

EDITOR E INTÉRPRETE DE ALGORITMOS REPRESENTADOS EN DIAGRAMAS DE FLUJO¹

**Fabián CÁRDENAS VARELA
Nelson CASTILLO IZQUIERDO
Eduardo DAZA CASTILLO**

RESUMEN

El uso de la informática como apoyo al proceso de aprendizaje y su asimilación dentro de entidades educativas aumenta día a día y ha tomado un gran auge en los últimos años, lo cual provoca la demanda creciente de software educativo de la más alta calidad. De ahí la necesidad de incorporar metodologías que hagan más productiva la preparación del desarrollador de software. Así como la práctica del desarrollo de software contribuye a satisfacer las necesidades educativas en muchos campos, también permite obtener mejoras en etapas básicas de capacitación en análisis y diseño de software, impulsando de esta manera su propio desarrollo. En este trabajo se describe el proyecto *Editor e Intérprete de Algoritmos Representados en Diagramas de Flujo* mostrando cómo se logra eliminar ciertas dificultades presentes en el estudio de algoritmos básicos y obtener significativas ventajas, mediante la implementación de herramientas de software.

INTRODUCCIÓN

Para el estudio formal de un problema de cualquier tipo es necesario seguir una serie de pasos que lleven a su entendimiento y solución, es decir, es necesaria la utilización de un algoritmo por medio del cual se resuelva dicho problema; esta herramienta es útil aún en problemas cotidianos. Uno de los conceptos más aceptados de algoritmo

¹ Trabajo ganador del *Premio Colombiano de Informática Educativa 1998, categoría Investigación y Desarrollo*. Este premio es auspiciado por el Ministerio de Educación Nacional y convocado y otorgado por RIBIE-COL, Red Iberoamericana de Informática Educativa - nodo Colombia. Los interesados en explorar el software DFD lo pueden solicitar al MEN (División de Medios Educativos), a RIBIE-COL (<http://ribie-col.eafit.edu.co>) o al autor (Smart@compunet.net.co).

dice: "es un procedimiento para la resolución de problemas de cualquier tipo por medio de una determinada secuencia de pasos simples y no ambiguos". El concepto fue utilizado originalmente para el cálculo matemático, pero, ahora es ampliamente usado en la programación de computadoras.

Para la representación, análisis y estudio de algoritmos ha sido de gran utilidad el uso de diagramas de flujo, los cuales muestran de forma clara y sencilla el comportamiento de un algoritmo. Un diagrama de flujo es una descripción gráfica de un procedimiento para la resolución de un problema. Son frecuentemente usados para describir algoritmos y programas de computadora debido a que pueden suprimir detalles innecesarios y tener un significado preciso si son usados correctamente.

Los diagramas de flujo son muy utilizados en los niveles básicos de enseñanza y aprendizaje de algoritmos. En algunos casos no son usados por presentar deficiencias en varios aspectos: La distribución y aprovechamiento del espacio, el cuidado de la estética mientras se están dibujando diagramas, la tendencia que tienen las personas a diagramar sin un estándar y el excesivo tiempo invertido, son aspectos que limitan la aplicabilidad de los diagramas de flujo. Estas limitaciones pueden ser superadas vinculando herramientas informáticas a la diagramación. Existen herramientas informáticas que facilitan la creación de diagramas de flujo, de hecho, son muy populares en el mercado. Sin embargo, estas herramientas están más orientadas al diseño gráfico que al estudio de algoritmos. Esto es, no interpretan algoritmos ni detectan errores en los mismos, fueron concebidas pensando más en los diagramas que en los algoritmos que éstos representan.

La base de la programación de computadoras esta en el manejo de los algoritmos; Por lo tanto los niveles básicos de enseñanza de estos últimos exigen al instructor buenos métodos, y al estudiante aptitud y mucho interés, por lo cual es muy importante el vínculo de la informática para hacer más fácil y productiva la adquisición de estos nuevos conocimientos.

El software Dfd[®] fue diseñado para estudiar algoritmos representados con diagramas de flujo, más que para dibujar diagramas. Sin embargo, su interfaz gráfica facilita en gran medida la creación de diagramas de flujo.

DESARROLLO DEL PROYECTO

El objetivo principal de la vinculación de la informática en este ámbito estuvo fijado en incrementar la calidad y eficiencia en el proceso educativo en materia de algoritmos básicos, para lograr así una introducción más formal y confiable al campo de la

programación de computadoras. Durante el diseño de este proyecto se presentó como alternativa eficaz el estudio y consecución de los siguientes puntos:

- Introducir un método didáctico e interactivo para la enseñanza y el estudio del comportamiento de algoritmos.
- Lograr que desde el inicio, los conocimientos adquiridos en el estudio de algoritmos, puedan ser llevados a la práctica.
- Propiciar en el estudiante la autoevaluación de estilo, conocimiento y aptitudes para la realización de algoritmos.
- Facilitar herramientas para el desarrollo de la creatividad y aumento del interés en estudiantes que se encuentran en niveles básicos de aprendizaje.
- Utilizar la Informática para economizar esfuerzos innecesarios en cálculos complejos que en ciertas ocasiones parecen ser más importantes que el objeto de estudio; esto es, reducir a 0 los errores en la evaluación de expresiones lógicas, y a valores despreciables, los errores por cálculos numéricos; así los esfuerzos pueden invertirse en la comprensión del algoritmo como tal.

