

Algoritmia-GEI-UPC

---

# Estudio de la Transición de Fase en la Conectividad de Grafos Percolados

---

Experimentación en Redes Cuadradas, Grafos  
Geométricos Aleatorios y Grafos Aleatorio BA con  
Percolación por Aristas y Nodos

Autor/a:

Huijing Cheng

Liangwei Dong

Sambhav Mayani

Paolo Neville Milner

# ÍNDICE

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Algoritmos.....</b>	<b>3</b>
2.1 Union-Find.....	3
2.1.1 Union-Find: implementación.....	3
2.1.2 Union-Find: optimizaciones.....	4
2.1.3 Union-Find: coste.....	4
2.2 Percolación.....	4
2.2.1 Percolación por aristas.....	5
2.2.2 Percolación por nodos.....	5
<b>3 Modelos a experimentar.....</b>	<b>6</b>
3.1 Red cuadrada $n \times n$ .....	6
3.2 Grafo geométrico aleatorio.....	6
3.3 Modelo aleatorio Barabási-Albert.....	7
3.3.1 Explicación del modelo.....	7
3.3.2 Explicación del algoritmo.....	7
<b>4. Diseño de experimentación.....</b>	<b>8</b>
4.1 Experimentación.....	8
4.2 Análisis.....	8
<b>5. Resultados y análisis.....</b>	<b>9</b>
5.1 Red cuadrada $N \times N$ .....	9
5.1.1 Percolación por aristas.....	9
5.1.2 Percolación por nodos.....	10
5.1.3 Conectividad.....	11
5.2 Grafo geométrico aleatorio.....	13
5.3 Modelo Barabási-Albert.....	15
5.3.1 Percolación por nodos modelo Barabási-Albert.....	16
5.3.2 Percolación por aristas modelo Barabási-Albert.....	18
<b>6. Conclusión.....</b>	<b>20</b>
<b>7. Bibliografía.....</b>	<b>21</b>

# 1. Introducción

El objetivo de este proyecto consiste en estudiar la existencia o no de la transición de fase para los grafos cuando las aristas y / o los nodos pueden fallar. Simulamos los fallos como un proceso de percolación por nodos o por aristas e intentamos estudiar la transición de fase en relación del número de componentes conexas de un grafo. En otras palabras, queremos observar si hay un cambio abrupto de número de componentes conexas en relación de la probabilidad de fallo.

Estudiamos tanto grafos deterministas como aleatorios, en concreto, investigamos sobre redes cuadradas de tamaño  $n \times n$ , grafos geométricos aleatorios y otro tipo de grafo aleatorio de modelo Barabási-Albert.

## 2. Algoritmos

Para llevar a cabo la experimentación, es imprescindible diseñar algoritmos para simular el proceso de percolación por nodos y por aristas, y también un algoritmo para calcular el número de componentes conexas tras la percolación.

El nuestro algoritmo usa la estructura de datos Union-Find para representar la estructura del grafo en el proceso de percolación. La implementación específica de Union-Find que empleamos se inspira en el código proporcionado en el libro *Algorithms 4th Edition*.

### 2.1 Union-Find

Union-Find, también conocido como disjoint-set, es una estructura de datos diseñada para gestionar dinámicamente la información de los componentes conexas de un grafo de una forma eficiente. Cada componente conexo se representa como un árbol y la raíz de ese árbol es el representante del componente.

#### 2.1.1 Union-Find: implementación

Dispone de dos operaciones fundamentales:

**UNION(p, q):** Unir los conjuntos a los que pertenecen los nodos p y q. En concreto, lo que hace es cambiar la raíz de un conjunto p por la raíz de raíz del conjunto q, o viceversa.

**FIND(p):** Encontrar la raíz del componente conexo al que pertenece el nodo p.

#### Representación de un grafo en Union-Find

Para representar un grafo con n nodos en Union-Find, se sigue el siguiente procedimiento:

1. **Inicialización:** Crear un vector de tamaño n, donde cada posición corresponde a un nodo y guarda el índice de la raíz del componente conexo al que pertenece ese nodo. Inicialmente, cada vértice se representa como un componente individual. Por lo tanto, el número de componentes es el número de nodos.

2. **Procesar las aristas:** Para cada arista, sea  $p$  y  $q$  los dos nodos de la arista, usar  $\text{FIND}(p)$  y  $\text{FIND}(q)$  para verificar si los dos nodos están en el mismo conjunto. Si pertenecen a componentes distintos, se usa  $\text{UNION}(p, q)$  para unir los dos conjuntos, lo que reduce el número de componentes conexas.

Con este procedimiento es posible mantener la información de las componentes conexas de un grafo y obtener el número de componentes conexas fácilmente.

### 2.1.2 Union-Find: optimizaciones

La eficiencia del algoritmo es un punto crucial para el nuestro experimento, ya que tenemos que experimentar con grafos de gran tamaño. Para obtener el mejor rendimiento, hemos seleccionado la versión de Union-Find optimizada con weighted quick-union y con path compression.

**Weighted quick-union:** Al unir dos conjuntos, se conecta el árbol más pequeño como subárbol del subárbol más grande, manteniendo así un árbol más balanceado y con una altura más baja, lo que hace las operaciones de UNION y FIND más rápidas.

**Path compression:** Al hacer una operación de FIND para buscar la raíz de un nodo, conectamos directamente todos los nodos en el camino hacia la raíz. Esta optimización reduce la profundidad del árbol y reduce el tiempo de búsqueda.

### 2.1.3 Union-Find: coste

Con la implementación del Union-Find optimizada, cada operación de UNION y de FIND tiene un coste muy cerca de constante, en concreto, el coste es  $O(\alpha(n))$ , donde  $\alpha(n)$  es la función inversa de Ackermann, que crece extremadamente lenta.

Sabiendo esto, vamos a calcular el coste de construir un grafo de  $n$  nodos y  $m$  aristas en Union-Find:

- **Inicialización:** Crear un vector de  $n$  elementos, esto tiene un coste  $O(n)$ .
- **Procesar aristas:** En esta fase, se hace como máximo  $m$  UNIONS y para cada UNION hace dos FINDs. En total, hace como máximo  $m$  UNIONS y  $2*m$  FINDs, que tiene un coste de  $3*O(m*\alpha(n)) \in O(m*\alpha(n))$ .

El coste total sería la suma del coste de los dos fases,  $O(n + m*\alpha(n))$ , como que  $\alpha(n)$  es casi constante, el coste es casi lineal, cercano a  $O(n + m)$ , lo cual es bastante rápida y eficiente para el nuestro experimento.

## 2.2 Percolación

En este proyecto, hemos desarrollado dos algoritmos de percolación, por aristas y por nodos, utilizando la estructura de Union-Find. Esta estructura, implementada con weighted quick-union y path compression, nos permite aplicar percolaciones en grafos de gran tamaño de manera eficiente.

### 2.2.1 Percolación por aristas

Dado un grafo representado mediante listas de adyacencias y una probabilidad  $q$ , que es la probabilidad de percolación (la probabilidad de que no falla), la percolación por aristas consiste en construir un Union-Find añadiendo solo las aristas que continúan en el grafo. El objetivo es obtener un grafo percolado, donde algunas aristas habrán fallado y serán eliminadas y otras mantendrán en el grafo. El procedimiento es el siguiente:

1. **Inicialización:** Empezamos con un Union-Find inicializado con el número de nodos, donde cada nodo es un componente conexo independiente.
2. **Recorrer las aristas y decidir si hay un fallo:** Para cada arista, generamos un número aleatorio utilizando un generador de número aleatorio uniformemente distribuido en el intervalo  $[0,1]$ . Si el número es menor o igual a la probabilidad  $q$ , esto significa que esta arista no falla y, por lo tanto, no se elimina. En este caso, realizamos un UNION de los dos nodos de la arista. Después de procesar todas las aristas, obtenemos un grafo percolado por aristas representado en Union-Find.
3. **Cálculo del número de componentes conexos:** En este caso, el número de componentes conexos resultantes ya se calcula automáticamente con la construcción de Union-Find. Basta con consultar el atributo count de UF para obtener el número de componentes conexos.

