



Grafos Sociales en Neo4j

DRA. KARINA RUBY PÉREZ-DANIEL

FACULTAD DE INGENIERÍA
UNIVERSIDAD PANAMERICANA

ENERO - MARZO, 2020



UNIVERSIDAD
PANAMERICANA

CONTENIDO

- 1 LOS DATOS EN UNA SOCIAL NETWORK SERVICES (SNS)
- 2 LOS DATOS DE UN SNS Y LAS MÉTRICAS PARA SU ANÁLISIS
- 3 MÉTRICAS DE CENTRALIDAD
- 4 NEO4J & CYPHER
- 5 CENTRALIDAD EN UN GRAFO EN NEO4J
- 6 TWITTER API



SECCIÓN 1 | LOS DATOS EN UNA SOCIAL NETWORK SERVICES (SNS)



PROPIEDADES DE LOS SNSs

De acuerdo a Danah Boyd (MSR), los SNSs están definidos por varias propiedades, tales como

- Perfil de usuario,
- Lista de contactos,
- Navegación entre contactos,
- Muro de contenidos,
- Sistema de recomendaciones
- Servicios típicos en una Red Social
 - Servicio de mensajería
 - Comunidades en línea
- Proveedor de Identidad Global



FUENTES DE INFORMACIÓN PERSONAL EN UN SNS

De acuerdo a la OCDE y al WEF, la información personal obtenida puede ser:

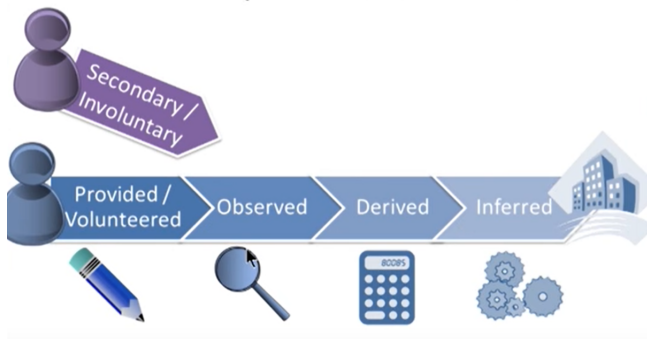


Figura 1. Tipos de información en un SNS



FUENTES DE INFORMACIÓN PERSONAL EN UN SNS

De acuerdo a la OCDE y al WEF, la información personal obtenida puede ser:

- **Proporcionada Voluntariamente:** Mediante el llenado de formularios al darse de alta en el SNS, la publicaciones, posts, comentarios, likes, RTs, etc.
- **Observada:** Los SNS sirven de intermediario en la transacción de redireccionamiento en un enlace compartido a través de un SNS, o al iniciar sesión en algún sitio usando como método de autenticación nuestra “identidad” en un SNS.
- **Información derivada:** Esta información se obtiene de la derivación directa de nuestros hábitos de navegación, y no requiere de cálculos complejos, ni de un análisis exhaustivo para ser obtenida.



FUENTES DE INFORMACIÓN PERSONAL EN UN SNS (CONTINUACIÓN)

- **Datos inferidos:** Se refieren a los datos obtenidos mediante técnicas estadísticas o de ML y que requieren de comparación con otros usuarios.
- **Datos secundarios:** Estos datos se obtienen mediante otros usuarios, al etiquetarnos en sus publicaciones.



TIPOS DE DATO EN UN SNS

Datos del usuario

- Nombre,
- Información personal,
 - ¿Verificada?
- Datos demográficos,
- Subscripciones, intereses,
- Relevancia,
- Relaciones,
- Contenido publicado,

Metadatos (Explícitos)

- **Entidades**: hashtags, símbolos, links, media, menciones, etc.,
- Usuarios relacionados y relaciones entre ellos,
- Temas, sentimiento/polaridad, sensibilidad,
- Geográficos,
- Conversación,
- **Alcance/Reach**: favoritos/
likes, veces que fue leído,
trending, alcance.

