

# Tarea 2

Jesus Angel Patlán Castillo (5621)

26 de febrero de 2019

En esta tarea se analizan distintos tipos de acomodo visual grafos generados a partir de la librería NetworkX [4] de Python [1]. Se estudian particularmente los grafos utilizados para la tarea 1, la librería Matplotlib [5] para generar el grafo en los distintos algoritmos estudiados y para guardar el grafo en el formato “.eps”. El código empleado se obtuvo consultando la documentación oficial de la librería NetworkX [3] y guías suplementarias [7, 13]. Las imágenes y el código se encuentran disponibles directamente en mi repositorio [12]. Las distintas formas de trazar los grafos son:

```
1 #Distintos algoritmos para trazar los grafos
2 nx.draw(G, pos=nx.spectral_layout(G), node_color='r', edge_color='b',
3         with_labels=True)
4 nx.draw(G, pos=nx.circular_layout(G), node_color='r', edge_color='b',
5         with_labels=True)
6 nx.draw(G, pos=nx.random_layout(G), node_color='r', edge_color='b',
7         with_labels=True)
8 nx.draw(G, pos=nx.shell_layout(G), node_color='r', edge_color='b',
9         with_labels=True)
10 nx.draw(G, pos=nx.spring_layout(G), node_color='r', edge_color='b',
11         with_labels=True)
```

## 1. Algoritmo Circular

Este algoritmo cuenta con las siguientes propiedades [15]:

- Los nodos del grafo se encuentran sobre una misma circunferencia.
- Las aristas se representa con lineas rectas.
- Requieren a lo más  $O(n)$  tiempo para su trazado, siendo  $n$  el número de aristas.
- Se utilizan particularmente para grafos en las que se desea mostrar claramente la biconectividad entre nodos.

### 1.1. Grafo simple no dirigido acíclico

Una red de ciudades en un estado puede ser un ejemplo de un grafo simple no dirigido acíclico, ya que podemos representar cada ciudad como un nodo

del grafo, y tomar en cuenta que cada arista representan las carreteras que conectan cada ciudad, considerando que sería un grafo no dirigido dado que una carretera puede ir de ida y vuelta entre ciudades, y es acíclico puesto que no se consideran carreteras que van de una ciudad a sí misma [8]. La figura 1 representa un ejemplo de este tipo de grafo.

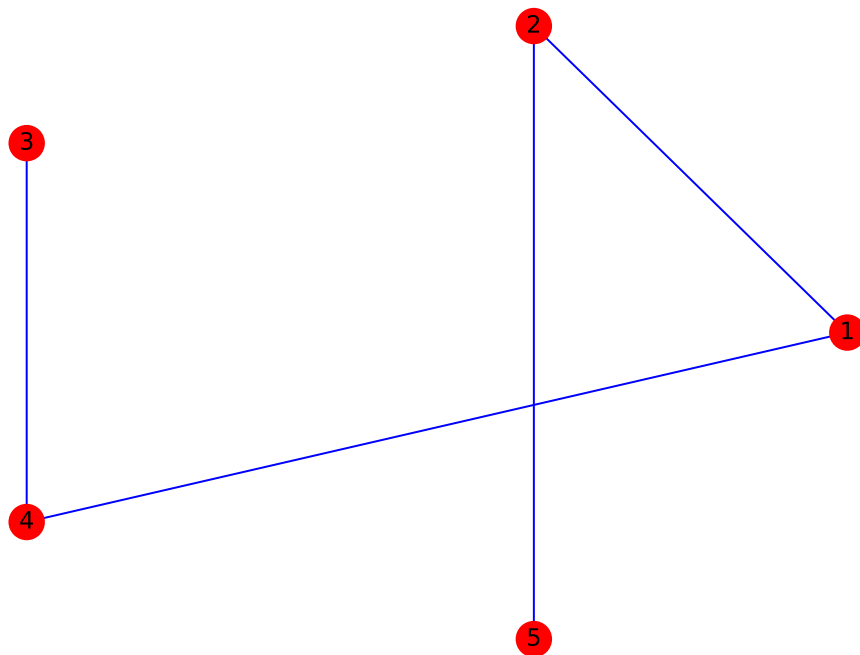


Figura 1: Grafo simple no dirigido acíclico.

