



**ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

---

**Laboratorio de:** ANALÍTICA DE DATOS – BIG DATA

**Práctica No.:** 4

**Tema:** Neo4j

**Objetivos:**

- Instalar Neo4j y realizar 3 nodos con sus relaciones.
- Usar una interfaz gráfica para crear nodos de manera más sencilla.
- Utilizar los grafos de GameOfThrones para practicar consultas.

**Marco teórico:**

**Grafos**

Un grafo se define como cualquier representación gráfica formada por vértices (se ilustran mediante círculos) y aristas (se muestran mediante líneas de intersección). Dentro de estas representaciones gráficas, tenemos varios tipos de grafos:

- Grafos no dirigidos: los nodos y las relaciones son intercambiables, su relación se puede interpretar en cualquier sentido. Las relaciones de amistad en la red social Facebook, por ejemplo, son de este tipo.
- Grafos dirigidos: los nodos y las relaciones no son bidireccionales por defecto. Las relaciones en Twitter son de este tipo. Un usuario puede seguir a determinados perfiles en esta red social sin que ellos le sigan a él.
- Grafos con peso: en este tipo de gráficas las relaciones entre nodos tienen algún tipo de valoración numérica. Eso permite luego hacer operaciones.

**BDOG**

Las bases de datos orientadas a grafos (BDOG) ayudan a encontrar relaciones entre los datos y extraer su verdadero valor.

**Neo4j**

Neo4j usa grafos para representar datos y las relaciones entre ellos.

Neo4j utiliza grafos de propiedad para extraer valor añadido de los datos de cualquier empresa con gran rendimiento y de una forma ágil, flexible y escalable.

