Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Análisis de redes sobre una base de datos usando Neo4j y NetworkX

Marina Márquez Macías

Enero 2024

Índice general

1. Introducción **5**

2. Análisis y gestión de bases de datos con Neo4j **7**
2.1. Aplicación a la base de datos de Marvel 7

3. Análisis de redes con Python y NetworkX **13**
3.1. Análisis exploratorio de los datos 13
3.1.1. Representación gráfica 14
3.1.2. Detección de vecinos 15
3.2. Análisis de centralidad 17
3.3. Componente conexa 22
3.4. Algoritmo de Girvan-Newman 23

A. Algunos resultados obtenidos en Neo4j y Python **27**

Capítulo 1

Introducción

El análisis de redes aborda el estudio de sistemas complejos que están formados por entidades interconectadas. Se centra en describir los patrones de estas conexiones o relaciones entre los elementos que forman la red. Este análisis basa sus fundamentos en la teoría de grafos, la cual es una rama de las matemáticas que estudia las propiedades de los grafos. Los grafos son estructuras que representan relaciones entre diferentes objetos. Un grafo está compuesto por un conjunto de vértices (también llamados nodos) y un conjunto de aristas que conectan esos vértices. Estos elementos pueden modelar muchas situaciones en la vida real, desde redes sociales hasta sistemas de transporte, circuitos eléctricos o relaciones en biología. Con el auge del big data, el análisis de redes ha cobrado cada vez más importancia en todos estos campos.

En esta memoria, vamos a introducir los conceptos básicos del análisis de redes y su visualización, analizando una base de datos en concreto. Se va a realizar este estudio usando la plataforma `Neo4j` (con su lenguaje de consulta `Cypher`) y el lenguaje de programación `Python`, del cual usaremos en mayor medida su librería `NetworkX`.

La base de datos con la que vamos a trabajar tiene el nombre de *The Marvel Comics Characters Partnerships*, la cual está compuesta por dos archivos en formato `CSV`: `nodes.csv` y `edges.csv`.

En el primer archivo encontramos los personajes de Marvel, que representarán los nodos de nuestro grafo, junto con dos propiedades asociadas a ellos: su grupo ('group') y su tamaño ('size'). El grupo al que pertenecen los personajes puede ser el grupo de héroes (grupo 0), el de villanos (grupo 1) o el de antihéroes (grupo 2), y la propiedad tamaño indica el número de conexiones que tiene dicho nodo en el grafo. En total hay 350 personajes (nodos).

El archivo `edges.csv` contiene una columna de origen y una de destino. El archivo enumera las conexiones de los ID de los personajes especificados en el archivo `nodes.csv`. El valor de la propiedad tamaño ('size') de cada nodo significa la cantidad de veces que aparecerá el nombre de ese nodo en la columna de origen.

Esta base de datos está disponible en https://www.kaggle.com/datasets/trnguyen1510/the-marvel-comic-characters-partnerships.

Capítulo 2

Análisis y gestión de bases de datos con Neo4j

Las bases de datos relacionales *(SQL)* son aquellas en las que los datos se almacenan en tablas que contienen filas y columnas. Cada fila representa una instancia y cada columna representa un atributo de esa instancia. Durante mucho tiempo, han sido las bases de datos más utilizadas. Sin embargo, a medida que las tecnologías fueron avanzando, empezaron a surgir limitaciones sobre estos modelos relacionales, como por ejemplo: el escalado con datos grandes y de alta velocidad, la necesidad de trabajar con grandes volúmenes de datos *(Big Data)*, y con datos demasiado complejos para estos modelos.

Por esta razón, se introdujo un nuevo modelo no relacional para satisfacer estas necesidades. Se implantaron las bases de datos *NoSQL (Not Only SQL)*. Estas bases de datos pueden ser de varios tipos: Key-Value (formada por pares clave-valor), orientadas a documentos, bases de datos de columnas, y orientadas a grafos. En esta memoria vamos a trabajar con bases de datos NoSQL orientadas a grafos.

Para ello, vamos a usar la plataforma de gestión de base de datos **Neo4j**, con su lenguaje de consulta **Cypher**. Mediante Neo4j, podremos realizar operaciones de inserción, modificado, borrado y consulta de los datos en bases de datos orientadas a grafos, así como obtener representaciones de las bases de datos en forma de grafos.

2.1. Aplicación a la base de datos de Marvel

Nuestra base de datos consta de dos archivos en formato `CSV`, uno que contiene los nodos del grafo (`nodes.csv`), los cuales serán los personajes de Marvel, acompañados de dos propiedades, el grupo al que pertenecen (group) y su tamaño (size). El otro archivo contiene a las aristas del grafo (`edges.csv`).

Para empezar, podemos representar nuestra base de datos como un grafo. Para ello, usamos Neo4j, escribiendo los siguientes comandos.

1. Contamos cuantos nodos y cuantas aristas tenemos. Usamos el código siguiente.

```
LOAD CSV WITH HEADERS FROM "file:///nodes.csv" AS row
RETURN count(row)
>count(row)
```

```
>350
```

Podemos observar que hay 350 nodos en total, cada uno representa a un personaje de Marvel, y tiene tres propiedades: su nombre, su grupo y su tamaño.

```
LOAD CSV WITH HEADERS FROM "file:///edges.csv" AS row
RETURN count(row)
>count(row)
>346
```

Hemos obtenido que hay 346 aristas, es decir, 346 relaciones entre los personajes de Marvel.

Podemos visualizar en una tabla todos los nodos y sus propiedades. Con el siguiente comando, imponemos que el nombre de las tres propiedades de cada nodo: (characterId, group, size).

```
LOAD CSV WITH HEADERS FROM "file:///nodes.csv" AS row
WITH toInteger(row.group) AS group, row.id AS characterId,
toInteger(row.size) AS size RETURN group, characterId, size;
```

La tabla resultante se puede observar en el Apéndice de esta memoria.

También podemos crear una tabla que nos muestre el origen y el destino de cada arista, de la forma siguiente:

```
LOAD CSV WITH HEADERS FROM "file:///edges.csv" AS row
WITH row.source AS origin, row.target AS destiny
RETURN origin, destiny;
```

De nuevo, podemos observar esta tabla en el Apéndice de la memoria.

2. Pasamos ahora a ver una representación en forma de grafo de nuestra base de datos. Para empezar, representamos los nodos de la siguiente manera.

```
LOAD CSV WITH HEADERS FROM "file:///nodes.csv" AS row
WITH toInteger(row.group) AS group, row.id AS characterId,
toInteger(row.size) AS size MERGE (n:Node {characterId: characterId})
SET n.group= group, n.size=size RETURN count(row);
>count(row)
>350
```

```
MATCH (n:Node) RETURN n;
```

El resultado de este comando, el cual es la representación de todos los nodos, se muestra en la Figura 2.1. Sin embargo, solo se visualizan 300 nodos ya que es el límite de representación del lenguaje Cypher. En ellos, el atributo característico es el nombre del personaje.

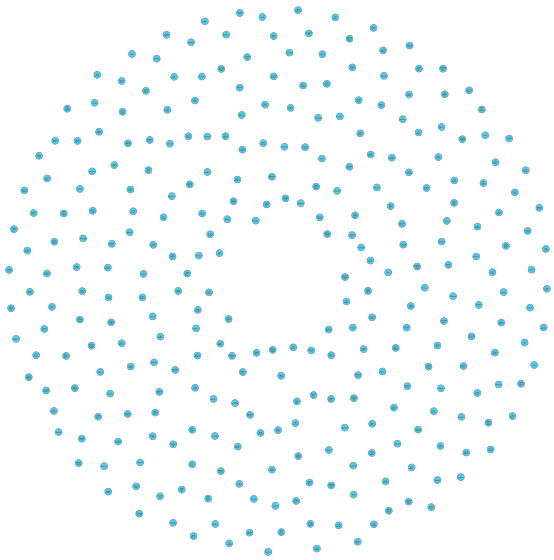
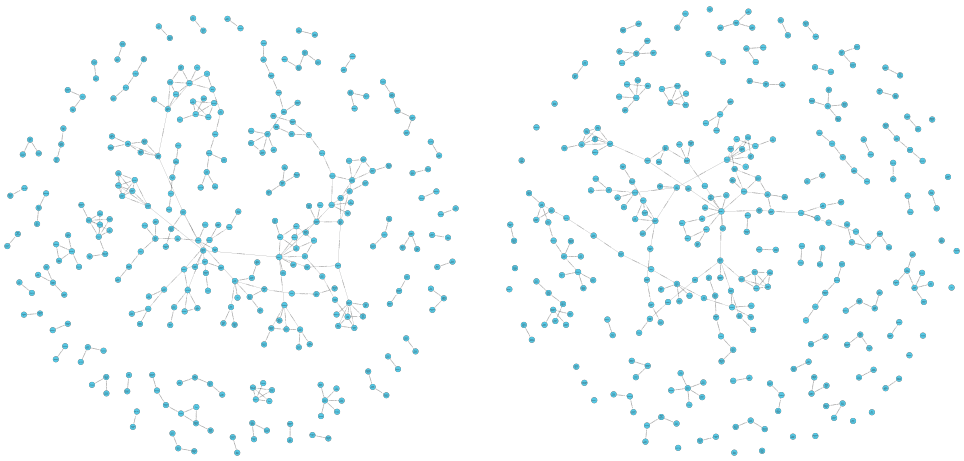


Figura 2.1: Nodos

Ahora vamos a introducir las aristas, es decir, las relaciones entre los nodos.

```
LOAD CSV WITH HEADERS FROM "file:///edges.csv" AS row
WITH row.source AS origin, row.target AS destiny
MATCH (n1:Node characterId: origin) MATCH (n2:Node characterId: destiny)
MERGE (n1)-[rel:COLABORATE_WITH]->(n2) RETURN count(rel);
>count(rel)
>346

MATCH (n1:Node)-[rel:COLABORATE_WITH]->(n2:Node) RETURN n1, rel, n2;
MATCH (n:Node) RETURN n;
```

El resultado de estos comandos, se muestra en la Figura 2.2. De nuevo, solo se visualizan 300 nodos.



(a) Nodos con conexiones (b) Grafo total con las conexiones

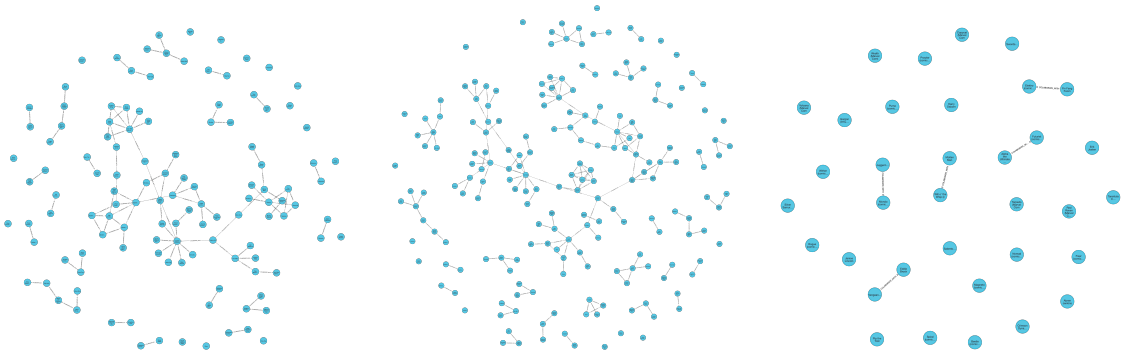
Figura 2.2: Nodos y aristas

Podemos también visualizar los grafos correspondientes a cada grupo de personajes: el grupo de héroes (0), el grupo de villanos (1) y el grupo de antihéroes

(2).

