



Alumnos:

Christian Hernandez

Docente:

Ing. Diego Quisi

Materia:

Inteligencia Artificial

Tema:

Neoj4

Ciclo

9no Ciclo

Cuenca 6 mayo 2020

CREAR NODOS CON PROPIEDADES

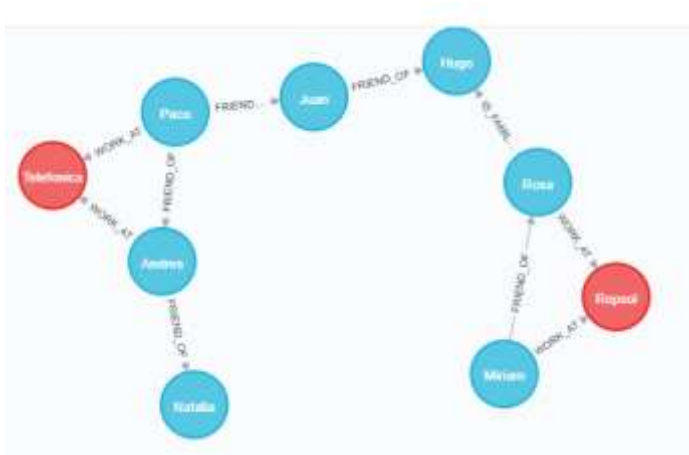
```
CREATE (Paco: Person {name:'Paco', born:1964})
CREATE (Juan: Person {name:'Juan', born:1967})
CREATE (Andres:Person {name:'Andres', born:1961})
CREATE (Hugo:Person {name:'Hugo', born:1960})
CREATE (Natalia:Person {name:'Natalia', born:1967})
CREATE (Miriam:Person {name:'Miriam', born:1965})
CREATE (Rosa:Person {name:'Rosa', born:1952})
CREATE
  (Telefonica:Company {name:'Telefonica', central_office:'Madrid',
    sector:'telecomunicaciones'}),
    (Repsol:Company {name:'Repsol', central_office:'Madrid',
    sector:'energia'})
```

CREAR RELACIONES ENTRE NODOS

```
CREATE
  (Paco)-[:FRIEND_OF {role:['Amigo de Trabajo']}]>(Juan),
  (Paco)-[:FRIEND_OF {role:['Amigo de Trabajo']}]>(Andres),
  (Juan)-[:FRIEND_OF {role:['Amigo de la infancia']}]>(Hugo),
  (Andres)-[:FRIEND_OF {role:['Amigo de la infancia']}]>(Natalia),
  (Miriam)-[:FRIEND_OF {role:['Amigo de Trabajo']}]>(Rosa)
CREATE
  (Paco)-[:WORK_AT {position:['Director de Marketing']}]>
  (Telefonica),
  (Andres)-[:WORK_AT {position:['Director de Marketing']}]>
  (Telefonica),
  (Miriam)-[:WORK_AT {position:['Director de Marketing']}]>
  (Repsol),
  (Rosa)-[:WORK_AT {position:['Director de Marketing']}]>(Repsol)
CREATE
  (Rosa)-[:IS_FAMILY_OF {position:['Prima']}]>(Hugo)
```

MOSTAR TODO EL GRAFO

```
MATCH (n) RETURN n
```



BUSCAR POR NODO Y RELACIÓN

Buscar amigos de Paco

```
MATCH (Paco {name: "Paco"})-[: FRIEND_OF]->(amigos) RETURN amigos
```



LISTADO BUSCANDO POR UNA RELACIÓN

Listado de personas y que trabajan en compañías

```
MATCH (persona)-[:WORK_AT]->(compania) RETURN persona.name, compania.name
```

```
| MATCH (persona)-[:WORK_AT]->(compania) RETURN persona.name, compania.name
```

persona.name	compania.name
"Andres"	"Telefonica"
"Paco"	"Telefonica"
"Miriam"	"Repsol"
"Rosa"	"Repsol"

CONTAR ELEMENTOS DE DOS RELACIONES CONCATENADAS

Buscar las personas con más amigos teniendo en cuenta amigos de amigos

```
MATCH (p1:Person)-[:FRIEND_OF]->(p2:Person)-[:FRIEND_OF]->(p3:Person)
RETURN p1.name AS Persona, COUNT(p2) + COUNT(p3) AS Amigos
```

```
j$ MATCH (p1:Person)-[:FRIEND_OF]->(p2:Person)-[:FRIEND_OF]->(p3:Person) RETURN p1.name AS Persona, COUNT(p2) + COUNT(p3) AS Amigos
```

