Análisis de la conversación de Twitter en Uruguay utilizando Neo4j

Rodrigo Gallardo
Juan Pablo Lago
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay

Resumen

En este trabajo se utiliza la herramienta Neo4j para la gestión y análisis de una base de datos de grafos construida a partir de información de la red social Twitter. En particular, se trabaja con un subconjunto de los *tweets* publicados en Uruguay durante el mes de mayo de 2022 y los usuarios que los publicaron.

Los objetivos de este análisis son dos: analizar la conversación de Twitter en Uruguay y evaluar las capacidades de la herramienta Neo4j y la librería Graph Data Science.

Habiendo utilizado una base con cerca de 500.000 nodos y 1.000.000 relaciones, se concluye que el ecosistema de Neo4j conforma una herramienta potente, flexible y eficiente para el análisis de datos sobre grafos.

El análisis de la conversación de Twitter confirmó características que hacen a la cultura e identidad uruguaya mostrando que una base de datos con las particularidades de la utilizada es una fuente fiable de información para explorar preguntas sobre estos temas.

I. Introducción

Las bases de datos de grafos almacenan nodos, relaciones y etiquetas sin una estructura predefinida. Estas abstracciones permiten representar de manera flexible conjuntos de datos donde las relaciones juegan un papel importante, como pueden ser redes sociales, recomendaciones de productos y transacciones bancarias, por nombrar algunos. Aunque las bases de datos relacionales permiten almacenar este tipo de información, las bases de datos de grafos fueron diseñadas para almacenar grandes volúmenes de relaciones y poder navegar a través de estas de forma eficiente [1]. A su vez, Neo4j cuenta con un amigable y potente lenguaje de consultas, Cypher, y una librería con una gran variedad de algoritmos sobre grafos [2].

En este trabajo se utiliza la herramienta Neo4j para la gestión y análisis de una base de datos de grafos construida a partir de información de la red social Twitter. En particular, se trabaja con un subconjunto de los *tweets* publicados en Uruguay durante el mes de mayo y los usuarios que los publicaron.

Los objetivos de este análisis son dos:

- Analizar la conversación de Twitter en Uruguay explotando los algoritmos sobre grafos implementados en Neo4j y el lenguaje de consultas Cypher: obtención de estadísticas generales, detección de nodos importantes y detección de comunidades.
- Evaluar las capacidades de la herramienta Neo4j al trabajar con una base de tamaño considerable. En este aspecto, se evalúan las siguientes tareas: carga de datos a la base, ejecución de consultas y ejecución de algoritmos sobre grafos.

La base construida cuenta con aproximadamente 500.000 nodos y 1.000.000 relaciones. Las tareas realizadas a lo largo de este trabajo permiten concluir que el ecosistema, conformado por Neo4j, Cypher, Graph Data Science Library y otras herramientas auxiliares, es potente, flexible y eficiente a la hora de analizar datos sobre grafos.

El análisis de la conversación brindó conclusiones a nivel de los temas más hablados, donde Política y Deporte destacan sobre el resto, sobre cuentas influyentes y otras conclusiones vinculadas a las costumbres e intereses de los uruguayos.

El resto de este informe se organiza de la siguiente forma: en la sección II se resumen trabajos previos similares; en la sección III se describen las etapas de obtención de los datos, diseño y creación de la base de datos; en la sección IV se detallan las consultas y algoritmos ejecutados y sus resultados; en la sección V se encuentran las conclusiones finales del trabajo y se plantean posibles líneas de trabajo futuro.

II. Trabajos relacionados

Siendo Twitter una plataforma altamente popular cuyo contenido puede ser naturalmente representado como un grafo, existen varios trabajos previos que intentan modelar un conjunto de *tweets* y usuarios utilizando una base de datos de grafos en Neo4j.

Un ejemplo de esto es el trabajo realizado en [3]. Se utilizan consultas de Cypher para responder la pregunta “Los usuarios que usan un *hashtag* X, que otros *hashtags* usan?” y se analiza el impacto de cada *hashtag* por país. Lo interesante de este trabajo son los nodos y relaciones utilizados para modelar los datos, así como la pregunta planteada y la facilidad que tiene contestar la misma utilizando Cypher. Por otro lado, el trabajo introduce a la librería `python-twitter` [4], la cual es un *wrapper* para Python para comunicarse con la API de Twitter, y al *driver* de Neo4j para Python [5], el cual permite la comunicación con la base y la carga de datos.

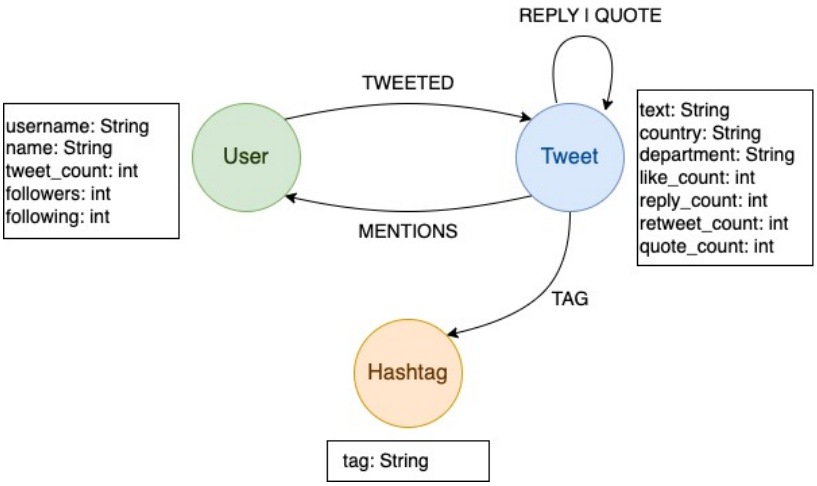

Figura 1: Diagrama de los nodos y relaciones modelados en la base, junto con los principales atributos de cada nodo y su tipo de datos.

En [6] también se modelan datos de Twitter utilizando Neo4j. El diseño de la base es similar al del trabajo anterior, aunque la carga de datos se hace utilizando código escrito en Java. El aporte importante de este trabajo es el uso de la librería Graph Data Science y la aplicación de algoritmos para la detección de usuarios populares y detección de comunidades sobre el grafo. Una de las conclusiones más relevantes del trabajo refiere a la eficiencia de la base; la mayoría de las consultas ejecutan en cuestión de segundos para una base con más de 3 millones de nodos y más de 6 millones de relaciones en una MacBook Pro del año 2014. Un trabajo similar es el presentado en [7], donde también se aplican los algoritmos de Graph Data Science sobre un grafo en Neo4j.

III. Desarrollo

III-A. Descarga de datos

Los datos usados para construir la base de datos fueron extraídos de la API de Twitter [8]. La misma es una API abierta por la cual Twitter permite la descarga de datos de *tweets* y usuarios con 3 niveles de acceso. Para poder armar una base restringida a *tweets* de Uruguay y con el volumen de datos al que se aspiraba, se realizó una solicitud al nivel mayor de acceso a la API, el *académico*, el cual nos fue concedido.

La descarga de los datos se realizó usando la librería de Python `Twarc2` [9]. La misma tiene la ventaja de gestionar la descarga de los *tweets* con el máximo de información disponible así como los límites de descarga de acuerdo al nivel de acceso. Se realizó una búsqueda de los *Tweets que fueron publicados con geolocalización activada dentro del territorio de Uruguay, con fecha de publicación en el mes de mayo de 2022*. Los resultados fueron 276.546 *tweets*, junto con la información de los usuarios que los publicaron así como de los *tweets* a los que estos respondieron o citaron.

