EIEI ACOFI 2025
5º Congreso Latinoamericano de Ingeniería
Más Información: www.acofi.edu.co/eiei
Ingeniería de impacto para cuidar y transformar la vida
ACOFI: 50 años por la excelencia de la educación en ingeniería.

Relación entre las percepciones y las dificultades del aprendizaje de la recursividad en estudiantes de ingeniería de sistemas y computación

Sergio Augusto Cardona Torres, Róbinson Pulgarín Giraldo, Sebastián Gaviria Valencia

Universidad del Quindío
Armenia, Colombia

Resumen

La recursividad se encuentra establecida como parte del cuerpo de conocimiento de disciplinas afines a las ciencias de la computación y la ingeniería de software. La importancia del uso de la recursividad es evidente, pues posee una estrecha relación con la programación de computadores, las estructuras de datos, las matemáticas discretas, la teoría de grafos, la inteligencia artificial y el análisis de algoritmos. La recursividad se presenta como uno de los temas de más difícil comprensión para los estudiantes que están desarrollando habilidades de algoritmia y programación. Desde hace algunos años se investiga este fenómeno para comprender las principales causas por las que resulta tan complejo la resolución de problemas utilizando esta estrategia de programación. En los antecedentes se identifican pocos estudios empíricos sobre la percepción propia de los estudiantes sobre la recursividad en combinación con datos que proporciona la resolución de problemas que proporciona una prueba de conocimiento. El presente trabajo tiene como objetivos (1) presentar el proceso de adaptación de un instrumento de percepciones del aprendizaje de la recursividad para el contexto colombiano y (2) analizar la relación entre los factores vinculados con las dificultades de los estudiantes sobre la programación recursiva. La adaptación del instrumento se soportó en un proceso de validación de contenido y constructo. Mediante un análisis factorial exploratorio y confirmatorio se verificó la propiedad psicométrica de validez de constructo. Para constatar la consistencia interna de los ítems del instrumento se realizó un análisis de fiabilidad basado en el coeficiente alfa de Cronbach. Los ítems del instrumento tuvieron discriminación positiva, por lo tanto, un alto grado de consistencia interna. La muestra final se constituyó de 155 estudiantes de un programa académico de ingeniería de sistemas y computación de una Universidad pública de Colombia. Los resultados de la estructura factorial sugieren que el modelo se ajusta a la fundamentación teórica y por tanto el instrumento se adapta adecuadamente al contexto. Se identificó que los estudiantes tienen una percepción favorable con relación a sus conocimientos sobre los diferentes factores vinculados a la

DOI: https://doi.org/10.26507/paper.4328

recursividad, sin embargo, los resultados de la prueba de conocimiento muestran un rendimiento académico moderado.

Palabras clave: aprendizaje; percepción; recursividad

Abstract

Recursion is established as part of the body of knowledge in disciplines related to computer science and software engineering. The importance of using recursion is evident, as it is closely related to computer programming, data structures, discrete mathematics, graph theory, artificial intelligence, and algorithm analysis. Recursion is one of the most difficult topics for students developing algorithmic and programming skills to understand. For several years, this phenomenon has been studied to understand the main reasons why problem-solving using this programming strategy is so complex. Previous research reveals few empirical studies that explore students' own perceptions of recursion in combination with data from problem-solving as assessed through a knowledge test. The objectives of this study are: (1) to present the process of adapting a recursion learning perception instrument for the Colombian context, and (2) to analyze the relationship between factors associated with students' difficulties in recursive programming. The adaptation of the instrument was based on a process of content and construct validation. Construct validity was verified through exploratory and confirmatory factor analysis. To assess the internal consistency of the items, a reliability analysis was performed using Cronbach's alpha coefficient. The items demonstrated positive discrimination, indicating a high degree of internal consistency. The final sample consisted of 155 students enrolled in a computer systems engineering program at a public university in Colombia. The results of the factorial structure suggest that the model fits the theoretical framework, and therefore the instrument is appropriately adapted to the context. It was found that students have a favorable perception regarding their knowledge of the different factors related to recursion; however, the knowledge test results show a moderate level of academic performance.

