**PRÁCTICA FINAL SIE II:**

**ANÁLISIS DE UNA BASE DE DATOS  
Documentación**

Alejandro Fernández Muñoz

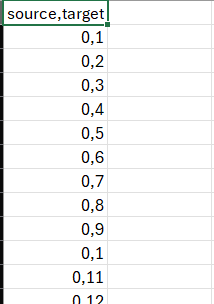
Gabriel Diaz Arevalillo

María Sánchez Nieto

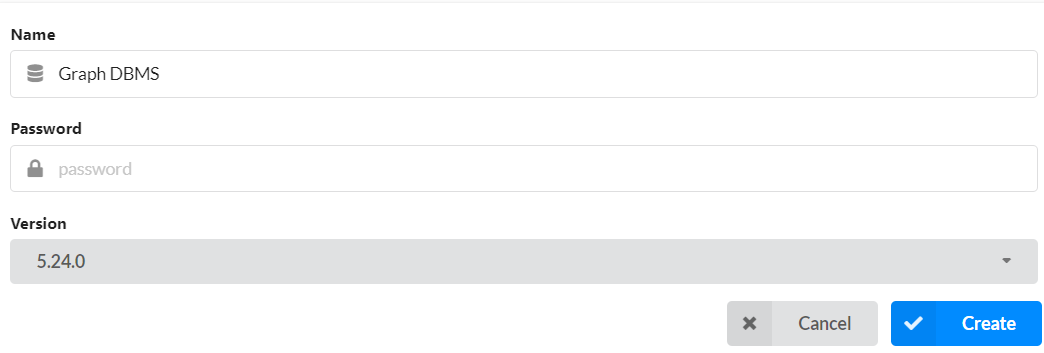
Daniel Gómez Obraztsov

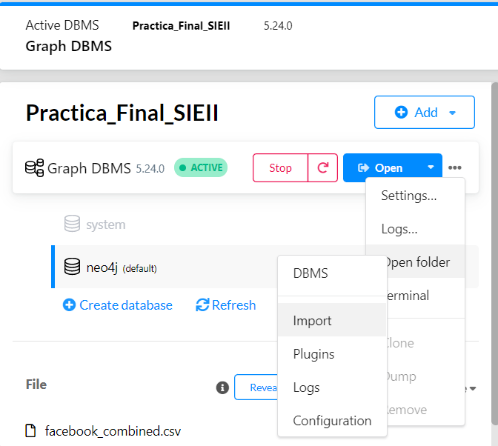
1. **Descargar el Dataset**
   1. Ir a esta página web: <https://snap.stanford.edu/data/egonets-Facebook.html>
   2. Descargar el archivo Facebook\_combined.txt.gz
   3. Lo encontraremos al final de la página
2. **Adaptar el dataset a un formato aceptado por neo4j**
   1. El txt nos vendrá con este formato