SECCIÓN 2 | LOS DATOS DE UN SNS Y LAS MÉTRICAS PARA SU ANÁLISIS



ACCEDER A LOS DATOS DE UNA SNS

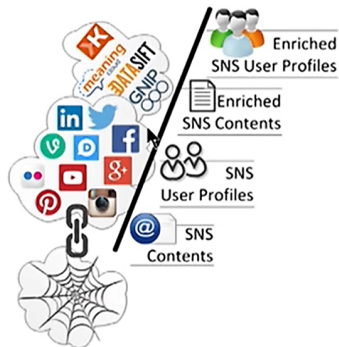


Figura 2. Acceso a Datos de un SNS

Métodos para acceder a los datos un SNS

- Content scraping.
- API publica
- APIs Enriquecidas de terceros,
- APIs de socios del SNS,
- Autorización(OAuth).
- Contenidos externos.
- Interfaces de ML
 - Internos
 - Delegados



ANÁLISIS DE CONTENIDO

- Anotación manual de la incidencia de “conceptos”, “categorías” o “relaciones” es un texto dado.
- Problemas:
 - Validación: Correspondencia, generalización
 - Confiabilidad: Estabilidad, ¿es reproducible, preciso?,

Krippendorff's alpha:

$$\alpha = 1 - \frac{D_o}{D_e} = 1 - \frac{\sum_{u=1}^N \frac{m_u}{n} D_u}{D_e} \quad (1)$$

- D_o = Discrepancia observada
- D_e = Discrepancia esperada

Se tiene que hacer por múltiples usuarios para observar patrones.



MÉTRICAS PARA EL ANÁLISIS DE LA RED

● Medidas para la Centralidad local o posición:

- Degree
- Betweenness
- Closeness
- eigenvector
- Page Rank (based on eigenvector)

● Medidas de cohesión global

- Clustering coefficient
- Path length distribution

● Redes libres de escala y mundos pequeños

● Detección de comunidad

- Clique overlap
- Simulated annealing
- Edge betweenness
- Modularidad

● Homofilía y proximidad



SECCIÓN 3

MÉTRICAS DE CENTRALIDAD



MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Las medidas de centralidad responden la siguiente pregunta
¿Cuál es el elemento más importante en la red?

Hay muchas respuestas a esta pregunta, y eso depende de a que nos referimos con **importante**. Hay diversas métricas de centralidad, tales como: degree, betweenness, closeness, eigenvector, page rank, etc.



MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Degree (Grado)

- **Definición:** El nodo con más alto grado es el central, es decir, asigna el grado de acuerdo al número de enlaces de cada nodo.
- **¿Qué nos dice?:** ¿Cuántas relaciones directas tiene cada nodo en la red?
- **¿Cuándo usarlo?:** Para encontrar individuos muy conectados/populares, o individuos que probablemente posean mayor información, o que puedan conectar fácilmente con varios miembros de la red.
- **Más detalles:** La premisa es que los nodos de grado más alto, son los más importantes y por lo tanto los **más influyentes**. Tiene 2 variantes, el **in-degree** → conexiones de entrada, y el **out-degree** → conexiones de salida.



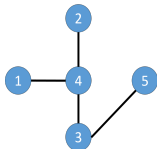
MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Degree (Grado)

$$C(n) = \frac{\sum_{j=1}^N a_{nj}}{N - 1} \quad (2)$$

donde a_{nj} son los caminos directos entre n y j ; y N es el número total de nodos.

Consideraciones iniciales: ¿Es dirigido el grafo?



¿Cómo
obtengo
degree?

# Nodo	1	2	3	4	5	Suma	N-1	Degree
1		0	0	1	0	1	4	1/4
2	0		0	1	0	1	4	1/4
3	0	0		1	1	2	4	2/4
4	1	1	1		0	3	4	3/4
5	0	0	1	0		1	4	1/4



MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Betweenness (Centralidad de intermediación)

- **Definición:** El número de veces que un nodo se encuentra en el camino más corto entre otros nodos.
- **¿Qué nos dice?:** ¿Qué nodos sirven como “puentes” entre nodos de una red?, es decir, cuantos mas nodos necesiten pasar por un nodo para hacer sus conexiones indirectas por los caminos más cortos, más central será el nodo.
- **¿Cuándo usarlo?:** Para encontrar individuos con mayor influencia en el flujo de un sistema.
- **Más detalles:** Es muy útil para analizar las dinámicas de comunicación. Sin embargo, un alto betweenness puede significar ya sea que un nodo influye sobre otros grupos, o que simplemente se encuentra en la periferia de ambos clusters.



MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Betweenness (Centralidad de intermediación)

$$B(n) = \sum_{s \neq n \neq t \in N} \frac{\sigma_{st}(n)}{\sigma_{st}} \quad (3)$$

donde σ_{st} es el número total de caminos más cercanos entre los nodos s y t ; y $\sigma_{st}(n)$ es el número de esos caminos que pasan por el nodo n .

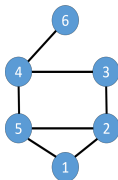
Consideraciones iniciales:

- ¿Es dirigido el grafo?
- $N = \#$ de nodos
- #Caminos Potenciales = $\frac{N(N-1)}{2} \rightarrow$ No dirigido
- #Caminos Potenciales = $N(N-1) \rightarrow$ Dirigido
- Factor de normalización = $\frac{(N-1)(N-2)}{2} \rightarrow$ No dirigido
- Factor de normalización = $(N-1)(N-2) \rightarrow$ Dirigido



MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Betweenness (Centralidad de intermediación)



¿Cómo obtengo
betweenness?

$$\begin{aligned} &\# \text{Caminos Potenciales} \\ &= \frac{N(N-1)}{2} = \frac{6(5)}{2} = 15 \end{aligned}$$

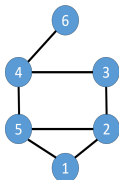
$$\begin{aligned} &\text{Factor de} \\ &\text{normalización} = \\ &\frac{(N-1)(N-2)}{2} = \frac{5(4)}{2} = 10 \end{aligned}$$

Nodo	<i>B</i>
1	0
2	1.5
3	1
4	4.5
5	3
6	0



MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Betweenness (Centralidad de intermediación)



¿Cómo
obtengo
betweenness?

#	De - A	Geodésico	Nodos					
			1	2	3	4	5	6
1	(1,2)	1,2	0	0	0	0	0	0
2	(1,3)	1,2,3	0	1	0	0	0	0
3	(1,4)	1,5,4	0	0	0	0	1	0
4	(1,5)	1,5	0	0	0	0	0	0
5	(1,6)	1,5,4,6	0	0	0	1	1	0
6	(2,3)	2,3	0	0	0	0	0	0
7	(2,4)	2,3,4	0	0	1	0	0	0
8	(2,5)	2,5	0	0	0	0	0	0
9	(2,6)	2,3,4,6 2,5,4,6	0	0	0	1	1	0
10	(3,4)	3,4	0	0	0	0	0	0
11	(3,5)	3,2,5 3,4,5	0	0.5	0	0.5	0	0
12	(3,6)	3,4,6	0	0	0	1	0	0
13	(4,5)	4,5	0	0	0	0	0	0
14	(4,6)	4,6	0	0	0	0	0	0
15	(5,6)	5,4,6	0	0	0	1	0	0
Betweenness			0	1.5	1	4.5	3	0
Factor de normalización			10	10	10	10	10	10

MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Closeness (Cercanía)

- **Definición:** Mide a cada nodo basado en la *cercanía* que tiene con otros nodos.
- **¿Qué nos dice?:** Calcula el camino más cercano entre todos los nodos y les asigna una puntuación basada en la suma de todos los caminos más cercanos.
- **¿Cuándo usarlo?:** Cuando se quiere encontrar a aquellos nodos mejor ubicados para influenciar a toda la red de forma más rápida.
- **Más detalles:** Puede ayudar a encontrar a aquellos nodos que puedan servir de *comunicadores* efectivos. Sin embargo, si la red está altamente conectada, es probable que todos los nodos tengan una puntuación similar. Es efectivo para encontrar *influencers* en un único cluster.

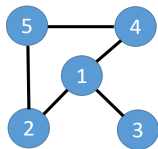


MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Closeness (Cercanía)

$$Closeness(n) = \frac{1}{\sum_{i \neq n \in N} d_{ni}} \quad (4)$$

donde d_{ni} es la distancia del camino más cercano entre los nodos i y n . Por lo tanto, *closeness* representa el promedio de la distancia de los caminos más cercanos a todos los nodos del grafo.