Keywords*: learning; perception; recursión*

1. Introducción

La recursividad se establece a nivel mundial como parte del currículo en planes de estudio de ciencias de la computación. La importancia del uso de la recursividad es evidente, pues posee una estrecha relación con la programación de computadores, las estructuras de datos, las matemáticas discretas, la teoría de grafos, la inteligencia artificial y el análisis de algoritmos. En currículos en ciencias de la computación, ingeniería de software e ingeniería informática de de IEEE y ACM, se contempla la recursividad como un concepto fundamental dentro su cuerpo de conocimiento. Se considera una estrategia de programación fundamentada en el principio de inducción matemática, la cual se utiliza para la resolución de diversos problemas computacionales tales como: teoría de juegos, problemas de optimización, problemas de búsquedas, algoritmos de ordenamiento, estrategias divide y vencerás, programación dinámica y algoritmos aplicado a grados. Los algoritmos recursivos o la recursividad (como se denomina a la técnica), es una estrategia de

programación en la cual se descompone un problema en varios subproblemas y posteriormente la solución se obtiene mediante la composición de los subproblemas. Desde el contexto educativo se ha identificado que la recursividad se presenta como uno de los temas de más difícil comprensión para los estudiantes que están aprendiendo programación (Zmuda & Hatch, 2007), (Ginat & Shifroni, 1999), (Gunion et al., 2009), (Sooriamurthi, 2001), (Mirolo, 2010) (Lacave et al., 2013). Desde hace algunos años se trabaja intentando tanto conocer las principales causas por las que resulta tan complicado su dominio, como también en proponer distintos métodos que ayuden a mitigar su comprensión y complejidad (Gunion et al., 2009). Esta situación ha generado el interés por realizar diferentes investigaciones para estudiar los posibles motivos de su entendimiento por parte de los estudiantes (Gunion et al., 2009), (Lacave et al., 2013). En este trabajo se presenta un estudio para analizar las posibles dificultades para aprendizaje de la recursividad en estudiantes de ingeniería de sistemas y computación de una Universidad pública de Colombia. El presente estudio empírico tiene alcance descriptivo, correlacional e inferencial.

2. Antecedentes

La recursividad es una técnica de programación se presenta como uno de los temas de más difícil comprensión para los estudiantes que están aprendiendo programación (Cardona & Jaramillo, 2015), (Zmuda & Hatch, 2007), (Gunion et al., 2009), (Mirolo, 2010), (Rinderknecht, 2014), y uno de los principales obstáculos que éstos tienen que enfrentar en los cursos de programación (Lacave et al., 2013). Este tema ha suscitado el interés por conocer las principales causas por las que resulta complicado su comprensión, por lo que desde hace décadas se trabaja intentando conocer las principales causas por las que resulta tan complicado dominarla (McCauley et al., 2015), así como proponer métodos que ayuden a mitigar su comprensión (Gunion et al., 2009) (Settle, 2014). Esta situación ha motivado el interés de la comunidad educativa por analizar el impacto de la recursividad a nivel curricular (Zmuda & Hatch, 2007), así como el esfuerzo por definir estrategias formativas orientadas a mejorar la enseñanza de la recursividad (Hsin, 2008). Se identifican estrategias basadas en juegos (Chaffin et al., 2009), de resolución de problemas (Wirth, 2008) y el uso de metáforas de delegación de tareas entre los estudiantes (Edgington, 2007). Se han desarrollado trabajos experimentales con estudiantes universitarios en cursos de programación (Scholtz & Sanders, 2010), (Zmuda & Hatch, 2007), (Quintana et al., 2016) y con estudiantes de nivel de educación secundaria (Gunion et al., 2009), cuyo propósito está orientado fundamentalmente a mejorar la comprensión de la recursividad.

Diferentes estudios (Sooriamurthi, 2001), (Gunion et al., 2009), (Mirolo, 2010) han sido orientados a identificar los problemas que tienen los estudiantes para la aplicación de la recursividad. Una de las tendencias actuales es utilizar técnicas de visualización, y a que a pesar de que no hay evidencias contundentes de que tenga como consecuencia una mejora en el proceso de aprendizaje, es un elemento motivador para el estudiante(Naps et al, 2003)(Hundhausen et al., 2002). Se han desarrollado herramientas educativas para la visualización del funcionamiento de los algoritmos recursivos (Pevac, 2012), (Terada, 2005), (Velázquez et al., 2009), (Perez, 2011), (Lacave et al., 2005), tutores interactivos diseñados para que los estudiantes visualicen el código (Hamouda et al., 2019) y sistemas de animaciones de la recursividad (Velázquez et al., 2008), videojuegos (Tessler et al., 2013), (Chaffin et al., 2009). Otros autores (George, 2000), (Dann et

al., 2001), han sugerido nuevos métodos para la enseñanza de la recursividad con el apoyo de recursos de software específicos. Se identifican estrategias para el aprendizaje basadas en juegos (Chaffin et al., 2009), (Tessler et al., 2013), la realización de actividades que aplican metáforas de delegación de tareas entre los estudiantes (Edgington, 2007) y la resolución de problemas reales que pueden ser resueltos mediante recursividad (Wirth, 2008).