La implementación específica de este proceso se encuentra en el método `percolateEdge(double p)` de la clase `Graph`.

### 2.2.2 Percolación por nodos

En el caso de percolación por nodos, dado un grafo y una probabilidad  $q$ , hay que eliminar los nodos que fallan. La implementación sería un poco más compleja porque el fallo de un nodo puede influir en varias aristas. El procedimiento es el siguiente:

1. **Inicialización:** Es lo mismo que percolación por aristas. Empezamos con un Union-Find inicializado con el número de nodos, donde cada nodo es un componente conexo independiente.
2. **Recorrer los nodos y decidir si hay un fallo:** Crear un vector de booleanos de tamaño  $n$  para guardar si un nodo falla o no. Para cada nodo, generamos un número aleatorio (igual que en la percolación por aristas), si es mayor o igual que  $p$ , significa que hay un fallo y marca un true en el vector.
3. **Recorrer las aristas y decidir si existe la arista:** Si los dos nodos de una arista no fallan, significa que la arista existe. En este caso, hacemos un UNION de los nodos.
4. **Cálculo del número de componentes conexas:** En este caso, hay que hacer un cálculo adicional de cuantos nodos son eliminados, y hacemos una resta para obtener el número total de componentes conexas.

La implementación completa de la percolación por nodos se encuentra en el método `percolateNodes(double p)` de la clase `Graph`.

## 3 Modelos a experimentar

### 3.1 Red cuadrada $n \times n$

Una **red cuadrada** es un tipo de grafo que tiene una estructura parecida a una cuadrícula, donde cada nodo se conecta a los vecinos cercanos en direcciones verticales y horizontales. De este modo, todos los nodos que no están al borde tienen un grado 4, ya que conectan cuatro nodos adyacentes. Este tipo de grafo tiene un uso común en el estudio de la percolación debido a su **regularidad** y **simetría**.

### 3.2 Grafo geométrico aleatorio

Un grafo aleatorio geométrico es un grafo donde los nodos es un punto en un espacio métrico, entre cada de par de nodos existe una arista si la distancia entre los nodos es inferior a una distancia dado.

Para generar un grafo geométrico aleatorio, lo que hace es crear nodos aleatorios en un espacio determinado, como en dimensiones 2D o dimensiones 3D, calcular la distancia entre los nodos y según el radio dado, decide si existe una arista entre ellos.

Este tipo de grafo puede tener propiedades muy diferentes dependiendo del radio, cuando el radio es pequeño, el grafo generado suele ser más disperso y tiene muchas componentes conexas. En cambio, si el radio es grande, el grafo generado sería más denso y la probabilidad que forma una única componente conexa es más grande.

### 3.3 Modelo aleatorio Barabási-Albert

Hemos escogido el modelo Barabási-Albert como modelo extra para estudiar la transición de fase bajo un proceso de percolación de nodos y aristas, para el apartado d).

#### 3.3.1 Explicación del modelo

El modelo Barabási-Albert es un algoritmo que forma redes aleatorias complejas, en nuestro caso “red” sería lo mismo que “grafo”.

La idea básica de este modelo es empezar con un grafo pequeño inicial, de tamaño  $m_0$ , sobre el cuál iremos añadiendo nodos con  $m$  aristas, las aristas las pondremos basándonos en la regla de “conexión preferencial”. En este caso cada nodo que añadamos al grafo construido hasta el momento, tendrá más posibilidades de conectarse a los vértices con mayor grado que los que tienen menos, siguiendo la siguiente fórmula:

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

La probabilidad de que el nuevo nodo se conecte a un nodo  $i$  ya presente en el grafo, es la división del grado del nodo  $i$  ( $k_i$ ) entre la suma de los grados de todos los nodos.

El grafo resultante se caracteriza por tener unos pocos nodos con muchas conexiones y la mayoría resultante con muy pocas, esto por la regla de la “conexión preferencial” comentada anteriormente, donde el “*rico se hace más rico*”, este tipo de redes se conocen como *redes libres de escala*, este tipo de redes están muy presentes en el mundo real, por ejemplo en las redes sociales, en internet e incluso en la naturaleza.

#### 3.3.2 Explicación del algoritmo

Al no encontrar algoritmos para generar redes siguiendo el modelo Barabási-Albert que nos convencieron, decidimos implementarlo por nosotros mismos.

En la función BA (int n, int m), presente en el documento *Graph.cpp*, dónde n es el tamaño que queremos que sea el grafo aleatorio generado y m el número de vértices que tendrá cada nodo “nuevo” que añadamos al grafo inicial de  $m_0$  en la creación de la red.

Primeramente, generamos un grafo aleatorio inicial de tamaño  $m_0$  (donde  $m_0 \geq 2$  y  $m_0 \geq m$ ) con el modelo Erdős-Rényi, pero asegurándonos que cada nodo tiene grado al menos 1.

Creamos a partir de ese grafo inicial, un vector llamado *degree\_list*, en el cual pondremos cada nodo  $i$  del grafo repetido  $k_i$  veces, o sea pondremos el índice que representa a cada nodo del grafo tantas veces como grado tiene. Este vector nos ahorrará recorrer todos los nodos del grafo cada vez que queramos añadir un “nuevo” nodo, para conectar cada nodo “nuevo” que añadamos para la generación de la red a  $m$  nodos, simplemente seleccionaremos  $m$  elementos aleatoriamente de *degree\_list* y nos conectaremos a ellos, y evidentemente actualizaremos el vector *degree\_list* respetando los nuevos grados del grafo construido hasta el momento.

## 4. Diseño de experimentación

Para cada uno de los 3 modelos mencionados anteriormente, realizamos percolaciones y analizamos el número de componentes conexos y de la conectividad, en función de la probabilidad de percolación y de tamaño del grafo.

### 4.1 Experimentación

1. Generación de grafos: creamos grafos de diferentes tamaños y parámetros (si los hay) con los algoritmos descritos anteriormente.
2. Percolaciones: Aplicamos percolación por nodos o aristas variando la probabilidad  $q$  en el rango  $[0,1]$ , con pequeños incrementos de "0.01" (1%, si lo miramos desde el punto de vista de porcentaje). Este proceso se repite varias veces para cada valor de  $q$ , ya que la percolación también es un proceso aleatorio, y analizamos la percolación con la misma probabilidad  $q$  varios grafos aleatorios distintos (menos en el caso de la red cuadrada, ya que no son grafos aleatorios si no que determinados, por lo tanto para la misma  $n$  siempre tendremos el mismo grafo). De esto modo, podemos calcular la media de todos los números de componentes conexas que obtenemos después de percolar grafos varias veces y la probabilidad de que el grafo sea conexo para cada probabilidad  $q$  y en función de los parámetros de entrada que tenga o no el modelo de generación de grafos aleatorios.
3. Resultados: Los resultados obtenidos serán guardados en archivos csv. Con los datos recopilados generamos gráficas que muestran la evolución del número de componentes conexos y de la conectividad.

### 4.2 Análisis

Una vez obtenidos los datos y las gráficas, vamos a investigar sobre el número de componentes conexos y la conectividad en relación de la probabilidad de percolación y en relación del tamaño del grafo. En el caso de grafos parametrizados, también estudiamos si un parámetro específico para crear un grafo influye en el la percolación y la transición de fase. A partir de eso, determinamos la existencia o no de la transición de fase. Si lo hay, intentamos encontrar el punto crítico.

