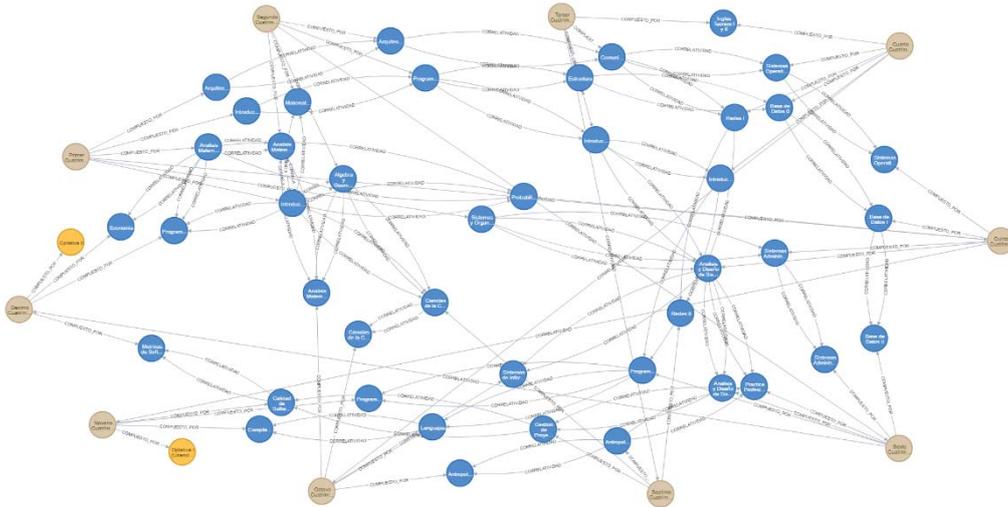


Fig. 5. Módulos y actividades del plan de estudio versión 415

- Actividades correlativas de una actividad en un plan de estudio. Ejemplo: consultamos las actividades correlativas a Análisis Matemático Básico cuyo código de actividad es 667.

```
match (amb:Actividad {codigo:667}) - [:CORRELATIVIDAD
{plan_version:415}] -> (mat:Actividad) return amb, mat
```



Fig. 6. Actividades correlativas de Análisis Matemático Básico.

- Actividades optativas de una genérica en un plan de estudio. Ejemplo: Actividades Optativas del plan 415 asociadas a la Actividad Genérica denominada 'Optativa I'

```
match (g:Generica {nombre:'Optativa I (Licenciatura)'})
- [:OPTATIVA {plan_version:415}] -> (mat:Actividad)
return g, mat
```



Fig. 7. Actividades Optativas de Genérica 'Optativa I'

- Actividades en un periodo lectivo en un plan de estudio. Ejemplo: actividades del Plan de Estudios versión 415, correspondientes al módulo Primer Cuatrimestre.

```
match (m:Modulo {nombre:'Primer Cuatrimestre'}) -
[:COMPUESTO_POR {plan_version:415}] -> (mat)
return m, mat
```



Fig. 8. Actividades del módulo Primer Cuatrimestre

5 Conclusiones y Trabajo Futuro

El modelo orientado a grafos alcanzado hasta el momento realza la importancia de los datos y principalmente, de las conexiones entre ellos.

Las consultas estudiadas son simples en comparación con el modelo relacional donde las 5 (cinco) entidades del modelo conceptual se convierten en tablas y para alcanzar los mismos resultados se requiere el join de todas estas. Esto evidencia un incremento en el rendimiento y la adaptabilidad del modelo en respuesta a los cambios del dominio.

Los datos de los planes de estudio utilizados son confiables ya que los mismos fueron extraídos directamente desde la base de datos de la universidad. La integridad del modelo en su primera etapa fue validada con las consultas planteadas al mismo.

El modelo planteado sobre los Planes de Estudio de las Propuestas de Grado de la Universidad corresponde al trayecto académico óptimo de los egresados. Actualmente se está trabajando en la implementación de los trayectos académicos propios de cada egresado, lo que permitirá a futuro, mediante técnicas de análisis de grafos buscar evidencias sobre problemáticas propias del currículo que inciden en la sobre-duración de las propuestas.

6 Referencias

1. Robinson, I., Webber, J. and Eifrem, E.: Graph Databases. Sebastopol, USA: O'Reilly (2015)
2. Angles R., Gutierrez C.: An Introduction to Graph Data Management. In: Fletcher G., Hidders J., Larriba-Pey J. (eds) Graph Data Management. Data-Centric Systems and Applications. Springer, Cham (2018)
3. Perez Sola C.: Diseño de una base de datos para analizar la actividad de usuarios en twitter. UOC
4. Pinilla, C.; Bello, M.; Peña, C.: Bases de datos orientadas a grafos. *TIA*, 5(2), pp. 153-160, . (2017).
5. Moneris, L.: Bases de datos orientadas a grafos aplicadas al estudio de informes radiológicos: utilizando entornos de computación en la nube para abordar estudios de gran dimensión. Tesis maestría, Valencia (2015)
6. Neo4j, <https://neo4j.com/graphacademy/>
7. Reconocimiento académico: pensar la formación con foco en el estudiante. Ministerio de Educación, Cultura, Ciencia y Tecnología (2018) <https://www.argentina.gob.ar/noticias/reconocimiento-academico-pensar-la-formacion-confoco-en-el-estudiante>
8. Habersfeld, L., Marquina M. y Morresi, S.: El Sistema universitario argentino. Situación, problemas y políticas. Centro de Estudios para el cambio estructural (2018) <http://fcece.org.ar/el-sistema-universitario-argentino/>