



## Tarea 2: Neo4j

Profesor: Jose Luis Martí Lara  
Ayudante: Daniela Sánchez Nizza

### 1. Introducción

Las bases de datos NoSQL de grafos almacenan los datos como nodos y relaciones, donde cada nodo tiene un conjunto de propiedades que lo caracterizan. Este tipo de base de datos permite visualizar las relaciones entre datos interconectados de manera gráfica e intuitiva.

Para esta tarea se utilizará Neo4j como gestor de base de datos.

### 2. Set Up: Instalación y conexión

Para empezar, es necesario realizar una conexión a Neo4j utilizando Python:

1. Instalar Neo4j Desktop. La página solicitará algunos datos, pueden ingresar información ficticia si lo prefieren. Al final, recibirán una clave que deberán usar más adelante. Luego, ejecuten el archivo .exe descargado y sigan las instrucciones de instalación.
2. Crear un archivo .py para realizar la tarea, en la ubicación del archivo, ejecuten el siguiente comando en la terminal: **pip install neo4j**.
3. Una vez completados los pasos anteriores, abran Neo4j Desktop y sigan las instrucciones del programa para inicializarlo y crear un nuevo proyecto. El botón "New" debería estar en la esquina superior izquierda.
4. En el proyecto, creen un DBMS usando la opción ".Add"(seleccionar la opción "Local DBMS") y asignen el nombre **Ponydb**, junto con una contraseña. Asegúrense de recordarla.
5. Luego, hagan clic en **Start**. Una vez que el nombre aparezca en la parte superior de la pantalla como activo, creen una base de datos llamada **ponydb**.
6. Finalmente, coloquen el archivo data.py de aula en el mismo directorio donde está su tarea. Ingresen la contraseña que definieron en la línea 182. Mientras la base de datos está activa, ejecuten el programa para poblarla.

Para más detalles sobre la instalación, consulten los siguientes enlaces: [Conexión a Python y Instalación de Neo4j Desktop](#).



### 3. Descripción del problema



En el universo de My Little Pony: La magia de la amistad, la Princesa Celestia, en su afán por mantener la paz y armonía en Equestria, ha decidido que es necesario realizar un censo de la amistad. Este plan piloto consiste en recabar información sobre todas las criaturas de Equestria:

1. **Ponys:** nombre, color, tipo (Alicornio, Pegaso, Unicornio o Pony terrestre), habilidad, cutiemark, gustos y bebida (Coca Cola o Sprite).
2. **Localidades:** nombre, tipo (ciudad o bosque), población (int).
3. **Relaciones:**
  - a) **Amigos:** Ej. Applejack es amiga de Twilight Sparkle.
  - b) **Enemigos:** Ej. Applejack es enemiga de Tempest Shadow.
  - c) **Colaboración:** Ej. Applejack colaboró con Twilight Sparkle (para algún fin).
  - d) **Vive en:** Ej. Applejack vive en Ponyville.

Debido al gran tamaño del reino de Equestria, solo se cuenta con algunos datos de ciertos ponys, pero se espera que en el futuro se expanda la cantidad de datos y consultas, de manera que esta información pueda ayudar a prevenir invasiones de cambiaformas y a fortalecer la magia de la amistad en todo Equestria.

La Princesa le ha encomendado esta tarea a Twilight Sparkle, su alumna prodigio. Sin embargo, en su formación nunca cursó un electivo sobre bases de datos avanzadas y solo las conoce de nombre, por lo que ha solicitado la ayuda de los estudiantes de INF-325 de la USM para diseñar un programa en Python que cumpla con las exigencias de Celestia.

