

UNIVERSIDAD DE LA COSTA CUC
RUBRICA SEGUNDO CORTE 2025-1
CURSO ALGORITMO II
UNIDAD TEMÁTICA: GRAFOS
FECHA DE ENTREGA: VIERNES 25 DE ABRIL DEL 2025

1. Descripción del proyecto a realizar

- Desarrollar en Python el algoritmo que se describe en el punto 2.
- Nombrar el archivo de la siguiente forma: **nombreakellido.py**
- Subir el archivo a Teams en el enlace correspondiente.

2. Sistema de Rutas Óptimas en una Ciudad Inteligente

Desarrollar un sistema que calcule la **ruta más corta** entre dos ubicaciones dentro de una ciudad, considerando una red vial representada como un **grafo ponderado y NO dirigido**.

Este sistema se puede aplicar en:

- Plataformas de navegación como Google Maps o Waze.
- Ruteo de vehículos de emergencia.
- Optimización logística de entregas.
- Gestión del tráfico urbano en ciudades inteligentes.

Características técnicas:

1. Representación del grafo:
 - Nodos: puntos de interés (Ej: "Hospital", "Parque", "Centro Comercial").
 - Aristas: calles con distancias.
 - Pesos: distancia en kilómetros.
 - Representar con diccionario de tuplas.
2. Algoritmo aplicado:
 - Dijkstra para hallar la ruta más corta desde un punto A hasta un punto B.
3. Interacción con el usuario:
 - El usuario ingresa los nodos de inicio (punto A) y destino (punto B).
 - Se muestra el camino óptimo y la distancia total.
4. Representación gráfica:
 - Graficar el grafo con networkx y matplotlib. (librerías) desde el mismo algoritmo.

Ejemplo de salida:

Ingrese ubicación de partida: Centro Comercial

Ingrese destino: Universidad

Ruta más corta: Centro Comercial → Hospital → Parque → Universidad

Distancia total: 4 km

Mapa de ejemplo:

Supongamos una pequeña ciudad con los siguientes **nodos** (lugares) y **aristas** (calles con distancias en kilómetros):

Nodos:

- A: Hospital
- B: Parque
- C: Centro Comercial
- D: Estación de Policía
- E: Universidad
- F: Terminal de Transporte

Aristas con pesos (no dirigidas):

Origen	Destino	Distancia (km)
A	B	2
A	C	4
B	C	1
B	D	2
C	E	3
D	E	1
D	F	2
E	F	5

¿Cómo usar las librerías networkx y matplotlib?

Instalar desde la terminal de Windows PowerShell, CMD o Git Bash las librerías con el siguiente comando:

```
pip install networkx matplotlib
```

```
MINGW64~/c/Users/Lorena
Collecting python-dateutil<=2.7 (from matplotlib)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting six>=1.5 (from python-dateutil<=2.7->matplotlib)
  Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Downloaded networkx-3.4.2-py3-none-any.whl (1.7 MB)
  1.7/1.7 MB 454.0 kB/s eta 0:00:00
Downloaded matplotlib-3.10.1-cp313-cp313-win_amd64.whl (8.1 MB)
  8.1/8.1 MB 473.3 kB/s eta 0:00:00
Downloaded contourpy-1.3.1-cp313-cp313-win_amd64.whl (220 kB)
Downloaded cycycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloaded fonttools-4.57.0-cp313-cp313-win_amd64.whl (2.2 MB)
  2.2/2.2 MB 421.7 kB/s eta 0:00:00
Downloaded kiwisolver-1.4.8-cp313-cp313-win_amd64.whl (71 kB)
Downloaded numpy-2.2.4-cp313-cp313-win_amd64.whl (12.6 MB)
  12.6/12.6 MB 624.7 kB/s eta 0:00:00
Downloaded packaging-24.2-py3-none-any.whl (65 kB)
Downloaded pillow-11.1.0-cp313-cp313-win_amd64.whl (2.6 MB)
  2.6/2.6 MB 895.7 kB/s eta 0:00:00
Downloaded pyparsing-3.2.3-py3-none-any.whl (111 kB)
Downloaded python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloaded six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, pyparsing, pillow, packaging, numpy, networkx, kiwisolver, fonttools, cycycler, python-dateutil, contourpy
Successfully installed contourpy-1.3.1 cycycler-0.12.1 fonttools-4.57.0 kiwisolver-1.4.8 matplotlib-3.10.1 networkx-3.4.2 numpy-2.2.4 packaging-24.2 pillow-11.1.0 python-dateutil-2.9.0 six-1.17.0
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
Lorena@LAPTOP-PELJ8MN7 MINGW64 ~
$
```

El algoritmo “**representación grafica.py**” muestra cómo se representa gráficamente en python.

Se encuentra en la carpeta “**algoritmo de dijkstra**”.

3. Indicadores de desempeño

- Identifica con claridad las necesidades que satisface las estructuras de datos tipo grafo.
- Construye bajo paradigma orientado a objetos estructura de datos tipo grafo.
- Estructura eficientemente algoritmos de recorridos y búsqueda en grafos.
- Genera algoritmos que dan solución a problemáticas, con el uso de grafos.