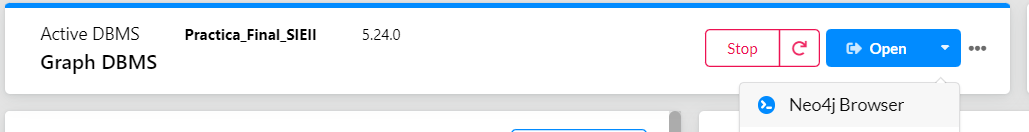
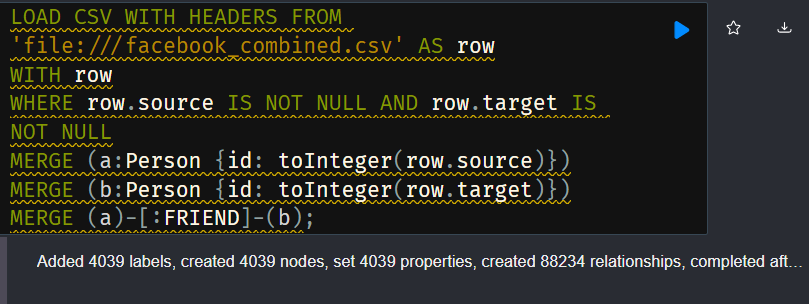
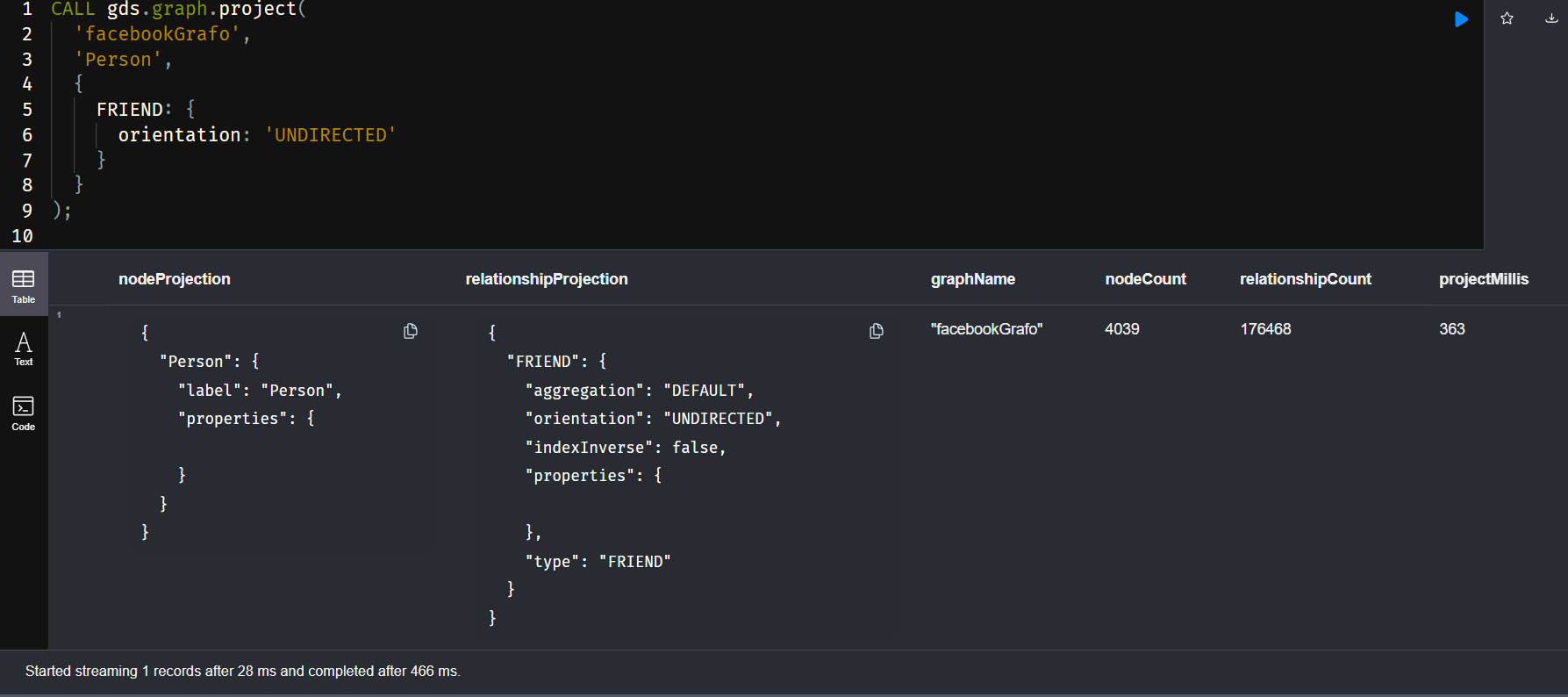
* 1. Tendremos que reemplazar los espacios por comas
  2. Y añadir arriba del todo esta línea: ‘source,target’
  3. Lo pasaremos a csv cambiando la extensión. Se tendría que ver tal que asíFFC
  4. 

1. **Creación de proyecto y carga de datos**
   1. Dentro de Neo4j desktop crearemos un nuevo proyecto y en la derecha pulsamos add àLocal DBMS
   2. Se nos abrirá una pestaña tal que así:

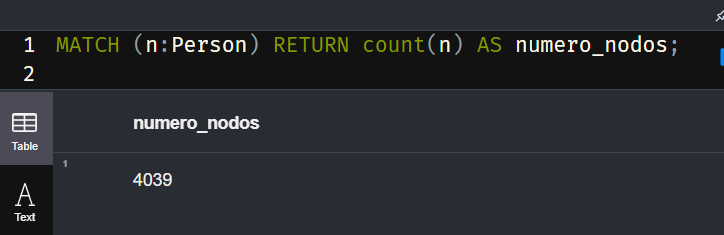


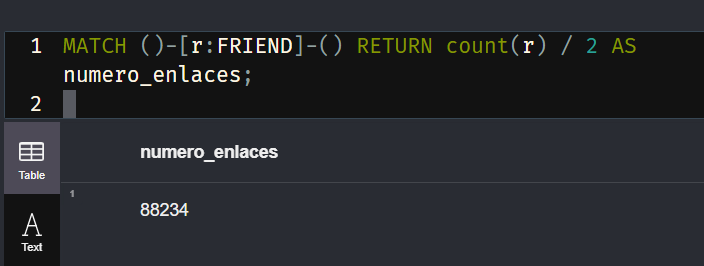
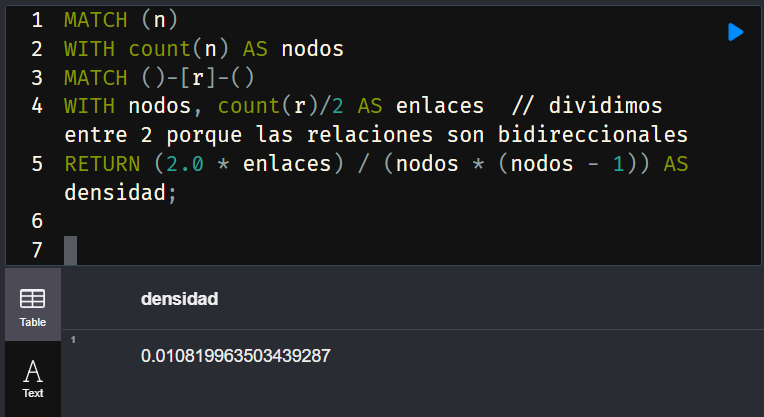
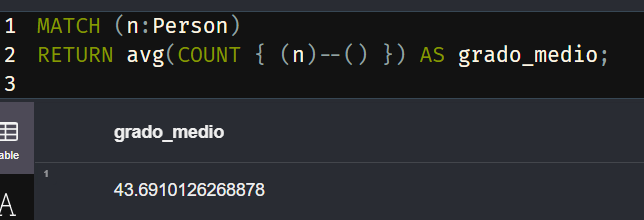
* 1. Ponemos la contraseña que configuramos al crear el usuario y le damos a crear
  2. Para añadir el documento csv a la base de datos vamos a los 3 puntitos al lado del boton azul open y vamos a aquí   
     