Se estima que nuestro conjunto de datos contiene entre el $2\,\%$ y el $3\,\%$ del total de *tweets* publicados en Uruguay en mayo (ver Apéndice A). Si bien es una proporción pequeña del total, se entiende que es representativa de los temas que se hablan en el país, del tipo de núcleos que se forman y que posiblemente contenga a los *tweets* y usuarios más importantes de la conversación de Twitter en Uruguay. [1]

En total, se descargaron 720 MB de datos en formato JSON.

III-B. Diseño de la base

En la figura 1 se puede observar un diagrama de los nodos y relaciones modelados en la base. Este diseño fue elegido tomando como inspiración el utilizado en [6] y pensando en las consultas a realizar. La semántica de las relaciones utilizadas es la siguiente:

- La relación `TWEETED` modela la relación entre un usuario y un *tweet* que este publicó.
- La relación `MENTIONS` modela la relación entre un *tweet* y un usuario que se menciona en el texto de ese *tweet*.
- La relación `REPLY` modela la relación entre dos *tweets*, donde uno es una respuesta del otro.
- La relación `QUOTE` modela la relación entre dos *tweets*, donde uno cita al otro.
- La relación `TAG` modela la relación entre un *tweet* y un *hashtag* que está contenido en el texto de ese *tweet*.

[1] Si un *tweet* es publicado sin geolocalización pero tiene muchas respuestas o citas dentro de la comunidad uruguaya, es altamente probable que haya sido eventualmente referenciado por un usuario con geolocalización y que por lo tanto aparezca en la base, junto con la información del usuario que lo publicó. Como ejemplo, el usuario `LuisLacallePou` no publica con geolocalización y sin embargo se tienen 26 *tweets* publicados por él durante mayo en la base de datos .

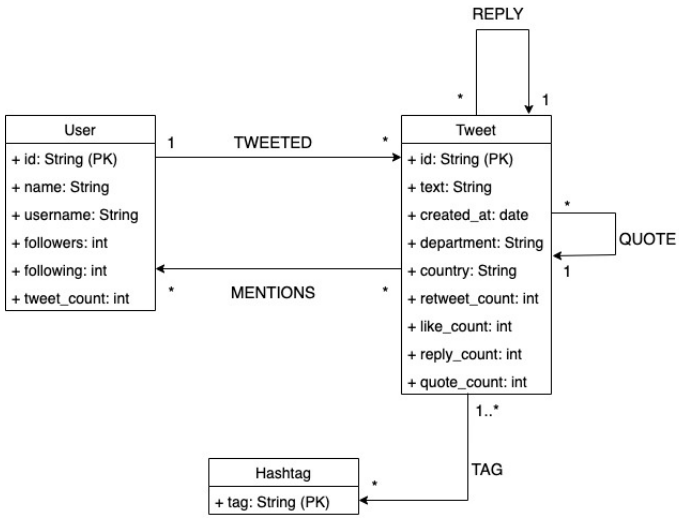

Figura 2: Diagrama de las clases creadas para trabajar con el OGM.

III-C. Carga de datos

Para la carga de datos se utilizó la librería `py2neo` [10], la cual ofrece una variada gama de herramientas para trabajar con Neo4j desde Python.

Una de estas herramientas, la cual fue utilizada para realizar la carga, es la componente de OGM (Object-Graph Mapping). Esta componente facilita la traducción de objetos creados en el lenguaje Python a información almacenada en la base de datos de grafos. Conceptualmente, esta idea es similar a la de los ORM (Object-Relational Mapping), los cuales traducen objetos de un lenguaje de programación a tuplas de tablas en una base de datos relacional.

Las clases creadas para trabajar con el OGM se pueden visualizar en el diagrama de la figura 2. Se puede observar que hay un mapeo directo entre las clases definidas y los nodos y relaciones presentados en el diseño de la sección III-B. Esto permite trabajar con un lenguaje de programación potente como Python, con una interfaz sencilla y flexible para la definición de clases, pudiendo fácilmente trasladar los objetos creados a una base de datos de grafos.

En el repositorio del proyecto [11] se puede acceder al código que realiza la carga de datos. A grandes rasgos, dicho código define estas clases, procesa los documentos JSON producto de la descarga de la API de Twitter, instancia objetos de las clases y utiliza el OGM de `py2neo` para cargar los datos a la base.

La carga de datos tomó un tiempo aproximado de 12 horas.

En cuanto a espacio de almacenamiento, el tamaño resultante de la base de datos es de 540 MB. La reducción en tamaño respecto a los datos descargados se debe a que al momento de la carga se descarta una gran cantidad de meta-data de los *tweets*.

IV. Experimentación

IV-A. Plataformas de hardware

Las plataformas de *hardware* utilizadas para ejecutar las siguientes consultas se ven resumidas en la tabla I. Estas referencias serán utilizadas para describir los tiempos de ejecución de los algoritmos.

IV-B. Descripción de la base de datos

A continuación se brindan estadísticas de la base de datos.

Número de nodos:

- Nodos `User`: 55.863
- Nodos `Tweet`: 405.389

ID	Modelo	OS	RAM	CPU	Storage
A	MacBook Pro 2017	Monterey 12.3.1	16 GB	Intel Core I7 Quad-Core 2,8 GHz	SSD
B	MacBook Pro 2020	BigSur 11.2.3	8 GB	Apple M1 (8x 2.06 GHz - 3.2 GHz)	SSD

Tabla I: Referencia para las plataformas de *hardware* utilizadas.

- Nodos `Hashtag`: 12.884
- Total de Nodos: 474.136

Del total de Tweets, 276.546 tienen geolocalización en Uruguay y corresponden a 9.309 usuarios. El resto de los Tweets y usuarios corresponden a referencias (menciones, citas, respuestas o autores de estas últimas) incluidas en los anteriores. La figura 3 permite visualizar esta composición de la base, mostrando ejemplos de nodos y relaciones, con *tweets* geolocalizados y *tweets* no geolocalizados referenciados por estos.

Número de relaciones:

- Relaciones `TWEETED`: 405.389
- Relaciones `REPLY`: 185.220
- Relaciones `MENTIONS`: 427.789
- Relaciones `TAG`: 41.782
- Relaciones `QUOTE`: 38.208
- Total de relaciones: 1.098.388

IV-C. Estadísticas de tweets

En esta sección se estudian estadísticas del flujo de *tweets*.

IV-C1. Tweets por hora del día: Se comienza analizando el volumen de *tweets* por hora del día a lo largo del mes. La consulta utilizada puede verse en el listado 1.

Listado 1 Consulta para conteo de *tweets* por hora del día

```
MATCH (t:Tweet)
WITH (datetime(t.created_at) - duration({hours: 3})).hour as hour, count(t) as total_tweets
RETURN hour, total_tweets
ORDER BY hour asc
```

En la figura 4 puede observarse que las horas en las que más se publican *tweets* son las 20-22 hs, con pico a las 21 hs. A su vez, se observa un aumento de frecuencia entre las 12 hs y las 13 hs, a la hora correspondiente al almuerzo en jornada laboral. Observando el gráfico entre las 23 hs y las primeras horas de la mañana pueden observarse cambios que posiblemente reflejen los patrones de sueño de la comunidad de Twitter uruguaya.