Persona	Amigos
"Paco"	4

BUSCAR CON RESTRICCIONES

Listado de personas nacidas después de los 60

```
MATCH (p:Person) WHERE p.born > 1960 RETURN p.name
```

```
§ MATCH (p:Person) WHERE p.born > 1960 RETURN p.name
```

p.name
"Paco"
"Juan"
"Andres"
"Natalia"
"Miriam"

BASE DE DATOS UTILIZADA

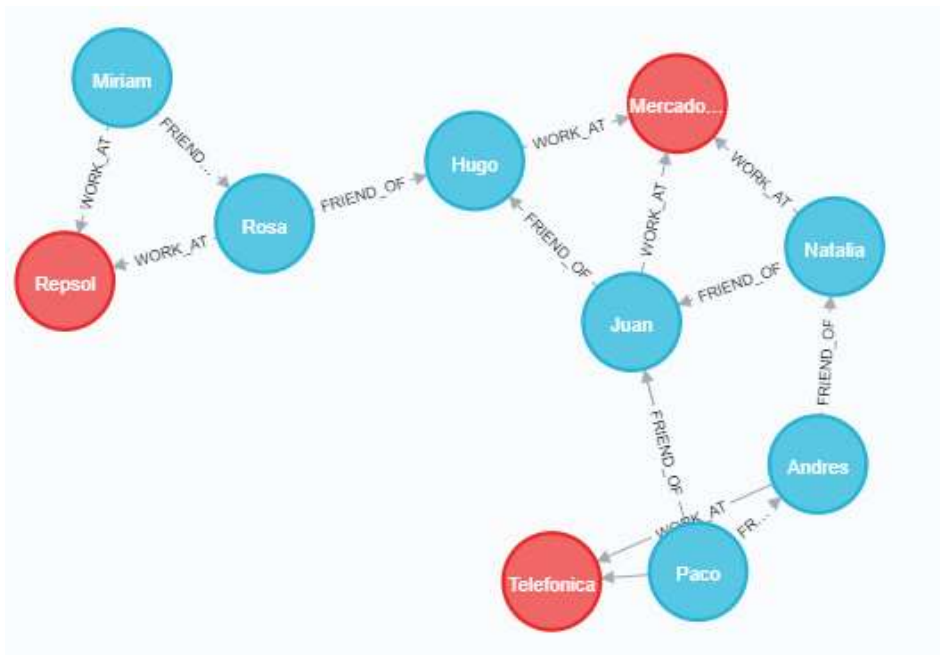
Se define primeramente la base de datos a utilizar en el ejemplo.

```
CREATE (Paco:Person {name:'Paco', born:1964}),
      (Juan:Person {name:'Juan', born:1967}),
      (Andres:Person {name:'Andres', born:1961}),
      (Hugo:Person {name:'Hugo', born:1960}),
      (Natalia:Person {name:'Natalia', born:1967}),
      (Miriam:Person {name:'Miriam', born:1965}),
      (Rosa:Person {name:'Rosa', born:1952})

CREATE
  (Telefonica:Company {name:'Telefonica', central_office:'Madrid',
sector:'telecomunicaciones'}),
  (Repsol:Company {name:'Repsol', central_office:'Madrid',
sector:'energia'}),
  (Mercadona:Company {name:'Mercadona', central_office:'Valencia',
sector:'alimentacion'})

CREATE
  (Paco)-[:FRIEND_OF {role:['Amigo de Trabajo']}]>(Juan),
  (Paco)-[:FRIEND_OF {role:['Amigo de Trabajo']}]>(Andres),
  (Juan)-[:FRIEND_OF {role:['Amigo de la infancia']}]>(Hugo),
  (Andres)-[:FRIEND_OF {role:['Amigo de la infancia']}]>(Natalia),
  (Miriam)-[:FRIEND_OF {role:['Amigo de Trabajo']}]>(Rosa),
  (Natalia)-[:FRIEND_OF {role:['Amigo de gimnasio']}]>(Juan),
  (Rosa)-[:FRIEND_OF {role:['Amigo de Trabajo']}]>(Hugo)

CREATE
  (Paco)-[:WORK_AT {position:['Director de Marketing']}]>
>(Telefonica),
  (Andres)-[:WORK_AT {position:['Director de Marketing']}]>
>(Telefonica),
  (Miriam)-[:WORK_AT {position:['Director de Marketing']}]>
>(Repsol),
  (Rosa)-[:WORK_AT {position:['Director de Marketing']}]>(Repsol),
  (Hugo)-[:WORK_AT {position:['Director de Marketing']}]>
>(Mercadona),
  (Juan)-[:WORK_AT {position:['Director de Marketing']}]>
>(Mercadona),
  (Natalia)-[:WORK_AT {position:['Director de Marketing']}]>
>(Mercadona)
```