Para ello, ejecutamos los siguientes coódigos, y observamos los tres grafos resultantes en la Figura 2.3. También podemos apreciar que hay 121 personajes del grupo de héroes, 194 que son villanos y 35 personajes considerados antihéroes.

```
MATCH (n:Node {group: 0}) RETURN count(n);
>count(n)
>121

MATCH (n:Node {group: 0}) RETURN n;

MATCH (n:Node {group: 1}) RETURN count(n);
>count(n)
>194

MATCH (n:Node {group: 1}) RETURN n;

MATCH (n:Node {group: 2}) RETURN count(n);
>count(n)
>35

MATCH (n:Node {group: 2}) RETURN n;
```



(a) Grupo 0 (b) Grupo 1 (c) Grupo 2

Figura 2.3: Conexiones entre los nodos de cada grupo

3. Para analizar los datos más útiles en esta base de datos, podemos observar cuáles son los nodos (personajes) con más relaciones a partir de la propiedad `size`, ya que esta característica de los nodos nos dice cuántas conexiones tiene dicho nodo. Lo realizamos mediante la siguiente consulta.

```
MATCH (n:Node) RETURN n ORDER BY n.size DESC;
```

Podemos observar cuales son los 11 personajes que tienen más conexiones con otros personajes en el Cuadro 2.1. Notamos que el personaje "Spider-Man" es el que tiene más relaciones, exactamente tiene 12 relaciones. Le siguen "Captain

America" con 10 relaciones y "Red Skull" con 9 conexiones.

(:Node {size: 12,characterId: "Spider-Man",group: 0})
(:Node {size: 10,characterId: "Captain America",group: 0)}
(:Node {size: 9,characterId: "Red Skull",group: 1})
(:Node {size: 8,characterId: "Unicorn (comics)",group: 1})
(:Node {size: 8,characterId: "Attuma",group: 1})
(:Node {size: 7,characterId: "Iron Fist (comics)",group: 0})
(:Node {size: 6,characterId: "Chameleon (comics)",group: 1})
(:Node {size: 6,characterId: "Venom (Marvel Comics character)",group: 1})
(:Node {size: 6,characterId: "Scarecrow (Marvel Comics)",group: 1}
(:Node {size: 6,characterId: "Grim Reaper (comics)",group: 1})
(:Node {size: 6,characterId: "Loki (comics)",group: 1})

Cuadro 2.1: Orden de los personajes en función a su propiedad tamaño (descendiente)

Capítulo 3

Análisis de redes con Python y NetworkX

En este capítulo vamos a tratar de crear y manipular nuestro grafo asociado a la base de datos de los personajes de Marvel, medir alguna de sus propiedades y descubrir comunidades y patrones. Vamos a realizar todo esto en el lenguaje de programación `Python`. En particular, usaremos la biblioteca `NetworkX`, la cual sirve para el estudio de grafos y análisis de redes.

Las librerías que usaremos serán las siguientes:

```
import networkx as nx
import networkx.algorithms.community as nx_comm
from sklearn import tree
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
import matplotlib as mpl
from matplotlib.lines import Line2D
import matplotlib.colors as mcolors
```

3.1. Análisis exploratorio de los datos

Nuestro conjunto de datos consta de 350 nodos que representan personajes de Marvel, y 346 aristas que representan las relaciones entre ellos. Para empezar, vamos a cargar nuestros datos en Python mediante la librería `Pandas`, cargando por un lado el archivo CSV de las aristas, y por otro el archivo CSV de los nodos.

```
edges = pd.read_csv('edges.csv')
nodes_data = pd.read_csv('nodes.csv')
```

Podemos observar la forma de los datos que hemos cargado `(edges, nodes_data)`. Mediante el siguiente código veremos las primeras 5 filas de nuestros conjuntos de datos en la Figura 3.1.

```
edges.head(5)
nodes_data.head(5)
```

Ahora que tenemos los datos almacenados con la librería `Pandas`, creamos la red utilizando una función de NetworkX, la cual es `nx.from_pandas_edgelist`.

	source	target
0	Baron Zemo	Blackout (Marcus Daniels)
1	Baron Zemo	Vermin (comics)
2	N'astirh	S'ym
3	N'astirh	Belasco (Marvel Comics)
4	N'astirh	Madelyne Pryor

(a) Conjunto de aristas

	group	id	size
0	1	Baron Zemo	2
1	1	N'astirh	3
2	0	Silver Sable	1
3	0	Hope Summers (comics)	1
4	0	Magik (Illyana Rasputin)	1

(b) Conjunto de nodos

Figura 3.1: Formato de nuestros conjuntos de datos

```
g = nx.from_pandas_edgelist(df=edges, source='source',target='target')
print(g)
```

```
>Graph with 350 nodes and 346 edges
```

Asociamos el atributo de interés para los nodos. En nuestro caso será el correspondiente a la columna *id*, atributo que representa el nombre de cada personaje de Marvel.

```
nodes_attributes = nodes_data.set_index('id').to_dict('index')
nx.set_node_attributes(g, nodes_attributes)
```

También podemos obtener una descripción del grafo que hemos creado.

```
print('Numero de nodos:', g.number_of_nodes())
print('Numero de aristas:', g.number_of_edges())

#Lista de nodos
print('Nodos:', g.nodes)

#Lista de aristas
print('Aristas:', g.edges)
```

```
>Número de nodos: 350
>Número de aristas: 346
>Nodos: ['Baron Zemo', 'Blackout (Marcus Daniels)', 'Vermin (comics)', "N'astirh", "S'ym", 'Belasco (Marvel
Comics)', 'Madelyne Pryor', ..., 'Black Talon (comics)', 'Kid Kaiju', 'Elsa Bloodstone', 'Jack Flag', 'Rikki
Barnes', 'Arnim Zola', 'Apocalypse (Marvel Comics)', 'Mister Sinister']
>Aristas: [('Baron Zemo', 'Blackout (Marcus Daniels)'), ('Baron Zemo', 'Vermin (comics)'), ("N'astirh", "S'ym"),
("N'astirh", 'Belasco (Marvel Comics)'), ("N'astirh", 'Madelyne Pryor'), ("S'ym", 'Belasco (Marvel Comics)'), ...,
('Devil Dinosaur', 'Moon Girl (Marvel Comics)'), ('Devil Dinosaur', 'Moon-Boy'), ('Moon Girl (Marvel Comics)',
'Kid Kaiju'), ('Spider-Girl', 'Buzz (comics)'), ('Jocasta (comics)', 'Ultron'), ('Psycho-Man', 'Shellshock (Marvel
Comics)'), ('Kid Kaiju', 'Elsa Bloodstone'), ('Apocalypse (Marvel Comics)', 'Mister Sinister')]
```

Podemos observar mediante la siguiente consulta los atributos de cualquier nodo. En este caso vemos los atributos del personaje *Spider-Man*, es decir, su grupo y su tamaño.

```
g.nodes(data=True)['Spider-Man']
```

```
>{'group': 0, 'size': 12}
```

3.1.1. Representación gráfica

Vamos a representar nuestro grafo con la función `nx.draw` de NetworkX. Para empezar, determinamos la posición de nuestros nodos con la función `nx.spring_layout`.

En cuanto a las aristas, vamos a asociarles distintos colores. Asociaremos el color azul a las aristas que conecten nodos de distintos grupos (héroes, villanos o antihéroes), el color rojo a las aristas que conecten nodos del grupo de héroes, el color verde a aquellas que conecten nodos del grupo de villanos y el color amarillo a las aristas que conecten nodos del grupo de antihéroes.

También vamos a definir la estética de los nodos. El tamaño de cada nodo será proporcional al grado que presente dicho nodo. Los nodos del grupo de héroes, serán nodos rojos; los nodos que representen villanos, tendrán el color verde; y los nodos del grupo de antihéroes serán nodos amarillos.

```
fig, ax = plt.subplots(figsize=(20, 15))

# Posicion de los nodos
pos = nx.spring_layout(g)

# Color de los enlaces
edge_colors = []
for u, v in g.edges():
    group_u = nx.get_node_attributes(g, 'group')[u]
    group_v = nx.get_node_attributes(g, 'group')[v]
    if group_u != group_v:
        edge_colors.append('blue')
    elif group_u == 0 :
        edge_colors.append('red')
    elif group_u == 1:
        edge_colors.append('green')
    else:
        edge_colors.append('yellow')

# Estetica de los nodos
nodes_size  = [10 * g.degree[node] for node in g.nodes()]
nodes_groups = list(nx.get_node_attributes(g, 'group').values())
nodes_colors = []
for group in nodes_groups:
  if group == 0:
    nodes_colors.append('red')
  elif group == 1:
    nodes_colors.append('green')
  else:
    nodes_colors.append('yellow')

# Leyenda
custom_lines = [Line2D([0], [0], color='blue', alpha=0.5, lw=4),
                Line2D([0], [0], color='red', alpha=0.5, lw=4),
                Line2D([0], [0], color='green', alpha=0.5, lw=4),
                Line2D([0], [0], color='yellow', alpha=0.5, lw=4)]

ax.legend(custom_lines, ['Conexiones entre grupos', 'Heroes', 'Villanos','
    Antiheroes'])

nx.draw_networkx_nodes(g, pos=pos, node_size=nodes_size, node_color=nodes_colors,
    ax=ax)
nx.draw_networkx_labels(g, pos=pos, ax=ax, font_size=3, font_color='grey')
nx.draw_networkx_edges(g, pos=pos, edgelist=g.edges,
                       edge_color=edge_colors, ax=ax);
```

El resultado de introducir este código se puede observar en la Figura 3.2.

3.1.2. Detección de vecinos

Para obtener los vecinos de un nodo concreto, podemos utilizar la función `neighbors()` de NetworkX que devuelve un iterador sobre todos los vecinos del nodo elegido. Por ejemplo, vamos a identificar a los vecinos del nodo *'Spider-Man'*, ya que es el nodo que más conexiones tiene.

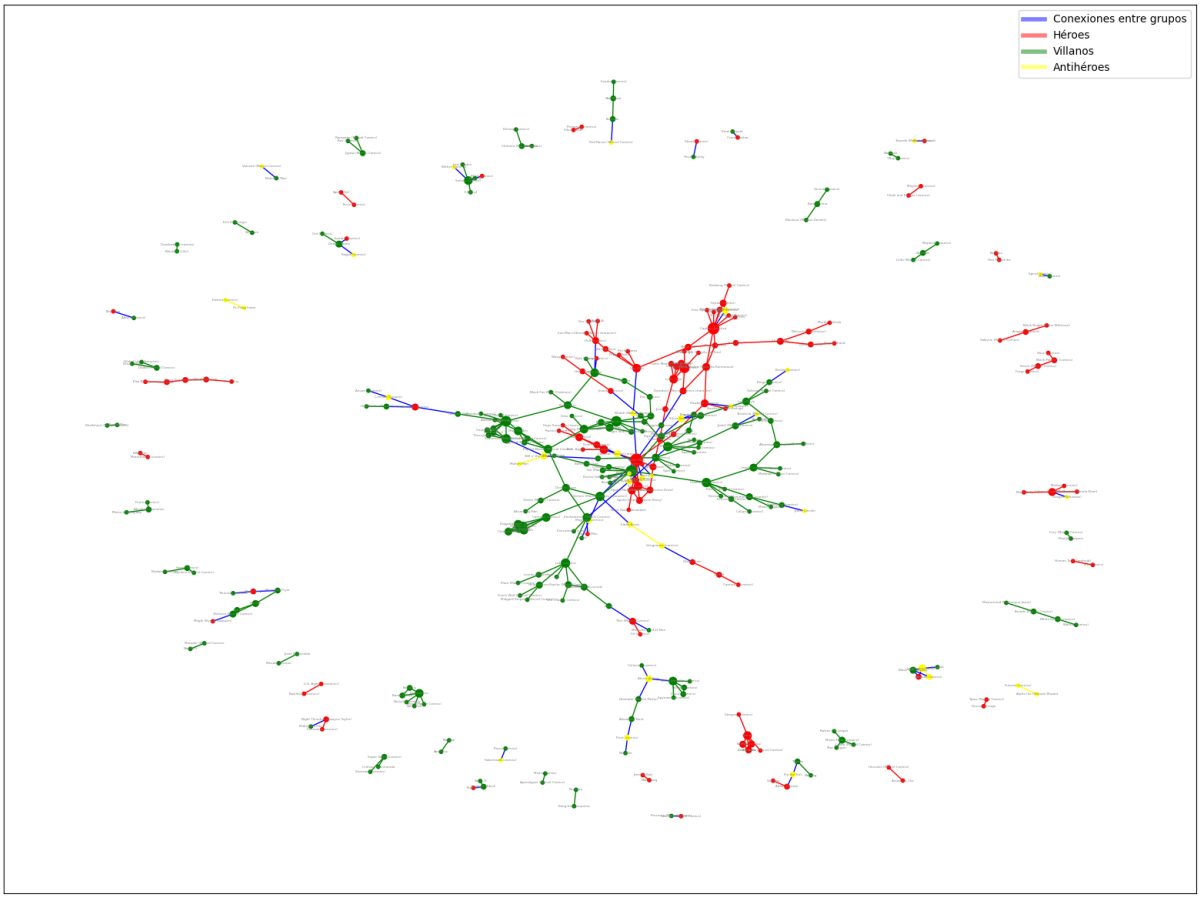


Figura 3.2: Representación gráfica con Python

```
node = 'Spider-Man'
neighbors = list(nx.neighbors(g,node))
print(neighbors)
```

```
>['Venom (Marvel Comics character)', 'Anya Corazon', 'Spider-Man (Miles Morales)', 'Prowler (comics)', 'Harry
Osborn', 'Mockingbird (Marvel Comics)', 'Deadpool', 'Wraith (Marvel Comics)', 'Puma (comics)', 'Black Cat (Marvel
Comics)', 'Kingpin (character)', 'Silk (comics)']
```

