

Tarea No.1 Representación de redes a través de la teoría de grafos

Dania Danielles Suarez Suarez

1. Grafo Simple No Dirigido Aciclico

En la Dulcería "Dulce Amor", el organigrama que representa la estructura de la jerárquica de la empresa está compuesto por un nodo raíz el cual sería la Dirección, de la empresa de esta se derivan los departamentos de Producción, Contabilidad y Recursos Humanos; estos serían los nodos hijos y del departamento de Producción se desprende un nodo externo llamado grupo de trabajo [6] .

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.Graph() #Se crea un grafo vacio
5
6 G.add_node("Direccion") #Se crea el nodo raíz
7 G.add_nodes_from(["Produccion", "Contabilidad", "Recursos Humanos", "Grupo de Trabajo"])
8
9 G.add_edges_from([("Direccion", "Produccion"), ("Direccion", "Contabilidad"), ("Direccion", "Recursos Humanos")])
10 G.add_edges_from([("Produccion", "Grupo de Trabajo")])
11
12 nx.draw(G, with_labels=True) #Se dibuja el grafo
13 plt.savefig("Tarea1_01.eps")
14 plt.show() #Se dibuja en pantalla
```

Tarea1_01.py

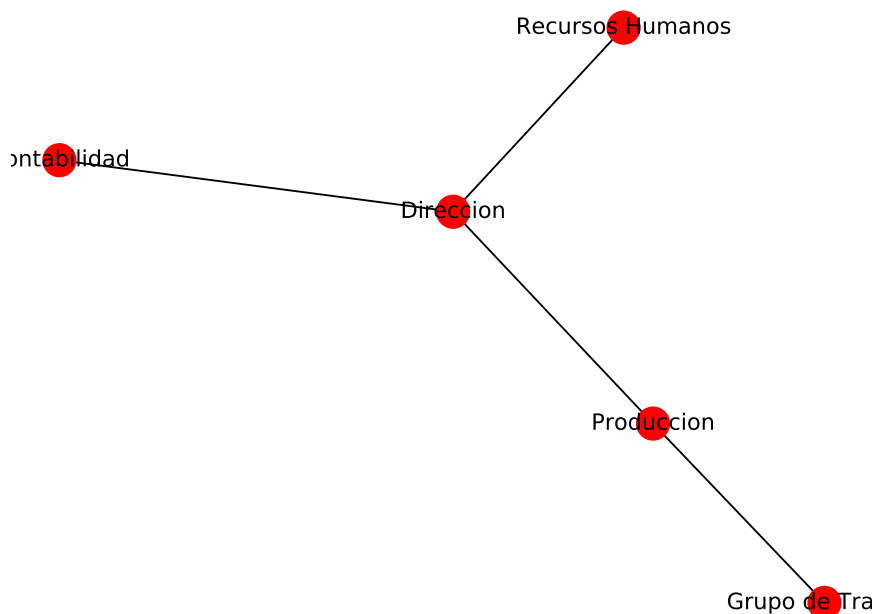


Imagen 1:Grafo Simple No Dirigido Aciclico

2. Grafo Simple No Dirigido Ciclico

Se tiene un grupo de ordenadores los cuales se quieren conectar entre ellos en una oficina, aquí los nodos serían las PC mientras los cables que la conectan sería una arista [1].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.Graph() #Se crea un grafo vacio
5
6 G.add_node(1)
7 G.add_nodes_from([2,3,4,5,6])
8 G.add_edges_from([(1,2),(1,3),(1,4),(1,5),(1,6)])
9 G.add_edges_from([(2,3)])
10 G.add_edges_from([(3,4)])
11 G.add_edges_from([(4,5)])
12 G.add_edges_from([(5,6)])
13 G.add_edges_from([(6,2)])
14
15 nx.draw(G, with_labels=True) #Se dibuja el grafo
16 plt.savefig("Tarea1_02.eps")
17 plt.show() #Se dibuja en pantalla
```

