

# Representación de un Grafo

## Funciones

- lista\_ady\_prueba
- insertar\_nodo
- eliminar\_nodo
- insertar\_arco
- eliminar\_arco
- graficar
- graficar\_peso

### **lista\_ady\_prueba():**

Función: Devuelve un dictionary de dictionarys que representan los nodos con sus nodos adyacentes, así como sus pesos, Ej. grafo con 10 nodos:

```
nodos={0:{1:12,6:16,8:2},
        1:{0:16,8:8,4:12,6:18,9:1},
        2:{4:12,6:8},
        3:{2:10,5:16,8:8},
        4:{1:4,2:6,3:10,5:18,9:2},
        5:{3:8,4:6},
        6:{0:2,1:4,2:1},
        7:{8:18,9:16},
        8:{0:6,3:14,7:1},
        9:{1:8,4:14,7:1}}
```

Return: Devuelve el dictionary

### **insertar\_nodo (dictionary , int):**

Parámetros: Recibe un dictionary donde se insertara el nodo y el nodo a insertar.

Función: Inserta un nodo en el dictionary en caso de que no exista.

Return: Devuelve True si el nodo se inserto, False si no se inserto.

### **eliminar\_nodo(dictionary, int):**

Parámetros: Recibe un dictionary de donde se eliminara el nodo y el nodo a eliminar.

Función: Elimina un nodo , siempre y cuando exista.

Return: Devuelve True si el nodo se elimino, False si no se elimino.

### **insertar\_arco (dictionary, int1, int2, int3):**

Parámetros: Recibe un dictionary donde se insertara un arco del nodo1(int1) al nodo2(int2) junto con su peso(int3).

Función: Insertar un arco en el grafo siempre y cuando los nodos existan.

Return: Devuelve True si se creo el arco , False en caso contrario.

### **eliminar\_arco (dictionary, int1, int2):**

Parámetros: Recibe un dictionary de donde se eliminara el arco del nodo1(int1) al nodo2(int2).

Función: Eliminar un arco entre nodo1 al nodo2 siempre y cuando exista.

Return: Regresa True si se elimino el arco , False en caso contrario.

**graficar(dictionary):**

Parámetros: Recibe el dictionary que se va a dibujar.

Función: Dibujar en pantalla el grafo y guardar la imagen con el nombre de Grafo.png, el grafo se puede dibujar como dirigido o no dirigido con las opciones:

#GAFO DIRIGIDO

`nx.draw_circular(G, arrows=True, with_labels=True)`

`nx.draw_spring(G, arrows=True, with_labels=True)`

#GRAFO NO DIRIGIDO

`nx.draw_shell(G, arrows=False, with_labels=True)`

**graficar\_peso(dictionary):**

Parámetros: Recibe el dictionary que se va a dibujar.

Función: Dibujar en pantalla el grafo y guardar la imagen con el nombre de Grafo.png, el grafo es no dirigido.