

El efecto de una herramienta de visualización de programas en la eficiencia y en la autoeficacia de los alumnos

Maximiliano Paredes-Velasco
Universidad Rey Juan Carlos
Móstoles, España
maximiliano.paredes@urjc.es

Isidoro Hernán-Losada
Universidad Rey Juan Carlos
Móstoles, España
isidoro.hernan@urjc.es

J. Ángel Velázquez-Iturbide
Universidad Rey Juan Carlos
Móstoles, España
angel.velazquez@urjc.es

Carlos-María Alcover
Universidad Rey Juan Carlos
Móstoles, España
carlosmaria.alcover@urjc.es

Abstract—La percepción que tenemos nosotros mismos sobre nuestras propias capacidades y habilidades constituye un factor muy importante a la hora de realizar tareas y alcanzar objetivos. El término “autoeficacia” define precisamente esta percepción. La autoeficacia constituye un referente importante en la motivación del alumno en el aprendizaje. En este artículo hemos realizado una experiencia con alumnos en el aprendizaje de la programación. En ella analizamos la implicación del uso de una herramienta interactiva de visualización de programas recursivos en la autoeficacia y eficiencia del aprendizaje. Los resultados obtenidos no nos permiten confirmar un incremento significativo de estos dos aspectos (autoeficacia y eficiencia) al no encontrarse diferencias estadísticamente significativas, pero sí que observamos una tendencia de mejora de los mismos con el uso de la herramienta.

Keywords—Autoeficacia; Motivación; Visualización de programas; Diseño algorítmico; Recursividad

I. INTRODUCCIÓN

En las carreras de ingeniería de informática, los alumnos presentan dificultades en el aprendizaje en algunas de las asignaturas más técnicas y específicas de la carrera, como son las asignaturas de introducción a la programación y las de algoritmia. Este aspecto genera mucha desmotivación entre los alumnos, llegando incluso a producir el abandono de sus estudios de informática [1]. La mayoría de los planes de estudio de las carreras de informática ofertan en segundo curso, y una vez que se han cursado las asignaturas de introducción a la programación, la asignatura de algoritmia, en la que se estudia principalmente la complejidad algorítmica y técnicas de diseño de algoritmos como divide y vencerás, algoritmos voraces y *backtracking*. Aunque estos temas son específicos de programación y por tanto deberían captar la atención e ilusión del alumno, en general ocurre lo contrario. El alumno encuentra demasiada complejidad en los mismos y se desmotiva en consecuencia.

La motivación precisamente es un aspecto esencial del aprendizaje de la programación, y en particular la autoeficacia, entendiendo ésta como la percepción que tenemos nosotros mismos sobre nuestras capacidades. En este artículo presentamos una experiencia realizada con alumnos en el aula en la asignatura de *Diseño y Análisis de Algoritmos* en la que estudiamos la autoeficacia en el aprendizaje de la programación de algoritmos de divide y vencerás. El objetivo es analizar el efecto de una herramienta interactiva de visualización de algoritmos recursivos sobre la autoeficacia del alumno y sobre su eficiencia para aprender los conceptos de recursividad y del esquema de divide y vencerás. En la sección II del artículo se presenta una introducción a la autoeficacia desde el punto de vista de la educación, para posteriormente en la sección III describir la experiencia realizada. En las secciones IV y V se presentan los resultados de la investigación y discusión y conclusiones de la misma respectivamente.

II. AUTOEFICACIA

Dentro del marco de la Teoría Cognitiva Social [2] el término autoeficacia se define como la creencia que tenemos en nuestras propias capacidades para organizar y ejecutar los cursos de acción necesarios para alcanzar objetivos o logros futuros [3]. La autoeficacia se genera a partir de cuatro fuentes de información que influyen en la sensación de eficacia que tiene el sujeto [3]: (1) las experiencias previas de éxito o fracaso. El éxito refuerza la creencia en nuestra capacidad personal mientras que situaciones de fracaso las debilita; (2) la experiencia vicaria, la cual la percibimos a través de observar el logro de los demás y compararnos con ellos. Sobrepasar el logro

de los demás hace crecer nuestra autoeficacia; (3) la persuasión verbal y *feedback*. Aumentamos nuestra confianza si oímos mensajes y opiniones de los demás que expresan confianza en nuestra propia capacidad; y (4) los estados fisiológicos, afectivos o emocionales en los que se encuentra el sujeto.