Desde el punto de vista funcional se han identificado las dificultades para el entendimiento de la recursividad. En el trabajo expuesto en (Ginat & Shifroni, 1999), se identifica que la recursividad resulta un tema complejo pues los estudiantes no apropian una representación exacta del modelo computacional. En (Zmuda & Hatch, 2007) se demuestra empíricamente que los estudiantes no comprenden el retorno del flujo de control después de alcanzar la llamada al caso base y generalmente resuelven los problemas mediante técnicas de programación como las estructuras de control repetitivas. Según (Stephenson, 2009), los estudiantes encuentran estas estructuras repetitivas más intuitivas ya que están proporcionando instrucciones para el computador de una manera similar a cómo han recibido instrucciones durante toda su vida. En la investigación de (Sooriamurthi, 2001) se plantea que la dificultad de los estudiantes para entender el concepto de la recursividad radica en que no tienen una metodología adecuada para expresar una solución en términos recursivos. Los estudiantes aplican métodos de rastreo mecánicos y no comprenden el flujo de las funciones recursivas (Scholtz & Sanders, 2010). En la Tabla 1, se presentan trabajos que hacen explícita la forma en la cual se reportan las dificultades para el aprendizaje de los algoritmos recursivos.

Tabla 1. Trabajos que reportan dificultades en la recursividad

Referencia	Modificación de variables	Secuencia de las llamadas activas	Recuperación de variables	Parámetros en la vuelta atrás de la llamada activa	Paso de objetos como	Llamados recursivos	Seguimiento de la traza del algoritmo
(Settle, 2014)		X					X
(Lim, 2006)						X	
(Haberman & Averbuch, 2004)		X					X
(Zmuda & Hatch, 2007)	X				X		
(Scholtz & Sanders, 2010)		X					
(Sooriamurthi, 2001)	X			X			
(Stephenson, 2009)				X			
(Quintana et al., 2016)						X	X
(Lacave et al., 2013)			X	X	X		
(Hamouda et al., 2019)		X				X	
(Gotschi et al., 2005)	X		X				
(McCauley et al., 2015)	X			X			
(Perez, 2011)			X			X	X
(Sanders & Scholtz, 2012)	X			X			
(Tessler et al., 2013)		X			X		
(Chaffin et al., 2009)			X		X		
(Alzoubi et al., 2014)				X			X

Se identificó que la mayoría de los estudios se orientaban a medir conocimientos a partir de las respuestas con base a las soluciones de ejercicios. Solo en los trabajos de (Mirolo, 2010), (Lacave et al., 2013), se han aplicado cuestionarios orientados a medir las dificultades de los estudiantes.

3. Método

La muestra se constituyó de 155 estudiantes de un programa académico de ingeniería de sistemas y computación de una Universidad pública de Colombia. El 63.6% corresponde a hombres y el 36.4% corresponde a mujeres. La edad promedio de los estudiantes es 20.1 años. Para el presente estudio se consideran dos instrumentos. El cuestionario propuesto por (Lacave et al., 2013) con las secciones: (1) conocimientos de recursividad (8 ítems), (2) estrategias para solucionar problemas con el uso de la recursividad (6 ítems), (3) dificultades de la comprensión de la recursividad (8 ítems). Los ítems presentan opción de respuesta en escala de Likert de 1 a 5. El segundo instrumento consiste en una prueba con cinco problemas a resolver mediante el uso de la programación recursiva. Para la adaptación del instrumento se aplicó validez de contenido, validez de constructo y consistencia interna de los ítems. Para la validez de contenido se usó juicio de expertos mediante la valoración de 10 profesores universitarios (formación de doctorado (40%) y formación de maestría (60%)) que evaluaron los ítems en cuanto a su pertinencia (si/no). Para 10 expertos el modelo de Lawshe establece un índice de validez de contenido (CVR) de 0.62. Con base en los análisis se estableció eliminar del instrumento: tres ítems de la primera sección, un ítem de la segunda sección y un ítem para la tercera sección. Las preguntas definitivas del instrumento se presentan en los resultados.

