

## Grafos

### Tabla de Contenidos

<b>Grafos</b>	<b>1</b>
1. Objetivos . . . . .	1
2. Introducción . . . . .	1
3. EJERCICIOS PLANTEADOS Y/O PROGRAMAS IMPLEMENTADOS . . . .	1
4. Conclusiones . . . . .	7
5 .Referencias bibliográficas . . . . .	7
6 .Declaración de uso de ia . . . . .	7

### 1. Objetivos

Crear un grafo de mínimo 20 nodos y 40 aristas para visualizarlo en Gephi. Aplicar un algoritmo de distribución, personalizar la apariencia, calcular el grado, detectar comuniddes y utilizando las métricas resultantes obtener dos o tres conclusiones del grafo.

### 2. Introducción

Utilizaremos el grafo de asoiaf traducido como “Canción de hielo y fuego” una novel de fantasía heroica que está a disposición en el aula virutal Utilizaremos la herramienta Gephi para realizar los ejercicios

### 3. EJERCICIOS PLANTEADOS Y/O PROGRAMAS IMPLEMENTADOS

Cree un grafo (o use un dataset de tipo grafo) que tenga al menos 20 nodos y 40 aristas. Visualice la data en Gephi, aplique un algoritmo de distribución, personalice la apariencia, calcule el grado (con o sin pesos, dependiendo del grafo), detecte comunidades y de las diferentes métricas obtenga dos o tres conclusiones a partir de su grafo.

Podemos resumir las tareas en: Visualce la data en Gephi Aplique un algoritmo de distribución Personalice la apariencia Calcule el grado Detecte comunidades Obtenga 2 o tres conclusiones

Al importar el grafo, que es un csv tenemos lo siguiente:

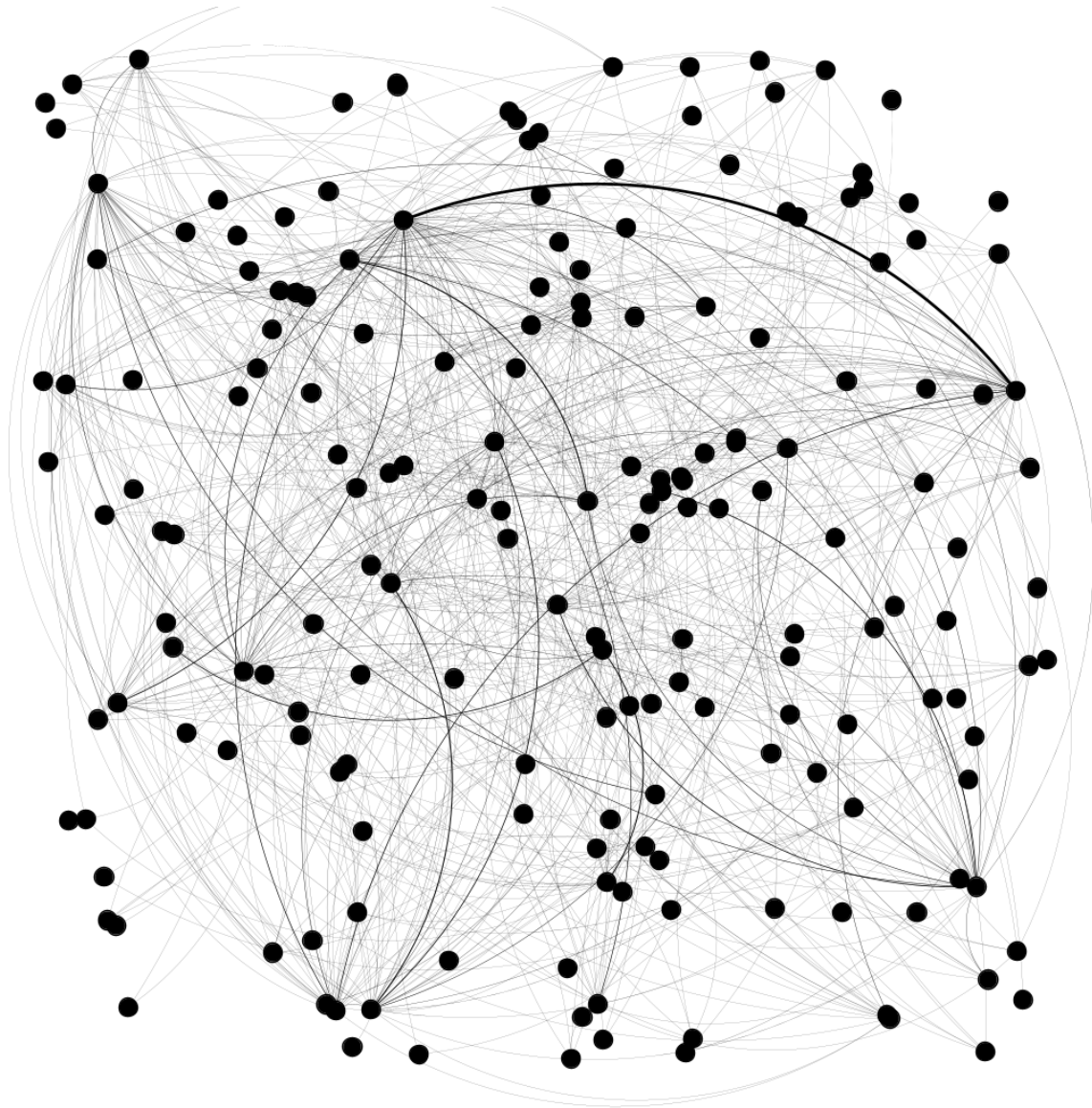


Figura 1: grafo distribuido, imagen\_muestra

Con esto completamos la parte de visualizar la data en Gephi

Ahora vamos a aplicar un algoritmo de distribución, en este caso usaremos el algoritmo de distribución de “Fruchterman Reingold”.

