

Capítulo 13

Herramienta de Visualización de Programas SRec: Un Estudio de la Autoeficacia del Estudiante

Maximiliano Paredes-Velasco, Isidoro Hernán-Losada, J. Ángel Velázquez-Iturbide, Carlos-María Alcover

Title—SRec program visualization system: Studying student’s self-efficacy.

Abstract— Learning to program is a complex and hard task. Program visualizations can assist students in learning programming, in particular in learning recursion. The SRec visualization system generates and allows students interacting with the recursion tree associated to given input data, assisting him/her in understanding the recursive process. This paper shows an experience with students in learning recursion, where the effect of the SRec system on students’ self-efficacy and learning efficacy is analysed. A total of 95 computer science students participated in the experience. The authors found that students who used SRec scored higher than students who used an IDE (Integrated Development Environment) in recursion learning.

Keywords— Self-efficacy; Motivation; Programs Visualization; Algorithm designing; Recursion

Abstract— El aprendizaje de la programación es un proceso complejo y difícil, al cual pueden ayudar las visualizaciones de la ejecución de programas. En concreto, esta situación se da en el aprendizaje de la recursividad. El sistema SRec genera y permite interactuar con el árbol de recursión correspondiente a cualquier ejecución, ayudando al estudiante a entender el proceso recursivo. En este artículo presentamos una experiencia con estudiantes en el aprendizaje de la recursividad, a partir de la cual hemos analizado el efecto del uso del sistema SRec en la autoeficacia y eficacia del aprendizaje de los estudiantes. En la experiencia participaron 95 estudiantes de informática y los resultados indicaron que los estudiantes que utilizaron SRec tuvieron mejores

puntuaciones de recursividad que los que usaron un entorno integrado de desarrollo.

I. INTRODUCCIÓN

EN los grados de ingeniería de informática, los estudiantes presentan dificultades para aprender algunas de las asignaturas más técnicas y específicas de la carrera, como son las asignaturas de introducción a la programación y de algoritmia. Este hecho genera mucha desmotivación entre los estudiantes, 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 la asignatura de algoritmia en segundo curso, una vez que se han cursado las asignaturas de introducción a la programación. En la asignatura de algoritmia se estudia principalmente la complejidad algorítmica y técnicas de diseño de algoritmos como divide y vencerás, algoritmos voraces y vuelta atrás. Aunque estos temas son específicos de programación y por tanto deberían captar la atención e ilusión del estudiante, en general ocurre lo contrario. El estudiante encuentra demasiada complejidad en los mismos y en consecuencia se desmotiva.

La motivación precisamente es un aspecto esencial del aprendizaje de la programación, y en particular la autoeficacia, entendiéndola ésta como la percepción que tenemos nosotros mismos sobre nuestras capacidades. En este artículo presentamos una experiencia realizada con estudiantes en el aula en la asignatura “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 estudiante y sobre su eficacia para aprender los conceptos de recursividad y del esquema de divide y vencerás.

J. Ángel Velázquez-Iturbide es profesor en la Escuela Técnica Superior de Ingeniería Informática de la Universidad Rey Juan Carlos, 28933 Móstoles, Madrid, [España{Angel.velazquez@urjc.es}](mailto:Angel.velazquez@urjc.es)

Carlos María-Alcover es profesor de la Facultad de Ciencias de la Salud de la Universidad Rey Juan Carlos, 28933, Móstoles, Madrid, España [{Carlosmaria.alcover@urjc.es}](mailto:Carlosmaria.alcover@urjc.es)

Este trabajo fue presentado originalmente al CONGRESO IX INTERNATIONAL SYMPOSIUM ON COMPUTERS IN EDUCATION (SIEE 2017).

Maximiliano Paredes-Velasco es profesor en la Escuela Técnica Superior de Ingeniería Informática de la Universidad Rey Juan Carlos, 28933 Móstoles, Madrid, [España{Maximiliano.paredes@urjc.es}](mailto:Maximiliano.paredes@urjc.es)

Isidoro Hernán-Losada es profesor en la Escuela Técnica Superior de Ingeniería Informática de la Universidad Rey Juan Carlos, 28933 Móstoles, Madrid, [España{Isidoro.hernan@urjc.es}](mailto:Isidoro.hernan@urjc.es)

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 respectivamente finalizando el artículo con las conclusiones en la sección VI.