Esta consulta tomó 1.6 segundos en ejecutarse en el *hardware* B.

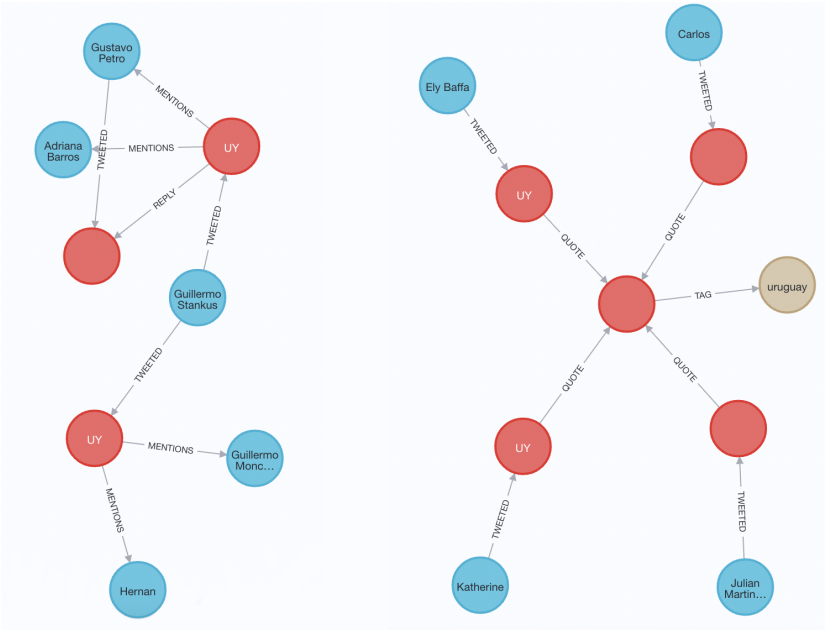

Figura 3: Grafo de ejemplo. En celeste nodos `User`, en rojo nodos `Tweet` y en marrón nodos `Hashtag`. Los nodos `Tweet` que tienen la etiqueta *UY* son *tweets* publicados con geolocalización en Uruguay.

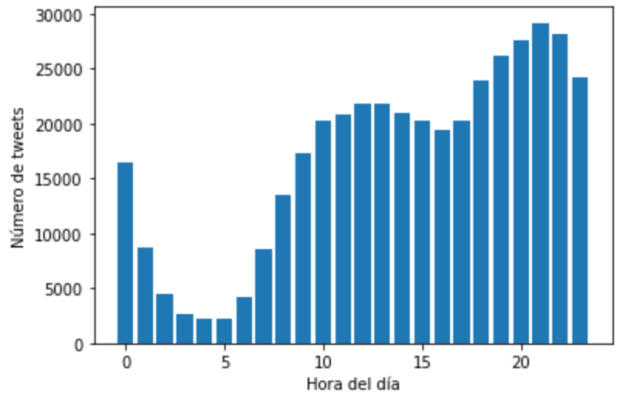

Figura 4: Volumen total de *tweets* por hora del día.

IV-C2. Flujo de tweets en el mes de mayo: Se continua analizando el flujo de *tweets* a lo largo del mes. Se utiliza una consulta análoga a la de la tabla 1. Los resultados pueden verse en la figura 5.

Hay un promedio cercano a 13.000 *tweets* diarios en la base que corresponde aproximadamente a 500.000 *tweets* uruguayos por día (ver Apéndice A). En el gráfico se observan picos en el volumen de *tweets* los días 17, 25 y en menor medida los días 28, 29 y 30. Para cada uno de estos picos se realizó una consulta para determinar los *hashtags* que más fueron usados. La correspondiente al 17 de mayo puede verse en el listado 2. Esta consulta tomó 1.7 segundos en el *hardware* B.

La figura 6 muestra los *hashtags* más usados en cada fecha. A partir de la observación de la tabla y una breve investigación en la web se detectan los siguientes eventos vinculados con el aumento de *tweets*:

- **Martes 17 de mayo**: Jugó Peñarol en la Libertadores y fue el día del Ciclón Subtropical.
- **Miércoles 25 de mayo**: Jugó nuevamente Peñarol en la Libertadores y se cumplieron 122 años del Gran Parque Central.
- **Sábado 28 de mayo - Lunes 30 de mayo**: El fin de semana coincidieron el Gran Premio de Mónaco (F1) y la final de la UEFA Champions League, lo cual fue seguido por la ola de frío que comenzó el lunes 30 (Ej: `Este frío es insalubre no seas malo #TeamVerano siempre`)

Listado 2 Consulta de *hashtags* más usados el 17 de mayo.

```
MATCH (t:Tweet) -[*1..2]-> (h:Hashtag)
WHERE (datetime(t.created_at) -duration({hours: 3})).day = 17
WITH h, count(t) as tweets
RETURN h.tag as hashtag, tweets
ORDER BY tweets desc
LIMIT 6
```

IV-D. Detección de usuarios importantes

En esta sección se utilizaron algoritmos de centralidad sobre grafos para determinar la importancia de los nodos dentro del grafo; en particular, de los usuarios. El objetivo de esta sección es identificar los usuarios más influyentes en la conversación en Twitter en Uruguay. Para esto se utiliza la librería Graph Data Science de Neo4j, la cual cuenta con diversos algoritmos de

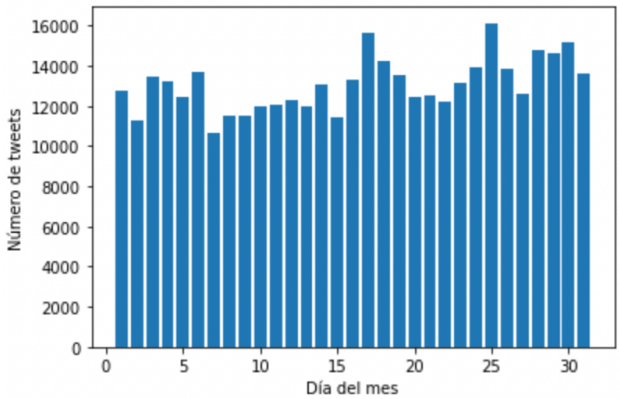

Figura 5: Volumen total de *tweets* por día del mes de mayo.

17 de mayo

	hashtag	tweets
0	ciclonextratropical	103
1	peñarol	71
2	elclubgigante	66
3	uruguay	56
4	libertadores	51
5	vamoscarbonero	45

25 de mayo

	hashtag	tweets
0	peñarol	83
1	libertadores	80
2	vamoscarbonero	53
3	elclubgigante	52
4	colón	42
5	122añosdelgpc	29

28-30 de mayo

	hashtag	tweets
0	monacogp	136
1	puntopenalenel10	111
2	uruguay	111
3	uclfinal	109
4	mytopfollowers	94
5	f1	86
6	elpeorgobiernodelahistoria	77
7	teamverano	77
8	peñarol	75
9	elclubgigante	68
10	bts	68
11	champ14ns	64

Figura 6: *Hashtags* de los días con mayor volumen de *tweets* del mes.

ciencia de datos para grafos. Tanto en esta sección como en la detección de comunidades se utilizaron solamente algoritmos en etapa de disponibilidad para uso en producción.