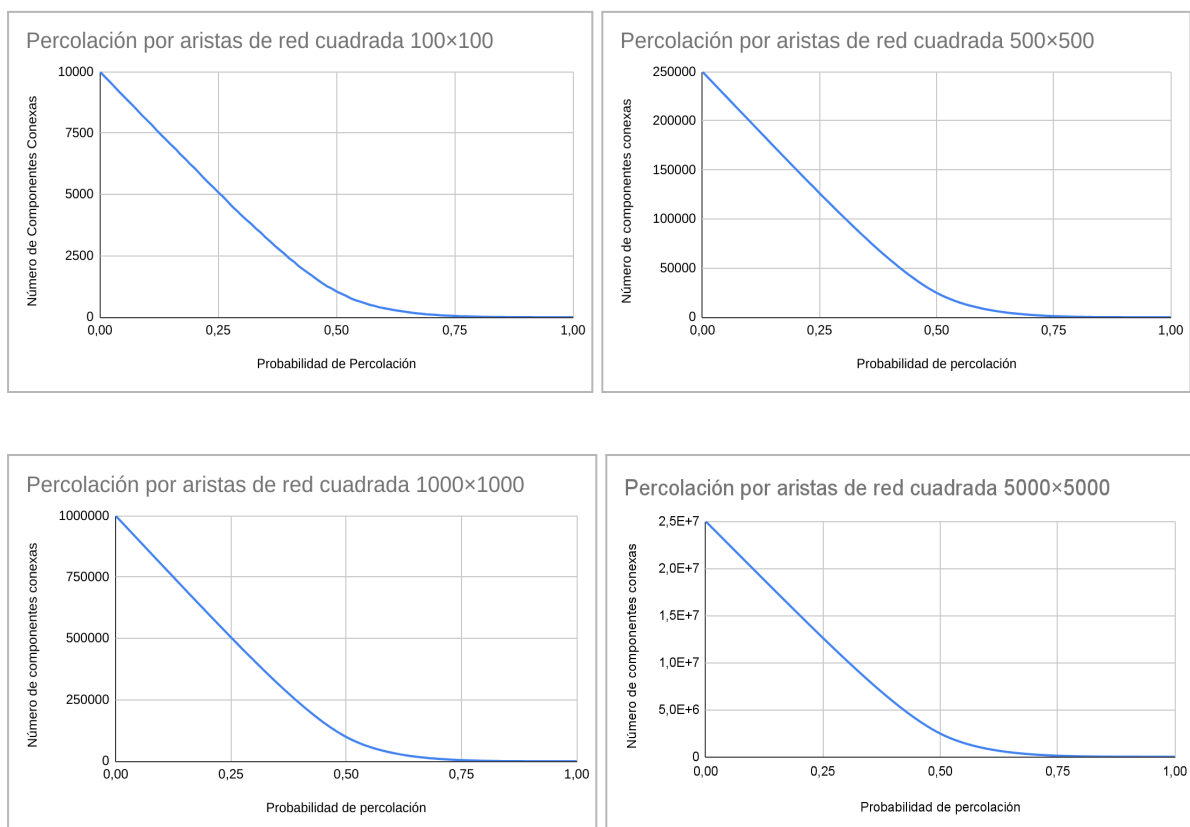


## 5. Resultados y análisis

### 5.1 Red cuadrada $N \times N$

El primer modelo que experimentamos es la red cuadrada. Aplicamos tanto la percolación por nodos como la percolación por aristas a este modelo para estudiar su comportamiento bajo diferentes probabilidades de percolación.

#### 5.1.1 Percolación por aristas.



Las gráficas anteriores muestran la relación entre la probabilidad de percolación y el número de componentes conexas tras aplicar la percolación por aristas en redes de tamaño  $n \times n$ , con  $n$  igual a **100, 500, 1000 y 5000**.

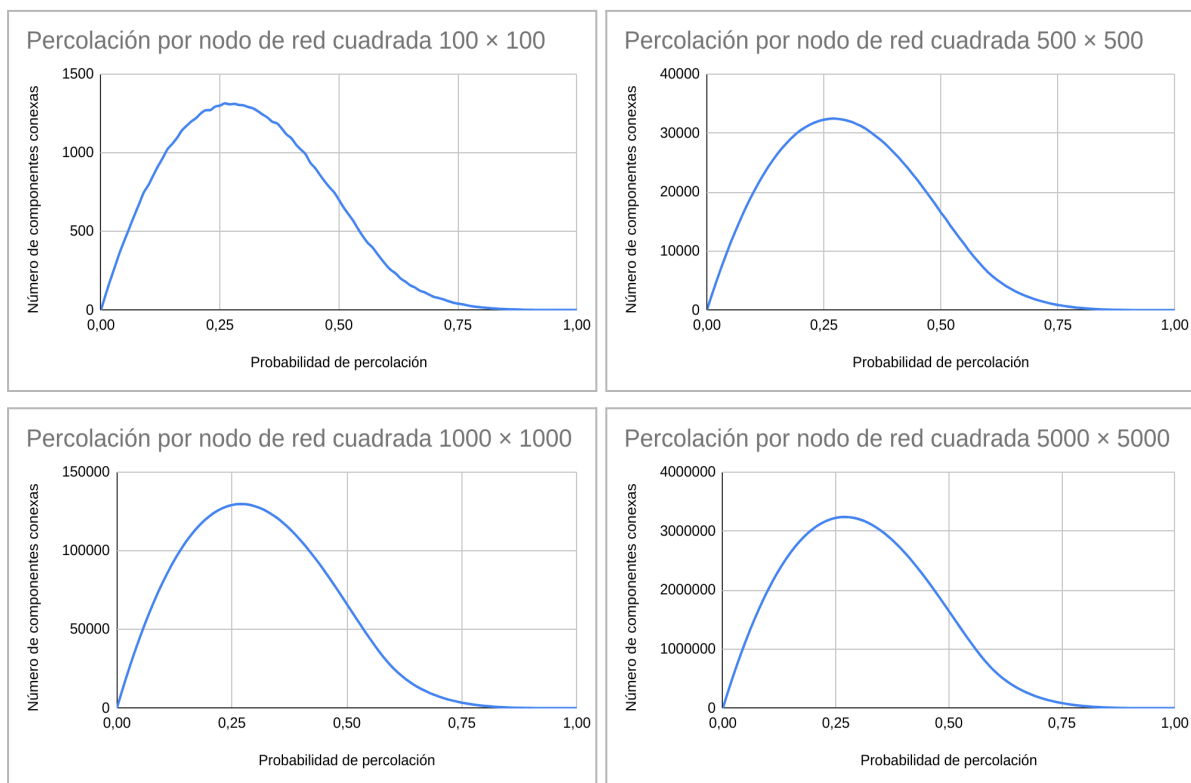
Inicialmente, cuando  $q = 0$ , todas las aristas fallan, lo que provoca que el número de componentes conexas sea igual al número de nodos, ya que cada nodo se convierte en un componente independiente.

A medida que aumenta la probabilidad  $q$ , el número de aristas incrementa y el número de componentes conexas decrece de manera similar a una función **lineal** hasta alrededor de 0.5. Para probabilidades altas, cercanas a 1, el grafo permanece mayoritariamente conexo y el número de componentes conexas es bajo, ya que pocas aristas fallan. Cuando  $q = 1$ , el grafo está completamente conexo, con un número de componentes conexas igual a 1.

En los gráficos **no se observa una transición de fase** donde hay un cambio brusco en el número de componentes conexas, sino un cambio decreciente gradual. Esto ocurre tanto en redes pequeñas (como  $n = 100$ ) como en redes más grandes (como  $n = 5000$ ) y el patrón que sigue son similares.

Aunque no se observa transición abrupta, podemos observar que entre 0.5 y 0.6, el decrecimiento de componentes conexas es mucho menor que para probabilidades más pequeñas, eso puede ser debido a que a partir de este punto, el grafo está mayormente conectado y comienza a formar una única componente gigante.

