

Herramienta para la enseñanza de estructuras de control en programación

*Estela Lizbeth Muñoz Andrade¹
Carlos Argelio Arévalo Mercado²*

Introducción

Las instituciones de educación superior están afrontando nuevos desafíos con la transformación digital. En la actualidad la aplicación de la tecnología en la educación no es sólo una opción, sino una necesidad (UNESCO, 2013). El uso de tecnología permite reforzar los contenidos de enseñanza y generar nuevas aptitudes y habilidades para la construcción de conocimiento,

1 Departamento de Sistemas Electrónicos, Centro de Ciencias Básicas, Universidad Autónoma de Aguascalientes, Ave. Universidad 940, Ciudad Universitaria, C.P. 20100, Aguascalientes, Aguascalientes, México. lizbeth.munoz@edu.uaa.mx, ORCID <https://orcid.org/0000-0003-4182-5044>

2 Departamento de Sistemas de Información, Centro de Ciencias Básicas, Universidad Autónoma de Aguascalientes, Ave. Universidad 940, Ciudad Universitaria, C.P. 20100, Aguascalientes, Aguascalientes, México. carlos.arevalo@edu.uaa.mx, ORCID <https://orcid.org/0000-0002-8349-7985>. Autor para correspondencia

logrando que el aprendizaje sea interactivo y dinámico (Sánchez-Otero, García-Guilianny, Steffens-Sanabria y Hernández-Palma, 2019).

Así, la brecha entre las necesidades de la industria y la educación para el desarrollo de software busca que los proyectos de la vida real y otras habilidades comerciales sean elementos que se incluyan en los programas de estudio, con el fin de lograr que la educación para el desarrollo de software sea accesible para una amplia gama de estudiantes (Liebenberg, Huisman y Mentz, 2014). Por tal motivo, los profesores realizan actividades de aprendizaje distintas a las tradicionales, donde la innovación en el diseño de las estrategias de enseñanza es parte fundamental, haciendo su trabajo más atractivo y eficiente, y con ello los estudiantes refuerzan los conocimientos sin importar el área de estudio y avanzan a su propio ritmo de aprendizaje; todo lo anterior con base en aplicaciones diseñadas para la enseñanza, conocidas como software educativo (Arroyo, 2006; Barragán, Aguilar, Cerpa y Núñez, 2009; Valencia y Caicedo, 2017).

Con respecto al software educativo, Vidal (2010) lo define como una forma genérica en la que las aplicaciones o programas computacionales facilitan el proceso de enseñanza-aprendizaje, mientras que Marqués (2005) señala que son aquellos programas creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje. Por otra parte, Arroyo (2006) menciona que se debe involucrar a la psicología y ciencias del aprendizaje en el diseño de los elementos teóricos, didácticos, pedagógicos y metodológicos que deben contener las aplicaciones y así contribuir al aprendizaje significativo.

En ese sentido, la tecnología ha permitido reforzar los contenidos de enseñanza, generando nuevas habilidades para la construcción de conocimiento (Cavazos y Torres, 2016) mediante el diseño de herramientas que permitan reforzar habilidades en conceptos de mayor complejidad, como son los temas de algoritmos y programación (Sánchez-Otero *et al.*, 2019), los cuales representan uno de los niveles de mayor dificultad en la preparación de estudiantes de ingeniería (Fuentes-Rosado y Moo-Medina, 2017).

Por otra parte, la enseñanza de la programación requiere del desarrollo del pensamiento lógico para el análisis, comprensión y solución de problemas computacionales. El desarrollo del pensamiento lógico y algorítmico está mediado por el uso de lenguajes de programación cuyo aprendizaje requiere un dominio de la sintaxis y semántica (Salgado, Alonso, Gorina y Tardo, 2013).

Partiendo de lo anterior, se ha identificado que una de las grandes dificultades en la comprensión de nociones abstractas se mantiene a lo largo de los cursos de programación (Jaramillo y Puga, 2016), por lo que uno de los grandes retos es lograr que las aplicaciones educativas sean capaces de ajustarse a las necesidades de los estudiantes (Vidal, Gómez y Ruiz, 2010).