Para la validez de constructo se aplicó análisis factorial (exploratorio y confirmatorio), con el propósito de identificar propiedades homogéneas de las variables independientes que se analizan. Para el análisis factorial exploratorio se verificaron los supuestos estadísticos y se analizaron los factores al interior de cada una de las dimensiones. Se realizó el análisis de los reactivos en el cual el estadístico Kaiser-Meyer-Olkin (KMO =0,773) sugiere que es factible la factorización de los datos originales. La prueba de esfericidad Bartlett ($p = 0.0001$) arrojó un valor significativo (χ^2= 1585, gl=120, $p<.05$). El coeficiente del determinante de la matriz de correlaciones fue 0.000001. Para la distribución de factores se aplicó el análisis de componentes principales. Las cargas factoriales fueron examinadas mediante la rotación Varimax. El método de extracción de componentes principales generó tres factores. El primer factor está constituido por variables asociadas a las dificultades relacionadas con la comprensión de la recursividad, el segundo está asociado a variables relacionadas con conocimientos de recursividad y el tercer factor está relacionado con estrategias que usa para solucionar problemas que impliquen la aplicación de la recursividad. La estructura de los componentes del análisis factorial explica el 54,1% de la variabilidad total. El análisis factorial confirmatorio se analizó por estimación de máxima verosimilitud. Los resultados del modelo (SBX2 = 190.121, gl = 109, $p < .001$; CFI =. 945, TLI = .937; RMSEA = .059, IC 90% [.044, .071]), SRMR = .062) y la fiabilidad global calculada a través del Alfa de Cronbach (α= .729, IC 95% [.705-.673]) y Omega de Mc Donald (ω= .786, IC 95% [.688-.806]), sugieren que los datos se ajustan a la fundamentación teoría. Con lo cual se puede afirmar que el instrumento se adapta adecuadamente al contexto.

4. Resultados

4.1 Dificultades sobre el aprendizaje de la recursividad

Para la sección de comprensión de los fundamentos de la recursividad, se identificó que un 72,7% valoró favorablemente el gusto la programación. El 60% de los estudiantes afirmó conocer los principios teóricos de la recursividad. El 41,9%, afirmó no comprender esta técnica de programación. Un 45,5% consideró que la recursividad no es útil para resolver problemas complejos de programación. Solo el 25,5% de los estudiantes afirmó que les cuesta dificultad diseñar algoritmos recursivos. En la Tabla 2 se muestran los descriptivos de la comprensión de la recursividad.

Tabla 2. Comprensión de la recursividad

Pregunta	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)
p1. Me gusta la programación de computadores.	1,8	3,6	21,8	41,8	30,9
p2. Conozco los principios teóricos de la recursividad.	3.6	14,5	21,8	45,5	14,5
p3. Comprendo el funcionamiento de la recursividad.	1,8	10,9	29,1	43,6	14,5
p4. Considero útil la recursión para solucionar problemas.	0,0	7,3	38,2	29,1	25,5
p5. Me cuesta trabajo diseñar algoritmos recursivos.	1,8	20,0	2,7	16,4	9,1

Para la categoría estrategias empleadas para resolver problemas mediante la recursividad, se identificó que un 52,8% de los estudiantes considera necesario que el profesor debe resolver más ejercicios. El 72,7% consideró fundamental que el profesor represente de forma visual la traza de los algoritmos. Un 65,4% piensa que es fundamental realizar la traza de los algoritmos recursivos para una mejor comprensión de su funcionamiento. Un 67.2% de los estudiantes considera que una herramienta para la visualización puede ser una estrategia importante para resolver problemas mediante la recursividad y para mejorar su comprensión. En la Tabla 3, se presentan las frecuencias relacionadas con las estrategias para resolver problemas con recursividad.

