



Facultad de Ingeniería  
Escuela de Ingeniería Informática

# **CENTRALIDAD DE INDIVIDUOS EN REDES SOCIALES BASADA EN MODELOS DE DISPERSIÓN DE INFLUENCIA**

Por

**José Vera Nahuelcoy**

Trabajo realizado para optar al Título de  
**INGENIERO CIVIL EN INFORMÁTICA**

Prof. Guía: Fabián Riquelme Csori

Enero 2022

Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.



---

Fabián Riquelme Csori Profesor Guía

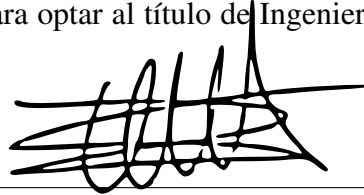
Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.



---

Eliana Providel Godoy Profesor Informante

Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.



---

Rodrigo Olivares Órdenes Profesor Informante

Aprobado por la Escuela de Ingeniería en Informática, UNIVERSIDAD DE VALPARAÍSO.

# Resumen

La centralidad y la dispersión de influencia son dos de los conceptos más importantes en la disciplina de análisis en redes sociales. La centralidad determina los actores más relevantes de la red y la dispersión de influencia aborda el tópico de viralización en la red. Estos dos conceptos usualmente han sido estudiados de manera independiente a pesar de su relación directa. Hace unos años comenzaron algunos estudios que utilizan estos dos conceptos y los trabajan en conjunto, dando como resultado la creación de una nueva medida de centralidad, llamada *Linear Threshold Rank* que está basada en el modelo de dispersión de influencia conocido como *Linear Threshold Model*. En este trabajo se generalizará esta medida, tomando ideas de otra medida de centralidad llamada *Diffusion Degree*, y sumando otro conocido modelo de dispersión de influencia, llamado *Independent Cascade Model*. El objetivo es comprender el rol de las vecindades de los nodos para dispersar la influencia. Esta nueva generalización se puso a prueba en redes de casos de laboratorio, donde se obtuvieron patrones de comportamiento que rigen sobre las redes que varía según la densidad de la red en base a las vecindades de los nodos.

# Agradecimientos

Quiero agradecer a mi familia, quienes me han acompañado y apoyado en toda decisión que he tomado respecto a mis estudios y vida llevándome a ser quien soy. A mis gatas que gracias a su compañía permitieron hacer más llevaderos los días de desarrollo de este trabajo. A mis amigos y compañeros de carrera que siempre tenían algún consejo que ofrecerme o alguna palabra de aliento cuando me estancaba. A mi profesor guía el Dr. Fabián Riquelme, quien siempre estuvo ahí dándome consejos, ayuda y confianza, sin él nada de este trabajo sería igual dado que me apoyó en cada etapa, mucho más de lo que hubiese podido imaginar que sería un acompañamiento de profesor guía. Finalmente, quiero agradecer a la Escuela de Ingeniería Civil Informática, que me brindó las herramientas para formarme como profesional, a todos los docentes pertenecientes a esta escuela, más de alguno me ofreció su ayuda, alguna reunión o consejo y estoy seguro que todos hubiesen ofrecido lo mismo dado que cada uno de ellos comparten la misma vocación de guiar nuevos profesionales por un buen camino.

Este trabajo también cuenta con el apoyo del proyecto Fondecyt de Iniciación 2020 titulado “The role of individual and group centrality in the influence spread process within social network”, Folio 11200113.

# Índice general

<b>Resumen</b>	<b>III</b>
<b>Agradecimientos</b>	<b>IV</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Marco conceptual y estado del arte</b>	<b>4</b>
2.1. Marco conceptual	4
2.1.1. Medidas de centralidad	5
2.1.2. Modelos de dispersión de influencia	6
2.2. Estado del arte	11
2.2.1. Medidas de centralidad basadas en modelos de dispersión de influencia	11
<b>3. Definición del problema</b>	<b>16</b>
3.1. Formulación del problema	16
3.2. Solución propuesta	17
3.3. Objetivos	17
3.3.1. Objetivo general	17
3.3.2. Objetivos específicos	17
3.4. Metodología	18
3.4.1. Metodología de investigación	18
3.4.2. Metodología de desarrollo	19
<b>4. Diseño de la solución</b>	<b>21</b>
4.1. Solución del problema	21
4.1.1. General Influence Spread Rank	21
4.1.2. Plataforma de experimentación	23
4.2. Diseño de pruebas	25
4.2.1. Pruebas de la plataforma	25
4.2.2. Pruebas de experimentación	25

4.3. Técnicas de análisis . . . . .	26
<b>5. Implementación</b>	<b>27</b>
5.1. Software utilizado . . . . .	27
5.2. Hardware utilizado . . . . .	32
5.3. Lenguajes de programación . . . . .	32
5.4. Estrategia de implementación . . . . .	33
5.5. Pruebas de la plataforma . . . . .	35
5.5.1. Pruebas a la plataforma . . . . .	35
5.5.2. Diseño de experimentos . . . . .	35
5.5.3. Datasets . . . . .	36
5.5.4. Interpretación de resultados . . . . .	38
<b>6. Experimentación</b>	<b>42</b>
6.1. Experimentos en la plataforma . . . . .	43
6.1.1. Datasets . . . . .	43
6.1.2. Ejecuciones de la plataforma . . . . .	45
6.2. Resultados . . . . .	46
6.3. Análisis de resultados . . . . .	47
6.3.1. Análisis de tiempos de ejecución . . . . .	47
6.3.2. Análisis gráfico de parámetros . . . . .	48
6.3.3. Análisis matrices de correlaciones . . . . .	60
6.4. Discusión de resultados . . . . .	77
6.4.1. Discusión de los resultados estadísticos . . . . .	77
6.4.2. Discusión de las matrices de correlación . . . . .	78
<b>7. Implantación</b>	<b>80</b>
7.1. Requerimientos . . . . .	80
7.1.1. Requerimientos de mínimos . . . . .	80
7.1.2. Requerimientos de recomendados . . . . .	81
7.2. Preparación de ambiente . . . . .	81
7.3. Manual de usuario . . . . .	82
<b>8. Conclusiones</b>	<b>83</b>
<b>Bibliografía</b>	<b>85</b>
<b>A. Matrices de análisis</b>	<b>92</b>

<b>B. Manual usuario plataforma experimentación GISR</b>	<b>99</b>
B.1. Índice	99
B.2. Introducción	99
B.3. Requisitos previos	100
B.4. Instalación y configuración	100
B.4.1. Configuración	100
B.5. Manejo del sistema	104
B.6. Cambios del sistema	109
B.7. Comentario finales	110

# Índice de tablas

5.1. Matriz de adyacencia dataset preparado para LT-Model.	36
5.2. Etiquetado dataset preparado para LT-Model.	37
5.3. Archivo de prueba para IC-Model.	37
5.4. Archivo de prueba para IC-Model.	38
5.5. Ejemplo ajuste para IC-Model.	38
6.1. Redes sociales utilizadas para los experimentos.	43
6.2. Valores de los parámetros en los experimentos.	45
6.3. Niveles de correlación (correlación inversa sigue el mismo rango pero con valores negativos).	61
A.1. Matriz de resultados para red Football en modelo IC.	93
A.2. Matriz de resultados para red Bitcoin Alpha en modelo IC.	94
A.3. Matriz de resultados para red Higgs en modelo IC.	95
A.4. Matriz de resultados para red Football en modelo LT.	96
A.5. Matriz de resultados para red Bitcoin Alpha en modelo LT.	97
A.6. Matriz de resultados para red Higgs Twitter en modelo LT.	98

# Índice de figuras

2.1. Linear Threshold Model (elaboración propia). Nodos amarillos representan nodos activos. . . . .	8
2.2. Independent Cascade Model (elaboración propia). Nodos amarillos representan nodos activos que pueden influenciar. Nodos grises presentan nodos activos que no pueden volver a influenciar. . . . .	10
2.3. Linear Threshold Rank con 1 nivel de profundidad (elaboración propia). Nodos amarillos representan nodos activos. . . . .	13
2.4. Diffusion Degree (elaboración propia). . . . .	15
3.1. Proceso de Metodología Cuantitativa. . . . .	19
3.2. Diagrama basado en metodología de prototipado (elaboración propia). . . . .	20
4.1. Diagrama de etapas . . . . .	24
5.1. Ejemplo de celda en Jupyter Notebook. . . . .	27
5.2. Vista principal del proyecto en GitHub. . . . .	28
5.3. Vista de edición en Visual Studio Code. . . . .	29
5.4. Vista de Spyder. . . . .	30
5.5. Vista de el uso de trello. . . . .	31
5.6. Ejemplo del patrón estructural aplicado a la plataforma de pruebas. . . . .	34
5.7. Ejecución medida generalizada nivel de profundidad 0 . . . . .	39
5.8. Ejecución medida generalizada nivel de profundidad 1 . . . . .	40
5.9. Ejecución medida generalizada nivel de profundidad 4 . . . . .	40
5.10. Ejecución medida generalizada nivel de profundidad 6 . . . . .	41
5.11. Ejecución medida generalizada nivel de profundidad 10 . . . . .	41
6.1. Ejemplo de archivo de resultados. . . . .	47
6.2. Ejemplo de archivo de resultados. . . . .	47
6.3. Gráfico de resultados red Football direccionalidad salida modelo IC. . . . .	49
6.4. Gráfico de resultados red Football direccionalidad Entrada modelo IC. . . . .	50
6.5. Gráfico de resultados red Football direccionalidad ambas modelo IC. . . . .	50
6.6. Gráfico de resultados red Football direccionalidad salida modelo LT. . . . .	51

6.7. Gráfico de resultados red Football direccionalidad Entrada modelo LT. . . . .	51
6.8. Gráfico de resultados red Football direccionalidad ambas modelo LT. . . . .	52
6.9. Gráfico de resultados red Bitcoin Alpha direccionalidad salida modelo IC. . . . .	53
6.10. Gráfico de resultados red Bitcoin Alpha direccionalidad entrada modelo IC. . . . .	53
6.11. Gráfico de resultados red Bitcoin Alpha direccionalidad Ambas modelo IC. . . . .	54
6.12. Gráfico de resultados red Bitcoin Alpha direccionalidad salida modelo LT. . . . .	55
6.13. Gráfico de resultados red Bitcoin Alpha direccionalidad entrada modelo LT. . . . .	55
6.14. Gráfico de resultados red Bitcoin Alpha direccionalidad ambas modelo LT. . . . .	56
6.15. Gráfico de resultados red Higgs Twitter direccionalidad salida modelo IC. . . . .	57
6.16. Gráfico de resultados red Higgs Twitter direccionalidad entrada modelo IC. . . . .	57
6.17. Gráfico de resultados red Higgs Twitter direccionalidad ambas modelo IC. . . . .	58
6.18. Gráfico de resultados red Higgs Twitter direccionalidad salida modelo LT. . . . .	58
6.19. Gráfico de resultados red Higgs Twitter direccionalidad entrada modelo LT. . . . .	59
6.20. Gráfico de resultados red Higgs Twitter direccionalidad ambas modelo LT. . . . .	59
6.21. Ejemplo de los códigos de colores y valores de correlación. . . . .	61
6.22. Extracto de ejemplo de una matriz real. . . . .	61
6.23. Vista general de la matriz Football. . . . .	62
6.24. Matriz Football sección IC-Model. . . . .	63
6.25. Matriz Football sección LT-Model. . . . .	64
6.26. Matriz Football sección IC-Model y LT-Model. . . . .	65
6.27. Matriz Football medidas clásicas con GISR. . . . .	66
6.28. Vista general de la matriz Bitcoin Alpha. . . . .	67
6.29. Matriz Bitcoin Alpha sección IC-Model. . . . .	69
6.30. Matriz Bitcoin Alpha sección LT-Model. . . . .	70
6.31. Matriz Bitcoin Alpha sección IC-Model y LT-Model. . . . .	71
6.32. Matriz Bitcoin Alpha medidas clásicas con GISR. . . . .	72
6.33. Vista general de la matriz matriz Higgs Twitter . . . . .	73
6.34. Matriz Higgs Twitter sección IC-Model . . . . .	74
6.35. Matriz Higgs Twitter sección LT-Model . . . . .	75
6.36. Matriz Higgs Twitter sección IC-Model y LT-Model . . . . .	76
6.37. Matriz Higgs Twitter sección medidas clásicas con GISR . . . . .	77
B.1. Extracto de archivo molde.py. . . . .	101
B.2. Extracto de archivo moldeado. . . . .	102
B.3. Extracto del Código Fuente a editar. . . . .	103
B.4. Ejecución plataforma con parámetro de ayuda. . . . .	103
B.5. Ejecución plataforma con nodo específico. . . . .	104
B.6. Ejecución plataforma sin nodo específico. . . . .	105
B.7. Resultado ejecución sin nodo específico. . . . .	105
B.8. Extracto de archivo de cálculo estadísticas. . . . .	106

<b>B.9. Archivo de resultados estadísticos.</b> . . . . .	107
<b>B.10. Extracto de archivo cálculo correlación.</b> . . . . .	108
<b>B.11. Matriz de correlación con mapa de calor.</b> . . . . .	109

# Capítulo 1

## Introducción

Debido a los avances tecnológicos y la facilidad que brindan para generar una mayor conectividad alrededor del mundo, el uso de Internet sólo ha ido en incremento con el paso de los años [1]. Internet resulta ser útil debido a que brinda accesibilidad a servicios como: comunicación, información, ocio, estudios, trabajos, compras, negocios, economía y un largo etcétera [2]. En 2021, un 59.5 % de la población mundial (4660M personas de un total de 7830M) cuenta con Internet [3]. Dentro de los servicios previamente nombrados, la gran mayoría pueden llevarse a cabo dentro de redes sociales online tales como Facebook, Twitter, LinkedIn, Orkut, Spotify, y así una larga lista de aplicaciones que contienen millones de usuarios. Existe una teoría de 6 grados de separación [4] que en resumen menciona que cada persona está a seis personas de conocer a cualquier persona en el mundo. Las redes sociales Online han ocasionado que la globalización tecnológica y la creciente conectividad entre los usuarios de las redes disminuya la teoría a sólo 4 grados de separación [5].

Una red social puede ser vista como un grafo donde los nodos son los usuarios o actores del sistema y las aristas son las conexiones que hay entre ellos [6]. En una misma red social se pueden identificar distintos tipos de relaciones. Por ejemplo, en muchas redes sociales Online, los actores podrían conectarse mediante relaciones de seguimiento (“follow”), de mensajerías (“post/reply”), de réplicas de contenido (“retweet”), de agrado o preferencia (“like/favorite”), entre otras.

Debido al creciente volumen y diversidad de datos y metadatos generados en las redes sociales, se hace necesario el estudio y desarrollo de diversos algoritmos de análisis para poder comprender mejor la estructura y evolución de estas redes, ya sea para fines de investigación, sociales o comerciales.

Una pregunta natural que emerge de este contexto es conocer aquellos nodos o actores más relevantes de una red y puede ser respondida a partir de diversas herramientas proveídas por el análisis de redes sociales y la teoría de grafos como las medidas de centralidad [7]: que se enfocan en los usuarios más influyentes dentro de una red. Así se

definen medidas de centralidad clásicas, como el *grado*, que juzga a un nodo por la cantidad de conexiones que posee; la *cercanía*, que apunta a que los mejores nodos son los que tienen los caminos más cortos para llegar a los demás, o la *intermediación*, que apunta a que los mejores nodos son los que se utilizan como puente entre conexiones con el resto de nodos[8]. Cada una de estas medidas representa un criterio de relevancia distinto, y permite por tanto jerarquizar a los nodos de la red de manera diferente. Como estos, existen más de un centenar de otras medidas de centralidad [9], algunas basadas en la topología de la red, pero las hay también basadas en otros aspectos de la misma.

También existen estudios para calcular la dispersión de influencia (o propagación de información) de un conjunto de nodos en la red, se denominan modelos de dispersión de influencia [10]. Dos de estos modelos, los más utilizados y estudiados. Son el *Linear Threshold Model* (LT-Model) o Modelo de Umbral Lineal, y el *Independent Cascade Model* (IC-Model) o Modelo de Cascada Independiente [10]. El primero está inspirado en el comportamiento colectivo [11] y el segundo en el marketing viral [12].

En 2019, se presentó una nueva medida de centralidad basada en el modelo LT-model, llamada *Linear Threshold Rank* (LTR) [13]. Esta medida considera a los nodos más relevantes como aquellos que junto a sus vecindades consiguen dispersar mejor su información o influencia al resto de la red [10]. Esta medida según los estudios realizados en [13] ofrece buenos resultados, siendo comparada con otras medidas de centralidad como Katz, Page-Rank e Independent Cascade Rank (ICR). Por otra parte, existen medidas de centralidad que se utilizan como heurísticas para poder encontrar top- $k$  actores más influyentes de una red, los cuales luego pueden usarse como semillas o conjuntos de activación sobre los cuales aplicar un modelo de dispersión de influencia. Un ejemplo de este tipo de medidas es el *difusión degree* [14], definido en 2011 y que generaliza la medida tradicional de *centralidad de grado*, considerando vecindades de hasta dos niveles de profundidad con una cierta probabilidad de difusión. En este trabajo, se busca generalizar la medida LTR, utilizando algunas ideas del *diffusion degree*, para poder comprender la función que cumplen las vecindades de los actores en los procesos de dispersión de influencia, a través de redes sociales y qué tanto influye el grado de la vecindad.

Esta medida de centralidad generalizada que denominamos General Influence Spread Rank (GISR), se ha desarrollado como una plataforma de experimentación. Esta plataforma permite moldear casi cualquier red para ser apta en el trabajo de esta medida, además admite distintos tipos de ajustes de parámetros para medir la centralidad de los actores que se incluyen como entrada al momento de ejecutar esta plataforma y finalmente es capaz de realizar cálculos estadísticos y de correlación para ayudar con el análisis de los resultados obtenidos.

El resto del documento está estructurado como se describe a continuación:

- El Capítulo 2 contiene el marco conceptual y estado del arte, donde se abarca más en detalle el contexto en base a lo que existe actualmente en otros estudios relacionados.

- El Capítulo 3 contiene la definición del problema, solución propuesta, objetivos de este trabajo y la metodología.
- El Capítulo 4 es el diseño de la solución del problema. Se presenta la nueva medida generalizada, se habla sobre la plataforma de experimentación y las técnicas de análisis.
- El Capítulo 5 es la implementación realizada del sistema. Se habla sobre los recursos para construir la plataforma de experimentación, ya sea en Software y Hardware y también sobre las primeras pruebas en la plataforma.
- El Capítulo 6 trata sobre la experimentación realizada, se hace una revisión de los datasets utilizados, los arreglos necesarios, los resultados, análisis de éstos y luego las discusiones.
- El Capítulo 7 se encuentra la implantación de la plataforma de experimentación. En este capítulo específico se habla sobre lo necesario para poder ejecutar plataforma y su manual de usuario.
- El Capítulo 8 se encuentran las conclusiones finales del trabajo.

# Capítulo 2

## Marco conceptual y estado del arte

En este capítulo nos centraremos en explicar los dos conceptos más importantes de este trabajo, los cuales son los modelos de dispersión de influencia y medidas de centralidad, además de mencionar algunos ejemplos y presentar algunos trabajos de estudio realizados.

### 2.1. Marco conceptual

Un *grafo* es la representación de una red de la forma  $G = (V, E)$ , donde  $V$  es el conjunto de nodos o vértices y  $E$  es el conjunto de aristas. Usaremos además las notaciones  $n = |V|$  y  $m = |E|$  para referirnos al número de nodos y aristas del grafo, respectivamente.

Un concepto importante que usaremos en este trabajo es el de *vecindad*. Este se refiere a las conexiones directas de un nodo. Para calcular la vecindad del nodo  $v$  se utiliza la siguiente ecuación:

$$N(v) = \{u \in V \mid (u, v) \in E \text{ o bien } (v, u) \in E\} \quad (2.1)$$

La *vecindad a nivel  $l$*  considera la distancia que hay entre los nodos. Lo ideal es encontrar el camino con la menor distancia posible para llegar de un nodo a otro. Esto está representado por la siguiente ecuación:

$$N(v, l) = \{u \in V \mid \text{dist}(v, u) \leq l\} \quad (2.2)$$

donde  $l$  es el nivel de profundidad de la vecindad, y  $\text{dist}(v, u)$  es la longitud del semicamino (es decir, el camino sin importar la dirección de las aristas) más corto que hay entre los nodos  $v$  y  $u$ .

Una versión menos usada, pero útil para este trabajo es la *vecindad probabilística*:

$$N(v, \lambda_v) = \{u \in N(v) \mid p(v, u) \geq 1 - \lambda_v\} \quad (2.3)$$

donde  $\lambda_v$  es la probabilidad del nodo  $v$  de poder incluir a sus vecinos, y  $p(v, u)$  el valor aleatorio obtenido para ver si se consigue o no superar la probabilidad necesaria. Por ejemplo, si  $\lambda_v = 0.8$ , entonces  $N(v, 0.8)$  incluirá a todos los vecinos de  $v$  para los que ejecutando una “tirada aleatoria” (*random*) en el dominio  $[0, 1]$  se obtenga al menos un valor 0.2. Note que  $N(v, 1) = N(v)$ . Todas las probabilidades asociadas a los distintos nodos de la red se pueden almacenar en un vector de probabilidades  $\lambda$ .

Asimismo, podemos también generalizar las definiciones anteriores para definir la *vecindad probabilística a nivel  $l$* :

$$N(v, l, \lambda_v) = \{u \in V \mid \text{dist}(v, u) \leq l \wedge p(v, u) \geq 1 - \lambda_v\} \quad (2.4)$$

donde  $\lambda_{(v,l)}$  es un vector de probabilidades del nodo  $v$  de poder incluir a sus vecinos que se encuentran a un nivel de profundidad  $l$ . Si la probabilidad es la misma para todos los niveles, entonces se puede escribir simplemente  $N(v, l, \lambda_v)$ , donde  $\lambda_v$  es un único valor, como en la Ecuación 2.3. Note que además, si este único valor es  $\lambda_v = 1$ , entonces se obtiene  $N(v, l, 1) = N(v, l)$ .

Finalmente, note que hasta ahora el concepto de vecindad utilizado no distingue entre la dirección de las aristas. Sin embargo, también podríamos estar interesados en obtener solo la vecindad de los nodos a los que apunta el nodo (criterio *forward*, hacia adelante), o bien sólo los que son apuntados por el nodo (criterio *backward*, hacia atrás). Por lo tanto, se define para ambos, casos, respectivamente:

$$N_{>}(v) = \{u \in V \mid (v, u) \in E\} \quad (2.5)$$

$$N_{<}(v) = \{u \in V \mid (u, v) \in E\} \quad (2.6)$$

Esta direccionalidad de la vecindad también se puede aplicar para las vecindades probabilísticas y/o a nivel  $l$ . Para este trabajo, nos interesa introducir estas posibilidades como parámetros, por lo que usaremos la notación  $N(v, l, \lambda_v, \text{dir})$ , donde  $\text{dir} \in \{\text{null}, <, >\}$ . Por defecto,  $\text{dir} = \text{null}$ , lo que significa que se usa la versión de la vecindad adireccional de la Ecuación 2.1. Así, por ejemplo, note que  $N(v, l, \lambda_{(v,l)}, >) = N_{>}(v, l, \lambda_{(v,l)})$  y  $N(v, l, \lambda_{(v,l)}, \text{null}) = N(v, l, \lambda_{(v,l)})$ .

### 2.1.1. Medidas de centralidad

Las medidas de centralidad en análisis de redes sociales se refieren a formas de medir e identificar cuantitativamente a los actores más relevantes dentro de una red. Esto usualmente se realiza a través de fórmulas matemáticas, aunque también existen variaciones un poco más elaboradas que utilizan algoritmos más sofisticados y que requieren de un estudio más específico como minería de datos, inteligencia artificial y otras áreas de las ciencias de datos.

La identificación de los actores con mayor centralidad dentro de una red es útil en muchos ámbitos tales como el marketing viral, análisis de redes sociales, decision-making, biología, investigación, seguridad, tráfico, transporte, educación, entre otros [15]. Hace ya muchos años que este estudio se popularizó debido a su versatilidad por ende, se vienen estudiando y proponiendo una gran cantidad de medidas de centralidad. Las medidas de centralidad más clásicas (definidas en los años 50's) son el *grado* o *degree*, bajo la cual el actor más relevante es el que tiene el mayor número de vecinos; la *intermediación* o *betweenness*, donde el actor más relevante es el más utilizado como puente para conectar un nodo o grupo de ellos con otros, y la *cercanía* o *closeness*, que busca al actor que posea el camino más corto para poder conectar con el resto de los actores de la red [7]. Además de éstas, existe un centenar de otras medidas de centralidad [9]. Cada una de las medidas son planteadas con distintos objetivos, como pueden ser medir la popularidad, actividad o influencia de un actor, predecir actores relevantes en el tiempo, entre otros. En este trabajo nos enfocaremos específicamente en las medidas para cuantificar la influencia de los nodos en la red. De este tipo, también existen muchas medidas aunque no basadas en modelos de dispersión, como *PageRank*, que es la medida que solía utilizar Google en su motor de búsqueda [16], basada en la medida *eigenvector* [17], y de la cual se derivan una gran cantidad de variaciones [18].

### 2.1.2. Modelos de dispersión de influencia

La dispersión de influencia dentro de una red social, es el proceso que se refiere a la dispersión de información a lo largo de la red, utilizando un conjunto de nodos iniciales. Los modelos de dispersión de influencia en el análisis de redes sociales se refieren a algoritmos que, a partir de una activación de nodos iniciales, siguiendo ciertas reglas y/o restricciones puedan dispersar información a través de la activación de otros nodos a lo largo de toda la red [19].

Este fenómeno se ha llevado a muchas áreas de aplicación, debido a que es un concepto bastante versátil. Puede adaptarse a estudios de redes sociales, seguimiento en las expansiones de enfermedades y/o pandemias en redes de contagio, crecimiento económico, propagación de CO<sub>2</sub> y un largo etcétera [20, 21, 22].

Uno de los problemas más estudiados y relacionados a los modelos de dispersión de influencia es el *problema de la máxima dispersión de influencia* (en inglés, *Maximum Influence Spread*) que consiste en encontrar el conjunto de nodos iniciales o semillas que permitan maximizar la dispersión de influencia dentro de la red. Dos de los modelos de dispersión de influencia más conocidos son el *Independent Cascade model* y el *Linear Threshold model*. Esto hace que ambos modelos tengan una gran cantidad de desarrollo y estudios basados y/o relacionados con ellos. En [10] se aplican ambos modelos para estudiar el problema del impacto que provoca la maximización. A continuación expondremos más detalles sobre estos dos modelos que son de suma importancia para este trabajo. Sólo

para aclarar que existen además otros modelos de dispersión, hablaremos también del modelo Susceptible-Infected (SI) y algunas de sus variantes, que son muy importantes para comprender y explicar las expansiones pandémicas [23, 24].

### Linear Threshold Model

El *Linear Threshold Model* (LT-Model) [10], trabaja como se declara a continuación:

Consideremos una red social representada como una tupla  $(G, w, f)$ , donde  $G = (V, E)$  es un grafo, con el conjunto de nodos  $V$  representando a los actores de la red y el conjunto de aristas  $E$  la influencia de las relaciones entre los actores. Entonces  $(a, b) \in E$  dice que el actor  $a$  ejerce una influencia directa sobre  $b$ . La función de peso  $w : E \rightarrow \mathbb{N}$  determina el poder de influencia  $w(a, b)$  que va desde  $a$  hacia  $b$  y la función de etiquetado  $f : V \rightarrow \mathbb{N}$  cuantifica la resistencia de cada actor a ser influenciado.

Sea  $X \subseteq V$  un conjunto de nodos semillas. La dispersión de influencia de  $X$  está definida por  $F(X) \subseteq V$  y formada por los nodos activados en un proceso iterativo. Generalizando, se define  $F_k(X)$  como el conjunto de nodos activados en el tiempo  $k$ . Inicialmente en el paso 0 sólo están activados los nodos de la semilla, por lo tanto  $F_0(X) = X$ . El conjunto de nodos en el tiempo  $i > 0$  serán todos los nodos  $F_{i-1}(X)$ , más los nodos cuya resistencia sea superada por la suma de los pesos de influencia de los nodos vecinos activos que apuntan a él, es decir:

$$F_i(X) = F_{i-1}(X) \cup \left\{ v \in V \mid \sum_{u \in F_{i-1}(X) \mid (u,v) \in E} w(u, v) \geq f(v) \right\} \quad (2.7)$$

El proceso de dispersar influencia termina cuando no hay posibles activaciones. El conjunto final está definido por  $F(X)$ .

En la Figura 2.1 se ve una aplicación del LT-Model. En 2.1a la semilla o activación inicial es el conjunto  $F_0(X) = \{a, c\}$ . Luego 2.1b,  $a$  influye en  $b$  dado que la fuerza de influencia de la arista  $(a, b)$  es 3 y es mayor que la resistencia de  $b$  que es 2, pero  $a$  no puede influir a  $d$  ya que la resistencia en este nodo es mayor que la fuerza de la arista  $(a, d)$ ; al mismo tiempo,  $c$  influye en  $e$  y el nuevo conjunto de nodos activos  $F_1(X) = \{a, b, c, e\}$ . Luego en la tercera y última instancia, 2.1c, debido a la suma de fuerzas de las aristas  $(a, d)$  y  $(e, d)$ , el nodo  $d$  es influido. El conjunto final de nodos activos es  $F_2(X) = F(X) = \{a, b, c, d, e\}$ .

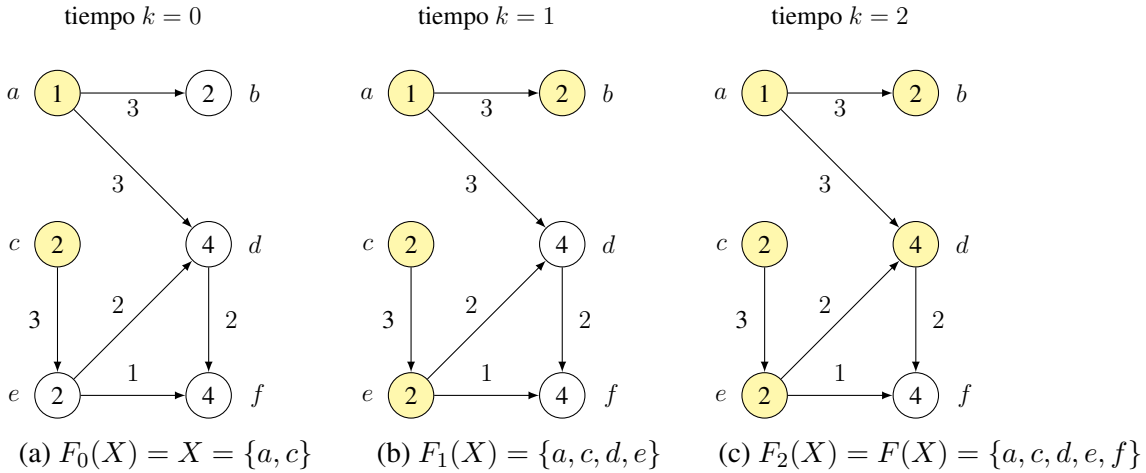


Figura 2.1: Linear Threshold Model (elaboración propia). Nodos amarillos representan nodos activos.

Este modelo de dispersión fue ideado originalmente para el estudio de comportamientos sociales colectivos [11]. Asimismo, también está ampliamente estudiado en casos de redes sociales buscando solucionar el problema de maximización de la dispersión de influencia [25, 26]. Además, al igual que en el caso del IC-Model, pueden encontrarse casos de estudios donde utilizando este modelo, se busca minimizar la emisión de gases contaminantes [27], predicción de probabilidad de dispersar información [28], entre otras aplicaciones de este modelo.

### Independent Cascade Model

El *Independent Cascade Model* (IC-Model) [10], trabaja de la siguiente manera.

Consideremos una red representada como un par ordenado  $(G, w)$ , donde  $G = (V, E)$  es un grafo con un conjunto de nodos  $V$ , los cuales representan los actores en la red, y un conjunto de aristas  $E$ , que representan las relaciones entre estos actores. Entonces  $(a, b) \in E$  hace referencia a una relación de influencia que ejerce el nodo  $a$  sobre el nodo  $b$ . Luego, la función de probabilidad  $w : E \rightarrow [0, 1]$  determina la probabilidad de que un nodo ejerza su influencia sobre otro a través de la arista que los conecta.

Dada una semilla o activación inicial  $X \subseteq V$ , la dispersión de  $X$  estará definida por  $F(X) \subseteq V$ , formado por los nodos activados a través de un proceso iterativo. De manera general, se tiene  $F_k(X)$  como el conjunto de nodos activados en el tiempo  $k$ . Inicialmente, en el paso  $F_0(X)$  sólo están activos los nodos semilla o iniciales, por lo tanto  $F_0(X) = X$ . El conjunto de nodos en el tiempo  $i > 0$  serán todos los nodos  $F_{i-1}(X)$ , más los nodos que se adhieran en ese mismo paso representados de la siguiente forma:

$$F_i(X) = F_{i-1}(X) \cup \{v \in V \mid P(u, v) \geq w(u, v); u \in F_{i-1}(X)\} \quad (2.8)$$

donde  $P(v, u)$  es la probabilidad única de influenciar  $v$  a partir de  $u$  perteneciente a IC-Model, no confundir con  $p(v, u)$  de las funciones de vecindad. Si en una iteración el nodo  $u$  que se encuentra activo, no influncia al nodo  $v$ , entonces  $u$ , no tendrá más opciones de influenciar a  $v$ . El proceso de dispersar influencia termina cuando no hay más activaciones posibles. El conjunto final de nodos activados o dispersión final está definido por  $F(X)$ .

En la Figura 2.2 se puede apreciar una aplicación del IC-Model. En 2.2a, la semilla o activación inicial es el conjunto  $F_0(X) = \{a, c\}$ . Luego en 2.2b,  $a$  influye en  $d$  dado que la probabilidad única de influencia, en la ejecución de este ejemplo, fue superior a la probabilidad en la arista  $(a, d)$  que es 0.6, pero, a su vez no fue mayor que la probabilidad de la arista  $(a, b)$ . Al mismo tiempo  $c$  influye en  $e$  y el nuevo conjunto de nodos activos es  $F_1(X) = \{a, c, d, e\}$ . Luego, en la tercera y última instancia, 2.2c, la probabilidad única de  $e$  consigue influenciar sobre  $f$ . El conjunto final de nodos activos es  $F_2(X) = F(X) = \{a, c, d, e, f\}$ .

Cabe mencionar que el IC-Model fue inicialmente pensado para aplicaciones en marketing viral [29, 30, 31]. Además, recientemente se ha utilizado en varias otras aplicaciones, incluyendo la dispersión de gases de CO<sub>2</sub> en redes de transporte y urbanismo [32, 33], o la dispersión de emociones donde se analizan mensajes de un nodo en específico y cómo éstos pueden influenciar en los nodos vecinos con las emociones que expresarán en los mensajes siguientes [34].

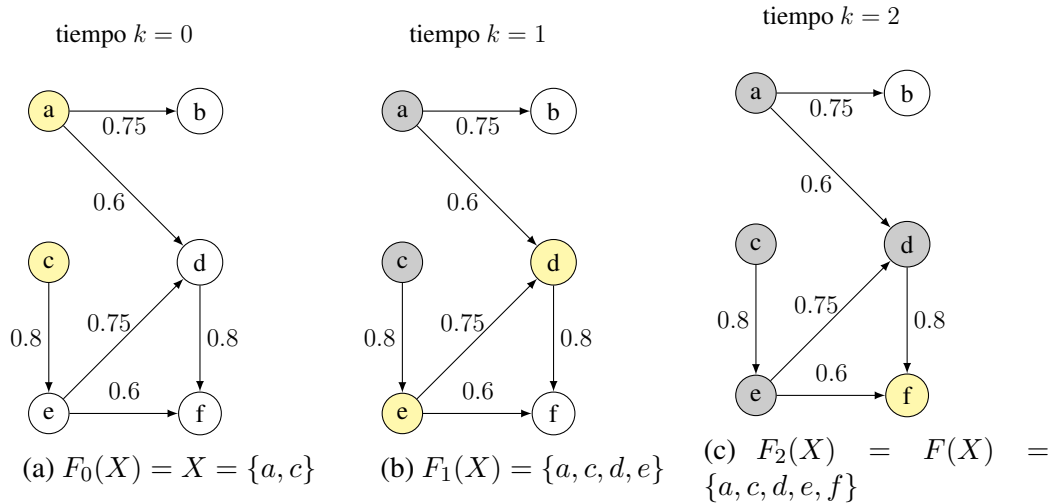


Figura 2.2: Independent Cascade Model (elaboración propia). Nodos amarillos representan nodos activos que pueden influenciar. Nodos grises presentan nodos activos que no pueden volver a influenciar.