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 ajena, que es percibida a través del logro de los demás y de compararnos con ellos. Sobrepasar el logro de los demás hace crecer nuestra autoeficacia.
3. La persuasión verbal y realimentación. 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 estudiante 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 subáreas como pueden ser el aprendizaje de la programación o de la algoritmia. 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 ni 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 Scratch, el cual se combinó con tareas de programación C# [13]. Estos trabajos no identificaron mejoras en la autoeficacia ni en la eficacia 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 estudiantes.

A. Asignatura: Diseño y Análisis de Algoritmos

La evaluación se ha realizado en la asignatura “Diseño y Análisis de Algoritmos” de segundo curso de los grados de informática. La 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].

B. La Herramienta SRec

SRec es un sistema de visualización de programas que muestra la ejecución de procesos recursivos [15]. Está concebido para ayudar a la docencia de la algoritmia, aunque también puede usarse en asignaturas de programación [16]. El estudiante programa métodos en Java y la herramienta genera varias representaciones gráficas del proceso recursivo: rastros (“trazas”), pila de control y árboles de recursión.

La utilización de la herramienta es muy sencilla. En primer lugar, el estudiante debe cargar el fichero con el código fuente Java, posteriormente seleccionar el método cuyo comportamiento quiere visualizar y posteriormente lanzar su ejecución.

SRec proporciona un diálogo para la ejecución del programa que ha cargado el estudiante. El usuario puede especificar un solo valor para los datos de entrada o varios. En el primer caso, se genera el árbol de recursión correspondiente a su ejecución. En el segundo caso, se genera un bosque de recursión, formado por tantos árboles como juegos de datos de entrada se hayan especificado. Una vez que el usuario selecciona uno de los árboles, puede interactuar con él. Por ejemplo, la Figura 1 muestra una captura de la pantalla principal de la herramienta en la que se

han generado 9 árboles de recursión para la serie de Fibonacci ($n=2..10$), aunque se muestra en tamaño grande el correspondiente a $n=8$ tras haber sido seleccionado.

A partir de este momento el estudiante puede interactuar con las visualizaciones que genera la herramienta. La herramienta permite 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.

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 estudiante 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 estudiantes 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 las 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 eficacia 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 respectivamente “Análisis de complejidad II” (final del bloque I de la asignatura) y “Divide y vencerás” (principio del bloque II) 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; (2) se realizan varias tareas (T1 y T2 sobre programación de recursividad y T3 sobre divide y vencerás); y (3) se realizan los postest de conocimiento y de autoeficacia.

E. Instrumentos y Contenidos

Como instrumento de medida de la variable eficacia 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 [17]. Esta escala fue diseñada por Midgley *et al.* [18] 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 de 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 eficacia de aprendizaje, se realizaron de forma anónima. Los estudiantes participaron voluntariamente y sin recibir bonificaciones en sus notas finales de la asignatura por su participación.

