Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Bases de Datos Orientadas a Grafos en la integración de datos para fines estadísticos, el caso del Sistema de Datos Integrados del ISTAC

Graph-Oriented Databases in the integration of data for statistical purposes, the case of the Integrated Data System of ISTAC

Jaime Rodríguez Pérez

La Laguna, 21 de noviembre de 2018

Dña. **Isabel Sánchez Berriel**, con N.I.F. 42.885.838-S profesora Contratada Doctora tipo 1 adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

Dña. **Luz Marina Moreno de Antonio**, con N.I.F. 45.457.492-Q profesora Contratada Doctora tipo 1 adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como co-tutora

C E R T I F I C A N

Que la presente memoria titulada:

"Bases de Datos Orientadas a Grafos en la integración de datos para fines estadísticos, el caso del Sistema de Datos Integrados del ISTAC"

ha sido realizada bajo su dirección por D.Jaime

Rodríguez Pérez, con N.I.F.78646909-L.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 21 de noviembre de 2018

Agradecimientos

XXX

XXX

XXX

XXX

Licencia

.

* Si quiere permitir que se compartan las adaptaciones de tu obra y quieres permitir usos comerciales de tu obra (licencia de Cultura Libre) indica:

Resumen

El objetivo de este Trabajo de Fin de Grado es el estudio de la viabilidad y beneficios del uso de un modelo de datos orientado a grafos para el Sistema de Datos Integrados (*iDatos*) utilizado en el banco de datos del ISTAC. Se analizan los registros y relaciones existentes entre fuentes administrativas así como otras posibles externas. Se pretende obtener el modelo de datos basado en grafos y la implementación del mismo en una prueba de concepto con un conjunto de datos reales o simulados en los casos en que estos no sean publicables. También se estudia la posibilidad de obtener relaciones que no están explícitas en las tablas en el sistema *iDatos* del ISTAC, pero que se puedan inferir gracias al uso de la base de datos orientada a grafos. En concreto, la construcción de esta BDD se hará sobre Neo4j, producto open-source implementado en java y con una amplia comunidad. Por último se pretende comparar el rendimiento del modelo orientado a grafos frente al modelo SQL ya existente, implementando una base de datos en la que se almacena el mismo conjunto de datos de las pruebas pero en un esquema relacional.

Palabras clave: Gestión de datos maestros, ISTAC, estadística pública, Bases de Datos Orientadas a Grafos

Abstract

The objective of this Final Degree Project is to study the feasibility and benefits of using a graph-oriented data model for the Integrated Data System (iDatos) used in the ISTAC database. The records and existing relationships between administrative sources as well as other possible external sources are analyzed. It is intended to obtain the data model based on graphs and its implementation in a proof of concept with a set of real or simulated data in cases where these are not publishable. The possibility of obtaining relationships that are not explicit in the tables in the ISTAC iDatos system, but that can be inferred thanks to the use of the graph-oriented database, is also studied. Specifically, the construction of this BDD will be done on Neo4j, an open-source product implemented in Java and with a large community. Finally, it is intended to compare the performance of the graph-oriented model against the existing SQL model, implementing a BDD in which the same test data set is stored but in a relational schema.

Keywords: Master Data Management, ISTAC,

Índice de figuras

Capítulo 1 Introducción

El Instituto Canario de Estadística (ISTAC), es el órgano central del sistema estadístico autonómico y centro oficial de investigación del Gobierno de Canarias, cuyas funciones principales son: proveer información estadística y coordinar la actividad estadística pública. Entre sus directrices está especificado que se constituirá un banco de datos administrativos para fines estadísticos provenientes fundamentalmente de los ficheros administrativos de la Comunidad Autónoma de Canarias.

En la actualidad, en el marco del Plan Estadístico 2018-2022 se pretende impulsar el Sistema de Datos Integrados (*iDatos*) con el fin de producir estadísticas *multifuente* apoyándose en una gestión eficiente de datos maestros compartidos en múltiples registros, de forma que faciliten el enlazamiento de los diferentes orígenes de datos. Dentro de este plan se han marcado objetivos que potencien tanto el uso de registros administrativos y fuentes de datos con el fin de mejorar la eficacia, disminuir los costes, reducir progresivamente la carga de encuestas a los usuarios y aumentar la oportunidad del dato, disponiendo en muchas ocasiones de indicadores en un tiempo menor. El gran volumen de datos manejado y su continuo crecimiento exige el uso de tecnologías Big Data

que garanticen la eficacia y el rendimiento de la solución que se proponga.

En este contexto encontramos que a una misma unidad de análisis le corresponderá datos que están dispersos en distintas fuentes ya sean administrativas u otras fuentes complementarias Big Data. El uso de datos maestros permite determinar un único elemento de referencia y facilita la construcción de las tablas que registran las relaciones entre los diferentes registros. El conjunto de datos maestros que se contemplan son: direcciones, edificios, viviendas y locales, población y hogares, y por último empresas y establecimientos. La solución actual almacena las relaciones en una base de datos relacional, sin embargo, en los últimos años han proliferado en diferentes contextos las bases de datos orientadas a grafos en las que las relaciones constituyen el elemento crucial en el modelo de datos. El beneficio del almacenamiento nativo de grafos viene dado por la infraestructura de distribución de los datos que se diseña y construye especialmente para tener un buen rendimiento y una alta escalabilidad en el tratamiento de los modelos de grafos, idóneos para la representación de las relaciones. Frente a las bases de datos relacionales y otras soluciones NoSQL, cuando se pretende explotar las relaciones entre datos masivos relacionados hay un aumento evidente de rendimiento. En las bases de datos relacionales el rendimiento de las consultas en el que intervienen operaciones JOIN decae a medida que el conjunto de datos crece, sin embargo, en bases de datos orientadas a grafos tiende a ser constante, ya que se limitan al subgrafo alcanzable desde el nodo.

