Manual de uso

Legendary guacamole

David Mendoza & Josué Rodríguez
Estructura de Datos I

Contenido

Introducción

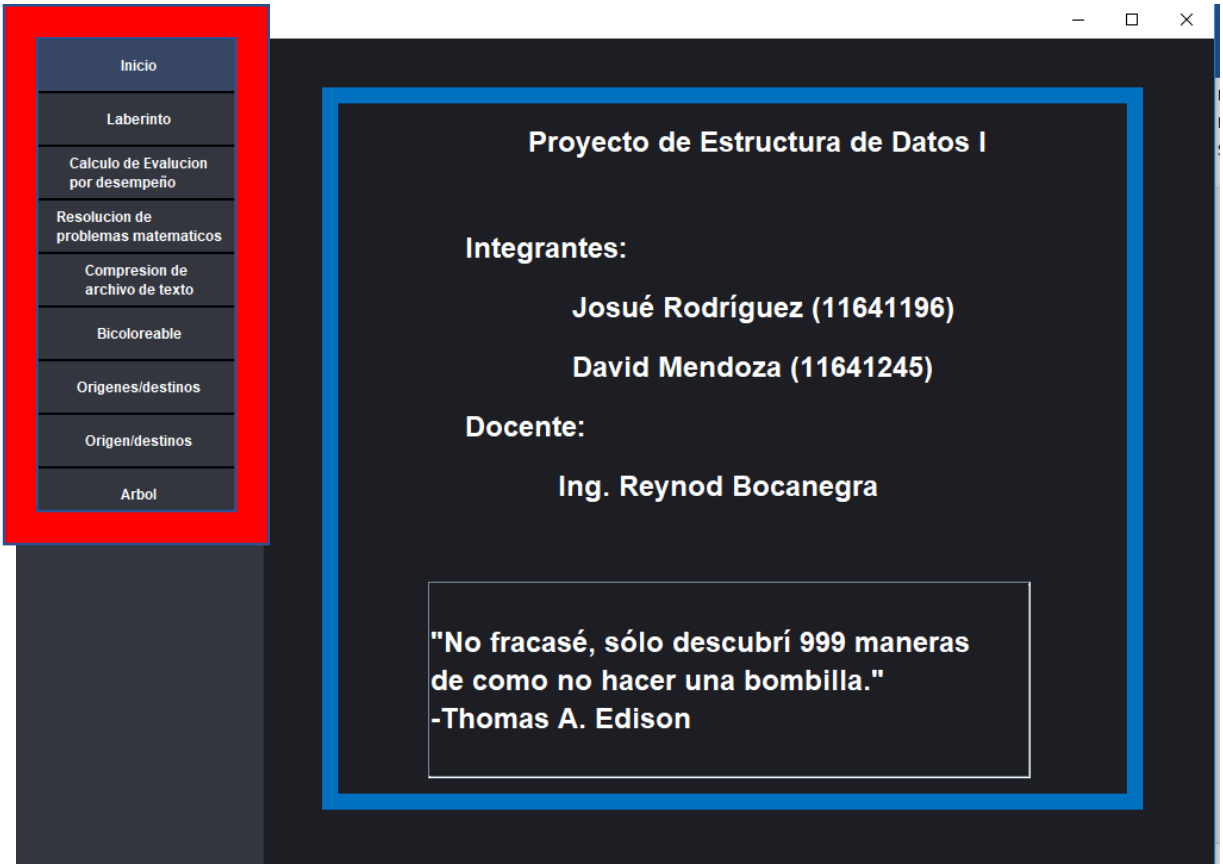
Este manual tiene como finalidad dar a conocer las formas de funcionamiento del programa “Legendary Guacamole”, proyecto de la clase “Estructura de Datos I” del primer semestre del año 2018.

Requerimientos previos

- Tener instalado java en su ordenador.
- Los grafos se leerán con la matriz de adyacencia, es decir, en el archivo de texto deberá tener escrita la matriz de adyacencia separada por comas (Véase Anexo).

Conociendo la interfaz

La pantalla principal, o menú principal, consta de dos elementos: una barra lateral que contiene los botones para cada uno de los problemas a resolver (rojo), y el panel del problema en sí (azul).

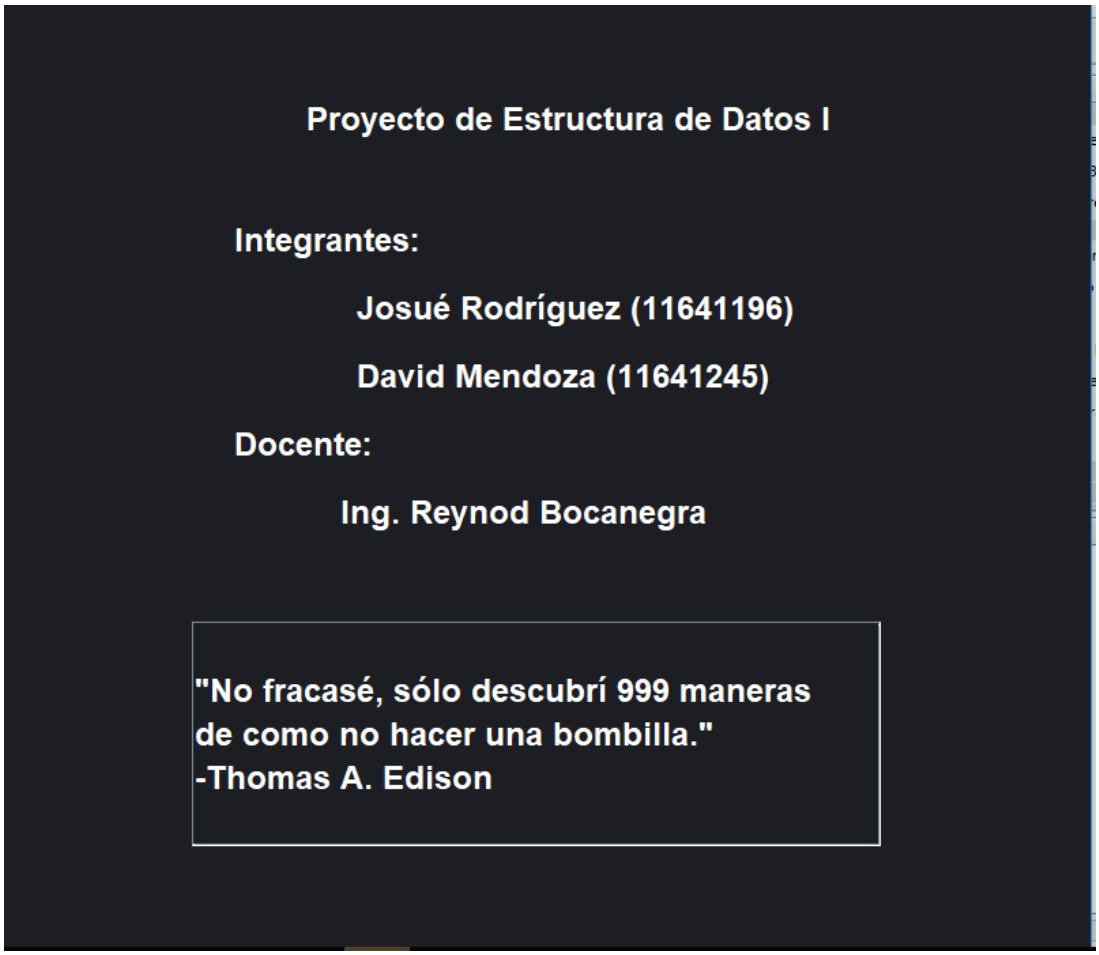


La barra lateral consta de 9 botones:

I. Inicio
II. Laberinto
III. Cálculo de evaluación por desempeño
IV. Resolución de problemas matemáticos
V. Compresión de archivos de texto
VI. Bicoloreable
VII. Orígenes/Destinos
VIII. Origen/Destinos
IX. Árbol de Expansión Mínima

I. Inicio

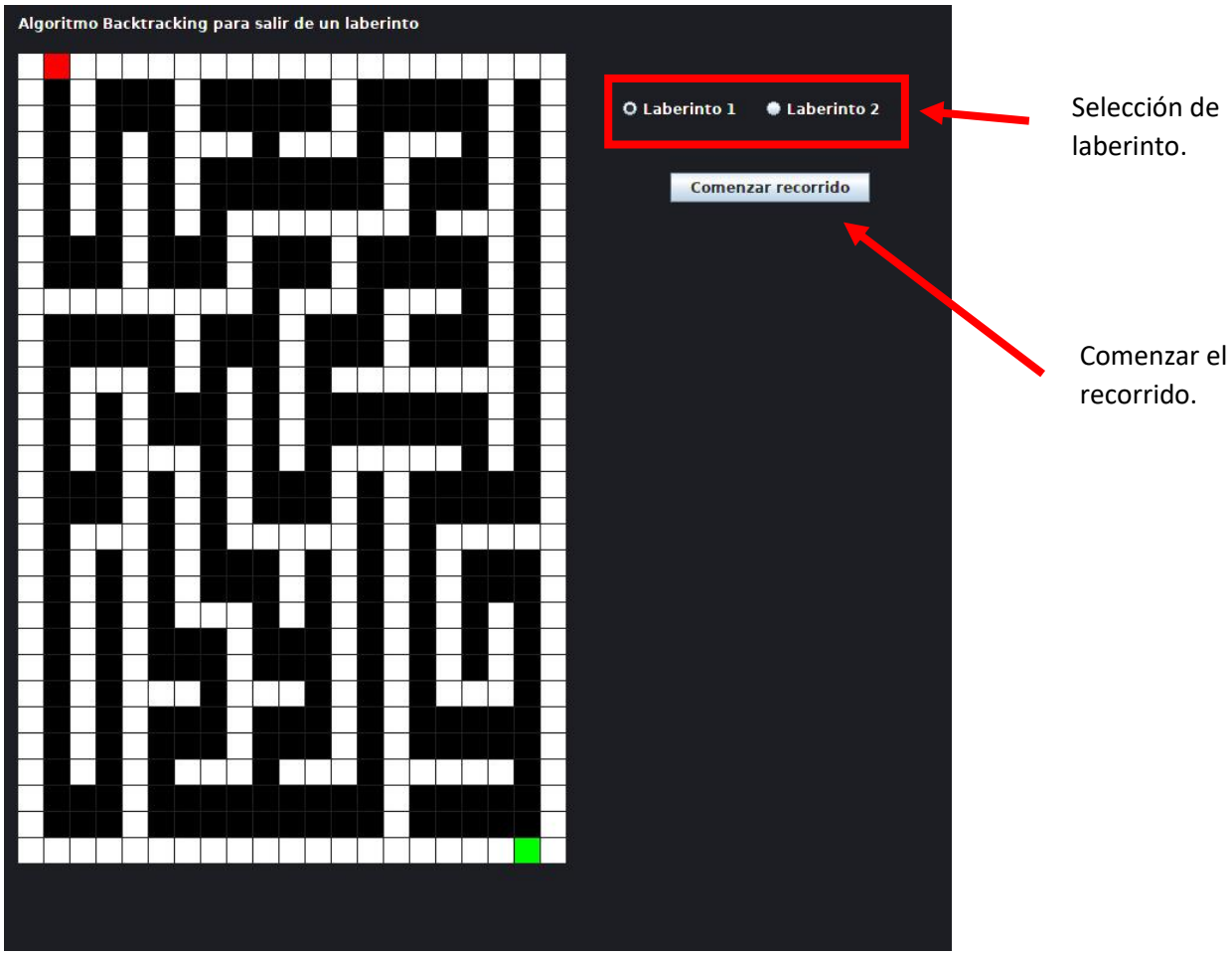
Primer panel que visualiza el usuario al momento de abrir la aplicación. En él se muestra el nombre de la clase, los desarrolladores, el docente que impartió la clase y una cita por Thomas A. Édison.



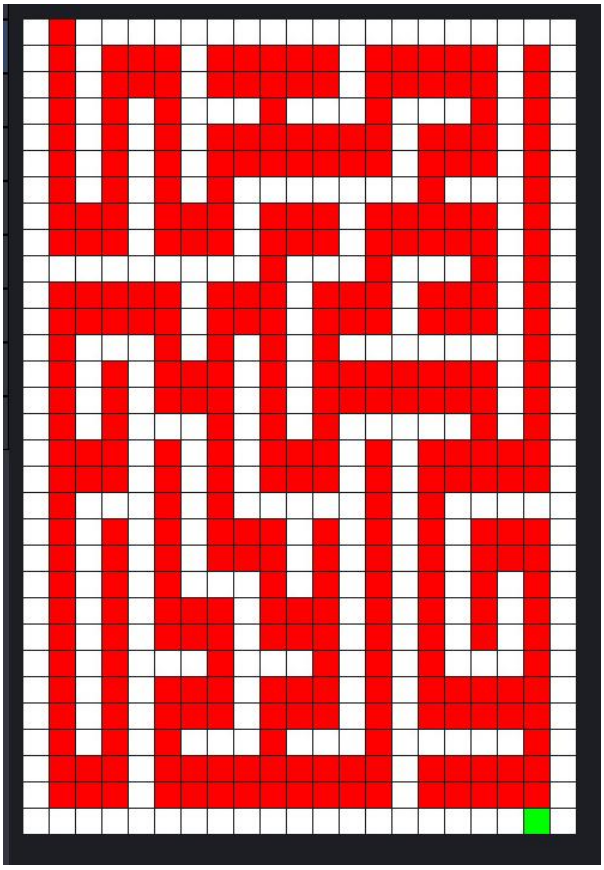
II. Laberinto

Objetivo: empleando el TDA adecuado, encontrar la salida de un laberinto haciendo uso de “backtracking”.

El TDA que decidimos emplear fue Pilas. Lo primero que verá el usuario será el laberinto por recorrer. La aplicación trae por defecto 2 laberintos. Estos se pueden elegir haciendo marcando la burbuja “Laberinto 1” o “Laberinto 2”. Luego para recorrerlo presionar “Comenzar recorrido”.