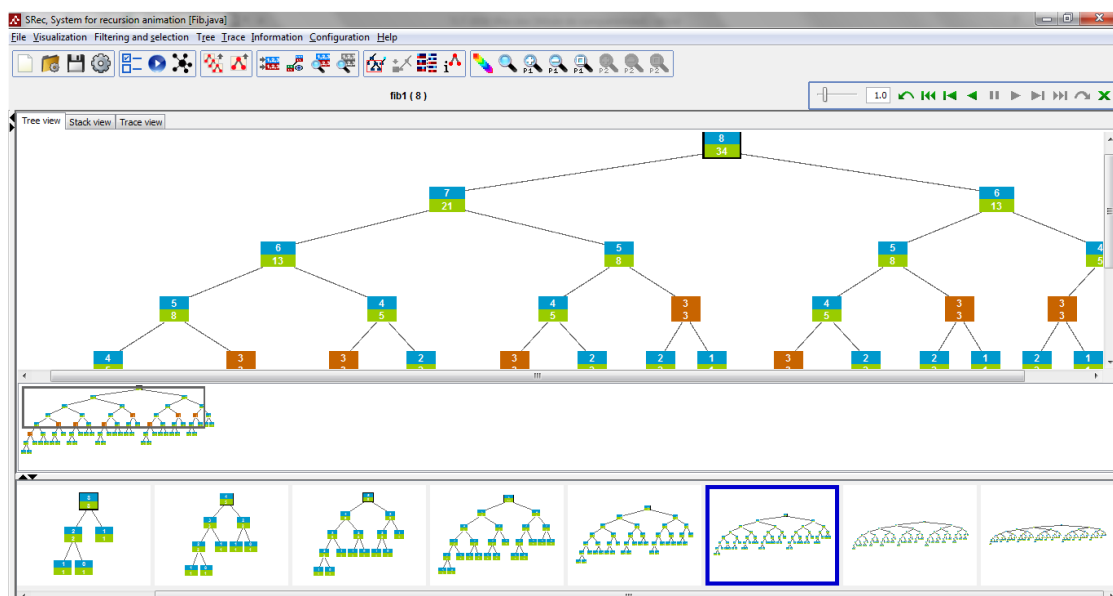


Figura .1. Herramienta SRec utilizada por el grupo experimental GE.

Como ya se ha indicado anteriormente, los estudiantes 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 estudiantes 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 estudiantes utilizaron el EDI NetBeans y adicionalmente los del GE usaron SRec (ver Figura 1).

IV. RESULTADOS

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

A. Eficacia de Aprendizaje

La experiencia comenzó con 95 participantes entre los dos grupos (experimental y control) sin embargo, hubo algunos sujetos que fueron eliminados del estudio estadístico ya que algunos estudiantes, o no realizaron los dos tests (pretest y postest) o bien los realizaron, pero se equivocaron al introducir el código personal anónimo en las cabeceras de los test. Por tanto, la muestra final se redujo a 74 participantes distribuyéndose en 36 estudiante del grupo experimental y 38 en el grupo de control.

La Tabla I muestra las variables estudiadas en el análisis estadístico. Estas variables contienen la media de las puntuaciones obtenidas agrupadas por pretest y postest y por el tipo de conocimiento que evalúa (recursividad y divide y vencerás). Además, se definen dos variables denominadas *Recursividad incremento* y *Divide y vencerás incremento* que miden el avance de aprendizaje del estudiante, tal y como se puede ver en las ecuaciones que las definen:

$$(I) \text{Recursividad incremento } (\Delta r) \\ = \text{Recursividad/postest} \\ - \text{Recursividad/pretest}$$

$$(II) \text{Divide y vencerás incremento } (\Delta d) \\ = \text{Divide y vencerás} \\ - \text{Divide y vencerás/Pretest}$$

La Tabla II visualiza la estadística descriptiva de las muestras de las distintas variables analizadas en el estudio estadístico. Como podemos observar en la Tabla II las medias de la puntuación obtenida en el postest de recursividad del grupo experimental (*Recursividad/Postest* = 5,53) es mayor que las del grupo de control (*Recursividad/Postest* = 4,84). Sin embargo, esto no ocurre en el aprendizaje de divide y vencerás, donde la puntuación del grupo experimental (*Divide y vencerás/Postest* = 3,22) es menor que la del grupo de control (*Divide y vencerás/Postest* = 3,63). Como ya se ha indicado anteriormente por las ecuaciones (I) y (II) interpretamos como incremento o avance de aprendizaje la diferencia que hay entre la puntuación que ha obtenido el estudiante en el postest de conocimiento y la obtenida en el pretest. La Tabla II muestra que este valor, para los conocimientos de recursividad, en el caso del grupo experimental, es positivo ($\Delta r = 0,61$) mientras que en el grupo de control es prácticamente nulo ($\Delta r = -0,11$). Para los

TABLA I
VARIABLES ANALIZADAS EN EL ESTUDIO ESTADÍSTICO

Variable	Descripción	Ítem
Recursividad/Pretest	Puntuación conocimientos de recursividad en el pretest	1-6
Recursividad/Postest	Puntuación conocimientos de recursividad en el postest	1-6
Divide y vencerás/Pretest	Puntuación conocimientos de divide y vencerás en el pretest	7-12
Divide y vencerás/Postest	Puntuación conocimientos de divide y vencerás en el Postest	7-12
Recursividad incremento	Diferencia puntuación de recursividad del postest y pretest	1-6
Divide y vencerás incremento	Diferencia puntuación de divide y vencerás del postest y pretest	7-12

conocimientos de divide y vencerás, ambos valores son positivos y mayor el del grupo de control que el del grupo experimental

