

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“ANÁLISIS DE MÉTRICAS DE REDES CONTANDO CON INFORMACIÓN PARCIAL.”

BENJAMÍN OPAZO CUBILLOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Andrés Moreira W.
Profesor Correferente: Claudio Lobos Y.

Septiembre - 2023

DEDICATORIA

Para mis Papás...

A mi Papá, por ser mi ejemplo a seguir.

A mi Mamá, por tener más fe en mí de la que yo tengo.

A pesar de que sus expectativas pesan como el plomo, me motivaron a seguir adelante y a mantenerme alejado de la mediocridad.

Muchas Gracias

AGRADECIMIENTOS

Además, me gustaría agradecer a...

...mi Profesor guía que, como dice el nombre, me guió en este proceso y me permitió llevar las riendas del trabajo.

...mis Amigos, por su buena compañía y hacer las tardes más amenas.

...mis Gatos, por venir a interrumpir las largas horas de trabajo.

RESUMEN

Resumen— La ciencia de redes es una disciplina que ha ganado reconocimiento en los últimos años por las perspectivas y herramientas que aporta al estudio y análisis de fenómenos relacionales. Sumando la importancia de *Big Data* en la informática contemporánea, estas dos disciplinas traen tanto nuevas posibilidades como problemáticas. La magnitud de datos a trabajar conlleva no siempre de disponer de la totalidad de la información necesaria. Por esto, en este trabajo se abarca mencionado problema en busca de un punto de vista que permita mitigar los efectos de esta carencia de *data*. Mediante simulaciones de falta de datos, se experimenta con el fin de dilucidar patrones y comportamientos en las métricas más importantes de las redes. A partir de la experiencia surge una metodología, construida a partir de ejemplos, que prepara al análisis estadístico de un grafo a los posibles efectos de no disponer de toda la información.

Palabras Clave— *Ciencia de redes; Métricas de redes; Información parcial*

ABSTRACT

Abstract— Network science is a discipline that has gained recognition in recent years because of the outlooks and tools it brings to the study and analysis of relational phenomena. Adding the relevance of *Big Data* in contemporary computer science, these two disciplines bring both new possibilities and issues. The magnitude of data to be worked with does not always entail to have all the needed information available. Therefore, this work covers the aforementioned problem in search of a point of view that allows to mitigate the effects of data shortage. Through lack-of-data simulations, experiments are ran with the purpose of resolving network metrics patterns and behaviors. A methodology arises from the experience, built from examples, that prepares statistical analysis of a graph to the possibles effects of missing data.

Keywords— *Network Science; Network metrics; Partial information*

GLOSARIO

ER: Modelo Erdős-Rényi, refiriéndose al tipo de redes aleatorias en [Erdős y Rényi, 1960]

SMB: *Stochastic Block Model*, refiriéndose al tipo de redes planteadas en [Holland et al., 1983]

WS: Modelo Watts-Strogatz, refiriéndose al tipo de redes planteadas en [Watts y Strogatz, 1998]

BA: Modelo Barabási-Albert, refiriéndose al tipo de redes planteadas en [Barabási y Albert, 1999]

NetworkX: Librería de Python que se utilizará para la realización de los experimentos. Link: Documentación NetworkX

BFS: *Breadth-first search*, reconocido algoritmo de búsqueda en grafos, más información en el **Marco Conceptual** y en GeeksForGeeks.

DFS: *Depth-first search*, reconocido algoritmo de búsqueda en grafos, más información en el **Marco Conceptual** y en GeeksForGeeks.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	XII
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
CAPÍTULO 2: ESTADO DEL ARTE	6
2.1 <i>Network Sampling</i>	7
2.1.1 <i>Random Walks</i>	7
2.2 Daño a la red	8
2.3 Trabajando con falta de información	9
2.3.1 Estimación de las propiedades de una red	10
2.3.2 Análisis del efecto de información incompleta	10
2.3.3 Otros trabajos relacionados	10
CAPÍTULO 3: MARCO CONCEPTUAL	11
3.1 Redes y sus propiedades	11
3.1.1 Grado promedio	11
3.1.2 Distribución de grados	12
3.1.3 Coeficiente de <i>clustering</i>	12
3.1.4 Asortatividad	13
3.1.5 Distancia promedio y Diámetro	14
3.1.6 Robustez	15
3.2 Modelos de redes	16
3.2.1 Modelo Erdős-Rényi	16
3.2.2 Modelo de bloques estocásticos (<i>Stochastic block model</i>)	17
3.2.3 Modelo Watts-Strogatz (<i>Small world networks</i>)	18
3.2.4 Modelo Barabási-Albert (<i>Scale-free networks</i>)	19
3.2.5 Mallas regulares	21
3.3 Tipos de falta de información	22
3.3.1 Eliminación de nodos	22
3.3.2 Eliminación de conexiones	23
3.3.3 Caminos Aleatorios	23
3.3.4 Muestreo por Exploración, BFS y DFS	25

CAPÍTULO 4: MARCO EXPERIMENTAL	26
4.1 Redes de muestra	26
4.1.1 Redes reales	26
4.1.2 Redes generadas	27
4.2 Procedimiento y Extracción	28
CAPÍTULO 5: RESULTADOS	33
5.1 Redes Generadas Originales	34
5.1.1 Modelo ER	35
5.1.2 Modelo SBM	36
5.1.3 Modelo WS	37
5.1.4 Modelo BA	38
5.1.5 Modelo Grid	39
5.2 Redes Reales Originales	40
5.3 Comparación Original - Alterado	43
5.3.1 Redes Generadas	43
5.3.2 Redes Reales	49
CAPÍTULO 6: DISCUSIÓN	56
6.1 Redes Generadas	56
6.1.1 Modelo ER	56
6.1.2 Modelo SBM	60
6.1.3 Modelo WS	63
6.1.4 Modelo BA	66
6.1.5 Modelo Grid	68
6.2 Redes Reales	72
6.2.1 Red de Interacción de Proteínas	72
6.2.2 Red de captura de Campaños	74
6.2.3 Red Eléctrica en Estados Unidos	74
6.2.4 Red Social Anybeat	75
6.2.5 Red Social Google Plus	76
6.3 Recapitulación	77
6.3.1 Resumen Redes Generadas	78
6.3.2 Resumen Redes Reales	80
CAPÍTULO 7: CONCLUSIONES	83
7.1 Comentarios Preliminares	83
7.2 Revisión de objetivos propuestos	85
ANEXOS	88
7.1 Algoritmo de generación bloques SBM	88
7.2 Algoritmo de generación probabilidades SBM	89
7.3 Algoritmo BFS modificado	90
7.4 Algoritmo DFS modificado	91

7.5	Detalles y Fuentes de Redes reales	92
7.6	Resultados originales para ER	93
7.7	Resultados originales para SBM	94
7.8	Resultados originales para WS	95
7.9	Resultados originales para BA	96
7.10	Resultados originales para Grid	97
7.11	Comparación de resultados restantes para ER	98
7.12	Comparación de resultados restantes para SBM	102
7.13	Comparación de resultados restantes para WS	106
7.14	Comparación de resultados restantes para BA	110
7.15	Comparación de resultados restantes para Grid	114
7.16	Comparación de resultados restantes para Red 1	118
7.17	Comparación de resultados restantes para Red 2	122
7.18	Comparación de resultados restantes para Red 2	126
7.19	Comparación de resultados restantes para Red 4	130
7.20	Comparación de resultados restantes para Red 5	134
REFERENCIAS BIBLIOGRÁFICAS		138

ÍNDICE DE FIGURAS

1	Ejemplificación de la experiencia.	2
2	Árbol del problema.	5
3	Ejemplos de <i>Network Sampling</i>	7
4	Ejemplo de <i>Random Walk</i>	8
5	Ejemplo de daño a una red.	9
6	Visualización de una red incompleta.	9
7	Grafo de ejemplo.	11
8	Distribución de grados de una red.	12
9	Ejemplo del diámetro de una red.	15
10	Ejemplo redes Erdős-Rényi.	17
11	Ejemplo red SBM con tres bloques.	18
12	Ejemplo red Watts-Strogatz	19
13	Comparación Red aleatoria versus Libre de escala.	19
14	Ejemplo de Ley de potencia en escala normal (izquierda) y logarítmica (derecha). 20	
15	Ejemplo de malla regular cuadrada (<i>lattice</i>).	21
16	Ejemplo de red generada con modelo ER.	30
17	Ejemplo de red alterada con el Algoritmo 1.	30
18	Ejemplo de red alterada con el Algoritmo 2.	31
19	Distribuciones de resultados para redes ER originales.	35
20	Distribuciones de resultados para redes SBM originales.	36
21	Distribuciones de resultados para redes WS originales.	37
22	Distribuciones de resultados para redes BA originales.	38
23	Distribuciones de resultados para redes Grid originales.	39

24	Distribuciones de resultados para red real 1.	40
25	Distribuciones de resultados para red real 2.	41
26	Distribuciones de resultados para red real 3.	41
27	Distribuciones de resultados para red real 4.	42
28	Distribuciones de resultados para red real 5.	42
29	Comparación de resultados algoritmo 1 para redes ER.	44
30	Comparación de resultados algoritmo 1 para redes SBM.	45
31	Comparación de resultados algoritmo 1 para redes WS.	46
32	Comparación de resultados algoritmo 1 para redes BA.	47
33	Comparación de resultados algoritmo 1 para redes Grid.	48
34	Comparación de resultados algoritmo 1 para red 1.	50
35	Comparación de resultados algoritmo 1 para red 2.	51
36	Comparación de resultados algoritmo 1 para red 3.	52
37	Comparación de resultados algoritmo 1 para red 4.	53
38	Comparación de resultados algoritmo 1 para red 5.	54
39	Distribuciones de resultados para redes ER originales. Completa.	93
40	Distribuciones de resultados para redes SBM originales. Completa.	94
41	Distribuciones de resultados para redes WS originales. Completa.	95
42	Distribuciones de resultados para redes BA originales. Completa.	96
43	Distribuciones de resultados para redes Grid originales. Completa.	97
44	Comparación de resultados algoritmo 2 para redes ER.	98
45	Comparación de resultados algoritmo 3 para redes ER.	99
46	Comparación de resultados algoritmo 4 (BFS) para redes ER.	100
47	Comparación de resultados algoritmo 4 (DFS) para redes ER.	101

48	Comparación de resultados algoritmo 2 para redes SBM.	102
49	Comparación de resultados algoritmo 3 para redes SBM.	103
50	Comparación de resultados algoritmo 4 (BFS) para redes SBM.	104
51	Comparación de resultados algoritmo 4 (DFS) para redes SBM.	105
52	Comparación de resultados algoritmo 2 para redes WS.	106
53	Comparación de resultados algoritmo 3 para redes WS.	107
54	Comparación de resultados algoritmo 4 (BFS) para redes WS.	108
55	Comparación de resultados algoritmo 4 (DFS) para redes WS.	109
56	Comparación de resultados algoritmo 2 para redes BA.	110
57	Comparación de resultados algoritmo 3 para redes BA.	111
58	Comparación de resultados algoritmo 4 (BFS) para redes BA.	112
59	Comparación de resultados algoritmo 4 (DFS) para redes BA.	113
60	Comparación de resultados algoritmo 2 para redes Grid.	114
61	Comparación de resultados algoritmo 3 para redes Grid.	115
62	Comparación de resultados algoritmo 4 (BFS) para redes Grid.	116
63	Comparación de resultados algoritmo 4 (DFS) para redes Grid.	117
64	Comparación de resultados algoritmo 2 para red 1.	118
65	Comparación de resultados algoritmo 3 para red 1.	119
66	Comparación de resultados algoritmo 4 (BFS) para red 1.	120
67	Comparación de resultados algoritmo 4 (DFS) para red 1.	121
68	Comparación de resultados algoritmo 2 para red 2.	122
69	Comparación de resultados algoritmo 3 para red 2.	123
70	Comparación de resultados algoritmo 4 (BFS) para red 2.	124
71	Comparación de resultados algoritmo 4 (DFS) para red 2.	125

72	Comparación de resultados algoritmo 2 para red 3.	126
73	Comparación de resultados algoritmo 3 para red 3.	127
74	Comparación de resultados algoritmo 4 (BFS) para red 3.	128
75	Comparación de resultados algoritmo 4 (DFS) para red 3.	129
76	Comparación de resultados algoritmo 2 para red 4.	130
77	Comparación de resultados algoritmo 3 para red 4.	131
78	Comparación de resultados algoritmo 4 (BFS) para red 4.	132
79	Comparación de resultados algoritmo 4 (DFS) para red 4.	133
80	Comparación de resultados algoritmo 2 para red 5.	134
81	Comparación de resultados algoritmo 3 para red 5.	135
82	Comparación de resultados algoritmo 4 (BFS) para red 5.	136
83	Comparación de resultados algoritmo 4 (DFS) para red 5.	137

ÍNDICE DE TABLAS

1	Magnitudes de redes conocidas.	6
2	Ejemplo de extracción de propiedades.	31
3	Tiempos de ejecución promedio para redes generadas.	33
4	Tiempos de ejecución promedio para redes reales.	33
5	Valores promedio de las redes originales ER.	35
6	Valores promedio de las redes originales SBM.	36
7	Valores promedio de las redes originales WS.	37
8	Valores promedio de las redes originales BA.	38
9	Valores promedio de las redes originales Grid.	39
10	Valores de las propiedades de las redes reales originales.	40

11	Resumen resultados para modelo ER.	78
12	Resumen resultados para modelo SBM.	78
13	Resumen resultados para modelo WS.	79
14	Resumen resultados para modelo BA.	79
15	Resumen resultados para modelo Grid.	80
16	Resumen resultados para Red 1.	80
17	Resumen resultados para Red 2.	81
18	Resumen resultados para Red 3.	81
19	Resumen resultados para Red 4.	82
20	Resumen resultados para Red 5.	82

INTRODUCCIÓN

Enraizada en la matemática, la teoría de grafos fue de los primeros grandes pasos que se dieron para el modelamiento, simulación y análisis de fenómenos de toda índole que destacan por su naturaleza relacional. Entiéndase por esto lo siguiente: múltiples entidades que, dentro del contexto de estudio, se podrían considerar de una misma categoría, por ejemplo, personas. Estas entidades, pueden relacionarse con otros miembros mediante una conexión relevante para marco de análisis. De esta manera, se modelan en grafos mediante vértices y aristas. Así, las herramientas que entrega la teoría de grafos permiten extraer información de estos modelos, tal como, obtener los caminos más cortos entre cada par de vértices, encontrar nodos importantes, entre otros.

Con el paso de los años, no solo crecieron las cantidades de datos a estudiar, si no que, a la par, fue avanzando el fundamento teórico y la capacidad de cómputo práctica. De trabajar con muestras de centenas de miembros, se pasó a las redes con millones de componentes. Del cálculo a mano, se pasó a computadores cada vez más poderosos. Y de la teoría de grafos, se pasó a la ciencia de redes. Basada firmemente en la teoría mencionada, esta disciplina naciente agregó a su repertorio múltiples herramientas, tales como la estadística, la minería de datos, la visualización de información, incluso indicios y estructuras sociales provenientes de la sociología.

Este campo se fue popularizando más y más en la tecnologías de la información. Muchos fenómenos actuales relevantes para la industria puede ser modelados mediante una red. Un ejemplo clásico son la redes sociales, ya sean o no digitales. El ámbito productivo, poco a poco, ha ido adaptando a sus estrategias de *business intelligence*, un enfoque de ciencia de redes. Predicción de conexiones y sistemas de recomendación, detección de fraudes, búsquedas y recuperación de información, entre muchos otros, son ejemplos de aplicaciones industriales de las herramientas mencionadas. El detalle de estos casos se desarrollan con más detalle en [Khraisha, 2020].

Considerando a la información como el principal foco de, valga la redundancia, la informática como disciplina, y en vista de la naturaleza de los datos en la actualidad, se puede llegar a la cuestión: ¿Qué pasa cuando no está disponible toda la información?

En la actualidad, y específicamente en el mundo de *BigData*, las magnitudes de datos con la que se trabaja pasaron ya de los *terabytes* a los *petabytes*. Este enorme volumen de *raw data* no siempre se encuentra disponible y a disposición de su uso. Ya sea por pérdida o daño de algún registro ya existente o imposibilidades prácticas de obtener el completo de los datos requeridos, existen variados motivos por los cuales, hoy en día, se trabajaría con información parcial. Empero, esto no deshabilita completamente el uso de data incompleta. Esta problemática se ha abarcado en literatura reciente, tomando distintos enfoques para reducir el impacto de esta carencia.

Dicho todo esto, en este trabajo se buscan generar perspectivas, *insights*, de como las men-

cionadas faltas de información afectan al análisis estadístico de redes. Primero, a continuación en este escrito, se delimitará formalmente el enfoque y alcance de la problemática a estudiar y de la propuesta generada en base a ésta. Luego, una discusión de las investigaciones recientes relevantes al contexto darán a conocer el estado del arte respecto a este tema. Una contextualización conceptual le sigue para establecer definiciones, modelos y procedimientos en los que se fundamenta este estudio. A continuación, se detallará el entorno de experimentación. Entiéndasela delimitación de muestras, sujetas al estudio y de control, procedimiento a aplicar, y datos a extraer para formular el conocimiento generado a partir de la experiencia. Ya realizado esto, los resultados serán expuestos en gráficos de comparación y; posibles patrones de comportamiento, presentados. Finalmente, las conclusiones discutirán sobre el cumplimiento de los objetivos del trabajo y potencial trabajo futuro.

Entrando en detalle sobre el experimento a conducir, en el **Marco Conceptual** se definen tipos, modelos específicamente, de redes y tipos de falta de información, que se traducen en procesos y técnicas para simular data faltante. En el **Marco Experimental**, las redes a utilizar en el experimento serán expuestas formalmente. Este grupo representará tanto redes reales como generadas. Las reales cumplen las características de los algunos modelos de red propuestos y las generadas, son producidas con métodos acorde a estos. Ya con la muestra presente, los algoritmos de simulación de falta de información son aplicados para generar un nuevo grupo de redes “alteradas”¹. A ambos grupos se le calcularán distintas estadísticas, también presentadas en el **Marco Conceptual**. Esto con el fin de compararlas entre sí y extrapolar comportamientos y patrones que los tipos de data faltante generen. Se espera que estas conductas sean útiles al momento de hacer estudios estadísticos sobre redes con faltas, complementando el resultado.

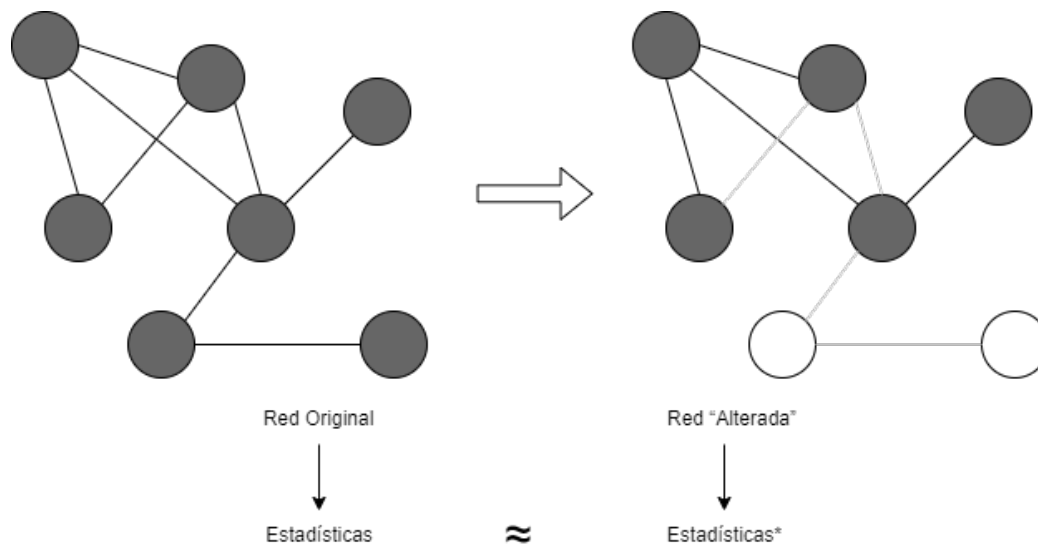


Figura 1: Ejemplificación de la experiencia.

Fuente: Elaboración propia.

¹“Alteradas” se refiere a las redes que simulan información faltante.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

“La perspectiva de redes nos permite abarcar preguntas profundas sobre sistemas humanos (sociales), biológicos, económicos, entre otros, que presenten una organización interdependiente” [Brandes *et al.*, 2013]. Los sistemas presentes en la realidad son sumamente complejos, dificultando la deducción de su comportamiento a partir del conocimiento de sus componentes [Barabási, 2016]. Para abarcar el análisis de fenómenos que destacan por tener una estructura subyacente interconectada, se utilizan métodos, prácticas y modelos desarrollados por la ciencia de redes.

Uno de sus mayores exponentes, Albert-László Barabási, destaca la importancia y efectividad de la ciencia de redes listando su características clave [Barabási, 2016]:

- **Naturaleza interdisciplinaria:** se establece un lenguaje común que distintas disciplinas puedan comprender, los grafos.
- **Conducido por datos empíricos:** sienta sus bases en la teoría de grafos, sumándole información real, funcional y útil.
- **Base matemática y cuantificable:** se formaliza mediante la precisión de las matemáticas.
- **Apoyo computacional:** dadas las magnitudes de los datos trabajados, la vanguardia del campo trabaja estrechamente con las ciencias de la computación.

Desde fines del siglo XX, esta disciplina ha evolucionado para tratar problemas de múltiples índoles y áreas. Dentro de sus objetivos principales, se encuentran el modelo de distintos tipos de fenómenos, la descripción de su comportamiento actual y predicciones sobre su posible futura conducta. La metodología preliminar para cumplir estos fines es hacer cálculos y mediciones provenientes principalmente de la teoría de grafos. La métricas resultantes de este proceso son una herramienta fundamental de la ciencia de redes. Sobre estos resultados, una herramienta común a aplicar son varios instrumentos estadísticos que permitan generar conocimiento útil sobre el fenómeno estudiado. Por ejemplo, si se quiere saber la estructura inherente de una red de relaciones entre personas, un posible paso es computar y revisar su distribución de grados ², además de otras medidas a partir del grafo. De esto se pueden extraer la existencia de distintas características y propiedades, tales como, la presencia de un “club de ricos” [Colizza *et al.*, 2006] en una red.

Un supuesto elemental con el que trabajó en sus comienzos esta ciencia es percibir la red subyacente al fenómeno por completo. Es decir, se tenía conocimiento de todos los nodos in-

²Sea grado la cantidad de conexiones que tiene un nodo. Este y otros conceptos sobre redes serán explicados más adelante.

volucrados y las conexiones entre ellos. Cuando se aplicó el análisis mencionado a ejemplos reales de redes sociales ³ se notó que el entendimiento al completo sobre la red no siempre era el caso percibido [Kossinets, 2006]. Distintos tipos de falta de información llevan a múltiples problemas al momento de aplicar las técnicas, ya sea de predicción, descripción o modelamiento. Un ejemplo conocido en la literatura es el desconocimiento de todas las conexiones de la red, cuyo principal impedimento práctico es dificultar la detección de comunidades [Warnke, 2016]. Esta falta de información causa un gran obstáculo a la efectividad de los métodos de la ciencia de redes y, dada la magnitud de los datos generados y estudiados en la actualidad [Lancichinetti *et al.*, 2011], es cada vez más común no disponer de la red completa. La carencia de completitud de un grafo que busca simular el fenómeno real de manera fiel no es de naturaleza determinista. No es posible en la mayoría de los casos, saber con certeza, por ejemplo, que porcentaje de los nodos son desconocidos, o de la cantidad de conexiones que faltan. Existen distintos procedimientos para extraer información de una red real que, dependiendo de esta, puede proporcionar grafos incompletos. Entre estos métodos se pueden encontrar las *ego-networks* [Arnaboldi *et al.*, 2016], *samples* de redes [Granovetter, 1976], redes ofuscadas ⁴, entre otros.

Es importante exponer que la relación entre las métricas de la red subyacente y los objetivos del análisis de ciencia de redes suele ser el mencionado estudio estadístico sobre las mediciones. Estos resultados llevan, por ejemplo, a hacer predicciones sobre el camino que un fenómeno puede tomar. Como se mencionó previamente, la falta de información afecta directamente al comportamiento de las propiedades del grafo estudiado. Por consecuencia, afecta indirectamente al análisis estadístico que busca cumplir los posibles objetivos planteados. Es por esto que identificar una posible relación entre el tipo de falta de información y el comportamiento de las métricas de la red puede complementar los resultados de los futuros estudios sobre fenómenos modelables mediante redes.

Se propone buscar una posible pauta que sigan las propiedades de las redes cuando se pasa de contar con información completa a perder parte del conocimiento del grafo. En capítulos futuros se definirán tipos de modelos de redes ⁵ (sean redes generadas aleatoriamente o redes reales), tipos de falta de información a aplicarle a estas ⁶ y métricas a extraer ⁷ que serán comparadas en la experimentación. Respecto a la experimentación, a lotes de redes se les aplicarán los tipos de falta de información en distintas medidas. A la versión con la falta aplicada y al grafo original, sus propiedades serán calculadas y comparadas con el fin de dilucidar el comportamiento de las métricas bajo la simulación.

Resumiendo, rescatar un patrón de comportamiento de las propiedades de una red sometida a distintos tipos de falta de información puede beneficiar a los análisis estadísticos que se apliquen sobre el grafo que modele algún fenómeno estudiado. Antes de cualquier expe-

³Red que modela conexiones entre personas, no necesariamente plataformas de redes sociales en línea.

⁴Redes obtenidas al completo, pero se sabe a ciencia cierta que el fenómeno real es más extenso.

⁵Tales como redes aleatorias Erdős-Rényi, redes *scale-free*, entre otros.

⁶Como se mencionó antes, nodos faltantes, conexiones faltantes, entre otros.

⁷Distribución de grados, coeficientes de *clustering*, etc. Junto a los conceptos de la experimentación, se describirán en profundidad más adelante.

rimento, se definen los tipos de redes y faltas de información a estudiar. En otras palabras, los objetivos de este trabajo abarcan la investigación y definición de los problemas surgidos a partir de la carencia de conocimiento total sobre redes y el diseño de experimentos con el fin de dilucidar conductas nacidas de la problemática planteada. Para conseguir el último objetivo, la propuesta es aplicar las mencionadas faltas de información a redes generadas y reales, calcular sus propiedades y hacer una comparación y análisis de los valores antes y después aplicar la falta. Se espera poder generar conocimiento en forma de relaciones cualitativas respecto a los comportamientos encontrados. En la figura 2 se desglosa el problema en un mapa conceptual tipo árbol.

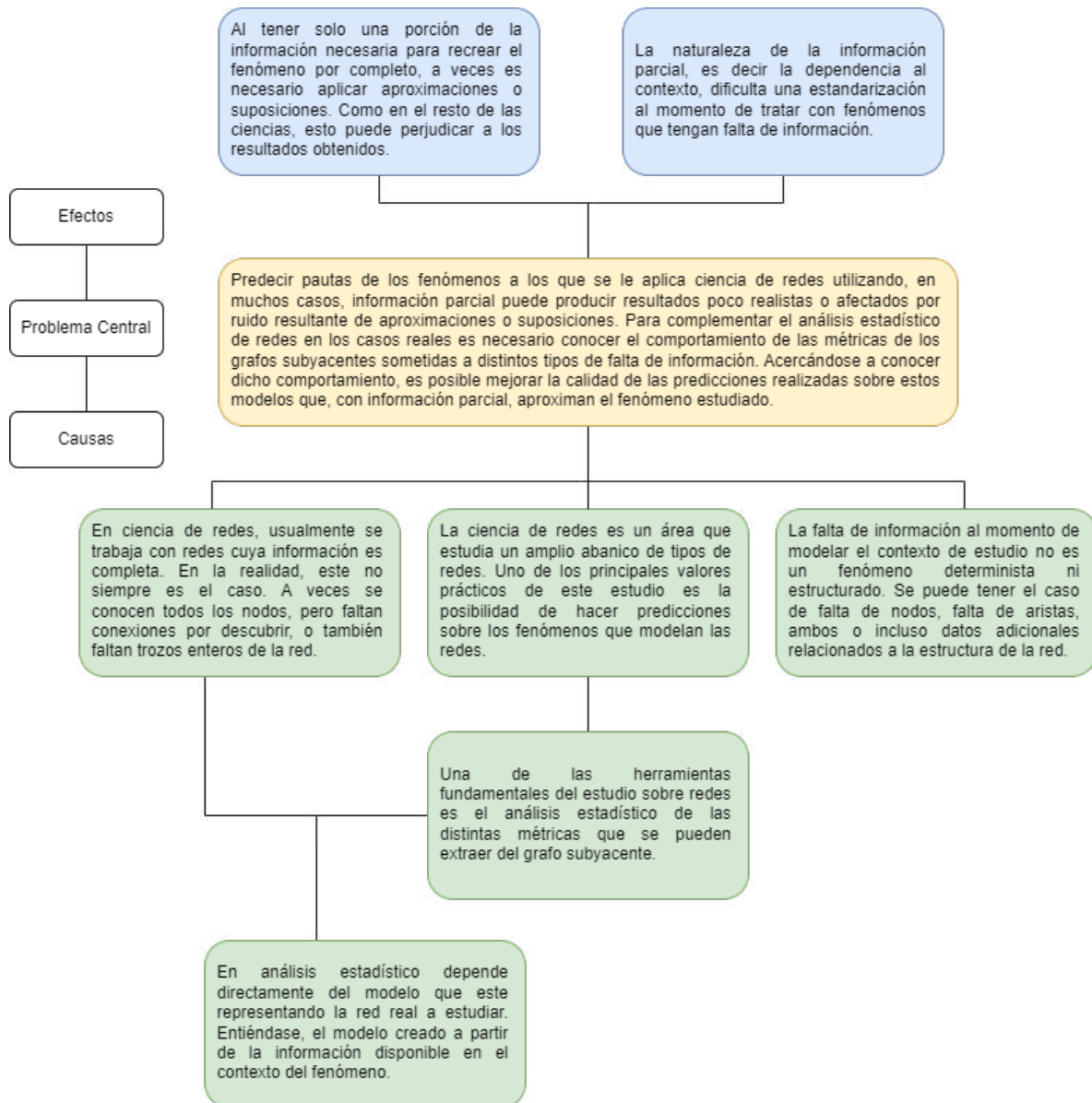


Figura 2: Árbol del problema.

Fuente: Elaboración propia.

CAPÍTULO 2

ESTADO DEL ARTE

El problema de no disponer del total de la información sobre una red a estudiar es conocido en el área de la ciencia de redes. Además, este acontecimiento no solo surge por la naturaleza del fenómeno observado (por ejemplo, parte de la información sobre la red simplemente no existe), sino que puede nacer de los métodos de extracción de datos sobre la estructura que se quiere modelar.

A veces, las estructuras interconectadas subyacentes a fenómenos de interés, ya sea puramente académico o con alguna finalidad práctica, pueden ser demasiado grandes o enigmáticos para disponer acceso a su largo y ancho. Para atacar este problema, existen distintos métodos de extracción de información, los cuales abarcan desde “caminar”⁸ a través de las conexiones de un nodo conocido, hasta tomar “trozos”, subgrafos, de lo que se conoce de la red original.

Red	Nodos	Enlaces	N	E	⟨K⟩
Internet	Routers	Conexiones de Red	192,244	609,066	6.34
WWW	Páginas Web	Hipervínculos	325,729	1,497,134	4.60
Red Eléctrica	Centrales, transformadores	Cables	4,941	6,594	2.67
Llamadas entre Celulares	Subscriptores	Llamadas	36,595	91,826	2.51
Email	Direcciones de Correo	Emails	57,194	103,731	1.81
Colaboración Científica	Científicos	Co-autorías	23,133	93,437	8.08
Red de Actores	Actores	Co-actuación	702,388	29,397,908	83.71
Red de Citaciones	Papers	Citaciones	449,673	4,689,479	10.43
Metabolismo de E. coli	Metabolitos	Reacciones químicas	1,039	5,802	5.58
Interacciones Proteicas	Proteínas	Interacciones vinculantes	2,018	2,930	2.90

Tabla 1: Magnitudes de redes conocidas.
Fuente: [Barabási, 2016]. Traducción propia.

⁸Se entiende “caminar” por una red como realizar un camino saltando de un nodo a otro mediante sus conexiones.

2.1. Network Sampling

Con el objetivo de obtener una muestra representativa de una red, extraer sus propiedades y extrapolar información útil sobre esta sin tener que observar la red en su completitud ⁹, fueron desarrolladas técnicas de *sampling*. Revisando la tabla 1 y considerando a **N**, **E** y **K** como cantidad de nodos, de enlaces y grado promedio respectivamente, se puede observar el volumen de datos que puede llegar a tratar la ciencia de redes ¹⁰.

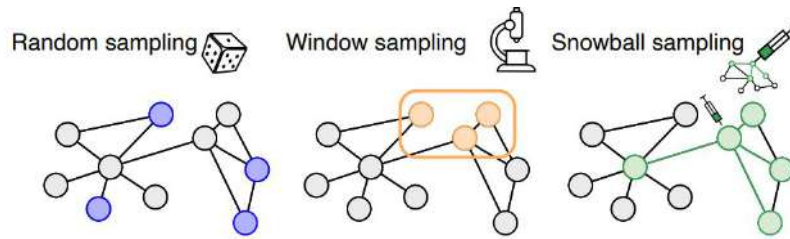


Figura 3: Ejemplos de *Network Sampling*.

Fuente: [Levina et al., 2022].

El *sampling* o muestreo de redes es, a grandes rasgos, tomar un subconjunto de nodos de la comunidad que componen la red completa (o conocida) [Granovetter, 1976]. Profundizando, el método clásico es tomar una muestra aleatoria de tamaño n , con $n < N$, siendo N la cantidad de total de nodos. Distintas técnicas pueden traer sus respectivos sesgos [Levina et al., 2022]. El mencionado muestreo aleatorio logra representar su conjunto progenitor, pero aplica un sesgo en los resultados. Otro ejemplo, es el muestreo por ventana, que consigue una buena perspectiva local, pero puede no reproducir el comportamiento global. Ejemplos gráficos del *sampling* pueden ser encontrados en la figura 3.

En suma, el *network sampling* es una valiosa herramienta utilizada en la ciencia de redes para abarcar fenómenos masivos. Como se discutió, esta técnica produce subgrafos que intentan representar la situación real a partir de información parcial. Lamentablemente, como mostró [Levina et al., 2022], el método trae consigo sesgos y distorsiones en sus resultados.

2.1.1. Random Walks

Otra técnica utilizada en el contexto de obtención de información de redes reales son los *Random Walks*, o Caminos Aleatorios. Siendo este método un “hijo” del *network sampling*, tiene un propósito similar, abarcar redes masivas en busca de un subconjunto representativo. Partiendo de un nodo conocido, se “camina” por sus conexiones y los vecinos de estas sucesivamente, generando así el subconjunto de muestra [Levina et al., 2022, Ribeiro et al., 2012].

⁹Revisar la red completa sería lo óptimo, pero por las magnitudes de estas, se vuelve infactible.

¹⁰Es posible tomar la oportunidad para recalcar la ubicuidad del campo viendo los contextos de las redes presentadas.

Este proceso es ilustrado en la figura 4. El principal error asociado a la estimación de parámetros usando *random walks* nace de la cantidad de pasos que tenga el camino versus el tamaño real de la red [Cooper *et al.*, 2014]. Más pasos traen más precisión a los resultados obtenidos, pero, en algunos contextos, cada paso puede ser caro, computacionalmente o no [Levina *et al.*, 2022, Cooper *et al.*, 2014]. El caso del costo no computacional puede nacer del método de obtención de la información. En la referencia mencionada se presenta el ejemplo de trazar una red de neuronas de algún individuo. Rescatar las conexiones es sumamente caro en su contexto, la neurociencia.

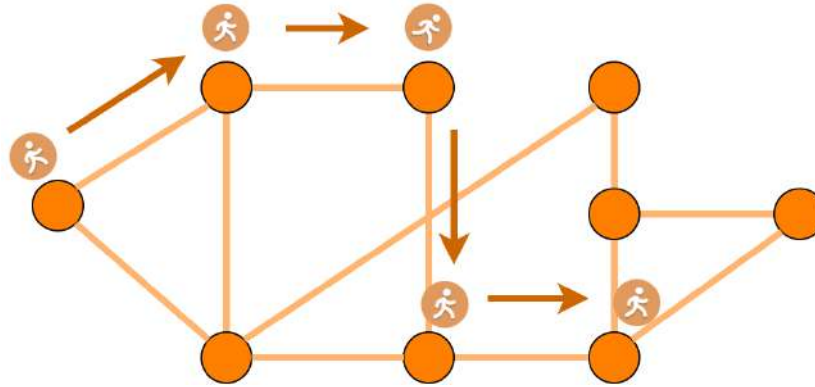


Figura 4: Ejemplo de *Random Walk*.
Fuente: [Özcelik, 2019]

2.2. Daño a la red

Si bien no es una común fuente “natural”¹¹ de falta de información como lo son los métodos anteriores, existen investigaciones sobre el comportamiento de las propiedades de una red cuando esta es dañada¹². Se puede encontrar una visualización en la figura 5. [Sarkar *et al.*, 2016] estudia esta conducta cuantificándola con dos medidas: sensibilidad, que mide la magnitud del cambio de las cualidades de una red al dañarla, y fiabilidad, relacionada a la efectividad de los procesos en la red al perder conexiones. La métrica relevante para este trabajo es la primera. Particularmente, la metodología en cuestión trata de aplicar, a las redes, distintos tipos de ruido, los cuales consistían en falta de conexiones de manera aleatoria o de algún mecanismo dependiente del contexto [Yan y Gregory, 2011]. Con esta alteración empleada, se calculaban las medidas definidas en este párrafo.

Otro estudio relevante fue desarrollado por [Borgatti *et al.*, 2006] busca comprender la robustez y resiliencia de las métricas extraídas de una red. Esto significa la “veracidad” de la información que proporciona una red sabiendo que puede estar distorsionada, dañada o

¹¹Se dice natural porque, en el contexto de ciencia de redes, el daño a una red se suele producir a propósito con fines académicos.

¹²Se entiende “daño” a una red cuando ésta pierde nodos o conexiones.

simplemente incompleta. Análogamente a los trabajos citados previamente, se le aplican alteraciones a las redes. Esta vez, no solo pérdidas, sino que también adiciones de nodos y conexiones.

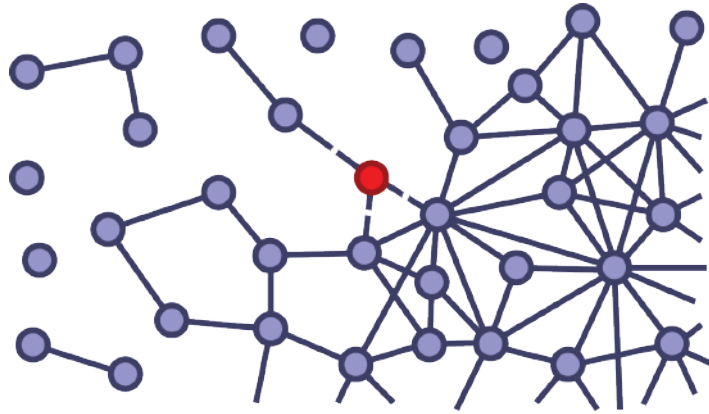


Figura 5: Ejemplo de daño a una red.

Fuente: [Gross, 2022]

2.3. Trabajando con falta de información

Habiendo establecido las fuentes de falta de información más comunes en la investigación reciente, se puede pasar al trabajo desarrollado a tratar con estas redes incompletas. Muchos de los trabajos en esta línea de investigación analizan medidas de centralidad, y por consecuencia, detección de comunidades cuando se aplican faltas de información. En este trabajo no se abarcarán las comunidades, pero es importante destacar el enfoque de los estudios actuales. Pese a eso, también existe literatura sobre la métricas que sí se abarcarán en esta experiencia ¹³.

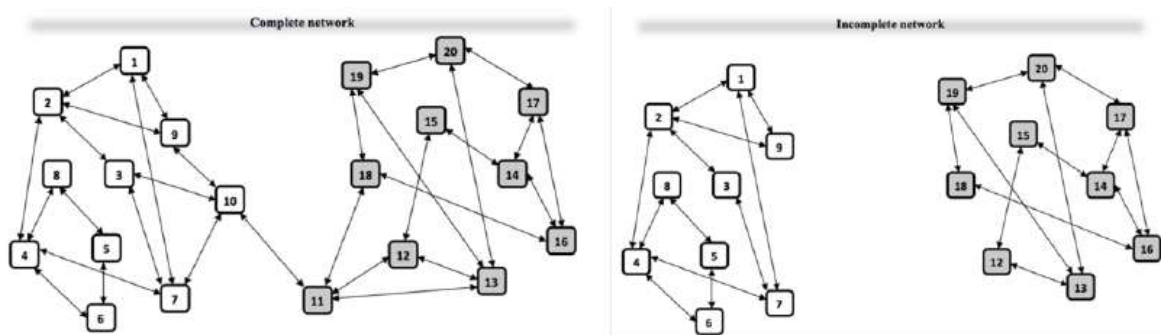


Figura 6: Visualización de una red incompleta.

Fuente: [Hristov, 2015]

¹³Las métricas a abarcar se definirán en la sección 3.1 del **Marco Conceptual**

2.3.1. Estimación de las propiedades de una red

[Cooper *et al.*, 2014, Ribeiro *et al.*, 2012] al trabajar con subconjuntos generados a partir de *random walks*, su trabajo se enfocó en rescatar las propiedades de redes tratando de estimar el valor que daría la red completa. El proceso consiste en calcular las métricas del subgrafo generado por el camino aleatorio e iterar agregando “pasos”, nodos, al camino. El procedimiento se repite hasta la convergencia. [Cooper *et al.*, 2014] muestra y menciona lo prometedor del *framework* de trabajo definido y su bajo costo computacional. Aun así, es importante recordar el sesgo que traen los muestreos de redes como mencionó [Levina *et al.*, 2022].

2.3.2. Análisis del efecto de información incompleta

[Kossinets, 2006], focalizando su trabajo específicamente en las redes sociales, analiza y simula “mecanismos” de falta de información característicos de las redes en las que se centra el estudio. Aplicando esto a un red real de colaboración científica, presenta los efectos sobre las distintas propiedades. A pesar de asemejarse a lo propuesto en este trabajo, aquí se propone ampliar el espectro de los tipos de redes, en este caso, no solo reales, sino que también generadas. Además, los tipos de faltas de información a considerar se mantendrán independientes del ámbito al que pertenezca la red. Sin embargo, es necesario recalcar el aporte que trae esta investigación al trabajo actual.

2.3.3. Otros trabajos relacionados

Finalmente, cabe destacar el creciente interés por el problema de *Missing Link Inference* [Litant, 2013] que escapa a los propósitos que se buscan cumplir, pero que nace de las mismas raíces. Desde redes sociales, para recomendar nuevas amistades, hasta redes criminales para predecir conexiones a organizaciones ilícitas, el área tiene una gran cantidad de literatura. Como se mencionó en la **Definición del Problema**, el objetivo es proporcionar conocimiento que pueda apoyar a los análisis estadísticos que se puedan efectuar en los tipos de redes que se han discutido hasta ahora, masivas, incompletas y/o dañadas.

CAPÍTULO 3

MARCO CONCEPTUAL

3.1. Redes y sus propiedades

La fundamentación matemática de la ciencia de redes se basa casi completamente en la teoría de grafos [Chartrand, 1977], nacida, a su vez, del famoso problema de los puentes de *Königsberg* [Euler, 1741]. Lo primero a entender es qué se conoce como red, o grafo en su definición fundamental ¹⁴.

Sea un grafo G un conjunto finito de N vértices denominado V y otro conjunto finito de aristas llamado E . Siendo estas aristas pares que representan relaciones entre los elementos de V [Chartrand, 1977]. Para los propósitos de este trabajo, se consideran sólo grafos no dirigidos.

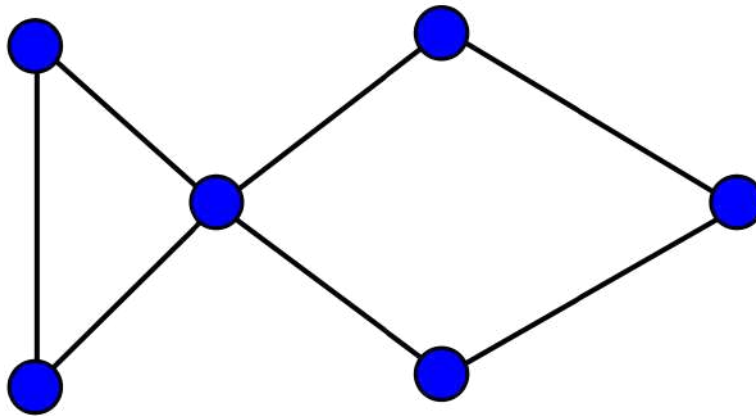


Figura 7: Grafo de ejemplo.
Fuente: Wikipedia.

En la figura 7, los círculos azules representan los nodos, miembros de V y la líneas entre ellos, las aristas, miembros de E .

3.1.1. Grado promedio

Cuando ya se tiene el fenómeno a estudiar representado en una red, una propiedad clave, y de las primeras que se revisan en ciencia de redes, es el grado de los nodos. Denotado con k_i , el grado del i -ésimo vértice miembro de V se determina mediante la cantidad de aristas

¹⁴En este trabajo, los conceptos red, nodo y conexión se usarán de manera sinonímica con grafo, vértice y arista respectivamente

en las que participa el nodo i . En el caso de la figura 7, el grado del nodo que se encuentra más a la derecha sería $k = 2$.

Teniendo estos valores, se puede extraer otra propiedad útil del grafo, el grado promedio $\langle k \rangle$, calculado mediante el cálculo clásico de media aritmética sobre los grados de todos los nodos:

$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^N k_i \quad (1)$$

3.1.2. Distribución de grados

También se puede derivar la, posiblemente, más importante propiedad de una red, la distribución de grados [Barabási, 2016]. Esta determina una probabilidad p_k de que un nodo aleatorio tenga grado k .

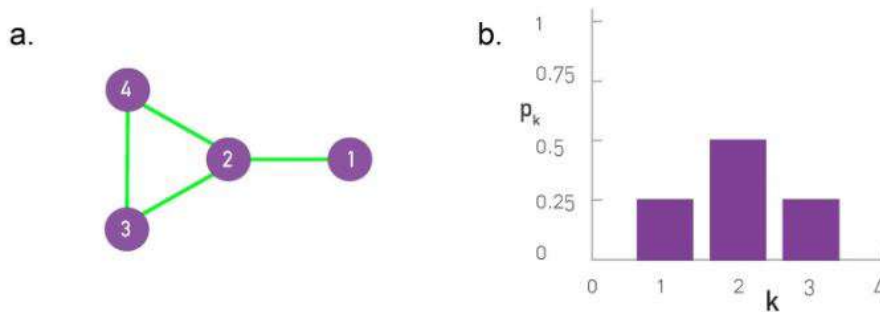


Figura 8: Distribución de grados de una red.
Fuente: [Barabási, 2016]

El gráfico **b** de la figura 8 permite visualizar la distribución de grados de la red presentada en la imagen **a** de la misma figura. El grado $k = 1$ tiene un cuarto de las probabilidades (0,25), ya que existe 1 de 4 nodos que tienen una sola conexión: el nodo 1.

3.1.3. Coeficiente de *clustering*

Manteniendo el enfoque hacia la topología de la red, la siguiente propiedad a definir es el coeficiente de *clustering*. En principio, esta propiedad mide, para cada nodo, “cuántas de mis conexiones se conocen entre ellas”. Esto es, la proporción en la que los vecinos ¹⁵ de un nodo están conectados entre sí [Barabási, 2016]. Esta métrica, para un nodo i con grado k_i ,

¹⁵Entiéndase vecino como los nodos conectados directamente a otro.

se define como:

$$C_i = \frac{2L_i}{k_i(k_i - 1)} \quad (2)$$

Donde L_i es la cantidad de conexiones entre los k_i vecinos de i . En la red **a** de la figura 8, el nodo 2 tiene un grado $k = 3$ y un $L_i = 1$ (conexión entre 3 y 4), por lo que su coeficiente de *clustering* C_i es $\frac{1}{3}$. Cabe notar que el valor de C_i fluctúa entre 0 y 1, representando la probabilidad de que dos vecinos del nodo estudiado tengan una conexión entre ellos.

Símil al grado, a este coeficiente también se le puede calcular un promedio, así, representando el *clustering* de toda la red. Se define de manera análoga:

$$\langle C \rangle = \frac{1}{N} \sum_{i=1}^N C_i \quad (3)$$

3.1.4. Asortatividad

Otra de las propiedades a estudiar es la asortatividad. Esta métrica, derivada del coeficiente de correlación de Pearson entre pares de nodos conectados [Newman, 2002] cuantifica la “preferencia” a conectarse a otros nodos similares. En la mayoría de los casos, que los vértices se consideren similares se refiere a que tengan grados de magnitudes semejantes. En otras palabras, la asortatividad mide si los nodos de grado alto, también llamados *hubs*, prefieren conectarse a otros con una cantidad parecida de conexiones. Análogamente, se incluye la inclinación de nodos de grado bajo a conectarse a otros con su misma condición. Conviene enfatizar que la disasortatividad se refiere a que los *hubs* prefieren conectarse a nodos de bajo grado [Barabási, 2016].

Antes de determinar el cálculo de la asortatividad, es necesario definir lo siguiente:

- Matriz de correlación de grados e_{ij} : esta matriz representa la distribución de probabilidad de encontrar nodos de grado i y j a los extremos de una conexión seleccionada aleatoriamente [Barabási, 2016]. Esta distribución cumple que:

$$\sum_{ij} e_{ij} = 1$$

- Fracción a_i de la distribución e_{ij} de aristas que comienzan con un nodo de grado i [Newman, 2003]. Factorizada de la siguiente forma:

$$\sum_j e_{ij} = a_i$$

- Fracción b_j de la distribución e_{ij} de aristas que terminan con un nodo de grado j [Newman, 2003]. Factorizada de la siguiente forma:

$$\sum_i e_{ij} = b_j$$

- Desviaciones estándar σ_a y σ_b ¹⁶ de las distribuciones a_i y b_j respectivamente.

Finalmente, el cálculo del coeficiente de asortatividad r [Newman, 2003]:

$$r = \frac{\sum_{ij} ij(e_{ij} - a_i b_j)}{\sigma_a \sigma_b} \quad (4)$$

El término se encuentra en el rango $-1 < r < 1$, donde $r = 1$ es asortatividad perfecta y $r = -1$ es disasortatividad perfecta. En el primer caso, la red es regular, todos los nodos tienen el mismo grado. El segundo, no es tan trivial como el primero. Aquí, la mayoría de conexiones existirán entre hojas (nodos de grado 1 o bajo) y vértices céntricos, o *hubs*.

3.1.5. Distancia promedio y Diámetro

También propiedades importantes de la topología de una red, la distancia promedio y el diámetro son otro par de métricas a estudiar. Cabe aclarar que, con distancia promedio, se busca decir camino más corto entre nodos promedio. Para esto, se aplica el reconocido algoritmo de Dijkstra [Dijkstra, 1959]¹⁷, el cual entrega las distancias más cortas entre cada par de nodos. Para los efectos de este trabajo, se calculará el promedio de estos valores por cada red.

Por su parte, el diámetro de una red es “la máxima distancia entre todos los caminos más cortos” [Lehman *et al.*, 2010]. En la práctica, se busca el mayor valor de los entregados por el algoritmo de Dijkstra.

¹⁶Se diferencian estas desviaciones cuando el grafo es dirigido. Para no dirigido, basta una de estas al cuadrado.

¹⁷La librería NetworkX, hasta la fecha, utiliza por defecto este algoritmo. Sin embargo, existen otras opciones.

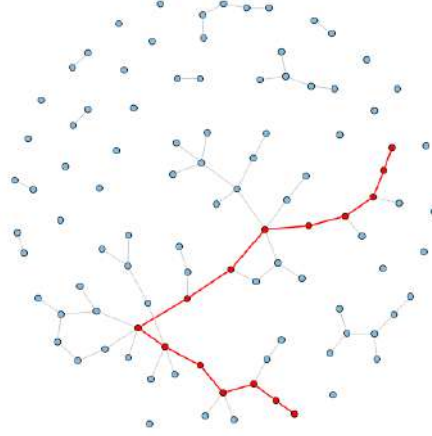


Figura 9: Ejemplo del diámetro de una red.
Fuente: [Franceschet, 1999]

3.1.6. Robustez

Por último, se revisarán umbrales de robustez. Específicamente, el límite, de nodos eliminados, en el que la red pierde su componente gigante ¹⁸ y el criterio *Molloy-Reed* para determinar si existe o no esta componente.

Siendo $\langle k \rangle$ y $\langle k^2 \rangle$ grado promedio y grado al cuadrado promedio respectivamente, el criterio *Molloy-Reed* κ se define como [Cohen et al., 2000]:

$$\kappa \equiv \frac{\langle k^2 \rangle}{\langle k \rangle} \quad (5)$$

El criterio tiene un punto crítico en 2, donde redes con $\kappa > 2$ tienen componente gigante y las que cumplen con $\kappa < 2$, su topología esta fragmentada en componentes menores desconectadas [Barabási, 2016].

Respecto al otro umbral mencionado, f_c determina la fracción de nodos que deben ser quitados de la red para que su componente gigante se fragmente en elementos desconexos. Su definición es la siguiente [Cohen et al., 2000, Barabási, 2016]:

$$f_c = 1 - \frac{1}{\frac{\langle k^2 \rangle}{\langle k \rangle} - 1} \quad (6)$$

¹⁸Componente conexas más grande de un grafo que tiene la propiedad de crecer proporcionalmente a la cantidad de nodos que se agreguen a la red [Bollobás, 2001].

3.2. Modelos de redes

En el origen de la disciplina, se tenía la creencia que el comportamiento de las redes que representaban fenómenos reales seguían un comportamiento aleatorio (siguiendo una red Erdős-Rényi, u otro modelo estocástico) o del tipo *small world* (explicados más adelante). Fueron A. Barabási junto a R. Albert, quienes, al modelar páginas web enlazadas, propusieron un modelo más acorde a la realidad: redes *scale-free* [Barabási y Albert, 1999]. No obstante, existen múltiples modelos de redes, tan variados como los fenómenos a los que representan. No obstante, es importante mencionar que los modelos aleatorios no quedaron obsoletos y existen modelos aún más simples que tienen utilidad. Por ejemplo, las mallas regulares¹⁹.