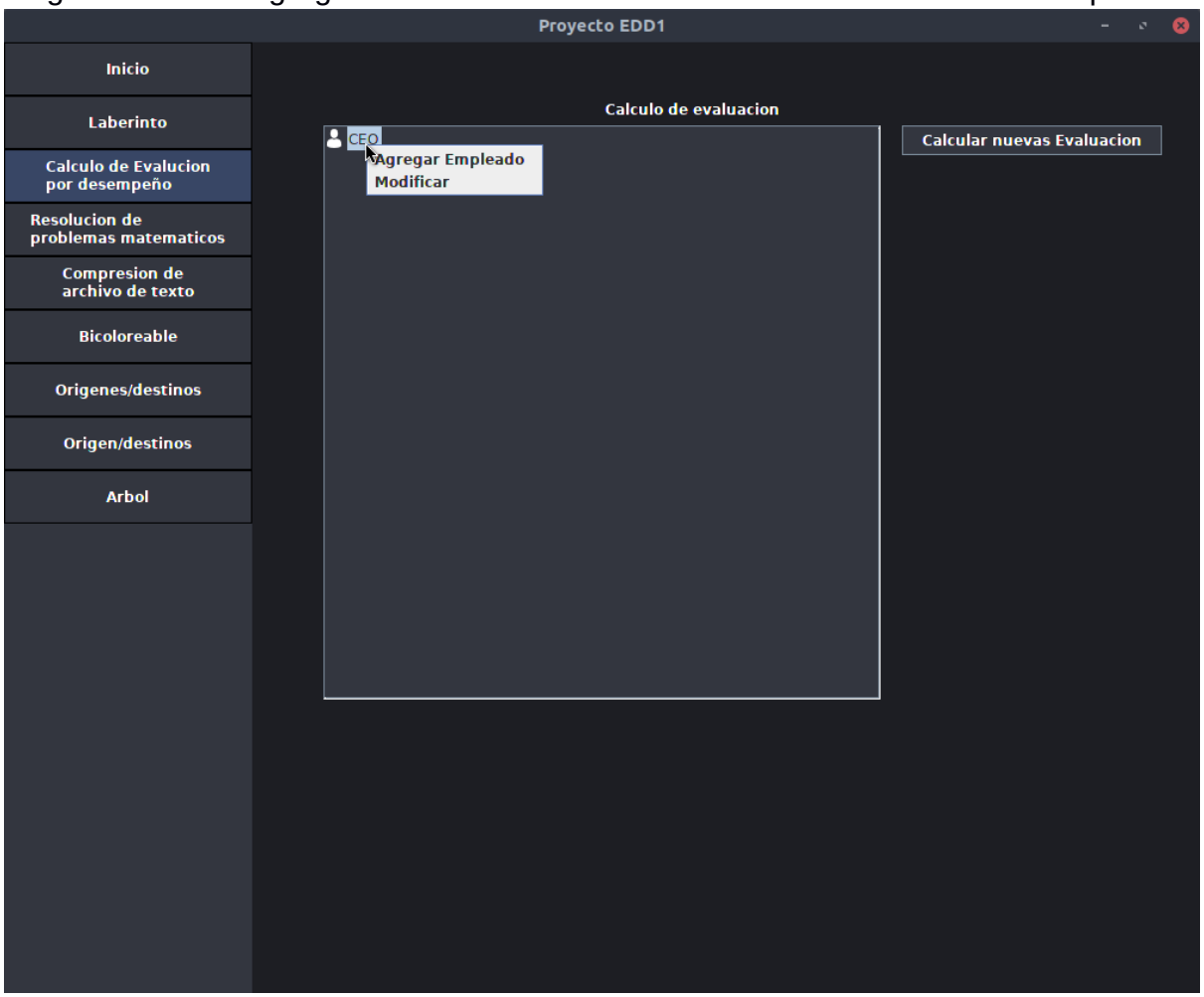
Una vez se presione el botón, comenzará el recorrido y en el laberinto se marcarán las posiciones que ya fueron visitadas, como se muestra en la siguiente imagen:



III. Cálculo de evaluación por desempeño

Objetivo: Hacer uso de la recursividad y del TDA árbol así crea la evaluación de desempeño de una empresa.

En la siguiente imagen, se muestra el TDA árbol general para poder hacer que cada nodo tenga sus respectivos empleados bajo el mismo. Para comenzar se debe agregar todos los empleados necesarios, comenzando por el Nodo padre, o CEO, luego agregar todos sus subempleados

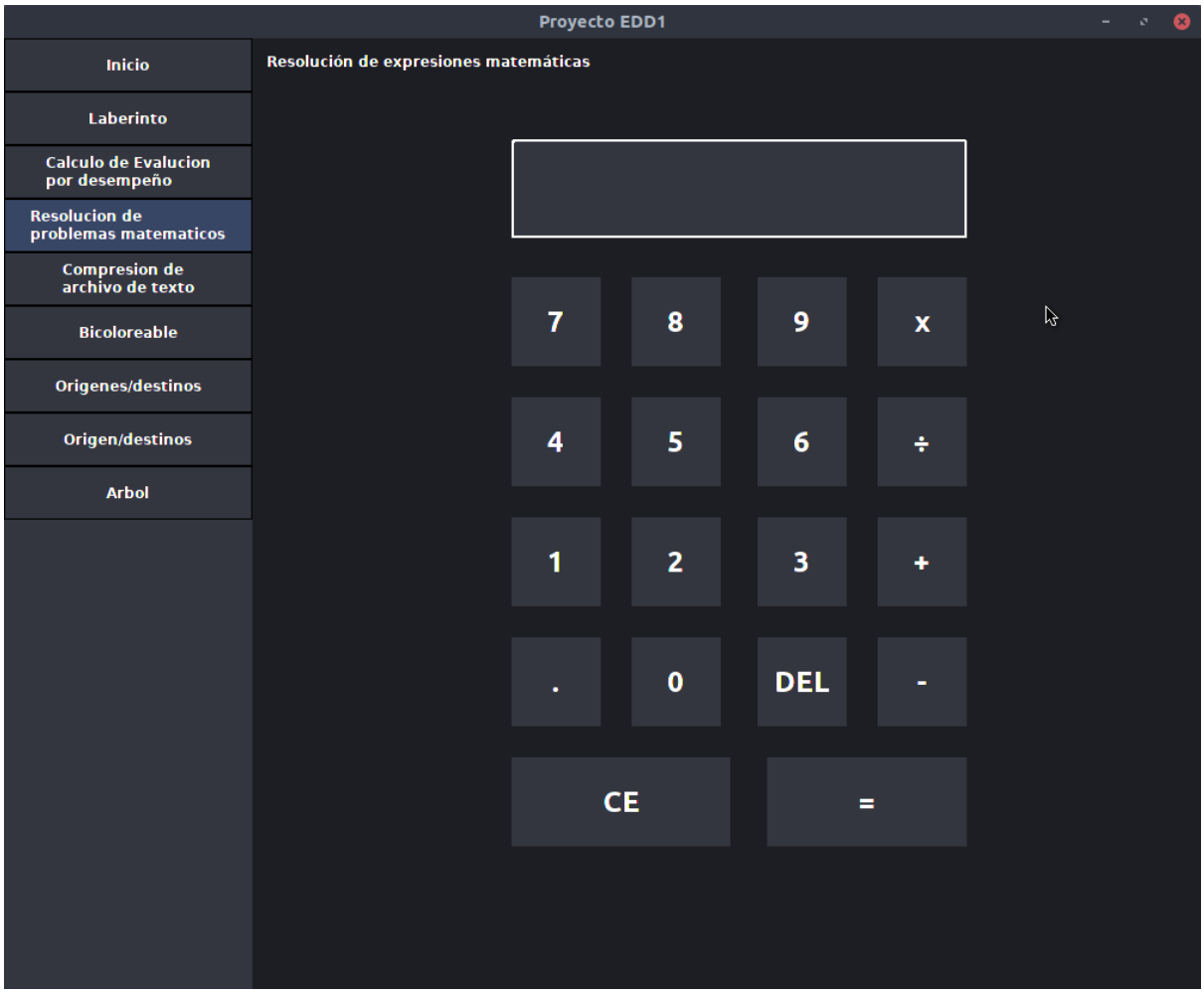


Luego de agregar a todos los empleados, se debe usar el botón “calcular nuevas Evaluaciones”.