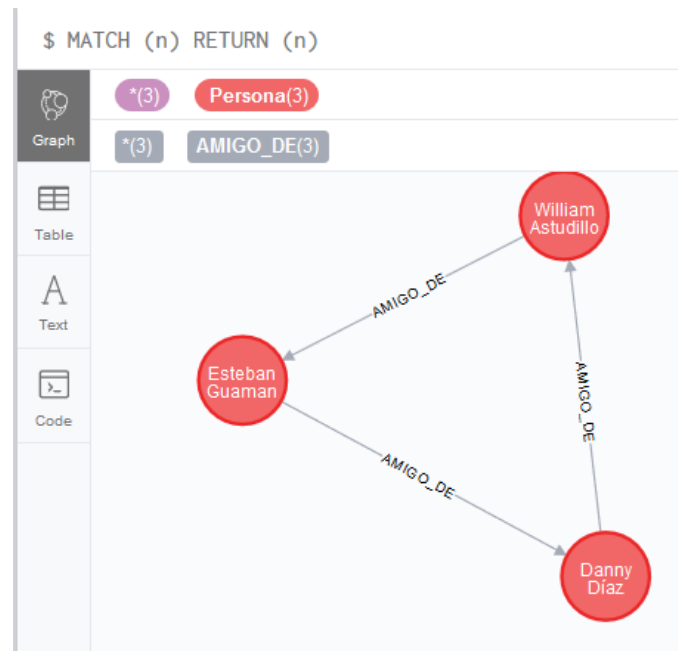


ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

## Desarrollo de la práctica:

### 1. Creación de nodos y relaciones básicas

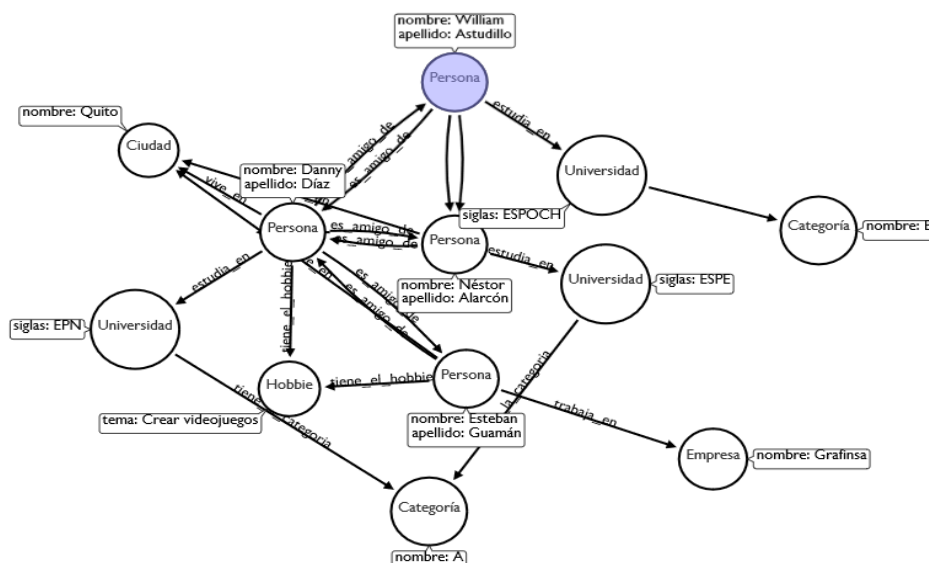
Como introducción al programa se creó tres simples nodos con la instrucción CREATE y las relaciones con la estructura (a)-[r]->(b).



### 2. Creación de varios nodos usando la interfaz gráfica de

<http://www.apciones.com/arrows/#>

Arrastrando nodos y relaciones la gráfica es la siguiente:



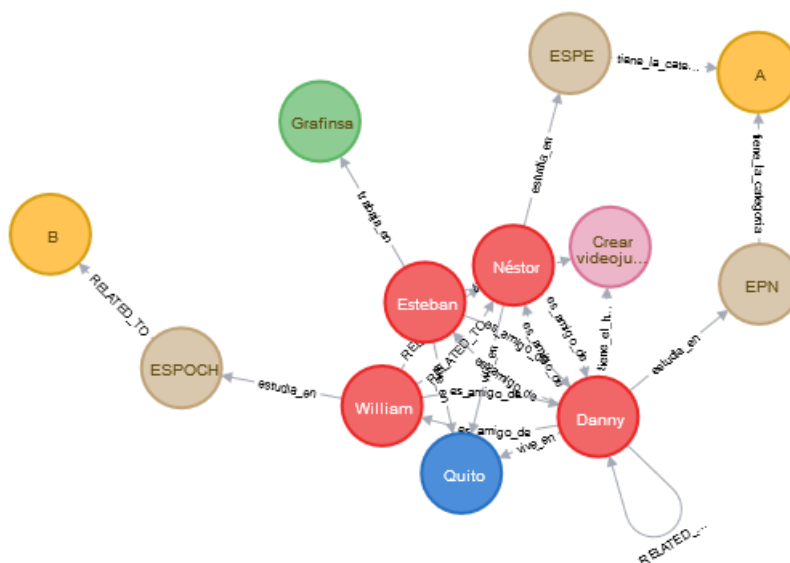


ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Y el código que genera para neo4j es:

```
CREATE
('0':Persona {nombre:'Danny',apellido:'Díaz'}) ,
('3':Persona {nombre:'William',apellido:'Astudillo'}) ,
('4':Persona {nombre:'Néstor',apellido:'Alarcón'}) ,
('6':Persona {nombre:'Esteban',apellido:'Guamán'}) ,
('7':Universidad {siglas:'EPN'}) ,
('8':Universidad {siglas:'ESPOCH'}) ,
('9':Universidad {siglas:'ESPE'}) ,
('10':Empresa {nombre:'Grafinsa'}) ,
('11':Hobbie {tema:'Crear videojuegos'}) ,
('12':Ciudad {nombre:'Quito'}) ,
('13':`Categoría` {nombre:'A'}) ,
('14':`Categoría` {nombre:'B'}) ,
('0')-[:`es_amigo_de` ]->('3'),
('0')-[:`es_amigo_de` ]->('4'),
('0')-[:`es_amigo_de` ]->('6'),
('3')-[:`es_amigo_de` ]->('0'),
('4')-[:`es_amigo_de` ]->('0'),
('6')-[:`es_amigo_de` ]->('0'),
('0')-[:`RELATED_TO` ]->('0'),
('0')-[:`estudia_en` ]->('7'),
('3')-[:`RELATED_TO` ]->('4'),
('3')-[:`RELATED_TO` ]->('4'),
('3')-[:`estudia_en` ]->('8'),
('4')-[:`estudia_en` ]->('9'),
('6')-[:`trabaja_en` ]->('10'),
('0')-[:`tiene_el_hobbie` ]->('11'),
('6')-[:`tiene_el_hobbie` ]->('11'),
('0')-[:`vive_en` ]->('12'),
('4')-[:`vive_en` ]->('12'),
('6')-[:`vive_en` ]->('12'),
('7')-[:`tiene_la_categoria` ]->('13'),
('9')-[:`tiene_la_categoria` ]->('13'),
('8')-[:`RELATED_TO` ]->('14')
```