Tabla 3. Estrategias para resolver problemas

Pregunta	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)
p6. Más ejercicios resueltos por el profesor en el curso.	0,0	7,3	40,0	25,5	27,3
p7. Uso de material sugerido por el profesor del curso.	0,0	14,5	40,0	32,7	12,7
p8. Visualización de la traza de los algoritmos recursivos.	1,8	3,6	21,8	43,6	29,1
p9. Realización de las trazas de los algoritmos recursivos.	1,8	3,6	29,1	43,6	21,8
p10. Herramienta de visualización de algoritmos recursivos.	3,6	3,6	25,5	23,6	43,6

Para las dificultades de comprensión sobre el funcionamiento de la recursividad. Un 21,9% de los estudiantes afirma tener problemas para entender la llegada al caso base. Un 87,2% responde que no tiene inconveniente para entender el acceso y la modificación de los parámetros. Un 29,1% afirmó tener dificultades para comprender el flujo de secuencia de las llamadas activas en ejecución. El 32,7% considera complejo entender cómo se recuperan los valores de las variables y de los parámetros en la vuelta atrás. Un 25,4% afirmó tener dificultad para gestionar el paso de objetos como parámetros dentro del algoritmo. El 45,5% consideró complejo la llamada a varios métodos

recursivas. El 38,2% identificó limitantes para hacer seguimiento a la traza del algoritmo en ejecución. En la Tabla 4 se muestra las dificultades de la comprensión de los algoritmos.

Tabla 4. Dificultades en la comprensión de los algoritmos recursivos

Pregunta	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)
p11. La llegada al caso base desde el caso inductivo.	3,6	34,5	40,0	16,4	5,5
p12. Modificación de parámetros y variables de métodos.	12,7	40,0	34,5	5,5	7,3
p13. Flujo de secuencia de las llamadas activas.	3,6	25,5	41,8	20,0	9,1
p14. Recuperación de los valores de variables y parámetros.	3,6	18,2	45,5	21,8	10,9
p15. Paso de objetos como parámetros.	7,3	36,4	30,9	21,8	3,6
p16. Funcionamiento de varias llamadas recursivas.	1,8	12,7	40,0	29,1	16,4
p17. Seguimiento de la traza de un algoritmo recursivo.	3,6	10,9	47,3	36,4	1,8

Para el análisis correlacional se verificó el supuesto de comportamiento normal mediante la prueba de Kolmogorov-Smirnov. El nivel de significancia para todas las variables analizadas tiene un p-valor < 0.05 (no existe distribución normal), por lo cual se aplicó la prueba no paramétrica de Rho de Spearman. En la Tabla 5 se presenta la matriz de correlaciones.

Tabla 5. Matriz de correlacionales

	p1	p2	p3	p8	p9	p12	p13	p14	p16
p1	1.00	.390**	.355**	.401**	.204*	.255**	.292**	.387**	.045
p2		1.00	.435**	.339**	.248**	.337**	.351**	.622**	.850**
p3			1.00.	.420**	.005	.101	.078	.480**	.037
p8				1.00	.266**	.293**	.248**	.303**	.161
p9					1.00	.663**	.812**	.512**	.221*
p12						1.00	.848**	.659**	.755**
p13							1.00	.683**	.839**
p14								1.00	.352**
p16									1.00.

De las 17 variables analizadas, se identificó que las variables p4, p5, p6, p7, p10, p11, p15 y p17 no presentan nivel de significancia (p_valor<0.05), que puedan resultar significativos para analizar con todas las otras variables del estudio. Los estudiantes que conocen los principios teóricos de la recursividad tienen relación directa con la dificultad que tienen de comprender el funcionamiento del programa cuando existen varias llamadas recursivas. La dificultad de comprender el acceso y modificación de parámetros y variables de métodos tiene relación directa con entender el funcionamiento del programa cuando existen varias llamadas recursivas. La dificultad que tienen los estudiantes para comprender el flujo de secuencia de las llamadas activas tiene correlación positiva alta con la dificultad que ellos tienen para comprender el funcionamiento del programa cuando existen varias llamadas recursivas.

4.2 Desempeño académico

El análisis de los factores se realizó en función de la incidencia en el desempeño académico de los estudiantes. Mediante la prueba de Shapiro-Wilk, se verificó la normalidad del conjunto de datos

de la variable rendimiento académico (agrupada en dos valores [aprobó – no aprobó]) con las variables. El p-valor>0.05 para todos los casos, confirmó la normalidad del conjunto de datos. Se aplicó un análisis de varianza ANOVA con p-valor>0.05 para todas las variables. En la Tabla 6 se presenta la prueba de hipótesis entre la variable desempeño académico y componente.

Tabla 6. Prueba de hipótesis variable – desempeño

Variable	Componente del instrumento	ANOVA
		Sig.
Rendimiento académico	Dificultades	,050
	Conocimiento	,673
	Estrategias	,413

Con base en el nivel de significancia se identifica que no existe diferencia estadísticamente significativa entre el rendimiento académico con el componente conocimiento de la recursividad y con las estrategias que aplica para aprender recursividad. Se identificó diferencia estadísticamente significativa de acuerdo con los resultados de las dificultades para el aprendizaje de la recursividad. Se identifica que los estudiantes que menor valoraron las dificultades presentan un menor rendimiento académico, con relación a los estudiantes que resaltaron la complejidad de la recursividad.