A continuación se presentaran los modelos de redes que se estudiarán en este trabajo:

3.2.1. Modelo Erdős-Rényi

Presentado originalmente alrededor del año 1960 [Erdős y Rényi, 1960], el modelo Erdős-Rényi (ER) genera grafos de manera aleatoria, siendo la aproximación más básica de una red y utilizada comúnmente como muestra de control en experimentos. Este modelo toma como parámetros la cantidad de nodos n a producir y la probabilidad p con la que crear una conexión entre estos. Otra opción es trabajar con la cantidad de conexiones M a producir, distribuyéndose de manera equiprobable entre los pares de nodos²⁰.

El proceso de generación consiste en partir con grafo de n vértices, para luego iterar sobre cada par de nodos. Bajo la probabilidad p , se decide si crear la conexión entre los nodos en cuestión. En el otro método se selecciona aleatoriamente un par de nodos disponible (sin conexión) y se crea un arista. Esto se repite M veces.

El resultado es una red con propiedad de *small world*²¹ y una distribución de grados binomial:

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \quad (7)$$

¹⁹Estas redes pueden utilizarse para modelar mapas y calles.

²⁰Esto asegura la cantidad de aristas deseadas a diferencia de la primera opción, que el valor esperado de aristas es $p \cdot n^2$ cuando $n \rightarrow \infty$.

²¹Se explicará en la subsección que lleva su mismo nombre.

Y un grado promedio de:

$$\langle k \rangle = p \cdot (n - 1) \quad (8)$$

La naturaleza aleatoria de este modelo permite que Erdős-Rényi se utilice de *benchmark* (muestra de control) para el estudio de otras redes. Al utilizar los mismos parámetros que el grafo estudiado (cantidad de nodos, aristas, etc), se puede desligar la topología del fenómeno e identificar si ésta tenía un efecto notable o no.

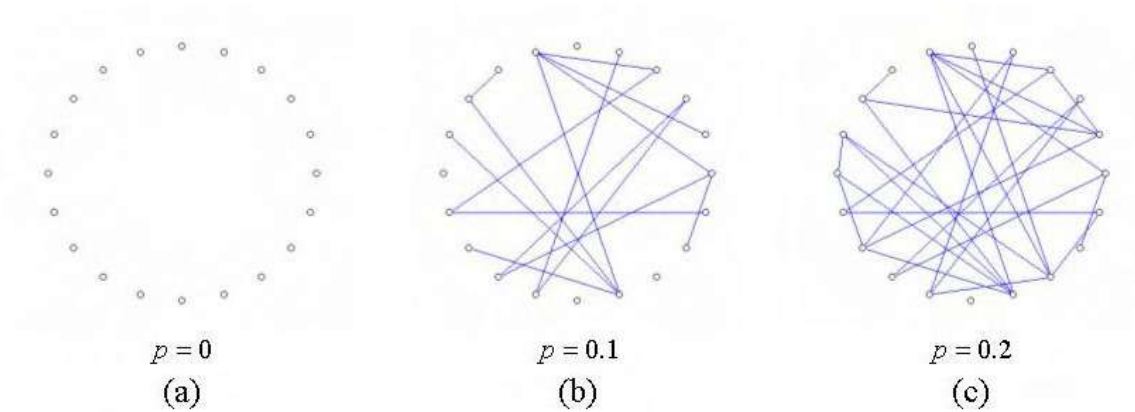


Figura 10: Ejemplo redes Erdős-Rényi.
Fuente: [Newth, 2006]

Teniendo $p = 0$, como se ve en la figura 10, se tiene un grafo sin conexiones; con $p = 1$, un grafo completamente conexo.

3.2.2. Modelo de bloques estocásticos (*Stochastic block model*)

También un modelo generativo aleatorio, el modelo de bloques estocásticos (SBM por sus siglas en inglés), propuesto por [Holland *et al.*, 1983], está diseñado para simular agrupaciones de nodos en comunidades. Este modelo recibe como parámetros la cantidad de nodos n a producir (opcional), un conjunto C de subconjuntos que particionen los nodos en “bloques” y una matriz (simétrica en el caso de grafos no dirigidos) P de probabilidades de tamaño $r \times r$, siendo r la cantidad de comunidades a generar. Cada casilla de la matriz p_{ij} determina la probabilidad de generación de una arista entre los pares de nodos (u, v) que cumplan con $u \in C_i$ y $v \in C_j$ ²².

La principal limitación de esta red es que el subgrafo de los miembros de una sola comunidad tiene el mismo comportamiento que una red aleatoria ER. Esto debido a que los nodos

²²Los algoritmos de generación se encuentran en los Anexos 7.1 y 7.2

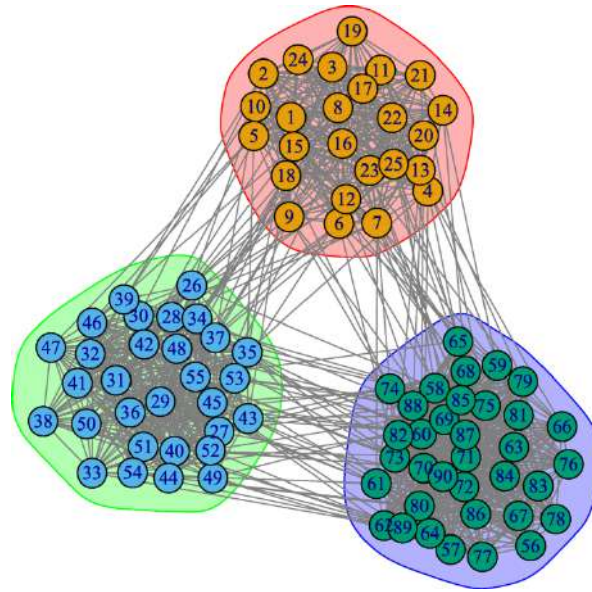


Figura 11: Ejemplo red SBM con tres bloques.
Fuente: [Lee y Wilkinson, 2019]

dentro de la misma comunidad siguen una sola probabilidad, las pertenecientes a la diagonal principal de la matriz P .

3.2.3. Modelo Watts-Strogatz (*Small world networks*)

A fines de los años 90, D. Watts y S. Strogatz (WS) notaron un fenómeno que se repetía en múltiples redes reales. El efecto de mundo pequeño puede ser encontrado en muchos ámbitos. Por ejemplo, en el contexto de red social dos personas que no se conocen están conectadas por una cadena de conocidos que es relativamente pequeña en comparación al cantidad de nodos en la red. Los grafos que poseen esta cualidad, tienen un “camino más corto”²³ promedio bajo ²⁴.

La generación de este modelo necesita tres parámetros: la cantidad de nodos n a producir, el grado k con el que comenzarán los nodos y la probabilidad p de “recablear” conexiones. El método propuesto por [Watts y Strogatz, 1998] parte por una topología regular de anillo ²⁵, donde cada nodo se conecta con los k (se suele considerar k par) más cercanos. Así se tiene un anillo donde los miembros son vecinos de los k vértices aledaños. Para conseguir el efecto *small world*, para cada arista que una un par de nodos (u, v) , se aplica la probabilidad p para redirigir la conexión a (u, w) , siendo w otro nodo elegido aleatoriamente. Esto genera

²³Camino, en este contexto, se refiere a una cadena de conexiones que une a dos nodos. El más corto, hace alusión al camino de menor longitud entre todos los disponibles.

²⁴Se considera bajo en la literatura cuando es equiparable al logaritmo de la cantidad de nodos de la red.

²⁵Los nodos se alinean en una circunferencia.

posibles conexiones entre “vecindades” en lados opuestos del anillo.

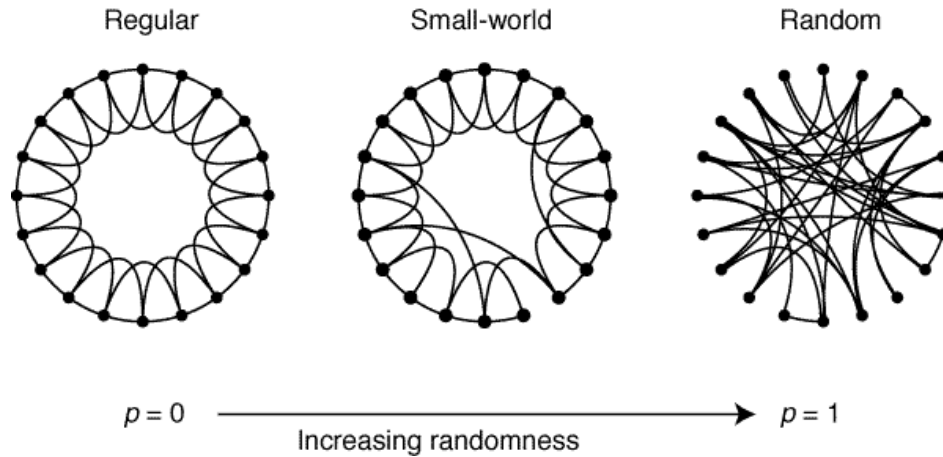


Figura 12: Ejemplo red Watts-Strogatz

Fuente: [Watts y Strogatz, 1998]

3.2.4. Modelo Barabási-Albert (*Scale-free networks*)

Repetidas a lo largo y ancho de múltiples áreas en las que existan fenómenos relacionales involucrados, las redes libres de escala, propuestas por [Barabási y Albert, 1999] (BA), son una de las principales razones de la creciente popularidad de la disciplina de ciencia de redes. El descubrimiento de “*hubs*” fue el principal hallazgo que llevó a la disciplina a entender el comportamiento de redes reales. Los *hubs* son nodos en la red que tienen **muchas** más conexiones que la mayoría del resto de miembros. Adicionalmente, suelen ser pocos los nodos que cumplen con esta propiedad. En la figura 13 (b), los *hubs* son los vértices sombreados. La mezcla de estas características hace que la distribución de grados del grafo siga una ley de potencia o *power law*.

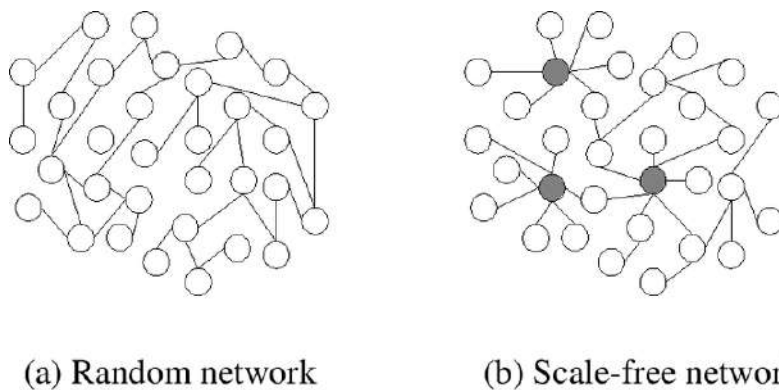


Figura 13: Comparación Red aleatoria versus Libre de escala.

Fuente: Wikimedia Commons.

La distribución *power law*, que merece una sección completa para describirla, se distingue por una alta frecuencia de grados bajos ²⁶ y baja frecuencia en grado **mucho** más altos. En la figura 14, el gráfico de la izquierda, se pueden ver la gran mayoría de vértices con pocos vecinos en la primera columna. Los *hubs* se encontrarían en las últimas. Matemáticamente, se puede expresar con la siguiente fórmula:

$$P(k) \sim k^{-\gamma} \quad (9)$$

El exponente γ suele estar en el rango $2 < \gamma < 3$. Esto consigue que su media aritmética este bien definida, mas su varianza diverge. Cabe mencionar que el parámetro γ dependerá de los datos de la red en cuestión.

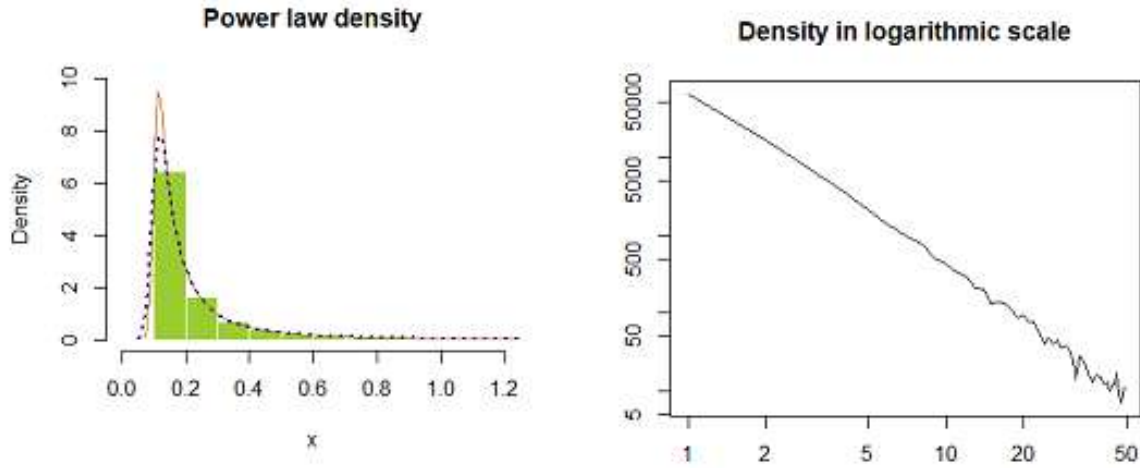


Figura 14: Ejemplo de Ley de potencia en escala normal (izquierda) y logarítmica (derecha).
Fuente: StackExchange.

El modelo propuesto en [Barabási y Albert, 1999], tomando un parámetro m_0 para indicar un tamaño inicial, crea una red completamente conectada de m_0 vértices. Luego agrega, hasta llegar a un total n , nuevos miembros bajo el siguiente criterio: Cada nodo nuevo se conecta con m miembros existentes, siendo $m < m_0$. Esta cantidad depende de la probabilidad p_i que tiene cada nodo i , la cual es directamente proporcional su grado “normalizado” al resto de grados de la red:

$$p_i = \frac{k_i}{\sum_j k_j}$$

²⁶Podría decirse grado bajo en relación al tamaño de la red, pero las redes se llaman libres de escala porque mantiene el comportamiento independiente de su magnitud.

Esto significa que los nodos con mayor grado tienen más probabilidad de recibir nuevas conexiones que el resto de los miembros, así acumulando cada vez más grado. También conocido como el fenómeno “*the rich get richer*”, este comportamiento es la base de la generación de *hubs* en la red y la característica definidora de una red libre de escala. Este modelo, en función del tamaño de la red que genere²⁷, hace que el exponente de su distribución *power law* resulte en $\gamma \sim 3$.

3.2.5. Mallas regulares

De las redes más simples que se revisarán en este trabajo, las mallas regulares, o *grid networks*, suelen usarse para representar topologías básicas relacionadas a figuras geométricas como cuadrados o triángulos²⁸.

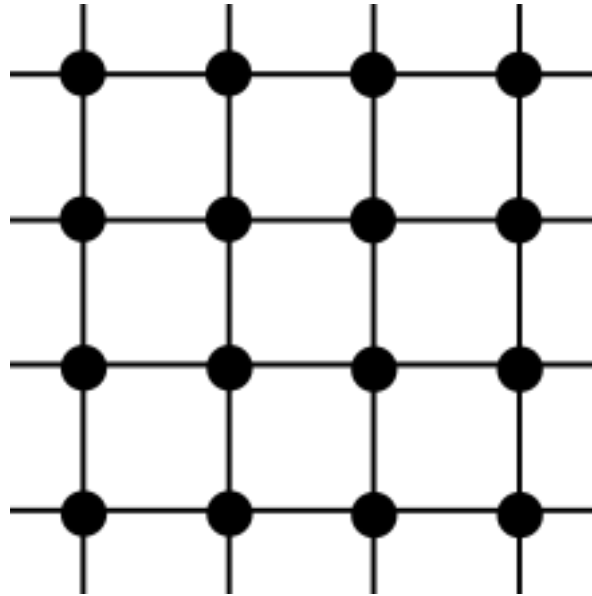


Figura 15: Ejemplo de malla regular cuadrada (*lattice*).

Fuente: Wikipedia.

Estas mallas se generan tomando un parámetro n para la cantidad de nodos (esto está restringido a la forma, por ejemplo, una malla cuadrada suele pedir un valor n tal que el resultado sea n^2 nodos) y otro k para el grado de los nodos. La construcción comienza con un vértice unido a k vecinos, los cuales se conectan, con nodos nuevos o ya presentes, hasta obtener el grado k . Se debe mantener todos los miembros del grafo, como se ve en la figura 15, con el mismo grado. Si es necesario, los nodos de las orillas de la malla “dan la vuelta” y conectan con la orilla opuesta, simulando una estructura toroidal. Esto asegura $\langle k \rangle = k$. Si no se realiza el “bucle”, el grado promedio cumple con $\langle k \rangle < k$.

²⁷Esto debido a la aleatoriedad del modelo.

²⁸Similar al teselado de un espacio 2D. También puede ser de mayor dimensión.

El principal propósito de considerar estas redes es su “estabilidad” respecto al grado ($\langle k \rangle = k$ y distribución de grados uniforme) y que no poseen el fenómeno *small world*.

3.3. Tipos de falta de información

La falta de información, como se discutió anteriormente, puede ser impredecible. Dependen de los métodos de extracción de los datos o simplemente del estado actual de la red debido a efectos de daño u otro agente externo. Es por esto que se definirán, a continuación, múltiples algoritmos, estocásticos y deterministas, que busquen simular los fenómenos a estudiar.

3.3.1. Eliminación de nodos

A partir de una red completa, elegidos bajo un cierto criterio, se van eliminando nodos de la red. Este proceso es iterativo y también se deshace de las conexiones que involucraban al vértice eliminado. Respecto al criterio de remoción, este puede ser aleatorio o dirigido. El caso dirigido tiene un subcriterio que determina qué nodo es el objetivo de la eliminación. Dichos criterios pueden ser por ejemplo: nodo con más grado, con más centralidad, entre otros. Conviene poner énfasis en que las eliminaciones dirigidas tienen un impacto mayor en las redes *scale-free*. Esto por la presencia de nodos *hubs*. Dadas suficientes remociones, la red podría perder su propiedad de ser libre de escala.

A continuación se presenta un pseudocódigo que representa el proceso²⁹:

Algoritmo 1 Eliminación de Nodos

```
1: Parámetros: (Grafo  $G$ , Entero  $n$ , Texto criterio)
2:
3:  $newG \leftarrow Graph(G)$  ▷ Copia el grafo de input.
4:  $V \leftarrow newG.nodes$ 
5:  $i \leftarrow 0$ 
6: while  $i < n$  do ▷  $n$  determina la cantidad de nodos a eliminar.
7:    $node \leftarrow selectNode(V, criterio)$  ▷ Función para elegir el nodo a eliminar.
8:    $newG.remove\_node(node)$  ▷  $V$  es modificado según la eliminación.
9:    $i \leftarrow i + 1$ 
10: end while
11:
12: return  $newG$ 
```

El parámetro n en el algoritmo 1 determina la cantidad de falta de información que se le

²⁹Los algoritmos presentados seguirán, a grosso modo, la nomenclatura de NetworkX y, por consecuencia, la de Python.

aplicará a la red. Cabe destacar que se debe cumplir que $n < |V|$. Este procedimiento puede simular, principalmente, daño en una red. Es más complejo imitar el *sampling* con este método, a menos que se afine minuciosamente el criterio de selección. No obstante, se presentarán mejores procedimientos para el caso de muestreo.

3.3.2. Eliminación de conexiones

Símil al método anterior, este procedimiento se encarga de remover aristas entre los nodos. Nuevamente existen los criterios aleatorio o dirigido. En el caso de dirigido, puede ser, por ejemplo, centralidad de conexiones ³⁰.

El pseudocódigo, para este método, es análogo al algoritmo 1, centrándose en las conexiones:

Algoritmo 2 Eliminación de Conexiones

```
1: Parámetros: (Grafo  $G$ , Entero  $n$ , Texto criterio)
2:
3:  $newG \leftarrow Graph(G)$  ▷ Copia el grafo de input.
4:  $E \leftarrow newG.edges$ 
5:  $i \leftarrow 0$ 
6: while  $i < n$  do ▷  $n$  determina la cantidad de conexiones a eliminar.
7:    $(u, v) \leftarrow selectEdge(E, criterio)$  ▷ Función para elegir el conexión a eliminar.
8:    $newG.remove\_edge(u, v)$  ▷ E es modificado según la eliminación.
9:    $i \leftarrow i + 1$ 
10: end while
11:
12: return  $newG$ 
```

Ahora, se tiene una técnica más apta para simular el muestreo de redes. Por ejemplo, se puede imitar una red que en la que se estudie el *Missing Link Problem*, discutido en el **Estado del Arte**, sección 2.3.3. Aún así, escapa al *sampling* del estilo *Ego-networks* o *Random Walks*.

3.3.3. Caminos Aleatorios

Esta vez, se simula directamente la técnica de *sampling*. Como base, se toma la red completamente conocida y se elige un nodo inicial aleatoriamente. Según un parámetro dado, n determina el largo del camino que se efectuará. En el vértice seleccionado se miran sus vecinos y, nuevamente, una extracción azarosa determina el siguiente paso del camino. El

³⁰La centralidad de conexiones se define por la cantidad de caminos-más-cortos que pasan por la arista en cuestión.

resultado de este proceso, como dice su nombre, genera un camino, una cadena unidimensional de conexiones. Para obtener un red más voluminosa con la que trabajar, se agregan las conexiones que los nuevos miembros tenían originalmente con nodos que ya ha sido seleccionados en el camino.

Algoritmo 3 Camino Aleatorio

```

1: Parámetros: (Grafo  $G$ , Entero  $n$ )
2:
3:  $newG \leftarrow Graph()$                                 ▷ Aquí quedará el resultado del camino.
4:  $rev \leftarrow list()$                                   ▷ Lista de nodos revisados.
5:  $i \leftarrow 0$ 
6:  $node \leftarrow random.choice(G.nodes)$                 ▷ Paso inicial.
7: while  $i < n$  do                                       ▷  $n$  determina el largo del camino.
8:    $newG.add\_node(node)$ 
9:    $rev.append(node)$ 
10:
11:   while  $node$  in  $rev$  do                                ▷ Selección del nuevo paso.
12:      $node \leftarrow random.choice(G.neighbors(node))$ 
13:   end while                                           ▷ Se puede optimizar haciendo pop.
14:    $i \leftarrow i + 1$ 
15: end while
16:
17: for  $node$  in  $newG.nodes()$  do
18:   for  $neigh$  in  $G.neighbors(u)$  do
19:     if  $neigh$  in  $newG.nodes$  then ▷ Agregar conexiones a nodos ya considerados.
20:        $newG.add\_edge(node, neigh)$ 
21:     end if                                           ▷ Conexiones repetidas no se consideran.
22:   end for
23: end for
24:
25: return  $newG$ 

```

Gracias a este algoritmo, el muestreo de una red generado a partir de caminos aleatorios está emulado. Además, se evita tener el camino unidimensional al agregar conexiones existentes entre nodos ya revisados. La efectividad de esta técnica para evitar la unidimensionalidad dependerá, al ser aleatorio, del largo del camino. Como añadido, para las redes reales, solían ocurrir ciclos cerrados pequeños que terminaban el recorrido prematuramente. Para solventar este caso, cuando el camino concluía antes de lo planeado, se retoma el recorrido en un nodo que ya pertenezca al camino. En otras palabras, se selecciona un nodo del recorrido y se crea una “rama” a partir del camino original.

3.3.4. Muestreo por Exploración, BFS y DFS

El último tipo de emulación de falta de información es el muestreo parecido al *ego-network* [Arnaboldi *et al.*, 2016]. Bastante similar a los caminos aleatorios, en este caso se retira el azar al momento de seleccionar el siguiente paso. Esto pues, se recorre la vecindad del vértice seleccionado bajo un criterio de anchura o profundidad. La aleatoriedad que no se quita es la elección del nodo inicial. Diferenciado de una *ego-network* como tal, es el hecho de iterar a un nuevo nodo y revisar su vecindad. Estas redes “ego”, en resumen, serían sólo la vecindad en sí.

Para los recorridos se utilizarán los algoritmos *Breadth First Search* (BFS) y *Depth First Search* (DFS) ³¹. De nuevo, esto generaría árboles, lo que disminuye drásticamente la cantidad de conexiones. Para esto, se repite el proceso de agregar las aristas entre nodos ya revisados. Los pseudocódigos de BFS y DFS con esta consideración se encuentran en los **Anexos 7.3 y 7.4** respectivamente.

Algoritmo 4 *Ego-Networks*

```
1: Parámetros: (Grafo  $G$ , Entero  $n$ )
2:
3:  $bfsG \leftarrow BFSMod(G, n)$ 
4:  $dfsG \leftarrow DFSMod(G, n)$ 
5:                                     ▷ Detalle de modificaciones en Anexos.
6: return ( $bfsG, dfsG$ )
```

La principal consideración de este último método es que entrega los casos extremos del *random-walk*. Junto a la técnica anterior, como resultado siempre se entrega subgrafos conexos del original. En cambio, los dos primeros métodos pueden llegar a retornar subgrafos disconexos.

³¹Estos ya tienen una implementación en NetworkX. Link: [NetworkX,]. De todas formas, se implementarán versiones modificadas.

CAPÍTULO 4

MARCO EXPERIMENTAL

Esta sección está dedicada a la definición de todo el proceso de experimentación que se llevará a cabo. Esto incluye la exposición de las muestras a utilizar, siendo estas las redes a estudiar, el procedimiento con el que se tratará el grupo de ejemplares y la extracción de resultados. Además, las condiciones del ambiente en el que se realizó la experiencia serán dadas a conocer.

4.1. Redes de muestra

4.1.1. Redes reales

Las redes reales, como lo dice su nombre, derivan de fenómenos en un contexto existente ya representados en un grafo. Estas provienen de múltiples ámbitos, desde redes de infraestructura hasta redes “sociales” animales. Sus detalles, tales como, propiedades principales, tipo de archivo ³² y sus fuentes serán precisados en el Anexo 7.5. En esta sección se mostrarán nombres, breves descripciones y mediciones de magnitud para tener un entendimiento de las muestras con las que se trabajará.

Antes de presentar el listado, se debe comentar sobre una particularidad que tienen muchas redes reales. Según cómo se haya creado la red que represente el fenómeno real que existe detrás, suele verse la existencia de una gran componente gigante y múltiples pequeños subgrafos desconectados de esta. Para los experimentos venideros, se extraerá solamente la componente gigante. Más comentarios sobre la implicancia de esta decisión en la sección de **Conclusiones**. Asimismo, algunas redes eran dirigidas en su versión original. Por motivos similares a la extracción de la componente gigante, se trabajó con la versión no dirigida³³. Esto produce que el número de conexiones disminuya, dado que si existe conexión en cualquiera de las dos direcciones, el resultado considera solo una arista. Se hará alusión a ambas cantidades, la original y la no dirigida.

La fuente específica de cada red quedará en los **Anexos 7.5**, pero el sitio centralizado donde se extrajeron es el siguiente: [Rossi y Ahmed, 2015], Link página web Network Repository: [Rossi y Ahmed,].

³²Los tipos de archivo para almacenar redes suelen ser *.net* o similares. La librería NetworkX posee los métodos para leerlos.

³³Se considera que existe una conexión cuando, en su versión dirigida, existe en cualquier dirección.

1. Red de Interacción de Proteínas.

- Descripción: Red de interacción de las proteínas del hongo de la levadura.
- Cantidad de nodos: 1,8k. Cantidad de conexiones: 4,4k. No dirigida: 2,2K.
- Número de iteraciones: 15.

2. Red de captura de Campaños.

- Descripción: Red de roedores mamíferos capturados en una misma trampa.
- Cantidad de nodos: 1,7k. Cantidad de conexiones: 4,7k. No dirigida: 4,6k.
- Número de iteraciones: 15.

3. Red Eléctrica en Estados Unidos.

- Descripción: Red que representa la infraestructura de estaciones eléctricas en Estados Unidos.
- Cantidad de nodos: 4,9k. Cantidad de conexiones: 6,6k. No dirigida: 6,6k.
- Número de iteraciones: 15.

4. Red Social Anybeat.

- Descripción: Red entre usuarios en la cerrada red social Anybeat.
- Cantidad de nodos: 12,6k. Cantidad de conexiones: 67,1k. No dirigida: 49,1k.
- Número de iteraciones: 5.

5. Red Social Google Plus

- Descripción: Red entre usuarios en la cerrada red social Google Plus.
- Cantidad de nodos: 23,6k. Cantidad de conexiones: 39,2k. No dirigida: 39,1K.
- Número de iteraciones: 5.

Ya que solo se tiene una red como muestra al experimentar en estos casos, en pos de reducir el error aleatorio, se repetirán las aplicaciones de los algoritmos. La cantidad específica es descrita por el presentado número de iteraciones.

4.1.2. Redes generadas

A diferencia de las redes presentadas en la sección anterior, estas serán generadas aleatoriamente según los cinco modelos propuestos en el **Marco Conceptual**, sección 3.2. Para cada modelo serán generadas 150 redes. El propósito de este número es intentar minimizar el efecto que tendrá el error aleatorio sobre la experiencia balanceado contra los tiempos de ejecución.

Concretamente, las redes a generar son las siguientes:

- 150 redes ER de 3000 nodos. Grado promedio aproximado: 8.
- 150 redes SBM de 3000 nodos. Bloques y probabilidades generadas con los algoritmos 5 y 6 respectivamente. Intra-comunidad $p = 10/3000$, Inter-comunidad $p = 4/3000$.
- 150 redes WS de 3000 nodos. Parámetros $k = 8$ y $p = 0,1$.
- 150 redes BA de 3000 nodos. Parámetro $m = 3$.
- 150 redes *Grid* de 54×54 nodos. Esto para aproximar, por lo bajo, la cantidad de 3000.

Los valores escogidos buscan generar redes de magnitudes y densidades similares, con el propósito de resaltar el efecto de la topología más que el de la cantidad de nodos y conexiones.

4.2. Procedimiento y Extracción

Con las redes definidas en la sección anterior, se puede poner en marcha el experimento. El desarrollo fue realizado en la plataforma Google Colab, versión gratuita. Esto conllevó trabajar con 12,7 GB de memoria RAM. Cabe destacar que, al momento del desarrollo, se trabajó con Python 3.9 y NetworkX 3.0. Las redes reales son cargadas a través de archivos. Las generadas se construyen vía los métodos implementados en la librería NetworkX. Luego, los algoritmos presentados en el **Marco Conceptual** fueron implementados en el mismo *notebook* de desarrollo. Así también, el flujo de trabajo del experimento.

Dicho flujo comienza en las cargas de las redes, ya sea por lectura o generación. Ya con todas, se puede pasar a la modificación de éstas usando los algoritmos ya discutidos. Previo a la alteración, se extraen las propiedades de las redes originales y se escriben en un archivo. Análogamente, luego de la modificación se repite la extracción para las redes resultantes.

En el caso de las generadas, para cada modelo, el flujo de trabajo teórico es el siguiente:

1. Generación de una red según el modelo especificado.
2. Escritura en archivo de propiedades originales.
3. Aplicación de Algoritmo 1.
 - Iteración a través de distintos parámetros del algoritmo.³⁴
 - Escritura en archivo de propiedades modificadas.

³⁴Con parámetro del algoritmo se hace alusión a principal n que toman como variable los algoritmos propuestos. Por ejemplo, el largo del recorrido en los procedimientos 3 y 4

4. Repetición del mismo proceso con el resto de algoritmos...
5. Paso a la siguiente red.

Existirán casos donde se aplican menos parámetros en las iteraciones. Por ejemplo, el algoritmo 1 usará cuatro parámetros y el 2 sólo dos. La razón de esto fue que la eliminación de conexiones necesita un número mayor para hacer notar su impacto. Esta superior magnitud impacta a los tiempos de ejecución.

El caso de las redes reales es similar, aunque necesita unas modificaciones:

1. Lectura de archivo que contenga la red.
2. Escritura en archivo de propiedades originales.
3. Ciclo iterador que repite los experimentos.
 - Aplicación de Algoritmo 1.
 - Iteración a través de distintos parámetros del algoritmo.
 - Escritura en archivo de propiedades modificadas.
 - Repetición del mismo proceso con el resto de algoritmos...
4. Paso a la siguiente red.

A continuación se presenta un pequeño ejemplo con la ejecución de dos Algoritmos con un parámetro específico a una red. En la figura 16 se muestra un ejemplo de red generada ³⁵ con el modelo ER ($n = 25$ y $p = 0,2$). Esta es la instancia donde se almacenan las propiedades originales en un archivo. A continuación en la figura 17 se presenta una demostración de la aplicación del Algoritmo 1 de eliminación de nodos tomando como parámetro n igual a 10 nodos con criterio aleatorio. Otro ejemplo, en la figura 18 se aplicó el Algoritmo 2, eliminando 25 conexiones, también de manera aleatoria. ³⁶

³⁵La magnitud de este ejemplo es pequeña, 25 nodos. Esto en pos de mejorar la visibilidad en esta instancia.

³⁶Los esquemas fueron dibujados bajo el criterio del método *draw()* de NetworkX. Por esto no se logra la misma posición de los nodos sobrevivientes.

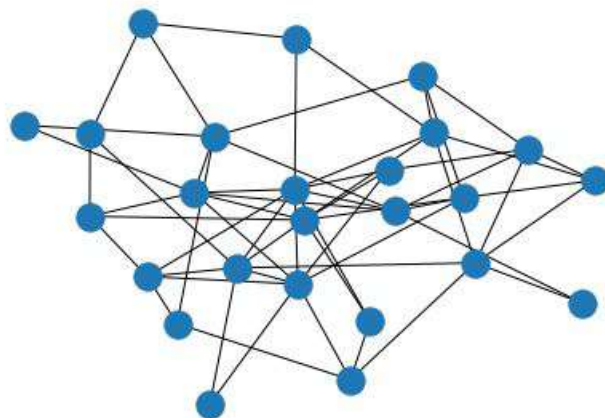


Figura 16: Ejemplo de red generada con modelo ER.
Fuente: Elaboración propia.

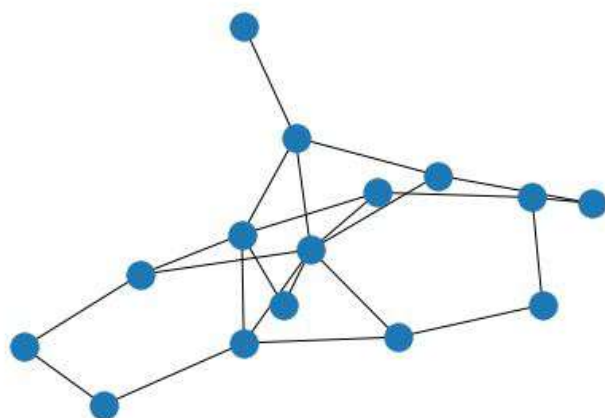


Figura 17: Ejemplo de red alterada con el Algoritmo 1.
Fuente: Elaboración propia.

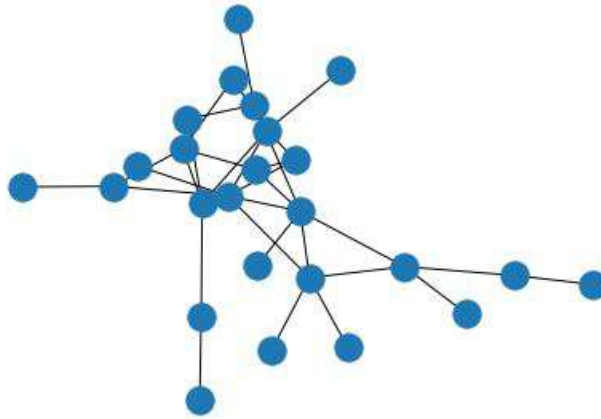


Figura 18: Ejemplo de red alterada con el Algoritmo 2.
Fuente: Elaboración propia.

Luego de aplicar los cuatro Algoritmos a todas las redes, reales y generadas, se dispondrá de dos grupos principales: las redes originales y las redes alteradas. Cabe destacar que las alteradas serán más numerosas, ya que se debe aplicar cuatro procedimientos resultando en cuatro nuevos grafos por cada uno de los 150 miembros por modelo. La extracción de propiedades es lo siguiente para estos grupos. Particularmente, el cálculo de las estadísticas propuestas en el **Marco Conceptual**, sección 3.1.

Dado que existe aleatoriedad en los métodos, existe la posibilidad de generar varias alteraciones por cada cruce Red-Algoritmo para disminuir errores aleatorios. Se menciona en el flujo de trabajo como “distintos parámetros del algoritmo”. Sin embargo, esta realización conlleva un crecimiento geométrico en la cantidad de muestras. Es por esto que se decidió mantener solo una red alterada por cada cruce Red-Algoritmo y, en pos de reducir varianzas en los resultados, aumentar el número de redes y sus tamaños.

A continuación, en la tabla 2, un ejemplo reducido es presentado. Los valores presentes corresponden a la redes ilustradas en las figuras 16, 17 y 18 respectivamente.

Red 1	Original	Alg 1 ($n = 10$)	Alg 2 ($n = 25$)
$\langle k \rangle$	4,72	3,06	2,72
$\langle C \rangle$	0,192...	0,073...	0,048...

Tabla 2: Ejemplo de extracción de propiedades.
Fuente: Elaboración propia.

En este pequeño ejemplo, se puede dar una reducida muestra del objetivo que busca cumplir este trabajo. Revisando la tabla 2, para ambas propiedades, aplicando ambos Algoritmos, se puede ver una clara disminución. Los dos algoritmos reducen la cantidad de conexiones, directamente en el caso 2 e indirectamente en el caso 1. Como las propiedades presentes

son consecuencia inmediata del número de aristas del grafo, es posible ver el efecto que tuvo la falta de completitud de la red. Cabe destacar que el impacto del Algoritmo 2 es mayor, puesto que este se enfoca en eliminar conexiones. La pequeña conclusión extraída de este ejemplo de “juguete” es lo que se busca nutrir y expandir con el experimento propuesto. Buscar y definir patrones de comportamiento es su principal objetivo. Además de una potencial metodología para extraerlos en caso de tratar con algún modelo no visto en este experimento.

CAPÍTULO 5

RESULTADOS

Antes de pasar a los resultados de la experiencia, en las tablas 3 y 4 serán presentados los tiempos de ejecución que tomaron los flujos de trabajo. La numeración de las redes reales esta en base al listado en el que se presentaron en la sección 4.1.1 del **Marco Experimental**.

Tipo de Red	Tiempo de Ejecución
ER	03 h 32 min 38 s
SBM	01 h 43 min 04 s
WS	01 h 52 min 06 s
BA	01 h 58 min 06 s
Grid	01 h 28 min 14 s
Total	10 h 34 min 08 s

Tabla 3: Tiempos de ejecución promedio para redes generadas.
Fuente: Elaboración propia.

Red	Tiempo de Ejecución	Número de Iteraciones
Red 1	00 h 01 min 30 s	15
Red 2	00 h 02 min 44 s	15
Red 3	00 h 21 min 20 s	15
Red 4	01 h 53 min 26 s	5
Red 5	04 h 02 min 46 s	5
Total	06 h 21 min 46 s	

Tabla 4: Tiempos de ejecución promedio para redes reales.
Fuente: Elaboración propia.

Lo más relevante a rescatar de estas tablas, y el motivo por el que se presentan, es la magnitud de algunos tiempos de ejecución que tomaron los flujos de trabajo de cada red, y más importante, de cada tipo de red en la tabla 3. Este tiempo se debe a la repetición de distintos algoritmos sobre todas las redes, tales como, la generación misma, en el caso de la segunda tabla, y el cálculo de propiedades. Por su parte, en la redes reales, las de mayor tamaño, la dicha extracción de métricas es la parte que más tiempo toma. Los largos tiempos de ejecución fueron un gran limitante para el desarrollo de esta experiencia.

Pasando a los resultados de los experimentos, en primer lugar, serán presentados los valores de las propiedades extraídas de las redes originales. Esto con el propósito de establecer el número que le corresponde a una recolección completa de información. Junto a los valores, una visualización de distribuciones, cuando esta exista, se adjuntará a cada red/tipo de red.

Finalmente, se ilustrará una comparación que logre esclarecer el efecto que tuvo la falta de información a estos valores.

5.1. Redes Generadas Originales

Para todos los modelos, las tablas que presentarán los resultados tendrán los valores promedio de cada propiedad a lo largo de todas las redes. Estas métricas serán ilustradas en líneas de tendencia tipo histograma (estimado de densidad o *kde* ³⁷) para facilitar la visualización de su distribución en el caso de grado y coef. de *clustering*. Cabe destacar que estos gráficos abarcan las distribuciones de todas las redes unidas en una. Caso que no se repite con el resto de métricas dado que es un valor por red. Para estas propiedades, se presentarán gráficos de violín. Estas imágenes al completo, por tamaño, se dejarán en los **Anexos**, desde el 7.6 al 7.10. Por lo que, en esta sección se mostrarán sólo distribución de grado y de *clustering*.

³⁷Al ser una estimación de tendencia, la distribución puede presentar valores negativos cuando no debería ser posible, en especial bajo el efecto de “suavizado” que se aplicó.

5.1.1. Modelo ER

Modelo ER	Valor Promedio
Grado Promedio	7,99
Coef. Cluster Promedio	0,00262
Asortatividad	-0,000377
Diámetro	7,086
Distancia Promedio	4,084
f_c	0,874

Tabla 5: Valores promedio de las redes originales ER.

Fuente: Elaboración propia.

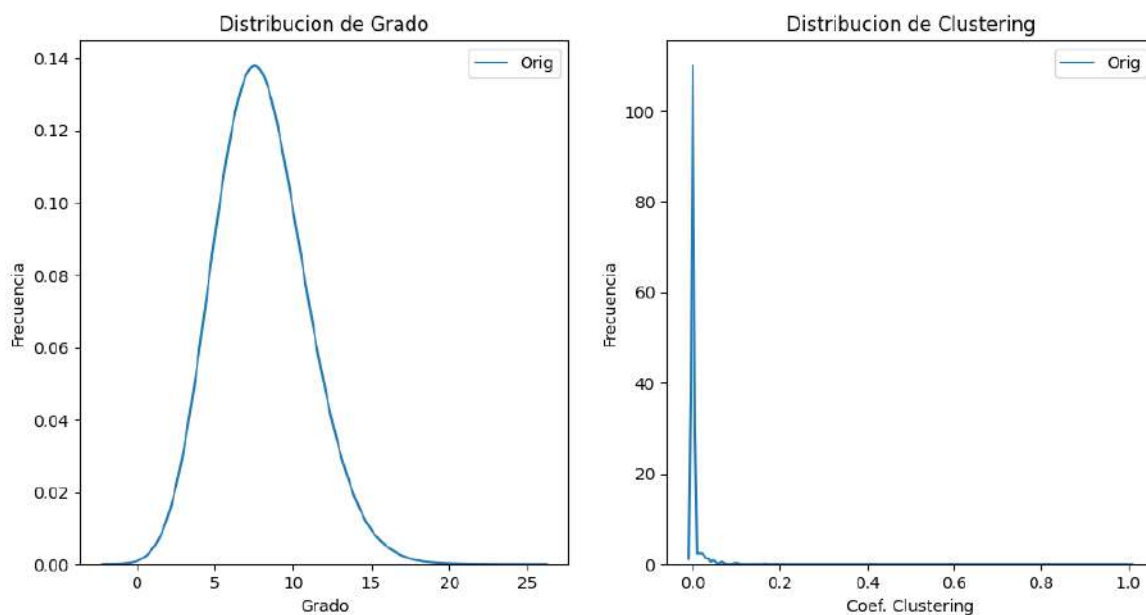


Figura 19: Distribuciones de resultados para redes ER originales.

Fuente: Elaboración propia.

5.1.2. Modelo SBM

Modelo SBM	Valor Promedio
Grado Promedio	5,056
Coef. Cluster Promedio	0,00164
Asortatividad	0,00174
Diámetro	10,246
Distancia Promedio	5,114
f_c	0,802

Tabla 6: Valores promedio de las redes originales SBM.

Fuente: Elaboración propia.

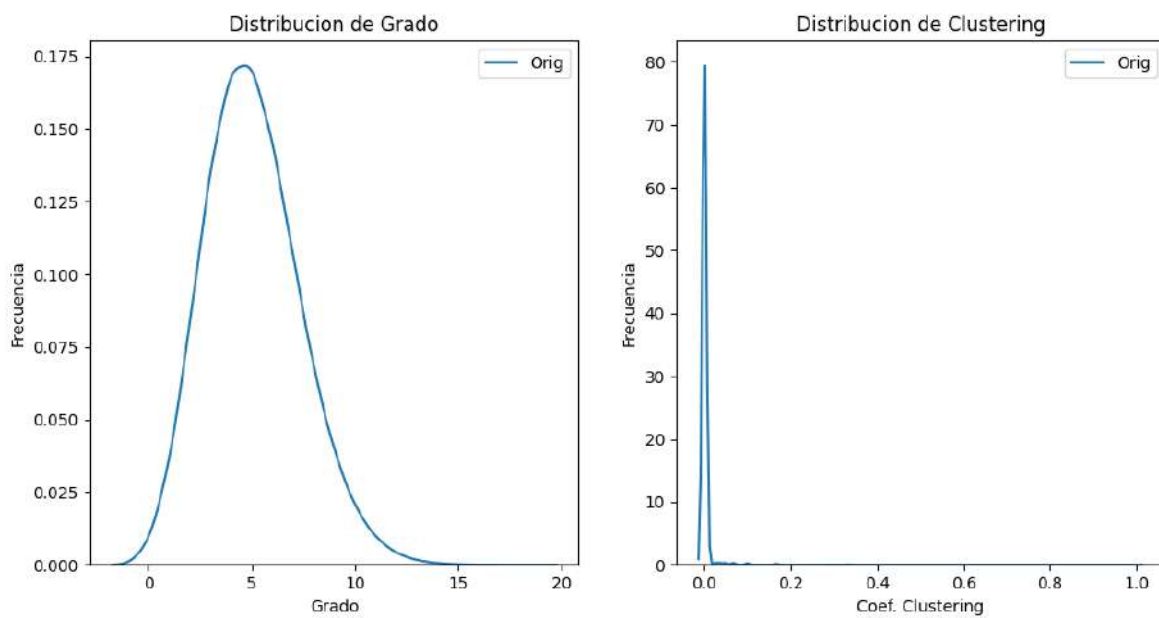


Figura 20: Distribuciones de resultados para redes SBM originales.

Fuente: Elaboración propia.

5.1.3. Modelo WS

Modelo WS	Valor Promedio
Grado Promedio	8,00
Coef. Cluster Promedio	0,474
Asortatividad	-0,0118
Diámetro	10,033
Distancia Promedio	6,0042
f_c	0,859

Tabla 7: Valores promedio de las redes originales WS.
Fuente: Elaboración propia.

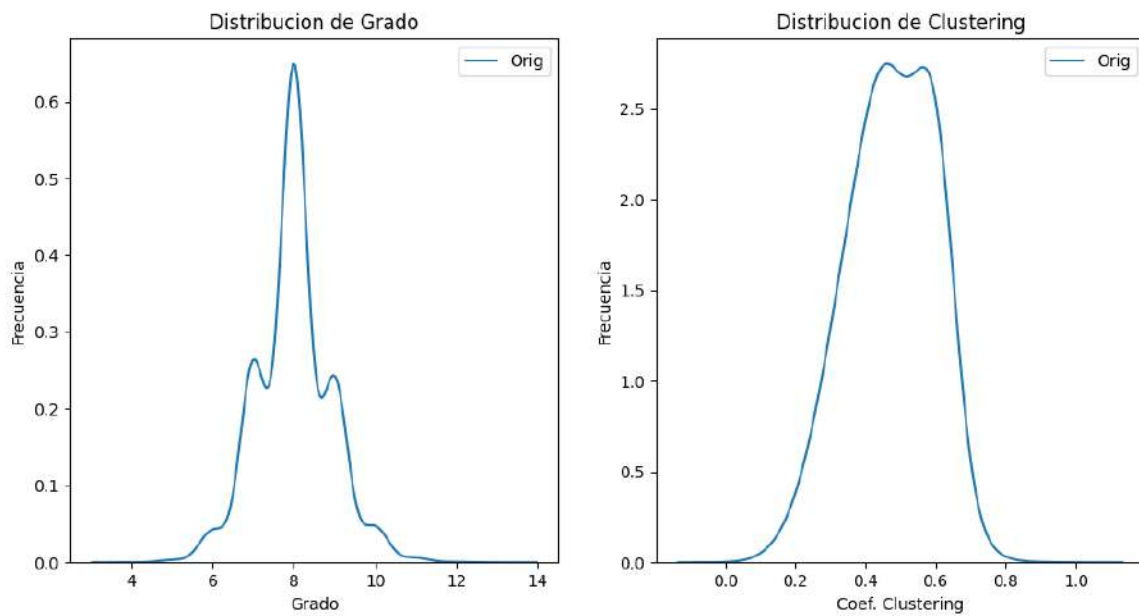


Figura 21: Distribuciones de resultados para redes WS originales.
Fuente: Elaboración propia.

5.1.4. Modelo BA

Modelo BA	Valor Promedio
Grado Promedio	5,994
Coef. Cluster Promedio	0,0133
Asortatividad	-0,0554
Diámetro	6,36
Distancia Promedio	3,882
f_c	0,933

Tabla 8: Valores promedio de las redes originales BA.

Fuente: Elaboración propia.

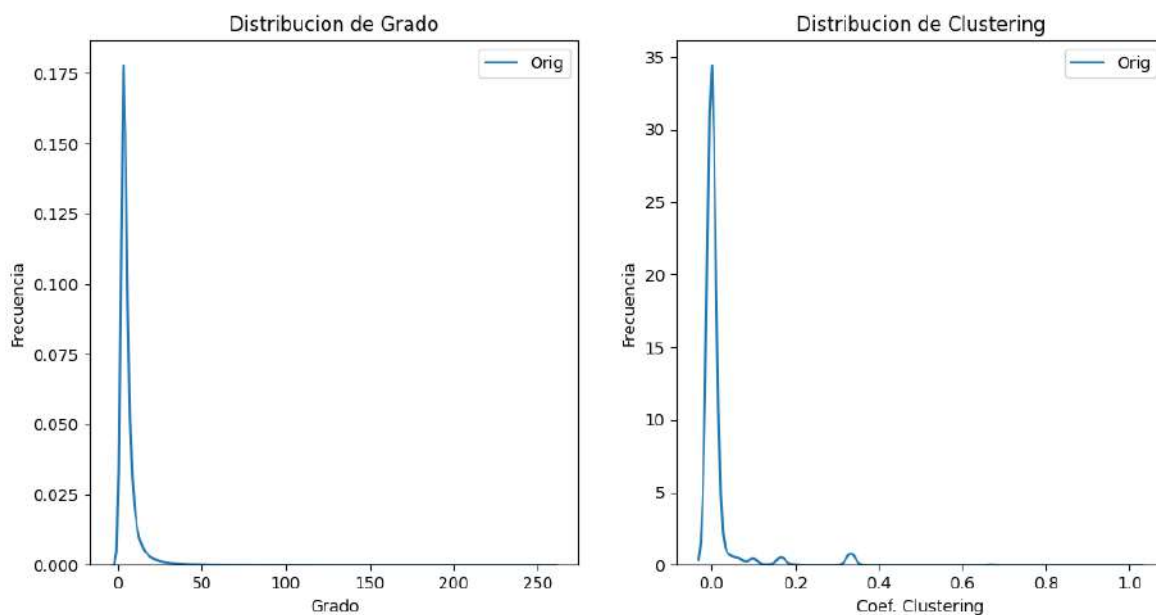


Figura 22: Distribuciones de resultados para redes BA originales.

Fuente: Elaboración propia.

5.1.5. Modelo Grid

Modelo Grid	Valor Promedio
Grado Promedio	4,00
Coef. Cluster Promedio	0,00
Asortatividad	—
Diámetro	54
Distancia Promedio	27,009
f_c	0,666

Tabla 9: Valores promedio de las redes originales Grid.

Fuente: Elaboración propia.

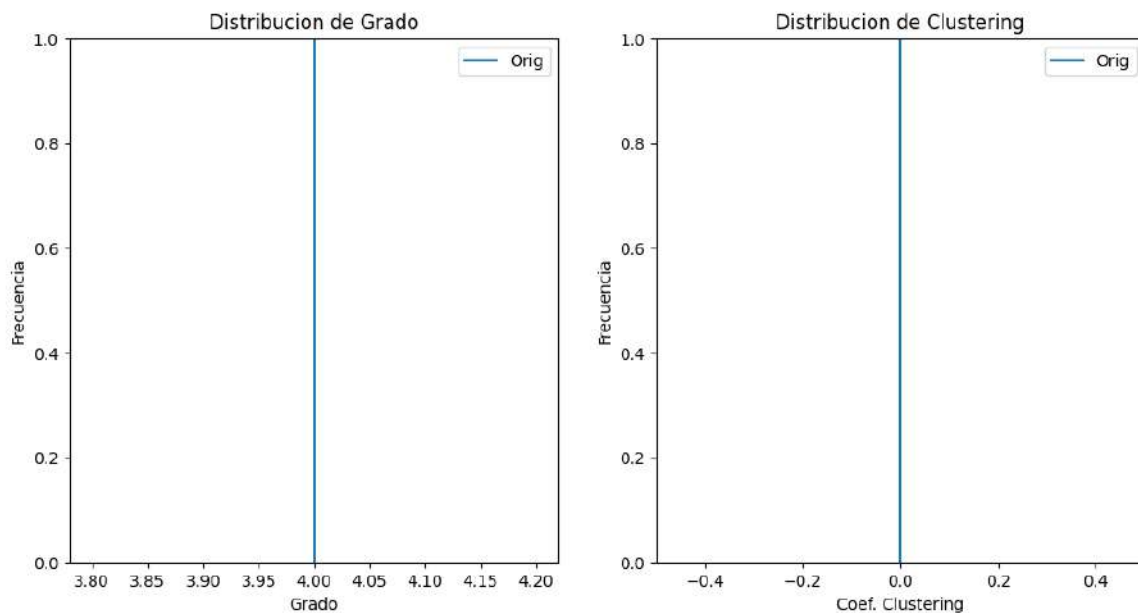


Figura 23: Distribuciones de resultados para redes Grid originales.

Fuente: Elaboración propia.

5.2. Redes Reales Originales

En este caso, al ser redes únicas, las reales presentarán los valores promedios de propiedades individuales³⁸ y el valor único de la red en el caso del resto de métricas. Asimismo, se presentarán las distribuciones con el mismo método *kde*.