Tarea1_02.py

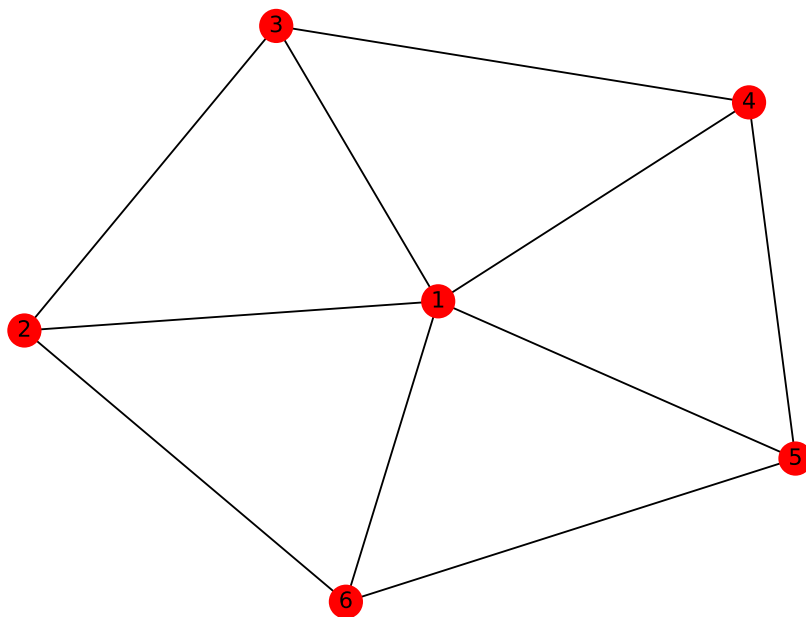


Imagen 2: Grafo Simple No Dirigido Ciclico

3. Grafo Simple No Dirigido Reflexivo

En un partido de football los jugadores serían los nodos, los cuales los pases que se realizan entre ellos conformarían las aristas, el jugador X realiza un autopase antes de darle continuidad a la pelota [4].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.Graph() #Se crea un grafo vacio
5
6 G.add_node(1) #Se crea el nodo raíz
7 G.add_nodes_from([2,3,4])
8
9 G.add_edges_from([(1,2),(2,3),(3,4),(2,2),(4,1),(2,4)])
10
11 nx.draw(G, with_labels=True) #Se dibuja el grafo
12 plt.savefig("Tarea1_03.eps")
13 plt.show() #Se dibuja en pantalla
```

Tarea1_03.py

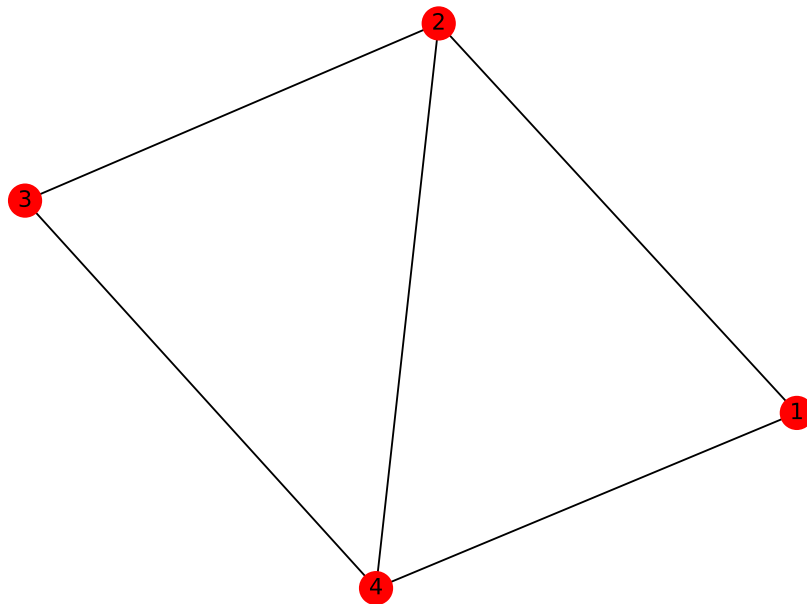


Imagen 3:Grafo Simple No Dirigido Reflexivo

4. Grafo Simple Dirigido Aciclico

Una colección de tareas que deben ordenarse con cierta secuencia, esta está sujeta a limitaciones dado que una tarea no puede realizarse antes que otras; tendría un vértice para cada tarea y un borde para cada restricción, la forma en que se puede ordenar es de manera topológica [12].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.DiGraph() #Se crea un grafo vacio
5
6 G.add_node(1)
7 G.add_nodes_from([3,2,4,5])
8 G.add_edges_from([(1,2),(1,3),(1,5),(1,5)])
9 G.add_edges_from([(2,4)])
10 G.add_edges_from([(3,2),(3,5),(4,5)])
11 G.add_edges_from([(4,5)])
12
13 nx.draw(G, with_labels=True) #Se dibuja el grafo
14 plt.savefig("Tarea1_04.eps")
15 plt.show() #Se dibuja en pantalla
```

