****

**Universidad Estatal a Distancia**

**Diplomado en Informática**

**Cátedra de Tecnología de Sistemas**

**Escuela de Ciencias Exactas y Naturales**

**Proyecto #1**

**Estructuras de Datos**

**Julio 2015**

Olman A Rojas Espinoza

INTRODUCCION

En el siguiente proyecto vamos a analizar el algoritmo de Djikstra implementado en el lenguaje de Programación JAVA.

El programa cuenta con una interfaz GUI de tal manera que se reduce la necesidad de crear varias excepciones y validaciones de los tipos de Datos a utilizar para la entrada de los mismos.

A su vez la Interfaz, sólo va a validar entrada de tipo numéricos enteros, obviando cualquier entrada de tipo letras o bien negativos o decimales.

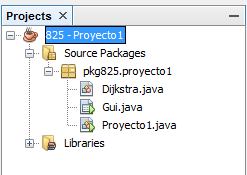
El sistema también valida si la matriz de pesos es simétrica o no.

En teoría se deberían de manejar dos matrices una de adyacencia y otra de pesos, sin embargo para este ejercicio se consideró una única matriz, de tal manera que si un dato es insertado este va a indicar que el vértice se utiliza y el número que este en ese vértice es el peso de la arista.

Cabe mencionar que todo el código se encuentra documentado internamente.

DESARROLLO

Para empezar este programa sólo cuenta con 3 archivos los cuáles son los siguientes:



Donde

**Dijkstra.java**

Es la clase que contiene la lógica para resolver el camino más corto, recibe como entrada una matriz de 5x5 y luego devuelve un String con el resultado de los caminos y la distancia total.

**Gui.java**

Es la clase que contiene la interfaz, valida la entrada de enteros positivos, contiene los botones para tomar la matriz, y a su vez instancia a la clase Dijkstra.java para resolver el algoritmo, una vez que recibe el resultado tipo String, imprime en pantalla para que el usuario pueda ver el resultado.

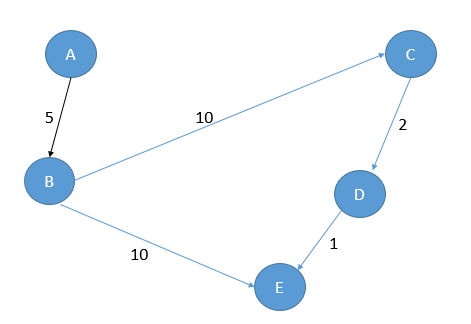
También existe un método llamado **calcular\_simetria()** , el cual consta de un ciclo sencillo que demuestra si la matriz es simétrica o no.

**Proyecto1.java**

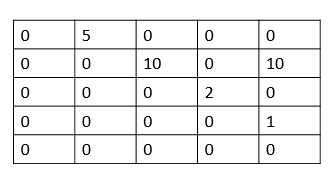
Es la clase principal que contiene el método main, e instancia el Gui.java.

Ejemplo de Funcionamiento:

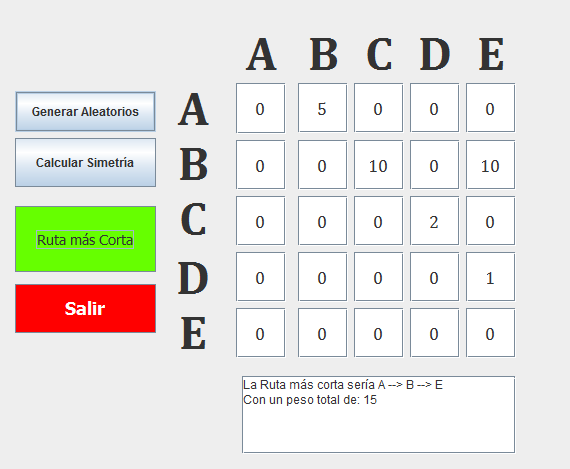
Observemos el siguiente grafo.



Para interpretar este grafo tenemos que traducirlo a una matriz, quedaría de la siguiente forma:



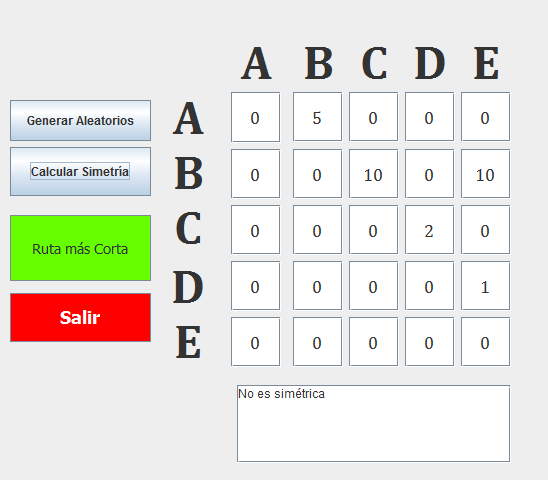
Dicho esto se procede a correr el programa.



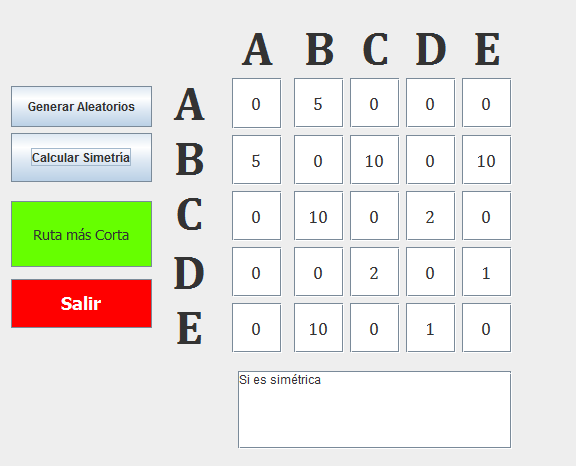
Al introducir la matriz del grafo anterior y haciendo click en el botón “Ruta más Corta”, se puede observar como el Sistema es capaz de proporcionar el camino correcto a la ruta más corta y a su vez indica el peso total o distancia que tomaría llegar del punto A al punto E.

**Prueba de Simetría.**

Utilizando el ejemplo anterior hacemos click en “Calcular Simetria”:



Si quisiéramos forzar la simetría cambiamos algunos valores:



CONCLUSION

Como se pudo observar en el programa anterior, el algoritmo de Djikstra permite calcular la ruta más corta para llegar a un destino incluyendo la distancia total para llegar al destino.

En las aplicaciones actuales un gran ejemplo es el Internet, hoy en día existen muchos caminos para acceder a un servidor remoto, sin embargo los enlaces pueden caer, o pueden estar despacio o bien el camino puede ser muy largo, hacer estos cálculos pueden ser tediosos, sin embargo con ese algoritmo se resuelve rápido y cada host es el encargado de brindarle al que peticiona una ruta, ya sería internamente el dispositivo o servidor el que analiza si toma o no esa ruta para acceder.

No sólo en Internet se pueden tomar este algoritmo, también puede utilizarse en logística, para envío y recepción de mercaderías y que estén a tiempo.

BIBLIOGRAFIA.

Weiss Mark, A. (2013). *Estructura de Datos en Java.* Madrid España: Pearson Educación S.A.