### 5.1.2 Percolación por nodos



Las gráficas obtenidas muestran la relación entre la probabilidad de percolación por nodos y el número de componentes conexas tras aplicar la percolación en redes cuadradas de tamaño  $n \times n$ , con  $n$  igual a 100, 500, 1000 y 5000.

Al punto  $q = 0$ , todos los nodos fallan, y por tanto tenemos 0 componentes conexas. A medida que incrementa la probabilidad, el número de componentes conexas aumenta hasta  $q = 0.25$  aproximadamente, donde la función alcanza su máximo de componentes conexas, y luego empieza a decrecer. A partir de  $q = 0.6$  vemos otro cambio en el comportamiento en la función, donde empieza a tender hacia el eje X de manera asintótica.

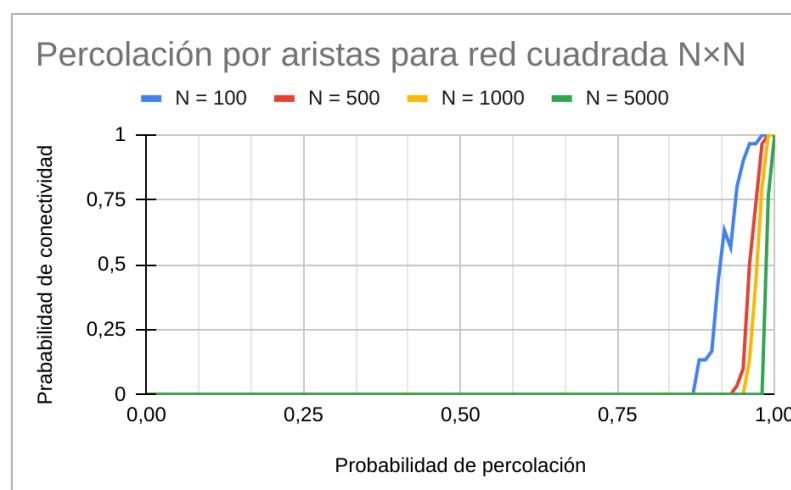
La siguiente tabla muestra la probabilidad donde el número de componentes conexas alcanza la máxima, el número máximo de componentes, y la proporción de componentes conexas respecto al número de nodos red de cada tamaño:

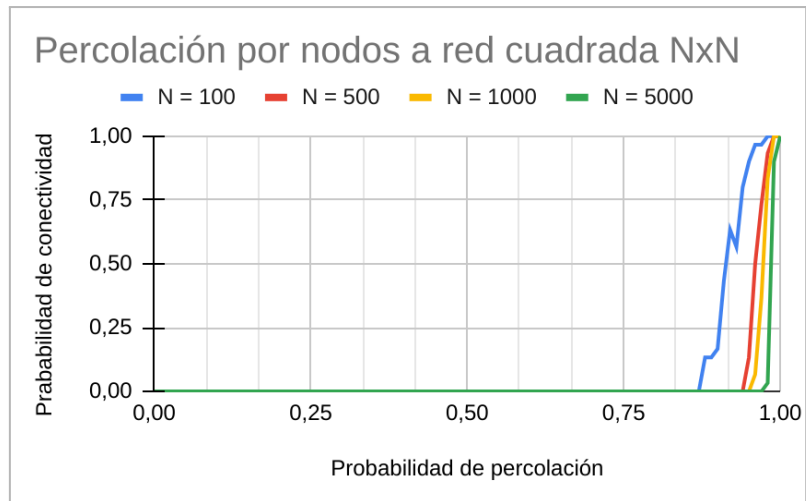
Tamaño de la red(N)	100	500	1000	5000
Probabilidad en el pico máximo	0,29	0,27	0,27	0,27
Número máximo de componentes conexas	1313,7	32469,2	129690	3239010
Proporción de componentes conexas respecto al número de nodos	0,13137	0,1298768	0,12969	0,1295604

A partir de los datos, observamos que en redes grandes, el pico máximo de componentes conexas aparece alrededor de una probabilidad de percolación de 0,27. En este punto, el número de componentes conexas alcanza su valor máximo, que es aproximadamente 12,96% del total de nodos de la red y la proporción es similar con redes de diferentes tamaños. Además, en las gráficas, observamos que la forma de la curva que presenta cada modelo son similares. Con esto, podemos decir que el tamaño de la red no afecta significativamente en el comportamiento general de los resultados, ya que sus comportamientos son similares.

Cuando a la transición de fase, igual que la percolación por aristas, aquí **tampoco se observa una transición de fase**, ya que no hay un cambio abrupto del número de componentes conexas. Todos los cambios són suaves.

### 5.1.3 Conectividad





Estas gráficas muestran la probabilidad de conectividad (la probabilidad de que el grafo es conexo tras percolación) en función de la probabilidad de percolación. A diferencia de las gráficas anteriores, aquí sí que se puede observar **transiciones claras** en probabilidades altas cercanas de 1.

Para redes más pequeñas como  $n = 100$ , la transición de fase es más gradual, al incrementar la probabilidad, la transición de fase es más abrupta y aparece más cercana a  $q = 1$ .

Esto nos indica que el tamaño de la red sí que tiene un impacto en la dinámica de la percolación, el punto de transición es alto para redes grandes y la transición es más abrupta. En el nuestro caso, el punto de transición de la probabilidad de conectividad es bastante alto, entre 0.9 y 1. Para  $n = 5000$ , este punto se aproxima a 0.99.

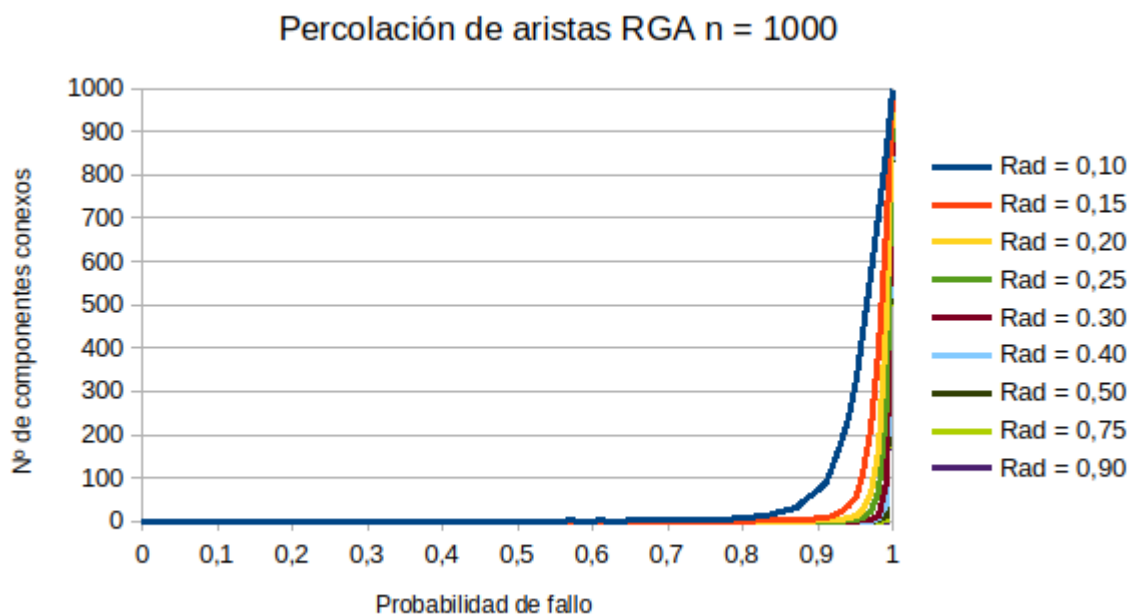
## 5.2 Grafo geométrico aleatorio

Para la experimentación de la percolación por aristas sobre grafos geométricos aleatorios decidimos utilizar los de dos dimensiones ya que necesitamos grafos un poco densos para poder hacer pruebas con radios pequeños, pero tampoco queremos que tengan demasiados nodos para que las ejecuciones no sean demasiada largas, así que descartamos los que tienen más dimensiones.