La autoeficacia constituye un referente importante en la motivación del alumno en el aprendizaje [4]. Bandura [3] afirma que las creencias de la autoeficacia se pueden trabajar a nivel de tarea o de dominio. Por tanto, el estudio de la autoeficacia desde el dominio particular del aprendizaje puede resultar de gran interés, incluso dentro del propio proceso de aprendizaje, a nivel de sub-áreas como pueden ser el aprendizaje de la informática o de la programación. Son varios los autores que señalan la importancia de la autoeficacia en el aprendizaje de la informática y del uso del computador. Compeau y Higgins [5] definen la autoeficacia en el uso de computadores (*computer self-efficacy*) como la creencia de la capacidad de uno mismo para usar el computador. Estos autores señalan que los sujetos con poca confianza en sus habilidades para usar los computadores pueden llevar a cabo peor tareas con computadores y [6] señala que la autoeficacia puede ser un factor muy importante para la adquisición de habilidades en el uso de computadores. La autoeficacia puede hacer que el estudiante invierta pequeños esfuerzos en aprender nuevos conceptos de informática [7], mejorando así la eficacia del aprendizaje.

Como consecuencia de la importancia que tiene la autoeficacia en el aprendizaje, son muchos los estudios y trabajos que se han llevado a cabo sobre la misma. En un análisis bibliométrico [8] realizado desde el año 2006 a 2015, ya tan solo en revistas, se recogen más de 81 publicaciones de trabajos de autoeficacia en entornos de aprendizaje basados en computador. Sin embargo, aunque es prolija la investigación en esta línea no hay una definición clara que constata la aportación y mejora de la relación del uso del computador con la autoeficacia, y menos aún en el aprendizaje de la programación. Si bien hay trabajos que señalan la correlación de la autoeficacia con el aprendizaje de la programación [9-11], otras investigaciones no son concluyentes y

dejan abiertas líneas exploratorias en este campo. San y sus colegas [7] no encontraron relación entre la autoeficacia y el uso de Internet en las tareas de aprendizaje, enfatizando además la necesidad de integrar correctamente el computador en la tarea para mejorar la autoeficacia. En el contexto del aprendizaje de programación Java, Jedege [12] no encontró relación de mejora entre la autoeficacia y las experiencias previas de programación y entre la autoeficacia y el uso de computadores, y señaló la necesidad de realizar estudios que identifiquen otros factores de la autoeficacia en el aprendizaje de Java, como pueden ser por ejemplo la integración de herramientas específicas para el aprendizaje.

Otros trabajos han intentado mejorar la autoeficacia en el aprendizaje de los conceptos de programación incorporando herramientas de programación visual. Este es el caso, por ejemplo, del uso de Scratch¹, el cual se combinó con tareas de programación C# [13]. Estos trabajos no identificaron mejoras en la autoeficacia ni en la eficiencia de aprendizaje. Sin embargo Quille y Bergin [13] señalaron que la incorporación de herramientas de programación visual podrían mejorar la autoeficacia del estudiante y su progreso del curso, indicando la necesidad de realizar investigaciones al respecto.

En este trabajo exploramos la inclusión de una herramienta de visualización de gráficos en el contexto de su implicación en la autoeficacia y en el avance del aprendizaje.

III. EXPERIENCIA EN EL AULA

Se describe a continuación el contexto y el desarrollo de la experiencia realizada en el entorno del aula con los alumnos.