Tarea1_04.py

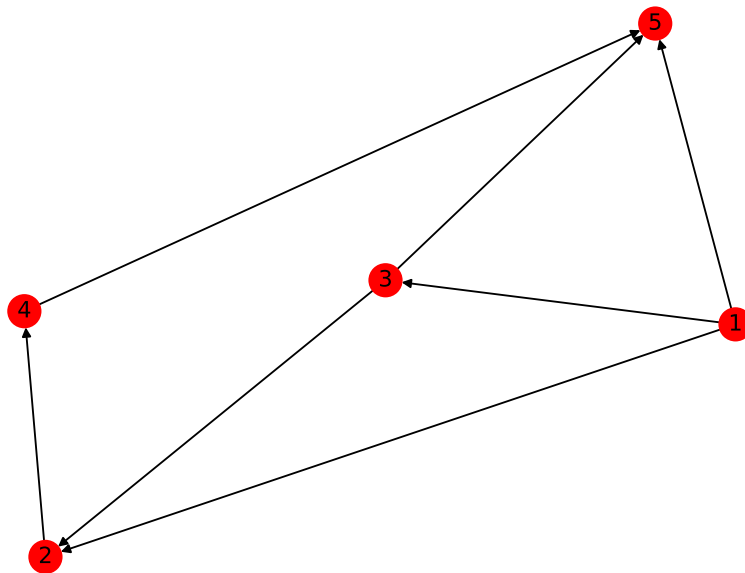


Imagen 4:Grafo Simple Dirigido Aciclico

5. Grafo Simple Dirigido Ciclico

Una Empresa "X" que se dedica a comercializar productos de limpieza necesita hacer la distribución de la mercancía, para esto se requiere encontrar un recorrido que garantice que se recorrerá todos los nodos de la red de modo que se minimice la distancia total recorrida y al final de la distribución el método de transporte vuelva al origen [8].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.DiGraph()
5
6 G.add_node("X")
7 G.add_nodes_from([1,2,3,4,5,6])
8
9 G.add_edges_from([("X",2),(2,4),(4,6),(6,5),(5,3),(3,1),(1,"X")])
10
11 nx.draw(G, with_labels=True) #Se dibuja el grafo
12 plt.savefig("Tarea1_05.eps")
13 plt.show() #Se dibuja en pantalla
```

Tarea1_05.py

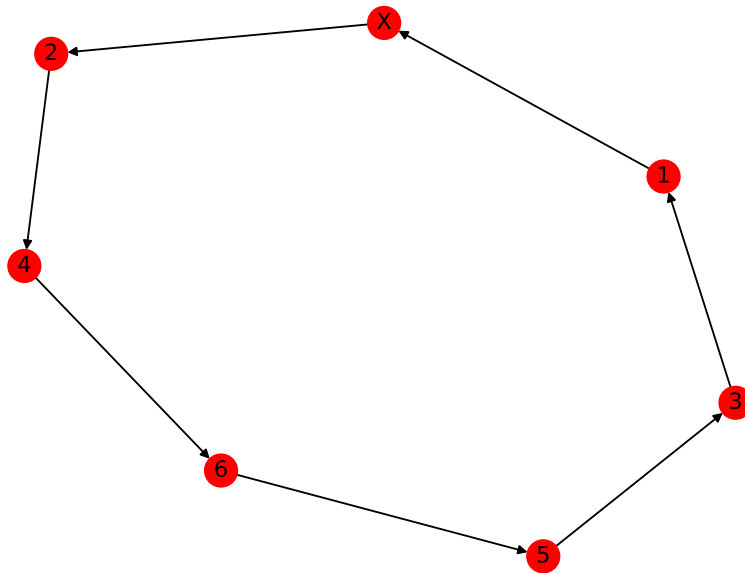


Imagen 5: Grafo Simple Dirigido Ciclico

6. Grafo Simple Dirigido Reflexivo

Sea el conjunto $A = \{x, y, z\}$. El grafo representa una relación binaria definida en A , puesto que los pares (x, z) , (y, x) (y, y) constituyen un subconjunto de $A \times A$ [11].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.DiGraph() #Se crea un grafo vacio
5
6 G.add_node("x")
7 G.add_nodes_from(["y", "z", "w", "s", "t"])
8 G.add_edges_from([("t", "s"), ("s", "w"), ("s", "z"), ("z", "x"), ("x", "y"), ("y", "y")])
9
10 nx.draw(G, with_labels=True) #Se dibuja el grafo
11 plt.savefig("Tarea1_06.eps")
12 plt.show() #Se dibuja en pantalla
```

