

Alternativas para la enseñanza de pseudocódigo y diagrama de flujo

Ana del Prado; Nancy Lamas

Facultad de Tecnología Y Ciencias Aplicadas. Universidad Nacional de Catamarca.
anadelprado@tecno.unca.edu.ar; nancylamas@gmail.com

RESUMEN

En este trabajo, se busca mejorar el proceso de enseñanza aprendizaje de la materia Fundamentos de Informática de la carrera Tecnicatura Industrial de la Facultad de Tecnología y Ciencias Aplicadas, en el que se trata de encontrar nuevas formas en las que los alumnos pueden desarrollar sus habilidades de solución de problemas a través del desarrollo de pseudocódigo y diagramas de flujo.

En la ciencia de la computación y en la programación, los algoritmos son más importantes que los lenguajes de programación o las computadoras. Un lenguaje de programación busca expresar un algoritmo y una computadora es un procesador que permite ejecutarlo. Es por ello que para enseñar a programar es necesario modificar los hábitos de enseñanza de la sintaxis de un lenguaje particular, para enseñar a resolver problemas de una forma sistematizada y enfocada al diseño, mediante el empleo de pseudocódigo o diagramas de flujo. Luego traducir un pseudocódigo o diagrama de flujo a un lenguaje de programación es una tarea sencilla, que no genera mayores problemas.

Se analizan las herramientas de enseñanza aprendizaje Pseint y FreeDfd, basadas en software libre, para la enseñanza de pseudocódigo y diagramas de flujo, determinando cuál de ellas cumple con requisitos de usabilidad y resulta más amigable a los alumnos para la resolución de problemas, comprobación de errores y explicación paso a paso de cada instrucción ejecutada.

Se espera, que el uso de las herramientas mejore el desarrollo de algunas habilidades de programador en los estudiantes, ya que

permite corregir errores y genera prácticas mentales lógicas necesarias, aprendiendo una metodología de la programación apropiada e independiente de algún lenguaje de programación.

Palabras Claves: Enseñanza-aprendizaje, pseudocódigo, diagramas de flujo, usabilidad.

Alternatives for teaching and flow diagram pseudocode

ABSTRACT

In this paper, we seek to improve the teaching-learning process of matter Computer Basics Tecnicatura race of the Faculty of Industrial Technology and Applied Sciences, which is about finding new ways in which students can develop their skills problem solving through the development of pseudocode and flowcharts. In computer science and programming, algorithms are more important than programming languages or computers. A programming language seeks to express an algorithm and a computer is a processor that can run it. That is why to teach programming is necessary to change teaching habits syntax of a particular language, to teach problem-solving in a systematic way and focuses on the design, by using pseudocode or flowcharts. Then translate a flowchart pseudocode or a programming language is a simple task that generates no major problems.

We analyze the teaching-learning tools and FreeDFD Pseint, based on free software, for the teaching of pseudocode and flow charts, determining which one meets usability requirements and is more friendly to the students to solve problems, check errors and step by step explanation of each instruction executed. It is expected that the use of the tools to improve the development of some programming skills in students, allowing correct errors and generates necessary logical mental practices, learning a proper programming methodology independent of any programming language.

Keywords: Teaching and learning, pseudocode, flowcharts, usability.

INTRODUCCION

En la carrera Tecnicatura Industrial de la Facultad de Tecnología y Ciencias Aplicadas de la Universidad Nacional de Catamarca, se dicta la materia Fundamentos de Informática, correspondiente al segundo año. Se pretende que después de haber cursado y acreditado esta materia, el alumno debe tener la capacidad de resolver problemas algorítmicos. Sin embargo, los resultados esperados no son los óptimos y la mayoría de los alumnos, no tienen la habilidad de realizar algoritmos que resuelvan problemas.

Una propuesta para ayudar a resolver esta problemática es implementar el uso de software orientados a la resolución de problemas; que enseñen a los alumnos a seguir un proceso (pensamiento algorítmico) y que contenga actividades de autoevaluación que se imparten a través de la plataforma moodle del aula virtual, provista por la Facultad de Tecnología y Ciencias Aplicadas, dirigida a los alumnos para lograr una autoevaluación de los temas impartidos en la materia.

Con la utilización de estas herramientas se pretende mejorar los resultados de aprendizaje y determinar la herramienta más adecuada para implementarla.

La problemática de la enseñanza de la programación no es nueva y es un tema importante para la comunidad de las ciencias computacionales.