Siguiendo en la Tabla II, podemos ver que los valores de los pretest de los dos grupos varían. En recursividad, el pretest del grupo experimental (*Recursividad/Pretest* = 4,92) es menor que el del grupo de control (*Recursividad/Pretest* = 4,95). Igual ocurre en divide y vencerás, donde el grupo experimental tiene un valor menor (*Divide y vencerás/Pretest* = 2,39) que el del grupo de control (*Divide y vencerás/Pretest* = 2,50). Además, los incrementos o avance de aprendizaje en divide y vencerás también es menor en el grupo experimental ($\Delta r = 0,83$) que en el grupo de control ($\Delta r = 1,13$). Con el objetivo de averiguar si las diferencias de los pretest son significativas entre los dos grupos se ha aplicado en primer lugar el test de normalidad para determinar si las muestras siguen una distribución normal o no. La tabla III muestra los resultados del test de *Shapiro-Wilk*, cuyo test ha sido aplicado al tratarse de un tamaño de muestra pequeño en los dos grupos ($N=38$ y $N=36$ de la Tabla II).

Este test indica que no siguen las muestras una distribución normal ($\alpha=0,05$) al ser *sig.* < 0,05. Para asegurar más aun el resultado de normalidad aplicamos también el test de *Kolmogorov-Smirnov* aunque estemos trabajando con muestras pequeñas. Este test también confirma que las muestras no siguen una distribución normal. En consecuencia, se deben aplicar test no paramétricos para la comparación de las medias de las diferentes muestras.

Para comparar si las medias de los pretest son significativas se ha aplicado el test no paramétrico para muestras independiente Mann-Whitney. La tabla IV muestra el resultado de este test, comparando por separado si la puntuación de los pretest de recursividad (*Recursividad/Pretest*) y de divide y vencerás (*Divide y vencerás/Pretest*) son diferentes en el grupo de control y en el experimental. El resultado del test muestra que no existe diferencia entre los grupos para un nivel de significancia de $\alpha < 0,05$.

TABLA II
ESTADÍSTICA DESCRIPTIVA POR GRUPOS

Grupo	Conocimiento/test	Estadísticos descriptivos					
		N	Mínimo	Máximo	Media (M)	Desviación estándar	Varianza
Control	Recursividad/Pretest	38	2	6	4,95	1,064	1,132
	Divide y vencerás/Pretest	38	0	6	2,50	1,856	3,446
	Recursividad/Postest	38	0	6	4,84	1,516	2,299
	Divide y vencerás/Postest	38	0	6	3,63	1,601	2,563
	Recursividad incremento	38	-4	3	-.11	1,448	2,097
	Divide y vencerás incremento	38	-3	6	1,13	2,016	4,063
Experimental	Recursividad/Pretest	36	2	6	4,92	1,251	1,564
	Divide y vencerás/Pretest	36	0	6	2,39	1,573	2,473
	Recursividad/Postest	36	4	6	5,53	.654	.428
	Divide y vencerás/Postest	36	1	6	3,22	1,929	3,721
	Recursividad incremento	36	-2	4	.61	1,315	1,730
	Divide y vencerás incremento	36	-2	5	.83	1,813	3,286

TABLA III
COMPROBACIÓN DE DISTRIBUCIÓN NORMAL SOBRE LAS MUESTRAS

	Grupo	Pruebas de normalidad					
		Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Estadístico	gl	Sig.	Estadístico	gl	Sig.
Recursividad/Pretest	GC	0,233	38	0,000	0,842	38	0,000
	GE	0,249	36	0,000	0,799	36	0,000
Divide y vencerás/Pretest	GC	0,159	38	0,017	0,925	38	0,014
	GE	0,173	36	0,008	0,933	36	0,032
Recursividad/Postest	GC	0,252	38	0,000	0,762	38	0,000
	GE	0,376	36	0,000	0,696	36	0,000
Divide y vencerás/Postest	GC	0,170	38	0,007	0,932	38	0,023
	GE	0,209	36	0,000	0,842	36	0,000
Recursividad incremento	GC	0,182	38	0,003	0,941	38	0,044
	GE	0,290	36	0,000	0,803	36	0,000
Divide y vencerás incremento	GC	0,265	38	0,000	0,901	38	0,003
	GE	0,233	36	0,000	0,920	36	0,012

a. Corrección de significación de Lilliefors

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 V, 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 es representativa, 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 (ver Tabla VI).