Redes Reales	Grado prom	Coef. Cluster prom	Asortatividad	Diámetro	Distancia prom	f_c
Red 1	2,733	0,0708	-0,2075	19	6,812	0,839
Red 2	5,656	0,472	0,0132	15	6,536	0,887
Red 3	2,669	0,0801	0,00345	46	18,989	0,651
Red 4	7,770	0,203	-0,123	10	3,171	0,997
Red 5	3,318	0,174	-0,388	8	4,032	0,997

Tabla 10: Valores de las propiedades de las redes reales originales.

Fuente: Elaboración propia.

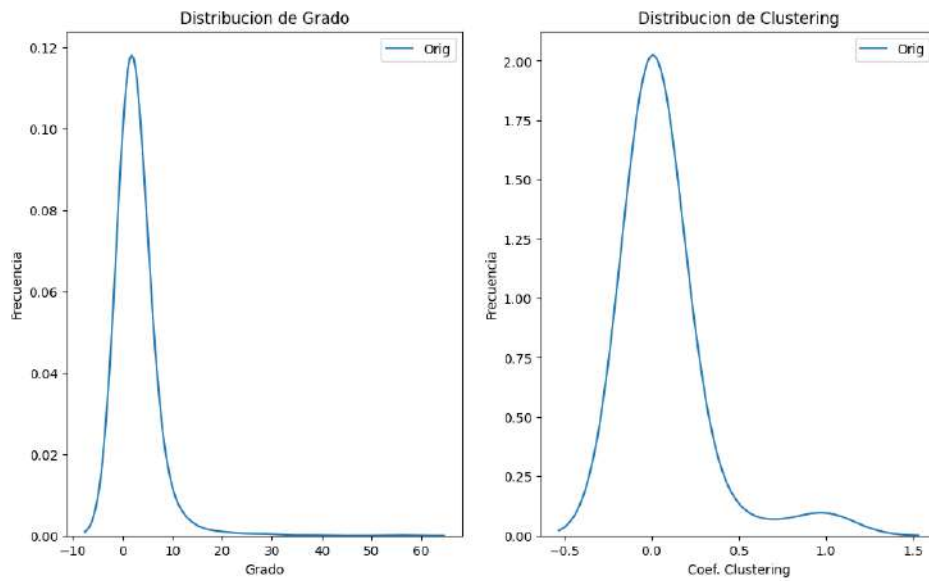


Figura 24: Distribuciones de resultados para red real 1.

Fuente: Elaboración propia.

³⁸Propiedad individual para cada nodo.

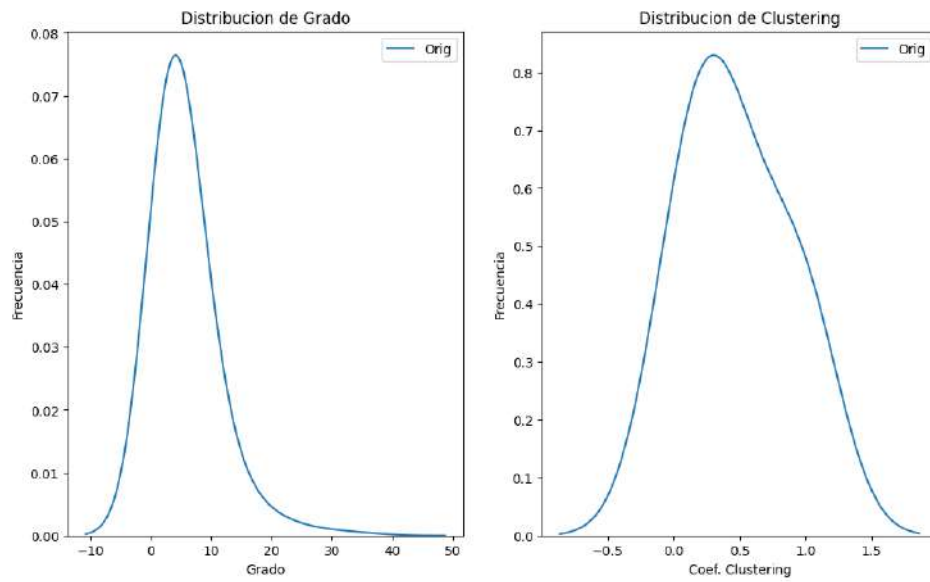


Figura 25: Distribuciones de resultados para red real 2.
Fuente: Elaboración propia.

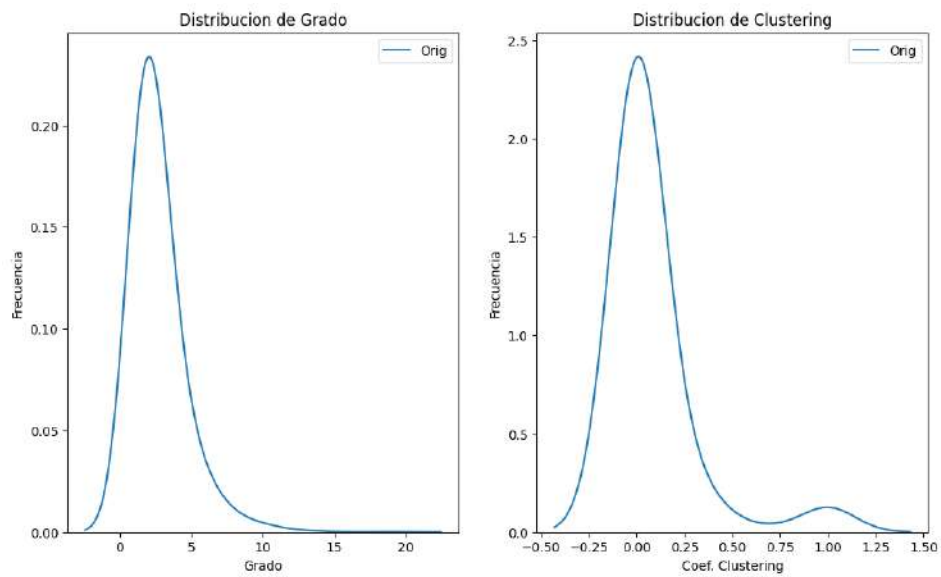


Figura 26: Distribuciones de resultados para red real 3.
Fuente: Elaboración propia.

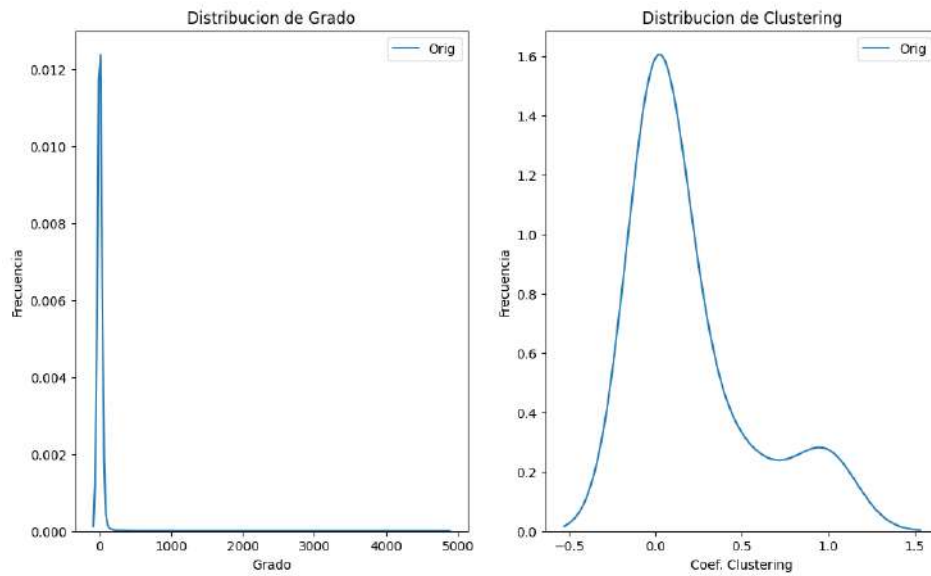


Figura 27: Distribuciones de resultados para red real 4.
Fuente: Elaboración propia.

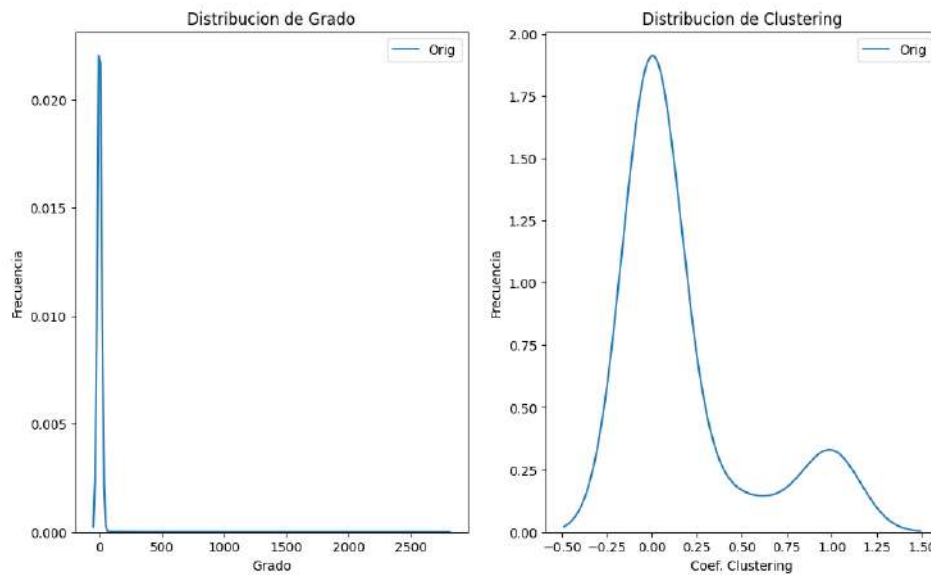


Figura 28: Distribuciones de resultados para red real 5.
Fuente: Elaboración propia.

5.3. Comparación Original - Alterado

5.3.1. Redes Generadas

Para visualizar la comparación de las métricas de redes originales versus alteradas se presentarán múltiples figuras con seis gráficos, donde cada uno de estos representará una propiedad en particular. Con el mismo formato a las imágenes en los **Anexos** del 7.6 al 7.10, los gráficos, en orden izquierda - derecha, arriba - abajo, muestran distribución de grado, de *clustering*, valores de asortatividad, de diámetro, de camino más corto promedio y de robustez (valor f_c). Cada color representa un valor del parámetro principal del algoritmo. Por ejemplo, para el algoritmo 1, eliminación de nodos, el número representa el parámetro n de cantidad de nodos a eliminar. Nuevamente, por temas de espacio en este escrito, quedarán, en esta sección, solo las figuras del algoritmo 1 aplicado a cada tipo de red. El resto de algoritmos se desplegarán en los **Anexos** del 7.11 al 7.15.

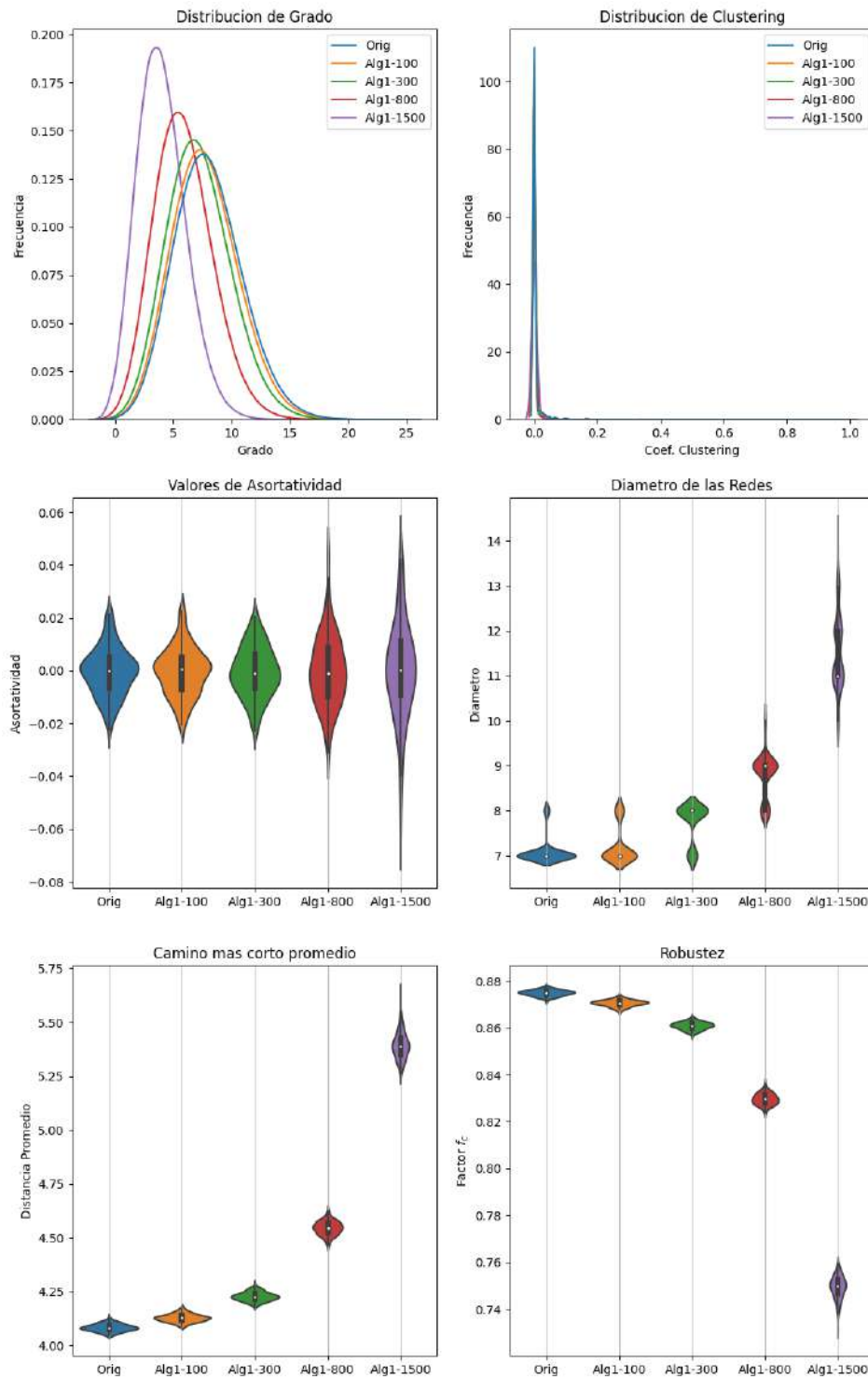


Figura 29: Comparación de resultados algoritmo 1 para redes ER.
Fuente: Elaboración propia.

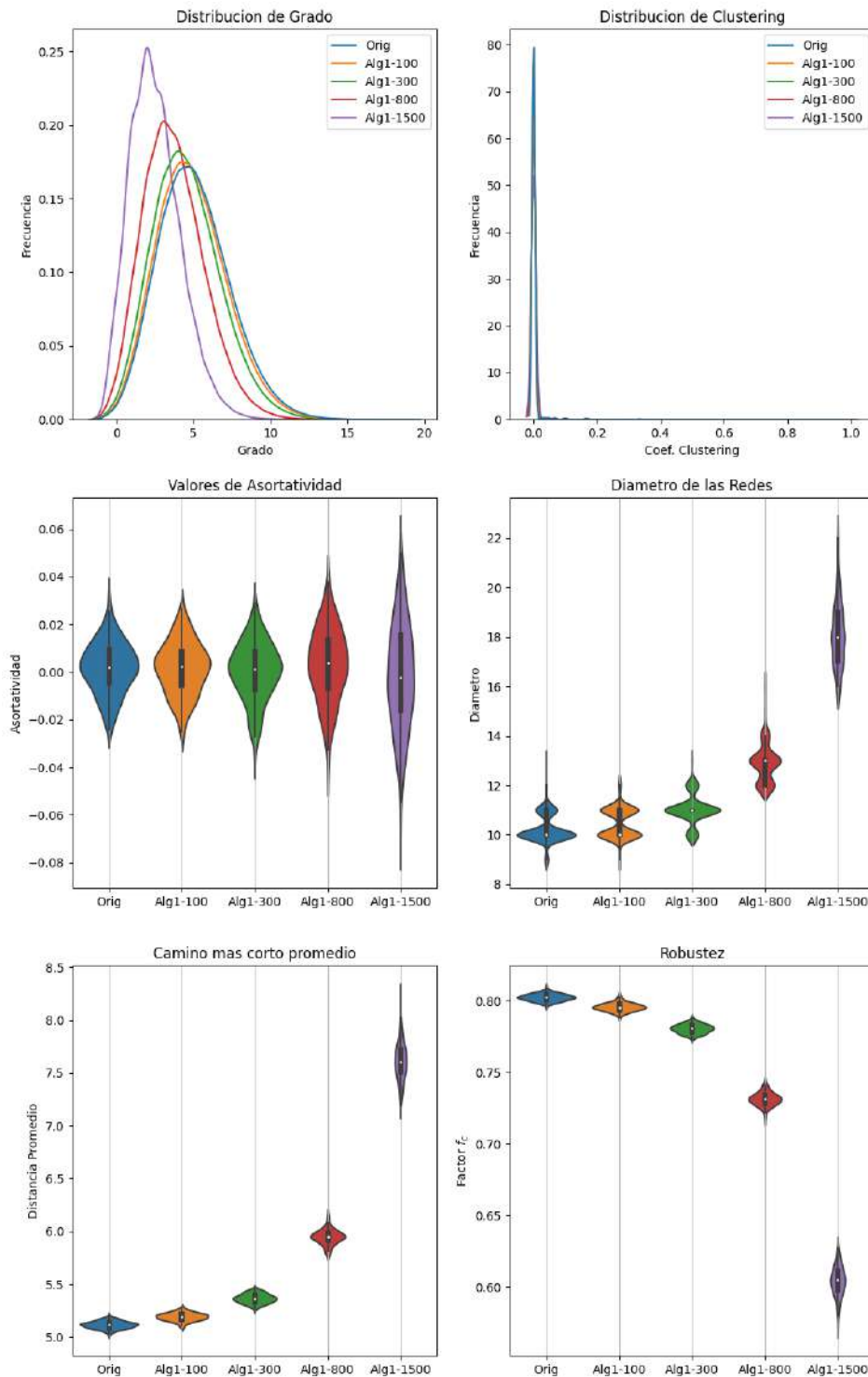


Figura 30: Comparación de resultados algoritmo 1 para redes SBM.

Fuente: Elaboración propia.

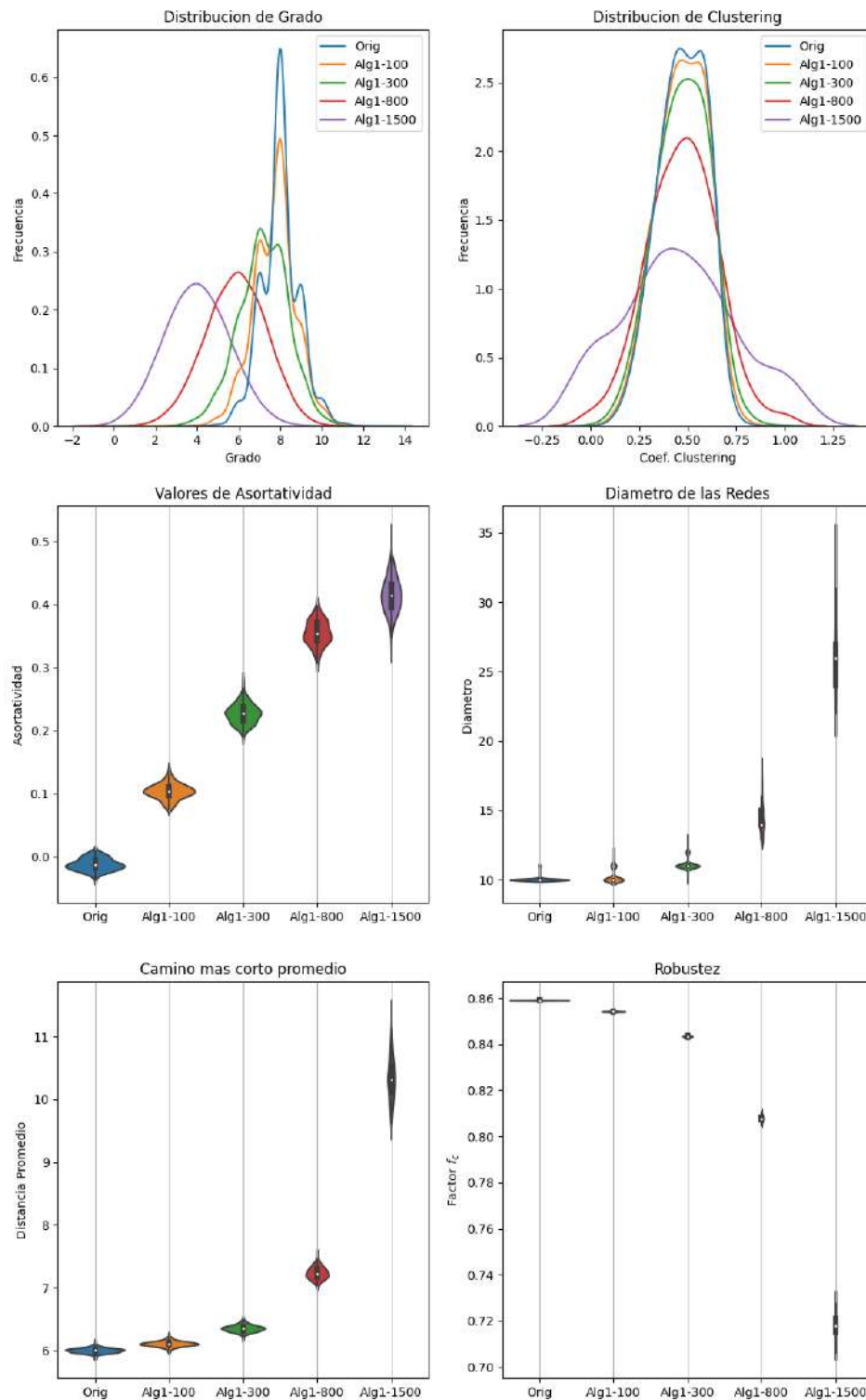


Figura 31: Comparación de resultados algoritmo 1 para redes WS.
Fuente: Elaboración propia.

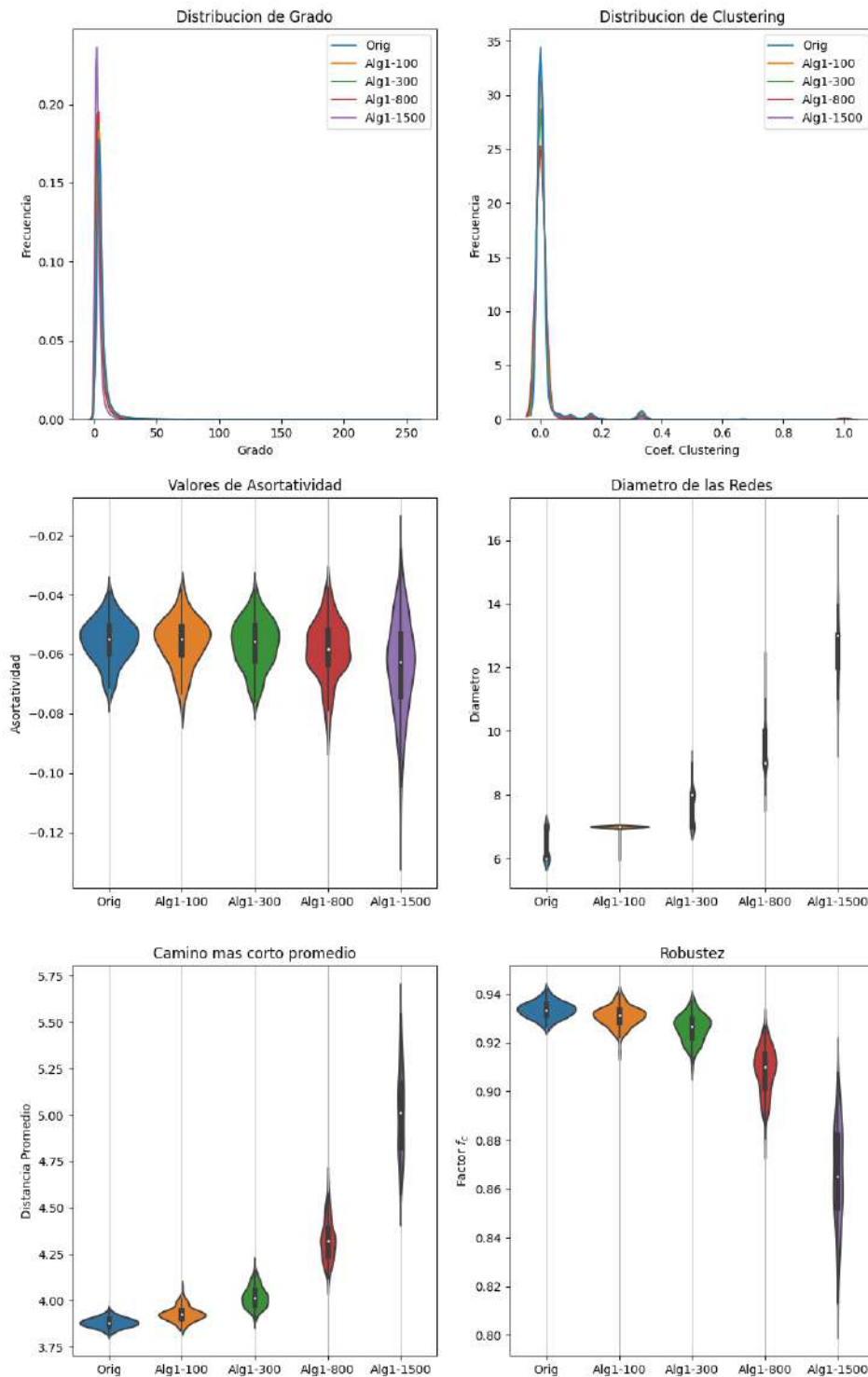


Figura 32: Comparación de resultados algoritmo 1 para redes BA.
Fuente: Elaboración propia.

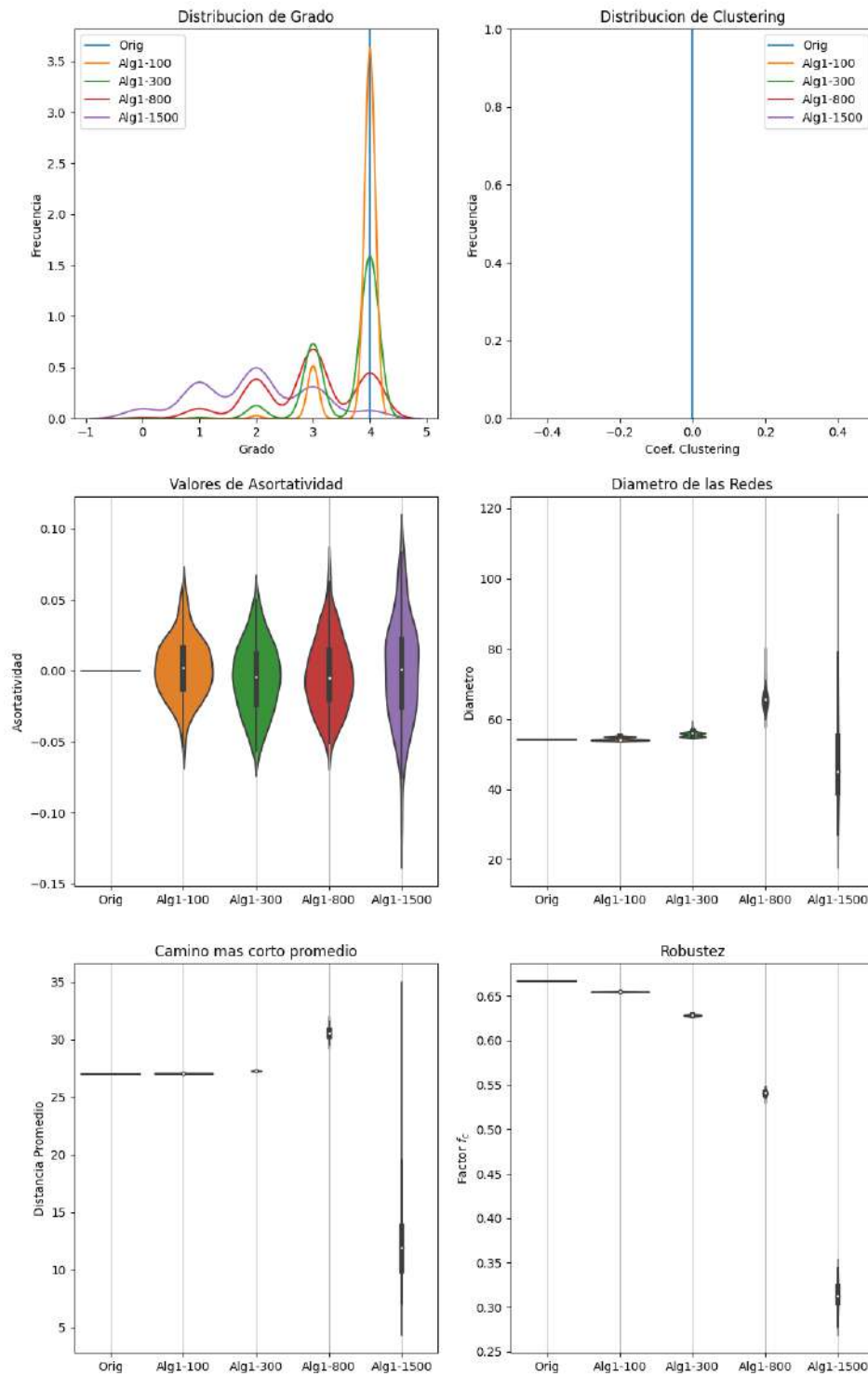


Figura 33: Comparación de resultados algoritmo 1 para redes Grid.
Fuente: Elaboración propia.

5.3.2. Redes Reales

No diferente de la sección anterior, a continuación se ilustrarán las comparaciones de red real original versus alteradas con sus distintos parámetros. Esto para el algoritmo 1. El resto de figuras quedarán en los **Anexos** 7.16 al 7.20.

Es importante mencionar que los valores del parámetro principal de cada algoritmo es un valor específico para cada red. Para las redes generadas, todas compartieron los mismos n , ya que también tuvieron en común la cantidad de nodos³⁹. En cambio, para las reales se buscó mantener proporciones similares, a las generadas, respecto a los valores de los parámetros y la cantidad de nodos. Para ejemplificar, en el algoritmo 1 la variable n va desde $1/30$ de la cantidad total de nodos hasta $1/2$ independiente de la red que se trabaje.

³⁹Para el caso de Grid, el valor fue aproximado.

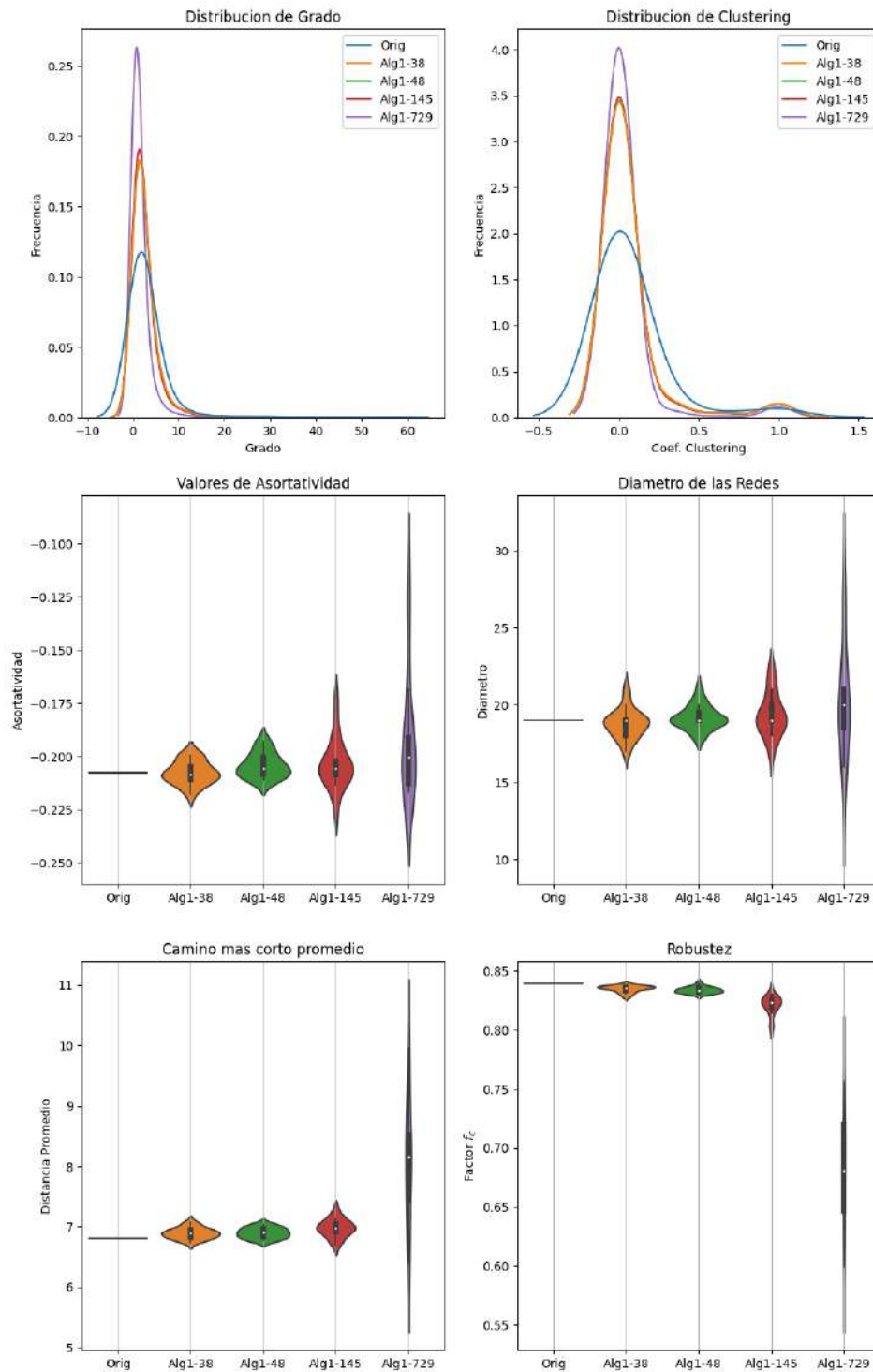


Figura 34: Comparación de resultados algoritmo 1 para red 1.
Fuente: Elaboración propia.

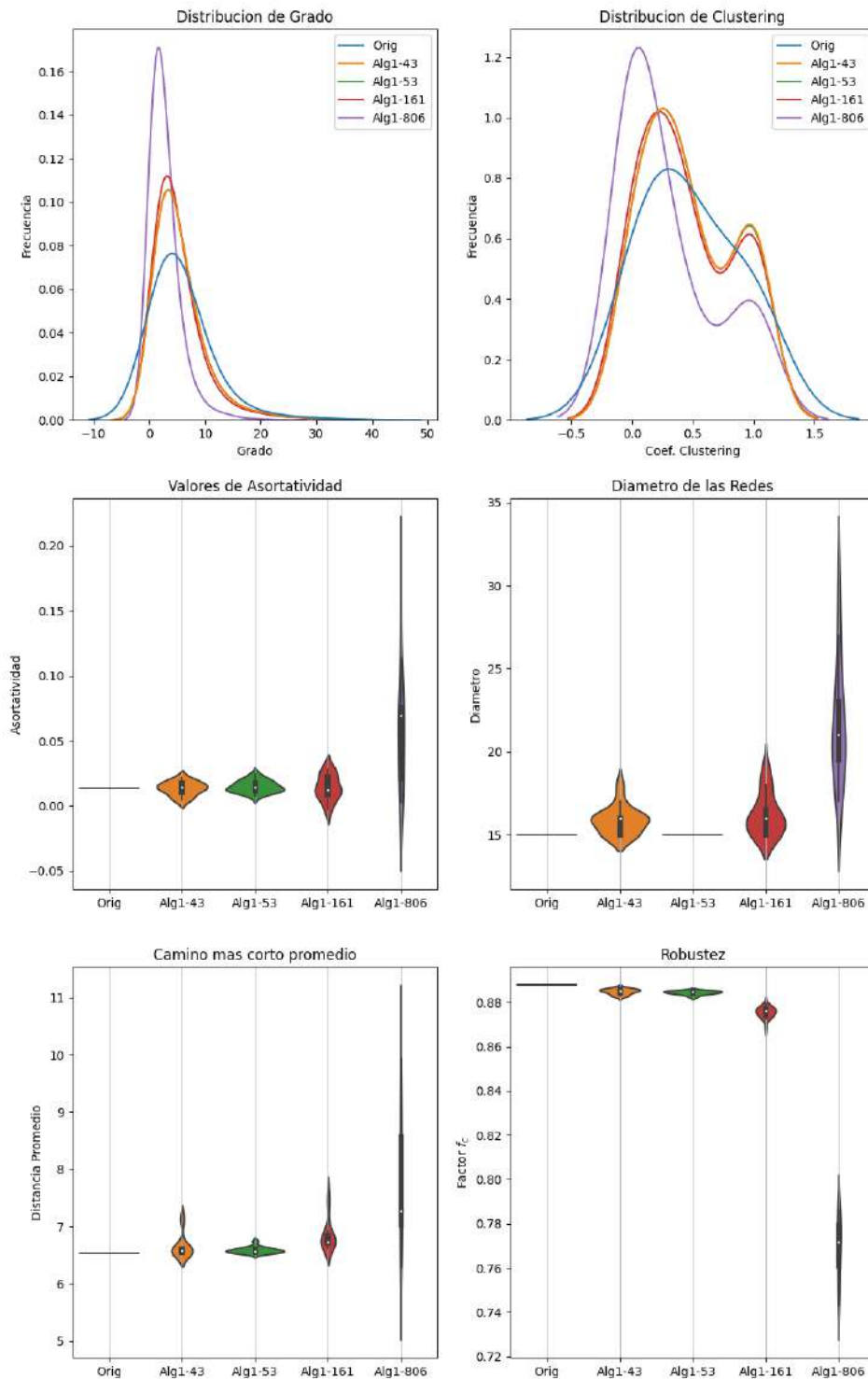


Figura 35: Comparación de resultados algoritmo 1 para red 2.
Fuente: Elaboración propia.

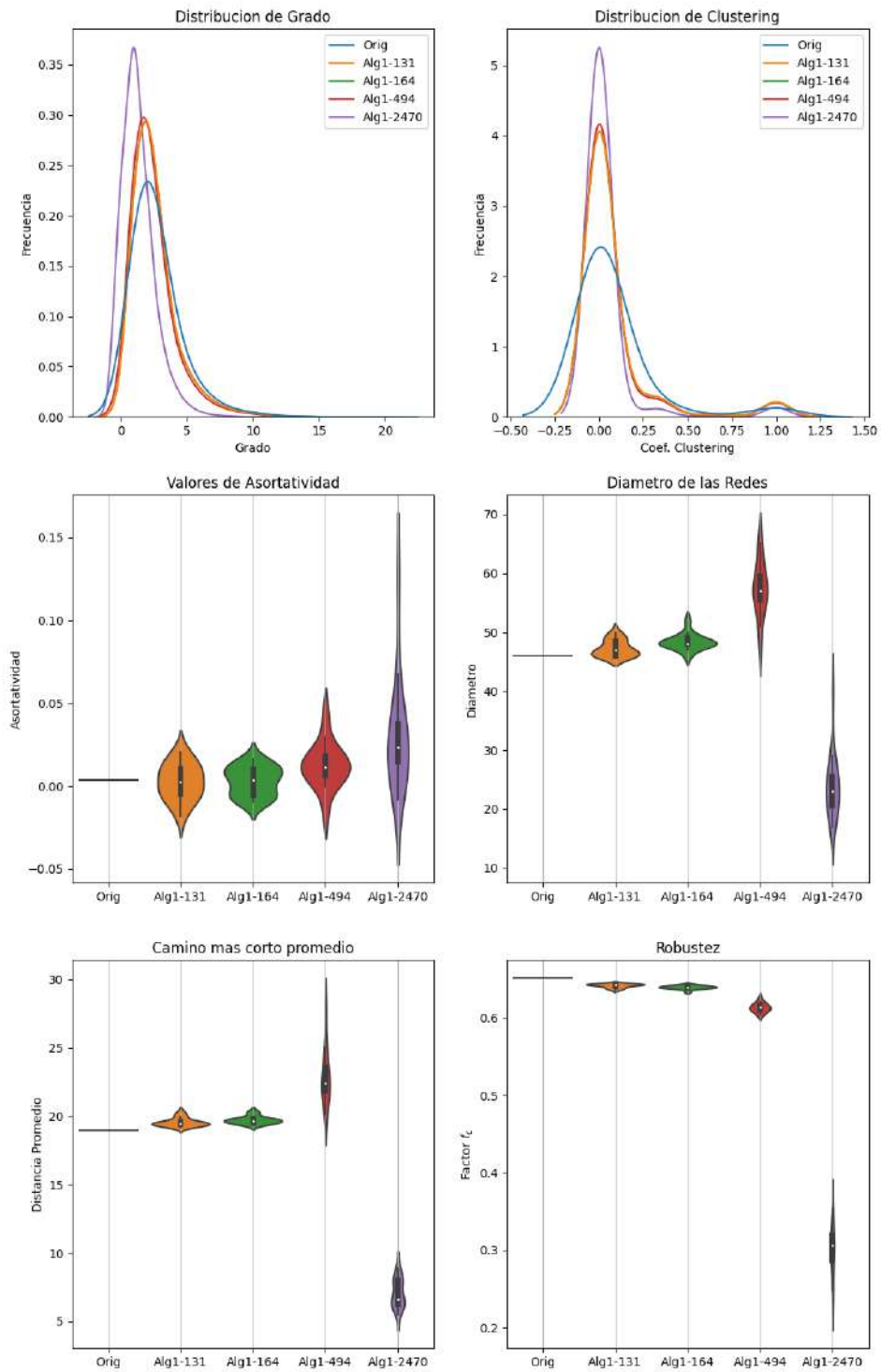


Figura 36: Comparación de resultados algoritmo 1 para red 3.
Fuente: Elaboración propia.

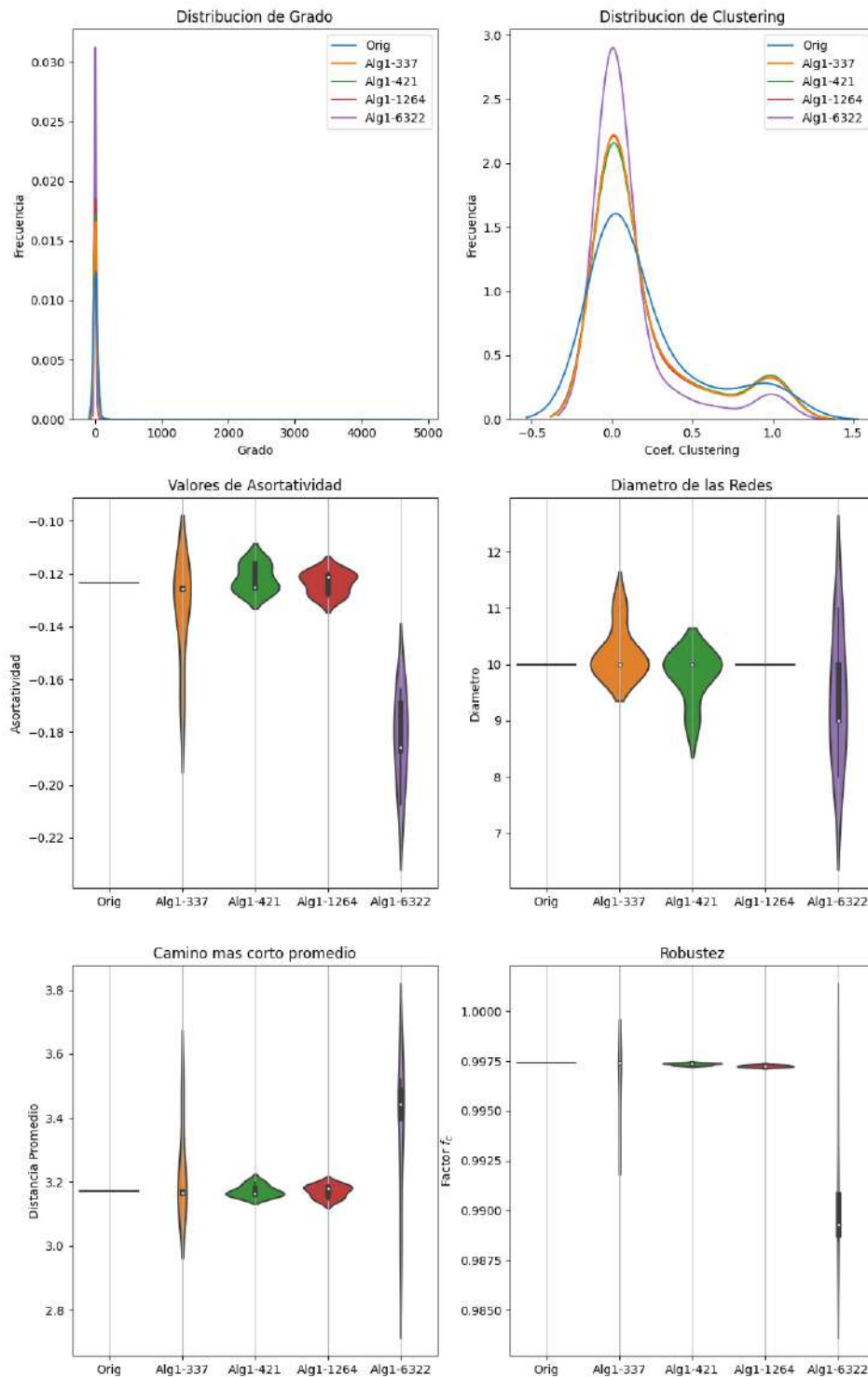


Figura 37: Comparación de resultados algoritmo 1 para red 4.
Fuente: Elaboración propia.

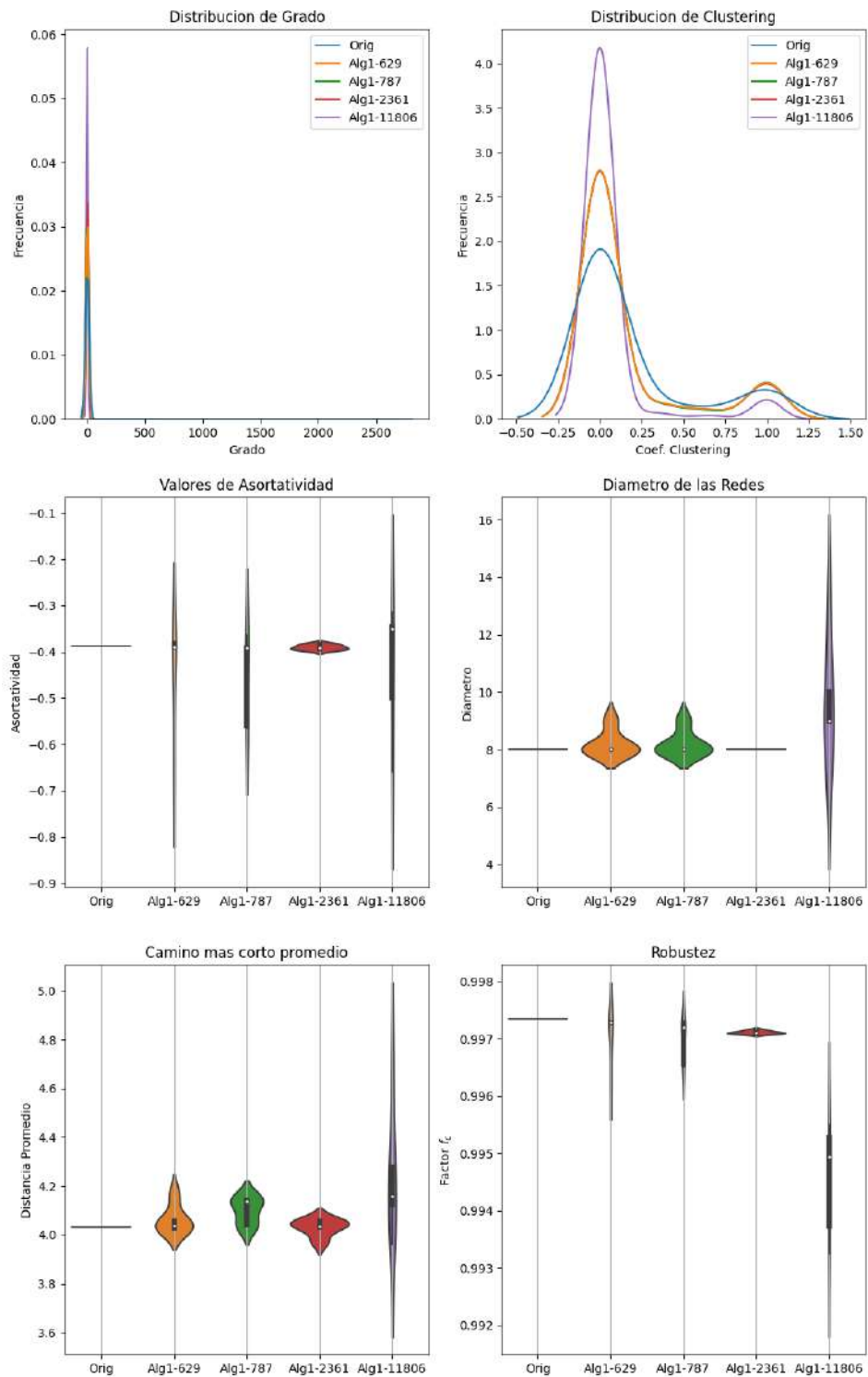


Figura 38: Comparación de resultados algoritmo 1 para red 5.
Fuente: Elaboración propia.

El análisis de estos resultados se desarrollará en la siguiente sección, **Discusión**. Cuando se evidencien las afirmaciones mencionando una figura, esta será referenciada, esté en esta sección o en los **Anexos**.

CAPÍTULO 6

DISCUSIÓN

Esta sección se utilizará para discutir los fenómenos más relevantes que se producen al tener una mirada incompleta sobre redes de distintos tipos. Primero, se presentará una perspectiva general del comportamiento presenciado. Luego, las figuras con los gráficos en la sección de **Resultados** 5.3 serán referenciadas según sea posible visualizar el efecto del algoritmo aplicado. Adjunta estará la explicación del impacto en las métricas y las posibles causas y consecuencias que trae consigo. Este proceso, adicionalmente, busca desarrollar, a través de ejemplos, un marco metodológico para repetir el análisis y búsqueda de patrones en otros modelos no vistos.

6.1. Redes Generadas

6.1.1. Modelo ER

Mirada general — El modelo Erdős–Rényi, dada su naturaleza simple y aleatoria, destaca por su uniformidad, ya sea en sus propiedades estadísticas como en su topología. Por esto, el principal propósito en la experiencia de este tipo de red fue hacer de muestra de control. En otras palabras, se buscó un caso donde la topología no sesgara fuertemente las alteraciones de los algoritmos, principalmente, los algoritmos 3 y 4. Por lo menos, esto se refleja en los casos con poco “daño”⁴⁰, manteniendo cierta semejanza a sus redes originales. A su vez, en la práctica, lo mencionado no significa que las redes alteradas sean “mini” redes ER que se escalaron según la diferencia en la cantidad de nodos. En los casos donde la redes resultantes tenían las menores cantidades de nodos, los gráficos reflejan el caos consecuencia del azar.

En general, el comportamiento de Erdős–Rényi sujeto a falta de información es el siguiente: A medida que aumenta el “daño” a la red, las propiedades pasan desde un estado similar al original a un aumento no insignificante de la varianza en los resultados, propio de su origen aleatorio. En necesario añadir, que esta varianza tiene límites. Por ejemplo, no es probable generar *hubs* al muestrear una ER, ya que los todos los nodos mantienen un grado promedio similar. No imposible, cabe mencionar, ya que se puede producir el caso extremo de tener un grado promedio original alto y realizar un muestreo BFS de una cantidad de nodos ligeramente mayor a dicho grado. Esto puede transformar al nodo inicial del recorrido en un *hub*. Asimismo, es posible generar árboles, grafos completamente conexos, entre otros. Nuevamente, ya que se mantienen topologías y propiedades uniformes⁴¹, el modelo permite ge-

⁴⁰Casos en los que las redes resultantes tienen casi o la misma cantidad de nodos/conexiones que la red original

⁴¹En otras palabras, no tiene tendencias específicas marcadas. Como por ejemplo, las redes BA siguen una topología *scale-free*.

nerar casos de todo tipo. Prematuramente, se puede mencionar que el comportamiento de aumentar la varianza según disminuyen los nodos existen en cierta medida en todos los tipos de redes.

Específicamente, el patrón de pasar de un comportamiento similar al original al caos, es posible evidenciarlo en la figura 29, gráficos de violín. Al corresponder al algoritmo 1, la cantidad de nodos disminuye de izquierda a derecha. También se puede visualizar en los gráficos de violín de las figuras 45, 46 y 47, las cuales representan los algoritmos 3 y 4. En este caso, la cantidad de nodos aumenta de izquierda a derecha, sin contar la primera columna. A medida que disminuye el tamaño de la red, la varianza (la altura de los violines) aumenta. Como ya se mencionó, esta tendencia existe en todos los tipos de redes estudiados, pero ER presenta los casos más prominentes.

Distribución de grado — Lo anterior envuelve a todas las propiedades en general, por lo que, a continuación, el enfoque pasa a propiedades en específico. Respecto al grado y su distribución, la homogeneidad de ER mantuvo la falta de información distribuida e uniforme a lo largo de toda la red cuando se aplicaron los algoritmos de esta experiencia. Cabe mencionar que esto puede tener raíz en utilizar el azar como principal criterio en los algoritmos, ya sea para el procedimiento completo o para el impulso inicial. La eliminación de miembros y conexiones, en los algoritmos 1 y 2, causó un desplazamiento de la “campana” hacia la izquierda. Es decir, disminuyó el grado promedio. Lo cual es lógico, dado que en ambos algoritmos se eliminan conexiones. También, existe un “adelgazamiento”, lo que muestra una disminución de la varianza de los grados. Lo interesante de este caso es que la distribución se mantiene acampanada, o, mejor dicho, con una sola campana. El desplazamiento “diagonal”⁴² de la colina presenta un comportamiento proporcional a la cantidad de eliminaciones que provocó el algoritmo. Esto es evidenciable en los gráficos de grado en las figuras 29 y 44.

