**UNIVERSIDAD DE PANAMÁ**

**CENTRO REGIONAL UNIVERSITARIO DE COCLÉ**

**FACULTAD DE INFORMÁTICA, ELECTRÓNICA Y COMUNICACIÓN**

**LICENCIATURA EN INGENIERÍA EN INFORMÁTICA**

**INFORMÁTICA TEÓRICA**

**PROYECTO FINAL**

**SEMESTRAL**

**ASIGNATURA**

**INFORMÁTICA TEÓRICA**

**PROFESOR**

**LUIS DOMÍNGUEZ**

**ESTUDIANTES:**

**NITSY BALABARCA**

**2-750-941**

**IVYS SÁNCHEZ**

**8-996-1251**

**II SEMESTRE**

**PENONOMÉ, DICIEMBRE DE 2023**

**PROYECTO FINAL – SEMESTRAL**

### Descripción del Proyecto:

Diseñar e implementar un sistema en Python que utilice el algoritmo de Dijkstra para encontrar la ruta más corta entre ciudades en la provincia de Coclé, Panamá. El proyecto incluirá la representación de las ciudades y sus conexiones mediante un grafo, permitiendo a los usuarios determinar la distancia más corta entre dos ubicaciones. Para ello investigué en internet, las distancia entre las ciudades.

**Modalidad:** Grupos de 2 estudiantes.

***Sugerencias a la hora de ejecutar el código:***

* Apertura de la Aplicación: Al ejecutar este código, se abrirá una ventana de interfaz gráfica con el título "Calculadora de Ruta Más Corta". Esta ventana es la aplicación a utilizar.
* Ingreso de Distritos: Verás dos etiquetas que dicen "Distrito de Inicio" y "Distrito de Destino". Estas etiquetas indican que debes ingresar los nombres de los distritos desde donde quieres comenzar y a dónde quieres llegar. Para hacer esto, simplemente escribe el nombre del distrito en las cajas de texto (entradas) correspondientes. Aquí solo se permiten ingresar las siguientes opciones: “Aguadulce”, “Natá”, “Olá”, “Penonomé”, “La Pintada”, “Antón”.
* Cálculo de la Ruta Más Corta: Después de ingresar los distritos de inicio y destino, haz clic en el botón que dice "Calcular Ruta". Este botón tiene la función de ejecutar el cálculo de la ruta más corta entre los distritos que ingresaste.
* Visualización del Resultado: La aplicación calculará automáticamente la ruta más corta y mostrará el resultado en la ventana. En la parte inferior, verás un mensaje que indica el camino corto y la distancia que toma en kilómetros.
* Además, se abrirá una nueva ventana con un gráfico que representa visualmente la red de distritos y resalta la ruta más corta en morado. También se guardará una imagen de este gráfico con el nombre "Graph.png".
* Cierre de la Aplicación: Puedes cerrar la aplicación haciendo clic en el botón de cierre en la esquina superior derecha de la ventana.

***Código:***

# Semestral - Nitsy Balabarca e Ivys Sánchez

# Importar las bibliotecas necesarias

import networkx as nx

import matplotlib.pyplot as plt

import tkinter as tk

from tkinter import ttk

# Función para calcular y mostrar la ruta más corta

def calcular\_ruta():

# Obtener los nombres de inicio y destino desde las entradas de la interfaz gráfica

inicio = entrada\_inicio.get()

destino = entrada\_destino.get()

# Calcular la ruta más corta y su tiempo

camino\_corto = nx.shortest\_path(Grafo, inicio, destino, weight='weight')

tiempo = nx.shortest\_path\_length(Grafo, inicio, destino, weight='weight')

leyenda = f"El Camino Corto es: {camino\_corto} y toma {tiempo} Kilómetros"

label\_resultado.config(text=leyenda)

# Especificar posiciones de nodos manualmente

pos = {

"Aguadulce": (0.002, 0.004),

"Penonomé": (1.151, 0.959),

"Antón": (1.987, 0.560),

"Natá": (0.687,0.0361),

"La Pintada": (0.199,0.944),

"Olá": (0.251, 0.555)