Así, en el trabajo de Zavaleta, Pérez y Cocón (2016) se presenta un análisis de distintas aplicaciones de software diseñadas para la enseñanza de la programación, cada uno de los trabajos muestra una herramienta de enseñanza general, entre ellas se encuentran los mapas conceptuales, tutoriales gráficos, aplicaciones multimedia, manejo de plantillas, diseño de algoritmos, entre otras. Varias de las herramientas son orientadas a niños y jóvenes, y otras más al manejo de pseudocódigo, pero ninguna orientada a un lenguaje de programación en específico. Se debe tomar en cuenta que las estructuras de control conocidas como ciclos son elementos básicos de un lenguaje de programación, y mediante este tipo de instrucciones es posible ejecutar un conjunto de sentencias un número determinado de veces. Los ciclos deben ser contruidos de manera tal que se tenga control sobre la cantidad de veces que se va a realizar el conjunto de instrucciones, evitando así una ejecución infinita (Cairo, 2005).

Partiendo de lo anterior, el objetivo principal de este trabajo consistió en probar la herramienta de software educativo que fue diseñada para mejorar el aprendizaje en las estructuras cíclicas de programación, y así evaluar si los estudiantes comprenden de forma intuitiva su uso.

La herramienta diseñada para el estudio tiene la característica de enfocarse específicamente en el manejo de las estructuras cíclicas *for*, *while* y *do-while*, las cuales representan conceptos de difícil visualización y, por lo tanto, generan complejidad para crear modelos mentales de las mismas (Lahtinen, Ala-Mutka y Järvinen, 2005). El software diseñado específicamente para el lenguaje de programación ANSI C cuenta con la ilustración del funcionamiento de cada ciclo, aplicando animación e interacción, y a su vez permite el seguimiento del flujo de instrucciones; elementos que no presentan otras propuestas.

Cabe mencionar que la aplicación busca ser una herramienta incluyente, de uso tanto para estudiantes y profesores, con la intención de hacer partícipes a todos los agentes en el proceso de enseñanza-aprendizaje.

Desarrollo del trabajo

Para comprobar si la herramienta de software educativo permite que los estudiantes comprendan de forma intuitiva su uso, se aplicó una prueba USE (Usefulness, Satisfaction and Ease of Use) para medir la satisfacción, usabilidad y utilidad, así como el aprendizaje de uso (Lund, 2001; Serrano-Angulo y Cebrián-Robles, 2014).

Los datos generados por este estudio exploratorio se obtuvieron a través de la participación de estudiantes del último grado escolar del programa educativo de Ingeniería en Sistemas Computacionales de la Universidad Autónoma de Aguascalientes, así como por profesores del área de programación. El estudio se desarrolló durante el periodo escolar agosto-diciembre del 2019. Se hizo una invitación abierta para participar a un grupo de 45 estudiantes, de los que se obtuvo la respuesta de 35 personas. Se estableció como prerrequisito haber cursado la materia de Ingeniería de Software. Como parte de la muestra se contó con la participación de cinco profesores con conocimientos en desarrollo de aplicaciones de software. Los profesores no corresponden con los profesores de los estudiantes. La prueba fue aplicada por un profesor distinto a los investigadores del proyecto, con el fin de evitar la subjetividad y permitir que los participantes se expresaran libremente (Rubin y Chisnell, 2008).

Para llevar a cabo la validación, se solicitó a los participantes hacer uso del software educativo (Figura 1) durante 30 minutos, para después contestar un cuestionario y anotar sus observaciones acerca de los ítems evaluados. La prueba constó de cinco preguntas para recabar información general, más 30 ítems en una escala Likert de siete puntos, desde muy fuertemente de acuerdo con la máxima puntuación, a muy fuertemente desacuerdo con la puntuación más baja. El acceso al cuestionario fue anónimo y en línea. El prerrequisito de haber cursado Ingeniería de Software y que los profesores impartan materias de desarrollo de software genera conocimiento previo de manejo de herramientas similares, por tal motivo se incluyó una pregunta para evaluar el uso de herramientas similares a la propuesta de este estudio.

