



Universidad de Los Andes  
Facultad de Ciencias  
Departamento de Física

## Implementación en ISyS de medidas para la caracterización de redes complejas

Autor: T.S.U. Hebert J. Avendaño L.

Tutor: Dr Kay Tucci.

Trabajo especial de grado presentado ante la ilustre Universidad de Los Andes como  
requisito parcial para optar al título de Licenciado en Física.

Mérida, Junio 2010

# Índice general

Índice general	2
Resumen	6
Agradecimientos	7
Introducción	8
<b>1. Fundamentos teóricos</b>	<b>11</b>
1.1. ¿Qué es la herramienta ISyS?	11
1.2. Grafos	14
1.2.1. Representación de grafos	16
1.2.2. Matriz Adyacencia	16
1.2.3. Conjunto de vecinos	17
1.3. Redes Complejas o Reales	18
1.3.1. Redes aleatorias de Erdos y Renyi	18
1.3.2. Redes de pequeño mundo de Watts Strogatz	19
1.3.3. Redes libres de escala de Barabasi y Albert	21
<b>2. Diseño e implementación del modulo de medidas estadísticas de ISyS</b>	<b>24</b>
2.1. Medidas	26
2.1.1. Grado	26
2.1.2. Distancia promedio	32
2.1.3. Eficiencia	34
2.1.4. Vulnerabilidad	37
2.1.5. Clustering o Agrupamiento	39
2.1.5.1. Espectro de autovalores	41

<b>3. Resultados</b>	<b>43</b>
3.1. Tiempo de ejecución. . . . .	43
3.2. Representación y comparación gráfica de las algunas de las medidas implementadas. . . . .	50
<b>Conclusiones y recomendaciones</b>	<b>55</b>
<b>Referencias</b>	<b>56</b>

# Índice de Figuras

<b>1</b>	<b>Fundamentos teóricos</b>	<b>11</b>
1.1.	Al izquierda podemos observar el retrato del matemático Leonhard Euler, quien estudio el problema de los Puentes de Konigsberg en el siglo <i>XVIII</i> . A la derecha se observa un plano del siglo <i>XVIII</i> , de la ciudad de Königsberg, con los siete puentes. . . . .	15
1.2.	Grafo que consta de 4 nodos ( <i>circulos</i> ) y 7 arcos ( <i>líneas</i> ). Los arcos no tienen dirección definida, es decir, éste es un grafo simple. . . . .	15
1.3.	Grafo dirigido. Los arcos dirigidos se representan con flechas. . . . .	16
1.4.	Representa una de las tantas redes aleatorias existentes en la naturaleza.[?] . . . . .	20
1.5.	Representación de una red social de internet basado en el experimento de S. Milgram, The Small World Problem[23]. . . . .	21
1.6.	Representa la interacción entre proteínas[22]. Ésta red es libre de escala . . . . .	23
<b>2</b>	<b>Diseño e implementación del modulo de medidas estadísticas de ISyS</b>	<b>24</b>
<b>3</b>	<b>Resultados</b>	<b>43</b>
3.1.	Tiempos de ejecución en función del tamaño de la red para las medidas (a) $k_{in}$ y (b) $k_{out}$ . BA (cuadrados), ER (circulos), Regular (triángulos), WS (rombos) . . . . .	46

3.2.	Tiempos de ejecución en función del tamaño de la red para las medidas $H_{in}$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos).	46
3.3.	Tiempos de ejecución en función del tamaño de la red para la Longitud característica $L$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos). . . . .	47
3.4.	Tiempos de ejecución en función del tamaño de la red para la Longitud característica $L$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos). . . . .	48
3.5.	Tiempos de ejecución en función del tamaño de la red para la vulnerabilidad máxima $V_{max}$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos). . . . .	49
3.6.	Tiempos de ejecución en función del tamaño de la red para el espectro de autovalores $\lambda$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos). . . . .	50
3.7.	Gráfica de la probabilidad en función del grado de la red $P_k$ con $N = 10000$ : A la izquierda la red $BA$ , a la derecha la red $ER$ . . . . .	50
3.8.	Gráfica de la probabilidad en función del grado de la red $P_k$ con $N = 10000$ , para una red del tipo $WS$ $P = 0,1$ y $P = 0,5$ . . . . .	51
3.9.	Variación de la entropía $H$ como función de la probabilidad de recableado $P$ en una red $WS$ de tamaño $N = 512$ en un anillo unidimensional con $k = 8$ . . . . .	52
3.10.	Gráfica de la longitud característica vs el coeficiente de agrupamiento. Note que están normalizados a 1 . . . . .	52
3.11.	Vulnerabilidades máximas $V_{max}$ de las redes $SW$ para diferentes valores de probabilidad de recableado $N = 512$ , $d = 1$ , $k = 8$ , . . . . .	53
3.12.	Autovalores de la matriz adyacencia para redes $ER$ con $N = 512$ y $k = 8$ . Cada red tiene una fracción diferente de arcos asimétricos ( $x$ ) $q = 0$ , $q = 0,20$ y ( $0$ ) $q = 1$ . . . . .	54

# Índice de Ejemplos

1.	Extracto del archivo de cabecera. . . . .	25
2.	Función para medir el $k_i^{in}$ de un nodo. . . . .	27
3.	Función para medir el $k_i^{out}$ de un nodo. . . . .	27
4.	Función para medir el $\langle k_i^{in} \rangle$ de la red. . . . .	29
5.	Función para medir el $\langle k_i^{out} \rangle$ de la red. . . . .	29
6.	Distribución de los arcos entrantes . . . . .	30
7.	Distribución de los arcos salientes . . . . .	31
8.	Entropía de la red . . . . .	32
9.	Distancia Promedio . . . . .	33
10.	Eficiencia global . . . . .	35
11.	Eficiencia sin el nodo $i$ . . . . .	36
12.	Media Armónica . . . . .	37
13.	Vulnerabilidad . . . . .	38
14.	Vulnerabilidad promedio de la red . . . . .	39
15.	Coficiente de clustering para grafos dirigidos . . . . .	41
16.	Autovalores . . . . .	42
17.	Mide el tiempo de ejecución de la longitud característica. . . . .	44
18.	Archivo para la construcción del modelo. . . . .	45

# Resumen

ISyS es una plataforma desarrollada para el estudio de redes de especial utilidad para modelar, simular y analizar sistemas dinámicos extendidos sobre espacios no uniformes, permitiendo implementar modelos de autómatas celulares, redes de mapas acoplados y redes neuronales; todo esto de una manera general y en un contexto integral. Con este trabajo se crea un módulo nuevo de ISyS con el propósito de caracterizar redes complejas. En el se implementaron los caracterizadores: Grado de la red, distribución de probabilidades de los grados entrantes y salientes, entropía de la distribución de grado, longitud característica, eficiencia, eficiencia sin un nodo, vulnerabilidad sin un nodo, vulnerabilidad máxima, coeficiente de agrupamiento para grafos simples y para dirigidos; y espectro de autovalores de la red. Para todos ellos se verificó su correcto funcionamiento mostrando, por razones de espacio en este manuscrito, solamente los resultados más ilustrativos. La documentación de todas las funciones implementadas se encuentra en el archivo de cabecera del módulo. Por último, hemos considerado útil para los futuros usuarios de este nuevo módulo mostrar como escala el tiempo de ejecución  $t$  de cada uno de los caracterizadores en función del tamaño de la red  $N$  a caracterizar. Hemos encontrado que generalmente el escalamiento obedece a una ley de potencia  $t \sim N^\gamma$ , donde el exponente  $\gamma$  depende del caracterizador.

**Palabras clave:**

ISyS, grafos dirigidos, redes, apuntadores, listas de listas, Diseño algorítmico, lenguaje C.

# Agradecimientos

*Agradezco infinitamente a Dios y la Virgen, por darme la vida y la salud necesaria para trabajar en este proyecto de investigación a pesar de los contratiempos.*

*Agradezco ampliamente con mucha sinceridad y humildad:*

*A mi familia (abuelos, padres y hermanos) por el cariño incondicional y apoyo moral que me han brindado desde mi nacimiento.*

*A mi tutor y jurados Kay, Mayerlin y Mario, por ser los orientadores que necesité durante el desarrollo de mi pre-grado. Dios me brindó nuevamente la oportunidad de adquirir nuevos y sabios conocimientos a través de ustedes en ésta etapa importante de mi vida.*

*A mis compañeros y amigos de pregrado, Elvis, Morena, Jesus y Fernando que con su apoyo moral me auxiliaron cuando lo necesitaba.*

*A la planta profesoral del Pregrado del Departamento de Física por transmitirme de manera eficiente los conocimientos necesarios para el desarrollo de éste trabajo especial de grado para optar al título de Licenciatura en Ciencias Física.*

*A todas aquellas personas e instituciones que de alguna manera colaboraron con este proyecto.*

*“El éxito no es para los que piensan que pueden hacer algo  
sino para quienes lo hacen.”  
anónimo*

# Introducción

ISyS [1], cuyas siglas traducen Inhomogeneous Systems Simulator, es un paquete diseñado con la intención de servir las necesidades computacionales de la comunidad de investigadores dedicados al estudio de sistemas espacio temporales definidos sobre redes. La herramienta ha sido utilizada en numerosas investigaciones realizadas en el grupo de Caos y Sistemas Complejos del Centro de Física Fundamental en la Facultad de Ciencias de nuestra Universidad.

Con el presente proyecto se pretende incorporar a la plataforma ISyS un módulo nuevo para el cálculo de algunas propiedades estadísticas de las redes que sirven de sustrato a las dinámicas de los sistemas modelados con ella. En el módulo estarán contenidas, de forma ordenada y sistematizada, las ecuaciones relacionadas con el cálculo de la eficiencia global, la media armónica, la vulnerabilidad, la entropía, el número promedio de arcos entrantes y salientes, los histogramas de frecuencia de los arcos salientes y entrantes, la distancia promedio y el espectro de autovalores de la matriz de adyacencia; todo esto para grafos dirigidos y no dirigidos [4, 3, 17]. Todas éstas medidas se basan en la teoría de grafos, que es una disciplina antigua con muchas aplicaciones modernas [5]. Sus ideas básicas las introdujo el gran matemático suizo Leonhard Euler en el siglo *XVIII*. Euler utilizó esta teoría, para resolver el famoso problema de los puentes de Königsberg. Pero no fue sino hasta 1959, que

Erdős y Rényi [2] realizaron un estudio estadístico de las propiedades de un tipo particular de grafos, los de conectividad aleatoria.

La mayoría de los problemas relacionados con la teoría de grafos pertenecen a la clase de complejidad  $NP$ -completo [18], por lo que el problema de estudiar redes más realistas que las de conectividad aleatoria tuvo que esperar hasta finales de los años 90s con el trabajo sobre redes complejas de Watts y Strogatz [20] y, posteriormente, el de Barabasi-Albert [10]. Más recientemente han surgido otros tipos de redes, como las de comunidades [11], y actualmente se está planteando el estudio de la evolución de redes [12].

Es por esto que la plataforma ISyS cuenta con los subprogramas para crear redes de elementos dinámicos interactuantes, tanto sobre redes regulares como complejas. Sin embargo, ISyS carecía de algunas de las estadísticas para la caracterización de las redes, y es por esto, que hemos planteado el desarrollo de éste trabajo.

### **Objetivo General**

Diseñar e implementar el módulo de ISyS para el cálculo de algunas cantidades estadísticas que permitan caracterizar a las redes complejas.

### **Objetivos específicos**

- Seleccionar las medidas a implementar.
- Implementar en ISyS las medidas para la caracterización de redes complejas.
- Calcular las diferentes medidas para caracterizar las redes complejas ya implementadas en ISyS.
- Comparar los resultados obtenidos con los resultados teóricos y experimentales.