### Susceptible Infective Model (SI-Model) y sus variaciones

Como ejemplo de otros modelos de dispersión existentes, la familia de los modelos de dispersión SI (por sus siglas en inglés, “Susceptibles” e “Infectados”) se basan en el comportamiento epidemiológico, utilizando actores que, representados por nodos en una red, van dispersando la epidemia en un proceso iterativo. Estos modelos están basados en ecuaciones diferenciales y tienen el carácter de ser continuos a diferencia de los anteriormente mencionados que son de tipo discreto. A continuación se mencionarán los modelos de dispersión más conocidos de este grupo.

- El *Susceptible Infective Model* (SI-Model) es uno de los más puros en lo que respecta el área de estudio de la propagación de una epidemia y sólo trabaja con estados S (actores susceptibles a ser infectados) e I (actores de la red ya infectados) [23]. Existen diversas aplicaciones y modificaciones a este modelo de dispersión [35, 36, 37].
- El *Susceptible Infective Recovered Model* (SIR-Model), utiliza el mismo concepto de S e I pero a diferencia del SI-Model, aplica un nuevo estado a los actores de la red, R, recuperados del virus. Este estado provoca inmunidad a los actores de su red, en específico cuando los actores ya han sido contagiados, pero luego del tiempo se han recuperado, haciendo que éstos no queden habilitados para volver a ser infectados

nuevamente [24]. Este modelo como el anterior también cuenta con aplicaciones y modificaciones [38, 39, 40].

En la práctica, existen muchas más variaciones referentes a este modelo, como: SIS-Model, MSIR-Model, SEIR-Model, SEIS-Model, entre otros [41, 42, 43, 44]. Si bien todos los modelos de este grupo están basados en la misma idea del comportamiento epidemiológico, a su vez, son distintas variaciones y se aplican en distintos contextos.

## 2.2. Estado del arte

En esta sección del trabajo se abordará en específico los estudios que hay sobre las medidas de centralidad basadas en modelos de dispersión de influencia que sirven para apoyar el estudio que se realizó en el trabajo.

### 2.2.1. Medidas de centralidad basadas en modelos de dispersión de influencia

Las medidas de centralidad y los modelos de dispersión de influencia son dos conceptos muy estudiados y analizados en el área de investigación sobre redes sociales [9], [19]. Además, estos dos conceptos están relacionados de una manera muy directa dado que un conjunto de nodos iniciales que mediante un modelo de dispersión permita influenciar a muchos nodos podría interpretarse como un conjunto de nodos centrales [13]. A pesar de esto último, estas dos técnicas usualmente han sido estudiadas de manera independiente. Sin embargo, recientemente se propuso una nueva medida de centralidad basada en modelos de dispersión de influencia [13]. Al ser una propuesta reciente, sus investigaciones y aplicaciones aún no han sido exploradas en profundidad. A continuación detallaremos esta medida, conocida como Linear Threshold Rank, y luego hablaremos del Diffusion Degree, otra medida que se usa como heurística para encontrar los rankings de nodos influyentes sobre los cuales después aplicar modelos de dispersión. Estas medidas y sus variaciones servirán como inspiración para la generalización de la medida de LTR que diseñamos más adelante en el Capítulo 4.

#### Linear Threshold Rank

Esta medida de centralidad, definida en [13] está basada en el LT-Model. Para trabajar con esta medida se debe fijar en primer lugar un nivel de profundidad. Este nivel de profundidad hace referencia a los nodos que estarán activos inicialmente. Luego de ello, la dispersión a lo largo de la red sigue las reglas del LT-Model definidas anteriormente.

La medida *Linear Threshold Rank* (LTR) está definida por la ecuación a continuación.

$$LTR(i) = |F(\{i\} \cup N(i))| \quad (2.9)$$

En [45] se define esta medida de forma generalizada, buscando cómo es que afecta el nivel de profundidad de la vecindad al cálculo de la centralidad. Esta nueva versión de la medida LTR está definida por la siguiente ecuación.

$$LTR(i, l) = |F(\{i\} \cup N(i, l))| \quad (2.10)$$

En el Algoritmo 1 se presenta la forma de trabajar del Linear Threshold Rank (en su versión generalizada).

---

**Algorithm 1:** Algoritmo de centralidad LTR
 

---

**Condition:**  $(G = (V, E), w, f)$  grafo de influencia adecuado para el modelo de dispersión.

**Input:** nodo  $a \in V$ ;  $l \in \mathbb{N}$  nivel de profundidad

**Output:** valor de centralidad del nodo  $a$ .

```

1 begin
2   | LTR ( $a, l$ ):
3   |   |  $X \leftarrow \{a\} \cup N(a, l)$ 
4   |   | Return  $|F(X)|$ 
5 end

```

---

En la Figura 2.3 se presenta un ejemplo de aplicación de la medida LTR donde se calculará la centralidad del nodo  $a$  en la red. Inicialmente en 2.3a se desconoce cuál es la centralidad  $LTR(a)$ , pero se cuenta con el parámetro de profundidad 1, con este valor, en 2.3b se calcula la vecindad de  $a$  que es  $N(a) = \{b, c\}$  los que formarán el conjunto de nodos iniciales  $\{a, b, c\}$ . El proceso de medición continúa con las reglas del LT-Model. Finalmente en 2.3c,  $b$  puede influenciar sobre  $d$  y  $e$  pero no sobre  $f$ , ya que la resistencia que este último posee es mayor a la fuerza de influencia de la arista  $(b, f)$ . Al mismo tiempo,  $c$  puede influenciar a  $g$  pero no a  $h$ . Por lo que la centralidad del nodo  $a$  en este ejercicio es  $|LTR(a)| = |\{a, b, c, d, e, g\}| = 6$ .

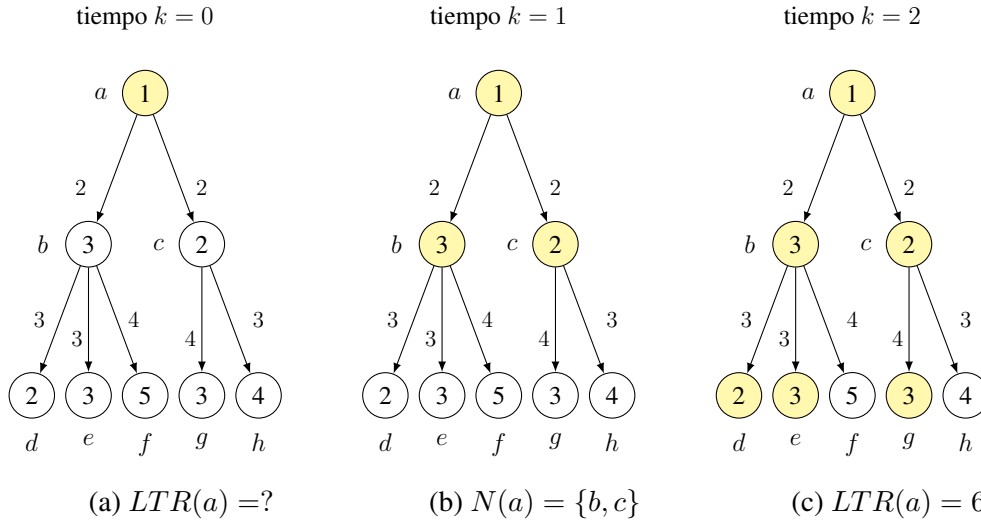


Figura 2.3: Linear Threshold Rank con 1 nivel de profundidad (elaboración propia). Nodos amarillos representan nodos activos.

Esta medida ha sido utilizada para realizar diversas investigaciones como en redes aleatorias [46] donde se descubre un comportamiento similar tanto en redes dirigidas como no dirigidas. En [47] se plantean nuevas variantes de la medida de centralidad LTR, donde se especifica la dirección de las aristas en las vecindades de los nodos:

- Forward Linear Threshold Rank. Sólo considera los nodos hacia los que apunta el nodo al cuál se le calcula la centralidad, como en la Ecuación 2.5.
- Backward Linear Threshold Rank. En la forma inversa a como lo hace la variación anterior, sólo considera los nodos que estén apuntando hacia el nodo que se le calcula la centralidad, como en la Ecuación 2.6.

En [48] se estudian ponderaciones y mecanismos de influencia de la variante Forward Linear Threshold Rank, se descubre que consiguen mejores resultados en LT-Model utilizando PageRank como medida de centralidad para conseguir los nodos iniciales.

### Diffusion Degree

Esta medida de centralidad, definida en [14] se utiliza como heurística para aplicar el IC-Model sobre semillas específicas de actores influyentes, y funciona como se describe a continuación. Primero es necesario calcular el grado del nodo en medición. El *grado* (en inglés, *degree*) del nodo  $v$  es la cardinalidad de su vecindad:

$$C_D(v) = |N(v)| \quad (2.11)$$

Para seguir con el proceso de difusión, se contará con la capacidad de dispersión del nodo  $v$  sobre sus vecinos directos, lo que se calcula multiplicando el grado del nodo por su probabilidad  $\lambda_v$  de activar a cada uno de sus nodos vecinos. Esto estará representado por la ecuación:

$$C'_{DD}(v) = \lambda_v * C_D(v) \quad (2.12)$$

El proceso de difusión continúa al siguiente nivel, donde el grado de los vecinos directos del nodo  $v$  se multiplican a su vez por sus propias probabilidades de activación:

$$C''_{DD}(v) = \sum_{i \in N(v)} C'_{DD}(i) \quad (2.13)$$

Por último, el cálculo del *diffusion degree* para el nodo  $v$  será la acumulación de las Ecuaciones 2.12 y 2.13. Por lo tanto, será dado por la ecuación:

$$\begin{aligned} C_{DD}(v) &= C'_{DD}(v) + C''_{DD}(v) \\ &= \lambda_v * C_D(v) + \sum_{i \in N(v)} C'_{DD}(i) \\ &= \lambda_v * C_D(v) + \sum_{i \in N(v)} \lambda_i * C_D(i) \end{aligned} \quad (2.14)$$

El Diffusion Degree ( $C_{DD}$ ) como medida de centralidad se puede aplicar sobre todos los nodos del grafo para obtener un ranking de los actores más relevantes de acuerdo con este criterio. Los nodos mejor ubicados en el ranking, son candidatos para ser los actores más influyentes de la red. En [14], aplican el IC-Model sobre los primeros nodos del ranking para conocer su capacidad de dispersión de influencia real dentro de la red.

En la Figura 2.4 se tiene un ejemplo de aplicación de la medida *diffusion degree* donde se calcula la centralidad del nodo  $a$  dentro de la red. En primera instancia se calcula el grado de vecindad del nodo  $a$  para calcular la centralidad y luego a sus vecinos en primer grado. En este ejemplo tendríamos  $C_D(a) = 2$ ,  $C_D(b) = 4$ ,  $C_D(c) = 3$ . Luego, se calcula la probabilidad de influencia respecto al valor  $\lambda$  de cada nodo,  $C'_{DD}(a) = 1.5$ ,  $C'_{DD}(b) = 2.4$ ,  $C'_{DD}(c) = 1.5$ , y finalmente se suman para conseguir el Diffusion Degree del nodo  $a$ ,  $C_{DD}(a) = 5.4$ .

Luego, otro grupo de investigadores en [49] generaliza esta medida de centralidad, donde en vez de buscar hasta dos niveles de profundidad, busca hasta el  $l$ -ésimo nivel. Esta generalización se puede representar mediante la siguiente ecuación:

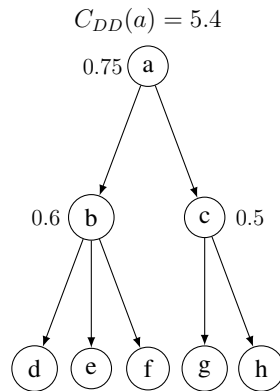


Figura 2.4: Diffusion Degree (elaboración propia).

$$C_{DD}^l(v) = \lambda_v * C_D(v) + \sum_{u \in N(v,l)} \lambda_u * C_D(u) \quad (2.15)$$

Esta medida en particular, ha sido utilizada para realizar estudios de maximización de influencia en redes sociales en múltiples capas. Para esto último, en [50] se utilizan cuentas de diferentes redes sociales online que estén asociadas al mismo correo electrónico.

# Capítulo 3

## Definición del problema

En este Capítulo se abarca cuál fue el problema encontrado para generar el desarrollo de este trabajo, además de la solución realizada, sus objetivos y la metodología de trabajo.

### 3.1. Formulación del problema

En la actualidad, dos de los problemas más relevantes en el análisis de redes sociales son el de centralidad, que tiene que ver con buscar a los actores más relevantes dentro de la red, y el de dispersión de influencia, preocupado de modelar, analizar y comprender los procesos de viralización en redes sociales [7], [10]. Para estos dos problemas existen una gran cantidad de técnicas y modelos de análisis. En efecto, son tantos que inclusive podemos encontrar cientos de distintos tipos de medidas de centralidad [9] y decenas de modelos de dispersión de influencia con variaciones [19]. Debido a que existen tantos modelos y variaciones aplicables a contextos o ámbitos diferentes, el desafío consiste en identificar qué modelo o medida es más útil en cada caso.

Aunque los problemas recién expuestos (centralidad y dispersión de influencia) estén bastante relacionados, dado que una activación de nodos que permita dispersar la influencia a lo largo de toda la red puede considerarse en una activación de nodos centrales o de alta centralidad, durante años fueron estudiados como conceptos independientes. Sin embargo, hace unos años nace la idea de estudiar estos problemas de manera conjunta, mediante medidas de centralidad basados en modelos de dispersión de influencia [13, 14].

Basándose en lo anteriormente dicho, el problema de este Trabajo de Título es profundizar en los estudios de las medidas de centralidad basados en los modelos de dispersión de influencia. Más concretamente, analizar medidas como el *Linear Threshold Rank* (LTR) [13] y sus posibles variantes, para identificar qué características deben cumplir los actores y sus vecindades en una red para maximizar sus procesos de dispersión de influencia.

## 3.2. Solución propuesta

Para resolver el problema, se creó un algoritmo que ejecutó la medida de centralidad con múltiples parámetros ajustables para la experimentación en redes sociales de medidas de centralidad basadas en el fenómeno de dispersión de influencia. El algoritmo recibe como entrada redes sociales representadas como grafos, los parámetros ajustables y como salida entrega los resultados obtenidos de esa ejecución en específico, los últimos son la cantidad de dispersión alcanzada por esta medida.

Se permite utilizar distintas redes sociales de diversos contextos ahorrándonos así el hecho de tener que buscar la medida que se ajuste mejor a nuestra red. Aunque todas las redes ingresadas al algoritmo deben ser ajustadas previamente a su uso. Los parámetros permiten aplicar diversos modelos de dispersión, probabilidades de vecindad, múltiples niveles de profundidad y direccionalidad ajustable (Forward, Backward o ambas).

En este trabajo se analiza esta nueva medida en distintas redes y con distintos ajustes de parámetros, utilizando métodos descriptivos estadísticos y matrices de correlación para visualizar las diferencias y características de los distintos ajustes.

Este Trabajo de Título es de investigación aplicada, por lo que aborda una problemática de investigación en ciencias de la computación y análisis de redes sociales, que cuenta con diversos dominios de aplicación, tales como la propagación de información [51], el marketing viral [52], estrategias de búsqueda [53], sistemas de recomendación [54], sistemas de comunidades [55], gestión de relaciones sociales con el cliente [56], teoría de percolación [57] o educación y procesos de aprendizaje [58], por mencionar solo algunos.

## 3.3. Objetivos

En esta sección se abarcan los objetivos de esta nueva medida, qué es lo que se quiere obtener de ella.

### 3.3.1. Objetivo general

Implementar una medida de centralidad general basada en el fenómeno de dispersión de influencia, que mediante parámetros ajustables permita estudiar el impacto de las vecindades de los nodos en su capacidad de influencia en las redes sociales.

### 3.3.2. Objetivos específicos

- Estudiar medidas de centralidad basadas en vecindades y modelos de dispersión de influencia, realizando una búsqueda bibliográfica de trabajos relacionados.

- Implementar medidas de centralidad basadas en niveles de vecindades y modelos de dispersión de influencia, utilizando herramientas de programación adecuadas para el análisis de redes sociales.
- Aplicar y comparar estas medidas de centralidad en redes sociales reales y de laboratorio, utilizando técnicas de análisis de redes sociales y métodos estadísticos.

## 3.4. Metodología

Dado que este trabajo de título es de investigación aplicada y como solución para acompañar la investigación se desarrolló una plataforma de experimentación, este trabajo cuenta con un componente de metodología tanto de investigación como de desarrollo.

### 3.4.1. Metodología de investigación

Como metodología de investigación se utilizaron los siguientes enfoques:

- Cuantitativo: Debido a que las medidas de centralidad y modelos de dispersión de influencia pueden ser medidos numéricamente esta metodología es ideal para este tipo de comparaciones.
- Correlacional: Se utilizó este enfoque de investigación dado que el objetivo de este trabajo es comparar los resultados obtenidos de las distintas medidas.
- Exploratorio: Debido a que las medidas de centralidad basadas en modelos de dispersión de influencia no cuenta con muchas investigaciones y no existen muchas comparaciones entre ellas. Además, el estudio se basa en buscar la respuesta de las siguientes preguntas de investigación: ¿Qué ventajas obtienen las medidas basadas en modelos de dispersión de influencia sobre las medidas de centralidad independientes del modelo? ¿Cómo afectan los niveles de profundidad de las vecindades de los actores para conformar activaciones iniciales o los procesos de dispersión de influencia?

En la Figura [3.1](#) se presenta un proceso de metodología cuantitativa clásico, dividido en sus distintas etapas [\[59\]](#).

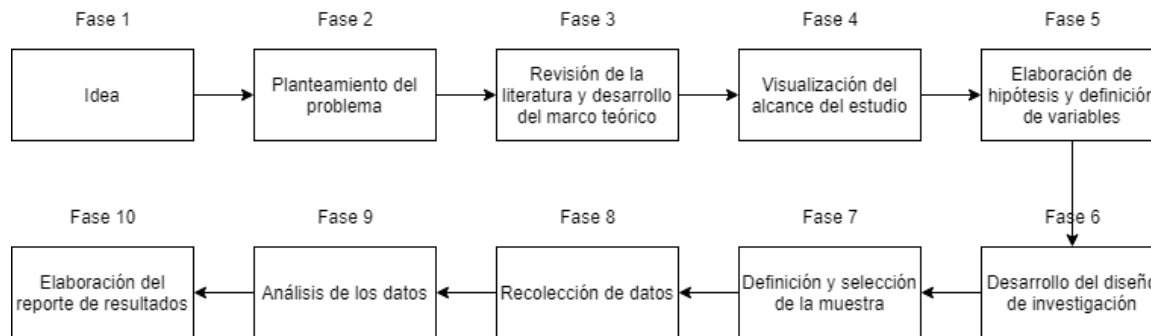


Figura 3.1: Proceso de Metodología Cuantitativa.

### 3.4.2. Metodología de desarrollo

Debido a que el objetivo de esta metodología fue obtener una plataforma de experimentación en tiempos acotados, lo óptimo fue utilizar una metodología de desarrollo ágil. En este caso se aplicó una basada en la metodología de prototipado evolutivo[60]. Consiste en realizar distintos avances con prototipos del programa final y junto al profesor guía que por el contexto del trabajo es al mismo tiempo el cliente, se realizaron respectivas modificaciones en cada entrega de prototipo. El equipo consta del estudiante de TT y el profesor guía. Además, este Trabajo de Título está guiado por la planificación del proyecto Fondecyt Folio 11200113.

En la Figura 3.2 se presenta un diagrama de avances de esta metodología propuesta que funciona de la siguiente manera:

- **Recolección y refinamiento de requisitos.** Fueron el conjunto de las reuniones iniciales con el cliente donde éste planteó sus necesidades para la aplicación. El conjunto de reuniones permitió definir las bases y estructura de la aplicación. Esta etapa se ejecutó en el marco del curso Seminario de Título I.
- **Diseño rápido.** Es la segunda fase y donde se realiza un pequeño bosquejo en base a lo pactado con el cliente en las reuniones. Sirve como ejemplo para construir las diferentes versiones del prototipo. Esta etapa se ejecutó en el marco del curso Seminario de Título I.
- **Construcción del prototipo.** Se crea ya el prototipo basado en el diseño rápido de la etapa anterior, a través de una maduración progresiva mediante entregas sucesivas al cliente. Esta etapa se ejecutó en el marco del curso Seminario de Título II.
- **Evaluación del prototipo por el cliente.** En esta etapa, el prototipo le fue presentado al cliente y éste realizó la respectiva retroalimentación. Es común que en esta etapa

se generen solicitudes de modificaciones, por lo que se recolectan nuevamente requisitos por parte del cliente. Este proceso se llevó a cabo una vez a la semana luego de haber superado la etapa de construcción del prototipo, por ende también se ejecutó en el marco del curso Seminario de Título II.

- Refinamiento del prototipo. Luego de la retroalimentación por parte del cliente, se realizan cambios al prototipo respecto a los nuevos requisitos. Cuando el prototipo cumple con los requisitos por parte del cliente, este pasa a la fase final que es el producto de ingeniería, de lo contrario, vuelve a la etapa de diseño rápido y el ciclo es repetido nuevamente. Esta etapa se ejecutó en el marco del curso Seminario de Título II.
- Producto de ingeniería. El prototipo aprueba todos los requisitos del cliente y se le entrega el producto a este. Esta etapa fue alcanzada al término del curso de Seminario de Título II.

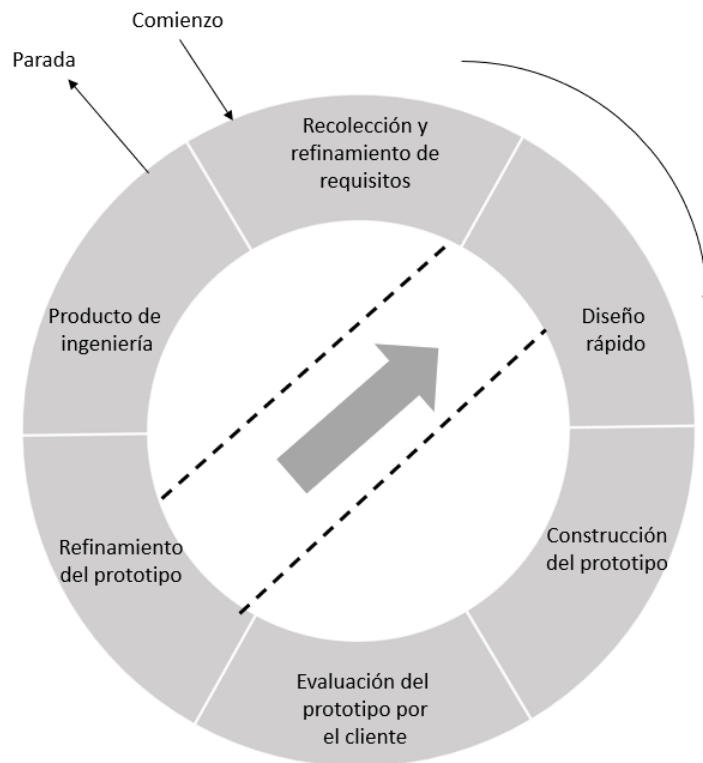


Figura 3.2: Diagrama basado en metodología de prototipado (elaboración propia).

# Capítulo 4

## Diseño de la solución

Este capítulo consiste en la etapa del diseño de la solución, donde se presenta la nueva medida de centralidad basada en modelos de dispersión de influencia generalizada. Esta nueva medida de centralidad dará una solución al problema planteado en el Capítulo 3 y ayudará a responder las preguntas de investigación. También se presenta la plataforma de experimentación donde será llevada a prueba esta nueva medida.

### 4.1. Solución del problema

Como solución al problema planteado del Capítulo 3, se propone una nueva medida generalizada de la medida Linear Threshold Rank y con ideas basadas en las probabilidades de Diffusion Degree. Esta será probada en una plataforma de experimentación, donde el usuario podrá introducir los parámetros requeridos que desee testear en cada ejecución.

#### 4.1.1. General Influence Spread Rank

Esta nueva medida basada utilizará la misma naturaleza de trabajar que Linear Threshold Rank, la medida en la que está basada y que fue previamente definida en el Capítulo 2 en la Sección 2.2.1. A su vez también trabaja con la idea probabilística del Diffusion Degree que está definido en la Sección 5. Debido a esto, dependiendo del contexto de las pruebas se podrán aplicar en ella tanto IC-Model como LT-Model.

**Definición 1** Dado un grafo de influencia  $(G, w, f)$  (apto para el LT-Model o para el IC-Model),  $l \leq n \in \mathbb{N}$ ,  $\lambda_v$  una función de probabilidades,  $dir$  la direccionalidad de las vecindades, el General Influence Spread Rank (GISR) se define como:

$$GISR(i, l, \lambda_v, dir) = |F(\{i\} \cup N(i, l, \lambda_v, dir))| \quad (4.1)$$

donde  $F$  es la dispersión de influencia dada por LT-Model o IC-Model. Note que este valor se puede normalizar sencillamente dividiendo la fórmula por  $n$  (o  $n - 1$  si la red no posee bucles).

La representación algorítmica en pseudocódigo de la medida es la siguiente:

---

**Algorithm 2:** Algoritmo de centralidad General Influence Spread Rank

---

**Condition:**  $G(V, E)$ ,  $w$ ,  $f$  grafo de influencia adecuado para el modelo de dispersión.  $\lambda$  vector de probabilidades.

**Input:** nodo  $a \in V$ ;  $l \in \mathbb{N}$  nivel de profundidad;  $model \in \{0, 1, 2\}$  es el modelo de dispersión (0 ninguno, 1 IC-Model, 2 LT-Model),  $dir \in \{0, 1, 2\}$  es la direccionalidad de la vecindad (0 ninguna, 1 Forward, 2 Backward)

**Output:** centralidad del nodo  $a$ .

```

1 begin
2   if model == 0 then
3     GISR(a, l, dir):
4     | X ← {a} ∪ N(a, l, λ, dir)
5     | return |X|
6   end
7   else if model == 1 then
8     GISR(a, l, p, dir):
9     | X ← {a} ∪ N(a, l, λ, dir)
10    | return |F(X)|
11  end
12  else GISR(a, l, dir):
13    | X ← {a} ∪ N(a, l, λ, dir)
14    | return |F(X)|
15  ;
16  ;
17 end

```

---

- Donde  $N(a, l, \lambda, dir)$  es la vecindad del nodo  $a$  a distancia  $l$  con vector de probabilidades  $\lambda$  y además en la dirección  $dir$  definidas en las Ecuaciones [2.5](#) y [2.6](#).
- $p$  es un vector de probabilidades perteneciente solamente a IC-Model
- $X$  es la vecindad influenciada por el nodo al que se le calcula la centralidad. Si no se utiliza un modelo de dispersión de influencia, se retorna la cardinalidad de este conjunto.
- $F(X)$  es el modelo de dispersión de influencia elegido por parámetros entre LT-Model e IC-Model (véase Sección [2.1.2](#)). Si en una ejecución de la plataforma de

experimentación se escoge alguno de estos dos implementados, se retorna la cardinalidad de este conjunto.

### 4.1.2. Plataforma de experimentación

Este programa estará diseñado para funcionar directamente en una consola de comandos o Shell de diferentes sistemas operativos que cuente con la versión de Python 3.6 o superior. Estos pueden ser Windows, Mac, Linux, entre otros sistemas operativos que también puedan correr código Python. En un comienzo éste solicitará un archivo como entrada que contenga la red social representada como grafo, cumpliendo con las condiciones descritas en la Sección 2.1.2 y los parámetros a utilizar en esa ejecución. Si todo está correcto, se realizará una ejecución como se muestra en el Algoritmo 2. Con este programa, se puede evaluar “cómo afectan los niveles de vecindad a la medida” verificando en diferentes ejecuciones, el comportamiento de la nueva medida con diversos parámetros de profundidad. Se recomienda tener niveles de profundidad entre el rango 1-6 dado que con estos, en teoría, podría cubrirse la red por completo o gran parte de ella [4].

Además, también se pueden calcular para los mismas redes, ejecuciones de medidas de centralidad que no estén basadas en modelos de dispersión de influencia, tales como el grado, la cercanía o la intermediación para poder comparar y con eso responder la otra pregunta de investigación; “¿Qué ventaja tienen las medidas basadas en modelos de dispersión de influencia por sobre las que no?”.

Para hacer uso de la plataforma de experimentación, es necesario utilizar las variables de diversos tipos, como las que se encuentran a continuación:

- **Variables independientes.** Éstas se refieren a la entrada que recibe el programa, tales como un nodo perteneciente al archivo que se le entrega al programa, un nivel de profundidad para la vecindad y el valor que representa el modelo de dispersión a utilizar en esa ejecución, la dirección de influencia o sentido y la probabilidad de influenciar vecinos.
- **Variables parámetros.** Éstas variables son las condiciones para que se ejecute el programa. Se encuentran dentro del archivo que contiene la red representada por un grafo y su respectivo vector  $\lambda$  de probabilidades. No confundir en el caso de el modelo de dispersión de influencia IC-Model, que cuenta con dos vectores de probabilidades. Uno que comparte con LT-Model y es el de las probabilidades de vecindad, y por otra parte el propio del modelo, dado que es un modelo probabilístico, es necesario que la red esté preparada con un vector de probabilidades para realizar la dispersión de su algoritmo.

- **Variables dependientes.** Éstas son el resultado que arroja finalmente el programa después de cada ejecución, debido a que para que éstas existan, las variables anteriormente mencionadas deben estar de manera correcta. Por lo tanto, la principal variable dependiente es el valor de la medida de centralidad para un actor específico dentro de una red. Adicionalmente, el tiempo de ejecución que tarda el algoritmo para cada ajuste de variables y parámetros, aunque este estudio no está enfocado específicamente en el tiempo que tardan las pruebas, aún así, se calcula.

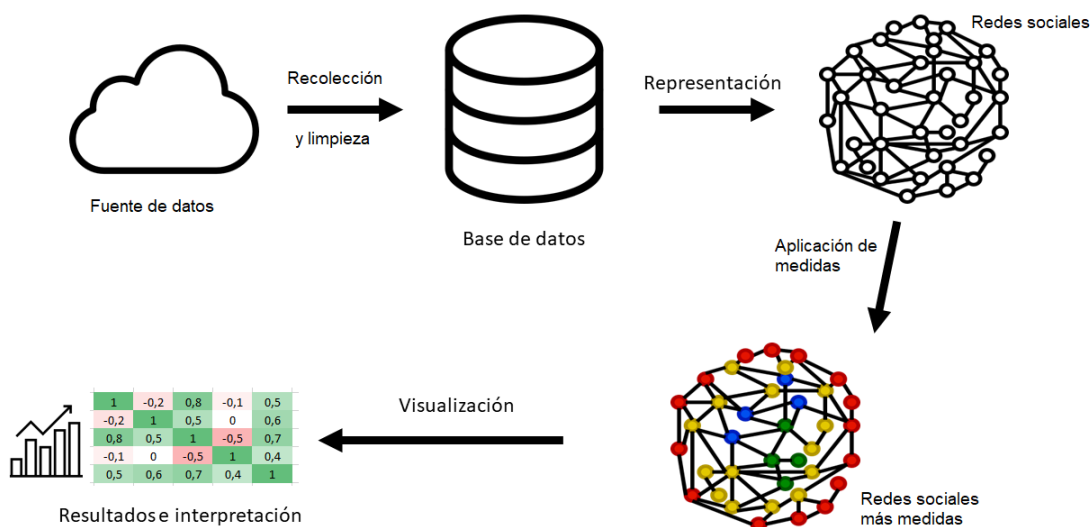


Figura 4.1: Diagrama de etapas

En la Figura 4.1 se ven las etapas del proceso de investigación que será utilizado para el estudio de las redes. Cada una de las fases se describen a continuación:

- **Gran cantidad de datos.** Esta fase o etapa del proceso de investigación se refiere a los datos duros tal como vienen del repositorio de pruebas que se vaya a estudiar como el repositorio de redes de Stanford <sup>1</sup>, repositorio de todo tipo de redes <sup>2</sup>, repositorio de análisis en redes sociales <sup>3</sup>, el repositorio de conjuntos de datos Pajek <sup>4</sup>, entre otros.

<sup>1</sup><https://snap.stanford.edu/data/#socnets>

<sup>2</sup><http://networkrepository.com/rt.php>

<sup>3</sup><https://sites.google.com/a/umn.edu/social-network-analysis/resources/dataset>

<sup>4</sup><http://vlado.fmf.uni-lj.si/pub/networks/data/>

- Base de datos. Luego de la primera fase, ésta está más dedicada a la búsqueda y recolección de datos. Estos son conservados en una base de datos y luego serán llamados desde la computadora donde se llevarán a cabo las pruebas del programa.
- Modelo de redes. Los datos previamente reservados, se presentan en una red sin haber realizado estudio, esto sirve para visualizar la red que generan estos datos tal como vienen. En esta etapa además se realiza una limpieza y adecuación de los datos para que al momento de trabajar con ellos no presenten errores. Por ejemplo, en algunos casos se podría requerir invertir el sentido de las aristas, para poder representar una relación de influencia en lugar de una relación de popularidad o prestigio
- Algoritmos. Los datos son procesados de manera informática utilizando la nueva medida GISR propuesta en el programa previamente mencionado. Esto para obtener los nodos más influyentes de la red.
- Resultados e interpretación. En esta fase el trabajo el procesamiento de los datos ya está listo. Luego estos son presentados en rankings para así visualizar los nodos con mayor influencia. Para comparar la medida bajo diversos ajustes de parámetros, se recurre a técnicas estadísticas como coeficientes de correlación y finalmente se visualizan en una matriz de correlación para ser interpretados.

## 4.2. Diseño de pruebas

### 4.2.1. Pruebas de la plataforma

Para que la plataforma se comporte de la manera que se necesitó en este trabajo, se desarrollan pruebas específicas. Se utilizan dos redes pequeñas de manera tal que se pueda comprobar fácilmente el resultado de las diversas ejecuciones. En esta fase sólo se utiliza el parámetro de profundidad. Los resultados son analizados en gráficos que representan la mayor cantidad de dispersión de influencia alcanzada por los nodos.

### 4.2.2. Pruebas de experimentación

Al terminas las pruebas de la plataforma, se comienza con la experimentación dentro de la plataforma. Nuevamente se realizan pequeñas pruebas ahora para comprobar todos los parámetros utilizados: Profundidad, dirección, probabilidad de vecindad, modelo de dispersión de influencia.

Luego se realizan pruebas a ambos modelos, comprobando el correcto comportamiento de estos.

Finalmente teniendo todas las pruebas necesarias realizadas, se comienza con las ejecuciones de plataforma donde se realizan las diversas variaciones de parámetros desarrollando un total de 234 pruebas utilizando sólo los parámetros luego también se agregan pruebas realizadas con las medidas de centralidad clásicas sumando un total de 255 pruebas. Los resultados de estas pruebas son analizados tanto en gráficos estadísticos como matrices de correlación.

### 4.3. Técnicas de análisis

Para comprobar que las soluciones que entrega el programa sean útiles, se aplicaron métricas de calidad como las siguientes:

- Eficiencia. En primera instancia el tiempo de ejecución de la medida debe ser polinomial. En este caso, esta medida cumple con este criterio debido a que tanto el cálculo de las vecindades como los modelos de dispersión son polinomiales.
- Número de soluciones distintas. Se busca que las soluciones otorgadas en cada una de las ejecuciones sean diversas debido a que si la medida entrega muchos valores idénticos no será útil. Esto se puede llevar a cabo utilizando diversos parámetros para identificar aquellos que provean una mayor diversidad de resultados.
- Desviación estándar. Se busca que los resultados no sólo sean diversos, sino que además sean distinguibles entre sí. Una mayor desviación estándar permite distinguir mejor los resultados, requiriéndose menos dígitos significativos en los valores de las medidas. Al igual que la métrica anterior igual puede verse influenciada positivamente modificando los parámetros de la medida.