}

# Crear y mostrar el gráfico con la ruta más corta

nx.draw(Grafo, pos, node\_size=1300, node\_color='pink', font\_size=8, font\_weight='bold', with\_labels=True)

nx.draw\_networkx\_edge\_labels(Grafo, pos, edge\_labels=nx.get\_edge\_attributes(Grafo, "weight"))

# Resaltar la ruta más corta en color morado

edge\_list = [(camino\_corto[i], camino\_corto[i + 1]) for i in range(len(camino\_corto) - 1)]

nx.draw\_networkx\_edges(Grafo, pos, edgelist=edge\_list, edge\_color='purple', width=2)

# Configurar título y guardar la visualización como una imagen PNG

plt.title("Grafo con Ruta Más Corta Entre Los Distritos de la Provincia de Coclé")

plt.savefig("Graph.png", format="PNG")

plt.show()

# Crear el grafo

Grafo = nx.Graph()

Grafo.add\_node("Aguadulce")

Grafo.add\_node("Penonomé")

Grafo.add\_node("Antón")

Grafo.add\_node("Natá")

Grafo.add\_node("La Pintada")

Grafo.add\_node("Olá")

# Agregar nodos y aristas al grafo

Grafo.add\_edge("Antón", "Penonomé", weight=34.9)

Grafo.add\_edge("Penonomé", "La Pintada", weight=15.3)

Grafo.add\_edge("La Pintada", "Olá", weight=55.5)

Grafo.add\_edge("Natá", "Olá", weight=37.6)

Grafo.add\_edge("Natá", "Aguadulce", weight=19.3)

Grafo.add\_edge("Natá", "Penonomé", weight=27.1)

Grafo.add\_edge("Natá","La Pintada", weight=61.0)

Grafo.add\_edge("Antón", "Natá", weight=63.2)

# Crear la interfaz gráfica con Tkinter

ventana = tk.Tk()

ventana.title("Calculadora de Ruta Más Corta")

# Crear etiquetas y entradas para ingresar los distritos de inicio y destino

etiqueta\_inicio = ttk.Label(ventana, text="Distrito de Inicio:")

etiqueta\_inicio.grid(column=0, row=0, padx=10, pady=10)

entrada\_inicio = ttk.Entry(ventana)

entrada\_inicio.grid(column=1, row=0, padx=10, pady=10)

etiqueta\_destino = ttk.Label(ventana, text="Distrito de Destino:")

etiqueta\_destino.grid(column=0, row=1, padx=10, pady=10)

entrada\_destino = ttk.Entry(ventana)

entrada\_destino.grid(column=1, row=1, padx=10, pady=10)

# Crear botón para calcular la ruta

boton\_calcular = ttk.Button(ventana, text="Calcular Ruta", command=calcular\_ruta)

boton\_calcular.grid(column=0, row=2, columnspan=2, pady=10)