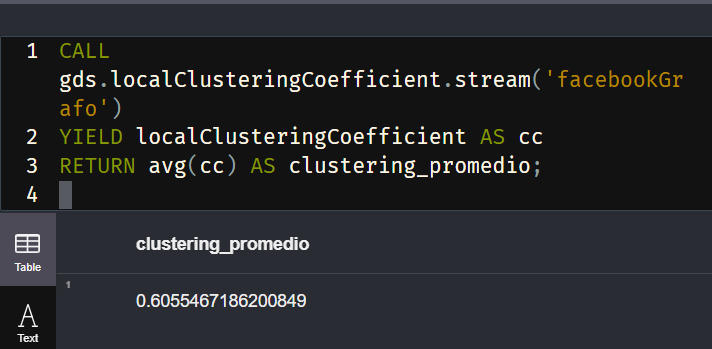
En esa carpeta metemos el csv

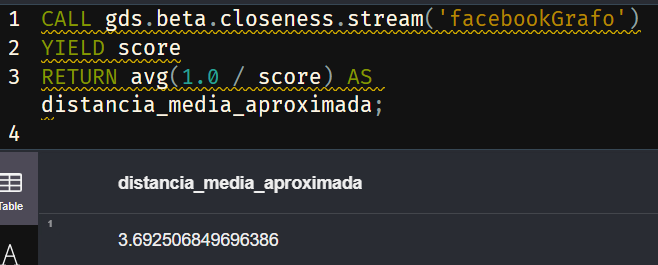
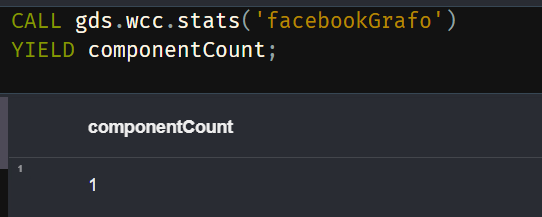
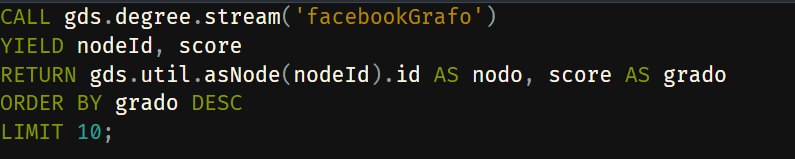
* 1. Volvemos a neo4j y hacemos à open à neo4j browser  
     
  2. Una vez dentro ejecutamos este comando para cargar los datos  
     
  3. Ahora creamos el grafo proyectado  
       
     El grafo proyectado es una copia temporal (cargada en la memoria RAM) y optimizada y preparada para para cálculos como centralidades, componentes conexas, clustering…

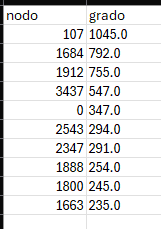
1. **Preparación de los datos para su posterior análisis**
   1. Nodos

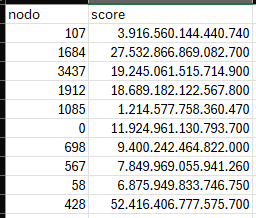
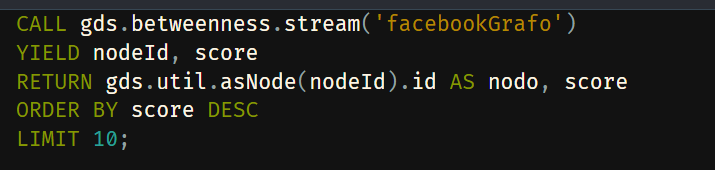
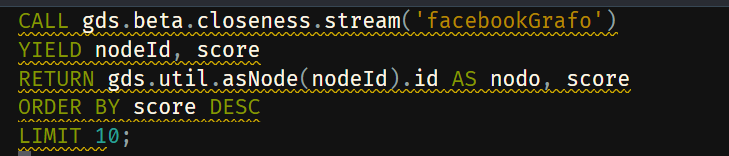