Tarea1_06.py

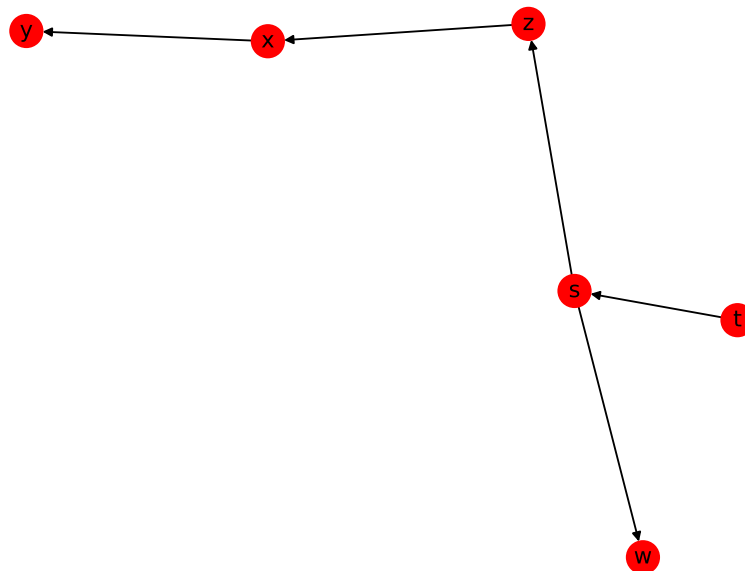


Imagen 6: Grafo Simple Dirigido Reflexivo

7. Multigrafo No Dirigido Aciclico

Un corredor sale desde su punto de partida pasando por varias paradas, al llegar a la primera parada hay dos alternativas de rutas para llegar al próximo destino, en este ejemplo las paradas constituyen los nodos mientras el recorrido entre los nodos son las aristas [2].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.MultiGraph() #Se crea un grafo vacio
5
6 G.add_node("A") #Se crea el nodo raíz
7 G.add_nodes_from(["B", "C", "D", "E"])
8
9 G.add_edges_from([("A", "B"), ("B", "C"), ("B", "C"), ("C", "D"), ("D", "E")])
10
11 nx.draw(G, with_labels=True) #Se dibuja el grafo
12 plt.savefig("Tarea1_07.eps")
13 plt.show() #Se dibuja en pantalla
```

Tarea1_07.py

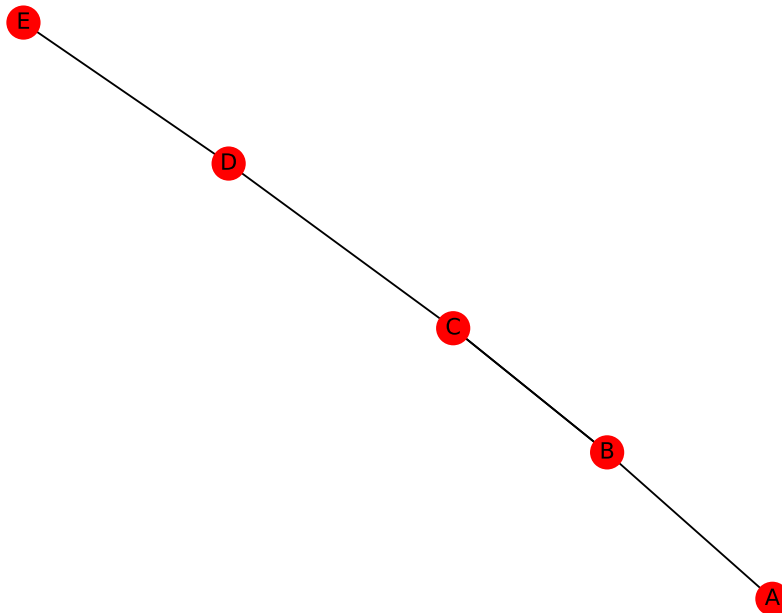


Imagen 7:Multigrafo No Dirigido Aciclico

8. Multigrafo No Dirigido Ciclico

Las llamadas telefónicas es una aplicación en la vida práctica de los grafos dado que estas se pueden representar mediante una red; cada número telefónico sería un nodo y cada llamada sería una arista, esta sale desde el número de teléfono que hace la llamada hasta el número que la recibe [7].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G=nx.MultiDiGraph()
5
6 G.add_node(1)
7 G.add_nodes_from([2,3,4,5])
8 H=nx.Graph()
9 G.add_edges_from([(1,2),(2,1),(1,3),(4,5),(4,1)])
10
11 nx.draw(G, with_labels=True) #Se dibuja el grafo
12 plt.savefig("Tarea1_08.eps")
13 plt.show() #Se dibuja en pantalla
```