A. Asignatura: Diseño y Análisis de Algoritmos

La asignatura en la que aplicamos el uso de SRec es Diseño y Análisis de Algoritmos (DAA) de segundo curso de los grados de informática. Esta asignatura está organizada en dos grandes bloques: (I), eficiencia algorítmica y recursividad, y (II), esquemas algorítmicos (divide y vencerás, voraz, vuelta atrás y ramificación y poda) [14].

¹ <https://scratch.mit.edu/>

B. La herramienta SRec

SRec es un sistema de visualización de programas que muestra la ejecución de procesos recursivos [15]. El alumno programa estos procesos en Java y la herramienta proporciona los recursos para generar varias representaciones gráficas de su comportamiento: rastros (“trazas”), pila de control y árboles de recursión. La utilización de la herramienta es muy sencilla. En primer lugar, el alumno debe cargar el fichero con el código fuente Java, posteriormente seleccionar el método cuyo comportamiento quiere visualizar y posteriormente lanzar su ejecución. A partir de este momento el alumno puede interactuar con las visualizaciones que genera la herramienta.

La herramienta permite principalmente cambiar las propiedades de los elementos gráficos de las visualizaciones, filtrar la cantidad de información a mostrar, cambiar el orden relativo de los datos en algunas visualizaciones, navegar por una visualización grande, buscar datos en la visualización y ampliar información sobre la misma. La herramienta SRec se introdujo al final del bloque I y al inicio del bloque II de la asignatura, de forma consecutiva.

C. Formulación de hipótesis

La hipótesis de trabajo de nuestra investigación se enuncia de la siguiente forma:

H1: El sistema de visualización de programas SRec mejora la eficacia educativa en aprendizaje de procesos recursivos y de estrategias de resolución de divide y vencerás.

H2: El sistema de visualización de programas SRec mejora la autoeficacia del alumno en el contexto de uso y aplicación de procesos recursivos en estrategias de resolución de divide y vencerás.

D. Sujetos, variables y procedimiento

La experiencia se realizó con dos grupos de alumnos del Grado de Ingeniería Informática y Grado de Ingeniería de Software de la Universidad Rey Juan Carlos durante el curso académico 2015-2016, participando un total de 95 estudiantes. Con el objetivo de validar la hipótesis, se organizaron dos grupos: grupo de control, en adelante GC, y grupo experimental, en adelante GE. Las variables dependientes de estudio fueron: 1) la autoeficacia y 2) la eficiencia de aprendizaje. La variable

independiente es la herramienta de soporte y apoyo que dispone el estudiante para el desarrollo de la tarea. Ambos grupos, tanto de control como experimental, realizaron la misma tarea.

Todos los contenidos y actividades se organizaron en clases teóricas (clases magistrales combinadas con resolución de problemas) y en clases prácticas (en laboratorios de computadores). El lenguaje de programación que se utilizó en las actividades prácticas fue Java. El mismo profesor impartió todas las clases a los dos grupos. La experiencia abarcó los temas 3 y 4 de la asignatura, titulados “Análisis de complejidad II” y “Divide y vencerás” respectivamente y duró cuatro semanas. Se distribuyó en varias sesiones de dos horas organizadas en tres fases: (1) se realizan los pretest de conocimientos de algoritmos recursivos y divide y vencerás, así como el pretest de autoeficacia; (2) se realizan varias tareas (T1 y T2 sobre programación de recursividad y T3 sobre divide y vencerás); y (3) se realizan los respectivos postest.

E. Instrumentos y contenidos

Como instrumento de medida de la variable eficiencia de aprendizaje se utilizó un test formado por 17 ítems con preguntas tipo selección y de texto libre. Como instrumento utilizado para medir la variable autoeficacia hemos utilizado la escala de Autoeficacia académica [16]. Esta escala fue diseñada por [17] y refleja la creencia del estudiante que tiene sobre su propia capacidad para realizar en el futuro tareas académicas. La escala incluye 5 ítems con opciones desde 1 (nunca) hasta 4 (siempre). La escala de Autoeficacia académica ha sido traducida al español y utilizada en otras experiencias en ambientes educativos universitarios. La elección de utilización de esta escala se debe a que está ampliamente utilizada en contexto educativo y está validada. El referente utilizado en los ítems es recursividad, complejidad y algoritmos divide y vencerás. Un ejemplo de ítem es: “*Voy a ser capaz de entender los temas más difíciles de recursividad, del esquema algorítmico Divide y Vencerás y de complejidad*”.