Luego de verificar la calidad de los resultados se compararán distintos ajustes de variables independientes y parámetros utilizando las siguientes herramientas:

- Coeficientes de correlación. Aplicar este método de análisis arroja información útil para poder hacer comparaciones entre las medidas y ver cuál se adecua mejor para diferentes ejecuciones. En este trabajo en específico, se utilizarán los coeficientes de Spearman [61], debido a que es un coeficiente no-paramétrico, que no requiere asumir una distribución particular de los datos. Además para asegurar que los valores/resultados son estadísticamente significativos se considerará un  $p$ -value inferior a 0.05.
- Matrices de correlación. Las matrices se utilizarán para visualizar de manera más sencilla la información obtenida desde los coeficientes de correlación y poder encontrar hallazgos.

# Capítulo 5

## Implementación

En este capítulo se considera lo necesario respecto a herramientas de software para ejecutar la plataforma de pruebas que se ofreció previamente como solución al problema. También se detalla en qué tipo de equipo se realizaron los experimentos.

### 5.1. Software utilizado

#### Jupyter

Para comenzar a desarrollar la plataforma de experimentación se utilizó el software *Jupyter* [62] el cual es un ambiente de desarrollo para poder programar *Jupyter Notebooks* que permite codificar varios lenguajes de programación o analizar datos con diversas herramientas, además es un software de código abierto contenido en GitHub.

Los Jupyter Notebooks son documentos web basados en JSON que siguen un esquema versionado y una lista ordenada de celdas que pueden contener: códigos de distintos lenguajes, texto de marcado Markdown, fórmulas matemáticas y ecuaciones, entre otras cosas. Para este trabajo en específico, se utilizarán los Jupyter Notebooks para desarrollar código en lenguaje Python.

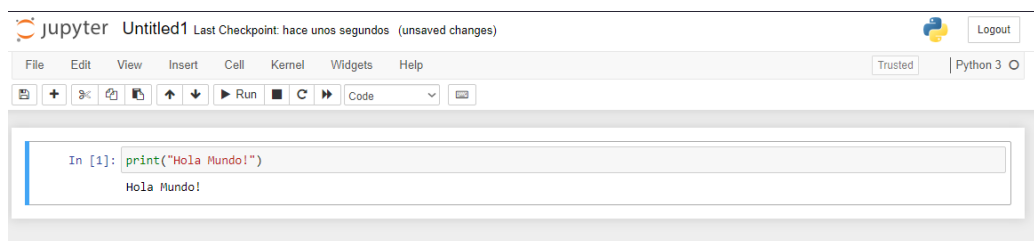


Figura 5.1: Ejemplo de celda en Jupyter Notebook.

En la Figura 5.1 se ve una celda que como entrada recibe código en Python y de salida entrega el resultado de la ejecución de ese código.

Se utiliza este software, debido a la versatilidad que ofrece en su herramienta y que es una manera de programar y probar código de una manera sencilla y rápida. Además tiene una opción que permite exportar el código ya sea en un Jupyter Notebook o en un archivo .py.

## GitHub

La herramienta *GitHub* [63] es un hosting sin fines de lucro de repositorios en la nube, que permite conservar sistemas de control de versiones *GIT*. Gracias a esto se logra compartir el código para no limitar el trabajo solamente a un área local. La creación de *ramas* permite trabajar varias áreas del mismo código al mismo tiempo sin modificar el código principal y evitar el conflicto entre modificaciones permitiendo volver a una versión previa. En la Figura 5.2 se puede ver el proyecto que contiene la plataforma de experimentación.

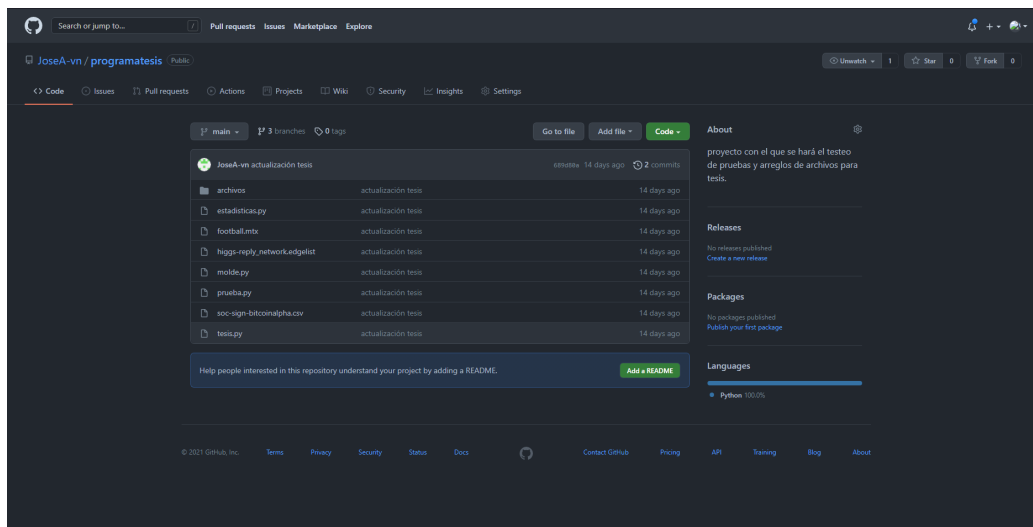


Figura 5.2: Vista principal del proyecto en GitHub.

## Visual Studio Code

Luego de haber avanzado en el desarrollo de la plataforma de experimentación se utiliza el editor de texto *Visual Studio Code* [64]. Que cuenta con varias funcionalidades, como: (i) poder iniciar una consola dentro del mismo programa, por lo tanto, se puede escribir código y probarlo inmediatamente dentro de la misma aplicación, (ii) poder agregar diferentes extensiones que permiten programar de manera más cómoda, (iii) permite manejar *GIT* de manera sencilla y con interfaz gráfica.

En la Figura 5.3 se puede ver el editor de Texto Visual Studio Code, además en esa misma figura se puede apreciar la consola dentro del programa que comentábamos antes.

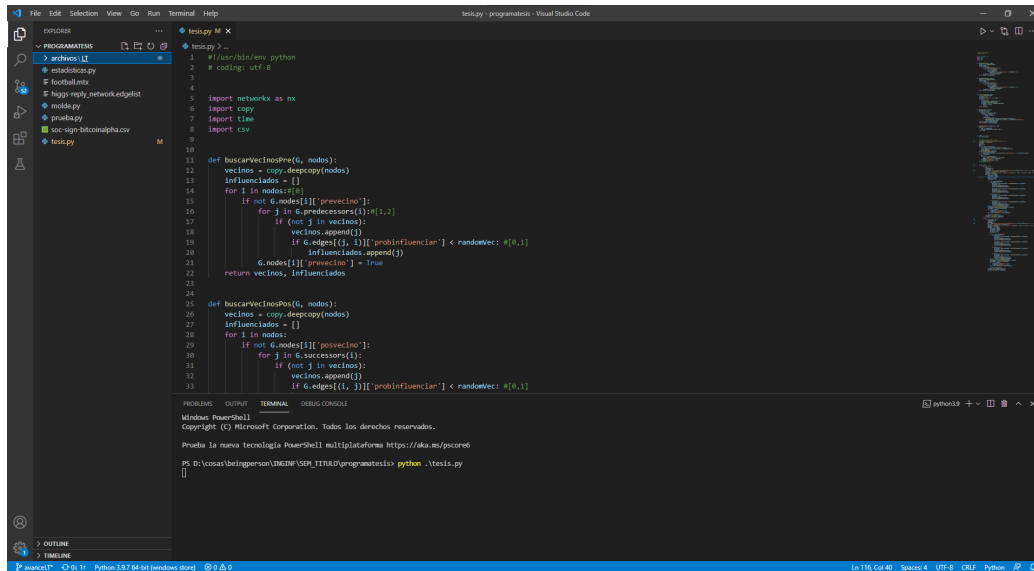


Figura 5.3: Vista de edición en Visual Studio Code.

## Spyder

*Spyder* es otro IDE que permite editar código Python [65]. Este a su vez, como Visual Studio Code, tiene una consola implementada que se utiliza para ver correctamente los gráficos creados por *Matplotlib*. Al formar parte de los programas que trae la distribución de Python, *Anaconda*, incluye la gran mayoría de lo necesario preinstalado y es más sencillo comprobar cosas como los gráficos en *Spyder* que en otro editor de código.

En la Figura 5.4 se ve la aplicación de *Spyder*. A la izquierda está ubicado el editor de texto, y a la derecha inferior, la consola para ejecutar los programas. La sección de la derecha arriba es un visor de diversas funciones, entre ellas se pueden ver el resultado de los gráficos de las aplicaciones.

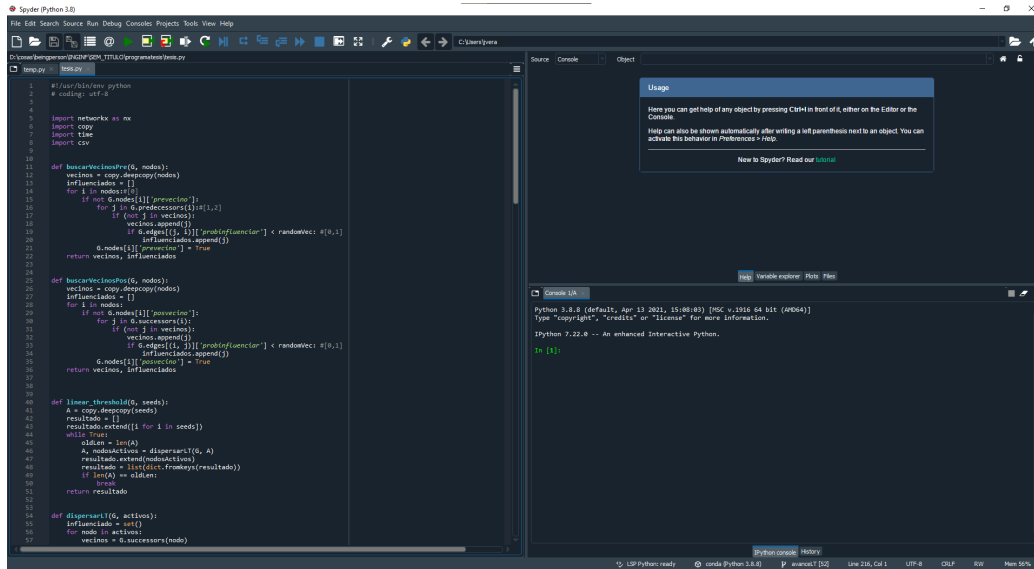


Figura 5.4: Vista de Spyder.

## Trello

Trello [66] es una herramienta online que permite trabajar utilizando la gestión de trabajo Kanban. Es muy útil para apoyar la metodología de desarrollo prototipado que se aplica en este trabajo y permite tener mayor conocimiento sobre las tareas realizadas y las que falta por realizar.

En la Figura 5.5 se puede ver el uso de la aplicación Trello con sus tarjetas tipo Kanban, con tareas ordenadas por su estado de realización.

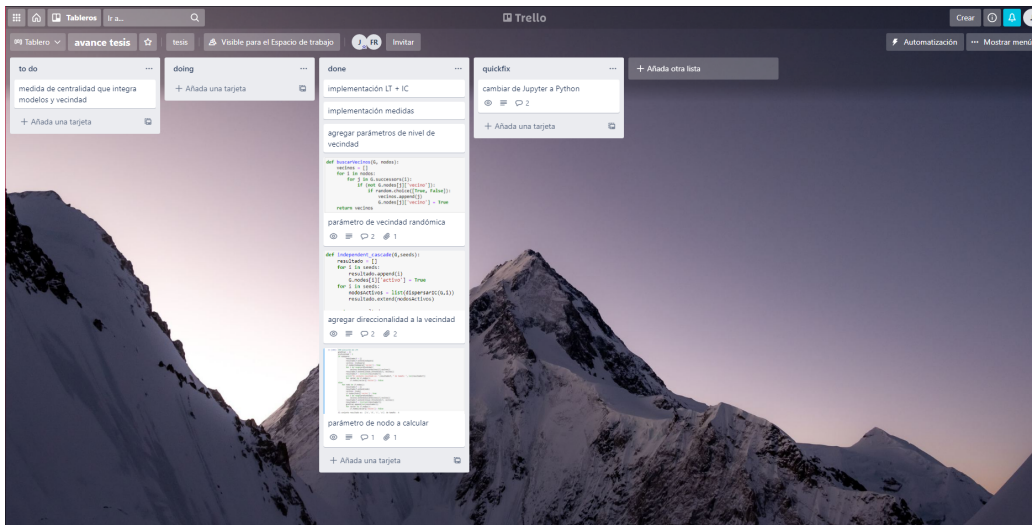


Figura 5.5: Vista de el uso de trello.

## Google Meet

*Google Meet* [67] es una aplicación online que permite realizar reuniones de vídeo llamada. Es muy útil debido que otorga la posibilidad de estar en contacto en tiempo real y tiene la funcionalidad de transmitir por pantalla, lo que también permite mostrar de manera más directa el trabajo que se realiza en el computador de los participantes de la reunión sin la necesidad de tenerla físicamente o realizar una conexión virtual. Esta aplicación fue imprescindible para comunicarse con el cliente o profesor guía en el contexto de trabajo a distancia por la pandemia

## eduVPN

*eduVPN* [68] es un servicio de VPN que permite conectarse a redes institucionales. Teniendo acceso a las redes institucionales de la Universidad de Valparaíso, se puede hacer uso de computadores que estén conectadas a estas redes de manera remota y/o acceder a sitios privados de esa red. Para el caso de este trabajo, sirvió para conectarse con una computadora perteneciente a la Escuela de Ingeniería Informática con mayores recursos que la computadora a la que tenía acceso el estudiante. Esto ayuda entonces a realizar los experimentos en un computador mejor y sin estar exigiendo a los recursos personales del estudiante.

## 5.2. Hardware utilizado

Para la realización de pruebas preliminares del trabajo se utiliza el computador portátil personal del estudiante tesista. Las especificaciones están a continuación.

- Procesador: Intel Core I5-9300H
- Ram: 16 GB
- Almacenamiento: 120 GB M2 PCIe nVME, 1000 GB HDD 5400 rpm

Además, al momento de realizar la fase de experimentación, también se contó con una computadora facilitada por la Escuela de Ingeniería Informática de la Universidad de Valparaíso para realizar pruebas. Sus especificaciones son las siguientes.

- Procesador: Intel Core i7-8700T
- Ram: 8 GB
- Almacenamiento: 120 GB M2 PCIe nVME, 1000 GB HDD 5400 rpm

## 5.3. Lenguajes de programación

Para la realización de la plataforma se utiliza solamente el lenguaje de programación Python 3 y algunas librerías específicas. Las librerías utilizadas son:

- NetworkX: Esta librería de código abierto [69] ofrece una manera sencilla de trabajar con los grafos en Python permitiendo entre otras cosas un manejo de las propiedades de los nodos, sus vecindades, conteo total de nodos, aristas, entre otros usos muy útiles para realizar el trabajo.
- Matplotlib: Esta librería de código abierto [70] permite la creación de gráficos en Python. Estos son utilizados para visualizar los resultados de las distintas pruebas de la plataforma.
- Numpy: Esta librería de código abierto [71] permite realizar operaciones matemáticas más orientadas a estudios científicos. Para este trabajo en específico se utilizan algunos métodos para el análisis de resultados y representación de datos.

A su vez también se utilizaron otras librerías propias del lenguaje como: Copy, que permite realizar copias profundas y superficiales (para este trabajo se utilizaron las copias profundas dado que estas permiten copiar una variable por completo, teniendo el mismo contenido pero en otra asignación de memoria, por ende se podría modificar esta nueva

variable sin modificar la primera asignación de memoria); Time, que permite conocer el tiempo del reloj de la computadora (para este trabajo se utilizó para medir el tiempo que tardaba la computadora en realizar los experimentos); CSV, que permite crear, editar y leer archivos con extensión .csv de una manera sencilla; Random, que entrega valores pseudo-aleatorios (en este trabajo se utilizó su función almacenando las semillas, de modo que los resultados podrían ser replicados para realizar un estudio en profundidad de ese caso en específico).

## 5.4. Estrategia de implementación

Teniendo ya el ambiente y el lenguaje definidos para la solución informática, se planificó el proceso de experimentación, donde en primera instancia fueron instaladas todas las dependencias requeridas y así tener el entorno de desarrollo preparado para comenzar el proceso de desarrollo. Habiendo realizado lo anterior, se preparan los avances debido a la metodología de desarrollo utilizada, la cual está definida en la Sección 3.4.2. Se comenzó por implementar los modelos de dispersión de influencia, siguiendo por la vecindad generalizada y terminando con la medida de centralidad que generaliza las medidas Linear Threshold Rank y las ideas obtenidas desde el Diffusion Degree.

Dado que el objetivo de este trabajo es crear una plataforma que trabaje en shell dentro del funcionamiento normal del sistema, no es necesario aplicar algún patrón estructural que esté asociado a las vistas y/o requerimientos del usuario con el sistema. Por lo tanto, el patrón estructural que mejor acoge a las necesidades de este sistema es el *Composite* o también llamado *Object tree u objeto compuesto*. Esto porque el usuario de la plataforma le entregará ciertos parámetros al sistema y este debe comunicarse desde un componente hacia otro en manera descendente para poder entregar un resultado. Esto se ve más claro en la Figura 5.6.

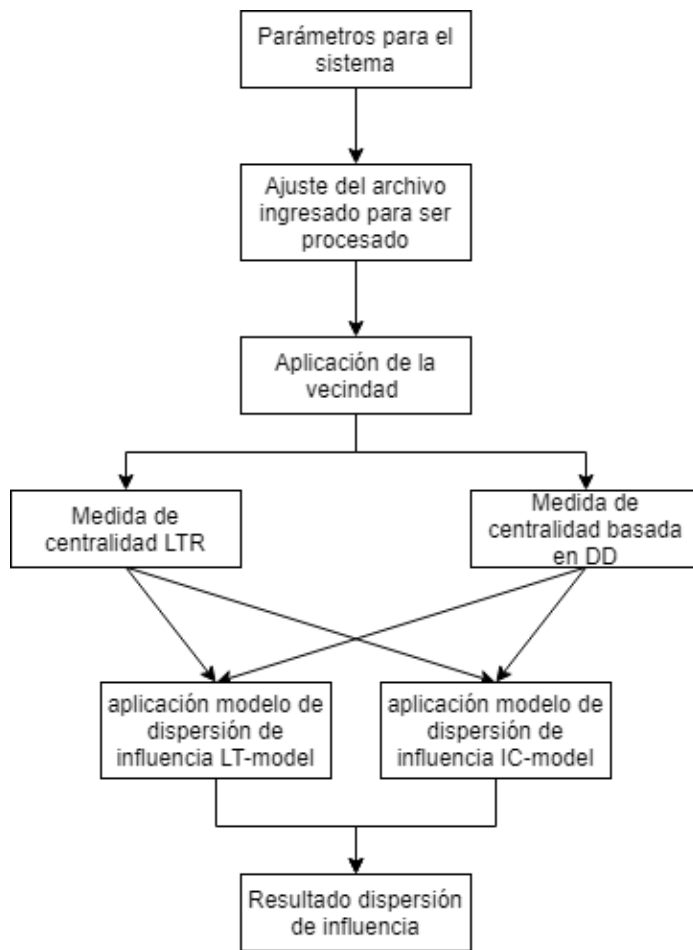


Figura 5.6: Ejemplo del patrón estructural aplicado a la plataforma de pruebas.

## 5.5. Pruebas de la plataforma

En esta sección se encuentran los diseños de las pruebas realizadas en la medida generalizada GISR. Se busca con esto, hallar una solución al problema y también responder a las preguntas de investigación planteadas en el Capítulo 3, además de también encontrar y solucionar cualquier error que pueda presentar la plataforma antes de hacer la experimentación final. Para esto último también, se detallarán los datasets utilizados en las pruebas, cómo se obtuvieron, su formato, y cómo se interpretarán los resultados.

### 5.5.1. Pruebas a la plataforma

Antes de realizar los experimentos de los que se obtienen los resultados reales y se hagan conclusiones en base a ellos, es necesario realizar una serie de pruebas para poder llegar a una versión madura de la plataforma. Las pruebas son enfocadas a una revisión en detalle de cada sección de la plataforma. El objetivo de estas pruebas es que cada una retorne el resultado previsto. A continuación, se entrará en detalle a las pruebas realizadas.

### 5.5.2. Diseño de experimentos

Dado el diseño de la solución planteado en el Capítulo 4, el diseño y la forma de realizar las pruebas será como se detalla a continuación (note que estas fueron pruebas para pulir posibles errores de programación; más tarde en el Capítulo 6 se describen los experimentos, con la aplicación ya testeada).

- Búsqueda o creación de datasets de prueba. En esta fase de la experimentación se buscará un dataset el cual cumpla con lo requerido para una ejecución de la nueva medida. En primera instancia, se cuenta con dos datasets de laboratorio.
- Ajuste en los datasets. Los datasets tal como son creados y/o recolectados de algún repositorio, no siempre cumplen con el formato necesario para ser procesados por la plataforma. Es por esto que se ajustan previamente antes de ser probados, adecuando al formato necesario para poder ser utilizados en pruebas en la plataforma. Dentro de los ajustes realizados se puede contar con agregar parámetros tales como la probabilidad de influenciar a un vecino (si el caso fuera para una dispersión bajo IC-Model) o la etiqueta de los nodos (si el caso fuera para una dispersión bajo LT-Model) y la probabilidad de que los vecinos sean parte de la semilla inicial dependiendo del nivel de profundidad escogido por el usuario (en ambos casos se realiza este ajuste).
- Ajuste en los parámetros. Dado que las medidas son paramétricas, en la sección de experimentación se probarán diversas variaciones de los parámetros y diversas entradas con las que se busca analizar la dispersión de influencia, estas variaciones

entregan diferentes resultados de dispersión de influencia. Algunos de los parámetros a testear son: nivel de profundidad, nodo en medición, modelo de dispersión, sentido de influencia, probabilidad de considerar a vecinos como conjunto de nodos semilla.

- Análisis de resultados. En esta etapa se corroboran los resultados obtenidos en los experimentos para responder a las preguntas de investigación, comparando las diferentes ejecuciones en matrices de correlación.

### 5.5.3. Datasets

Para poner a prueba la plataforma utilizaremos dos datasets. Cada uno corresponde a la representación de una red donde los nodos representarán actores de la red y sus aristas, las interacciones entre ellos que representan relaciones de influencia. Además, deben ajustarse para trabajar con su respectivo modelo de dispersión de influencia, IC-Model o LT-Model. Los datasets son definidos a continuación.

#### Dataset utilizado para LT-Model

Aquí se detallará el dataset utilizado [2](#) para probar la plataforma con el modelo de dispersión de influencia LT-Model previamente definido en el Capítulo [2](#) Sección [2.1.2](#) y que es uno de los dos utilizados para las pruebas dentro de la plataforma de experimentación propuesta por este trabajo.

En la Tabla [5.1](#) está la matriz de adyacencia del dataset utilizado para las pruebas de la plataforma. Debido a que este grafo es dirigido, la matriz es asimétrica. Luego, cada celda representa el peso de la arista que conecta. Por ejemplo, la arista  $(a, c)$  posee un peso o poder de influencia de 8 desde el nodo  $a$  sobre  $c$ . Además dado que es un dataset preparado para LT-Model cuenta con las etiquetas en los nodos, definidas en [5.2](#) donde nuevamente, los nodos son representados por las letras y los valores de las celdas son la etiqueta o resistencia del nodo a ser influenciado.

	a	b	c	d	e	f
a	0	0	8	5	2	0
b	4	0	0	5	3	8
c	2	0	0	5	1	0
d	4	1	3	0	1	0
e	2	0	0	0	0	0
f	0	4	7	5	1	0

Tabla 5.1: Matriz de adyacencia dataset preparado para LT-Model.

a	b	c	d	e	f
8	5	10	15	3	8

Tabla 5.2: Etiquetado dataset preparado para LT-Model.

Este dataset en particular, fue comprobado por fuerza bruta<sup>1</sup> por lo que se conocen todos los comportamientos de antemano. Por esto último, la red se convierte en un caso ideal para realizar las pruebas de este modelo. En la Tabla 5.3 se muestra el formato del archivo MTX<sup>2</sup> que debido a sus cualidades se adecua para trabajar con él en la plataforma. La primera fila contiene las etiquetas de la manera nodo y etiqueta juntos. Luego cada línea siguiente será una arista que contendrá como primer valor, el nodo que apunta hacia el siguiente nodo representado por el siguiente valor y por último contiene un valor que representa el peso de la arista.

a8	b4	c11	d15	e3	f8
a	c	8			
a	d	5			
a	e	2			
b	a	4			
b	d	5			

Tabla 5.3: Archivo de prueba para IC-Model.

### Dataset utilizado para IC-Model

Aquí se detallará el dataset utilizado para probar la plataforma con el modelo de dispersión de influencia IC-Model previamente definido en la Sección 2 y que es uno de los dos utilizados dentro de la plataforma de experimentación propuesta para este trabajo.

Para este dataset, no se cuenta con la matriz de adyacencia como en el caso anterior, dado que se trata de una red obtenida desde un repositorio, en específico Network Repository [72], de 829 nodos y más de 1500 aristas, por lo que la matriz de adyacencia de esta red en específico es mucho más grande que la anterior. Su formato está definido en la Tabla 5.4 donde nuevamente es una red dirigida y en cada fila están definidas las aristas

<sup>1</sup>La técnica de fuerza bruta se refiere a realizar todas las combinaciones posibles obteniendo todos los posibles resultados. En este caso, se probaron las dispersiones de influencia para las  $2^6 = 64$  activaciones iniciales posibles.

<sup>2</sup>MTX, es un formato de matrices que permite ordenar los valores en conjuntos separados por espacios dentro de cada línea, además realizar comentarios, especificar el tamaño del archivo, entre otras cosas.

de la manera que el primer valor representa el nodo que apunta al siguiente valor. En esta ocasión los nodos son enumerados y no representados por letras.

Nodo inicial	Nodo final
1	2
1	3
1	4
2	5
2	6

Tabla 5.4: Archivo de prueba para IC-Model.

Para este dataset se utilizará de momento el IC-Model, por lo que es necesario agregarle otro valor a cada fila, un valor decimal entre 0 y 1 que representará la probabilidad de que el nodo de la primera columna influence al nodo de la segunda columna. Esto está representado en la Tabla 5.5 y funcionaría de la siguiente manera: la arista (1, 2) tiene una probabilidad de 0.3 por lo tanto el nodo 1 tiene la probabilidad 0.3 de influenciar al nodo 2. El valor agregado es de carácter aleatorio por lo tanto en cada ejecución, este valor cambiaría.

Nodo inicial	Nodo final	Prob. influenciar
1	2	0.3
1	3	0.4
1	4	0.7
2	5	1.0
2	6	0.3

Tabla 5.5: Ejemplo ajuste para IC-Model.

#### 5.5.4. Interpretación de resultados

Para esta fase del trabajo, sólo se consideró la red preparada para IC-Model. Dado que para LT-Model debido a fuerza bruta se conocen todos los resultados y además la red es muy reducida para ser visualizada.

Los resultados serán visualizados solamente en gráficos. Estos datos sólo contarán con el valor de la medida de centralidad y éstos se interpretarán respecto al nivel de profundidad que se utilice en cada ejecución. Se harán diversas pruebas con distintos niveles de profundidad para comprobar teorías como la de los 6 grados de separación [4] y las

diferencias con la de 4 grados de separación [5], así apoyando a la pregunta de investigación sobre cómo afecta el nivel de profundidad a la vecindad para conformar activaciones iniciales planteada en el Capítulo 3.

En la Figura 5.7 se presenta la ejecución de la medida generalizada GISR con nivel de profundidad 0, es decir, la dispersión de sólo los nodos sin considerar vecindad como los nodos iniciales. Note que la gráfica representa bastante bien el comportamiento típico del principio de Pareto, que dice que el 80 % de los datos suele provenir del 20 % de los sujetos.

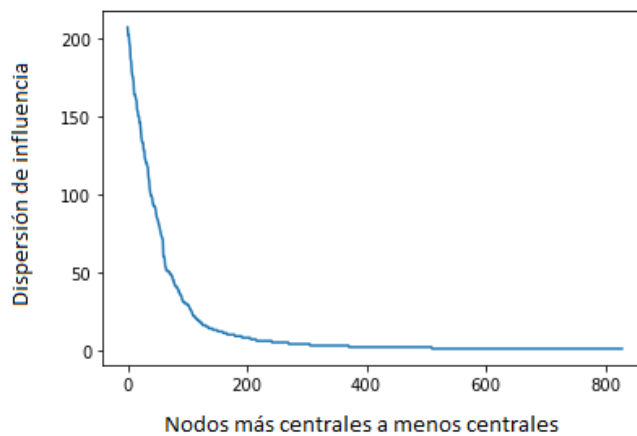


Figura 5.7: Ejecución medida generalizada nivel de profundidad 0

En la Figura 5.8 se presenta la ejecución de GISR con nivel de profundidad 1, es decir, se considera como conjunto de nodos iniciales al nodo sobre el que se calcula la centralidad, más sus vecinos inmediatos. De acuerdo a lo esperado, note que los primeros 400 nodos aproximadamente logran dispersar más su influencia a través de la red, al contar con mayor apoyo por parte de sus vecinos.

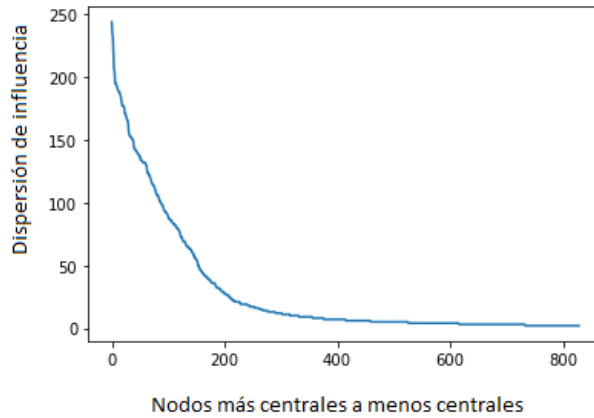


Figura 5.8: Ejecución medida generalizada nivel de profundidad 1

En la Figura 5.9 se presenta la ejecución de GISR con nivel de profundidad 4. Note que en este caso la curva de Pareto ya desaparece. Además, la mayor parte de los nodos de la red junto con sus vecinos hasta un nivel de profundidad 4 ya son capaces de poder dispersar una amplia gama de la red.

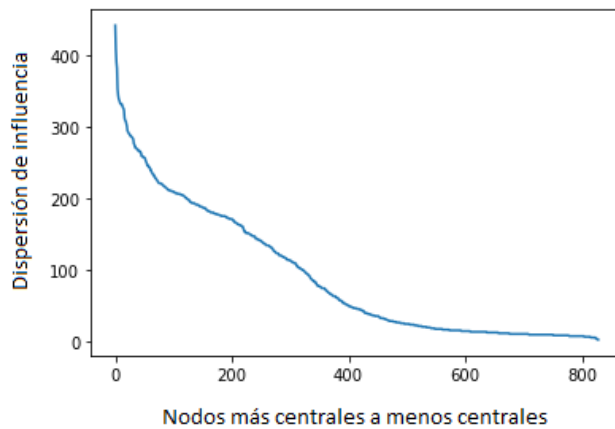


Figura 5.9: Ejecución medida generalizada nivel de profundidad 4

En la Figura 5.10 se presenta la ejecución de GISR con nivel de profundidad 6. En este caso el comportamiento anterior se acentúa un poco más.

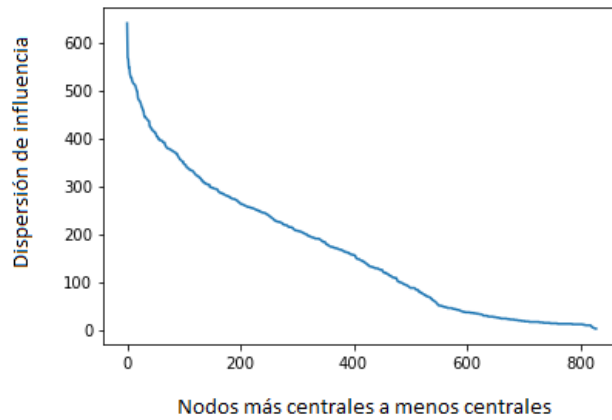


Figura 5.10: Ejecución medida generalizada nivel de profundidad 6

En la Figura 5.11 se presenta la ejecución de GISR con nivel de profundidad 10. En este caso la curva resulta casi lineal, quedando claro que aún con vecindades con un nivel de profundidad 10, para esta red en particular hay actores que no consiguen dispersar todos los nodos de la red.

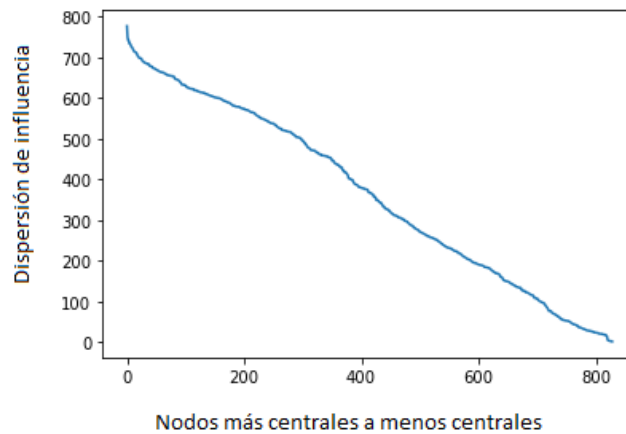


Figura 5.11: Ejecución medida generalizada nivel de profundidad 10

De estas diversas ejecuciones se puede concluir que el nivel de profundidad genera una gran diferencia en la dispersión de influencia dentro de cada ejecución. Para 4 grados de separación, ya se distorsiona el principio de Pareto. Entre el nivel de profundidad 4 y 6 se encontraron el mayor incremento de dispersión.

# Capítulo 6

## Experimentación

En este capítulo se detalla todo lo referente a los experimentos realizados en este trabajo. Desde cómo pasó de estar en fase de pruebas iniciales y su evolución para ser una plataforma que funcione como es esperado. Se muestran los experimentos realizados y los análisis de sus resultados, permitiendo responder a las preguntas de investigación planteadas en la Sección [3.4.1](#).

### Pruebas de vecindad

Son las pruebas que estuvieron enfocadas en la vecindad de cada nodo. Su finalidad es que efectivamente retorne los vecinos del nodo en cuestión y que éstos a su vez efectivamente coincidían con los parámetros ingresados para la búsqueda de vecinos. Los parámetros asociados a la vecindad son tres: (i) Direccionalidad, existen tres sentidos: Salida, que es el sentido al cual apunta el nodo (out-degree). Entrada, donde apuntan hacia el nodo en cuestión (in-degree) y también pueden ser ambos sentidos, es decir, hacia los nodos que se apunta y los que apuntan a él. (ii) El nivel de profundidad que fija el límite de hasta dónde se considerará o no un vecino. (iii) La probabilidad de la vecindad, éste es un valor que puede ingresar el usuario para realizar variaciones en las redes y puede ser 25 %, 50 %, 75 % o 100 %. Además, es necesario de igual manera que se ingrese la red y el nodo perteneciente a la red del que se busca su vecindad. Finalmente, la vecindad es utilizada como la semilla inicial antes de realizar el proceso de dispersión de influencia en la plataforma.

### Pruebas a Modelos de Dispersión de Influencia

Si bien, las pruebas a los modelos de dispersión de influencia fueron realizadas de manera separada dado que los modelos son distintos, se hablará de los dos aquí, ya que las pruebas son idénticas. Las pruebas se enfocan en que los modelos de dispersión implementados en la plataforma de experimentación (IC-Model y LT-Model) funcionen de la misma

forma en la que fueron definidos en un comienzo en el Capítulo 2. Por lo tanto al igual que en el caso de pruebas de vecindad, el objetivo de estas pruebas es que los modelos de dispersión retornen un resultado que coincida con los parámetros entregados a estos. En este caso los parámetros son solamente la vecindad y la red. En redes pequeñas, se puede seguir fácilmente como trabaja el modelo de dispersión, entonces, así se puede comprobar si coinciden o no los resultados. Esto último se realizó sólo en redes con un tamaño reducido de nodos, dado que en una red con gran cantidad de nodos, es muy difícil seguir el comportamiento del modelo.