- Sistematizar la información haciendo uso de las normas de sintaxis establecidas para lograr la estabilidad y cohesión con los módulos ya existentes en ISyS.

Para sistematizar la información, la metodología de trabajo fué la siguiente: en primera instancia se seleccionaron las medidas a implementar de las propuestas por Newman [13], Dorogovtsev y Mendes [14], y Costa et al [3]; y así diseñar e implementar en ISyS los algoritmos respectivos. Luego, para las diferentes redes ya implementadas en ISyS, se probaron las medidas verificando que concuerden con los resultados teóricos o experimentales obtenidos previamente. Por último se escribió el manual del usuario y del analista.

Para cubrir los tópicos de este trabajo, hemos considerado conveniente organizar el documento de la siguiente manera:

En el primer capítulo presentaremos al lector, la base conceptual que proporcione el marco teórico o cuerpo de conocimiento necesarios para abordar los aspectos de la herramienta ISyS y de la teoría de grafos, importantes para el óptimo desarrollo del módulo a incorporar. Igualmente, describiremos los cuatro tipos de redes utilizadas en este trabajo, a saber: las redes regulares, las aleatorias (*ER*), con característica de pequeño mundo (*WS*) y las libre de escala (*BA*).

En el segundo capítulo nos concentraremos en el diseño e implementación del nuevo módulo contentivo de las diferentes medidas seleccionadas para caracterizar a las redes, siempre en consonancia con las normas que requiere ISyS [1].

En el tercer capítulo se presentan los resultados del diseño e implementación de las diferentes medidas seleccionadas para caracterizar a las redes en dos fases: tiempo de ejecución y comparación con los resultados teóricos y experimentales.

*“Saber que se sabe lo que se sabe y que no se sabe lo que no se sabe;  
he aquí el verdadero saber.”*  
*Confucio*

# Capítulo 1

## Fundamentos teóricos

En este capítulo se definirán las bases conceptuales que hemos considerado como marco teórico básico a fin de facilitar al lector, la comprensión de cada uno de los temas que se desarrollarán a lo largo del documento. En primer lugar, en la sección 1.1, haremos una revisión de la herramienta ISyS [1], haciendo un esbozo general acerca de su arquitectura y la sintaxis necesaria para desarrollar nuevos módulos de dinámica local, interacciones y topología de conexión. Seguidamente, de acuerdo con la formalidad matemática requerida para la construcción del módulo para la caracterización de redes, en la sección 1.2, presentaremos algunos conceptos relacionados con grafos [5], así como también ejemplos de las aplicaciones de la presente teoría a la realidad. Posteriormente, en la sección 1.3, se presenta algunas redes complejas, las cuales se caracterizan por tener un gran número de elementos y una estructura topológica no trivial.

### 1.1. ¿Qué es la herramienta ISyS?

ISyS es una herramienta desarrollada con la intención de servir las necesidades computacionales de la comunidad de investigadores dedicados al estudio de sistemas

espacio-temporales definidos sobre redes [1]. Se trata de una herramienta amigable para el usuario, simple de usar, y de especial utilidad para modelar, simular y analizar sistemas dinámicos extendidos sobre espacios no uniformes. ISyS permite implementar modelos de autómatas celulares, redes de mapas acoplados y redes neuronales; de una manera general y en un contexto integral.

Por su diseño modular, ISyS permite construir o modificar en forma independiente los sistemas dinámicos espacio-temporales en las siguientes características: la dinámica local de los elementos o celdas del sistema, la forma como interactúan entre sí, la topología de interconexión inicial del sistema y las propiedades estadísticas que se calculan en la simulación.

Su diseño está orientado para que el usuario solamente se ocupe de enlazar los diferentes módulos para construir el modelo que desea simular. Además, ISyS permite, en caso de que se requiera, integrar a la herramienta de simulación nuevos módulos desarrollados por el usuario, sin que sea necesario modificar los otros componentes de la plataforma, de este modo el usuario no requiere una gran experiencia como programador para ampliar la plataforma.

Adicionalmente, ISyS acumula los algoritmos desarrollados por diferentes usuarios para la construcción de topologías de interconexión, dinámicas locales, interacciones entre los elementos y estadísticos; los cuales pueden ser utilizados en otro momento, redundando en una disminución del tiempo de desarrollo de los modelos de simulación. Con ISyS se puede simular una amplia variedad de sistemas compuestos de elementos discretos cuyas variables de estado dependan de una dinámica local y de las interacciones o influencias de los estados de vértices vecinos. La herramienta ISyS está conformada por varios módulos, los cuales se pueden clasificar en tres grupos. El primer grupo, es el núcleo del sistema y lo conforman el motor de simulación, el módulo gráfico y el manejador de errores. El motor es el responsable de hacer avanzar la simulación del modelo en forma coherente llamando al resto de los módulos en el orden correcto. Además, se encarga de otras funciones, como solicitar al sistema

operativo los recursos necesarios para la ejecución del modelo. El módulo gráfico realiza el despliegue en pantalla u otro dispositivo de la simulación del sistema cuando así lo solicita el usuario; y el manejador de errores identifica y controla los errores que pueden ocurrir en tiempo de ejecución.

El segundo módulo lo conforman una serie de rutinas de soporte, como por ejemplo los generadores de números aleatorios, las paletas de colores y algunas funciones estadísticas. Por último, el tercer grupo lo conforman los módulos que definen al sistema dinámico espacio temporal, es decir, el módulo de la dinámica local de cada elemento del sistema y el módulo de la topología de las interrelaciones existentes entre los elementos.

El usuario, mediante un archivo de configuración para la compilación, selecciona cuáles módulos desea utilizar en su modelo y cuales estadísticas se desean calcular, así como también activa o desactiva el componente gráfico de ISyS. Una vez seleccionado el modelo y las estadísticas, la interfaz enlaza estos módulos con el motor y el resto de las herramientas que sean necesarias para poder construir el programa ejecutable.

Ya con el programa ejecutable del modelo, el usuario puede ejecutarlo con los parámetros con los que se desea simular y, a medida que la simulación avanza, ISyS muestra al usuario los resultados parciales de las estadísticas e imágenes de la evolución temporal del sistema; esto último siempre que el componente gráfico haya sido activado al momento de la construcción del programa ejecutable. Si las combinaciones de los módulos de dinámicas locales, interacciones entre elementos y topologías existentes no son suficientes para que el usuario construya un modelo adecuado, ISyS le permite desarrollar aquellos módulos que sean necesarios. Generalmente se deben crear dos archivos en lenguaje C; uno de cabecera, en el que están las declaraciones y definiciones; y otro con el código fuente de las nuevas funciones. Para mantener la compatibilidad con el resto del sistema, el usuario de ISyS debe seguir las normas de sintaxis al escribir estos dos archivos.

En particular, el aporte de éste trabajo se realiza en el módulo de las estadísticas de

ISyS; para ello se cumplió con los estándares específicos para lograr su interconexión con los demás módulos.

## 1.2. Grafos

Para 1736 la ciudad de Königsberg; (actualmente Kaliningrado) tenía siete puentes que comunicaban las islas en el río Pregel, entre sí y con tierra firme como se muestra en la figura 1.1. En aquel entonces los habitantes de Königsberg se preguntaban, ¿es posible dar un paseo empezando por una cualquiera de las cuatro partes de tierra firme, cruzando cada puente una sola vez y volviendo al punto de partida?. La respuesta a ésta pregunta la dio el matemático suizo Leonhard Euler, (ver a la izquierda de la figura 1.1) quien para resolver este problema, introdujo la teoría de grafos. Euler determinó, en el contexto del problema, que los puntos intermedios de un recorrido posible necesariamente han de estar conectados a un número par de arcos. En efecto, si llegamos a un punto desde algún arco, entonces el único modo de salir de ese nodo es por un arco diferente. Esto significa que tanto el punto inicial como el final serían los únicos que podrían estar conectados con un número impar de arcos. Sin embargo, el requisito adicional del problema dice que el nodo inicial debe ser igual al final, por lo que no podría existir más de un único nodo conectado con un número impar de arcos. En particular, los cuatro nodos poseen un número impar de arcos incidentes (dos de ellos inciden en tres arcos, y el restante incide en cinco), entonces se concluye que es imposible definir un camino con las características buscadas.

Los problemas que se pueden resolver con técnicas de la teoría de grafos aparecen resultados prácticamente en cualquier disciplina [5]. Desde el punto de vista matemático, los grafos son estructuras discretas que constan de vértices o nodos y de aristas o arcos que conectan a los vértices tal como se observa en la figura 1.2. Hay varios tipos de grafos, que se diferencian entre sí por el tipo y el número de aristas que pueden conectar cada par de vértices, todos ellos se pueden clasificar en dos tipos: los grafos no dirigidos o simples, y los grafos dirigidos o digrafos.



Figura 1.1: Al izquierda podemos observar el retrato del matemático Leonhard Euler, quien estudio el problema de los Puentes de Königsberg en el siglo *XVIII*. A la derecha se observa un plano del siglo *XVIII*, de la ciudad de Königsberg, con los siete puentes.

**Definición 1.1** *Un grafo simple o no dirigido denotado por  $G = (V, E)$  consta de  $V$ , un conjunto no vacío de vértices; y de  $E$ , un conjunto de pares no ordenados de elementos de  $V$ , llamados aristas.*

El grafo simple de la figura 1.2 es el grafo asociado al problema de los Puentes de Königsberg, donde los 4 nodos pertenecientes al conjunto  $V = \{(1, 2, 3, 4)\}$  representan a las 4 porciones de tierra conectadas por los puentes y los 7 arcos pertenecientes al conjunto  $E = \{(1, 2); (1, 2); (2, 3); (2, 3); (1, 4); (2, 4); (3, 4)\}$ , representan a los 7 puentes de Königsberg.

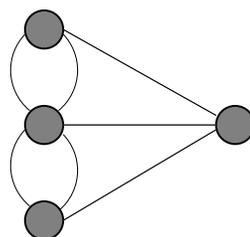


Figura 1.2: Grafo que consta de 4 nodos (*circulos*) y 7 arcos (*líneas*). Los arcos no tienen dirección definida, es decir, éste es un grafo simple.

**Definición 1.2** *Un grafo dirigido o digrafo es denotado por  $G = (V, E)$  consta de  $V$ , un conjunto no vacío de vértices y de un conjunto  $E$  de aristas, que son pares ordenados de elementos de  $V$ .*

Gráficamente, utilizamos una flecha apuntando desde el nodo  $u$  hacia el nodo  $v$  para indicar la dirección de la arista  $(u, v)$ . La figura 1.3 muestra un digrafo de 4 nodos y 7 arcos. Cuando existen los arcos que unen a dos nodos en ambas direcciones se representan como una línea sin flecha.

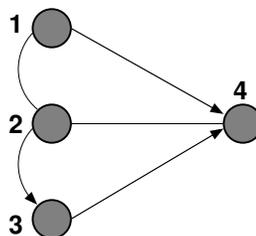


Figura 1.3: Grafo dirigido. Los arcos dirigidos se representan con flechas.

### 1.2.1. Representación de grafos

Además de la representación gráfica, existen muchas otras maneras de representar los grafos [5], y al trabajar con grafos conviene poder tener la posibilidad de elegir su representación más conveniente. ISyS utiliza grafos dirigidos sin arcos múltiples ni autoenlaces o bucles. Una forma de representar grafos sin arcos múltiples es enumerar todas las aristas del grafo. Otra forma de representar grafos sin arcos múltiples es utilizar listas de adyacencia, que especifican los nodos que son adyacentes a cada uno de los nodos del grafo. En esta sección presentaremos dos tipos de representación de grafos que se utilizan con frecuencia. Uno se basa en la adyacencia de nodos y el otro se basa en la incidencia entre nodos y arcos.