Ambos cuestionarios, tanto de autoeficacia como eficiencia de aprendizaje, se realizaron de forma anónima. Los alumnos participaron voluntariamente y sin recibir bonificaciones en sus notas finales de la asignatura por su participación.

Como ya se ha indicado anteriormente, los alumnos realizan tres tareas en la Fase 2 de la experiencia. Las tareas T1 y T3 son las mismas para los dos grupos, mientras que la tarea T2 tiene una parte igual y otra diferente: a los alumnos del GE se le muestra el uso de SRec y se les facilita un par de ejemplos (el cálculo de una potencia y los números de Fibonacci) con los que trabajar y familiarizarse con el uso de SRec. Adicionalmente en esta tarea T2, a ambos grupos se les solicitó desarrollar dos programas (inversión de los dígitos de un número y el cálculo de un número combinatorio). Todos los alumnos utilizaron el EDI NetBeans y adicionalmente los del GE usaron SRec (ver Figura 1).

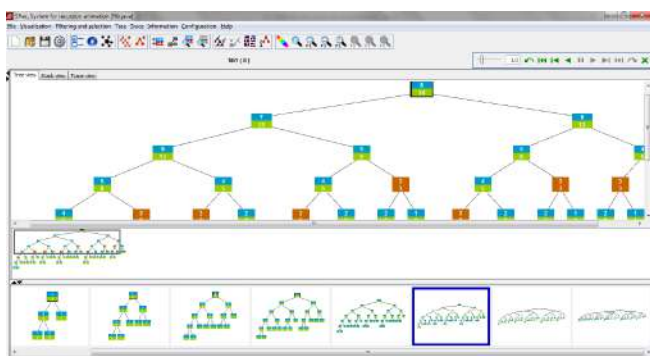


Fig. 1. Herramienta SRec utilizada por el grupo experimental GE.

IV. RESULTADOS

Se presentan los resultados organizados por las dos variables de estudio.

A. Eficiencia de aprendizaje

La muestra final del grupo experimental para la variable eficiencia de aprendizaje fue de 34 alumnos y la del grupo de control fue 41. La tabla I muestra un resumen de las medias de los test de conocimiento en recursividad y en divide y vencerás.

TABLA I. MEDIA DE LOS TEST DE EFICIENCIA DE APRENDIZAJE

	Recursividad			Divide y vencerás			N
	M-Pre	M-Pos	Dif	M-Pre	M-Pos	Dif	
GE	5,15	5,41	0,26	2,91	3,79	0,88	34
GC	4,90	4,83	-0,07	2,49	3,67	1,18	41

Los alumnos del grupo experimental han obtenido mayores puntuaciones a la finalización de la experiencia tanto en conocimientos de recursividad como en el diseño de divide y vencerás (ver Tabla I, columnas M-Pos).

Interpretamos como avance de aprendizaje la diferencia que hay entre la puntuación que ha obtenido el alumno en el postest de conocimiento y la obtenida en el pretest. La Tabla I muestra que este valor, para los conocimientos de recursividad, en el caso del grupo experimental, es positivo mientras que en el grupo de control es negativo (casi nulo). Para los conocimientos de divide y vencerás, ambos valores son positivos.

