

Desarrollo de aplicación didáctica para el desarrollo de pensamiento algorítmico

Development of didactic application for the development of algorithmic thinking

DURAN-BELMAN, Israel†, MAGDALENO-ZAVALA, Juan Antonio y GALLARDO-ALVAREZ, Dennise Ivonne

Instituto Tecnológico Superior de Irapuato, Carretera Irapuato-Silao, Km. 12.5, Col. El Copal, C.P. 36821, Irapuato, Gto.

ID 1^{er} Autor: *Israel, Duran-Belman* / **ORC ID:** 0000-0002-1394-0486, **Researcher ID:** V-7806-2019, **CVU CONACYT ID:** 691483

ID 1^{er} Coautor: *Juan Antonio, Magdaleno-Zavala* / **ORC ID:** 0000-0001-9762-0086, **Researcher ID:** S-5543-2018, **CVU CONACYT ID:** 691690

ID 2^{do} Coautor: *Dennise Ivonne, Gallardo-Alvarez* / **ORC ID:** 0000-0002-9197-6425, **Researcher ID:** S-4921-2018, **CVU CONACYT ID:** 264516

DOI: 10.35429/JPD.2019.7.3.1.6

Recibido 11 de Enero, 2019; Aceptado 25 de Marzo, 2019

Resumen

Los programas de computador tienen como finalidad resolver problemas específicos usando un conjunto de instrucciones compuestas por operaciones, decisiones lógicas y ciclos repetitivos. El proceso enseñanza-aprendizaje utiliza en las primeras etapas los Diagramas de Flujo que permite al estudiante representar gráficamente la secuencia de un conjunto finito de instrucciones pero que resulta insuficiente para aprender a programar debido a la poca experiencia y conocimientos de los estudiantes en el área de programación, que por consiguiente aumenta los índices de reprobación o deserción durante los primeros semestres de su formación universitaria en la carrera de ingeniería en sistemas computacionales, por lo que se requiere desarrollar recursos adicionales para que los alumnos logren generar de manera gradual el pensamiento algorítmico. La metodología seguida para el desarrollo del proyecto está basada en el ciclo de vida en cascada con subproyectos; Finalmente con un grupo de 35 estudiantes, al inicio 24 (68%) de ellos presentaron dificultades para analizar y programar, utilizando el software didáctico como apoyo en clases se redujo a 8 (22%).

Pensamiento Algorítmico, Diagramas de Flujo, Software didáctico

Abstract

The purpose of the computer programs is to solve specific problems using a set of instructions composed of operations, logical decisions and repetitive cycles. The teaching-learning process uses in the early stages the Flow Diagrams that allow the student to graphically represent the sequence of a finite set of instructions but that is insufficient to learn to program due to the lack of experience and knowledge of the students in the area of programming, which consequently increases the rates of failure or attrition during the first semesters of their university education in the engineering career in computer systems, so it is necessary to develop additional resources for students to generate algorithmic thinking gradually. The methodology followed for the development of the project is based on the cascade life cycle with subprojects; Finally, with a group of 35 students, at the beginning 24 (68%) of them presented difficulties to analyze and program, using the didactic software as support in classes was reduced to 8 (22%).

Algorithmic thinking, Flow diagrams, Didactic software

Citación: DURAN-BELMAN, Israel, MAGDALENO-ZAVALA, Juan Antonio y GALLARDO-ALVAREZ, Dennise Ivonne. Desarrollo de aplicación didáctica para el desarrollo de pensamiento algorítmico. Revista de Didáctica Práctica. 2019. 3-7: 1-6.

† Investigador contribuyendo como primer autor.

Introducción

La programación es un conocimiento que se aprende a través de la práctica siempre y cuando se sigan ciertas reglas y procedimientos empezando desde lo más básico e ir aumentando la complejidad de manera gradual, hasta que el alumno se capaz de escribir un programa que resuelva una necesidad o problema del entorno real.

En el ámbito de la computación, los Algoritmos son una herramienta que permite describir claramente un conjunto finito de instrucciones, ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible; con frecuencia cuando se habla de este concepto de describen tres tipos de pensamiento que generalmente se relacionan entre ellos, los cuales destacan al Pensamiento Computacional, Pensamiento Algorítmico y Pensamiento Procedimental.

