

- CYPHER Y NEO4J -

Guillermo Facundo Colunga, Pablo Menéndez Suárez y Jorge Vila Suárez.

PROFESORADO:

Darío Álvarez Gutiérrez - darioa@uniovi.es

Resumen. En este documento se presentan la propuesta, realización y conclusiones de la primera práctica de la asignatura de Repositorios de la Información 2017 de la Universidad de Oviedo. En ella se exploran la creación de una base de datos en modelo de grafo y consultas aplicadas sobre la misma.

Palabras clave: neo4j, cypher, base de datos, grafo, noSQL.

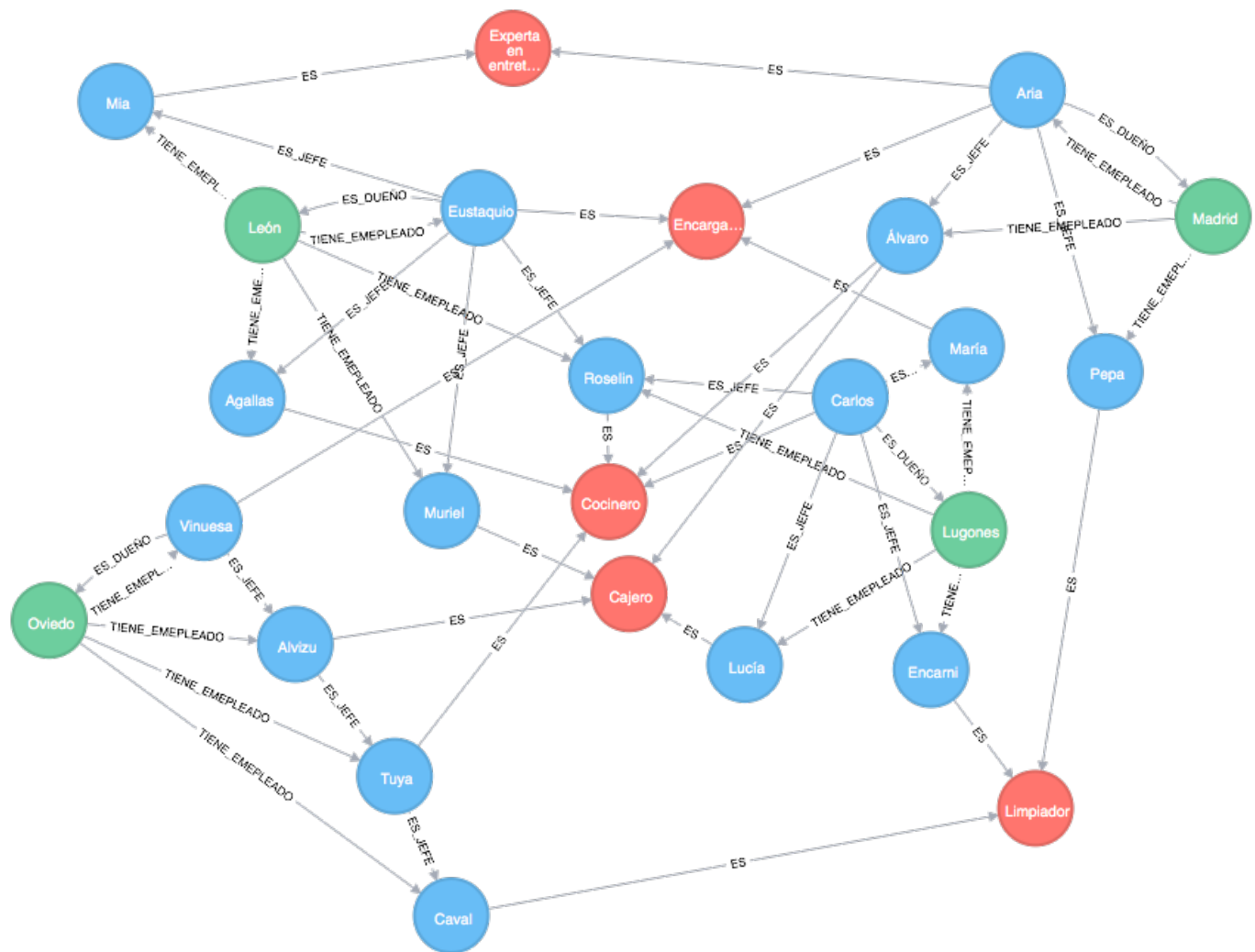
Dominio de Aplicación

Para esta práctica elegimos un dominio que fuera representativo del mundo real. Así mismo como se trata de una base de datos en grado también intentamos aproximarnos lo más posible a lo que en la vida real podían ser los nodos y sus relaciones. Es por esto que finalmente escogimos un modelo de aplicación basado en las famosas franquicias Kentucky Fried Chicken (KFC). Dentro de las franquicias intentamos definir las relaciones entre trabajadores, dueños, puestos y franquicias en si. De esa forma podemos relacionar a una persona con una franquicia, el tipo de relación que tiene y el resto de relaciones con otras personas de la misma o distinta franquicia.