B. Autoeficacia

La muestra final del grupo experimental para la autoeficacia fue 44 y la del grupo de control fue 51. Como se puede observar en la Tabla II, la media del valor de la autoeficacia del grupo experimental ($M=2,84$) es mayor que la del grupo de control ($M=2,54$). Con el objeto de determinar si esta diferencia entre medias son representativas, procedemos a realizar pruebas de normalidad en primer lugar y determinar en consecuencia la aplicación de estudios paramétricos o no paramétricos. Se realiza el estudio de normalidad con un intervalo de confianza al 95% ($\alpha=0,05$) usando el método de Shapiro-Wilk, obteniendo que el grupo experimental sí sigue una distribución normal pero sin embargo no ocurre así en el grupo de control.

TABLA II. ESTADÍSTICA DESCRIPTIVA DE LAS MUESTRAS DE AUTOEFICACIA

	M		SD		N	
	M-Pre	M-Pos	SD-Pre	SD-Pos	N-Pre	N-Pos
GE	3,05	2,84	0,5239	0,4961	58	44
GC	2,90	2,54	0,4407	0,4738	51	51

Al no seguir una de las dos muestras una distribución normal aplicamos métodos no paramétricos para el estudio de las medias de ambos grupos. El test de contraste de medias entre grupos que aplicamos es Wilcoxon (prueba de suma de rangos Wilcoxon) dando como resultado $W=1507,5$ y $p\text{-value}=0,003719$. Como podemos ver el p-valor (0,003719) es menor que $\alpha=0,05$ por lo tanto, las diferencias son estadísticamente significativas.

Veamos ahora los resultados de los pretest de autoeficacia (no se aportan las tablas de estos resultados por restricciones de espacio). La Tabla II muestra la media de los pretest en ambos grupos ($M=3,05$ y $M=2,90$ para GE y GC respectivamente). Con el objeto de determinar si la diferencia entre estas medias es significativa estadísticamente comprobamos la normalidad de las muestras

aplicando Shapiro-Wilk y concluimos que la muestra del pretest del GC sigue una distribución normal mientras que la del GE no es así. En consecuencia, aplicamos el Test de contraste de medias no paramétrico Wilcoxon (prueba de suma de rangos Wilcoxon) (para $\alpha=0,05$) dando como resultado $W=1724,5$ y $p\text{-value}=0,1332$, por lo que no hay diferencia significativa.

V. DISCUSIÓN Y CONCLUSIONES

En lo relativo al aprendizaje del esquema algorítmico divide y vencerás, los resultados muestran que la diferencia entre el postest y el pretest para el grupo experimental es de 0,81 y para el grupo de control es 1,18. Además, la media del pretest en el grupo de control es menor que la del grupo experimental. Ante estos resultados, y en una primera lectura, nos surge la duda de si es más fácil que el grupo de control experimente un avance mayor que el grupo experimental, al partir de unos conocimientos más bajos. Además, parece ser (a falta de confirmar si las diferencias entre postest y pretest son significativas) que los alumnos del grupo de control han aprendido más que los del grupo experimental. Por lo tanto, no podemos aceptar la hipótesis H1 en la que se afirma que la herramienta SRec mejora el aprendizaje del esquema divide y vencerás. Sin embargo, poniendo el foco en las medias de los postest, vemos que la del grupo experimental, $M=3,79$, es mayor que la del grupo de control ($M=3,67$). Esto indica que a la finalización de la tarea, los alumnos del grupo experimental terminaron con un mayor nivel de conocimientos en divide y vencerás que el grupo de control. Debido a que las diferencias de estas medias no son estadísticamente significativas, debemos tomar este hecho como que observamos una tendencia de mejoría en el aprendizaje del esquema divide y vencerás con el uso de la herramienta SRec.

En lo relativo al aprendizaje de la recursividad podemos ver que el grupo experimental obtiene mejores resultados finales ($M=5,41$) frente a los del grupo de control ($M=4,83$), además, el avance de aprendizaje del grupo experimental ha sido mayor (0,26) que el de control (-0,07), el cual ha experimentado retroceso. A falta de confirmar si las diferencias son estadísticamente significativas, podemos afirmar también que observamos tendencia

en la mejora del aprendizaje de la recursividad con la herramienta. No sabemos muy bien la causa del retroceso del grupo de control en el aprendizaje de la recursividad, podría estar relacionada con un sentimiento de desmotivación del alumno en el proceso de aprendizaje, o bien podría estar relacionada con posibles confusiones e incertidumbres que se generen en el alumno cuando programa recursividad sin entender totalmente los árboles de recursión que generan dichos programas.