Este Trabajo de Fin de Grado se ha ocupado del análisis del problema y diseño del esquema de una Base de Datos Orientada a Grafos que de soporte al sistema iDatos. Para obtener conclusiones se ha trabajado con un conjunto de datos de prueba que permite implementar el grafo resultante. Diseñando consultas que permiten comparar el rendimiento en una base de datos similar a la utilizada actualmente por el ISTAC.

Capítulo 2 ORGANIZACIÓN Y GESTIÓN DE LA INFORMACIÓN DEL SISTEMA

2.1 LOS ESQUEMAS-TIPO DEL ENTORNO REPOSITORIO DEL BANCO DE DATOS

El ISTAC organiza la arquitectura de su banco de datos al procesamiento supervisado de datos por lotes.El procesamiento por lotes es un método de ejecución de tareas de datos repetitivas y de gran volumen. Este método permite a los usuarios procesar datos cuando se disponga de recursos informáticos y con poca o nula interacción del usuario.

En el procesamiento por lotes, existen varios tipos de entornos, nosotros trabajaremos con el entorno repositorio. El entorno repositorio se organiza en función de los diferentes tipos de archivos que contiene, en este caso nosotros contamos con tres tipos de niveles diferentes, cartografías (representación gráfica/digital de mapas), microdatos(cantidades de datos bajas, factibles para que un ser humano pueda comprenderlos) o macrodatos(grandes

cantidades de datos, de difícil comprensión para un ser humano).

Dentro de cada nivel podemos observar diferentes tipos de datos, en las cartografías tenemos, los Raw cartography(RC) , los Support Cartography(IGS) y los Geography information reference (IGR), en los microdatos, tendríamos los Raw Data(RD), los Master Data(ID), los Statistical Data(SD), los Scientific Data(CD), y los Public Data(PD), mientras que en los macrodatos los MacroDataSet(MDS), DataSetCube(DSC) y los IndicatorsCube(DSI). El Sistema de Datos Integrados trabajará con los siguientes tipos de datos:

- Raw Cartography,
- Support Cartography,
- Geography Information,
- Raw Data
- Master Data.

2.2. ORGANIZACIÓN DE LOS MICRODATOS PARA FACILITAR LA INTEGRACIÓN

El artículo 32 de la Ley 1/1991 indica que “El banco de datos administrativos para fines estadísticos debe facilitar la fusión de los ficheros para fines estadísticos”. Para cumplir con el artículo 32 de esta ley el ISTAC organiza los datos de la siguiente forma. Organiza los microdatos dentro del Banco de Datos de la Infraestructura de datos y metadatos en diferentes tipos de tablas, que a su vez se dividen en función de si son: *datos, metadatos o relaciones.*

Las tablas tipo dentro del grupo de los *datos* serían los Datos o DAT, que almacenan microdatos, las tablas tipo Georreferencias o GEO que almacenan georreferencias, y las tablas tipo Datos Longitudinales o LON que almacenan identificadores normalizados de una unidad de observación. Dentro del grupo de los metadatos, tenemos la tabla de Diseño de registro,almacena como su nombre indica diseños de los registros del conjunto de las tablas , Registro de datos, son tablas que contienen los registro de los datos y la relacion que los une con el diseño de registro, y las Extensiones de códigos,almacenan extenciones de los codelist de las tablas de microdatos cuando se requieren . Y por ultimos en el grupo de las relaciones tenemos las Relaciones entre unidades de información o URD, que almacenan relaciones de microdatos con otros microdatos.