# Etiqueta para mostrar el resultado

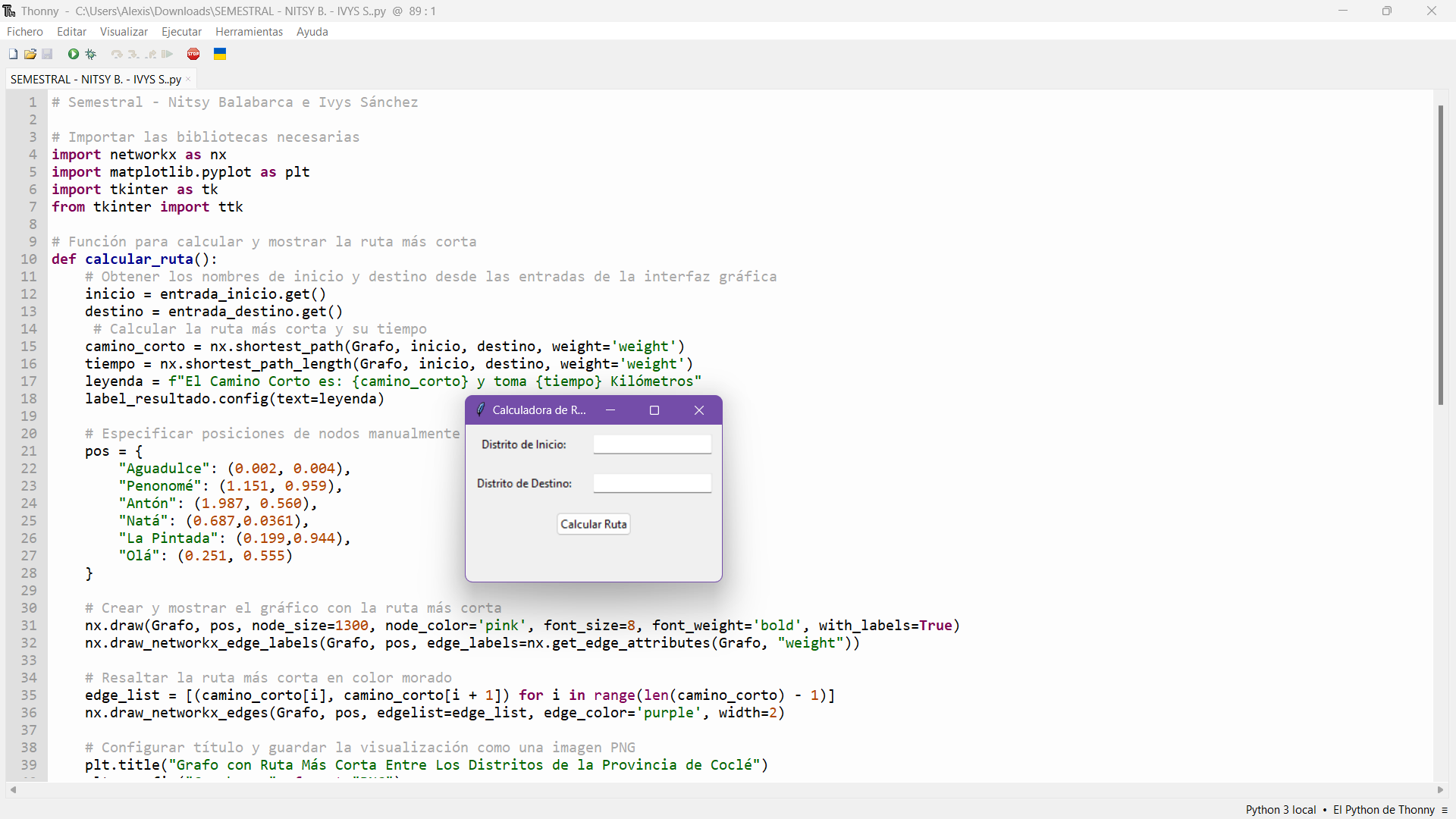
label\_resultado = ttk.Label(ventana, text="")

label\_resultado.grid(column=0, row=3, columnspan=2, pady=10)

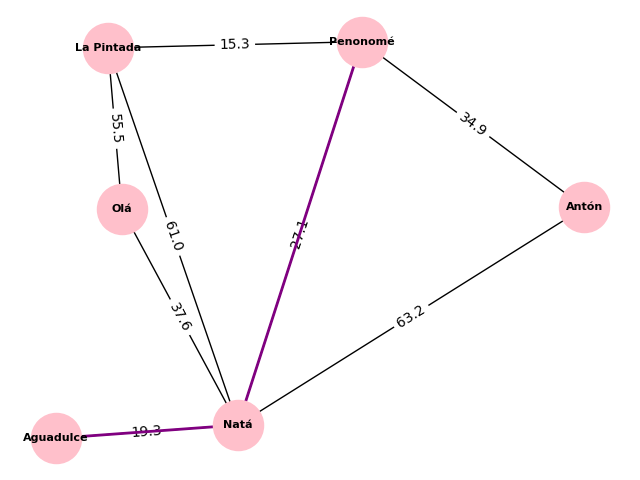
# Iniciar el bucle principal de la interfaz gráfica