El caso de los recorridos, los algoritmos 3 y 4, conciben otros resultados. Para el 4, se puede ver, en la figuras 46 y 47, que la distribución tiene dos colinas. Esto contrasta más evidentemente con recorridos cortos, n bajo. La cima más pequeña, a la derecha, coincide con el grado promedio original; la más grande, a la izquierda, con grado pequeños, probablemente hojas del subgrafo resultante. A medida que crece el recorrido, la distribución se estabiliza en una sola campana que asemeja a las resultantes de los algoritmos 1 y 2. La pequeña colina, situada en el grado promedio original, representa a los nodos que se les revisaron todos sus vecinos al momento de iterar el camino. En el caso de BFS, estos nodos son los primeros seleccionados, ya que se revisa a anchura. Para DFS, esto ocurre cuando se llega a una hoja en la red original y se tiene que volver al siguiente vecino del nodo anterior. Por su parte, los recorridos aleatorios, como lo indica su nombre, presentan mucho más caos en los caminos cortos. Al momento en que se disipa el “desorden”, el subgrafo es suficientemente grande para estabilizarse en una sola cima. Esto se puede notar en el gráfico de distribución de grado de la figura 45.

Lo principal a rescatar sobre el grado de las redes alteradas es lo siguiente: para poca falta de

⁴²Se refiere al movimiento izquierda-arriba de la cima de la distribución.

información, es decir, una red resultante casi completa, la nueva distribución estará sesgada hacia la izquierda y tendrá menos varianza que la original. Este patrón se acrecenta junto con n . Para los recorridos, los deterministas crean dos colinas que convergen a una sola a medida que crece el subgrafo. Los estocásticos presentan caos hasta converger en los caminos más largos.

Coefficiente de *clustering* — La generación de la muestra ER se hizo con un valor p bajo, por lo que el coeficiente de *clustering* original ya es notablemente reducido. Solo los subgrafos más pequeños lograron hacer una muesca en la cantidad total de “triángulos” disponibles. Esto se ve en las figuras 45, 46 y 47, mediante cimbras menos pronunciadas para n bajo, cuyas faldas se adentran a valores mayores. Dado que el gráfico es un *kde*, una estimación de tendencia, se pueden ver valores negativos, lo que no es posible para el *clustering*. La interpretación de esto es que el largo de la “intrusión” en los valores negativos es resultado de la desviación estándar de la distribución.

Asortatividad — Los algoritmos de eliminación no sesgaron el valor promedio de asortatividad. Aleatoriamente, se pueden haber eliminado nodos hoja conectados a un *mini-hub*, aumentando el valor o quitado vértices que pertenecieran a un grupo fuertemente conectado, disminuyéndolo. Nuevamente, la uniformidad de ER, da que el valor promedio original sea alrededor de 0. Esto significa que no hay preferencia ni por los grupos fuertemente conectados, ni por dichas hojas. Los gráficos de violín de asortatividad de las figuras 29 y 44 pueden evidenciar este comportamiento. Dicho esto, el efecto en la asortatividad parece depender más del algoritmo aplicado que de la topología original.

Los recorridos, potencialmente pueden generar valores altos de asortatividad. Un grafo completamente lineal, un camino unidimensional básicamente, significa muchos nodos de grado 2 conectados a otros del mismo grado, formando una cadena. Como se ha expuesto anteriormente, la aleatoriedad del algoritmo 3 permite todo tipo de casos. La altura⁴³ en los violines del gráfico de asortatividad en la figura 45 muestran lo descrito. Asimismo, también evidencian el sesgo hacia valores mayores, en especial para n bajo. El algoritmo 4 presenta un caso para nada nimio. En contra del pronóstico los valores promedio disminuyen, como se puede ver en las figuras 46 y 47. Como ocurrió con las dos colinas para el caso de la distribución de grado, existirán nodos a los que el recorrido revisó todos sus vecinos, en especial para BFS. Esto logra recuperar los valores de asortatividad de la red original y al agregar hojas⁴⁴, se crea la disparidad que disminuye la asortatividad.

Diámetro y Camino más corto promedio — Aterrizando la discusión sobre los caminos internos de la red, hacia el diámetro y camino más corto promedio, nuevamente, es posible divergir dos comportamientos generales: Algoritmos de eliminación, 1 y 2, y de recorridos, 3 y 4. Quitar conexiones, indudablemente, tendrá un detrimento a las opciones de caminos que se pueden tomar desde a un nodo a otro. Quitar opciones puede significar en tomar

⁴³Al mencionar “altura” de los violines, se hace alusión a la varianza de las distribuciones que representan.

⁴⁴En este caso, nodos en los que el recorrido no pasó por sus vecinos o vértices que ya eran hojas originalmente.

rutas más largas para llegar de un punto a otro. Por consecuencia, la eliminación de nodos y/o conexiones (la eliminación de nodos también conlleva quitar conexiones), en la mayoría de los casos, mantiene o aumenta el diámetro de la red. Los gráficos de violín de diámetro en las figuras 29 y 44 ilustran como la concentración de valores “sube” por el eje vertical.

Se manifestó que esto ocurría en la mayoría de los casos, en especial en el algoritmo 2. En este procedimiento puede quedar un nodo completamente aislado, lo que inhabilita los valores de caminos hacia dicho vértices. Empero, en el 1 es posible que se elimine una hoja perteneciente al camino que establece el diámetro. Quitar este vértice puede disminuir el diámetro⁴⁵, ya que el camino no necesitar llegar a este.

Por su parte, los algoritmos de recorrido tienen un gran impacto en las métricas relacionadas a caminos. Como dice su nombre, estos métodos se basan en un camino. Es bastante probable que en los recorridos no en anchura, el mismo “tronco”, el camino principal, sea el que defina el nuevo diámetro. Como se ha visto, y se verá en el resto de los tipos de redes, el algoritmo 3, al ser completamente estocástico, logra ignorar, en mayor medida que el 4, la topología inherente del grafo. Viendo el gráfico de diámetro de la figura 45, este algoritmo permite diámetros de hasta la mitad de la red. Y, aunque tiene las mayores varianzas, también presenta los mayores valores promedio. En cambio, el algoritmo 4, al basar su iteración en vecindarios, BFS siempre y DFS cuando llegar a una hoja, sus gráficos en las figuras 46 y 47 ilustran violines con menos altura y más cerca del valor promedio original⁴⁶. Para BFS, con n menores, incluso el valor promedio es menor. Inequívocamente, esto surge de la prioridad a la anchura sobre la profundidad del método. En contraste, DFS muestra valores más altos, pero el paso de volver a vecindarios ya parcialmente revisados cuando se llega a una hoja hace converger hacia el valor promedio original a medida que aumenta n .

En suma, a lo largo del resto de tipos de redes, se podrá ver que las propiedades de camino, diámetro y camino más corto promedio, recibirán un impacto más marcado por el algoritmo aplicado, en especial para 3 y 4, que por la topología de la red.

El caso de camino más corto promedio tiene un comportamiento similar, y vinculado (ya que el diámetro es, simplemente, el máximo camino más corto), al del diámetro. Para los algoritmos 1 y 2, símil al patrón en el diámetro, la eliminación de elementos del grafo hace crecer a los caminos. En las figuras 29 y 44 es notable el aumento proporcional al n de las posiciones de los violines. Análogamente, el comportamiento de esta métrica sigue un patrón similar al caso del diámetro según su alteración aplicada. Las figuras 45, 46 y 47 revelan esto.

Robustez — Para terminar con ER, queda discutir la conducta de la robustez bajo falta de información. La robustez, específicamente el factor f_c mide la cantidad de nodos que se deben eliminar para que la componente gigante se descomponga. Dicho sea de paso que su cálculo recibe como uno de sus parámetros el grado promedio en la red, por lo que la eliminación de conexiones también afecta al valor. Naturalmente, en los algoritmos de eliminación, las figuras 29 y 44 reflejan directamente dicho ataque hacia la componente gigante. A medida que

⁴⁵Si es que el diámetro se ve en más de un camino, no disminuiría el valor.

⁴⁶Para ver esto, es necesario notar las escalas y valores que se recorren en los ejes verticales de los gráficos.

crece n , más elementos han sido eliminados y la nueva fracción a eliminar, f_c , decrece. Para los recorridos, el análisis es más confuso y complejo. Como se ha venido repitiendo, cuando el camino va creciendo, el comportamiento del subgrafo converge a asimilarse al original. Para n bajo, el caos prevalece sobre los resultados, pero aún así, en 4 se mantiene en valores altos, como se ve en las figuras 46 y 47. El caso de interés está en el algoritmo 3. Al no tener el enfoque en vecindarios del 4, se permite que las redes resultantes sean más “lineales”. A su vez, esto puede causar que la componente gigante posea una sola conexión, un puente, que la mantenga unida, lo que hunde el valor del factor f_c .

6.1.2. Modelo SBM

Mirada general — El modelo de bloques estocásticos tiene una concepción muy similar a Erdős-Rényi, con la diferencia de ahora se fuerza una estructura de comunidades. Lo que se logra es tener una topología, todavía aleatoria, pero marcada por la agrupación en bloques. Ya que en este trabajo no se trató la detección de comunidades, el uso de este modelo no llegó a todo su potencial. La definición de dos probabilidades distintas⁴⁷, la intra-comunidad y la inter-comunidad, buscaba diferenciar a este tipo de redes de ER. Con todo, esta desigualdad fue menor de la esperada. A grandes rasgos, el comportamiento de SBM fue considerablemente similar a la conducta de ER. Lo que por su cuenta es racional, debido que este modelo es generado con la misma metodología que ER.

No obstante, aún se abarcarán los detalles que más resaltaron por sobre la similitud a ER.

Distribución de grado — En esta propiedad, la única diferencia con ER es el grado promedio. Y, así, los subsecuentes valores medios al momento de alterar la red. La figura 48, a primera vista, puede parecer un poco distinta. Sin embargo, esta discrepancia surge del suavizado que se le aplicó a la estimación de tendencia.

Un resultado que sí tiene un cambio considerable respecto a ER es en la aplicación de n grande para el algoritmo 3. En la figura 49 se puede visualizar que las líneas para $n = 800$ y $n = 1500$, roja y morada respectivamente, trazan distribuciones de grado mucho más achatas que en el caso de ER. Además de mostrar menos sesgo desde el valor promedio original. La particularidad que pudo haber causado esto parte de la generación de este modelo. Dados los valores de p entregados al método, se estima un grado promedio entre bloques de 10 y 4 para las conexiones que conectaban comunidades distintas. Como la cantidad de bloques y su tamaño se determinaron de manera aleatoria, la cantidad de conexiones que podrían existir entre bloques varía.

Para explicar mejor esta particularidad, se hará un paréntesis:

Se busca generar una red SBM de 30 nodos. Como se mencionó, la cantidad de bloques y su tamaño es aleatorio. El parámetro de conexión intra-comunidad es de $10/n$, lo que en

⁴⁷El modelo soporta múltiples probabilidades, pero, por simplicidad, se utilizaron dos.

este caso de ejemplo sería $1/3$. Por simplificar los cálculos, se tomará el valor esperado de la cantidad de conexiones en cada bloque, definido por:

$$|E| = \text{tamaño bloque}^2 \cdot \text{prob. de conexión}$$

Entonces, es posible generar los siguientes casos:

- Caso 1: 1 bloque de tamaño 10 y otro de tamaño 20.
- Caso 2: 3 bloques de tamaño 10.

Por consiguiente, computar el valor esperado resultaría en:

Caso 1

$$|E| = 10^2 \cdot \frac{1}{3} + 20^2 \cdot \frac{1}{3}$$

$$|E| = 166.\bar{6}$$

Caso 2

$$|E| = 10^2 \cdot \frac{1}{3} + 10^2 \cdot \frac{1}{3} + 10^2 \cdot \frac{1}{3}$$

$$|E| = 100$$

Ignorando la probabilidad inter-comunidad ⁴⁸, la cantidad de conexiones es favorecida para menos grupos grandes que más bloques pequeños. Puesto que la cantidad de bloques, y sus tamaños, era determinada al azar dentro de un rango de 5 a 20, se dio rienda a redes menos densas de lo esperado. Lo que, a su vez, se puede notar en que ER tuviera un grado promedio más alto.

Por si solo, esto no explica el fenómeno observado. No obstante, es menester recordar que los recorridos dan un punto de vista más local. Los caminos, en especial los aleatorios, tienen más chances de quedarse “atrapados” dentro de un bloque, dado al mayor grado que viaja internamente que hacia afuera. Por consecuencia, el grado rescatado será alto. Es posible ver una segunda aparición de este fenómeno en las figuras para BFS y DFS, 50 y 51, respectivamente. El sesgo es, proporcionalmente, menor que el visto en ER.

Coefficiente de *clustering* — Como es apreciable en todas las figuras de SBM, 30, 48, 49, 50 y 51, el efecto es prácticamente el mismo que en ER. Como ya se discutió para dicha

⁴⁸Esta sí se beneficia de mayor cantidad de grupos, pero el valor esperado estaría escalado por un 4 en vez de un 10.

red, la generación del modelo con un valor de p bajo, deja libres muchos de los triángulos disponibles, por lo que el coeficiente es, consecuentemente, bajo.

Asortatividad — Ya que el modelo tiene una generación aleatoria, como ER, se podría argumentar que no presenta una topología particularmente marcada. Lo cual es cierto para el contexto de esta experiencia, puesto que no se revisa modularidad. Es sabido que SBM, en comparación al otro modelo aleatorio visto, si presenta una estructura particular: el agrupamiento por comunidades. Localmente, los bloques deberían presentar una red ER en “miniatura”, lo que ya mostró una asortatividad centrada en 0. La diferencia entra en las “fronteras” del bloque. En otras palabras, los nodos que forman puentes entre comunidades. Por probabilidad, sin considerar conexiones puente, estos vértices presentan el valor esperado de grado dentro de la comunidad. Al agregar las aristas entre bloques, aumenta el grado de los nodos entre ambas comunidades conectadas. Los nodos frontera, por consecuencia, tendrán conexiones tipo “rico con rico”. A su vez, esto aplica naturalmente un pequeño sesgo positivo respecto al modelo anterior. Más allá de eso, para los efectos de esta experiencia, la alteración aplicada no se diferencia mucho de la vista en ER.

Diámetro y Camino más corto promedio — Nuevamente, el efecto visto sobre los caminos, y por consecuencia, el diámetro, es similar al modelo ER. La principal diferencia es la magnitud del sesgo y los rangos de valores que se ven. En contraste, el otro modelo estocástico tiene una disponibilidad uniforme de acceso desde un punto de la red a otro. Es decir, no hay inclinaciones particulares respecto a sí, posicionando un vértice inicial en un sector de la red, el resto de la red es de una pareja facilidad de acceso.

Dada a la agrupación en bloques y junto a su principal causa, la menor probabilidad de conexión inter-comunidad, los nodos “puente” resultan siendo las únicas opciones de ir de un bloque a otro. Sin alteraciones, se puede ver en las imágenes que los promedios originales de camino más corto y diámetro son mayores en SBM. Naturalmente, esta diferencia se acrecienta cuando se eliminan las pocas opciones de rutas cortas que existen. No solo las figuras 30 y 48 muestran que las eliminaciones generan un rango de valores mayor, sino que también se pueden visualizar caminos más largos en las figuras 49 y 51. Se dejó fuera a BFS puesto que, primero, priorizando la anchura se consiguen, en comparación, valores menores para esta propiedad y, segundo, este algoritmo puede quedarse en, y revisar al completo, un bloque antes de intentar salir. Bloques que, por chance, tienen más rutas internas que externas.

Robustez — Cambiando el contexto a robustez, los mencionados puentes podrían volverse una debilidad del modelo. Matemáticamente, al cómputo no le importa la existencia de estos puentes. Empero, estos si hacen muesca en el cociente de Kappa, ecuación 5. Analíticamente, los puentes son los puntos débiles de la componente gigante. No obstante, es importante tener en mente que, ya que la probabilidad inter-comunidad es constante, existen puentes entre cada par de bloques. Por esto, el valor de robustez no es “abismal”, si no que es ligeramente menor que en ER. Como se vió, y se verá en otras redes, los algoritmos de eliminación ponen en evidencia el efecto que eliminar conexiones tiene en la robustez de la componente gigante. Por su parte, los recorridos, a medida que crece n , convergen a rescatar los valores originales, pero con un sesgo negativo.

6.1.3. Modelo WS

Mirada general — El propósito de incluir el modelo Watts-Strogatz en esta experiencia fue tener un modelo que mezclara el determinismo con la aleatoriedad. El inicio como anillo le da una estructura regular, cuya configuración se modifica estocásticamente con el “recableado”. Dado que se utilizó $k = 8$ como parámetro, cada nodo comienza conectado con 8 “pre-vecinos” en una topología de anillo. Sin modificaciones, se tiene un camino que rodea el anillo completo. Empezar a redefinir conexiones es lo que consigue que este modelo tenga el efecto *small-world*. Mencionada estructura trae distintos casos interesantes al momento de simular un falta de información.

Distribución de grado — Ya que todo nodo es forzado, al inicio, a tener k aristas, naturalmente, este será el grado promedio, con varianza 0, en el estado previo al “recableado”. Aplicar una probabilidad p a este proceso, tendrá un efecto, proporcional, de aumento en la desviación de los grados, aún centrado en k como media. Dado que en esta experiencia el valor de p fue bajo, un 10 %, la campana de la distribución de grado no ensanchó mucho. En la figura 21, también se puede evidenciar que la mayor parte de los nodos tiene el grado promedio. En consecuencia, los algoritmos de eliminación tiene más probabilidades de apuntar a uno de estos vértices o sus conexiones. Las figuras 31 y 52 muestran que, a medida que se eliminan elementos, la distribución de grado se achata y desplaza a una media menor. Este efecto, cuando llega hasta casi la mitad de la red, logra que se empiece a asemejar un distribución de grado tipo ER. Al quitar tantos nodos de grado k (parámetro k del modelo WS), se comienza a perder la topología de anillo y sobrevive la red aleatoria que produjo el “recableado”.

El caso de los algoritmos de recorrido es de interés, ya que aprovechan completamente la propiedad de *small-world*. Pueden, gracias a esto, viajar a cualquier parte del anillo. Desgraciadamente, la distribución de grado no puede ilustrar este hecho por si sola, empero, aún así presenta resultados no triviales. Como se mencionó, a menor cantidad de nodos, más similitud se tiene con una red ER. No obstante, los tres algoritmos de recorrido “reconstruyen” el anillo de distintas formas. DFS, representado por la figura 55, tiene el potencial de dar la vuelta completa al anillo. Sumado a la modificación de agregar conexiones que existían en la red original, aprovecha el efecto de mundo pequeño para rescatar, posiblemente, largos arcos del anillo. El grueso del anillo, expuesto anteriormente, son, principalmente, nodos de grado k . En otras palabras, ya que DFS puede circunvalar la topología, rescata más de estos nodos.

Por su parte, BFS, tiene a su favor la probabilidad de construir pequeños arcos del anillo, mas en cualquier parte de este. Puede revisar un vecindario en un sector de la topología y luego saltar al lado opuesto de la circunferencia, gracias a los nuevos “cables”, y repetir el proceso. Se estarían generando pequeños subgrafos casi regulares ⁴⁹ conectados por puentes de un lado a otro del anillo. Así, la figura 54 muestra campanas más regulares que su contraparte. Por último, visto varias veces ya, el algoritmo 3 toma rutas caóticas que generar un gran

⁴⁹El anillo es una topología regular, pero pierde esta condición por el recableado.

diversidad de casos. Lo sustancial a rescatar es que el sesgo de las distribuciones es más hacia la izquierda que en los otros algoritmos. Así como, más pronunciadas las cimas en la figura 53.

Coefficiente de *clustering* — La topología de anillo, que inicialmente tiene a cada nodo conectado con sus 8 vértices “adyacentes”⁵⁰, hace que, localmente, se formen triángulos entre los miembros de la red. Por ejemplo, visualizando tres nodos seguidos en la circunferencia del anillo, identificados con 1, 2 y 3. Estos se conectan a sus 8 vecinos, por lo que existen las conexiones 1-2 y 1-3. A su vez, 2 y 3 también son adyacentes, por consiguiente, se conectan. Este patrón se replica 8 veces por nodo. Dicho esto, el modelo WS es el que presenta los mayores valores de *clustering*. Por sí sola, la topología tendría un coeficiente constante a lo largo de todas las redes generadas. Al introducir el recableado es cuando los valores se dispersan en una distribución.

Eliminar miembros, en el contexto de este trabajo, trae dos posibles resultados. Eliminar un nodo se deshace de todos los triángulos disponibles en lo que podría participar, aumentando el coeficiente de la red. Por contraparte, también se eliminan los triángulos de los que sí participa, afectando tanto a dividendo como divisor en el cálculo de *clustering* de los nodos supervivientes. Este va y ven resulta en que el gráfico de distribución del coeficiente de la figura 31 presente mayor varianza, pero poco sesgo al valor promedio. El otro resultado posible viene de solo eliminar conexiones. El impacto posible, al solo aumentar los triángulos vacíos, es de disminuir el valor del coeficiente. Por esto, la figura 52 presenta menos dispersión y más sesgo en comparación.

Símil al caso del grado, BFS, rescatando pequeños trozos de la circunvalación, presenta más varianza. Comparando las figuras 54 y 55, en DFS se ve más sesgo positivo, posiblemente resultado del rescate de mayor parte del grueso del anillo mencionado. Como ya se expuso, la parte regular del anillo tiene un *clustering* localmente alto.

No siendo la última vez, el algoritmo 3, retorna *snippets* de la red. Las diferencias del valor promedio en la figura 53 es, en contraste, menor. Este fue uno de los casos donde el algoritmo destacó por ser el “intermedio” entre BFS y DFS.

Asortatividad — La asortatividad de la topología inicial de este modelo, el anillo, es teóricamente⁵¹, 1. Esto porque es una estructura regular independiente del valor de k , lo que conlleva que todos los nodos sean iguales respecto al grado. Como se detallará más adelante, en el modelo Grid, el valor de asortatividad debería ser 1, pero no se muestra así en los violines que representan las redes originales. Este caso borde hace que se compute la asortatividad para las conexiones recableadas. Por esto los resultados, cuando la red resultante es de magnitudes equiparables a la original, y, claro, los valores de las mismas redes originales, son poco fiables.

⁵⁰Adyacente bajo el contexto de una topología en circunferencia.

⁵¹Se dice teóricamente porque la librería NetworkX tiene un caso borde con topología regulares. Más detalles en la sección del modelo Grid.

De todas formas, es todavía posible analizar los casos de recorrido y cuando no se revisa una gran parte de la red. Viendo la figura 31, se puede ver el sesgo positivo a medida que aumenta n . La circunvalación previa al recableado posee lo que se conoce como asortatividad perfecta. En consecuencia, cambiar conexiones solo puede disminuir este valor. Eliminar nodos que participaron en este proceso logra tanto, deshacer el recableado aumentando la asortatividad, como disminuir el grado de sus vecinos, posiblemente disminuyéndola. Lamentablemente, por la particularidad del cálculo de esta métrica, los gráficos para los algoritmos de eliminación no son del todo íntegros.

A diferencia del resto de modelos, los sesgos para los recorridos BFS y DFS son positivos. Nuevamente, el rescate de trozos del anillo, ya sean pequeños para BFS o grandes para DFS, recuperan la parte que aporta a aumentar la asortatividad: el grueso del anillo original. Lo que no dista del resto de los tipos de red es la varianza presente en las figuras 54 y 55. Por su parte, el algoritmo 3 mantiene su comportamiento ya visto: asortatividades mayores debido a los subgrafos “lineales”.

Diámetro y Camino más corto promedio — El efecto sobre las métricas de camino no se aleja mucho del ya visto anteriormente en la experiencia, en especial para los algoritmos de recorrido. La principal distinción vista en las figuras es que WS muestra valores más altos. Inicialmente, los caminos de una topología de anillo son equiparables a la mitad del perímetro de esta ⁵². Para k mayor a 2, estos largos disminuyen. Esto al no tener que pasar por cada nodo y tomar atajos a través de los vecinos no directamente adyacentes. Aún con esto, los largos son proporcionales al tamaño de la red, a su cantidad de nodos. Contrariamente, lo que destaca a este modelo es tener la propiedad de *small-world*, tener camino cortos, relativamente, independiente de la magnitud del grafo. Esto se consigue gracias al recableado.

El impacto del recableado sobre disminuir el largo inicial de los caminos depende del parámetro p . En este caso, al ser 10 %, el efecto es moderado, por lo que las redes originales ya tienen valores más altos que ER. Como se ha visto, la eliminación genera, por lo general, aumentos en el largo de las rutas. Eliminar las conexiones recableadas estaría removiendo la propiedad de *small-world* que el proceso agregó inicialmente. Dado que la eliminación es aleatoria, no logró quitarla por completo. No obstante, viendo las figuras 31 y 52, se puede ver que, proporcionalmente, existe un mayor sesgo en los valores promedio de las redes alteradas.

Robustez — Un anillo no es particularmente robusto. Para $k = 2$, basta quitar dos conexiones para partir la circunferencia por la mitad. Aumentar k , y posteriormente recablear, hacen que la robustez de las redes originales sea un poco menor al del visto en redes ER.

Dicho esto, el impacto que tuvieron los algoritmos prácticamente no dista del presentado por ER. Matemáticamente, la particularidad de aumentar el grado de algunos, los que el recableado los dejó con nuevas conexiones, hace crecer el factor κ ⁵³. Lo que a su vez, aumenta la robustez de la red. Analíticamente, el anillo es una red regular casi “lineal” en la que la “co-

⁵²Mitad porque, al ser no dirigido, se puede hacer la ruta en ambas direcciones.

⁵³Hacen crecer ambas partes de la fracción, pero el numerador tiene un crecimiento al cuadrado.

la” se conecta con el inicio, lo que no la hace muy distinta del modelo Grid (más adelante se verá que el modelo Grid tiene la peor robustez de todas las redes estudiadas). Redirigir las conexiones a otros sectores aleatorios de la red, logra una interconectividad local (en la vecindad de un nodo) que se asemeja a ER.

6.1.4. Modelo BA

Mirada general — Si no el más importante, de los más importantes, el modelo BA es el caso de estudio con mayor relevancia en esta experiencia. Dada su naturaleza *scale-free*, este tipo de redes fue usado para representar a una gran cantidad de fenómenos en la realidad que siguen leyes de potencia. Es más, si el modelo ER fue seleccionado por la poca influencia de su topología, para este modelo es todo lo contrario. Descrita ya en la sección **Marco Conceptual** 3.2.4, la presencia de *hubs* y su interacción con los otros nodos de la red generan resultados interesantes al privar la red de información completa.

Para los algoritmos de recorrido (3 y 4), dada la misma particularidad que hace *hubs* a los *hubs*, es muy probable que los caminos, ya sean estocásticos o deterministas, lleguen eventualmente a uno de dichos nodos. Esto a su vez, hace que las redes resultantes sean, probablemente, *scale-free*. Es necesario mencionar que el recorrido tipo DFS previsiblemente pase por un nodo *hub*, pero no llegue a generar una ley de potencia como distribución resultado. Esto es porque el camino potencialmente “pasa de largo” a otros los *hubs* al priorizar profundidad antes que anchura. Sin embargo, el caso de originar una topología diferente pasa con recorridos de n bajo, ya que al aumentar la cantidad de nodos a rescatar, es más probable llegar a otros nodos céntricos. Un poco difícil de notar por los límites del eje vertical, en los gráficos de violín de las figuras 58 y 59, se hace énfasis a la comparación entre las alturas de los violines cuando n es bajo. Aunque distorsionada por dichos límites, la dispersión es mayor en DFS. A medida que crece la cantidad de nodos, las fluctuaciones se estabilizan más rápido en BFS. Esto se debe, posiblemente, a que se llega a un *hub* y este da acceso, con la búsqueda a lo ancho en sus vecinos, a la mayoría del resto de *hubs* de la red.

Distribución de grado — Respecto a las distribuciones de grado, este modelo destaca por el poco impacto que tuvo la falta de información. La topología *scale-free*, y dicho sea de paso, las leyes de potencia, son estructuras extremadamente marcadas y prevalentes en el contexto de redes. Los *hubs*, son pocos, por lo que es poco probable eliminarlos de manera aleatoria. Encima, poseen los grados más altos de la red, por lo que es tan poco probable que un recorrido, de cualquier tipo, esquivе y no pase por un *hub*. O, por otro lado, eliminar estas conexiones apenas hacen muesca al grado de estos centros. Esta improbabilidad de escapar a los *hubs* hace que, independiente de la simulación de falta de información, la red resultante mantiene la topología libre de escala y una *powerlaw* como distribución de grado.

En todas las figuras sobre BA, 32, 56, 57, 58 y 59, los gráficos de distribución de grado mantienen una ley de potencia para todos los casos. El algoritmo 3 pareciera tener una discrepancia. La *powerlaw* para $n = 100$, la línea amarilla, es mucho más pronunciada que las demás.

El subgrafo rescatado presentó una mayor de concentración de hojas, aunque no pierde la forma de ley de potencia. Cuando el tamaño de la red resultante es muy pequeño, cabe la posibilidad de no rescatar a todos los *hubs*, pero aún están presentes en la versión alterada.

Coeficiente de *clustering* — Pasando el foco al *clustering*, en una red BA, el coeficiente de *clustering* tiene un comportamiento parecido al grado. Los *hubs*, dado el procedimiento de generación del modelo BA, tienen alta probabilidad de ser vecinos de otros *hubs*. Estos son los pocos nodos que completan los triángulos disponibles. Es más, es también prevalente el caso de que un nodo de grado bajo se conecte con múltiples *hubs*, que, a su vez, son vecinos entre sí. El resto, en su mayoría hojas o vértices con pocas aristas, los que componen la mayor parte de la red, computan un *clustering* bajo. Por ende, la distribución del **clustering** sigue, también, una ley de potencia.

En la experiencia, los algoritmos de eliminación, por lo mencionado con anterioridad, tienen poco efecto en los *hubs*. Como se ve en las figuras 32 y 56, el *clustering* mantiene una distribución muy similar para el caso original y todos los n aplicados. Viendo los recorridos, BFS, como fue notado para estos algoritmos, llegará eventualmente a un nodo céntrico. Al buscar en anchura, se aprovechará al completo la particularidad de ser *hub* del nodo elegido. Lo que, por probabilidad, significa rescatar el resto de los centros. Sumado al comentario del *clustering* en este tipos de nodos, la imagen 58 muestra colinas considerables, para la escala del gráfico completo, en valores altos. DFS, que puede tomar más pasos, pero de todas formas recorrerá por los *hubs*, también presenta dichas cimas. Nuevamente, el enfoque de vecindades, ya sea prioritario (BFS) o de respaldo (DFS), de estos métodos saca mayor provecho de la topología de la red. En cambio, la contraparte aleatoria del algoritmo 3, se desentiende, en mayor medida, de la estructura del grafo.

Asortatividad — La estructura *scale-free*, con su clara distinción entre *hubs* y hojas, y la cantidad de nodos que manejan ambos grupos, calcula asortatividades más bajas que un modelo aleatorio como ER. Aquí, la eliminación de una conexión entre hoja y *hub* tendría un impacto mayor sobre el valor de la propiedad, aumentando más que lo hizo en el caso de ER. A pesar de esto, el efecto es local, y este tipo de conexiones son el grueso de la red. Lo que mezclado con la dificultad de eliminar un nodo céntrico de manera aleatoria, hacen que ambos algoritmos de eliminación, como ilustran 32 y 56, tengan un efecto no distinto al visto en ER.

Por su parte, los recorridos, con su particularidad de inevitablemente rescatar los *hubs*, incluyen menos hojas en el subgrafo final. Esto lleva, al haber menos probabilidad de que las hojas tengan una conexión previa, a hundir el valor de la asortatividad. En cualquier caso, el efecto sigue siendo similar a ER. En las figuras 58 y 59 se puede vislumbrar el sesgo. El caso destacable, como en el resto de tipo de redes, es el algoritmo 3. De nuevo, la propensión de formar redes más “lineales”, sesga el valor hacia arriba.

Diámetro y Camino más corto promedio — En las métricas de diámetro y camino más corto, los algoritmos de eliminación han tenido un efecto constante a lo largo de los tipos de redes, variando menos, que el resto de métricas, según la topología subyacente. Para el diámetro, basta que el singular camino que lo define, si es que no comparte distancias con otros, se

tenga que alargar por eliminación de opciones para que el valor crezca. Esta facilidad de aumentar, hace que hasta la topología *scale-free* tenga dificultades al mantener la estabilidad de esta métrica. Estabilidad que se pierde según aumenta el n de eliminación. De manera análoga, y como se ha descrito anteriormente, todos los caminos de la red sufren por la eliminación. Por ende, el efecto en el promedio de estos no dista del descrito en ER. Para evidenciarlo, los gráficos pertinentes en la figuras 32 y 56 muestran sesgos similares a la redes aleatorias.

Los algoritmos BFS y DFS siguen el comportamiento esperado en el diámetro. Respectivamente, 58, que muestra el recorrido anchura, sesga la media hacia valores menores. Por su parte, 59, destaca la profundidad con valores más altos como media. Cabe destacar que, gracias a los *hubs*, la topología libre de escala también presenta el efecto *small-world*, por lo que los rangos de los largos de camino se mantienen estables. Como contraste, la figura 57 recalca, nuevamente, que el algoritmo 3 se ve menos influenciado por las características que posee la topología. Es necesario, para notar esto, tener en consideración los rangos de valores que presenta el eje vertical⁵⁴.

Robustez — Las redes libres de escala, entre los modelos revisados, son el tipo que más robustez presenta. La principal evidencia es el poco sesgo visto en las figuras 32 y 56, que representan a los algoritmos de eliminación. Con n de la magnitud de la mitad de la red, es cuando se comienza a ver un impacto no nimio a la integridad de la red. Cabe mencionar, que el *outlier* más bajo equivale a un f_c del 80 %. Cuando se están eliminando tantos elementos, crece la probabilidad de tener como objetivo a un *hub*, nodos que son los pilares fundamentales de la robustez de una *scale-free*.

Finalmente, BFS, en la figura 58 presenta redes aún más robustas. Aprovechando las vecindades de los *hubs* y saltando entre estos, rescata subgrafos, relativamente densos respecto a las conexiones. Ya que DFS, en la figura 59, también usa las vecindades, aunque de “respaldo”, también hace uso de los nodos céntricos. Empero, su prioridad por la profundidad no permite redes tan robustas como en BFS. Siguiendo su propio carril, el algoritmo 3 permite, para n bajo, casi todo el espectro disponible de robustez. Una vez más, el caos de este método lo hace el más impredecible y complejo de analizar.

6.1.5. Modelo Grid

Mirada general — El caso del modelo grilla es uno de los más interesantes. Su generación, completamente determinista, da paso a que el efecto del azar solo entre en la aplicación de los algoritmos. Esta particularidad llevó a su elección como muestra de control, símil a ER, pero con aún menos dependencia de probabilidades. Como el resto de tipos de red, también sus resultados presentan mayor varianza a medida que disminuyen los tamaños

⁵⁴Múltiples veces se ha hecho esta consideración. Lamentablemente, estandarizar el rango de valores para todos los gráficos haría que los que abarcan menos espacio no se pudieran apreciar correctamente.

de los grados resultantes. No obstante, este modelo es más resistente⁵⁵ al “daño” o falta de información. Esto es, se necesita eliminar o perder más nodos/conexiones para que los resultados diverjan, en comparación a las otras redes.

Una evidencia de esto se puede ver en los gráficos de violín de la figura 33. No es tan claro en el de asortatividad, pero en el resto se visualiza que, incluso eliminando más del 25 %, los resultados mantienen baja desviación⁵⁶. Aun, en la figura 62, donde se aplica BFS, se presentan variaciones nulas o casi nulas. Cabe mencionar que esto se debe a que el recorrido *breadth-first*, va poco a poco generando los cuadrados que forman la grilla. Por lo que, independiente del punto de inicio, los muestreos tendrán topología y características similares.

Los algoritmos que más creaban dispersión en los resultados fueron el 3 y el DFS del algoritmo 4. Proyectando el grafo como la grilla cuadrada que representa, los nodos solo están conectados con los cuatro vecinos de los cuatro cuadrados en los que forma parte como vértice. En otras palabras, los nodos no tienen conexiones con otros sectores de la grilla. Esto produce que los recorridos no en anchura (aleatorio y, en especial, DFS) tengan más probabilidades de generar redes lineales o árboles⁵⁷. El caso de grafo lineal⁵⁸, principalmente generado por DFS, permite, como se ve en los gráficos de diámetro de la figura 63, redes más “largas”. Esto se refiere a que una gran parte de los nodos pertenecen al camino más largo. Por su parte, los caminos aleatorios, representados en la figura 61, convergen a valores menores de diámetro, pero aún así, más grandes que el recorrido en anchura.

Antes de adentrarse en el análisis, preliminarmente se puede descartar la propiedad de *clustering*. La red sigue una estructura de grilla cuadrada, lo que significa que las conexiones solo existen en las aristas de las figuras que forman la teselación. Por esto, ninguno de los triángulos disponibles está definido, computando un coeficiente de *clustering* nulo. Dado que todos los algoritmos no crean información nueva, sino construyen o destruyen, a partir de redes existentes, ninguna de las figuras asociadas al modelo Grid tendrá un *clustering* definido. De hecho, una consecuencia de esta topología y de no crear conexiones nuevas es que los subgrafos rescatados siempre seguirán la estructura de grilla, aunque, por la falta de información, no siempre se cerrarán los cuadrados.

Distribución de grado — Eliminar un nodo en este patrón de topología provocará que los cuatro vecinos del removido bajen su grado a 3. En un comienzo, la probabilidad apunta a repetir este proceso en otros sectores de la grilla. Sin embargo, con suficientes selecciones, se puede escoger un nodo que es vecino de un vértice de grado 3, lo que a su vez, bajaría su grado a 2. Es por esto que las distribuciones de la figura 33, a medida que crece n , van cada vez aplanando más las colinas, redistribuyéndose en grado menores.

Al tener grado promedio 4, este modelo tiene la mitad de conexiones que el valor esperado

⁵⁵No en el mismo sentido que la propiedad de robustez.

⁵⁶Eliminar el 25 % puede parecer poco, pero los modelos anteriores presentaban fluctuaciones mucho antes.

⁵⁷Árboles de más profundidad que anchura.

⁵⁸No es estrictamente lineal, donde los nodos se conectarían formando un solo camino, sino que existe un camino principal, el que define el diámetro de la red, al cual le crecen ramificaciones a lo largo.

de ER, siendo de las menores cantidades de conexiones en la redes generadas. Como se ve la figura 60, el efecto de eliminar conexiones tiene un impacto considerable para n alto. Incluso, este fue el caso con más nodos de grado 0 entre los modelos presentados. Se dijo, anteriormente, que el modelo era robusto respecto a la divergencia de los resultados. Empero, como se discutirá más adelante, en la propiedad de robustez, es el modelo más débil.

En los recorridos, la figura 61 muestra como, para cualquier n , la distribución presenta como la mayoría de nodos tiene grado 2. Puesto que el algoritmo 3 no utiliza las vecindades al completo (simplemente elige un nodo de esta y se olvida completamente), se favorecen las cadenas lineales por sobre árboles más complejos para el recorrido preliminar. Estas cadenas causan muchos grados igual a 2, pero, gracias a la inclusión de conexiones que no fueron visitadas, mas sus nodos sí, los resultados presentan valores 3 y 4. La atención a este caso procede de que para todos los n , la distribución resultante es prácticamente igual, Símil al caso de WS. La posible razón de esto deriva que la red es completamente uniforme, por lo que, probablemente, los caminos resultantes sean isomorfos y “modulares”. Modular porque el largo del camino no afecta a la distribución. Entonces, pasar de $n = 100$ a $n = 300$, triplicar el largo, o pega tres recorridos uno en la cola del anterior, lo que resulta en un reparto de nodos proporcional.

Asortatividad — Este resulta ser un caso particular. Visto en la ecuación 4, en el denominador, se esta dividiendo por desviaciones estándar de distribuciones de grado en particular. El modelo Grid, dada su formulación, no tiene varianza en su distribución de grado. Por ende, se estaría dividiendo por 0. Este es un caso borde conocido de la librería NetworkX [Cetinkaya, 2012]. Específicamente, el método retorna NaN, mas en los gráficos, por defecto en la experiencia, se llenaron con 0 estos casos. Por esta razón, todas la figuras asociadas a este modelo marcan la asortatividad original como 0, cuando, teóricamente, el valor debería ser 1. Inequívocamente, en la redes inalteradas, todos los nodos tienen el mismo grado. Haciendo pruebas asociadas a esta problemática, se eliminó un solo nodo de una red grid de 20×20 . El resultado de computar la asortatividad resultó ser un valor cercano a 0. El efecto esperado, al tener 396 nodos de grado 4 conectados entre sí y tan solo 4 vértices distintos (grado 3 por el agujero que dejó el nodo eliminado), debería haber sido un valor cercano a 1.

Teniendo esto en mente, las figuras 33 y 60, con n no muestran resultados fiables en el gráfico de asortatividad. Por su parte, si son más íntegros los resultados plasmados en las figuras 61, 62 y 63. Exceptuando la columna que representa a la red original. BFS, que a diferencia de los otros, sesga los valores hacia el sector negativo. Hacer un recorrido en anchura por la grilla, podría generar un camino lineal con hojas a ambos lados en cada paso. Casi como una estructura molecular en química orgánica. El tronco de carbonos sería el camino lineal y los hidrógenos serían las hojas que resultaron de revisar la vecindades locales. De vuelta al contexto de redes, se estaría aumentando la asortatividad con el tronco principal, pero disminuyendo con las hojas. Cabe mencionar que la modificación impuesta a BFS hace que agregue conexiones entre las hojas cuando tenían originalmente una relación. Por consiguiente, los hojas no son estrictamente hojas, pero tienen menor grado que el tronco, así,

restando a la asortatividad. Asimismo, es preciso recordar que el grado máximo es 4, así que las diferencias son pequeñas, pero existentes.

DFS, priorizando la profundidad, y el algoritmo 3, en su habitual caos, causaron un sesgo hacia valores más altos. Se ha establecido varias veces que, aunque DFS tenga una predilección por la hondura, mantiene, como respaldo, la revisión de vecindades. Puesto que la estructura de grilla es periódica, pensando en tres dimensiones, los caminos se están desplazando por un toroide. Mientras no “choque con su cola”, un recorrido DFS puede llegar a su n objetivo sin necesidad de volver a revisar una vecindad pasada. Esto sumado a que los subgrafos lineales tienen inclinación por la asortatividad alta, se puede encontrar en la figura 61 evidencia de este comportamiento.

Diámetro y Camino más corto promedio — Se hizo mención de que este modelo es débil frente a las eliminaciones y no es inesperado que la pocas opciones que se tienen para los caminos afecten tanto a las métricas de caminos como al grado. A fin de cuentas, solo se deben eliminar 4 conexiones para aislar completamente a cualquier nodo de la red. Por esto, en la figuras 33 y 60, el diámetro sigue una tendencia en la misma dirección que el n aplicado. Como es evidente en los mismo gráficos, con el n más alto, el patrón propuesto es interrumpido. Visto tanto en diámetro como en camino más corto promedio, al eliminar tantos elementos, muchos vértices quedan completamente aislados de la red. Nodos compartidos con el caso de grado 0.

Los recorridos siguen un patrón no distinto al resto de modelos. BFS sesga hacia abajo con su anchura, visualizado en la figura 62 y DFS sesga hacia arriba, con bastante más varianza en las profundidades que llega a revisar. En la figura 63 están ilustrados los rangos que pudo recorrer antes de “morderse la cola”. El *outlier* aquí es el algoritmo 3. En la mayoría de los tipos de redes, mejor dicho en los otros cuatro modelos, este procedimiento generaba un sesgo negativo. Aquí, en contra el pronóstico, la figura 61 presenta un sesgo hacia valores mayores. Una posible explicación es que, dada la naturaleza aleatoria, se tienen más posibilidades que DFS de “chocar” con un nodo ya revisado.

Robustez — Previamente adelantado, la robustez no es el punto fuerte de este modelo. La principal diferencia del impacto que tuvieron los algoritmos de eliminación es, como se ve en las figuras 33 y 60, el mayor sesgo y menor varianza de los resultados. Incluso, para el algoritmo 2 con n grande, los valores son negativos. Hecho que, analizando según el significado de f_c , no tiene sentido. Se computa un número menor a 0 porque la componente gigante ya esta irremediablemente rota. En otras palabras, no hay fracción de nodos a eliminar, pues la red ya se encuentra fragmentada.

El algoritmo 3, como se discutió en la distribución de grado, generó recorridos muy similares a lo largo de sus iteraciones. No solo se vio afectado el grado por esto, sino que el resto de propiedades. Los gráficos de violín de la figura 61 muestran distribuciones muy parecidas. Dichos violines no representan redes de una misma magnitud, pero aún así mantienen un reparto análogo independiente de n .

Las figuras 62 y 63 permiten visualizar que los valores promedio de BFS y DFS, respectivamente, convergen a valores similares a medida que aumenta n . BFS, prácticamente sin varianza, se diferencia de la dispersión vista en DFS. Como se mencionó, DFS puede avanzar largos trechos sin tener que devolverse a una vecindad ya revisada. El largo de este tramo es susceptible al azar de la elección del primer vecino a revisar ⁵⁹. Sumando a esto a que las redes “lineales” o profundas son menos robustas, explica la dispersión de los resultados.

6.2. Redes Reales

La inclusión de redes reales en este trabajo se enfocó bajo varios objetivos. El primero fue ver si los patrones observados en la experiencia con redes generadas volvían a surgir al alterar topologías reales. Asimismo, el siguiente objetivo se enfocó en no solo ver si una estructura real se comportaba de manera similar, sino también estudiar la conducta de la red al tener distintas magnitudes. Anteriormente se mencionó que las leyes de potencia eran de los modelos más comunes en los fenómenos representables por redes. Por esto se eligió como distribución en la redes reales a escoger.

Sin embargo, este no es el único modelo seleccionado. Restricciones físicas, principalmente de posición o locación, son responsables por una distribución que también se ve con frecuencia. Estas limitantes suelen ser la causa de que una red tenga una distribución de grado exponencial. Dicho modelo, podría decirse que es una reducción, a grandes rasgos, de una ley de potencia. Los *hubs* en una exponencial no llegan a ser los verdaderos centros que son su contra parte en una *scale-free*. Principalmente, su grado no cumple con ser ordenes de magnitud mayor que el del grueso de la red.

Para esta sección, no se realizará un análisis tan exhaustivo como ocurrió con las redes generadas. La principal razón es que, dada las limitaciones de tiempos de ejecución, las repeticiones de los experimentos fueron muchas menos. Por consiguiente, el efecto del azar no es despreciable en los resultados. De todas formas, se destacarán las mayores diferencias entre lo visto y lo esperado a partir de lo ya rescatado en la sección anterior.

6.2.1. Red de Interacción de Proteínas

Mirada general — Esta red en particular podría considerarse que está en un punto medio entre una ley de potencia y una exponencial. Esto, a primera vista, porque, teniendo en cuenta la magnitud de la red, existen nodos que podrían ser considerados *hubs* ⁶⁰. A partir de esto surge una tangente sobre qué se considera una ley de potencia o no.

⁵⁹En la práctica, esto depende del proceso de generación de la red y de como se ordena la lista de vecinos de cada nodo. Además, el vértice de partida siempre es aleatorio.

⁶⁰Como nota aparte, múltiples redes de interacción de proteínas siguen leyes de potencia. [Lucchetta *et al.*, 2023]

Se mencionó en el **Marco Conceptual** que las leyes de potencia tienen un parámetro γ , que define el exponente de la curva de la distribución de grado. Junto a esto, un rango de valores comunes para redes *scale-free* reales fue determinado: $2 < \gamma < 3$. Calcular este número para una distribución real puede ser una tarea compleja. No obstante, existe múltiples métodos para estimar este valor. Uno de estos es mediante la pendiente de la recta que forma el aplicar logaritmo a ambos de sus ejes. Este método, podría argumentarse, que es de los más imprecisos entre los disponibles. No obstante, para los efectos prácticos de lo que se busca responder, esta técnica basta. El exponente calculado para esta red es de $1,9 \pm 0,1$. Esto ayuda a evidenciar que la red cerca de un punto crítico.

Retornando del paréntesis, las discrepancias más notorias, comparadas con BA y el comportamiento general con el resto de modelos, ocurren en las métricas de caminos y en el coeficiente de *clustering*.

Coefficiente de *clustering* — Para empezar, la red, originalmente, tiene una pequeña cima en el valor 1, por lo que existen varios nodos que completan todos, o la mayoría, de sus triángulos disponibles. Estos detalles que se diferenciaron de las redes con topología similar, BA, sin duda surgen del contexto real del fenómeno representado. Este hecho es una de las mayores diferencias entre generadas y reales. Como consecuencia, para los nodos que tienen un *clustering* alto, pero no 1, la eliminación, o no consideración por recorrido, de los vecinos que no cerraban los triángulos que faltaban hace que el valor de la métrica suba. Por esto, en todas las figuras se puede ver una cima más acentuada en valores altos comparando con el original.

Este fenómeno ocurrirá en las próximas redes *scale-free*. E incluso en las que siguen una distribución exponencial. En múltiples y variados entornos reales, en especial redes sociales, existe una concentración de *clustering* alto no menor vista en nodos de grado bajo [Varga, 2015]. Por ejemplo, es probable que los amigos de una persona sean amigos entre sí, cuando ninguno de ellos podría considerarse un *hub* dentro del contexto estudiado. Dada la formación de una red BA, e incluso otros modelos similares, el poco *clustering* se concentrará solo en los *hubs*. Esta es de las principales discrepancias con las redes generadas y, más adelante en redes que siguen una ley de potencia, se discutirá más a fondo.

Diámetro y Camino más corto promedio — Exceptuando las figuras 66 y 67, que muestran los algoritmos de recorrido con revisión de vecindades en alguno de sus pasos (el trabajo con vecindades termina acortando caminos), existe poco sesgo respecto a los largos de las rutas. Esto incluso cuando se elimina una buena porción de las conexiones. Es posible que, dado el contexto, la red tenga una buena interconectividad que mantenga los caminos cortos. Por ejemplo, existen mecanismos redundantes en las interacciones de genes [Kafri *et al.*, 2008]. Otra evidencia está presente en la robustez de la figura 34. Aún eliminando una gran porción, la robustez baja poco.

6.2.2. Red de captura de Campaños

Mirada general — La siguiente red, evidenciada por su distribución de grado acampanada en valores bajos, sigue una exponencial ⁶¹. Como pasó en el caso anterior, se necesita conocer el contexto de la red para dar sentido a su topología. Una de las causas de una distribución exponencial, mencionada previamente, son las restricciones del mundo físico. Para esta red, se podría argumentar que las trampas que definen las relaciones entre miembros de la red están distribuidas espacialmente. Lo que, a su vez, limita a que los roedores fueran capturados solo por trampas que estuvieran en sus respectivos territorios.

De hecho, el impacto de los algoritmos sobre las propiedades es considerablemente parecido al presenciado en la red anterior. Una posible causa yace en la cercanía a una distribución exponencial que tiene la red de proteínas. Asimismo, el grafo mantiene magnitudes similares en nodos y conexiones. Ya que el efecto de la simulación de falta de información es similar, a continuación se destacarán las diferencias más destacables con la red anterior.

Coefficiente de *clustering* — Se pueden ver dos colinas en la distribución de *clustering*. Enfocando el criterio de relación como una propiedad transitiva, cuando los vecinos de un campañol son capturados por una misma trampa, es posible que sus territorios se superpongan. Esto llega a una probabilidad no menor que se formen triángulos completos. Hecho que se acentúa cuando, eliminando elementos, las colinas en valores altos de las figuras 35 y 68 aumentan cuando n no llega a ser la mitad de la red.

Se ve el efecto que se tuvo con la red de proteínas, con la diferencia de que esta red tiene una mayor concentración original de valores altos. A pesar de no ser una ley de potencia, se puede ver el fenómeno de *clustering* alto.

6.2.3. Red Eléctrica en Estados Unidos

Mirada general — De la redes reales estudiadas, esta es la última exponencial. Si bien, de las que comparten su distribución, es la más masiva. Nuevamente, se ve una fuerte limitación en el espacio físico. La propagación de estaciones eléctricas está subyugada a la localización de poblaciones o industrias. Este hecho puede justificar la distribución exponencial de grado.

Exceptuando la distribución de grado, y el *clustering*, los resultados de esta red asemejan al visto en el modelo Grid. Teniendo en mente lo que esta red representa, hace sentido la semejanza a una grilla. Por lo menos a un grafo con pocos, o ningún, cruce.

Coefficiente de *clustering* — Como se ha visto, la formación de triángulos completos es más abundante en la vida real que en los modelos generativos vistos. Al igual que en la red de proteínas, una colina en *clustering* alto destaca en las figuras. En el contexto de la red, al

⁶¹Cabe recordar que las figuras tienen una estimación de tendencia *kde*. Una distribución exponencial no tiene una campana.

igual que en el mundo de la interacción entre proteínas, también existen mecanismos de redundancia. Crear un árbol a partir de la red eléctrica permitiría que un solo fallo corte el grafo en dos partes.

Métricas de Caminos y Robustez — Ligado a la última afirmación, las métricas de caminos y la robustez de la red tienen un comportamiento distinto a la red exponencial anterior. Aquí es donde parte la semejanza a las grillas. Aunque exista redundancia, esta no es todopoderosa. Siguen existiendo puntos débiles en el grafo, cuya posible falla puede inhabilitar partes de la red. Evidenciando esto, la figura 36 muestran que aplicando un n alto, el diámetro y el camino más corto promedio hunden su media. Sumando el gráfico de robustez, se puede argumentar que, al eliminar suficientes elementos, la red se comienza a fragmentar. El largo de las rutas disminuye porque ya no es posible acceder a nodos que, con su lejanía, mantenían el diámetro en el valor original o cercano.

Como contra parte, las figuras 74 y 75, con algoritmos que revisan localidades eventualmente, permiten rescatar la interconectividad redundante local. Esto visto en los posibles valores altos de robustez.

6.2.4. Red Social Anybeat

Mirada general — Ahora pasando a la leyes de potencia, las siguientes redes ya pasan los diez mil miembros. En contraste, la repetición de los experimentos efectuados fue menor. Mantener los tiempos de ejecución bajo el umbral de lo práctico fue el motivo de esta reducción. Previamente se hizo un breve comentario fue hecho sobre el motivo de elegir estas redes, principalmente por su ubicuidad. No obstante, también ayudan a validar los resultados que entregaron las redes generadas.

Como se adelantó en la red de proteínas, BA y otros modelos *scale-free* no llegan a representar del todo la realidad. Simular este tipo de redes es más complejo que solo generar una ley de potencia como distribución de grado. Como respaldo, por ejemplo, ya se vieron diferencias en el coeficiente de *clustering*. Fenómeno que reaparecerá en las siguientes redes.