OBJETIVOS PROPUESTOS

A partir de nuestra experiencia como estudiantes de Ingeniería de Sistemas y programadores de computadoras, y de la experiencia y conocimiento de estudiantes de diferentes niveles en áreas de Ingeniería de Sistemas e Informática, se pudo observar las ventajas y deficiencias que tienen los métodos de enseñanza en niveles básicos. Con base en esto, se planteó un problema y un objetivo general para la investigación.

Para lograr el objetivo propuesto, se realizó un riguroso estudio de las siguientes alternativas: a) Creación de un software tutor de algoritmos, éste contendría una amplia explicación de ejemplos y conceptos necesarios para aprender algoritmos, disponibles a través de gráficas, animaciones, sonido y texto. b) Elaboración de un manual de algoritmos, el cual mostraría a través de gráficos y texto los ejemplos y conceptos enunciados en la alternativa anterior; éste sería más portable que el software, ya que no sería indispensable el uso de medios electrónicos. c) Desarrollo de un software editor e interprete de diagramas de flujo con las características de Dfd[®]². De acuerdo con el estudio realizado, se optó por el diseño de la herramienta de software Dfd[®], la cual reúne las condiciones para el cumplimiento del objetivo planteado.

² La descripción del software Dfd[®] se encuentra al final del documento.

Para llevar a cabo el desarrollo del proyecto fue necesario la utilización de computadoras personales, aulas, software de base y aplicación, conexión a Internet y material bibliográfico adecuado.

RESULTADOS OBTENIDOS

Como resultado de la investigación, se ha desarrollado la versión en español del software Dfd[®]. Una herramienta informática como ésta no se encuentra disponible en el mercado y no se tiene conocimiento de alguna entidad que se encuentre utilizándola³. Como método para el afianzamiento de los conocimientos obtenidos en determinadas áreas, se han implementado laboratorios que llevan a la práctica la teoría adquirida en el estudio de las mismas; Sin embargo, no siempre se cuenta con las herramientas necesarias para el desarrollo de un laboratorio y debido a esto, la necesidad de practicar lo aprendido termina siendo ignorada. Los algoritmos básicos no son la excepción. En la Universidad del Magdalena se encuentra en curso el proyecto "Evaluación de los Resultados de la Implementación de un Laboratorio de Algoritmos Básicos en el Programa de Ingeniería de Sistemas de la Universidad del Magdalena", el cual utiliza como herramienta principal el software Dfd[®]. El proyecto anterior pretende obtener los beneficios de un laboratorio de algoritmos implementado en etapas básicas de aprendizaje.

Por las características del software, su aplicación abarca todas las instituciones y personas que se encuentren relacionadas con la informática. Por ser un software de base que reúne los requisitos para ser un intérprete de un lenguaje de programación de alto nivel⁴, los beneficios obtenidos dependen de la creatividad del usuario. Éste último puede crear software que corra en Dfd[®] y que pueda ser fácilmente traducido a cualquier otro lenguaje de programación estructurada.

³ Después de exhaustivas indagaciones, incluyendo búsquedas en Internet, no se logró encontrar una herramienta similar al software Dfd[®].

⁴ "Los programadores de computadoras necesitan expresar problemas de forma que sean independientes del diseño de la CPU. También prefieren no expresar cada detalle cuando escriben un software para computadora. Por esta razón, Los lenguajes de programación de alto nivel han sido desarrollados. Con un lenguaje de alto nivel el usuario puede obtener la suma de dos números dando a la computadora un comando como "PRINT 512 + 637." Para que esto sea posible, los lenguajes de alto nivel deben traducir a lenguajes de bajo nivel. Esto se hace a través de un computador con un intérprete o con un compilador".