En primer lugar fue necesario realizar una proyección del grafo original a uno que modele las relaciones entre usuarios, inspirándose en el trabajo de [6]. Dicha proyección solo considera nodos `User` y añade las siguientes relaciones entre ellos:

- Se añade una relación desde `user2` a `user1` si `user2` respondió o citó un *tweet* que `user1` publicó.
- Se añade una relación desde `user2` a `user1` si `user2` respondió o citó un *tweet* que menciona a `user1`.
- Se añade una relación desde `user2` a `user1` si `user2` publicó un *tweet* que menciona a `user1`.

El listado 3 contiene la consulta que construye estas relaciones.

Listado 3 Consulta que construye las relaciones para la proyección de usuarios

```
MATCH (u1:User)-[:TWEETED]->(t1:Tweet)<-[:QUOTE|REPLY]-(t2:Tweet)<-[:TWEETED]-(u2:User)
RETURN id(u2) as source, id(u1) as target
UNION
MATCH (u1:User)<-[:MENTIONS]-(t1:Tweet)<-[:QUOTE|REPLY]-(t2:Tweet)<-[:TWEETED]-(u2:User)
RETURN id(u2) as source, id(u1) as target
UNION
MATCH (u1:USER)<-[:MENTIONS]-(t1:Tweet)<-[:TWEETED]-(u2:User)
RETURN id(u2) as source, id(u1) as target
```

El grafo proyectado cuenta con 55860 nodos y 187877 relaciones. Una vez realizada la proyección, se procede a ejecutar los algoritmos sobre el grafo proyectado.

IV-D1. Estructura de la consulta: El listado 4 tiene el código de una consulta que invoca a un algoritmo de centralidad sobre el grafo proyectado y extrae información de los veinte usuarios más importantes: nombre, nombre de usuario, cantidad de seguidores, cantidad total de *tweets*, puntaje del algoritmo y cantidad de *tweets* en la base.

Listado 4 Consulta que invoca un algoritmo de centralidad y extrae información relevante.

```
CALL gds.<ALGORITHM>.stream('tweets')
YIELD nodeId, score
WITH gds.util.asNode(nodeId) AS node, score
MATCH (u:User)-[:TWEETED]->(t:Tweet)
WHERE u.username = node.username
WITH u, score, count(t) as may_tweets
RETURN
   u.name as Name,
   u.username as Username,
   u.followers as Followers,
   u.tweet_count as TotalTweets,
   score as Score,
   may_tweets as MayTweets
ORDER BY Score DESC, Name ASC
LIMIT 20
```

IV-D2. Algoritmo Page Rank Centrality: El algoritmo *Page Rank* le asigna importancia a los nodos dependiendo de la cantidad de arcos entrantes y la importancia de los nodos origen de esos arcos. En la figura 7 se pueden ver los veinte usuarios con mayor puntaje según el algoritmo *Page Rank*.

Se puede ver que los tres primeros lugares están ocupados por algunos de los noticieros más populares de Uruguay, y que muchas otras fuentes de noticias o cuentas de instituciones públicas se encuentran entre los primeros veinte lugares. Esto es esperable, debido a que es muy posible que los *tweets* publicados por estos usuarios sean respondidos y citados por muchos otros usuarios, alguno de estos también usuarios importantes: figuras públicas, instituciones u otras fuentes de noticias.

Se puede observar que los usuarios en las posiciones 5, 11, 13 y 18 dentro de los primeros veinte no son figuras públicas reconocidas. El hecho de que estos usuarios califiquen dentro de los primeros veinte puede deberse a la alta cantidad de *tweets* que tienen dentro de la base (ver columna `MayTweets` en la figura 7), lo cual provoca que tengan una mayor presencia e interacción con los demás usuarios y *tweets*.

Esta consulta tomo un total de 1.4 segundos en ejecutar en la plataforma de *hardware* A.

	Name	Username	Followers	TotalTweets	Score	MayTweets
0	Telemundo	TelemundoUY	281014	147473	204.974342	367
1	Subrayado	Subrayado	833617	200657	129.424268	199
2	EL PAÍS	elpaisuy	792004	417586	121.168402	398
3	leo sarro press	leosarro	51348	14180	114.979262	133
4	PEÑAROL | Basketball	BasketCAPuy	33843	3006	85.854922	53
5	Isabel_	Isabel66991411	5478	78777	67.521515	3502
6	Inumet	MeteorologiaUy	146749	7438	57.485985	21
7	Nacional	Nacional	341359	29775	50.366663	154
8	MSP - Uruguay	MSPUruguay	164823	15172	37.949762	31
9	Montevideo Portal	portalmvd	571372	456315	37.355354	324
10	PEÑAROL	OficialCAP	434933	43753	35.323001	142
11		boniatero75	2129	42533	31.685419	1943
12	Diego Píriz	diegopirizg	11530	7520	28.869217	12
13	Antonella Gordillo	AntoGordillo	5616	14396	28.841041	785
14	Coalición Multicolor	CoalicionMulti	10652	8402	26.430641	5
15	infobae	infobae	3183161	879621	26.171829	91
16	Graciela Bianchi	gbianchi404	74276	60786	24.904681	137
17	Luis Lacalle Pou	LuisLacallePou	416367	27270	23.152289	26
18	scestau	SCestau	3074	32112	21.788438	769
19	Pedro González	Pedringonsan	9431	79739	21.415263	675

Figura 7: Veinte usuarios más relevantes según el algoritmo *Page Rank Centrality*.

IV-D3. Algoritmo Article Rank Centrality: El algoritmo *Article Rank* es una variante de *Page Rank* donde la influencia de los nodos con menor *out-degree* (cantidad de arcos salientes) se ve disminuida. Esto es una diferencia importante con *Page Rank*, que asigna mayor influencia a los nodos con menor *out-degree*. En la figura 8 se pueden ver los veinte usuarios con mayor puntaje según el algoritmo *Article Rank*.

Se observa que en este caso, aquellos nodos que representan figuras públicas o instituciones reconocidas siguen presentes en la tabla pero en posiciones menos importantes. Sin embargo, aquellos nodos que representan usuarios comunes y corrientes pero con alto volumen de *tweets* en la base aumentan su posición; por ejemplo, los usuarios “Isabel66991411” o “boniatero75”.

Esto tiene sentido con la premisa del algoritmo *Article Rank*, que disminuye el peso de los nodos con menor *out-degree*. Es muy probable que usuarios con menor volumen de *tweets* en la base tengan menor *out-degree* en la proyección ya que seguramente interactúen con menos usuarios. Sin embargo, usuarios con mayor volumen, por más de que no sean populares, interactúan con mayor cantidad de usuarios poco populares.

La consulta tomó un total de 1.4 segundos en ejecutar en la plataforma de *hardware* A. Resulta interesante observar la velocidad de ejecución de los algoritmos *Page Rank* y *Article Rank*. Una explicación puede ser que Graph Datascience Library se encargue de calcular la meta-data de los nodos utilizada por estos algoritmos al momento de almacenar la proyección del grafo; por ejemplo, computar los grados de cada nodo.