De esta forma definimos las que las siguientes entidades irán representadas a través de nodos: Persona, Franquicia y Puesto. Y como relaciones encontraremos: ES, ES_DUEÑO, ES_JEFE y TIENE_EMPLEADO.

// Ampliar descripción del dominio... TODO

Representación de la Instancia



En la instancia anterior encontramos un grafo formado por 25 nodos y 51 relaciones. Entre los nodos nos encontramos los siguientes tipos:

Persona. Representan una persona del mundo real, en nuestro dominio pueden tener nombre y apellido.

Puesto. Representa el puesto que puede desempeñar un empleado dentro de una franquicia.

Franquicia. Es la representación del restaurante en sí.

Y entre las relaciones:

ES. Se aplica a los nodos de tipo Persona y tiene como fin un nodo de tipo Puesto, en dirección Persona -> Puesto. Representando así el puesto de desempeña una persona.

ES_DUEÑO. Se aplica a nodos de tipo Persona y tiene como fin un nodo de tipo Franquicia, en dirección Persona -> Franquicia. Representando que una persona es dueña de una franquicia.

ES_JEFE. Se aplica entre nodos de tipo Persona {a, b} y (a)->(b) representa que (a) es jefe de (b).

TIENE_EMPLEADO. Se aplica a nodos de tipo Franquicia y tiene como destino nodos de tipo Persona, representando así la relación entre una franquicia y un empleado.

Código de creación del grafo

```
// Creando nodos de tipo Persona.
CREATE (MiaK:Persona {nombre:'Mia', apellidos: 'Kalifa'})
CREATE (Eustaquio:Persona {nombre:'Eustaquio'})
CREATE (Muriel:Persona {nombre:'Muriel'})
CREATE (Agallas:Persona {nombre:'Agallas'})
CREATE (Roselin:Persona {nombre:'Roselin'})
CREATE (Carlos:Persona {nombre:'Carlos'})
CREATE (María:Persona {nombre:'María'})
CREATE (Pepa:Persona {nombre:'Pepa'})
CREATE (Aria:Persona {nombre:'Aria'})
CREATE (Álvaro:Persona {nombre:'Álvaro'})
CREATE (Encarni:Persona {nombre:'Encarni'})
CREATE (Lucía:Persona {nombre:'Lucía'})
CREATE (Vinuesa:Persona {nombre:'Vinuesa'})
CREATE (Alvizu:Persona {nombre:'Alvizu'})
CREATE (Tuya:Persona {nombre:'Tuya'})
CREATE (Caval:Persona {nombre:'Caval'})

// Creando nodos de tipo Franquicia.
CREATE (Leon:Franquicia {ciudad:'León', nombre : 'Franquicia de León'})
CREATE (Madrid:Franquicia {ciudad:'Madrid', nombre : 'Franquicia de Madrid'})
CREATE (Lugones:Franquicia {ciudad:'Lugones', nombre : 'Franquicia de Lugones'})
CREATE (CAU:Franquicia {ciudad:'Oviedo', nombre : 'Franquicia del CAU'})

// Creando nodos de tipo Puesto
CREATE (Encargado:Puesto {puesto:'Encargado'})
CREATE (Limpiador:Puesto {puesto:'Limpiador'})
CREATE (Cocinero:Puesto {puesto:'Cocinero'})
CREATE (Cajero:Puesto {puesto:'Cajero'})
CREATE (ExpertaEnEntretenimiento:Puesto {puesto:'Experta en entretenimiento'})

// Creando relaciones entre nodos.
CREATE
  (MiaK)-[:ES]->(ExpertaEnEntretenimiento),
  (Aria)-[:ES]->(ExpertaEnEntretenimiento),
  (Aria)-[:ES]->(Encargado),
  (Eustaquio)-[:ES]->(Encargado),
  (María)-[:ES]->(Encargado),
  (Vinuesa)-[:ES]->(Encargado),
  (Alvizu)-[:ES]->(Cajero),
  (Lucía)-[:ES]->(Cajero),
  (Muriel)-[:ES]->(Cajero),
  (Álvaro)-[:ES]->(Cajero),
  (Agallas)-[:ES]->(Cocinero),
  (Roselin)-[:ES]->(Cocinero),
  (Carlos)-[:ES]->(Cocinero),
  (Álvaro)-[:ES]->(Cocinero),
  (Tuya)-[:ES]->(Cocinero),
  (Encarni)-[:ES]->(Limpiador),
  (Caval)-[:ES]->(Limpiador),
```

```

(Pepa)-[:ES]->(Limpiador),
(Leon)-[:ES_TRABAJADOR]->(Muriel),
(Leon)-[:ES_TRABAJADOR]->(Roselin),
(Leon)-[:ES_TRABAJADOR]->(Agallas),
(Leon)-[:ES_TRABAJADOR]->(Eustaquio),
(Leon)-[:ES_TRABAJADOR]->(MiaK),
(CAU)-[:ES_TRABAJADOR]->(Vinuesa),
(CAU)-[:ES_TRABAJADOR]->(Alvizu),
(CAU)-[:ES_TRABAJADOR]->(Tuya),
(CAU)-[:ES_TRABAJADOR]->(Caval),
(Madrid)-[:ES_TRABAJADOR]->(Aria),
(Madrid)-[:ES_TRABAJADOR]->(Pepa),
(Madrid)-[:ES_TRABAJADOR]->(Álvaro),
(Lugones)-[:ES_TRABAJADOR]->(Encarni),
(Lugones)-[:ES_TRABAJADOR]->(María),
(Lugones)-[:ES_TRABAJADOR]->(Roselin),
(Lugones)-[:ES_TRABAJADOR]->(Lucía),
(Eustaquio)-[:ES_DUEÑO]->(Leon),
(Aria)-[:ES_DUEÑO]->(Madrid),
(Carlos)-[:ES_DUEÑO]->(Lugones),
(Vinuesa)-[:ES_DUEÑO]->(CAU),
(Eustaquio)-[:ES_JEFE]->(MiaK),
(Eustaquio)-[:ES_JEFE]->(Muriel),
(Eustaquio)-[:ES_JEFE]->(Roselin),
(Eustaquio)-[:ES_JEFE]->(Agallas),
(Aria)-[:ES_JEFE]->(Pepa),
(Aria)-[:ES_JEFE]->(Álvaro),
(Carlos)-[:ES_JEFE]->(María),
(Carlos)-[:ES_JEFE]->(Encarni),
(Carlos)-[:ES_JEFE]->(Lucía),
(Carlos)-[:ES_JEFE]->(Roselin),
(Vinuesa)-[:ES_JEFE]->(Alvizu),
(Alvizu)-[:ES_JEFE]->(Tuya),
(Tuya)-[:ES_JEFE]->(Caval)