En la autoeficacia vemos que el grupo experimental tiene mayor índice de autoeficacia ($M=2,84$) que el grupo de control ($M=2,54$). Teniendo en cuenta que las diferencias de estas medias son estadísticamente significativas podemos pensar con cautela que los alumnos que usaron SRec acabaron obteniendo mayor confianza en sus propias habilidades para abordar problemas de programación de recursividad, de divide y vencerás y de complejidad algorítmica que los alumnos que no lo utilizaron, generando por tanto el uso de SRec un mayor grado de confianza en sí mismos.

De especial interés son los resultados de los pretest ($M=3,05$ y $M=2,90$ para GE y GC respectivamente) y su comparación con los valores obtenidos en los postest por grupo. Como se puede ver comparando las medias de los pretest y los postest por grupo, la autoeficacia decrece en ambos grupos. Bien es cierto que en el grupo experimental decrece menos (-0,21) que en el de control (-0,36). Creemos que esta disminución en la autoconfianza que experimentan los alumnos puede estar relacionada con la complejidad intrínseca de la tarea desarrollada por el alumno y el hecho de que la fuente con mayor repercusión e impacto en la autoeficacia es la experiencia previa [3]. La complejidad intrínseca de las tareas de programación en las que trabajan los alumnos podría constituir una experiencia de gran impacto sobre la creencia que tiene el alumno en sus propias capacidades, en la propia autoconfianza, que además, al ser tareas complejas para los alumnos y, en general presentan más experiencias de fracaso que de éxito, harían que decreciera la propia autoconfianza. El éxito refuerza la creencia en nuestra capacidad personal mientras que situaciones de fracaso las debilita.

Esto explicaría que la autoeficacia baje en los dos grupos, tanto de control como experimental, si bien

en el grupo experimental no baja tanto por la intervención de la herramienta SRec. Como dato adicional, señalamos que en una experiencia distinta, que hemos realizado en el año 2017, hemos detectado este mismo extremo: los resultados del postest de la autoeficacia disminuyen respecto a los del pretest, siendo esta experiencia en un dominio totalmente diferente, con alumnos del Grado de Enfermería y sin intervención del uso de los computadores como elemento central de la tarea. La coincidencia de este fenómeno en ambas experiencias nos hace pensar que puede haber una relación entre la complejidad intrínseca de la propia tarea de aprendizaje con la autoeficacia percibida por el alumno.

Como conclusión final, y a tenor de los resultados obtenidos en esta experiencia, no se aceptan las dos hipótesis planteadas, por lo que no podemos afirmar que el uso de SRec mejora la eficiencia ni la autoeficacia del alumno en el aprendizaje del esquema divide y vencerás, si bien hemos observado que su uso tiende a mejorarlo.

Aunque los resultados obtenidos no han sido los esperados, no es una situación desconocida en el uso educativo de visualizaciones. Existen numerosas experiencias en la documentación científica donde la visualización no ha tenido los efectos esperados, siendo variadas las razones de dichos resultados adversos [18]. Como trabajo futuro pretendemos profundizar en el análisis estadístico de los datos obtenidos en la experiencia, analizando en primer lugar las puntuaciones obtenidas en los test de árboles de recursión y en segundo lugar, realizar análisis de correlación entre la autoeficacia y la eficiencia de aprendizaje. También es aconsejable volver a repetir la evaluación, bien para comprobar si los resultados se reproducen bien para intentar introducir cambios en la experiencia de forma que puedan llevar a mejores resultados del grupo experimental.

