

LABORATORIO DE PROGRAMACIÓN III (Curso 2011/2012)

PRÁCTICA 2 FECHA LÍMITE: 20 de abril a las 19h

Objetivos de la práctica:

1. Diseño de gráficos: objetos gráficos, figuras básicas, formas bidimensionales
2. Algoritmos de recorridos de grafos
3. Casos de prueba usando JUnit
4. Tipos genéricos en Java
5. Patrones de diseño
6. Clases e interfaces en las bibliotecas de Java
(Map, HashMap, Set, HashSet, Comparator, Iterator, PriorityQueue, etc.)

Desarrolla una aplicación Java que permita introducir un grafo ponderado y obtener:

1. El árbol de expansión mínimo utilizando el algoritmo de Kruskal.
2. El camino mínimo entre dos vértices del grafo, utilizando los algoritmos de Dijkstra y Floyd.

La aplicación ofrecerá los siguientes menús para facilitar el diseño del grafo:

- Un menú **Grafo** para introducir nuevos vértices y aristas entre ellos con las siguientes opciones:
 1. *Introducir vértice*: para introducir un nuevo vértice en el grafo. El nombre del nuevo vértice se obtendrá a través de una ventana de diálogo y su ubicación en el grafo se seleccionará pulsando con el ratón en la posición deseada.
 2. *Introducir arista*: para introducir una nueva conexión en el grafo. El valor de la arista se obtendrá a través de una ventana de diálogo. Con dos clicks sucesivos de ratón se seleccionarán dos vértices entre los ya introducidos en el grafo.

En cualquier momento de la ejecución de la aplicación debe ser posible introducir nuevos vértices y aristas. Opcionalmente, también puedes añadir opciones en el menú para eliminar vértices y aristas previamente introducidos.

- Un menú **Archivo** para guardar y recuperar en un archivo la información sobre un grafo previamente creado a través del menú **Grafo** con las siguientes opciones.
 1. *Guardar*: para guardar la información actual en un archivo cuyo nombre se solicita a través de una ventana de diálogo.
 2. *Abrir*: para leer la información de un archivo cuyo nombre se solicita a través de una ventana de diálogo.

El archivo puedes guardarlo en el formato que quieras.

- Un menú **Operaciones** con los submenús:
 1. *Dijkstra*: basándose en el algoritmo voraz para búsqueda de caminos mínimos de Dijkstra. Con dos clicks sucesivos de ratón se seleccionarán los vértices origen y destino. El programa marcará sobre el grafo, con un trazo de distinto color, el trayecto óptimo.
 2. *Floyd*: como 1, pero basándose en el algoritmo de programación dinámica para búsqueda de caminos mínimos de Floyd descrito en la pág. 301 de Brassard y Bratley.
 3. *Kruskal*: basándose en el algoritmo voraz para calcular el árbol mínimo de expansión de Kruskal. El programa marcará sobre el grafo, con un trazo de distinto color, el árbol.

Implementa casos de prueba utilizando el framework JUnit, para demostrar que las implementaciones realizadas son correctas. Los casos de prueba contendrán al menos tres grafos, con al menos 5 vértices cada uno.

Los detalles de implementación se ampliarán en clase.