



# **Neo4j: Medidas de centralidad**

**Big Data Aplicado**

Julio Alberto López Gómez  
JulioAlberto.Lopez@uclm.es



**Big\_Data**  
**Aplicado**



Curso Especialización  
Inteligencia\_Artificial y  
Big\_Data

# Centralidad

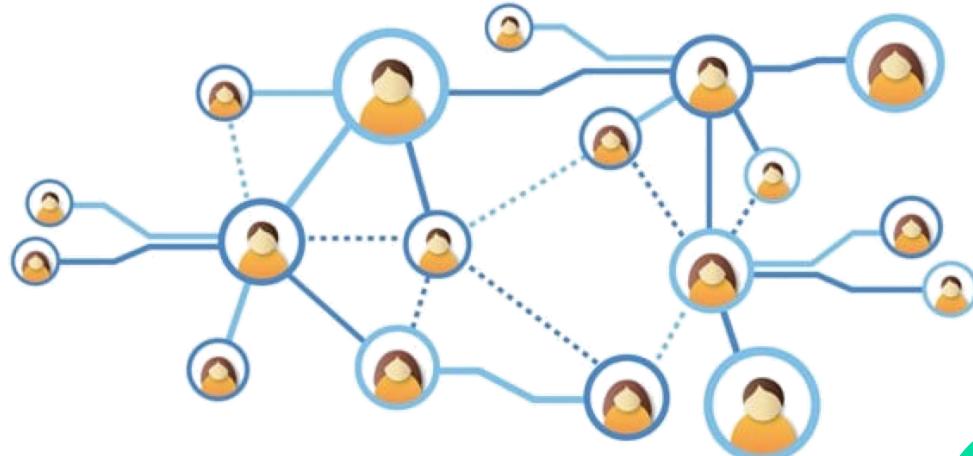
## Concepto:

- La centralidad permite identificar la relevancia de un nodo dentro de un grafo
- Identifican nodos relevantes o importantes
- Permiten entender el comportamiento de la red
  - Credibilidad
  - Accesibilidad
  - Velocidad

# Centralidad

## Aplicaciones:

- Publicidad/Marketing
  - Encontrar un actor relevante dentro de la red para anunciar un producto
  - Identificar actores relevantes para, a través de ellos, alcanzar a más usuarios



# Centralidad

## Medidas de Centralidad:

- No existe una única medida de centralidad
- En función del problema, se elegirá la medida que se considere más adecuada
- Algunas medidas de centralidad:
  - Centralidad de grado
  - Cercanía
  - Intermediación
  - ...

# Centralidad

## Centralidad de grado:

- Grado: Número de conexiones directas de un nodo.
  - Grado de entrada: Número de conexiones que entran a un nodo (Prominencia)
  - Grado de salida: Número de conexiones que salen de un nodo (influencia)
- Aplicaciones:
  - Identificar actores relevantes en una red social
  - Identificar conductas fraudulentas

# Centralidad

## Centralidad de grado:

- Creación de un grafo social.
  - Creación de nodos

CREATE

```
(alice:User {name: 'Alice'}) ,  
(bridget:User {name: 'Bridget'}) ,  
(charles:User {name: 'Charles'}) ,  
(doug:User {name: 'Doug'}) ,  
(mark:User {name: 'Mark'}) ,  
(michael:User {name: 'Michael'}) ,
```

# Centralidad

## Centralidad de grado:

- Creación de un grafo social.
- Creación de relaciones

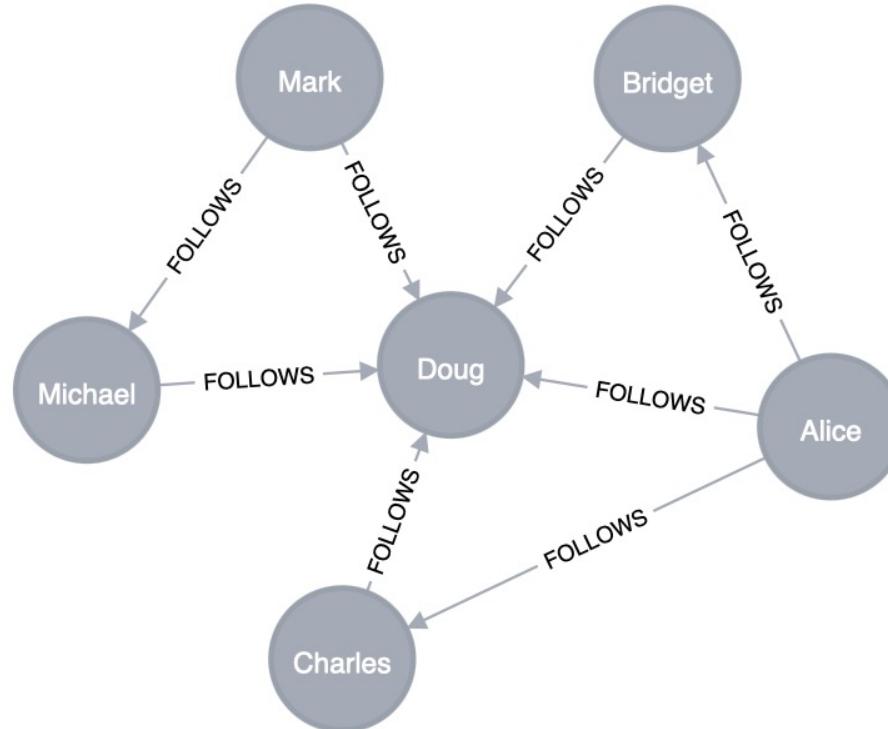
```
(alice) -[:FOLLOWS {score: 1}]->(doug) ,  
    (alice) -[:FOLLOWS {score: -2}]->(bridget) ,  
        (alice) -[:FOLLOWS {score: 5}]->(charles) ,  
            (mark) -[:FOLLOWS {score: 1.5}]->(doug) ,  
                (mark) -[:FOLLOWS {score: 4.5}]->(michael) ,  
                    (bridget) -[:FOLLOWS {score: 1.5}]->(doug) ,  
                        (charles) -[:FOLLOWS {score: 2}]->(doug) ,  
                            (michael) -[:FOLLOWS {score: 1.5}]->(doug)
```