TABLA IV
COMPARACIÓN DE MEDIAS DE PRETEST

	Estadísticos de prueba ^a	
	Recursividad /Pretest	Divide y vencerás/Pretest
U de Mann-Whitney	667,000	671,500
W de Wilcoxon	1408,000	1337,500
Z	-0,194	-0,137
Sig. asintótica (bilateral)	0,847	0,891

a. Variable de agrupación: Grupo (GE y GC)

TABLA V
ESTADÍSTICA DESCRIPTIVA DE AUTOEFICACIA

Grupo	N	Estadísticos descriptivos				
		Mínimo	Máximo	Media (M)	Desviación estándar	Varianza
Control	51	1,6	6	2,54	0,474	0,224
Experimental	44	2	4	2,84	0,4961	0,246

Al no seguir una de las dos muestras una distribución normal aplicamos métodos no paramétricos para el estudio de las medias de ambos grupos. Aplicamos un test no paramétrico para muestras no relacionadas para comparar las medias entre grupos con un intervalo de confianza al 99% ($\alpha=0,01$). En concreto aplicamos Mann-Whitney (Tabla VII) donde podemos ver que el p-valor (0,004) es menor $\alpha=0,01$, por lo tanto las diferencias son estadísticamente significativas. Para reforzar más aún el resultado aplicamos el test de Wilcoxon (prueba de suma de rangos Wilcoxon) dando como resultado $W=1507,5$ y $p\text{-value}=0,003719$, que confirma el resultado anterior.

V. DISCUSIÓN

En lo relativo al aprendizaje del esquema algorítmico divide y vencerás, podemos ver que hay una sensible diferencia entre los pretest de ambos grupos (2,50 del grupo de control frente a 2,39 del grupo experimental), por lo que podría parecer que los estudiantes del grupo de control tenían más conocimientos en el esquema algorítmico que los del grupo experimental, y por tanto los grupos estaban desequilibrados desde el punto de vista de los conocimientos iniciales al comenzar la experiencia. Sin embargo, los resultados muestran que esta diferencia no es

TABLA VI
NORMALIDAD DE MUESTRAS

Grupo	W	Sig.
Control	0,94346	0,01694
Experimental	0,95474	0,08269

TABLA VII
COMPARACIÓN DE MEDIAS DE POSTEST

Estadísticos de prueba ^a	Valor
U de Mann-Whitney	736,500
W de Wilcoxon	2062,500
Z	-2,905
Sig. asintótica (bilateral)	0,004

a. Variable de agrupación: Grupo (GE y GC)

estadísticamente significativa (Tabla IV), por lo que los grupos están compensados a nivel de conocimiento del esquema algorítmico divide y vencerás. Los resultados también muestran que la diferencia entre el postest y el pretest es menor en el grupo experimental ($\Delta d=0,83$) que en el grupo de control ($\Delta d=1,13$). 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, partiendo ambos grupos con el mismo nivel de conocimientos. Además, parece ser (a falta de confirmar si las diferencias entre postest y pretest son significativas) que los estudiantes del grupo de control han aprendido más ($M=3,63$) que los del grupo experimental ($M=3,22$). 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. Esto parece indicar que a la finalización de la tarea, los estudiantes del grupo experimental terminaron con un menor nivel de conocimientos en divide y vencerás que el grupo de control. Sin embargo, no se puede afirmar esta cuestión ya que sería necesario analizar estadísticamente si esta diferencia es significativa (tarea que se contempla en un trabajo futuro).

En lo relativo al aprendizaje de la recursividad podemos ver que el grupo experimental obtiene mejores resultados finales ($M=5,53$) frente a los del grupo de control ($M=4,84$), además, el avance de aprendizaje del grupo experimental ha sido mayor ($\Delta r=0,61$) que el de control ($\Delta r=-0,11$), el cual ha experimentado retroceso. A falta de confirmar si las diferencias son estadísticamente significativas, podemos afirmar que observamos tendencia en la mejora del aprendizaje de la recursividad con la herramienta. Aunque pueda parecer que los estudiantes que utilizaron la herramienta tenían unos conocimientos de recursividad menores que el grupo que no la usaron (medias en pretest de 4,92 y 4,95 respectivamente), hay que tener en cuenta que esta diferencia no es estadísticamente significativa, con lo cual ambos grupos partieron aproximadamente del mismo nivel de conocimientos en recursividad.