Tarea1_08.py

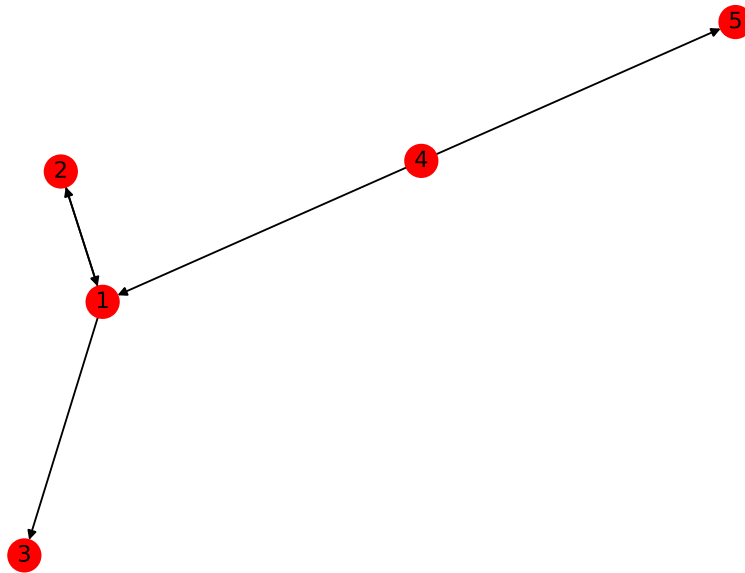


Imagen 8: Multigrafo No Dirigido Ciclico

9. Multigrafo No Dirigido Reflexivo

En el juego del Policías y Ladrones el laberinto que conforma este juego está representado con un grafo, donde las calles son las aristas donde se mueven los policías y los ladrones de vértice en vértice los cuales están ubicados en las intersecciones de las calles. Este juego se juega en grafos reflexivos, que son aquellos cuyos vértices tienen bucles, de esta forma, los jugadores pueden permanecer en el mismo vértice varias rondas seguidas y también permite que contenga aristas múltiples [9].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.MultiDiGraph() #Se crea un grafo vacio
5
6 G.add_node(1) #Se crea el nodo raíz
7 G.add_nodes_from([2,3,4,5,6,7,8,9])
8
9 G.add_edges_from([(1,2),(2,1),(2,3),(3,2),(3,4),(4,3),(3,5),(5,3),(5,6),(6,5),(6,7),(7,6),
10                  ,(7,8),(8,7),(7,9),(9,7)])
11