¿Cómo
obtengo
closeness?

# Nodo	1	2	3	4	5	Suma	N-1	Closeness
1		1	1	1	2	5	4	4/5
2	1		2	2	1	6	4	4/6
3	1	2		2	3	8	4	4/8
4	1	2	2		1	6	4	4/6
5	2	1	3	1		7	4	4/7



MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

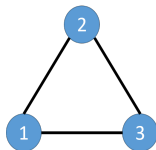
Eigenvalores

- **Definición:** Al igual que el grado, los *eigenvalores* miden la influencia de un nodo basado en el número de *links* que tiene con otros nodos en la red. Sin embargo, los *eigenvalores* también consideran qué tan bien está conectado un nodo y cuántos links tienen sus vecinos.
- **¿Qué nos dice?:** Al calcular las conexiones extendidas de un nodo, los *eigenvalores* pueden identificar nodos con influencia sobre toda la red, no únicamente la influencia directa.
- **¿Cuándo usarlo?:** Es una medida que da una visión general de los nodos de la red,.
- **Más detalles:** *Page Rank* es una variante de los *Eigenvalores*.



MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Eigenvalores Consideraciones iniciales: ¿Es dirigido el grafo?




¿Cómo
obtengo
eigenvalue?

# Nodo	1	2	3
1		1	1
2	1		1
3	1	1	

$$\rightarrow \det \begin{vmatrix} 0 - \lambda & 1 & 1 \\ 1 & 0 - \lambda & 1 \\ 1 & 1 & 0 - \lambda \end{vmatrix}$$

$$\det = -\lambda \begin{vmatrix} -\lambda & 1 \\ 1 & -\lambda \end{vmatrix} - 1 \begin{vmatrix} 1 & 1 \\ 1 & -\lambda \end{vmatrix} + 1 \begin{vmatrix} 1 & -\lambda \\ 1 & 1 \end{vmatrix}$$

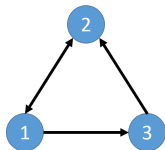
$$\begin{aligned} \det &= -\lambda(\lambda^2 - 1) - 1(-\lambda - 1) + 1(1 + \lambda) \\ &= -\lambda^3 + \lambda + \lambda + 1 + 1 + \lambda \\ &= -\lambda^3 + 3\lambda + 2 = -(\lambda + 1)(\lambda + 1)(\lambda - 2) \dots \end{aligned}$$

$$E = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$


UNIVERSIDAD
PANAMERICANA

MEDIDAS PARA LA CENTRALIDAD LOCAL O POSICIÓN

Eigenvalores Consideraciones iniciales: ¿Es dirigido el grafo?



¿Cómo
obtengo
eigenvalue?

Hasta	Desde		
	1	2	3
1		1	0
2	1		1
3	1	0	

$$\rightarrow \det \begin{vmatrix} -\lambda & 1 & 0 \\ 1 & -\lambda & 1 \\ 1 & 0 & -\lambda \end{vmatrix}$$

$$E = \begin{pmatrix} 1.325 \\ 1.755 \\ 1 \end{pmatrix}$$

- Está métrica ayuda a determinar quién tiene **mayor influencia** sobre la red, así como
- Quién es importante en mi red en un contexto de **macro escala**



RESUMEN

¿Qué me dice cada métrica de centralidad?.

Resumen

- **Degree/Grado:** ¿Qué tanta exposición tiene un nodo en la red?
¿Qué oportunidad tiene para influir directamente en la red?
- **Betweenness/Intermediación:** Potencial informal, control en el flujo, importancia de un nodo como puente entre grupos.
- **Closeness/Cercanía:** Estima qué tanto tomará difundir información desde un nodo a todos los demás nodos de forma secuencial.
- **Eigenvalores:** Determina qué tan bien está conectado un nodo a los nodos influyentes de algo *grado*, es decir, lo más importante no es qué sabes sino a quién conoces.