También podemos detectar los vecinos de un orden concreto con la función `ego_graph()` de NetworkX, la cual devuelve un subgrafo que contiene a los nodos conectados al nodo elegido por caminos del orden especificado. Vamos a fijar un orden de 3 unidades, es decir, buscamos caminos de hasta tres aristas.

```
node = 'Spider-Man'
order = 3
ego = nx.ego_graph(g, node, radius=order)
neighbors = list(ego.nodes())
print(neighbors)
```

```
>['Silk (comics)', 'Carnage (comics)', 'Nova (Sam Alexander)', 'Hawkeye (comics)', 'Garrison Kane', 'Hope Summers
(comics)', 'Garokk', 'Sin (Marvel Comics)', 'Vengeance (comics)', 'Deadpool', 'Black Cat (Marvel Comics)', 'War
Machine', 'Anya Corazon', 'Spider-Woman (Jessica Drew)', 'Daredevil (Marvel Comics character)', 'Enchantress
(Marvel Comics)', 'Copycat (Marvel Comics)', 'Titania (Marvel Comics)', 'Puppet Master (Marvel Comics)', 'Kingpin
(character)', 'Spider-Woman (Gwen Stacy)', 'Prowler (comics)', 'Bob, Agent of Hydra', 'Riri Williams', 'Baron
Strucker', 'Shriek (comics)', 'Hawkeye (Kate Bishop)', 'Maria Hill', 'Arnim Zola', 'Magneto (comics)', 'Wraith
(Marvel Comics)', 'Puma (comics)', 'Iron Man', 'Red Skull', 'Domino (comics)', 'Doppelganger (comics)',
'Spider-Man (Miles Morales)', 'Crossbones (comics)', 'Rachel Summers', 'Doctor Doom', 'Cable (comics)', 'Carrion
(comics)', 'Spider-Man', 'Iron Fist (comics)', 'Harry Osborn', 'Doctor Faustus (comics)', 'Pepper Potts', 'Viper
(Madame Hydra)', 'Power Broker', 'Venom (Marvel Comics character)', 'Chameleon (comics)', 'Demogoblin', 'Jessica
Jones', 'Carol Danvers', 'Eddie Brock', 'Swordsman (comics)', 'Mockingbird (Marvel Comics)', 'Black Widow (Natasha
Romanova)']
```

3.2. Análisis de centralidad

Las métricas de centralidad caracterizan a los nodos de un grafo, determinando la importancia o relevancia de estos dentro del grafo. Hay varias formas de medir la importancia de los nodos en un grafo, por lo tanto, tenemos distintas métricas de centralidad, como son:

- La centralidad de grado. Indica que un nodo es más importante o relevante cuanto mayor sea su grado, es decir, cuantas más conexiones tenga dentro del grafo.
- La centralidad de cercanía (o closeness). Con esta medida, un nodo es relevante si su distancia promedio a todos los demás nodos es baja.
- La centralidad betweenness. Indica que un nodo es más relevante si pertenece a más caminos mínimos. Es decir, los nodos más relevantes son los que hacen de puente en comunicaciones directas entre otros nodos.
- La centralidad eigenvector. Dentro de este esquema los nodos que poseen una alta importancia están conectados de forma más eficiente dentro de un grafo determinado o posee un alto nivel de conexiones y por lo tanto son buenos candidatos para difundir información.
- Coeficiente de Clustering. Indica cómo de conectados entre sí están los nodos. El coeficiente de clustering de un nodo será alto si todos los vecinos de ese nodo están conectados entre sí.

Vamos a calcular las métricas de centralidad de cada nodo, y las ordenaremos de mayor a menor:

1. **Centralidad de grado.**

```
cgrado = nx.algorithms.degree_centrality(g)
sorted(cgrado.items(), key= lambda elemento : elemento[1], reverse=True)

```

```
>[('Spider-Man', 0.034383954154727794), ('Captain America', 0.028653295128939826), ('Red Skull',
0.025787965616045846), ('Unicorn (comics)', 0.022922636103151862), ('Attuma', 0.022922636103151862), ('Iron
Fist (comics)', 0.020057306590257878), ('Venom (Marvel Comics character)', 0.017191977077363897), ('Loki
(comics)', 0.017191977077363897), ... , ('Rikki Barnes', 0.0028653295128939827), ('Arnim Zola',
0.0028653295128939827), ('Apocalypse (Marvel Comics)', 0.0028653295128939827), ('Mister Sinister',
0.0028653295128939827)]
```

2. **Centralidad de cercanía.**

```
ccerc = nx.algorithms.closeness_centrality(g)
sorted(ccerc.items(), key= lambda elemento : elemento[1], reverse=True)

```

```
>[('Venom (Marvel Comics character)', 0.11038843783325213), ('Doctor Doom', 0.10720170463945156),
('Spider-Man', 0.10246873754720202), ('Puppet Master (Marvel Comics)', 0.09762005911436912), ('Red Skull',
0.09690676014380484), ('Enchantress (Marvel Comics)', 0.09367979436706865), ('Carnage (comics)',
0.09191750120570795), ('Eddie Brock', 0.09173584606498522), ... , ('Shellshock (Marvel Comics)',
0.0028653295128939827), ('Apocalypse (Marvel Comics)', 0.0028653295128939827), ('Mister Sinister',
0.0028653295128939827)]
```

3. **Centralidad betweenness.**

```
cbet = nx.algorithms.betweenness_centrality(g)
sorted(cbet.items(), key= lambda elemento : elemento[1], reverse=True)

```

```
>[('Venom (Marvel Comics character)', 0.1762754009814577), ('Doctor Doom', 0.14998517933010572),
('Spider-Man', 0.13016253334650724), ('Puppet Master (Marvel Comics)', 0.10640417613542798), ('Krang (Marvel
Comics)', 0.0945064716925205), ... , ('Arnim Zola', 0.0), ('Apocalypse (Marvel Comics)', 0.0), ('Mister
Sinister', 0.0)]
```

4. **Centralidad eigenvector.**

```
ceigen = nx.algorithms.eigenvector_centrality(g)
sorted(ceigen.items(), key= lambda elemento : elemento[1], reverse=True)

```

```
>[('Carrion (comics)', 0.4184390227206002), ('Demogoblin', 0.4184390227206002), ('Shriek (comics)',
0.4184390227206002), ('Doppelganger (comics)', 0.41843902272060013), ('Carnage (comics)',
0.37839257728461917), ... , ('Shellshock (Marvel Comics)', 1.8843809643712696e-37), ('Apocalypse (Marvel
Comics)', 1.8843809643712696e-37), ('Mister Sinister', 1.8843809643712696e-37)]
```

5. **Coeficiente de clustering.**

```
cclust = nx.algorithms.cluster.clustering(g)
sorted(cclust.items(), key= lambda elemento : elemento[1], reverse=True)

```

```
>[("S'ym", 1.0), ('Siryn', 1.0), ('Demolition Man (comics)', 1.0), ('Llyra', 1.0), ('Bucky Barnes', 1.0),
... , ('Arnim Zola', 0), ('Apocalypse (Marvel Comics)', 0), ('Mister Sinister', 0)]
```

Podemos visualizar una tabla con las medidas de centralidad de cada nodo usando el siguiente código. Visualizamos en la Figura 3.3 las primeras filas de esta tabla.

```
grado       = pd.Series(dict(nx.degree(g)), name='degree')
cercania     =  pd.Series(nx.closeness_centrality(g), name='closeness')
betweenness =  pd.Series(nx.betweenness_centrality(g), name='betweenness')
eigenvector =  pd.Series(nx.eigenvector_centrality(g), name='eigenvector')
clustering = pd.Series(nx.cluster.clustering(g), name = 'clustering')
centralidad = pd.concat([grado, eigenvector, cercania, betweenness, clustering],
    axis=1)
centralidad.index.name = 'node'
centralidad.head(5)
```

node	grado	eigenvector	cercania	betweenness	clustering
Baron Zemo	2	6.528657e-31	0.005731	0.000016	0.000000
Blackout (Marcus Daniels)	1	4.616458e-31	0.003820	0.000000	0.000000
Vermin (comics)	1	4.616458e-31	0.003820	0.000000	0.000000
N'astirh	3	1.038081e-19	0.010315	0.000148	0.333333
S'ym	2	8.675562e-20	0.007935	0.000000	1.000000

Figura 3.3: Tabla de métricas de centralidad para cada nodo

Con el siguiente código obtendremos los nodos con mayor valor en cada una de las métricas de centralidad:

```
for col in centralidad.columns:
top_nodos = centralidad[col].sort_values(ascending=False).head(5).index.to_list()
top_nodos.sort()
print(f"Nodos con mayor {col}: {top_nodos}")
```

```
>Nodos con mayor degree: ['Attuma', 'Captain America', 'Red Skull', 'Spider-Man', 'Unicorn (comics)']
>Nodos con mayor eigenvector: ['Carnage (comics)', 'Carrion (comics)', 'Demogoblin', 'Doppelganger (comics)', 'Shriek
(comics)']
>Nodos con mayor closeness: ['Doctor Doom', 'Puppet Master (Marvel Comics)', 'Red Skull', 'Spider-Man', 'Venom (Marvel
Comics character)'] >Nodos con mayor betweenness: ['Doctor Doom', 'Krang (Marvel Comics)', 'Puppet Master (Marvel Comics)',
'Spider-Man', 'Venom (Marvel Comics character)']
>Nodos con mayor clustering: ['Colleen Wing', 'Crossbones (comics)', 'Discus (comics)', 'Sin (Marvel Comics)', 'Siryn']
```

Se puede observar que *'Spider-Man', 'Venom (Marvel Comics character)'* y *'Doctor Doom'* son nodos con bastante relevancia, ya que tienen valores altos en más de una métrica.