## 6.1. Experimentos en la plataforma

Luego de haber terminado las pruebas de la plataforma y conseguir la etapa madura sin errores, se realizan los experimentos con los que se busca responder a las preguntas de investigación propuestas en el Capítulo 3. Para realizar los experimentos se debe hacer recolección de datasets, ajustes a las redes para que sean adecuadas con la plataforma para luego pasar a las ejecuciones específicas con diversos parámetros.

### 6.1.1. Datasets

Los datasets escogidos para la experimentación de la plataforma son tres redes de distintos tamaños. Esto, para comprobar el comportamiento de la plataforma dependiendo del tamaño de la red en específico de la cantidad de nodos y aristas que posea. El detalle de las redes se encuentra resumido en la Tabla 6.1. Estas redes tal como fueron recolectadas, no cumplían con lo necesario para ser ejecutadas por la plataforma. Por lo tanto se procede a hacer algunos pequeños ajustes. A continuación se detalla cada red, sus necesarios arreglos y luego cómo se adaptaron para poder ser trabajadas tanto por IC-Model como por LT-Model.

ID	Nombre	n	m	URL
G1	Football	35	118	<a href="https://networkrepository.com/football.php">https://networkrepository.com/football.php</a>
G2	Bitcoin Alpha	3783	24186	<a href="https://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html">https://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html</a>
G3	Higgs Twitter	38918	32523	<a href="https://snap.stanford.edu/data/higgs-twitter.html">https://snap.stanford.edu/data/higgs-twitter.html</a>

Tabla 6.1: Redes sociales utilizadas para los experimentos.

- Football. Esta red es la más pequeña de las tres recolectadas. Como se puede ver en la tabla contiene sólo 35 nodos y 118 aristas. Dado que el número de aristas es ligeramente mayor a la cantidad de nodos, la red no es muy densa ni dispersa. Los efectos de esto se analizarán luego en la sección de resultados. Esta red contiene datos de los clubes de fútbol de Universidades Estadounidenses en clases de verano una temporada del 2000. Cada nodo representa un equipo distinto; las aristas, la

relación entre los equipos. La interpretación de las relaciones  $(a, b)$  es, el equipo  $a$  influyendo sobre el equipo  $b$  con una fuerza de influencia representada por  $w(a, b)$ .

- **Bitcoin Alpha.** Dentro del trabajo, esta red es la de tamaño medio teniendo 3.783 nodos y 24.186 aristas. Dado que el número de aristas es varias veces más grande que la cantidad de nodos (para ser exactos, es 6.3 veces más grande), nos encontramos frente a una red más bien densa. Esto último, en teoría, provoca que la dispersión de influencia de la red pueda ser muy grande dado que existe mucha conectividad en ella. Esto será analizado más adelante en la sección de resultados. La red representa relaciones de confianza de intercambios en Bitcoins realizadas en la plataforma BTC Alpha. En esta clase de plataformas los usuarios son anónimos y no conocen la identidad de la persona con la que están haciendo intercambios. Para prevenir riesgos o intercambios fraudulentos, los usuarios deben evaluar sus transacciones, así creando una reputación de cada usuario en la red. Cada nodo representa a un usuario y las aristas que unen a los usuarios representan la transacción en bitcoin. La red original trabaja de la siguiente manera: una arista del tipo  $(a, b)$  representa la transacción recibida por el usuario  $a$  desde el usuario  $b$ . Por lo tanto la fuerza de influencia de esa arista  $w(a, b)$  representa la evaluación dada por el usuario  $a$  del usuario  $b$  que es representado por un valor entero que va en el rango -10 (desconfianza total) y 10 (completa confianza). Para este trabajo, cambiamos el valor de confianza entre el rango 1 (desconfianza total) y 21 (confianza total). También se invierten las aristas, de modo que ahora la misma arista  $(a, b)$  es  $(b, a)$ ; de esta manera, tiene el sentido de influencia que ejerce  $b$  sobre  $a$  dependiendo de la confianza que le genera  $b$  a  $a$ .
- **Higgs Twitter.** Esta red es la más grande respecto a la cantidad de nodos que se usó para la experimentación, teniendo un total de 38.918, aunque solamente tiene 32.523 aristas. Esto último provoca que la densidad de la red sea muy baja, siendo una red dispersa e incluso disconexa. Esto será analizado nuevamente en la sección de resultados. Esta red representa respuestas de usuarios en Twitter entre el 1 y el 7 de Julio del año 2012 durante y luego del anuncio de un nuevo descubrimiento de una partícula con características consistentes con el Boson de Higgs el 4 de Julio de 2012. Este dataset fue actualizado el 31 de Marzo de 2015. En esta red en específico, los nodos representan los usuarios de la red mientras que una arista del tipo  $(a, b)$  representa que el usuario  $a$  responda al usuario  $b$ ; su fuerza de influencia  $w(a, b)$  representa la cantidad de veces que respondió. Al igual que en el caso de la red Bitcoin Alpha, se invierte el sentido de las aristas por lo tanto ahora  $(b, a)$  representa la influencia que ejerce  $b$  sobre  $a$ .

Por último los archivos que contienen estas redes deben ajustarse para IC-Model y LT-Model, además de incluir un valor de probabilidad para cuando se calcule la vecindad.

- El ajuste a los archivos para ser utilizados con IC-Model, consiste en agregar un parámetro probabilístico creado con la librería *random* de Python, almacenando el parámetro semilla para que estos puedan ser reproducibles. Este parámetro probabilístico creado por la librería es necesario debido a la forma en la que trabaja el modelo, dado que dispersa su influencia de manera probabilística. El valor de probabilidad para cuando se calcule la vecindad también utiliza una semilla.
- El ajuste a los archivos para ser utilizados con LT-Model, consiste en agregar un parámetro que representará la resistencia de cada nodo a ser influenciado por el resto de nodos. Para conseguir este parámetro se utilizará el criterio de mayoría que considera a todos los nodos que apuntan directamente al nodo en cuestión de la siguiente manera. Dado un grafo de influencia  $(G = (V, E), w, f)$ , para cada actor  $i \in V$ , su etiqueta queda como  $f(i) = \lfloor \bar{w}(i)/2 \rfloor + 1$ , donde  $\bar{w}(i) = \sum \{w(j, i) | (j, i) \in E\}$ . Además también se agrega el valor de la probabilidad del cálculo de vecindad, este valor se crea al igual que en el caso anterior de IC-Model, con una semilla para que sea replicable y la librería *random* de Python.

### 6.1.2. Ejecuciones de la plataforma

Debido a la gran cantidad de ajustes de parámetros posibles, estos experimentos se centrarán en la búsqueda de respuestas para las preguntas de investigación.

En la Tabla 6.2 se encuentran los parámetros y sus valores dentro de las ejecuciones. Por cada red se pueden calcular al menos 78 pruebas únicas gracias a estas variaciones de parámetros y ya que en este trabajo se cuenta con tres redes, la experimentación de pruebas cuenta con 234 distintos resultados considerando cada una de las redes como un solo experimento, a pesar de que todas estas variaciones se le aplican a cada nodo perteneciente de la red, sumando las mediciones de las medidas clásicas, se realizaron un total de 255 pruebas.

Niv. Profundidad	Modelo	Direccionalidad	Probabilidad vecinos
0, 1, 4, 6	IC-Model, LT-Model	Salida, Entrada, Ambos	25 %, 50 %, 75 %, 100 %

Tabla 6.2: Valores de los parámetros en los experimentos.

Dado que la idea principal de este trabajo es ver cómo afecta el nivel de profundidad de la vecindad al momento de dispersar la influencia, este es el parámetro de mayor interés. Luego, los valores más importantes en este parámetro serán: 0 donde cada nodo debe dispersar influencia por sí mismo, 1 donde ahora será el nodo y sus vecinos directamente, 4 para comprobar cuál es el nivel de dispersión utilizando los grados de separación definidos en la literatura para las redes sociales online, y 6 en donde se busca confirmar la teoría de los 6 grados de separación.

El parámetro del modelo de dispersión, podrá asumir tres valores: 0, para ningún modelo (es decir, se calcula solamente los nodos alcanzables a partir de la vecindad); 1, para el IC-Model; y 2, para el LT-Model. El parámetro de direccionalidad, podrá tomar tres valores: 0 (ambos), 1 (salida o forward), 2 (entrada o backward). Por último, el parámetro que define la probabilidad de que los vecinos sean influenciados, podrá tomar cuatro valores: 25 %, 50 %, 75 % y 100 %, donde 100 % representa que todos los vecinos dentro del nivel de profundidad serán utilizados como nodo semilla. Cabe destacar que este parámetro de probabilidad de vecinos sólo funciona cuando el nivel de profundidad es mayor o igual a 1. Tener en cuenta que no se utilizará la función de buscar a un nodo específico de la plataforma ya que lo que se busca es conocer distintos casos y por lo tanto se utilizarán todos los nodos de cada red.

## 6.2. Resultados

Cada una de las 255 ejecuciones mencionadas anteriormente crearán un archivo distinto con sus resultados, donde se podrá encontrar los parámetros utilizados para la ejecución, cada nodo con su respectiva vecindad, la cantidad de nodos influenciados (es decir, el valor de la medida de centralidad GISR y el tiempo de ejecución de la prueba.

En las Figuras 6.1 y 6.2 se aprecian archivos reales sólo que acortados. La primera fila muestra los encabezados de las líneas siguientes, donde  $i$  es el nodo,  $|Xi|$  es la cardinalidad de la vecindad del nodo  $i$ ,  $|F(Xi)|$  es la cardinalidad de la dispersión de influencia del nodo  $i$  junto a su vecindad. Luego los parámetros de profundidad que en la Figura 6.1 es 0 y para la Figura 6.2 es 6, la direccionalidad en ambos casos es 0, lo que representa ambas direcciones. La probabilidad de vecindad en la Figura 6.1 no es relevante, debido a la profundidad 0, y para la Figura 6.2 es 100 % representado por 1. Luego, las siguientes tres filas del archivo representan las vecindades y dispersiones de influencia de los nodos 7, 1, 9, y finalmente la última línea es el tiempo de ejecución de la prueba en segundos. Para más detalle de resultados en distintos casos u otras redes, visitar GitHub<sup>1</sup> referente al proyecto.

---

<sup>1</sup><https://github.com/JoseA-vn/programatesis>

	A	B	C	D	E	F	G
1	i	Xi	F(Xi)	profundidad 0	dirección 0	prob vecinos 0	
2		7	1	1			
3		1	1	1			
4		9	1	2			
5	0.0019757747650146484						
6							

Figura 6.1: Ejemplo de archivo de resultados.

	A	B	C	D	E	F	G
1	i	Xi	F(Xi)	profundidad 6	dirección 0	prob vecinos 1	
2		7	9	9			
3		1	7	8			
4		9	12	12			
5	0.013962507247924805						
6							

Figura 6.2: Ejemplo de archivo de resultados.

### 6.3. Análisis de resultados

Luego de haber terminado todas las pruebas, pueden analizarse finalmente los resultados obtenidos. Se comenzará analizando lo mencionado previamente respecto a la densidad de las redes en la sección de descripción de los datasets. Luego se analizarán los resultados enfocándonos en los parámetros seleccionados. Finalmente se analizarán los resultados en matrices de correlación para dar paso a una discusión final respecto a los experimentos realizados.

#### 6.3.1. Análisis de tiempos de ejecución

Respecto a los tiempos de ejecución, se debe mencionar que mientras más densa sea la red, mayor es el tiempo que tardaron las pruebas realizadas. Esto último fue específicamente más notorio en las redes Bitcoin Alpha y Higgs Twitter. Pese a que la red Bitcoin Alpha tiene menos del 10 % de nodos que la red Higgs Twitter, en todos los experimentos la primera demoró una cantidad de tiempo mayor que la segunda, y en algunos

experimentos esta diferencia fue considerablemente mayor. Esto es más visible en pruebas con nivel de profundidad más alto. En el caso del dataset Football, no se puede asegurar algo al respecto dado que la cantidad de nodos hacía que las pruebas tardaran muy poco, siendo el tiempo de estas pruebas cercano al orden de centésimas de segundos. Estos datos pueden ser revisados con mayor detalle en Apéndice [A](#).

La razón de lo anterior se debe al cómputo de los modelos de dispersión incluidos en la medida de centralidad generalizada GISR. Mientras más densa es la red, hay un mayor número de iteraciones en ambos modelos de dispersión considerados, lo que aumenta los tiempos de cómputo. Sin embargo, entre los dos modelos, se aprecian mayores tiempos de ejecución para el LT-model que para el IC-model, dado que el primero debe ir guardando los nodos ya activos, porque estos pueden seguir apoyando el proceso de dispersión en las iteraciones siguientes, mientras que en el IC-model cada nuevo nodo activo sólo tiene una única posibilidad de activar a sus vecinos aún inactivos. Esto significa que LT-model además requiere mayores costos de almacenamiento de memoria volátil. No obstante lo anterior, ambos modelos de dispersión siguen siendo polinomiales, y para niveles de profundidad bajos (menores de 4), los tiempos de ejecución de GISR son análogos al de otras medidas de centralidad tradicionales, como closeness o betweenness.

### 6.3.2. Análisis gráfico de parámetros

Como se mencionó anteriormente dentro de los parámetros utilizados para las pruebas, el más importante para cumplir con el objetivo de este trabajo es el nivel de profundidad. Como un primer análisis, consideraremos la dispersión de influencia promedio a la que pueden llegar los distintos nodos a partir de las distintas profundidades de vecindad. Por esto último, las gráficas de los resultados tendrán como variable independiente en el eje  $X$  el valor de la profundidad y como variable dependiente o resultado en el eje  $Y$  la cardinalidad de la dispersión de influencia promedio. Finalmente, cada gráfico representará una red en particular, utilizando un modelo de dispersión, una direccionalidad, todos los niveles de profundidad y todas las probabilidades de influenciar a la vecindad incluidas como parámetros.

#### Análisis gráficos red Football

En general, mientras mayor sean los parámetros de probabilidad de vecinos y profundidad, podríamos suponer que mayor debiesen ser los resultados de dispersión de influencia. Pero, en los casos de la red Football para el IC-Model, las Figuras [6.3](#), [6.4](#), [6.5](#) demuestran que para esta red en específico, esto no se cumple. Esto es debido a la cantidad de nodos y al modelo de dispersión de influencia. El IC-Model es probabilístico, haciendo que no siempre puedan influenciar a los mismos nodos, por lo tanto, si en una ejecución se influencia un nodo que a su vez influencia luego a más y luego en otra ejecución este

primer nodo no se influencia, la segunda ejecución tendrá menor dispersión que la primera. Este caso hipotético ocurre específicamente en esta red debido a la baja cantidad de nodos. Luego en las otras redes como Bitcoin Alpha y Higgs Twitter, utilizando el mismo modelo, esto no ocurre dado que cuentan con muchos nodos.

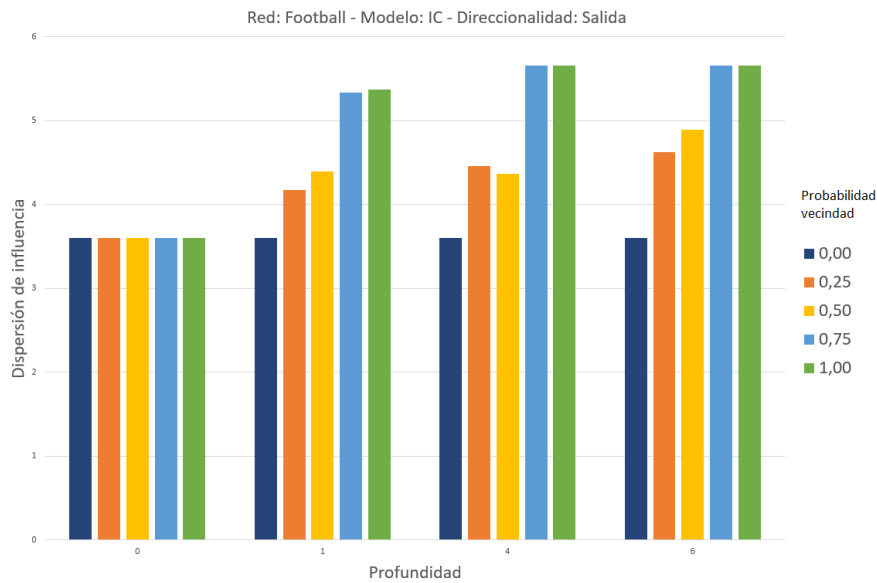


Figura 6.3: Gráfico de resultados red Football direccionalidad salida modelo IC.

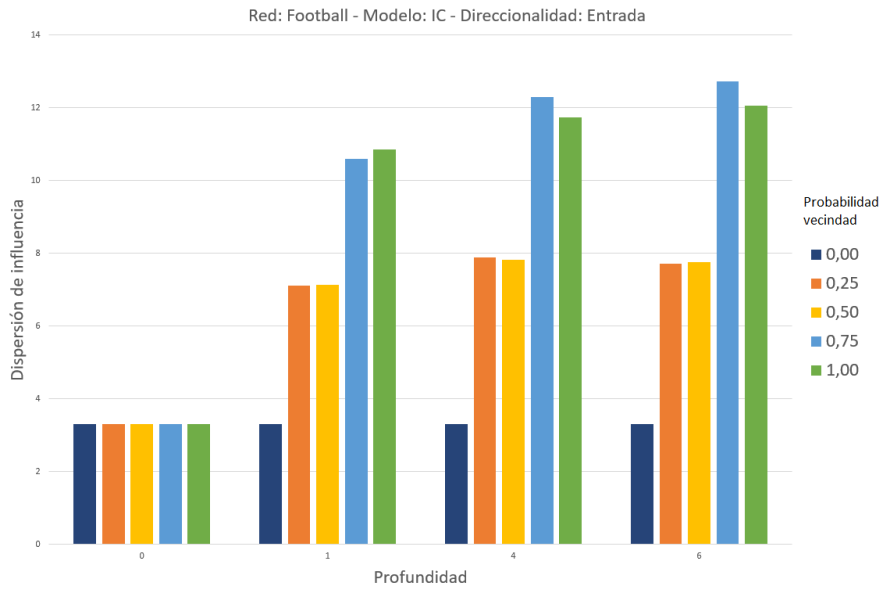


Figura 6.4: Gráfico de resultados red Football direccionalidad Entrada modelo IC.

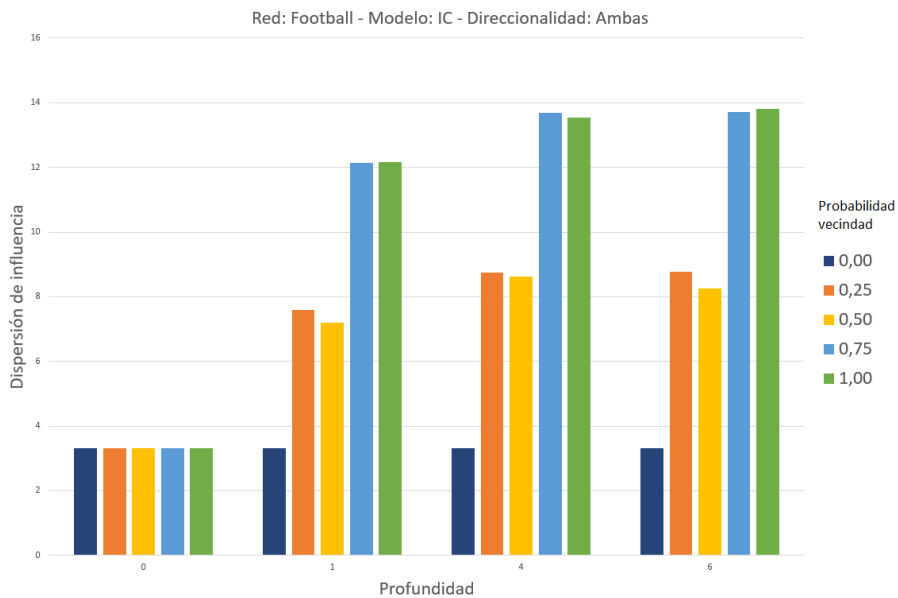


Figura 6.5: Gráfico de resultados red Football direccionalidad ambas modelo IC.

Luego en los gráficos de la red Football pero en el modelo de LT (Figuras 6.6, 6.7 y 6.8), se puede ver un comportamiento más esperado: a medida que suben los niveles de

profundidad, mayor será la dispersión de influencia. Para una profundidad cero, evidentemente, la dispersión es la misma, ya que solo se considera el promedio de las dispersiones para cada nodo por separado.

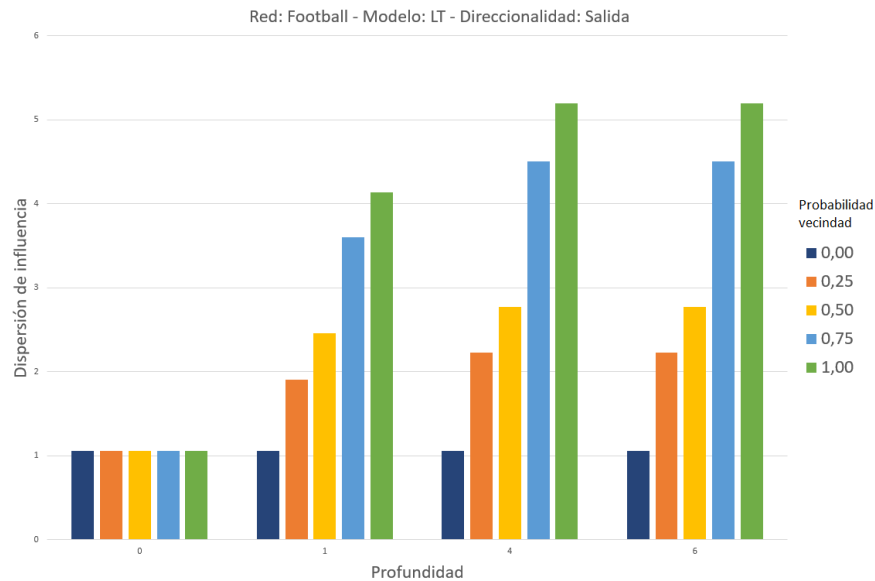


Figura 6.6: Gráfico de resultados red Football direccionalidad salida modelo LT.

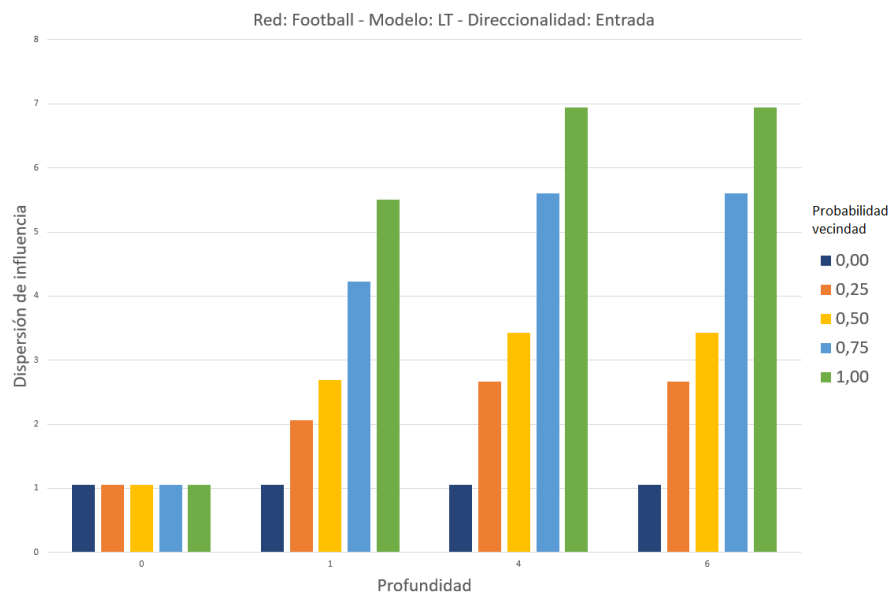


Figura 6.7: Gráfico de resultados red Football direccionalidad Entrada modelo LT.

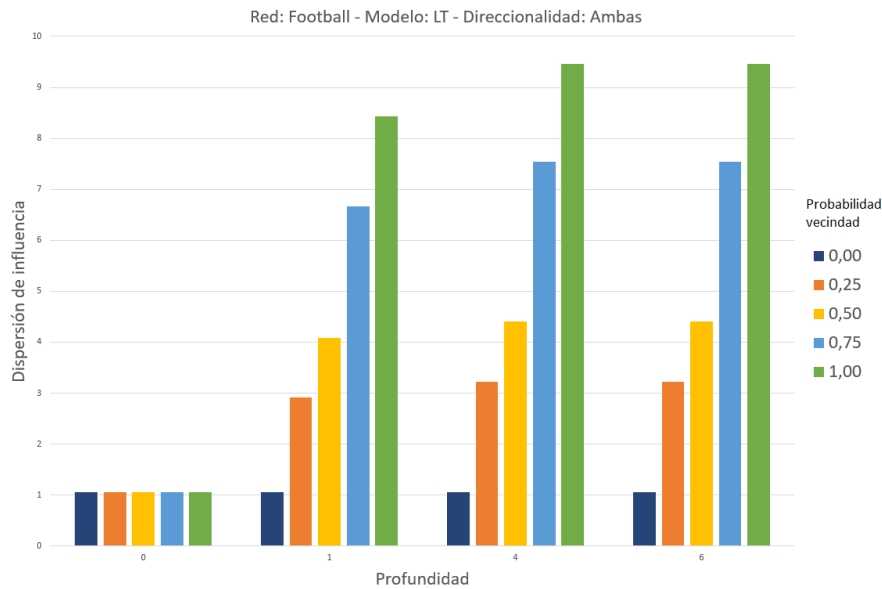


Figura 6.8: Gráfico de resultados red Football direccionalidad ambas modelo LT.

### Análisis gráficos red Bitcoin Alpha

En los gráficos de Bitcoin Alpha, utilizando el modelo de dispersión IC (Figuras [6.9](#), [6.10](#) y [6.11](#)), ocurre algo muy propio de la red y es debido a lo que se analizó previamente respecto a la densidad. Alcanza una dispersión muy grande a pesar de tener parámetros bajos y esto ocurre por el modelo, dado que es probabilístico y la red es muy densa, entonces hay una alta probabilidad de que un nodo influencia a algunos de sus múltiples vecinos, y por consecuencia la dispersión será alta.

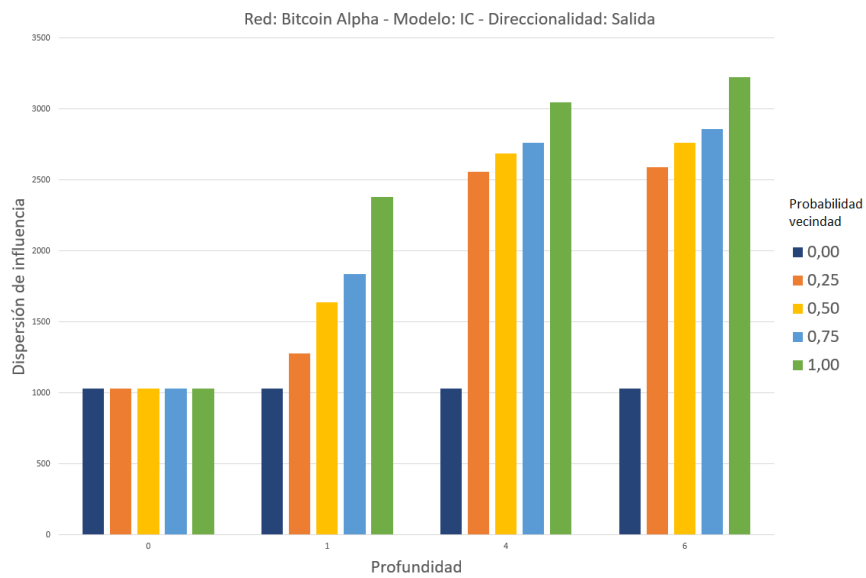


Figura 6.9: Gráfico de resultados red Bitcoin Alpha direccionalidad salida modelo IC.

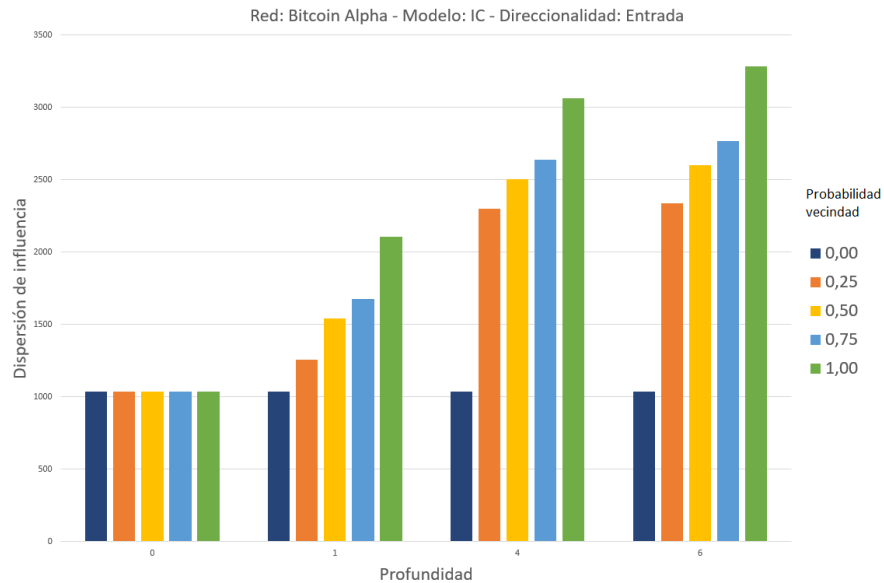


Figura 6.10: Gráfico de resultados red Bitcoin Alpha direccionalidad entrada modelo IC.

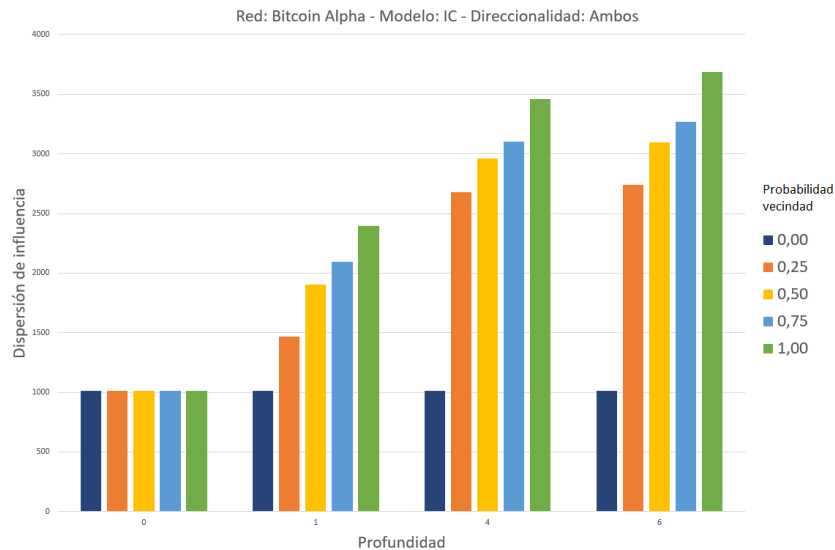


Figura 6.11: Gráfico de resultados red Bitcoin Alpha direccionalidad Ambas modelo IC.

Luego, para la misma red pero utilizando el modelo de dispersión LT (Figuras 6.12, 6.13 y 6.14), se nota algo bastante curioso: a profundidad 1, la dispersión es ligeramente mayor a 0. Esto se debe a que este es un modelo determinista y no probabilístico, por lo que a ese nivel de profundidad, la semilla sigue siendo demasiado pequeña y por tanto no es posible dispersar a más nodos de la red. En cambio, para profundidades 4 y 6 la semilla ya es suficientemente grande como para poder continuar con el proceso de dispersión en la red.

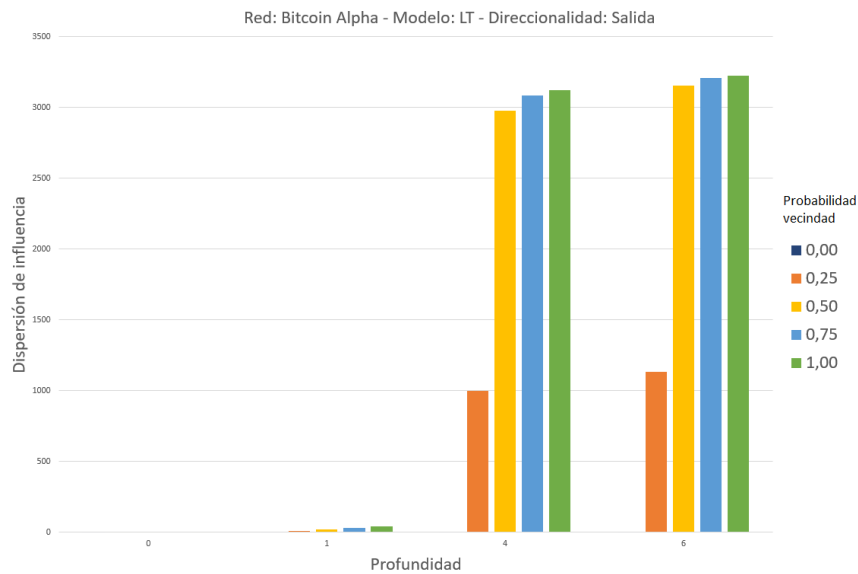


Figura 6.12: Gráfico de resultados red Bitcoin Alpha direccionalidad salida modelo LT.

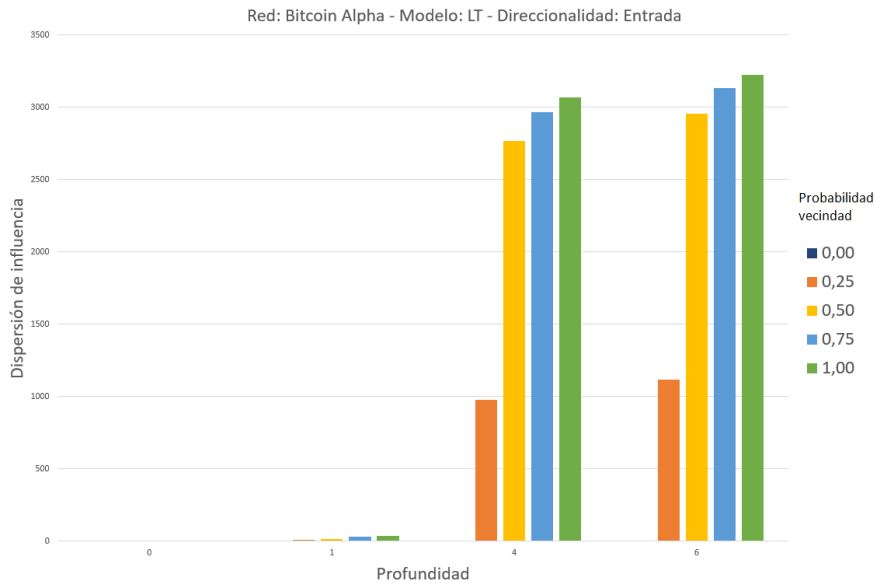


Figura 6.13: Gráfico de resultados red Bitcoin Alpha direccionalidad entrada modelo LT.

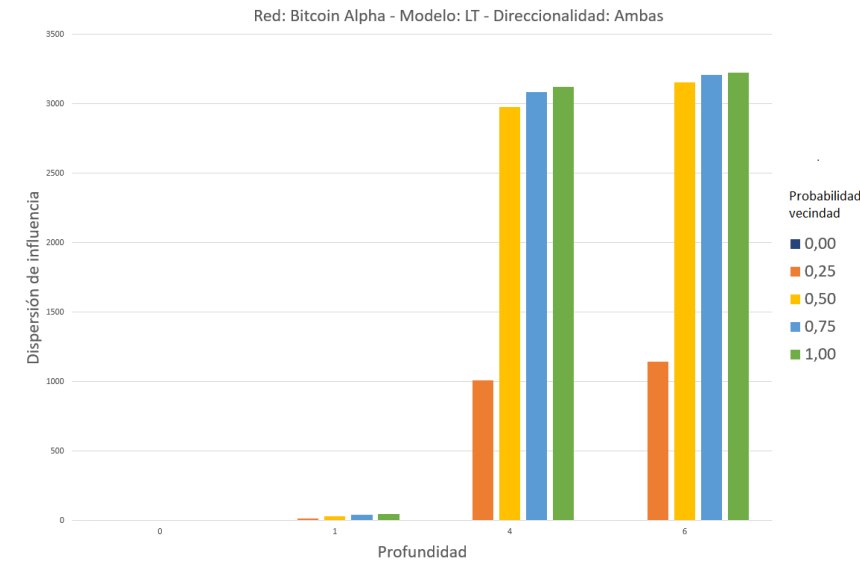


Figura 6.14: Gráfico de resultados red Bitcoin Alpha direccionalidad ambas modelo LT.