Elegimos  $n = 1000$ ,  $n = 2500$  y  $n = 5000$  como las cantidades de nodos para crear los grafos. La  $n$  no puede ser demasiada pequeña porque los grafos no serían conexos con un radio pequeño.

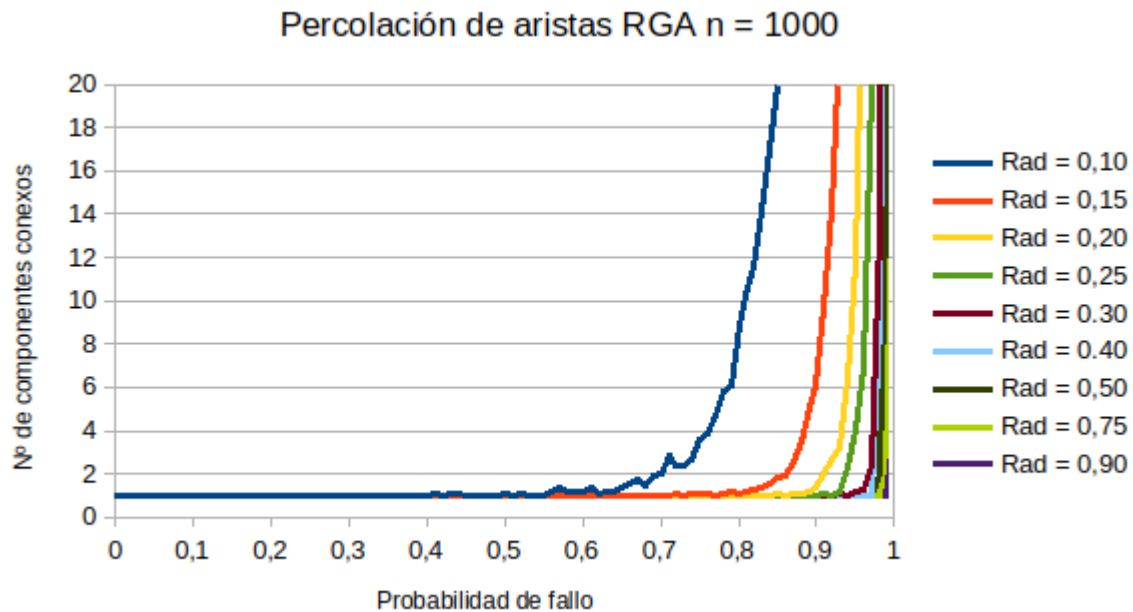
Para cada cantidad de nodos creamos distintos grafos aleatorios conexos con diferente radio y aplicamos la percolación de aristas distintas veces sobre cada grafo para obtener la media de componentes y la probabilidad de no conectividad para cada probabilidad de fallo. Nos aseguramos de que el grafo creado sea conexo contando cuantos componentes conexos tiene con un DFS. La probabilidad de no conectividad será 0 si el grafo percolado es conexo y 1 si no lo es y obtenemos la media con las repeticiones y hacemos la gráfica con esta media.

Para  $n = 1000$  obtenemos la siguiente gráfica de componentes conexos:

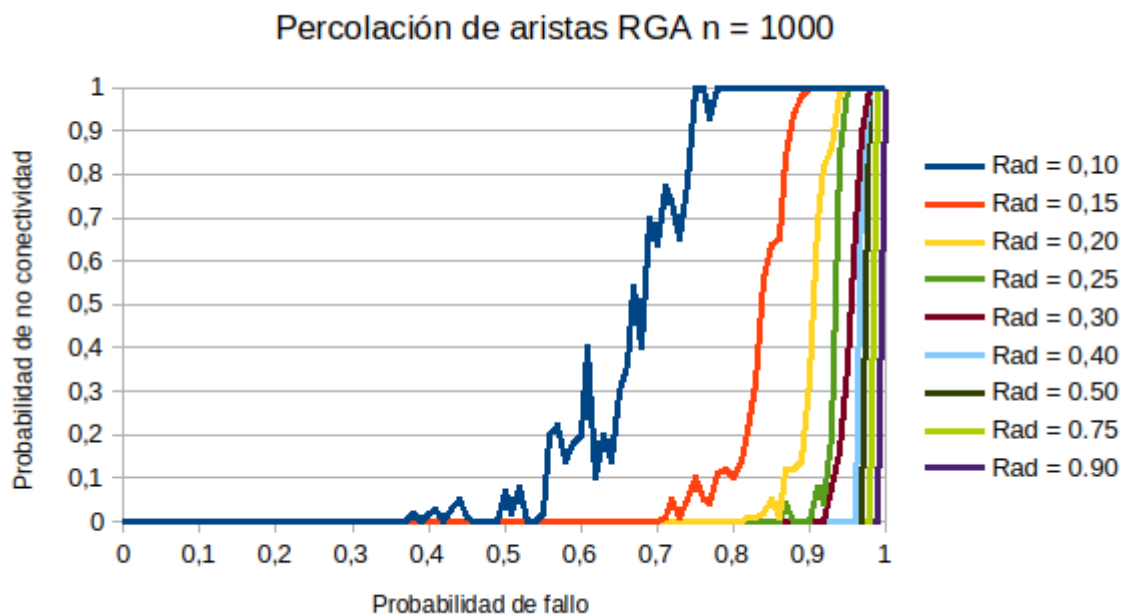


A partir de esta gráfica podemos ver que aunque el radio sea relativamente pequeño, los grafos que se crean tienen conexiones muy fuertes entre ellos. Vemos que el número de componentes conexos no empieza a crecer hasta que la probabilidad de fallo sea elevada, en este caso empieza a crecer después de la probabilidad de 0,5.

Si hacemos un zoom-in, podemos ver que después de la probabilidad de 0,7, el número de componentes empieza a crecer de forma exponencial, como podemos ver en la siguiente gráfica. No sé si podríamos considerar esto como una transición de fase, pero sí que pasamos de un estado donde el grafo es conexo a otro donde no lo es.



Con la siguiente gráfica estudiamos la conectividad de las percolaciones. El eje x representa la probabilidad de fallo de la percolación y el eje y representa la probabilidad de no conectividad, es decir, si esta se acerca a 0, significa que es probable que el grafo sea conexo; de lo contrario, si se acerca a 1, es probable que el grafo no sea conexo. En esta gráfica podemos apreciar claramente transiciones de fase, sobre todo con radio de 0,15.



Para las otras cantidades mayores de nodos nos dan unas gráficas muy parecidas a las de  $n = 1000$ , solo que están más aplazados al lado de la probabilidad de fallo de 1 ya que son grafos más densos y más conexos y por lo tanto necesitan una probabilidad de fallo más alto para que crezca el número de componentes conexos.

## 5.3 Modelo Barabási-Albert

Ahora estudiaremos si hay o no transición de fase en el modelo extra que hicimos para el apartado 'd)', el Barabási-Albert, explicado anteriormente. La experimentación para obtener los datos que se analizarán a continuación, consistía en, para cada probabilidad de que un nodo/arista no falle abarcada desde 0 hasta 1 en incrementos de "0,01", estudiar el número de componentes conexas de una red generado por el modelo BA percolado de tamaño  $n$  y parámetro  $m$  ( $n$  y  $m$  previamente definidas) por esa probabilidad.