### 1.2.2. Matriz Adyacencia

**Definición 1.3** *Supongamos que  $G = (V, E)$  es un grafo con  $N$  nodos. Supongamos además que los nodos de  $G$  se enumeran de manera arbitraria como  $1, 2, 3, \dots, N$ .*

La matriz de adyacencia  $A$  de  $G$  es la matriz booleana de tamaño  $(N \times N)$  que tiene 1 en la posición  $(i, j)$ , si  $i$  y  $j$  son adyacentes, y tiene un 0 en la posición  $(i, j)$ , si los

nodos  $i$  y  $j$  no lo son. En otras palabras los elementos de la matriz de adyacencia es  $a_{ij}$ , vienen dados por

$$a_{ij} = \begin{cases} 1, & \text{si } (i, j) \in E \\ 0, & \text{si } (i, j) \notin E, \end{cases} \quad (1.1)$$

Un ejemplo de ésta representación es

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Representación por medio de la matriz de adyacencia del grafo de la figura 1.3

Cuando el número de elementos es muy alto, es más eficiente trabajar con listas de arcos a la hora de representar y de trabajar con éste tipo de grafos. La matriz de adyacencia de un grafo simple es simétrica, esto es,  $a_{ij} = a_{ji} \forall i, j = 1, 2, 3, \dots, N$ . Mientras que para un digrafo la matriz es asimétrica, es decir que existe al menos un elemento  $a_{ij} \neq a_{ji}$ . Por otra parte, un grafo que no contiene bucles es aquel donde cada elemento  $a_{ij} = 0 \forall i = 1, 2, 3, \dots, N$ .

### 1.2.3. Conjunto de vecinos

Por razones de rendimiento para representar a los grafos, ISyS, en vez de utilizar matrices de adyacencia, utiliza conjuntos de vecinos incidentes. El conjunto de vecinos incidentes del nodo  $\nu_i$  se define como el conjunto de nodos de los que proviene un arco que incide en  $i$  y viene dado por  $\nu = \{j : (j, i) \in E\}$  los conjuntos de vecinos del grafo de la figura son  $\nu_1 = \{2\}; \nu_2 = \{1, 4\}; \nu_3 = \{3\}; \nu_4 = \{1, 2, 3\}$ .

### 1.3. Redes Complejas o Reales

Según M. Aldana [15], las redes complejas son conjuntos de muchos nodos conectados de forma no trivial que interactúan de alguna forma. Es decir, los sistemas dinámicos que se encuentran sobre un grafo, entonces, ¿qué hay que añadir a la teoría de grafos si éste es un lenguaje poderoso y general?. Según M. Newman, A. Barabasi y D. Watts la ciencia de las redes que ha ido tomando forma durante los últimos años, se distingue de la teoría de grafos al trabajar en redes de tres maneras novedosas: la primera es, centrándose en las propiedades de las redes del mundo real; la segunda tomando en cuenta las preguntas frecuentes que consideran que las redes no son estáticas sino evolucionan en el tiempo de acuerdo con varias reglas dinámicas, y la tercera entender las redes no sólo como objetos topológicos, sino también como el marco en el que están contruidos los sistemas dinámicos [17].

#### 1.3.1. Redes aleatorias de Erdos y Renyi

Los grafos aleatorios investigados por Erdős y Rényi pueden ser considerados como el modelo de redes más fundamental [3, 25]. En la publicación de 1959 ellos introdujeron un modelo para generar grafos aleatorios. Para construir estos grafos consideremos  $N$  nodos sin conectar y distribuidos de forma aleatoria. Podemos imaginar que, en un instante inicial enlazamos dos nodos cualesquiera, y en pasos sucesivos, vamos enlazando aleatoriamente nodos de dos en dos, descartando los nodos ya enlazados, si repetimos este proceso  $M$  veces, al final habremos establecido  $M$  enlaces entre parejas de nodos. Si  $M$  es un valor muy pequeño con respecto al total de nodos  $N$ , muchos de los nodos estarán desconectados, y otros nodos formarán pequeñas islas. En cambio, si  $M$  es grande en comparación con  $N$ , es muy posible que casi todos los nodos estén enlazados entre sí.

También se puede construir una red aleatoria de tipo Erdős-Rényi definiendo  $N$  nodos y una probabilidad  $P$  con la que cada par de nodos están conectados. Este modelo es ampliamente conocido como el modelo Erdős-Rényi (ER)[3]. Las redes

$ER$ , cuando son de tamaño grande, es decir en el límite cuando  $N \rightarrow \infty$ , el número promedio de conexiones de cada nodo  $\langle k \rangle$ , que viene dado por

$$\langle k \rangle = P(N - 1) , \quad (1.2)$$

donde  $\langle k \rangle$ , que se conoce como el grado promedio de la red, diverge si  $P$  es fijo. En cambio, si se escoge  $P$  como

$$P = \frac{\langle k \rangle}{(N - 1)} , \quad (1.3)$$

se puede mantener el grado  $\langle k \rangle$  fijo al aumentar  $N$ .

Para este modelo  $P_k$ , que representa la probabilidad de encontrar un nodo de grado  $k$ , es decir con  $k$  conexiones, sigue una distribución de Poisson, cuya ecuación tiene la forma

$$P_k = \frac{e^{-\langle k \rangle} \langle k \rangle^k}{k!} , \quad (1.4)$$

Un ejemplo cotidiano de red aleatoria lo podemos observar en la figura 1.4, en la que vemos como, luego de colocar una cantidad razonable de grapas separadas dentro de un recipiente y luego volcarlas sobre una mesa, éstas se agrupan de una forma aleatoria.

### 1.3.2. Redes de pequeño mundo de Watts Strogatz

Muchas de las redes del mundo real poseen la propiedad llamada *small world* o pequeño mundo[3]. A pesar de que el conjunto de vecinos de cada elemento es compartido en buena medida por todos sus vecinos muchos nodos pueden alcanzarse desde

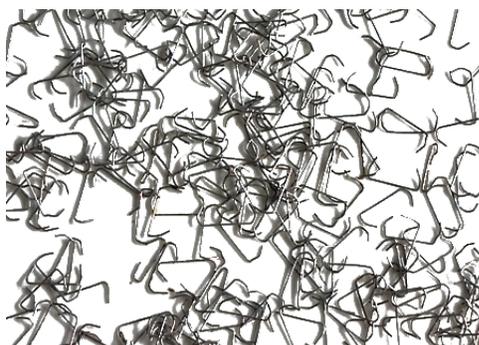


Figura 1.4: Representa una de las tantas redes aleatorias existentes en la naturaleza.[?]

otros vértices a través de un pequeño número de pasos.

Esta característica se encuentra, por ejemplo, en redes sociales, donde todo el mundo puede buscarse a través de las cadenas de vínculos sociales. Esta propiedad se observó por primera vez con los experimentos hechos por Milgram en 1967[35], quien encontró que dos ciudadanos americanos elegidos aleatoriamente están conectados en promedio por seis conocidos.

Otra propiedad de cualquier red tipo pequeño mundo, es la presencia de un gran número de ciclos de tamaño tres [19], medida generalmente con el coeficiente de agrupamiento. Ésto se puede expresar como la probabilidad de que dos nodos  $j$  y  $k$  estén conectados entre sí dado que ambos están conectados a un tercer nodo  $i$  por ejemplo en la red de amigos, si  $B$  y  $C$  son amigos de  $A$ , entonces existe una alta probabilidad que  $B$  y  $C$  sean amigos entre sí, la figura 1.5 muestra una red social con características de pequeño mundo.

Las redes (ER) tienen longitud característica pequeña pero un coeficiente de agrupamiento promedio muy bajo. Por otra parte, las redes regulares con coeficiente de agrupamiento alto son fáciles de construir, pero ellas tienen longitudes características grandes. El modelo más popular para construir redes de pequeño mundo fué desarrollado por Watts y Strogatz [20].

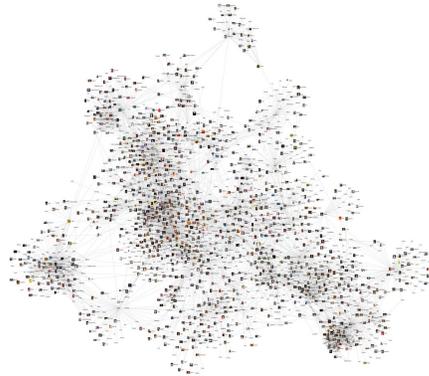


Figura 1.5: Representación de una red social de internet basado en el experimento de S. Milgram, The Small World Problem[23].

Además de proponer un algoritmo para la construcción de redes pequeño mundo, ellos demostraron que éstas redes son comunes dentro de una variedad de rangos reales que van desde los elegantes sistemas neuronales hasta las redes de energía eléctrica.

Para construir una red de pequeño mundo, se comienza con una retícula regular de  $N$  nodos en la que cada vértice está conectado con los  $k$  vecinos más cercanos, donde  $N \gg k \gg \log(N) \gg 1$ . Luego, con probabilidad  $P$  cada arista, se recablea aleatoriamente. Cuando  $P = 0$ , tenemos una estructura ordenada con un gran número de lazos pero grandes distancias y cuando  $P \rightarrow 1$ , la red se convierte en un grafo aleatorio con distancias cortas pero pocos lazos. La distribución de grado para las redes de pequeño mundo son similares a la de redes aleatorias con un pico en  $\langle k \rangle$ .

### 1.3.3. Redes libres de escala de Barabasi y Albert

Luego de la aparición del modelo de Watts y Strogatz, Barabasi y Albert [9] mostraron que la distribución de grado de muchas redes reales se caracterizan por ser desigual. Los nodos de éstas redes no tienen un patron aleatorio de conexiones con grados característicos, tal como ocurre con los modelos *ER* y *WS*. Por el contrario, Barabasi-Albert observaron que algunos nodos están altamente conectados mientras que otros tiene pocas conexiones, lo que expresa la ausencia de un grado característico.

Más específicamente, se ha comprobado que la distribución de grado sigue la ley de potencia dada por

$$P(k) \sim k^{-\gamma}, \quad (1.5)$$

donde  $\gamma$  no es universal, sino que depende del tipo específico de red. Para la mayor parte de los sistemas se encontró que dicho parámetro se encontraba en el rango  $2 < \gamma \leq 3$ . Cuando  $\gamma \leq 2$ , la varianza de la distribución del número de enlaces por nodo es infinita [36].

Estas redes son llamadas redes libres de escala. Una característica siempre presente en ellas es la existencia de *hubs* o nodos que están conectados mediante una fracción significativa del número total de arcos de la red. El modelo de red de Barabási-Albert (BA) se basa en dos componentes básicos: crecimiento y conexión preferencial. La red se genera comenzando con un conjunto de  $m_0$  nodos; posteriormente, en cada paso de la construcción, la red crece con la suma de nuevos nodos, por cada uno de ellos se crean,  $m$  nuevos arcos que van entre los nuevos nodos y algunos nodos ya existentes, estos últimos elegidos siguiendo una norma lineal de conexión preferencial. Es decir, la probabilidad de que un nuevo nodo  $i$  se conecte con el nodo  $j$  existente es proporcional al grado de  $j$ , esto es

$$P((i, j) \in E) = \frac{k_j}{M}, \quad (1.6)$$

donde  $M$  es el número total de arcos que tiene la red al momento de conectar al nodo  $j$ .