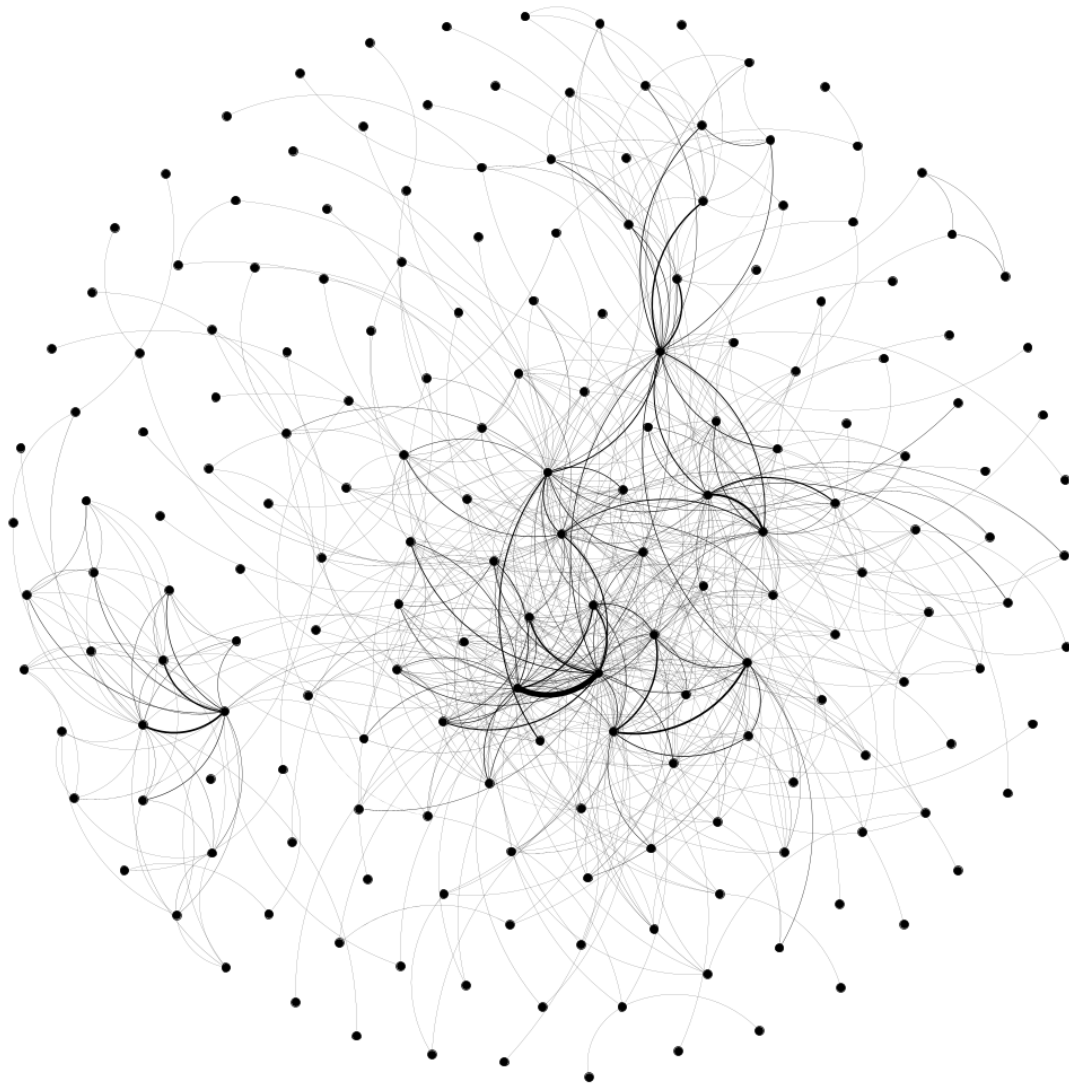


Figura 2: grafo distribuido, imagen\_muestra

Con esto completamos la tarea “aplicar un algoritmo de distribución”

Ahora vamos a personalizar la apariencia, para esto, primero vamos a mostrar los labels, los cuales serán los ids de los nodos ya agregados y luego cambiaremos el color de los nodos dependiendo de su peso. Bajo peso -> Negro algo de peso -> Amarillo medio peso -> Verde Alto peso -> Rojo