Por lo tanto, es importante puntualizar a qué se refiere cada uno de estos pensamientos. Según Moursund (2006), el pensamiento computacional hace referencia a la representación y solución de problemas utilizando inteligencia humana, de máquinas o de otras formas que ayuden a resolver el problema; el pensamiento algorítmico se refiere al desarrollo y uso de algoritmos que puedan ayudar a resolver un tipo o tarea específica; por su parte, el pensamiento procedimental se ocupa del desarrollo y utilización de procedimientos diseñados para resolver un problema, pero que no necesariamente, siempre resulta exitoso.

En el Instituto Tecnológico Superior de Irapuato (ITESI) cuenta diversas carreras, una de ellas es la de Ingeniería en Sistemas Computacionales el cual tiene como objetivo generar en los estudiantes la capacidad para analizar situaciones reales y plantear soluciones a los problemas detectados en el desarrollo del procesamiento de datos en una organización, motivo por el cual ofrece en el transcurso de la carrera diversas materias enfocadas en el área de programación; una de las dificultades que encuentran los maestros en las materias de los primeros tres semestres es el de desarrollar el pensamiento algorítmico en los estudiantes y que estos comprendan la lógica de la programación, por tanto se les enseña partiendo de los diagramas de flujo los cuales se les considera como una representación gráfica (a través de símbolos) de una secuencia de pasos que se realizan para obtener un cierto resultado, esto puede ser un producto, un servicio, o bien una combinación de ambos, ya que según el autor Osvaldo Cairo del Libro Metodología de la programación, es una de las herramientas básicas necesarias para aprender a programar y posteriormente iniciar en cualquier lenguaje de programación.

Debido a las circunstancias mencionadas anteriormente se detectó mediante una evaluación diagnóstica que un grupo de alumnos de la carrera ingeniería en sistemas computacionales de la materia de fundamentos de programación tiene problemas para desarrollar el pensamiento algorítmico ya que se les dificulta traducir a lenguaje de alto nivel de programación partiendo de un diagrama de flujo, por consiguiente se determinó desarrollar un software didáctico que permitiera a los alumnos de manera autodidacta arrojando la siguiente información el 78% aprende a través de imágenes, organizadores gráficos, diagramas, mapas y símbolos; por lo que si en el proceso se utilizan solo libros de texto propicia un entorno complejo en el proceso enseñanza-aprendizaje y por ende no se desarrollarán las habilidades necesarias para un profesional en el desarrollo de software.

Descripción del Método

Software

Según la definición del IEEE (Institute of Electrical and Electronics Engineers), citada por [Lewis 1994] "software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según el mismo autor, "un producto de software es un producto diseñado para un usuario". El software, como programa, consiste en un código en un lenguaje máquina específico para un procesador individual. El código es una secuencia de instrucciones ordenadas que cambian el estado del hardware de una computadora.

Lenguaje de programación.

Un lenguaje de programación es el medio a través del cual le comunicamos a la computadora la secuencia de instrucciones que debe ejecutar para llevar a cabo actividades, tareas o solución de problema. Todo lenguaje permite el manejo de los tres elementos que componen un programa; a saber: estructuras de datos, operaciones primitivas elementales y estructuras de control. Recordemos que mediante un programa podemos representar en forma de datos cualquier situación de nuestra realidad, a los datos se le da entrada a la computadora mediante dispositivos de entrada como teclado, lectores ópticos de caracteres, ratón, etc.; una vez que los datos están en la computadora, se les procesan para convertirlos en información, la cual será emitida hacia el exterior de la computadora mediante dispositivos de salida como son la pantalla, impresora, etcétera.

Diagramas de flujo.

Los diagramas de flujo son una manera de representar visualmente el flujo de datos a través de sistemas de tratamiento de información. Los diagramas de flujo describen que operaciones y en que secuencia se requieren para solucionar un problema dado. Un diagrama de flujo u organigrama es una representación diagramática que ilustra la secuencia de las operaciones que se realizarán para conseguir la solución de un problema. Los diagramas de flujo se dibujan generalmente antes de comenzar a programar el código frente a la computadora.