12 nx.draw(G, with_labels=True) #Se dibuja el grafo
13 plt.savefig("Tarea1_09.eps")
14 plt.show() #Se dibuja en pantalla
```

Tarea1_09.py

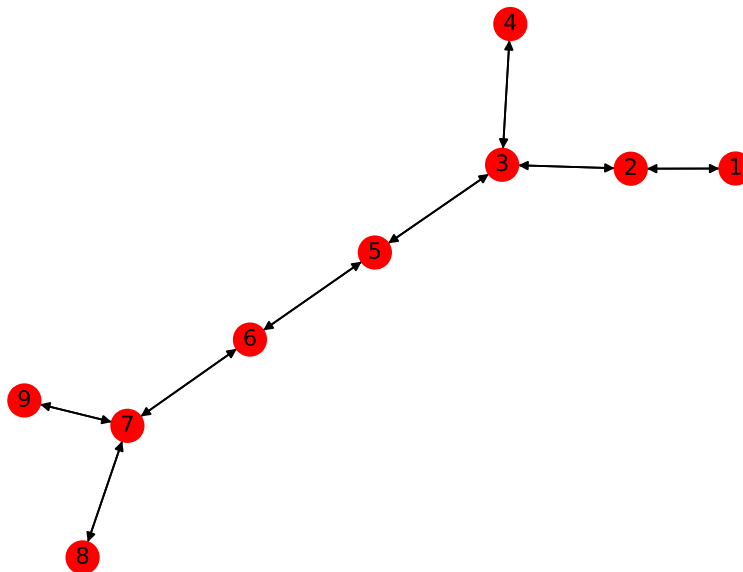


Imagen 9:Multigrafo No Dirigido Reflexivo

10. Multigrafo Dirigido Aciclico

Se puede considerar una representación grafica de un mapa de carreteras en el cual una arista (con flecha de sentido) entre dos ciudades corresponde a un carril en una autopista entre las dos ciudades. Como a menudo hay autopistas de varios carriles entre pares de ciudades, esta representación origina un multígrafo [3].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.MultiDiGraph() #Se crea un grafo vacio
5
6 G.add_node("A") #Se crea el nodo raíz
7 G.add_nodes_from(["C", "B", "R", "G"])
8
9 G.add_edges_from([("A", "C"), ("C", "A"), ("C", "B"), ("B", "C"), ("B", "R"), ("R", "B"), ("R", "G")])
10
11 nx.draw(G, with_labels=True) #Se dibuja el grafo
12 plt.savefig("Tarea1_10.eps")
13 plt.show() #Se dibuja en pantalla
```