### Análisis gráficos red Higgs Twitter

En la red Higgs, ocurre algo particular que se repite para ambos modelos (ver figuras [6.15](#), [6.16](#), [6.17](#), [6.18](#), [6.19](#) y [6.20](#)) y es que debido a la naturaleza de la red, la direccionalidad influye demasiado en la dispersión de influencia (ver los valores del eje Y). Luego, los valores de dispersión de influencia o nodos influenciados son demasiado bajos, esta es la red con mayor cantidad de nodos 38.918 y las dispersiones son muy bajas, llegando a un máximo en el gráfico [6.17](#) de 176.03. Esto es debido nuevamente a que la red es muy dispersa, por lo tanto existen muchos valores que dispersan 1 (es decir, no son capaces de influenciar nuevos nodos) y por tanto, en promedio, se observan valores muy bajos de dispersión (aunque con una muy alta desviación estándar).

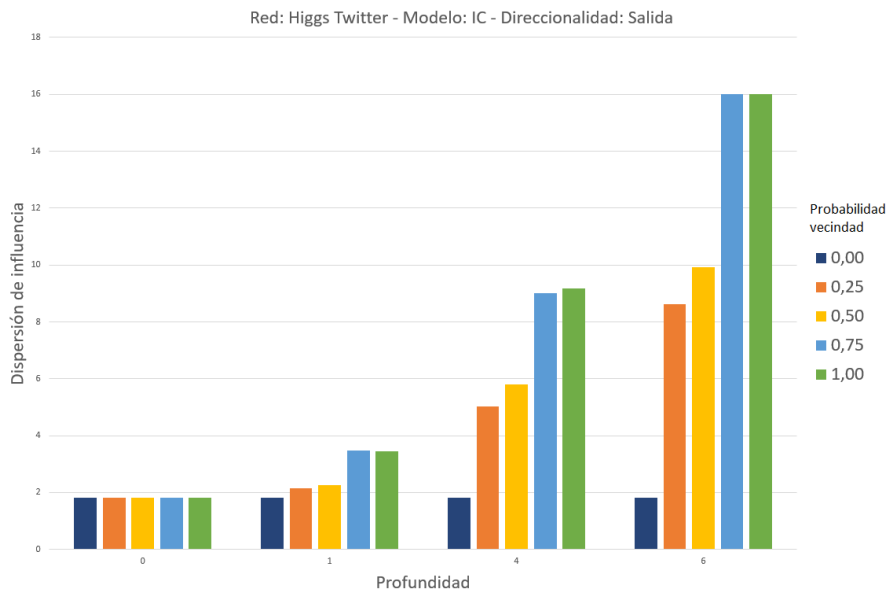


Figura 6.15: Gráfico de resultados red Higgs Twitter direccionalidad salida modelo IC.

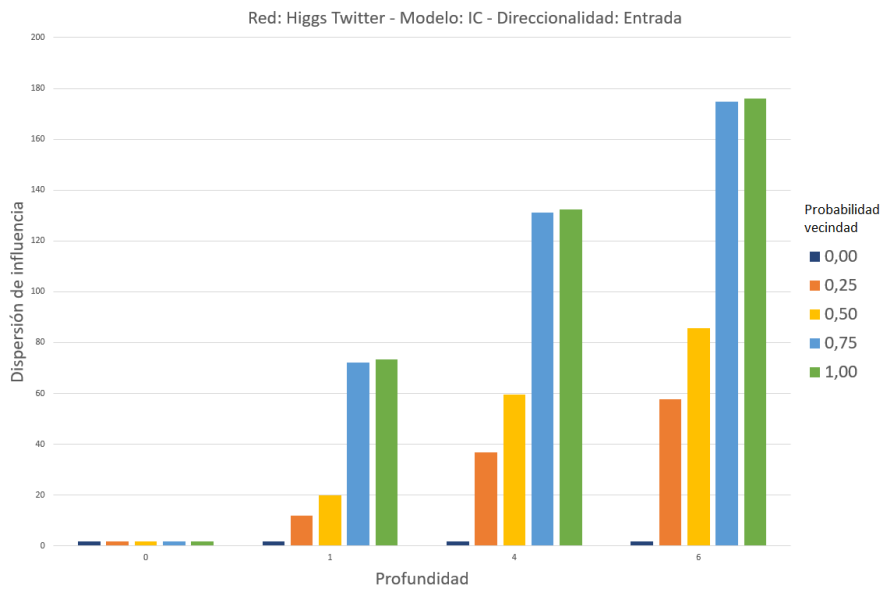


Figura 6.16: Gráfico de resultados red Higgs Twitter direccionalidad entrada modelo IC.

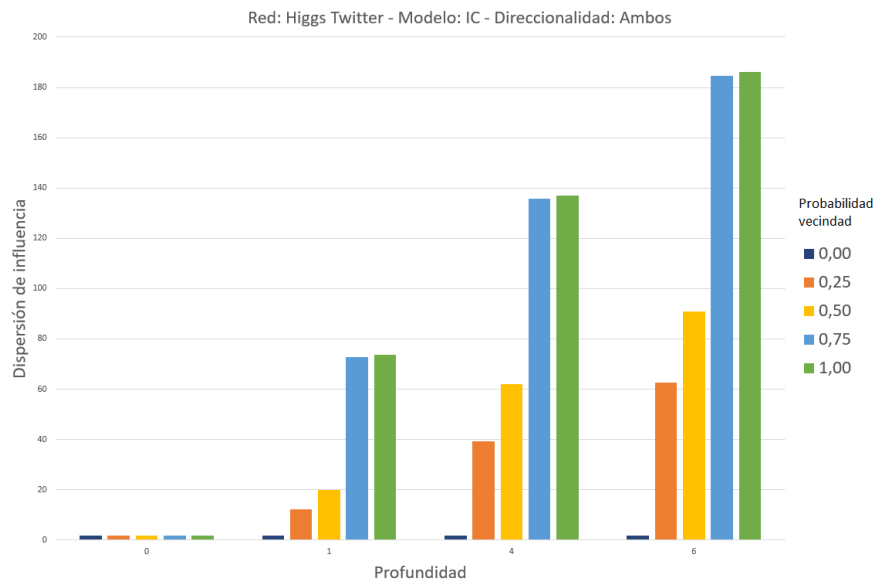


Figura 6.17: Gráfico de resultados red Higgs Twitter direccionalidad ambas modelo IC.

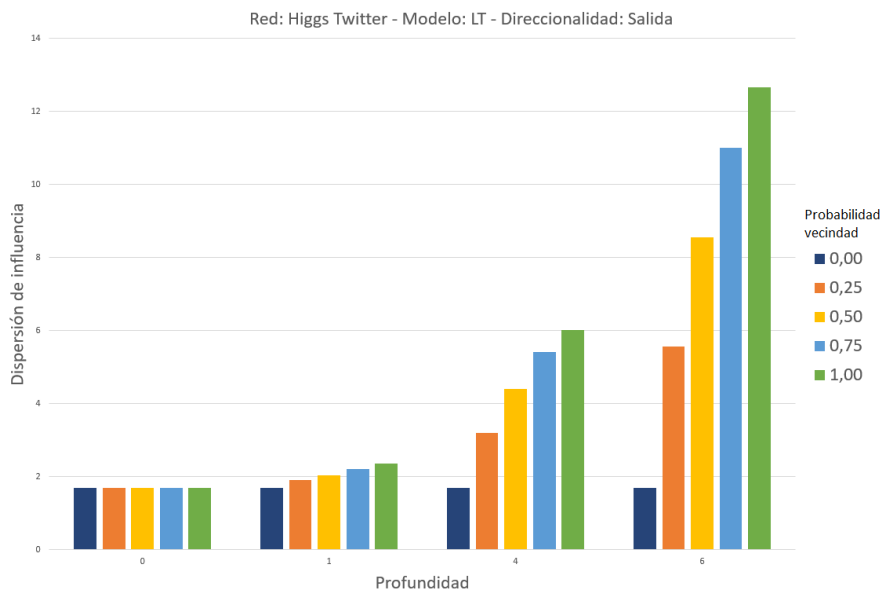


Figura 6.18: Gráfico de resultados red Higgs Twitter direccionalidad salida modelo LT.

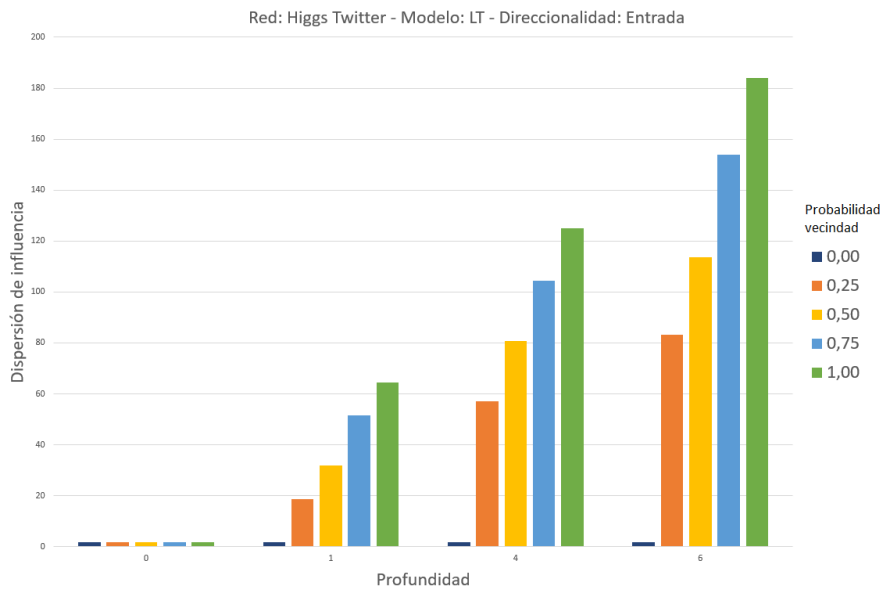


Figura 6.19: Gráfico de resultados red Higgs Twitter direccionalidad entrada modelo LT.

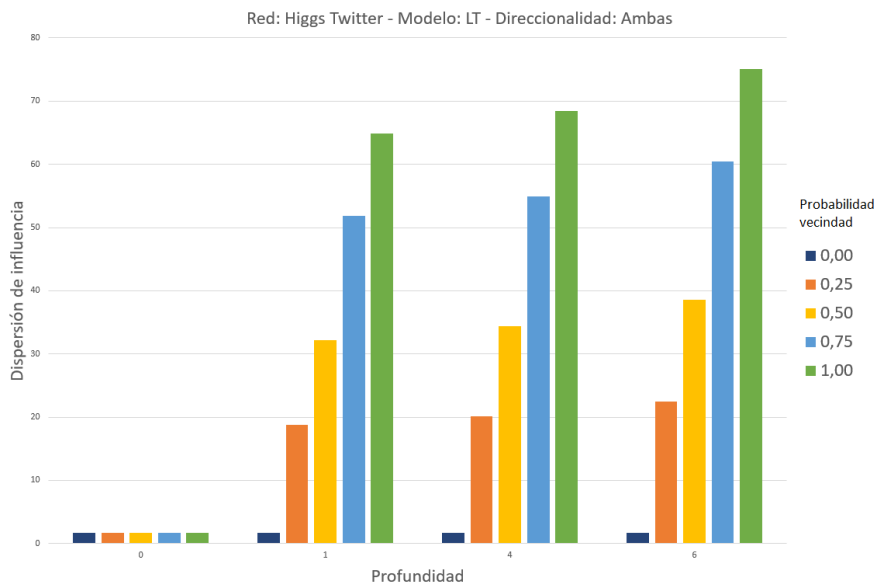


Figura 6.20: Gráfico de resultados red Higgs Twitter direccionalidad ambas modelo LT.

Note que las gráficas anteriores reflejan comportamientos en promedio, que en general tienen altas desviaciones estándar, y que por tanto, si bien ayudan a describir de alguna manera el comportamiento de la medida, no son suficientes para comprender el verdadero

sentido de los parámetros. Es por eso que a continuación recurrimos a análisis de correlación, para entender mejor la incidencia de los distintos ajustes de parámetros en la medida de centralidad generalizada.

### 6.3.3. Análisis matrices de correlaciones

Para entrar en detalle del comportamiento de GISR, se utilizan matrices de correlación, con todas las combinaciones posibles de los ajustes de parámetros que fueron originalmente definidos. Para cada red, existe una matriz de correlación. Los resultados nos entrega información como: comportamientos recurrentes de la medida en las redes, similitudes entre parámetros, comportamientos en distintos tipos de redes y similitudes, entre modelos. Además incluimos comparaciones con algunas medidas de centralidad clásicas y algunas de sus variantes: in-closeness, out-closeness, degree, in-degree, out-degree y betweenness.

De acuerdo con la Sección 4.3, recordemos que se usará el coeficiente de correlación de Spearman, ya que no podemos asumir distribución normal en los datos, y porque lo que nos interesa es comparar los rankings de las centralidades de los nodos, más allá de las diferencias específicas entre sus valores.

Para interpretar las matrices antes es necesario dejar en claro ciertos códigos que se utilizan en ellas.

- Colores. Los colores presentes en la matriz serán tres y significan lo siguiente: Rojo significa una correlación inversa, blanco significa una correlación nula, y verde una correlación positiva. A mayor intensidad de color, mayor correlación de cada tipo. Por último, el color negro se utiliza en aquellas celdas en que el  $p\text{-value} \geq 0.05$ , es decir, cuando los resultados no fueron estadísticamente significativos. Esta gama de colores puede verse más claramente en la Figura 6.21.
- Valores de Correlación. Los valores de correlación están directamente relacionados con los colores. Son el contenido de cada celda de la matriz y se trabajará con un rango para poder clasificarlos, como se observa en la Tabla 6.3 (para valores negativos, es la misma escala pero considerando correlación inversa):
- Códigos de Parámetros. Estos son solamente para dejar claro cuáles parámetros se están comparando y en qué orden aparecen en las matrices. El código sigue la forma  $AXYZ$ , donde:  $AA$  representa el modelo de dispersión, que puede ser  $LT$  o  $IC$ ;  $X$  es el nivel de profundidad, que para estos experimentos fueron 0, 1, 4, 6;  $Y$  es la dirección de la vecindad, donde 0 es ambas direcciones, 1 es sólo salida y 2 es sólo entrada;  $Z$  es la probabilidad de influenciar vecinos, que para estos experimentos se utilizaron 0, 0.25, 0.50, 0.75 y 1.00. Ejemplo: Si la red fuera Football y el código fuera  $IC010.5$ , sería la red Football con modelo de dispersión de influencia

Rango	Nivel de correlación
0.90-1.00	Muy alta
0.70-0.89	Alta
0.50-0.69	Moderada
0.26-0.49	Baja
0.00-0.25	Muy baja

Tabla 6.3: Niveles de correlación (correlación inversa sigue el mismo rango pero con valores negativos).

IC a profundidad 0, con dirección de influencia de salida al 50 % de probabilidad de influenciar vecinos.

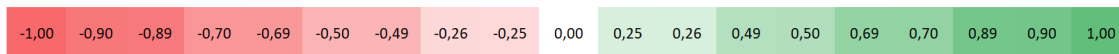


Figura 6.21: Ejemplo de los códigos de colores y valores de correlación.

Antes de hacer la revisión en específico de las matrices de cada red, cabe dejar en claro la composición de esta. En la Figura 6.22 se puede ver que es una matriz cuadrada de 84x84. Como en toda matriz de correlación, la diagonal siempre tendrá valores 1 (máxima correlación positiva), dado que se comparan las variables con sí mismas. La matriz es además simétrica, por lo que la triangular inferior es idéntica a la triangular superior.

	IC000	IC010	IC020	IC100.25	IC100.5	IC100.75
IC000	1,0	0,7	0,7	0,5	0,2	0,1
IC010	0,7	1,0	0,7	0,5	0,2	0,2
IC020	0,7	0,7	1,0	0,5	0,3	0,2
IC100.25	0,5	0,5	0,5	1,0	0,4	0,2
IC100.5	0,2	0,2	0,3	0,4	1,0	0,3
IC100.75	0,1	0,2	0,2	0,2	0,3	1,0

Figura 6.22: Extracto de ejemplo de una matriz real.

Para hacer un análisis, se presentará en primera instancia una vista muy general de la matriz y luego se entrará en detalle de cada una de las redes revisando los patrones que se

observen en la vista general. Se hará especial énfasis en las correlaciones altas y muy altas, debido a que son las que entregan una información más determinante.

### Matrices de correlación red Football

En la Figura 6.23 se ve la vista general de la matriz de correlación para la red Football. Se observan algunos patrones de comportamiento pero lo más llamativo es la cantidad de celdas negras, que como se definieron anteriormente, no representan nada debido a que los datos utilizando la correlación de Spearman arrojaban un  $p$ -value mayor a 0.05. La cantidad de celdas negras se debe al tamaño del archivo; dado que esta red es muy pequeña, no consigue en todos los casos obtener la cantidad de datos suficientes para lograr tener una confianza mayor al 95%.

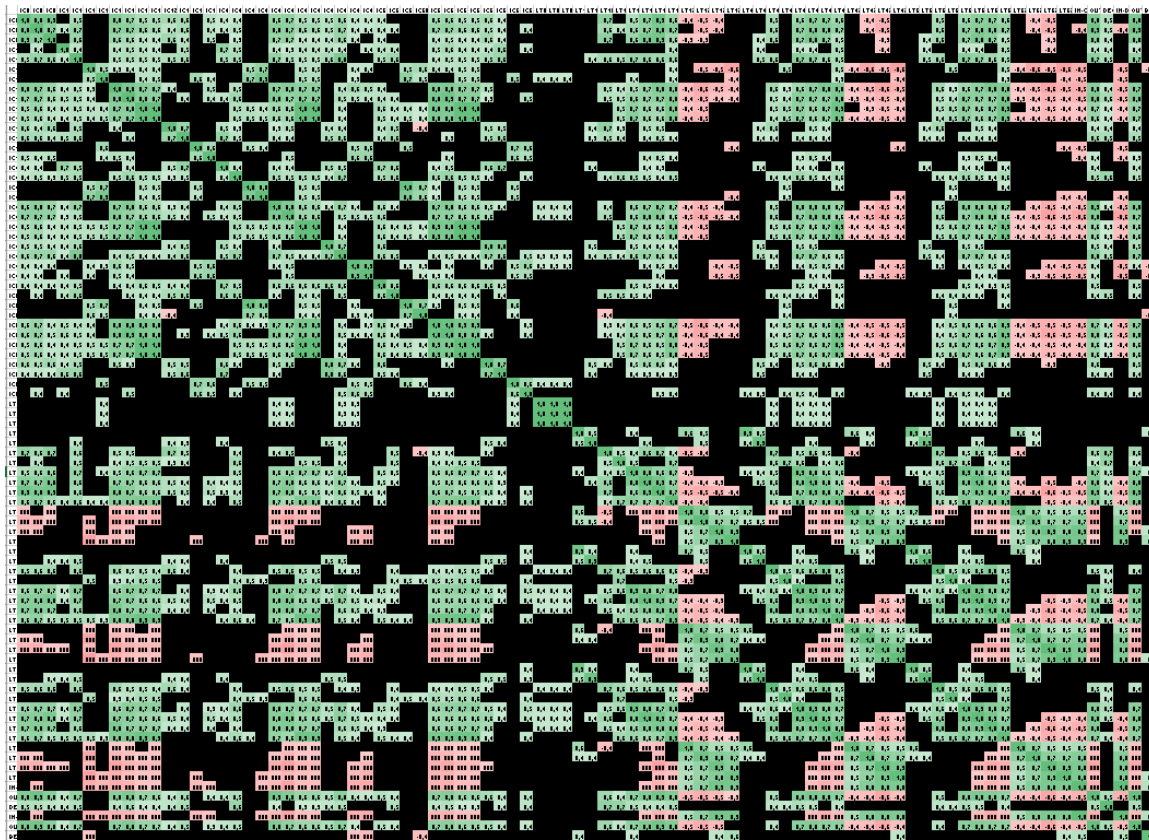


Figura 6.23: Vista general de la matriz Football.

En la Figura 6.24 se encuentra la sección de la matriz que contiene los resultados utilizando IC-Model comparados con sí mismos, utilizando diversos parámetros. Se ve un claro ejemplo de altas correlaciones cuando se utiliza dirección de salida (1) en todas las

probabilidades pero observando un peak en 0.75 y 1 luego la red en específico no muestra patrones ni tampoco variaciones muy altas. El comportamiento de GISR en esta red utilizando IC-Model es bastante estable, dejando de lado el único caso de una correlación inversa que se asocia a un comportamiento del modelo dado que es probabilístico.

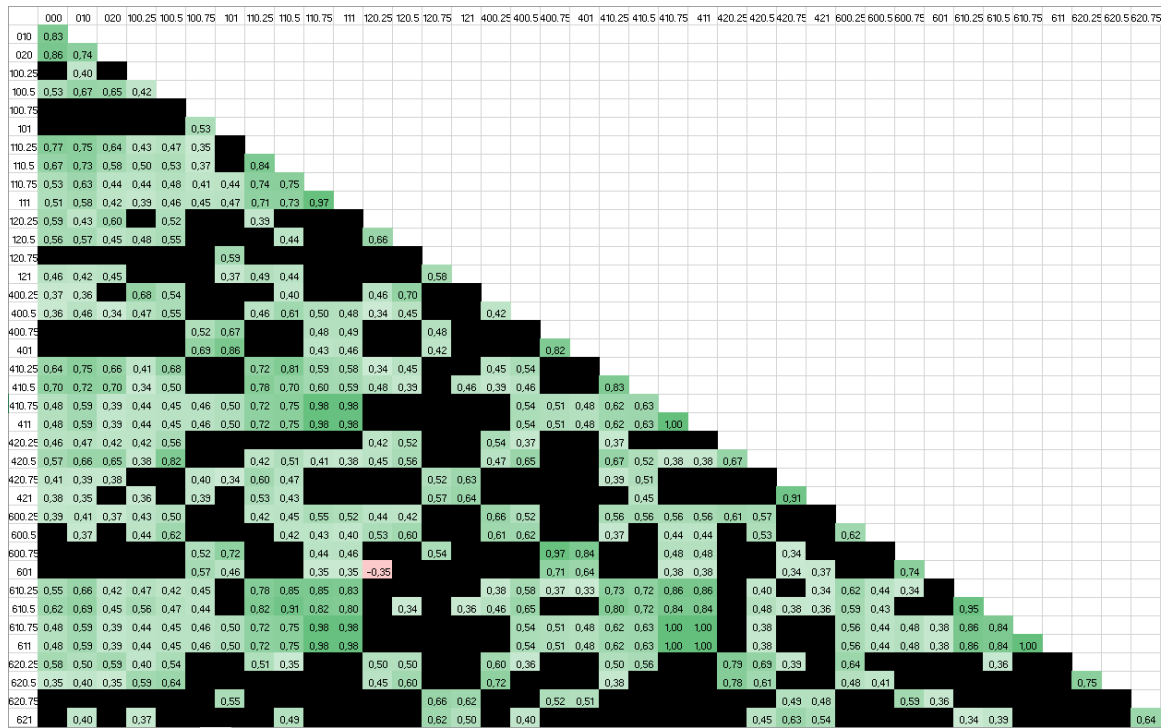


Figura 6.24: Matriz Football sección IC-Model.

En la Figura 6.25 se encuentra la sección de la matriz que contiene los resultados utilizando LT-Model comparados con sí mismos utilizando diversos parámetros antes definidos profundidad, sentido, probabilidad de vecindad. Existe una diagonal de alta correlación que se crea cuando la dirección utilizada como parámetro es la misma y la probabilidad de influenciar vecinos también es la misma. Respecto a la profundidad con los resultados obtenidos no se obtiene mayor información dado que faltarían profundidades para comprobar si esto se replica como si lo hace con los parámetros anteriores. Hay cuadros de correlación inversa que ocurren siempre y cuando se comparen direcciones de salida (1) con las de entrada (2); esto pasa cuando la red tiene muchos nodos que apunten hacia el mismo nodo (caso dirección entrada dominante) y en el sentido inverso cuando un nodo apunta hacia muchos nodos (caso dirección de salida dominante).

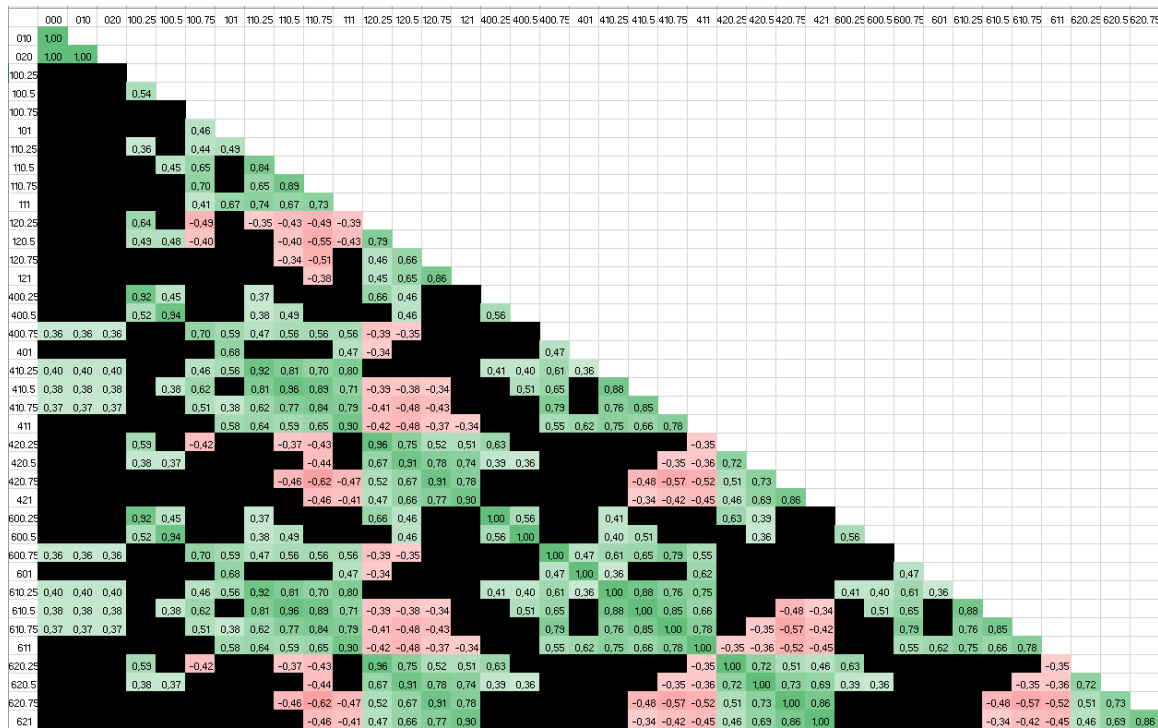


Figura 6.25: Matriz Football sección LT-Model.

En la Figura 6.26 se encuentra la sección de la matriz que contiene los resultados de las comparaciones entre IC-Model (filas) y LT-Model (columnas). Se ve un patrón reiterativo. Cuando LT-Model utiliza dirección de entrada (2), las correlaciones se vuelven inversas para todas las comparaciones con IC-Model. En otro caso, cuando LT-Model utiliza una dirección de salida (1) o ambos (0), tienen correlaciones directas consiguiendo un peak cuando se comparan casos con dirección de salida (1) y probabilidades de dispersión de influencia 0.75 o 1.

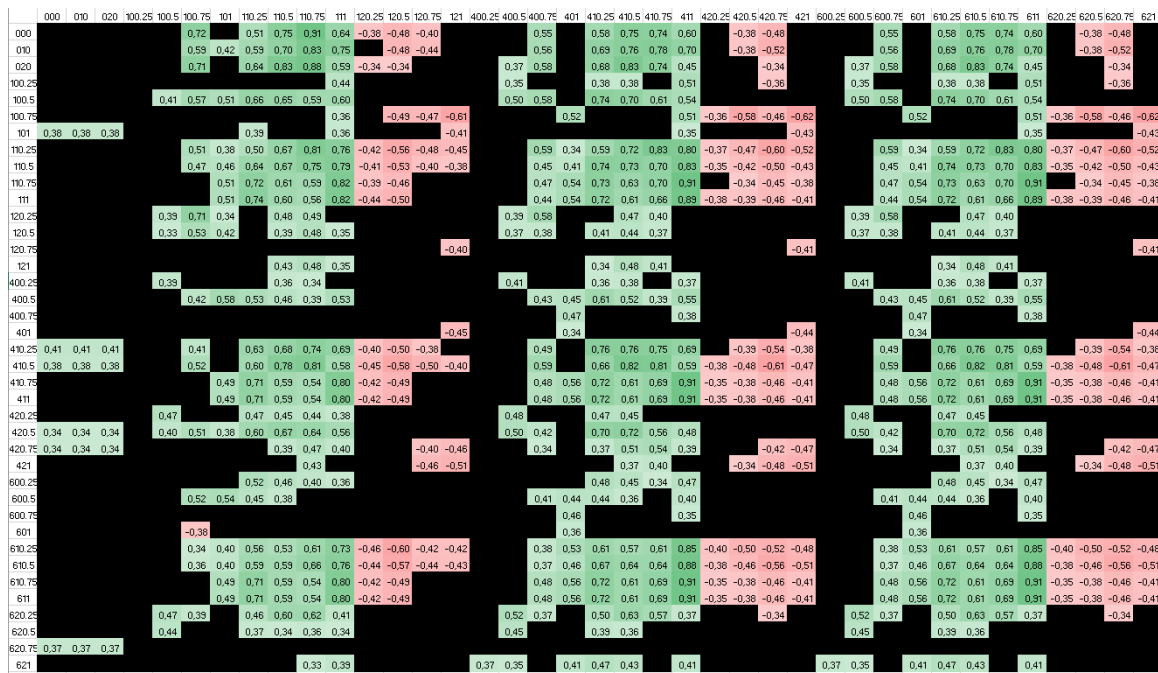


Figura 6.26: Matriz Football sección IC-Model y LT-Model.

En adición, para comparar esta medida basada en modelos de dispersión de influencia con medidas de centralidad clásicas, se tiene la Figura 6.27, que es una sección de la misma matriz general donde se encuentra comparado GISR con las medidas clásicas In-Closseness, Out-Closseness, Degree, In-Degree, Out-Degree, Betweenness.

Las medidas clásicas en esta red tienen un comportamiento esperado. Si vemos los resultados del LT-Model comparado con las medidas clásicas, donde las relaciones van variando entre directas e inversas dependiendo de los parámetros de GISR. En cambio, para IC-Model, es un caso distinto, ya se puede ver que tanto IN-Closseness e IN-Degree tienen correlación inversa no importa los parámetros utilizados en GISR y se repite de manera inversa con OUT-Closseness y OUT-Degree.



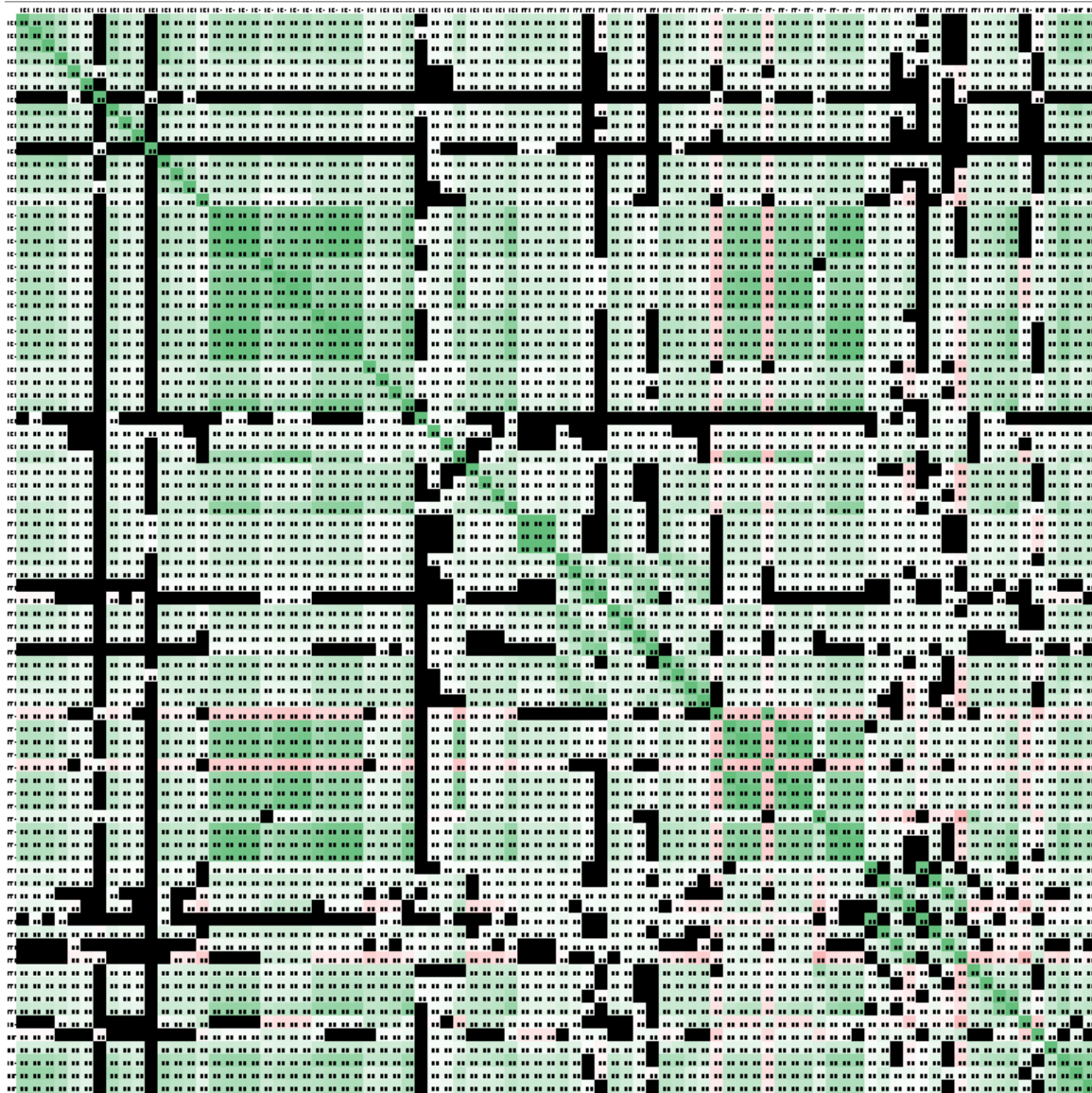


Figura 6.28: Vista general de la matriz Bitcoin Alpha.

En la Figura [6.29](#) se encuentra una sección de la matriz de correlación, en específico donde están los resultados de IC-Model comparados con ellos mismos pero con distintos parámetros.

Se puede ver claramente una correlación alta y muy alta en el nivel de profundidad 4 para la mayoría de combinaciones de parámetros restantes, salvo cuando la direccionalidad de las vecindades es opuesta. Este es uno de los cuatro bloques que se pueden ver en la Figura [6.28](#). Esto se da debido a la densidad de la red. Dado que la profundidad 4 permite una alta dispersión a lo largo de toda la red, produce un comportamiento similar en casi

todas las variaciones de las probabilidades, aunque las correlaciones son aún más altas para probabilidades de influenciar vecinos iguales o mayores a 0.5.

Sin embargo, es muy interesante notar que para profundidades más altas (6), este patrón no se mantiene, aunque pudiésemos pensar erradamente que a mayor profundidad de las vecindades, mayores serán las dispersiones de influencia y por tanto más parecidos los rankings. La explicación de esto nuevamente tiene que ver con la densidad de la red, con la probabilidad de influencia y con el modelo en este caso específico. Para niveles de profundidad más grandes, para probabilidades de activación de los nodos menores a 1, es bastante probable que la activación inicial esté formada por nodos muy dispersos entre sí, de modo que no consiguen entre ellos ayudarse para influenciar a muchos más nodos dentro de la red. Así, la centralidad de los nodos varía bastante entre sí y por ello las correlaciones disminuyen.