## 1.2. Grafo simple no dirigido cíclico

Una ruta de autobús puede ser representada por este tipo de grafos, en donde los autobuses pasan a través de las calles y avenidas (representadas con las aristas), y cada nodo del grafo se puede representar una parada donde se recogen personas. La figura 2 representa un ejemplo de este tipo de grafo.

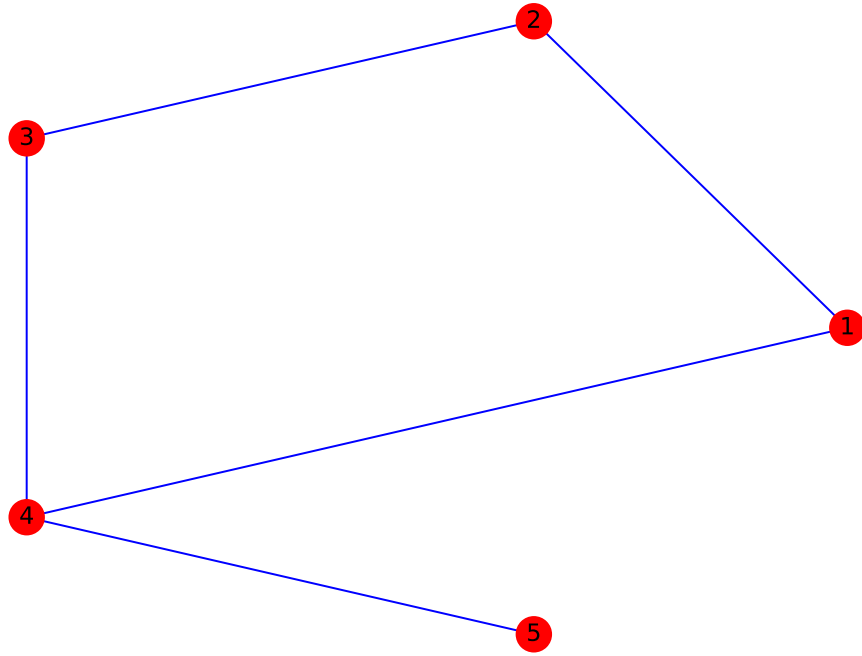


Figura 2: Grafo simple no dirigido cíclico.

### 1.3. Grafo simple dirigido cíclico

En algunos juegos se tienen distintos estados en los que el jugador se puede encontrar, y esto puede ser representado en un grafo dirigido cíclico, dado que cada estado se puede visualizar por medio de un nodo, y las aristas representarían las maneras en las que un estado puede cambiar a otro [9]. La figura 3 representa un ejemplo de este tipo de grafo.

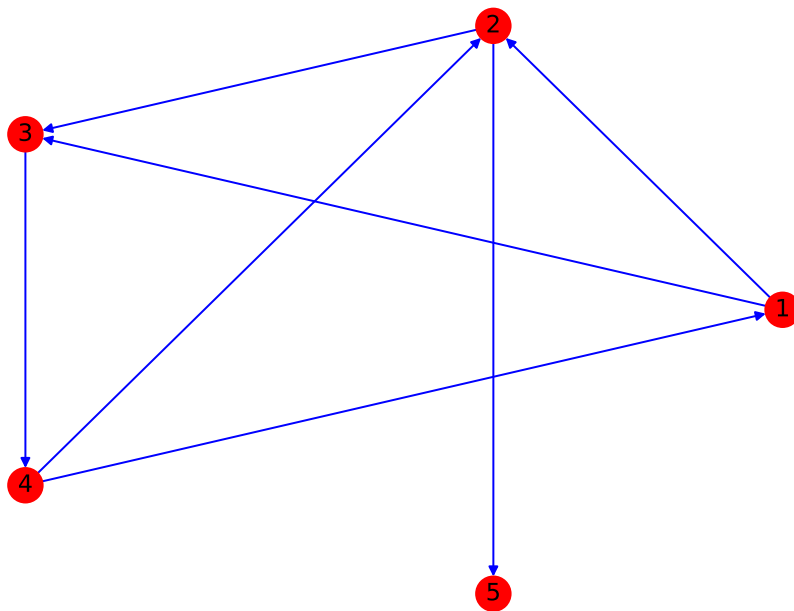


Figura 3: Grafo simple dirigido cíclico.