Coefficiente de *clustering* — En la disciplina de ciencia de redes se suele categorizar la topología, o estructura general, por la distribución de grado. Como se ha visto en las redes reales analizadas, el enfocarse solo en el grado genera categorías cuyos miembros no siempre siguen las misma reglas. Representado en todas las figuras de esta red, 37, 76 hasta 79, el *clustering* sigue el fenómeno ya descrito y citado en [Varga, 2015]. Además, en el contexto de la experiencia, el efecto de los algoritmos de simulación de falta de información también es análogo al previamente descrito. Considerando la notable diferencia entre la cantidad de hojas, y/o nodos de grado bajo, y los *hubs*, visualizada en los gráficos de grado, es más lejana la conclusión de que el *clustering* alto es causa de los *hubs*, como lo es en BA.

Añadiendo a los análisis hechos en las redes pasadas, los algoritmos BFS y DFS hacen notar

que existen vecindades con alta concentración de *clustering*, valga la redundancia. Más claro en la figura 78, la colina en valores altos crece considerablemente al tener un enfoque local. Por su parte, el algoritmo 3 no lo presenta tan evidente, pero aún así sesga la media hacia la derecha. Si el *clustering* se asemejara a BA, es decir que los valores altos dependan de los *hubs*, no se verían los efectos mencionados.

Métricas de Caminos y Robustez — No distinto a lo ocurrido en la red de proteínas, estas redes reales tienen una mejor interconectividad que la vista en las redes generadas. Hecho que puede asociarse al fenómeno del *clustering*. En otras palabras, la alta presencia de triángulos completos crean redundancia. Lo que a su vez, deja más opciones de rutas y menos puntos débiles en la integridad de la componente gigante.

Comparando con BA, es notable el menor sesgo presente en las métricas de camino de los algoritmos de eliminación. Figuras 32 y 56 de BA contrastadas con 37 y 76 para esta red. La mencionada interconectividad vuelve a surgir en los recorridos, donde las figuras 77, 78 y 79 presentan sesgos exclusivamente negativos.

Ya se argumentó una vez que las redes *scale-free* eran de las más robustas. En este caso, las reales son aún más íntegras que sus compañeras libres de escala generadas. Por ejemplo, BA puede presentar valores “abismales” al aplicar caminos aleatorios, figura 57. Independiente del procedimiento que se le aplique a esta red social, ninguno logra hacer una muesca considerable en su robustez.

6.2.5. Red Social Google Plus

Mirada general — Siendo la más masiva vista en este trabajo, esta última red mantiene fuertes similitudes con la anterior, puesto que ambas representan redes sociales. Como comentario preliminar, este grafo tiene casi el doble de miembros, pero el grado máximo visto está alrededor de la mitad en la red Anybeat. La mayoría de las propiedades tuvieron una conducta análoga en ambas redes. A primera vista, las figuras no parecen ser muy similares. Sin embargo es menester recordar que las formas de los violines son susceptibles al rango de valores que tiene el eje vertical. A pesar de esto, se dijo que específicamente que la mayoría de las propiedades sigue un patrón similar, puesto que entre ellas destaca una con un comportamiento distinto.

Asortatividad — Que la red tenga más miembros, pero presente *hubs* de menor grado que la red anterior, da un pista a la diferencia en estructura que tienen entre sí. Lamentablemente, el *trade-off* entre cantidad de experimentos versus tiempo de ejecución perjudica a la confianza que se le pueda tener a las varianzas de estas métricas. Aún así, volviendo a la comparación, el valor original de asortatividad ya es menor que en la red Anybeat. Con todo lo comentado, se puede argumentar que la estructura de la red se basa en más conexiones desde hojas hacia *hubs* que entre solo hojas. También se puede corroborar con la menor colina en *clustering* alto.

En la figura 82 se puede ver que, aun con la revisión de vecindades de BFS, que debería poder encontrar los mencionados grupos de amigos que no son *hubs*, la asortatividad se sesga negativamente con n menor a la mitad de la red. Otro caso curioso fue el algoritmo 3, figura 81, que, a diferencia de la leyes de potencia reales vistas, aumentó la media de la métrica. Eso se puede relacionar más con el modelo BA, que siguió un patrón similar. El análisis sobre como esta similitud puede ayudar a entender el ambiente interno de la red social escapa a este trabajo.

6.3. Recapitulación

A lo largo de esta sección se han hecho múltiples análisis cualitativos a los resultados vistos. Por su lado, las limitaciones que el experimento tuvo respecto a las variables mermó la certeza de un posible análisis cuantitativo. Más comentarios sobre esto se encontrarán en la siguiente sección, en **Conclusiones**. Una gran cantidad de datos fue generada a partir de la experiencia, empero revisar a fondo cada uno de ellos sería una tarea monumental. En otras palabras, se tendría que estudiar cada cruce red-algoritmo- n . Por esto, se presentaron los *insights* más relevantes respecto a los patrones visualizados y sus posibles causas.

En principal enfoque dado fue intentar discernir las causas y razones de los impactos vistos en los resultados. La motivación a utilizar redes de magnitudes similares (para las generadas) fue usar la topología como principal variable de experimentación. A partir de esto se buscó un balance entre dicha magnitud de la redes, cantidad de nodos y conexiones, la cantidad de grafos que se generarían y el tiempo de ejecución de la experiencia. El primer parámetro define qué tan marcada será la topología esperada bajo ciertas excepciones. BA puede parecer una red aleatoria cuando tiene pocos nodos y, por otro lado, Grid mantiene la misma estructura independiente de su magnitud. El segundo, busca controlar el error aleatorio resultante de una generación azarosa. Nuevamente, con excepciones como Grid. Y el último factor de balance es el tiempo que tomaría el trabajo, el cual debió regularse para hacer la experiencia factible dentro de plazos establecidos. Por último, se escogieron distintas redes reales para, o corroborar lo visto, o detectar discrepancias en lo observado. Asimismo, la oportunidad fue utilizada para un “test de estrés” sobre los experimentos al aumentar las magnitudes con las que se trabajaría.

La siguiente fase fue el desarrollo, aplicación y ejecución de los experimentos. Una vez que se tenían los resultados se dio el enfoque de “ver que pasa por detrás” según los parámetros y variables conocidas. En otras palabras, se buscó entender el cómo y el por qué del efecto visto en la falta de información. Así, fue posible identificar los impactos que tienen distintos tipos de falta de información (simulada a través de los algoritmos) según una mirada general a lo largo de todas las redes, o enfocando el análisis a una topología en específico.

6.3.1. Resumen Redes Generadas

A continuación, tablas que resumen los resultados serán presentadas. Por espacio, se abrevia sesgo como “S” y varianza como “V”. El sesgo se representará con la dirección del cambio en el valor promedio de la métrica y la varianza con una comparación de magnitud.

Modelo ER Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	S – V <	Sin efectos Notables	S ~ 0 V >	S + V >	S + V >	S – V ~
Eliminación de Conexiones	S – V <	Sin efectos Notables	S ~ 0 V >	S + V ~	S + V ~	S – V ~
Camino Aleatorio	S – V <	S ~ 0 V >	S + V >	S + V >>	S + V >>	S – V >>
Exploración BFS	S – V <	S ~ 0 V >	S – V >	S ± V ~	S ± V >	S – V >
Exploración DFS	S – V <	S ~ 0 V >	S – V ~	S + V >	S + V >	S – V >

Tabla 11: Resumen resultados para modelo ER.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

Modelo SBM Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	S – V <	Sin efectos Notables	S ~ 0 V >	S + V ~	S + V ~	S – V ~
Eliminación de Conexiones	S – V <	Sin efectos Notables	S ~ 0 V >	S + V >	S + V >	S – V >
Camino Aleatorio	S – V <<	S ~ 0 V >	S + V >>	S + V >>	S + V >>	S – V >>
Exploración BFS	S – V <	S ~ 0 V >	S – V >	S – V >	S ~ 0 V >	S – V >
Exploración DFS	S – V <	S ~ 0 V >	S – V ~	S ~ 0 V >	S ~ 0 V >	S – V >

Tabla 12: Resumen resultados para modelo SBM.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

Modelo WS Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	S – V >	S ~ 0 V >	S + V ~	S + V >	S + V ~	S – V ~
Eliminación de Conexiones	S – V >	S – V >	S ~ 0 V >	S + V >	S + V >	S – V ~
Camino Aleatorio	S – V >	S ~ 0 V >>	S + V >>	S + V >>	S + V >>	S – V >>
Exploración BFS	S – V >>	S + V >>	S ~ 0 V >>	S – V ~	S – V >	S – V >>
Exploración DFS	S – V >>	S + V >	S + V >	S + V >>	S + V >>	S – V >

Tabla 13: Resumen resultados para modelo WS.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

Modelo BA Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	Sin efectos Notables	Sin efectos Notables	S ~ 0 V >	S + V >	S + V >	S – V >
Eliminación de Conexiones	Sin efectos Notables	Sin efectos Notables	S ~ 0 V >	S + V >	S + V ~	S – V ~
Camino Aleatorio	S ~ 0 V <	S ~ 0 V >	S + V >>	S + V >>	S + V >>	S – V >>
Exploración BFS	Sin efectos Notables	S ~ 0 V >	S – V >>	S – V ~	S – V >>	S ~ 0 V >>
Exploración DFS	Sin efectos Notables	S ~ 0 V >	S – V >>	S + V >	S – V >>	S – V >>

Tabla 14: Resumen resultados para modelo BA.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

Modelo Grid Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	S – V >>	No Aplica	Comparación poco Fiable	S ± V >	S – V >	S – V ~
Eliminación de Conexiones	S – – V >>	No Aplica	Comparación poco Fiable	S – V ~	S – V ~	S – – V ~
Camino Aleatorio	S – V >>	No Aplica	Comparación poco Fiable	S – V >>	S – V >>	S – V >>
Exploración BFS	S – V >>	No Aplica	Comparación poco Fiable	S – – V ~	S – – V ~	S – V ~
Exploración DFS	S – V >	No Aplica	Comparación poco Fiable	S + V >>	S – V >>	S – V >>

Tabla 15: Resumen resultados para modelo Grid.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

La asortatividad tiene una comparación poco fiable por lo discutido acerca del cómputo de la métrica versus su valor teórico = 1.

6.3.2. Resumen Redes Reales

La principal consideración a tener en esta sección es que la comparación de varianzas para los valores que no son distribuciones (asortatividad, diámetro, entre otros) no tiene mucho sentido. El motivo de esto es que el valor original es único, así que siempre tendrá varianza 0. Por consecuencia, la comparación siempre será >.

Red 1 Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	S ~ 0 V <	S ~ 0 V <	S ~ 0 -	S ~ 0 -	S + -	S – -
Eliminación de Conexiones	S ~ 0 V <	S ~ 0 V <	S + -	S + -	S + -	S – -
Camino Aleatorio	S ~ 0 V <	S ~ 0 V >	S + -	S – -	S – -	S – -
Exploración BFS	S ~ 0 V >	S ~ 0 V >	S – -	S – – -	S – -	S ~ 0 -
Exploración DFS	S ~ 0 V ~	S ~ 0 V >	S – -	S – -	S – -	S ~ 0 -

Tabla 16: Resumen resultados para Red 1.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

Red 2 Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	S – V <	S – V <	S + -	S + -	S + -	S – -
Eliminación de Conexiones	S ~ 0 V <	S – V <	S – -	S + -	S + -	S – -
Camino Aleatorio	S ~ 0 V <	S ~ 0 V <	S + -	S – -	S – -	S – -
Exploración BFS	S ~ 0 V ~	S + V ~	S – -	S – -	S – -	S – -
Exploración DFS	S ~ 0 V ~	S + V ~	S – -	S – -	S – -	S ~ 0 -

Tabla 17: Resumen resultados para Red 2.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

Red 3 Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	S – V <	S ~ 0 V <	S ~ 0 -	S ± -	S ± -	S – -
Eliminación de Conexiones	S ~ 0 V <	S ~ 0 V <	S ~ 0 -	S + -	S + -	S – -
Camino Aleatorio	S ~ 0 V <	S ~ 0 V ~	S + -	S – -	S – -	S – -
Exploración BFS	S ~ 0 V ~	S ~ 0 V >	S – -	S – -	S – -	S ± -
Exploración DFS	S ~ 0 V ~	S ~ 0 V >	S – -	S – -	S – -	S – -

Tabla 18: Resumen resultados para Red 3.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

Red 4 Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	$S \sim 0$ $V <$	$S \sim 0$ $V <$	$S -$ -	$S \sim 0$ -	$S +$ -	$S -$ -
Eliminación de Conexiones	$S \sim 0$ $V <$	$S \sim 0$ $V <$	$S -$ -	$S +$ -	$S +$ -	$S -$ -
Camino Aleatorio	$S \sim 0$ $V <$	$S +$ $V >$	$S -$ -	$S -$ -	$S -$ -	$S -$ -
Exploración BFS	$S \sim 0$ $V <$	$S +$ $V >$	$S -$ -	$S -$ -	$S -$ -	$S -$ -
Exploración DFS	$S \sim 0$ $V <$	$S +$ $V >$	$S -$ -	$S -$ -	$S -$ -	$S -$ -

Tabla 19: Resumen resultados para Red 4.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

Red 5 Algoritmo	Distribución de Grado	Distribución de <i>Clustering</i>	Asortatividad	Diámetro	Camino más Corto prom	Robustez
Eliminación de Nodos	$S \sim 0$ $V <$	$S \sim 0$ $V <$	$S \sim 0$ -	$S \sim 0$ -	$S \sim 0$ -	$S -$ -
Eliminación de Conexiones	$S \sim 0$ $V <$	$S \sim 0$ $V <$	$S -$ -	$S \sim 0$ -	$S +$ -	$S -$ -
Camino Aleatorio	$S \sim 0$ $V <$	$S +$ $V >$	$S +$ -	$S \pm$ -	$S -$ -	$S -$ -
Exploración BFS	$S \sim 0$ $V <$	$S \sim 0$ $V >$	$S -$ -	$S -$ -	$S -$ -	$S -$ -
Exploración DFS	Sin efectos Notables	$S \sim 0$ $V >$	$S -$ -	$S -$ -	$S -$ -	$S -$ -

Tabla 20: Resumen resultados para Red 5.

La comparación es entre original versus resumen de alteradas. Fuente: Elaboración propia.

CAPÍTULO 7

CONCLUSIONES

7.1. Comentarios Preliminares

Antes de comentar los objetivos y resultados de la experiencia como tal, cabe, nuevamente, mencionar las limitaciones de este trabajo. Mencionado en la sección de **Resultados**, el tiempo fue el principal factor acotante de este trabajo, específicamente, los tiempos de ejecución del experimento.

Ya sea de generación de redes, su alteración o posterior extracción de propiedades, varios algoritmos utilizados en el flujo de trabajo tenían complejidades temporales de orden cuadrático o peor ⁶². Siendo este n , en múltiples casos, la cantidad de nodos o conexiones. En las redes generadas, estos valores rondaban $|V| \sim 3000$ y $|E| \sim 24000$ ⁶³. Estas magnitudes se repetían en las 150 redes a través de los cinco modelos y de sus cinco versiones (una original y cinco alteradas ⁶⁴). Todo este cómputo, el flujo de trabajo generación - extracción - alteración - extracción, apilado uno sobre otro, fue el responsable de los largos tiempos de ejecución. Esta es la razón de no aumentar más la magnitud y cantidad de redes. Así, se tuvo que balancear el error aleatorio generado por el tamaño de la muestra versus la extensión de tiempos de cómputo.

Como aparte, los tiempos presentados fueron de la última ejecución exitosa. Si algo salía mal o existía un error en el código, se tenía que pasar estos tiempos de nuevo para *debuggear*.

En retrospectiva, las variaciones de valores en la data se mantuvo dentro de márgenes aceptables. Se mantuvo baja en los casos donde la aleatoriedad tenía poco impacto, como en las distribuciones de grado. Y, esperablemente, permaneció alta en los casos donde el azar de los algoritmos alteradores tenían un alto impacto. Por ejemplo, la asortatividad depende mucho de si los nodos sobrevivientes, bajo alguna alteración, mantienen magnitudes similares o dispares de grado. Los gráficos lograron capturar líneas de tendencia claras. Aunque se aplicó un efecto de *smoothing* ⁶⁵ a los *kdeplot*, la intensidad de esta alteración fue leve y puramente estética ⁶⁶, es decir, sin esta, la pauta sigue siendo evidente.

Para cerrar este primer comentario conclusivo, cabe mencionar las posibles optimizaciones que se podrían haber aplicado para mejorar los tiempos de ejecución. Por ejemplo, las redes eran generadas, desde cero, cada vez que se ejecutaba el flujo de trabajo. Una alternativa

⁶²Por ejemplo, el de generación de redes ER.

⁶³Esto en ER, SBM y WS.

⁶⁴Son cuatro algoritmos, pero el 4 aplica dos alteraciones, BFS y DFS.

⁶⁵Este es un efecto que “suaviza” el estimado de tendencia. Su intensidad se mide con un parámetro de *bandwidth*. Link de documentación.

⁶⁶Sin este efecto la línea de tendencia salta, de manera discreta, de 0 hasta los valores pico.

podría haber sido, tener un *dataset* de redes ya generado preliminarmente y al comenzar el flujo, simplemente, cargarlas a memoria en vez de generarlas. Otro ejemplo, ligado a la alternativa recién propuesta, es un análisis exhaustivo del *dataset* de redes a procesar. Esto para elegir los algoritmos que más se ajusten a las cualidades de las redes. En particular, el algoritmo de Dijkstra tiene un desempeño “estable” a lo largo de las posibles densidades de los grafos sobre los que trabaja. En cambio, el algoritmo de Bellman-Ford varía su desempeño según la configuración de la red [Upadhyay, 2023]. Estas y otras mejoras podrían haber reducido los tiempo de ejecución y, quizás, permitir aumentar el tamaño de las muestras y de las redes en sí.

Otra anotación sobre posibles mejoras a la experiencia trata sobre las componentes conexas de los grafos. Muchas redes reales, y la mayoría de los fenómenos en la realidad que representan, tienen múltiples componentes conexas. A su vez, una gran fracción de este grupo tienen una componente gigante principal y varias más pequeñas. En especial, las topologías *scale-free* tienen este comportamiento. Por sí solo, esto no representa un contratiempo, pero, al aplicar los algoritmos, puede existir casos problemáticos. Por ejemplo, si el nodo aleatorio que comienza un recorrido, ya sea un camino estocástico o determinista, cae en una de las componente pequeñas. Si se da este caso, el camino puede terminar prematuramente y no ser una buena representación de la red.

En este trabajo, por simplicidad, se extrae la componente gigante de las redes reales y se trabaja solo con esta. Así se logra evitar choques con los algoritmos de recorrido. Adicionalmente, reduce los problemas con la extracción de propiedades, tal como el camino más corto entre nodos. Por sí mismo, este recurso puede considerarse como simular falta de información por sí mismo. El problema yace en que no se consideró el impacto de la eliminación de las componentes pequeñas.

Dado esto, existe la posibilidad de mejorar el manejo de subgrafos desconexos. Por ejemplo, permitir “saltos” en los caminos para pasar de una componente a otra cuando ya se hayan revisado todos los nodos locales. Un contraargumento contra esto es que no es muy realista cuando el recorrido simula un muestreo a través de conexiones. Para cerrar este segundo comentario, como anotación, dada su formulación, varios modelos no pueden generar componentes desconexas, por lo que esta problemática no se ve tanto en las redes generadas. Es menos probable pero no imposible, así que también existen casos, en especial en ER, donde se encuentran con pequeños subgrafos separados de la componente gigante. Un nuevo enfoque respecto a las componentes de las redes también beneficiarían a la experimentación con redes generadas.

Similar al trato de componentes, se puede dar nuevos enfoques a trabajos futuros con redes dirigidas. Cómo se mencionó en la sección de **Marco Experimental**, al incluir las redes reales, algunas eran dirigidas. Para simplificar y estandarizar el trabajo, respecto a las redes generadas, la decisión fue por tomar la versión no dirigida de la red. Esto se logró dejando una conexión entre nodos si es que existe al menos una arista en cualquier dirección en su versión original. También se discutió que esto disminuiría la cantidad de conexiones de la red tal y como se extrajo de la fuente. Nuevamente, esto podría considerarse como falta de

información, pero en la experiencia desarrollada no se consideró su impacto.

Adaptar los algoritmos al manejo de redes dirigidas puede ser un posible enfoque a su manejo. Tal como el algoritmo de Dijkstra, varios de los métodos utilizados para la extracción de métricas tienen sus versiones para tratar la dirección de las conexiones. Esto por si solo sería generar un análogo del experimento propuesto, si bien considerando el sentido de las conexiones. Un planteamiento más interesante sería utilizar el hecho de transformar una red dirigida a una no dirigida para analizar el efecto de la falta de información. Así, se podría comparar distintas métricas cuando la red tiene, o no, dirección.

Como última anotación previa a comentar los objetivos, la capacidad de memoria RAM también fue un limitante del alcance de la experiencia. Trabajar con redes grandes, y en especial densas, ocupa bastantes recursos. Quizás sea posible cargar un grafo con millones de nodos por si solo, pero al momento de trabajarlo pueden surgir problemas. Tanto como complejidades temporales, los algoritmos también tienen complejidades espaciales. Algunos procedimientos necesitan, por ejemplo, espacio de $O(n^2)$, siendo n la cantidad de nodos.

Lamentablemente, la forma de solventar esta problemática conlleva costos monetarios, ya sean planes de pago en Google Colab, para utilizar una máquina con más recursos, o en la adquisición de *hardware*. Aún así, existen variantes de los algoritmos utilizados que sacrifican tiempo de ejecución por utilizar menos espacio. Como los procesos sobre redes grandes ya demoran tiempos considerables, hacer que demoren aún más no suena como una opción deseable. Bajo recursos limitados se debe encontrar un equilibrio. En este trabajo y en cualquier ámbito que trabaje con datos suficientemente grandes, siempre están presentes este tipo de limitaciones donde debe existir un *trade-off* para hacer factibles las experiencias. En este caso particular, se sacrificó la magnitud de la redes a revisar.

7.2. Revisión de objetivos propuestos

Existe muchas formas de perder información, o no disponerla, cuando se esta modelando una red. Entre estas, pérdidas aleatorias, ataques direccionados, limitaciones de magnitud, incluso que el fenómeno estudiado esté oculto a propósito. Es complejo definir un modelo que englobe todas las formas de tener datos incompletos al momento de generar una red.

La investigación al inicio de este trabajo dio a luz distintos enfoques que ha tomado la literatura al momento de abarcar la falta de información. Varias veces se presentó el caso del muestro, o “sampling”, que no es una pérdida de información per se, pero aún así conlleva trabajar con una red parcial. A través de los algoritmos propuestos, una aproximación fue desarrollada para simular posibles ausencias de los miembros de un grafo. Como lo amerita su nombre, esta aproximación no es perfecta. A lo largo de la experiencia se encontraron distintas faltas y/o potenciales mejoras. Por ejemplo, la modificación hecha a los algoritmos BFS y DFS (el hecho de agregar conexiones entre nodos que tuvieran relación en la red original, pero que no fueran revisadas por el recorrido) puede no ser una simulación muy realista

de un muestreo tipo recorrido. No obstante, se decidió por evitar la generación de árboles al momento de hacer recorridos. Por su parte, una posible mejora que se planteó, mas no fue usada, fueron los distintos métodos de selección de los miembros de la red. Por simplicidad, y complejidad temporal, el criterio usado fue el aleatorio. Sin embargo, podría generar casos interesantes el hecho de seleccionar mediante métricas de centralidad, grado, entre otras.

Al momento de aplicar estos experimentos se llegó a otra encrucijada. Magnitud de la red, topologías a revisar, cantidad de redes, parámetros de generación para los modelos y parámetros de algoritmos de simulación (el mencionado n durante la discusión) fueron las principales variables de experimentación a considerar. Como es esperable, la escala de posibles combinaciones crecería exponencialmente si no se fijan algunas. Dado que el enfoque de la experiencia recae en revisar los efectos de las simulaciones en distintas topologías, fueron estas las dos variables seleccionadas.

En retrospectiva, la exclusión de la magnitud de la red como variable tuvo un dejo “agridulce”. Por un lado, haría crecer la cantidad de redes a trabajar, manteniendo una cantidad considerable para cada magnitud, y, por consiguiente, los tiempos de ejecución. Por el otro, daría una mejor mirada sobre las proporciones vistas en los resultados. Como ejemplo, dado el experimento ejecutado, no sería seguro exponer que el sesgo visto es proporcional, según algún número computable a través de los resultados, al n del algoritmo aplicado. En términos más concisos, no hay seguridad de decir que el grado promedio cae, por ejemplo, un cuarto cuando se elimina la mitad de la red. Ese análisis cuantitativo no sería del todo íntegro, ya que nada asegura que, cambiando la magnitud de la red, las proporciones, que en el ejemplo fue un cuarto por un medio, también cambien.

Ligado a lo anterior, la delimitación de los parámetros de generación también tienen el potencial de afectar dichas proporciones. Como se planteó en el **Marco Conceptual**, estos parámetros, excluyendo la variable de cantidad de nodos, puede modificar considerablemente la red, aun manteniendo la magnitud. Una instancia de esto es el parámetro p de ER. Este valor no toca la cantidad de nodos, sino la de conexiones, así modificando la densidad de la red resultante. Considerando esto, surge el mismo dilema que ocurrió con la cantidad de nodos por red. Tiene un impacto en los resultados, pero aumenta geométricamente las combinaciones.

En suma, las consideraciones que tuvieron que dejarse de lado, son un buen punto de partida para trabajo futuro. Quizás, enfocándose en una topología y en un algoritmo sería más realizable un experimento que logre rescatar un patrón cuantitativo entre las proporciones vistas en los sesgos producidos.

Volviendo del paréntesis de potenciales mejoras, se espera que este trabajo haya logrado más allá de una lista topología-algoritmo-efecto, más bien una metodología para tratar la falta de información. Expresado múltiples veces a lo largo del trabajo, la topología de una red es una de sus características definitorias más importantes. Y, así como los fenómenos de relación expresables con redes se encuentran en variadas disciplinas, son igual de diversas las topologías existentes. Quizás, la categorización más amplia de tipos de redes, al igual

a la presentada en la experiencia, no logre representar la afirmación anterior. Un ejemplo visto durante el experimento, fue la diferencia entre una red BA y una red real *scale-free*. Ambas caen en la categoría de seguir una ley de potencia como distribución de grado. No obstante, bajo una mirada más escrutadora, diferencias fundamentales son encontradas. Y, los resultados obtenidos evidenciaron discrepancias en los comportamientos de ambas redes. En resumen, la categorización por topología puede presentar “sub-categorías” que no deben ser ignoradas al momento de analizar un tipo de red.

La discusión tuvo el enfoque de buscar las causas, motivos y consecuencias de los comportamientos vistos en los cruces tipo de red - algoritmo aplicado. El principal motivo de esto fue, a través del ejemplo, delimitar y definir una posible metodología a aplicar cuando se está trabajando con falta de información en redes. Desafortunadamente, el planteamiento de un modelo matemático “*one size fits all*” no fue posible dada la estructura y formulación del experimento. Incluso solventando las restricciones temporales con, como se discutió previamente, una reformulación factible de la experiencia (enfocar en una topología, un algoritmo y variar la magnitud y densidad de la red) tampoco lograría representar cuantitativamente todos los modelos de red en existencia. En suma, se hicieron concesiones en pos de lograr una estrategia que permitiera el análisis de la mayor cantidad de combinaciones posibles.

Este trabajo rescató ejemplos de simulaciones de falta de información al aplicarla en distintas topologías. A su vez, un marco de comportamientos, patrones y causas fue descrito en busca de entender que es lo que pasaba “por detrás”. La expectativa yace en que este marco metodológico pueda ser replicable en otros contextos y modelos de redes, aún cuando no se hayan revisado en este estudio. Así, por ejemplo, si se tiene un fenómeno que no se conoce por completo y es representado por un grafo, al momento de aplicar un análisis estadístico sobre este ya se disponga de un indicio previo de la posible conducta que tendrá la red al completo. Ya sea un caso como conocer la dirección del sesgo que habrá entre los valores promedios reales versus observados. O, por el otro lado del espectro, saber que un tipo de muestreo tendrá tanta aleatoriedad involucrada que la variable original puede tener un valor completamente inesperado.

ANEXOS

7.1. Algoritmo de generación bloques SBM

En resumen, este algoritmo genera una cantidad de comunidades aleatoria (dentro del rango dado), a las cuales le asigna un número de miembros al azar. Basado en una respuesta del foro StackExchange [“david.pfx”, 2014].

Algoritmo 5 comunSBM(n , $minCom$, $maxCom$)

```

1: Parámetros: (Entero  $n$ , Entero  $minCom$ , Entero  $maxCom$ )
2:                                     ▷  $n$  es la cantidad total de nodos.
3:                                     ▷  $minCom$  y  $maxCom$  son el rango de tamaño de comunidades.
4:
5:  $result \leftarrow list()$ 
6:  $temp \leftarrow list()$ 
7:  $numComun \leftarrow random.randint(minCom, maxCom)$        ▷ Cantidad de bloques.
8:  $i \leftarrow 0$ 
9: while  $i < n$  do
10:    $temp.append(random.randint(\sqrt{n}, n))$ 
11:    $i \leftarrow i + 1$                                      ▷ Rango aproximado de tamaño de bloques.
12: end while
13:  $sum \leftarrow \sum(temp)$ 
14:  $i \leftarrow 0$ 
15: while  $i < n$  do
16:    $result.append(int(temp[i] * (n/sum)))$ 
17:    $i \leftarrow i + 1$                                      ▷ Los números generados en  $temp$  son escalados.
18: end while
19:  $result[-1] \leftarrow result[-1] + (n - \sum(result))$ 
20:                                     ▷ La diferencia se compensa en la ultima comunidad.
21: return  $result$ 

```

7.2. Algoritmo de generación probabilidades SBM

Este algoritmo genera la matriz de probabilidades del modelo de generación de redes SBM.

Algoritmo 6 probSBM(numComun, intraProb, interProb)

```

1: Parámetros:
2: (Entero numComun, Flotante intraProb, Flotante interProb)
3:                                     ▷ numComun es la cantidad de comunidades.
4:                                     ▷ intraProb es la probabilidad de conexión dentro de comunidades.
5:                                     ▷ interProb es la probabilidad de conexión entre de comunidades.
6:
7: result ← matrix(dim = [numComun, numComun])
8: i ← 0
9: while i < numComun do
10:   j ← i
11:   while j < numComun do
12:     if i == j then                                     ▷ Diagonal principal, caso misma comunidad.
13:       result[i][j] ← intraProb
14:     else                                                 ▷ Caso entre comunidades.
15:       result[i][j] ← interProb
16:     end if
17:     j ← j + 1
18:   end while
19:   i ← i + 1
20: end while
21: result ← HacerSimetrica(result)                         ▷ SBM necesita una matriz simétrica.
22: return result

```

7.3. Algoritmo BFS modificado

Algoritmo BFS con las modificaciones mencionadas.

Algoritmo 7 BFSMod(G, n)

```

1: Parámetros: (Grafo  $G$ , Entero  $n$ ) ▷  $n$  mide la cantidad de nodos a revisar.
2:
3:  $newG \leftarrow Graph()$ 
4:  $rev \leftarrow list()$  ▷ Lista de nodos revisados.
5:  $Q \leftarrow queue()$ 
6:  $i \leftarrow 0$ 
7:  $Q.enqueue(random.choice(G.nodes))$  ▷ Paso inicial.
8:
9: while not  $Q.empty()$  and  $i < n$  do
10:    $u \leftarrow Q.dequeue()$ 
11:    $newG.add\_node(u)$ 
12:    $rev.append(u)$ 
13:
14:   for  $neigh$  in  $G.neighbors(u)$  do
15:     if  $neigh$  not in  $rev$  then
16:       if  $i + 1 < n$  then ▷ Limita a  $n$  nodos.
17:          $Q.enqueue(neigh)$  ▷ Iteración del BFS.
18:          $i \leftarrow i + 1$  ▷  $i$  limita que no se revise el grafo completo.
19:       else
20:         break
21:       end if
22:     end if
23:   end for
24: end while
25:
26: for  $node$  in  $newG.nodes()$  do
27:   for  $neigh$  in  $G.neighbors(node)$  do
28:     if  $neigh$  in  $newG.nodes$  then ▷ Agregar conexiones a nodos ya considerados.
29:        $newG.add\_edge(node, neigh)$ 
30:     end if ▷ Conexiones repetidas no se consideran.
31:   end for
32: end for
33:
34: return  $newG$ 

```

7.4. Algoritmo DFS modificado

Algoritmo DFS con las modificaciones mencionadas.

Algoritmo 8 DFSMod(G, n)

```

1: Parámetros: (Grafo  $G$ , Entero  $n$ ) ▷  $n$  mide la cantidad de nodos a revisar.
2:
3:  $newG \leftarrow Graph()$ 
4:  $rev \leftarrow list()$  ▷ Lista de nodos revisados.
5:  $S \leftarrow stack()$ 
6:  $i \leftarrow 0$ 
7:  $S.push(random.choice(G.nodes))$  ▷ Paso inicial.
8:
9: while not  $S.empty()$  and  $i < n$  do
10:    $u \leftarrow S.pop()$ 
11:    $newG.add\_node(u)$ 
12:    $rev.append(u)$ 
13:
14:   for  $neigh$  in  $G.neighbors(u)$  do
15:     if  $neigh$  not in  $rev$  then
16:        $S.push(neigh)$  ▷ Iteración del DFS.
17:        $i \leftarrow i + 1$  ▷  $i$  limita que no se revise el grafo completo.
18:     end if
19:   end for
20: end while
21:
22: for  $node$  in  $newG.nodes()$  do
23:   for  $neigh$  in  $G.neighbors(node)$  do
24:     if  $neigh$  in  $newG.nodes$  then ▷ Agregar conexiones a nodos ya considerados.
25:        $newG.add\_edge(node, neigh)$ 
26:     end if ▷ Conexiones repetidas no se consideran.
27:   end for
28: end for
29:
30: return  $newG$ 

```

7.5. Detalles y Fuentes de Redes reales

Todas las redes fueron extraídas del sitio **Network Repository** [Rossi y Ahmed, 2015].

- Red 1: Red de interacción de proteínas.
 - Nombre archivo: `bio-yeast-protein-inter.edges`
 - Distribución de grado / Tipo de red: Ley de potencia / Exponencial.
 - Cantidad de Nodos: $1,8k$
 - Cantidad de Conexiones: $4,4k$
 - Fuente original: [Jeong *et al.*, 2001]
- Red 2: Red captura de campañoles
 - Nombre archivo: `mammalia-vole-bhp-trapping.edges`
 - Distribución de grado / Tipo de red: Exponencial.
 - Cantidad de Nodos: $1,7k$
 - Cantidad de Conexiones: $5,3k$
 - Fuente original: [Davis *et al.*, 2015]
- Red 3: Red eléctrica USA
 - Nombre archivo: `power-US-Grid.mtx`
 - Distribución de grado / Tipo de red: Exponencial.
 - Cantidad de Nodos: $4,9k$
 - Cantidad de Conexiones: $6,6k$
 - Fuente original: No especificada.
- Red 4: Red social Anybeat.
 - Nombre archivo: `soc-anybeat.edges`
 - Distribución de grado / Tipo de red: Ley de potencia.
 - Cantidad de Nodos: $12,6k$
 - Cantidad de Conexiones: $67,1k$
 - Fuente original: [Fire *et al.*, 2012]
- Red 5: Red social Google Plus.
 - Nombre archivo: `soc-gplus.edges`
 - Distribución de grado / Tipo de red: Ley de potencia.
 - Cantidad de Nodos: $23,6k$
 - Cantidad de Conexiones: $38,2k$
 - Fuente original: No especificada.

7.6. Resultados originales para ER

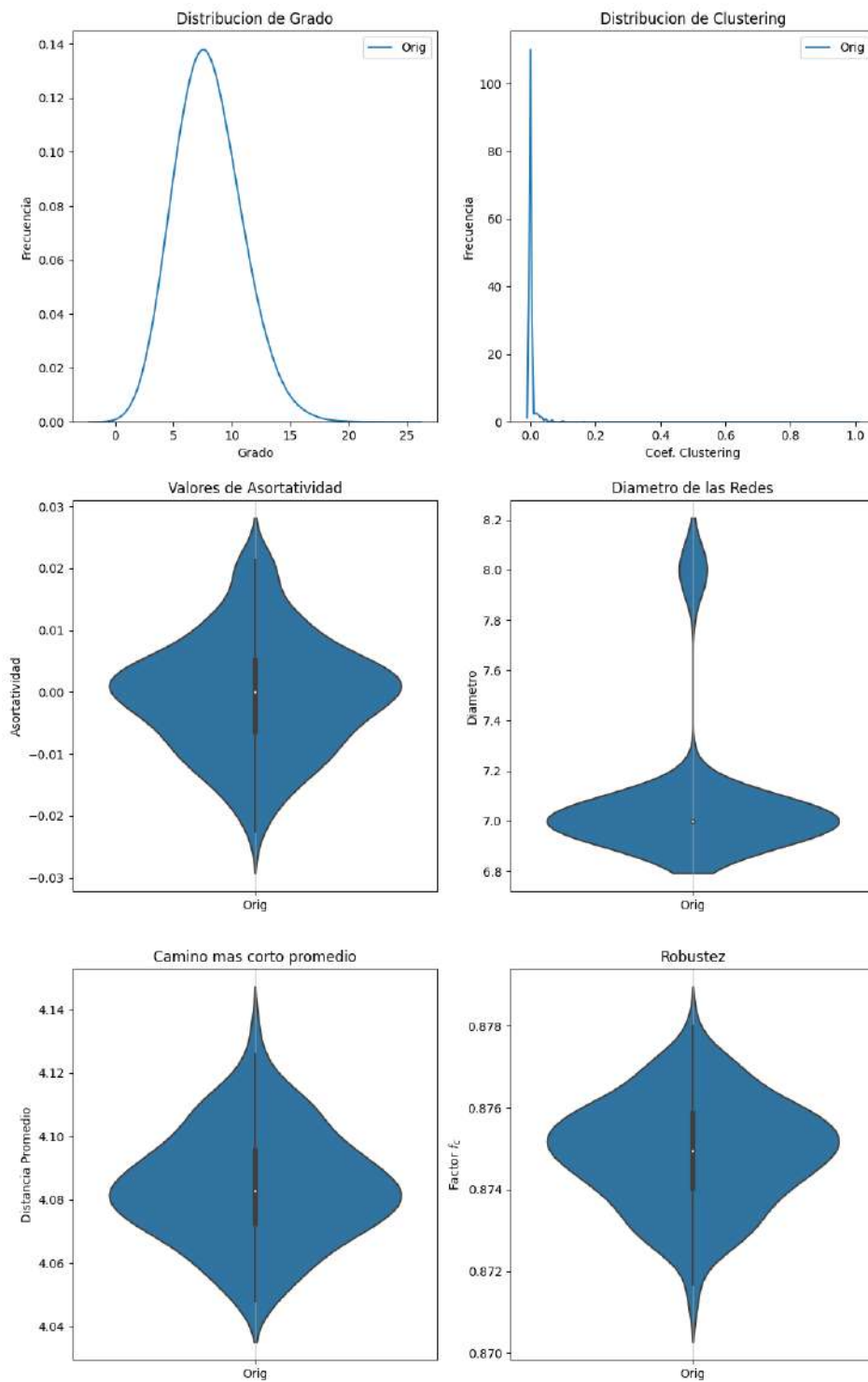


Figura 39: Distribuciones de resultados para redes ER originales. Completa.
Fuente: Elaboración propia.

7.7. Resultados originales para SBM

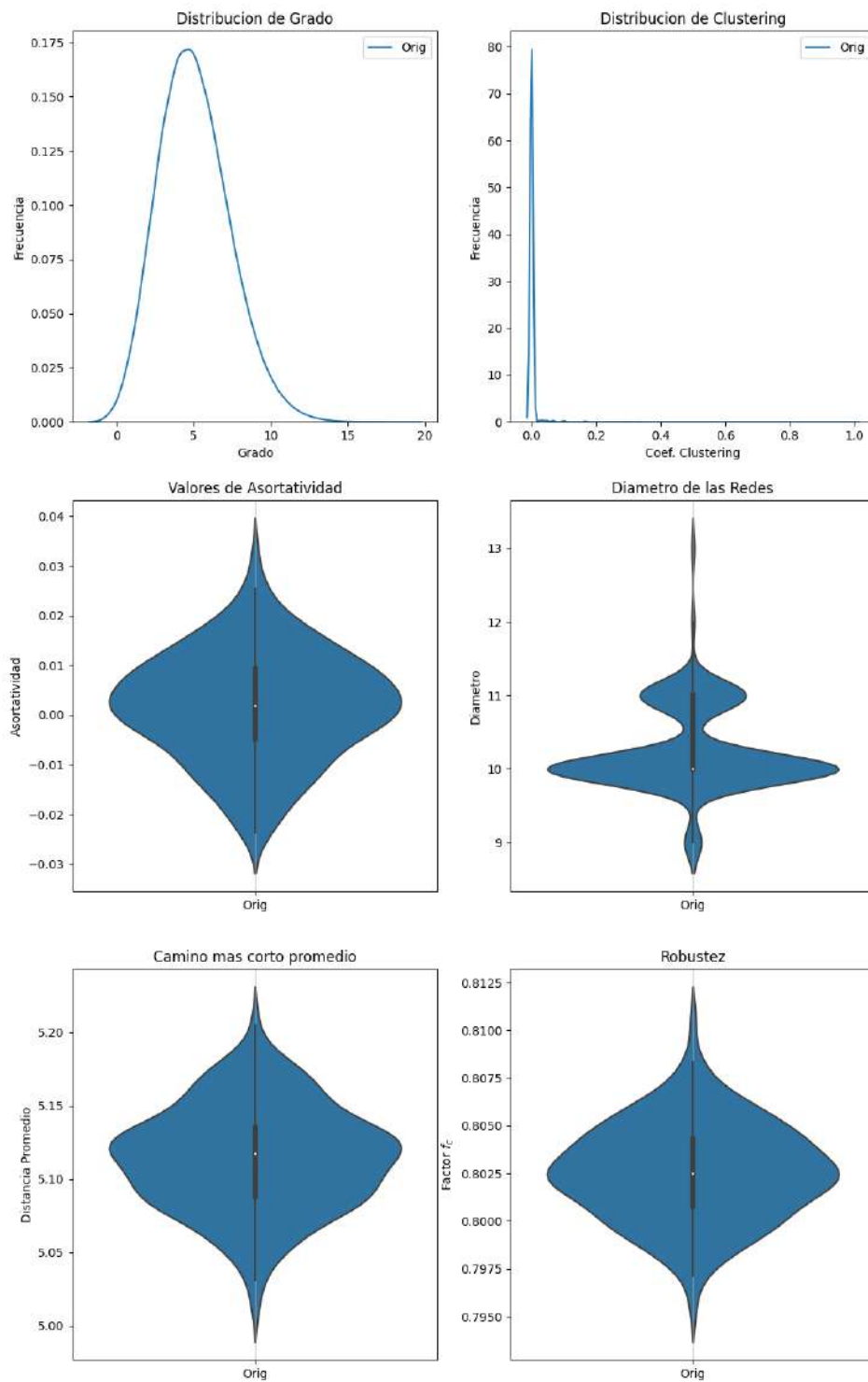


Figura 40: Distribuciones de resultados para redes SBM originales. Completa.
Fuente: Elaboración propia.

7.8. Resultados originales para WS

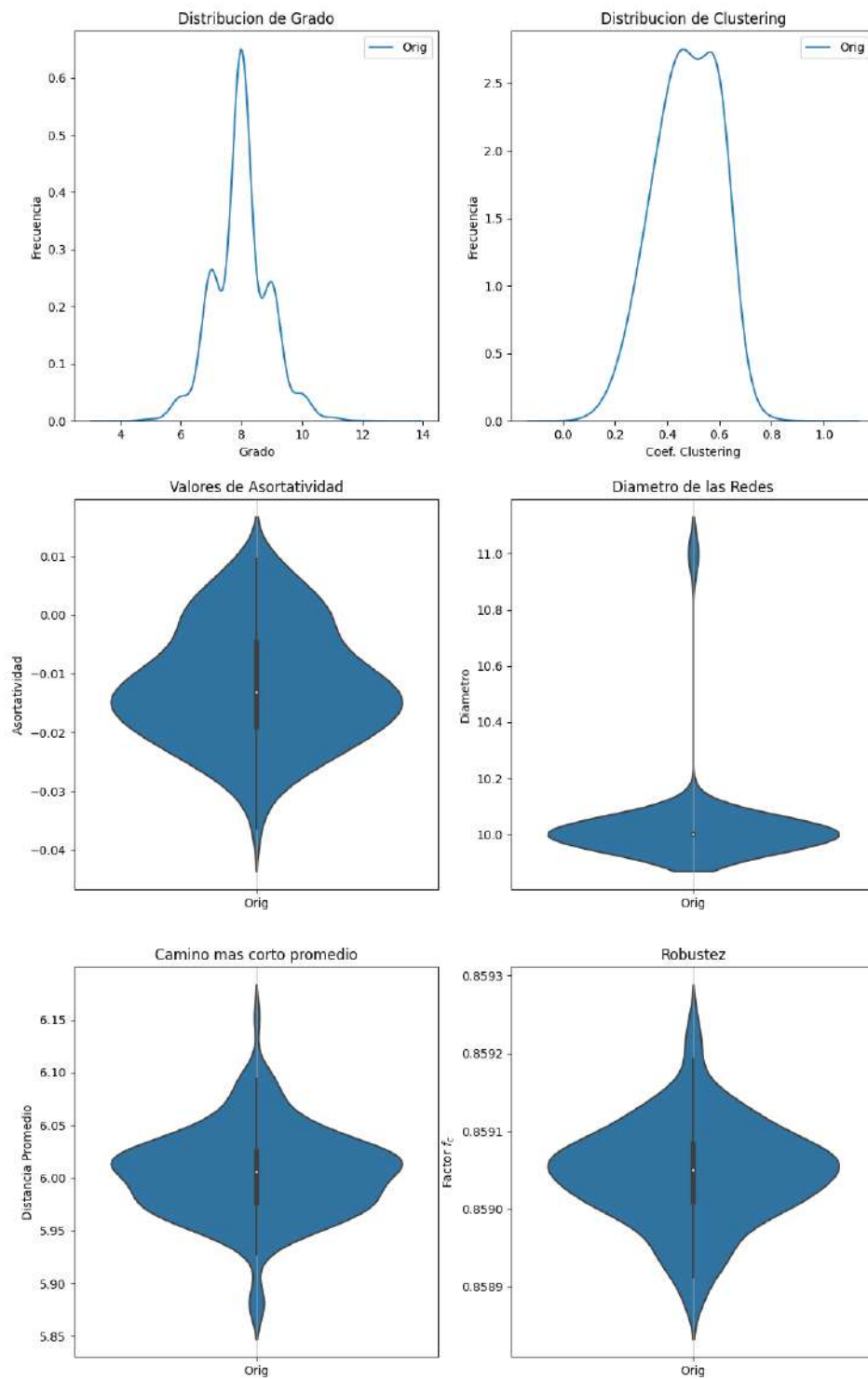


Figura 41: Distribuciones de resultados para redes WS originales. Completa.
Fuente: Elaboración propia.

7.9. Resultados originales para BA

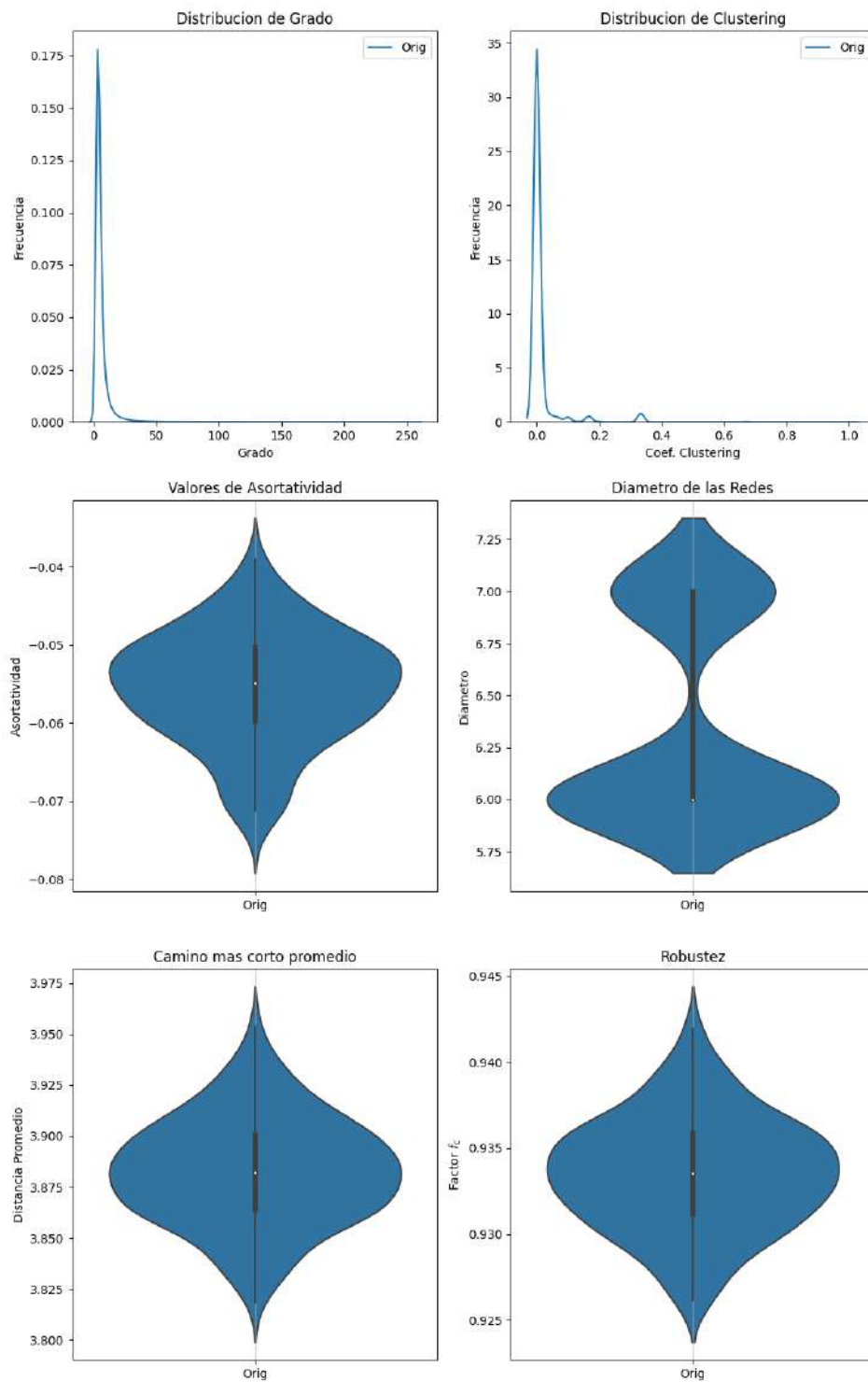


Figura 42: Distribuciones de resultados para redes BA originales. Completa.
Fuente: Elaboración propia.

7.10. Resultados originales para Grid

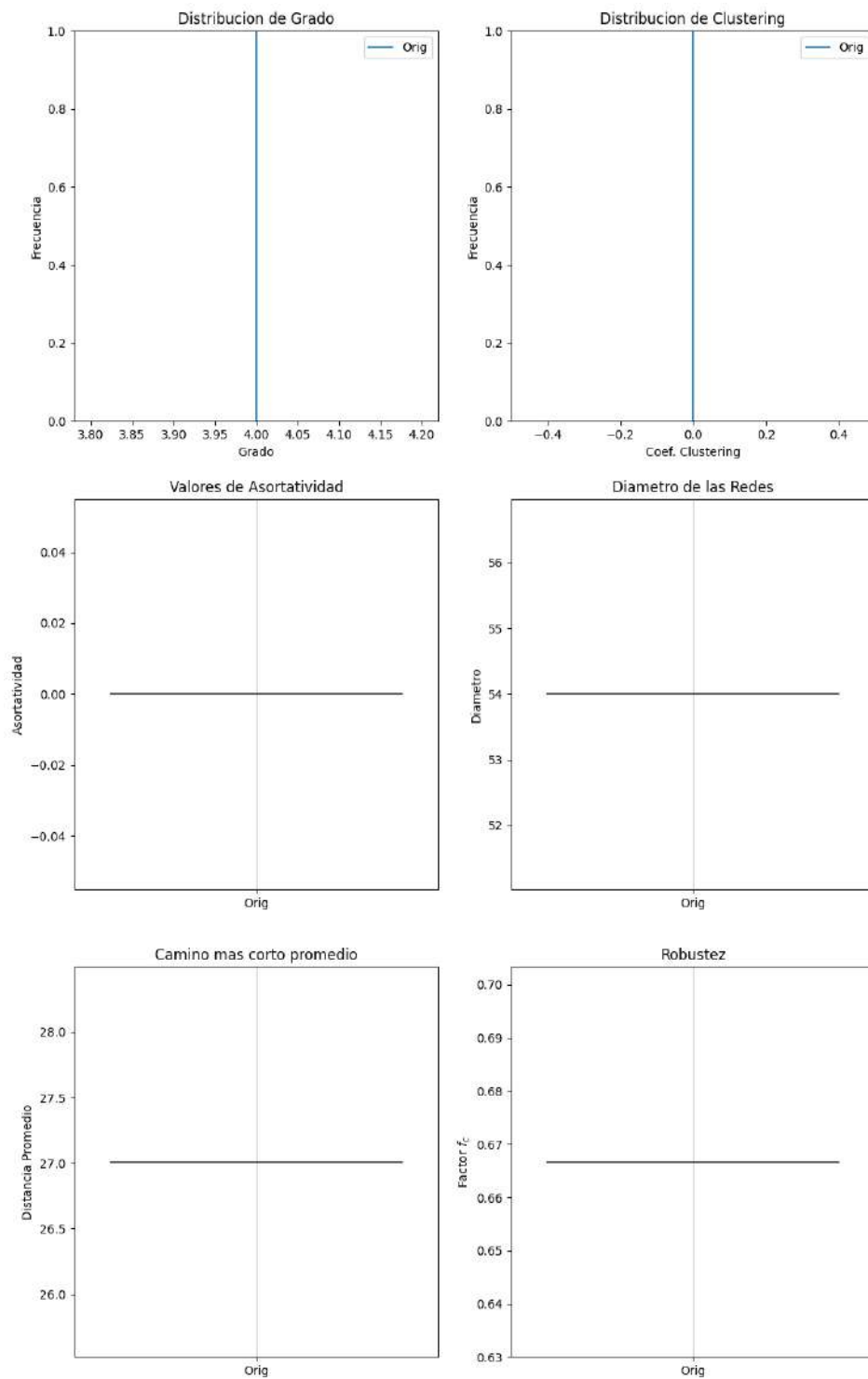


Figura 43: Distribuciones de resultados para redes Grid originales. Completa.
Fuente: Elaboración propia.