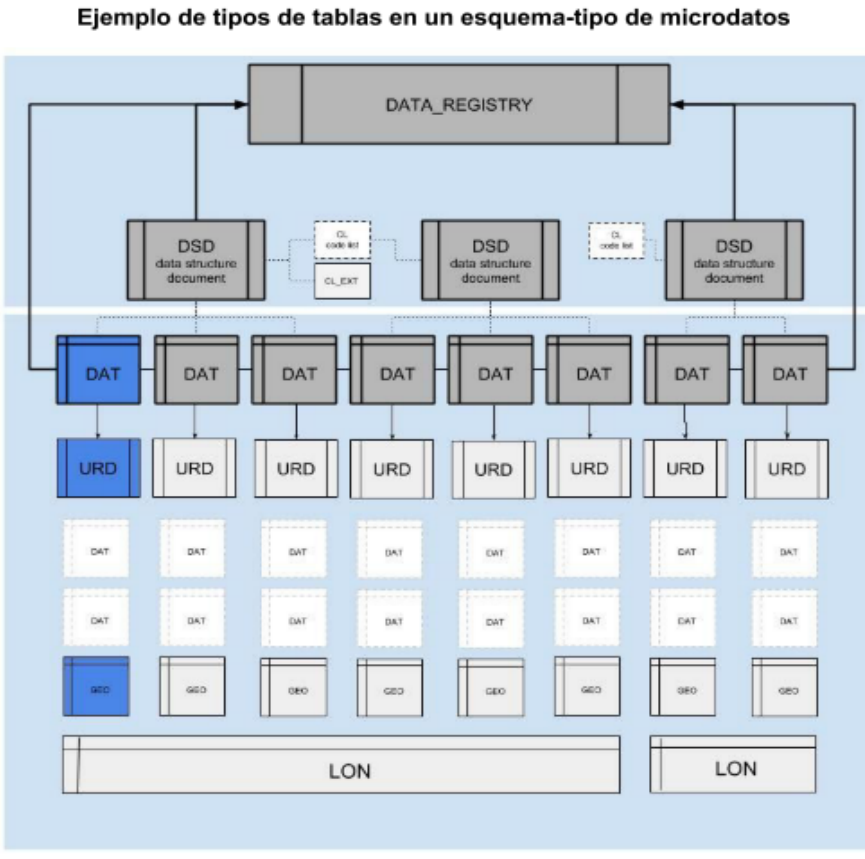


Figura 1. (Autores: Jesús Alberto González Yanes, Noelia Martín Morales, Andrés Rodríguez González,Domingo J. Lorenzo Díaz, Rafael Betancor Villalba, Esther Torres Medina.
Marco de desarrollo del Sistema de Datos Integrados de Canarias(iDatos), p.10)

Como podemos observar, el almacenamiento de las relaciones dentro de los esquemas de microdatos se realiza en tablas Relaciones entre unidades de información o URD, son tablas de relaciones, que nos permiten elaborar estadísticas multifuentes, almacenando para cada fichero de datos el conjunto de relaciones que se establezcan con otros

ficheros de datos. Estableciendo así relaciones entre las distintas tablas tipo del grupo de los datos que comentamos en el párrafo anterior.

El Banco de Datos organiza estas tablas tipo de la siguiente manera:

Para cada fila u observación, se crea un identificador único universal que llamaremos uuid, un identificador de tabla o stid, un identificador único local o luid y la fecha en la que ha sido creado, que la denominaremos marcatiempo . Por tanto estos 4 identificadores los encontraremos en todas las observaciones de las tablas tipo que existan.

En las tablas URD por tanto, se almacenan las relaciones entre dos ficheros de datos, uno origen (DAT_A) y otro destino (DAT_B). La tabla URD por lo tanto necesitará de identificadores que diferencie los ficheros de origen y los de destino, por ello precisa de:

Un UUID_A y un UUID_B, así como un STID_A y un STID_B que son tanto al identificador único universal o uuid como el identificador de esquema de tabla o stid de los correspondientes DAT_A y DAT_B.

Además, como todas las otras tablas cuenta con los identificadores únicos universales, uuid y stid, identificando así cada relación existente.

2.3 PRODUCCIÓN DE DIRECTORIOS Y ESTADÍSTICAS MULTIFUENTES

Los directorios dentro del Banco de Datos se almacenan en esquemas tipo Master Data o ID, que como vimos anteriormente se encuentran dentro del nivel de los microdatos. Dentro de cada directorio encontraremos los registros. Los registros son ficheros relacionados entre sí, los registros que nosotros usaremos serán los registro población, portales y empresas, pero existen otros como viviendas, hogares, o establecimientos.

Cada uno de estos tipos de registros tiene tres clases de tablas tipo DAT. Las tablas IDT,que se crean de una fuente básica, construyendo así una tabla única en la realización del registro en un momento(t). Las tablas IDF, se construyen de distintas fuentes(permitiendo la producción de estadísticas multifuente) y enriqueciendo a la fuente básica que era la tabla IDT, para por último con la tabla IDL conectar esas diferentes fuentes pero de la misma unidad a lo largo del tiempo.

2.4 ESTADÍSTICA DE POBLACIÓN ACTIVA REGISTRADA (EPA-Reg)

2.4.1 DIRECTORIOS DE UNIDADES ECONÓMICAS: REGISTRO EMPRESAS

El directorio de unidades económicas contiene el registro de referencia de tanto las empresas como los autónomos, en el IDT del registro empresa podemos encontrarnos las variables nucleares, siendo además variables de entidad relacionadas, UUID (identificador único universal de esas variables) y STID (identificador tanto del esquema como de la tabla), en cuanto a los ficheros IDF, obtenemos diferentes versiones de todos los registros distintos de cada empresa única.

A continuación, podemos observar el IDF de una empresa con algunas de las variables comunes de ese fichero,