5. Conclusiones

La revisión del estado del arte permitió identificar las diferentes estrategias formativas desarrolladas para contribuir al aprendizaje de la programación recursiva. Un análisis de los estudios muestra que no en todos ellos se han planteado investigaciones formales que permitan conocer la opinión de los estudiantes sobre la recursividad y su posible relación con variables que pueden incidir en su rendimiento académico. Con la realización de este estudio se conocen variables de incidencia del aprendizaje de la recursividad. Con lo cual se amplía en el marco de referencia y se complementan los trabajos realizados en (Mirolo, 2010), (Lacave et al., 2013).

Se identifica que los aspectos más problemáticos son los relacionados con la recursividad son: la vuelta atrás, la recuperación de valores en las llamadas activas, así como el funcionamiento de la recursividad en los casos en los que existen múltiples llamadas recursivas. Los estudiantes resaltan que la visualización de la traza de los algoritmos recursivos puede contribuir a un mejor entendimiento de la recursividad, lo cual está acorde con lo reportado en (Zmuda & Hatch, 2007) en el cual se demostró que los estudiantes no comprenden el retorno del flujo de control después de alcanzar la llamada al caso base, es decir no tienen dominio sobre el flujo de la secuencia ni la incidencia de las llamadas activas. Se confirma lo realizado en investigaciones previas (Scholtz & Sanders, 2010), (Stephenson, 2009), en las cuales los estudiantes aplican métodos de seguimientos mecánicos en las cuales no comprenden el flujo de las funciones recursivas.

En correspondencia con lo que se afirma en (Tessler et al., 2013), (Sanders & Scholtz, 2012), (Lacave et al., 2013), la recursividad es una herramienta muy importante y potente para la solución de problemas complejos en computación. Sin embargo, aunque los alumnos consideran que

entienden los principios teóricos y el funcionamiento de la recursividad, la experiencia llevada a cabo permite afirmar que las dificultades para su aprendizaje son evidentes. Se ha realizado un estudio que combina las dificultades objetivas (mediante ejercicios) con las percepciones subjetivas sobre los aspectos que más dificultad le suponen. Estas preguntas directas a los estudiantes sobre las dificultades del aprendizaje de la recursividad, es el elemento distintivo de nuestro estudio, frente a otros identificados en la literatura.

Para la validez de contenido, el juicio de experto permite afirmar que el instrumento refleja el dominio de conocimiento específico. La matriz de componentes rotados generó los factores: dificultades del aprendizaje de la recursividad, comprensión de los aspectos teóricos de la recursividad y estrategias para el aprendizaje de la recursividad. Estos tres componentes están alineados con el constructo teórico objeto de estudio.

Con relación a los resultados de los estudiantes en los descriptivos, las opiniones favorables de los estudiantes sobre el conocimiento y comprensión de la recursividad están en dirección opuesta a los resultados de la resolución de los ejercicios planteados. Solo un 12,7%% de los estudiantes resolvieron todos los ejercicios planteados y solo el 21.8% resolvió correctamente 4 ejercicios, lo que permite concluir que a los estudiantes les cuesta resolver problemas en lo que tienen que aplicar la recursividad. En los descriptivos se destacan las opiniones de los estudiantes sobre las dificultades para el aprendizaje de la recursividad. Un porcentaje muy bajo de los estudiantes afirmaron tener problemas para comprender la llegada al caso base desde el caso general, lo cual está en correspondencia con los resultados de los tres primeros ejercicios planteados. Solo 12,8% de los estudiantes afirman no tener inconveniente para entender el acceso y la modificación de los parámetros, y las variables de método, lo cual también es coherente con los resultados pues la actividad 3, tiene un porcentaje de 76,4% de respuestas correctas por parte de los estudiantes. Solo un 29,1% afirma tener dificultades para comprender el flujo de secuencia de las llamadas activas cuando el algoritmo está en ejecución, lo cual está en correspondencia con los resultados de los estudiantes. Un 72,2% respondió correctamente la actividad 2 y un 67,3% respondió bien la actividad 1. Para la actividad 4 solo el 47,3%, la resolvió adecuadamente y para la actividad 5 solo un 34,4%.