Vamos a representar ahora el grafo coloreado por cada métrica de centralidad. El resultado se observa en la Figura 3.4. Para ello usamos el siguiente código. Ahora, la posición de los nodos va a estar dada por la función `nx.kamada_kawai_layout` de NetworkX, y vamos a normalizar las métricas para poder compararlas. El tamaño de los nodos estará ponderado por su valor en cada métrica.

```python
fig, axs = plt.subplots(1, 5, figsize=(9, 3.5))
pos = nx.kamada_kawai_layout(g)

# Mapa de colores
cmap =  mpl.colormaps['Blues']

# Escalado de las metricas para pooder compararlas
scaler = MinMaxScaler(feature_range=(0, 1))
scaler.fit(centralidad)
centralidad_scaled = pd.DataFrame(
                        scaler.transform(centralidad),
                        columns=scaler.feature_names_in_)
centralidad_scaled.head(3)

for i, metrica in enumerate(centralidad_scaled.columns):
    node_size = 100 * centralidad_scaled[metrica] + 1
    node_color = cmap(centralidad_scaled[metrica])
    nx.draw_networkx_nodes(
        g,
        pos=pos,
        node_size=node_size,
        node_color=node_color,
        linewidths=0.5,
        edgecolors='black',
        ax=axs[i])
    nx.draw_networkx_edges(g, pos=pos, edgelist=g.edges, alpha=0.3, ax=axs[i])
    axs[i].set_title(metrica)
    axs[i].axis('off')
```

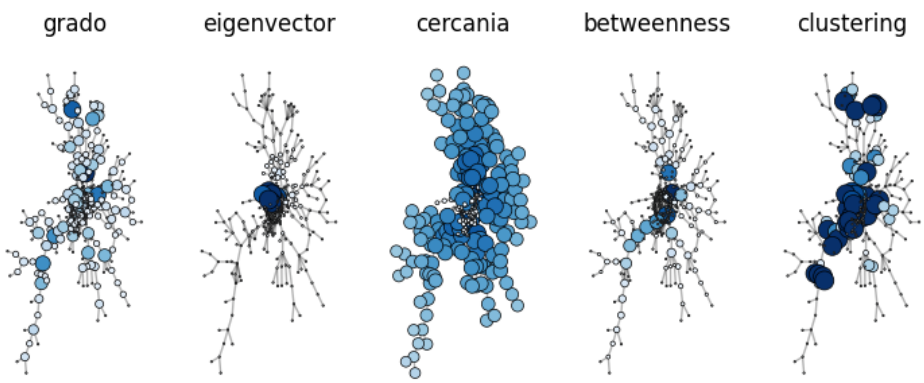


Figura 3.4: Representación del grafo coloreado por cada métrica

A continuación vamos a crear un modelo de árbol de decisión en el que intervengan dos de estas métricas o características. Por ejemplo, la centralidad de grado y la

centralidad de cercanía. Creamos el árbol con el módulo `DecisionTreeClassifier` de `Scikit-learn` y lo entrenamos con el módulo `train_test_split`. La etiqueta a predecir es el grupo al que pertenece cada nodo.

```
centralidad_grado=list(nx.algorithms.degree_centrality(g).values())
centralidad_cercania=list(nx.algorithms.closeness_centrality(g).values())
clf=tree.DecisionTreeClassifier(max_depth=3)
X=list([x,y] for x,y in zip(centralidad_grado, centralidad_cercania))
y=list(nx.get_node_attributes(g,"group").values())
X_train, X_test, y_train, y_test = train_test_split(X,y) #coge algunos ejemplos al
    azar de X e y
clf.fit(X_train,y_train)
```

Graficamos el árbol (Figura 3.5) y calculamos el rendimiento del modelo en los conjuntos de entrenamiento, de prueba y total.

```
tree.plot_tree(clf)
print('Rendimiento del modelo sobre el conjunto de entrenamiento:', clf.score(
    X_train,y_train))
print('Rendimiento del modelo sobre el conjunto de test:', clf.score(X_test,y_test)
    )
print('Rendimiento del modelo sobre el conjunto total:', clf.score(X,y))
plt.show()
```

```
>Rendimiento del modelo sobre el conjunto de entrenamiento:
0.5916030534351145
Rendimiento del modelo sobre el conjunto de test: 0.5681818181818182
Rendimiento del modelo sobre el conjunto total: 0.5857142857142857
```

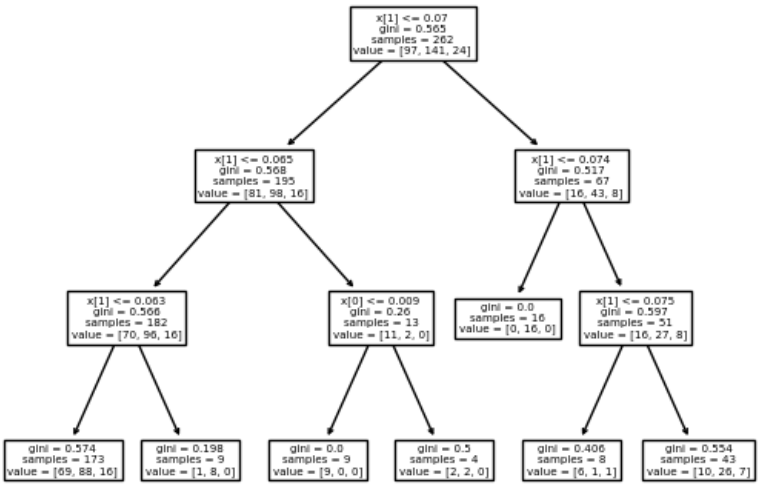


Figura 3.5: Modelo de árbol de decisión con la centralidad de grado y de cercanía.

Podemos observar que el rendimiento es bastante bajo. Esto se debe a que el grupo al que pertenece un nodo es difícil de predecir según las métricas de centralidad.

Probamos a predecirlo según otras métricas, como por ejemplo la centralidad betweenness y la centralidad eigenvector. El árbol correspondiente se puede observar en la Figura 3.6.

```
centralidad_betweness=list(nx.algorithms.betweenness_centrality(g).values())
centralidad_eigenvector=list(nx.algorithms.eigenvector_centrality(g).values())
clf1=tree.DecisionTreeClassifier(max_depth=3)
X1=list([x,y] for x,y in zip(centralidad_betweness, centralidad_eigenvector))
X1_train, X1_test, y1_train, y1_test = train_test_split(X1,y)
clf1.fit(X1_train,y1_train)
tree.plot_tree(clf1)
print('Rendimiento del modelo sobre el conjunto de entrenamiento:', clf1.score(
    X1_train,y1_train))
print('Rendimiento del modelo sobre el conjunto de test:', clf1.score(X1_test,
    y1_test))
print('Rendimiento del modelo sobre el conjunto total:', clf1.score(X1,y))
plt.show()
```

```
Rendimiento del modelo sobre el conjunto de entrenamiento:
0.5763358778625954
Rendimiento del modelo sobre el conjunto de test: 0.5795454545454546
Rendimiento del modelo sobre el conjunto total: 0.5771428571428572
```

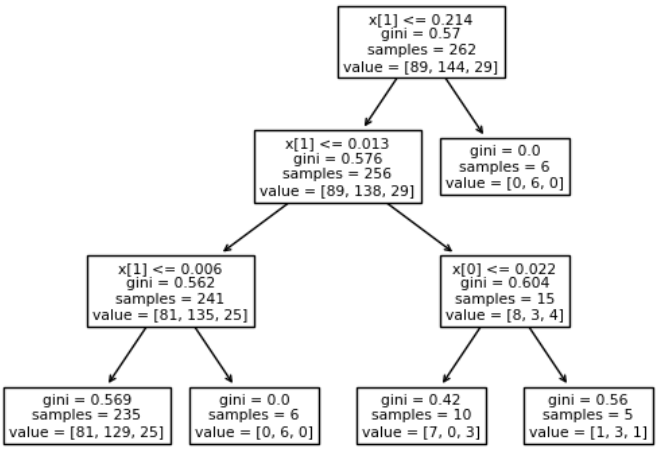


Figura 3.6: Árbol de decisión con la centralidad betweenness y eigenvector.

De nuevo, estos resultados no son satisfactorios. Probamos un último árbol de decisión, con la centralidad eigenvector y el coeficiente de clustering. El árbol está representado en la Figura 3.7.

```
coeficientes_clustering=list(nx.algorithms.cluster.clustering(g).values())
clf2=tree.DecisionTreeClassifier(max_depth=3)
X2=list([x,y] for x,y in zip(centralidad_eigenvector, coeficientes_clustering))
X2_train, X2_test, y2_train, y2_test = train_test_split(X2,y)
clf2.fit(X2_train,y2_train)
tree.plot_tree(clf2)
print('Rendimiento del modelo sobre el conjunto de entrenamiento:', clf2.score(
    X2_train,y2_train))
print('Rendimiento del modelo sobre el conjunto de test:', clf2.score(X2_test,
    y2_test))
print('Rendimiento del modelo sobre el conjunto total:', clf2.score(X2,y))
plt.show()
```

```
Rendimiento del modelo sobre el conjunto de entrenamiento:
0.5801526717557252
Rendimiento del modelo sobre el conjunto de test: 0.5795454545454546
Rendimiento del modelo sobre el conjunto total: 0.58
```

Obtenemos un rendimiento similar en todos los casos que hemos probado, el cual no es muy alto.

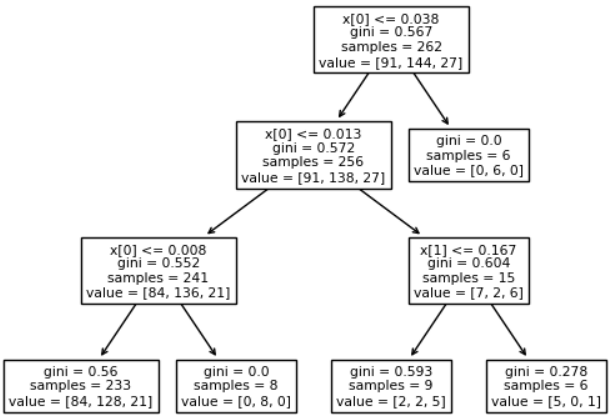


Figura 3.7: Árbol de decisión con la centralidad eigenvector y coeficientes de clustering.

3.3. Componente conexa

En esta sección, vamos a encontrar la componente conexa más grande del grafo. Para ello, obtenemos todas las componentes conexas de nuestro grafo mediante la función `nx.connected_components` de NetworkX, calculamos su número de nodos, y nos quedamos con la componente conexa más grande.

```
# Lista del numero de nodos de todas las componentes conexas del grafo
[len(c) for c in sorted(nx.connected_components(g), key=len, reverse=True)]
```

```
[181, 11, 7, 6, 6, 6, 5, 5, 5, 5, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,2,2]
```

Se puede notar que la componente conexa con más nodos tiene 181 nodos.

```
mayorcc = max(nx.connected_components(g), key=len)
g_mayorcc = g.subgraph(mayorcc).copy()
print(g_mayorcc)
```

```
Graph with 181 nodes and 224 edges
```

Representamos esta componente conexa como un grafo en la Figura 3.8 mediante el siguiente código:

```
plt.figure(figsize=(10,8))
pos = nx.spring_layout(g_mayorcc)
nx.draw_networkx_nodes(g_mayorcc, pos=pos, node_size=10, node_color='red')
nx.draw_networkx_labels(g_mayorcc, pos=pos, font_size=3)
nx.draw_networkx_edges(g_mayorcc, pos=pos, edge_color='blue')
```

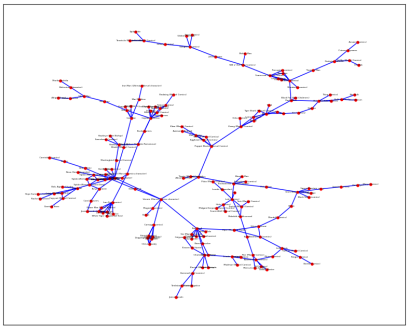


Figura 3.8: Componente conexa con más nodos

3.4. Algoritmo de Girvan-Newman

El Algoritmo de Girvan Newman es un método de orden jerárquico que sirve para detectar comunidades dentro de sistemas complejos de datos. Este algoritmo se centra en la eliminación iterativa de los enlaces por los que pasa el mayor número de caminos más cortos del grafo (mayor valor de centralidad betweenness). Al eliminar enlaces uno a uno, el grafo se descompone en subgrafos más pequeños que llamaremos comunidades. El algoritmo Girvan-Newman se divide en cuatro pasos principales:

1. Calcular la centralidad betweenness de cada enlace en el grafo.
2. Eliminar el enlace con mayor valor de esta centralidad. Si todos son iguales, se escoge aleatoriamente.
3. Calcular de nuevo los valores de esta métrica para los enlaces restantes.
4. Repetir los pasos 2, 3 y 4 hasta que no queden enlaces.