Tarea1_10.py

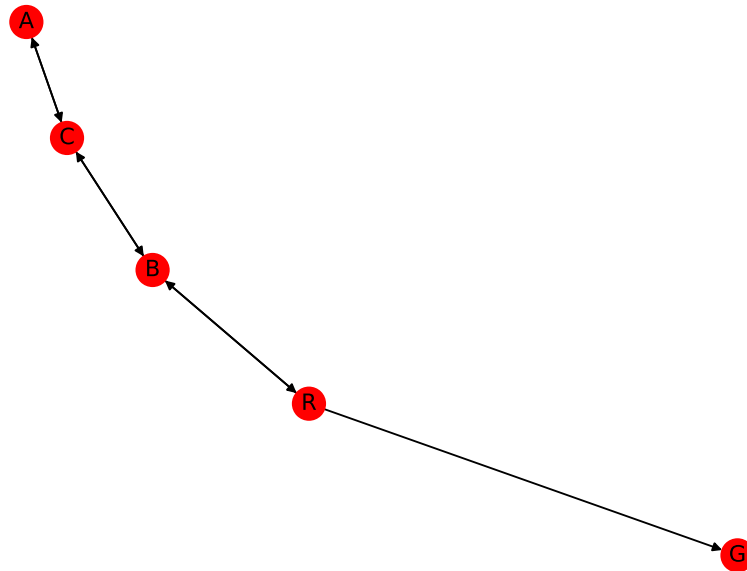


Imagen 10:Multigrafo Dirigido Aciclico

11. Multigrafo Dirigido Ciclico

Para modelar las posibles conexiones de vuelo ofrecidas por una aerolínea se tendría un multígrafo dirigido, donde cada nodo es una localidad y donde pares de aristas paralelas conectan estas localidades, según un vuelo es hacia o desde una localidad a la otra [10].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.MultiDiGraph() #Se crea un grafo vacio
5
6 G.add_node(1) #Se crea el nodo raíz
7 G.add_nodes_from([2,3,4,5])
8
9 G.add_edges_from([(1,2),(2,1),(2,3),(3,4),(4,5),(5,2)])
10
11 nx.draw(G, with_labels=True) #Se dibuja el grafo
12 plt.savefig("Tarea1.11.eps")
13 plt.show() #Se dibuja en pantalla
```

Tarea1.11.py

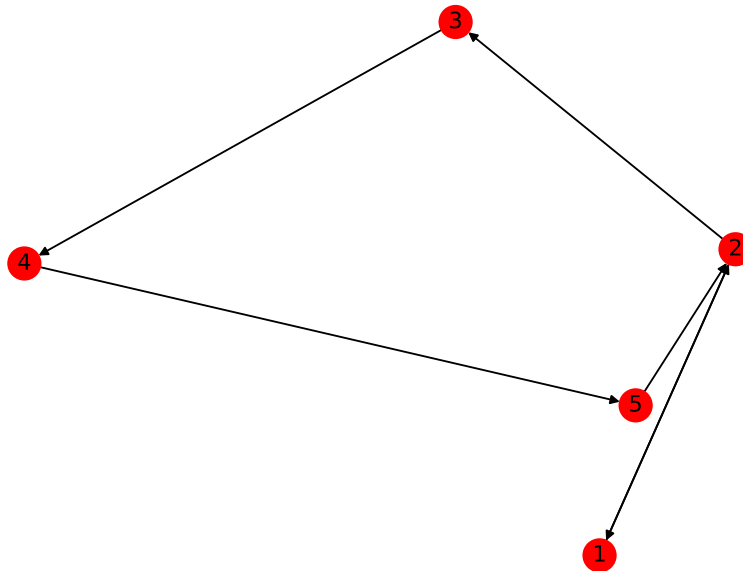


Imagen 11:Multigrafo Dirigido Ciclico

12. Multigrafo Dirigido Reflexivo

Una ruta de camión empieza su recorrido en la mañana donde tiene dos alternativas de rutas para comenzar, cuando llega al final del recorrido descarga al personal y comienza nuevamente su recorrido en ese mismo lugar con personal que se encuentre en esa parada. Los nodos aquí serían las paradas y las aristas las calles que unen cada parada [5].

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.MultiDiGraph() #Se crea un grafo vacio
5
6 G.add_node(1) #Se crea el nodo raíz
7 G.add_nodes_from([2,3,4,5])
8
9 G.add_edges_from([(1,2),(1,2),(2,3),(3,4),(3,5),(4,5),(5,1)])
10
11 nx.draw(G, with_labels=True) #Se dibuja el grafo
12 plt.savefig("Tarea1.12.eps")
13 plt.show() #Se dibuja en pantalla
```