7.11. Comparación de resultados restantes para ER

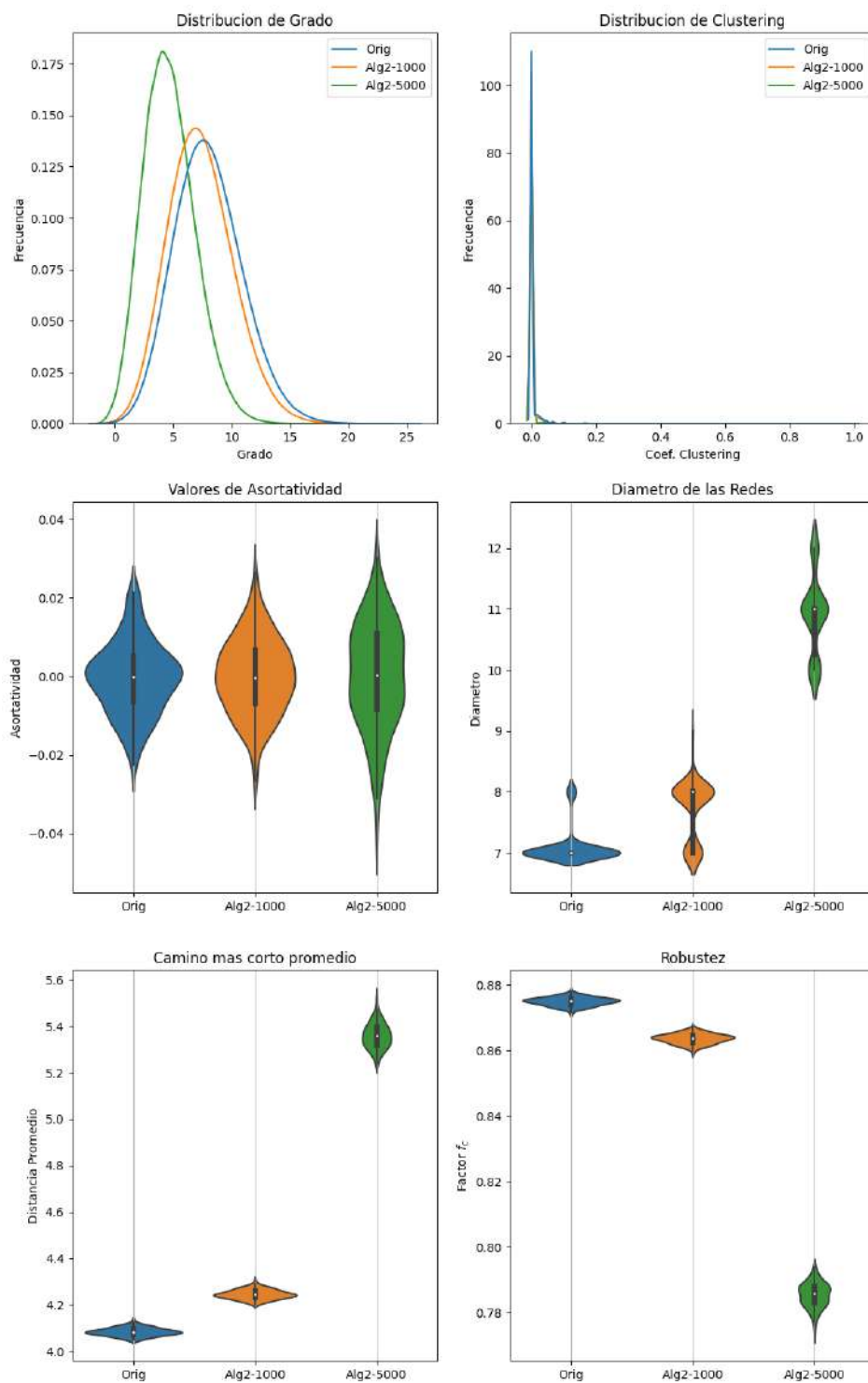


Figura 44: Comparación de resultados algoritmo 2 para redes ER.

Fuente: Elaboración propia.

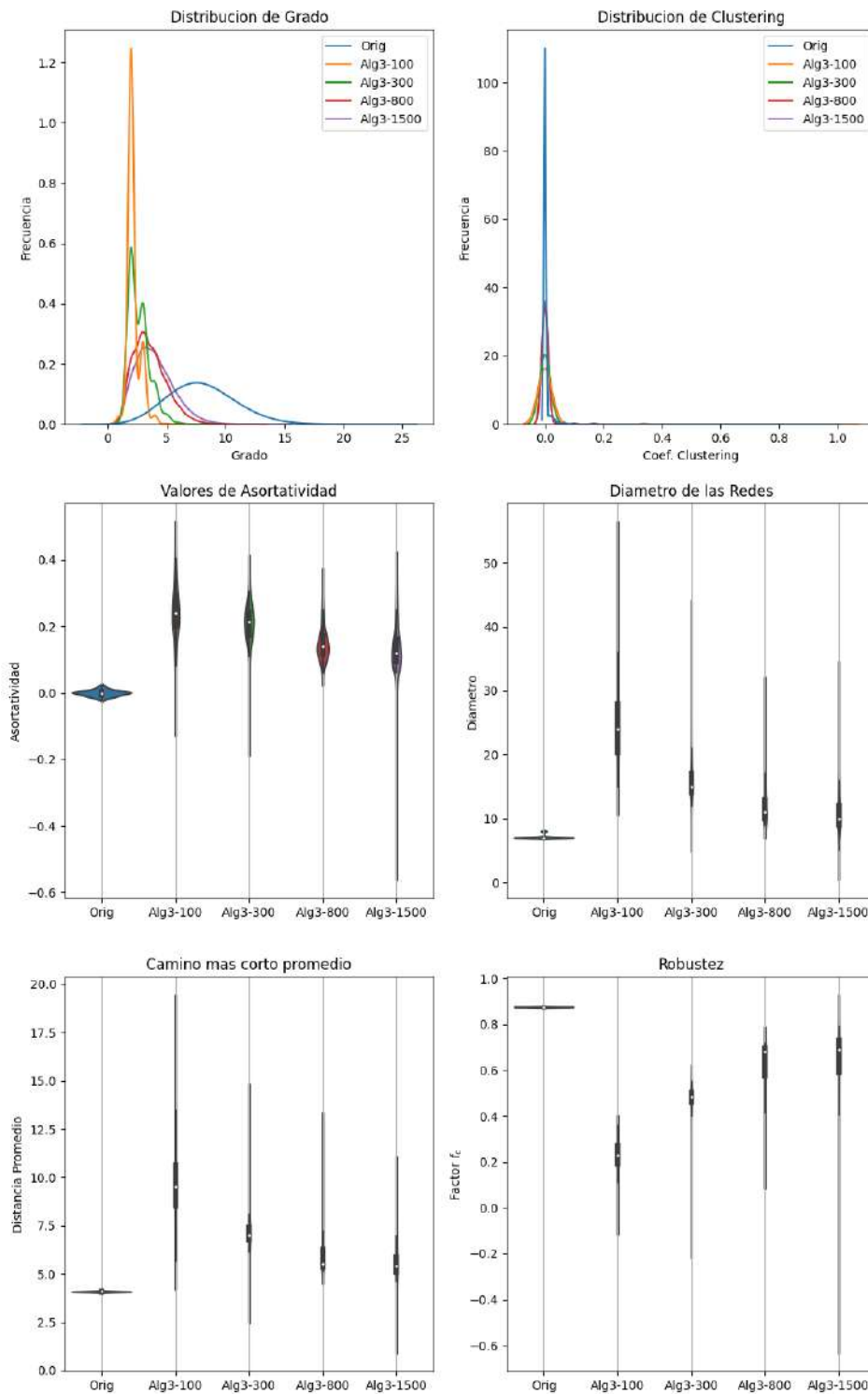


Figura 45: Comparación de resultados algoritmo 3 para redes ER.
Fuente: Elaboración propia.

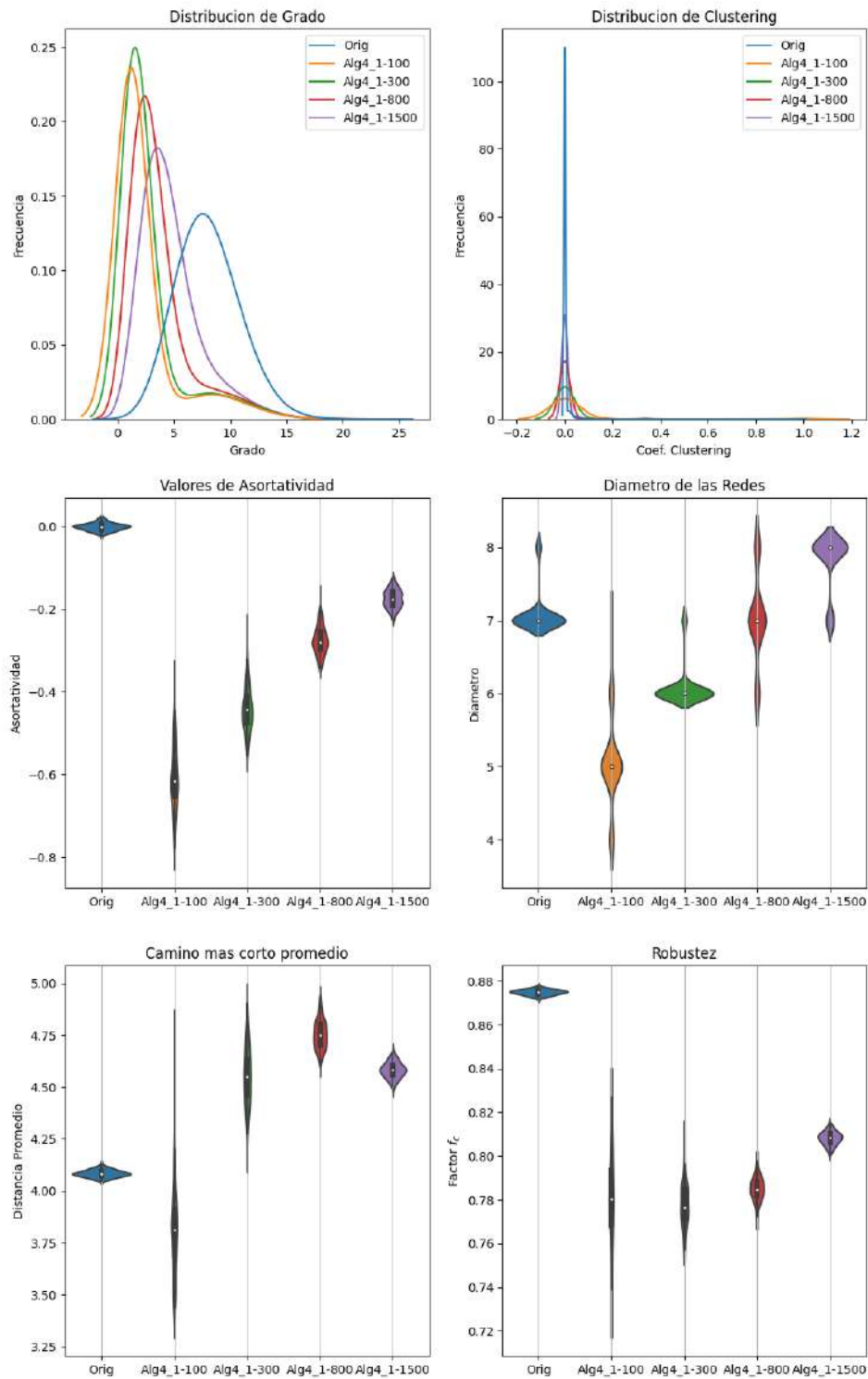


Figura 46: Comparación de resultados algoritmo 4 (BFS) para redes ER.
Fuente: Elaboración propia.

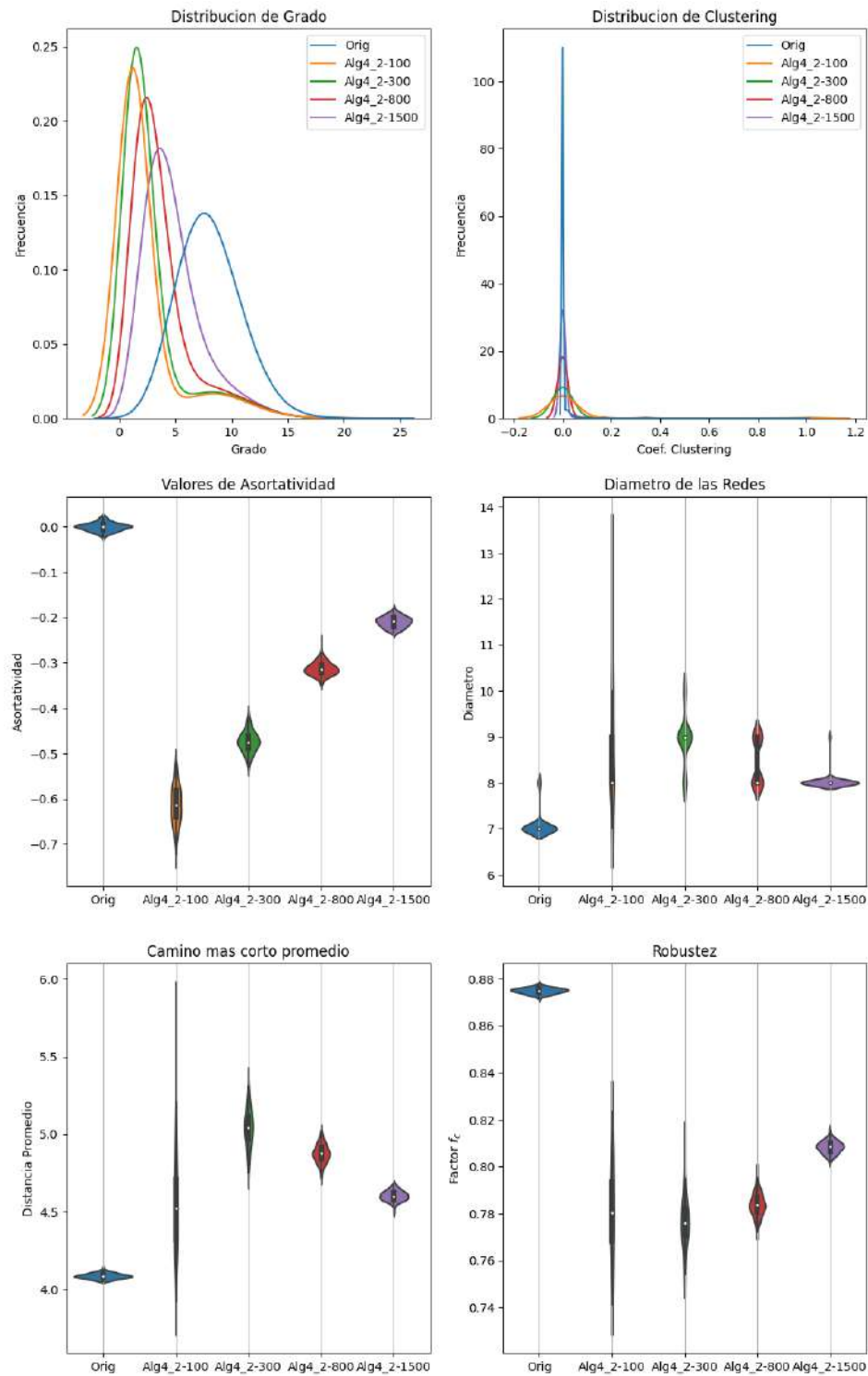


Figura 47: Comparación de resultados algoritmo 4 (DFS) para redes ER.
Fuente: Elaboración propia.

7.12. Comparación de resultados restantes para SBM

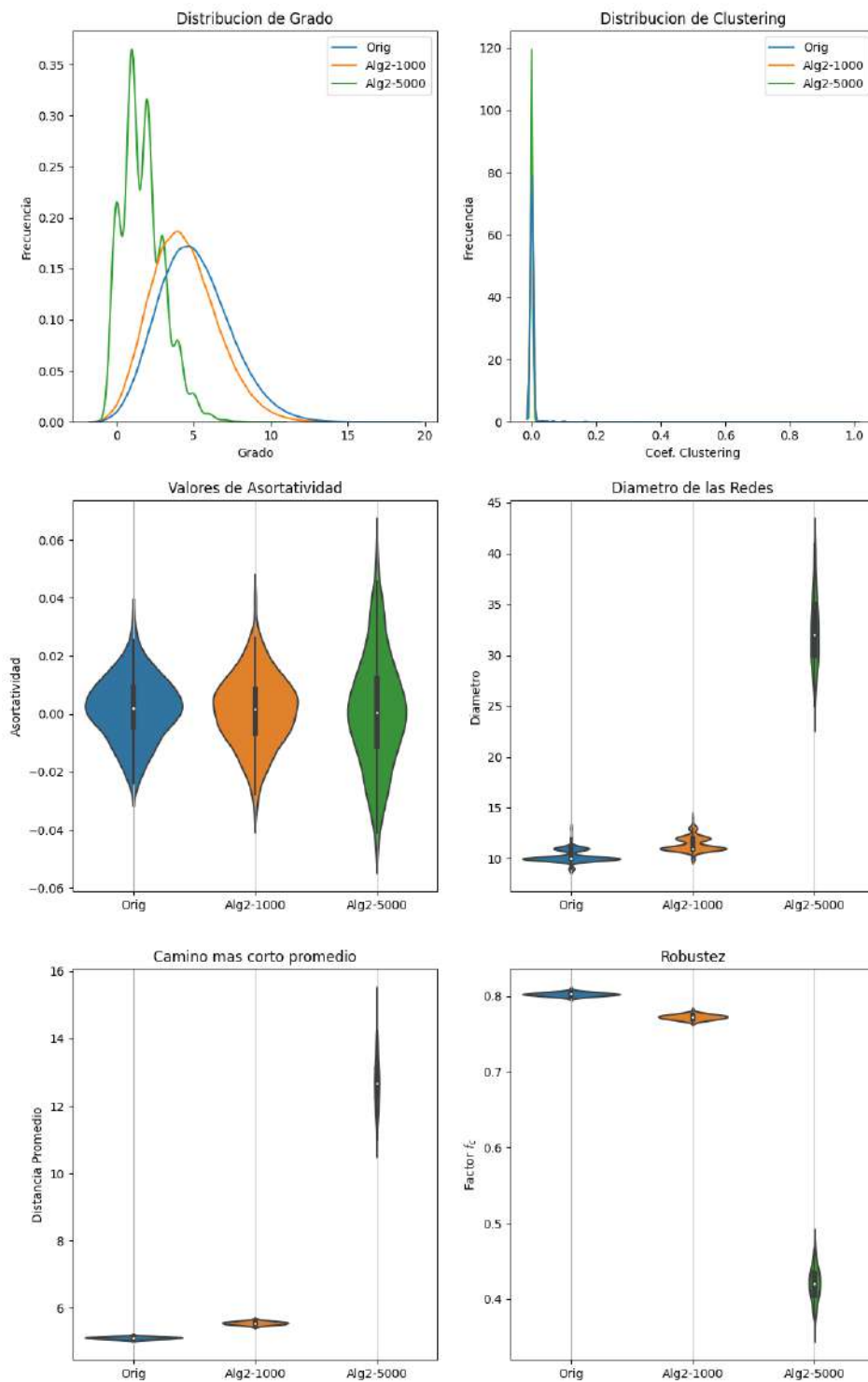


Figura 48: Comparación de resultados algoritmo 2 para redes SBM.

Fuente: Elaboración propia.

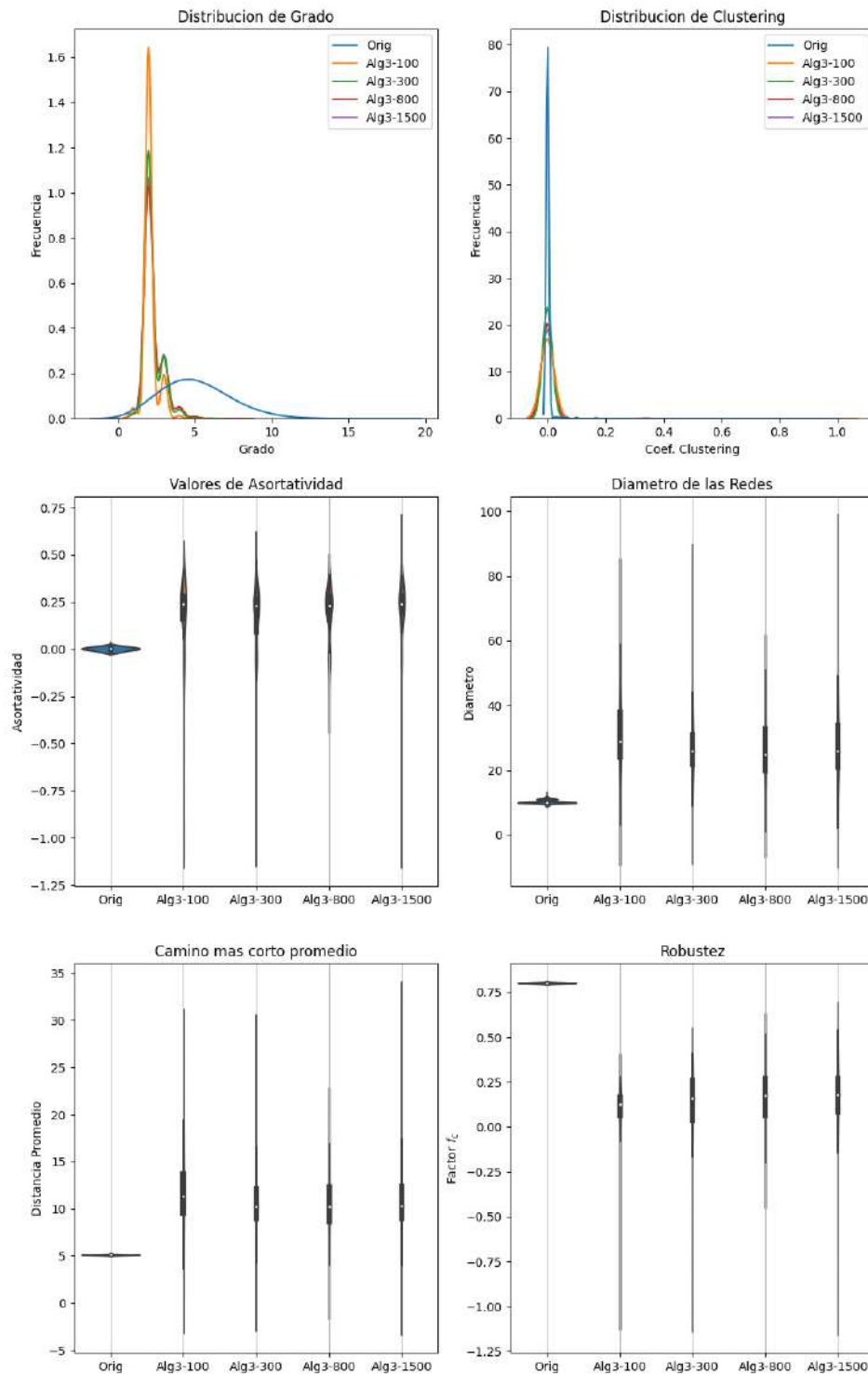


Figura 49: Comparación de resultados algoritmo 3 para redes SBM.
Fuente: Elaboración propia.

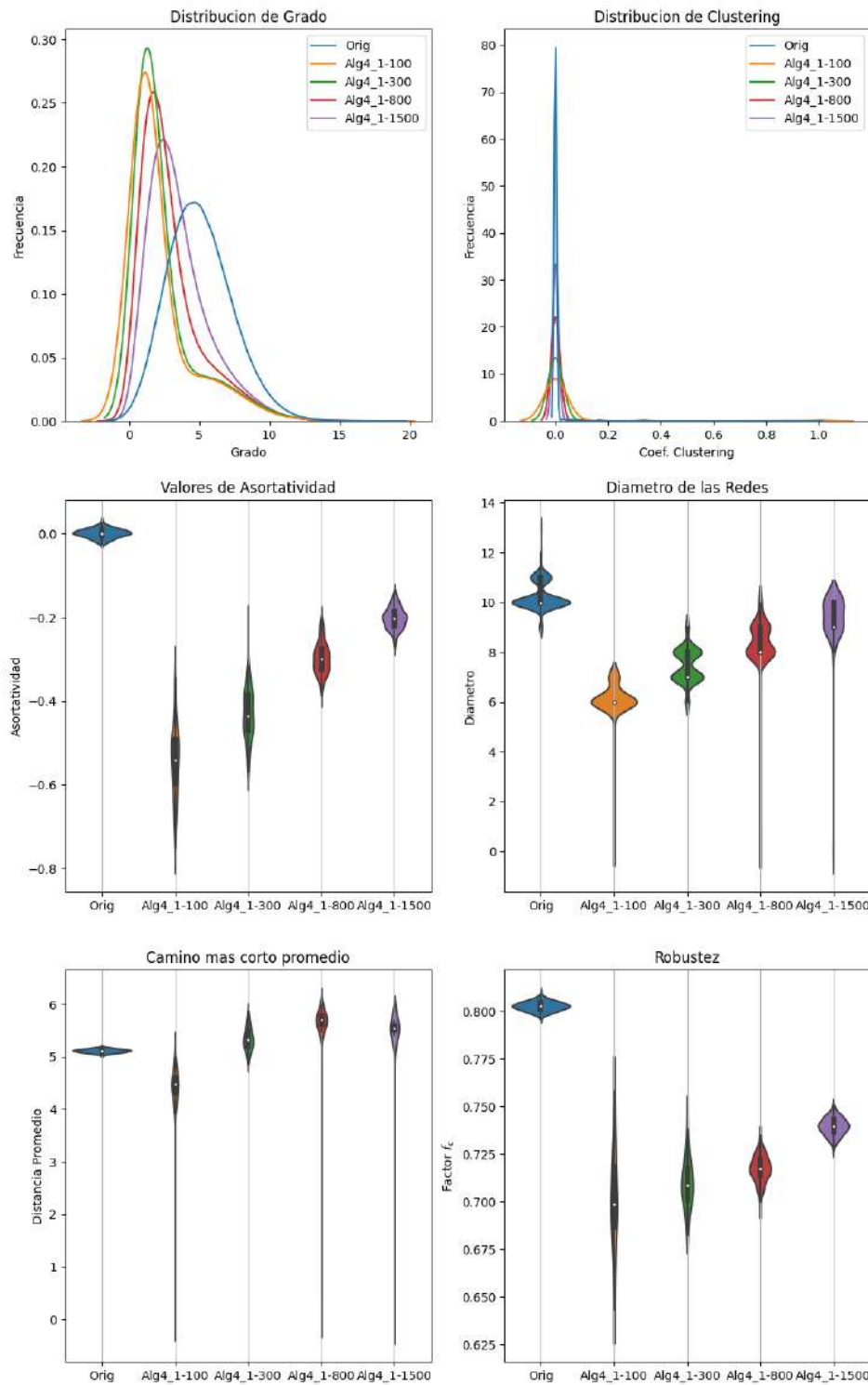


Figura 50: Comparación de resultados algoritmo 4 (BFS) para redes SBM.
Fuente: Elaboración propia.

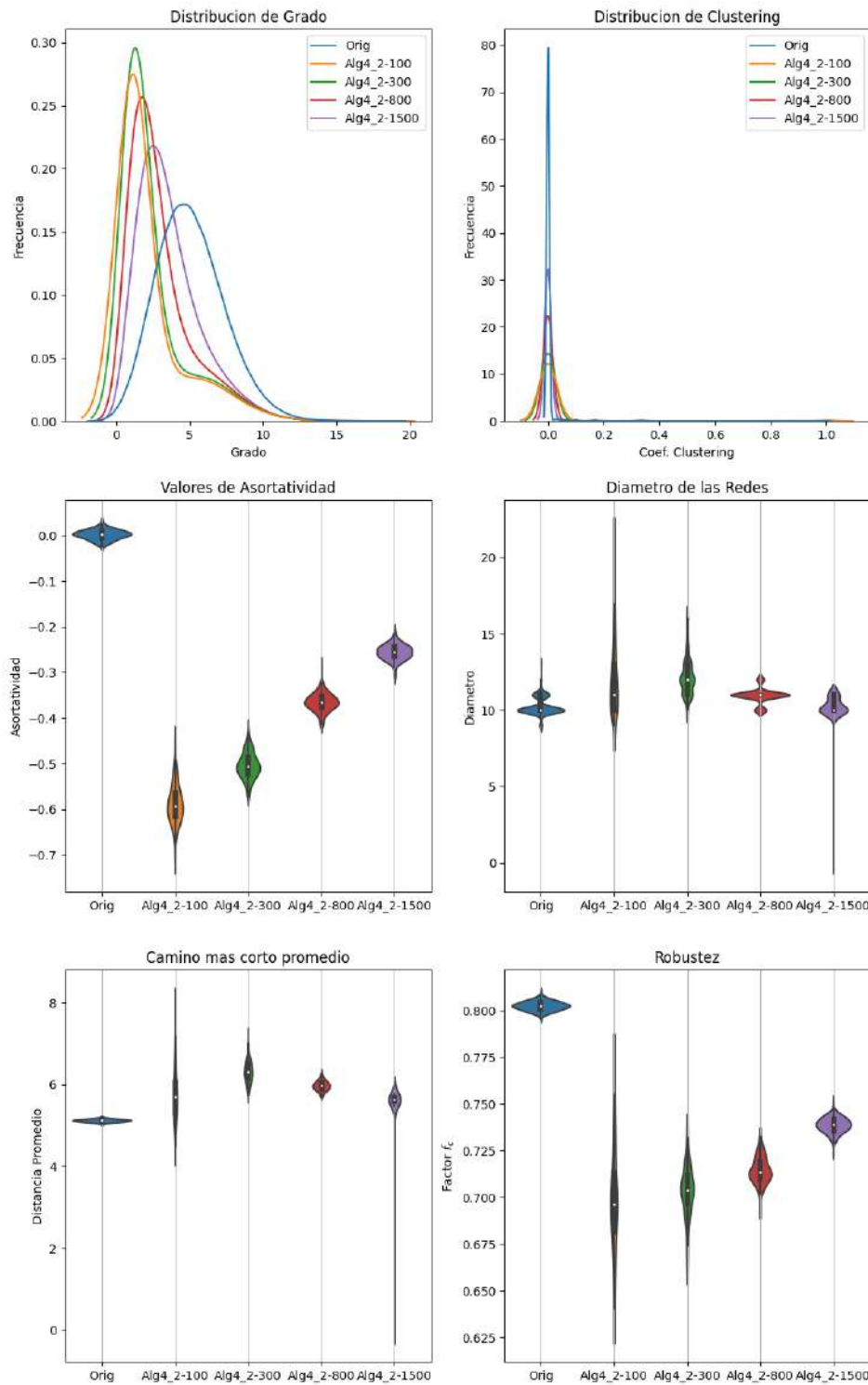


Figura 51: Comparación de resultados algoritmo 4 (DFS) para redes SBM.
Fuente: Elaboración propia.

7.13. Comparación de resultados restantes para WS

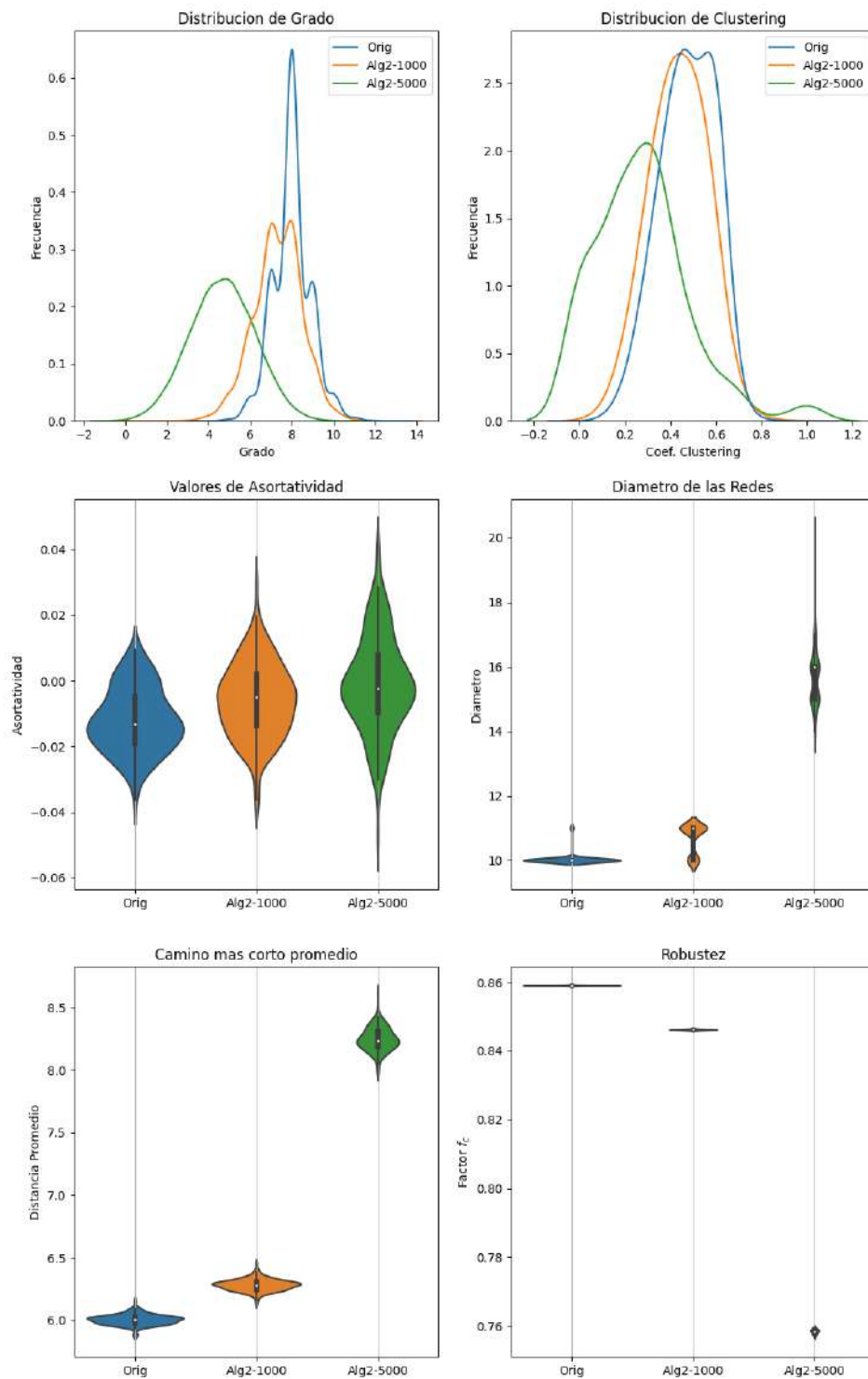


Figura 52: Comparación de resultados algoritmo 2 para redes WS.

Fuente: Elaboración propia.

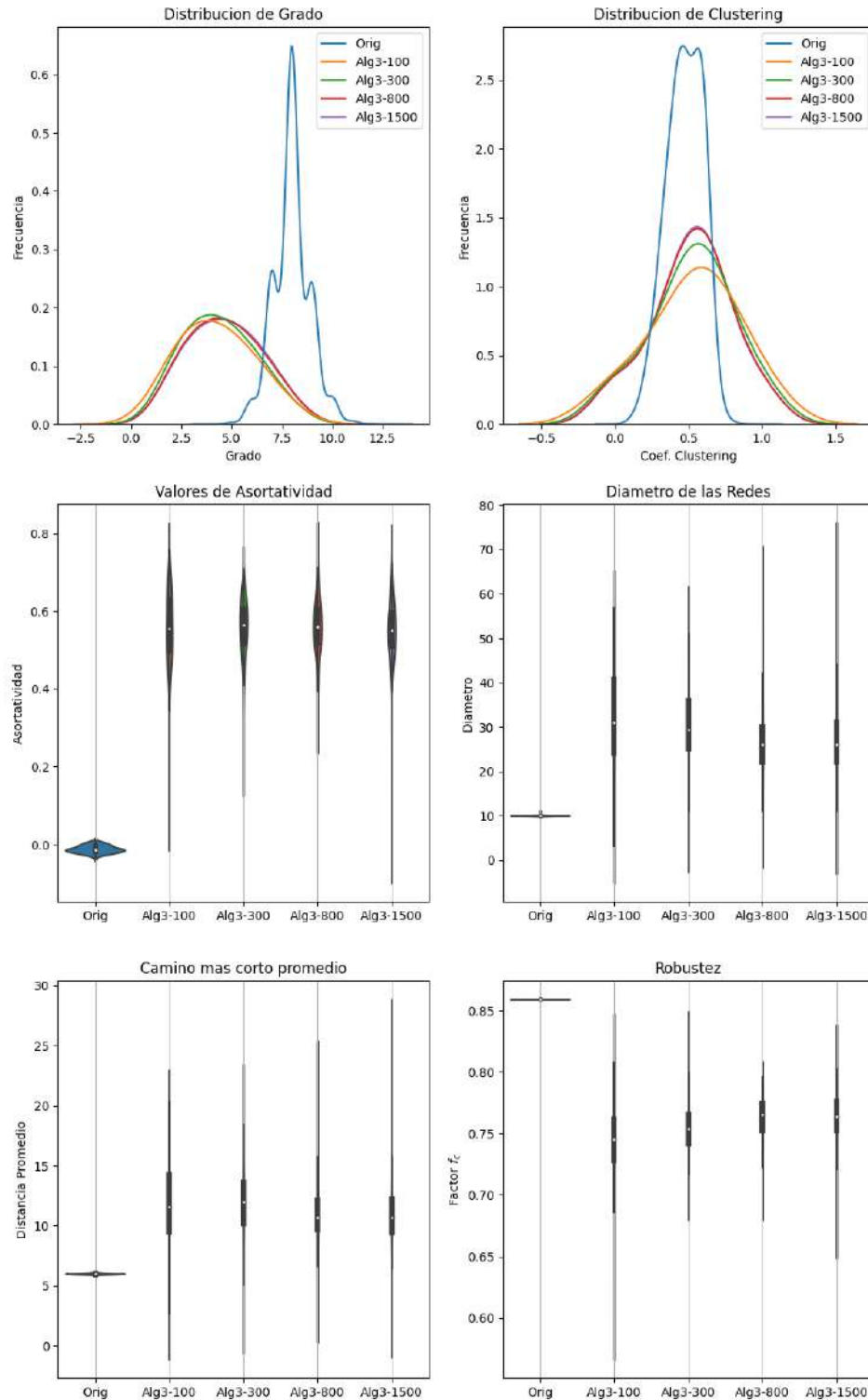


Figura 53: Comparación de resultados algoritmo 3 para redes WS.
Fuente: Elaboración propia.

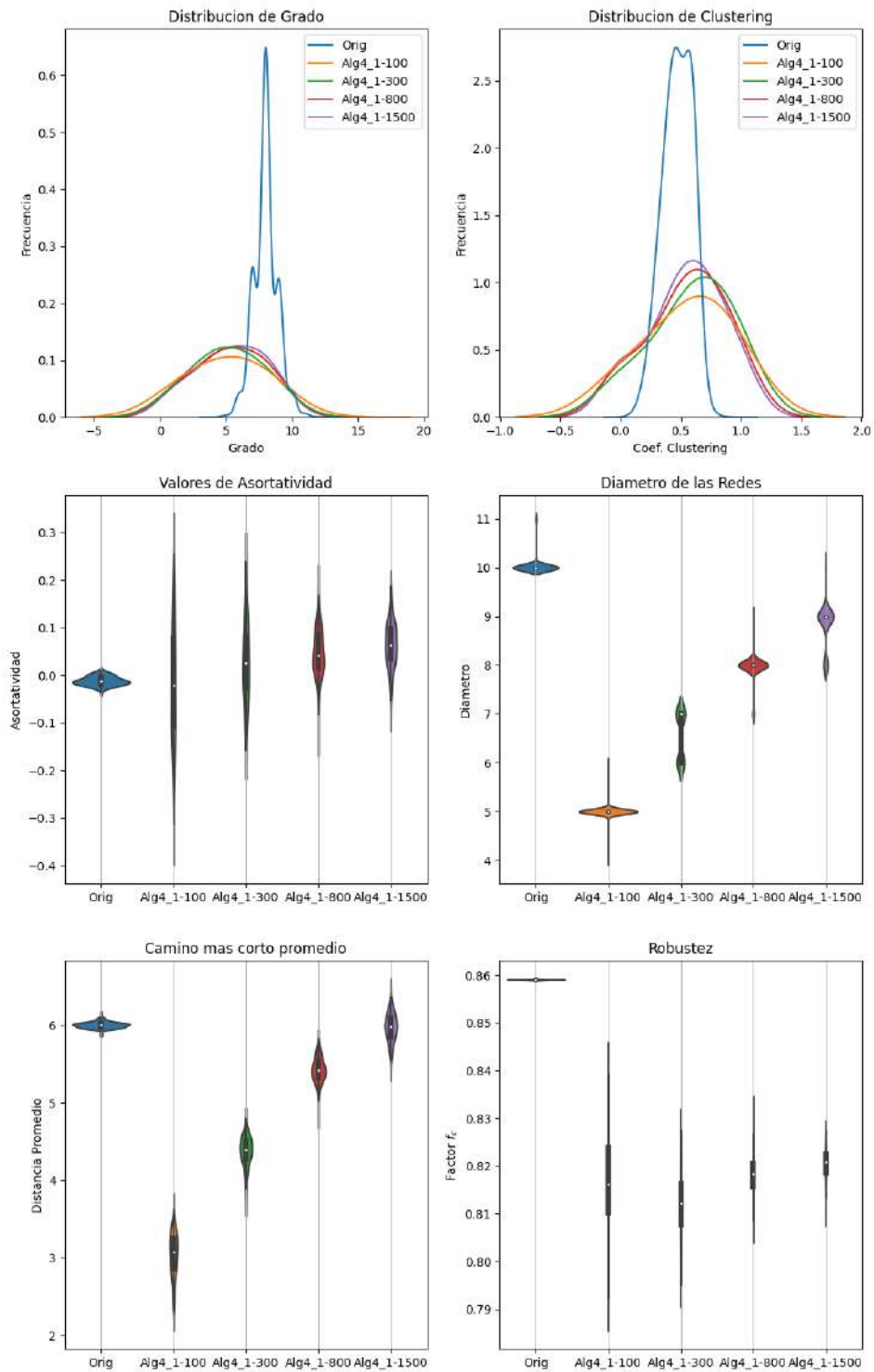


Figura 54: Comparación de resultados algoritmo 4 (BFS) para redes WS.
Fuente: Elaboración propia.

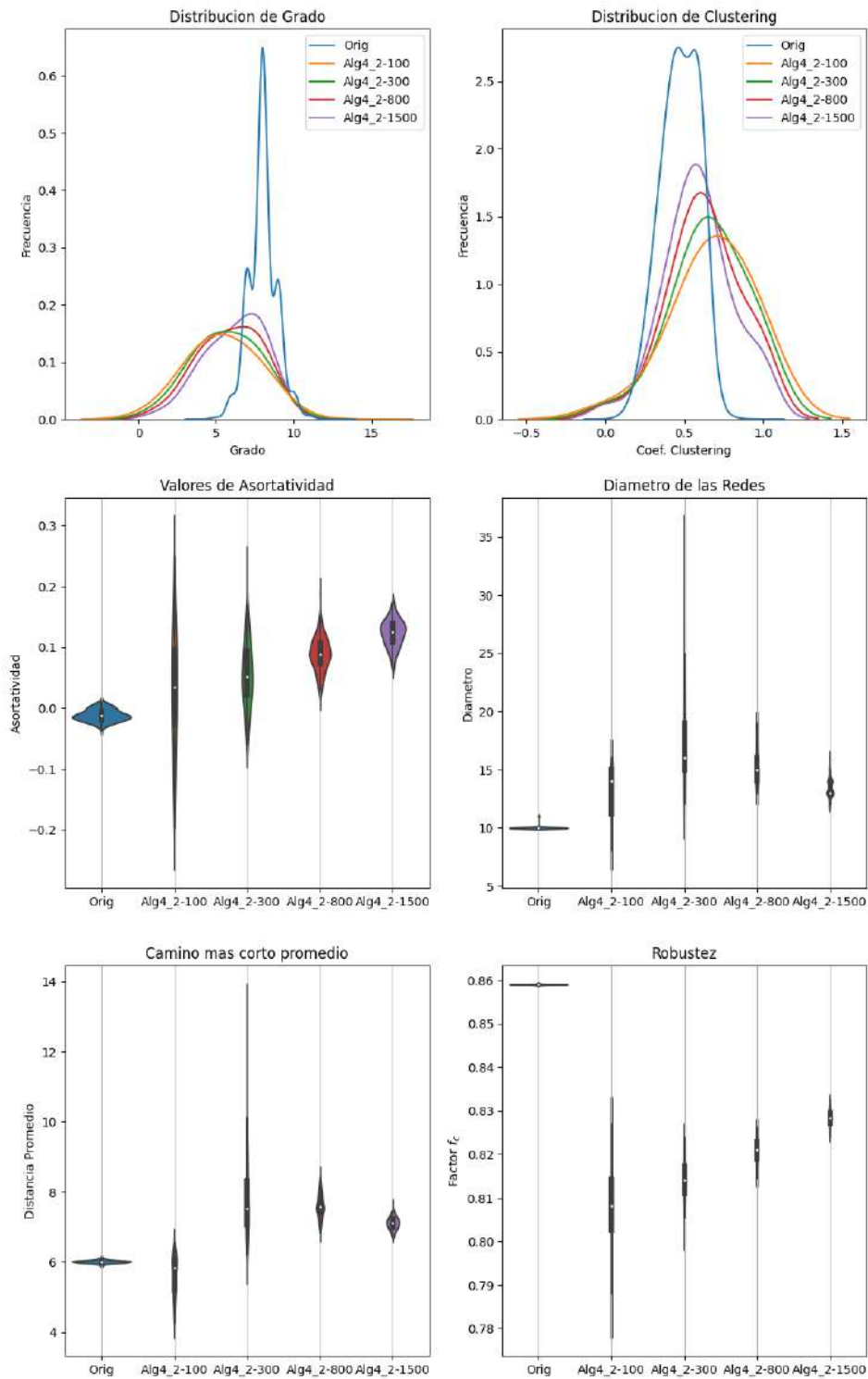


Figura 55: Comparación de resultados algoritmo 4 (DFS) para redes WS.
Fuente: Elaboración propia.

7.14. Comparación de resultados restantes para BA

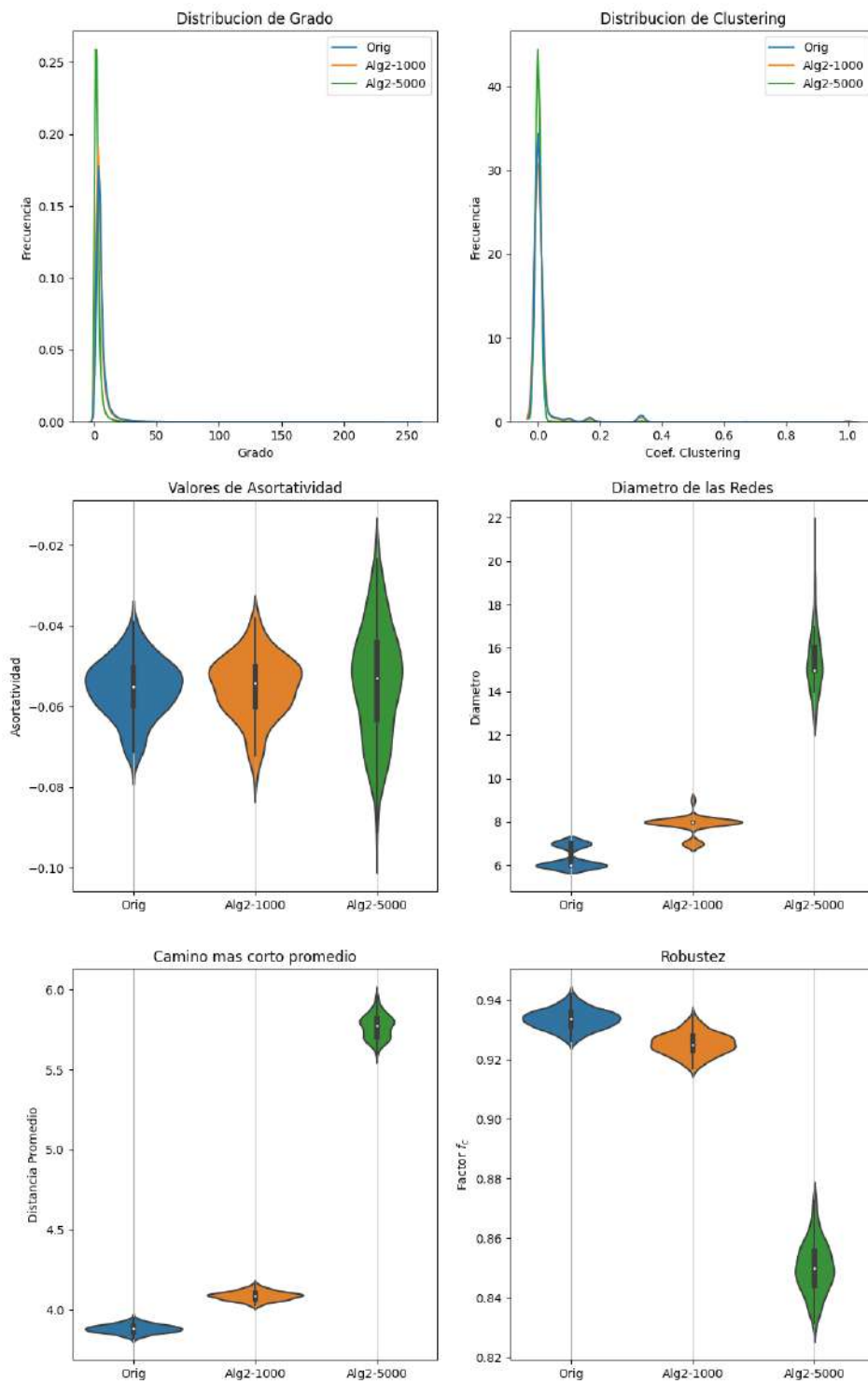


Figura 56: Comparación de resultados algoritmo 2 para redes BA.

Fuente: Elaboración propia.

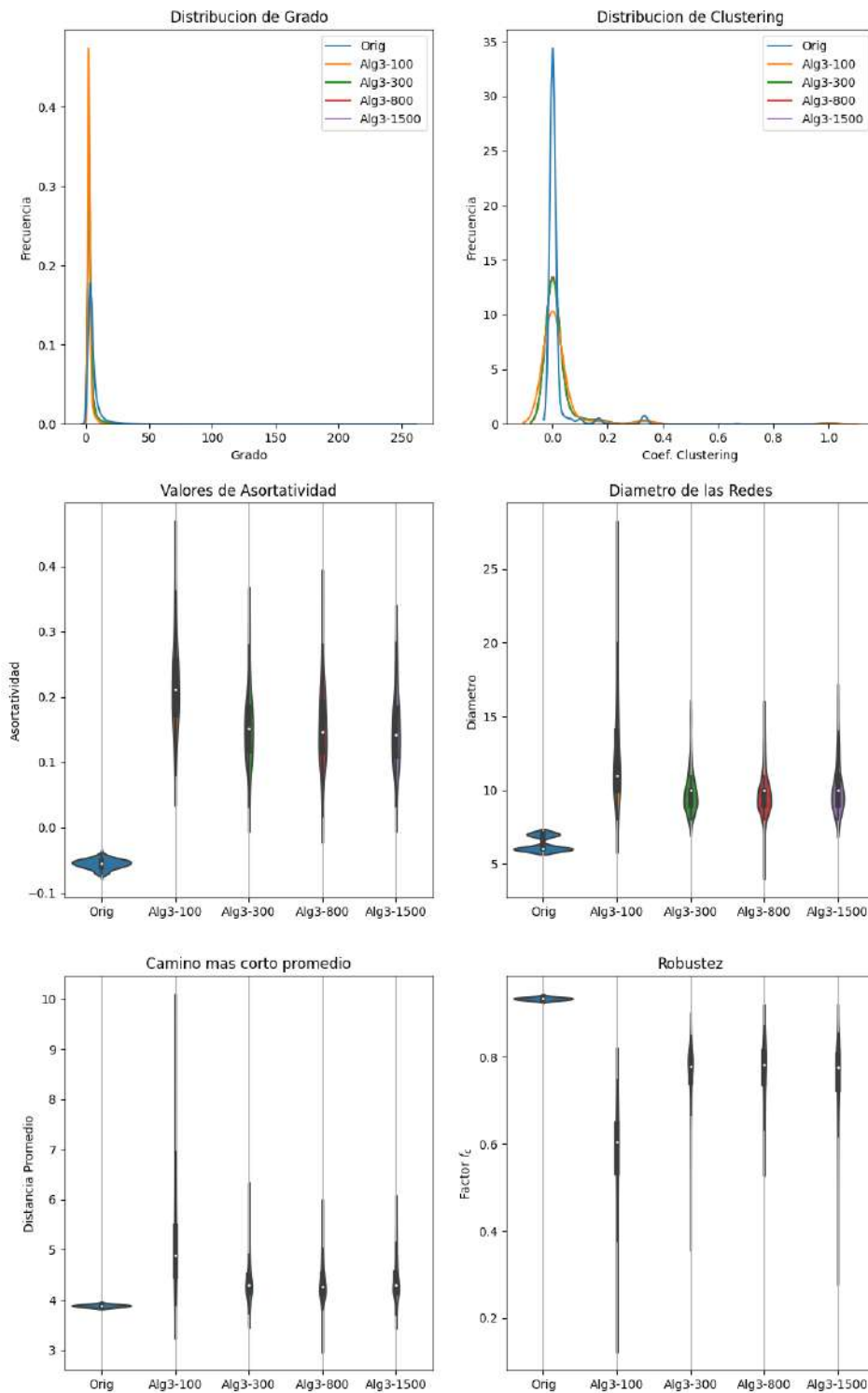


Figura 57: Comparación de resultados algoritmo 3 para redes BA.
Fuente: Elaboración propia.