Esto hace que los nodos más conectados tengan una alta probabilidad de recibir a los nuevos nodos. En la figura 1.6 podemos ver una de las tantas redes libres de escala, la interacción entre proteínas.

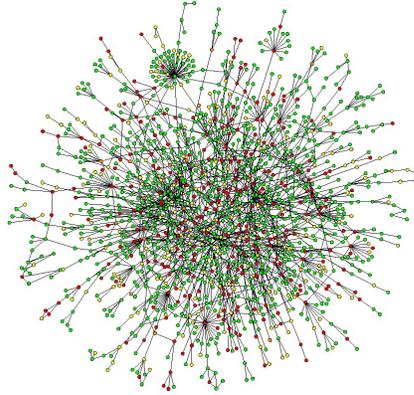


Figura 1.6: Representa la interacción entre proteínas[22]. Ésta red es libre de escala

En el cuadro 1.1 se resumen las características analíticas de algunas medidas básicas para los modelos *ER*, *WS*, y *BA*

Erdo-Renyi	Watts-Strogatz	Barabasi-Albert
$P_k = \frac{e^{-\langle k \rangle} \langle k \rangle^k}{k!}$	$P(k) = \frac{(\sum_{i=1}^{\min(k-K, K)} \binom{K}{i} (1-p)^i p^{K-i} \frac{(pK)^{k-K-i}}{(k-K-i)!})}{e^{-pK}}$	$P(k) \sim k^{-3}$
$\langle k \rangle = p(N - 1)$	$\langle k \rangle = 2Kp$	$\langle k \rangle = 2m$
$C = p$	$C(p) = \frac{3(K-1)}{2(2K-1)} (1-p)^3$	$C \sim N^{-0,75}$
$l \sim \frac{\ln(N)}{\ln(\langle k \rangle)}$	$l(N, p) \sim \begin{cases} p^\tau, & \text{Constante} \\ f(Np^\tau), & \frac{\ln(Np^\tau)}{Np^\tau} \end{cases}$	$l \sim \frac{\text{Log}N}{\text{Log}(\text{Log}N)}$

Cuadro 1.1: Tabla comparativa de la distribución de grados  $P_k$ , el grado promedio de la red  $\langle k \rangle$ , el coeficiente de agrupamiento  $C$  y la longitud carasterística  $l$ ; para redes *ER*, *WS*, y *BA*.

\*En la función  $f(u)=\text{constante}$ , si  $u \ll 1$  o  $f(u) = \frac{\ln(u)}{u}$  si  $u \gg 1$

*“Saber no es suficiente, debemos aplicar.  
Desear no es suficiente, debemos hacer.”*

*Johann W. Von Goethe*

## Capítulo 2

# Diseño e implementación del modulo de medidas estadísticas de ISyS

En el capítulo anterior desarrollamos algunos conceptos que nos van a ser de utilidad al momento de implementar las medidas para la caracterización de redes reales o complejas. En este capítulo nos concentraremos en el diseño e implementación del nuevo módulo contentivo de las diferentes medidas seleccionadas para caracterizar a las redes siempre en consonancia con las normas que requiere ISyS [1].

Primeramente, para la creación del módulo se requiere tener claras las reglas de sintáxis que son las que mantienen la compatibilidad de los tres módulos. Por cada nueva función a implementar se debe crear un archivo en lenguaje C e incluir la declaración de la función en el archivo de cabecera.

A manera de ejemplo, en el código 1 se muestra un extracto del archivo de cabecera. En él se muestra la declaración y documentción de la función `Avedistance`, cuya

---

**Código 1** Extracto del archivo de cabecera.
 

---

```

/* -----
A_. NOMBRE
    Avedistance
B_. SINOPSIS
    "hebert.h"
    double Avedistance(void)
    Ademas utiliza el la funciones:
    free(),
    calloc(),
    sizeof().
    Asi como los Registros:
    node[].n_nei
    node[].nb[]
C_. DESCRIPCION
    Esta funcion calcula la el promedio de las distancias
    mas cortas entre cada par de vertices.
D_. VALOR QUE DEVUELVE
    En caso de exito, retorna un valor de tipo double con el
    promedio de las distancias mas cortas entre cada par de vertices.
    En caso de de que el grafo no sea conexo devuelve el valor nn+1 y
    la siguiente advertencia:
    a -.warning_message(1), "que significa que existe al menos un
    par de vertices no conectados."
    En caso de no tener exito devuelve la siguiente error:
    a -.error_message(2): indicando que no se agoto la memoria.
E_. VER ADEMAS
    "isys.h"
*/
extern double Avedistance(void);

```

---

implementación se discute en la sección 2.1.2, junto con su documentación. La documentación es un comentario en lenguaje C y consta generalmente de 5 partes: `NOMBRE`, `SINOPSIS`, `DESCRIPCION`, `VALOR QUE DEVUELVE` y `VER ADEMAS`. Luego del comentario donde se encuentra la documentación, encontramos la declaración de la función, en este caso `Avedistance`. Este mismo esquema se repite para cada una de las funciones implementadas.

Se decidió incluir la documentación en el archivo de cabecera ya, al momento de distribuir la herramienta ISyS, estos archivos conservan su formato de texto y por lo tanto pueden ser leídos por el usuario.

## 2.1. Medidas

Antes de presentar las diferentes medidas que se implementaron es importante mencionar que para ISyS los grafos son, en general, digrafos sin arcos duplicados, es decir, que en el conjunto de vértices  $E$  cada par ordenado sólo está presente una vez, y además las autoconexiones no están permitidas, esto es, que los dos elementos de cada par ordenado de  $E$  deben ser diferentes.

### 2.1.1. Grado

En el caso de grafos simples de tamaño  $N$ , el grado del nodo  $i$ , denotado por  $k_i$ , se define como el número de arcos conectados con  $i$ , que viene dado por

$$k_i = \sum_{j=1}^N a_{ij} = \sum_{j=1}^N a_{ji} , \quad (2.1)$$

donde  $a_{ij}$ , es el elemento ubicado en la fila  $i$  y columna  $j$  de la matriz de adyacencia del grafo definida en la sección 1.2.2. Pero para los digrafos, el grado de los arcos entrantes de un nodo  $i$ , definidos por

$$k_i^{in} = \sum_{j=1}^N a_{ji} , \quad (2.2)$$

no necesariamente coincide con el grado de los arcos salientes, dado por

$$k_i^{out} = \sum_{j=1}^N a_{ij} . \quad (2.3)$$

Es por esta razón que se implementaron ambas medidas. El código 2 muestra la implementación de  $k_i^{in}$  mientras que el código 3 muestra la de  $k_i^{out}$ .

---

**Código 2** Función para medir el  $k_i^{in}$  de un nodo.

---

```
#include "isys.h"
long kiIn(i)
long i;
{
    return node[i].n_nei;
}
```

---



---

**Código 3** Función para medir el  $k_i^{out}$  de un nodo.

---

```
#include "isys.h"
long kiOut(i)
long i;
{
    long m,n;
    long k;
    k=0;
    for (n=0;n<nn;n++){
        for (m=0;m<node[n].n_nei;m++){
            if(node[n].nb[m]==i){
                k++;
                break;
            }
        }
    }
    return k;
}
```

---

Generalmente, los usuarios de ISyS están interesados en el grado de la red definido como el promedio de los grados de sus nodos. En el caso de arcos entrantes este promedio viene dado por

$$\langle k^{in} \rangle = \frac{1}{N} \sum_{i=1}^N k_i^{in}, \quad (2.4)$$

donde su respectiva desviación estandar es

$$\sigma(k^{in}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (k_i^{in})^2 - \langle k^{in} \rangle^2}. \quad (2.5)$$

La función `kIn`, mostrada en el código 4, implementa las ecuaciones (2.4) y (2.5) para  $k_i^{in}$  y adicionalmente retorna su valor máximo y mínimo, todo ello en un arreglo de 4 `doubles`.

La función `kOut`, mostrada en el código 5, implementa las ecuaciones (2.4) y (2.5) y adicionalmente retorna el valor máximo y mínimo de los  $k_i^{out}$ , todo ello en un arreglo de 4 `doubles`.

Para el cálculo de la distribución de probabilidades de los grados entrantes se determina la probabilidad de que un nodo cualquiera tenga una grado  $k$  mediante la expresión

$$P_k^{in} = \frac{\sum_{i=1}^N \delta_{kk_i^{in}}}{N}, \quad (2.6)$$

donde,  $\delta$  es la delta de Dirac y  $N$  número total de nodos de la red. Cada una de las probabilidades se almacena en un arreglo, tal como lo podemos ver en el código 6.

---

**Código 4** Función para medir el  $\langle k_i^{in} \rangle$  de la red.
 

---

```

#include "isys.h"
#include <math.h>
double kIn(k)
double *k;
{
    double s=0,
           ss=0;
    long i, ki;
    k[2] = nn; // kMin
    k[3] = 0; // kMax
    for (i=0;i<nn;i++){
        ki = kiIn(i);
        s += ki;
        ss += ki*ki;
        if(ki>k[3]) k[3] = ki; // kMax
        if(ki<k[2]) k[2] = ki; // kMin
    }
    k[0] = s/nn; // <k>
    k[1] = ss/nn - k[0]*k[0]; // Desv. Estandar de k
    if(k[1]>0) k[1] = sqrt(k[1]); // Desv. Estandar de k
    else k[1] = 0; // Desv. Estandar de k
    return k[0];
}

```

---



---

**Código 5** Función para medir el  $\langle k_i^{out} \rangle$  de la red.
 

---

```

#include "isys.h"
#include <math.h>
double kIn(k)
double *k;
{
    double s=0,
           ss=0;
    long i, ki;
    k[2] = nn; // kMin
    k[3] = 0; // kMax
    for (i=0;i<nn;i++){
        ki = kiOut(i);
        s += ki;
        ss += ki*ki;
        if(ki>k[3]) k[3] = ki; // kMax
        if(ki<k[2]) k[2] = ki; // kMin
    }
    k[0] = s/nn; // <k>
    k[1] = ss/nn - k[0]*k[0]; // Desv. Estandar de k
    if(k[1]>0) k[1] = sqrt(k[1]); // Desv. Estandar de k
    else k[1] = 0; // Desv. Estandar de k
    return k[0];
}

```

---

---

**Código 6** Distribución de los arcos entrantes
 

---

```

void kindistribution(){
    long n; double entro,ff,*w; long double s = 0; long double ss = 0;
    long kMin = nn, kMax = 0, *h, nHisto;
    for(n=0; n<nn; n++){
        if(node[n].n_nei > kMax) kMax = node[n].n_nei;
        if(node[n].n_nei < kMin) kMin = node[n].n_nei;
    } nHisto = kMax-kMin+1;
    if((h = (long*)calloc(nHisto,sizeof(long))) == NULL) error_message(2);
    if((w = (double*)calloc(nHisto,sizeof(double))) == NULL) error_message(2);
    for(n=0; n<nHisto; n++){
        h[n] = 0;
        w[n] = 0;}
    for(n=0; n<nn; n++){
        s += node[n].n_nei;
        ss += node[n].n_nei*node[n].n_nei;
        h[node[n].n_nei-kMin]++;}
    ff = 0;
    for(n=0; n<nHisto; n++)
        printf("Histograma del nodo %ld con frecuencia = %ld\n",n+kMin,h[n]);
}

```

---

De forma equivalente, para el cálculo de la distribución de los grados salientes se implementó en el código 7 la siguiente ecuación

$$P(k) = \frac{\sum_{i=1}^N \delta_{kk_i^{out}}}{N}, \quad (2.7)$$

La entropía es un concepto clave en termodinámica, la mecánica estadística [26] y teoría de la información [27]. Cuenta con importantes implicaciones físicas, para el orden y la información en un sistema. En teoría de la información, la entropía describe que tanta aleatoriedad está presente en una señal o en un evento al azar [28]. Éste concepto puede fácilmente aplicarse a las redes complejas.