Figura 1. Sistema para la enseñanza de estructuras de control.



Fuente: Elaboración propia.

Una vez terminado el tiempo de uso de la herramienta, los participantes contestaron la encuesta en línea, la cual permitió conocer su percepción del sistema. De los 40 participantes que representan 100% de la muestra, sólo 82.5% contestó de forma completa la encuesta, mientras que 17.5% dejó algunos ítems sin contestar. De tal forma que se trabajó con los datos de los 33 participantes que sí concluyeron la evaluación. El 15.15% era profesores y 84.8% estudiantes. Dado que existía un prerrequisito para formar la muestra, se solicitó contestar una pregunta de conocimiento de herramientas similares: 90.9% menciona no haber utilizado alguna herramienta similar o igual. El 36.3% de los participantes son mujeres y 63.6% hombres. Con respecto a las edades de los participantes, 75.5% se encuentra en el rango de 18 y 22 años, mientras que el resto, 24.24%, es mayor a 22 años. Como ya se mencionó, la evaluación de usabilidad se realizó aplicando la encuesta USE (Lund, 2001), la Tabla 1 muestra los ítems seleccionados para el estudio:

Tabla 1. Ítems para evaluar la usabilidad del sistema educativo.

Elemento	Variable
Usabilidad	Utilidad
	Efectividad
Facilidad de Uso	Facilidad
	Rapidez
Facilidad de Aprendizaje	Aprendizaje
Satisfacción	Satisfacción

Fuente:

La facilidad de aprendizaje del sistema, así como su flexibilidad, consistencia y facilidad para la corrección de errores, fue evaluada con 33.3%, seguido de la facilidad de uso y satisfacción que obtuvieron el mismo puntaje, 21.2%. La efectividad y utilidad fueron aspectos con una puntuación media de 12.1% y 9% respectivamente, por lo que se considera que son elementos que pueden tener mejoras. Por último, la rapidez es el ítem con menor valoración: 3.03%, considerando que el sistema no arroja los resultados de forma inmediata, lo que lo convierte en un elemento a revisar y mejorar en el sistema.

Conclusiones

La falta de evaluación del software genera herramientas de baja confianza que no aseguran el logro de los objetivos, por lo que evaluar la usabilidad de una herramienta diseñada específicamente para atender un tema y dirigirla a un grupo de estudiantes con necesidades específicas, es un paso ineludible en el proceso de diseño de software y de herramientas para la mejora del aprendizaje.

Los resultados obtenidos en el estudio muestran que la herramienta es del gusto de los usuarios en general, que es fácil de aprender y entender, así como de usar, sin embargo, aún existen detalles a corregir como la rapidez de respuesta.

Este tipo de evaluación proporciona aspectos de calidad que permiten crear productos usables y cuyo diseño está mejor adaptado y orientado a profesores y estudiantes del área de programación. Además, permite realizar mejoras en el diseño del sistema, identificar posibles problemas de ejecución, ahorro de tiempo y costos, así como conocer los gustos y preferencias de los usuarios y su reacción ante el diseño de la aplicación.

Como estudios futuros se tiene proyectado realizar un comparativo entre la enseñanza tradicional y la enseñanza con la herramienta educativa, para medir la efectividad del aprendizaje en los estudiantes.

