

# Sigraf: Simulador de GRAFos.

Judith Antolín Sendino

María Ruiz Ruiz

Escuela Politécnica Superior,  
Universidad de Burgos  
{jas00, mrr0012}@alu.ubu.es

Carlos Pardo Aguilar

Juan J. Rodríguez Díez

Dpto. de I. Civil, Lenguajes y Sistemas Informáticos  
Universidad de Burgos  
{cpardo, jjrodriguez}@ubu.es

## Resumen

En el presente artículo se presenta una herramienta de apoyo al aprendizaje en un campo tan importante de la informática como las Estructuras de Datos, centrándose en el diseño, desarrollo y aplicación del Tipo Abstracto de Datos (TAD) grafo para la resolución de diversos tipos de problemas.

Aunque se encuentran disponibles diversas animaciones de algoritmos sobre grafos, normalmente éstas se limitan a un solo algoritmo (e.g., Dijkstra), o a un solo problema (e.g., árboles expandidos mínimos). La posibilidad de introducir un grafo y poder ejecutar interactivamente toda una gama de algoritmos sobre el mismo, es claramente ventajosa con respecto a tener que introducir el mismo grafo para diversos algoritmos, utilizando una interfaz diferente para cada uno de ellos.

## 1. Introducción

La disponibilidad de herramientas que permitan al alumno trabajar con conceptos adquiridos en las clases teóricas, así como resolver dudas, es escasa, además las horas de prácticas de las que se dispone en la universidad son limitadas. Por ello, es importante la disponibilidad de herramientas que proporcionen un entorno de pruebas y desarrollo de las diferentes cuestiones que van aprendiendo, que redunde en un mayor aprovechamiento de las horas dedicadas a clases prácticas.

Se considera de especial interés el estudio de los grafos que se usan para modelar cualquier situación en la que existan elementos unidos con otros tales como redes de alcantarillado, redes de comunicación, circuitos eléctricos, entre otros.

Una vez modelado el problema mediante un grafo se pueden hacer estudios sobre diversas propiedades, para ello se utilizan algoritmos concretos que resuelvan ciertos problemas. La teoría de grafos ha sido aplicada en el estudio de problemas que surgen en áreas diversas de las ciencias, como la química, la ingeniería eléctrica o la investigación operativa. El primer paso siempre será representar el problema como un grafo. En esta representación cada elemento, cada objeto del problema se representa mediante un nodo. La relación, comunicación o conexión entre los nodos da lugar a una arista, que puede ser dirigida o bidireccional (no dirigida) [1].

En el ámbito de asignaturas de Estructuras de Datos, que tienen gran carga teórica y aplicación práctica, se pueden encontrar demostraciones de algunos algoritmos sueltos [3], [6]. Sin embargo, no se ha encontrado disponible ninguna herramienta que abarque aquéllos que normalmente se incluyen dentro de estas asignaturas. La utilización de una herramienta distinta para cada algoritmo es incómoda, puesto que cada una de ellas tiene su peculiar manera de editar un grafo. Por ello se planteó el desarrollo de una aplicación para el estudio de este TAD y de varios de sus algoritmos. Aunque sería deseable disponer de un entorno de visualización de todas las estructuras de datos habituales, entendemos que la cantidad de algoritmos que se pueden realizar sobre grafos justifica una herramienta para su estudio. Por otro lado, dentro de las estructuras de propósito general, la de grafo es la más compleja, así que se considera que es el primer paso más adecuado hacia un sistema de animación de cualquier estructura de datos.

El resto del artículo se organiza del siguiente modo. La sección 2 presenta los objetivos planteados. La herramienta se presenta en la

sección 3. Finalmente se exponen las conclusiones en la sección 4.

## 2. Objetivos

El objetivo era crear una aplicación que permitiera diseñar grafos, guardarlos, modificar otros ya existentes, imprimirlos y ver de forma didáctica y atractiva los distintos algoritmos que se estudian sobre ellos. Todo ello de forma gráfica y fácil de utilizar para ayudar en su comprensión a los alumnos.

La herramienta se ha desarrollado utilizando el paradigma Orientado a Objetos, ya que facilita la reutilización de las aplicaciones y es probable que esta herramienta en un futuro sea ampliada en posteriores proyectos finales de carrera u otros trabajos de investigación.

Uno de los aspectos que se tuvo en cuenta fue la necesidad de diferenciar entre dos modos de funcionamiento del grafo, por un lado el grafo abstracto que es el que se encarga de modelar la realidad y sobre el que se ejecutan los algoritmos, y por otro lado las propiedades gráficas para poder representarlos en la pantalla (posición, color, etc.).

Como consecuencia se generaron dos clases para cada modo de funcionamiento, con las relaciones mostradas en la Figura 1.