UUID	STID	MARCATIEMPO	EMPRESA_PERSONA_FISICA
003f7f92-2265-4de5-9be4-b30c4b3faf45	c00021a_id.idf_empresas	20FEB2020	6
007ee1dd-ccca-4e01-9ff7-58e6a1bc7672	c00021a_id.idf_empresas	20FEB2020	6
00928afe-e133-4c52-8362-639a89fccec5	c00021a_id.idf_empresas	20FEB2020	6
00a8c26b-a0f7-4688-a2a3-0bbc84f1d8c9	c00021a_id.idf_empresas	26FEB2020	6
01375e02-b8ff-4ff7-a940-b26e54647192	c00021a_id.idf_empresas	21FEB2020	6
014b2687-3fd0-4eea-8405-e0ed464f46eb	c00021a_id.idf_empresas	26FEB2020	6
022c803c-b4ae-4d92-b08a-73d02168b7c7	c00021a_id.idf_empresas	20FEB2020	6
0259230a-e013-4aff-834e-cd3d146babac	c00021a_id.idf_empresas	20FEB2020	6
026267be-8377-41d7-bef3-cdf8a3717ff4	c00021a_id.idf_empresas	21FEB2020	6
026a6138-9bdc-4a13-9203-8065a6745355	c00021a_id.idf_empresas	20FEB2020	6
0283609f-ba71-49ac-8e24-41c2b327d736	c00021a_id.idf_empresas	20FEB2020	1
02b780f1-e3ec-4cc1-8770-8455818e2071	c00021a_id.idf_empresas	20FEB2020	1
031eefa2-b48e-4ed4-bff1-f85fb807f97d	c00021a_id.idf_empresas	26FEB2020	1

Figura 2. IDF Empresa

2.4.2 Directorio de Calles y portales: Registro Portales

El directorio de calles y portales registra puntos en el espacio que representan los portales únicos existentes. De guardar estos puntos se encarga el fichero IDT, mientras que el IDF del registro de portales guarda las diferentes versiones de cada portal único, es decir las diferentes formas de llamar a una dirección en concreto, o el cambio de nombre de ese punto a lo largo del tiempo.

CONCEPT_ID	TECH_TYPE	TECH_SIZE	LABEL	CODELIST
uuid	varchar	36	Identificador Único Universal	
luid	serial		Identificador Único Local	
stid	varchar	61	Identificador de esquema y tabla	
marcatiempo	date		Sello de tiempo de creación de la observación	
tvia_nn	varchar	13	Tipo de vía no normalizado	
nvia_nn	varchar	50	Nombre de vía no normalizado	
numer_nn	varchar	7	Número de portal no normalizado	
tvia	varchar	13	Tipo de vía	
cvia	varchar	5	Código de vía	
nvia	varchar	50	Nombre de vía	
numer	varchar	4	Número de portal	
kmt	**varchar**	**3**	**Punto kilométrico**	
nomedif	**varchar**	**50**	**Nombre del edificio**	
codmun	varchar	5	Código de municipio	ISTAC:CL_AREA_ES(02.000)
nommun	varchar	35	Nombre del municipio	ISTAC:CL_AREA_ES(02:000)
direccion	varchar	255	Dirección (tvia+nvia+numer+nommun)	

Figura 3. IDF portal (2019, ISTAC, estadística de población activa registrada(EPA-reg), p.9)

2.4.3 DIRECTORIO DE POBLACIÓN Y HOGARES: REGISTRO POBLACIÓN

. El fichero IDT del directorio contiene las variables nucleares para poder identificar al representante único de la entidad, mientras que el IDF contiene cada una de las diferentes versiones de la entidad en todos los registros administrativos que intervienen en una estadística.

UUID	STID	MARCATIEMPO	NOMEPER_TIPO_IPF	NOMEPER_FNAC	NOMEPER_SEXO	NOMEPER_CODMUNNAC	NOMEPER_PAISNAC
000811e2-fd35-485a-82c9-4e42a25438e7	c00063a_id.idf_poblacion	19JUN2019	1	07OCT1957	2	_U	_U
000b7f3f-bdc2-4a29-af3b-8b8f3fbe4220	c00063a_id.idf_poblacion	19JUN2019	1	08AUG1983	2	35009	724
0011a03f-4dbf-4f1e-98ce-4507df6632c2	c00063a_id.idf_poblacion	17JAN2020	1	26MAR1966	2	35016	724
0017c7ff-6591-4079-8279-9d407deead40	c00063a_id.idf_poblacion	06NOV2019	1	02MAY1968	2	35034	724
00301239-d1aa-4e60-aa56-4b8c60c6690f	c00063a_id.idf_poblacion	08AUG2019	1	03MAY1985	2	_U	_U
00401700-a8fb-4de7-b9bc-54f96aeaa17c	c00063a_id.idf_poblacion	19JUN2019	1	12MAY1979	2	28079	724
0059b1f0-ad10-4825-9532-62f8390b6497	c00063a_id.idf_poblacion	19JUN2019	1	27JAN1970	2	35006	724
0064d334-61ea-41db-bd8a-6e587063d14f	c00063a_id.idf_poblacion	19JUN2019	1	08AUG1968	1	35016	724
0070a1f4-5c10-4e1a-a4a4-6b5329b47768	c00063a_id.idf_poblacion	19JUN2019	1	21APR1965	2	35016	724
00816b41-3b7d-4259-abf2-8ec5e0ec57ea	c00063a_id.idf_poblacion	19JUN2019	1	01AUG1989	2	35009	724
0081f0b3-fca2-40d4-925f-ff73c98b10c4	c00063a_id.idf_poblacion	19JUN2019	1	08NOV1992	2	_Z	170
008a2007-8e4b-4d07-839b-3a5791e71af4	c00063a_id.idf_poblacion	19JUN2019	1	18SEP1965	2	35023	724
00a55d3f-a9f8-47ba-9b54-db9a0f9a315b	c00063a_id.idf_poblacion	19JUN2019	1	08FEB1985	2	35019	724

Figura 4. IDF Población

Esta imagen representa al IDF del directorio población, como vemos debe tener características propias de una persona para así poder identificarlo como un individuo diferente al resto. Evitando así que ocurran problemas de duplicidad. Los problemas de duplicidad surgen cuando un individuo tiene varios registros, pero con diferentes propiedades, creando así duplicidad en las tablas IDT de ese individuo, por ejemplo, un problema muy común son los apellidos que contienen un ‘del’ o ‘de’ o nombres compuestos, puesto que un registro puede que ese nombre se haya apuntado como ‘nombre: “*Iván*” apellido:”*Del Castillo*”’ mientras que en otro lugar se haya podido registrar como: ‘nombre: “*Iván Del* ” apellido:”*Castillo*” ’. Cuando este tipo de problemas ocurre, actuamos igual que en el directorio empresa.