Y en Neo4j:



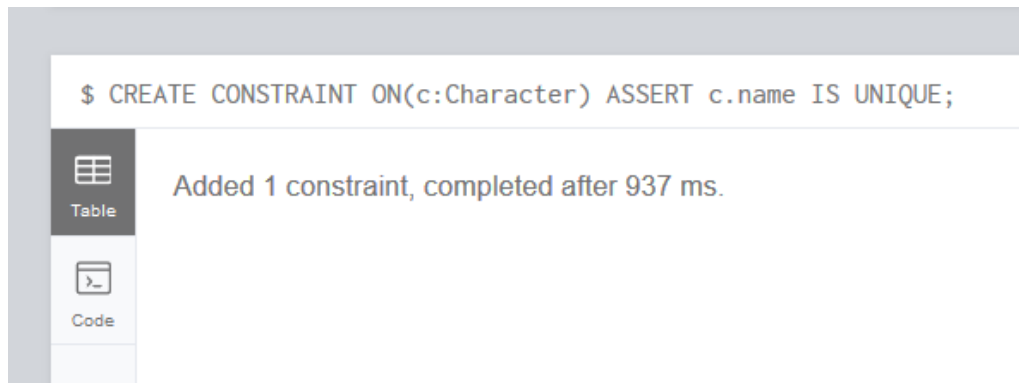


ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

### 3. Ejecución de las instrucciones vistas en clase para Game of Thrones

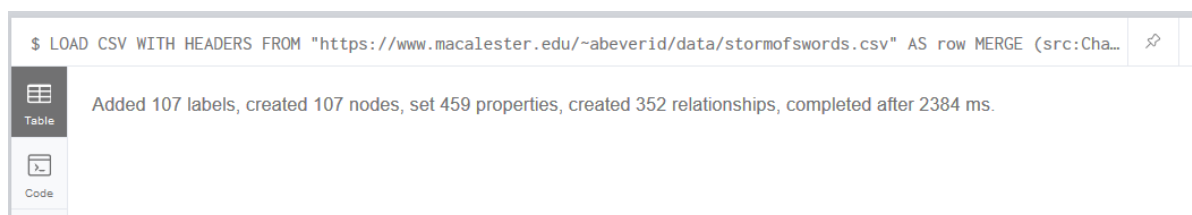
- a. Creación de una restricción sobre Character

```
CREATE CONSTRAINT ON(c:Character) ASSERT c.name IS UNIQUE;
```



- b. Carga de datos de "GameOfThrones"

```
LOAD CSV WITH HEADERS FROM  
"https://www.maclester.edu/~abeverid/data/stormofswords.csv" AS row MERGE  
(src:Character{name: row.Source}) MERGE(tgt:Character{name: row.Target}) MERGE(src-  
[r:INTERACTS]->(tgt) ON CREATE SET r.weight=toInteger(row.Weight)
```