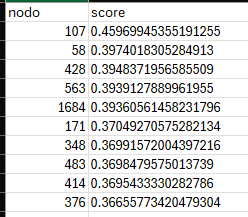
* 1. Enlaces  
     
  2. Densidad del grafo  
     
  3. Grado medio  
     
  4. Coeficiente de clustering promedio



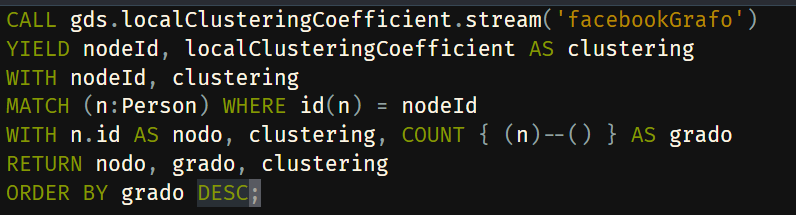
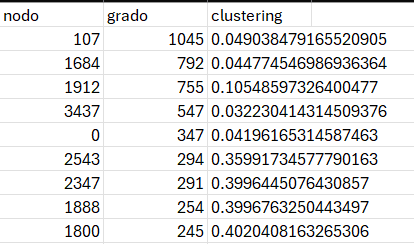
* 1. Distancia media  
     
  2. Componentes conexas  
     
  3. Centralidad de grado  
     



* 1. Intermediación  
     
  2. Cercanía  
     



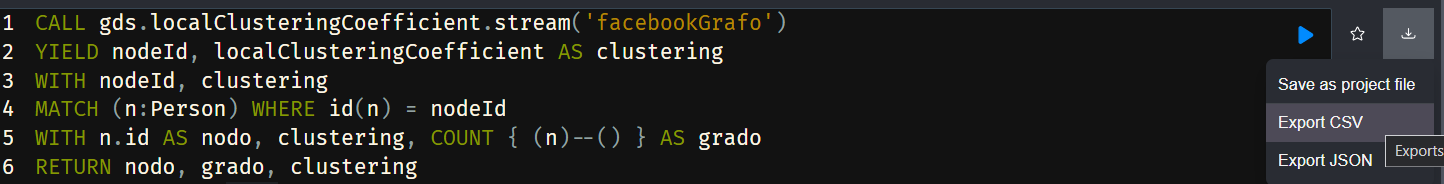
* 1. Relación clustering-grado

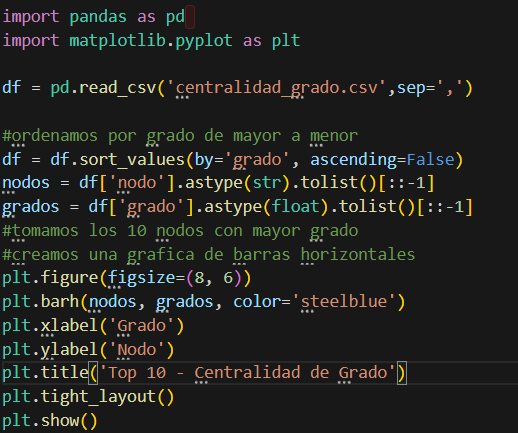
  


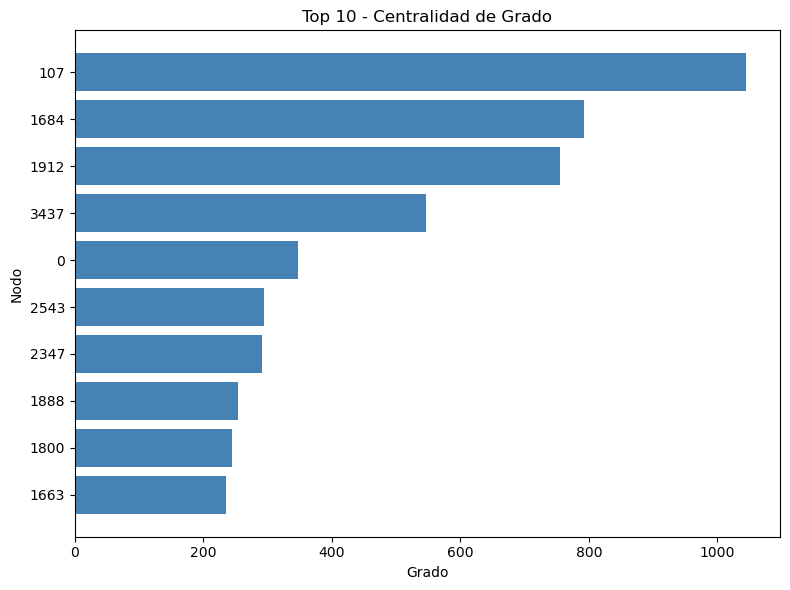
1. **Análisis de la estructura del grafo**
   1. Número de nodos obtenidos: 4039 --> esto quiere decir que hay 4039 usuarios en la red
   2. Número de enlaces: 88234 --> es el nº de amistades totales entre ellos
   3. Densidad del grafo: 0.0108 --> Quiere decir los usuarios tienen pocas conexiones en comparación a las que podrían tener
   4. Grado medio: 43.69 --> Es el nº de amistades que tiene cada usuario de media
   5. Coef. de clustering medio: 0.6055 --> esto quiere decir que hay una alta probabilidad de que se formen grupos cerrados de vecinos
      1. Mide cuanto tienden los vecinos de un nodo a estar conectados entre sí. Para un nodo la formula sería: C(u) = nº de enlaces entre vecinos de u / nº máximo de enlaces posibles entre ellos
   6. Distancia media: 3.69 --> los usuarios pueden llegar a cualquier otro a través de 3.7 amistades. Esto significa que la red presenta un comportamiento tipo 'mundo pequeño', similar a las redes sociales reales, donde la conexión global se logra con un nº bajo de intermediarios.
   7. Número de componentes conexas: 1 --> esto quiere decir que toda la red está conectada, no hay ningún grupo aislado.
      1. una componente conexa es un grupo de nodos donde cada nodo está conectado directa o indirectamente
2. **Análisis y comparación de las centralidades**

Como son muchos valores, se entenderá mejor con una gráfica

Para exportar los csv que vamos a utilizar para las gráficas se hace aquí



* 1. Centralidad de grado
     1. Importando como csv el resultado que nos dio el comando para generar las centralidades de grado, hemos hecho este programa en Python para generar la gráfica
     2. 

