

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

MATEMATICA PARA COMPUTACION 2

CATEDRÁTICO: ING. JOSE ALFREDO GONZALEZ DIAZ

TUTOR ACADÉMICO: EDUARDO RENÉ AGUSTIN



OSCAR ALFREDO PÉREZ CIFUENTES 202110342

OSCAR MELENDREZ 202308486

DOUGLAS AJÚ 202300559

JOSÉ ROBERTO DE LEÓN 202308202

GUATEMALA, 25 DE OCTUBRE DEL 2,024

# ÍNDICE

<b>ÍNDICE</b>	<b>1</b>
<b>INTRODUCCIÓN</b>	<b>2</b>
<b>OBJETIVOS</b>	<b>2</b>
1. GENERAL	2
2. ESPECÍFICOS	2
<b>ALCANCES DEL SISTEMA</b>	<b>2</b>
<b>ESPECIFICACIÓN TÉCNICA</b>	<b>3</b>
• REQUISITOS DE HARDWARE	3
• REQUISITOS DE SOFTWARE	3
<b>DESCRIPCIÓN DE LA SOLUCIÓN</b>	<b>4</b>
<b>LÓGICA DEL PROGRAMA</b>	<b>5</b>
❖ Proyecto Grafos	5
➤ Librerías	5
➤ Función Main	5
➤ Métodos y Funciones utilizadas	6

# **INTRODUCCIÓN**

El propósito de este manual es mostrar de manera clara la forma en la que está hecho el programa, así como explicar todos los procesos que realiza junto con sus funciones y métodos creados para la ejecución del mismo, para la correcta comprensión del lector.

## **OBJETIVOS**

### **1. GENERAL**

- 1.1. Mostrar la información más importante sobre el programa al lector

### **2. ESPECÍFICOS**

- 2.1. Objetivo 1: Categorizar las diferentes clases y funciones que contiene el código del programa
- 2.2. Objetivo 2: Determinar el uso que tiene cada parte del código dentro del programa

## **ALCANCES DEL SISTEMA**

Se busca compartir de manera comprensible para el lector la forma en la que está hecho el programa, partiendo de explicar que realiza cada método y función que se encuentre dentro del programa y mostrar la razón por la que se optó por hacerlo de esa manera. También explicando los requisitos que se deben de tener para la correcta ejecución del programa y alguna que otra consideración extra.

## **ESPECIFICACIÓN TÉCNICA**

- **REQUISITOS DE HARDWARE**

- Ram: 8 GB
- Memoria: 250 GB
- Procesador Intel i5 - 10500

- **REQUISITOS DE SOFTWARE**

- Windows 10
- Sistema operativo de 64 bits
- Visual Studio Code
- Python

## DESCRIPCIÓN DE LA SOLUCIÓN

- Al ejecutar el programa se encontrará con una ventana donde le aparecerán una serie de botones y “textbox” para el mejor manejo de la solución, primero deberá ingresar las aristas como se le muestra en el ejemplo puesto en el programa para empezar a crear su grafo, una vez se introduce dicha información se creó un botón para una función que lea los datos puestos en el campo de texto y los almacene en forma de vértices y aristas, luego se pusieron dos “Listbox” para mostrar de manera visual todos los datos que se vayan colocando dentro del programa para finalmente colocar los últimos pero no menos importantes tres botones que servirán para poder mostrar el grafo ingresado, y aplicar las dos búsquedas aprendidas en la clase magistral (Búsqueda en Anchura y Búsqueda en Profundidad).

# LÓGICA DEL PROGRAMA

## ❖ Proyecto Grafos

```
import tkinter as tk
from tkinter import simpledialog, messagebox
import networkx as nx
import matplotlib.pyplot as plt
```

### ➤ Librerías

La librería tkinter, se utilizó para crear la interfaz gráfica de usuario (GUI). Permitió construir ventanas, botones, cuadros de texto y otros elementos interactivos útiles para el programa.

La librería network, Se utilizó para trabajar con grafos y redes, permitió añadir nodos, aristas y realizar análisis de redes para el programa.

La librería matplotlib.pyplot, se utilizó para generar gráficos y poder hacer visualizaciones de datos.

### ➤ Función Main

La función Main se utilizó para que contuviera al menú inicial el cual es el que tiene las opciones para ingresar a todos los métodos que se crearon para este programa, utilizando botones para realizar cada acción que el usuario elija de las opciones del menú.

```
if __name__ == "__main__":
    root = tk.Tk()
    app = GrafoApp(root)
    root.mainloop()
```

## ➤ Métodos y Funciones utilizadas

A continuación se dará una explicación general de lo que hace cada función:

- ```
7 def __init__(self, root):
```

Sirve para crear la ventana donde se maneja el flujo del programa y demás opciones a utilizar.

- ```
16 def inicializar_interfaz(self):
```

Sirve para crear la interfaz gráfica del programa, botones, tablas, labels, textbox.

- ```
49 def agregar_arista(self):
```

Sirve para agregar una arista al grafo, y actualizar los datos que se muestran visualmente con la información que tenga agregada al programa.

- ```
71 def actualizar_tabla(self):
```

Sirve para actualizar la tabla de datos que tenga la información de las aristas y vértices en el programa.

- ```
85 def mostrar_grafo(self):
```

Se encarga de agregar un grafo y mostrarlo visualmente según los datos que contenga el programa.

- ```
91 def aplicar_bfs(self):
```

Se encarga de hacer toda la lógica del método de búsqueda en anchura en el programa.

- ```
101 def aplicar_dfs(self):
```

Se encarga de hacer toda la lógica del método de búsqueda en profundidad en el programa.

- ```
111 def mostrar_grafo_modificado(self, grafo_modificado, titulo):
```

Muestra el nuevo gráfico que se forma luego de aplicar cualquiera de las búsquedas antes mencionadas.