En particular, la entropía de la distribución de grado ofrece una medición promedio de la heterogeneidad de la red, que puede definirse por

---

**Código 7** Distribución de los arcos salientes
 

---

```

void koutdistribution(){
  long n,k; long double s = 0; long double ss = 0;
  long kMin = nn, kMax = 0, *kout, *h, nHisto;
  if((kout = (long*)calloc(nn,sizeof(long))) == NULL) error_message(2);
  for(n=0; n<nn; n++)
    for(k=0; k<node[n].n_nei; k++)
      kout[node[n].nb[k]]++;
  for(n=0; n<nn; n++){
    if(kout[n] > kMax) kMax = kout[n];
    if(kout[n] < kMin) kMin = kout[n]; }
  nHisto = kMax-kMin+1;
  if((h = (long*)calloc(nHisto,sizeof(long))) == NULL) error_message(2);
  for(n=0; n<nHisto; n++)
    h[n] = 0;
  for(n=0; n<nn; n++) {
    s += kout[n];
    ss += kout[n]*kout[n];
    h[kout[n]-kMin]++; }
  if(ss<0)
    printf("kOutBar\t%Lg\t0\n",s/nn);
  else
    for(n=0; n<nHisto; n++)
      printf("Histograma del nodo %ld con frecuencia = %ld\n",n+kMin,h[n]);
}

```

---

$$H = - \sum_k ((P(k)) * \log(P(k))) , \quad (2.8)$$

donde  $P(k)$  es la probabilidad definida en las ecuaciones (2, 6) y (2, 7), para el cálculo de la entropía de los arcos salientes y entrantes, respectivamente, tal como podemos observar en los códigos 8, 9.

El valor máximo de la entropía se obtiene de una distribución de grado uniforme y el valor mínimo es cuando  $Hmin = 0$ , y se alcanza cuando todos los vértices tienen el mismo grado.

La entropía de redes, se ha relacionado con la robustez de las redes, es decir, su resistencia a los ataques [29], y la contribución de los vértices de la entropía de la red, se correlaciona con la letalidad en las interacciones de redes de proteínas[30].

Además se puede aplicar para los  $k_i^{in}$  y  $k_i^{out}$  de forma independiente, dado que podemos realizar las comparaciones del nivel de orden de los grados promedios entrantes y

salientes de la red independientemente del tipo de red.

---

### Código 8 Entropía de la red

---

```

void entropy(){
    long n;    double entro,ff,*w; long double s = 0;
    long double ss = 0; long kMin = nn,kMax = 0, *h, nHisto;
    for(n=0; n<nn; n++){
        if(node[n].n_nei > kMax) kMax = node[n].n_nei;
        if(node[n].n_nei < kMin) kMin = node[n].n_nei;  }
    nHisto = kMax-kMin+1;
    if((h = (long*)calloc(nHisto,sizeof(long))) == NULL)    error_message(2);
    if((w = (double*)calloc(nHisto,sizeof(double))) == NULL)    error_message(2);
    for(n=0; n<nHisto; n++) {
        h[n] = 0;    w[n] = 0; }
    for(n=0; n<nn; n++){
        s += node[n].n_nei;
        ss += node[n].n_nei*node[n].n_nei;
        h[node[n].n_nei-kMin]++;  }
    ff = 0;
    for(n=0; n<nHisto; n++){
        ff += ((double)h[n])/nn;
        w[n] = ((double)h[n])/nn; }
    entro = 0;
    for(n=0; n<nHisto; n++){
        if(w[n] > 0){
            entro += w[n]*log10(w[n]);  }  }
    printf(" Entropia  \t%lf  \n\n",(-1.0)*(entro));
}

```

---

### 2.1.2. Distancia promedio

Una medida muy utilizada al momento de caracterizar a una red es su logitud característica, que viene dada por

$$l = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij} , \quad (2.9)$$

donde  $d_{ij}$  es la distancia mínima medida en arcos, que existe entre el nodo  $i$  y el nodo  $j$ . Al implementar ésta definición surge el problema de que diverge si un par de vértices no están conectados, es decir que el  $d_{ij} \rightarrow \infty$ . Hemos resuelto éste problema,

---

**Código 9** Distancia Promedio

---

```
double Avedistance(void)
{
    double acum;    long *length, n, m, k, neiNode;    char changes;
    if((length = (long*)calloc(nn,sizeof(long))) == NULL)    error_message(2);
    acum=0;
    for(n=0; n<nn; n++){
        for(m=0; m<nn; m++){
            length[m] = nn+1;
            length[n] = 0;
            do{
                changes = false;
                for(m=0; m<nn; m++){
                    if(length[m] <= nn){
                        for(k=0; k<node[m].n_nei; k++){
                            neiNode = node[m].nb[k];
                            if(length[neiNode] > length[m]+1){
                                length[neiNode] = length[m]+1;
                                changes = true;}    }    }
            }while(changes);
            for(m=0; m<nn; m++){
                if(length[m] > nn){ warning_message(1);
                return nn+1;}
                acum += length[m];}    }
    free(length);
    printf("La Distancia Promedio\t%f\v\n", (float)acum/(nn*(nn-1)));
    return (float)acum/(nn*(nn-1));
}
```

---

emitiendo un mensaje de advertencia `warning_message(1)` y retorna el valor `nn+1`, tal como se puede observar en el código 9.

### 2.1.3. Eficiencia

Latora y Marchiori [6] propusieron medir la eficiencia global de la red como solución a la divergencia de la longitud características cuando un par de nodos no se encuentran conectados. La eficiencia global viene dada por

$$E = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}}, \quad (2.10)$$

donde  $N$  es el número de nodos de la red y  $d_{ij}$  es la distancia mínima entre el nodo  $i$  y el nodo  $j$ . Dicha medición cuantifica la eficiencia de la red en el envío de información entre los nodos.

El código 10 muestra la implementación de la ecuación 2.10. Notese que es muy similar al código 9 que implementa la medición de la longitud característica discutida en la sección 2.1.2

Por otra parte, V.Gol'dshtein [7] propuso el cálculo de la eficiencia sin un nodo, como el cálculo de la eficiencia global de la red sin tomar en cuenta el vértice desde cual se está realizando la medición. Es decir, estamos alterando la red original y realizando el cálculo de la medida de acuerdo a la siguiente expresión:

$$E_n = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}^n}, \quad (2.11)$$

donde  $n$  es el nodo que no se toma en cuenta,  $(i, j) \in \{1, 2, 3, 4, \dots, N\} - \{n\}$  y  $N$  es el número total de nodos.

---

**Código 10** Eficiencia global
 

---

```

double globalefficiency()
{
    double acum,r;    long *length, n, m, k, neiNode;    char changes;
    if((length = (long*)calloc(nn,sizeof(long))) == NULL) error_message(2);
    acum=0;
    areNeighbors(0,1);
    for(n=0; n<nn; n++){
        for(m=0; m<nn; m++){
            length[m] = nn+1;
            length[n] = 0;
            do{ changes = false;
                for(m=0; m<nn; m++){
                    if(length[m] <= nn){
                        for(k=0; k<node[m].n_nei; k++){
                            neiNode = node[m].nb[k];
                            if(length[neiNode] > length[m]+1){
                                length[neiNode] = length[m]+1;
                                changes = true;}    }    }
                    }while(changes);
                for(m=0; m<nn; m++){
                    if(length[m] > nn){warning_message(1);
                        return nn+1;}
                }
            if ((n!=m)&&(length[m]!=0))
                acum += ((1.0)/length[m]); }    }
            free(length);
            r= acum/(nn*(nn-1));
            return (r);
        }
    }

```

---

La ecuación 2.11 está implementada en el código 11.

---

**Código 11** Eficiencia sin el nodo  $i$

---

```

void eficienciaI(double *q)
{ double acum; long *length, n, m, k, neiNode,i; char changes, isJoined;
  if((length = (long*)calloc(nn,sizeof(long))) == NULL) error_message(2);
  for(i=0; i<nn; i++){
    acum=0;
    isJoined=true;
    for(n=0; n<nn && isJoined; n++){
      if(n!=i){
        for(m=0; m<nn; m++){
          length[m] = nn+1;
          length[n]=0;
          do{
            changes = false;
            for(m=0; m<nn; m++){
              if((m!=i) && (length[m] <= nn)){
                for(k=0; k<node[m].n_nei; k++){
                  neiNode = node[m].nb[k];
                if(neiNode!=i)
                  if(length[neiNode] > length[m]+1){
                    length[neiNode] = length[m]+1;
                    changes = true; } } } }
              }while(changes);
            for(m=0; m<nn; m++){
              if(m!=i){
                if(length[m] > nn){
                  acum=-nn*nn;
                  isJoined=false;
                  break; }
                if((n!=m)&&(length[m]!=0))
                  acum += ((1.0)/length[m]); } } } }
            q[i]=acum/(nn*nn); }
            free(length);
          }
        }
      }
    }
  }

```

---

Asumiendo que la eficiencia para el envío de información entre 2 nodos  $i$  y  $j$  es proporcional al recíproco de la distancia que los separa. Entonces decimos que el recíproco de la eficiencia global es la media armónica de la distancia geodésica, denotada por

$$h = \frac{1}{E}, \quad (2.12)$$

donde  $E$  es la eficiencia global de la red. La implementación de la media armónica de las distancia geodesica se muestra el código 12, siendo ésta otra alternativa para medir promedios de distancias entre los nodos.

---

**Código 12** Media Armónica

---

```
double harmonicmean()
{
double p;
p = (1.0)/globalefficiency();
return p;
}
```

---

#### 2.1.4. Vulnerabilidad

Para saber cuales componentes de la red son cruciales se determina la vulnerabilidad. Intuitivamente, se puede pensar que siempre los vértices críticos de una red son los hubs (los nodos con alto grado), sin embargo, hay situaciones en las cuales no necesariamente ésto es así. Por ejemplo, todos los nodos intermedios de una red tipo árbol binario tienen grado  $k = 3$ , la raíz tiene grado  $k = 2$  y las hojas tienen grado  $k = 1$  por lo tanto allí no hay *hubs*. Pero la desconexión de un nodo intermedio o de la raíz, ésta tiene un gran impacto en las propiedades del grafo.

Un camino para encontrar los componentes críticos de una red es observando los nodos más vulnerables. Si la eficiencia global de la red  $E$  y la eficiencia de la red al sustraerle uno a uno sus nodos  $E_i$ , 2.1.3 la vulnerabilidad de cada nodo [7] se define como

$$V_i = \frac{E - E_i}{E}, \quad (2.13)$$

Generalmente el orden de la distribución de nodos con respecto a su vulnerabilidad  $V_i$

está relacionada con la jerarquía de la red, de ésta manera, el vértice más vulnerable (crítico) ocupa una posición muy alta en la jerarquía de la red.

La implementación de la ecuación 2.13, se muestra en el código 13.