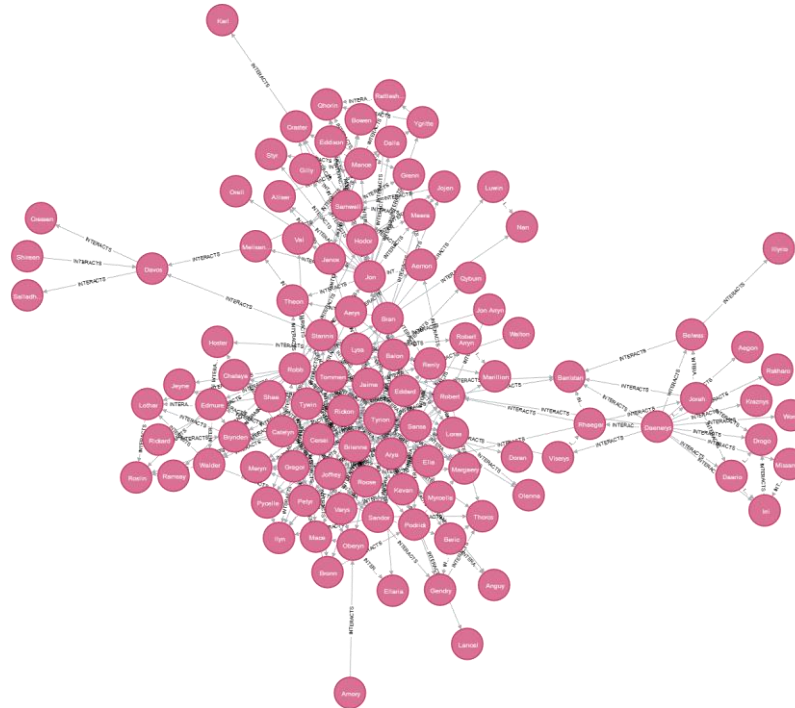




ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

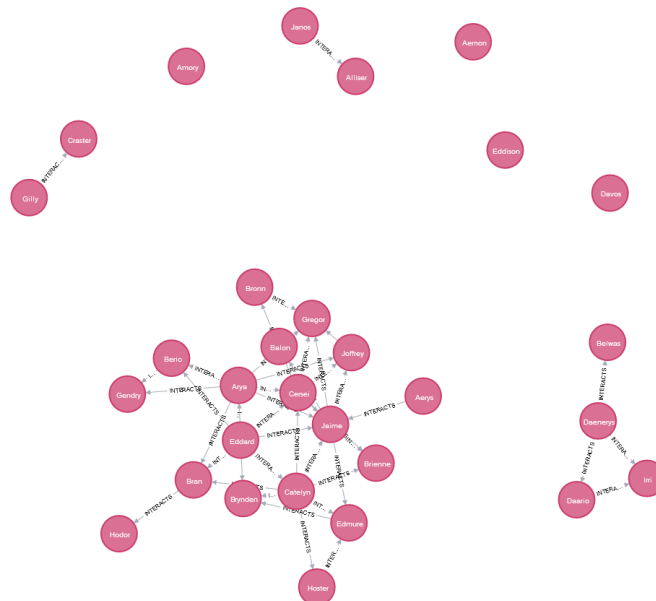
c. Visualización de las interacciones entre Personajes de la serie

MATCH p=(Character)-[:INTERACTS]-(Character) RETURN p



d. Consulta limitada a 30 registros

MATCH (n) RETURN (n) LIMIT 30





ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

e. Conteo de la cantidad de personajes

MATCH (c:Character) RETURN count(c)

\$ MATCH (c:Character) RETURN count(c)	
count(c)	107

f. Estadísticas de los números de interacciones

MATCH (c:Character)-[:INTERACTS]->( ) WITH c, count(\*) AS num RETURN min(num) AS min, max(num) AS max, avg(num) AS avg\_characters, stdev(num) AS stdev