No sabemos muy bien la causa del retroceso del grupo de control en el aprendizaje de la recursividad ($\Delta r=-0,11$), podría estar relacionada con algún sentimiento de desmotivación del estudiante durante la actividad, o bien podría estar relacionada con posibles confusiones e incertidumbres que se generen en el estudiante cuando programa recursividad sin entender totalmente los árboles de recursión que generan dichos programas.

En la autoeficacia vemos que el grupo experimental tiene mayor valor ($M=2,84$) que el grupo de control ($M=2,54$).

Teniendo en cuenta que las diferencias de estas medias son estadísticamente significativas (ver Tabla VII) podemos pensar con cautela que los estudiantes 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 estudiantes que no lo utilizaron, generando por tanto el uso de SRec un mayor grado de confianza en sí mismos. SRec da realimentación al estudiante, siendo uno de los factores [3] que aumenta la sensación de autoeficacia.

Como se puede observar en el diseño de la experiencia (sección III) no se realizaron medidas de autoeficacia en al inicio de la experiencia. Hay que tener en cuenta que la fuente con mayor repercusión e impacto en la autoeficacia es la experiencia previa [3]. Al inicio del experimento, esta experiencia previa es nula en los estudiantes para algunas de las tareas que desarrollan a lo largo de la experiencia. Debemos ser conscientes que los estudiantes no tienen conocimientos previos sobre el esquema algorítmico de divide y vencerás. Por tanto, el intentar medir el grado de autoconfianza que experimenta un sujeto sobre una actividad que no tiene ninguna experiencia previa puede ser poco representativo del estado real de ésta, y más bien pueda representar el grado de optimismo que pueda tener éste.

Además de este aspecto hay que tener en cuenta la complejidad de la tarea desarrollada por el estudiante. Las tareas de programación en las que trabaja el estudiante durante la experiencia tienen una complejidad intrínseca considerable. Sin duda alguna, esta complejidad intrínseca podría constituir una experiencia de gran impacto sobre la creencia que tiene el estudiante en sus propias capacidades, en la propia autoconfianza, que además, al ser tareas complejas para los estudiantes y, en general presentan más experiencias de fracaso que de éxito, harían que probablemente decreciera la autoconfianza. El éxito refuerza la creencia en nuestra capacidad personal mientras que situaciones de fracaso las debilita. SRec permite revertir esta situación, permitiendo encontrar los errores de forma inmediata y convertir este fracaso en futuro éxito.

Estos dos aspectos, el de la inexistencia de experiencia previa del estudiante y la complejidad intrínseca de la tarea, son variables que están igualmente presente en ambos grupos (control y experimental) y que afectarían a la hora de comparar las medidas de autoeficacia iniciales de la experiencia con las finales, no siendo representativa estas comparaciones para valorar la intervención de la herramienta y distorsionando el estudio.

Como dato adicional, señalamos que en una experiencia distinta, que hemos realizado en el año 2017, en la que los estudiantes no tenían ningún conocimiento previo sobre la tarea a realizar se midieron la autoeficacia al inicio de la experiencia y al final. En esta experiencia los resultados del postest de la autoeficacia disminuyeron respecto a los del pretest en ambos grupos (control y experimental), siendo esta experiencia en un dominio totalmente diferente, con estudiantes del Grado de Enfermería y sin intervención del uso de los computadores como elemento central de la tarea. Esto nos hace pensar que la medida de autoeficacia que percibe el sujeto ante una tarea que no tiene experiencia previa puede ser poco representativa, siendo mucho más

representativa de ésta la medida una vez que se tenga alguna experiencia.

Como conclusión final, y a tenor de los resultados obtenidos en esta experiencia, no podemos aceptar la hipótesis *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*”, por lo que no podemos afirmar que el uso de SRec mejora la eficacia del estudiante en el aprendizaje del esquema divide y vencerás, aunque sí se observa una tendencia de mejora en el aprendizaje de la recursividad.

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 [19].

También, como conclusión final, se puede aceptar la hipótesis *H2*: “*El sistema de visualización de programas SRec mejora la autoeficacia del estudiante en el contexto de uso y aplicación de procesos recursivos en estrategias de resolución de divide y vencerás*”. Por tanto, podemos afirmar que SRec mejora la autoeficacia del estudiante en el proceso de aprendizaje respecto a otras herramientas de programación como EDI NetBeans.