IV-D4. Algoritmo Betweenness Centrality: El algoritmo *Betweenness* intenta medir la influencia que tiene un nodo sobre el flujo de información en el grafo. Para esto, se computan todos los caminos más cortos entre cada par de nodos en el grafo y se asigna un puntaje a cada nodo dependiendo de la cantidad de caminos más cortos que pasan por él. En la figura 9 se

	Name	Username	Followers	TotalTweets	Score	MayTweets
0	Isabel_	Isabel66991411	5478	78777	16.372308	3502
1	Nacional	Nacional	341359	29775	14.412332	154
2	PEÑAROL	OficialCAP	434933	43753	11.890014	142
3	Telemundo	TelemundoUY	281014	147473	11.822032	367
4	Montevideo Portal	portalmvd	571372	456315	9.218680	324
5	Luis Lacalle Pou	LuisLacallePou	416367	27270	8.952443	26
6	EL PAÍS	elpaisuy	792004	417586	8.804899	398
7		boniatero75	2129	42533	8.525932	1943
8	Subrayado	Subrayado	833617	200657	8.059107	199
9	leo sarro press	leosarro	51348	14180	7.796260	133
10	Graciela Bianchi	gbianchi404	74276	60786	7.703635	137
11	Antonella Gordillo	AntoGordillo	5616	14396	7.160323	785
12	gaby	sgabyo	2084	28238	6.759563	1820
13	scestau	SCestau	3074	32112	5.819341	769
14	Pedro González	Pedringonsan	9431	79739	5.794296	675
15	Min. Diego González González	diegodelacurva	336400	136196	5.791689	257
16	PEÑAROL | Basketball	BasketCAPuy	33843	3006	5.442293	53
17	Punto Penal	Punto_Penal	124801	48936	4.967155	62
18	Frente Amplio	Frente_Amplio	123620	24552	4.547760	34
19	Gabriela	zurdayo	3573	31947	4.443434	1150

Figura 8: Veinte usuarios más relevantes según el algoritmo *Article Rank Centrality*.

	Name	Username	Followers	TotalTweets	Score	MayTweets
0	Isabel_	Isabel66991411	5478	78777	3.662321e+07	3502
1		boniatero75	2129	42533	2.682564e+07	1943
2	gaby	sgabyo	2084	28238	2.297714e+07	1820
3	Insoportablemente Bolso	KiwiNacional	1355	53954	1.675676e+07	941
4	LMP	Lelengo86	513	5257	1.669779e+07	1586
5	Antonella Gordillo	AntoGordillo	5616	14396	1.615455e+07	785
6	Jorge Andrés	JorgeAndresBusi	2554	94890	1.609382e+07	1017
7	Antonella	Antonnella1891	3718	63855	1.352389e+07	962
8	Carmen rinaldi	CarmenRinaldi4	5453	147724	1.332232e+07	620
9	Nati	NatiMArbiza	1643	53603	1.280733e+07	739
10	me dicen Fidel	Fidel_Espanta	4541	9474	1.246657e+07	242
11	Pedro González	Pedringonsan	9431	79739	1.226639e+07	675
12	JACK_TORRANCE	LaMoscaFliesu2	638	13809	1.151280e+07	2151
13	MaGnUs	lordmagnusen	4898	103162	1.126975e+07	971
14	Min. Diego González González	diegodelacurva	336400	136196	1.115275e+07	257
15	Charles Coubrough	CharlesDVM	3115	46811	1.081555e+07	783
16	Tefaa1899	tefaaa29	742	3000	1.068354e+07	703
17	Tía Brishith #87CEEB	latiabrishith	2661	35290	1.042024e+07	924
18	scestau	SCestau	3074	32112	1.029457e+07	769
19	Estela	EstelaF68436226	3484	84862	1.005411e+07	1404

Figura 9: Veinte usuarios más relevantes según el algoritmo *Betweenness Centrality*

pueden ver los veinte usuarios con mayor puntaje según el algoritmo *Betweenness*.

Es importante destacar la diferencia en el resultado de este algoritmo en comparación con los algoritmos *Page Rank* y *Article Rank*. En este caso, salvo el usuario en la posición 14, ninguno de los demás es una figura pública o institución "popular". Sin embargo, todos los usuarios dentro de los primeros veinte lugares tienen un alto volumen de *tweets* en la base. Esto tiene sentido, ya que estos usuarios, al tener una alta presencia en el grafo, probablemente tengan un mayor número de interacciones con otros usuarios.

La consulta tomó un total de 2 minutos en ejecutar en la plataforma de *hardware* A. El tiempo de ejecución es mayor que los algoritmos de centralidad probados antes debido a que el algoritmo *Betweenness* es más costoso computacionalmente; computa todos los caminos más cortos entre cada par de nodos. Sin embargo, según la documentación de Neo4j [12] la implementación utiliza una estimación de la cantidad de caminos más cortos que pasan por cada nodo. Esta implementación tiene orden de ejecución $O(n * m)$ donde n es el número de nodos y m el número de relaciones.

IV-D5. Algoritmo Degree Centrality: El algoritmo *Degree Centrality* trabaja con el *in-degree* y el *out-degree* de cada nodo para asignarle una medida de importancia. El algoritmo ha sido utilizado previamente para computar la importancia de nodos en redes sociales; en particular, en Twitter [13]. En la tabla 10 se pueden ver los veinte usuarios con mayor puntaje según el algoritmo *Degree Centrality*.

En este caso, igual que con el algoritmo *Betweenness*, se puede observar que las veinte posiciones más importantes están dominadas por usuarios que no son figuras públicas o instituciones, pero que sin embargo tienen un alto volumen de *tweets* en la base. Esto tiene sentido ya que estos usuarios probablemente tengan un alto número de interacciones con otros usuarios

	Name	Username	Followers	TotalTweets	Score	MayTweets
0	Isabel_	Isabel66991411	5478	78777	961.0	3502
1	máximo j gutierrez z	bonnevilleminas	894	83471	912.0	1271
2	Olga Alonso	OlgaAlonso9	1206	41509	788.0	1340
3	gaby	sgabyo	2084	28238	773.0	1820
4	Jorge Andrés	JorgeAndresBusi	2554	94890	747.0	1017
5	Jenny Parada Martino	MartinoParada	1137	62402	662.0	1625
6	Fernandomassa	Fernand47402543	2017	11601	581.0	814
7	Gabriela	zurdayo	3573	31947	566.0	1150
8	Insoportablemente Bolso	KiwiNacional	1355	53954	532.0	941
9	Ángela Ruso	anyelinaruso	2740	15821	499.0	765
10		boniatero75	2129	42533	481.0	1943
11	Thalos	Axulon	11657	68845	478.0	872
12	Charles Coubrough	CharlesDVM	3115	46811	468.0	783
13	Pedro González	Pedringonsan	9431	79739	468.0	675
14	Guillermo Stankus	GuillermoStank2	1678	23767	443.0	575
15	MaGnUs	lordmagnusen	4898	103162	437.0	971
16	ProgressiveYorugua	PandaYorugua	970	8771	431.0	435
17	JACK_TORRANCE	LaMoscaFliesu2	638	13809	425.0	2151
18	Antonella Gordillo	AntoGordillo	5616	14396	422.0	785
19	Rudhy Weiss	rew610521	1331	31086	416.0	666

Figura 10: Veinte usuarios más relevantes según el algoritmo *Degree Centrality.*

que sí se encuentran presentes en la base.

Se destaca que el usuario "Isabel66991411" se posiciona primero en los resultados de tres de los algoritmos y sexto en los resultados de *Page Rank*, implicando que este nodo probablemente sea central en el grafo. Sorprendentemente, este usuario no corresponde a una persona "popular" de la realidad, sino a alguien que publicó muchos de los *tweets* almacenados en la base. A su vez, el $90\,\%$ de los *tweets* publicados por este usuario son respuestas, lo cual provoca que esté relacionado con un gran número de otros usuarios en la proyección del grafo.