CONTAR ELEMENTOS DERIVADOS DE DOS RELACIONES

Contar para cada persona el número de amigos que tiene trabajando en los diferentes sectores

```
MATCH (p1:Person)-[:FRIEND_OF]->(p2:Person)-[:WORK_AT]->(c:Company)
RETURN p1.name AS Persona , COUNT(c.sector) AS Sectores
```

```
$ MATCH (p1:Person)-[:FRIEND_OF]->(p2:Person)-[:WORK_...
```

Persona

Sectores

"Paco"

2

"Miriam"

1

"Rosa"

1

"Juan"

1

"Natalia"

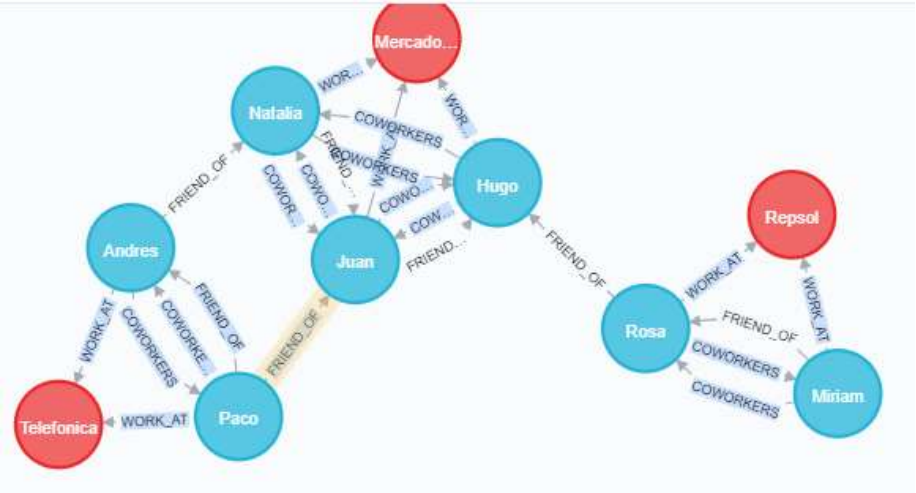
1

"Andres"

1

GENERAR RELACIONES A PARTIR DE CONSULTAS

Crear la relación de compañeros de trabajo «coworkers» para las personas que trabajen en la misma compañía.



AGRUPACIONES

Para cada persona agrupar por sectores en que trabajan sus amigos

```
MATCH (p1:Person)-[:FRIEND_OF]->(p2:Person)-[:WORK_AT]->(c:Company)
RETURN p1.name, COLLECT(c.sector)
```

p1.name	p2.name	COLLECT(c.sector)
"Paco"	"Andres"	["telecomunicaciones"]
"Miriam"	"Rosa"	["energia"]
"Rosa"	"Hugo"	["alimentacion"]
"Juan"	"Hugo"	["alimentacion"]
"Natalia"	"Juan"	["alimentacion"]
"Paco"	"Juan"	["alimentacion"]
"Andres"	"Natalia"	["alimentacion"]

MODIFICAR PROPIEDADES A UN NODO

```
MERGE (p:Person {name: 'Paco'}) SET p.age = 34, p.coat = 'Yellow'  
RETURN p
```



MODIFICAR PROPIEDADES A UNA RELACIÓN

```
MERGE (Paco)-[r:FRIEND_OF]->(Juan) SET r.ages = 34 RETURN r
```

```
§ MERGE (Paco)-[r:FRIEND_OF]->(Juan) SET r.ages = 34 ...
```

"r"
{"ages":34,"role":["Amigo de gimnasio"]}
{"ages":34,"role":["Amigo de Trabajo"]}
{"ages":34,"role":["Amigo de Trabajo"]}
{"ages":34,"role":["Amigo de Trabajo"]}
{"ages":34,"role":["Amigo de la infancia"]}
{"ages":34,"role":["Amigo de la infancia"]}
{"ages":34,"role":["Amigo de Trabajo"]}

BORRAR TODO EL GRAFO

```
MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r
```

Server version	Neo4j/4.0.3
Server address	localhost:7687
Query	MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r
Summary ▶	{, "query": {, "text": "MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r", ...
Response ▶	[] ...