### 4. Desarrollo

Para esta tarea, se le solicita un menú, cada uno de los items a continuación debe estar en una opción distinta:

1. Agregar un pony a la base de datos y definir una relación de amistad con el nodo de nombre Vinyl (nombre: Vinyl Scratch). Se debe solicitar cada atributo por consola. (5 puntos)
2. Dada una ciudad, retornar la cantidad de pegazos, ponis terrestres y unicornios. (Se debe solicitar el nombre de la ciudad por consola). (5 puntos)
3. Agregar un campo extra llamado **anexo** a los ponis según el número de amigos que tengan:



- A los unicornios que tengan 3 o más amigos se les asignará el valor "Sociable".
- A los Alicornios se les asignará el valor Realeza"
- A aquellos unicornios que tengan solo 2 amigos se les asignará el valor Reservado".
- A los unicornios con solo 1 amigo se les asignará el valor "Solitario".
- A los ponis terrestres que tengan 4 o más amigos se les asignará el valor "Hipersociable".
- A aquellos ponis terrestres que tengan 2 amigos o menos se les asignará el valor Reservado".
- Al resto asignar el valor "Por completar".

Todo esto debe realizarse en una única query, de lo contrario el puntaje será de un máximo de 4 puntos. (10 puntos)

4. Encontrar el camino más corto entre ponis, considerando únicamente la relación de amistad. Se debe mostrar por consola la longitud y los nodos que lo componen. Si no hay camino entre dos ponis, se debe mostrar "No existe relación". Por consola debe solicitar los **nombres** de ambos ponis, no los nombres de los nodos (10 puntos)
5. Encontrar los amigos de los amigos de un pony. Se debe solicitar el nombre del pony por consola y retornar los nombres de los amigos de los amigos, sin contar a los amigos directos. Ejemplo: Applejack es amiga de Applebloom, y Applebloom es amiga de Sweetie Belle. En este caso, solo se debe retornar a Sweetie Belle. Todo debe realizarse en una única query, caso contrario el puntaje es 4 puntos. (10 puntos)
6. Retornar los ponis en los que el atributo "habilidad.esté relacionado con la magia (buscar la palabra "magia.en el atributo, cuidado con la sensibilidad a mayúsculas). (5 puntos)
7. Retornar nombres de los ponis que tienen relaciones de amistad unidireccionales, y con quién. (5 puntos)
8. Retornar la cantidad de pegasos, alicornios, ponis terrestres y unicornios que tienen preferencia por Coca-Cola o Sprite. Se debe solicitar por consola el tipo de pony y retornar la cantidad de preferencias por cada bebida. (5 puntos)
9. Retornar los ponis que tienen más enemigos que colaboraciones con otros ponis. (10 puntos)
10. Encontrar los ponis que tienen preferencia por Coca-Cola y que tengan al menos un amigo pony terrestre que prefiera Sprite. (10 puntos)
11. Discutir el trade-off entre el rendimiento de las consultas y el costo de mantenimiento de índices (Search performance indexes). Cómo se puede encontrar un equilibrio? Máximo 20 líneas. (10 puntos)
12. Es siempre adecuado usar EXPLAIN por sobre PROFILE en Neo4j? Mencione una situación donde PROFILE podría ser preferible a EXPLAIN. No más de 20 líneas. (15 puntos)



## 5. Sobre la entrega

1. La tarea se debe realizar en parejas utilizando el software mencionado, además deben inscribirse por aula en la sección correspondiente. Cualquier problema con su compañero debe informarle a su profesor.
2. Es necesario entregar un informe en pdf que contenga las respuestas a cada punto, con capturas que muestren el proceso y explicando cada paso brevemente. El informe debe tener nombre, rol y paralelo de los integrantes, puede ser realizado en word o latex.
3. Deben entregar el archivo .py comprimido en formato zip junto con un archivo readme.txt que contenga sus nombres y rol, se descontarán 10 puntos si no está en el .zip.
4. Neo4j desktop tiene un par de aplicaciones utiles como Neo4j browser y neo4j Bloom. La primera permite ejecutar consultas y ver el resultado en un grafo, La segunda también pero tiene una interfaz distinta. Puede ser util ver y manipular el dataset.
5. La fecha de entrega es hasta las **23:59 del jueves 17 de Octubre de 2024**. Consultas sobre la tarea pueden ser realizadas a mi correo: [daniela.sanchezn@usm.cl](mailto:daniela.sanchezn@usm.cl), no se responderán consultas 24 horas antes de la fecha de entrega. Errores en el formato del documento tendrán un descuento de 5 puntos, se descontarán 10 puntos por hora de atraso o fracción.