Capítulo 3 BASES DE DATOS ORIENTADA A GRAFOS

3.1 ¿Qué es una base de datos orientada a grafos?

Las bases de datos orientadas a grafos se caracterizan como su nombre indica por el uso de grafos que nos permiten representar los datos interconectados de forma visual y comprensible. El grafo está formado por un conjunto de vértices o nodos, que son nuestros datos u objetos que contienen la información, y arcos o aristas que nos permiten comprender las relaciones existentes entre los diferentes nodos.

Los principales aspectos positivos de las bases de datos orientadas a grafos se pueden resumir en los siguientes puntos:

- Buen rendimiento cuando existe un crecimiento exponencial del volumen de datos, por lo que las bases de grafos son capaces de adaptarse a las exigencias estructurales que van surgiendo
- Flexibilidad, ofreciendo diferentes métodos analíticos

3.2 TECNOLOGÍAS USADAS

3.2.1 NEO4j

Neo4J es una de las bases de datos orientada a grafos (BDOG) más conocidas del mercado, se trata de una tecnología implementada en java.

El desarrollador de eBay Volker Pacher dijo lo siguiente sobre el rendimiento de uso de neo4j en su plataforma, “Nuestra solución Neo4j es literalmente mil veces más rápida que la solución anterior MySQL, con búsquedas que requieren entre 10 y 100 veces menos código”.

Una de sus otras ventajas más importantes es su agilidad a la hora de gestionar los datos, si llegáramos al límite de su capacidad la aplicación, el volumen total de datos tendría que superar los 34.000 millones de nodos y relaciones. Y por último su flexibilidad, debido a su alta capacidad de añadir nodos y relaciones a grafos creados con anterioridad.

Los principales casos de uso de Neo4J, son detección de fraudes, recomendaciones en tiempo real y redes sociales, gestión de centros de datos, y por último y la que nosotros hemos utilizado para este tipo de proyecto, la gestión de sistemas de datos maestros.

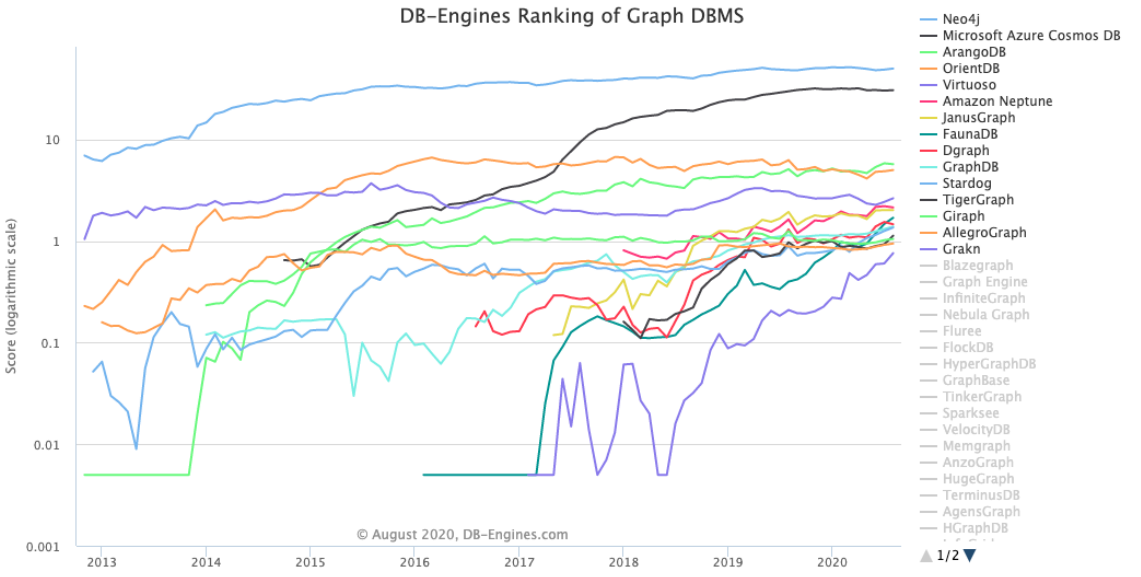
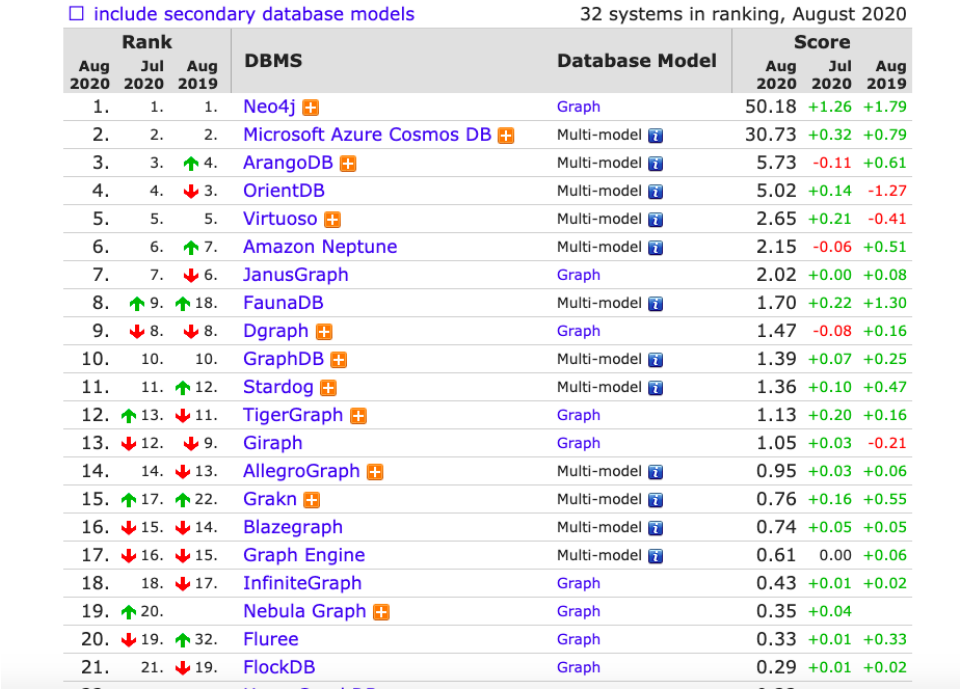