Los diagramas de flujo facilitan la comunicación entre los programadores y la gente del negocio. Estos diagramas de flujo desempeñan un papel vital en la programación de un problema y facilitan la comprensión de problemas complicados y sobre todo muy largos. Una vez que se dibuja el diagrama de flujo, llega a ser fácil escribir el programa en cualquier idioma de alto nivel. Vemos a menudo cómo los diagramas de flujo nos dan ventaja al momento de explicar el programa a otros. Por lo tanto, está correcto decir que un diagrama de flujo es una necesidad para la documentación mejor de un programa complejo.

Algoritmo.

En el ámbito de la computación, los Algoritmos son una herramienta que permite describir claramente un conjunto finito de instrucciones, ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible. Vale la pena recordar que un programa de computador consiste de una serie de instrucciones muy precisas y escritas en un lenguaje de programación que el computador entiende (Java, C++, etcétera).

Análisis

Para el presente proyecto se realizó debido a que se detectó un problema de aprendizaje y comprensión en la resolución de problemas a través de algoritmos, se fijó el objetivo de desarrollar una herramienta de software didáctico que ayudará al alumno. Por lo que la herramienta creada en este proyecto tiene la ventaja de facilitar el aprendizaje del alumno en el área de programación. Para la elaboración del proyecto se siguió la metodología de un ciclo de vida en cascada en el cual se trabajó en seis fases:

- Análisis del sistema: en esta fase se levantaron los requerimientos del sistema, se especifican tanto requerimientos funcionales, como los no funcionales.
- Diseño del sistema: se diseñaron los algoritmos necesarios para la elaboración del área específica al igual que la forma en cómo se visualizaría al usuario.

- Codificación: se realizaron líneas de codificación en el lenguaje de programación Visual Studio C# 2008, de acuerdo a la función especificada para cada área.
- Pruebas: Sé probó el sistema ya integrado, que las interfaces cumplan con el entorno. Algunas de las pruebas fueron funcionales, de comunicación, de rendimiento y de volumen.
- Integración se realizaron casos de prueba de caja negra con el objetivo de descubrir errores que se pueden producir en la interacción entre los módulos.
- Implementación: Se realizó el instalable para que los estudiantes lo utilizarán en sus computadoras personales.

Los principales símbolos estandarizados por ISO y ANSI para la elaboración de diagramas de flujo:

- Herramienta Terminador (Inicio/Fin).
- Herramienta Entrada General.
- Herramienta Acción/Proceso General.
- Herramienta Decisión.
- Herramienta Impresión.
- Herramienta Flujo.

Para la parte de análisis y diseño del sistema se centró en la realización de diagramas de clases y de secuencia ya que permiten representar gráficamente la estructura del sistema a desarrollar con sus interacciones tal y como se muestra a continuación:

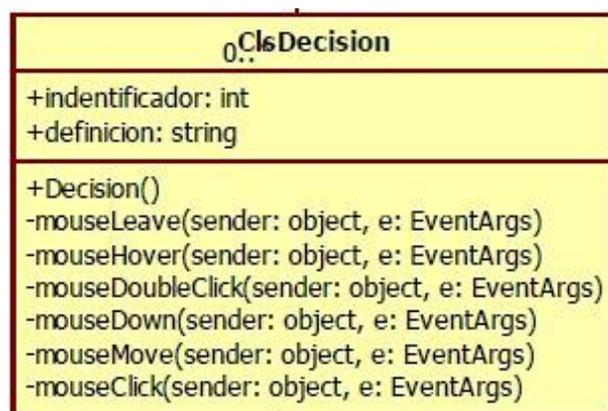


Figura 1 Diagrama de Clase llamada ClsDecision

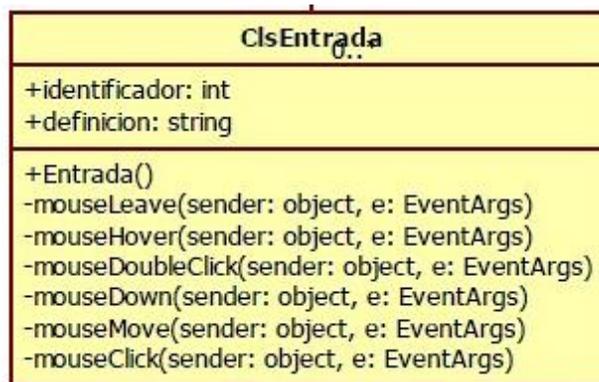


Figura 2 Diagrama de Clase llamada ClsEntrada

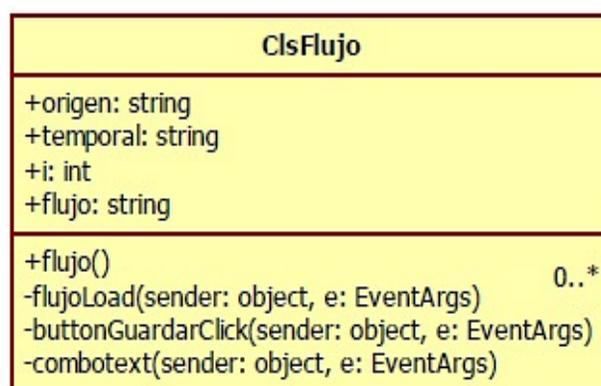


Figura 3 Diagrama de Clase llamada ClsFlujo

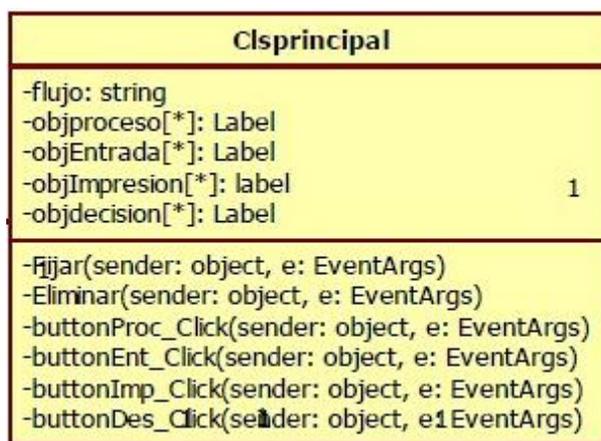


Figura 4 Diagrama de Clase llamada ClsPrincipal

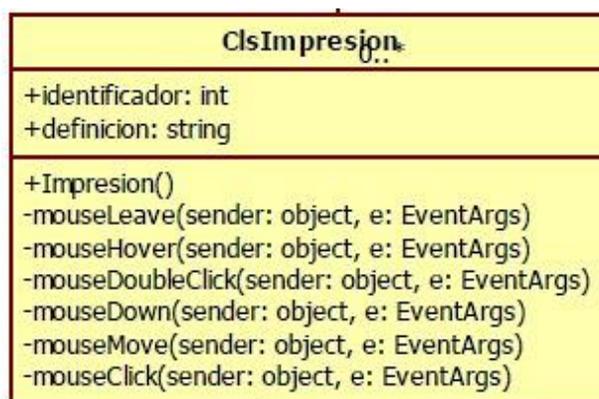


Figura 5 Diagrama de Clase llamada ClsImpresion



Figura 6 Diagrama de Clase llamada ClsProceso

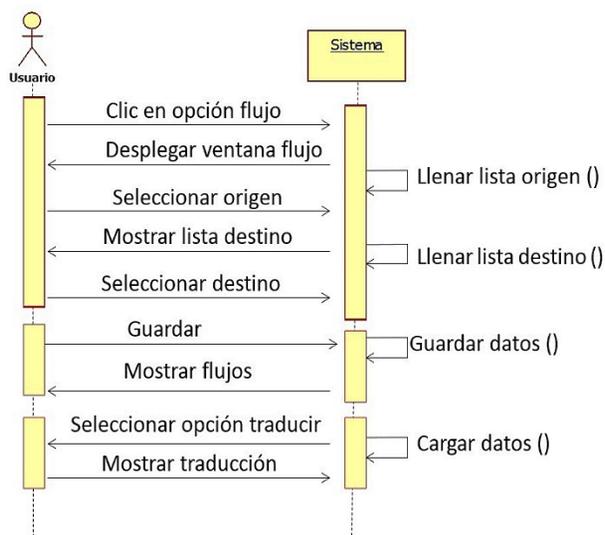


Figura 7 Diagrama de Secuencia

Una vez que se definió las características y eventos por los cuales el sistema va interactuar se diseñó la Interfaz gráfica de usuario (GUI) para facilitar la comunicación del usuario con la computadora a través de la utilización de un conjunto de imágenes, texto y objetos pictóricos. Se tomó en consideración que la operación de este sistema se enfocará en el uso del mismo para el auto aprendizaje en el apoyo de la transformación de un diagrama de flujo a lenguaje de programación, esto significa que el operador del proyecto deberá tener ciertos conocimientos básicos como son: manejo básico de una computadora, conocimientos acerca de programación orientada a objetos y programación estructurada, y conocimientos acerca del diseño de un diagrama de flujo. El sistema se divide en un área de herramientas en las cuales aparecen los símbolos que se utilizan para realizar los diagramas de flujo, en la parte central se encuentra el área de dibujo que sirve para construir dichos diagramas y finalmente el área de traducción que ayuda a codificar partiendo del dibujo en pseudocódigo.



Figura 8 Interfaz del Sistema

Resultados

De acuerdo a los primeros datos del grupo A que cuenta con 35 alumnos de Fundamentos de programación de la carrera de ingeniería en sistemas computacionales se detectó que 24 (68%) de estudiantes presentan dificultades para analizar y programar sistemas de información, utilizando la herramienta como apoyo en clases se redujo a 8 (22%).



Gráfico 1 Porcentaje que representa dificultad para programar sin apoyo de herramienta didáctica



Gráfico 2 Porcentaje que representa dificultad para programar con apoyo de herramienta didáctica