```

Consultas realizadas sobre la instancia

| C.1 | Dificultad: Media |
|---|-------------------|
| Objetivo Consulta: | |
| <p>En esta consulta se esperan obtener aquellos trabajadores que siendo encargados de una franquicia no son jefe de ningún otro empleado ni propietarios de la franquicia. Una situación que pudiendo ser poco frecuente se puede dar en una franquicia mal coordinada y que puede estar inflando el gasto en sueldos para trabajadores que no desempeñan la función que deberían pues según el contrato de la franquicia “todo encargado deberá ser jefe de al menos otro empleado”.</p> | |
| Código Consulta: | |

```
MATCH ((p:Persona)-[:ES]->(pu:Puesto) {puesto: "Encargado"})
WHERE NOT ((p)-[:ES_JEFE]->(:Persona))
AND NOT ((p)-[:ES_DUEÑO]->(:Franquicia))
RETURN p
```

Resultado:

```
+-----+
| "p"    |
+-----+
| {"nombre":"María"} |
+-----+
```

C.2

Dificultad: Elemental

Objetivo Consulta:

En esta consulta se espera obtener los trabajadores de la franquicia de León.

Código Consulta:

```
MATCH (n:Persona)
WHERE algo
RETURN n
```

Resultado:

```
+-----+
| n.name |
+-----+
| "neo4j" |
+-----+
```

C.3

Dificultad: Intermedia

Objetivo Consulta:

En esta consulta se espera obtener los trabajadores de la franquicia de León.

Código Consulta:

```
MATCH (n:Persona)
WHERE algo
RETURN n
```

Resultado:

```
+-----+
| n.name |
+-----+
| "neo4j" |
+-----+
```

C.4

Dificultad: Intermedia

Objetivo Consulta:

En esta consulta se espera obtener los trabajadores de la franquicia de León.

Código Consulta:

```
MATCH (n:Persona)
WHERE algo
RETURN n
```

Resultado:

```
+-----+
| n.name |
+-----+
| "neo4j" |
+-----+
```

C.5

Dificultad: Avanzada

Objetivo Consulta:

En esta consulta se espera obtener los trabajadores de la franquicia de León.

Código Consulta:

```
MATCH (n:Persona)
WHERE algo
RETURN n
```

Resultado:

```
+-----+
| n.name |
+-----+
| "neo4j" |
+-----+
```

C.6

Dificultad: Avanzada

Objetivo Consulta:

En esta consulta se espera obtener los trabajadores de la franquicia de León.

Código Consulta:

```
MATCH (n:Persona)
WHERE algo
RETURN n
```

Resultado:

```
+-----+
| n.name |
+-----+
| "neo4j" |
+-----+
```

Conclusiones

Tras la realización del trabajo hemos blaojlkasdnvoenvlñqkfnrlnkndveqorivnlñkadsnvodfknv lkn o jlcñ knñoc jnñlk ndao ndlñfn dfoin lñdfkm aodfj lñadm ñdlakfvàdo.