Podemos implementar este algoritmo en `Python` usando la función `girvan_newman` de la librería `NetworkX`.

```
communities_generator = nx_comm.girvan_newman(g, most_valuable_edge=None)
```

Calculamos la primera iteración de este algoritmo con el siguiente código, y se puede observar que en esta iteración ya ha conseguido obtener todas las comunidades de nuestro grafo. Podemos ver la lista completa de comunidades en el Apéndice de esta memoria.

```
communities = next(communities_generator)
communities
```

También podemos representar gráficamente las comunidades obtenidas dentro del grafo. Vamos a representar las 9 primeras comunidades obtenidas. Los nodos de la primera comunidad serán rojos, los de la segunda tendrán el color azul, los de la tercera serán rosas, los nodos de la cuarta comunidad irán asociados al color amarillo, los de la quinta serán naranjas, los nodos de la sexta tendrán color negro, los de la séptima comunidad serán morados, los de la octava marrones y los nodos de la novena comunidad obtenida serán de color violeta. Los nodos que no pertenecen a ninguna de estas 9 comunidades serán de color verde. El resultado se puede observar en la Figura 3.9.

```
communities_generator = nx_comm.girvan_newman(g, most_valuable_edge=None)
communities = next(communities_generator)

node_colors = []
for node in g:
    if node in communities[0]:
        node_colors.append('red')
    elif node in communities[1]:
        node_colors.append('blue')
    elif node in communities[2]:
        node_colors.append('pink')
    elif node in communities[3]:
        node_colors.append('yellow')
    elif node in communities[4]:
        node_colors.append('orange')
    elif node in communities[5]:
        node_colors.append('black')
```

```
    elif node in communities[6]:
        node_colors.append('purple')
    elif node in communities[7]:
        node_colors.append('brown')
    elif node in communities[8]:
        node_colors.append('violet')
    else:
        node_colors.append('green')

fig, ax = plt.subplots(figsize=(8, 5))
nx.draw(g, node_color=node_colors, node_size=10, with_labels=False, ax=ax)
plt.show()
```

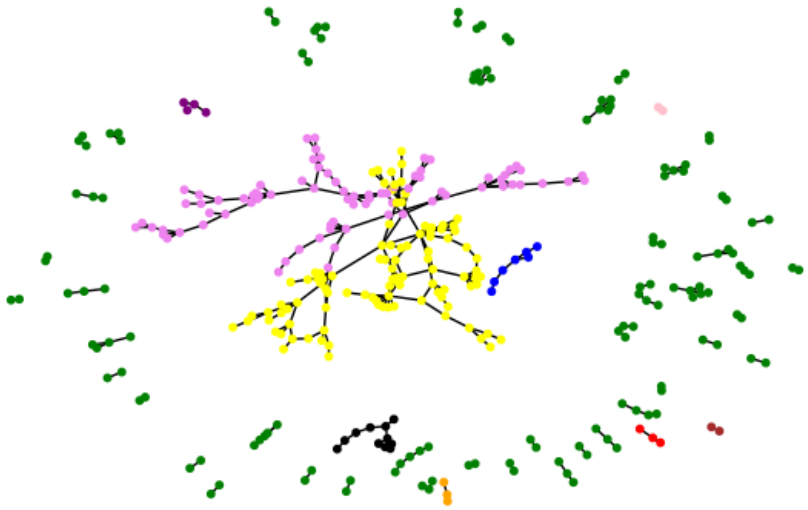


Figura 3.9: Primeras 9 comunidades obtenidas con el algoritmo Girvan-Newman

Se puede notar que las comunidades más grandes que podemos obtener son las coloreadas de amarillo y violeta, las cuales corresponden a la cuarta y la novena comunidad obtenida. Podemos observar cuáles son estas comunidades en la lista obtenida anteriormente, la cual está visible en el Apéndice de esta memoria. Dichas comunidades son las siguientes:

- Cuarta comunidad obtenida:

```
{'Abigail Brand', 'Abomination (comics)', 'Anya Corazon', 'Arnim Zola', 'Baron Strucker', 'Beast (comics)',
'Beetle (comics)', 'Black Cat (Marvel Comics)', 'Black Widow (Natasha Romanova)', 'Bob, Agent of Hydra',
'Boomerang (comics)', 'Bucky (Marvel Comics)', 'Bucky Barnes', 'Cable (comics)', 'Calypso (comics)',
'Captain America', 'Caretaker (comics)', 'Carnage (comics)', 'Carol Danvers', 'Carrion (comics)', 'Chameleon
(comics)', 'Cletus Kasady', 'Colleen Wing', 'Copycat (Marvel Comics)', 'Crossbones (comics)', 'Daredevil
(Marvel Comics character)', 'Deadpool', 'Demogoblin', 'Demolition Man (comics)', 'Doctor Faustus (comics)',
'Doctor Octopus', 'Domino (comics)', 'Doppelganger (comics)', 'Eddie Brock', 'Electro (Marvel Comics)',
'Falcon (comics)', 'Free Spirit (comics)', 'Galaxy Master', 'Garokk', 'Garrison Kane', 'Ghost Rider', 'Glob
(comics)', 'Hammerhead (comics)', 'Harry Osborn', 'Hawkeye (Kate Bishop)', 'Hawkeye (comics)', 'Helmut
Zemo', 'Hope Summers (comics)', 'Iron Fist (comics)', 'Iron Man', 'Iron Man (Ultimate Marvel character)',
'Jack Flag', 'Janice Lincoln', 'Jessica Jones', 'Kingpin (character)', 'Kraven the Hunter', 'Leader
(comics)', 'Leila Davis', 'Luke Cage', 'Madman (Marvel Comics)', 'Magneto (comics)', 'Maria Hill', 'Mariko
Yashida', 'Master Izo', 'Mister Negative', 'Misty Knight', 'Mockingbird (Marvel Comics)', 'Nick Fury',
'Nomad (comics)', 'Nova (Sam Alexander)', 'Pepper Potts', 'Power Broker', 'Power Man (Victor Alvarez)',
'Prowler (comics)', 'Puma (comics)', 'Rachel Summers', 'Red Skull', 'Redwing (Marvel Comics)', 'Rhino
(comics)', 'Rikki Barnes', 'Ringer (comics)', 'Riri Williams', 'Shiver Man', 'Shriek (comics)', 'Shriker',
'Silk (comics)', 'Silver Samurai', 'Sin (Marvel Comics)', 'Spider-Man', 'Spider-Man (Miles Morales)',
'Spider-Woman (Gwen Stacy)', 'Spider-Woman (Jessica Drew)', 'Swordsman (comics)', 'Tombstone (comics)',
'Vengeance (comics)', 'Venom (Marvel Comics character)', 'Viper (Madame Hydra)', 'Vulture (Marvel Comics)',
'War Machine', 'White Tiger (Angela del Toro)', 'Wolverine (character)', 'Wraith (Marvel Comics)'}
```

- Novena comunidad encontrada:

```
'Absorbing Man', 'Answer (comics)', 'Attuma', 'Awesome Android', 'Black Fox (Raul Chalmers)', 'Black Talon
(comics)', 'Byrrah', 'Count Nefaria', 'Crimson Dynamo', 'Darkstar (comics)', 'Doc Samson', 'Doctor Doom',
```

```
'Eel (comics)', 'Egghead (Marvel Comics)', 'Enchantress (Marvel Comics)', 'Erik Josten', 'Executioner
(comics)', 'Fenris Wolf (Marvel Comics)', 'Ghaur', 'Gibbon (comics)', 'Giganto', 'Griffin (Marvel Comics)',
'Grim Reaper (comics)', 'Grizzly (comics)', 'Hank Pym', 'Headlok', 'Hela (comics)', 'Hulk (comics)', 'Jackal
(Marvel Comics)', 'Jocasta (comics)', 'Jonas Harrow', 'Kala (comics)', 'Kangaroo (comics)', 'Klaw (Marvel
Comics)', 'Krang (Marvel Comics)', 'Kro (comics)', 'Kurse', 'Lemuel Dorcas', 'Llyra', 'Loki (comics)',
'Lorelei (Asgardian)', 'Mad Thinker', 'Malekith the Accursed', 'Man-Ape', 'Mandarin (comics)', 'Mandrill
(comics)', 'Mercurio the 4-D Man', 'Midgard Serpent (Marvel Comics)', 'Mole Man', 'Molten Man', 'Nekra',
'Orka (comics)', 'Plantman', 'Pluto (Marvel Comics)', 'Porcupine (comics)', 'Puppet Master (Marvel Comics)',
'Red Ghost', 'Scarecrow (Marvel Comics)', 'She-Hulk', 'Sif (comics)', 'Space Phantom', 'Spidercide', 'Spot
(comics)', 'Surtur (Marvel Comics)', 'Tarantula (Marvel Comics)', 'Thing (comics)', 'Thor (Marvel Comics)',
'Tiger Shark (Marvel Comics)', 'Titania (Marvel Comics)', 'Titanium Man', 'Tyrak', 'Ulik', 'Ultron',
'Unicorn (comics)', 'Wasp (comics)', "Will o' the Wisp (comics)", 'Wizard (Marvel Comics)', 'Wonder Man',
'Ymir (Marvel Comics)'
```

Nótese que cualquiera de estas dos comunidades sería útil para hacer posibles introducciones de personajes en películas o comics de Marvel, ya que los personajes de estas comunidades están muy conectados entre sí.

Apéndice A

Algunos resultados obtenidos en Neo4j y Python

La tabla obtenida en Neo4j que representa a los nodos del grafo y sus características *group* y *size* se puede observar en el Cuadro A.1.

Group	Character ID	Size
1	"Baron Zemo"	2
1	"N'astirh"	3
0	"Silver Sable"	1
0	"Hope Summers (comics)"	1
0	"Magik (Illyana Rasputin)"	1
0	"Night Thrasher (Dwayne Taylor)"	2
1	"Orb (comics)"	1
0	"Forge (comics)"	1
0	"U.S. Agent (character)"	1
1	"Venom (Marvel Comics character)"	6
0	"Hercules (Marvel Comics)"	1
1	"Selene (comics)"	5
1	"Hood (comics)"	2
2	"Sabretooth (comics)"	1
2	"Beetle (comics)"	1
0	"Siryn"	2
1	"Malekith the Accursed"	3
1	"Mister Negative"	1
0	"Jessica Jones"	2
0	"Maria Hill"	3
0	"Master Izo"	1
0	"Valkyrie (Marvel Comics)"	1
0	"Nova (Sam Alexander)"	1
1	"Rampage (Marvel Comics)"	1
0	"Cable (comics)"	4
1	"Cyclone (Marvel Comics)"	1
1	"Mercurio the 4-D Man"	1
1	"Mojo (comics)"	1
1	"Trevor Fitzroy"	1
1	"Rev (comics)"	1
1	"Matsu'o Tsurayaba"	1
0	"Elsie-Dee"	1
0	"Motormouth (comics)"	1
1	"Mephisto (comics)"	1
1	"Fury (Marvel Comics)"	1
0	"Andromeda (Marvel Comics)"	1
1	"Mad Jim Jaspers"	1
2	"Magneto (comics)"	2
0	"Wasp (comics)"	1
1	"Vermin (comics)"	1
1	"Spidercide"	1

Sigue en la página siguiente.

Group	Character ID	Size
2	"Crimson Dynamo"	2
2	"Nomad (comics)"	3
1	"Kro (comics)"	2
1	"Walrus (comics)"	1
1	"Spymaster (comics)"	1
0	"Hawkeye (Kate Bishop)"	1
1	"Headlok"	1
0	"Gamora"	1
0	"War Machine"	2
1	"Titania (Marvel Comics)"	2
0	"Bob, Agent of Hydra"	1
1	"T-Ray (comics)"	1
0	"Nick Fury"	2
0	"Aragorn (comics)"	2
1	"Mandarin (comics)"	1
1	"Cobra (Marvel Comics)"	1
1	"Kraven the Hunter"	2
0	"Daredevil (Marvel Comics character)"	3
1	"Llyra"	2
0	"Crane Mother"	1
0	"Anya Corazon"	3
0	"Silhouette (comics)"	1
1	"Ulik"	1
0	"Iron Man (Ultimate Marvel character)"	1
2	"Abner Jenkins"	3
2	"Fixer (comics)"	2
0	"Bucky Barnes"	2
2	"Red Raven (Marvel Comics)"	1
2	"Tarantula (Marvel Comics)"	1
1	"Enchantress (Marvel Comics)"	5
1	"Venomm"	1
0	"Spider-Man (Miles Morales)"	4
0	"Thing (comics)"	1
1	"Tombstone (comics)"	2
1	"Blackout (Marcus Daniels)"	1
0	"Toro (comics)"	1
0	"Iron Man"	5
2	"Torpedo (Marvel Comics)"	1
1	"Answer (comics)"	1
2	"Erik Josten"	2
1	"Mastermind (Martinique Jason)"	1
1	"Kurse"	2
1	"White Rabbit (comics)"	2
1	"Lemuel Dorcas"	4
1	"Justin Hammer"	5
1	"Satannish"	1
1	"Mendel Stromm"	1
0	"Adam Warlock"	2
2	"Fin Fang Foom"	1
1	"Lorelei (Asgardian)"	2
0	"Shiver Man"	3
0	"Namorita"	3
0	"Shriker"	2
0	"Riri Williams"	1
1	"Gilded Lily (character)"	1
1	"Carrion (comics)"	5
0	"Ms. Marvel (Kamala Khan)"	1
1	"Space Phantom"	1
0	"Gambit (comics)"	1
1	"Glob (comics)"	1
1	"Gladiator (Melvin Potter)"	2
1	"Exodus (comics)"	1
1	"Super Sabre (comics)"	2
1	"S'ym"	2

Sigue en la página siguiente.