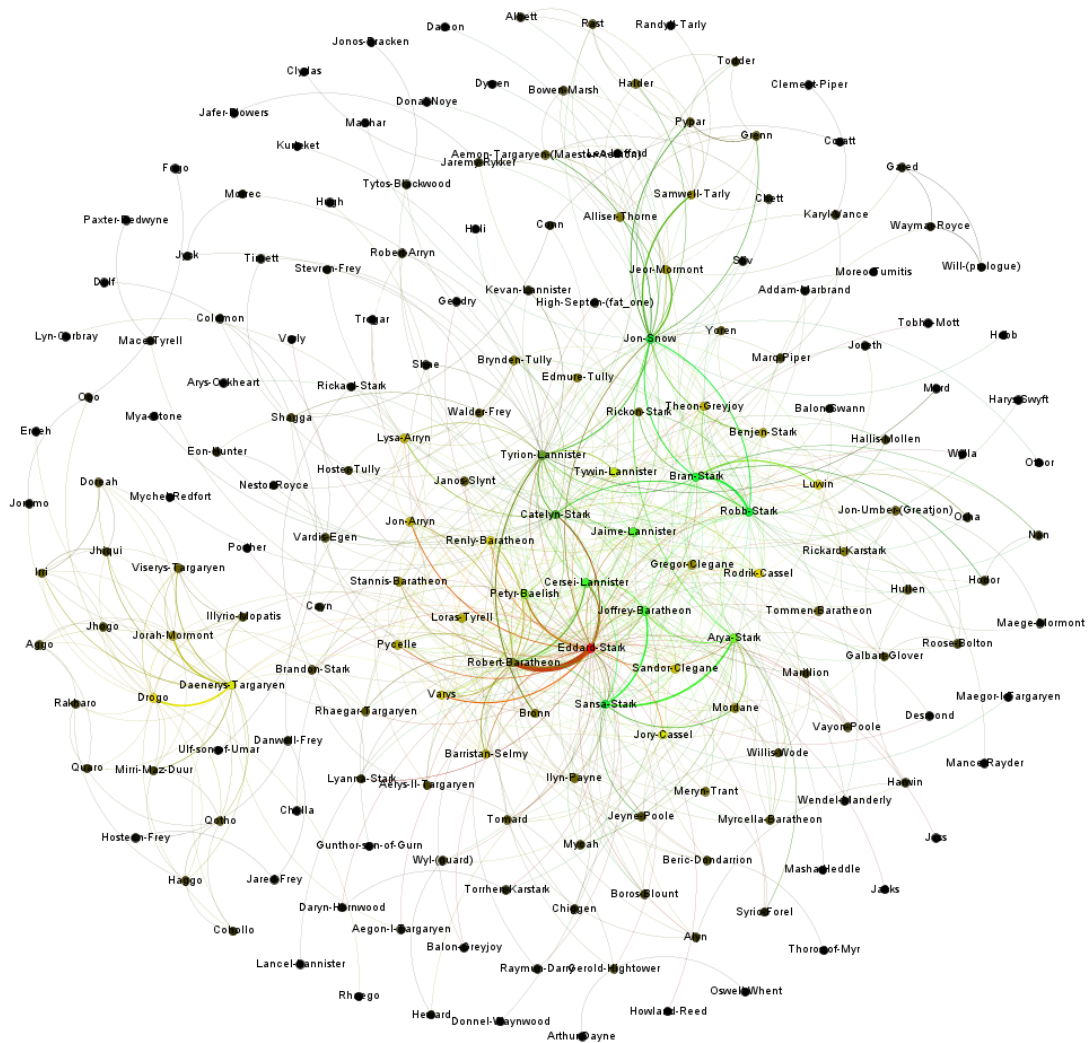


Figura 3: grafo personalizado, imagen\_muestra

Con esto completamos la tarea “Personalice la apariencia”

Para calcular el grado y verlo reflejado en el Data Table de Data Laboratory utilizamos las funciones de “Average Degree” y “Avg. Weighted Degree” que están en la sección “Statistics” de Overview”, usando eso tendremos la siguiente tabla:

Id	Label	Degree	Weighted Degree	Interval
Addam-Marbrand	Addam-Marbrand	2	9.0	
Jaime-Lannister	Jaime-Lannister	29	241.0	
Tywin-Lannister	Tywin-Lannister	22	181.0	
Aegon-I-Targaryen	Aegon-I-Targaryen	2	9.0	
Daenerys-Targaryen	Daenerys-Targaryen	21	443.0	
Eddard-Stark	Eddard-Stark	66	1284.0	
Aemon-Targaryen-(Maester-Aemon)	Aemon-Targaryen-(Maester-Aemon)	7	74.0	
Allysen-Thorne	Allysen-Thorne	10	80.0	
Bowen-Marsh	Bowen-Marsh	5	28.0	
Chett	Chett	3	19.0	
Clydas	Clydas	1	5.0	
Jeor-Mormont	Jeor-Mormont	13	181.0	
Jon-Snow	Jon-Snow	37	784.0	
Samwell-Tarly	Samwell-Tarly	12	148.0	
Aerys-II-Targaryen	Aerys-II-Targaryen	6	37.0	
Brandon-Stark	Brandon-Stark	6	41.0	
Gerold-Hightower	Gerold-Hightower	5	17.0	
Jon-Arryn	Jon-Arryn	15	169.0	
Robert-Baratheon	Robert-Baratheon	50	941.0	
Aggo	Aggo	6	36.0	
Drogo	Drogo	19	256.0	
Jhogo	Jhogo	6	43.0	
Jorah-Mormont	Jorah-Mormont	13	154.0	
Quaro	Quaro	5	21.0	
Rakharo	Rakharo	5	24.0	
Albrett	Albrett	3	11.0	
Halder	Halder	8	56.0	