SECCIÓN 4

NEO4J & CYPHER



LOS GRAFOS EN EL MERCADO: FB



```

MATCH (person:Person)-[:IS_FRIEND_OF]->(friend),
         (friend)-[:LIKES]->(restaurant),
         (restaurant)-[:LOCATED_IN]->(loc:Location),
         (restaurant)-[:SERVES]->(type:Cuisine)
WHERE person.name = 'Philip'
AND loc.location='New York'
AND type.cuisine='Sushi'
RETURN restaurant.name
  
```

Figura 3. Consulta en una Red Social



CONSULTAS EN NEO4J

Componentes de una consulta en Neo4j.

MATCH

```
MATCH (m:Pelicula)
RETURN m
```

- MATCH y RETURN son palabras reservadas de Cypher.
- m es una variable
- :Pelicula es la etiqueta de un nodo.

```
MATCH (p:Persona)-[r:ACTUA_EN]->(m:Pelicula)
RETURN p,r,m
```

- MATCH y RETURN son palabras reservadas de Cypher.
- p, r, m son variables.
- :ACTUA_EN es el tipo de relación.



CONSULTAS EN NEO4J

Componentes de una consulta en Neo4j.

```
MATCH path = (:Persona)-[ACTUA_EN]->(:Pelicula)
RETURN path
```

- MATCH y RETURN son palabras reservadas de Cypher.
- path es una variable
- :Persona y :Pelicula son etiqueta de nodos.
- :ACTUA_EN es el tipo de relación.



COMANDOS EN CYPHER PARA ACCEDER A UN GRAFO

- **MATCH**: Se usa para establecer el patrón de coincidencia en la búsqueda.
- **WHERE**: Es complementario a **MATCH**.
- **WITH**: Agrega restricciones a un patrón o filtro para la búsqueda
- **RETURN**: Retorno.



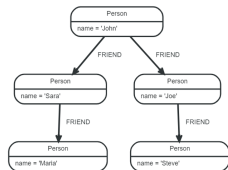
COMANDOS EN CYPHER PARA MODIFICAR UN GRAFO

- **CREATE** y **DELETE**: Crea y borrar nodos y/o relaciones.
- **SET** y **REMOVE**: Agrega o retira propiedades y etiquetas de los nodos.
- **MERGE**: Empareja nodos existentes o crea nuevos nodos. Suele utilizarse con la restricción **UNIQUE**.



CREANDO UN GRAFO EN CYPHER

```
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)
```



Consulta 1

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

Figura 4. Ejemplo de un Grafo

John.name	fof.name
"John"	"Maria"
"John"	"Steve"

Consulta 2

```
MATCH (user)-[:FRIEND]->(follower)
WHERE user.name IN ['Joe', 'John', 'Sara', 'Maria', 'Steve']
AND follower.name = 'S.*'
RETURN user.name, follower.name
```

user.name	follower.name
"Joe"	"Steve"
"John"	"Sara"

CONSULTAS EN CYPHER

En Cypher es posible leer, match o hacer actualizaciones, pero no de forma simultánea.

Cypher no devuelve ningún match hasta que se le indique lo contrario.

Consulta 3

```
MATCH (n name: 'John')-[:FRIEND]-(friend)
WITH n, count(friend) AS friendsCount
WHERE friendsCount > 3
RETURN n, friendsCount
```

Displaying 0 nodes, 0 relationships.

Consulta 4

```
MATCH (n name: 'John')-[:FRIEND]-(friend)
WITH n, count(friend) AS friendsCount
SET n.friendsCount = friendsCount
RETURN n.friendsCount
```

n.friendsCount
2



UNIQUENESS EN CYPHER

Cypher se asegura de que la misma relación no sea devuelta más de una vez in la misma consulta.