Además, se identificó que el 54% de los alumnos carecen de razonamiento lógico y una vez utilizada la herramienta disminuyó al 26%.

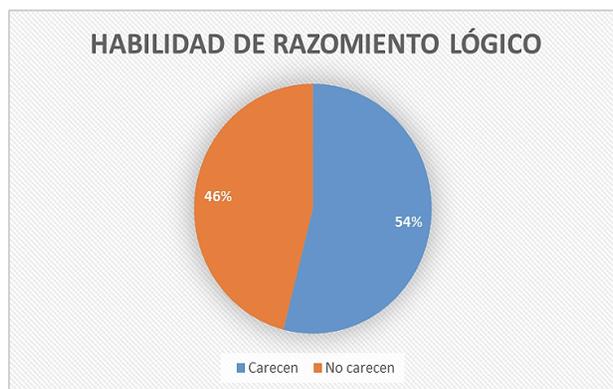


Gráfico 3 Porcentaje que representa habilidad de razonamiento lógico sin apoyo de herramienta didáctica



Gráfico 4 Porcentaje que representa habilidad de razonamiento lógico con apoyo de herramienta didáctica

Después de la implementación del software didáctico durante el transcurso de un semestre en la materia Fundamentos de programación de la carrera de ingeniería en sistemas computacionales se obtuvo un mayor índice de aprovechamiento por lo cual se disminuyó el índice de reprobación y deserción por lo cual la mayormente de estudiantes continuaron su carrera sin adeudo. Institucionalmente se creó una herramienta informática que apoya al docente como complemento en el proceso enseñanza-aprendizaje, logrando que los estudiantes no se rezaguen en los temas y puedan de manera autodidacta aprender por su cuenta.

Conclusiones

Se desarrolló una herramienta útil para entender la relación que existe entre el diagrama de flujo y un lenguaje de programación. Por lo que la herramienta creada en este proyecto tiene la ventaja de facilitar el aprendizaje del alumno en el área de programación. Para los alumnos que estén cursando la materia de fundamentos de programación se les recomienda que consideren la herramienta de software didáctica como un método que ayuda a la comprensión en el área de razonamiento lógico.

Referencias

Basogain Olabe, X., Olabe Bosagain, M., & Olabe Basogain, J. C. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. RED-Revista de Educación a Distancia, 1-33.

López García, J. C. (2009). Algoritmos y Programación. Colombia: Fundación Gabriel Piedrahita Uribe.

LÓPEZ ROMÁN, L. (2006). Metodología de la Programación Orientada a Objetos. México: Alfaomega.

Valverde Berrocoso, J., Fernández Sánchez, M., & Garrido Arroyo, M. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. RED-Revista de Educación a Distancia, 1-18.

Vidal, C. L., Cabezas, C., Parra, J. H., & López, L. P. (2015). Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para. Centro de Información Tecnológica, 23-32.