Figura 5. Gráficos en base al tiempo de base de datos orientada a grafos.[1]

Por qué elegimos NEO4j frente a otras tecnologías tales como Amazon Neptune, SAP hana graph o Orient DB que son las bases de datos más conocidas. En primer lugar, porque Neo4J es una de las principales bases de datos a nivel mundial, empresas importantes como eBay, Walmart, Lufthansa, CISCO o UBS confían en sus servicios, además se trata de una tecnología open source.

[1] https://db-engines.com/en/ranking_trend/graph+dbms



☐ include secondary database models — 32 systems in ranking, August 2020

Rank Aug 2020	Rank Jul 2020	Rank Aug 2019	DBMS	Database Model	Score Aug 2020	Score Jul 2020	Score Aug 2019
1.	1.	1.	Neo4j	Graph	50.18	+1.26	+1.79
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	30.73	+0.32	+0.79
3.	3.	↑ 4.	ArangoDB	Multi-model	5.73	-0.11	+0.61
4.	4.	↓ 3.	OrientDB	Multi-model	5.02	+0.14	-1.27
5.	5.	5.	Virtuoso	Multi-model	2.65	+0.21	-0.41
6.	6.	↑ 7.	Amazon Neptune	Multi-model	2.15	-0.06	+0.51
7.	7.	↓ 6.	JanusGraph	Graph	2.02	+0.00	+0.08
8.	↑ 9.	↑ 18.	FaunaDB	Multi-model	1.70	+0.22	+1.30
9.	↓ 8.	↓ 8.	Dgraph	Graph	1.47	-0.08	+0.16
10.	10.	10.	GraphDB	Multi-model	1.39	+0.07	+0.25
11.	11.	↑ 12.	Stardog	Multi-model	1.36	+0.10	+0.47
12.	↑ 13.	↓ 11.	TigerGraph	Graph	1.13	+0.20	+0.16
13.	↓ 12.	↓ 9.	Giraph	Graph	1.05	+0.03	-0.21
14.	14.	↓ 13.	AllegroGraph	Multi-model	0.95	+0.03	+0.06
15.	↑ 17.	↑ 22.	Grakn	Multi-model	0.76	+0.16	+0.55
16.	↓ 15.	↓ 14.	Blazegraph	Multi-model	0.74	+0.05	+0.05
17.	↓ 16.	↓ 15.	Graph Engine	Multi-model	0.61	0.00	+0.06
18.	18.	↓ 17.	InfiniteGraph	Graph	0.43	+0.01	+0.02
19.	↑ 20.		Nebula Graph	Graph	0.35	+0.04	
20.	↓ 19.	↑ 32.	Fluree	Graph	0.33	+0.01	+0.33
21.	21.	↓ 19.	FlockDB	Graph	0.29	+0.01	+0.02

Figura 6. Ranking base de datos orientada a grafos.[2]

Se trata también de un programa muy intuitivo a la hora de empezar a trabajar con bases de datos orientada a grafos.

3.2.2 CYPHER

Neo4j usa su propio lenguaje, cypher, es un tipo de lenguaje declarativo, fue diseñado con la mente en el lenguaje SQL, pero teniendo en cuenta los componentes y las necesidades de una base de datos orientada grafos, para así simplificar el trabajo y que sea un lenguaje más intuitivo.

Estructura

[2] https://db-engines.com/en/ranking_trend/graph+dbms

Cypher usa cláusulas, al igual que SQL, las query(sentencias) se realizan construyendo con varias cláusulas, la cláusulas son condiciones de modificación, que realizan funciones con los datos que se desean manipular.