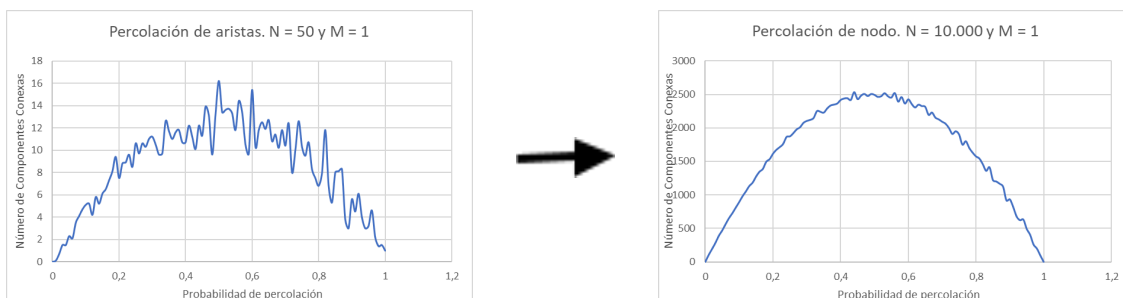
Básicamente, para la probabilidad "0", generamos 10 grafos aleatorios siguiendo el modelo BA, lo sometemos al proceso de percolación de aristas para la probabilidad "0" de que no falle cada arista del grafo, y hacemos la media del número de componentes conexas que quedan en el grafo después someterlo a la percolación, hacemos lo mismo para la probabilidad "0,01" (que sería 1% si lo miramos en porcentaje), luego para la probabilidad "0,02"... así hasta llegar a la probabilidad de que no falle una arista "1". Con este proceso, tenemos como datos las siguientes dos columnas:

<b>(p)</b> Probabilidad de que no falle arista/nodo	<b>(I.C.C)</b> Número de componentes conexas que queda de media después de percolar el grafo por aristas con la probabilidad $p$ como parámetro
---	---

Hacemos el mismo proceso con la percolación por nodos, pero con la probabilidad de que un nodo no falle en vez de que una arista no falle. Decidimos hacer la media de 10 grafos aleatorios BA percolados para obtener muestras más exactas y reducir la aleatoriedad de los datos.

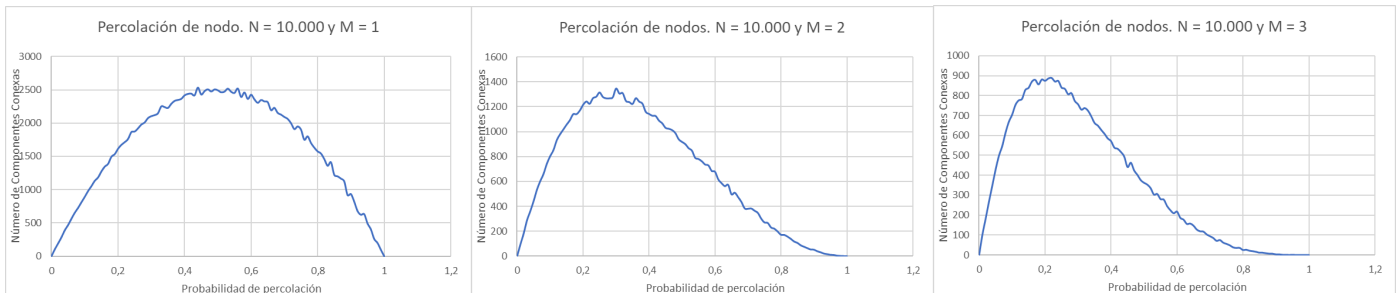
La " $n$ " (tamaño del grafo) y la " $m$ " (necesaria para crear una red siguiendo el modelo BA) las íbamos modificando, hemos probado con los siguientes valores de  $n = [20, 50, 5.000 \text{ y } 10.000]$ , y los valores de  $m = [1, 2, 3]$  (posteriormente experimentamos con casos más extremos para confirmar hipótesis que teníamos), esto para estudiar el comportamiento en grafos pequeños y grandes y ver si la  $m$  afecta significativamente al resultado, y así poder estudiar si hay o no transición de fase en todos los casos.

Experimentando con  $n$ , nos dimos cuenta que la gráfica resultante para todas los valores de  $n$  era muy parecida, lo único en que afectaba  $n$  era que la función que definía la gráfica se iba perfilando con valores de  $n$  superiores, como se ve en los siguientes ejemplos:

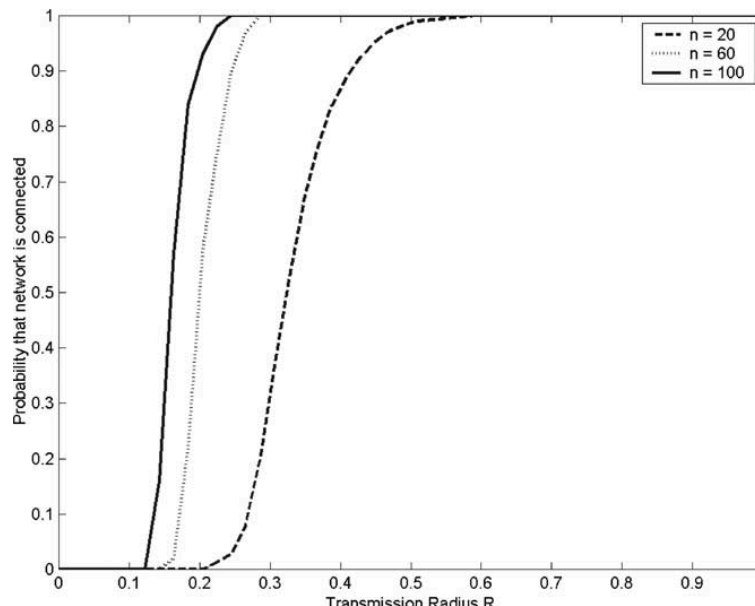


Así que para analizar los resultados, nos vamos a quedar con las gráficas con la  $n$  más grande (10.000), donde se ve cómo varían los datos más claramente y con menos variabilidad aleatoria.

### 5.3.1 Percolación por nodos modelo Barabási-Albert



Como se puede observar en las gráficas de arriba, no hay ninguna transición de fase, lo único que se puede observar es una función convexa, es que a medida que aumentamos la  $m$  es que va creciendo más y más rápido al principio, hasta llegar a un punto de inflexión, a partir del cual la función decrece, este cambio de pendiente en la función evidentemente no sería catalogado como una transición o transiciones de fase, ya que no “pasamos de una fase a otra”, no vemos un cambio brusco en la función donde pasemos de estar en un “rango” de número de componentes conexas y que ese “rango” varíe a uno muy distinto (ya sea mayor o menor) a partir de una probabilidad de fallo de nodo determinada, como sí sucede en la siguiente imagen que se muestra como ejemplo en el enunciado del proyecto:



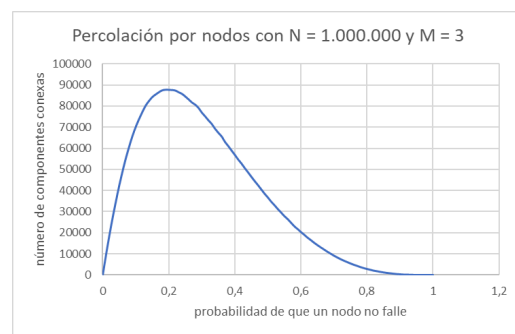
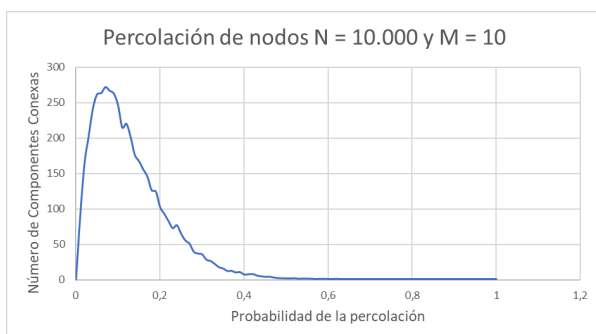
Lo único que se puede interpretar de las gráficas, es que cuanto menor es la probabilidad de que no falle un nodo (y por lo tanto mayor la de que sí falle), se puede ver que para probabilidades pequeñas es más grande el número de componentes conexas generadas, no muy pequeñas, ya que por ejemplo si tenemos  $p = “0,01”$ , entonces hay un 99% de posibilidades para cada nodo a que falle, entonces nos quedamos sin la mayoría de nodos, y por ende tendremos menos componentes conexas, pero para  $p = “0,2”$  en la gráfica donde estudiamos el comportamiento de grafos generados por el modelo BA con  $M = 2$ , el número



