Neo4j

Contenido

[Recursos útiles 1](#_Toc203585220)

[Objetivo de la práctica 2](#_Toc203585221)

[Instalación Neo4j y herramientas asociadas 2](#_Toc203585222)

[Instalación servidor Neo4j y Neo4j browser como Docker 2](#_Toc203585223)

[Tutorial Movie Graph 4](#_Toc203585224)

[Ejercicios sistema de recomendación 7](#_Toc203585225)

[Integración con Python 8](#_Toc203585226)

[Conexión y librería 8](#_Toc203585227)

[Ejemplos de consultas: 9](#_Toc203585228)

[Entregables de la práctica 10](#_Toc203585229)

# Recursos útiles

Recuerde el directorio de recursos compartidos disponible en:

<https://nascorformacion0-my.sharepoint.com/:f:/g/personal/juan_pinuela_docente_nascorformacion_com/EtGSBITnqFpJp6-XpG6REOQBFycP_G9zgcB89jxPymjV2Q?e=IBIEER>

Para esta práctica disponemos de una carpeta con nombre “**neo4j”** donde podemos encontrar todos los ficheros utilizados y creados en este guion.

# Objetivo de la práctica

Esta práctica tiene como objetivo introducir al estudiante en los conceptos fundamentales del uso de bases de datos orientadas a grafos mediante Neo4j. El estudiante será capaz de crear y consultar grafos sencillos, entender cómo modelar relaciones entre entidades mediante nodos y relaciones, y desarrollar consultas básicas y avanzadas usando Cypher. Además, se pretende familiarizar al alumno con el entorno visual de Neo4j Browser para la interpretación visual del grafo, aplicando conceptos básicos de sistemas de recomendación mediante grafos

# Instalación Neo4j y herramientas asociadas

## Instalación servidor Neo4j y Neo4j browser como Docker

Creamos la carpeta “C:\Docker\neo4j” donde copiamos el siguiente docker-compose.yml, que podemos lanzar sin más que ejecutando desde la línea de comandos:

>> docker-compose up -d

version: '3.8'

services:

neo4j:

image: neo4j:5

container\_name: neo4j

restart: always

ports:

- 7474:7474

- 7687:7687

environment:

NEO4J\_AUTH: neo4j/password

Como vemos tenemos dos puertos en uso en esta ocasión. Por un lado está el puerto 7687 que es el estándar de Neo4j y que utilizaremos para conectarnos a la instancia de Neo4j mediante la cadena:



Con usuario y base de datos neo4j (el usuario es especificado con su password como variables de entorno en el fichero Docker).

Ese usuario y contraseña lo usamos también para navegar a <http://localhost:7474/> que es el puerto por el que accedemos a Neo4j Browser. Esta será la herramienta que utilizaremos como interfaz gráfico para operar con Neo4j. Es especialmente interesante seleccionar la opción “Try Neo4j with the Movie Graph” que nos lleva a un tutorial o guía paso a paso de las características claves de esta base de datos basada en grafos.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Es importante saber como reiniciar todo y poder volver a experimentar con esta base de datos. Para ello, tenga en cuenta ya estos primeros comandos:

Selecciona todos los nodos para eliminarlos junto con las relaciones en las que están implicados:

MATCH (n)

DETACH DELETE n

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Si se ha borrado todo veremos que tenemos 0 nodos y 0 relaciones en el panel de información siguiente:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Será vital también usar saltos de línea en la zona de consultas mediante la combinación de teclas SHIFT+ENTER.

Podemos reiniciar el tutorial cerrando la sesión y volviendo a conectar.

## Tutorial Movie Graph

|  |  |
| --- | --- |
| Interfaz de usuario gráfica  El contenido generado por IA puede ser incorrecto. | En cada paso debemos ir seleccionado la opción resaltada en rojo que carga en el editor de Cypher las sentencias oportunas para editarlas, analizarlas y finalmente ejecutarlas o también podemos seleccionar la zona resaltada en amarillo donde directamente se ejecuta la consulta.  Si cargamos los datos nos aparecerán tal y como se muestra a continuación que tendremos en base de datos 171 nodos entre 38 películas y el resto personas (entre actores, directores, etc.).  Podemos ver un subconjunto del grafo con:  MATCH (person:Person {name: 'Tom Hanks'})  MATCH path = (person)-[:ACTED\_IN]->(m)<-[:DIRECTED]-(d)  RETURN path; |

Tabla

El contenido generado por IA puede ser incorrecto.

Explicando esta consulta tenemos:

**MATCH (person:Person {name: 'Tom Hanks'})**

* Busca en el grafo el nodo etiquetado como Person cuyo atributo name sea exactamente "Tom Hanks".
* La variable person almacena ese nodo.
* Es el **punto de partida** de la consulta.

**MATCH path = (person)-[:ACTED\_IN]->(m)<-[:DIRECTED]-(d)**

* A partir del nodo almacenado en person (Tom Hanks), busca todos los caminos con esta estructura:
  + Desde person (Tom Hanks):
    - Relación **:ACTED\_IN** (actuó en).
    - Hacia un nodo de película (variable m).
  + Desde ese nodo película (m):
    - Relación **:DIRECTED** (dirigida por).
    - Desde un nodo de persona (variable d), que representa un director.
* La variable path almacena el **camino completo**:
  + Tom Hanks → película → director

**RETURN path**

* Devuelve el **subgrafo** formado por:
  + El nodo person (Tom Hanks).
  + Todos los nodos m (películas en las que actuó).
  + Todos los nodos d (directores de esas películas).
  + Las relaciones :ACTED\_IN y :DIRECTED.