Para esta red densa, entonces, el nivel de profundidad 4 es un peak, que permite obtener una activación inicial relativamente dispersa, pero como solo están a lo más separadas por caminos de tamaño 4, estos nodos alcanzan a colaborar para influenciar a varios otros nodos de la red, alcanzando valores más semejantes, que se traducen en correlaciones más altas.

También se ven unos patrones interesantes entre las interacciones de la profundidad 6 con profundidad 4, las correlaciones son altas solamente cuando en profundidad 6 se tiene una probabilidad 1 de que sus vecinos sean influenciados y para el resto de probabilidades se alcanzan correlaciones muy bajas.



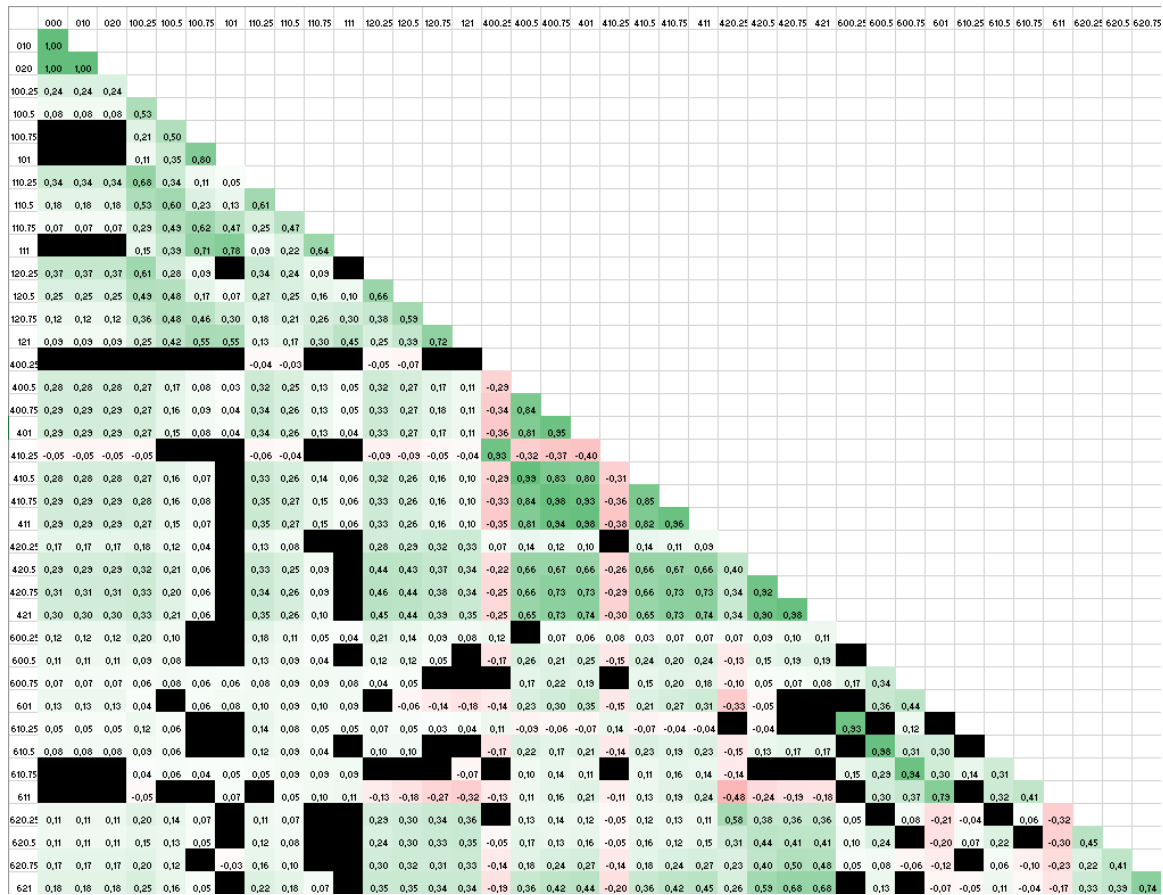


Figura 6.30: Matriz Bitcoin Alpha sección LT-Model.

En la Figura 6.31 está la sección de la matriz donde se encuentran los resultados de las comparaciones entre IC-Model (filas) y LT-Model (columnas).

Nuevamente se ve el bloque de correlaciones presente a profundidad 4. Para LT-Model con probabilidad de activar vecinos de 0.25, existe una correlación inversa o muy baja. El comportamiento de ambos modelos de dispersión es muy similar si las probabilidades son 0.5 o mayores. Luego se ve que las correlaciones más altas alcanzadas son cuando se compara LT-Model en dirección ambas (0) o salida (1) con probabilidades 0.5 o mayores, e IC-Model en salida (1) con probabilidades 0.5 o mayores. También si es que LT-Model tiene dirección entrada (2) con probabilidades 0.5 o mayores e IC-Model con dirección ambas (0) y entrada (2) con cualquier probabilidad.

Para IC-Model con nivel de profundidad 6, hay correlaciones altas cuando la probabilidad de influenciar vecinos es (1), comparándolo con LT-model a profundidad (4) y probabilidades 0.5 o mayores.

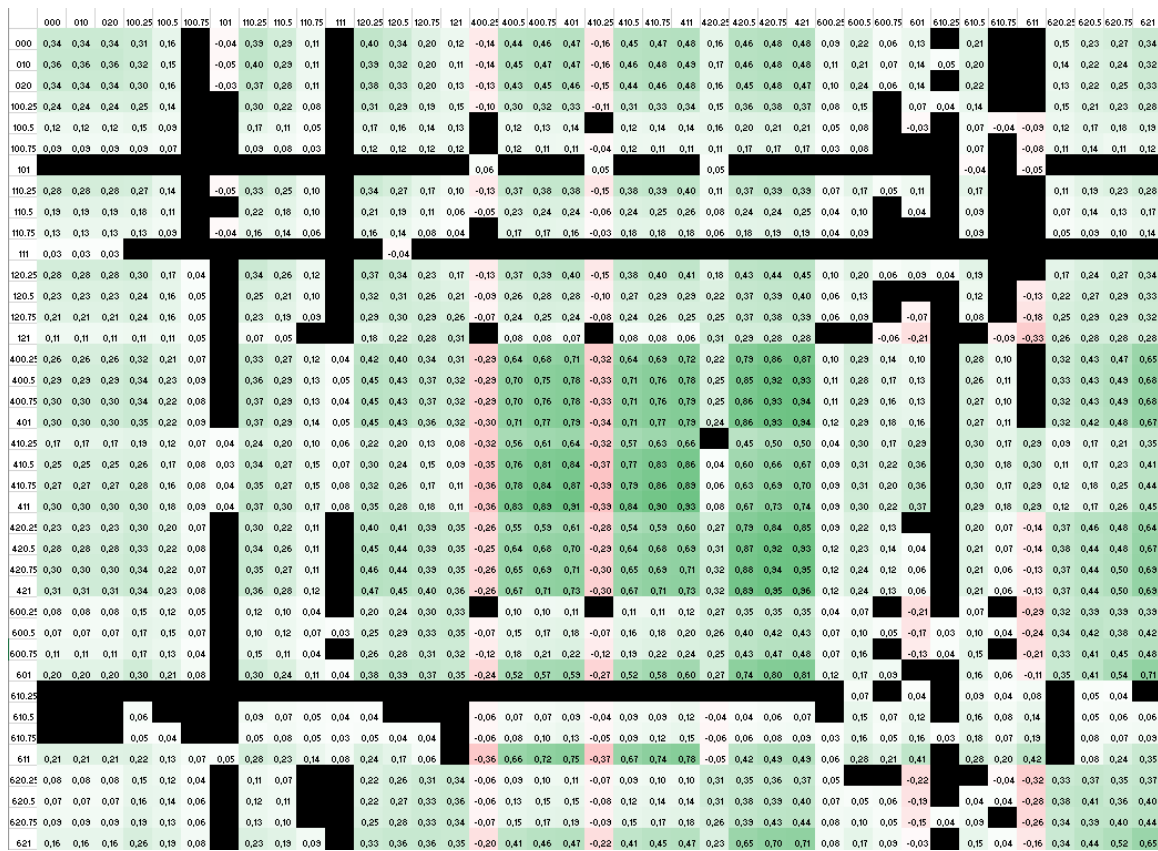


Figura 6.31: Matriz Bitcoin Alpha sección IC-Model y LT-Model.

En adición, para comparar esta medida basada en modelos de dispersión de influencia con medidas clásicas, se tiene la Figura 6.32, donde podemos ver la medida GISR comparada con medidas clásicas como In-Closeness, Out-Closeness, Degree, In-Degree, Out-Degree y Betweenness.

En líneas generales, para esta red no se aprecian altas correlaciones entre dichas medidas, siendo de moderadas a bajas, exceptuando por In-Degree que posee una correlación moderada-alta cuando la profundidad en IC-Model es 4 y las direcciones son ambas (0) o entrada (2). También hay similitudes de comportamiento entre In-Degree, Out-Degree y Betweenness con IC-Model cuando su profundidad es 0.

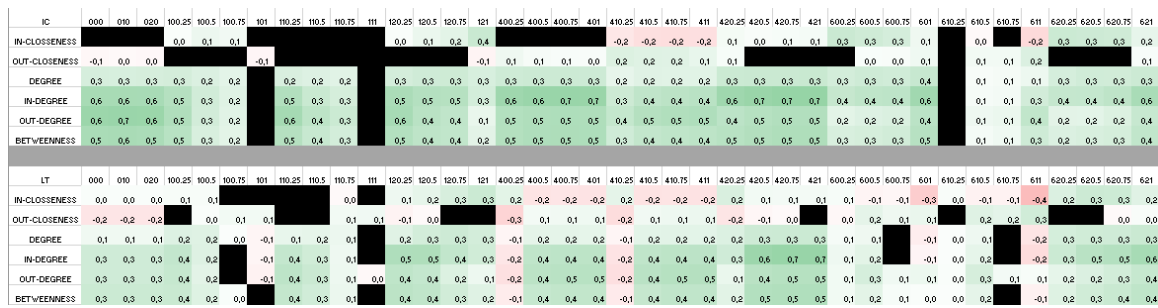


Figura 6.32: Matriz Bitcoin Alpha medidas clásicas con GISR.

### Matrices de correlación red Higgs Twitter

En la Figura 6.33 se ve la vista general de la matriz de correlación para la matriz Higgs Twitter. Se pueden observar diversos patrones que se repiten a lo largo de toda la matriz como cuadrados con alta correlación, algunas líneas horizontales con correlaciones inversas inclusive. Antes de comenzar el análisis de esta red, cabe recordar que es una red muy dispersa y que algunos de sus patrones están directamente relacionados con esto.





comparan direcciones de entrada (2) y ambos (0) y las probabilidades de influenciar vecinos son las mismas en cada caso. Se ve nuevamente el patrón que muestra una correlación directa entre la profundidad 0 y dirección de salida (1) con el resto de profundidades pero en este caso, las correlaciones son altas y muy altas.

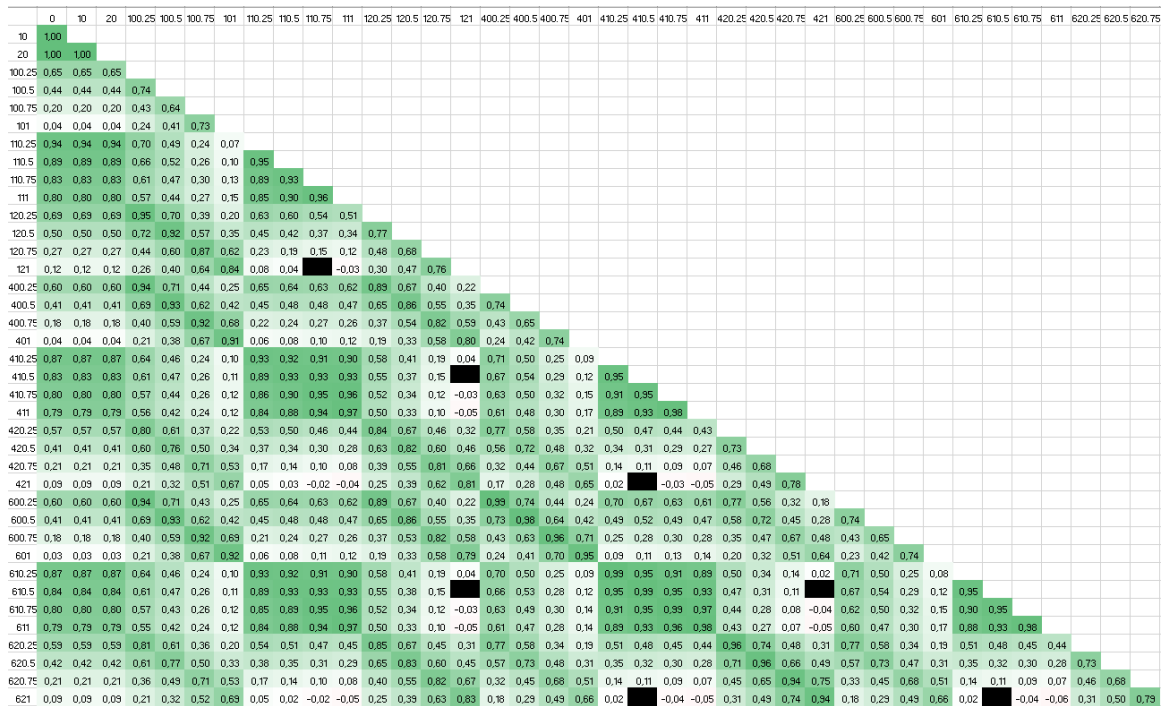


Figura 6.35: Matriz Higgs Twitter sección LT-Model

En la Figura 6.36 se encuentra la sección de la matriz donde se encuentran los resultados de las comparaciones entre IC-Model (filas) y LT-Model (columnas).

Nuevamente se ve el bloque de alta correlación visto en los anteriores análisis. Se nota alta correlación cuando las direcciones son de salida (1) con todas sus probabilidades aunque notando un peak cuando las probabilidades son 0.75 o 1.

	000	010	020	100.25	100.5	100.75	101	110.25	110.5	110.75	111	120.25	120.5	120.75	121	400.25	400.5	400.75	401	410.25	410.5	410.75	411	420.25	420.5	420.75	421	600.25	600.5	600.75	601	610.25	610.5	610.75	611	620.25	620.5	620.75	621		
000		0.53	0.53	0.59	0.41	0.31	0.17	0.08	0.62	0.63	0.66	0.67	0.39	0.26	0.10	0.44	0.33	0.18	0.08	0.63	0.65	0.66	0.67	0.33	0.21	0.07	-0.01	0.44	0.33	0.18	0.08	0.64	0.65	0.66	0.67	0.33	0.22	0.07	-0.01		
010	0.53		0.53	0.59	0.41	0.31	0.17	0.08	0.61	0.63	0.66	0.67	0.38	0.26	0.10	0.43	0.32	0.17	0.07	0.63	0.64	0.66	0.67	0.32	0.21	0.07	-0.01	0.43	0.32	0.17	0.07	0.63	0.64	0.66	0.67	0.33	0.22	0.07	-0.01		
020	0.53	0.53		0.59	0.42	0.31	0.17	0.08	0.62	0.63	0.66	0.67	0.39	0.26	0.10	0.44	0.33	0.18	0.07	0.63	0.65	0.66	0.67	0.33	0.22	0.08	0.44	0.33	0.18	0.08	0.63	0.65	0.66	0.67	0.34	0.22	0.08	0.44			
100.25	0.46	0.46	0.46		0.34	0.27	0.17	0.11	0.48	0.50	0.52	0.53	0.32	0.23	0.12	0.05	0.37	0.29	0.18	0.10	0.50	0.52	0.52	0.53	0.28	0.20	0.09	0.03	0.37	0.29	0.18	0.10	0.50	0.51	0.52	0.53	0.29	0.20	0.10	0.03	
100.5	0.38	0.38	0.38	0.38		0.28	0.18	0.14	0.40	0.42	0.44	0.45	0.28	0.22	0.14	0.09	0.33	0.27	0.18	0.13	0.42	0.43	0.44	0.44	0.26	0.19	0.11	0.06	0.32	0.27	0.18	0.13	0.42	0.43	0.44	0.44	0.28	0.19	0.11	0.07	
100.75	0.10	0.10	0.10	0.10	0.20		0.34	0.45	0.02	0.04	0.06	0.08	0.08	0.16	0.27	0.37	0.14	0.22	0.34	0.43	0.06	0.07	0.07	0.07	0.13	0.18	0.25	0.23	0.14	0.21	0.34	0.43	0.06	0.07	0.07	0.07	0.12	0.17	0.26	0.32	
110.25	0.62	0.62	0.62	0.62	0.44	0.33		0.18	0.08	0.65	0.67	0.70	0.72	0.41	0.27	0.10	0.48	0.35	0.19	0.08	0.67	0.69	0.70	0.71	0.35	0.23	0.07	-0.02	0.46	0.35	0.19	0.08	0.67	0.69	0.70	0.71	0.35	0.23	0.08	-0.02	
110.5	0.65	0.65	0.65	0.65	0.45	0.34	0.19		0.03	0.68	0.70	0.73	0.75	0.42	0.28	0.11	0.48	0.36	0.20	0.08	0.70	0.72	0.74	0.74	0.36	0.23	0.08	-0.02	0.48	0.36	0.20	0.09	0.70	0.72	0.74	0.74	0.37	0.24	0.08	-0.02	
110.75	0.78	0.78	0.78	0.78	0.55	0.42	0.24	0.11		0.84	0.88	0.93	0.97	0.50	0.33	0.10	-0.05	0.61	0.46	0.26	0.11	0.85	0.93	0.95	0.96	0.43	0.27	0.07	-0.05	0.61	0.46	0.25	0.11	0.85	0.92	0.95	0.96	0.44	0.28	0.07	-0.06
120.25	0.42	0.42	0.42	0.42	0.32	0.25	0.16	0.10		0.44	0.45	0.47	0.48	0.30	0.22	0.12	0.05	0.34	0.27	0.17	0.10	0.45	0.47	0.47	0.48	0.27	0.19	0.09	0.04	0.34	0.26	0.16	0.10	0.45	0.46	0.47	0.48	0.27	0.19	0.10	0.08
120.5	0.31	0.31	0.31	0.31	0.26	0.22	0.16	0.12	0.33		0.34	0.35	0.36	0.24	0.20	0.13	0.09	0.27	0.23	0.16	0.12	0.34	0.35	0.35	0.36	0.23	0.18	0.10	0.07	0.27	0.23	0.16	0.12	0.34	0.35	0.35	0.36	0.22	0.18	0.11	0.08
120.75	-0.18	-0.18	-0.18	-0.18	-0.03	0.07	0.23	0.34	-0.19	-0.19	-0.19	-0.20	-0.02	0.09	0.27	0.39	-0.03	0.06	0.21	0.32	-0.18	-0.19	-0.20	-0.20	0.05	0.13	0.25	0.33	-0.03	0.06	0.21	0.32	-0.18	-0.19	-0.20	-0.20	0.03	0.12	0.26	0.35	
400.25	0.32	0.32	0.32	0.32	0.25	0.20	0.13	0.07	0.35	-0.20	-0.20	-0.21	-0.02	0.10	0.27	0.41	-0.03	0.07	0.21	0.33	-0.18	-0.20	-0.21	-0.22	0.05	0.13	0.25	0.34	-0.03	0.06	0.21	0.33	-0.19	-0.20	-0.21	-0.22	0.03	0.12	0.26	0.37	
400.5	0.24	0.24	0.24	0.24	0.21	0.18	0.13	0.10	0.27	0.29	0.32	0.34	0.18	0.14	0.07	0.04	0.25	0.22	0.15	0.10	0.31	0.33	0.34	0.34	0.27	0.20	0.09	0.02	0.25	0.22	0.15	0.10	0.31	0.33	0.34	0.34	0.26	0.18	0.08	0.03	
400.75	0.09	0.16	0.23	0.40	0.02	0.04	0.06	0.08	0.07	0.12	0.23	0.31	0.11	0.18	0.33	0.43	0.04	0.07	0.09	0.10	0.09	0.18	0.30	0.39	0.11	0.19	0.33	0.43	0.04	0.07	0.09	0.10	0.18	0.30	0.39	0.11	0.18	0.30	0.37		
410.25	0.60	0.60	0.60	0.60	0.43	0.33	0.19	0.10		0.84	0.87	0.92	0.97	0.50	0.32	0.10	-0.05	0.60	0.47	0.28	0.15	0.88	0.93	0.97	0.99	0.43	0.27	0.07	-0.05	0.60	0.48	0.29	0.14	0.88	0.93	0.97	0.99	0.44	0.28	0.07	-0.06
410.5	0.62	0.62	0.62	0.62	0.45	0.35	0.21	0.11	0.67	0.71	0.75	0.78	0.40	0.27	0.10	-0.02	0.51	0.40	0.23	0.11	0.73	0.76	0.78	0.78	0.35	0.23	0.07	-0.03	0.51	0.40	0.23	0.11	0.73	0.76	0.78	0.78	0.36	0.23	0.07	-0.03	
410.75	0.78	0.78	0.78	0.78	0.55	0.41	0.24	0.11	0.83	0.87	0.93	0.96	0.50	0.32	0.10	-0.05	0.60	0.47	0.28	0.15	0.88	0.92	0.97	0.98	0.42	0.27	0.07	-0.05	0.60	0.48	0.29	0.14	0.88	0.93	0.97	0.98	0.44	0.28	0.07	-0.06	
420.25	0.28	0.28	0.28	0.28	0.22	0.18	0.11	0.05	0.30	0.31	0.33	0.34	0.21	0.16	0.08	0.03	0.24	0.19	0.11	0.05	0.32	0.33	0.34	0.34	0.30	0.20	0.07	-0.01	0.24	0.19	0.11	0.04	0.32	0.33	0.33	0.34	0.28	0.18	0.07	-0.06	
420.5	0.18	0.18	0.18	0.18	0.17	0.15	0.10	0.08	0.20	0.21	0.22	0.23	0.16	0.13	0.09	0.07	0.18	0.15	0.10	0.07	0.21	0.22	0.22	0.23	0.23	0.25	0.19	0.10	0.04	0.18	0.15	0.10	0.07	0.21	0.22	0.22	0.23	0.24	0.18	0.10	0.06
420.75	-0.20	-0.20	-0.20	-0.20	-0.06	0.04	0.19	0.30	-0.20	-0.21	-0.21	-0.21	-0.05	0.06	0.23	0.35	-0.06	0.03	0.18	0.29	-0.20	-0.21	-0.22	-0.22	0.14	0.31	0.43	-0.06	0.03	0.18	0.28	-0.20	-0.21	-0.21	-0.22	0.14	0.31	0.41			
600.25	0.32	0.32	0.32	0.32	0.25	0.20	0.13	0.07	0.35	-0.20	-0.20	-0.21	-0.02	0.10	0.27	0.41	-0.03	0.07	0.21	0.33	-0.18	-0.20	-0.21	-0.22	0.05	0.13	0.25	0.34	-0.03	0.06	0.21	0.33	-0.19	-0.20	-0.21	-0.22	0.03	0.12	0.26	0.37	
600.5	0.26	0.26	0.26	0.26	0.22	0.19	0.15	0.11	0.29	0.31	0.34	0.36	0.20	0.15	0.08	0.04	0.26	0.23	0.16	0.11	0.33	0.35	0.36	0.36	0.26	0.20	0.10	0.03	0.26	0.23	0.16	0.11	0.33	0.35	0.36	0.36	0.26	0.20	0.10	0.03	
600.75	0.10	0.17	0.31	0.42	0.03	0.04	0.07	0.09	0.08	0.14	0.24	0.33	0.12	0.20	0.34	0.44	0.05	0.07	0.09	0.10	0.18	0.29	0.37	0.12	0.21	0.35	0.46	0.05	0.08	0.10	0.11	0.10	0.19	0.32	0.41						
610.25	0.60	0.60	0.60	0.60	0.44	0.33	0.19	0.10	0.85	0.88	0.92	0.97	0.50	0.32	0.10	-0.05	0.61	0.47	0.27	0.14	0.89	0.92	0.96	0.98	0.42	0.27	0.06	-0.06	0.60	0.47	0.29	0.15	0.88	0.93	0.97	0.98	0.44	0.28	0.07	-0.06	
610.5	0.63	0.63	0.63	0.63	0.45	0.35	0.20	0.11	0.88	0.91	0.96	0.99	0.41	0.27	0.10	-0.02	0.51	0.39	0.23	0.11	0.74	0.76	0.78	0.78	0.36	0.23	0.07	-0.03	0.51	0.40	0.23	0.11	0.74	0.77	0.78	0.78	0.36	0.23	0.07	-0.03	
610.75	0.78	0.78	0.78	0.78	0.55	0.41	0.24	0.11	0.83	0.87	0.93	0.96	0.50	0.32	0.10	-0.05	0.60	0.47	0.28	0.15	0.88	0.92	0.96	0.98	0.42	0.27	0.06	-0.06	0.60	0.47	0.28	0.15	0.88	0.92	0.97	0.98	0.43	0.28	0.07	-0.06	
620.25	0.23	0.23	0.23	0.23	0.23	0.18	0.11	0.06	0.31	0.32	0.34	0.35	0.21	0.16	0.08	0.03	0.24	0.19	0.11	0.05	0.32	0.34	0.34	0.34	0.30	0.20	0.07	-0.01	0.24	0.19	0.11	0.05	0.32	0.34	0.34	0.34	0.28	0.18	0.07	-0.06	
620.5	0.20	0.20	0.20	0.20	0.17	0.15	0.11	0.08	0.21	0.22	0.24	0.24	0.16	0.14	0.09	0.06	0.18	0.16	0.11	0.07	0.22	0.23	0.24</																		

Figura 6.37: Matriz Higgs Twitter sección medidas clásicas con GISR

## 6.4. Discusión de resultados

Respecto a las pruebas realizadas, los resultados y los cálculos estadísticos obtenidos, finalmente se puede responder a las preguntas de investigación propuestas en el capítulo 3: ¿Cómo afectan los niveles de profundidad de las vecindades de los actores para conformar activaciones iniciales o los procesos de dispersión de influencia? y ¿Qué ventajas obtienen las medidas basadas en modelos de dispersión de influencia sobre las medidas de centralidad independientes del modelo?

Cabe destacar que para el análisis de estadísticos simples, los resultados que se obtuvieron tenían desviaciones estándar en algunos casos mayores a los promedios, por lo que no resultaron suficientemente descriptivos. Es por eso que los resultados más importante de este trabajo son los obtenidos de los análisis de correlación, donde se comparan los rankings de las medidas de centralidad para todos los nodos de cada red, a través de lo cual se obtuvieron patrones significativos para ciertos ajustes de parámetros.

### 6.4.1. Discusión de los resultados estadísticos

Revisando lo analizado en la sección anterior, se puede decir que el nivel de profundidad afecta de manera significativa en el resultado de la dispersión de influencia, incluso, en general a mayor nivel de profundidad, mayores serán los resultados de la dispersión de influencia. Sin embargo, para redes muy densas se ve que con menor profundidad se puede llegar a la misma o similar cantidad de usuarios influenciados. Para la red densa Bitcoin Alpha, se observa que un nivel de profundidad 4 es un punto crítico para poder alcanzar altos valores de centralidad (dispersión de influencia) para la mayoría de nodos de la red. En cambio, para la red dispersa Higgs Twitter, se ve que el peak se alcanza más tarde, para una profundidad de 6. Estos valores coinciden con la teoría de los 6 grados de separación, aunque asumiendo altos niveles de conectividad en la red de comunicación, y con los 4 grados de separación considerados en la actualidad para redes sociales online.

Note que la densidad de una red tiene relación directa con su naturaleza y no con los parámetros utilizados en la medida de centralidad generalizada. Sería interesante como trabajo futuro hacer experimentos para redes con distintos niveles de densidad.

Otro dato interesante que tiene relación directa con la naturaleza de la red y se puede acceder vía parámetros es la direccionalidad. Dependiendo de la red, puede ser más beneficioso realizar una búsqueda de vecindad en sentido de salida o llegada. Dado que utilizar la direccionalidad en ambos sentidos es la unión de las dos direcciones previas, siempre será mejor que utilizar una sola en términos de dispersión de influencia a través de la red. En general, se observa que la direccionalidad no varía demasiado los resultados para redes más densas como la de Bitcoin Alpha. Será interesante a futuro verificar si esto se cumple en general para las redes densas.

Respecto a la probabilidad de vecindad, en el conjunto de semillas iniciales se puede ver en los gráficos de los resultados estadísticos que en líneas generales ofrecen una variación sobre la dispersión de influencia. Pero esta última variación mencionada, depende de la topología de la red. Esta mejoría puede cambiar completamente el resultado o solamente agregar un poco de dispersión como los casos de red dispersa Higgs Twitter, donde la variación es notoria entre 50 % de probabilidades y 75 % en cambio dentro del rango 25-50 % y el rango 75-100 % no genera tanta variación. Luego en redes densas como Bitcoin Alpha utilizar el 25 % de probabilidad es completamente inviable si se busca maximizar la dispersión, pero utilizar el 50 % en una búsqueda de minimizar recursos es igual de viable que 75 % y 100 %.

### 6.4.2. Discusión de las matrices de correlación

En líneas generales, gracias al estudio de las matrices de correlación se puede concluir que en redes no densas, como los casos de red Football y Higgs Twitter, existen patrones de comportamiento que se repiten a lo largo de los niveles de profundidad. Esto último dado que, independiente de los niveles de profundidad, los resultados arrojan una alta o muy alta correlación, para probabilidades positivas (o al menos mayores o iguales a 0.25) de influenciar vecindades, y mientras coincidan las direccionalidades de las vecindades. Por lo tanto, en redes más dispersas, el nivel de profundidad no es un parámetro tan crítico como veremos más adelante para las redes densas. En cambio, los parámetros de direccionalidad y probabilidad de vecindades generan una gran variabilidad en el comportamiento de la medida de centralidad. Respecto a la direccionalidad, queda claro que los comportamientos serán similares si se compara la misma dirección de vecinos sólo cuando la red en general tengan más vecinos de salida que de entrada o en el caso contrario, en cambio si fuese equivalente o una red sin direccionalidad, este parámetro no influiría.

Para redes más densas, el comportamiento de la medida de centralidad es bastante diferente. En específico en este estudio se trabajó con la red Bitcoin Alpha una red muy densa. Consigue un peak de correlación a profundidad 4, manteniendo las altas correlaciones sólo a esta profundidad sin generar una gran variación respecto a los otros parámetros, exceptuando la probabilidad de influenciar las vecindades 0.25 o bien 25 % que también se generan correlaciones bajas o inversas. Por lo tanto, a diferencia de las redes más dispersas

en redes densas, el nivel de profundidad es el parámetro crítico dado a que el comportamiento de altas correlaciones se consigue en cierto nivel de profundidad.

Respecto a los modelos utilizados para las dispersiones de influencia, en general, los comportamientos que presentan comparándose entre modelos son similares a los comportamientos generados comparados con ellos mismos. Por lo tanto se puede concluir que independiente de la densidad de la red y parámetros, ambos modelos de dispersión de influencia se comportan de manera similar. Existen sutiles diferencias que se dan en base que IC-Model es probabilístico y LT-Model es determinista, pero, al comprobar patrones, no son tan relevantes dado que los principales comportamientos vistos de altas y muy altas correlaciones se mantienen.

Para terminar con la discusión de resultados tanto estadísticos como de correlación, hay que recordar que todo lo hablado anteriormente se puede aplicar directamente al análisis de redes sociales, donde los actores son usuarios, personas, organizaciones o entidades. Así, para redes no densas, los resultados dependerán mucho de qué personas se requiera influenciar, si los que tienen contacto directo con el actor o si el actor tiene el contacto directo con ellas (esto hace referencia a la direccionalidad y en la práctica, puede ser que ellos te conozcan y no tú a ellos, o viceversa) o qué tanto confíen en el actor, si todos sus vecinos pueden ser influenciados por él por la alta confianza o ninguno de ellos es influenciado por la baja confianza (refiriéndose a la probabilidad de vecindades). En casos de redes densas, dado que los actores están altamente interrelacionados entre sí, los dos parámetros anteriores pierden más importancia, y pasa a ser más relevante qué tan lejos puede llegar el actor en la activación inicial de su vecindad a distintos niveles, es decir, si es capaz o no de influenciar los contactos de sus contactos, a los contactos de dichos contactos, y así sucesivamente.

# Capítulo 7

## Implantación

En este Capítulo se abarca todo lo relacionado con la implantación de la plataforma de experimentación GISR. Considerando los requerimientos de Hardware y Software, cómo preparar el ambiente del sistema para poder ejecutar esta plataforma, documentación asociada y su manual de usuario. Por lo tanto este capítulo es de suma importancia para que el usuario logre entender cómo utilizar la plataforma.

### 7.1. Requerimientos

#### 7.1.1. Requerimientos de mínimos

Dado que la plataforma de experimentación se ejecuta en Shell o Línea de Comandos, presenta esencialmente requerimientos de Software y de Hardware mínimos para utilizar la computadora como tal.

- Linux- Ubuntu 16.04 a 17.10, o Windows 7 a 10.
- 2GB RAM y preferiblemente 4GB.
- 5GB de espacio en disco duro para instalación de librerías.

en caso de utilizar Anaconda como distribución de Python.

- Windows 8 o superior, 64-bit macOS 10.13 o superior, Linux, including Ubuntu, RedHat, CentOS 7 o superior, y otros.
- Windows 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bit aarch64 (AWS Graviton2 / arm64), 64-bit Power8/Power9.
- 5GB de espacio en disco duro para descarga e instalación.

### 7.1.2. Requerimientos de recomendados

Como recomendación, se darán las especificaciones de las máquinas utilizadas para la ejecución de las pruebas.

- Procesador: Intel Core I7-8700T.
- RAM: 8GB
- Almacenamiento: 50GB
- SO: Windows 10 Home Single Language 64-bit.

## 7.2. Preparación de ambiente

Para poder realizar pruebas en la plataforma de experimentación, es necesario tener instaladas ciertas librerías, programas y lenguaje de programación cruciales para el funcionamiento de la plataforma. A continuación se detallarán los necesarios de tener instalados:

- Python 3: Este es el lenguaje de programación en el que está desarrollada la totalidad de la plataforma por ende es obligatoria su instalación para lograr ejecutarla.
- Anaconda: Esta es una distribución de Python que además, instala Python y además algunas dependencias necesarios para el funcionamiento, todas pueden ser instaladas por separado pero esto acelera el proceso de buscar la web oficial de la librería en este caso.
- NetworkX: Es una librería que permite la manipulación de grafos en Python. Es una de las librerías que Anaconda trae integradas.
- Matplotlib: Es una librería que dota al lenguaje Python con funciones que permiten generar y visualizar gráficos entre otras cosas. Es una de las librerías que Anaconda trae integradas
- Numpy: Es una librería que dota al lenguaje Python con cálculos matemáticos avanzados. Es una de las librerías que anaconda trae integradas.
- SciPy: Es una librería que dota al lenguaje Python con cálculos matemáticos avanzados. Es una de las librerías que anaconda trae integradas.
- GIT: Esta herramienta permite utilizar GIT en nuestra computadora. Con ella podremos descargar el proyecto.
- Spyder o Visual Studio Code: Estas dos herramientas son editores de código y ambos incluyen su propia línea de comandos donde es posible testear la plataforma de experimentación. Ambas vienen integradas dentro de Anaconda.

### 7.3. Manual de usuario

En Apéndice [B](#), se encuentra la totalidad del manual de usuario, donde nos encontramos con lo necesario para lograr cumplir con la instalación y ejecución de la plataforma de experimentación. A continuación se detalla cada sección del manual:

- En Introducción [B.2](#) se recuerda cómo está compuesta la plataforma y dónde poder descargar el proyecto que la contiene.
- En Requisitos Previos [B.3](#) se encuentra lo necesario en Software para levantar esta plataforma y ejecutarla.
- En Instalación y Configuración [B.4](#) se encuentra la manera de ejecutar la plataforma y cómo modificar cada archivo importante para las entradas de esta.
- En Manejo del Sistema [B.5](#) se encuentra todo lo relacionado a utilización del sistema. Habiendo cubierto las entradas en el capítulo anterior, este muestra las ejecuciones reales y resultados reales de todo el proceso, desde que se ejecuta la plataforma hasta que se hacen cálculos estadísticos y correlaciones en ella.
- Cambios al Sistema [B.6](#) le menciona al lector/usuario del sistema, cuáles son algunas de las visiones de los primeros desarrolladores de esta plataforma.
- Comentarios Finales [B.7](#) es sólo una despedida al usuario.