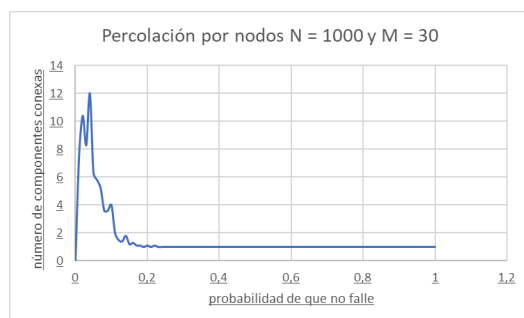
de componentes conexas es alto, entonces hay 80% de posibilidades de que cada nodo falle, entonces es muy probable que acabemos “fragmentando” mucho el grafo en distintas componentes conexas, sobretodo porque es probable que eliminemos a los nodos con mayor grado, dejando a todos sus vecinos, que acostumbran a ser de grado inferior, sin estar conectados entre ellos (recordemos que una red/grrafo generado por el modelo BA, son redes de libre escala, que se caracterizan por tener pocos nodos muy conectados y muchos nodos muy poco conectados).

El porqué de comprensión de que la “función” vista en la gráfica a medida que aumentamos el parámetro ‘ $m$ ’, se puede justificar que al tener una ‘ $m$ ’ más grande, cada vértice que añadimos en el proceso de creación del grafo tiene más aristas (literalmente  $m$  aristas), entonces formamos un grafo “más conexo”, donde es más difícil fragmentarlo en componentes conexas, por lo tanto requerimos más probabilidad de que un nodo falle para fragmentarlo, y por ende, menor probabilidad de que no falle.

Para cerciorarnos de que estábamos en lo correcto, decidimos probar el caso con una  $n$  muy grande (de 1000000 de nodos) y con una  $m$  relativamente grande, y las gráficas formadas por los datos obtenidos nos dan la razón, siendo compatibles con el razonamiento que hemos descrito:



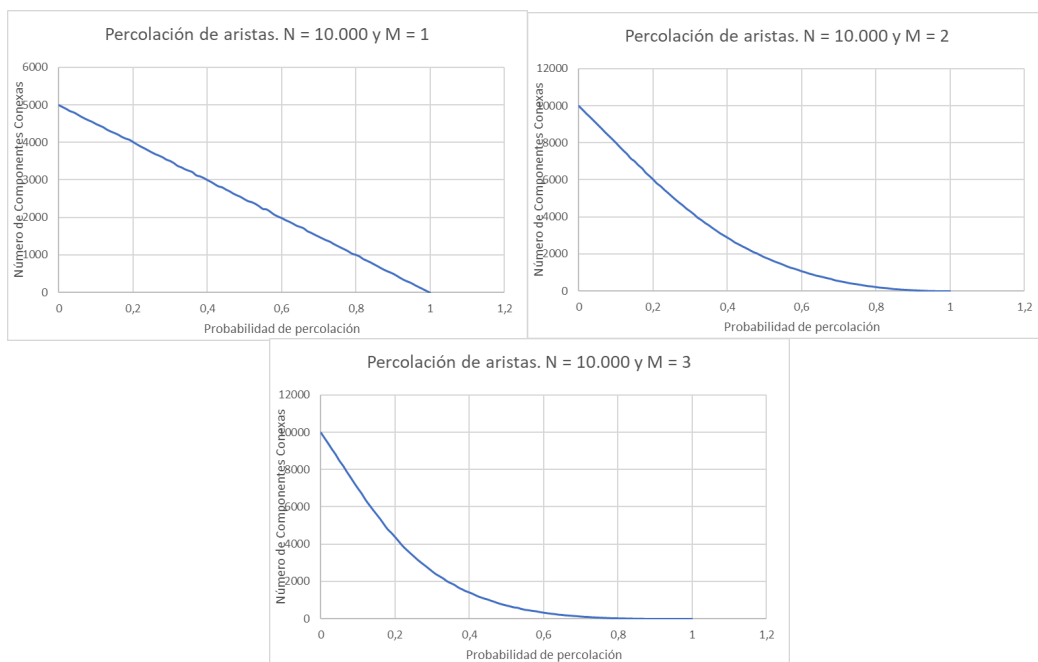
Finalmente, aunque en un inicio no lo teníamos pensado, hemos decidido probar con valores de  $m$  aún más grandes, por si hay transición de fase respecto al parámetro  $m$ , que ya hemos visto que para  $m$  pequeñas no hay. El porqué no teníamos pensado hacerlo es por como es el modelo y como se aplica en el mundo real, las redes formadas por el modelo, redes de escala libre, se caracterizan por lo que ya se ha comentado anteriormente cuando describimos el modelo, por tener pocos nodos muy conectados y muchos muy poco conectados, si tenemos valores de  $m$  que no son pequeños, esa caracterización deja de existir, todos los nodos empiezan a estar muy conectados. Pero aún así, por curiosidad y ver si en el modelo hay transición de fase, aunque no tenga aplicaciones prácticas ya que nunca pondremos valores de  $m$  tan grandes.



La tendencia es la misma, además de las 2 expuestas en el documento, también generamos varias con valores intermedios de  $m$ , y no se veía ninguna transición de fase respecto al parámetro  $m$ , no hay ningún valor de  $m$  a partir del cuál cambie drásticamente el número de componentes conexas de  $m+i$  ( $i > 0$ ) siguientes...

Pero sí que podríamos llegar a interpretar que, cuando la  $m$  crece mucho, el crecimiento y decrecimiento de componentes conexas en el gráfico es tan drástico, que podríamos llegar a considerar esos dos cambios como fases de transición, ya que, respecto a una probabilidad de que un nodo no falle, hay un cambio brusco en el número de componentes conexas.

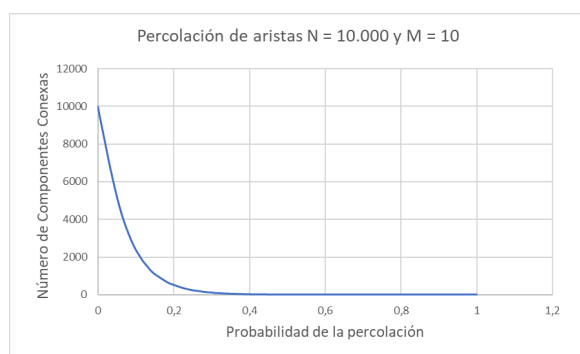
### 5.3.2 Percolación por aristas modelo Barabási-Albert