# Centralidad

Centralidad de grado:

- Creación de un grafo social.



# Centralidad

## Centralidad de grado:

- Almacenamiento del grafo

```
CALL gds.graph.create(
  'myGraph',
  'User',
  {
    FOLLOWS: {
      orientation: 'REVERSE',
      properties: ['score']
    }
  }
)
```



# Centralidad

## Centralidad de grado:

- Cálculo de la centralidad de grado

```
CALL gds.degree.stream('myGraph')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).name AS name,
      score AS followers
ORDER BY followers DESC, name DESC
```



# Centralidad

## Centralidad de grado:

### ■ Resultado

	name	followers
1	"Doug"	5.0
2	"Michael"	1.0
3	"Charles"	1.0
4	"Bridget"	1.0
5	"Mark"	0.0
6	"Alice"	0.0

# Centralidad

## Cercanía:

- Identifica nodos que expanden rápida y eficientemente la información a través de la red.
- Se calcula como la suma del inverso de las distancias de un nodo al resto.
- $$C(u) = \frac{1}{\sum_{v=1}^{n-1} d(u,v)}$$
  - u es un nodo
  - n es el número de nodos del grafo
  - $d(u,v)$  camino mínimo entre u y v

# Centralidad

## Cercanía:

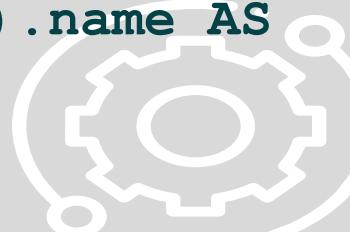
- Aplicaciones:
  - Descubrir actores en posiciones privilegiadas de adquirir información y recursos importantes
  - Heurística para estimar tiempos de llegada en logística
  - Minería de textos, importancia o prominencia de las palabras en un documento

# Centralidad

- Cálculo de la cercanía

## Cercanía:

```
CALL gds.alpha.closeness.stream({  
    nodeProjection: 'User',  
    relationshipProjection: 'FOLLOWS'  
})  
  
YIELD nodeId, centrality  
  
RETURN gds.util.asNode(nodeId).name AS user, centrality  
  
ORDER BY centrality DESC
```



# Centralidad

Cercanía:

## ■ Resultado

	user	centrality
1	"Doug"	1.0
2	"Alice"	0.7142857142857143
3	"Bridget"	0.625
4	"Charles"	0.625
5	"Mark"	0.625
6	"Michael"	0.625

# Centralidad

## Intermediación:

- Identifica "al actor que está en medio"
- Detecta la influencia de un actor en el flujo de información o recursos de un grafo
- ¿Quién es el actor que hace puente entre unas porciones y otras del grafo?
- Un nodo tiene mayor intermediación si está en los caminos mínimos de otros pares de nodos

# Centralidad

## Intermediación:

- $B(u) = \sum_{s \neq u \neq t} \frac{p(u)}{p}$ 
  - u es un nodo
  - p es el número total de caminos mínimos entre s y t
- Aplicaciones:
  - Identificación de influencers en una organización
  - Viralización de tweets
  - Puentes en redes eléctricas

# Centralidad

## Intermediación:

- Cálculo de la intermediación:

```
CALL gds.betweenness.stream('myGraph')  
YIELD nodeId, score  
RETURN gds.util.asNode(nodeId).name AS name, score  
ORDER BY name ASC
```



# Centralidad

## Intermediación:

- Resultado:



	name	score
1	"Michael"	0.0
2	"Mark"	0.0
3	"Doug"	0.0
4	"Charles"	0.0
5	"Bridget"	0.0
6	"Alice"	0.0



# **Neo4j: Medidas de centralidad**

**Big Data Aplicado**

Julio Alberto López Gómez  
JulioAlberto.Lopez@uclm.es



**Big\_Data**  
Aplicado



Curso Especialización  
Inteligencia\_Artificial y  
Big\_Data