# Capítulo 8

## Conclusiones

Las medidas de centralidad y modelos de dispersión de influencia son dos de los conceptos más estudiados en el área de análisis en redes sociales. Aunque cada uno se ha estudiado en profundidad de manera independiente, existen menos estudios que los estudian de manera conjunta. Las medidas de centralidad basadas en modelos de dispersión de influencia son un ejemplo de esta convergencia. Es por esto que este trabajo se enfoca en este nuevo concepto realizando una plataforma de experimentación para una medida de centralidad basada en modelos de dispersión de influencia generalizada, que mide cuál es la capacidad de un nodo y sus vecinos para poder dispersar su influencia a través de una red.

Dentro de la plataforma de experimentación recién mencionada, se pueden probar distintos parámetros para conocer diversas formas de ejecución de la misma red: Profundidad, se refiere a los niveles de vecindad que considerará como nodos de activación iniciales sobre los cuales aplicará el modelo de dispersión de influencia; Modelo, se refiere a cuál será el modelo de dispersión de influencia a aplicar; Dirección, que hace referencia en qué sentido se activarán los vecinos del actor; Probabilidad de vecindad, que es la probabilidad de que los vecinos de un nodo sean activados antes de aplicar el modelo de dispersión de influencia.

A partir de diversos ajustes de parámetros, se realizaron 255 experimentos que abarcan tres redes sociales diferentes, de lo cuales se obtuvieron distintos resultados llamativos, que se describen a continuación.

Para redes dispersas, a mayor profundidad, mayor es la cantidad de nodos influenciados. Para este tipo de redes, un nivel de profundidad 6 tiene un gran impacto en el resultado final, habiendo influenciado a casi toda la red. Sin embargo, el comportamiento de estas redes parece estar más asociado con las probabilidades de las vecindades que con la profundidad. Esto último ya que en los resultados se obtiene un comportamiento de nodos influenciados similar si se comparan profundidades 1, 4 y 6 con probabilidades de vecindad del 50 %, 75 % y 100 %. Entonces si bien la profundidad 6 es la que genera una mayor

dispersión de influencia, variando las probabilidades de vecindad, podemos obtener comportamiento similar con menor profundidad, esto podría ser útil para un estudio distinto al de dispersión de influencia como el de comportamiento de redes.

En cambio en el caso de utilizar redes densas, se nota que la profundidad es el parámetro más importante para conocer el comportamiento de la red como la dispersión de influencia de la red. Utilizando el valor de profundidad 4, la red alcanza una muy alta dispersión de influencia siendo muy levemente superado por profundidad 6. Se puede notar además que otros parámetros influyentes en redes dispersas, como la probabilidad de vecindad e incluso la dirección, no afectan tan significativamente los comportamientos y patrones a los que obedece la red densa.

Por lo tanto, para las redes dispersas la importancia de variar los parámetros es alta. En cambio cuando es una red densa, es importante enfocarse en cuál es la profundidad idónea considerando el gasto de recursos y la dispersión que se desea alcanzar.

Las principales dificultades dentro de este trabajo se presentaron al momento de aplicar los modelos de dispersión de influencia. Esto debido a que estos modelos son muy versátiles y pueden aplicarse en muchas áreas, por lo tanto también existen muchas versiones de estas en Internet pero no todas se ajustan a lo requerido para este trabajo. Por lo tanto, se procedió a implementar una versión adecuada al contexto del trabajo.

Ya que la finalidad de este trabajo era principalmente exploratoria, surgen algunas otras ideas, mejoras y proyecciones de lo que podría llegar a ser finalmente quizá como una aplicación web u otro tipo de proyecto no tan relacionado con la investigación.

La plataforma de experimentación está programada para que trabaje de manera serial que en trabajos futuros podría ser paralelizada. Esto reduciría considerablemente el tiempo de demora en las pruebas.

Ya que la plataforma se conforma de archivos separados, otra mejora sería compactar todo en un mismo archivo y automatizar todas las etapas desde el moldeado de las redes a aplicar GISR, luego las estadísticas y cálculos de correlación.

También sería muy útil llevar esta plataforma a una aplicación web, ya que de esta manera sería más sencilla de utilizar gracias al aporte que haría la interfaz gráfica de la aplicación.

# Bibliografía

- [1] ITU, “Measuring Digital Development: Facts and Figures 2021,” [https://www.itu.int/pub/D-IND-ICT\\_MDD-2021](https://www.itu.int/pub/D-IND-ICT_MDD-2021), 2021, [Accedido en 23 Enero 2022].
- [2] Z. Dentzel, “El impacto de internet en la vida diaria,” <https://www.bbvaopenmind.com/articulos/el-impacto-de-internet-en-la-vida-diaria/>, 2013, [Accedido en 23 Enero 2022].
- [3] Hootsuite, “digital 2021 the latest insights into how people around the world use the internet, social media, mobile devices, and ecommerce global overview report,” [https://hootsuite.widen.net/s/zcdrtxwczn/digital2021\\_globalreport\\_en](https://hootsuite.widen.net/s/zcdrtxwczn/digital2021_globalreport_en), 2021, [Accedido en 13 Abril 2021].
- [4] D. J. Watts, *Six degrees: The science of a connected age*. WW Norton & Company, 2004.
- [5] E. Y. Daraghmi and S.-M. Yuan, “We are so close, less than 4 degrees separating you and me!” *Computers in Human Behavior*, vol. 30, pp. 273–285, 2014.
- [6] J. Sun and J. Tang, *A Survey of Models and Algorithms for Social Influence Analysis*. Boston, MA: Springer US, 2011, pp. 177–214.
- [7] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.
- [8] C. Ni, C. Sugimoto, and J. Jiang, “Degree, closeness, and betweenness: Application of group centrality measurements to explore macro-disciplinary evolution diachronically,” in *Proceedings of ISSI*, 2011, pp. 1–13.
- [9] F. Riquelme and P. González-Cantergiani, “Measuring user influence on twitter: A survey,” *Information Processing & Management*, vol. 52, no. 5, pp. 949–975, Sep. 2016.
- [10] D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137–146.

- [11] M. Granovetter, “Threshold models of collective behavior,” *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, May 1978.
- [12] J. Goldenberg, B. Libai, and E. Muller, “Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata,” *Academy of Marketing Science Review*, vol. 9, no. 3, pp. 1–18, 2001.
- [13] F. Riquelme, P. Gonzalez-Cantergiani, X. Molinero, and M. Serna, “Centrality measure in social networks based on linear threshold model,” *Knowledge-Based Systems*, vol. 140, pp. 92–102, 2018.
- [14] S. Kundu, C. Murthy, and S. K. Pal, “A new centrality measure for influence maximization in social networks,” in *International Conference on Pattern Recognition and Machine Intelligence*. Springer, 2011, pp. 242–247.
- [15] K. Das, S. Samanta, and M. Pal, “Study on centrality measures in social networks: a survey,” *Social network analysis and mining*, vol. 8, no. 1, pp. 1–11, 2018.
- [16] T. L. Griffiths, M. Steyvers, and A. Firl, “Google and the mind: Predicting fluency with pagerank,” *Psychological science*, vol. 18, no. 12, pp. 1069–1076, 2007.
- [17] B. Ruhnau, “Eigenvector-centrality—a node-centrality?” *Social networks*, vol. 22, no. 4, pp. 357–365, 2000.
- [18] P. Bonacich, “Some unique properties of eigenvector centrality,” *Social networks*, vol. 29, no. 4, pp. 555–564, 2007.
- [19] Y. Yujie, “A survey on information diffusion in online social networks,” in *Proceedings of the 2020 European Symposium on Software Engineering*, 2020, pp. 181–186.
- [20] H. Zhang, T. N. Dinh, and M. T. Thai, “Maximizing the spread of positive influence in online social networks,” in *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 2013, pp. 317–326.
- [21] M. Morris, “Epidemiology and social networks: Modeling structured diffusion,” *Sociological methods & research*, vol. 22, no. 1, pp. 99–126, 1993.
- [22] Z. Song, “Economic growth and carbon emissions: Estimation of a panel threshold model for the transition process in china,” *Journal of Cleaner Production*, vol. 278, p. 123773, 2021.
- [23] S. Tang and W. Li, “An epidemic model with adaptive virus spread control for wireless sensor networks,” *International Journal of Security and Networks*, vol. 6, no. 4, p. 201, 2011.

- [24] S. Funk, M. Salathé, and V. A. Jansen, “Modelling the influence of human behaviour on the spread of infectious diseases: a review,” *Journal of the Royal Society Interface*, vol. 7, no. 50, pp. 1247–1256, 2010.
- [25] W. Chen, Y. Yuan, and L. Zhang, “Scalable influence maximization in social networks under the linear threshold model,” in *2010 IEEE International Conference on Data Mining*, 2010, pp. 88–97.
- [26] X. He, G. Song, W. Chen, and Q. Jiang, “Influence blocking maximization in social networks under the competitive linear threshold model,” in *Proceedings of the 2012 siam international conference on data mining*. SIAM, 2012, pp. 463–474.
- [27] M. Kimura, K. Saito, and H. Motoda, “Solving the contamination minimization problem on networks for the linear threshold model,” in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2008, pp. 977–984.
- [28] K. Saito, R. Nakano, and M. Kimura, “Prediction of information diffusion probabilities for independent cascade model,” in *International conference on knowledge-based and intelligent information and engineering systems*. Springer, 2008, pp. 67–75.
- [29] J. Kim, W. Lee, and H. Yu, “Ct-ic: Continuously activated and time-restricted independent cascade model for viral marketing,” *Knowledge-Based Systems*, vol. 62, pp. 57–68, 2014.
- [30] F. Lu, W. Zhang, L. Shao, X. Jiang, P. Xu, and H. Jin, “Scalable influence maximization under independent cascade model,” *Journal of Network and Computer Applications*, vol. 86, pp. 15–23, 2017.
- [31] A. Nazemian and F. Taghiyareh, “Influence maximization in independent cascade model with positive and negative word of mouth,” in *6th International Symposium on Telecommunications (IST)*. IEEE, 2012, pp. 854–860.
- [32] S. Atchariyawut, R. Jiraratananon, and R. Wang, “Mass transfer study and modeling of gas–liquid membrane contacting process by multistage cascade model for co2 absorption,” *Separation and purification technology*, vol. 63, no. 1, pp. 15–22, 2008.
- [33] T. Huo, R. Cao, H. Du, J. Zhang, W. Cai, and B. Liu, “Nonlinear influence of urbanization on china’s urban residential building carbon emissions: New evidence from panel threshold model,” *Science of The Total Environment*, vol. 772, p. 145058, 2021.
- [34] Q. Wang, Y. Jin, T. Yang, and S. Cheng, “An emotion-based independent cascade model for sentiment spreading,” *Knowledge-Based Systems*, vol. 116, pp. 86–93, 2017.

- [35] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou, “K-center: An approach on the multi-source identification of information diffusion,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2616–2626, 2015.
- [36] B. A. Prakash, J. Vreeken, and C. Faloutsos, “Spotting culprits in epidemics: How many and which ones?” in *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 11–20.
- [37] R. Paluch, X. Lu, K. Suchecki, B. K. Szymański, and J. A. Hołyst, “Fast and accurate detection of spread source in large complex networks,” *Scientific Reports*, vol. 8, no. 1, Feb. 2018.
- [38] Z. Chen, K. Zhu, and L. Ying, “Detecting multiple information sources in networks under the sir model,” *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 1, pp. 17–31, 2016.
- [39] K. Zhu and L. Ying, “Information source detection in the sir model: A sample-path-based approach,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 408–421, 2014.
- [40] I. Cooper, A. Mondal, and C. G. Antonopoulos, “A sir model assumption for the spread of covid-19 in different communities,” *Chaos, Solitons ‘I&’ Fractals*, vol. 139, p. 110057, 2020.
- [41] R. Parshani, S. Carmi, and S. Havlin, “Epidemic threshold for the susceptible-infectious-susceptible model on random networks,” *Physical review letters*, vol. 104, no. 25, p. 258701, 2010.
- [42] D. Bichara, A. Iggidr, and G. Sallet, “Global analysis of multi-strains sis, sir and msir epidemic models,” *Journal of Applied Mathematics and Computing*, vol. 44, no. 1, pp. 273–292, 2014.
- [43] M. Y. Li and J. S. Muldowney, “Global stability for the seir model in epidemiology,” *Mathematical biosciences*, vol. 125, no. 2, pp. 155–164, 1995.
- [44] M. López, A. Peinado, and A. Ortiz, “A seis model for propagation of random jamming attacks in wireless sensor networks,” in *International Joint Conference SOCO’16-CISIS’16-ICEUTE’16*. Springer, 2016, pp. 668–677.
- [45] F. Riquelme, P. Gonzalez-Cantergiani, X. Molinero, and M. Serna, “The neighborhood role in the linear threshold rank on social networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 528, p. 121430, 2019.

- [46] L. Iacovissi, “Linear threshold rank on random social networks,” B.S. thesis, Università degli Studi di Padova, 2020.
- [47] P. García Rodríguez, “Estudio experimental del forward linear threshold rank,” B.S. thesis, Universitat Politècnica de Catalunya, 2019.
- [48] A. Domínguez Besserer, “Estudio de mecanismos de ponderación de influencia y su efecto en el forward linear threshold rank,” B.S. thesis, Universitat Politècnica de Catalunya, 2020.
- [49] I. Gaye, G. Mendy, S. Ouya, and D. Seck, “New centrality measure in social networks based on independent cascade (ic) model,” in *2015 3rd International Conference on Future Internet of Things and Cloud*. IEEE, 2015, pp. 675–680.
- [50] I. Gaye, G. Mendy, S. Ouya, I. Diop, and D. Seck, “Multi-diffusion degree centrality measure to maximize the influence spread in the multilayer social networks,” in *International Conference on e-Infrastructure and e-Services for Developing Countries*. Springer, 2016, pp. 53–65.
- [51] D. Gruhl, D. Liben-Nowell, R. V. Guha, and A. Tomkins, “Information diffusion through blogspace,” *SIGKDD Explorations*, vol. 6, no. 2, pp. 43–52, 2004.
- [52] P. M. Domingos and M. Richardson, “Mining the network value of customers,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, D. Lee, M. Schkolnick, F. J. Provost, and R. Srikant, Eds. ACM, 2001, pp. 57–66.
- [53] L. A. Adamic and E. Adar, “How to search a social network,” *Social Networks*, vol. 27, no. 3, pp. 187–203, 2005.
- [54] X. Song, B. L. Tseng, C. Lin, and M. Sun, “Personalized recommendation driven by information flow,” in *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, Eds. ACM, 2006, pp. 509–516.
- [55] X. Zhang, J. Zhu, Q. Wang, and H. Zhao, “Identifying influential nodes in complex networks with community structure,” *Knowl.-Based Syst.*, vol. 42, pp. 74–84, 2013.
- [56] J. Li, W. Peng, T. Li, T. Sun, Q. Li, and J. Xu, “Social network user influence sense-making and dynamics prediction,” *Expert Syst. Appl.*, vol. 41, no. 11, pp. 5115–5124, 2014.

- [57] F. Morone and H. A. Makse, “Influence maximization in complex networks through optimal percolation,” *Nature*, vol. 524, pp. 64–68, 2015.
- [58] R. Noël, F. Riquelme, R. Mac Lean, E. Merino, C. Cechinel, T. S. Barcelos, R. Villarroel, and R. Muñoz, “Exploring collaborative writing of user stories with multimodal learning analytics: A case study on a software engineering course,” *IEEE Access*, vol. 6, pp. 67 783–67 798, 2018.
- [59] R. Hernández, C. Fernández, and L. Baptista, *Metodología De La Investigación*. McGraw-Hill Interamericana, 2014.
- [60] A. Davis, “Operational prototyping: a new development approach,” *IEEE Software*, vol. 9, no. 5, pp. 70–78, Sep. 1992. [Online]. Available: <https://doi.org/10.1109/52.156899>
- [61] R. Artusi, P. Verderio, and E. Marubini, “Bravais-pearson and spearman correlation coefficients: meaning, test of hypothesis and confidence interval,” *The International journal of biological markers*, vol. 17, no. 2, pp. 148–151, 2002.
- [62] Jupyter, “Project Jupyter,” <https://jupyter.org>, 2021, [Accedido en 17 Julio 2021].
- [63] GitHub, “GitHub,” <https://github.com>, 2021, [Accedido en 15 Octubre 2021].
- [64] Microsoft, “Visual Studio Code,” <https://code.visualstudio.com>, 2021, [Accedido en 15 Octubre 2021].
- [65] Spyder, “Spyder,” <https://www.spyder-ide.org>, 2021, [Accedido en 15 Octubre 2021].
- [66] Trello, “Trello,” <https://trello.com/es>, 2021, [Accedido en 15 Octubre 2021].
- [67] Google, “Google Meet,” <https://meet.google.com>, 2021, [Accedido en 15 Octubre 2021].
- [68] EduVPN, “EduVPN,” <https://www.eduvpn.org>, 2021, [Accedido en 15 Octubre 2021].
- [69] NetworkX, “Network Analisis in Python,” <https://networkx.org>, 2021, [Accedido en 17 Julio 2021].
- [70] Matplotlib, “Matplotlib: Visualization with Python,” <https://matplotlib.org/stable/index.html>, 2021, [Accedido en 17 Julio 2021].
- [71] Numpy, “Numpy,” <https://numpy.org/doc/stable/index.html#>, 2021, [Accedido en 15 Octubre 2021].

- [72] R. A. Rossi and N. K. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *AAAI*, 2015. [Online]. Available: <http://networkrepository.com>

# **Apéndice A**

## **Matrices de análisis**

### **Matrices de Modelo IC**

En esta sección de Apéndice se encuentran las tablas referentes a las pruebas de las redes utilizando el modelo de dispersión de influencia IC-Model.

prof	dir	prob vec	min xi	max xi	min fx	max fx	prom xi	prom fx	mediana xi	mediana fx	desv est xi	desv est fx	cant dist xi	cant val dist fx	t eje
0	0	0	1	1	1	9	1	3.31	1	2	0.0	2.55	1	8	0
0	1	0	1	1	1	12	1	3.6	1	2	0.0	2.86	1	10	0,01
0	2	0	1	1	1	9	1	3.31	1	2	0.0	2.61	1	9	0
1	0	0.25	1	8	1	20	3.23	7.6	3	8	1.73	3.93	7	14	0
1	0	0.5	1	8	1	19	3.23	7.2	3	7	1.73	4.01	7	13	0
1	0	0.75	2	20	2	28	7.74	12.14	7	11	4.71	6.35	14	19	0
1	0	1	2	20	2	29	7.74	12.17	7	11	4.71	6.32	14	18	0,01
1	1	0.25	1	5	1	10	2.11	4.17	2	4	1.12	2.85	5	10	0
1	1	0.5	1	5	1	10	2.11	4.4	2	4	1.12	2.92	5	10	0
1	1	0.75	1	11	1	13	4.37	5.34	4	6	2.89	3.47	10	12	0
1	1	1	1	11	1	13	4.37	5.37	4	6	2.89	3.48	10	12	0
1	2	0.25	1	7	1	18	2.11	7.11	1	7	1.77	4.26	7	15	0
1	2	0.5	1	7	1	20	2.11	7.14	1	7	1.77	4.76	7	13	0
1	2	0.75	1	19	2	30	4.37	10.6	2	9	5.12	7.35	11	19	0
1	2	1	1	19	1	30	4.37	10.86	2	8	5.12	7.53	11	18	0
4	0	0.25	1	12	1	24	4.14	8.74	4	8	2.6	5.73	10	14	0,01
4	0	0.5	1	12	1	24	4.14	8.63	4	8	2.6	4.8	10	14	0
4	0	0.75	2	26	2	34	10.31	13.69	9	11	6.62	8.67	16	18	0
4	0	1	2	26	2	34	10.31	13.54	9	11	6.62	8.47	16	17	0,01
4	1	0.25	1	6	1	10	2.51	4.46	2	5	1.46	2.89	6	10	0
4	1	0.5	1	6	1	9	2.51	4.37	2	4	1.46	2.84	6	9	0
4	1	0.75	1	13	1	13	5.66	5.66	6	6	3.59	3.59	13	13	0,01
4	1	1	1	13	1	13	5.66	5.66	6	6	3.59	3.59	13	13	0
4	2	0.25	1	12	1	24	2.63	7.89	1	7	2.87	5.81	7	14	0
4	2	0.5	1	12	1	24	2.63	7.83	1	7	2.87	5.58	7	15	0
4	2	0.75	1	26	1	34	5.66	12.29	2	9	7.69	9.55	12	20	0,01
4	2	1	1	26	1	33	5.66	11.74	2	8	7.69	9.73	12	18	0
6	0	0.25	1	12	1	24	4.14	8.77	4	8	2.6	5.53	10	15	0
6	0	0.5	1	12	1	24	4.14	8.26	4	7	2.6	5.4	10	17	0,01
6	0	0.75	2	26	2	34	10.31	13.71	9	11	6.62	8.59	16	19	0
6	0	1	2	26	2	34	10.31	13.8	9	11	6.62	8.72	16	17	0,01
6	1	0.25	1	6	1	12	2.51	4.63	2	5	1.46	3.02	6	10	0
6	1	0.5	1	6	1	12	2.51	4.89	2	5	1.46	3.24	6	10	0
6	1	0.75	1	13	1	13	5.66	5.66	6	6	3.59	3.59	13	13	0
6	1	1	1	13	1	13	5.66	5.66	6	6	3.59	3.59	13	13	0,01
6	2	0.25	1	12	1	24	2.63	7.71	1	6	2.87	5.77	7	17	0
6	2	0.5	1	12	1	25	2.63	7.77	1	7	2.87	6.06	7	16	0
6	2	0.75	1	26	1	34	5.66	12.74	2	10	7.69	9.43	12	20	0
6	2	1	1	26	1	34	5.66	12.06	2	8	7.69	9.99	12	21	0

Tabla A.1: Matriz de resultados para red Football en modelo IC.

prof	dir	prob vec	min xi	max xi	min fx	max fx	prom xi	prom fx	mediana xi	mediana fx	desv est xi	desv est fx	cant dist xi	cant val dist fx	tejec
0	0	0	1	1	1	2552	1	1012.25	1	2	0.0	1220.4	1	116	471,36
0	1	0	1	1	1	2554	1	1031.25	1	2	0.0	1223.82	1	122	508,34
0	2	0	1	1	1	2552	1	1034.08	1	2	0.0	1224.52	1	121	514,79
1	0	0.25	1	111	1	2556	2.74	1470.16	2	2466	4.65	1221.55	43	130	679,07
1	0	0.5	1	278	1	2557	5.03	1903.28	2	2479	10.77	1053.18	78	134	876,09
1	0	0.75	1	346	1	2584	6.08	2094.9	3	2483	13.5	906.17	90	137	959,06
1	0	1	2	512	2	2636	8.47	2395.86	3	2487	20.05	467.62	113	144	1099,23
1	1	0.25	1	53	1	2545	1.92	1276.51	1	2447	2.47	1242.0	28	124	589,17
1	1	0.5	1	145	1	2558	3.38	1638.44	2	2472	6.13	1178.14	56	123	750,47
1	1	0.75	1	195	1	2550	4.21	1836.78	2	2477	8.23	1091.81	63	122	846,75
1	1	1	1	399	1	2564	7.39	2378.82	3	2486	16.41	506.03	103	138	1071,78
1	2	0.25	1	64	1	2549	1.92	1256.03	1	2442	2.69	1242.89	32	116	570,12
1	2	0.5	1	176	1	2556	3.38	1540.56	2	2470	6.76	1206.52	57	130	697,21
1	2	0.75	1	247	1	2551	4.21	1677.68	2	2474	9.15	1164.09	64	126	761,10
1	2	1	1	491	1	2635	7.39	2104.78	3	2483	18.29	896.78	107	135	950,04
4	0	0.25	1	960	1	2826	739.37	2680.68	841	2720	219.59	183.13	652	352	3552,95
4	0	0.5	1	2168	1	3180	1713.35	2961.84	1936	3045	481.16	236.94	1032	637	3889,80
4	0	0.75	1	2730	1	3356	2174.64	3101.4	2449	3208	588.46	280.62	1135	749	3914,94
4	0	1	2	3766	2	3771	3193.27	3460.01	3473	3603	690.14	346.86	1150	946	3441,23
4	1	0.25	1	506	1	2664	405.85	2560.17	441	2590	95.09	252.93	372	191	2274,89
4	1	0.5	1	1230	1	2834	1018.63	2690.04	1109	2737	235.61	269.29	671	339	2621,83
4	1	0.75	1	1634	1	2905	1363.88	2762.77	1484	2818	312.97	280.21	774	398	2684,38
4	1	1	1	3244	1	3253	2732.61	3048.03	2968	3137	627.96	337.26	1052	649	2269,07
4	2	0.25	1	548	1	2727	382.54	2302.44	469	2631	180.4	863.61	411	255	2204,06
4	2	0.5	1	1410	1	3022	1011.21	2506.22	1249	2893	478.01	952.83	715	503	2677,31
4	2	0.75	1	1947	1	3195	1390.29	2637.43	1720	3055	656.8	1001.34	852	586	2784,37
4	2	1	1	3742	1	3762	2732.61	3062.37	3376	3575	1287.21	1170.49	1046	861	2368,90
6	0	0.25	1	978	1	2845	869.41	2739.44	935	2773	166.99	178.01	160	238	4727,31
6	0	0.5	1	2202	1	3200	2014.12	3094.9	2154	3153	347.65	209.72	212	210	5171,10
6	0	0.75	1	2761	1	3371	2550.26	3267.44	2714	3340	408.09	237.35	230	196	5251,20
6	0	1	2	3771	2	3775	3682.04	3688.05	3769	3771	278.17	261.99	146	120	4573,14
6	1	0.25	1	512	1	2684	477.75	2591.51	482	2615	47.45	251.0	73	121	2883,35
6	1	0.5	1	1251	1	2844	1200.42	2765.04	1212	2791	117.91	267.58	111	113	3304,21
6	1	0.75	1	1665	1	2939	1603.26	2857.45	1620	2885	157.19	276.47	124	102	3379,33
6	1	1	1	3261	1	3261	3219.12	3224.03	3256	3257	314.72	311.96	116	84	2847,18
6	2	0.25	1	555	1	2730	448.26	2338.34	518	2665	180.08	874.17	87	167	2794,60
6	2	0.5	1	1440	1	3026	1198.54	2603.79	1392	2976	481.66	977.42	117	161	3093,50
6	2	0.75	1	1964	1	3204	1646.56	2766.91	1913	3160	661.77	1030.42	132	163	3207,65
6	2	1	1	3749	1	3771	3219.12	3282.03	3746	3762	1293.8	1228.06	107	147	2695,80

Tabla A.2: Matriz de resultados para red Bitcoin Alpha en modelo IC.

**Matrices de Modelo LT**

En esta sección de Apéndice, se encuentran las tablas referentes a las pruebas de las redes utilizando el modelo de dispersión de influencia LT-Model.

prof	dir	prob vec	min xi	max xi	min fx	max fx	prom xi	prom fx	mediana xi	mediana fx	desv est xi	desv est fx	cant dist xi	cant val dist fx	t ejec
0	0	0	1	1	1	1207	1	1.83	1.0	1.0	0.0	11.67	1	113	2081,84
0	1	0	1	1	1	1096	1	1.82	1.0	1.0	0.0	11.36	1	112	2005,20
0	2	0	1	1	1	994	1	1.87	1.0	1.0	0.0	12.51	1	123	2013,56
1	0	0.25	1	168	1	1275	1.23	12.26	1.0	1.0	1.33	94.64	21	502	1998,03
1	0	0.5	1	295	1	1303	1.42	19.95	1.0	1.0	2.28	123.14	28	658	1986,95
1	0	0.75	1	1189	1	2064	2.5	72.86	2.0	2.0	8.77	239.81	59	1046	2041,99
1	0	1	1	1208	1	1994	2.52	73.79	2.0	2.0	8.92	241.05	61	1035	2039,25
1	1	0.25	1	168	1	1296	1.12	2.16	1.0	1.0	1.29	15.62	21	140	1987,89
1	1	0.5	1	295	1	1314	1.21	2.27	1.0	1.0	2.23	16.57	28	150	1983,07
1	1	0.75	1	1188	1	1710	1.82	3.47	1.0	1.0	8.76	24.85	58	231	2001,03
1	1	1	1	1207	1	1528	1.83	3.44	1.0	1.0	8.9	24.9	60	221	1987,31
1	2	0.25	1	10	1	1247	1.12	11.97	1.0	1.0	0.34	93.67	8	481	2002,55
1	2	0.5	1	15	1	1289	1.21	19.79	1.0	1.0	0.46	123.19	10	645	2007,94
1	2	0.75	1	36	1	1951	1.82	72.15	2.0	2.0	0.9	240.09	25	1044	2048,54
1	2	1	1	36	1	1888	1.83	73.43	2.0	2.0	0.9	241.87	25	1058	2041,77
4	0	0.25	1	308	1	2131	2.0	39.2	1.0	1.0	5.74	183.88	90	979	2025,41
4	0	0.5	1	547	1	2395	2.83	62.15	1.0	2.0	10.42	236.46	132	1155	2042,72
4	0	0.75	1	2036	1	3146	8.02	135.82	2.0	3.0	40.82	375.57	299	1684	2107,44
4	0	1	1	2073	1	3266	8.11	137.13	2.0	3.0	41.48	378.15	300	1674	2114,96
4	1	0.25	1	308	1	1396	1.49	5.04	1.0	1.0	4.91	39.51	70	346	2041,93
4	1	0.5	1	547	1	1566	1.94	5.8	1.0	1.0	9.16	42.92	105	377	2008,76
4	1	0.75	1	2036	1	2436	4.66	9.02	1.0	1.0	36.93	67.95	228	504	2002,94
4	1	1	1	2073	1	2460	4.72	9.17	1.0	1.0	37.55	68.59	235	510	2006,07
4	2	0.25	1	57	1	2185	1.52	36.91	1.0	1.0	2.18	180.45	42	910	2010,24
4	2	0.5	1	86	1	2264	1.91	59.43	1.0	2.0	3.5	234.04	61	1135	2031,23
4	2	0.75	1	312	1	2931	4.67	131.11	2.0	2.0	13.51	370.22	178	1620	2097,60
4	2	1	1	319	1	3021	4.72	132.47	2.0	2.0	13.69	372.81	176	1623	2100,87
6	0	0.25	1	382	1	2413	3.37	62.52	1.0	1.0	12.97	251.52	178	1301	2077,89
6	0	0.5	1	700	1	2627	5.38	90.92	1.0	2.0	23.75	317.94	266	1595	2114,03
6	0	0.75	1	2716	1	4041	17.57	184.67	2.0	3.0	89.54	522.33	575	2180	2201,76
6	0	1	1	2757	1	3948	17.8	186.14	2.0	3.0	90.96	524.63	578	2179	2175,17
6	1	0.25	1	379	1	1539	2.15	8.63	1.0	1.0	10.18	66.44	136	496	2006,97
6	1	0.5	1	693	1	1734	3.23	9.91	1.0	1.0	19.43	73.36	202	530	2015,00
6	1	0.75	1	2698	1	2890	9.58	16.01	1.0	1.0	75.5	119.16	393	638	2021,43
6	1	1	1	2739	1	2890	9.72	16.01	1.0	1.0	76.76	119.94	411	647	2009,48
6	2	0.25	1	85	1	2224	2.24	57.81	1.0	1.0	5.86	244.4	71	1243	2041,68
6	2	0.5	1	141	1	2484	3.19	85.54	1.0	2.0	9.7	310.3	115	1467	2073,48
6	2	0.75	1	554	1	3482	9.62	174.72	2.0	2.0	38.09	506.87	340	2049	2191,05
6	2	1	1	561	1	3538	9.72	176.03	2.0	2.0	38.58	509.05	348	2032	2171,57

Tabla A.3: Matriz de resultados para red Higgs en modelo IC.

prof	dir	prob vec	min xi	max xi	min fx	max fx	prom xi	prom fx	mediana xi	mediana fx	desv est xi	desv est fx	cant dist xi	cant val dist fx	t eje
0	0	0	1	1	1	2	1	1.06	1	1	0.0	0.23	1	2	0.0
0	1	0	1	1	1	2	1	1.06	1	1	0.0	0.23	1	2	0.0
0	2	0	1	1	1	2	1	1.06	1	1	0.0	0.23	1	2	0.0
1	0	0.25	1	7	1	9	2.77	2.91	2	2	1.55	1.83	7	8	0.0
1	0	0.5	1	9	1	11	3.86	4.09	3	4	2.06	2.47	9	10	0.01
1	0	0.75	2	16	2	23	6.14	6.66	5	5	3.4	4.39	11	12	0.0
1	0	1	2	18	2	25	7.29	8.43	7	7	4.1	5.96	13	17	0.0
1	1	0.25	1	5	1	5	1.89	1.91	2	2	1.06	1.08	5	5	0.01
1	1	0.5	1	7	1	7	2.43	2.46	2	2	1.55	1.57	7	7	0.0
1	1	0.75	1	9	1	9	3.57	3.6	3	3	2.19	2.21	8	8	0.0
1	1	1	1	10	1	10	4.14	4.14	4	4	2.76	2.76	9	9	0.0
1	2	0.25	1	7	1	9	1.89	2.06	1	1	1.53	1.82	7	7	0.0
1	2	0.5	1	8	1	10	2.43	2.69	1	2	2.13	2.57	7	8	0.01
1	2	0.75	1	15	1	23	3.57	4.23	2	2	3.82	5.07	9	10	0.0
1	2	1	1	16	1	25	4.14	5.51	2	3	4.62	6.92	11	12	0.01
4	0	0.25	1	7	1	9	3.09	3.23	3	3	1.65	1.93	7	8	0.0
4	0	0.5	1	9	1	11	4.17	4.4	4	4	2.08	2.49	9	10	0.01
4	0	0.75	2	17	2	23	7.06	7.54	7	7	3.33	4.22	13	13	0.0
4	0	1	2	18	2	25	8.34	9.46	8	8	4.04	5.7	15	17	0.015
4	1	0.25	1	5	1	6	2.2	2.23	2	2	1.39	1.46	4	5	0.0
4	1	0.5	1	7	1	7	2.74	2.77	2	2	1.83	1.88	7	7	0.0
4	1	0.75	1	10	1	10	4.49	4.51	4	4	2.73	2.77	10	10	0.0
4	1	1	1	12	1	12	5.2	5.2	5	5	3.45	3.45	12	12	0.01
4	2	0.25	1	11	1	12	2.4	2.66	1	1	2.55	2.95	7	9	0.0
4	2	0.5	1	12	1	14	3	3.43	1	2	3.29	4.01	10	9	0.0
4	2	0.75	1	21	1	27	4.51	5.6	2	2	5.79	7.75	11	12	0.01
4	2	1	1	24	1	31	5.26	6.94	2	3	6.83	9.37	12	13	0.01
6	0	0.25	1	7	1	9	3.09	3.23	3	3	1.65	1.93	7	8	0.01
6	0	0.5	1	9	1	11	4.17	4.4	4	4	2.08	2.49	9	10	0.0
6	0	0.75	2	17	2	23	7.06	7.54	7	7	3.33	4.22	13	13	0.01
6	0	1	2	18	2	25	8.34	9.46	8	8	4.04	5.7	15	17	0.03
6	1	0.25	1	5	1	6	2.2	2.23	2	2	1.39	1.46	4	5	0.0
6	1	0.5	1	7	1	7	2.74	2.77	2	2	1.83	1.88	7	7	0.0
6	1	0.75	1	10	1	10	4.49	4.51	4	4	2.73	2.77	10	10	0.01
6	1	1	1	12	1	12	5.2	5.2	5	5	3.45	3.45	12	12	0.0
6	2	0.25	1	11	1	12	2.4	2.66	1	1	2.55	2.95	7	9	0.0
6	2	0.5	1	12	1	14	3	3.43	1	2	3.29	4.01	10	9	0.01
6	2	0.75	1	21	1	27	4.51	5.6	2	2	5.79	7.75	11	12	0.01
6	2	1	1	24	1	31	5.26	6.94	2	3	6.83	9.37	12	13	0.01