#### 1.4. Multigrafo dirigido acíclico

Como en el grafo simple dirigido reflexivo, en una página web se pueden tener múltiples hipervínculos que te redirigen a una misma página web. La figura 9 representa un ejemplo de este tipo de grafo.

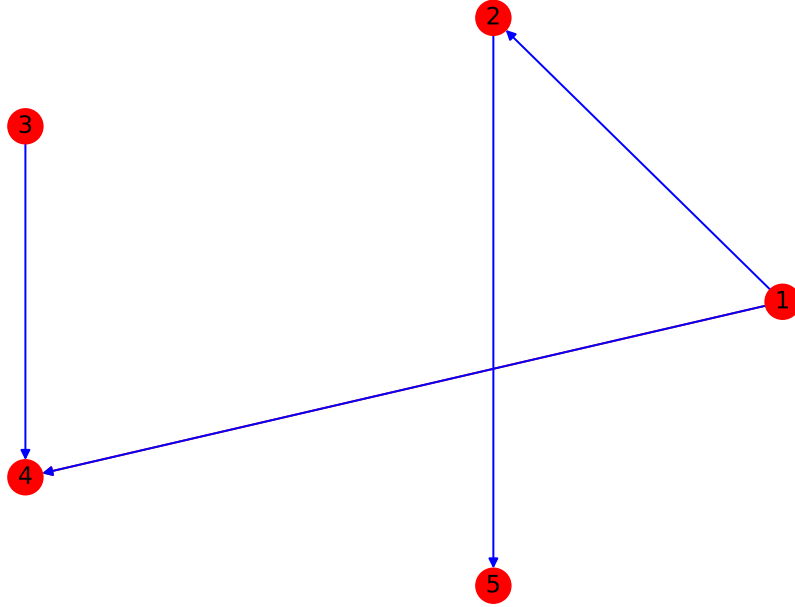


Figura 4: Multigrafo dirigido acíclico (los nodos 1 y 4 tienen múltiples aristas).

## 2. Algoritmo Random

Realiza el posicionamiento de los nodos de manera aleatoria, teniendo únicamente de restricción la ubicación de los nodos para no acomodar nodos encima de otros. Dada esta única restricción, su complejidad es nula [11].

### 2.1. Multigrafo no dirigido cíclico

Considerando el ejemplo de la ruta de autobuses, podemos tener entre dos paradas (nodos) múltiples rutas para llegar a la parada siguiente. La figura 5 representa un ejemplo de este tipo de grafo.