\$ MATCH (c:Character)-[:INTERACTS]->( ) WITH c, count(*) AS num RETURN min(num) AS min, max(num) AS max, avg(num) AS avg_characters, stdev(num) AS stdev				
min	max	avg_characters	stdev	
1	24	4.957746478873241	6.2276723918750845	

g. Camino más corto entre los personajes (consulta de solo 4)

MATCH (a:Character), (b:Character) WHERE id(a) > id(b) MATCH p=shortestPath((a)-[:INTERACTS\*]-(b)) RETURN length(p) AS len, extract(x IN nodes(p) | x.name) AS path ORDER BY len DESC LIMIT 4

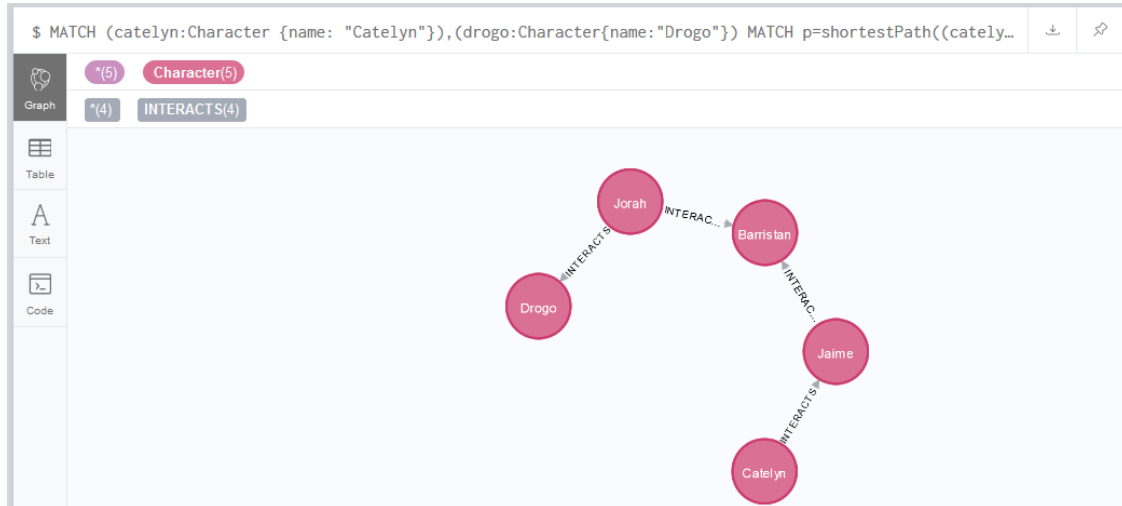
len	path
6	["Illyrio", "Belwas", "Daenerys", "Viserys", "Tyrion", "Oberyn", "Amory"]
6	["Illyrio", "Belwas", "Daenerys", "Robert", "Eddard", "Bran", "Jojen"]
6	["Illyrio", "Belwas", "Barristan", "Jaime", "Stannis", "Davos", "Shireen"]
6	["Illyrio", "Belwas", "Daenerys", "Robert", "Eddard", "Bran", "Luwin"]