Para Winslow (1996) el aprender a programar requiere de habilidades generales de resolución de problemas. Es un proceso que involucra:

1. Entender el problema.
2. Determinar cómo resolver el problema.
 - a. En alguna forma.
 - b. En alguna forma compatible con la computadora.
3. Traducir la solución en un lenguaje de programación
4. Probar y depurar el programa

Cabe resaltar que en la solución de un problema es importante tener un claro entendimiento de éste. Los programadores expertos piensan en algoritmos no en programas. La traducción de un algoritmo en un programa debe ser una tarea sencilla, no un problema.

Este trabajo tiene el objetivo lograr un aprendizaje mediante un pensamiento algorítmico que lo lleve darse cuenta de que la mejor manera de programar es siguiendo un proceso sistematizado que incluye el análisis del problema, la selección de una posible solución, la utilización de lenguaje natural y diagramas de flujo para el desarrollo de una solución algorítmica.

USO DE HERRAMIENTAS EN EDUCACION

Actualmente existen diversas herramientas de software que sirven como recurso didáctico en la enseñanza-aprendizaje de algoritmos estructurados a nivel superior.

La estructura y funcionalidad de las herramientas propuestas contribuye a que el estudiante adquiera, practique y ejercite la capacidad de resolver problemas de forma metódica a través de soluciones algorítmicas estructuradas.

En este trabajo se realiza una identificación de herramientas que permita realizar una búsqueda sobre las diferentes herramientas al nivel de software libre, sus beneficios o inconvenientes, con el fin de establecer los aportes que pueden proporcionar para el desarrollo de diferentes actividades que permitan favorecer el ambiente de aprendizaje.

Además, se utiliza el aula virtual (bajo la plataforma Moodle) de la Facultad de Tecnología y Ciencias Aplicadas para incorporar las actividades diseñadas, la planeación de los contenidos y los recursos (páginas Web, guías de trabajo, videos, evaluaciones, recursos multimedia), teniendo en cuenta todos los aspectos con los que cuenta para una oportuna distribución de todo el material

de trabajo sobre el aula, además del software seleccionado que permitirá el desarrollo de algunas tareas.

PSeInt

PSeInt es una herramienta para aprender la lógica de programación, orientada a estudiantes sin experiencia en dicha área. Mediante la utilización de un simple y limitado pseudo-lenguaje, intuitivo y en español, permite comenzar a comprender conceptos básicos y fundamentales de un algoritmo computacional.

Este software pretende facilitarle al principiante la tarea de escribir algoritmos en pseudocódigo presentando un conjunto de ayudas y asistencias para brindarle la capacidad de encontrar errores y comprender la lógica del pseudo-lenguaje generando el diagrama de flujo a partir de este.

En su página Web, PSeInt dispone de versiones ejecutables para las plataformas Windows, Linux y Mac OS. Además en la página también es posible descargar diversos manuales, documentación y ejemplos, así como acceder a foros de discusión.

FreeDfd

Anteriormente llamado Smart Dfd (1997), es un Software desarrollado por 3 jóvenes de la Universidad de Magdalena – Colombia, el cual pretende ser base en la enseñanza y práctica simple de algoritmos.

Con FreeDFD se puede dar forma gráfica a cualquier algoritmo o diagrama de flujo, ejecutarlo y corregirlo (depurarlo) en caso de que existan errores. Cuenta con una interfaz bastante intuitiva, disponible en español-inglés-portugués, en donde se agrupan todos los objetos necesarios en el diseño de un diagrama, tales como: Inicio, lectura, ciclos, decisiones, subprogramas, salida, etc. El programa incluye un directorio con diagramas de ejemplo, en donde se puede encontrar algoritmos básicos, de matrices/vectores/funciones/juegos y otros para mayor comprensión.

Su implementación se realizó en el lenguaje C++ y se ejecuta bajo plataformas Windows de 32 bits. La última versión soporta los idiomas: español, inglés y portugués. No requiere de la instalación adicional de librerías y viene con un directorio de ejemplos y ayuda al estilo Win32. Como trabajo futuro pretenden portarlo completamente a entornos Linux.

ANALISIS DE LAS HERRAMIENTAS

Para realizar la comparación de estas herramientas, se toman en cuenta los siguientes criterios:

1. Que la finalidad de la herramienta sea un recurso didáctico en la enseñanza-aprendizaje de algoritmos.
2. Representar mediante diagramas de flujo las soluciones algorítmicas.
3. Que cumpla con los requisitos de usabilidad (es decir que los alumnos puedan hacer sus tareas de forma rápida y fácil)
4. Que la herramienta se amigable a los alumnos para la resolución de problemas
5. Que se realicen comprobación de errores.
6. Que se explique paso a paso cada instrucción ejecutada.
7. Ejecución multiplataforma (Linux, Windows y MacOS).
8. Facilidad de instalación en distintas plataformas.
9. Página Web dedicada específicamente al software para proveer de nuevas versiones, documentación y ejemplos.
10. Blog de ayuda para el usuario.