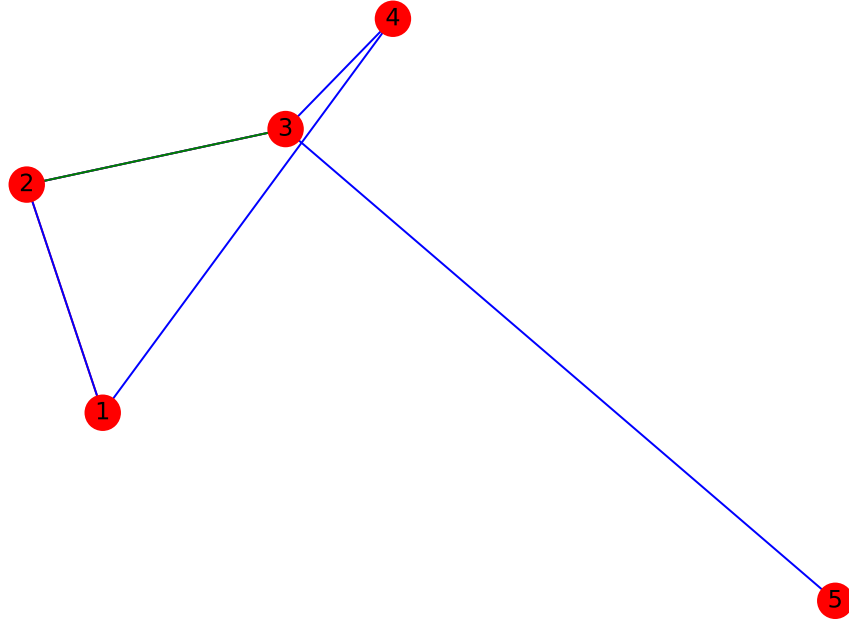


Figura 5: Multigrafo no dirigido cíclico (los nodos 1, 2 y 3 tienen múltiples aristas).

### 3. Algoritmo *Spectral*

El algoritmo *spectral* utiliza eigenvectores de la matriz del grafo (también conocido como el laplaciano). Dado un grafo ponderado, el algoritmo tiene como fin dar una relación entre los pesos de las aristas y la longitud de las aristas entre los nodos, esto es, entre mayor sea el peso de las aristas, más corta será la distancia entre los nodos [6].

### 3.1. Grafo simple dirigido reflexivo

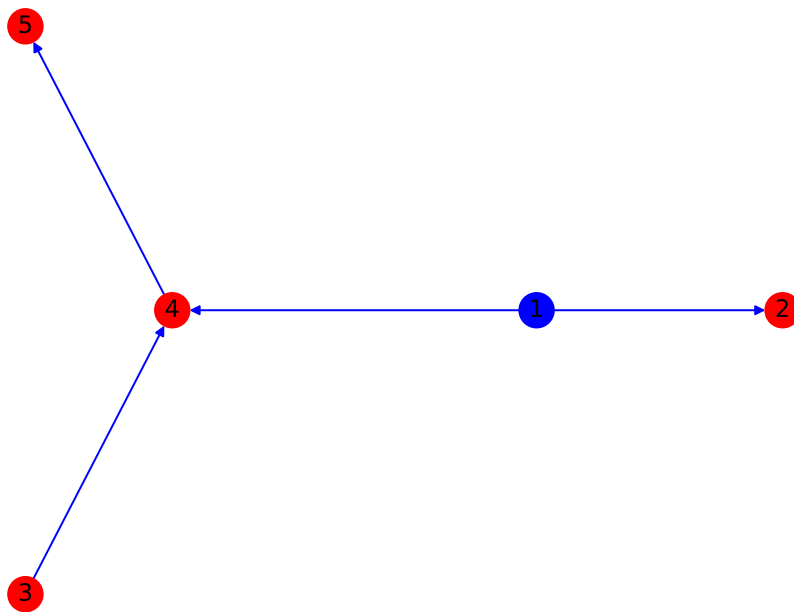


Figura 6: Grafo simple dirigido reflexivo (el nodo 1 tiene una arista reflexiva).

### 3.2. Multigrafo no dirigido acíclico

De manera similar al ejemplo de un grafo simple no dirigido acíclico, una red de ciudades puede ser representado por un multigrafo, el cual puede proporcionar más información que un grafo simple al añadir diversos caminos por el que se puede trasladar de un nodo a otro. La figura 7 representa un ejemplo de este tipo de grafo.