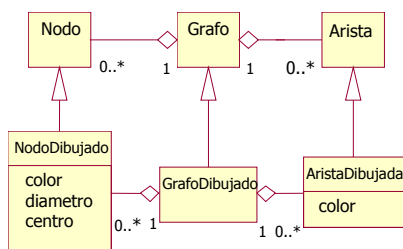


Figura 1: Diagrama de clases

## 3. La herramienta

Para la realización de esta aplicación se ha estudiado el comportamiento de otros programas en situaciones específicas, intentando obtener un producto con el que el usuario estuviera

familiarizado y su uso fuese lo más intuitivo posible, e intentando adaptar su modo de trabajo y sus funcionalidades a un entorno como el de la enseñanza, así como mejorar los aspectos más importantes. Se consideró relevante incrementar al máximo la disponibilidad y portabilidad de la aplicación, estando disponible ésta en la página web de la asignatura para que los alumnos puedan descargarla cuando deseen, y pudiendo ser ejecutada en distintos sistemas operativos, ya que el hecho de estar implementada en Java, lenguaje robusto, seguro y de gran portabilidad ayuda a la obtención de estas características.

La aplicación consta de una interfaz gráfica sencilla y fácil de manejar, que permite al usuario trabajar con esta estructura de datos de forma rápida y eficaz (ver Figura 2). Sobre este aspecto se ha de comentar uno de los requisitos más relevantes, la facilidad y comodidad de manejo que incluye que todas las operaciones de la aplicación se podrán realizar tanto por ratón como por teclado. Se debe tener en cuenta que si además del temario de la asignatura y de una herramienta específica, el alumno debe aprender a trabajar con otra adicional, ésta debe intentar facilitar su trabajo sin añadir complejidad.

Desde el menú principal, se puede acceder en cualquier momento a la ayuda de la aplicación para obtener información sobre cualquier aspecto relacionado, tanto con la forma de diseñar un grafo como con la aplicación de los algoritmos con la herramienta, además de las explicaciones teóricas sobre la estructura para facilitar su aprendizaje. Para todas estas funcionalidades se proporciona una barra de herramientas con botones significativos para cada opción (ver Figura 3).

Durante todo el proceso de desarrollo y modificación del grafo se realizan diversas comprobaciones, evitando por ejemplo que se dibujen nodos solapados, que antes de poder dibujar una arista existan los nodos que esta desea unir, que cuando se borra un nodo se borren todas sus aristas asociadas. En la barra de estado de la aplicación se indican mensajes de error sobre las comprobaciones anteriores, se visualizan las coordenadas del puntero de ratón en todo momento y si se selecciona un elemento del grafo pueden obtenerse sus propiedades.

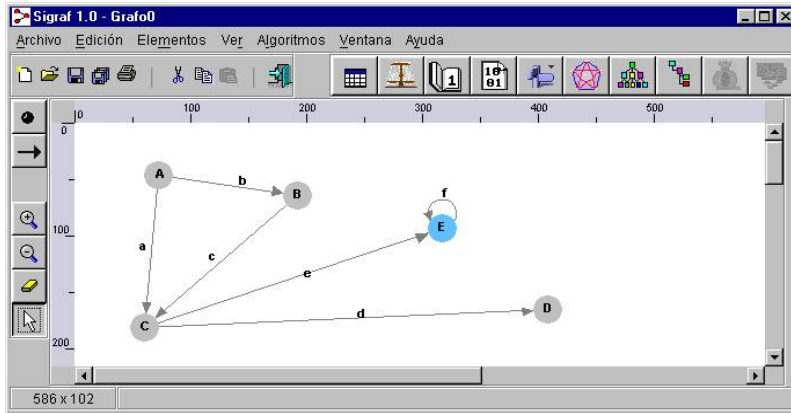


Figura 2: Interfaz de la aplicación.



Figura 3 Barra de herramientas de Dibujo

### 3.1. Representaciones de los grafos

Dentro de las diferentes visualizaciones de los grafos se encuentran la matriz de adyacencias, la matriz de costes y la lista de adyacencias. Cada una de estas puede obtenerse a través del menú 'Ver' de la aplicación, desde el cual también podemos configurar qué propiedades de las aristas del grafo se desean mostrar y obtener más información sobre cada uno de los nodos, del grafo.

### 3.2. Ejecución de los algoritmos sobre grafos

La ejecución y visualización de los algoritmos, igual que el resto de opciones de la herramienta, se puede realizar a través del menú o desde la barra de herramientas específica de estos. Dichas visualizaciones se realizan en una ventana auxiliar, bien de forma gráfica mostrando los cambios mediante colores sobre

el propio grafo, bien con los datos obtenidos (matrices de caminos, caminos mínimos...) o con ambas cosas según el algoritmo.