---

### Código 13 Vulnerabilidad

---

```
void vulnerability(double* v,double* q)
{
    double eg;
    long i;
    eg = globalefficiency();
    for(i=0; i<nn; i++){
        v[i]=((eg-q[i])/eg);
    }
    for (i=0; i<nn; i++){
        if (q[i]<0)
            v[i]=0;
    }
}
```

---

Al igual que al medir los grados, los usuarios de ISyS muchas veces están interesados en calcular la vulnerabilidad promedio de la red definida como

$$\langle V \rangle = \max_i V_i, \quad (2.14)$$

con su respectiva desviación estandar.

$$\sigma(V) = \sqrt{\frac{1}{N} \sum_{i=1}^N (V)^2 - \langle V \rangle^2}. \quad (2.15)$$

En el código 13, se muestra la función que implementa las ecuaciones (2.14) y (2.15) y además devuelve la vulnerabilidad máxima, mínima, de todo en un arreglo de 4 doubles.

y por medio de un método de ordenamiento iterativo, se obtiene el valor asociado a la vulnerabilidad máxima.

---

**Código 14** Vulnerabilidad promedio de la red
 

---

```

double MaxVulnerability(double* v)
{
  int i;
  double vMax = v[0];
  for (i=1; i<nn; i++)
    if (v[i]>vMax)
      vMax=v[i];
  return vMax;
}

```

---

### 2.1.5. Clustering o Agrupamiento

Una forma de caracterizar la localidad en una red es gracias a la presencia de bucles de orden 3, medida mediante el coeficiente de clustering o agrupamiento [3]. Se utilizan con frecuencia dos coeficientes diferentes de agrupamiento. El primero, también conocido como transitividad [33], se basa en la siguiente definición para redes no dirigidas no ponderadas dados dos nodos deben estar adyacentes a otro nodo (el nodo central). Por lo tanto tenemos que:

$$N_{\Delta} = \sum_{k>j>i} a_{ij}a_{ik}a_{jk} , \quad (2.16)$$

$$N_3 = \sum_{k>j>i} (a_{ij}a_{ik} + a_{ji}a_{kj} + a_{ki}a_{kj}) , \quad (2.17)$$

donde los  $a_{ij}$  son los elementos de la matriz de adyacencia  $A$  y la suma se toma sobre todos los triples de vértices distintos  $i, j, k$ , y sólo una vez, entonces

$$C = \frac{3 * N_{\Delta}}{N_3} , \quad (2.18)$$

donde  $N_{\Delta}$  es el número de triángulos en la red y  $N_3$  es el número de tripletas conectadas. El factor (3) explica el hecho de que cada triángulo puede ser visto como un conjunto de tres tripletas distintas que están conectadas, una con cada uno de los vértices como vértice central, y asegura que  $0 \leq C \leq 1$ . Un triángulo es un conjunto de tres vértices con los bordes entre cada par de vértices, una tripeleta conectada es un conjunto de tres vértices donde cada vértice puede llegar a los otros (directa o indirectamente), es decir,

También es posible definir el coeficiente de agrupamiento de un vértice dado  $i$  [20] como:

$$N_{\Delta}(i) = \sum_{k>j} a_{ij}a_{ik}a_{jk} \quad (2.19)$$

$$N_3(i) = \sum_{k>j} a_{ij}a_{ik} \quad (2.20)$$

$$C_i = \frac{N_{\Delta}(i)}{N_3(i)} \quad (2.21)$$

donde  $N_{\Delta}(i)$  es el número de triángulos alrededor del vértice  $i$  y  $N_3(i)$  es el número de triples conectados con  $i$  como el vértice central, si el número de vecinos del vértice  $i$ , a continuación:  $N_3 = (k_i * (k_i - 1))/2$ , donde  $N_{\Delta}(i)$  cuenta el número de aristas entre los vecinos de  $i$ , en representación del número de bordes entre los vecinos de  $i$  como  $l_i$ , la ecuación puede ser reescrita como:

$$C_i = \frac{2 * l_i}{k_i * (k_i - 1)}. \quad (2.22)$$

Ésta ecuación permite cuantificar el valor del coeficiente clustering o de apilamiento para grafos dirigidos o digrafos, y además contiene como caso particular el cálculo del coeficiente para grafos no dirigidos, en donde se verifica de forma iterativa la conexión directa de la tripleta desde cada nodo, tal como se puede visualizar en el código 15.

---

**Código 15** Coeficiente de clustering para grafos dirigidos

---

```
double Newcluster(void)
{
long    i, j, k, n, m, li, ki; float c;
j=0; ki=0; c=0;
  for(i=0; i<nn; i++){
    li=0;
    ki=node[i].n_nei;
    if(ki>1){
      for(n=0; n<ki; n++){
j=node[i].nb[n];
if(j!=i)
  for(m=0; m<ki; m++){
k=node[i].nb[m];
if ((k!=j)&&(k!=j)){
  if(isNeighborOf(k,j))
li++;
  if(isNeighborOf(j,k))
li++; } } }
c += li/(ki*2*(ki-1.0)); } }
  c /= nn;
printf("\n\nEl Coeficiente de Apilamiento Promedio es%lf\t\n",c);
}
```

---

### 2.1.5.1. Espectro de autovalores

El espectro de una red corresponde al conjunto de valores propios  $\lambda_i = 1, 2, 3, 4, 5, 6 \dots N$  de su matriz adyacencia  $A$ . La densidad espectral de la red se define como [31], [32]

$$\rho(\lambda) = \frac{1}{N} \sum_i \delta(\lambda - \lambda_i) \quad (2.23)$$

donde  $\delta(x)$  es la función delta de Dirac y  $\rho$  se aproxima a una función continua como  $N \rightarrow \infty$ , por ejemplo, para redes de tipo Erdos-Rényi, si  $P$  es constante cuando

$N \rightarrow \infty$ , entonces,  $\rho(\lambda)$  converge a un semicírculo [31]. Además los valores propios se pueden utilizar para calcular los momentos  $l_h$ .

---

### Código 16 Autovalores

---

```

autovalores(){
    double A[nn][nn], lr[nn], li[nn], DUMMY[1][nn], WORK[nn];
    double* AT; /* for transformed matrix */
    int i, j, ok, c1, c2, c3;
    char c4;
    if((AT=(double*)calloc(nn*nn,sizeof(double)))==NULL)error_message(2);
    for (i=0; i<nn; i++){
        for(j=0; j<nn; j++){
            AT[(j+nn*i)]=isNeighborOf(i,j); } }
    c1=nn;    c2=3*nn;    c3=1; c4='N';
    dgeev_(&c4,&c4,&c1,AT,&c1,lr,li,DUMMY,&c3,DUMMY,&c3,WORK,&c2,&ok);
    if (ok==0){
        for (i=0; i<nn; i++){
            printf("%lf\t%lf\n", lr[i],li[i]); } }
    else printf("An error occured");
    free(AT);
}

```

---

*“Lo que puedes hacer, o has soñado que podrías hacer,  
debes comenzar. La osadía lleva en sí, genio, poder y magia.”*

*Goethe*

## Capítulo 3

# Resultados

En este capítulo presentaremos algunas medidas relacionadas con grafos de tipo Erdős-Rényi, Barabási-Albert, Watts-Strogatz y regulares con la finalidad de ilustrar como se utilizan las nuevas medidas implementadas. En primer lugar mostraremos la relación entre el tamaño de la red y el tiempo que requieren las medidas. Luego en la sección 3.2 se presentan los resultados más ilustrativos de las medidas tomadas sobre los diferentes tipos de redes.

### 3.1. Tiempo de ejecución.

Para cada medida a implementar se creó un archivo en lenguaje C (ver código 17) con el código fuente de la función `statistic`, la cual llama a la función a evaluar. El ejemplo del código 17 permite determinar el tiempo de ejecución de la función `Avedistance`.

Para la compilación y creación del ejecutable se requiere un archivo de nombre `isys.cf` como el que se muestra en el código 18. Notese que el archivo `isys.cf`, incluye a `Avedistance.o` que es el archivo objeto que implementa la función `Avedistance`.

---

**Código 17** Mide el tiempo de ejecución de la longitud característica.

---

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include "isys.h"
#include "hebert.h"
#include <time.h>
#include <sys/time.h>
#include <math.h>

int timeval_subtract (result, x, y)
struct timeval *result, *x, *y;
{
    if (x->tv_usec < y->tv_usec) {
        int nsec = (y->tv_usec - x->tv_usec) / 1000000 + 1;
        y->tv_usec -= 1000000 * nsec;
        y->tv_sec += nsec;
    }
    if (x->tv_usec - y->tv_usec > 1000000) {
        int nsec = (y->tv_usec - x->tv_usec) / 1000000;
        y->tv_usec += 1000000 * nsec;
        y->tv_sec -= nsec;
    }
    result->tv_sec = x->tv_sec - y->tv_sec;
    result->tv_usec = x->tv_usec - y->tv_usec;
    return x->tv_sec < y->tv_sec;
}

void statistics(){
    struct timeval inicio, fin, resta;
    gettimeofday(&inicio,0);
    Avedistance();
    gettimeofday(&fin,0);
    timeval_subtract(&resta,&fin,&inicio);
    printf("Time\t%ld.%ld\n",resta.tv_sec,resta.tv_usec);
}
```

---

---

**Código 18** Archivo para la construcción del modelo.
 

---