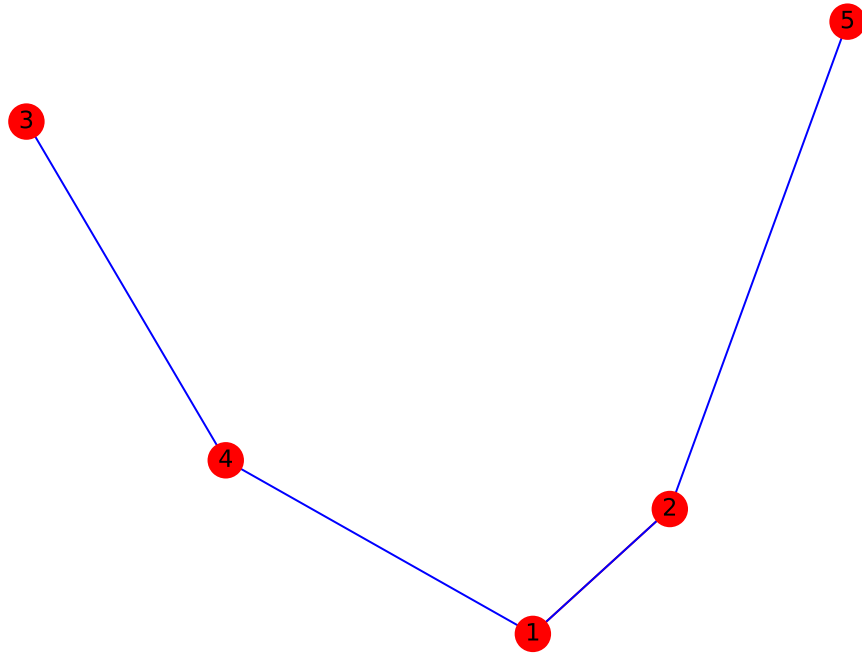


Figura 7: Multigrafo no dirigido acíclico (Los nodos 1 y 2 tienen múltiples aristas)

### 3.3. Multigrafo dirigido cíclico

Tomando el ejemplo utilizado en los grafos simples dirigidos cíclicos, en los juegos es posible que haya diferentes maneras de pasar de un estado a otro, lo cual es representado por aristas múltiples entre los nodos. La figura 8 representa un ejemplo de este tipo de grafo.



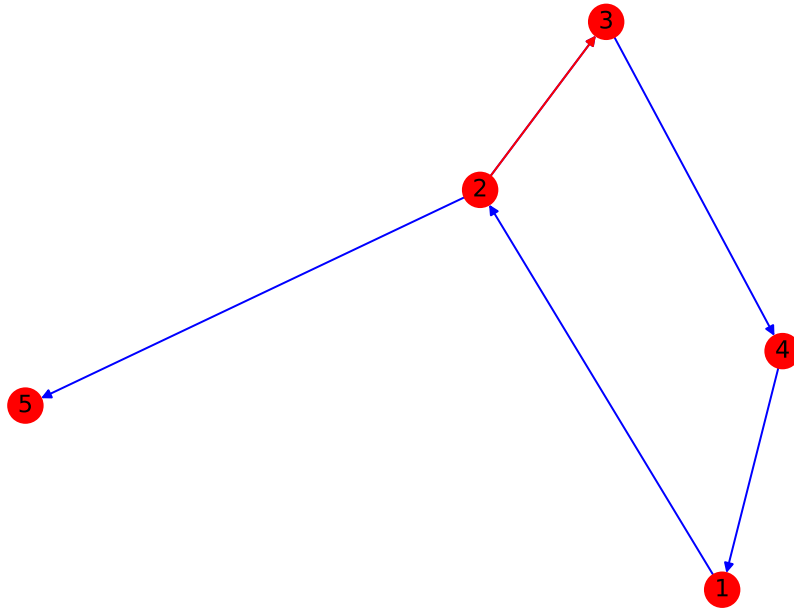


Figura 8: Multigrafo dirigido cíclico (los nodos 2 y 3 tienen múltiples aristas).

### 3.4. Multigrafo dirigido reflexivo

Como en el grafo simple dirigido reflexivo, en una página web se pueden tener múltiples hipervínculos que te redirigen a una misma página web. La figura 9 representa un ejemplo de este tipo de grafo.