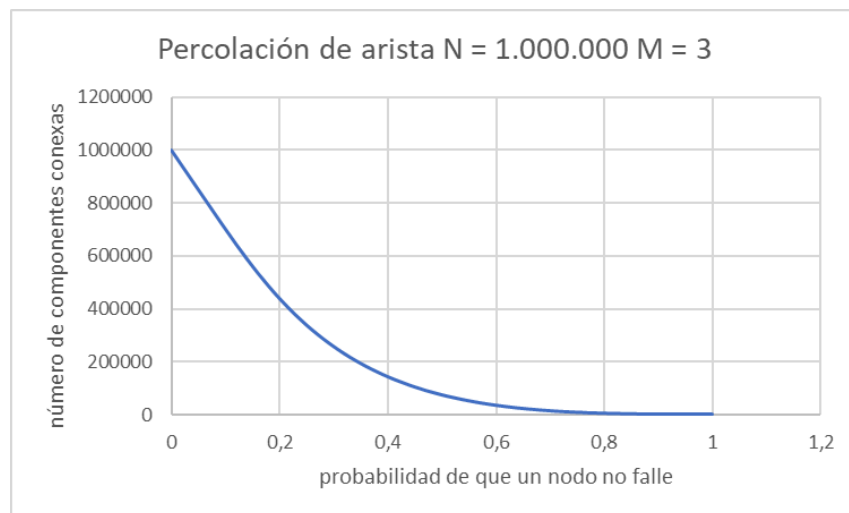
Analizando las gráficas de la percolación por aristas del modelo BA, se puede ver a simple vista que no hay ningún tipo de transición de fase. Lo único que vemos es una función descendente, que en el caso particular de  $m = 1$ , decrece linealmente, y en el resto de casos  $m > 1$ , decrecen cada vez más como una función exponencial invertida (p.e.j.  $(1/2)^x$ ), donde a medida que aumentamos la  $x$  decrece más rápido, hasta el punto en el que en la gráfica de  $N = 10.000$  y  $M = 10$  llegamos a 1 componente conexo en la probabilidad 0,61.

$N = 10.000$  y  $M = 10$

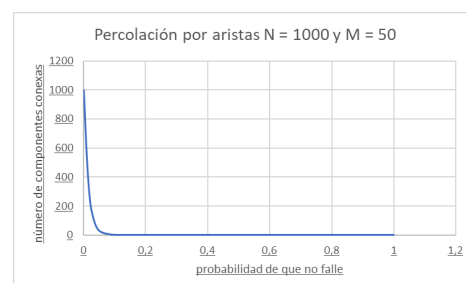
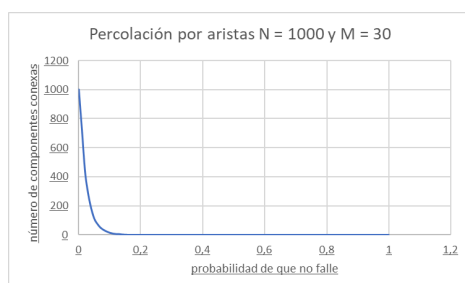
p: 0,61	C.C : 1
---------	---------



La justificación de porqué sucede esto sería que, a medida que aumentamos la probabilidad de que una arista cualquiera falle (y por lo tanto disminuimos la probabilidad de que no falle), el número de componentes conexas se dispara, ya que es más probable que eliminemos a cada vez más a algunos de los pocos nodos con grado muy alto, fragmentando más el grafo en más componentes conexas. Y de la misma manera en la que lo describimos en la percolación por nodos, cuando la  $m$  es mayor, entonces el grafo está más “conexo”, hay más vértices entre los nodos, por lo tanto necesitamos una probabilidad mayor de que una arista cualquiera falle (menos de que no falle) para segmentar el grafo en muchas componentes conexas. También vemos que se cumple esta premisa con una  $n$  absurdamente grande y  $m = 3$ , donde la “función” que aparece en la gráfica de abajo es idéntica a la de  $n = 10.000$  y la misma  $m$ , reafirmando otra vez, que la  $n$  no afecta al comportamiento de la gráfica, sólo reduce la “variabilidad” de la “función” que aparece, haciéndola más clara, en cambio modificar la  $m$  si que afecta ligeramente a la gráfica porque modifica el “grado de conectividad del grafo”.



Pero en cambio, si estudiamos, como en los nodos, grafos con valores de  $m$  grandes, cosa que no se ve en el mundo real, vemos entonces que la función es tan decreciente al principio, que podríamos considerar transición de fase una la probabilidad de que un nodo no falle de 0 a la probabilidad en la cual la media de componentes conexas baja a 1 (el mínimo). A medida que aumentamos la  $m$ , vemos que



*Nota: No ponemos gráficas con valor de  $m$  mayor ya que es muy costoso generar un grafo con una  $m$  grande. Lo que sí que llegamos a hacer es ejecutar solo el principio del programa, para las primeras probabilidades de percolación, y ver que para una  $n = 1000$  y  $m = 500$ , a partir de 0,1 ya tenemos*

*solamente 1 componente conexo, así que no hacía falta ejecutar más para ver que el para el resto de probabilidades también acabaremos en 1 c.c..*

## 6. Conclusión

En conclusión, a partir de todas los experimentos que hemos realizado, podemos confirmar que en **redes cuadradas no se observan transiciones de fase de componentes conexos**, tanto en percolación por aristas como en percolación por nodos. Pero cuando estudiamos la conectividad, sí que podemos observar **una transición de fase obvia en la conectividad**, y el punto crítico aumenta al incrementar el tamaño de red. Este punto tiene un valor aproximado de 0.99.

En el caso de los **grafos geométricos aleatorios** podemos decir que **sí que se observan transiciones de fase, tanto de componentes conexos como de conectividad**. Estas transiciones dependen del radio del grafo ya que si este es mayor significa que el grafo es más conexo y por lo tanto la transición estará situada en una probabilidad de fallo más alta. También depende del número de nodos ya que si tenemos un grafo con más nodos que otro en un mismo espacio con el mismo radio, entonces es más conexo que el otro y por lo tanto su transición también estará en una probabilidad de fallo más alto.

Finalmente, en el modelo BA podemos observar que **la existencia de transición de fase depende directamente del parámetro  $M$** , y que con valores de  $M$  altos podemos observar una transición de fase respecto a  $p$  (la probabilidad de no fallo). En cambio, según el comportamiento observado al incrementar gradualmente la  $M$ , podemos decir con certeza que no hay una transición de fase respecto a  $M$  en general, ya que no hay un valor de  $M$  a partir del cual obtengamos un comportamiento en los gráficos radicalmente diferente del anterior.

## 7. Bibliografía

1. R. Sedgewick and K. Wayne. Algorithms. Addison-Wesley, 4th edition, United States: Princeton University, 2011, ISBN: 9780321573513. Contenido online disponible en: <https://algs4.cs.princeton.edu/>
2. Wikipedia. Erdős–Rényi model [en línea]. 26 Marzo 2024. Disponible en: [https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi\\_model](https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi_model)
3. Wikipedia. Percolation Theory [en línea]. 10 Octubre 2024. Disponible en: [https://en.wikipedia.org/wiki/Percolation\\_theory](https://en.wikipedia.org/wiki/Percolation_theory)
4. Wikipedia. Random Graph [en línea]. 22 Septiembre 2024. Disponible en: [https://en.wikipedia.org/wiki/Random\\_graph](https://en.wikipedia.org/wiki/Random_graph)
5. Wikipedia. Percolation Threshold [en línea]. 3 Septiembre 2024. Disponible en: [https://en.wikipedia.org/wiki/Percolation\\_threshold](https://en.wikipedia.org/wiki/Percolation_threshold)
6. Wikipedia. Dynamic Connectivity [en línea] 19 Octubre 2024. Disponible en: [https://en.wikipedia.org/wiki/Dynamic\\_connectivity](https://en.wikipedia.org/wiki/Dynamic_connectivity)
7. Wikipedia. Barabási–Albert model [en línea] 12 Octubre 2024. Disponible en: [https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%93Albert\\_model](https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%93Albert_model)
8. Albert-László Barabási. Network Science: the Barabási-Albert model.[en línea] 2014. Disponible en: <https://barabasi.com/f/622.pdf>
9. Wikipedia. Conexión preferencial. [en línea] 20 Abril 2022. Disponible en: [https://es.wikipedia.org/wiki/Conexi%C3%B3n\\_preferencial](https://es.wikipedia.org/wiki/Conexi%C3%B3n_preferencial)
10. Emanuel Melachrinoudis. Percolation Theory and Forest Fires. [en línea] Disponible en: <https://www1.coe.neu.edu/~emelas/type.htm>