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

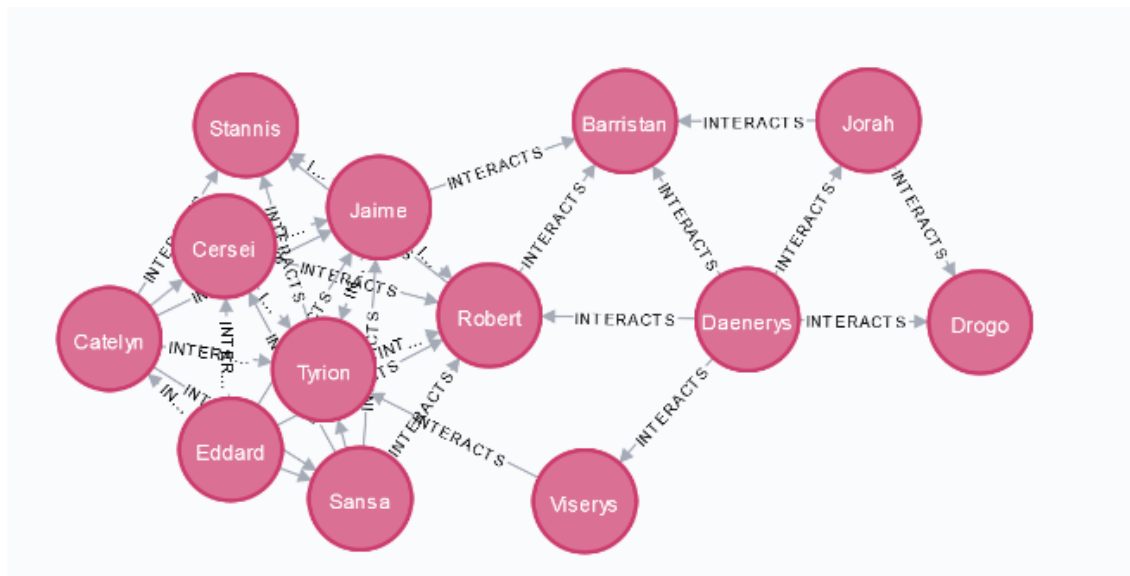
h. Camino más corto entre Catelyn y Drogo

```
MATCH (catelyn:Character {name: "Catelyn"}),(drogo:Character{name:"Drogo"}) MATCH p=shortestPath((catelyn)-[INTERACTS*]-(drogo)) RETURN p
```



i. Todos los caminos desde Catelyn hasta Drogo

```
MATCH (catelyn:Character {name: "Catelyn"}), (drogo:Character {name: "Drogo"}) MATCH p=allShortestPaths((catelyn)-[INTERACTS*]-(drogo)) RETURN p
```





**ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

---

j. Grado de popularidad

MATCH (c:Character) RETURN c.name AS character, size((c)-[:INTERACTS]-()) AS degree ORDER BY degree DESC

MATCH (c:Character) RETURN c.name AS character, size((c)-[:INTERACTS]-()) AS degree ORDER BY degree DESC

character	degree
"Tyrion"	36
"Jon"	26
"Sansa"	26
"Robb"	25
"Jaime"	24
"Tywin"	22
"Cercei"	20
"Arya"	19
"Catelyn"	18
"Joffrey"	18
"Robert"	18
"Samwell"	15
"Bran"	14
"Daenerys"	14

**Análisis de resultados:**

La interfaz permite construir grafos de manera rápida pero no es una opción escalable porque mientras más relaciones se introducía, más complicado era entender lo que estaba pasando.

Al dibujar varios nodos (cientos) se ralentizaba un poco el procesamiento.

**Conclusiones y recomendaciones:**

- Es mejor crear los nodos por código, de esa manera se tiene más control.
- El desempeño de una base de datos orientada a grafos es mayor a la de una base de datos relacional cuando se trata de conexiones entre nodos.
- Se logró instalar Neo4j y realizar 3 nodos con sus relaciones.
- Usar una interfaz gráfica para crear nodos de manera más sencilla.
- Utilizar los grafos de GameOfThrones para practicar consultas.
- Se recomienda limitar las consultas para mejorar el procesamiento.
- Es recomendable usar Neo4j con un API Rest hacia nuestra página Web con su propia base de datos relacional para utilizar las bases de datos que ya tenemos.

**Referencia:**

- [1] «WordPress,» [En línea]. Available: <https://networkofthrones.wordpress.com/>. [Último acceso: 28 10 2019].
- [2] «Neo4J,» [En línea]. Available: <https://neo4j.com/blog/graphcast-graph-algorithms-game-of-thrones/>. [Último acceso: 28 10 2019].
- [3] «Macalester,» [En línea]. Available: <https://www.macalester.edu/~abeverid/index.html>. [Último acceso: 28 10 2019].
- [4] «BBVA,» [En línea]. Available: <https://www.bbva.com/es/que-es-neo4j-y-para-que-sirve-una-base-de-datos-orientada-a-grafos/>. [Último acceso: 28 10 2019].