

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
MATEMÁTICA PARA COMPUTACIÓN 2
CATEDRÁTICO: ING. JOSÉ ALFREDO GONZÁLES
TUTOR AUXILIAR: ROBERTO GÓMEZ



Marcos Daniel Bonifasi de León - 202202410

Kevin Daniel Catún Landaverde - 202200378

José Sebastian Pirir Romero - 202300335

SECCIÓN: A

GUATEMALA, 29 DE ABRIL DEL 2,024

ÍNDICE

ÍNDICE	1
OBJETIVOS DEL SISTEMA	2
GENERAL	2
ESPECÍFICOS	2
INTRODUCCIÓN	3
INFORMACIÓN DEL SISTEMA	4
REQUISITOS DEL SISTEMA	6
FLUJO DE LAS FUNCIONALIDADES DEL SISTEMA	7

OBJETIVOS DEL SISTEMA

GENERAL

El objetivo principal del manual de usuario es proporcionar a los usuarios una guía clara y concisa sobre cómo utilizar el visualizador de grafos. Este manual busca facilitar la comprensión de las funcionalidades disponibles en la aplicación y cómo interactuar con ellas de manera efectiva. Además, tiene como objetivo ayudar a los usuarios a aprovechar al máximo las capacidades del visualizador de grafos para crear, modificar y analizar grafos de manera intuitiva y eficiente. En resumen, el manual de usuario busca ser una herramienta práctica y útil que permita a los usuarios utilizar la aplicación de manera autónoma y satisfactoria.

ESPECÍFICOS

- Objetivo 1: El manual proporcionará instrucciones detalladas sobre cómo utilizar los algoritmos de búsqueda en anchura y en profundidad dentro del visualizador de grafos.
- Objetivo 2: El manual incluirá ejemplos prácticos de cómo aplicar los algoritmos de búsqueda en anchura y profundidad en situaciones reales utilizando el visualizador de grafos.

INTRODUCCIÓN

El presente manual de usuario proporciona una guía detallada sobre el uso y funcionamiento del visualizador de grafos desarrollado en Python. Esta herramienta, diseñada para la creación, modificación y análisis de grafos, ofrece una plataforma intuitiva y eficiente para estudiantes, profesionales y entusiastas de los grafos. Con el objetivo de facilitar la comprensión y maximizar la utilidad del programa, este manual abarca todas las funcionalidades disponibles, desde la manipulación básica de nodos y aristas hasta la ejecución de algoritmos de búsqueda en anchura y profundidad. A través de instrucciones paso a paso, ejemplos prácticos y consejos útiles, se pretende proporcionar al usuario una experiencia completa y satisfactoria en el uso del visualizador de grafos.

INFORMACIÓN DEL SISTEMA

El programa funciona como un visualizador interactivo de grafos que permite al usuario realizar diversas operaciones sobre un grafo, como agregar nodos y aristas, ejecutar algoritmos de búsqueda en anchura (BFS) y profundidad (DFS), y visualizar el grafo resultante en una interfaz gráfica de usuario (GUI). Al iniciar el programa, se crea una instancia del objeto GraphVisualizer, que inicializa un nuevo grafo vacío y crea una ventana de interfaz gráfica utilizando la biblioteca Tkinter.

El usuario interactúa con la interfaz de usuario para realizar diversas operaciones. Puede agregar nodos y aristas ingresando los nombres de los nodos y haciendo clic en los botones correspondientes. También puede ejecutar algoritmos de búsqueda en anchura y profundidad haciendo clic en los botones respectivos. Cuando el usuario agrega un nodo, el nombre del nodo ingresado se agrega al grafo y se actualiza la visualización en el lienzo de Matplotlib. Si el usuario agrega una arista entre dos nodos, se agrega una conexión entre los nodos correspondientes en el grafo y se actualiza la visualización del grafo. Al ejecutar un algoritmo de búsqueda en anchura (BFS) o en profundidad (DFS), se utiliza la biblioteca NetworkX para calcular los bordes del grafo que representan el recorrido realizado por el algoritmo, y se actualiza la visualización del grafo para resaltar estos bordes. El grafo se visualiza utilizando la biblioteca Matplotlib, que proporciona una representación gráfica del grafo en la ventana de la interfaz de usuario. Los nodos se representan como círculos y las aristas como líneas entre los nodos correspondientes. La visualización se actualiza dinámicamente según las operaciones realizadas por el usuario. El programa continúa ejecutándose hasta que el usuario cierra la ventana de la interfaz de usuario. Una vez cerrada, el programa finaliza su ejecución.

