

Universidad Mariano Gálvez de Guatemala

Boca del Monte

Facultad de Ingeniería en Sistemas de la Información

Programación 3

Sección: A



Christopher Herrera 7690-20-20773

Cinthia Robles 7690-16-13986

Selvin Alvarez 7690-23-18974

Mahuerk Vazquez 7690-17-20456

Manual Técnico

Este documento constituye la documentación técnica del proyecto final de la asignatura Programación 3. El proyecto implementa algoritmos de recorrido de grafos, específicamente búsqueda en profundidad (DFS) y búsqueda en anchura (BFS), utilizando el lenguaje de programación Python. Además, incorpora una integración con una base de datos MySQL para el almacenamiento y gestión de la información procesada por los algoritmos.

Tabla de MySQL

La base de datos grafo_guatemala contiene la tabla grafo_municipios, la cual modela un grafo ponderado que representa la conexión entre municipios del departamento de Guatemala. Esta tabla almacena las aristas del grafo, donde cada fila representa una conexión directa entre dos municipios, junto con la distancia que los separa.

```
CREATE DATABASE grafo_guatemala;

CREATE TABLE grafo_municipios (
  id INT AUTO_INCREMENT PRIMARY KEY,
  origen VARCHAR(255) NOT NULL,
  destino VARCHAR(255) NOT NULL,
  distancia INT NOT NULL
);
```

A continuación, se describen los campos de la tabla:

- **id**: Clave primaria autoincremental que identifica de forma única cada registro o conexión entre municipios.
- **origen** (VARCHAR(255)): Nombre del municipio de origen en la conexión.
- **destino** (VARCHAR(255)): Nombre del municipio de destino en la conexión.
- **distancia** (INT): Representa el peso de la arista, es decir, la distancia entre los municipios de origen y destino. Este valor puede usarse en algoritmos como BFS, DFS o algoritmos de caminos mínimos (como Dijkstra) para determinar rutas más eficientes.

Esta estructura permite representar el grafo de forma flexible y eficiente, facilitando operaciones como búsqueda de rutas, análisis de conectividad y visualización de caminos entre municipios.

Librerías usadas

1. **import sys**

La librería estándar `sys` proporciona acceso a variables y funciones que interactúan directamente con el intérprete de Python.

2. **import matplotlib.pyplot as plt**

`matplotlib.pyplot` es un módulo de la biblioteca `matplotlib`, utilizada para crear gráficos y visualizaciones.

3. **import networkx as nx**

`networkx` es una librería especializada en la creación, manipulación y análisis de grafos y redes complejas.

4. **import csv**

`csv` es una librería estándar de Python que permite leer y escribir archivos en formato CSV (Comma-Separated Values).

5. **import mysql.connector**

`mysql.connector` es un conector oficial de MySQL para Python.

6. **from PyQt5.QtWidgets import ...**

Este módulo pertenece a la biblioteca **PyQt5**, un framework que permite crear interfaces gráficas (GUI) en Python usando componentes visuales.

Los elementos importados son:

- **QApplication**: Inicializa y gestiona el ciclo de vida de la aplicación GUI.
- **QWidget**: Clase base para todos los componentes visuales (ventanas, paneles).
- **QPushButton**: Botón interactivo.
- **QVBoxLayout, QHBoxLayout**: Administradores de diseño para organizar widgets en forma vertical u horizontal.
- **QLabel**: Etiqueta de texto estática.
- **QLineEdit**: Campo de texto editable.
- **QFileDialog**: Diálogo para abrir o guardar archivos.

Estructura del CSV

El archivo CSV utilizado en este proyecto tiene como propósito definir las conexiones entre municipios del departamento de Guatemala para construir el grafo correspondiente. Cada fila representa una arista del grafo, especificando los nodos de origen y destino, así como la distancia entre ellos.

Formato del archivo

El archivo debe tener la siguiente estructura:

 Copiar

```
origen,destino,kms
Ciudad de Guatemala,Mixco,10
Ciudad de Guatemala,Villa Nueva,20
```

Descripción de las columnas

- **origen:** Nombre del municipio de donde parte la conexión (nodo de origen).
- **destino:** Nombre del municipio al que llega la conexión (nodo de destino).
- **kms:** Distancia entre ambos municipios en kilómetros. Representa el peso de la arista en el grafo.

Requisitos importantes

- **Los nombres de los municipios en las columnas origen y destino deben coincidir exactamente** con los utilizados en otras partes del sistema, incluyendo:
 - La base de datos (grafo_municipios)
 - Las visualizaciones del grafo
 - La lógica interna de los algoritmos de recorrido (DFS, BFS)
- La coincidencia incluye **mayúsculas, tildes y espacios**. Por ejemplo, "Villa Nueva" y "villa nueva" serán interpretados como municipios distintos si no se normaliza el texto.

Este requisito garantiza que los algoritmos puedan procesar correctamente los datos y evitar inconsistencias o nodos duplicados involuntarios en el grafo.