Referencias

- Arroyo, F.E. (2006). Software educativo y colaborativo para el aprendizaje de la asignatura Tecnología Didáctica I. *Omnia*, 12(3), 109-122.
- Barragán, A., Aguilar, M., Cerpa, G. y Núñez, H. (2009). Relaciones docente-alumno y rendimiento académico. Un caso del Centro Universitario de Ciencias Exactas e Ingenierías de la Universidad de Guadalajara. *Sinéctica*, (33), 1-15. Recuperado de http://www.scielo.org.mx/scielo.php?pid=S1665-109X2009000200006&script=sci_arttext
- Cairo Battistutti, O. (2005). *Metodología de la programación. Algoritmos, diagramas de flujo y programas*. AGREGAR DATOS FALTANTES: Ciudad: Editorial.
- Cavazos Salazar, R. y Torres Flores, S. (2016). Diagnóstico del uso de las tecnologías en el proceso de enseñanza y aprendizaje en la educación superior. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 7(13), 1-20. Recuperado de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-74672016000200273&lng=es&tlng=es.
- Fuentes-Rosado, J.I. y Moo-Medina, M. (2017). Dificultades de aprender a programar. *Revista Educación en Ingeniería*, 12(24), 76. DOI: 10.26507/rei.v12n24.728
- Jaramillo Naranjo, L.M. y Puga Peña, L.A. (2016). El pensamiento lógico-abstracto como sustento para potenciar los procesos cognitivos en la educación. *Sophia, Colección de Filosofía de la Educación*, 2(21), 31. DOI: 10.17163/soph.n21.2016.01

- Lahtinen, E., Ala-Mutka, K. & Järvinen, H.M. (2005). A study of the difficulties of novice programmers. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 14-18. DOI: 10.1145/1067445.1067453
- Liebenberg, J., Huisman, M. & Mentz, E. (2014). Knowledge and Skills Requirements for Software Developer Students. *International Journal of Computer and Information Engineering*, 8(8), 2604-2609. DOI: 10.5281/ZENODO.1094433
- Lund, A.M. (2001). Measuring usability with the USE questionnaire. *Usability Interface*, 8(2), 3-6. DOI: 10.1177/1078087402250360
- Marquès, P. (2005). Evaluación y selección de software educativo. *Comunicación y Pedagogía: Nuevas Tecnologías y Recursos Didácticos*, 185, 31-37.
- Rubin, J., & Chisnell, D. (2008). *Handbook of Usability Testing. How to Plan, Design, and Conduct Effective Tests* (2nd ed.). Indianapolis, Indiana: Wiley Publishing, Inc.
- Salgado Castillo, A., Alonso Berenguer, I., Gorina Sánchez, A. y Tardo Fernández, Y. (2013). Lógica, algorítmica para la resolución de problemas de programación computacional: una propuesta didáctica. **AGREGAR NOMBRE DE LA REVISTA**, IV(1), 57-76.
- Sánchez-Otero, M., García-Guiliany, J., Steffens-Sanabria, E. y Hernández-Palma, H. (2019). Estrategias pedagógicas en procesos de enseñanza y aprendizaje en la educación superior incluyendo tecnologías de la información y las comunicaciones. *Información Tecnológica*, 30(3), 277-286. DOI: 10.4067/s0718-07642019000300277
- Serrano-Angulo, J. y Cebrián-Robles, D. (2014). Usabilidad y satisfacción de la e-Rúbrica. *REDU. Revista de Docencia Universitaria*, 12(1), 177-195.
- UNESCO. (2013). *Enfoques estratégicos sobre las TICs en educación en América Latina y el Caribe*. Oficina Regional de Educación para América Latina y el Caribe.
- Valencia Serrano, M. y Caicedo Tamayo, A.M. (2017). Diseño de tareas apoyadas en TIC para promover aprendizaje autorregulado. *Pensamiento Psicológico*, 15(2), 15-28. DOI: 10.11144/Javerianacali.PPSI15-2.dtat
- Vidal Ledo, M., Gómez Martínez, F. y Ruiz Piedra, A.M. (2010). Software educativos . Educación Médica Superior . scielocu. **AGREGAR DATOS QUE FALTAN**.

Zavaleta Carrillo, P., Pérez Cruz, D. y Cocón Juárez, J.F. (2016). Comparación de herramientas para la enseñanza de la programación. *Pistas Educativas*, (120), 300-316.

