

MANUAL TÉCNICO Y USUARIO

Nombre: Jencer Hamilton Hernández Alonzo.

Carnet: 202002141.

Nombre: Billy David Must Ochoa.

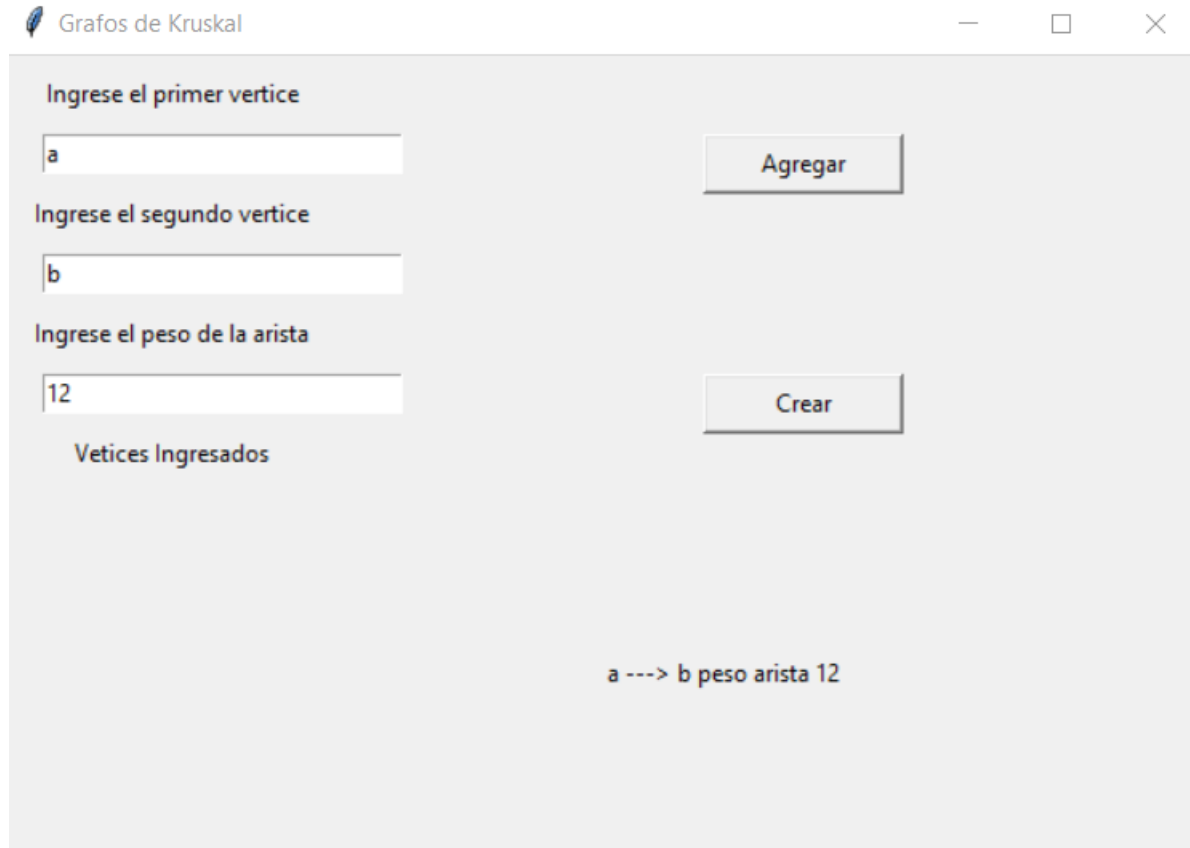
Carnet: 202000353.

Se debe Instalar previamente el programa donde se ejecutará el programa.

En la pantalla de inicio, se muestran las cajas donde se agregará la arista, desde el punto “a” al punto “b” y el peso que tendrá dicha arista.

Podrá agregar cuantas aristas quiera y se mostrará en un label la información ingresada anteriormente.

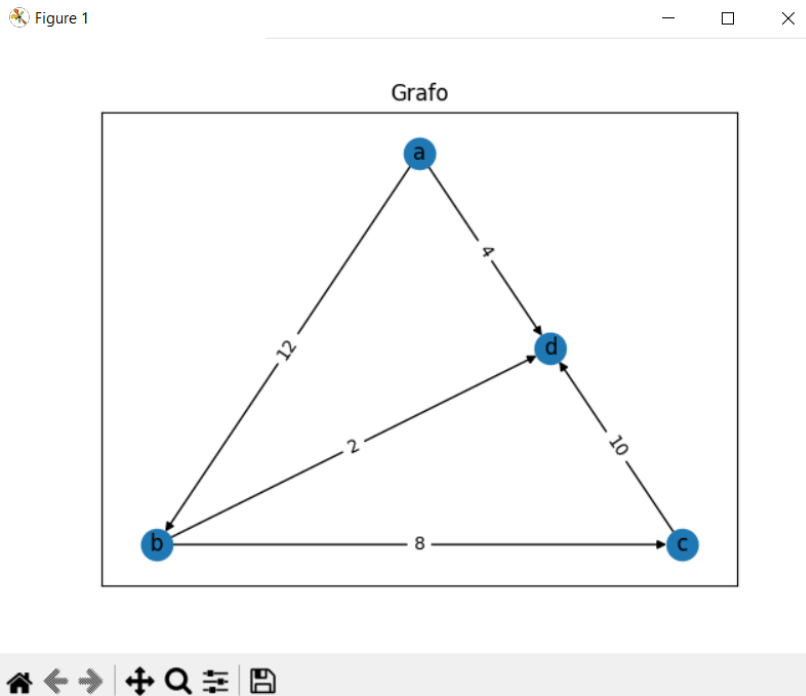
El botón de agregar se utilizará para agregar la arista.