Tabla A.4: Matriz de resultados para red Football en modelo LT.

prof	dir	prob vec	min xi	max xi	min fx	max fx	prom xi	prom fx	mediana xi	mediana fx	desv est xi	desv est fx	cant dist xi	cant val dist fx	t ejec
0	0	0	1	1	1	92	1	1.35	1	1	0.0	2.02	1	18	20,26
0	1	0	1	1	1	92	1	1.35	1	1	0.0	2.02	1	18	20,91
0	2	0	1	1	1	92	1	1.35	1	1	0.0	2.02	1	18	19,55
1	0	0.25	1	194	1	382	3.85	17.37	2	4	7.81	34.65	62	171	33,66
1	0	0.5	1	318	1	710	5.7	28.59	2	9	12.67	54.1	84	229	55,70
1	0	0.75	1	461	1	2355	7.62	41.77	3	14	17.86	88.42	107	288	92,86
1	0	1	1	505	1	2814	8.35	46.56	3	15	19.75	101.07	113	302	114,58
1	1	0.25	1	94	1	242	2.6	10.68	2	2	4.21	23.31	39	124	27,13
1	1	0.5	1	167	1	332	3.9	18.51	2	5	7.55	35.26	65	169	34,83
1	1	0.75	1	292	1	683	5.79	30.99	3	10	12.38	55.29	82	231	56,51
1	1	1	1	385	1	1717	7.07	39.75	3	13	15.64	75.69	94	270	72,81
1	2	0.25	1	127	1	261	2.6	9.71	1	2	4.74	22.17	45	123	24,36
1	2	0.5	1	222	1	438	3.9	16.96	2	3	8.4	35.43	64	169	31,20
1	2	0.75	1	379	1	1658	5.79	28.44	2	8	13.86	61.7	84	235	52,92
1	2	1	1	466	1	2423	7.07	37.19	3	11	17.44	85.51	103	275	85,50
4	0	0.25	1	905	1	3102	693.59	1011.91	755	1056	162.57	225.34	540	714	5764,84
4	0	0.5	1	1581	1	3196	1247.93	2980.21	1362	3106	287.66	520.35	768	455	26052,57
4	0	0.75	1	2526	1	3300	2042.94	3087.69	2224	3171	468.04	425.84	956	393	13641,17
4	0	1	1	3164	1	3328	2594.18	3122.69	2817	3201	595.89	403.82	1061	364	8539,01
4	1	0.25	1	850	1	1429	692.39	1000.01	754	1050	161.86	204.54	538	677	5403,48
4	1	0.5	1	1514	1	3190	1246.3	2979.98	1361	3105	286.7	520.47	754	442	25710,26
4	1	0.75	1	2444	1	3240	2041.59	3087.42	2224	3170	467.3	426.24	954	376	12213,42
4	1	1	1	3084	1	3255	2593.66	3122.37	2817	3201	595.64	404.31	1061	345	7628,91
4	2	0.25	1	960	1	3204	680.39	977.83	840	1174	319.55	445.44	564	732	6048,89
4	2	0.5	1	1752	1	3452	1247.13	2766.83	1546	3322	587.7	1204.55	807	546	19755,05
4	2	0.75	1	2826	1	3642	2048.03	2966.48	2535	3525	963.87	1252.52	1005	547	11774,67
4	2	1	1	3564	1	3749	2601.02	3067.95	3217	3644	1225.49	1284.29	1051	540	7262,27
6	0	0.25	1	906	1	3108	820.02	1143.95	828	1141	81.17	152.49	108	371	6668,31
6	0	0.5	1	1586	1	3214	1471.22	3155.54	1489	3187	145.45	304.88	154	96	22575,28
6	0	0.75	1	2538	1	3300	2404.26	3207.5	2430	3238	235.58	309.32	154	70	11514,88
6	0	1	1	3179	1	3328	3056.57	3224.38	3090	3255	298.8	310.85	146	68	6295,75
6	1	0.25	1	873	1	1498	818.83	1132.04	827	1132	80.98	129.3	100	349	6098,07
6	1	0.5	1	1556	1	3198	1469.59	3155.36	1488	3187	145.23	305.07	139	83	22552,98
6	1	0.75	1	2484	1	3244	2402.91	3207.24	2428	3238	235.47	309.92	148	51	11241,32
6	1	1	1	3116	1	3258	3056.06	3224.07	3090	3255	298.84	311.53	138	44	6002,93
6	2	0.25	1	975	1	3255	801.25	1119.5	928	1278	321.98	457.43	115	377	7944,20
6	2	0.5	1	1766	1	3468	1472.44	2956.38	1708	3427	591.79	1186.52	142	102	21677,77
6	2	0.75	1	2843	1	3648	2409.31	3131.97	2799	3637	968.28	1256.97	142	65	13566,43
6	2	1	1	3586	1	3753	3063.82	3226.16	3563	3748	1231.4	1294.77	129	62	6446,45

Tabla A.5: Matriz de resultados para red Bitcoin Alpha en modelo LT.

prof	dir	prob vec	min xi	max xi	min fx	max fx	prom xi	prom fx	mediana xi	mediana fx	desv est xi	desv est fx	cant dist xi	cant val dist fx	t ejec
0	0	0	1	1	1	1004	1	1.69	1.0	1.0	0.0	7.44	1	55	2112,26
0	1	0	1	1	1	1004	1	1.69	1.0	1.0	0.0	7.44	1	55	2077,05
0	2	0	1	1	1	1004	1	1.69	1.0	1.0	0.0	7.44	1	55	2068,22
1	0	0.25	1	318	1	1890	1.41	18.78	1.0	2.0	2.36	118.34	27	138	2058,52
1	0	0.5	1	561	1	1970	1.73	32.15	2.0	2.0	4.14	156.59	39	205	2090,03
1	0	0.75	1	919	1	2007	2.17	51.79	2.0	2.0	6.76	199.48	52	273	2144,28
1	0	1	1	1145	1	2007	2.45	64.85	2.0	2.0	8.49	222.78	58	320	2168,02
1	1	0.25	1	315	1	1105	1.21	1.9	1.0	1.0	2.31	12.07	27	65	2022,97
1	1	0.5	1	559	1	1171	1.38	2.04	1.0	1.0	4.1	13.51	39	69	2016,14
1	1	0.75	1	918	1	1276	1.62	2.22	1.0	1.0	6.73	15.03	49	80	2025,72
1	1	1	1	1144	1	1336	1.79	2.37	1.0	1.0	8.48	17.1	58	84	2022,09
1	2	0.25	1	13	1	1890	1.21	18.57	1.0	2.0	0.46	117.97	11	121	2083,60
1	2	0.5	1	19	1	1970	1.38	31.89	1.0	2.0	0.61	156.37	16	186	2084,92
1	2	0.75	1	31	1	2007	1.62	51.45	2.0	2.0	0.8	199.36	20	253	2158,92
1	2	1	1	35	1	2007	1.79	64.47	2.0	2.0	0.88	222.65	24	297	2174,20
4	0	0.25	1	520	1	1890	2.14	20.07	1.0	2.0	9.52	119.58	103	240	2054,88
4	0	0.5	1	953	1	1970	3.05	34.45	2.0	2.0	17.4	159.52	154	348	2082,22
4	0	0.75	1	1556	1	2138	4.32	54.9	2.0	2.0	28.37	203.3	200	448	2158,72
4	0	1	1	1966	1	2283	5.19	68.48	2.0	2.0	35.74	227.36	234	491	2197,29
4	1	0.25	1	518	1	1409	1.94	3.21	1.0	1.0	9.49	20.57	107	185	2073,44
4	1	0.5	1	951	1	1774	2.71	4.41	1.0	1.0	17.37	34.42	149	240	2052,35
4	1	0.75	1	1555	1	2137	3.77	5.41	1.0	1.0	28.34	42.68	196	271	2065,76
4	1	1	1	1966	1	2282	4.53	6.02	1.0	1.0	35.72	48.21	222	288	2048,35
4	2	0.25	1	78	1	2196	2.06	57.15	1.0	2.0	3.68	224.22	60	396	2189,58
4	2	0.5	1	137	1	2870	2.77	80.69	1.0	2.0	6.13	272.91	95	602	2290,63
4	2	0.75	1	241	1	3626	3.86	104.49	2.0	2.0	10.5	324.85	146	740	2446,61
4	2	1	1	306	1	3736	4.57	124.9	2.0	2.0	13.12	379.2	176	847	2642,46
6	0	0.25	1	692	1	1890	3.43	22.43	1.0	2.0	19.65	123.87	206	368	2105,74
6	0	0.5	1	1256	1	2209	5.37	38.59	2.0	2.0	35.55	169.22	298	520	2120,12
6	0	0.75	1	2059	1	2657	8.02	60.49	2.0	2.0	57.27	216.65	370	612	2166,23
6	0	1	1	2599	1	2826	9.94	75.1	2.0	2.0	72.81	243.0	421	646	2212,07
6	1	0.25	1	689	1	1742	3.23	5.57	1.0	1.0	19.62	38.04	208	326	2066,13
6	1	0.5	1	1254	1	2208	5.02	8.55	1.0	1.0	35.51	66.3	295	419	2072,21
6	1	0.75	1	2058	1	2656	7.48	11.01	1.0	1.0	57.23	85.97	369	442	2087,62
6	1	1	1	2598	1	2825	9.28	12.66	1.0	1.0	72.79	98.45	402	439	2090,04
6	2	0.25	1	152	1	2621	3.37	83.11	1.0	2.0	9.94	294.57	117	684	2364,95
6	2	0.5	1	249	1	3424	4.97	113.5	1.0	2.0	16.61	372.05	183	829	2726,25
6	2	0.75	1	428	1	4188	7.71	153.96	2.0	2.0	29.21	494.35	295	928	2889,63
6	2	1	1	540	1	4518	9.41	183.98	2.0	2.0	37.09	582.7	348	980	3239,09

Tabla A.6: Matriz de resultados para red Higgs Twitter en modelo LT.

# Apéndice B

## Manual usuario plataforma experimentación GISR

Este proyecto ha sido desarrollado en base al lenguaje de programación Python y algunas librerías de código abierto que nutren a este lenguaje y permiten realizar un análisis de datos y grafos más exhaustivo.

### B.1. Índice

- Introducción [B.2](#)
- Requisitos Previos [B.3](#)
- Instalación y Configuración [B.4](#)
- Manejo del Sistema [B.5](#)
- Cambios al Sistema [B.6](#)
- Comentarios Finales [B.7](#)

### B.2. Introducción

Este proyecto se encuentra desarrollado en un conjunto de archivos Python dentro del [GitHub](#) del Proyecto que deben ser ejecutados de manera local. Los archivos siguiendo el orden que se detalla más adelante en este mismo manual permitirán adaptar archivos para la realización de pruebas e investigaciones utilizando la medida de centralidad GISR.

### B.3. Requisitos previos

Para poder ejecutar la plataforma de experimentación o implementar nuevas características o mejoras, es necesario tener instalado en nuestra máquina algunos requisitos de software que se detallan a continuación.

- Python [ir al sitio Web](#) Es el lenguaje de programación en el que está escrito la totalidad de la plataforma de experimentación.
- Anaconda [ir al sitio Web](#) Es una distribución de Python y si bien no obligatorio tenerla en el equipo sí acorta gran cantidad de tiempo debido a que ya posee varias de las librerías que si se utilizan en el sistema.
- NetworkX [ir al sitio Web](#) Es la librería que permite la manipulación de los grafos en este trabajo.
- Matplotlib [ir al sitio Web](#) Es la librería que permite visualizar gráficos para este trabajo.
- Numpy [ir al sitio Web](#) Es una librería que dota al lenguaje Python con algunas características que permiten análisis de datos a mayor escala.
- SciPy [ir al sitio Web](#) Es una librería que dota al lenguaje Python con algunas características que permiten análisis de datos a mayor escala.

Tener un ligero conocimiento de Python, sobre algunas de estas librerías y grafos permitirá al usuario comprender fácilmente cómo funciona la plataforma de experimentación cómo poder editarla para agregar sus propias funciones y/o mejoras.

### B.4. Instalación y configuración

Una vez instalado todos los requisitos anteriores y habiendo descargado el proyecto GitHub, basta con ejecutar los códigos en Shell o línea de comandos de su preferencia.

#### B.4.1. Configuración

##### Moldear archivos

Si nuestros datos son archivos que contengan en dos filas (o más) los datos de un grafo, en primera instancia hay que editarlos utilizando el archivo molde.py perteneciente al conjunto de archivos del proyecto en GitHub esto hará que del archivo que le ingresemos vía código cree uno nuevo apto para la ejecución de este.

## APÉNDICE B. MANUAL USUARIO PLATAFORMA EXPERIMENTACIÓN GISR101

En la Figura [B.1](#) se ve el archivo molde.py donde se debe editar las secciones `open('', 'r')` para poder darle la ubicación del archivo que se desea moldear y por otro lado `open('', 'w')` para poder dar la ubicación donde se guardará el archivo.

```
### preparación para un archivo IC-Model
archivo = open('', 'r') # Colocar la ruta directa al archivo que se desee moldear
lineas = archivo.readlines()
archivo.close()
archivoMoldeado = open('', 'w') # Colocar la ruta directa donde quiere que se guarde el archivo moldeado se recomienda que el nombre del archivo contenga IC como distintivo
archivoSemilla = open('', 'w') # Colocar la ruta directa donde se guardarán las semillas del nuevo archivo se recomienda que el nombre del archivo contenga IC como distintivo

seed = int(time.time())
archivoSemilla.write(str(seed))

for numerolinea in range(len(lineas)):
    seed = seed + 1
    objetoRandom = random.Random(seed)
    objetoRandomVec = random.Random(objetoRandom)
    lineas[numerolinea] = lineas[numerolinea].rstrip("\n").split(" ")
    aux = lineas[numerolinea][0], lineas[numerolinea][1], round(objetoRandom.random(), 1), round(objetoRandomVec.random(), 1)
    archivoMoldeado.write(str(aux[1]) + ' ' + str(aux[0]) + ' ' + str(aux[2]) + ' ' + str(aux[3]) + '\n')
    print(aux[0], aux[1], aux[2], aux[3], seed)

archivoMoldeado.close()
archivoSemilla.close()
```

Figura B.1: Extracto de archivo molde.py.

### Archivos moldeados

En la Figura [B.2](#) se puede ver un ejemplo de un archivo apto para ser ingresado al sistema.

El primer dato sería el nodo que apunta hacia el segundo nodo y en específico este archivo está moldeado para trabajar con IC-Model por lo tanto el tercer valor representa la probabilidad que tiene el primer nodo de influenciar al siguiente nodo y por último el cuarto valor se utiliza para ver cuál es la probabilidad de influenciar vecinos antes de utilizar el modelo de dispersión de influencia.

## APÉNDICE B. MANUAL USUARIO PLATAFORMA EXPERIMENTACIÓN GISR102

```
8614 161345 0.8 0.6
11792 428368 0.7 0.8
10701 77904 0.0 0.6
286277 124554 0.8 0.6
194873 194873 0.1 0.8
16460 341375 0.5 0.6
220 436133 0.6 0.6
274149 274148 0.5 0.8
22252 12866 0.0 0.6
35248 425029 0.8 0.6
39410 126440 0.2 0.8
179783 279630 0.8 0.6
322 107949 0.4 0.6
456 387599 0.7 0.8
52504 158380 0.3 0.6
191708 201087 0.7 0.6
345940 345941 0.9 0.8
190354 402405 0.5 0.6
13537 3228 0.2 0.6
88 126681 0.8 0.8
8438 27095 0.9 0.6
88 245841 0.3 0.6
30798 138795 0.5 0.8
3726 318765 0.9 0.6
88 424421 0.3 0.6
57105 49154 0.5 0.8
146177 188537 0.1 0.6
351168 129481 0.3 0.6
4986 30359 1.0 0.8
79530 256008 0.4 0.6
88 188556 0.8 0.6
2177 455874 0.5 0.8
28919 20187 0.2 0.6
53360 53342 0.1 0.6
358743 307829 0.0 0.8
220962 271255 0.2 0.6
377924 307876 0.8 0.6
1574 310821 0.5 0.8
7034 67087 0.5 0.6
12508 427962 0.7 0.6
144131 222612 0.9 0.8
72698 55655 0.7 0.6
41747 288546 0.4 0.6
395758 316677 0.6 0.8
```

Figura B.2: Extracto de archivo moldeado.

### Edición código fuente

Para la primera versión de la plataforma, es necesario editar el código fuente para insertar la ruta del archivo que contenga la red como se muestra en la Figura [B.3](#) a continuación, donde se busca el archivo, se conserva el contenido y luego se cierra el archivo, dado que sólo necesitamos su contenido.

```
archivo_red = open('insertar ruta al archivo aquí', 'r')
mensaje = archivo_red.readlines()
archivo_red.close()
```

Figura B.3: Extracto del Código Fuente a editar.

### Parámetros de utilización

En la Figura [B.4](#) se puede ver como se ejecuta el archivo **plat\_GISR.py** el cual es la plataforma de experimentación de la medida GISR con el parámetro **-h** el cual nos muestra cuáles son los distintos parámetros y formas de ejecución de la plataforma.

```
PS C:\Users\jvera\Desktop\programatesis> python.exe .\plat_GISR.py -h
usage: plat_GISR.py [-h] [-a A] [-l L] [-d D] [-r R] -m M

Medida de centralidad generalizada.

optional arguments:
  -h, --help  show this help message and exit
  -a A        Nodo en específico al que se le calculará la centralidad.
  -l L        Nivel de profundidad.
  -d D        Sentido de influencia 0 Ambos sentidos, 1 Salida, 2 Entrada.
  -r R        valor de influencia en la vecindad en probabilidades. 0, 0.25, 0.50, 0.75, 1
  -m M        Modelo de dispersión a utilizar 1 IC-Model 2 LT-Model.
```

Figura B.4: Ejecución plataforma con parámetro de ayuda.

A continuación, se describen los diferentes parámetros:

- -a Se refiere al cálculo en específico de un nodo perteneciente a la red testada.
- -l Se refiere al nivel de la profundidad, que se refiere a qué distancia se buscan los vecinos por cada nodo.
- -d Se refiere a la dirección en la cual se buscarán los vecinos por cada nodo, pueden ser nodos de salida (1) nodos de entrada (2) y ambos (0).
- -r Se refiere a la capacidad de influenciar vecinos previo a aplicar el modelo de dispersión de influencia, trabaja de manera probabilística y los valores pueden ser: 0 este es el mismo caso a cuando el nivel de profundidad es 0; 0.25 que se refiere al 25 % de probabilidades de influenciar a los vecinos; 0.5 se refiere al 50 %; 0.75 se refiere al 75 %; 1 se refiere al 100 %.

- -m se refiere a qué modelo de dispersión utilizar, para la primera versión de esta plataforma sólo existen IC-Model (1) y LT-Model (2).

## B.5. Manejo del sistema

Como se planteó en el anterior capítulo de este manual de usuario en primera instancia debemos moldear el archivo, modificar la ruta en el archivo fuente plat\_GISR.py y luego ejecutar este con los distintos parámetros. A continuación se muestran diferentes ejecuciones de plat\_GISR.py con diversos parámetros utilizando la red Football que se encuentra dentro del GitHub del proyecto.

- En la Figura B.5 se ve como se busca el resultado del nodo **1** a profundidad **3** con sentido de dirección **0** (ambos), influencia de probabilidad **1** (100 %) y utilizando el modelo **1** (IC-Model) los resultados se plantean como se ven en la imagen a continuación.

```
PS C:\Users\jvera\Desktop\programatesis> python.exe .\plat_GISR.py -a 1 -l 3 -d 0 -r 1 -m 1
7 esta es la cantidad de vecinos
El conjunto resultado es: [1, 5, 7, 9, 12, 13, 14, 18, 20, 25, 33] de tamaño: 11
PS C:\Users\jvera\Desktop\programatesis> █
```

Figura B.5: Ejecución plataforma con nodo específico.

- En la Figura B.6 se ve la ejecución con los siguientes parámetros profundidad **3** con sentido de dirección **0** (ambos), influencia de probabilidad **1** (100 %) y utilizando el modelo **1** (IC-Model) los resultados se plantean como se ven en la Figura B.7. En este último caso dado que no buscamos la centralidad de un nodo en concreto nos crea un archivo con los resultados de todos los nodos de la red y en el nombre del archivo se ve además los códigos de los parámetros utilizados. El archivo creado se ve luego de la ejecución, nos arroja en la primera fila las cabeceras de cada columna donde:  $i$  será el nodo del cual se calcula la centralidad;  $|X_i|$  se refiere a la cardinalidad de la cantidad de vecinos influenciados;  $|F(X_i)|$  se refiere a la cantidad de nodos influenciados luego de haber aplicado el modelo de dispersión de influencia; los siguientes datos arrojan los parámetros utilizados para el cálculo. Las siguientes filas serán los resultados y la última es el tiempo demorado de ejecución en milisegundos.

Ejecución

APÉNDICE B. MANUAL USUARIO PLATAFORMA EXPERIMENTACIÓN GISR105

```
PS C:\Users\jvera\Desktop\programatesis> python.exe .\plat_GISR.py -l 3 -d 0 -r 1 -m 1
El nombre del archivo con sus resultados es: footballIC301.0.csv
PS C:\Users\jvera\Desktop\programatesis> █
```

Figura B.6: Ejecución plataforma sin nodo específico.

Resultado

i	Xi	F(Xi)	profundidad 3	dirección 0	prob vecinos 1.0
7	9	9			
1	3	6			
9	9	11			
8	11	11			
2	5	14			
16	7	8			
35	10	10			
24	11	13			
3	3	15			
29	8	10			
5	3	11			
28	4	7			
6	3	11			
10	19	29			
4	5	8			
11	9	9			
17	2	2			
22	7	8			
25	1	5			
26	7	8			
31	1	2			
33	1	3			
34	7	7			
12	21	30			
18	1	3			
13	16	24			
21	3	3			
30	1	3			
14	13	20			
19	2	2			
15	4	14			
20	5	11			
23	2	2			
27	5	13			
32	8	21			
0.0040209293365478516					

Figura B.7: Resultado ejecución sin nodo específico.

Luego teniendo los resultados, puede trabajarse con otros dos archivos *estadisticas.py* y *spearmangeneral.py* que permiten el análisis de datos, ambos necesitan al menos uno y es preferible más archivos de resultados como el anterior.

La Figura B.8 presenta el archivo *estadisticas.py* donde se busca dentro de una carpeta en concreto todos los archivos de resultados y finalmente arroja un archivo como el que se

## APÉNDICE B. MANUAL USUARIO PLATAFORMA EXPERIMENTACIÓN GISR106

aprecia en la Figura [B.9](#) por red donde se consiguen en base a todos los resultados, diversos cálculos y datos respecto a conjunto de los archivos de resultados. Con estos datos pueden realizarse análisis como gráficos.

```
import os
import csv
import statistics
import numpy as np

archivos = os.listdir('insertar ruta de la carpeta con resultados')

football = open('insertar ruta del archivo donde se guardarán los resultados', 'w', newline='')
escribir = csv.writer(football, delimiter=',')
escribir.writerow(['profundidad', 'direccion', 'provecinos', 'min xi', 'max xi', 'min fx', 'max fx', 'promedio xi', 'promedio fx', 'mediana xi', 'mediana fx', 'desviacion estandar xi', 'desviacion estandar fx'])
for cant_archivos in archivos: ## se recorren los nombres de los archivos en la carpeta de resultados.
    documento = open('inserte la ruta de la carpeta don resultados'+cant_archivos, 'r')
    lineas = documento.readlines()
    primeraline = lineas.pop(0).rstrip("\n").split(";")
    ultimaline = lineas.pop().rstrip("\n").split(";")
    ultimaline[0] = ultimaline[0].replace('.', ',')
    xi = []
    fx = []
    for linea in lineas:
        aux = linea.rstrip("\n").split(";")
        xi.append(int(aux[1]))
        fx.append(int(aux[2]))
    minimo = min(xi), min(fx)
    maximo = max(xi), max(fx)
    escritura = primeraline[3][-1:], primeraline[4][-1:], primeraline[5][12:], minimo[0], maximo[0], minimo[1], maximo[1], round(statistics.mean(xi), 2), round(statistics.mean(fx), 2), round(statistics.stdev(xi), 2), round(statistics.stdev(fx), 2)
```

Figura B.8: Extracto de archivo de cálculo estadísticas.

## APÉNDICE B. MANUAL USUARIO PLATAFORMA EXPERIMENTACIÓN GISR107

### Resultados estadísticos

prof	dir	prob_vec	min xi	max xi	min fx	max fx	prom xi	prom fx	mediana xi	mediana fx	des_est xi	des_est fx	cant_dist xi	cant_dist fx	tiempo ejecucion
0	0	0	1	1	1	9	1 3.31		1	2 0.0	2.55		1	8 0.0	
0	1	0	1	1	1	12	1 3.6		1	2 0.0	2.86		1	10 0.015654325485229492	
0	2	0	1	1	1	9	1 3.31		1	2 0.0	2.61		1	9 0.0	
1	0	0.25	1	8	1	20 3.23	7.6		3	8 1.73	3.93		7	14 0.0	
1	0	0.5	1	8	1	19 3.23	7.2		3	7 1.73	4.01		7	13 0.0	
1	0	0.75	2	20	2	28 7.74	12.14		7	11 4.71	6.35		14	19 0.0	
1	0	1	2	20	2	29 7.74	12.17		7	11 4.71	6.32		14	18 0.01550912857055664	
1	1	0.25	1	5	1	10 2.11	4.17		2	4 1.12	2.85		5	10 0.0	
1	1	0.5	1	5	1	10 2.11	4.4		2	4 1.12	2.92		5	10 0.0	
1	1	0.75	1	11	1	13 4.37	5.34		4	6 2.89	3.47		10	12 0.0	
1	1	1	1	11	1	13 4.37	5.37		4	6 2.89	3.48		10	12 0.0	
1	2	0.25	1	7	1	18 2.11	7.11		1	7 1.77	4.26		7	15 0.0	
1	2	0.5	1	7	1	20 2.11	7.14		1	7 1.77	4.76		7	13 0.0	
1	2	0.75	1	19	2	30 4.37	10.6		2	9 5.12	7.35		11	19 0.0	
1	2	1	1	19	1	30 4.37	10.86		2	8 5.12	7.53		11	18 0.0	
4	0	0.25	1	12	1	24 4.14	8.74		4	8 2.6	5.73		10	14 0.015538454055786133	
4	0	0.5	1	12	1	24 4.14	8.63		4	8 2.6	4.8		10	14 0.0	
4	0	0.75	2	26	2	34 10.31	13.69		9	11 6.62	8.67		16	18 0.0	
4	0	1	2	26	2	34 10.31	13.54		9	11 6.62	8.47		16	17 0.015652894973754883	
4	1	0.25	1	6	1	10 2.51	4.46		2	5 1.46	2.89		6	10 0.0	
4	1	0.5	1	6	1	9 2.51	4.37		2	4 1.46	2.84		6	9 0.0	
4	1	0.75	1	13	1	13 5.66	5.66		6	6 3.59	3.59		13	13 0.015536069869995117	
4	1	1	1	13	1	13 5.66	5.66		6	6 3.59	3.59		13	13 0.0	
4	2	0.25	1	12	1	24 2.63	7.89		1	7 2.87	5.81		7	14 0.0	
4	2	0.5	1	12	1	24 2.63	7.83		1	7 2.87	5.58		7	15 0.0	
4	2	0.75	1	26	1	34 5.66	12.29		2	9 7.69	9.55		12	20 0.015653371810913086	
4	2	1	1	26	1	33 5.66	11.74		2	8 7.69	9.73		12	18 0.0	
6	0	0.25	1	12	1	24 4.14	8.77		4	8 2.6	5.53		10	15 0.0	
6	0	0.5	1	12	1	24 4.14	8.26		4	7 2.6	5.4		10	17 0.015535831451416016	
6	0	0.75	2	26	2	34 10.31	13.71		9	11 6.62	8.59		16	19 0.0	
6	0	1	2	26	2	34 10.31	13.8		9	11 6.62	8.72		16	17 0.01565408706665039	
6	1	0.25	1	6	1	12 2.51	4.63		2	5 1.46	3.02		6	10 0.0	
6	1	0.5	1	6	1	12 2.51	4.89		2	5 1.46	3.24		6	10 0.0	
6	1	0.75	1	13	1	13 5.66	5.66		6	6 3.59	3.59		13	13 0.0	
6	1	1	1	13	1	13 5.66	5.66		6	6 3.59	3.59		13	13 0.015536785125732422	
6	2	0.25	1	12	1	24 2.63	7.71		1	6 2.87	5.77		7	17 0.0	
6	2	0.5	1	12	1	25 2.63	7.77		1	7 2.87	6.06		7	16 0.0	
6	2	0.75	1	26	1	34 5.66	12.74		2	10 7.69	9.43		12	20 0.0	
6	2	1	1	26	1	34 5.66	12.06		2	8 7.69	9.99		12	21 0.0	

Figura B.9: Archivo de resultados estadísticos.

La Figura [B.10](#) presenta el archivo `spearmangeneral.py` donde se busca dentro de una carpeta en concreto todos los archivos de resultados y finalmente arroja un archivo como el que se aprecia en la Figura [B.11](#) de matriz de correlación por red donde se consiguen en base a todos los resultados, la correlación entre ellos.

## APÉNDICE B. MANUAL USUARIO PLATAFORMA EXPERIMENTACIÓN GISR108

```
from scipy import stats
import csv
import os

correlacion = open('Insertar ruta donde se dejará el resultado','w',newline='')
escribir = csv.writer(correlacion, delimiter=',')
archivos = os.listdir('insertar ruta de carpeta donde se encontraran los resultados')
primeraescritura = True
primeraescribir = ['.',]
escribirdata = []
cont = 0
spearmanuno = []
spearmandos = []
for nombrearchivo in archivos:
    documento = open('insertar ruta de carpeta donde se encontraran los resultados'+nombrearchivo, 'r')
    lineas = documento.readlines()
    primeralinea = lineas.pop(0).rstrip("\n").split(";")
    ultimalinea = lineas.pop().rstrip("\n").split(";")# haciendo esto me quedo con sólo los resultados.

    escribirdata.append([i[7]+i[8]+primeralinea[3][-1:]+primeralinea[4][-1:]+primeralinea[5][12:]]
    spearmanuno = []
    for cantidadLineas in range(len(lineas)):
        lineas[cantidadLineas] = lineas[cantidadLineas].rstrip("\n").split(";")
        spearmanuno.append(lineas[cantidadLineas][2])
    for nombrearchivos in archivos:
        documentodos = open('insertar ruta de carpeta donde se encontraran los resultados'+nombrearchivos, 'r')
        lineasdos = documentodos.readlines()
        ultimaledos = lineasdos.pop().rstrip("\n").split(";")
        primerados = lineasdos.pop(0).rstrip("\n").split(";")
        if primeraescritura:
            primeraescribir.append(j[7]+j[8]+primerados[3][-1:]+primerados[4][-1:]+primerados[5][12:]]
            spearmandos = []
            for h in range(len(lineasdos)):
                lineasdos[h] = lineasdos[h].rstrip("\n").split(";")
                spearmandos.append(lineasdos[h][2])
                # print(lineasdos[h][2]) así accedo al resultado
            valor, aux = stats.spearmanr(spearmanuno, spearmandos)
            if aux <= 0.05:
                escribirdata[cont].append(valor)
            else:
                escribirdata[cont].append('x')
        cont = cont + 1
    if primeraescritura:
        # print(primeraescribir) la primera línea del archivo, se escribe una vez
        primeraescritura = False
    escribir.writerow(primeraescribir)
    escribir.writerows(escribirdata)
```

Figura B.10: Extracto de archivo cálculo correlación.

Matriz de correlación con mapa de calor

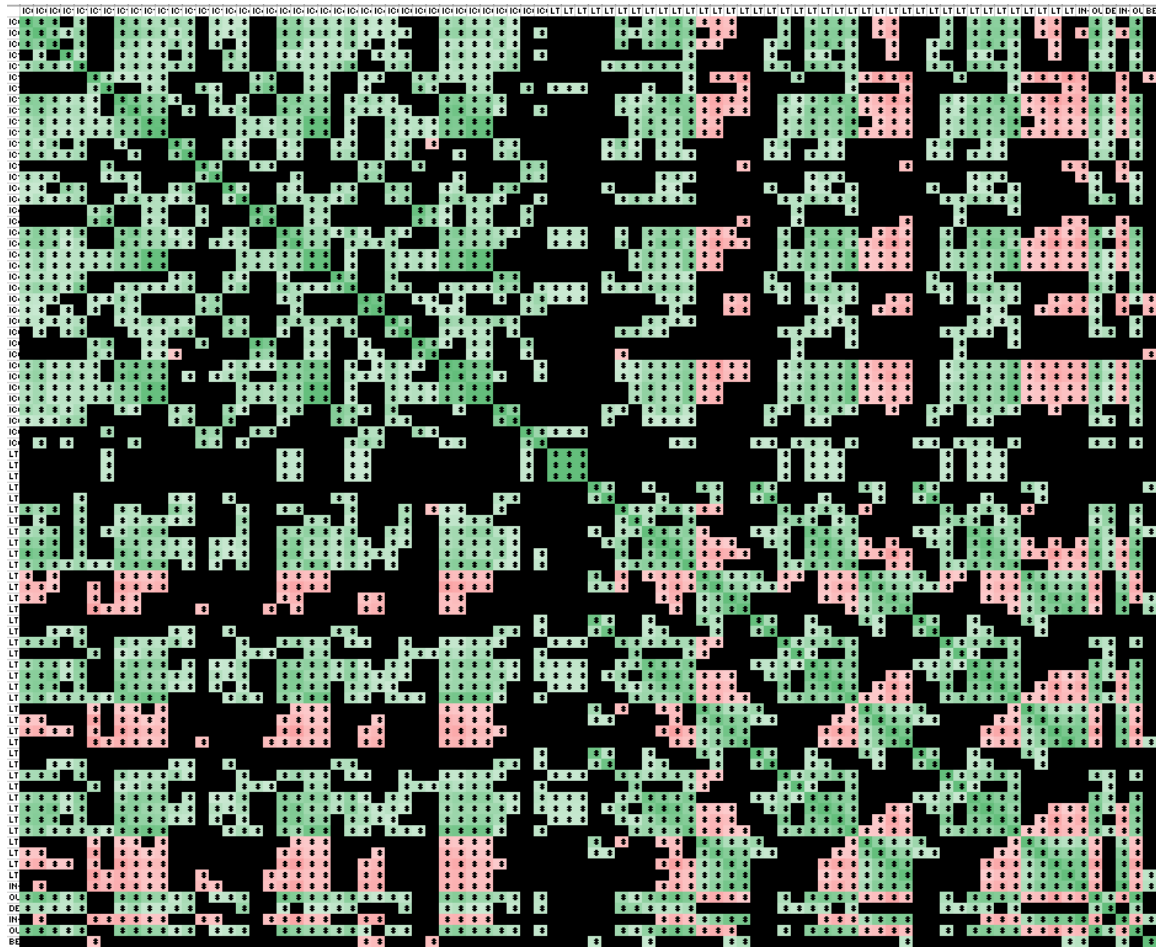


Figura B.11: Matriz de correlación con mapa de calor.

## B.6. Cambios del sistema

En cambios futuros o futuros proyectos relacionados con esta plataforma, se podría proponer como una buena idea, incluir nuevos modelos de dispersión de influencia, también llevar esta serie de archivos a un archivo que permita una ejecución en conjunto sin necesidad de ir archivo por archivo, una aplicación web que además de permitir lo anterior pueda también ser más sencilla de utilizar gracias al aporte de la interfaz gráfica.

## **B.7. Comentario finales**

Esperando que esta guía sea de ayuda para el usuario y permita mejorar el sistema a futuro, además de que cada punto haya sido abordado y desarrollado satisfactoriamente para el lector, se espera que pueda levantarlo y manejarlo de manera satisfactoria, deseando lo mejor del camino de desarrollo para el usuario.

**Hasta la Próxima y Suerte!**

**Nota: Este Manual de usuario requiere que el lector tenga un conocimiento previo de Python, las librerías que se utilizan en el proyecto y además un conocimiento ligero de lo que son grafos y estudios de este como son las medidas de centralidad y los modelos de dispersión de influencia.**