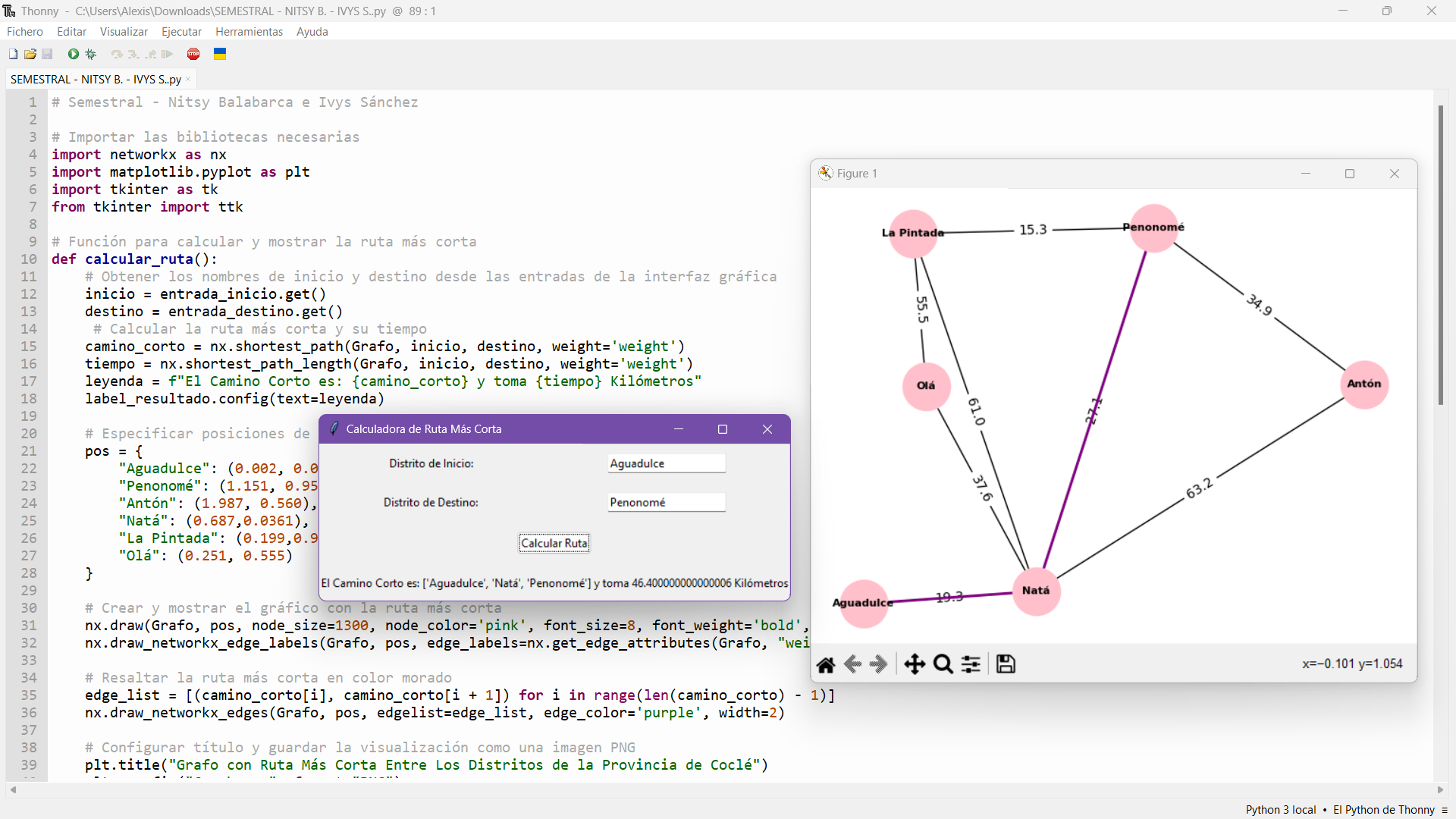
ventana.mainloop()

***Imagen de la interfaz:***



***Diagrama de Camino Rápido que visualice la ruta más corta en el grafo (PNG/JPG):***

******

***Vista de todo el programa:***