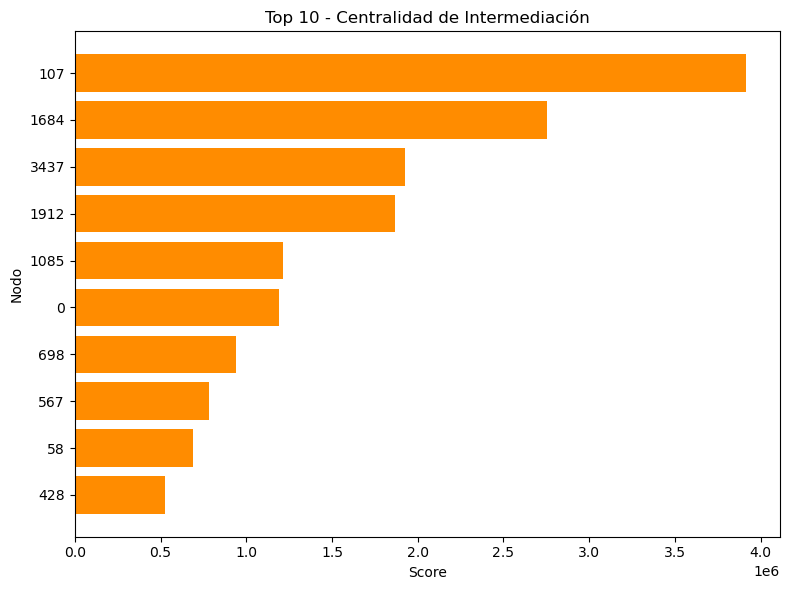


* + 1. La centralidad de grado mide la cantidad de conexiones directas que tienen un nodo con otros nodos
    2. El nodo 107 es el más conectado de toda la red (más de 1000 amigos).

Los nodos 1684, 1912 y 3437 también tienen altos grados, lo que sugiere que son muy populares.

A partir del nodo 0, el grado cae notablemente, aunque siguen siendo hubs locales importantes.

* 1. Intermediación
     1. El programa es el mismo solo que adaptándolo a intermediación

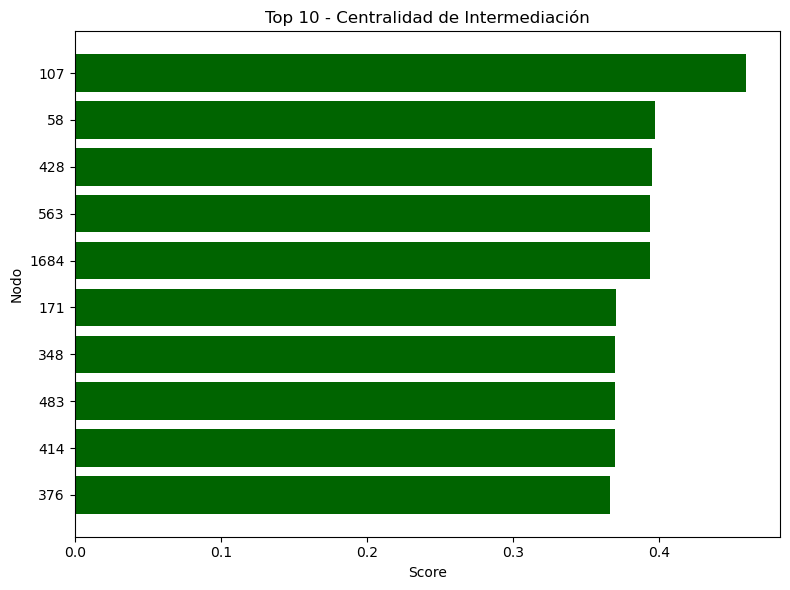


* + 1. La intermediación mide cuantas veces un nodo aparece en los caminos más cortos entre pares de otros nodos
    2. El nodo 107 no solo tiene muchas conexiones, sino que conecta partes importantes del grafo.

Los nodos 1684 y 3437 repiten como claves estructurales.

Aparecen nuevos nodos (como 1085, 698, 567) que no tienen tanto grado, pero sí un rol como “puente” entre comunidades.