Con relación al rendimiento académico, se identifica que esta variable tiene relación directa cuando el estudiante realiza adecuadamente seguimiento a la traza del algoritmo. Así mismo, la variable rendimiento académico mostró correlación negativa cuando los estudiantes los estudiantes deben recuperar los valores de las variables y de los parámetros en la vuelta atrás de la recursividad. Lo anterior confirma los resultados de las actividades pues se evidenció qua muchos estudiantes se les dificulta obtener los valores de las variables en la vuelta atrás.

Como escenario de trabajo futuro, es necesario realizar investigaciones empíricas en las cuales se implementen actividades que permitan hacer seguimiento continuo al aprendizaje de los estudiantes, buscando que sigan con mayor precisión los indicadores para establecer los logros, los aspectos por mejorar y el nivel de desempeño alcanzado en diferentes actividades de aprendizaje de la recursividad.

6. Referencias

- Alzoubi, O., Fossati, D., Eugenio, B., & Green, N. (2014). ChiQat-Tutor: An Integrated Environment for Learning Recursion. *Second Workshop on AI-Supported Education for Computer Science*, 1–4.
- Cardona, S., & Jaramillo, S. (2015). Dificultades en el aprendizaje de la programación recursiva. *Revista de Investigación IngEam*, *2*(2), 55–63.
- Chaffin, A., Doran, K., Hicks, D., & Barnes, T. (2009). Experimental evaluation of teaching recursion in a video game. *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, *1*(212), 79–86. https://doi.org/10.1145/1581073.1581086
- Dann, W., Cooper, S., & Pausch, R. (2001). Using Visualization To Teach Novices Recursion. *ACM SIGCSE Bulletin*, *33*(3), 109–112. https://doi.org/10.1145/507758.377507
- Edgington, J. (2007). Teaching and viewing recursion as delegation. *Computing Sciences in Colleges*, *23*(1), 241–246.
- George, C. E. (2000). EROSI - Visualising Recursion and Discovering New Errors. *ACM SIGCSE Bulletin*, *32*(1), 305–309. https://doi.org/10.1145/331795.331875
- Ginat, D., & Shifroni, E. (1999). Teaching Recursion in a Procedural Environment How much should we emphasize the Computing Model ? *ACM SIGCSE Bulletin*, *31*(1), 127–131. https://doi.org/10.1145/384266.299718
- Gotschi, T., Sanders, I., & Galpin, V. (2005). Mental Models of Recursion. *Journal of Educational Computing Research*, *14*(1), 1–23. https://doi.org/10.2190/AGG9-A5UD-DEK0-80EN
- Gunion, K., Mildford, T., & Stege, U. (2009). Curing Recursion Aversion. *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, 124–128. https://doi.org/10.1145/1562877.1562919
- Haberman, B., & Averbuch, H. (2004). The case of base cases. *ACM SIGCSE Bulletin*, *34*(3), 84. https://doi.org/10.1145/637610.544441
- Hamouda, S., Edwards, S. H., Elmongui, H. G., Ernst, J. V., & Shaffer, C. A. (2019). Recurtutor: an interactive tutorial for learning recursion. *ACM Transactions on Computing Education*, *19*(1), 1–25. https://doi.org/10.1145/3218328
- Hsin, W. (2008). Teaching recursion using recursion graphs. *Journal of Computing Sciences in Colleges*, *23*(4), 217–222.
- Hundhausen, C., Douglas, S., & Stasko, J. (2002). A Meta-Study of Algorithm Visualization Effectiveness. *Journal of Visual Languages and Computing*, *13*, 259–290. https://doi.org/10.1006/jvlc.2002.0237
- Lacave, C., Molina, A. I., & Giralt, J. (2013). Identificando algunas causas del fracaso en el aprendizaje de la recursividad : Análisis experimental en las asignaturas de programación. *XIX Jornadas Sobre La Enseñanza Universitaria de La Informática (Jenui 2013)*, 1–8.
- Lacave, C., Paredes Velasco, M., Velazquez Iturbide, J. A., & Hernán Losada, I. (2005). Experiencia para la evaluación de VisBack, una herramienta para la visualización de algoritmos de backtracking. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa*, *26*, 82–93. https://dialnet.unirioja.es/servlet/articulo?codigo=6231886
- Lim, N. (2006). Supplementing lectures on recursive C programming with an interactive learning environment. *7th International Conference on Information Technology Based Higher Education and Training, ITHET*, 236–244. https://doi.org/10.1109/ITHET.2006.339769
- McCauley, R., Grissom, S., Fitzgerald, S., & Murphy, L. (2015). Teaching and learning recursive programming: a review of the research literature. *Computer Science Education*, *25*(1), 37–66. https://doi.org/10.1080/08993408.2015.1033205
- Mirolo, C. (2010). Learning (through) recursion : a multidimensional analysis of the competences achieved by cs1 students. *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, 160–164.