En la mayoría de ellos puede realizarse su ejecución paso a paso para seguir la evolución del algoritmo de forma más didáctica para el alumno.

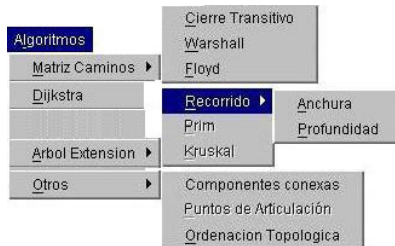


Figura 4 Menú de algoritmos.

### 3.3. Tratamiento de ficheros

Para el almacenamiento de los grafos se generan ficheros de texto con un formato específico y extensión ".grf" para identificarlos. Las operaciones sobre éstos (crear, abrir, cerrar,

guardar...), se realizan a través del menú 'Archivo' o desde la barra de herramientas estándar. Al permitir tener varios ficheros abiertos a la vez, se lleva un control de los que han sido modificados y se proporciona la posibilidad de cambiar de uno a otro desde el menú 'Ventana'.

### 3.4. Ejemplo de ejecución de un algoritmo.

El *cálculo de las componentes conexas* de un grafo consiste en obtener aquellos subconjuntos de nodos entre los cuales la comunicación es total. En este caso no se proporciona la opción de realizarlo paso a paso, ya que no se cree necesario para que el alumno comprenda el desarrollo del mismo. Un grafo no dirigido  $G$  es conexo si existe un camino entre cualquier par de nodos que forman el grafo. En el caso de que el grafo no sea conexo se pueden determinar todas las componentes conexas del mismo.

En un grafo dirigido podemos distinguir entre grafo dirigido conexo y grafo fuertemente conexo. Un grafo dirigido es conexo si para cada par de vértices existe una cadena que los une. Y un grafo dirigido es fuertemente conexo si para cada par de vértices existe un camino que los une. Por tanto, para saber si un grafo es o no conexo se calculan sus componentes, si tras visitar todos sus nodos se obtienen una única componente se dice que el grafo es conexo.

La visualización realizada para este algoritmo consta de una pantalla dividida en dos paneles. En el de la izquierda se visualiza el grafo formado por las componentes conexas obtenidas, mientras que en la parte derecha se muestran los nodos que forman la componente elegida o todo el grafo. Para facilitar la comprensión al alumno, se realiza una asociación de colores usando el mismo para una componente en el panel de la izquierda, que para los nodos por los que está formado en la parte derecha.

## 4. Conclusiones y trabajos futuros

Debido a que la herramienta se ha desarrollado recientemente y dada la situación temporal del tema dedicado a grafos al final de la asignatura de Estructuras de Datos, es demasiado pronto como para establecer conclusiones respecto a su

utilidad. No obstante, se considera que aunque sólo sea como "transparencias interactivas" la herramienta es útil. Queda por determinar hasta que punto los alumnos utilizan la herramienta por su cuenta, y su valoración de la misma.

Dentro de las posibles ampliaciones, algunas de las cuales ya están en marcha, se pueden comentar las siguientes. Por un lado, las mejoras en la capacidad de edición y visualización, como mostrar la traza del algoritmo además del grafo, permitir arcos curvados o colocar automáticamente los nodos y arcos. Para este último aspecto se planea utilizar alguna herramienta como Graphviz [4]. El formato de fichero utilizado para almacenar los grafos es actualmente un formato propio, por lo que parece interesante incluir la posibilidad de importar y exportar a otros formatos, por ejemplo GraphML [2]. Aunque en la versión actual están disponibles bastantes algoritmos, este apartado se puede completar, con por ejemplo, cuestiones relativas a flujo en redes o problemas NP.

## Referencias

- [1] Mark Allen Weiss "Estructuras de datos en Java". Addison-Wesley.
- [2] U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, and M.S. Marshall: GraphML Progress Report: Structural Layer Proposal. Proc. 9th Intl. Symp. Graph Drawing (GD'01), LNCS 2265, pp. 501-512. Springer-Verlag, 2002.
- [3] Peter Brummund, Ngozi V. Uti. "The Complete Collection of Algorithm Animations". <http://www.cs.hope.edu/~alganim/ccaa/>
- [4] Emden R. Gansner "Drawing graphs with GraphViz" 2003. <http://www.research.att.com/sw/tools/graphviz/libguide.pdf>
- [5] Luis Joyanes Aguilar, Ignacio Zahonero Martínez "Estructuras de datos. Algoritmos, abstracción y objetos". Mac Graw Hill 1999.
- [6] M. Gloria Sánchez Torrubia, Victor M. Lozano Terrazas. "Algoritmo de Dijkstra. Un tutorial interactivo". VII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2001)