Group	Character ID	Size
1	"Matador (Marvel Comics)"	1
1	"Tiger Shark (Marvel Comics)"	2
1	"Mad Thinker"	5
2	"Alpha the Ultimate Mutant"	1
0	"Mariko Yashida"	1
1	"Demogoblin"	5
1	"Crimson Commando"	1
1	"Scarecrow (Marvel Comics)"	6
0	"Redwing (Marvel Comics)"	1
0	"Hiro Takachiho"	1
1	"Lilith (Marvel Comics)"	1
2	"Futurist (comics)"	1
1	"Baron Strucker"	2
1	"Presence (Marvel Comics)"	1
2	"Swordsman (comics)"	1
1	"Blackout (Lilin)"	1
0	"Falcon (comics)"	3
1	"Krang (Marvel Comics)"	5
1	"Mandrill (comics)"	2
2	"Elektro (comics)"	1
1	"Black Tom Cassidy"	3
1	"Baron Mordo"	2
0	"Nate Grey"	2
1	"Annihilus"	1
1	"Jamie Braddock"	2
1	"Diablo (Marvel Comics)"	2
2	"Molten Man"	1
0	"Shard (comics)"	1
1	"Abomination (comics)"	3
1	"Ghaur"	3
1	"Danger Room"	3
2	"Copycat (Marvel Comics)"	3
0	"Killpower"	1
1	"Mangog"	1
0	"Hulk (comics)"	3
1	"Ymir (Marvel Comics)"	1
1	"Threnody (comics)"	1
1	"Thanos"	2
1	"Midnight's Fire"	1
1	"Chimera (Marvel Comics)"	2
1	"Spot (comics)"	1
1	"Albert (comics)"	1
2	"Prowler (comics)"	1
0	"Psylocke"	1
0	"Black Bolt"	1
2	"Harry Osborn"	1
0	"Battlestar (comics)"	1
1	"Stonewall (comics)"	1
1	"Tyrak"	1
0	"Shuri (comics)"	1
1	"Fenris (comics)"	1
0	"Garrison Kane"	1
0	"Jennifer Kale"	1
1	"Mister Hyde (comics)"	3
0	"Starlight (Marvel Comics)"	1
0	"Mockingbird (Marvel Comics)"	2
1	"Man-Ape"	1
0	"Iron Fist (comics)"	7
1	"Crossbones (comics)"	2
1	"Egghead (Marvel Comics)"	2
0	"Demolition Man (comics)"	2
1	"Kaecilius"	2
0	"Deadpool"	5
1	"Sin (Marvel Comics)"	2

Sigue en la página siguiente.

Group	Character ID	Size
1	"Slayback"	1
0	"Wonder Man"	1
1	"Helmut Zemo"	1
0	"Rachel Summers"	1
1	"Eel (comics)"	2
1	"Mentallo"	1
1	"Wizard (Marvel Comics)"	2
0	"Beast (comics)"	2
1	"Nekra"	2
1	"Hela (comics)"	3
2	"Juggernaut (comics)"	4
1	"Umar (Marvel Comics)"	1
1	"Doctor Doom"	4
0	"Namor"	5
1	"Eli Bard"	1
1	"Ord (comics)"	1
0	"Hawkeye (comics)"	4
2	"Wraith (Marvel Comics)"	2
0	"Inferno (Dante Pertuz)"	4
1	"Senyaka"	1
2	"Puma (comics)"	2
2	"Volcana (Marvel Comics)"	1
0	"Black Cat (Marvel Comics)"	3
1	"Fenris Wolf (Marvel Comics)"	1
0	"Spider-Man"	12
1	"Pasco (comics)"	1
1	"Ringer (comics)"	2
1	"Power Broker"	1
0	"Misty Knight"	3
2	"Gorgon (Inhuman)"	1
1	"Doctor Octopus"	1
0	"Luke Cage"	5
2	"Wither (comics)"	1
1	"Ravonna"	1
1	"Leila Davis"	4
1	"Executioner (comics)"	1
2	"Silver Samurai"	3
1	"Rhino (comics)"	2
0	"Baymax"	1
2	"Will o' the Wisp (comics)"	3
1	"Black Fox (Raul Chalmers)"	1
1	"Dormammu"	5
0	"Cloak and Dagger (comics)"	1
2	"Rogue (comics)"	1
0	"Darkstar (comics)"	3
1	"Dirtnap (comics)"	1
0	"Human Torch (android)"	1
1	"Mole Man"	5
0	"Thor (Marvel Comics)"	3
1	"Surtur (Marvel Comics)"	3
1	"Unicorn (comics)"	8
0	"Black Widow (Natasha Romanova)"	4
2	"Mondo (comics)"	2
2	"Janice Lincoln"	1
1	"Leader (comics)"	4
1	"Sat-Yr-9"	1
1	"Collector (comics)"	1
0	"Doctor Strange"	1
0	"Ghost Rider"	2
1	"Cletus Kasady"	4
0	"Black Knight (Dane Whitman)"	1
0	"Colleen Wing"	3
1	"Gibbon (comics)"	1
0	"Devil Dinosaur"	2

Sigue en la página siguiente.

Group	Character ID	Size
1	"Shriek (comics)"	5
1	"Discus (comics)"	2
0	"Black Panther (comics)"	2
0	"Spider-Girl"	1
1	"Doctor Faustus (comics)"	1
1	"Attuma"	8
1	"Erik Killmonger"	1
0	"Storm (Marvel Comics)"	2
1	"Mac Gargan"	1
1	"Arcade (Marvel Comics)"	2
1	"Vulture (Marvel Comics)"	1
0	"Blink (comics)"	1
1	"Kala (comics)"	1
0	"Jocasta (comics)"	2
1	"Electro (Marvel Comics)"	1
1	"Chameleon (comics)"	6
0	"Spider-Woman (Jessica Drew)"	3
0	"Doc Samson"	1
1	"Grizzly (comics)"	2
1	"Deathwatch (comics)"	1
0	"She-Hulk"	1
2	"Vengeance (comics)"	2
0	"Bucky (Marvel Comics)"	1
0	"Topaz (Marvel Comics)"	1
1	"Molecule Man"	1
1	"Porcupine (comics)"	2
1	"Kangaroo (comics)"	4
1	"Madame Masque"	1
1	"Abraham Cornelius"	2
0	"Stingray (comics)"	1
0	"Free Spirit (comics)"	1
1	"Psycho-Man"	1
1	"Jason Macendale"	1
1	"Klaw (Marvel Comics)"	1
1	"Kingpin (character)"	2
0	"Amadeus Cho"	1
1	"Red Ghost"	4
0	"Hank Pym"	2
1	"Titanium Man"	2
1	"Madelyne Pryor"	2
1	"Absorbing Man"	1
1	"Nicholas Scratch"	1
1	"Grim Reaper (comics)"	6
1	"Boomerang (comics)"	3
2	"Pip the Troll"	2
1	"Black Talon (comics)"	1
0	"Medusa (comics)"	1
0	"Rom (comics)"	1
1	"Blastaar"	1
0	"Kid Kaiju"	2
0	"Jack Flag"	1
0	"Elsa Bloodstone"	1
0	"Power Man (Victor Alvarez)"	2
0	"Abigail Brand"	1
1	"Awesome Android"	1
0	"Spider-Woman (Gwen Stacy)"	3
0	"Domino (comics)"	3
0	"Wolverine (character)"	2
1	"Lois London"	1
1	"Zarathos"	2
1	"Madman (Marvel Comics)"	1
1	"Cyttorak"	1
1	"Carnage (comics)"	5
1	"Steel Serpent"	1

Sigue en la página siguiente.

Group	Character ID	Size
2	"Eddie Brock"	2
1	"Man-Beast"	2
0	"Moon Girl (Marvel Comics)"	2
1	"Midgard Serpent (Marvel Comics)"	1
0	"Captain America"	10
2	"Spiral (comics)"	1
1	"Calypso (comics)"	1
0	"Man-Thing"	1
1	"Red Skull"	9
1	"Apocalypse (Marvel Comics)"	1
0	"Silk (comics)"	5
1	"Stiletto (comics)"	2
0	"Mayhem (comics)"	1
1	"Loki (comics)"	6
1	"Jigsaw (Marvel Comics)"	2
1	"Orka (comics)"	1
1	"Batroc the Leaper"	1
1	"Pluto (Marvel Comics)"	1
1	"Mister Sinister"	1
0	"Sif (comics)"	1
1	"Griffin (Marvel Comics)"	2
1	"Garokk"	1
0	"Pepper Potts"	1
0	"Moon-Boy"	1
0	"Caretaker (comics)"	1
0	"Buzz (comics)"	1
1	"Arnim Zola"	1
1	"Jonas Harrow"	2
0	"Rikki Barnes"	1
0	"White Tiger (Angela del Toro)"	1
1	"Hammerhead (comics)"	3
1	"Viper (Madame Hydra)"	3
1	"Puppet Master (Marvel Comics)"	5
1	"Man-Bull"	1
1	"Ultron"	2
1	"Byrrah"	3
1	"Kang the Conqueror"	1
0	"Carol Danvers"	2
1	"Jackal (Marvel Comics)"	3
1	"Galaxy Master"	1
1	"Doppelganger (comics)"	5
1	"Plantman"	2
1	"Shellshock (Marvel Comics)"	1
1	"Count Nefaria"	2
1	"Emplate"	1
0	"Peregrine (comics)"	1
1	"Belasco (Marvel Comics)"	3
1	"Dragon Man"	1
1	"Bi-Beast"	2
0	"Namora"	3
1	"Alexander Bont"	2
1	"Giganto"	2
0	"Fin (comics)"	3

Cuadro A.1: Tabla de nodos con sus propiedades

La tabla que almacena información sobre las aristas, es decir, su origen y su destino, también obtenida en Neo4j, se puede observar en el Cuadro A.2.

Origin	Destiny
Baron Zemo	Blackout (Marcus Daniels)
Baron Zemo	Vermin (comics)

Sigue en la página siguiente.

Origin	Destiny
N'astirh	S'ym
N'astirh	Belasco (Marvel Comics)
N'astirh	Madelyne Pryor
Silver Sable	Peregrine (comics)
Hope Summers (comics)	Cable (comics)
Magik (Illyana Rasputin)	Belasco (Marvel Comics)
Night Thrasher (Dwayne Taylor)	Midnight's Fire
Night Thrasher (Dwayne Taylor)	Silhouette (comics)
Orb (comics)	Justin Hammer
Forge (comics)	Storm (Marvel Comics)
U.S. Agent (character)	Battlestar (comics)
Venom (Marvel Comics character)	Carnage (comics)
Venom (Marvel Comics character)	Doctor Doom
Venom (Marvel Comics character)	Spider-Man
Venom (Marvel Comics character)	Magneto (comics)
Venom (Marvel Comics character)	Red Skull
Venom (Marvel Comics character)	Eddie Brock
Hercules (Marvel Comics)	Amadeus Cho
Selene (comics)	Senyaka
Selene (comics)	Eli Bard
Selene (comics)	Wither (comics)
Selene (comics)	Lois London
Selene (comics)	Blink (comics)
Hood (comics)	Cyclone (Marvel Comics)
Hood (comics)	Madame Masque
Sabretooth (comics)	Pasco (comics)
Beetle (comics)	Ringer (comics)
Siryn	Black Tom Cassidy
Siryn	Juggernaut (comics)
Malekith the Accursed	Surtur (Marvel Comics)
Malekith the Accursed	Kurse
Malekith the Accursed	Loki (comics)
Mister Negative	Hammerhead (comics)
Jessica Jones	Carol Danvers
Jessica Jones	Luke Cage
Maria Hill	Iron Man
Maria Hill	Nick Fury
Maria Hill	Captain America
Master Izo	Iron Fist (comics)
Valkyrie (Marvel Comics)	Aragorn (comics)
Nova (Sam Alexander)	Spider-Man (Miles Morales)
Rampage (Marvel Comics)	Jigsaw (Marvel Comics)
Cable (comics)	Domino (comics)
Cable (comics)	Rachel Summers
Cable (comics)	Deadpool
Mercurio the 4-D Man	Thor (Marvel Comics)
Mojo (comics)	Spiral (comics)
Trevor Fitzroy	Shard (comics)
Rev (comics)	Jigsaw (Marvel Comics)
Matsu'o Tsurayaba	Abraham Cornelius
Elsie-Dee	Albert (comics)
Motormouth (comics)	Killpower
Mephisto (comics)	Zarathos
Fury (Marvel Comics)	Mad Jim Jaspers
Andromeda (Marvel Comics)	Namor
Magneto (comics)	Garokk
Wasp (comics)	Hank Pym
Spidercide	Jackal (Marvel Comics)
Crimson Dynamo	Answer (comics)
Crimson Dynamo	Darkstar (comics)
Nomad (comics)	Demolition Man (comics)
Nomad (comics)	Falcon (comics)
Nomad (comics)	Captain America
Kro (comics)	Ghaur

Sigue en la página siguiente.