ttps://doi.org/10.1145/1822090.1822136
- Naps et al, T. (2003). Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin*, *35*(2), 131–152. https://doi.org/10.1145/782941.782998
- Perez, A. (2011). *Sistema Generador de Animaciones Interactivas para la Docencia de Algoritmos Recursivos*. Universidad Rey Juan Carlos.
- Pevac, I. (2012). First experiences with tutor for recursive algorithm time efficiency analysis. *Journal of Computing Sciences in Colleges*, *28*(1), 56–65.
- Quintana, M., Sagula, J. E., & Monzón, F. I. (2016). *Estrategias de enseñanza de funciones recursivas en ciencias de la computación*.
- Rinderknecht, C. (2014). A Survey on teaching and learning recursive programming. *Informatics in Education*, *13*(1), 87–119. https://doi.org/10.15388/infedu.2014.06
- Sanders, I., & Scholtz, T. (2012). First year students' understanding of the flow of control in recursive algorithms. *African Journal of Research in Mathematics, Science and Technology Education*, *16*(3), 348–362. https://doi.org/10.1080/10288457.2012.10740750
- Scholtz, T., & Sanders, I. (2010). Mental Models of Recursion : Investigating Students ' Understanding of Recursion. *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, 103–107. https://doi.org/10.1145/1822090.1822120
- Settle, A. (2014). *What' s Motivation Got to Do with It? A Survey of Recursion in the Computing Education Literature* (Vol. 23). http://via.library.depaul.edu/tr/23
- Sooriamurthi, R. (2001). Problems in comprehending recursion and suggested solutions. *ACM SIGCSE Bulletin*, *33*(3), 25–28. https://doi.org/10.1145/507758.377458
- Stephenson, B. (2009). Using graphical examples to motivate the study of recursion. *Using Graphical Examples to Motivate the Study of Recursion*, *25*(1), 42–50.
- Terada, M. (2005). ETV : a Program Trace Player for Students. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 118–122. https://doi.org/10.1145/1067445.1067480
- Tessler, J., Beth, B., & Lin, C. (2013). Using Cargo-Bot to Provide Contextualized Learning of Recursion. *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 161–168. https://doi.org/10.1145/2493394.2493411
- Velázquez, J. Á., Pérez, A., & Urquiza, J. (2008). SRec : An Animation System of Recursion for Algorithm Courses. *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, 225–229. https://doi.org/10.1145/1384271.1384332
- Velázquez, J. Á., Pérez, A., & Urquiza, J. (2009). A Design of Automatic Visualizations for Divide-and-Conquer Algorithms. *Electronic Notes in Theoretical Computer Science*, *224*, 159–167. https://doi.org/10.1016/j.entcs.2008.12.060
- Wirth, M. (2008). Introducing Recursion by Parking Cars. *ACM SIGCSE Bulletin*, *40*(4), 52–55. https://doi.org/10.1145/1473195.1473219
- Zmuda, M., & Hatch, M. (2007). Scheduling topics for improved student comprehension of recursion. *Computers and Education*, *48*(2), 318–328. https://doi.org/10.1016/j.compedu.2005.02.003

- **Sobre los autores**

- **Sergio Augusto Cardona Torres**: Ingeniero de Sistemas, Magister en Ingeniería, Doctor en Ingeniería - Universidad Pontifica Bolivariana. Profesor titular. sergio_cardona@uniquindio.edu.co
- **Róbinson Pulgarín Giraldo**: Ingeniero de Sistemas, Magister en Ingeniería – Universidad EAFIT. Profesor Asociado. robinsonpg@uniquindio.edu.co

- **Sebastián Gaviria Valencia**: Ingeniero Mecánico, Máster en Análisis y Visualización de Datos, Universidad Internacional de la Rioja. Profesor Auxiliar. sgaviria@uniquindio.edu.co

---

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2025 Asociación Colombiana de Facultades de Ingeniería (ACOFI)