En cada una de las herramientas a analizar se resolverá un problema sencillo que es: “Obtener los primeros 100 números pares” a fin de poder comparar las que si cumplen con todos los criterios definidos.

Tabla 1: Comparación de criterios de las herramientas Pseint y FreeDfd

Criterios	PSEINT	FreeDFD
1	x	x
2	x	x
3	x	
4	x	
5	x	x
6	x	x
7	x	
8	x	
9	x	
10	x	

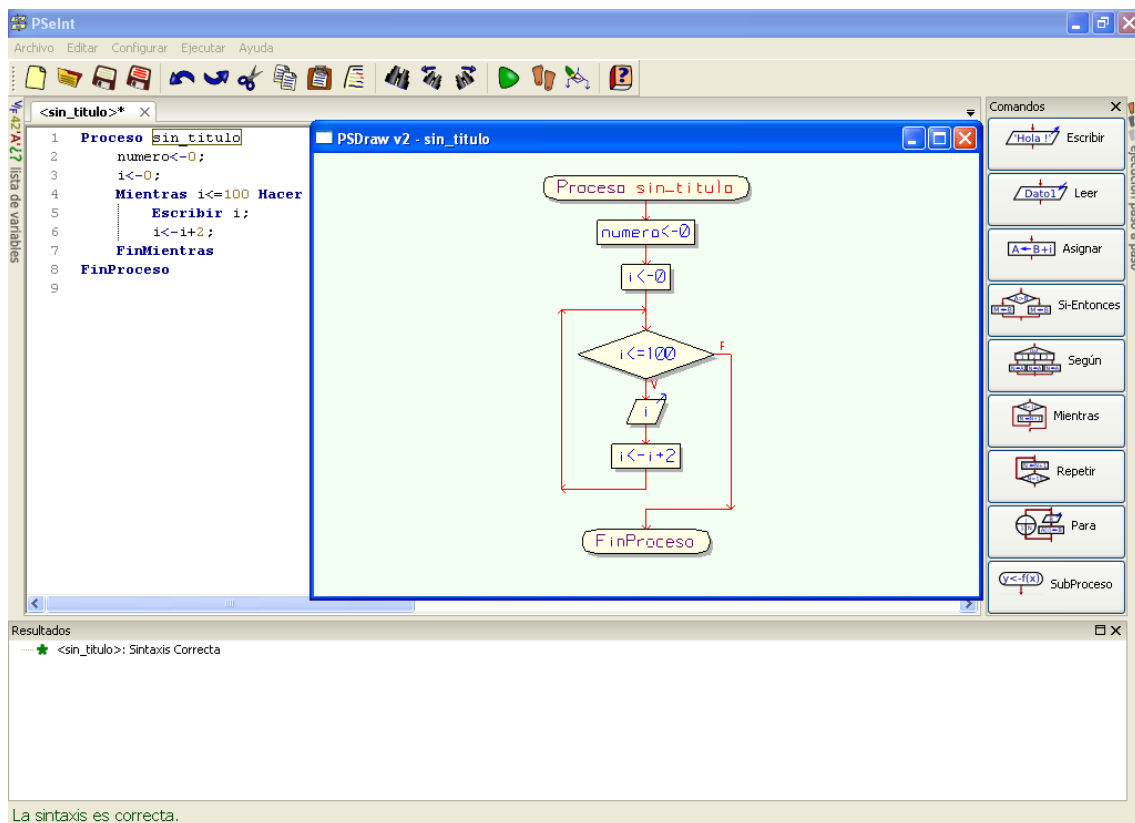


Gráfico 1: Diagrama de Flujo realizado con Pseint

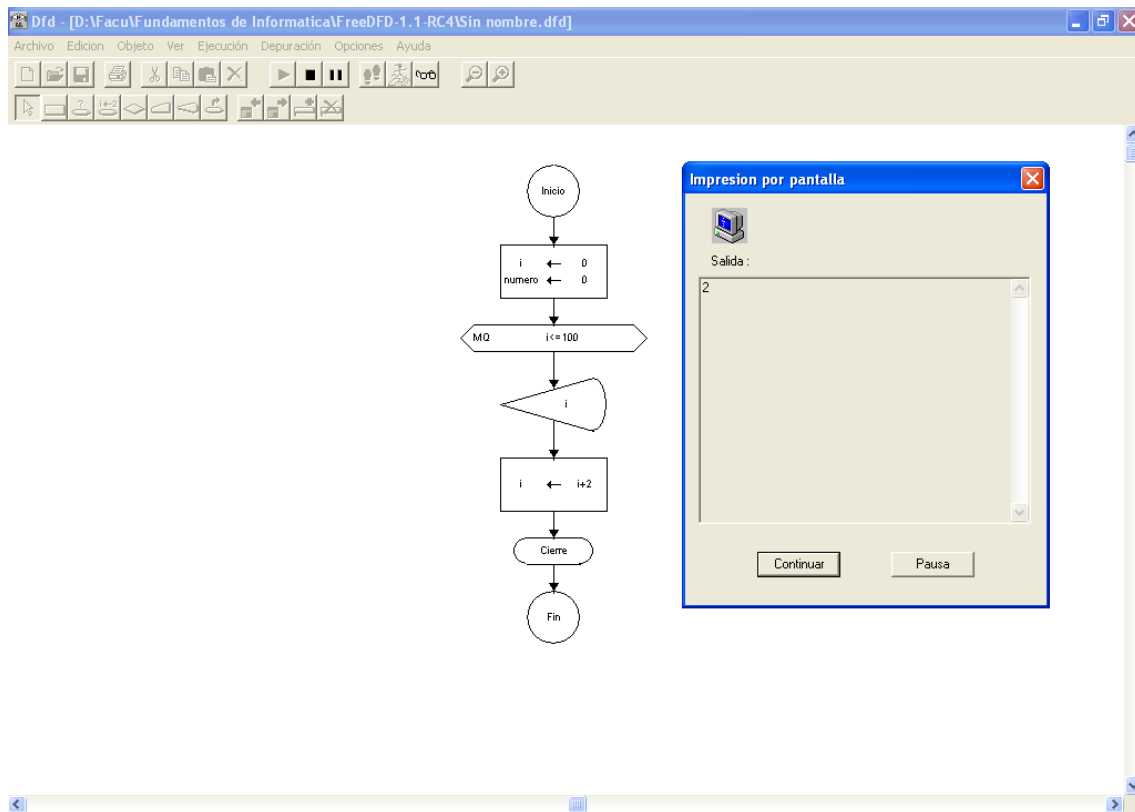


Grafico 2: Diagrama de Flujo realizado con FreeDfd

PSelnt

La interfaz gráfica de PSelnt permite la edición del pseudocódigo con resaltado tipográfico, autocompletado, ayudas emergentes, plantillas de comandos e indentado inteligente para facilitar su empleo y aprendizaje. Además el lenguaje del pseudocódigo es configurable.

- Al usar algoritmos los alumnos muchas veces tenían problemas para entender el pseudocódigo y escribían instrucciones que no eran correctas ya sea porque no existían, porque abrían una instrucción y no la cerraban. Al utilizar la herramienta, los alumnos no pueden utilizar instrucciones incorrectas ya que el software no lo permite.
- Los alumnos pueden verificar si su diagrama de flujo o pseudocódigo funciona o no antes de entregarlo, por lo tanto entregan mejores tareas al tener la oportunidad de corregir sus errores; y además su grado de