Tarea1.12.py

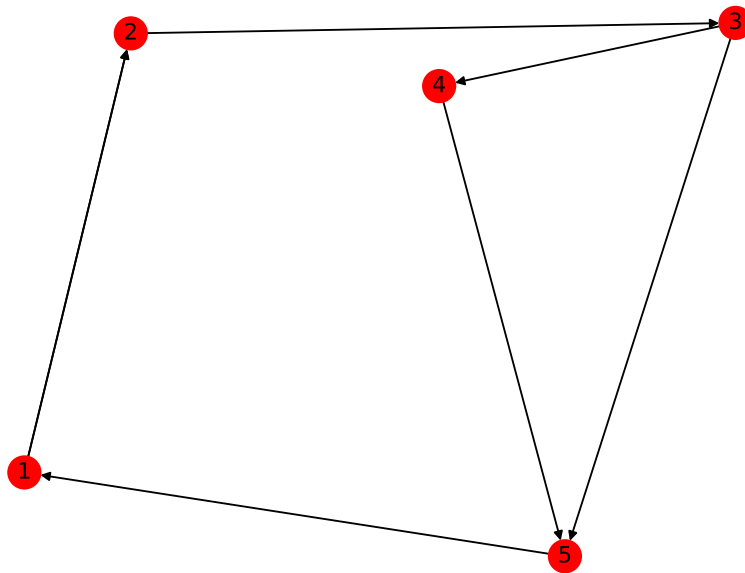


Imagen 12:Multigrafo Dirigido Reflexivo

Referencias

- [1] Armando de Jesus Ruiz Calderon, Sofia Barron Perez, Abel Gonzalez Canas, Daniel Roberto Del Toro Arias, and Argel Frias Escudero. Modelado de grafos en la red.
- [2] ESTER MARTOS CARRION. *Analisis sobre las nuevas formas de comunicacion a traves de las comunidades virtuales o redes sociales*. PhD thesis, 2011.
- [3] Maria del Carmen Somoza Lopez. *Estructuras metricas en grafos*. PhD thesis, Universidade de Vigo, 2015.
- [4] Natalia Martinez, Gheisa Ferreira, and Zoila Zenaida Garcia. Sistema de ensenanza/aprendizaje inteligente para grafos. 2007.
- [5] Johani Olivos Iparraguirre. Algoritmos para caminos minimos. 2009.
- [6] Jorge J Frias Perles. Grafos:las redes que mueven el mundo.
- [7] CAROLINA RAMIREZ. Aplicaciones de grafos.
- [8] Antonio Rifon Sanchez. Los ordenes semanticos. 2009.
- [9] Blanca Maria Pozuelo Rollon. Juego de policias y ladrones en grafos.
- [10] Jorge E SAGULA and Rene J TESEYRA. Algoritmo de busqueda de rutas con preferencias. 2011.
- [11] Andres Ramos Pedro Linares Pedro Sanchez Angel Sarabia, Begona Vitoriano. Teoria de grafos o redes.
- [12] Fabiola Werlinger and Dante D Caceres. Aplicacion de grafos aciclicos dirigidos en la evaluacion de un set minimo de ajuste de confusores: un complemento al modelamiento estadistico en estudios epidemiologicos observacionales. *Revista medica de Chile*, 146(7):907–913, 2018.