

Estudio de Algoritmos sobre Grafos

Versión: 1.0

Autores:

**Jhair Alexander Peña Aguirre
Bryan Mauricio Gonzalez Giraldo**

**18 de Junio 2024
Universidad de Caldas - Manizales**

2. Índice

1. Introducción
2. Descripción General del Sistema
3. Arquitectura del Sistema
4. Diseño del Sistema
5. Interfaz de Usuario
6. Configuración e Instalación
7. Mantenimiento y Actualizaciones
8. Pruebas y Validación

3. Introducción

Propósito del Documento: Este manual técnico proporciona una descripción detallada del proyecto "Estudio de Algoritmos sobre Grafos", incluyendo su arquitectura, diseño, configuración, y procedimientos de mantenimiento.

Alcance del Proyecto: El proyecto se centra en el estudio de algoritmos de grafos con un enfoque en problemas de optimización, y proporciona una interfaz gráfica de usuario (GUI) para facilitar la visualización de los resultados.

Audiencia Objetivo: Desarrolladores de software, investigadores en teoría de grafos y personal técnico.

4. Descripción General del Sistema

Resumen del Sistema: Este sistema es una aplicación web que permite la creación, edición y análisis de grafos mediante una interfaz gráfica de usuario. Se enfoca en problemas de optimización y la representación de procesos en grafos.

Funcionalidades Principales:

- Creación y edición de grafos (personalizados o automáticos).
- Visualización y manipulación de nodos y aristas (agregar, editar, eliminar).
- Ejecución de algoritmos de grafos, como la determinación de grafos bipartitos y componentes conexas.
- Exportación e importación de grafos en diferentes formatos.
- **Requisitos del Sistema:**

Hardware: Se recomienda computador, ya que permite visualizar de mejor manera el aplicativo.

Software: Sistema Operativo Windows, macOS o Linux, Python 3.10.x.

5. Arquitectura del Sistema

Diagrama de Arquitectura:

Componentes Principales:

- **Frontend y Backend:** Monolito desarrollado con Streamlit.

- **Librerías y Herramientas:**

- **Streamlit:** Para el desarrollo de la GUI.
- **NetworkX:** Para el manejo y funcionalidades con grafos.
- **Pandas:** Para la creación de tablas y exportación de archivos Excel.
- **NumPy:** Para cálculos de matrices.
- **SciPy:** Para cálculos de distribución de probabilidades.
- **PyAutoGUI:** Para tomar y exportar capturas de grafos.
- **Streamlit-agraph:** Para la visualización interactiva de grafos.

6. Diseño del Sistema

Diagrama de Clases:

Diagrama de Secuencia:

7. Interfaz de Usuario

Descripción de la Interfaz: La interfaz permite al usuario crear y editar grafos de forma manual o automática. Los usuarios pueden:

- Agregar o eliminar nodos y aristas.
- Definir características del grafo como número de nodos, si es dirigido, ponderado, completo o conexo.
- Utilizar menús contextuales para modificar nodos y aristas.
- Mover nodos sobre el área de trabajo.
- Cargar un grafo desde un archivo de texto plano.
- Exportar grafo en imagen, json, xlsx.

Menú Principal:

- **Archivo:** Permite guardar, importar y exportar grafos.
- **Editar:** Permite agregar, eliminar, y renombrar nodos y aristas.
- **Ejecutar:** Permite ejecutar algoritmos de grafos.
- **Herramientas:** Permite
- **Ayuda:** Permite obtener una guía sobre lo que hace el aplicativo y el manual de usuario.
- **Ventana:** Permite ver el grafo en formato tablas.

Mockups:

8. Configuración e Instalación

Requisitos Previos:

- Python 3.10.x instalado.
- Instalación de las librerías necesarias mediante pip:
Copiar código
`pip install -r requirements.txt`

Instrucciones de Instalación:

- Clonar el repositorio del proyecto:
bash
Copiar código
`git clone https://github.com/AguirreCode/project-analysis.git`
- Navegar al directorio del proyecto:
bash
Copiar código
`cd project-analysis`
- Instalar las dependencias del proyecto:
bash
Copiar código
`pip install -r requirements.txt`
- Ejecutar la aplicación:
bash
Copiar código
`streamlit run index.py`

9. Pruebas y Validación

Estrategia de Pruebas:

- Pruebas de extremo a extremo, replican el comportamiento de un usuario con el software en un entorno de aplicación completo. Además, verifican que diversos flujos de usuario funcionen según lo previsto.

Se realiza el seguimiento de diversos flujos de usuario, desde crear un grafo manual y aleatorio decidiendo los parámetros y características que tendrá.

Se realiza el seguimiento de carga de grafo desde un archivo de texto plano hasta poder visualizarlo en la interfaz de usuario.

Se realiza la adición, edición y eliminación tanto para nodos como para aristas y la opción de deshacer último cambio.

Casos de Prueba:

- **Caso de Prueba 1:** Verificar la creación manual de un grafo.
- **Caso de Prueba 2:** Verificar la creación automática de un grafo.
- **Caso de Prueba 3:** Determinar si un grafo es bipartito.
- **Caso de Prueba 4:** Determinar las componentes conexas de un grafo.

Resultados de Pruebas:

- **Caso de prueba 1**

Choose

Personalizado ▾

Grafo ▴

Nombre del grafo

mi_grafo

☒ ¿Es dirigido?

☒ ¿Es ponderado?

Crear grafo

Nodo ▴

ID del nodo

2

Nombre del nodo

Nodo 2

Color del nodo

Forma del nodo

diamond ▾

Agregar nodo

Arista ▴

Nodo de inicio de la arista

1 ▾

Nodo final de la arista

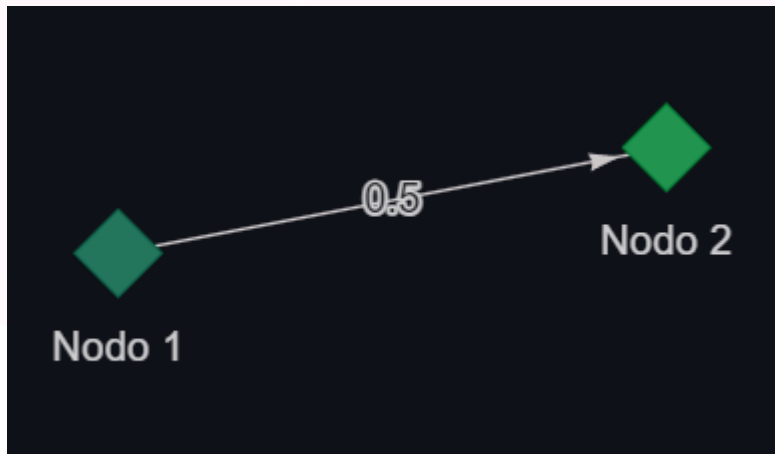
2 ▾

Etiqueta de la arista

0.5

Color de la arista

Agregar arista



- **Caso de prueba 2**

Choose

Aleatorio ▾

Grafo ^

Nombre del Grafo

mi_grafo

☒ ¿Es dirigido?

☒ ¿Es ponderado?

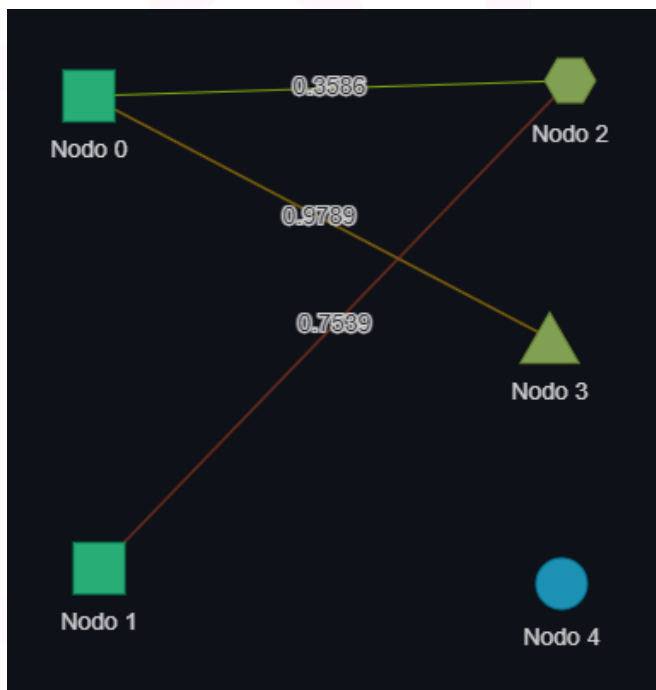
☐ ¿Es completo?

☐ ¿Es conexo?

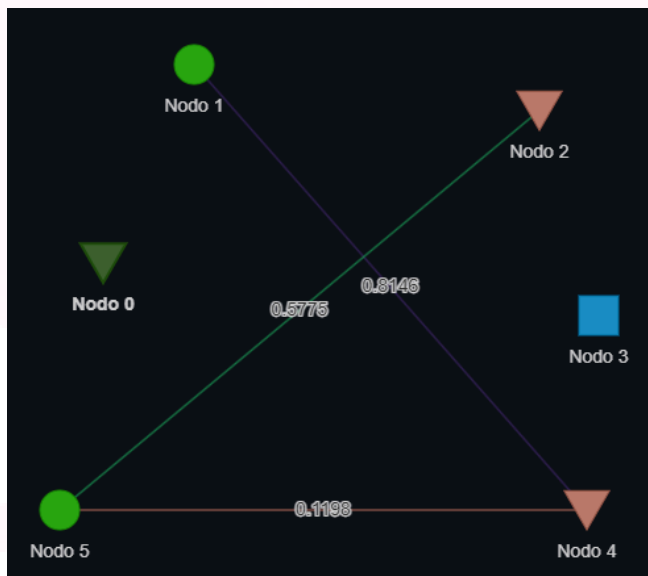
Cantidad de nodos

5

Generar



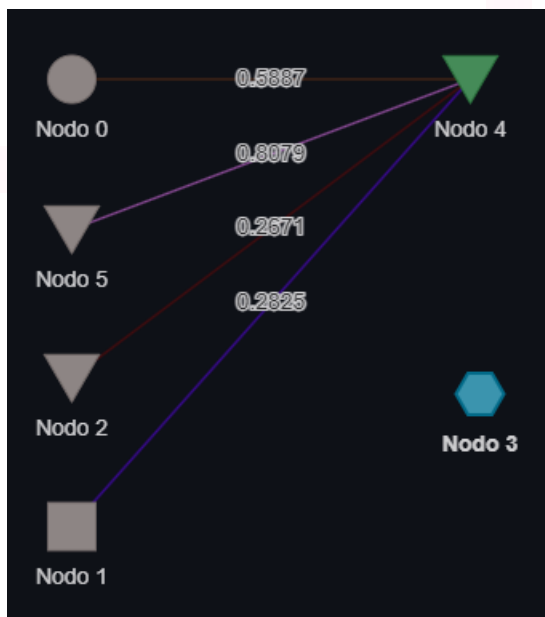
- Caso de prueba 3



Análisis de Grafo Bipartito

El grafo es bipartito y desconexo

- Caso de prueba 4



Componentes conectados:

Componente 1: ([0, 5, 2, 1], [4])

Componente 2: ([3], [])