IV. Resolución de problemas matemáticos

Objetivo: Hacer uso del TDA necesario (en este caso nosotros usamos listas) para solucionar un problema matemático básico.

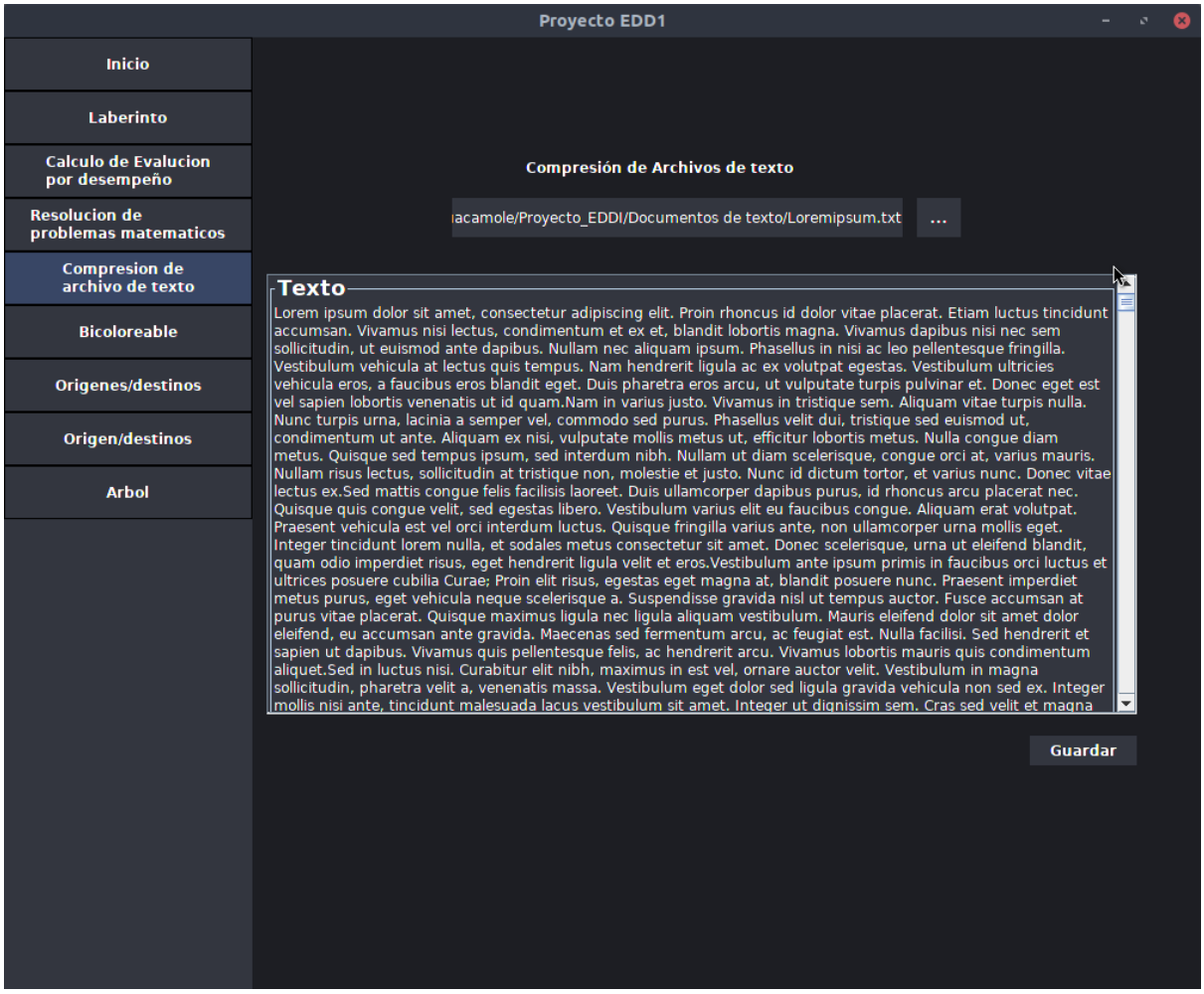
En la siguiente imagen, se verá nuestra interfaz principal, donde podrá usar su puntero para ingresar números y/o símbolos matemáticos.



Luego de ingresar su expresión es tan sencillo de usar el botón con el texto “=” y le dará su respuesta.

IV. Compresión de archivos de texto

Objetivo: Usar el TDA árbol binario, para la compresión de archivos de texto. En la siguiente imagen, se ve el uso en un archivo que consiste en 100 kb de peso