```

PROGRAM= demo
SRC= salida.c
ISyS_HOME= /opt/ISyS
X11_DIR= /usr/lib
LATTICE = $(ISyS_HOME)/lib/Lattices/smallworld.o
DYNAMIC = $(ISyS_HOME)/lib/TestLattice/scan.o
NETDYNAMIC = $(ISyS_HOME)/lib/NetDynamics/static.o
XDISPLAY = $(ISyS_HOME)/lib/Display/nopgplot.o \
$(ISyS_HOME)/lib/Display/color.o
STATISTIC = ../lib/Avedistance.o
CF_OBJS = $(LATTICE) \
$(DYNAMIC) \
$(NETDYNAMIC) \
$(STATISTIC) \
$(XDISPLAY)
CC= cc
F77= gfortran
CFLAGS= -O2 -W -Wall -Wtraditional
FFLAGS= -O2 -Wall
CF_INCL= -I../include
CF_LIBS= -llapack

```

---

Para las mediciones del tiempo de ejecución se seleccionaron redes conexas cuyos tamaños variaron desde  $N = 128$  duplicándose hasta  $N = 131072$ , con grado  $k = 10$  para todas las redes a excepción de las redes Erdö-Renyi (ER), en las que esta cantidad va creciendo a medida que crece el tamaño de la red, y así garantizar que no hayan nodos aislados. Para las redes pequeño mundo (WS) y Regulares la dimensión  $d = 1$ . La probabilidad de asimetría en las redes (ER)  $P = 0$  y la probabilidad de recableado para las redes WS es  $P = 0,01$ .

Todas las medidas se tomaron en un computador ADM Atlon 64 X2 de 2.8 GHz con 4Gb de memoria Ram con bus HyperTransport de 1000 MHz. El sistema operativo utilizado fué linux con kernel 2.6.24. Las fluctuaciones del tiempo de ejecución de las medidas se deben a que el procesador de la máquina en el momento de la ejecución, no estaba dedicado exclusivamente a esta actividad.

Las figuras 3.1 (a) y (b) muestran el tiempo de ejecución de las medidas  $k_{in}$  y  $k_{out}$  en función del tamaño de la red para los cuatro tipos de redes utilizadas en esta tesis. El tiempo de ejecución de las medidas  $k_{in}$  y  $k_{out}$ , implementadas en la sección 2.1.1,

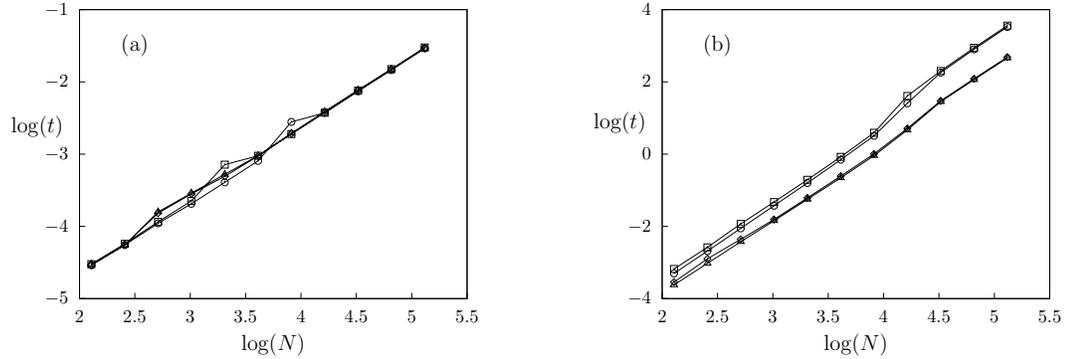


Figura 3.1: Tiempos de ejecución en función del tamaño de la red para las medidas (a)  $k_{in}$  y (b)  $k_{out}$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos)

sigue la relación funcional de una ley de potencia  $t \sim N^{\alpha_k}$ . donde numéricamente se determinó que

$$\begin{cases} \alpha_{k_{in}} & 0,86 \pm 0,02, \text{ para las cuatro redes} \\ \alpha_{k_{out}} & 0,96 \pm 0,01, \text{ para las cuatro redes} \end{cases} \quad (3.1)$$

El tiempo de ejecución de la Entropía  $H_{in}$ , implementada también en la sección 2.1.1, en función del tamaño de la red  $N$ , se muestra en la figura 3.2. Notese que, a

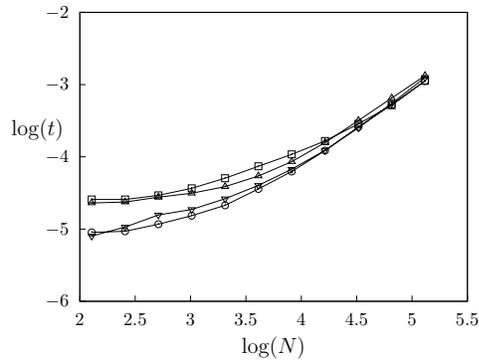


Figura 3.2: Tiempos de ejecución en función del tamaño de la red para las medidas  $H_{in}$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos).

medida que aumenta el tamaño de la red los tiempos de ejecución convergen para los cuatro tipos de redes.

Al igual que estas tres medidas, el resto de las medidas relacionadas con los grados de la red, requieren tiempos pequeños de ejecución que escalan linealmente con el tamaño de la red.

Para el caso del cálculo de la distancia promedio  $L$ , el tiempo de ejecución en función del tamaño de la red.

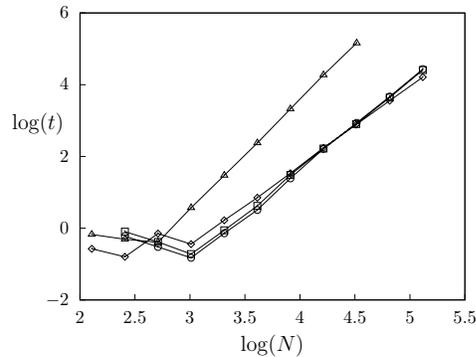


Figura 3.3: Tiempos de ejecución en función del tamaño de la red para la Longitud característica  $L$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos).

Para tamaños grandes de la red, el tiempo de ejecución de la longitud característica  $L$ , implementada en la sección 2.1.2, va como  $t \sim N^{\alpha_L}$ , con un exponente diferente para cada tipo de red

$$\alpha_L = \begin{cases} 2,45 \pm 0,03, & \text{para la red BA} \\ 2,53 \pm 0,04, & \text{para la red ER} \\ 2,22 \pm 0,01, & \text{para la red WS} \\ 3,07 \pm 0,02, & \text{para la red Regular} \end{cases} \quad (3.2)$$

Cabe destacar que para la red regular solamente se llegó a  $N = 32768$  debido a que para ese tamaño el tiempo de ejecución requerido es superior a las 40 horas y su para esta red  $\alpha = 3,07$ , por lo que el tiempo estimado al duplicar el tamaño de la red

sería de 320 aproximadamente (13días).

Las figuras 3.4 muestra el tiempo de ejecución de las medidas del coeficiente de agrupamiento  $C$  en función del tamaño de la red para los cuatro tipos de redes utilizadas.

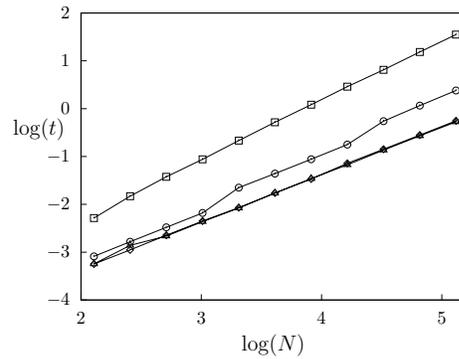


Figura 3.4: Tiempos de ejecución en función del tamaño de la red para la Longitud característica  $L$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos).

El tiempo de ejecución de las medidas del coeficiente de agrupamiento, implementada en la sección 2.1.5, sigue la relación funcional de una ley de potencia  $t \sim N^{\alpha_C}$  donde numéricamente se determinó que

$$\alpha_L = \begin{cases} 1,244 \pm 0,006, & \text{para la red BA} \\ 1,10 \pm 0,02, & \text{para la red ER} \\ 0,995 \pm 0,003, & \text{para la red WS} \\ 0,97 \pm 0,01, & \text{para la red Regular} \end{cases} \quad (3.3)$$

Para el caso del cálculo de la vulnerabilidad máxima  $V_{max}$ , el tiempo de ejecución en función del tamaño de la red se muestra en la figura 3.5.

Para tamaños grandes de la red, el tiempo de ejecución de la vulnerabilidad máxima  $V_{max}$   $L$ , implementada en la sección 2.1.4, va como  $t \sim N^{\alpha_V}$ , con un exponente

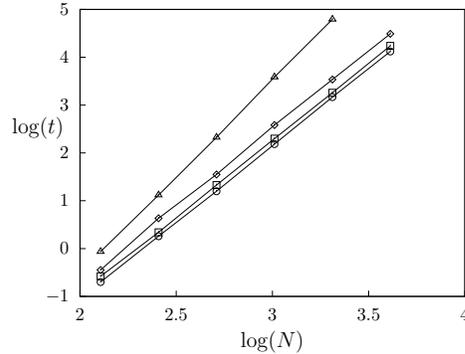


Figura 3.5: Tiempos de ejecución en función del tamaño de la red para la vulnerabilidad máxima  $V_{max}$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos).

diferente para cada tipo de red

$$\alpha_L = \begin{cases} 3,20 \pm 0,01, & \text{para la red BA} \\ 3,20 \pm 0,01, & \text{para la red ER} \\ 3,26 \pm 0,03, & \text{para la red WS} \\ 4,04 \pm 0,02, & \text{para la red Regular} \end{cases} \quad (3.4)$$

Cabe destacar que para la vulnerabilidad máxima  $V_{max}$   $L$  solamente se llegó a  $N = 4096$  debido a que para ese tamaño el tiempo de ejecución requerido es superior a las 10 horas y su para esta red  $\alpha = 4,07$ , por lo que el tiempo estimado al duplicar el tamaño de la red sería de 320 aproximadamente (13 días).

El tiempo de ejecución de los autovalores  $\lambda$ , implementada también en la sección 2.1.5.1, en función del tamaño de la red  $N$ , se muestra en la figura 3.6. Notese que, a medida que aumenta el tamaño de la red los tiempos no dependen del tipo de red. De acuerdo a la gráfica 3.6 se puede afirmar que la medida es totalmente independiente del tipo de red, y además sigue la relación funcional de una ley de potencia  $t \sim N^{\alpha_\lambda}$ , donde  $\alpha_\lambda = 3,25 \pm 0,06$ .

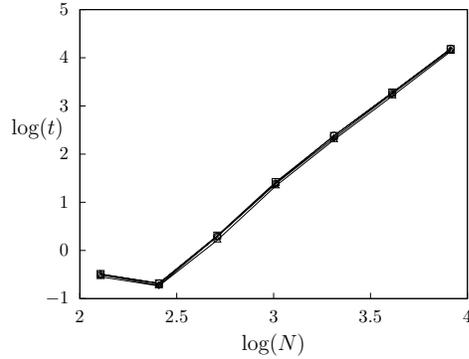


Figura 3.6: Tiempos de ejecución en función del tamaño de la red para el espectro de autovalores  $\lambda$ . BA (cuadrados), ER (círculos), Regular (triángulos), WS (rombos).

## 3.2. Representación y comparación gráfica de las algunas de las medidas implementadas.

En esta sección presentaremos los resultados de algunas de las medidas que se realizaron y que hemos considerado relevantes para ilustrar el desempeño de los códigos implementados.

En la figura 3.7(a) y (b) se muestra la función de probabilidad  $P_k$  en función del grado de la red  $k$ , para una red libre de escala y una red aleatoria, respectivamente. La medida se basa en la ecuación (2.6), que tiene asociado el código 6.

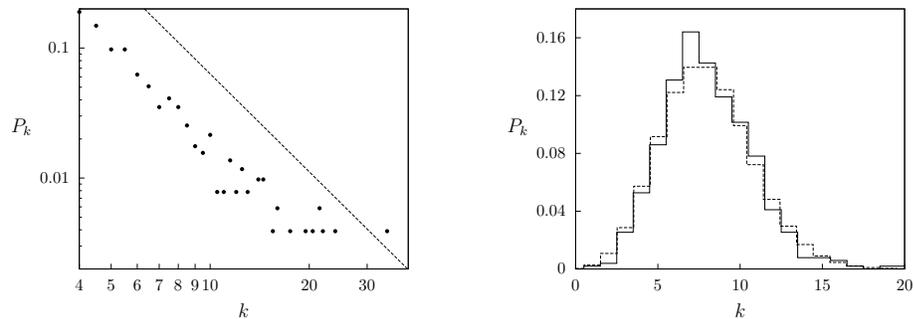


Figura 3.7: Gráfica de la probabilidad en función del grado de la red  $P_k$  con  $N = 10000$ : A la izquierda la red BA, a la derecha la red ER.

Para la red BA, 3.7(a), se aprecia como la distribución sigue una ley de potencia por su parte, la función de distribución de una red  $ER$  de la figura 3.7(b) se ajusta a una distribución de poisson representada por la línea punteada.

En la figura 3.8 se muestra la función de distribución de probabilidad  $P_k$  en función del grado de la red  $k$ , para una red con característica de pequeño mundo para dos valores diferentes de probabilidad de recableado  $P = 0,1$  y  $P = 0,5$ . Note que a me-

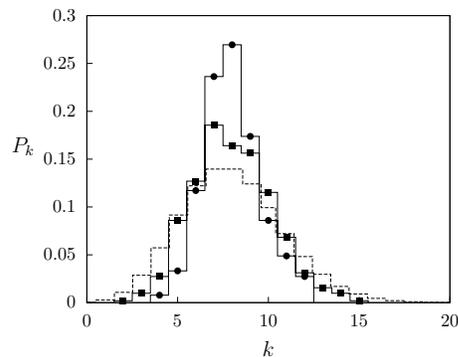


Figura 3.8: Gráfica de la probabilidad en función del grado de la red  $P_k$  con  $N = 10000$ , para una red del tipo  $WS$   $P = 0,1$  y  $P = 0,5$ .

didamente que aumenta la probabilidad de recableado de los arcos de la red la distribución se asemeja cada vez más a una Poisson representada por la línea punteada.