Por ejemplo si creamos un grafo simple con las siguientes sentencias:

```
CREATE (john:Person {name: 'John'})
CREATE (joe:Person {name: 'Joe'})
CREATE (steve:Person {name: 'Steve'})
CREATE (sara:Person {name: 'Sara'})
CREATE (maria:Person {name: 'Maria'})
CREATE (john)-[:FRIEND]->(joe)-[:FRIEND]->(steve)
CREATE (john)-[:FRIEND]->(sara)-[:FRIEND]->(maria)
```

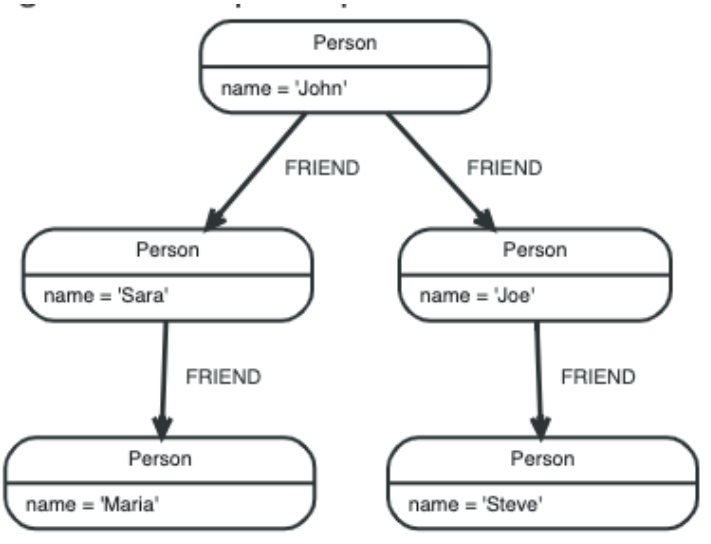


Figura 7. Ejemplo de estructura. [3]

Por ejemplo, en esta consulta encuentra un usuario llamado 'John' y amigos de 'John' (aunque no sus amigos directos) antes de devolver tanto a 'John' como a los amigos de

[3] https://neo4j.com/developer/cypher/intro-cypher/

amigos que se encuentren.

```
MATCH (john {name: 'John'})-[:FRIEND]->()-[:FRIEND]->(fof)
RETURN john.name, fof.name
```

Resulting in:

```
+----------------------+
| john.name | fof.name |
+----------------------+
| "John"    | "Maria"  |
| "John"    | "Steve"  |
+----------------------+
2 rows
```

Capítulo 4 IMPLEMENTACIÓN

4.1 Cláusulas

4.2 Datos Maestros

Después de estudiar estas tablas descritas en el capítulo dos, decidimos que todos los DAT,, diferenciando cada tabla DAT con su propiedad STID, propia de cada archivo, que nos otorga el esquema de la tabla DAT correspondiente. Esto es debido a que nos encontramos con el problema de que algunas relaciones en las tablas URD usaban “Dats” antiguas o de las cuales no contábamos con ese archivo, por lo que de alguna forma teníamos que crear

el nodo de ese DAT “inexistente” en nuestra base de datos para poder crear las relaciones correspondientes. Sin embargo, nos encontramos con un problema a la hora de trabajar con el fichero DAT de demandantes, el fichero en sí, era demasiado largo como para hacer una única sentencia con ese fichero, por ello, para simplificarlo dividimos el nodo DAT de demandantes, apartando las propiedades dentro de los demandantes que, al aplanarse el dato, obtenemos más de una columna. Creando así un tipo de nodo más pequeño que tendría una relación con el tipo de dato DAT de demandantes el cual, éste a su vez, se relacionaría con el tipo de dato DAT general, como habíamos comentado anteriormente.