satisfacción con el curso se incrementa, porque ahora pueden ver claramente la razón por la que obtienen una cierta calificación.

- Los alumnos están más motivados a utilizar la herramienta porque pueden ver lo que hace su “programa” y en general prefieren trabajar en la computadora que hacerlo a mano.
- Es más fácil revisar los problemas ya que se puede ejecutar el diagrama y verificar si hace lo que debe hacer.

La herramienta PSeInt tiene los siguientes inconvenientes:

- Aun cuando permite editar comentarios inmersos en el pseudocódigo, se carece de soporte para el análisis del problema y planteamiento de la solución.
- El diagrama de flujo generado no concuerda del todo con la notación estándar de la mayoría de los libros de algoritmos en lo que respecta a entrada, salida y el ciclo para.
- Al tratarse de una herramienta que emplea principalmente el pseudocódigo, aun cuando se tiene autocompletado, se incrementa la posibilidad de errores de sintaxis y construcción de las estructuras secuenciales y de control para la especificación de algoritmos, lo cual es un problema común en este tipo de herramientas.

FreeDFD

Con esta herramienta el usuario puede trabajar con expresiones complejas que involucren constantes, variables, funciones y operadores. Los tipos de datos que maneja son: reales, cadenas de caracteres y lógicos; también se permite el uso de arreglos.