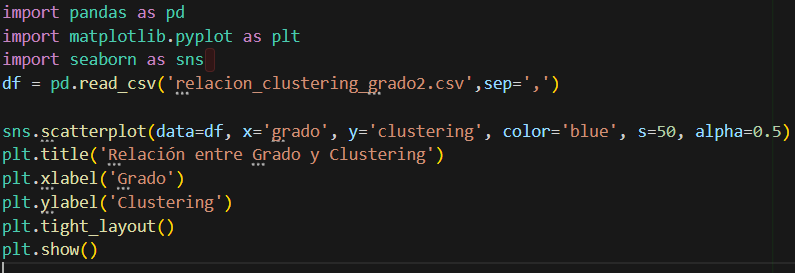
* 1. Cercanía
     1. También adaptamos el programa

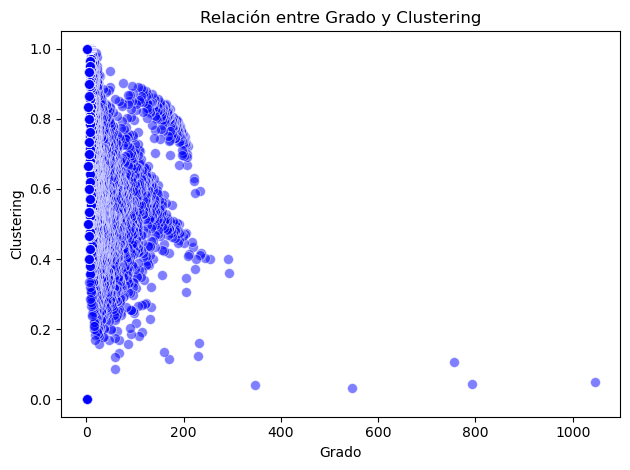


* + 1. La cercanía mide que tan ‘cerca’ está un nodo del resto de nodos en promedio
    2. El nodo 107 nuevamente es el más central: está más cerca del resto de la red que nadie.
    3. 1684 vuelve a aparecer (muy consistente en las 3 métricas).
    4. Nodos como 58, 428, 563, etc. no tienen tanto grado, pero tienen acceso rápido a casi toda la red (pueden “alcanzar” a otros nodos en pocos saltos).
  1. Comparación directa de los 3 top 10

|  |  |  |  |
| --- | --- | --- | --- |
| **Nodo** | **Grado** | **Intermediación** | **Cercanía** |
| 107 | ✅ | ✅ | ✅ |
| 1684 | ✅ | ✅ | ✅ |
| 1912 | ✅ | ✅ | ❌ |
| 3437 | ✅ | ✅ | ❌ |
| 0 | ✅ | ✅ | ❌ |
| 105 | ❌ | ✅ | ❌ |
| 698 | ❌ | ✅ | ❌ |
| 58 | ❌ | ✅ | ✅ |
| 428 | ❌ | ✅ | ✅ |
| 563 | ❌ | ❌ | ✅ |

* + 1. Nodo 107 es un supernodo absoluto: tiene más conexiones, conecta comunidades y está cerca de todos.
    2. Nodo 1684 también es muy importante en todos los sentidos.
    3. Nodos como 58 y 428 son interesantes: no tienen muchos amigos, pero son excelentes puentes y tienen acceso rápido al resto.
    4. Algunos nodos (como 698 o 105) tienen mucha intermediación sin tener tanto grado: eso sugiere que conectan comunidades separadas.

1. **Relación clustering – grado**
   1. Al analizar la relación de los 100 primeros nodos ordenados por grado de mayor a menor, obtenemos que:
      1. El nodo más conectado es el 107, con grado 1045 y clustering ≈ 0.5 à esto quiere decir que apenas un 5% de sus amigos están conectados entre sí.
      2. A medida que el grado desciende (grado 200-250), el clustering aumenta (0.5 – 0.79)
      3. Muchos nodos entre los grados 190 y 200 tienen un clustering por encima de 0.7 à esto indica que están altamente relacionados entre sí (muchos se “siguen” entre sí) lo que podríamos considerar grupos cerrados
   2. Este patrón encaja con las redes sociales
      1. Cuanto más popular es una persona, más diversos son sus contactos y menor probabilidad hay de que estén conectados entre sí. A diferencia de nodos con menor grado que suelen estar dentro de comunidades más cerradas donde todos o casi todos se conocen/se siguen.
   3. El programa utilizado para crear la gráfica es este:

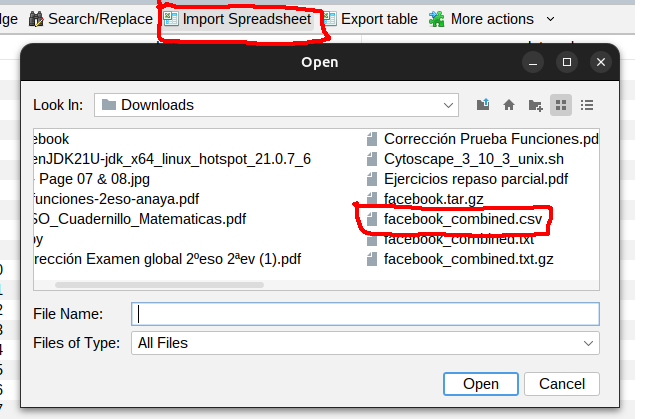


**DIAGRAMA DE NODOS**

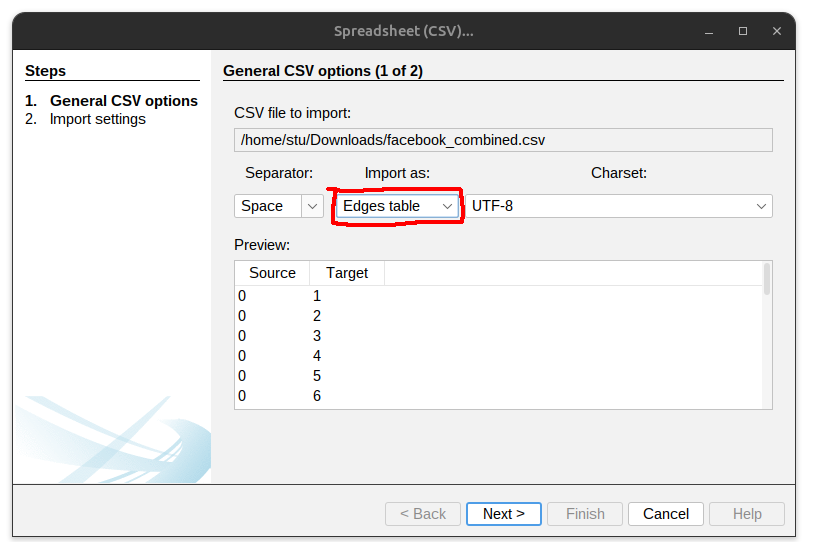
Para hacer el diagrama de nodos vamos a usar los datos de la página, específicamente el archivo facebook\_combined.txt.

Para generar este diagrama, vamos a usar el programa Gephi.

Convertimos el archivo de txt a csv, para que el programa lo acepte. Y después importamos el dataset.



El archivo facebook\_combined.txt proporciona una tabla de arcos/conexiones entre los nodos. Con esta tabla



Una vez se importan los datos, esto nos deja un cluster de nodos con esta pinta, solo son las conexiones, pero no los ordena entre grupos. Esto nos podría valer en nuestro caso para observar cuales están más conectados, pero no nos permite Observar los grupos que se forman.

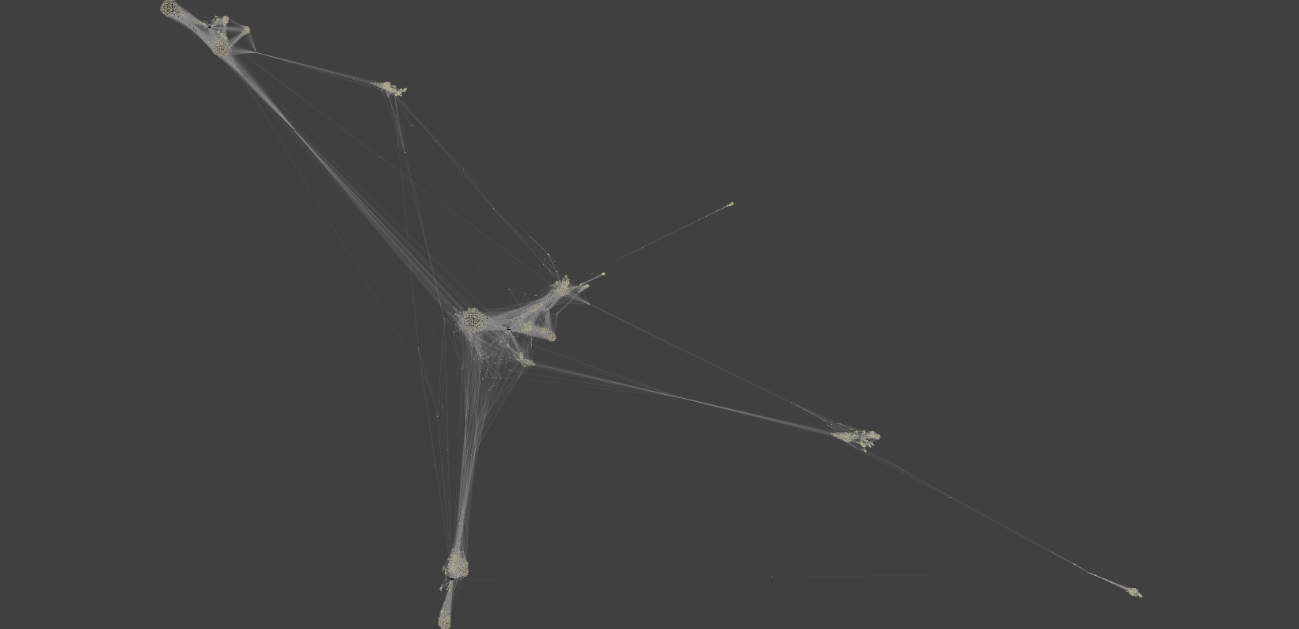


Para observar poder observar cómo se reparten los nodos, podemos usar algoritmos como ‘ForceAtlas 2’ o ‘Yifan Hu’. En nuestro caso hemos decidido usar el primero.