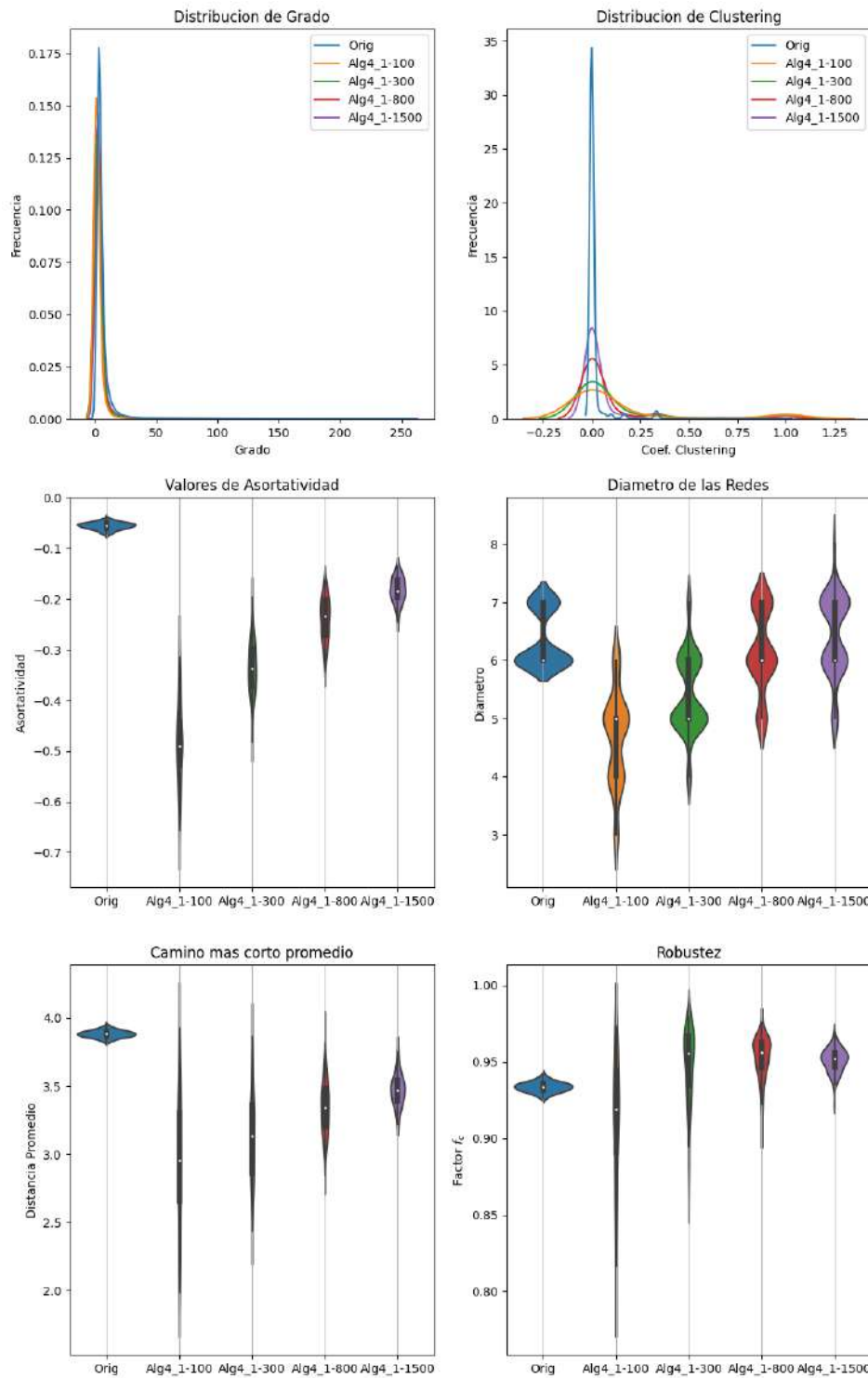


Figura 58: Comparación de resultados algoritmo 4 (BFS) para redes BA.
Fuente: Elaboración propia.

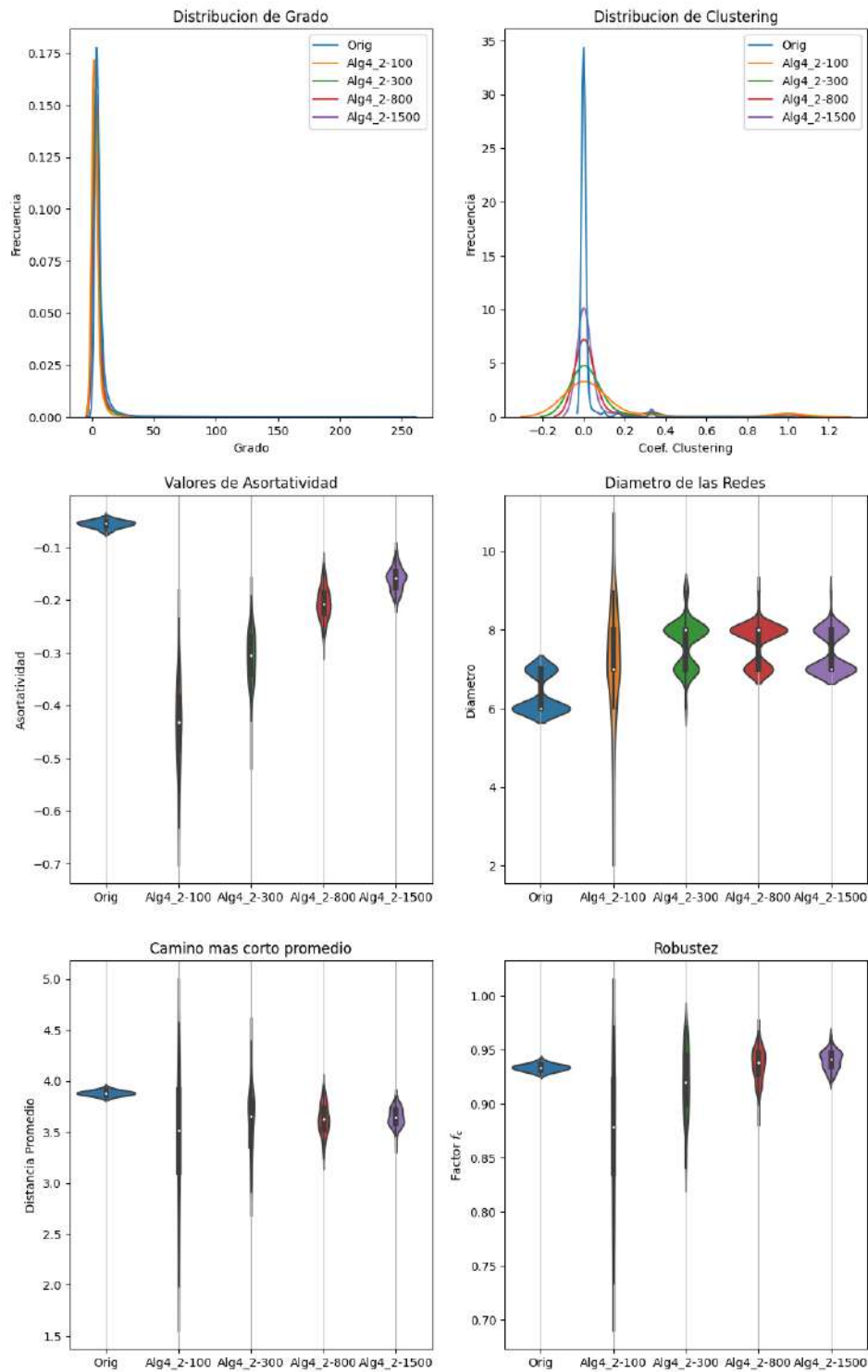


Figura 59: Comparación de resultados algoritmo 4 (DFS) para redes BA.
Fuente: Elaboración propia.

7.15. Comparación de resultados restantes para Grid

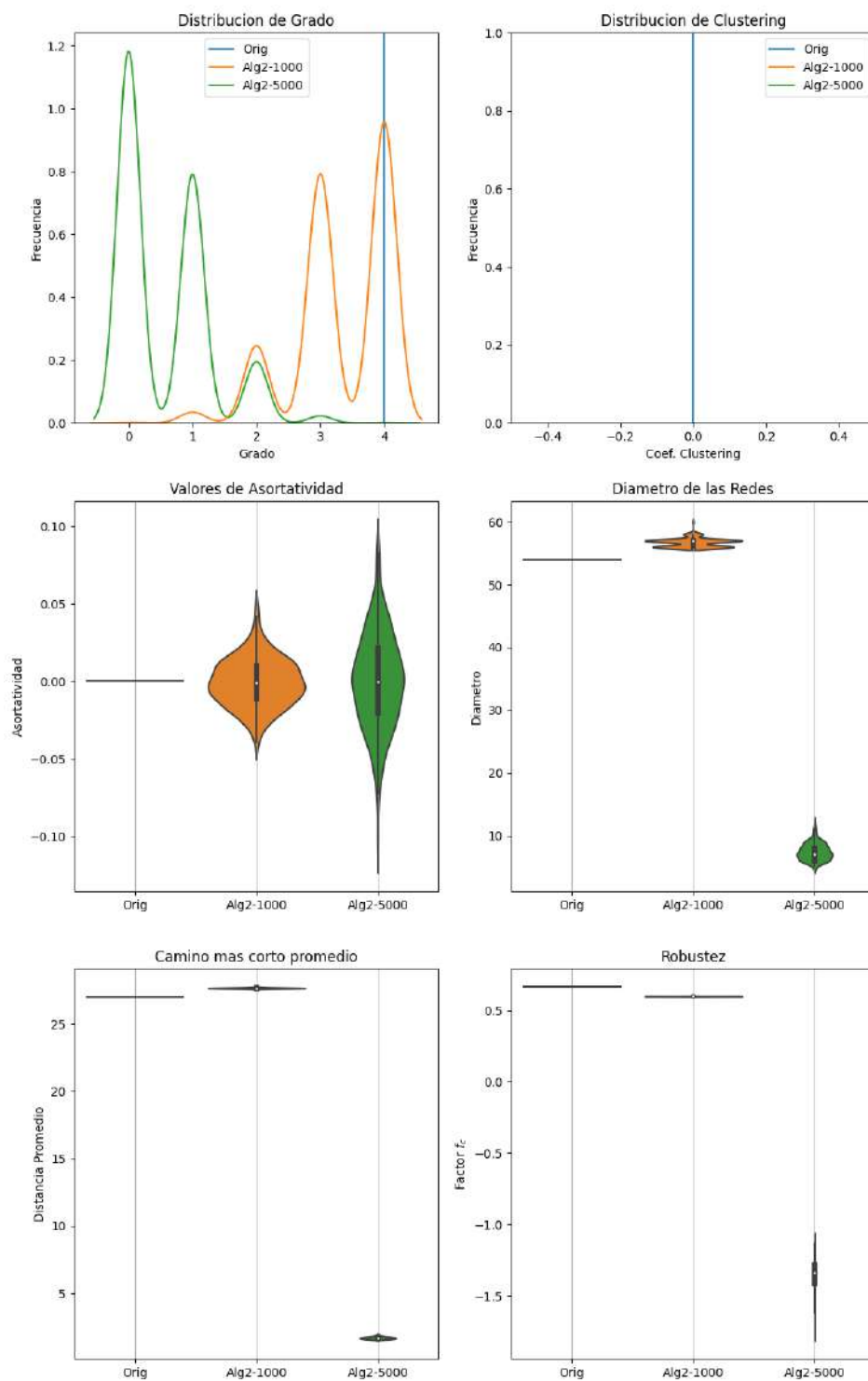


Figura 60: Comparación de resultados algoritmo 2 para redes Grid.

Fuente: Elaboración propia.

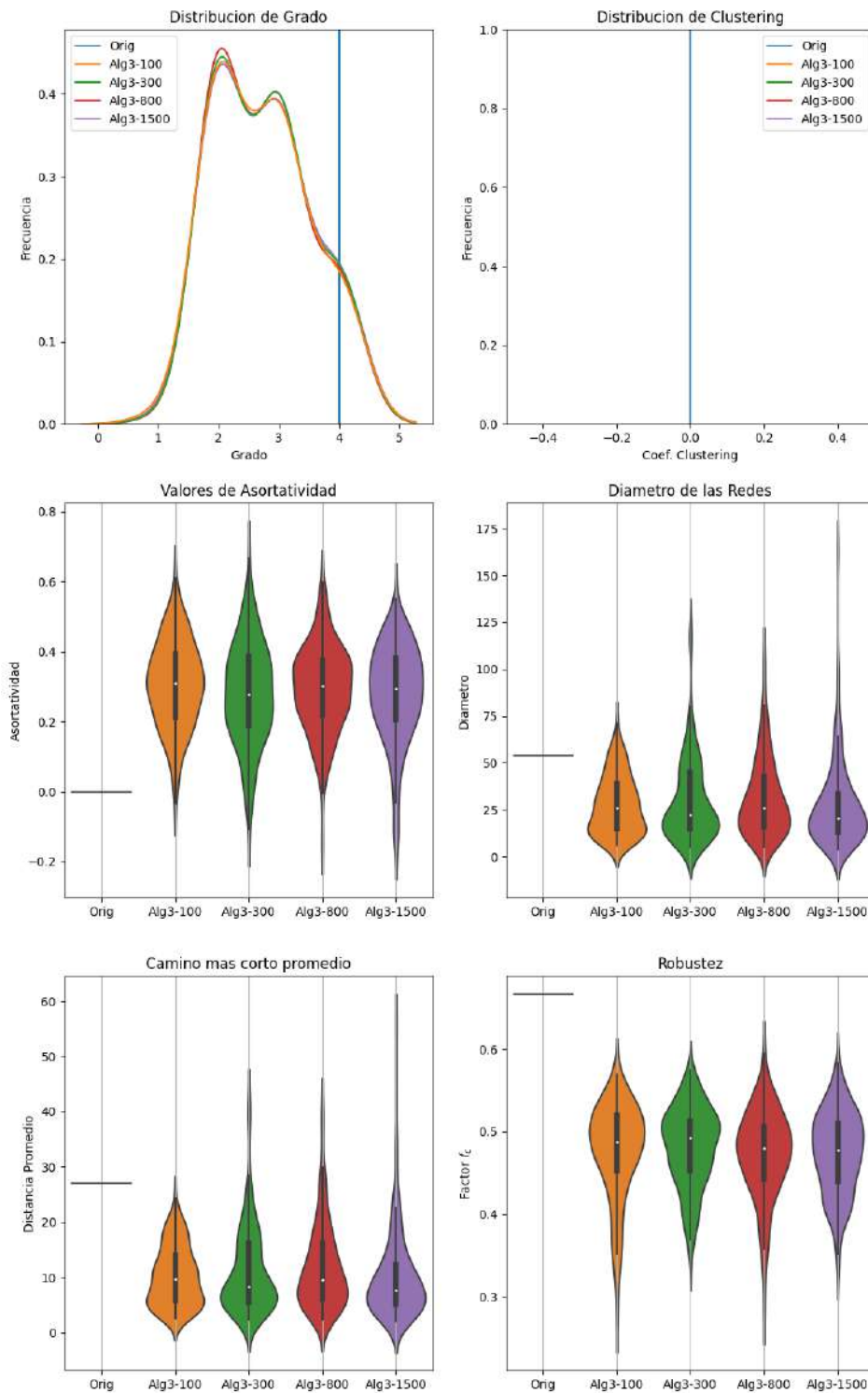


Figura 61: Comparación de resultados algoritmo 3 para redes Grid.
Fuente: Elaboración propia.

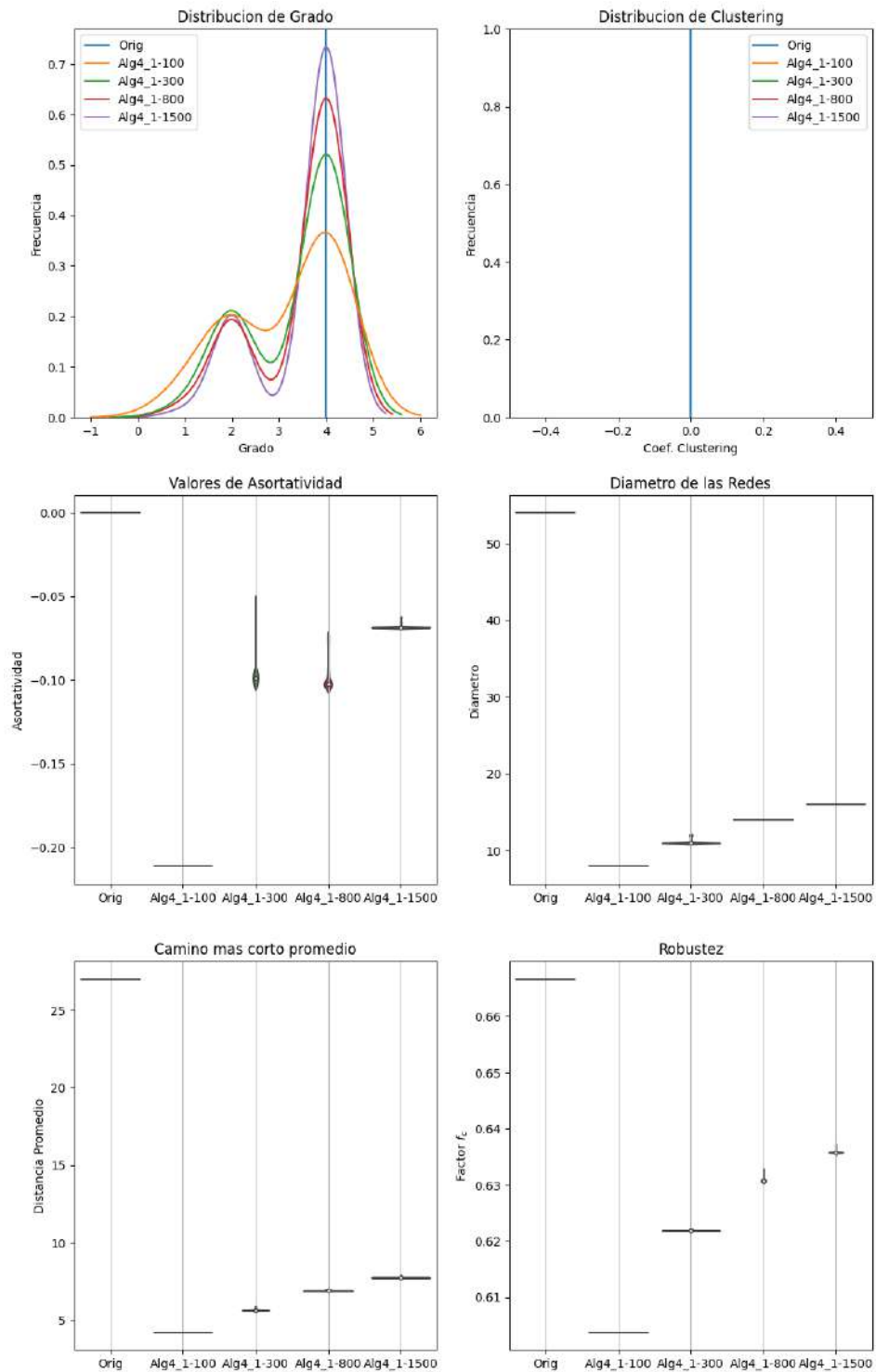


Figura 62: Comparación de resultados algoritmo 4 (BFS) para redes Grid.
Fuente: Elaboración propia.

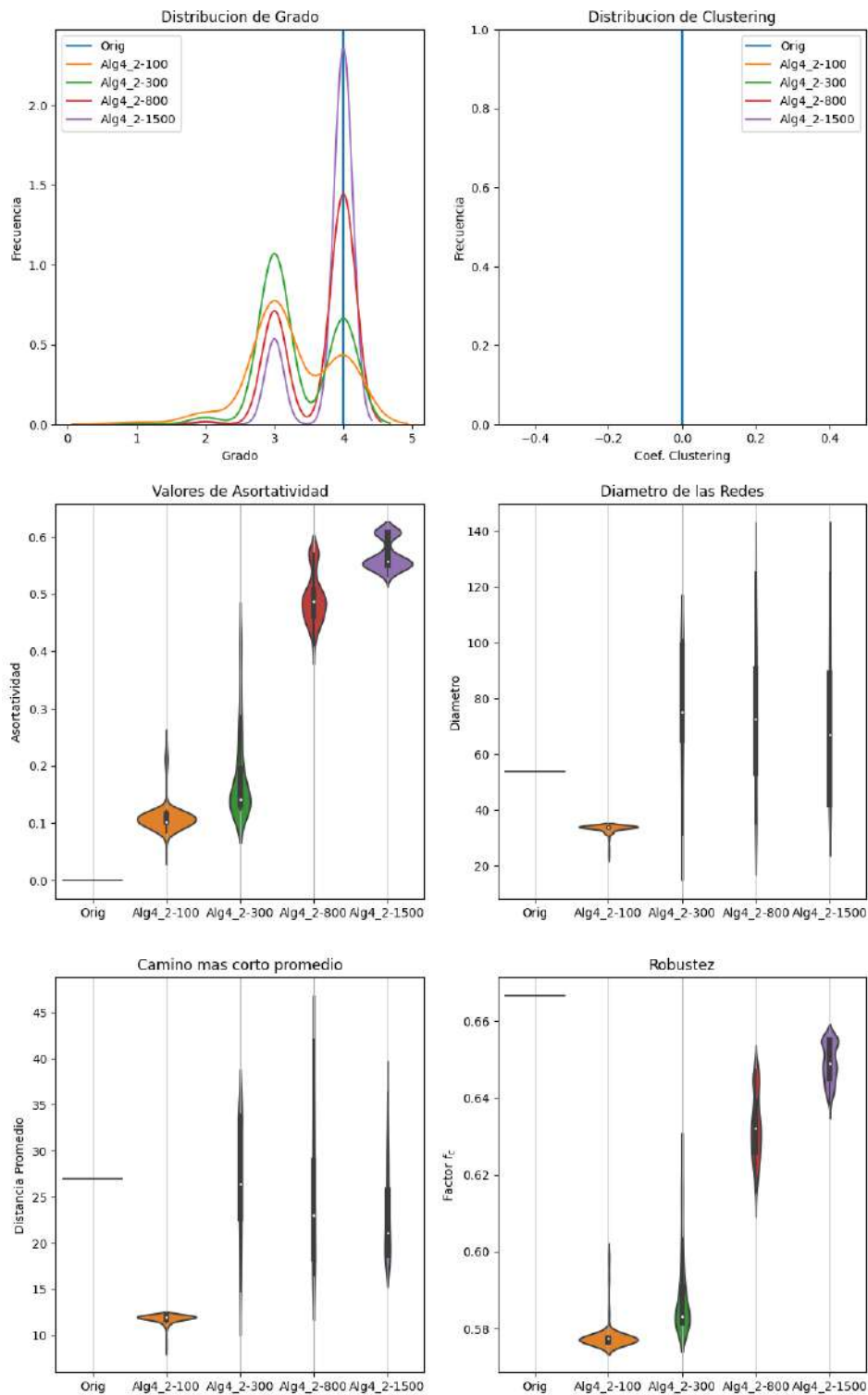


Figura 63: Comparación de resultados algoritmo 4 (DFS) para redes Grid.
Fuente: Elaboración propia.

7.16. Comparación de resultados restantes para Red 1

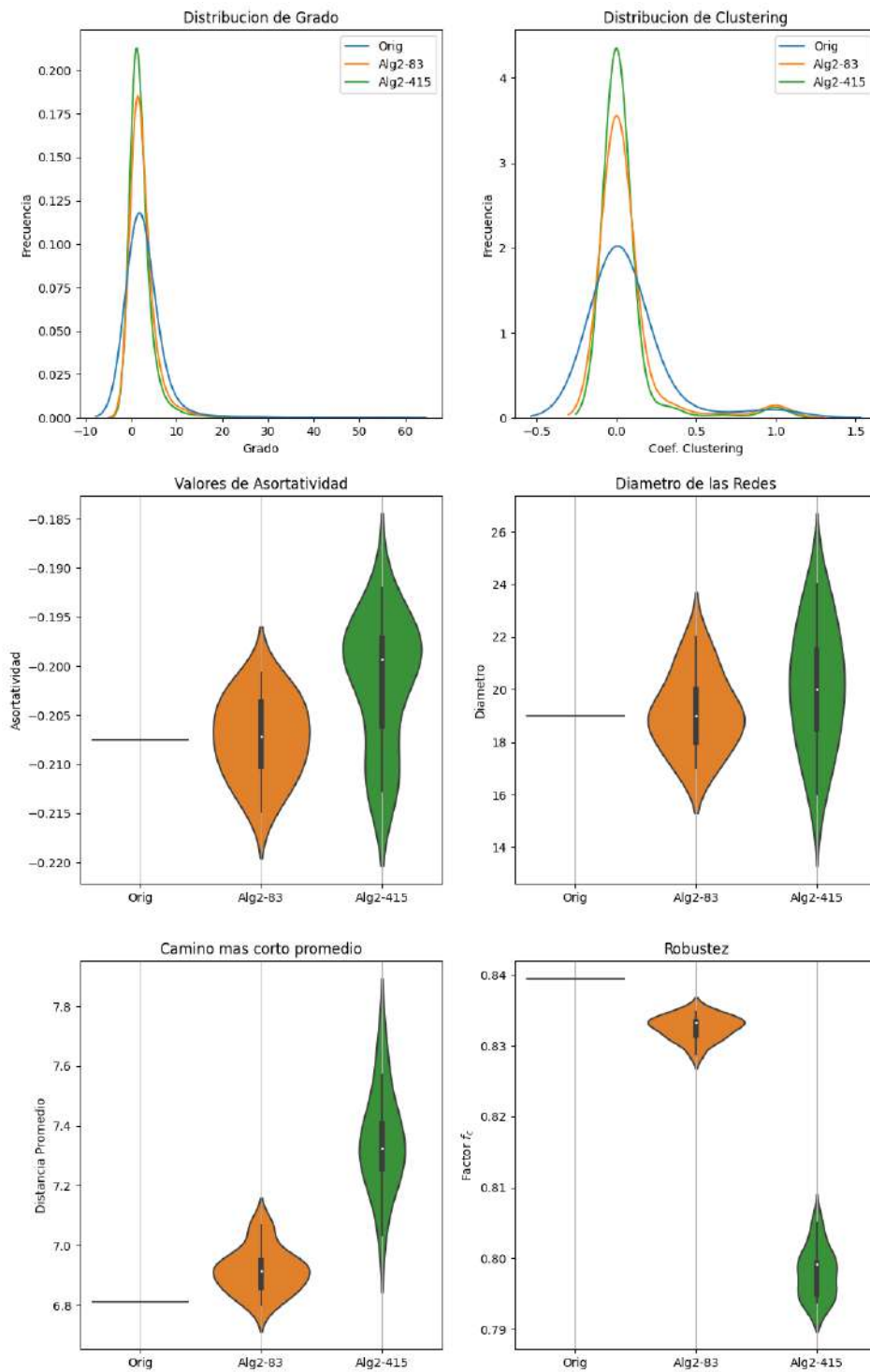


Figura 64: Comparación de resultados algoritmo 2 para red 1.

Fuente: Elaboración propia.

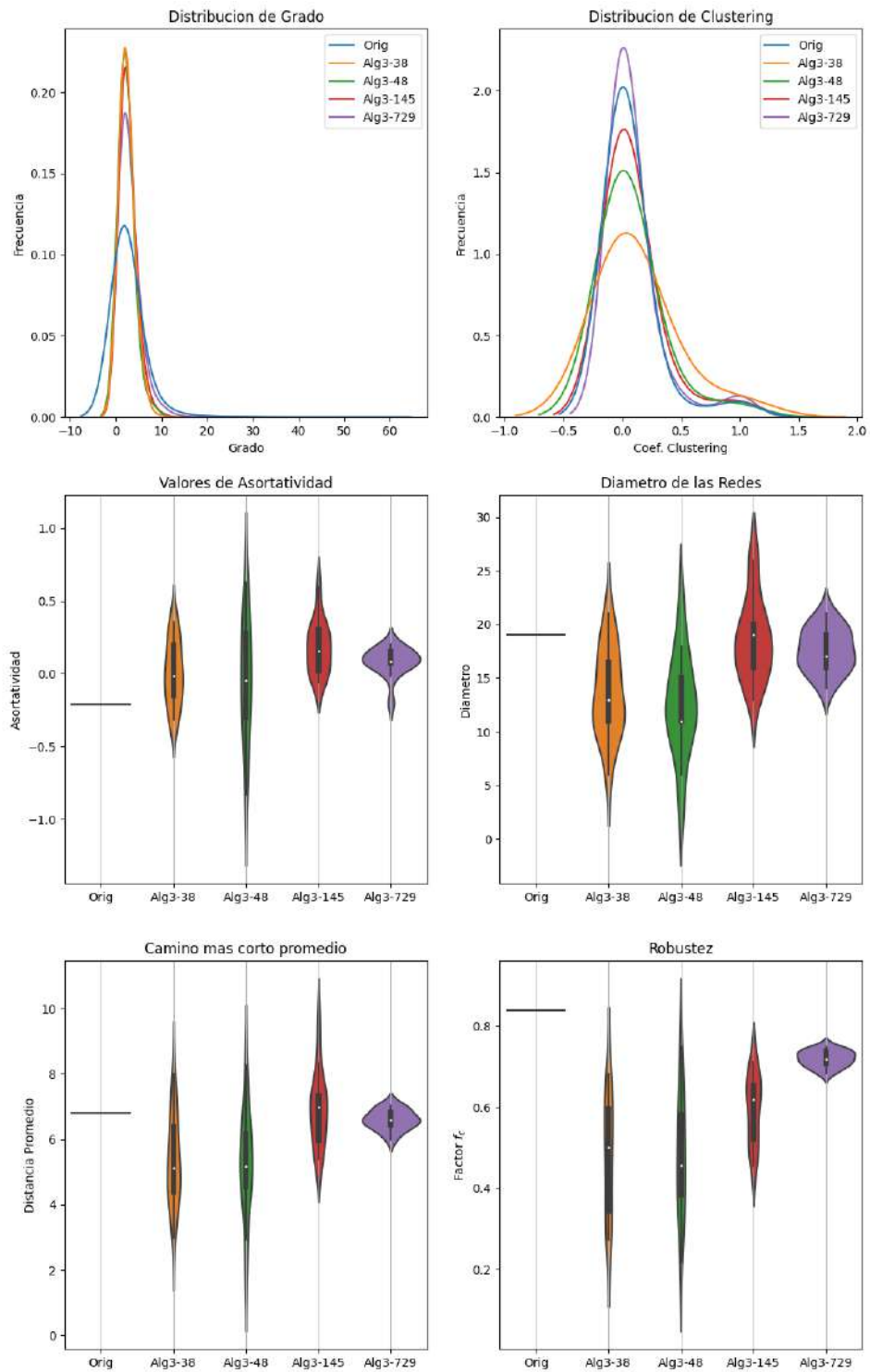


Figura 65: Comparación de resultados algoritmo 3 para red 1.
Fuente: Elaboración propia.

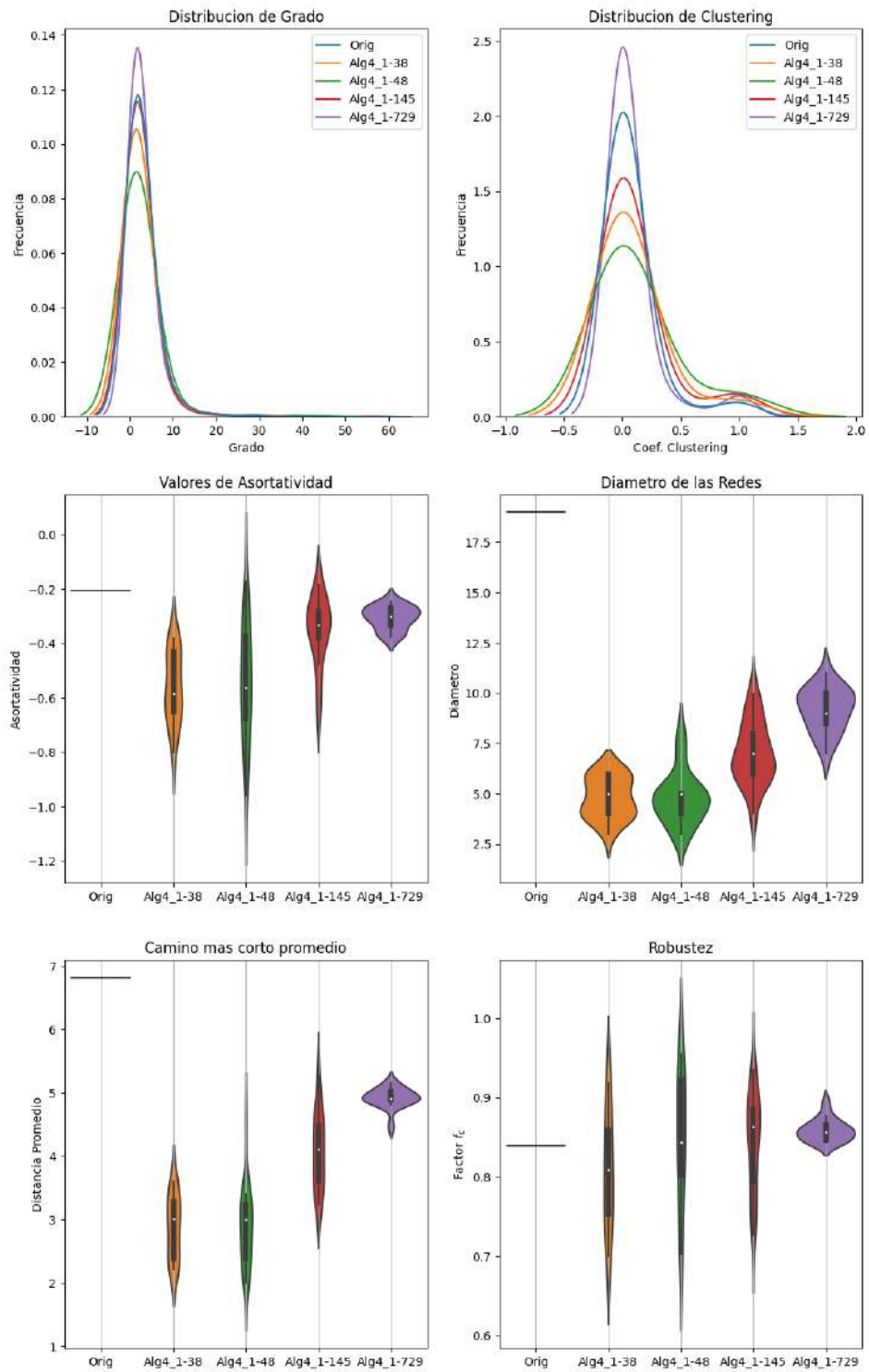


Figura 66: Comparación de resultados algoritmo 4 (BFS) para red 1.
Fuente: Elaboración propia.

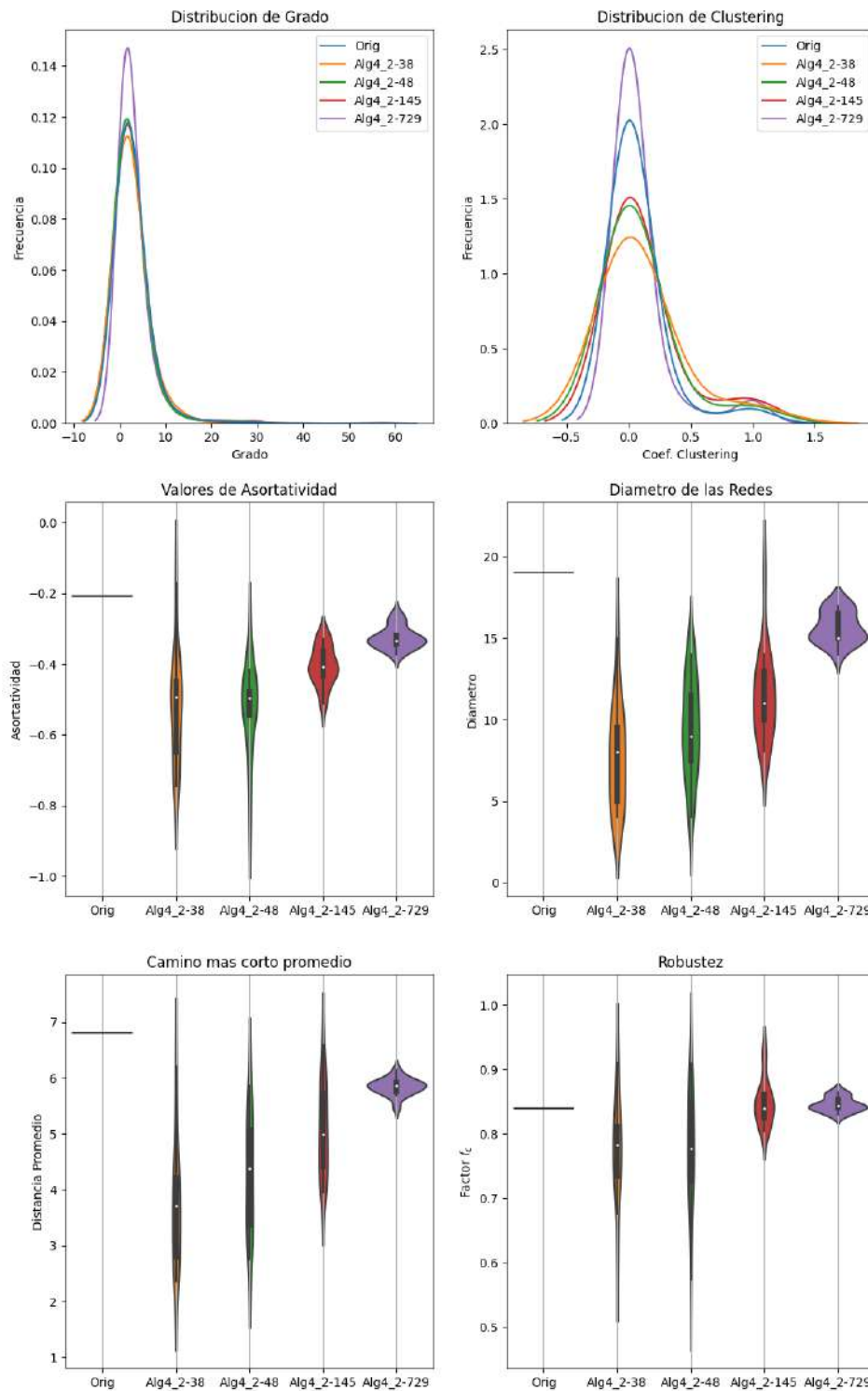


Figura 67: Comparación de resultados algoritmo 4 (DFS) para red 1.
Fuente: Elaboración propia.

7.17. Comparación de resultados restantes para Red 2

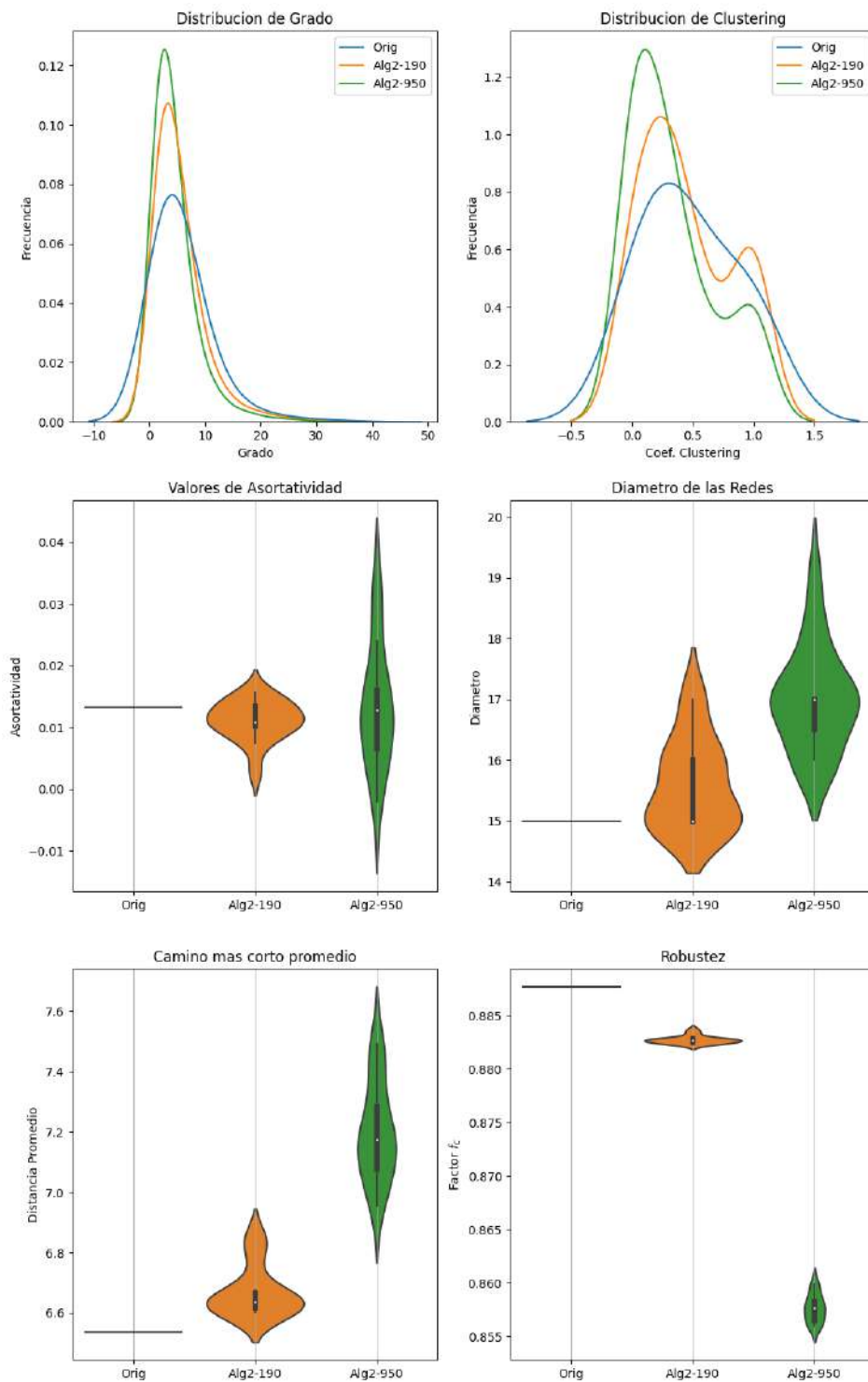


Figura 68: Comparación de resultados algoritmo 2 para red 2.

Fuente: Elaboración propia.

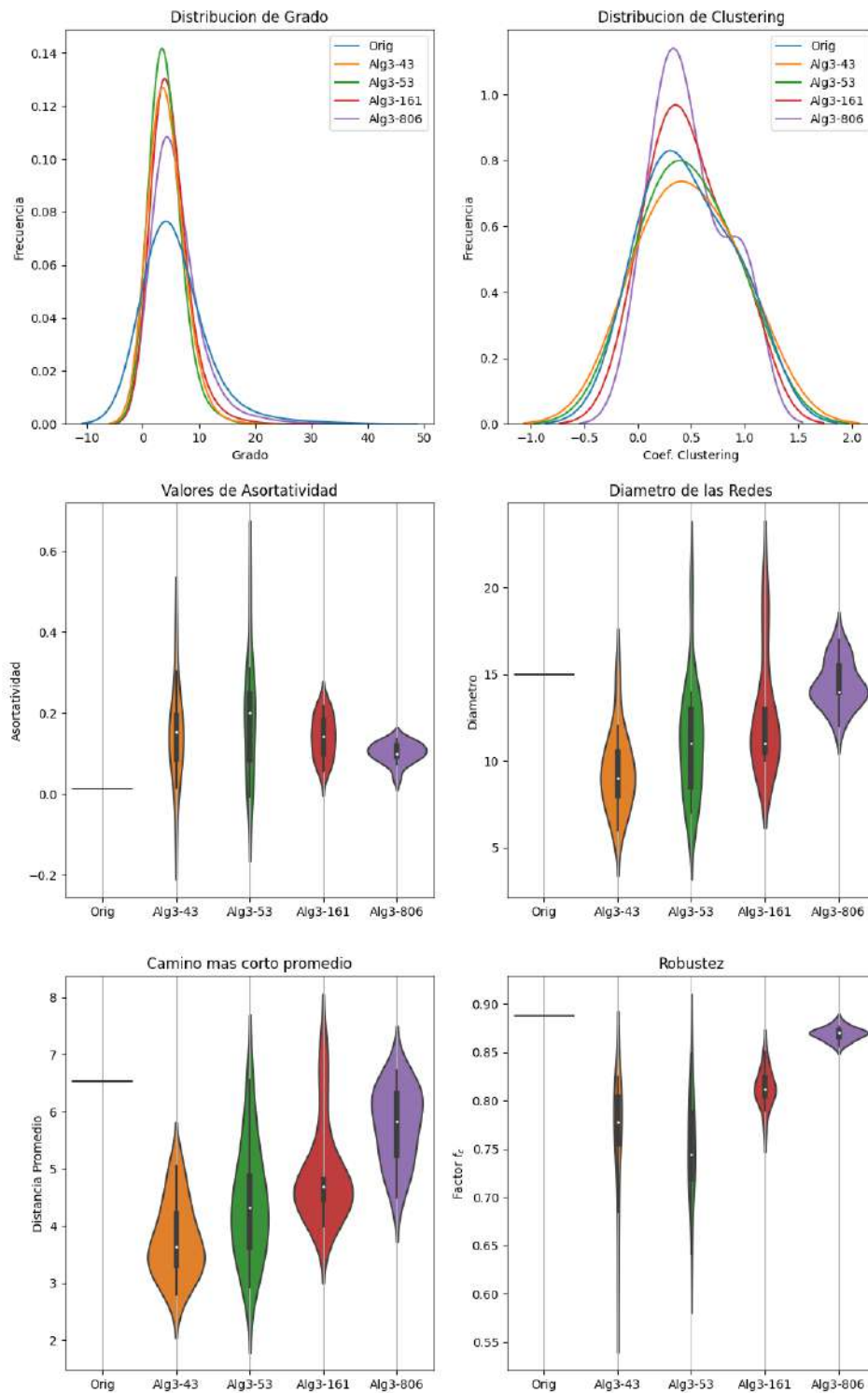


Figura 69: Comparación de resultados algoritmo 3 para red 2.

Fuente: Elaboración propia.

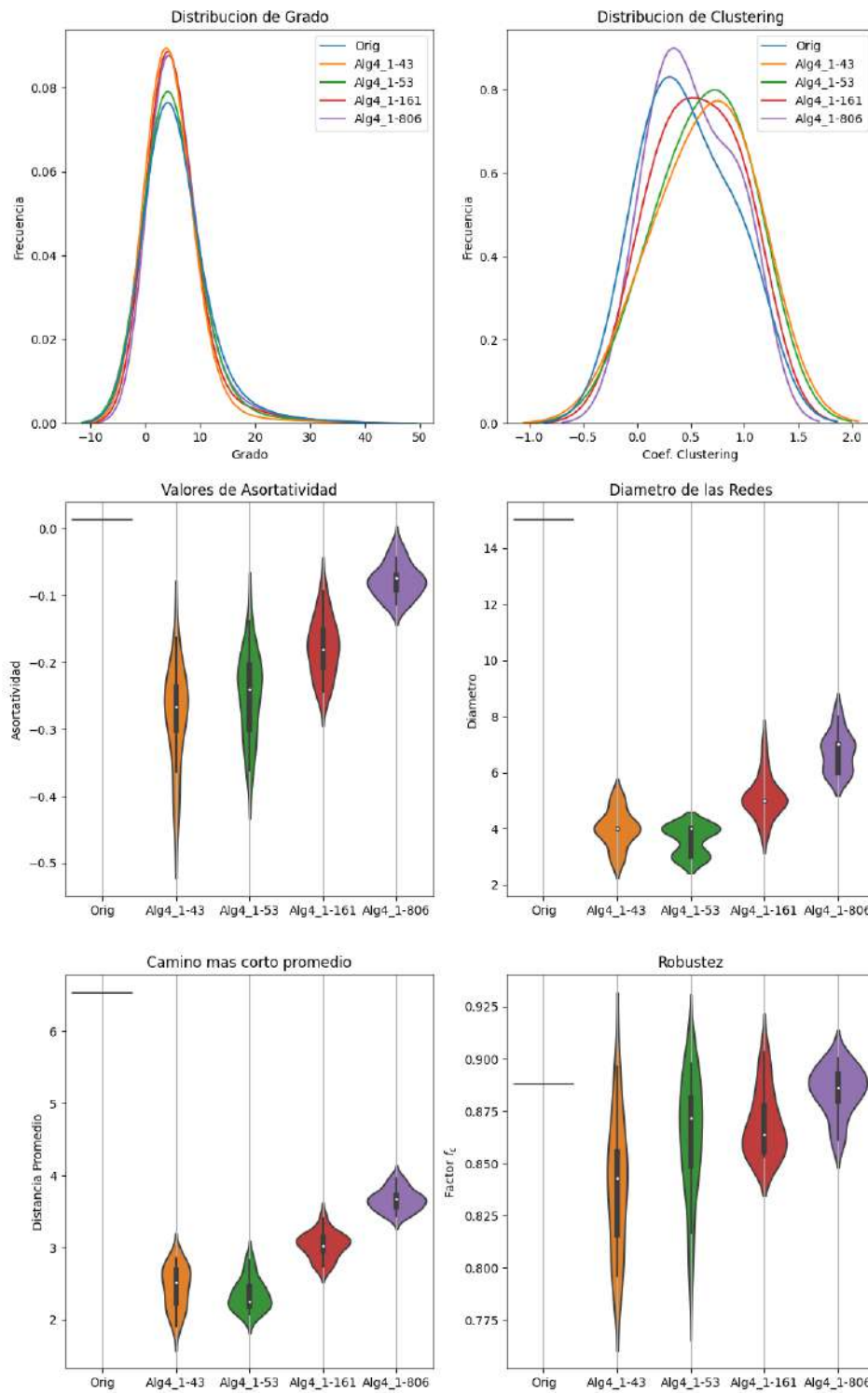


Figura 70: Comparación de resultados algoritmo 4 (BFS) para red 2.
Fuente: Elaboración propia.

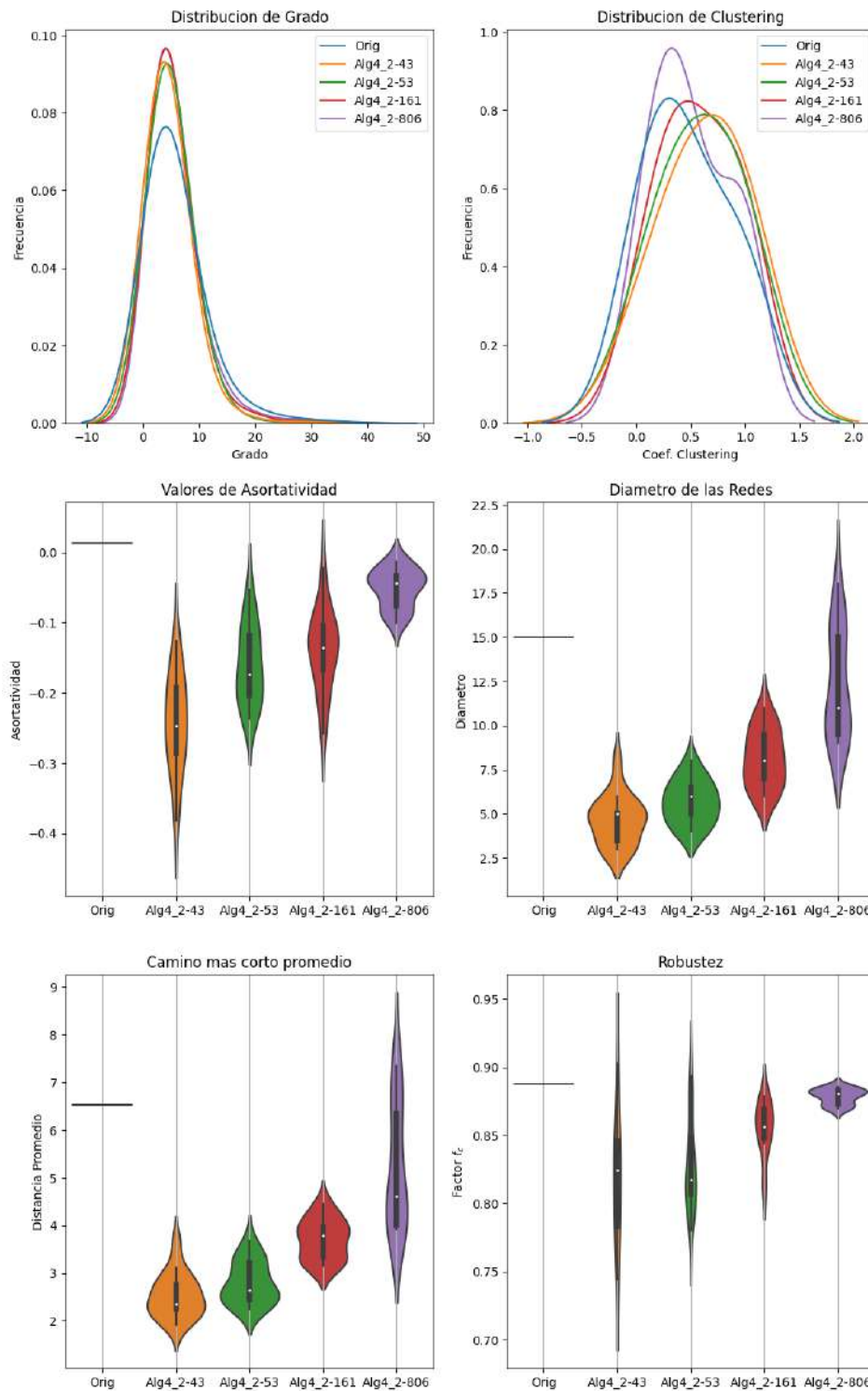


Figura 71: Comparación de resultados algoritmo 4 (DFS) para red 2.
Fuente: Elaboración propia.