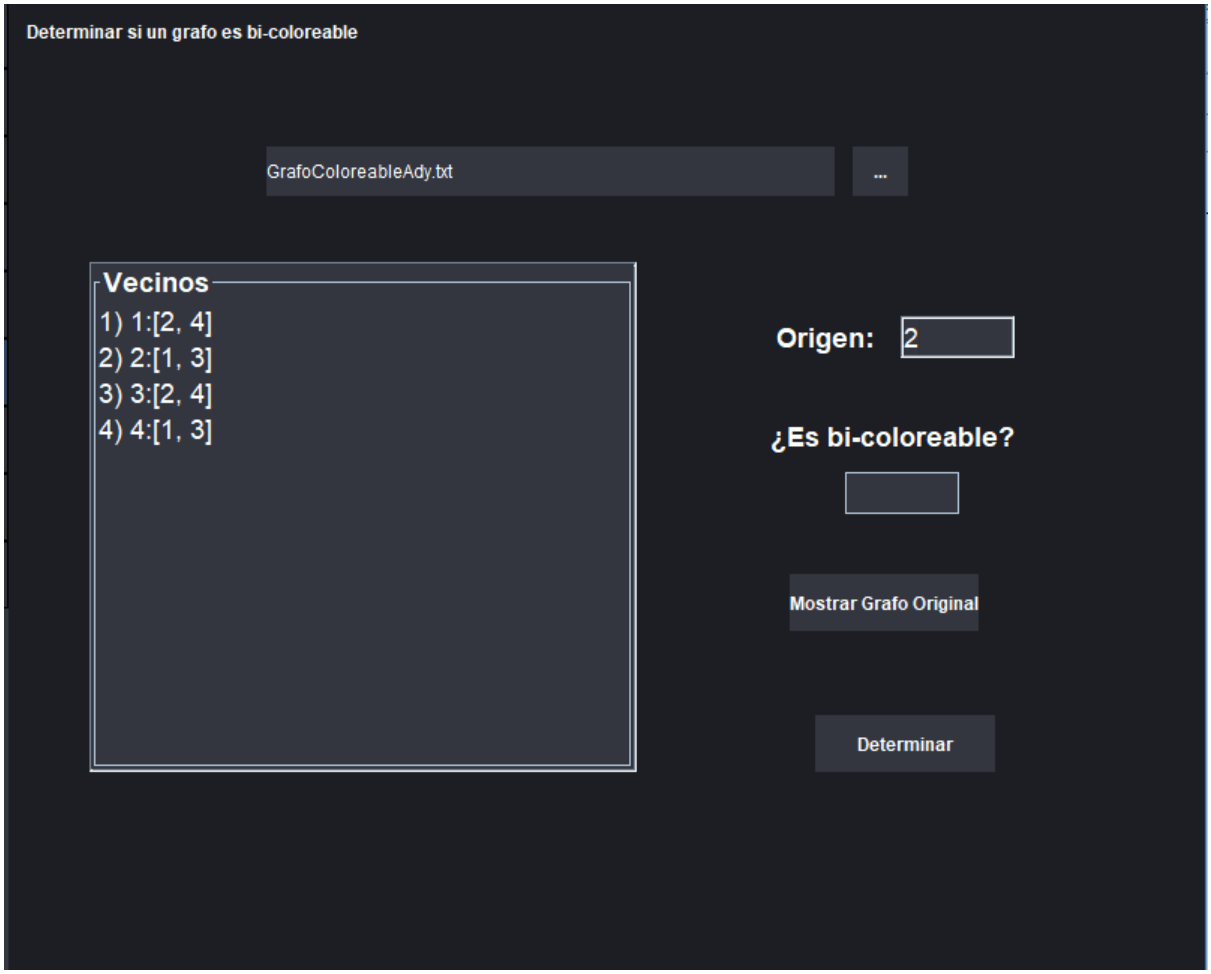


Luego de que su archivo sea cargado y procesado, podrá usar el botón Guardar, debido al tamaño del archivo de texto tomo aproximadamente 34 segundos.

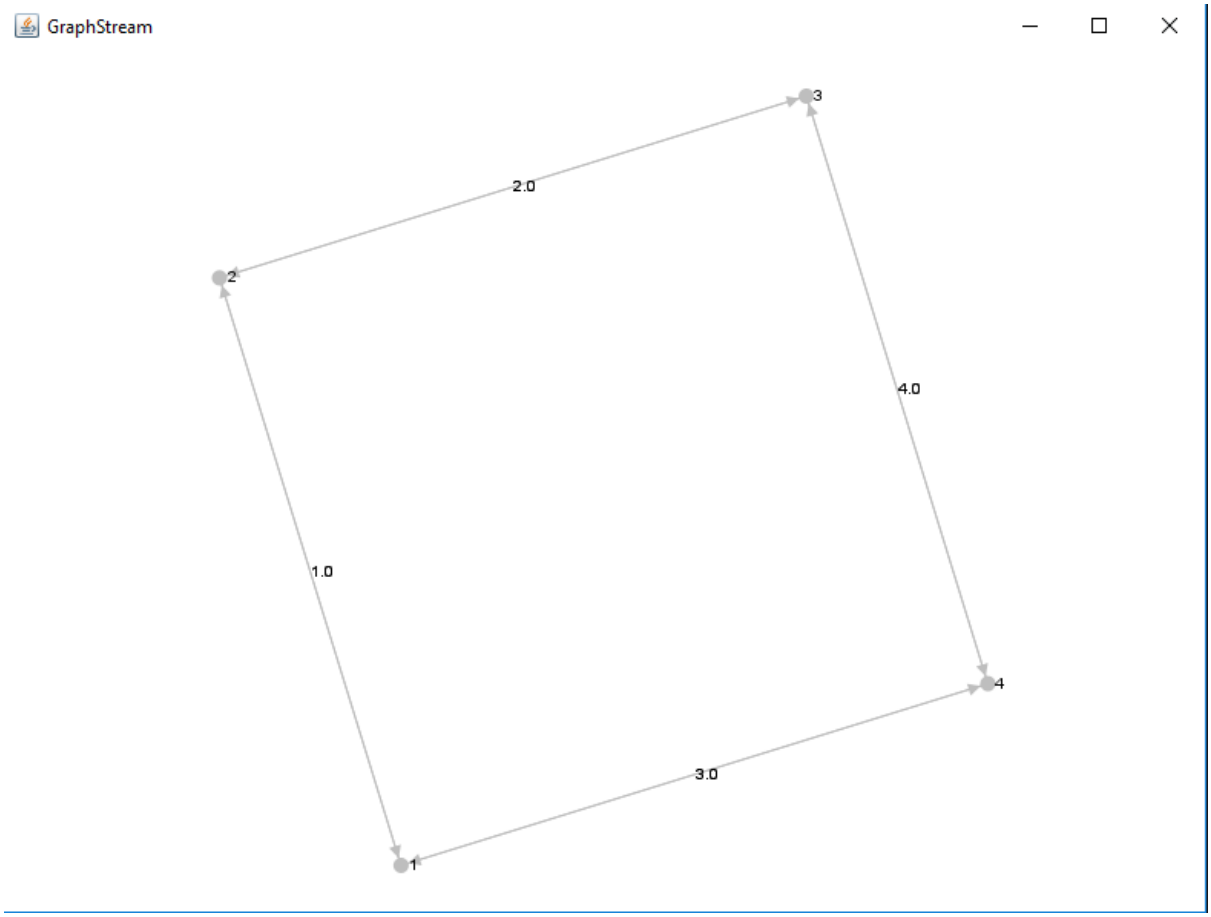
IV. Bicoloreable

Objetivo: Utilizar el recorrido en profundidad para determinar si un grafo es bicoloreable o no.

Empezamos cargando un archivo de texto (.txt), presionando el botón "…", que contenga la matriz de adyacencia del grafo no dirigido que queremos determinar. Una vez cargado el programa, nos mostrará los vecinos de cada nodo. En la casilla "Origen" colocamos el nodo en cual queremos comenzar el recorrido. De dejarlo en blanco, se tomara de inicio el primer nodo. Para ver el grafo original, presione "Mostrar Grafo Original". Para determinar la bicoloreabilidad del grafo, presione "Determinar".