Demostrando Uniqueness

```
CREATE (adam:User name: 'Adam' ),(pernilla:User name:
'Pernilla' ),(david:User name: 'David' ),

(adam)-[:FRIEND]->(pernilla), (pernilla)-[:FRIEND]->(david)
```

Consulta 1

```
MATCH (user:User name: 'Adam'
)-[r1:FRIEND]-()-[r2:FRIEND]-(friend_of_a_friend)

RETURN friend_of_a_friend.name AS fofName
```

Consulta 2

```
MATCH (user:User name: 'Adam' )-[r1:FRIEND]-(friend)
MATCH (friend)-[r2:FRIEND]-(friend_of_a_friend)
RETURN friend_of_a_friend.name AS fofName
```

Consulta 3

```
MATCH (user:User name: 'Adam'
)-[r1:FRIEND]-(friend), (friend)-[r2:FRIEND]-(friend_of_a_friend)

RETURN friend_of_a_friend.name AS fofName
```

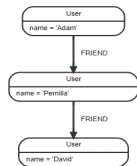


Figura 5. Ejemplo de un Grafo

fofName
"David"

fofName
"David"
"Adam"

fofName



EXPRESIONES

Expresiones

- Cypher puede manejar expresiones de comparación entre múltiples valores.

Consulta

```
CASE test
WHEN value THEN result
WHEN ...
ELSE default
END
```



EXPRESIONES EN GRAFOS

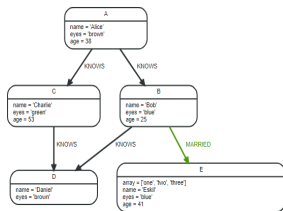


Figura 6. Ejemplo de un Grafo

Consulta 1

```

MATCH (n)
RETURN
CASE n.eyes
WHEN 'blue'
THEN 1
WHEN 'brown'
THEN 2
ELSE 3 END AS result

```

Consulta 2

```

MATCH (n)
RETURN
CASE
WHEN n.eyes = 'blue'
THEN 1
WHEN n.age < 40
THEN 2
ELSE 3 END AS result

```



EXPRESIONES EN GRAFOS

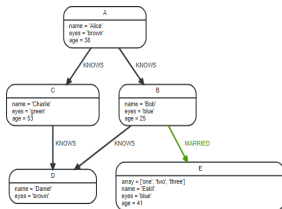


Figura 7. Ejemplo de un Grafo

Consulta 3

```

MATCH (n)
RETURN n.name,
CASE n.age
WHEN n.age IS NULL THEN -1
ELSE n.age END AS current_age
  
```

Consulta 4

```

MATCH (n)
RETURN n.name,
CASE
WHEN n.age IS NULL THEN -1
ELSE n.age END AS current_age
  
```



ACTUALIZAR UN NODO

- **Setting properties**

- **SET**: Se usa para actualizar las etiquetas, propiedades y relaciones en nodos.

- ```
MATCH (n{ name: 'Bob' })
SET n.surname = 'Taylor'
RETURN n.name, n.surname
```

- ```
MATCH (n{ name: 'Bob' }) SET (  
CASE n.age  
WHEN '36'  
THEN n END ).worksIn = 'UP'  
RETURN n.name, n.worksIn
```



ACTUALIZAR UN NODO

- **Actualizar una propiedad**

- ```
MATCH (n { name: 'Bob' })
 SET n.age = toString(n.age)
 RETURN n.name, n.age
 DETACH DELETE n
```

- **Borra una propiedad**

- ```
MATCH (n { name: 'Andy' })  
  SET n.name = NULL RETURN n.name, n.age
```



MERGE NODES

- **Merge a single node with a label**

- `MERGE (robert:Critic)`
`RETURN robert, labels(robert)`
Si no existe el nodo, se crea uno nuevo

- **Merge a single node with properties**

- `MERGE (charlie{ name: 'Charlie Sheen', age: 10})`
`RETURN charlie`

- **Merge a single node with labels and properties**

- `MERGE (michael:Person{ name: 'Michael Douglas'})`
`RETURN michael.name, michael.bornIn`



SECCIÓN 5

CENTRALIDAD EN UN GRAFO EN

NEO4J



DEGREE CENTRALITY

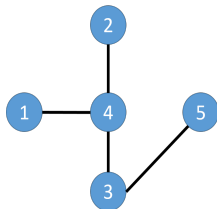


Figura 8. Ejemplo de un Grafo

CREATE

```
(uno:Number{id:"1"}),
(dos:Number{id:"2"}),
(tres:Number{id:"3"}),
(cuatro:Number{id:"4"}),
(cinco:Number{id:"5"})
```

CREATE

```
(uno)-[:CONECTA]->(cuatro),
(dos)-[:CONECTA]->(cuatro),
(tres)-[:CONECTA]->(cuatro),
(cinco)-[:CONECTA]->(tres)
```