7.18. Comparación de resultados restantes para Red 2

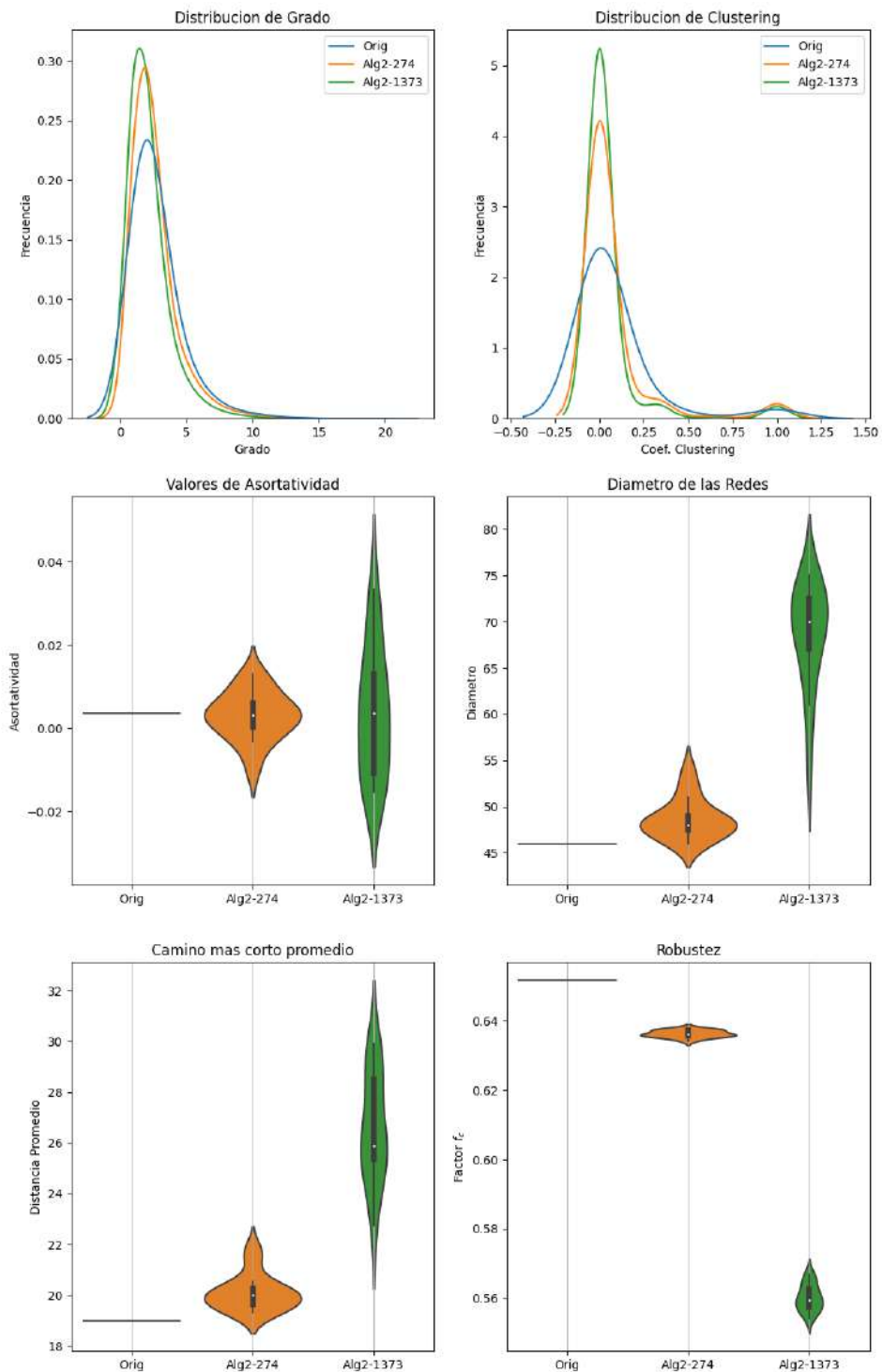


Figura 72: Comparación de resultados algoritmo 2 para red 3.
Fuente: Elaboración propia.

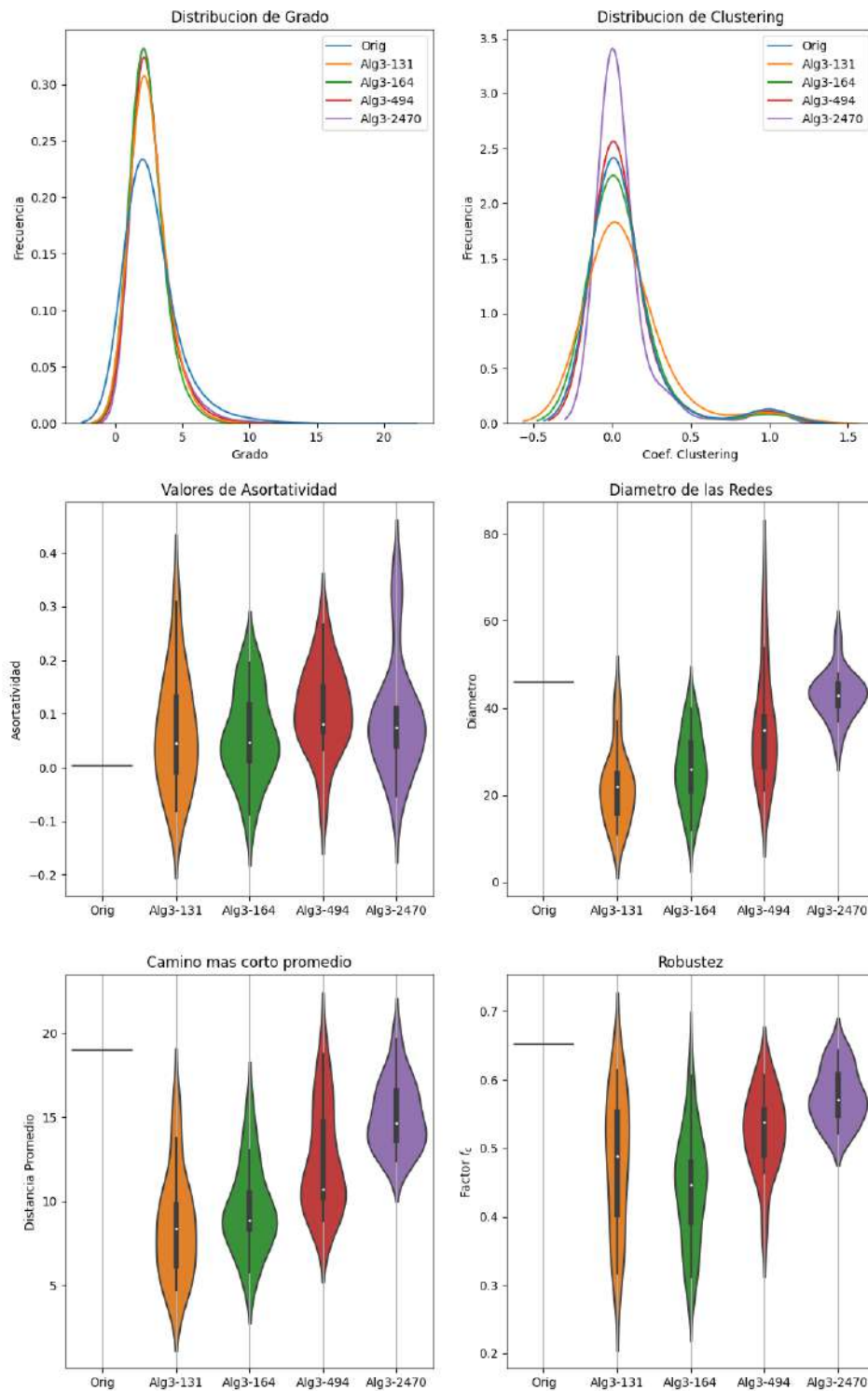


Figura 73: Comparación de resultados algoritmo 3 para red 3.

Fuente: Elaboración propia.

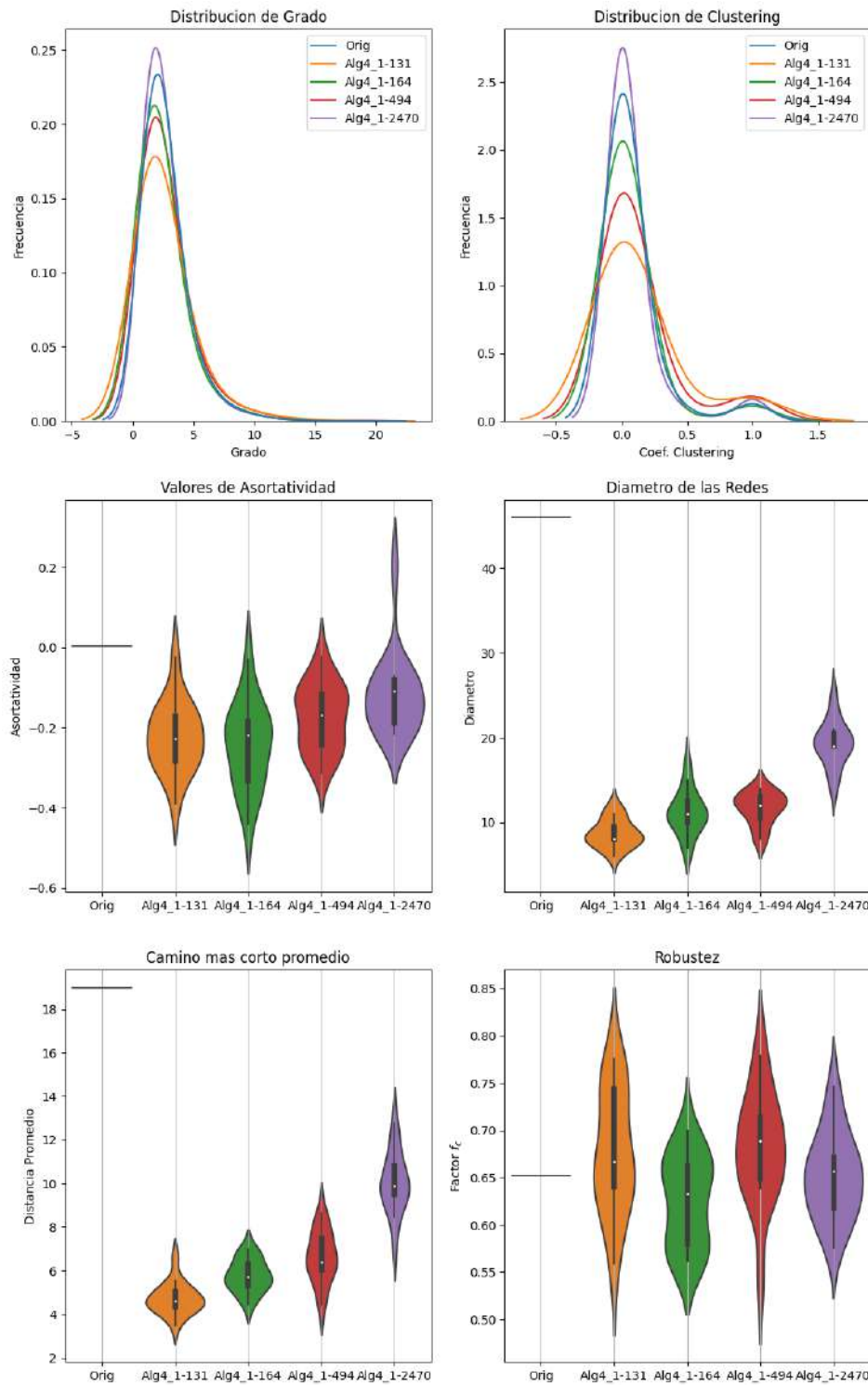


Figura 74: Comparación de resultados algoritmo 4 (BFS) para red 3.
Fuente: Elaboración propia.

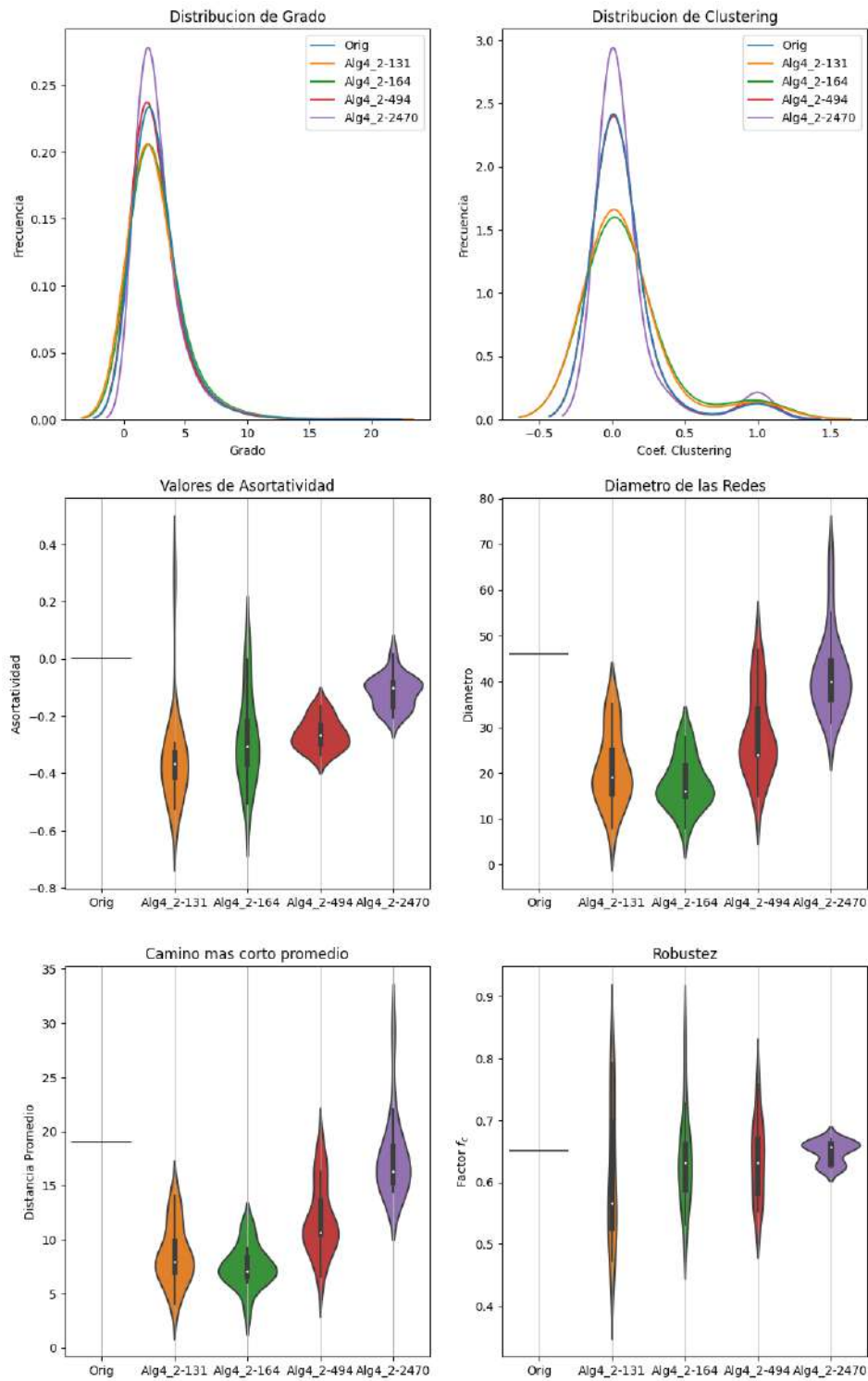


Figura 75: Comparación de resultados algoritmo 4 (DFS) para red 3.
Fuente: Elaboración propia.

7.19. Comparación de resultados restantes para Red 4

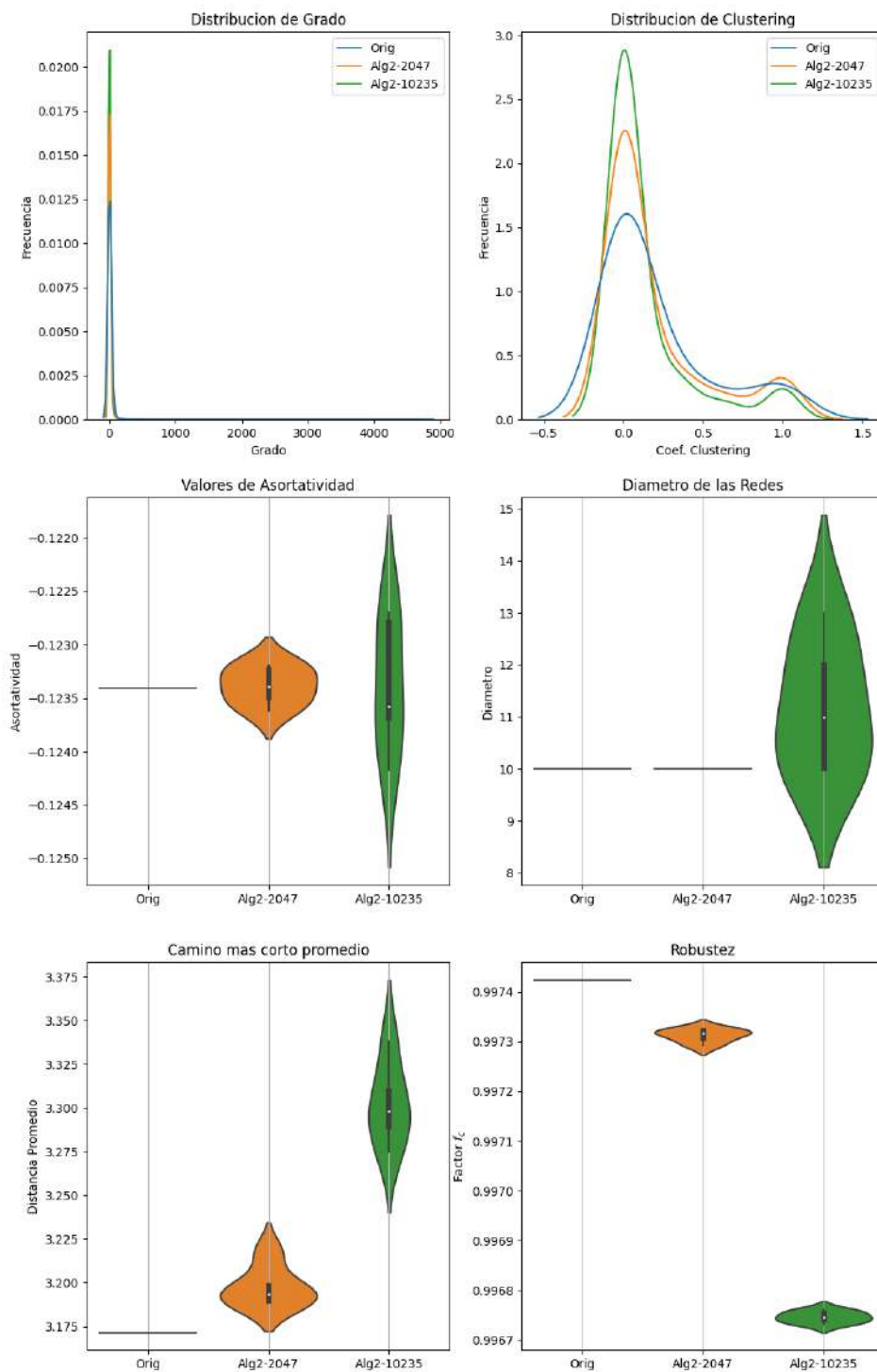


Figura 76: Comparación de resultados algoritmo 2 para red 4.
Fuente: Elaboración propia.

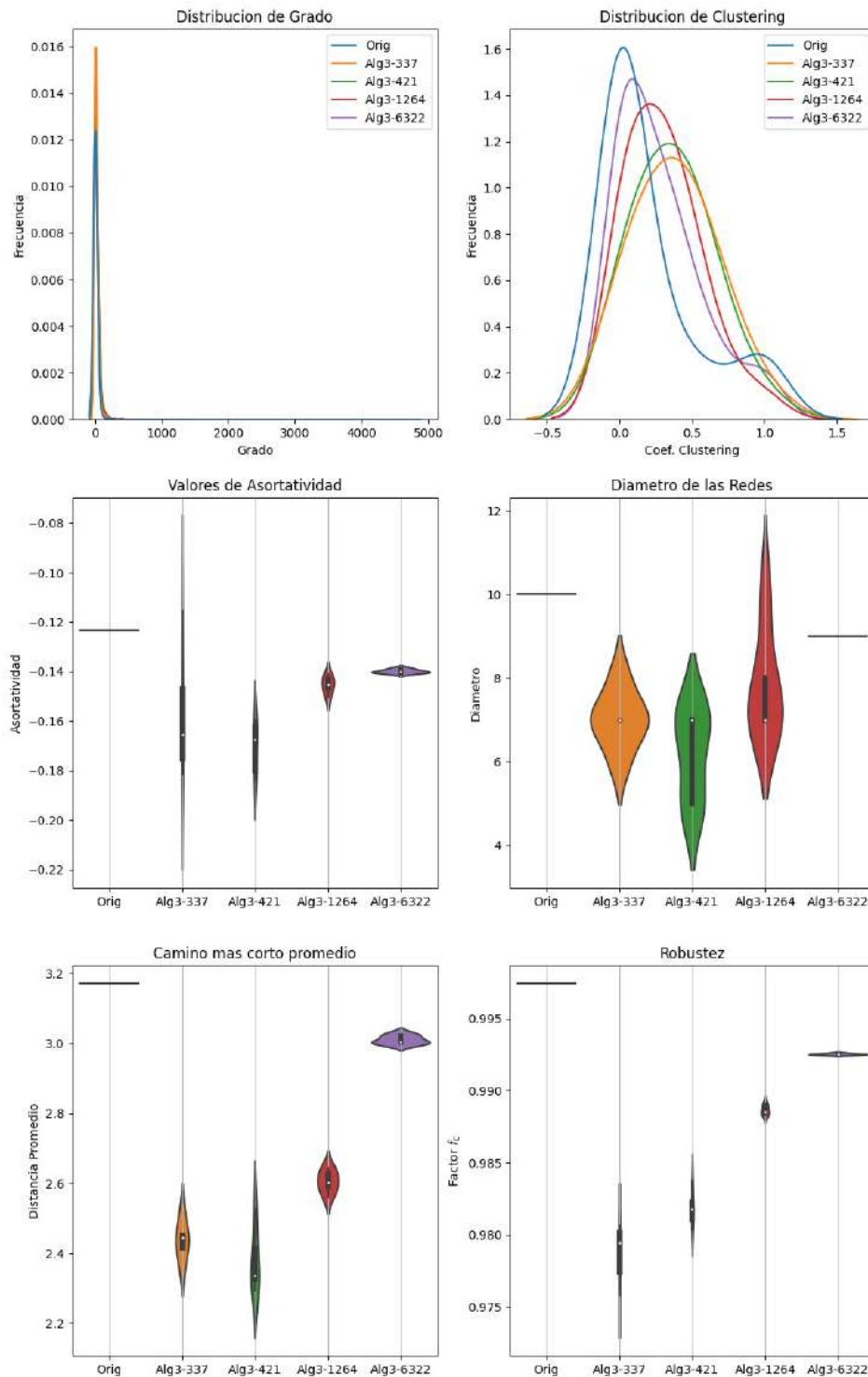


Figura 77: Comparación de resultados algoritmo 3 para red 4.
Fuente: Elaboración propia.

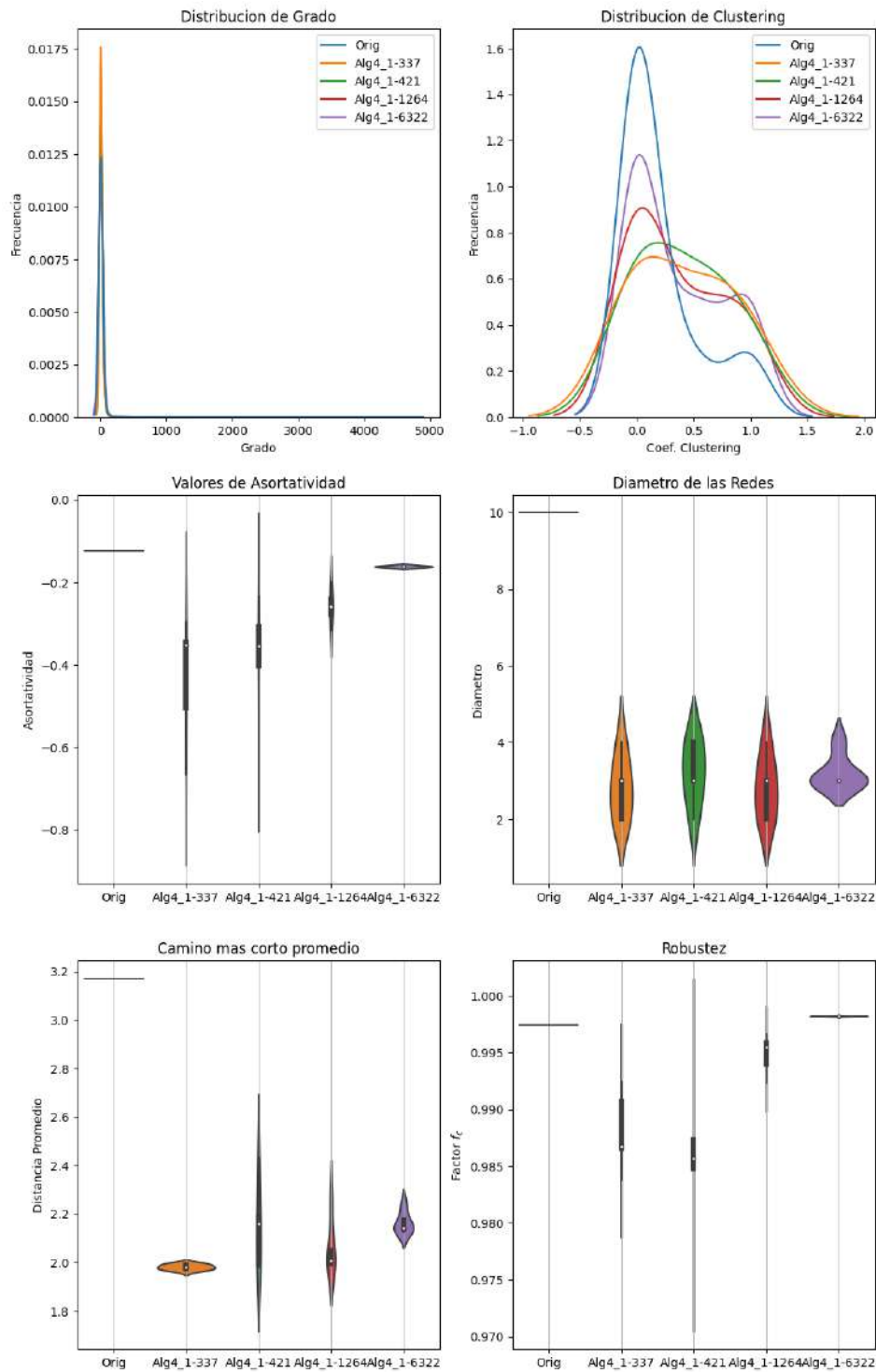


Figura 78: Comparación de resultados algoritmo 4 (BFS) para red 4.
Fuente: Elaboración propia.

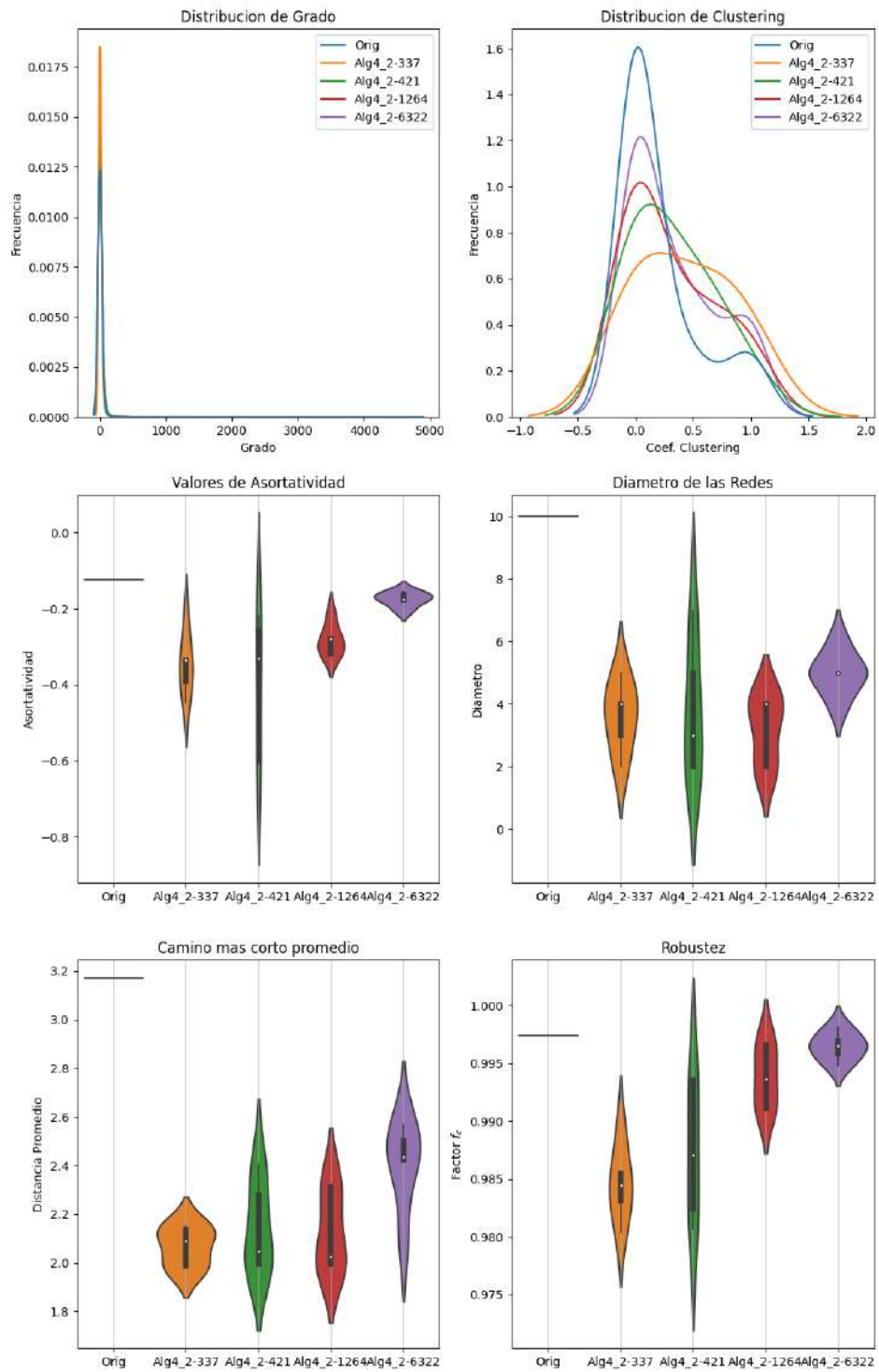


Figura 79: Comparación de resultados algoritmo 4 (DFS) para red 4.
Fuente: Elaboración propia.

7.20. Comparación de resultados restantes para Red 5

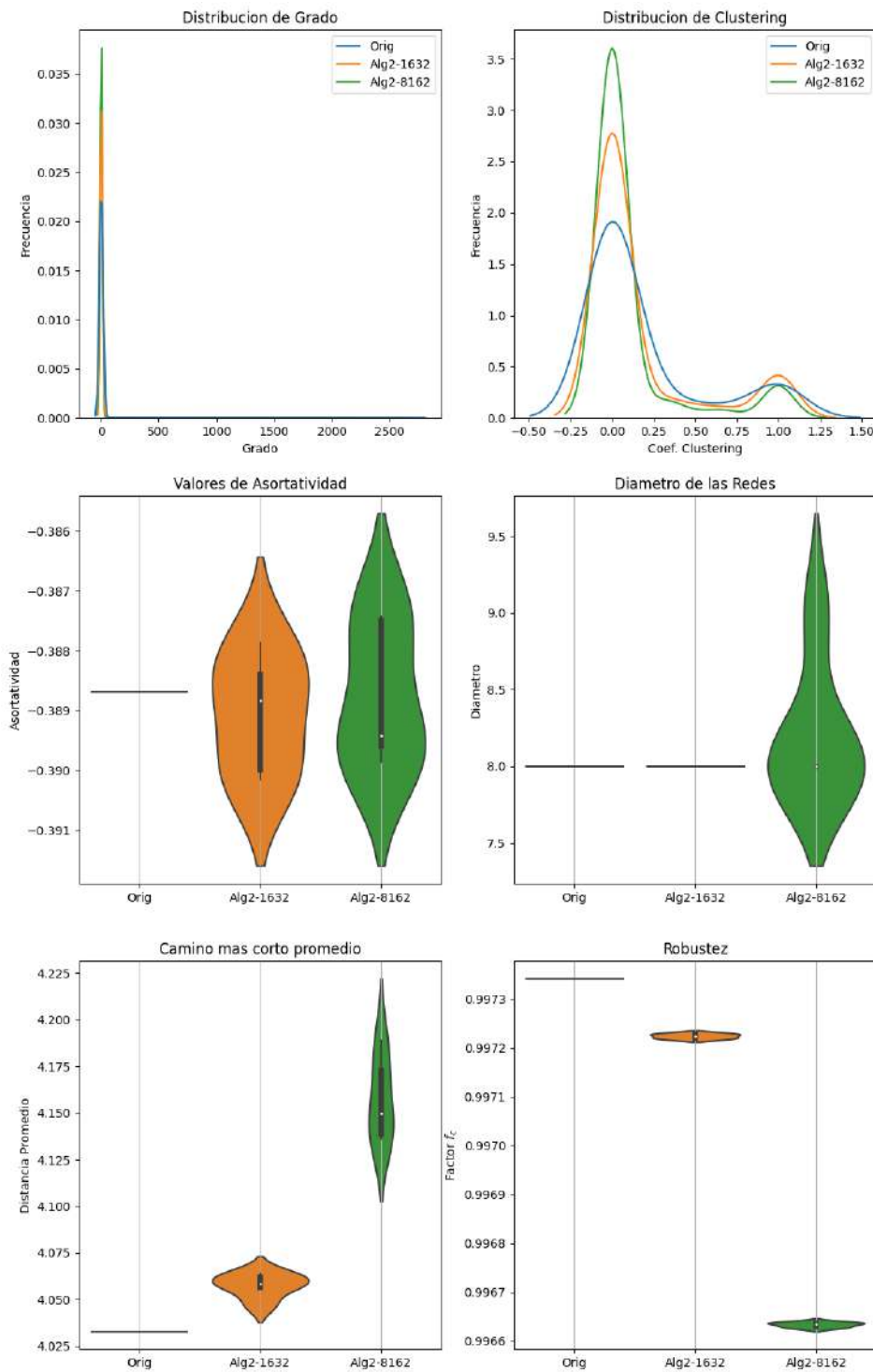


Figura 80: Comparación de resultados algoritmo 2 para red 5.

Fuente: Elaboración propia.

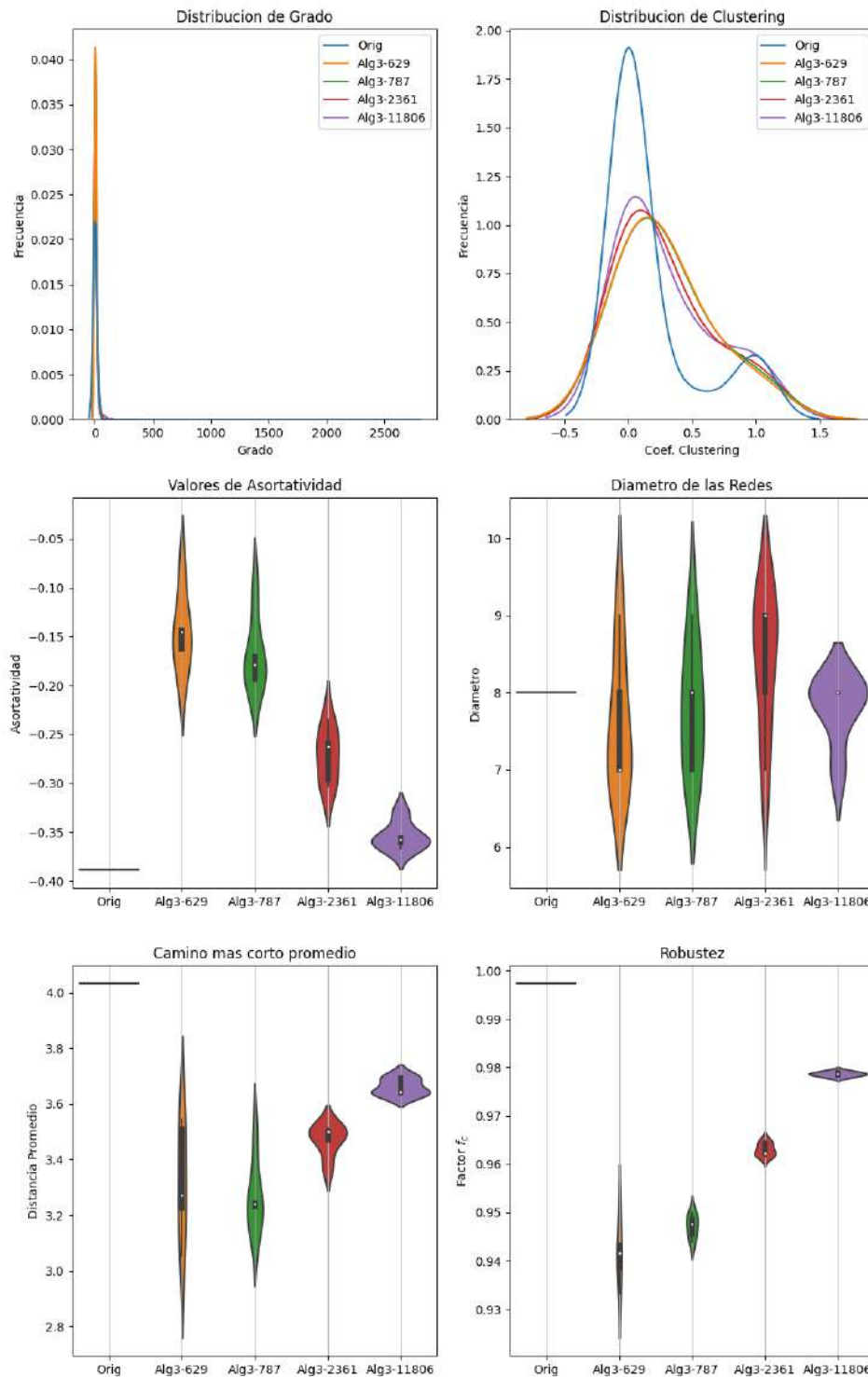


Figura 81: Comparación de resultados algoritmo 3 para red 5.
Fuente: Elaboración propia.

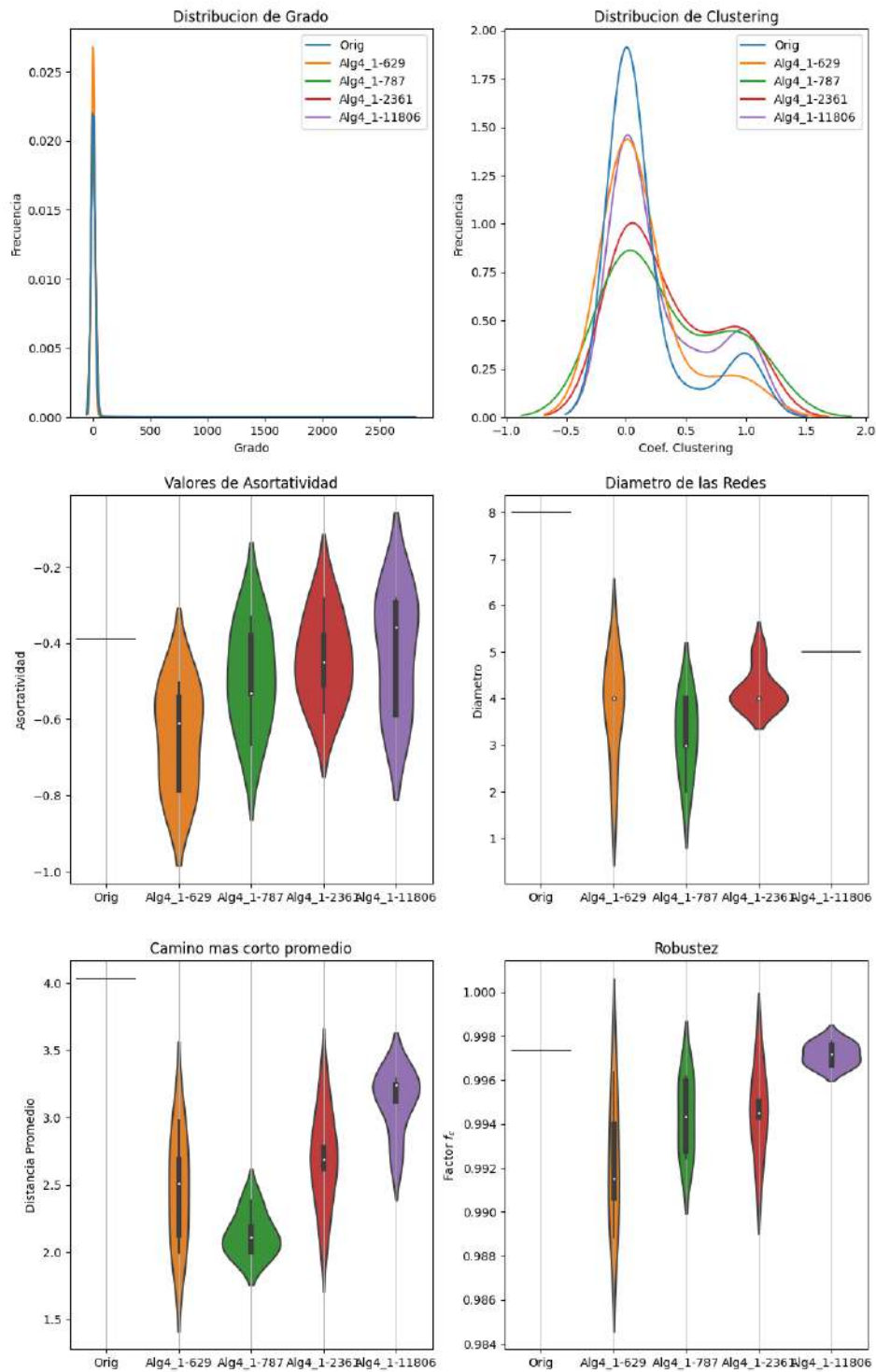


Figura 82: Comparación de resultados algoritmo 4 (BFS) para red 5.
Fuente: Elaboración propia.

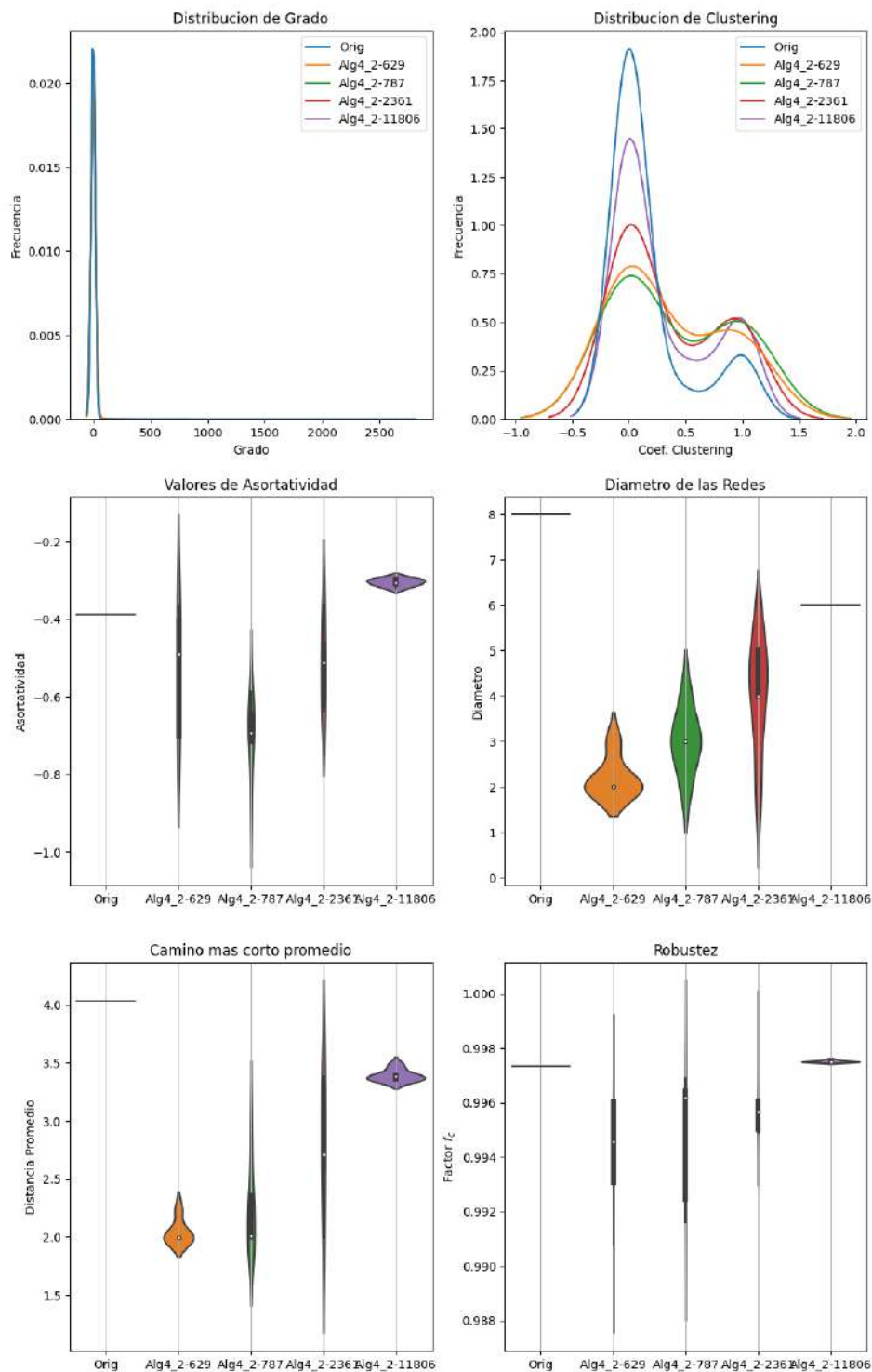


Figura 83: Comparación de resultados algoritmo 4 (DFS) para red 5.
Fuente: Elaboración propia.

REFERENCIAS BIBLIOGRÁFICAS

- [Arnaboldi *et al.*, 2016] Arnaboldi, V., Conti, M., La Gala, M., Passarella, A., y Pezzoni, F. (2016). Ego network structure in online social networks and its impact on information diffusion. *Computer Communications*, 76:26–41.
- [Barabási, 2016] Barabási, A. (2016). *Network Science*. Cambridge University Press, Cambridge.
- [Barabási y Albert, 1999] Barabási, A.-L. y Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.
- [Bollobás, 2001] Bollobás, B. (2001). The evolution of random graphs—the giant component. En *Random graphs*, volumen 184, pp. 130–59. Cambridge University Press Cambridge.
- [Borgatti *et al.*, 2006] Borgatti, S. P., Carley, K. M., y Krackhardt, D. (2006). On the robustness of centrality measures under conditions of imperfect data. *Social networks*, 28(2):124–136.
- [Brandes *et al.*, 2013] Brandes, U., Robins, G., McCranie, A., y Wasserman, S. (2013). What is network science? *Network science*, 1(1):1–15.
- [Cetinkaya, 2012] Cetinkaya, E. (2012). *Nan when calculating equal degree network*. Google Groups. <https://groups.google.com/g/networkx-discuss/c/o2zl40LMmqM>. Accedido: 27-06-2023.
- [Chartrand, 1977] Chartrand, G. (1977). *Introductory graph theory*. Courier Corporation.
- [Cohen *et al.*, 2000] Cohen, R., Erez, K., Ben-Avraham, D., y Havlin, S. (2000). Resilience of the internet to random breakdowns. *Physical Review Letters*, 85(21):4626.
- [Colizza *et al.*, 2006] Colizza, V., Flammini, A., Serrano, M. A., y Vespignani, A. (2006). Detecting rich-club ordering in complex networks. *Nature Physics*, 2(2):110–115.
- [Cooper *et al.*, 2014] Cooper, C., Radzik, T., y Siantos, Y. (2014). Estimating network parameters using random walks. *Social Network Analysis and Mining*, 4(1):1–19.
- [“david.pfx”, 2014] “david.pfx”(2014). *Algorithm to generate N random numbers between A and B which sum up to X. Respuesta de david.pfx*. <https://softwareengineering.stackexchange.com/a/254334>. Accedido: 28-03-2023.
- [Davis *et al.*, 2015] Davis, S., Abbasi, B., Shah, S., Telfer, S., y Begon, M. (2015). Spatial analyses of wildlife contact networks. *Journal of the Royal Society Interface*, 12(102):20141004.
- [Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs:(*Numerische Mathematik*, 1 (1959), p 269-271).

- [Erdős y Rényi, 1960] Erdős, P. y Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5(1):17–60.
- [Euler, 1741] Euler, L. (1741). *Solutio problematis ad geometriam situs pertinentis. Commentarii Academiae Scientiarum Petropolitanae*, pp. 128–140.
- [Fire et al., 2012] Fire, M., Puzis, R., y Elovici, Y. (2012). Link prediction in highly fractional data sets. *Handbook of Computational Approaches to Counterterrorism*.
- [Franceschet, 1999] Franceschet, M. (1999). *Shortest paths*. www.sci.unich.it/~francesc/teaching/network/geodesic.html. Accedido: 16-05-2023.
- [Granovetter, 1976] Granovetter, M. (1976). Network sampling: Some first steps. *American Journal of Sociology*, 81(6):1287–1303.
- [Gross, 2022] Gross, T. (2022). 20 years ago some groundbreaking papers on network robustness were among the advances that led to the inception of modern network science [...]. <https://twitter.com/thilogross/status/1494285806248341514>.
- [Holland et al., 1983] Holland, P. W., Laskey, K. B., y Leinhardt, S. (1983). Stochastic block-models: First steps. *Social networks*, 5(2):109–137.
- [Hristov, 2015] Hristov, D. (2015). Investigating dmos through the lens of social network analysis: Theoretical gaps, methodological challenges and practitioner perspectives. *Advances in Hospitality and Tourism Research (AHTR)*, 3(1):18–39.
- [Jeong et al., 2001] Jeong, H., Mason, S. P., Barabási, A.-L., y Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, 411(6833):41–42.
- [Kafri et al., 2008] Kafri, R., Dahan, O., Levy, J., y Pilpel, Y. (2008). Preferential protection of protein interaction network hubs in yeast: evolved functionality of genetic redundancy. *Proceedings of the National Academy of Sciences*, 105(4):1243–1248.
- [Khraisha, 2020] Khraisha, T. (2020). *Industry Applications of Network Science and Graph Algorithms*. *LinkedIn*. <https://www.linkedin.com/pulse/industry-applications-network-science-graph-tamer-khraisha-phd>. Accedido: 13-03-2023.
- [Kossinets, 2006] Kossinets, G. (2006). Effects of missing data in social networks. *Social networks*, 28(3):247–268.
- [Lancichinetti et al., 2011] Lancichinetti, A., Radicchi, F., Ramasco, J. J., y Fortunato, S. (2011). Finding statistically significant communities in networks. *PloS one*, 6(4):e18961.
- [Lee y Wilkinson, 2019] Lee, C. y Wilkinson, D. J. (2019). A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):1–50.

- [Lehman et al., 2010] Lehman, E., Leighton, F., y Meyer, A. (2010). *Mathematics for Computer Science. LibreTexts*. [https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_and_Computation_Fundamentals/Mathematics_for_Computer_Science_\(Lehman_Leighton_and_Meyer\)/](https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_and_Computation_Fundamentals/Mathematics_for_Computer_Science_(Lehman_Leighton_and_Meyer)/). Accedido: 16-05-2023.
- [Levina et al., 2022] Levina, A., Priesemann, V., y Zierenberg, J. (2022). Tackling the subsampling problem to infer collective properties from limited data. *Nature Reviews Physics*, 4(12):770–784.
- [Litant, 2013] Litant, T. (2013). Dark networks and the missing link inference problem. *American Intelligence Journal*, 31(1):102–115.
- [Lucchetta et al., 2023] Lucchetta, M., List, M., Blumenthal, D. B., y Schaefer, M. H. (2023). Emergence of power-law distributions in protein-protein interaction networks through study bias. *BioRxiv*, pp. 2023–03.
- [NetworkX,] NetworkX, D. Documentación networkx. <https://networkx.org/documentation/stable/reference/algorithms/traversal.html>. Accedido: 10-08-2023.
- [Newman, 2002] Newman, M. E. (2002). Assortative mixing in networks. *Physical Review Letters*, 89(20):208701.
- [Newman, 2003] Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*, 67(2):026126.
- [Newth, 2006] Newth, D. (2006). The structure of social networks. *Complex Science for a Complex World*, p. 71.
- [Ribeiro et al., 2012] Ribeiro, B., Wang, P., Murai, F., y Towsley, D. (2012). Sampling directed graphs with random walks. En *2012 Proceedings IEEE Infocom*, pp. 1692–1700. IEEE.
- [Rossi y Ahmed, 2015] Rossi, R. y Ahmed, N. (2015). The network data repository with interactive graph analytics and visualization. En *Proceedings of the AAAI conference on artificial intelligence*, volumen 29.
- [Rossi y Ahmed,] Rossi, R. A. y Ahmed, N. K. Network repository. <https://networkrepository.com/>. Accedido: 10-08-2023.
- [Sarkar et al., 2016] Sarkar, S., Kumar, S., Bhowmick, S., y Mukherjee, A. (2016). Sensitivity and reliability in incomplete networks: Centrality metrics to community scoring functions. En *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 69–72. IEEE.
- [Upadhyay, 2023] Upadhyay, S. (2023). *Bellman-Ford Algorithm: Pseudocode, Time Complexity and Examples. Simplilearn*. https://www.simplilearn.com/tutorials/data-structure-tutorial/bellman-ford-algorithm#the_complexity_of_bellmanford_algorithm. Accedido: 25-05-2023.

- [Varga, 2015] Varga, I. (2015). Scale-free network topologies with clustering similar to on-line social networks. En *Proceedings of the international conference on social modeling and simulation, plus econophysics colloquium 2014*, pp. 323–333. Springer International Publishing.
- [Warnke, 2016] Warnke, S. D. (2016). Partial information community detection in a multilayer network. Technical report, Naval Postgraduate School Monterey United States.
- [Watts y Strogatz, 1998] Watts, D. J. y Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442.
- [Yan y Gregory, 2011] Yan, B. y Gregory, S. (2011). Finding missing edges and communities in incomplete networks. *Journal of Physics A: Mathematical and Theoretical*, 44(49):495102.
- [Özçelik, 2019] Özçelik, R. (2019). *An Intuitive Explanation of DeepWalk*. Medium. <https://medium.com/analytics-vidhya/an-intuitive-explanation-of-deepwalk-84177f7f2b72>. Accedido: 10-12-2022.