En la figura 3.9 se observan los resultados de la entropía  $H$  vs la probabilidad de recableado  $P$  de una red pequeño mundo, basados en las ecuaciones (2.8) y (2.6), cuyos códigos son 8 y 6, respectivamente.

El incremento de la entropía  $H$ , al aumentar la probabilidad de recableado de la red  $P$  mostrada en la figura 3.9 nos revela que esta medida caracteriza el grado de desorden de la red siendo su límite  $H_{in} \rightarrow 0$  cuando  $P_{in} \rightarrow 0$  el que corresponde a las redes regulares.

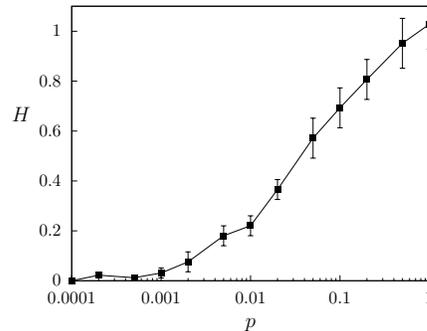


Figura 3.9: Variación de la entropía  $H$  como función de la probabilidad de recableado  $P$  en una red  $WS$  de tamaño  $N = 512$  en un anillo unidimensional con  $k = 8$ .

En la figura 3.10 podemos ver la longitud característica y el coeficiente de agrupamiento normalizados en función de la probabilidad de recableado de una red  $WS$ . Ambas medidas están basadas en las ecuaciones (2.9) y 2.22 de acuerdo a los códigos 9 y 15. En ella se aprecia que la red  $WS$  está en el régimen pequeño mundo cuando

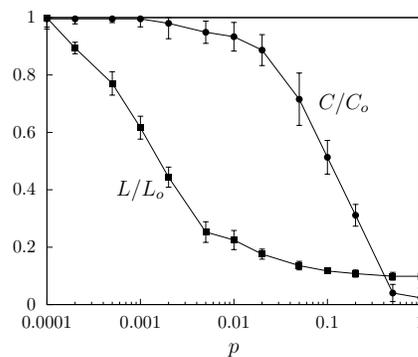


Figura 3.10: Gráfica de la longitud característica vs el coeficiente de agrupamiento. Note que están normalizados a 1

la probabilidad de recableado se encuentra entre  $P \in [0,001; 0,03]$ . En este intervalo el coeficiente de agrupamiento se mantiene elevado mientras que la longitud característica es pequeña. Para  $P = 1$  ambos valores corresponden a los de una red  $ER$ .

En 3.11 podemos observar la variación de la vulnerabilidad máxima  $V_{max}$  para

varias redes del tipo  $WS$  donde el parametro de probabilidad de recableado se varió desde  $P = 0$  hasta  $P = 1$ . Esta medida esta basada en la ecuación 2.14, de acuerdo al código 13. En ella, se observa que las redes  $WS$  son más vulnerables cuando están

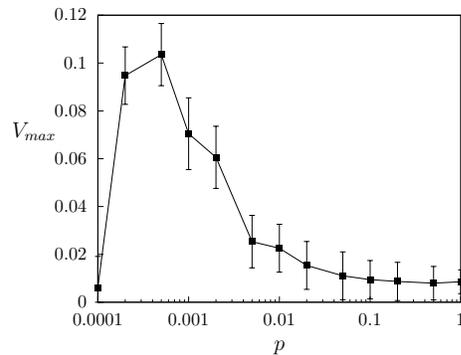


Figura 3.11: Vulnerabilidades máximas  $V_{max}$  de las redes  $SW$  para diferentes valores de probabilidad de recableado  $N = 512$ ,  $d = 1$ ,  $k = 8$ ,

pasando del régimen regular, esto es  $P = 0$ , al régimen de pequeño mundo; para luego decaer a un valor bajo que corresponde a la vulnerabilidad de una red aleatoria.

En la figura 3.12 se muestran los diferentes autovalores de la matriz de adyacencia de tres redes del tipo  $ER$ , obtenidas usando el código 16, variando en ellas la fracción de arcos dirigidos.

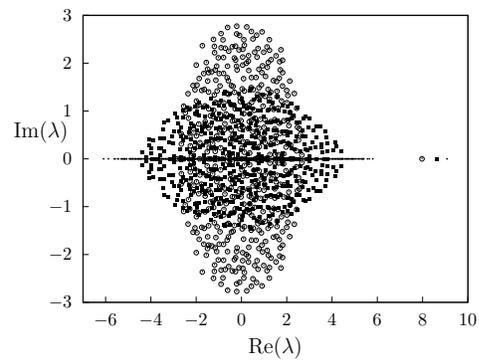


Figura 3.12: Autovalores de la matriz adyacencia para redes  $ER$  con  $N = 512$  y  $k = 8$ . Cada red tiene una fracción diferente de arcos asimétricos ( $x$ )  $q = 0$ ,  $q = 0,20$  y ( $0$ )  $q = 1$

*“Odio las citas, dime lo que sabes.”*

*Ralph Waldo Emerson*

## Conclusiones y recomendaciones

Posterior a la revisión bibliográfica específicamente la publicación de L. da F. Costa [3], se seleccionaron las medidas: Grado de la red, la distribución de probabilidades de los grados entrantes y salientes, la entropía de la distribución de grado, la logitud característica, la Eficiencia, la eficiencia sin el nodo  $i$ , la vulnerabilidad sin el nodo  $i$ , la Vulnerabilidad máxima, el coeficiente de agrupamiento para grafos simples y dirigidos y el espectro de autovalores de la red.

En la sección 2.1 se muestra la implementación de cada una de ellas en ISyS.

Luego se calcularon para las diferentes medidas ya implementadas en ISyS los tiempos de ejecución para diferentes tamaños de red estableciendo generalmente una relación funcional de una ley de potencia  $t \sim N^\gamma$  entre el tamaño de la red y el tiempo requerido por la implementación de la medida.

También, para todas las medidas se verificó su funcionamiento mostrando en la sección 3.2 aquellos resultados más ilustrativos.

Por último en el archivo de cabecera de las medidas implementadas está el manual del usuario de las mismas, que se debe distribuir con las nuevas versiones de ISyS.

*“El hombre nunca sabe de lo que es capaz  
hasta que lo intenta.”  
Charles Dickens*

## Referencias

- [1] Tucci, K. *Procesos dinámicos espaciotemporales en redes inhomogéneas*. Tesis Doctoral, Doctorado en Física Fundamental, Universidad de Los Andes. Mérida. Venezuela. (2002)
- [2] P. Erdős, A. Rényi, *On random graphs. I.*, Publicationes Mathematicae **6**, 290-297 (1959).
- [3] L. da F. Costa, F.A. Rodriguez, G. Travieso, P.R. Villas Boas, *Characterization of Complex Networks: A Survey of measurements*, Instituto de Física de São Carlos, Universidade de São Paulo, Física, SP. Brazil, 16 August (2006).
- [4] R. Johnsonbaugh, *Matemáticas Discretas*, Prentice Hall, México (1997).
- [5] K.H. Rosen. *Matemática Discreta y sus aplicaciones*. (5ta Edicion) McGraw-Hill Interamericana Editores, 2004.
- [6] V. Latora, M. Marchiori *Efficient behavior of small-world network*. Physics Review Letters, 87:198701,(2001).
- [7] V. Gol'dshtein, G. A. Koganov, G.I. Surdutovich. *Vulnerability and hierarchy of complex networks*. cond-mat/0409298, (2004)

- [8] V. Latora M. Marchiori *Vulnerability and protection of critical infrastructure*. Physics Review E, 71:015103R,(2005).
- [9] A.-L. Barabási, R. Albert. (1997). *Emergence of scaling in random networks*. Science, 286:509-5123.
- [10] R. Albert, A. L. Barabási, *Statistical mechanics of complex networks*, Rev. Mod. Phys. **74**, 47-97 (2002).
- [11] M. A. Porter, J.P. Onnela, P. J. Mucha *Communities in Networks*, Not. Amer. Math. Soc., **56**, 1082-1097 (2009)
- [12] E. Lieberman, C. Hauert, M.A. Nowak, *Evolutionary dynamics on graphs*. *Nature*, **433**, 312-316 (2005).
- [13] M. E. J. Newman *The structure and function of complex networks*, SIAM Review **45**, 167-256 (2003).
- [14] S.N. Dorogovtsev, J.F.F. Mendes, *Evolution of networks*, Advances in Physics **51**, 1079-1187 (2002).
- [15] *UNAM, 2006*  
<http://www.fis.unam.mx/max/English/notasredes.pdf>
- [16] H.M. Deitel, Prentice Hall Hispanoamericana. *Como programar en C/C++*. (2da Edición) Prentice Hall Hispanoamericana Editores, (1995).
- [17] M. Newman, A. Barabasi, D. Watt, *The Structure and Dynamics of NETWORKS*. (1ra Edición) Princeton University Press, (2006).
- [18] M. Garey, D. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., New York (1979).
- [19] E. A.Bender, E. R.Can., *The asymptotic number of labeled graphs with given degree sequences*. Journal of Combinatorial Theory, Ser. A 24:296-307, (1978).

- [20] D. J. Watts, S. H. Strogatz., *Collective dynamics of small-world networks*. Nature, 393:440-442, (1998).
- [21] *Wikipedia, 2010*  
[http://es.wikipedia.org/wiki/teoría\\_de\\_grafos](http://es.wikipedia.org/wiki/teoría_de_grafos).
- [22] *2010*  
<http://osf1.gmu.edu/~rcouch/chem665.htm>
- [23] *2006*  
<http://www.estudiolivres.org/tiki-index.php?page=SistemaVizster>
- [24] *Wikipedia, 2010*  
<http://es.wikipedia.org/wiki/Grafos#V.C3.A9rtice>
- [25] *Wikipedia, 2010*  
[http://es.wikipedia.org/Modelos\\_Erd%C3%B6s%E2%80%93Erd%C3%A9nyi](http://es.wikipedia.org/Modelos_Erd%C3%B6s%E2%80%93Erd%C3%A9nyi)
- [26] F. Reif. *Fundamentals of statistical and thermal physics*. McGraw-Hill, 1965.
- [27] L. Brillouin. *Science and Information Theory*. Dover Phoenix Editions, 2004.
- [28] C. E. Shannon, Weaver. *The Mathematical Theory of communication*. University of Illinois, 1963.
- [29] B. Wang, H. Tang, C. Guo, Z. Xiu. *Entropy optimization of scale-free networks robustness to random failures*. cond-mat/0506725, 2005.
- [30] L. Demetrius and T. Manke. *Robustness and network evolution - an entropic principle*. Physica A, 306(3-4):682-696, 2004.

- [31] I. J. Farkas, I. Derenyi, A. -L. Barabási, and T. Vicsek. *Spectra of real-world graph: beyond the semicircle law*.  
Physical Review E, 64:026704, 2001.
- [32] K. -I. Goh, B. Kahng, and D. Kim. *Spectra and eigenvectors of scale-free networks*.  
Physical Revii, 64:051903, 2001.
- [33] M. E. J. Newman. *Scientific collaboration networks: Networks construction and fundamental result*.  
Physical Review E, 64:016131, 2001.
- [34] *Wikipedia, 2010*  
[http://es.wikipedia.org/wiki/Ciclo\\_euleriano#Historia](http://es.wikipedia.org/wiki/Ciclo_euleriano#Historia)
- [35] <http://www.stanleymilgram.com/milgram.php>
- [36] *Wikipedia, 2010*  
[http://es.wikipedia.org/wiki/Red\\_libre\\_de\\_escalas](http://es.wikipedia.org/wiki/Red_libre_de_escalas)