Durante la ejecución de un diagrama se detectan errores de sintaxis y de conformación de subprogramas.

La herramienta FreeDFD presenta los siguientes inconvenientes:

- Carece de soporte para el análisis del problema y planteamiento de la solución, inclusive no cuenta con algún elemento para editar el enunciado del problema.
- La prueba del diagrama basa su ejecución en dos cuadros de diálogo, uno presenta las salidas a pantalla y el otro solicita las entradas al usuario sin mostrar la traza completa del algoritmo que incluya la evaluación de las expresiones lógicas inmersas en las condiciones de selección o cíclicas.
- No utiliza la notación estándar de la mayoría de los libros de algoritmos para representar las estructuras cíclicas para y mientras.
- El tamaño de los elementos del diagrama no se ajusta a la longitud del texto que contienen y tampoco acepta la edición de múltiples líneas de texto
- No cuenta con la estructura cíclica hacer hasta.
- No traduce el diagrama a algún lenguaje de programación estructurado.
- Solo se ejecuta de forma completa en plataformas Windows, para las plataformas Linux debe instalarse el Wine.

Considerando las características que posee cada herramienta y sus desventajas, se considera más adecuada la implementación del software PSEINT para enseñanza aprendizaje de la materia Fundamentos de Informática de la carrera Tecnicatura Industrial de la facultad de Tecnología y Ciencias Aplicadas.

CONCLUSIONES

En la enseñanza de la programación siempre ha existido la tentación de enseñar el lenguaje lo antes posible, sacrificando el desarrollo de la lógica, y se cae en enseñar a operar lenguajes y no a programar; porque programar implica primero diseñar lógicamente la solución y después, codificar usando un lenguaje.

En consecuencia, se están formando muchos programadores buenos para codificar usando lenguajes, pero sin bases lógicas sólidas.

El estudiante primero debe desarrollar las habilidades mentales lógicas necesarias, aprendiendo una metodología de la programación apropiada, porque la programación es lógica y debe ser independiente de algún lenguaje de programación, y después, sobre esas bases aprender el lenguaje.

El software para la enseñanza-aprendizaje de algoritmos estructurados es un recurso didáctico muy valioso que debe ser considerado por profesores y alumnos relacionados con cursos introductorios de algoritmos y programación a nivel superior.

La herramienta seleccionada PSEINT contribuye a que el estudiante adquiera, practique y ejercite la capacidad de resolver problemas de forma metódica aventajando a herramientas similares desde la perspectiva de los recursos didácticos de los cuales dispone.

Es importante mencionar que lo que hace la diferencia no es el uso de los diagramas de flujo en sí mismos, sino todo el ambiente que permite primero hacer el diagrama sin preocuparse de la sintaxis ni de tener un ambiente de desarrollo real y después de poder ejecutar el diagrama paso a paso y ver que valores van tomando las variables.

Los resultados de la integración de las herramientas como el uso del aula virtual, como apoyo a la presencialidad, junto con la sustentación pedagógica y didáctica soportada en el uso de las TICs, han permitido identificar y orientar el papel del docente frente a los procesos de aprendizaje, logrando que los estudiantes construyan su conocimiento de forma cooperativa.

REFERENCIAS

- Cárdenas F., Castillo N., Daza E., Editor e intérprete de algoritmos representados en diagramas de flujo. Informática educativa, Vol. 11, No. 1, pp. 101 106. Disponible en:
<http://www.colombiaaprende.edu.co/html/estudiantesuperior/1608/article109522.html> (1998)
- Dumas, J., Redish, J. A practical guide to usability testing Intellect Books. (1994)
- FreeDFD: editor de algoritmos representados en diagramas de flujo, (2008). Disponible en: <http://code.google.com/p/freedfd/>, consultado el 02 de Mayo de 2013
- Levine, G. Introducción a la computación y programación estructurada. McGraw-Hill, (1989)
- Pérez Peregrino, Luis, Metallana Adriana, Rodríguez Pérez Frey. Proyecto TEACH-Me: La experiencia construida en Uniminuto Inventum Revista No. 6 Facultad de Ingeniería Uniminuto - ISSN 1909 – 2520 (2009)
- PSEINT: Herramienta para aprender la lógica de programación. Disponible en: <http://pseint.sourceforge.net/>, consultado el 02 de Mayo de 2013
- Winslow, L. Programming Pedagogy. A psychological Overview SIGCSE BULLETIN 17 — 25 (1996)