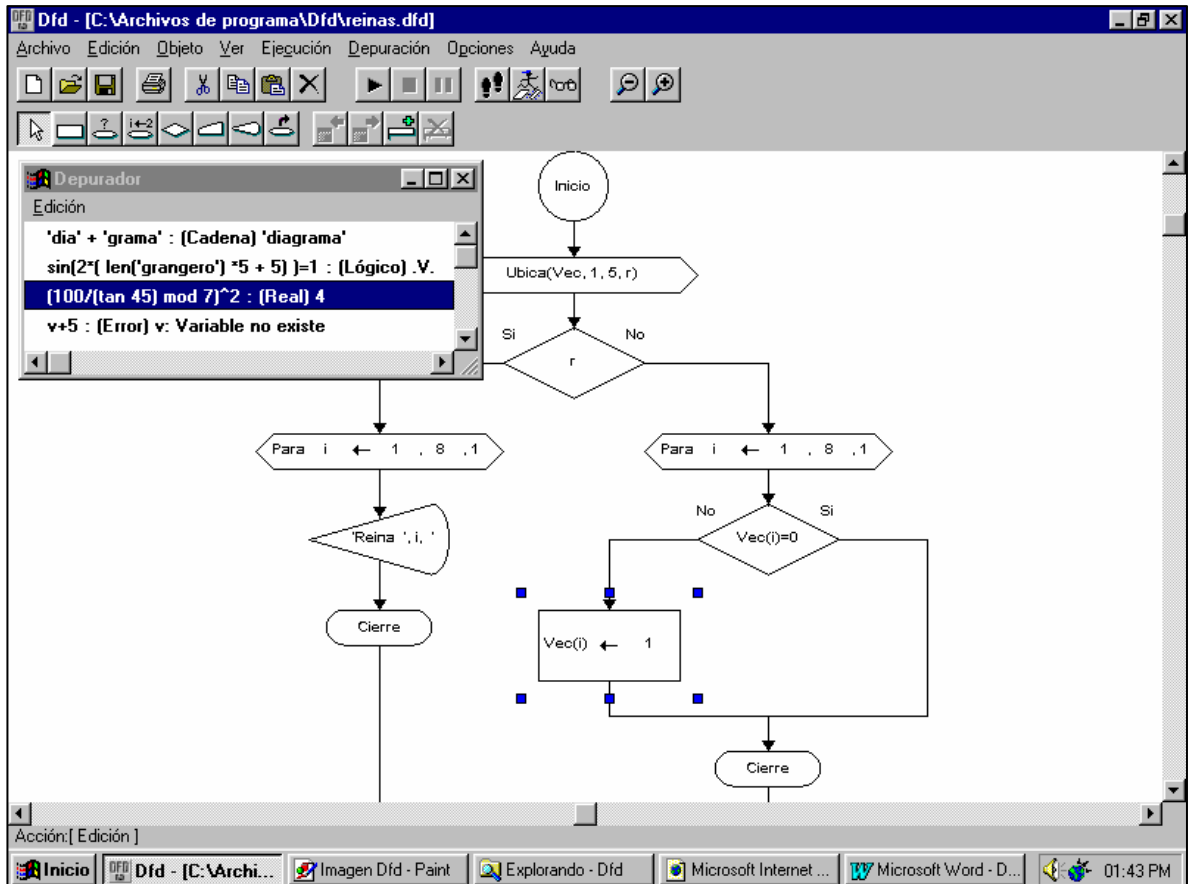
DESCRIPCIÓN DEL SOFTWARE DFD®⁵

Ilustración 1 : Edición de una diagrama en Dfd®

Dfd® es un editor e intérprete de diagramas de flujo. Su interfaz gráfica facilita en gran medida la creación de diagramas de flujo para la representación de algoritmos que solucionan problemas por computadora. Dichos diagramas pueden ser guardados en disco, recuperados de disco y pueden ser impresos en diferentes tamaños sin importar el

⁵ El software Dfd® se encuentra inscrito en el Registro Nacional de Derechos de Autor de Colombia, correspondiente al libro 13, tomo 3, partida 166.

tipo de impresora. En un disco de 3^{1/2}" de alta densidad se pueden almacenar más de 360 algoritmos de mediana complejidad como archivos de Dfd[®].

Los algoritmos creados pueden ser ejecutados para observar su comportamiento. Para un estudio detallado de los mismos, el programa brinda herramientas para la depuración como la "Ejecución Paso a Paso", "Ejecutar Hasta", una "Ventana de Depuración" que muestra cómo cambian los valores de expresiones a medida que un algoritmo se ejecuta (prueba de escritorio) y un "Evaluador de Expresiones" que funciona como una calculadora.

El usuario puede trabajar con expresiones complejas que involucren campos constantes, campos variables, funciones y operadores. Se permite la utilización de arreglos de cualquier dimensión. Existen tres tipos de datos: Real, Cadena de Caracteres y Lógico; además 24 operadores y 19 funciones que operan con datos de estos tipos. Con la utilización de subprogramas que permiten el paso de argumentos de dos formas (por referencia y por valor), el usuario se introduce en un ambiente de programación estructurada y logra así elaborar y estudiar algoritmos que impliquen recursión directa e indirecta ó simples llamadas para modularizar los procesos.

Los errores presentes en la sintaxis o en la evaluación de las expresiones, en la conformación de los subprogramas y llamadas a subprogramas, y en general, en cualquier lugar del diagrama, son detectados y mostrados a través de mensajes, indicando el lugar en el que se presentan. Antes de la ejecución del algoritmo se realiza una revisión completa del diagrama. La ejecución del algoritmo puede ser puesta en pausa (quedando en "Ejecución Paso a Paso") o detenida (quedando en estado de Edición), evitando así la pérdida del control en la ejecución del algoritmo.

El programa Dfd[®] en su totalidad, fue realizado en lenguaje C++ utilizando programación orientada a objetos y corre bajo Windows a 32 bits. El software Dfd[®] incluye un archivo de ayuda para Windows, que provee un fácil acceso a información necesaria para trabajar con Dfd[®], documentación sobre los símbolos u objetos que conforman los diagramas, las funciones, los operadores, los mensajes de error y demás aspectos relacionados con Dfd[®].