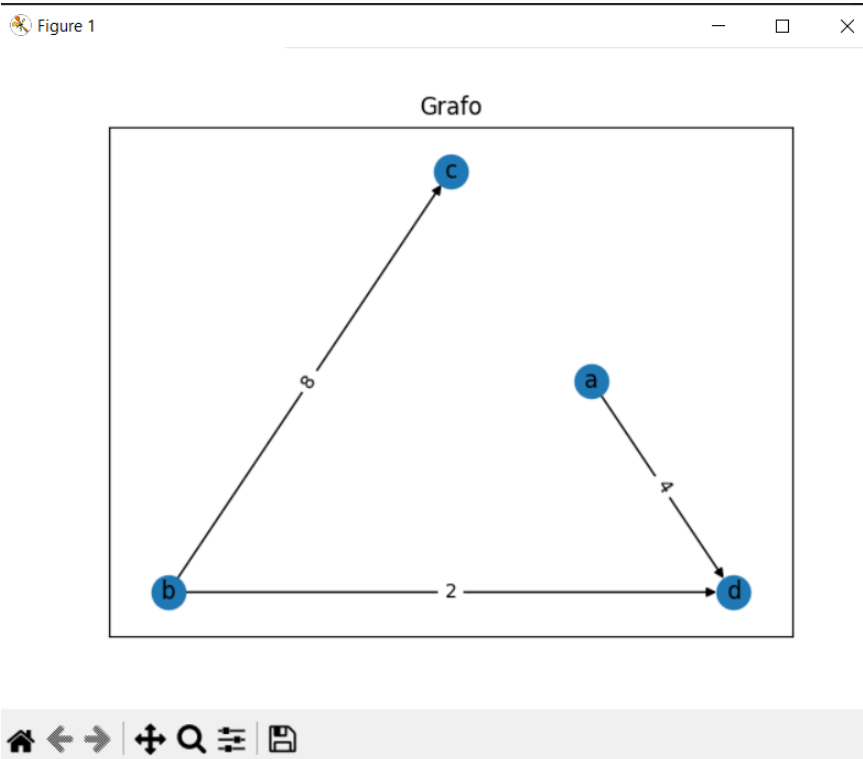
The screenshot shows a window titled "Grafos de Kruskal" with standard Windows window controls (minimize, maximize, close). The window contains the following elements:

- Input fields:**
 - "Ingrese el primer vertice": A text box containing the letter "a".
 - "Ingrese el segundo vertice": A text box containing the letter "b".
 - "Ingrese el peso de la arista": A text box containing the number "12".
- Buttons:**
 - "Agregar": A button located to the right of the first two input fields.
 - "Crear": A button located to the right of the third input field.
- Labels:**
 - "Vetices Ingresados": A label positioned below the input fields.
- Output:**
 - A label at the bottom center of the window displays the text "a ---> b peso arista 12".

El botón de crear, creará el grafo y con ello la visualización del grafo.



Al cerrar la pestaña del grafo se creará un grafo de kruskal.



Se utilizó la librería networkx con el objetivo de graficar el grafo.

```
import networkx as nx
```

Se usó una función donde G es el grafo, U sería el vértice 1, V el vértice 2 y W sería el peso, si el grafo no es dirigido se agrega otra arista en sentido contrario.

```
def agregar_arista(G, u, v, w=1, di=True):
    G.add_edge(u, v, weight=w)
    # Si el grafo no es dirigido
    if not di:
        # Agrego otra arista en sentido contrario
        G.add_edge(v, u, weight=w)
```

Se ordenó el archivo en orden inverso.

```
Ordenada = [(a,b,c) for c,a,b in Ordenada]
```

Con este método se verifica cuál la arista con el menor peso del grafo y luego se guarda en un arreglo.

```
for Dato in Ordenada:
    peso, u, v = Dato
    if Find_set(u) != Find_set(v):
        resultante.append(Dato)
        print ("=====")
        print ("Paso:", cont)
        print ("=====")
        resultante = [(a,b,c) for c,a,b in resultante]
        print ("Resultante: ", resultante)
        resultante = [(c,a,b) for a,b,c in resultante]
        cont+=1
        Union(u, v, Ordenada)
return resultante
```