La consulta tomó 1.1 segundos en ejecutar en la plataforma de *hardware* A. Este velóz tiempo de ejecución puede no solo deberse a que el algoritmo *Degree Centrality* es un algoritmo sencillo, sino también a que los puntajes asignados a cada nodo ya se encuentren calculados desde la etapa de proyección del grafo.

IV-D6. Eficiencia de los algoritmos de centralidad: Como parte de la evaluación de Neo4j y la Graph Data Science Library, se decidió ejecutar los algoritmos de centralidad sobre el grafo completo (incluyendo nodos `User`, `Tweet` y `Hashtag`) para observar si el tiempo de ejecución se degrada significativamente.

Se señala que el grafo proyectado tiene un $11.7\,\%$ de la cantidad de nodos del grafo completo y un $17.1\,\%$ de la cantidad de relaciones; el grafo proyectado es al menos 5 veces menor que el grafo completo. Por otro lado, el grafo proyectado tiene una densidad de 6×10^{-5} comparado con 5×10^{-6} para el grafo completo. Se cree que ambos factores, tamaño y densidad, pueden afectar los tiempos de ejecución de algunas consultas.

Luego de ejecutar los algoritmos sobre el grafo completo, se destaca que para los algoritmos *Page Rank*, *Article Rank* y *Degree Centrality* los tiempos de ejecución no sufren degradaciones significativas. Los tres algoritmos ejecutan en aproximadamente 5 segundos versus un promedio de 1.3 segundos sobre el grafo proyectado. Esto respalda la creencia de que estos algoritmos trabajan sobre meta-datos de los nodos computados en la etapa de proyección del grafo, lo cual resulta en consultas mucho más veloces.

Por otro lado, el tiempo de ejecución del algoritmo *Betweenness* para el grafo completo fue de 25 horas versus 2 minutos sobre el grafo proyectado [2]. Esta degradación es razonable considerando la naturaleza del algoritmo, que calcula una estimación de la cantidad de los caminos más cortos que pasan por cada nodo. Sin embargo, se observa que esta diferencia en tiempos de ejecución es bastante más elevada de lo que da a entender la documentación de Neo4j cuando dice que el orden de ejecución es $O(n * m)$.

IV-E. Detección de comunidades

En esta sección se corre el algoritmo de Louvain para detección de comunidades. El método de Louvain es un algoritmo para detectar comunidades en redes de gran tamaño. Maximiza un puntaje de modularidad, donde la modularidad cuantifica la calidad de la asignación de nodos a comunidades. Esto implica que evalúa cuánto más densamente conectados están los nodos dentro de una comunidad, comparado con cuánto lo estarían en una red aleatoria.

Para correr este algoritmo se utiliza una proyección diferente a la usada para detección de usuarios importantes. Se proyecta sobre nodos `User` y nodos `Tweet`, manteniendo todas las relaciones entre estos. La heurística tras esta decisión es que los *tweets* (con sus referencias y menciones) también son 'miembros' de la comunidad de diálogo que los usa.[3] Por otro lado, se convierten las relaciones en no dirigidas, ya que de esta manera suelen obtenerse mejores resultados para este algoritmo de acuerdo a la documentación de Neo4j [14]. La consulta utilizada, que puede verse en la tabla 5, fue la segunda más costosa entre las realizadas habiendo tomado 42.6 segundos en ejecutarse en el *hardware* B.

Listado 5 Consulta que corre el algoritmo de Louvain y extrae información relevante.

```
CALL gds.louvain.stream('tweets_no_hashtags')
YIELD nodeId, communityId
WITH communityId, count(nodeId) as size, collect(nodeId) as ids
MATCH (t:Tweet)-[*1..2]->(n)
WHERE id(t) in ids AND (n:Hashtag or n:User)
WITH communityId, size, n.tag as tag, n.name as name, count(t) as t_count
ORDER BY t_count DESC
RETURN communityId, size, collect(name)[0..15] AS users, collect(tag)[0..15] AS hashtags
ORDER BY size DESC
LIMIT 12
```

[2]La ejecución sobre el grafo completo se realizó dos veces obteniendo un resultado muy similar

[3]Se probó también el uso de la misma proyección utilizada en la detección de usuarios importantes, pero los resultados obtenidos con esta proyección tienen mayor sentido.