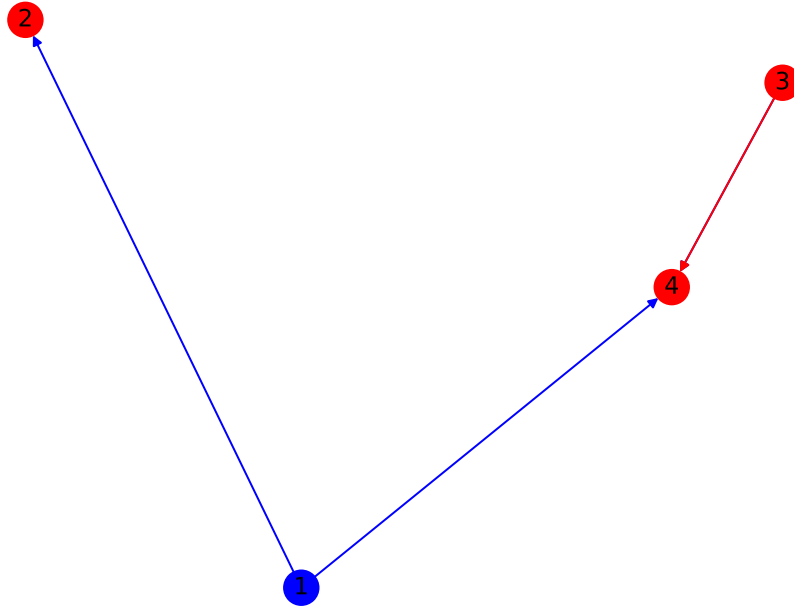


Figura 9: Multigrafo dirigido reflexivo (los nodos 3 y 4 tienen múltiples aristas, el nodo 1 tiene una arista reflexiva).

## 4. Algoritmo *Spring*

Basado en la física, el algoritmo *spring* trata de mantener que la suma de las fuerzas entre los nodos (los pesos de las aristas) sea de 0. Se basa esencialmente en la ley de Coulomb y en la métrica euclidiana para obtener estos valores, teniendo una complejidad computacional de  $O(n^2)$  [2].

### 4.1. Grafo simple no dirigido reflexivo

Podemos representar una red de sistemas informáticos por medio de un grafo reflexivo, donde cada computadora es representada por medio de un nodo, y la conexión hacia las demás computadoras son representadas por las aristas. La arista reflexiva representaría la conexión de una computadora consigo misma [14]. La figura 10 representa un ejemplo de este tipo de grafo.

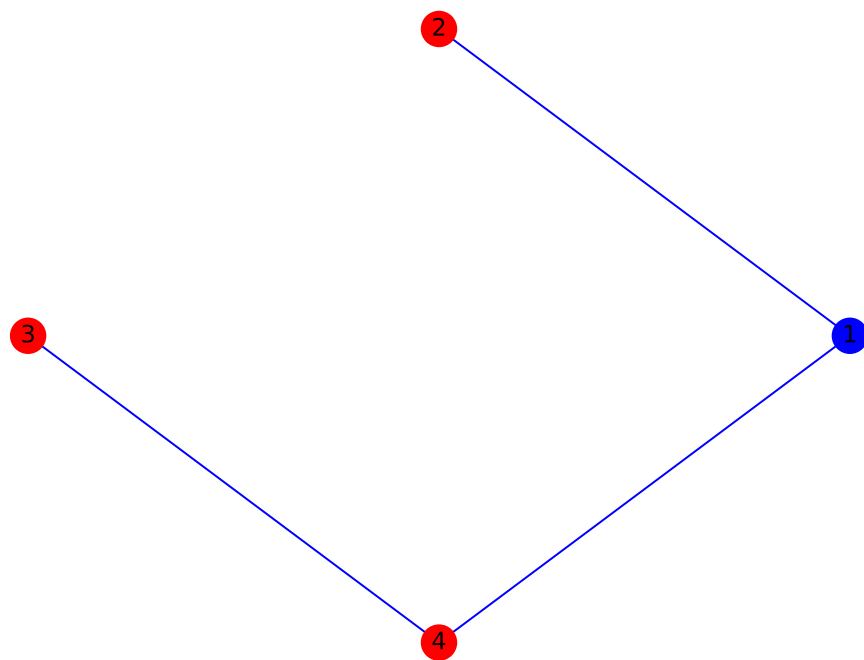


Figura 10: Grafo simple no dirigido reflexivo (el nodo 1 tiene una arista reflexiva).