AGRADECIMIENTOS

Este trabajo se ha financiado con los proyectos de investigación TIN2015-66731-C2-1-R del Ministerio de Economía y Competitividad,

S2013/ICE-2715 de la Comunidad Autónoma de Madrid, y 30VCP1G15 de la Universidad Rey Juan Carlos.

REFERENCES

- [1] Vallerand, Robert J.; Fortier, Michelle S.; Guay, Frédéric (1997). "Self-determination and persistence in a real-life setting: Toward a motivational model of high school dropout. *Journal of Personality and Social Psychology*", Vol 72(5), May 1997, 1161-1176.
- [2] Bandura, A. (1999). A social cognitive theory of personality. In L. Pervin & O. John (Ed.), *Handbook of personality* (2nd ed., pp. 154-196). New York: Guilford Publications.
- [3] Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York, NY: Freeman.
- [4] Zimmerman, B.J. (2000). Self-Efficacy: An Essential Motive to Learn. *Contemporary Educational Psychology* 25: 82-91.
- [5] Compeau, D. R., & Higgins, C. A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly*, 19, 189-211.
- [6] Miura, I. T. (1987). The relationship of computer self-efficacy expectations to computer interest and course enrollment in college. *Sex Roles*, 16, 303-311.
- [7] Sam, H. K., Othman, A. E. A., & Nordin, Z. S. (2005). Computer Self-Efficacy, Computer Anxiety, and Attitudes toward the Internet: A Study among Undergraduates in Unimas. *Educational Technology & Society*, 8 (4), 205-219.
- [8] Valencia-Vallejo, N., López-Vargas, O., & Sanabria-Rodríguez, L. (2016). Self-Efficacy in Computer-Based Learning Environments: A Bibliometric Analysis. *Psychology*, 7, 1839-1857.
- [9] Ramalingam, V., LaBelle, D. & Wiedenbeck, S. (2004). Self-Efficacy and Mental Models in Learning to Program. *ITICSE'04 June 28-30*, pp. 171-175.
- [10] Zingaro, D. (2014). Peer Instruction Contributes to Self-Efficacy in CS1, *SIGCSE'14, March 3-8, 2014, Atlanta*
- [11] Govender, I., Govender, D., Havenga, M., Mentz, E., Breed, B., Dignum, F. & Dignum, V. (2014) Increasing self-efficacy in learning to program: exploring the benefits of explicit instruction for problem solving, *TD The Journal for Transdisciplinary Research in Southern Africa*, 10(1), pp. 187-200.
- [12] Jedge, P. O. (2009). Predictors of java programming self-efficacy among engineering students in a nigerian university. *IJCSIS, International Journal of Computer Science and Information Security Vol. 4, No. 1 & 2*.
- [13] Quille, K. & Bergin, S. (2016). Does Scratch improve self-efficacy and performance when learning to program in C#? An empirical study. *International Conference on Engaging Pedagogy (ICEP)*, Maynooth University, Ireland.
- [14] Brassard, G., & Bratley, P. (1996). *Fundamentals of Algorithmics*. Englewood Cliffs, NJ: Prentice-Hall.
- [15] Velázquez-Iturbide, J.A., Pérez-Carrasco, A. & Urquiza-Fuentes J. (2008). SRec: An animation system of recursion for algorithm courses," en *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2008, ACM, Madrid*, pp. 225-229.
- [16] Bresó, E., Schaufeli, W.B. and Salanova, M. (2011). Can a self-efficacy-based intervention decrease burnout, increase engagement, and enhance performance? A quasi-experimental study. *High Educ*, 61, 339-355.
- [17] Midgley, C., Maehr, M. L., Hruda, L. Z., Anderman, E., Anderman, L., Freeman, K. E., et al. (2000). *Manual for patterns of adaptive learning scales*. Ann Arbor: University of Michigan Press.
- [18] Urquiza-Fuentes, J. & Velázquez-Iturbide, J. Á. (2009). A survey of successful evaluations of program visualization and algorithm animation systems, *ACM Transactions on Computing Education*, vol. 9, no. 2, artículo 9.