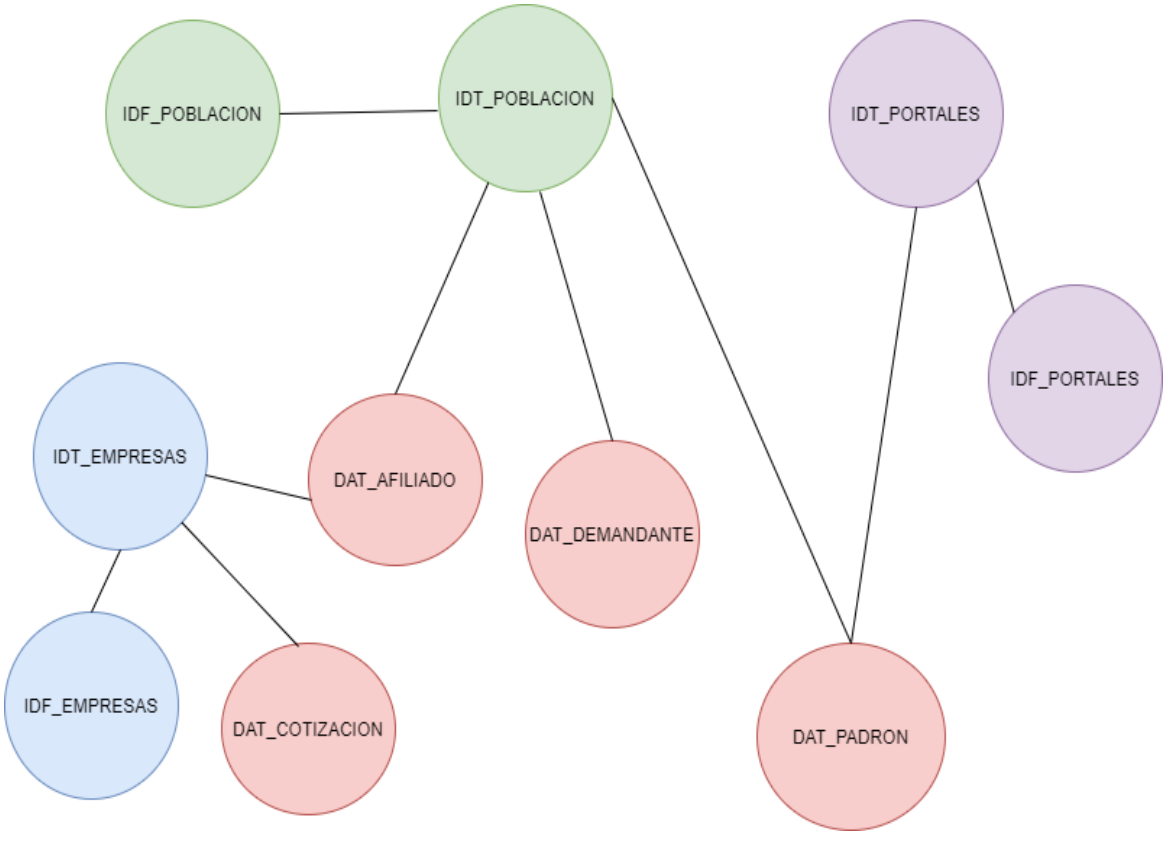


Figura 8. Organización de tipos de nodos.

Tendríamos por tanto una organización como en la figura 8, donde los IDT se relacionan con sus IDFs correspondientes, y a su vez con el tipo de nodo DAT, que tiene relación con todos los demás nodos DAT existentes.

4.2.1 FORMATO DE LOS TIPOS DE DATOS MAESTROS

Los datos se nos entregaron en forma de ficheros tipo ***CSV***, para así poder realizar una importación sencilla, nos entregaron una carpeta con todos los datos de tipo *DAT*, en concreto, datos de afiliados, del padrón y de demandantes de distintos años, y otra carpeta con las muestras de cada registro (empresa, población y portales), con tres archivos distintos dentro de esta carpeta, el archivo *IDT*, el *IDF* y el *URD*.

Todos estos archivos vienen con una cabecera, que representa el identificador de concepto de cada uno de los valores separados por “;”, y cada fila, a partir de la cabecera, se tratan de las observaciones.

4.3 MIGRACIÓN DE DATOS (QUERYS)

Para aumentar el rendimiento de la base de datos se recomienda crear índices y declarar las claves únicas de los nodos. Por ello en primer lugar creamos la clave única que automáticamente creará un índice sobre esa clave.

La sentencia utilizada es las siguiente:

```
CREATE CONSTRAINT ON (n:Idt_Poblacion) ASSERT
n.uuid IS UNIQUE;
```

Una vez cargado los índices y las claves únicas, nuestro siguiente paso, será cargar la base de datos de nodos, para ello, importamos de nuestra carpeta '*import*' en la aplicación de neo4j el archivo CSV, que queremos importar. Usamos el comando MERGE, cuando sea necesario, el cual comprueba de primeras, si ese tipo de nodo ya existe o está creado, en cuyo caso no lo volvería a crear.

```
LOAD CSV WITH HEADERS FROM
"file:///data/poblacion/IDT_SELECT_TFG.csv" AS line
FIELDTERMINATOR ";" MERGE(:Idt_Poblacion{uuid:line.UUID_IDT,
stid:line.STID_IDT})
```

Actuaremos de igual forma para todos los demás ficheros. Para poder acceder a todos los registros con las sentencias si fuera necesario, se ha incluido un anexo con todas ellas para que cualquier persona pueda si le interesara usarlas o verlas.

4.2 CONSULTAS

Capítulo 5
Conclusiones y líneas futuras

Este capítulo es obligatorio. Toda memoria de Trabajo de Fin de Grado debe incluir unas conclusiones y unas líneas de trabajo futuro

Capítulo 6
Summary and Conclusions

This chapter is compulsory. The memory should include an extended summary and conclusions in english.

Capítulo 7
Presupuesto

Este capítulo es obligatorio. Toda memoria de Trabajo de Fin de Grado debe incluir un presupuesto.

7.1 Sección Uno

Tipos	Descripción
AAAA	BBBB
CCCC	DDDD
EEEE	FFFF
GGGG	HHHH

Tabla 7.1: Resumen de tipos

Capítulo 8
Título del Apéndice 1

8.1 Algoritmo XXX

```
/***************************************************************************** *
* Fichero .h
*
***************************************************************************** *
* AUTORES
*
*
* FECHA
*
*
* DESCRIPCION
*
*
*****************************************************************************/
```

8.2 Algoritmo YYY

```
/***************************************************************************** *
 * Fichero .h
 *
 ***************************************************************************** *
 * AUTORES
     *
     * FECHA
     *
```


```
* DESCRIPCION
*
*
**********************************************************************************/ 10
```


Capítulo 9
Título del Apéndice 2

9.1 Otro apéndice:

Sección 1 texto

9.2 Otro apéndice:

Sección 2 texto

Bibliografía

https://neo4j.com/developer/cypher/intro-cypher/