Consultando el Grado

```
MATCH (n:Number)
RETURN n.id AS NAME,
size((n)-[:CONECTA]->()) AS outDegree,
size((n)-[:CONECTA]<-()) AS inDegree,
size((n)-[:CONECTA]-()) AS Degree
```



BETWEENNESS CENTRALITY

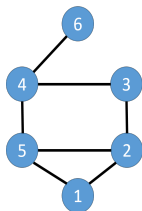


Figura 9. Ejemplo de un Grafo

CREATE

```
(uno:Number{id:"1"}),
(dos:Number{id:"2"}),
(tres:Number{id:"3"}),
(cuatro:Number{id:"4"}),
(cinco:Number{id:"5"}),
(seis:Number{id:"6"})
```

CREATE

```
(uno)-[:CONECTA]->(dos),
(uno)-[:CONECTA]->(cinco),
(dos)-[:CONECTA]->(cinco),
(dos)-[:CONECTA]->(tres),
(tres)-[:CONECTA]->(cuatro),
(cinco)-[:CONECTA]->(cuatro),
(cuatro)-[:CONECTA]->(seis)
```

Consultando Betweenness Centrality

```
CALL algo.betweenness.stream("Number",
"CONECTA", direction:"out")
YIELD nodeId, centrality
MATCH (number:Number) WHERE id(number) = nodeId
RETURN number.id AS user, centrality
ORDER BY centrality DESC;
```



CLOSENESS CENTRALITY

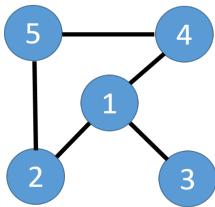


Figura 10. Ejemplo de un Grafo

CREATE

```
(uno:Number{id:"1"}),
(dos:Number{id:"2"}),
(tres:Number{id:"3"}),
(cuatro:Number{id:"4"}),
(cinco:Number{id:"5"})
```

CREATE

```
(uno)-[:CONECTA]->(dos),
(unos)-[:CONECTA]->(cuatro),
(unos)-[:CONECTA]->(tres),
(dos)-[:CONECTA]->(cinco),
(cuatro)-[:CONECTA]->(cinco)
```

Consultando Closeness Centrality

```
CALL algo.closeness.stream("Number",
"CONECTA")
YIELD nodeId, centrality
MATCH (n:Number) WHERE id(n) = nodeId
RETURN n.id AS node, centrality
ORDER BY centrality DESC
LIMIT 3;
```



EIGENVALUE CENTRALITY

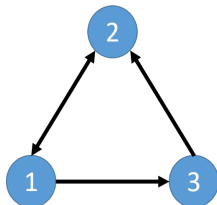


Figura 11. Ejemplo de un Grafo

CREATE

```
(uno:Number{id:"1"}),
(dos:Number{id:"2"}),
(tres:Number{id:"3"})
```

CREATE

```
(uno)-[:CONECTA]->(dos),
(uno)-[:CONECTA]->(tres),
(dos)-[:CONECTA]->(uno),
(tres)-[:CONECTA]->(dos)
```

Consultando Eigenvalue Centrality

```
CALL algo.eigenvector.stream("Number",
"CONECTA", {normalization: "max"})
YIELD nodeId, score
RETURN algo.asNode(nodeId).id AS
number, score
ORDER BY score DESC
```



SECCIÓN 6

TWITTER API



TWITTER API

- **Es necesario contar con una cuenta developer**
- Crear un app para obtener datos
- Instalar tweepy y py2neo



CREAR UNA APP EN TWITTER PARA LA OBTENCIÓN DE DATOS

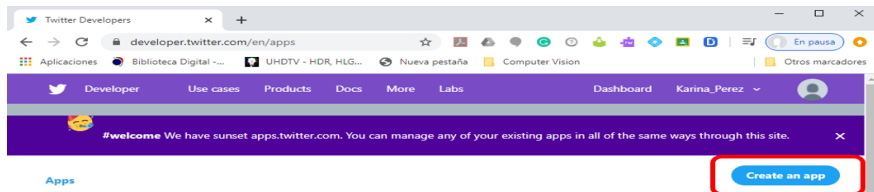


Figura 12. Paso 1



CREAR UNA APP EN TWITTER PARA LA OBTENCIÓN DE DATOS

Client applications — Twitter Developer

developer.twitter.com/en/apps/create

Aplicaciones Biblioteca Digital ... UHDTV - HDR, HLG... Nueva pestaña Computer Vision Otros marcadores

En pausa

Apps Create an app

Readiness: We have issued application codes. You can manage any of your existing apps in all of the same ways through this site.