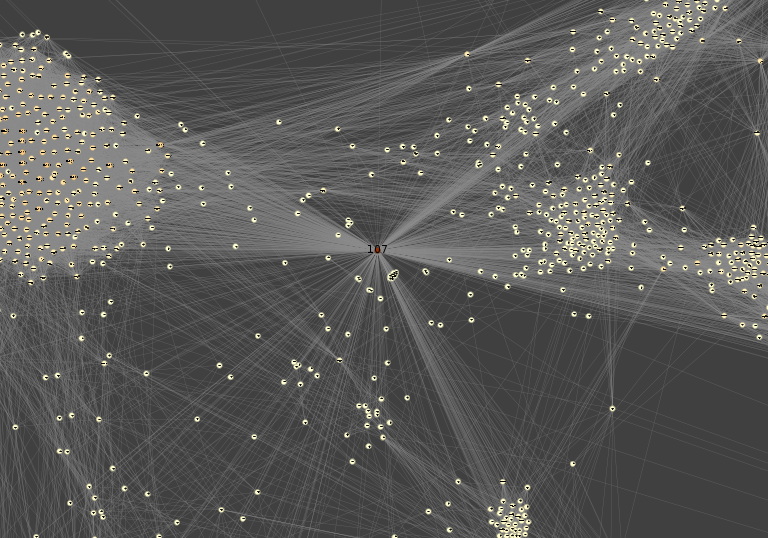
ForceAtlas 2 es un algoritmo muy interesante para usar en este caso, ya que se usa mucho para estudiar cómo reaccionan unos nodos a otros por sus conexiones, acercándose y rechazándose. Es una simulación física. Esto lo hace un algoritmo muy comúnmente usado para estudia grupos sociales.

Transforma la estructura de la red en proximidad espacial, haciendo que las comunidades y los nodos clave sean visualmente evidentes. Es altamente interactivo, configurable y muy adecuado para el análisis de redes sociales. Y afortunadamente ya está implementado en Gephi.

Una vez aplicado y dejando que corra durante aproximadamente 15 minutos (se para hasta la propia satisfacción, ya que no para por su cuenta, pero si deja de hacer cambios significativos). Y el resultado es el siguiente:



Se crean unos distintivos grupos, unas comunidades donde podemos ver que se dividen en varias zonas, conectadas entre sí por diferentes nodos que participan entre ellas.

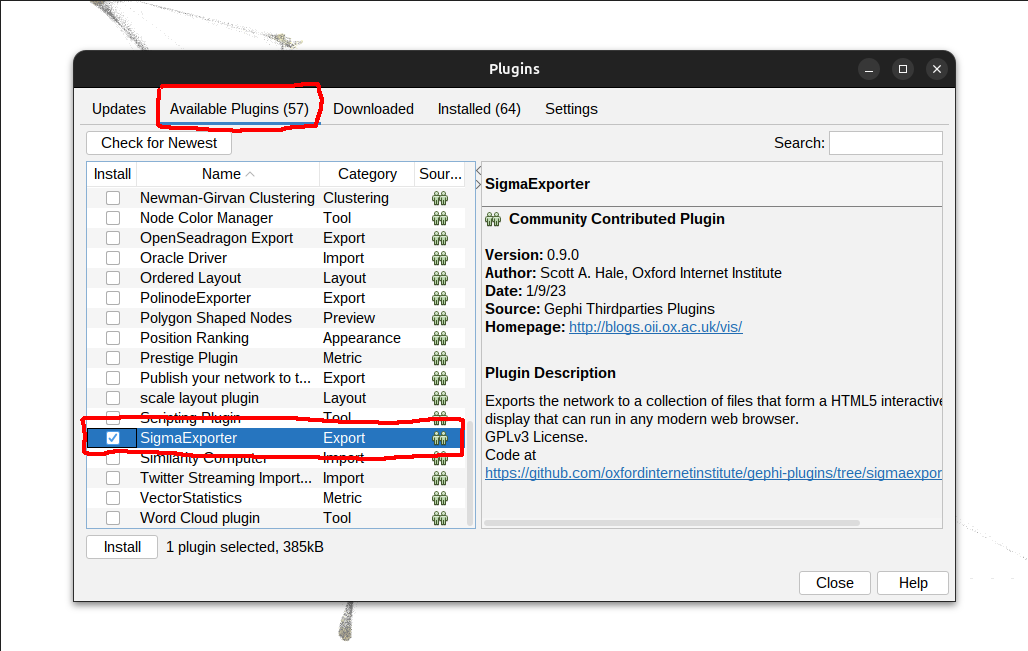
Si observamos el nodo de interés 107, el cual mencionamos como el que tiene mayor número de conexiones/grado.

Podemos sacar más información interesante haciendo un cálculo del degree, y ya que proviene de una tabla de edges/arcos, por lo que está dirigido. Con herramientas propias de Gephi, obtenemos un interesante resultado:

|  |  |  |
| --- | --- | --- |
| ID | In-Degree | Out-Degree |
| 107 | 2 | 1043 |

Esto nos está diciendo que las conexiones del nodo 107 son de él a otros nodos más que de otros nodos a él. Tanto esto, que la diferencia entre su In-Degree y su Out-Degree es la más grande de todos los nodos.

Para hacer una mejor visualización, decidimos expórtalo con un plug-in llamado ‘Sigma.js generator’ el cual permite crear una página web visualizando el diagrama de nodos.



Este funciona mejor si se hace host de un servidor web para visualizarlo:

*python3 -m http.server 8000*

Y tenemos el resultado final:

