

Requerimientos: Análisis y Representación Computacional de Grafos mediante Python

J. C. Barrera Guevara , D. A. Machado Tovar, J. G. Delgado
Facultad de Ciencias Básicas e Ingeniería, Universidad de los Llanos
Villavicencio, Colombia
jc.bguevara@unillanos.edu.co
damachado@unillanos.edu.co
jg.delgado@unillanos.edu.co

1. Introducción

1.1 Propósito

El propósito de este documento es especificar los requerimientos funcionales y no funcionales del software desarrollado para la Práctica No. 3: Teoría de Grafos. El sistema permite construir, validar y analizar grafos dirigidos y no dirigidos mediante una interfaz gráfica implementada en Python, integrando el algoritmo de Dijkstra para determinar rutas de costo mínimo.

1.2 Alcance

El software está diseñado como herramienta académica que permite ingresar una matriz de adyacencia, validar su consistencia, representar el grafo gráficamente y calcular caminos mínimos. No está orientado a entornos industriales, sino educativos.

1.3 Definiciones, acrónimos y abreviaturas

Grafo: conjunto de vértices y aristas.

Matriz de adyacencia: representación matricial de las conexiones.

Dijkstra: algoritmo de caminos mínimos.

1.4 Visión general del documento

Este documento describe los requerimientos del sistema, su funcionalidad, restricciones y características no funcionales relevantes.

2. Descripción general

2.1 Perspectiva del producto

Aplicación de escritorio en Python con Tkinter, modular, sin dependencias externas.

2.2 Funciones del sistema

Permite crear grafos, ingresar matrices, validar estructuras, visualizar aristas, calcular rutas mínimas y mostrar resultados.

2.3 Características de los usuarios

Dirigido a estudiantes y docentes con conocimientos básicos de grafos.

2.4 Restricciones

Requiere Python y Tkinter.

2.5 Suposiciones y dependencias

El usuario entiende la matriz de adyacencia. Requiere Python 3 y Tkinter.

3. Requerimientos específicos

3.1 Requerimientos funcionales

RF-01 Creación del grafo: definir número de vértices.

RF-02 Ingreso de matriz de adyacencia.

RF-03 Selección de tipo de grafo.

RF-04 Validación estructural.

RF-05 Visualización gráfica.

RF-06 Selección de nodos origen/destino.

RF-07 Ejecución del algoritmo de Dijkstra.

RF-08 Resaltado visual del camino mínimo.

RF-09 Manejo de errores.

3.2 Requerimientos no funcionales

RNF-01 Usabilidad: interfaz intuitiva.

RNF-02 Fiabilidad: manejo estable de datos.

RNF-03 Mantenibilidad: código modular.

RNF-04 Interfaz: diseño claro y visual.

3.3 Requerimientos de interfaz

Entrada de pesos y tipo de grafo, salida gráfica del grafo y mensajes de resultado.

3.4 Requerimientos de validación

Sin aristas duplicadas, detección de grafos no conexos.

4. Aprobación

Documento elaborado conforme al formato IEEE 830 basado en la Guía de Laboratorio No. 3 del curso de Optimización, como referencia para implementación y validación del software académico.