Origin	Destiny
Kro (comics)	Giganto
Walrus (comics)	White Rabbit (comics)
Spymaster (comics)	Justin Hammer
Hawkeye (Kate Bishop)	Hawkeye (comics)
Headlok	Griffin (Marvel Comics)
Gamora	Adam Warlock
War Machine	Iron Man
War Machine	Iron Man (Ultimate Marvel character)
Titania (Marvel Comics)	Doctor Doom
Titania (Marvel Comics)	Absorbing Man
Bob, Agent of Hydra	Deadpool
T-Ray (comics)	Slayback
Nick Fury	Shiver Man
Aragorn (comics)	Black Knight (Dane Whitman)
Mandarin (comics)	Unicorn (comics)
Cobra (Marvel Comics)	Mister Hyde (comics)
Kraven the Hunter	Calypso (comics)
Kraven the Hunter	Chameleon (comics)
Daredevil (Marvel Comics character)	Black Widow (Natasha Romanova)
Daredevil (Marvel Comics character)	Kingpin (character)
Daredevil (Marvel Comics character)	Iron Fist (comics)
Llyra	Attuma
Llyra	Ghaur
Crane Mother	Steel Serpent
Anya Corazon	Spider-Man
Anya Corazon	Spider-Woman (Jessica Drew)
Anya Corazon	Carol Danvers
Ulik	Loki (comics)
Abner Jenkins	Gladiator (Melvin Potter)
Abner Jenkins	Justin Hammer
Abner Jenkins	Collector (comics)
Fixer (comics)	Mentallo
Fixer (comics)	Alexander Bont
Bucky Barnes	Black Widow (Natasha Romanova)
Bucky Barnes	Captain America
Red Raven (Marvel Comics)	Bi-Beast
Tarantula (Marvel Comics)	Jackal (Marvel Comics)
Enchantress (Marvel Comics)	Wonder Man
Enchantress (Marvel Comics)	Executioner (comics)
Enchantress (Marvel Comics)	Doctor Doom
Enchantress (Marvel Comics)	Erik Josten
Enchantress (Marvel Comics)	Loki (comics)
Venom	Erik Killmonger
Spider-Man (Miles Morales)	Spider-Man
Spider-Man (Miles Morales)	Spider-Woman (Gwen Stacy)
Spider-Man (Miles Morales)	Silk (comics)
Thing (comics)	Mole Man
Tombstone (comics)	Hammerhead (comics)
Tombstone (comics)	Janice Lincoln
Toro (comics)	Human Torch (android)
Iron Man	Wraith (Marvel Comics)
Iron Man	Pepper Potts
Iron Man	Riri Williams
Torpedo (Marvel Comics)	Rom (comics)
Erik Josten	Grim Reaper (comics)
Mastermind (Martinique Jason)	Arcade (Marvel Comics)
Kurse	Thor (Marvel Comics)
White Rabbit (comics)	Arcade (Marvel Comics)
Lemuel Dorcas	Attuma
Lemuel Dorcas	Byrrah
Lemuel Dorcas	Krang (Marvel Comics)
Lemuel Dorcas	Tiger Shark (Marvel Comics)
Justin Hammer	Stiletto (comics)
Justin Hammer	Discus (comics)

Sigue en la página siguiente.

Origin	Destiny
Satannish	Dormammu
Mendel Stromm	Jason Macendale
Adam Warlock	Pip the Troll
Fin Fang Foom	Elektro (comics)
Lorelei (Asgardian)	Pluto (Marvel Comics)
Lorelei (Asgardian)	Loki (comics)
Shiver Man	Beast (comics)
Shiver Man	Wolverine (character)
Namorita	Fin (comics)
Namorita	Namor
Namorita	Namora
Shriker	Caretaker (comics)
Shriker	Ghost Rider
Gilded Lily (character)	Diablo (Marvel Comics)
Carrion (comics)	Carnage (comics)
Carrion (comics)	Demogoblin
Carrion (comics)	Shriek (comics)
Carrion (comics)	Cletus Kasady
Carrion (comics)	Doppelganger (comics)
Ms. Marvel (Kamala Khan)	Inferno (Dante Pertuz)
Space Phantom	Grim Reaper (comics)
Gambit (comics)	Danger Room
Glob (comics)	Leader (comics)
Gladiator (Melvin Potter)	Alexander Bont
Exodus (comics)	Man-Beast
Super Sabre (comics)	Stonewall (comics)
Super Sabre (comics)	Crimson Commando
S'ym	Belasco (Marvel Comics)
Matador (Marvel Comics)	Man-Bull
Tiger Shark (Marvel Comics)	Attuma
Mad Thinker	Egghead (Marvel Comics)
Mad Thinker	Puppet Master (Marvel Comics)
Mad Thinker	Awesome Android
Mad Thinker	Wizard (Marvel Comics)
Mad Thinker	Klaw (Marvel Comics)
Alpha the Ultimate Mutant	Futurist (comics)
Mariko Yashida	Wolverine (character)
Demogoblin	Carnage (comics)
Demogoblin	Doppelganger (comics)
Demogoblin	Cletus Kasady
Demogoblin	Shriek (comics)
Scarecrow (Marvel Comics)	Eel (comics)
Scarecrow (Marvel Comics)	Porcupine (comics)
Scarecrow (Marvel Comics)	Count Nefaria
Scarecrow (Marvel Comics)	Will o' the Wisp (comics)
Scarecrow (Marvel Comics)	Unicorn (comics)
Scarecrow (Marvel Comics)	Plantman
Redwing (Marvel Comics)	Falcon (comics)
Hiro Takachiho	Baymax
Lilith (Marvel Comics)	Zarathos
Baron Strucker	Helmut Zemo
Baron Strucker	Red Skull
Presence (Marvel Comics)	Starlight (Marvel Comics)
Swordsman (comics)	Hawkeye (comics)
Blackout (Lilin)	Deathwatch (comics)
Falcon (comics)	Captain America
Krang (Marvel Comics)	Attuma
Krang (Marvel Comics)	Orka (comics)
Krang (Marvel Comics)	Puppet Master (Marvel Comics)
Krang (Marvel Comics)	Byrrah
Mandrill (comics)	Silver Samurai
Mandrill (comics)	Nekra
Black Tom Cassidy	Mondo (comics)
Black Tom Cassidy	Juggernaut (comics)

Sigue en la página siguiente.

Origin	Destiny
Baron Mordo	Dormammu
Baron Mordo	Kaecilius
Nate Grey	Madelyne Pryor
Nate Grey	Threnody (comics)
Annihilus	Blastaar
Jamie Braddock	Psylocke
Jamie Braddock	Sat-Yr-9
Diablo (Marvel Comics)	Dragon Man
Molten Man	Will o' the Wisp (comics)
Abomination (comics)	Leader (comics)
Abomination (comics)	Rhino (comics)
Abomination (comics)	Galaxy Master
Ghaur	Attuma
Danger Room	Rogue (comics)
Danger Room	Ord (comics)
Copycat (Marvel Comics)	Domino (comics)
Copycat (Marvel Comics)	Deadpool
Copycat (Marvel Comics)	Garrison Kane
Mangog	Thanos
Hulk (comics)	Mole Man
Hulk (comics)	She-Hulk
Hulk (comics)	Doc Samson
Ymir (Marvel Comics)	Surtur (Marvel Comics)
Thanos	Pip the Troll
Chimera (Marvel Comics)	Emplate
Chimera (Marvel Comics)	Dirtnap (comics)
Spot (comics)	Kangaroo (comics)
Prowler (comics)	Spider-Man
Black Bolt	Inferno (Dante Pertuz)
Harry Osborn	Spider-Man
Tyrak	Attuma
Shuri (comics)	Black Panther (comics)
Fenris (comics)	Abraham Cornelius
Jennifer Kale	Man-Thing
Mister Hyde (comics)	Batroc the Leaper
Mister Hyde (comics)	Mac Gargan
Mockingbird (Marvel Comics)	Spider-Man
Mockingbird (Marvel Comics)	Hawkeye (comics)
Man-Ape	Grim Reaper (comics)
Iron Fist (comics)	Power Man (Victor Alvarez)
Iron Fist (comics)	Luke Cage
Iron Fist (comics)	Misty Knight
Iron Fist (comics)	Colleen Wing
Iron Fist (comics)	White Tiger (Angela del Toro)
Crossbones (comics)	Red Skull
Crossbones (comics)	Sin (Marvel Comics)
Egghead (Marvel Comics)	Puppet Master (Marvel Comics)
Demolition Man (comics)	Captain America
Kaecilius	Dormammu
Deadpool	Domino (comics)
Deadpool	Spider-Man
Sin (Marvel Comics)	Red Skull
Eel (comics)	Unicorn (comics)
Wizard (Marvel Comics)	Puppet Master (Marvel Comics)
Beast (comics)	Abigail Brand
Nekra	Grim Reaper (comics)
Hela (comics)	Midgard Serpent (Marvel Comics)
Hela (comics)	Fenris Wolf (Marvel Comics)
Hela (comics)	Loki (comics)
Juggernaut (comics)	Mondo (comics)
Juggernaut (comics)	Cyttorak
Umar (Marvel Comics)	Dormammu
Doctor Doom	Puppet Master (Marvel Comics)
Namor	Fin (comics)

Sigue en la página siguiente.

Origin	Destiny
Namor	Namora
Namor	Stingray (comics)
Hawkeye (comics)	Black Widow (Natasha Romanova)
Wraith (Marvel Comics)	Spider-Man
Inferno (Dante Pertuz)	Gorgon (Inhuman)
Inferno (Dante Pertuz)	Medusa (comics)
Puma (comics)	Spider-Man
Puma (comics)	Black Cat (Marvel Comics)
Volcana (Marvel Comics)	Molecule Man
Black Cat (Marvel Comics)	Spider-Man
Black Cat (Marvel Comics)	Silk (comics)
Spider-Man	Kingpin (character)
Spider-Man	Silk (comics)
Ringer (comics)	Leila Davis
Power Broker	Red Skull
Misty Knight	Colleen Wing
Misty Knight	Luke Cage
Doctor Octopus	Chameleon (comics)
Luke Cage	Power Man (Victor Alvarez)
Luke Cage	Colleen Wing
Ravonna	Kang the Conqueror
Leila Davis	Boomerang (comics)
Leila Davis	Vulture (Marvel Comics)
Leila Davis	Rhino (comics)
Silver Samurai	Boomerang (comics)
Silver Samurai	Viper (Madame Hydra)
Will o' the Wisp (comics)	Jonas Harrow
Black Fox (Raul Chalmers)	Red Ghost
Dormammu	Nicholas Scratch
Cloak and Dagger (comics)	Mayhem (comics)
Darkstar (comics)	Griffin (Marvel Comics)
Darkstar (comics)	Titanium Man
Mole Man	Kala (comics)
Mole Man	Red Ghost
Mole Man	Giganto
Thor (Marvel Comics)	Sif (comics)
Surtur (Marvel Comics)	Loki (comics)
Unicorn (comics)	Red Ghost
Unicorn (comics)	Titanium Man
Unicorn (comics)	Porcupine (comics)
Unicorn (comics)	Count Nefaria
Unicorn (comics)	Plantman
Black Widow (Natasha Romanova)	Captain America
Leader (comics)	Madman (Marvel Comics)
Leader (comics)	Chameleon (comics)
Doctor Strange	Topaz (Marvel Comics)
Ghost Rider	Vengeance (comics)
Cletus Kasady	Doppelganger (comics)
Cletus Kasady	Shriek (comics)
Gibbon (comics)	Kangaroo (comics)
Devil Dinosaur	Moon Girl (Marvel Comics)
Devil Dinosaur	Moon-Boy
Shriek (comics)	Carnage (comics)
Shriek (comics)	Doppelganger (comics)
Discus (comics)	Stiletto (comics)
Black Panther (comics)	Storm (Marvel Comics)
Spider-Girl	Buzz (comics)
Doctor Faustus (comics)	Red Skull
Attuma	Red Ghost
Attuma	Byrrah
Jocasta (comics)	Ultron
Jocasta (comics)	Hank Pym
Electro (Marvel Comics)	Chameleon (comics)
Chameleon (comics)	Hammerhead (comics)

Sigue en la página siguiente.