## 4.2. Grafo simple dirigido acíclico

En la Programación Orientada a Objetos es común realizar herencia entre clases, y esta puede ser representada por medio de un grafo dirigido acíclico, teniendo a cada nodo como una clase distinta, y señalando la herencia por medio de una arista dirigida a las clases hijas [9]. La figura 11 representa un ejemplo de este tipo de grafo.

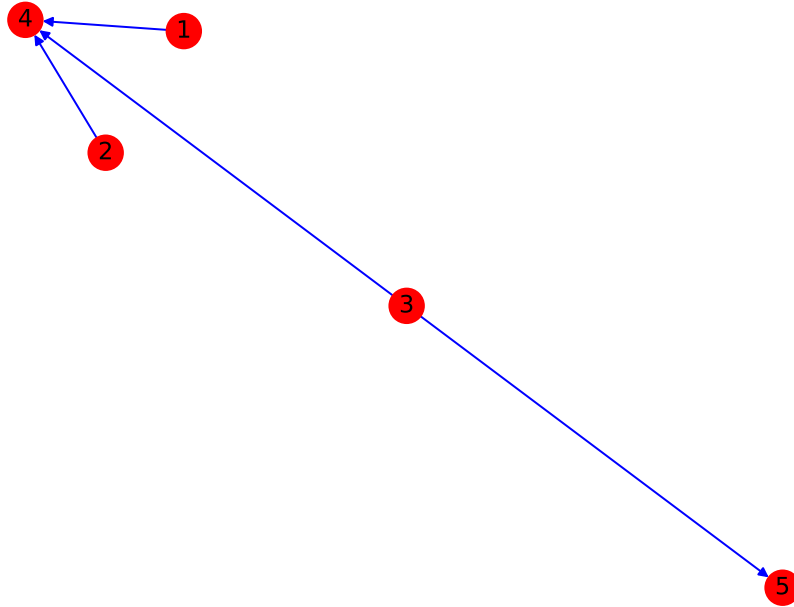


Figura 11: Grafo simple dirigido acíclico

## 5. Algoritmo *Shell*

Similar al algoritmo circular, el algoritmo *shell* realiza agrupaciones de nodos en círculos, con la diferencia que los nodos se pueden separar en distintos círculos a lo largo del grafo, por lo que la complejidad de este algoritmo es igual al algoritmo circular de  $O(n)$  siendo  $n$  el número de aristas [10].

### 5.1. Multigrafo no dirigido reflexivo

En las redes sociales, se puede utilizar un multigrafo reflexivo para representar las menciones que se hacen entre usuarios por medio de publicaciones, donde cada nodo representa un usuario y cada publicación representa la arista con la que conecta el usuario que realizó la publicación con el que es mencionado. La figura 12 representa un ejemplo de este tipo de grafo.