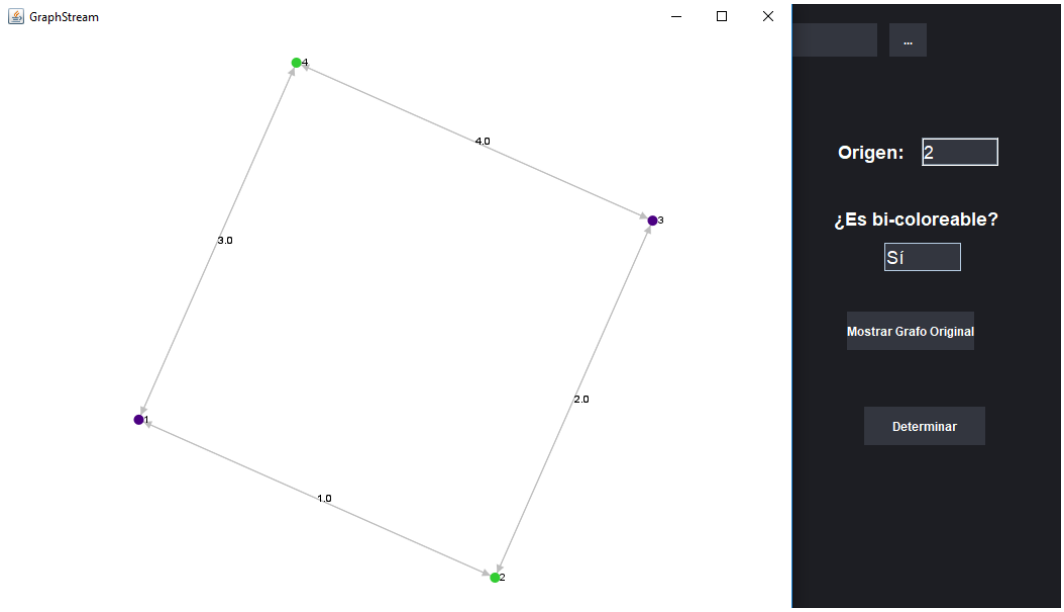


De igual manera, nos mostrara una ventana nueva con el grafo que cargamos.



Nota: así funcionan todos los ejercicios que involucran grafos, se tomara este inciso como referencia.

Una vez presione determinar, se le mostrará un mensaje diciendo si es bicoloreable o no, y posteriormente el grafo coloreado. En la casilla "¿Es bi-coloreable?" Se pondrá "Sí" o "No".



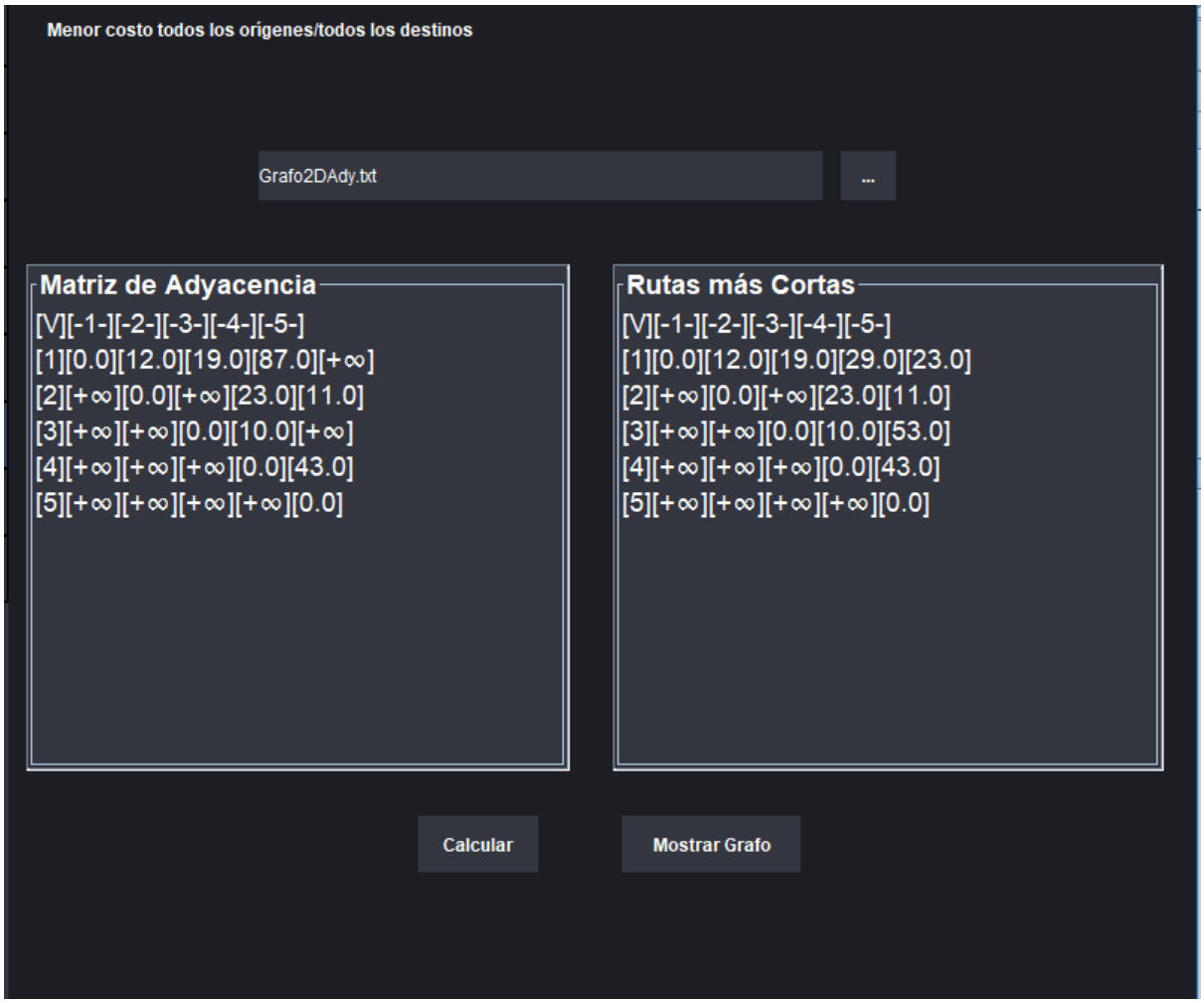
V. Orígenes/Destinos

Objetivo: Implementar el algoritmo de Floyd para determinar el costo mínimo entre todos los orígenes y todos los destinos.

Comenzamos cargando un grafo (Referencia: **IV. Bicoloreable**), que contenga la matriz de adyacencia del grafo dirigido que queremos calcular. Una vez cargado el grafo, se mostrará el mismo en una nueva ventana y se llenará el cuadro “Matriz de Adyacencia” con la correspondiente al grafo. Presione “Calcular” para calcular las rutas mas cortas de todos los orígenes, todos los destinos. Una vez calculadas, estas serán mostradas en el cuadro “Rutas más Cortas”.

Presione “Mostrar Grafo” para volver a ver el grafo.