	communityId	size	users	hashtags
0	187790	28815	[Graciela Bianchi, Tía Brishith #87CEEB, leo sarro press, Luis Lacalle Pou, Frente Amplio, EL PAÍS, Sebas1, MonicaBatlle, Rudhy Weiss, GRAZIANO PASCALE, Montevideo Portal, Telemundo, Yamandú Orsi, Romina Pesce, Adry More]	[uruguay, hayordendenoaflojar, mayomesdelamemoria, marchadelsilencio2022, porquetodosesabe, ladiariajusticia, santoyseña, lamascarauy, eltuitazo2, 1demayo, elpeorgobiernodelahistoria, peñarol, hoy, ahora, todossomosfamiliares]
1	59914	28601	[Nacional, Insoportablemente Bolso, Tefaa1899, laabdon.com.uy, BOLSO AMARTE ES UN PLACER, PASIÓN TRICOLOR ((1010 Am)), CONMEBOL Libertadores, Mauri Pérez, Lea Garcia, Valentín Canale, Estudiantes de La Plata, PEÑAROL, Carmen rinaldi, Dahi, Lucas]	[elclubgigante, nacional, libertadores, decano1solo, jueganacional, 123años, laradiodelhincha, puntopenalenel10, orgullonacional, mayotricolor, sudamericana, 123añosdeverdad, oídossanos, lafiestadelhincha, canterainagotable]
2	288489	27787	[PEÑAROL, Negro y de Peñarol ™, Nati, Nacional, PEÑAROL | Basketball, Punto Penal, Wilson Méndez, TONGA, Cecilia Magdalena, PUMA Uruguay, BUYSAN, me dicen Fidel, padreydecano.com, Martin Charquero, daniel cuadrado1891]	[peñarol, puntopenalenel10, vamoscarbonero, libertadores, apertura2022, nacional, todosjuntos, lamascarauy, marchadelsilencio2022, elequipoquenosune, auf, nuncamasterrorismodeestado, espnenstarplus, elclubgigante, campeonatouruguayo]
3	1747	26166	[JACK_TORRANCE, Radio Belcha®, Laura, India, Diego, AmableDonante, Stepha, Daniel Bengoa, Mika, medicenrubia, Sandra, Carol, edu, Nicolás Hidalgo]	[f1, monacogp, marchadelsilencio2022, laradioestuya, mayomesdelamemoria, miamigp, f12022, todossomosfamiliares, lamascarauy, essereferrari, laaldea, tip, nuevafotodeperfil, teaminvierno, elpeorgobiernodelahistoria]
4	147333	17015	[Isabel_, gaby, Lunaro, Vicky, Jazmín, Silvia, Dr. Guillermo de los Santos, Genny Rodríguez, Jacquelin Sánchez Quintero, Gladys Nohemi, Toñi Valiente. ® ™, Joyce, Eduardo Flores, Luis Pagani, Pablo]	[mytopfollowers, grupodeamigosentwitter, tiktok, nuevafotodeperfil, lunagitana, hayordendenoaflojar, teamverano, uruguay, lavidaesbella, seanfelices, sefeliz, buenlunes, lamascarauy, teaminvierno, inflacion]
5	91250	14877	[Gabriela, Jorge Andrés, Subrayado, El Pala, Jenny Parada Martino, Luis Lacalle Pou, Mika, Graciela Bianchi, Bea, Marta Alvarez, Gabriel Pereyra, Alejandro, Telemundo, rita silvan, Harley]	[renunciaheber, marchadelsilencio2022, mayomesdelamemoria, ahora, uruguay, elpeorgobiernodelahistoria, polémicaenel10, todossomosfamiliares, blancospillos, lahiena, presente, marchadelsilenciopresente, verdadmemoriayjusticia, ley19728, graciastabaré]
6	1199	13777	[Luis Lacalle Pou, Daniel Salinas, Luis Alberto Heber, MSP - Uruguay, Partido Nacional, Ministerio de Desarrollo Social, Jose Luis Satdjian, Sebastian Da Silva, Martin Lema, Alfin, Pablo Mieres, Beatriz Argimón, Frente Amplio, ASSE Comunica, Javier García]	[hayordendenoaflojar, uruguay, ahora, haciauncompromisonacional, marchadelsilencio2022, mayomesdelamemoria, calibertades25, somosdelpueblo, fotografia, diamundialdelalibertadeprensa, renunciaheber, elpeorgobiernodelahistoria, photography, elclubgigante, todossomosfamiliares]
7	447429	10767	[Estela, Alejandro Raffo, Majo, Graciela Gadea, Ricardo Pons, Nestor Grajales, Julio Cesar Pradie Colmán, Flora Cukierman, jose ernesto costemalle, MaRosa, Nora, CIUDADANOJorgeCASTRO, Luis Lacalle Pou, Maria, Mi Nona Celia]	[felizviernes, felizjueves, felizmiercoles, mytopfollowers, nuevafotodeperfil, felizsabadoatodos, eltuitazo2, felizmartesatodos, uruguay, felizviernesatodos, teamverano, pintouruguay, felizsabado, felizmiercolesatodos, redwilsonista]
8	486	10559	[Club Atlético Aguada, PEÑAROL | Basketball, FUBB, Pablo Batista, Val, PablO RiverA, Pablo, Daniel, Liga Uruguaya de Básquetbol (Oficial), Franco Fernández, Hinchada Aguatera, Salvador, Mario, Alen Banewur Rubin, Al Thornton]	[lub, aguada100años, vamoscarbonero, peñarol, aguada, miclub, 100pretrouville, libertadores, 40aniversario, vamoslaroja, lamascarauy, vamoslaspibas, somoslaliga, finaleslub, puntopenalenel10]
9	78433	9973	[ANALI BENTENCOURT, Beatriz García Montejo, Susy, Michele Piquet, DECOS HIPPOCRATES, Roberto, Bea, Patricia, Juana Lalo, Luz Rodríguez, #CadenaDeZurdos, angel camarano, nefridio, Colo, mónica, JOTACEPE J.C.P.]	[marchadelsilencio2022, todossomosfamiliares, mayomesdelamemoria, mytopfollowers, renunciaheber, nuevafotodeperfil, buenlunes, montevideo, memoriaverdadyjusticia, nadielee, buenmartes, elpeorgobiernodelahistoria, niolvidoniperdon, lavidaesbella, seanfelices]
10	289194	7851	[Antonella Gordillo, Charles Coubrough, Fernanda Guerra, Pablo Carrasco, Mariano, Pablo Bonasso, Lic. Tricolor 2, Gustavo Zubia, Nacional, Rurales El País Uy, Hernán Zorrilla de San Martín, CIUDADANOJorgeCASTRO, El Mago, Lic. Nancy Pacheco, Esc. Lic. Eva Sora]	[uruguay, hereforduruguay, tiktok, agro, criandofuturo, uclfinal, teaminvierno, paloenlarueda, vieragro, agribusiness, durazno, ahora, farming, sucive, ucl]
11	128365	7801	[Teledoce, Quién Es La Máscara Uruguay, Angel, elisa, eltrece, @adrianabravista, ElEjercitoDeLAM, El hotel de los famosos, Lali, Eugenia Suarez, Juanito Say, Ana María Guimaraens, Marí, Carina Zampini, Esta Boca es Mía]	[lamascarauy, elhoteldelosfamosos, lam, lamáscarauy, martinfierro, lpa, lamatina, premiosplatino2022, johnnydepp, uruguay, amberheard, justiceforjohnnydepp, intrusos, sdtv, lavozuruguay]

Figura 11: Resultados del algoritmo de detección de comunidades. `users` y `hashtags` son los usuarios y hashtags más mencionados y utilizados en los *tweets* de cada comunidad.

La figura 11 muestra los resultados de correr el algoritmo. En casi la totalidad de las comunidades detectadas hay un tema claro que destaca entre los *hashtags* y, en algunos casos, dos. Una posible separación de temas detectados podría ser la siguiente:

- **Política**: Índices 0 (Política y Prensa), 5 (Política y Prensa), 6 (Partido Nacional).
- **Deportes**: Índices 1 (Nacional), 2 (Peñarol), 8 (Básquetbol).
- **"Photologuers versión Twitter"**: Índices 4 y 7. En estos grupos dominan hashtags como `mytopfollowers`, `grupodeamigosentwitter`, `nuevafotodeperfil`, `felizviernes`, `felizjueves`, etc.
- **Agro**: Índice 10.
- **TV y famosos**: Índice 11.

A su vez las comunidades con Índice 3 y 9 son un poco más híbridas a nivel de temas: en la primera predomina la Fórmula 1 y otros temas variados, y la segunda parece compartir afiliación al Frente Amplio y el perfil "Photologuers versión Twitter".

V. Conclusiones y trabajo futuro

V-A. Potencia de Neo4j

A lo largo de este trabajo se exploraron diversas funcionalidades de Neo4j y librerías externas que trabajan con el mismo. Aunque muchas de estas aún se encuentran en proceso de madurez, se cree que este ecosistema de soluciones es una herramienta potente para el análisis de datos modelados como grafos.

Por un lado, la librería `py2neo` presenta una interfaz sencilla en Python para la comunicación con una base de datos de grafo en Neo4j. La ejecución de consultas utilizando Cypher y Graph Data Science Library permite la recuperación de datos de mucho interés de la base, los cuales pueden ser analizados o trabajados con mayor profundidad con programas de Python. También, el componente OGM facilita enormemente la carga de datos, ofreciendo una integración potente y flexible entre las clases de Python y los datos a almacenar en el grafo. La existencia de librerías como `py2neo`, que facilitan la integración de Neo4j con tecnologías ya conocidas, promueven la adopción de la herramienta y expanden los limites de lo que se puede realizar, lo cual finalmente aporta a la madurez del ecosistema en general.

El lenguaje de consultas Cypher demostró una gran facilidad de uso y flexibilidad para realizar las consultas planteadas en este trabajo. Con consultas en su mayoría simples e intuitivas, se lograron obtener estadísticas generales de la base y proyecciones de la misma.

Finalmente, se cree que la Graph Data Science Library es una herramienta de gran importancia para la ejecución de algoritmos sobre grafos. Aunque muchas implementaciones de algoritmos aún se encuentran en etapas de evaluación, la librería cuenta con un variado catálogo de algoritmos listos para sistemas en producción. Dentro de las principales fortalezas de la librería

se encuentra su facilidad de uso, capacidad para ejecutar los algoritmos sobre proyecciones de grafos, eficiencia e integración con Cypher, lo cual facilita la implementación de consultas más complejas. Sin dudas Graph Data Science Library posiciona a Neo4j como una buena opción para trabajos académicos o industriales.