Figura 4: grado con y sin peso del grafo, imagen\_muestra

Con esto completamos la tarea “calcule el grado”

Para detectar comunidades usamos las funciones de Statics en Overview, en la sección de “Community Detection” y usamos la función “Modularity” con los valores por defecto, de esta forma la tabla quedaría de la siguiente forma:

Id	Label	Interval	Degree	Weighted Degree	Modularity Class
Addam-Marbrand	Addam-Marbrand		2	9.0	5
Jaime-Lannister	Jaime-Lannister		29	241.0	3
Tywin-Lannister	Tywin-Lannister		22	181.0	5
Aegon-I-Targaryen	Aegon-I-Targaryen		2	9.0	0
Daenerys-Targaryen	Daenerys-Targaryen		21	443.0	0
Eddard-Stark	Eddard-Stark		66	1284.0	3
Aemon-Targaryen-(Maester-Aemon)	Aemon-Targaryen-(Maester-Aemon)		7	74.0	2
Allysen-Thorne	Allysen-Thorne		10	80.0	2
Bowen-Marsh	Bowen-Marsh		5	28.0	2
Chett	Chett		3	19.0	2
Clydas	Clydas		1	5.0	2
Jeor-Mormont	Jeor-Mormont		13	181.0	2
Jon-Snow	Jon-Snow		37	784.0	2
Samwell-Tarly	Samwell-Tarly		12	148.0	2
Aerys-II-Targaryen	Aerys-II-Targaryen		6	37.0	3
Brandon-Stark	Brandon-Stark		6	41.0	3
Gerold-Hightower	Gerold-Hightower		5	17.0	3
Jon-Arryn	Jon-Arryn		15	169.0	3
Robert-Baratheon	Robert-Baratheon		50	941.0	3
Aggo	Aggo		6	36.0	0
Drogo	Drogo		19	256.0	0
Jhogo	Jhogo		6	43.0	0
Jorah-Mormont	Jorah-Mormont		13	154.0	0
Quaro	Quaro		5	21.0	0
Rakharo	Rakharo		5	24.0	0
Albrett	Albrett		3	11.0	2
Halder	Halder		8	56.0	2

Figura 5: tabla nodos con comunidad, imagen\_muestra

y si aplicamos color a cada comunidad tenemos lo siguiente:

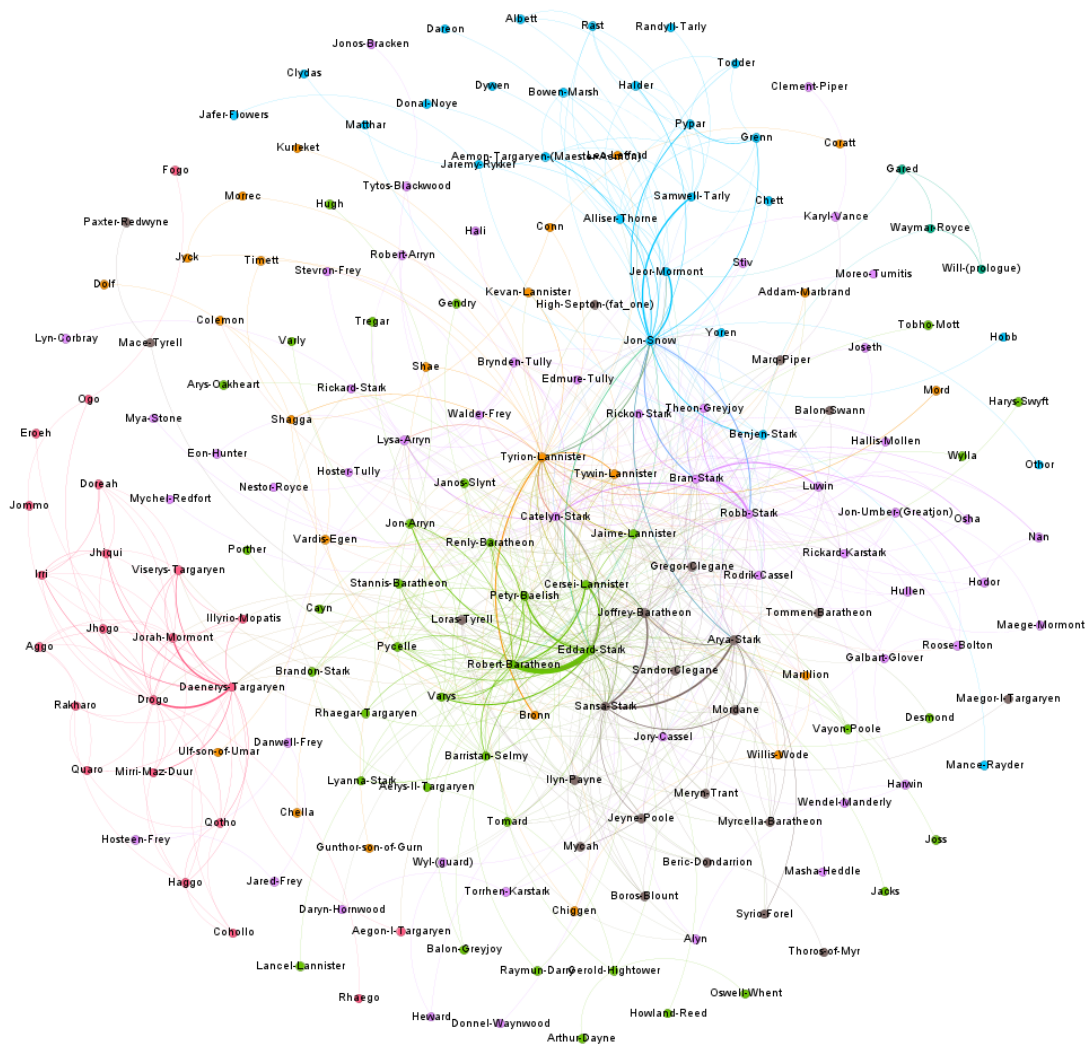


Figura 6: tabla nodos con comunidad, imagen\_muestra

Con esto completamos la tarea “Detecte comunidades”

Podemos obtener algunas conclusiones del grafo como las siguientes: 1. La persona con mayor impacto es “Eddard-Stark” ya que tiene la mayor grado con 66, y un peso de 1284, podemos verlo ya que si aplicamos un color a los nodos por peso, es el nodo con el color de más peso del grafo. 2. Hay peronsajes más importantes que otros, en el centro vemos nodos que tienen gran cantidad de conexiones y alrededor tenemos nodos con peso entre 1 y 2 que poco se conectan con los demás 3. Hay un total de 7 comunidades que se basan en las conexiones que existen entre los nodos, en otras palabras la función de “Modularity” tomo en cuenta unos 7 nodos que influyen o se relacionan más sobre los otros, y en este caso diriamos que son 7 personajes que tienen gran importancia dentro de los cuales 1 no se

relaciona con los otros personajes con un gran peso o importancia, ya que está más apartado, sería el de Daenerys Targaryen, que su comunidad es de un color Rosa oscuro.

Con esto completamos la tarea “Obtenga 2 o tres conclusiones”

#### **4. Conclusiones**

Aplicando multiples funciones y filtros al grafo podemos observar comportamientos que costarían ver usando una tabla, como las relaciones entre los nodos, la importancia entre ellos, la forma en que se relacionan, etc. Existen nodos que tienen mucha conexión entre los demás nodos y se relacionan a la vez entre sí, pero hay otros que no se relacionan entre ellos o están cerca uno del otro, es un ejemplo de lo importante que es poder filtrar un grafo por pesos y por comunidad.

#### **5 .Referencias bibliográficas**

No se utilizo ninguna bibliografía o consulta durante el desarrollo de este informe

#### **6 .Declaración de uso de ia**

La inteligencia artificial ayudó en la guía del uso de funciones en Gephi para poder obtener grados y comunidades.