En resumen, el programa permite al usuario interactuar con un grafo de manera intuitiva y visual, brindando herramientas para la manipulación y análisis de grafos de una manera eficiente y accesible.

REQUISITOS DEL SISTEMA

Para poder ejecutar la aplicación del visualizador de grafos desarrollado en Python, el usuario necesita cumplir con los siguientes requisitos del sistema:

1. Python: El usuario debe tener instalado Python en su computadora. La aplicación está desarrollada en Python y requiere que el intérprete de Python esté disponible para ejecutar el código.
2. Bibliotecas Python: Es necesario tener instaladas las bibliotecas de Python utilizadas en el programa, que incluyen NetworkX, Tkinter y Matplotlib. Estas bibliotecas pueden instalarse utilizando un gestor de paquetes de Python como pip.
3. Sistema operativo: La aplicación debería ser compatible con los sistemas operativos principales como Windows, macOS y Linux. Sin embargo, se recomienda verificar la compatibilidad específica con el sistema operativo del usuario antes de la instalación.
4. Recursos de Hardware: Aunque la aplicación no requiere recursos de hardware muy exigentes, se recomienda tener un equipo con al menos 2 GB de RAM y un procesador de al menos 1 GHz para un rendimiento óptimo.

FLUJO DE LAS FUNCIONALIDADES DEL SISTEMA

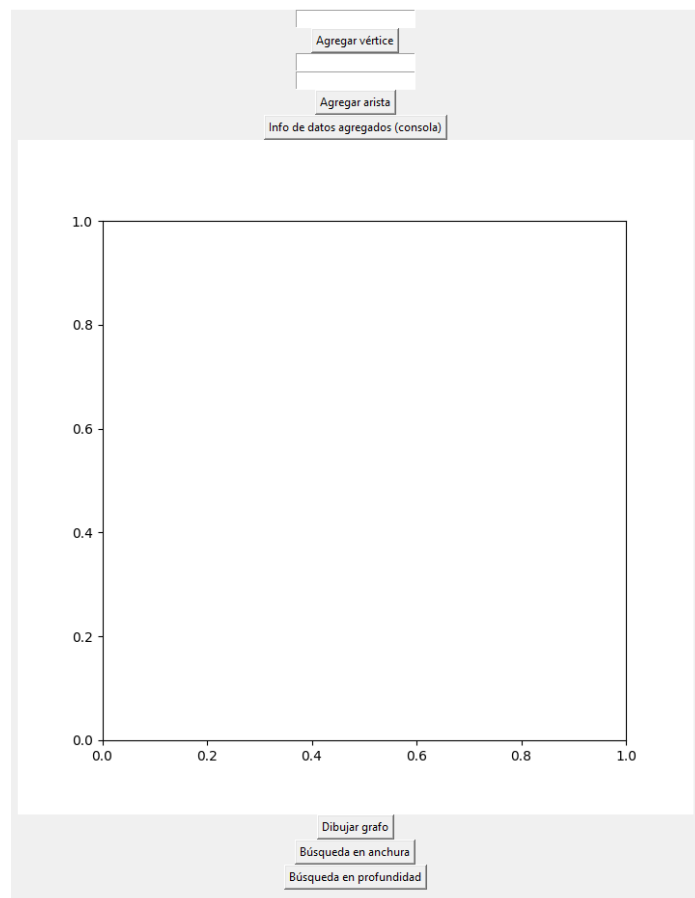
El flujo de funcionalidad del sistema, considerando que el usuario tiene un archivo ejecutable de la aplicación, se puede describir en los siguientes pasos:

1. Inicio de la Aplicación

- El usuario inicia la aplicación haciendo doble clic en el archivo ejecutable proporcionado.

2. Interfaz de Usuario

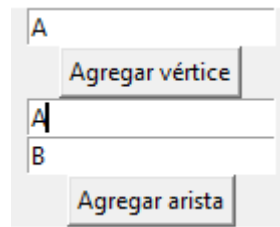
- Después de iniciar la aplicación, se abre una ventana de interfaz de usuario (GUI) que muestra las opciones y controles disponibles para interactuar con el programa.



3. Agregar Nodos y Aristas

- Para agregar nodos al grafo, el usuario puede ingresar el nombre del nodo en el campo de entrada designado y hacer clic en el botón "Agregar vértice".

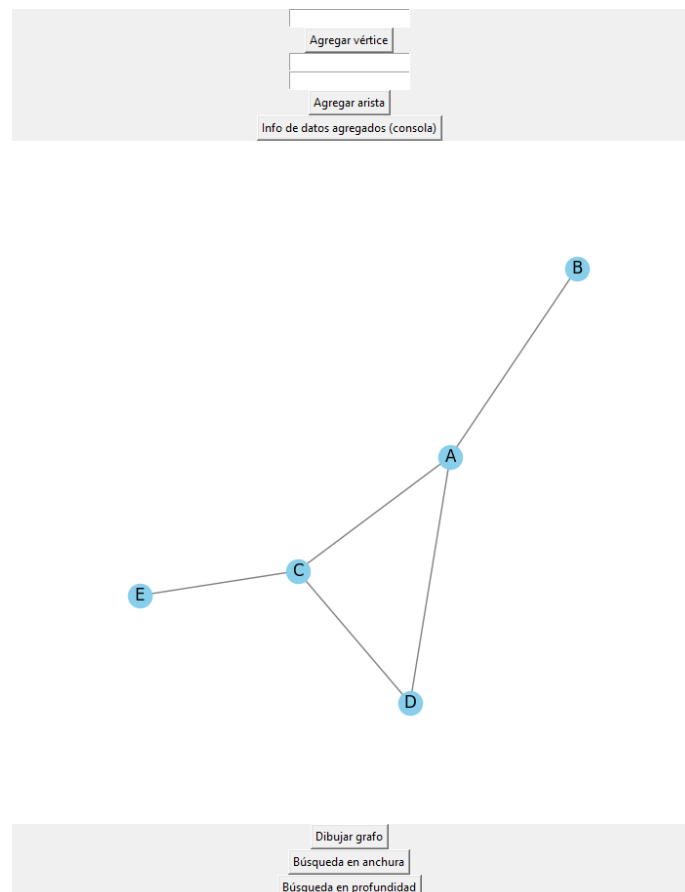
- Para agregar aristas, el usuario debe ingresar los nombres de los nodos de origen y destino en los campos de entrada correspondientes y hacer clic en el botón "Agregar arista".



Formulario de interfaz de usuario para agregar nodos y aristas. Incluye un campo de entrada con el texto 'A', un botón 'Agregar vértice', otro campo de entrada con el texto 'A', un tercer campo de entrada con el texto 'B', y un botón 'Agregar arista'.

4. Visualización del Grafo

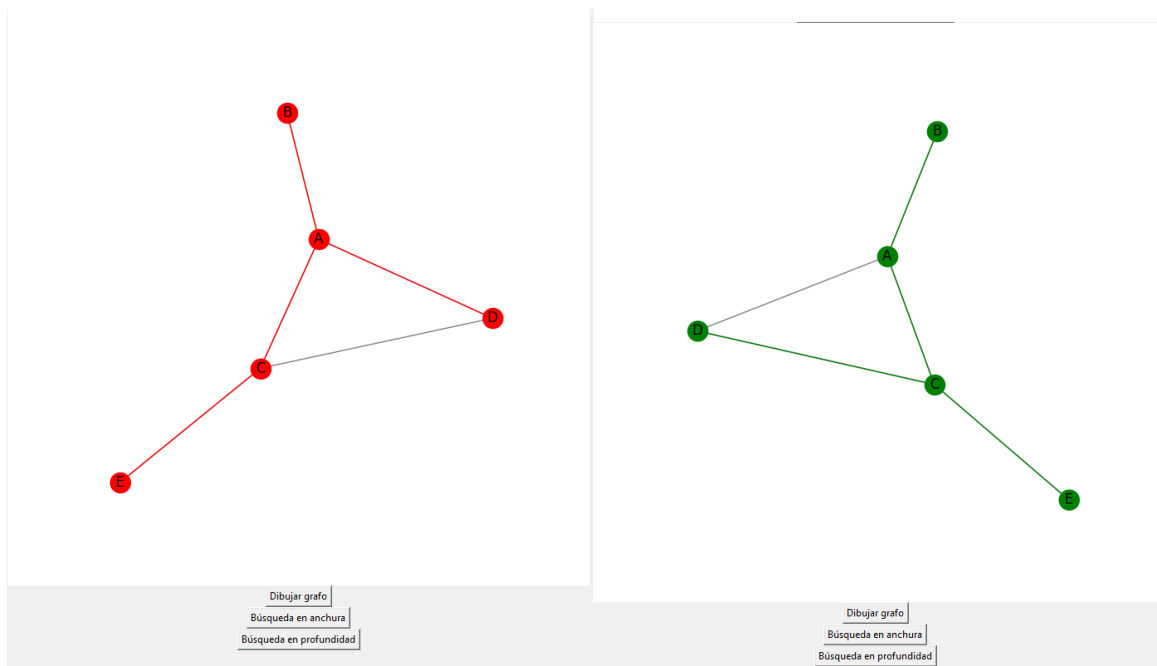
- Después de agregar nodos y aristas, el grafo se visualiza automáticamente en el lienzo de Matplotlib en la interfaz de usuario. Los nodos se representan como círculos y las aristas como líneas entre los nodos correspondientes.