App details

This information is only visible to you and your organization. It is not visible to the public.

App name (required) Maximum characters: 32

App description (required) Maximum characters: 512

Website URL (required) Between 10 and 255 characters

Callback URL (required) Between 10 and 255 characters

Terms of Service URL (required) Between 10 and 255 characters

Primary testing URL (required) Between 10 and 255 characters

Organization name (required) Between 10 and 255 characters

Organization website URL (required) Between 10 and 255 characters

Tell us how this app will be used (required) Maximum characters: 512

Cancel Create

Figura 13. Paso 2



CREAR UNA APP EN TWITTER PARA LA OBTENCIÓN DE DATOS

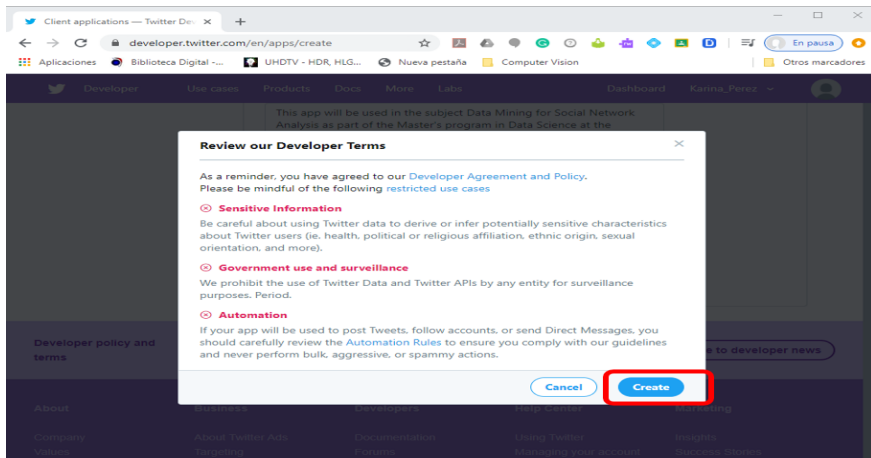


Figura 14. Paso 3



CREAR UNA APP EN TWITTER PARA LA OBTENCIÓN DE DATOS

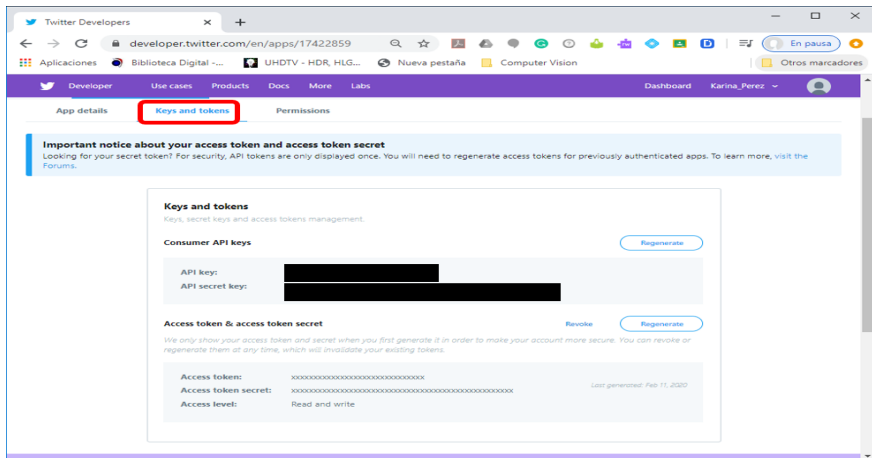


Figura 15. Paso 4

Gracias por su atención

Fin



UNIVERSIDAD
PANAMERICANA