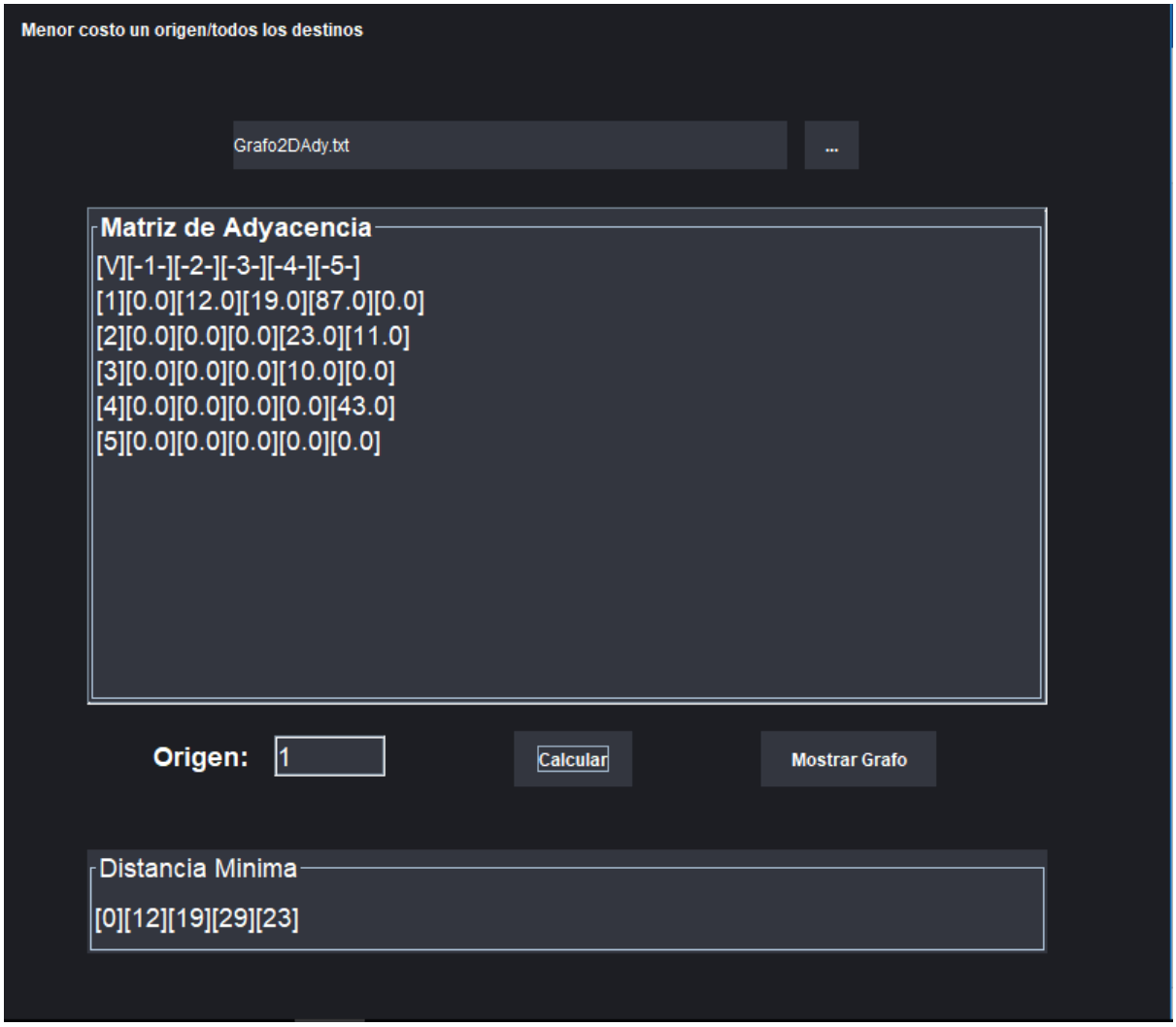
Hágase notar que “+∞” representa que no hay conexión directa entre esos dos nodos.



VI. Origen/Destinos

Objetivo: Implementar el algoritmo de Dijkstra para determinar el costo mínimo entre uno de los orígenes y todos los destinos.

Comenzamos cargando un grafo (Referencia: **IV. Bicoloreable**), que contenga la matriz de adyacencia del grafo dirigido que queremos calcular. Una vez cargado el grafo, se mostrará el mismo en una nueva ventana y se llenará el cuadro “Matriz de Adyacencia” con la correspondiente al grafo. En la casilla “Origen” deberá ingresar de que nodo se quiere comenzar. A diferencia del Bicoloreable, este problema no aceptará que deje en blanco esta casilla. Una vez ingresado el origen, presione “Calcular”. Una vez calculado, se llenará el cuadro “Distancia Mínima” con las rutas más cortas desde el origen ingresado hasta todos los destinos.



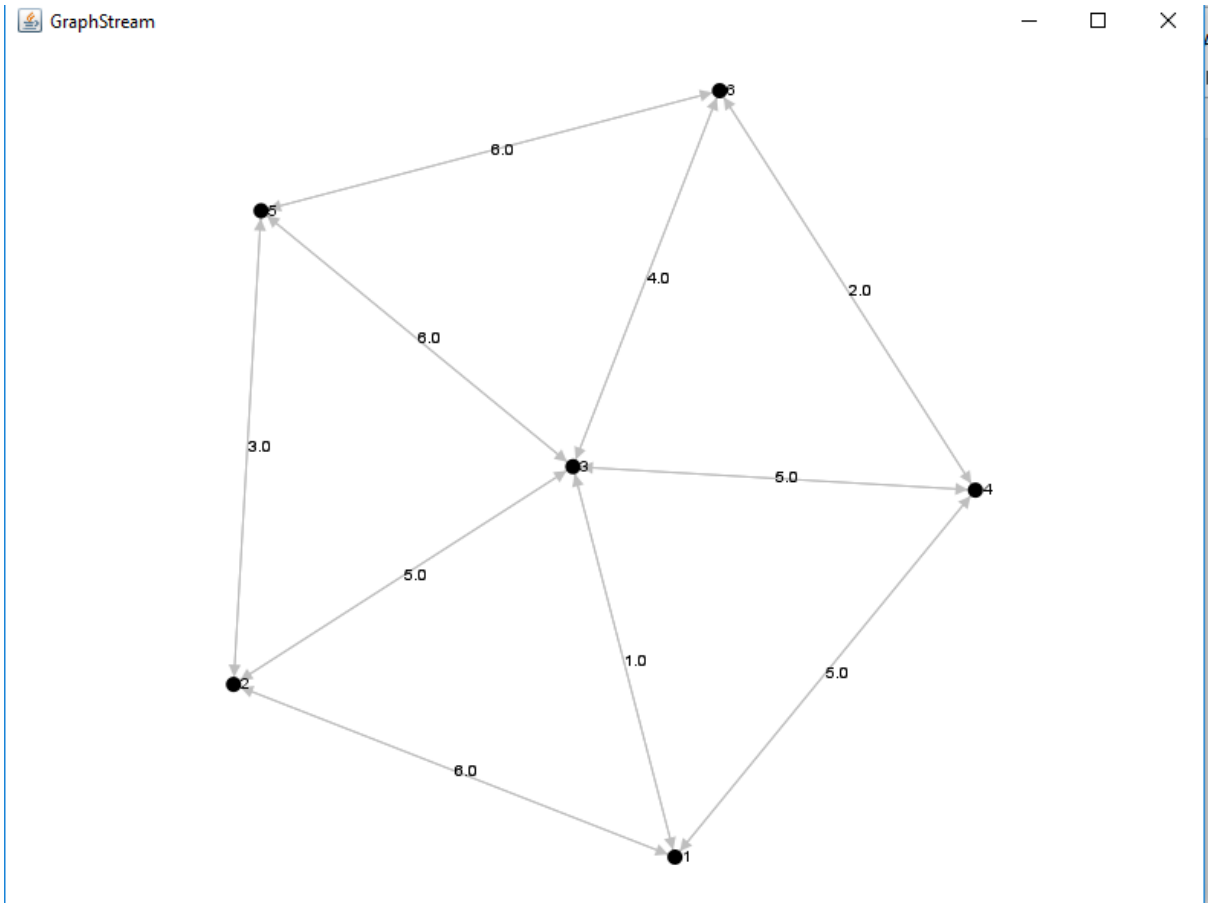
VI. Árbol de Expansión Mínima

Objetivo: Implementar el algoritmo de Prim para determinar el árbol de expansión mínima para un grafo dado.