5. Ejecutar búsqueda por anchura y profundidad

- El usuario puede ejecutar los algoritmos de búsqueda en anchura y profundidad haciendo clic en los botones "Búsqueda en anchura" y "Búsqueda en profundidad", respectivamente.

- Al hacer clic en estos botones, se ejecutan los algoritmos correspondientes en el grafo y se resaltan visualmente los bordes que representan el recorrido realizado por el algoritmo.



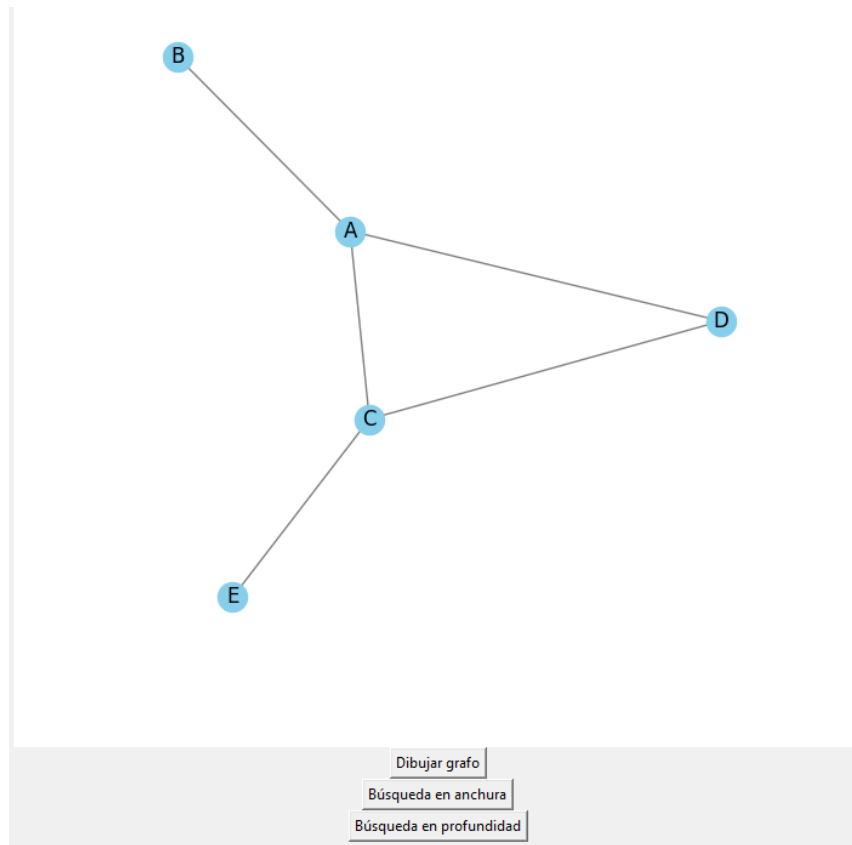
6. Información de Datos

- El usuario puede obtener información sobre los datos agregados al grafo haciendo clic en el botón "Info de datos agregados (consola)". Esto imprimirá en la consola la cantidad de nodos y aristas presentes en el grafo.

```
Número de nodos: 5
Número de bordes: 5
```

7. Dibujar Grafo

- El usuario puede actualizar la visualización del grafo en cualquier momento haciendo clic en el botón "Dibujar grafo". Esto volverá a dibujar el grafo en el lienzo de Matplotlib según los nodos y aristas actuales en el grafo.



8. Cierre de la Aplicación

- Para cerrar la aplicación, el usuario puede hacer clic en el botón para cerrar la ventana de la interfaz de usuario o cerrar directamente la ventana de la aplicación. Esto finalizará la ejecución del programa.