VI. CONCLUSIÓN

En este artículo hemos presentado una experiencia en el aula con estudiantes de las asignaturas de algoritmia en la que hemos utilizado dos herramientas de apoyo diferentes para realizar tareas de aprendizaje de recursividad y del esquema algorítmico divide y vencerás. Los estudiantes se han distribuido en dos grupos: Grupo de Control, donde se utilizó la herramienta EDI NetBeans, y Grupo Experimental, que trabajó con EDI NetBeans y con la herramienta de visualización y animación algorítmica SRec. Al inicio final de la experiencia se realizaron medidas del nivel de conocimiento de los estudiantes para medir la eficacia de aprendizaje de las herramientas, y al final de la misma, para poder comparar si hubo incremento. Para medir este conocimiento se diseñó una escala específica. Adicionalmente se realizó una medida de la autoeficacia del estudiante al finalizar la experiencia, tanto en un grupo como en el otro.

Como conclusión a partir de los resultados obtenidos no podemos afirmar que en la eficacia del aprendizaje haya habido una mejora con el uso de la herramienta de visualización algorítmica. Sin embargo, hemos encontrado una mejora estadísticamente significativa en la autoeficacia del estudiante cuando se usa la herramienta de visualización SRec respecto cuando se usa la herramienta de desarrollo profesional NetBeans.

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 30VCPIGI15 de la Universidad Rey Juan Carlos.

REFERENCIAS

- [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] Jedege, 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] Velázquez-Iturbide, J.A. & Pérez-Carrasco, A. (2016). How to use the SRec visualization system in programming and algorithm courses, *ACM Inroads*, vol. 7, no. 3, pp. 42-49.
- [17] 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.
- [18] 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.
- [19] 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.



Maximiliano Paredes Velasco es titulado por la Universidad de Castilla-La Mancha en Ingeniería Técnica en Informática de Sistema. En 1998 obtuvo el título de Ingeniero en Informática de la Universidad de Sevilla y se doctoró en Ingeniería Informática por la Universidad de Castilla-La Mancha en 2006. En 1998 comenzó a trabajar como profesor de la Universidad de Alcalá de Henares para pasar posteriormente al Departamento de Lenguajes y Sistemas Informáticos de la Universidad Rey Juan Carlos en 1999, donde desarrolla su labor como profesor actualmente. Su investigación se centra en aprendizaje colaborativo soportado por computador (CSCL) e Interacción Persona-Ordenador. Es autor de números artículos en congresos y revistas y participa en varios proyectos de investigación.



Isidoro Hernán Losada es titulado por la Universidad Complutense de Madrid en CC. Físicas especialidad de Calculo Automático. En 2012 obtuvo el título de Doctor en Informática y Modelización Matemática por la Universidad Rey Juan Carlos (URJC). Actualmente es profesor colaborador en la URJC. Su investigación se centra en la informática educativa. Es autor de numerosos artículos en congresos nacionales e internacionales y participa en varios proyectos de investigación.



J. Ángel Velázquez Iturbide recibió los títulos de Licenciado y de Doctor en Informática por la Universidad Politécnica de Madrid, España, en los años 1985 y 1990, respectivamente. Actualmente es Catedrático de Universidad en la Universidad Rey Juan Carlos, donde también es el Director del Laboratorio de Tecnologías de la Información para la Educación (LITE, <http://www.lite.etsii.urjc.es/>). Sus áreas de investigación son *software* y metodologías docentes para la enseñanza de la programación y la visualización del *software*. Actualmente, es Presidente de la Asociación para el Desarrollo de la Informática Educativa (ADIE). Es miembro senior de IEEE (IEEE *Computer Society* e IEEE *Education Society*) y de ACM (incluyendo ACM SIGCSE).



Carlos María Alcover es Licenciado y Doctor en Psicología Social por la Universidad Complutense de Madrid. Desde 1998 pertenece a la Universidad Rey Juan Carlos, Madrid, donde es catedrático de Psicología Social (Psicología del Trabajo y de las Organizaciones). Dirige el Equipo INPSITRO/SOC (Investigación en Psicología del Trabajo y las Organizaciones y en Psicología Social), y colabora de manera estable con equipos de investigación de universidades españolas, europeas y americanas en diversas líneas de investigación interdisciplinar. En la actualidad, su investigación se centra en el contrato psicológico y las relaciones de intercambio en las organizaciones; las dinámicas de cambio en los equipos de trabajo; y el papel desempeñado por las identidades y las pertenencias a múltiples grupos en la salud y el bienestar.