V-B. Eficiencia

Se destaca la rapidez en tiempo de ejecución de las consultas y algoritmos sobre el grafo, de los cuales la gran mayoría ejecuta en el orden de los segundos.

Esto es evidencia de una de las más importantes premisas de Neo4j, que es la eficiencia en la navegación a través de las relaciones. Esta capacidad de Neo4j permite que las consultas de Cypher ejecuten en tiempos cortos en una base de tamaño considerable.

Por otro lado, se cree que muchos de los algoritmos sobre grafos se benefician de meta-data calculada al momento de la proyección, lo cual acelera enormemente los tiempos de ejecución.

Un caso excepcional es el algoritmo de centralidad *Betweenness*, el cual tomó un total de 2 minutos sobre el grafo proyectado y 25 horas sobre el grafo completo. Este es un algoritmo computacionalmente costoso, por lo que son esperables tiempos de ejecución mayores que los demás algoritmos. Sin embargo, se cree que la documentación de Neo4j no es precisa en cuanto al orden del tiempo de ejecución del algoritmo, dando a creer que debería ser más velóz.

V-C. Análisis de Tweets

Las consultas realizadas sobre la base de datos permiten observar temas de interés y costumbres que identifican a la cultura uruguaya. Los tópicos que involucran la mayoría de la conversación son naturalmente la Política y el Deporte, seguidos por usuarios que usan la red social como forma de vincularse, y en menor medida Agro y TV y Famosos. De los algoritmos de centralidad se entiende que *Page Rank Centrality* es el que permite capturar de mejor manera en la base de datos a las cuentas más influyentes en la conversación uruguaya, donde destacan las principales cadenas de noticias, las cuentas de Nacional y Peñarol, seguidas por cuentas de la actual Coalición Multicolor, incluyendo la del presidente de la república Luis Lacalle Pou. Los eventos del mes que generaron picos de conversación correspondieron al aliento de los principales clubes en el ámbito futbolístico, así como otros eventos internacionales a nivel deportivo, y dos sucesos climáticos destacables. Otro tema político que estuvo presente a lo largo de todo el mes fue la Marcha del Silencio.

Se entiende que las características de la base de datos, con un promedio de 13.000 *tweets* diarios (estimado en un $2-3\,\%$ del total en Uruguay), permiten utilizarla para responder de forma fiable preguntas de gran escala sobre la conversación de Twitter, lo cual es una fuente de información muy rica sobre la cultura e identidad del país.

V-D. Trabajos a futuro

Una posible línea de trabajo en el futuro sería extender la base de datos utilizando la propiedad *location* de los usuarios, la cual es un string de ingreso libre, completada por cerca de la mitad de los usuarios. Procesando estas entradas podrían extraerse usuarios uruguayos y ampliar la base con los tweets publicados por ellos. Una base de mayor tamaño permitiría validar las conclusiones ya obtenidas y profundizar el análisis de *performance* de Neo4j.

También se podría expandir el análisis realizado sobre la conversación de Twitter, utilizando distintas proyecciones del grafo o algoritmos que se agreguen a la Graph Data Science Library.

Por otro lado, podría resultar interesante repetir el trabajo sobre una base relacional y analizar si la misma mantiene o supera todas las ventajas observadas de Neo4j. Esta comparación podría ser útil para ayudar a justificar la adopción de los motores de bases de datos de grafos, en caso de que las bases relacionales no logren estar a la altura.

APÉNDICE

A. Estimación de la proporción de tweets capturados por la base

Para obtener una estimación de la proporción de *tweets* publicados con geolocalización activada sobre el total de los *tweets* publicados en Uruguay en el mes de mayo, se eligieron 3 *hashtags* entre los más usados en el mes y se comparó el volumen de *tweets* de la base con dichos *hashtags* versus el volumen total de *tweets* que los utilizaron. En la figura 12 se observan los diez *hashtags* más utilizados.

Se eligieron los *hashtags* `lamascarauy`, `marchadelsilencio2022` y `renunciaheber` para la comparación, teniendo en cuenta que es muy poco probable que hayan sido utilizados por fuera de Uruguay. Para obtener los valores totales para estos *hashtags* en mayo de 2022 se realizaron 3 nuevas descargas de *tweets* utilizando `twarc2`. En la figura 13 se ve como las proporciones para la base de datos construida varían entre $2\,\%$ y $3\,\%$, a partir de lo cual se estima que ese es el rango aproximado de captura de *tweets* de nuestra base.

	hashtag	hashtag_count
0	uruguay	623
1	lamascarauy	553
2	marchadelsilencio2022	528
3	peñarol	411
4	mayomesdelamemoria	390
5	todossomosfamiliares	276
6	bts	238
7	nacional	234
8	puntopenalenel10	231
9	renunciaheber	228

Figura 12: Los diez *hashtags* más usados en el mes.

	hashtag	db_hashtag_count	total_hashtag_count	db_proportion
0	marchadelsilencio2022	528	25866	0.020413
1	renunciaheber	228	9044	0.025210
2	lamascarauy	553	16966	0.032595

Figura 13: Comparación de *tweets* publicados capturados por la base versus los totales en mayo para tres *hashtags*.

REFERENCIAS

[1] Neo4j, "What is a Graph Database? - Developer Guides." [Online]. Available: https://neo4j.com/developer/graph-database/
[2] ——, "Graph Data Science," Jun 2022. [Online]. Available: https://neo4j.com/product/graph-data-science/
[3] L. Fura, "Analyzing Twitter Hashtag Impact using Neo4j, Python and JavaScript," Oct 2017. [Online]. Available: https://neo4j.com/blog/twitter-hashtag-impact-neo4j-python-javascript/
[4] Bear, "python-twitter," Oct 2018. [Online]. Available: https://github.com/bear/python-twitter
[5] Neo4j, "Neo4j Python Driver," Oct 2018. [Online]. Available: https://neo4j.com/docs/api/python-driver/current/?ref=blog
[6] M. Saengsuwan, "Using Neo4j Graph Database to Analyze Twitter Data," Apr 2021. [Online]. Available: https://towardsdatascience.com/using-neo4j-graph-database-to-analyze-twitter-data-6e3d38042af1
[7] M. Needham, "Finding influencers and communities in the Graph Community," May 2019. [Online]. Available: https://medium.com/neo4j/finding-influencers-and-communities-in-the-graph-community-e3d691296325
[8] Twitter, "Twitter API," Jun 2022. [Online]. Available: https://developer.twitter.com/en/docs/twitter-api
[9] twarc, "Twarc2 Library," Jun 2022. [Online]. Available: https://twarc-project.readthedocs.io/en/latest/twarc2_en_us/
[10] py2neo, "The Py2neo Handboook," May 2021. [Online]. Available: https://py2neo.org/2021.1/
[11] G. R. and L. J.P., "Repositorio del Proyecto: twitter-uy." [Online]. Available: https://github.com/rodrigallardo/twitter-uy
[12] Neo4j, "Betweenness Centrality - Graph Data Science," Jun 2022. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/algorithms/betweenness-centrality/
[13] Brandwatch, "The Most Influential Men and Women on Twitter 2017." [Online]. Available: https://www.brandwatch.com/blog/react-influential-men-and-women-2017/
[14] Neo4j, "Louvain Community Detection," Jun 2022. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/algorithms/louvain/