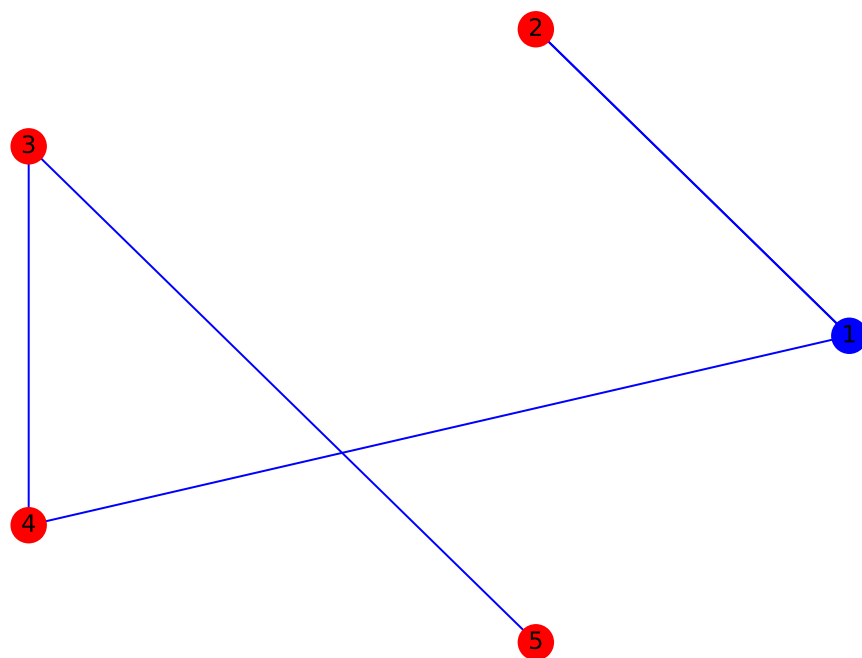


Figura 12: Multigrafo no dirigido reflexivo (los nodos 1 y 2 tienen múltiples aristas, el nodo 1 tiene una arista reflexiva).

## 6. Conclusiones entre algoritmos

Los algoritmos estudiados tienen distintos propósitos, siendo el tipo de visualización que se desea del grafo para determinar el algoritmo más adecuado. Por ejemplo, para simples pruebas de visualización se utilizaría el algoritmo *random* por su complejidad más simple entre los algoritmos, mientras que para otros tipos de grafos se utilizarían los otros tipos de algoritmos según el caso.

## Referencias

- [1] Python Software Foundation Versión 3.7.2. <https://www.python.org/>. Copyright ©2001-2019.
- [2] Artículo: “Praktikum Algorithmen-Entwurf”. <http://www.mayr.informatik.tu-muenchen.de/lehre/2012WS/algoprak/uebung/tutorial11.english.pdf>.
- [3] NetworkX developers con última actualización el 19 de Septiembre 2018. <https://networkx.github.io/documentation/stable/index.html>. Copyright ©2004-2018.

- [4] NetworkX developers Versión 2.0. <https://networkx.github.io/>. Copyright ©2004-2018.
- [5] The Matplotlib development team Versión 3.0.2. <https://matplotlib.org/>. Copyright ©2002-2012.
- [6] Artículo: “Spectral Graph Drawing”. <http://www.cis.upenn.edu/~cis515/-15-graph-drawing.pdf>.
- [7] Plotting NetworkX graph in Python Pregunta en Stackoverflow. <https://stackoverflow.com/questions/44692644/plotting-networkx-graph-in-python>. Copyright ©2019.
- [8] E. Novo and A. Méndez Alonso. Aplicaciones de la teoría de grafos a algunos juegos de estrategia. *Suma*, 46:31–35, 2004.
- [9] Example of Digraphs Applications Oxford Math Center. <http://www.oxfordmathcenter.com/drupal7/node/678>.
- [10] Documentación oficial Networkx: Shell Layout. <https://networkx.github.io/documentation/networkx-1.10/modules/networkx/drawing/layout.html#shell-layout>.
- [11] Documentación oficial Rogue Wave Software. <https://docs.roguewave.com/visualization/views/6.1/views.html#page/Options/layouts.5.1.19.html>.
- [12] Jesús Angel Patlán Castillo. Repositorio Optimización Flujo en Redes. <https://github.com/JAPatlanC/Flujo-Redes>.
- [13] «How to set colors for nodes in NetworkX» Respuesta de Abdallah So-behy. <https://stackoverflow.com/questions/27030473/how-to-set-colors-for-nodes-in-networkx-python>.
- [14] “What is the application of reflexive graph?” Respuesta de Prenoy Kumar. <https://www.quora.com/What-is-the-application-of-reflexive-graph>.
- [15] Janet M Six and Ioannis G Tollis. A framework for circular drawings of networks. In *International Symposium on Graph Drawing*, pages 107–116. Springer, 1999.