Comenzamos cargando un grafo (Referencia: **IV. Bicoloreable**), que contenga la matriz de adyacencia del grafo dirigido que queremos calcular. Presione “Determinar” para calcular el árbol de expansión mínima. Una vez calculado, se llenará el cuadro “Árbol de Expansión Mínima” con una representación escrita del árbol y simultáneamente se mostrará el árbol generado.

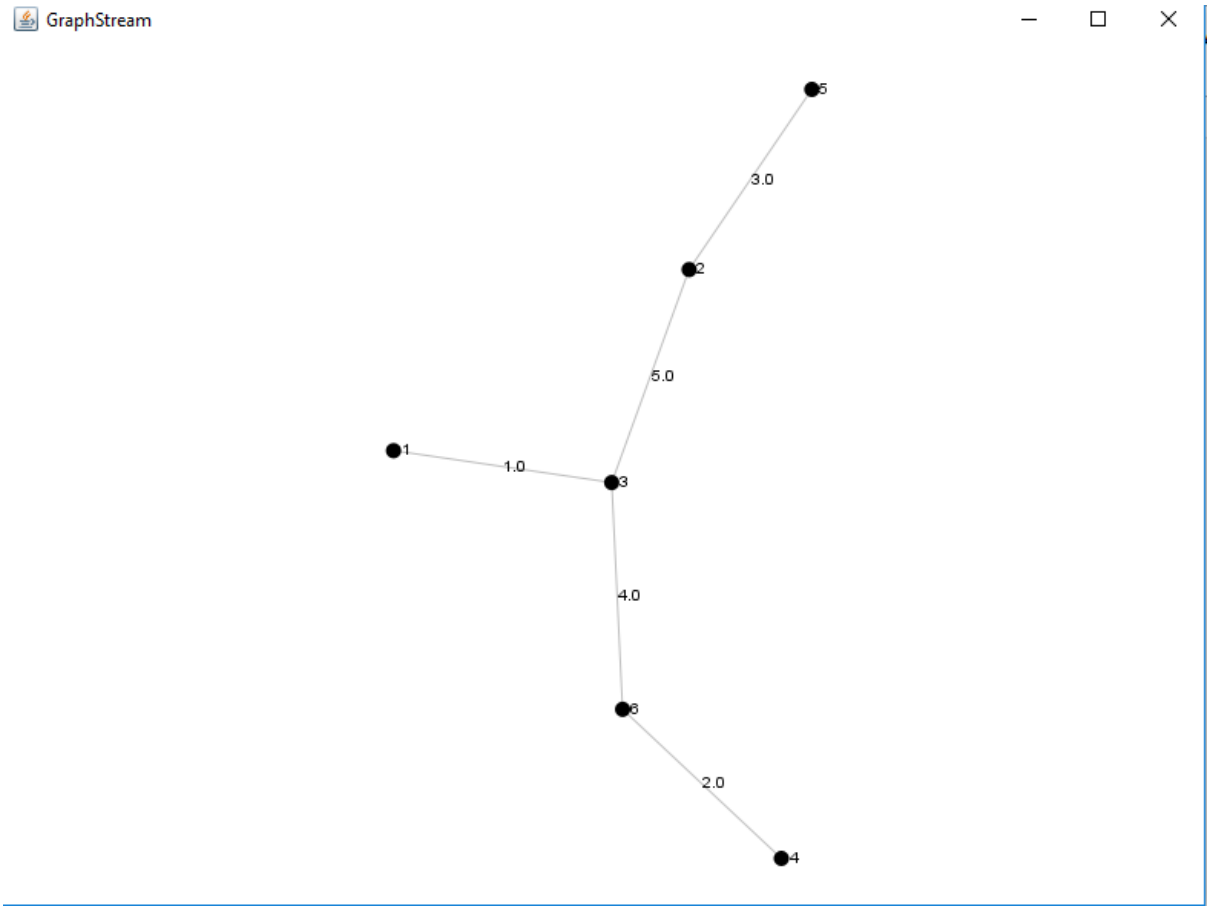
Si quiere volver a ver el grafo, presione “Mostrar Grafo”. Si quiere ver el árbol, presione “Mostrar Árbol”.

Grafo por Calcular:



Una vez ya calculado el árbol:

Árbol en forma gráfica:



Árbol en forma escrita:

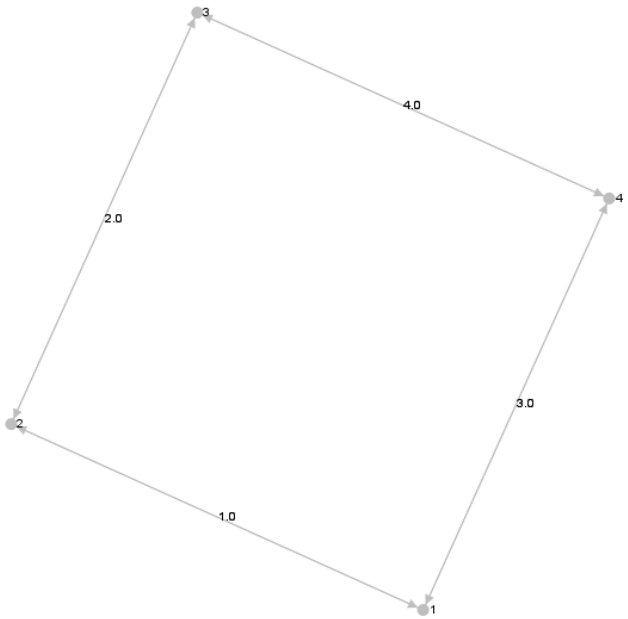
Árbol de Expansión Mínima

Desde 3 hasta 2 con costo 5.0
Desde 1 hasta 3 con costo 1.0
Desde 6 hasta 4 con costo 2.0
Desde 2 hasta 5 con costo 3.0
Desde 3 hasta 6 con costo 4.0

VII. Anexos

Sobre el archivo de texto y la matriz de adyacencia

Un grafo puede ser representado por una matriz de adyacencia. Existe una diferencia entre la matriz de adyacencia para un grafo dirigido, y la matriz de adyacencia para un grafo no dirigido. Pero ambas se leen y representan de la misma manera. Tómese de ejemplo el siguiente grafo no dirigido:



Su representación en una matriz de adyacencia sería la siguiente:

0,1,0,3
1,0,2,0
0,2,0,4
3,0,4,0

Donde el número de la fila es el nodo origen, el numero de la columna es el nodo destino, y el numero contenido en esa coordenada es el peso de la arista.

Así será como se leerá el grafo desde el archivo de texto, mediante la matriz de adyacencia.