Origin	Destiny
Chameleon (comics)	Red Skull
Spider-Woman (Jessica Drew)	Silk (comics)
Spider-Woman (Jessica Drew)	Spider-Woman (Gwen Stacy)
Grizzly (comics)	Kangaroo (comics)
Grizzly (comics)	Jackal (Marvel Comics)
Vengeance (comics)	Eddie Brock
Bucky (Marvel Comics)	Captain America
Kangaroo (comics)	Jonas Harrow
Free Spirit (comics)	Captain America
Psycho-Man	Shellshock (Marvel Comics)
Grim Reaper (comics)	Ultron
Grim Reaper (comics)	Black Talon (comics)
Boomerang (comics)	Viper (Madame Hydra)
Kid Kaiju	Moon Girl (Marvel Comics)
Kid Kaiju	Elsa Bloodstone
Jack Flag	Captain America
Spider-Woman (Gwen Stacy)	Silk (comics)
Carnage (comics)	Doppelganger (comics)
Man-Beast	Bi-Beast
Captain America	Rikki Barnes
Red Skull	Viper (Madame Hydra)
Red Skull	Arnim Zola
Apocalypse (Marvel Comics)	Mister Sinister
Namora	Fin (comics)

Cuadro A.2: Tabla de aristas

Por último, podemos observar la lista de todas las comunidades que hay dentro de nuestro grafo, la cual hemos obtenido con Python:

```
({'Baron Zemo', 'Blackout (Marcus Daniels)', 'Vermin (comics)'}
{'Belasco (Marvel Comics)', 'Madelyne Pryor', 'Magik (Illyana Rasputin)', "N'astirh",
'Nate Grey', "S'ym", 'Threnody (comics)'}
{'Peregrine (comics)', 'Silver Sable'}
{'Abigail Brand', 'Abomination (comics)', 'Anya Corazon', 'Arnim Zola', 'Baron Strucker',
'Beast (comics)', 'Beetle (comics)', 'Black Cat (Marvel Comics)', 'Black Widow (Natasha
Romanova)', 'Bob, Agent of Hydra', 'Boomerang (comics)', 'Bucky (Marvel Comics)', 'Bucky
Barnes', 'Cable (comics)', 'Calypso (comics)', 'Captain America', 'Caretaker (comics)',
'Carnage (comics)', 'Carol Danvers', 'Carrion (comics)', 'Chameleon (comics)', 'Cletus
Kasady', 'Colleen Wing', 'Copycat (Marvel Comics)', 'Crossbones (comics)', 'Daredevil
(Marvel Comics character)', 'Deadpool', 'Demogoblin', 'Demolition Man (comics)', 'Doctor
Faustus (comics)', 'Doctor Octopus', 'Domino (comics)', 'Doppelganger (comics)', 'Eddie
Brock', 'Electro (Marvel Comics)', 'Falcon (comics)', 'Free Spirit (comics)', 'Galaxy
Master', 'Garokk', 'Garrison Kane', 'Ghost Rider', 'Glob (comics)', 'Hammerhead (comics)',
'Harry Osborn', 'Hawkeye (Kate Bishop)', 'Hawkeye (comics)', 'Helmut Zemo', 'Hope Summers
(comics)', 'Iron Fist (comics)', 'Iron Man', 'Iron Man (Ultimate Marvel character)', 'Jack
Flag', 'Janice Lincoln', 'Jessica Jones', 'Kingpin (character)', 'Kraven the Hunter',
'Leader (comics)', 'Leila Davis', 'Luke Cage', 'Madman (Marvel Comics)', 'Magneto
(comics)', 'Maria Hill', 'Mariko Yashida', 'Master Izo', 'Mister Negative', 'Misty
Knight', 'Mockingbird (Marvel Comics)', 'Nick Fury', 'Nomad (comics)', 'Nova (Sam
Alexander)', 'Pepper Potts', 'Power Broker', 'Power Man (Victor Alvarez)', 'Prowler
(comics)', 'Puma (comics)', 'Rachel Summers', 'Red Skull', 'Redwing (Marvel Comics)',
'Rhino (comics)', 'Rikki Barnes', 'Ringer (comics)', 'Riri Williams', 'Shiver Man',
'Shriek (comics)', 'Shriker', 'Silk (comics)', 'Silver Samurai', 'Sin (Marvel Comics)',
'Spider-Man', 'Spider-Man (Miles Morales)', 'Spider-Woman (Gwen Stacy)', 'Spider-Woman
(Jessica Drew)', 'Swordsman (comics)', 'Tombstone (comics)', 'Vengeance (comics)', 'Venom
(Marvel Comics character)', 'Viper (Madame Hydra)', 'Vulture (Marvel Comics)', 'War
Machine', 'White Tiger (Angela del Toro)', 'Wolverine (character)', 'Wraith (Marvel
Comics)'}
{"Midnight's Fire", 'Night Thrasher (Dwayne Taylor)', 'Silhouette (comics)'}
{'Abner Jenkins', 'Alexander Bont', 'Collector (comics)', 'Discus (comics)', 'Fixer
(comics)', 'Gladiator (Melvin Potter)', 'Justin Hammer', 'Mentallo', 'Orb (comics)',
'Spymaster (comics)', 'Stiletto (comics)'}
```

{'Black Panther (comics)', 'Forge (comics)', 'Shuri (comics)', 'Storm (Marvel Comics)'}
{'Battlestar (comics)', 'U.S. Agent (character)'}
{'Absorbing Man', 'Answer (comics)', 'Attuma', 'Awesome Android', 'Black Fox (Raul Chalmers)', 'Black Talon (comics)', 'Byrrah', 'Count Nefaria', 'Crimson Dynamo', 'Darkstar (comics)', 'Doc Samson', 'Doctor Doom', 'Eel (comics)', 'Egghead (Marvel Comics)', 'Enchantress (Marvel Comics)', 'Erik Josten', 'Executioner (comics)', 'Fenris Wolf (Marvel Comics)', 'Ghaur', 'Gibbon (comics)', 'Giganto', 'Griffin (Marvel Comics)', 'Grim Reaper (comics)', 'Grizzly (comics)', 'Hank Pym', 'Headlok', 'Hela (comics)', 'Hulk (comics)', 'Jackal (Marvel Comics)', 'Jocasta (comics)', 'Jonas Harrow', 'Kala (comics)', 'Kangaroo (comics)', 'Klaw (Marvel Comics)', 'Krang (Marvel Comics)', 'Kro (comics)', 'Kurse', 'Lemuel Dorcas', 'Llyra', 'Loki (comics)', 'Lorelei (Asgardian)', 'Mad Thinker', 'Malekith the Accursed', 'Man-Ape', 'Mandarin (comics)', 'Mandrill (comics)', 'Mercurio the 4-D Man', 'Midgard Serpent (Marvel Comics)', 'Mole Man', 'Molten Man', 'Nekra', 'Orka (comics)', 'Plantman', 'Pluto (Marvel Comics)', 'Porcupine (comics)', 'Puppet Master (Marvel Comics)', 'Red Ghost', 'Scarecrow (Marvel Comics)', 'She-Hulk', 'Sif (comics)', 'Space Phantom', 'Spidercide', 'Spot (comics)', 'Surtur (Marvel Comics)', 'Tarantula (Marvel Comics)', 'Thing (comics)', 'Thor (Marvel Comics)', 'Tiger Shark (Marvel Comics)', 'Titania (Marvel Comics)', 'Titanium Man', 'Tyrak', 'Ulik', 'Ultron', 'Unicorn (comics)', 'Wasp (comics)', "Will o' the Wisp (comics)", 'Wizard (Marvel Comics)', 'Wonder Man', 'Ymir (Marvel Comics)'}
{'Amadeus Cho', 'Hercules (Marvel Comics)'}
{'Blink (comics)', 'Eli Bard', 'Lois London', 'Selene (comics)', 'Senyaka', 'Wither (comics)'}
{'Cyclone (Marvel Comics)', 'Hood (comics)', 'Madame Masque'}
{'Pasco (comics)', 'Sabretooth (comics)'}
{'Black Tom Cassidy', 'Cyttorak', 'Juggernaut (comics)', 'Mondo (comics)', 'Siryn'}
{'Aragorn (comics)', 'Black Knight (Dane Whitman)', 'Valkyrie (Marvel Comics)'}
{'Jigsaw (Marvel Comics)', 'Rampage (Marvel Comics)', 'Rev (comics)'}
{'Mojo (comics)', 'Spiral (comics)'}
{'Shard (comics)', 'Trevor Fitzroy'}
{'Abraham Cornelius', 'Fenris (comics)', "Matsu'o Tsurayaba"}
{'Albert (comics)', 'Elsie-Dee'}
{'Killpower', 'Motormouth (comics)'}
{'Lilith (Marvel Comics)', 'Mephisto (comics)', 'Zarathos'}
{'Fury (Marvel Comics)', 'Mad Jim Jaspers'}
{'Andromeda (Marvel Comics)', 'Fin (comics)', 'Namor', 'Namora', 'Namorita', 'Stingray (comics)'}
{'Arcade (Marvel Comics)', 'Mastermind (Martinique Jason)', 'Walrus (comics)', 'White Rabbit (comics)'}
{'Adam Warlock', 'Gamora', 'Mangog', 'Pip the Troll', 'Thanos'}
{'Slayback', 'T-Ray (comics)'}
{'Batroc the Leaper', 'Cobra (Marvel Comics)', 'Mac Gargan', 'Mister Hyde (comics)'}
{'Crane Mother', 'Steel Serpent'}
{'Bi-Beast', 'Exodus (comics)', 'Man-Beast', 'Red Raven (Marvel Comics)'}
{'Erik Killmonger', 'Venomm'}
{'Human Torch (android)', 'Toro (comics)'}
{'Rom (comics)', 'Torpedo (Marvel Comics)'}
{'Baron Mordo', 'Dormammu', 'Kaecilius', 'Nicholas Scratch', 'Satannish', 'Umar (Marvel Comics)'}
{'Jason Macendale', 'Mendel Stromm'}
{'Elektro (comics)', 'Fin Fang Foom'}
{'Diablo (Marvel Comics)', 'Dragon Man', 'Gilded Lily (character)'}
{'Black Bolt', 'Gorgon (Inhuman)', 'Inferno (Dante Pertuz)', 'Medusa (comics)', 'Ms. Marvel (Kamala Khan)'}
{'Danger Room', 'Gambit (comics)', 'Ord (comics)', 'Rogue (comics)'}
{'Crimson Commando', 'Stonewall (comics)', 'Super Sabre (comics)'}
{'Man-Bull', 'Matador (Marvel Comics)'}
{'Alpha the Ultimate Mutant', 'Futurist (comics)'}
{'Baymax', 'Hiro Takachiho'}
{'Presence (Marvel Comics)', 'Starlight (Marvel Comics)'}
{'Blackout (Lilin)', 'Deathwatch (comics)'}
{'Annihilus', 'Blastaar'}
{'Jamie Braddock', 'Psylocke', 'Sat-Yr-9'}
{'Chimera (Marvel Comics)', 'Dirtnap (comics)', 'Emplate'}
{'Jennifer Kale', 'Man-Thing'}
{'Molecule Man', 'Volcana (Marvel Comics)'}

```
{'Kang the Conqueror', 'Ravonna'}
{'Cloak and Dagger (comics)', 'Mayhem (comics)'}
{'Doctor Strange', 'Topaz (Marvel Comics)'}
{'Devil Dinosaur', 'Elsa Bloodstone', 'Kid Kaiju', 'Moon Girl (Marvel Comics)',
'Moon-Boy'}
{'Buzz (comics)', 'Spider-Girl'}
{'Psycho-Man', 'Shellshock (Marvel Comics)'}
{'Apocalypse (Marvel Comics)', 'Mister Sinister'})
```