Y con esta consulta visualizamos el siguiente subgrafo:

Diagrama

El contenido generado por IA puede ser incorrecto.

El propio tutorial nos plantea realizar algunas pruebas.

Step 4/8. Viendo consultas para encontrar en el grafo a Tom Hanks es:

MATCH (p:Person {name: "Tom Hanks"})

RETURN p

O consultas con condiciones WHERE como:

MATCH (m:Movie) WHERE m.released >= 1990 AND m.released < 2000

RETURN m.title, m.released

1. **Cómo encontraría una película llamada “*Cloud Atlas*”**

O si ya utilizamos relaciones como en:

MATCH (p:Person {name: "Tom Hanks"})-[:ACTED\_IN]->(movie)

RETURN p;

1. **¿Puedes especificar un patrón para saber quién dirigió Cloud Atlas? Pista: La relación entre un director y la película que dirigió es :DIRECTED.**

Finalemente tenemos consultas muchos más complejas que tienen que ver con teoría de grafos y que quedan fuera del alcance de la práctica como son

MATCH (:Person {name:"Kevin Bacon"})-[\*1..6]-(n)

RETURN DISTINCT n

El patrón [\*1..6] significa:

* Busca caminos con entre 1 y 6 saltos de relaciones.
* Es decir, nodos que estén a una distancia máxima de 6 relaciones de Kevin Bacon.
* No importa el tipo de relación ni la dirección
* Devuelve todos los nodos encontrados, eliminando duplicados con DISTINCT

Siempre es más atractivo mostrar el grafo o subgrafo haciendo:

MATCH path = (:Person {name:"Kevin Bacon"})-[\*1..6]-(n)

RETURN path LIMIT 20;

La última consulta que se destaca es:

MATCH path=shortestPath(

(:Person {name:"Kevin Bacon"})-[\*]-(:Person {name:"Meg Ryan"})

)

RETURN path, length(path) / 2 as distance

Con el siguiente resultado visual por todos conocido:

Gráfico, Gráfico de burbujas

El contenido generado por IA puede ser incorrecto.

# Ejercicios sistema de recomendación

Disponemos en la carpeta compartida del curso el script “clase1.txt”

No olvidar ejecutar en primer lugar:

MATCH (n)

DETACH DELETE n

Ejecutamos el contenido del fichero y resultarán 12 nodos y 14 relaciones que podemos visualizar con:

MATCH path = ()-->()

RETURN path

1. **Mostrar todas las películas junto con su género**

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

1. **Mostrar usuarios con las películas que valoraron y la puntuación dada**
2. **Películas con puntuación IMDb superior a 8.0**
3. **Sistema de recomendación**

MATCH (m:Movie)<-[r:RATED]-(:User)

MATCH (m)-[:HAS\_IMDB\_RATING]->(imdb:ImdbRating)

WITH m.name AS Pelicula,

count(r) AS TotalVotos,

avg(r.rating) AS RatingUsuarios,

imdb.rating AS RatingIMDb

WITH Pelicula, TotalVotos, RatingUsuarios, RatingIMDb,

((RatingUsuarios \* TotalVotos) + RatingIMDb) / (TotalVotos + 1) AS ScorePonderado

RETURN Pelicula, TotalVotos, RatingUsuarios, RatingIMDb, ScorePonderado

ORDER BY ScorePonderado DESC;

**Tabla

El contenido generado por IA puede ser incorrecto.**

# Integración con Python

## Conexión y librería

En este caso necesitamos instalar el paquete ***neo4j*** y en concreto el objeto ***GraphDatabase*** con:

from neo4j import GraphDatabase

uri = "bolt://localhost:7687"

user = "neo4j"

password = "password"

driver = GraphDatabase.driver(uri, auth=(user, password))

Esta conexión y la carga de clientes y productos incluyendo la compra está ya implementado en el fichero insertar\_en\_neo4j.py disponible como recurso de la práctica.

Una vez ejecutado tenemos haciendo:

MATCH path = ()-->()

RETURN path

Gráfico, Diagrama

El contenido generado por IA puede ser incorrecto.

## Ejemplos de consultas:

1. **¿Cuántas compras totales se han realizado por cliente? (ojo que el programa hace compras aleatorias)**

**Tabla

El contenido generado por IA puede ser incorrecto.**

1. **¿Qué productos han sido comprados por más de un cliente?**

**Gráfico, Gráfico de burbujas

El contenido generado por IA puede ser incorrecto.**

En mi caso el monitor de 27” es el que tiene más compras

1. **¿Qué productos ha comprado alguien que también compró lo mismo que Ana?**

* Busca los productos que Ana ha comprado.
* Encuentra otros clientes que también hayan comprado esos productos.
* Busca qué otros productos han comprado esos clientes.
* Excluye los productos que Ana ya haya comprado.

Diagrama

El contenido generado por IA puede ser incorrecto.

Como Ana ha comprado “video”, “pantallas” y “almacenamiento” le falta el “audio”. Ese será e producto recomendado “

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

# Entregables de la práctica

El alumno deberá entregar las siguientes evidencias al finalizar la práctica:

1. Capturas de pantallas de las consultas realizadas insertadas en este mismo documento junto con la sintaxis del Cypher usado