

# PROGRAMACIÓN POR IMITACIÓN. APROXIMACIÓN AL APRENDIZAJE EN INGENIERÍA DE SISTEMAS

PROGRAMMING BY IMITATION: A LEARNING APPROACH TO SYSTEMS ENGINEERING

Omar Iván Trejos Buriticá \*

<https://orcid.org/0000-0002-3751-6014>

Luis Eduardo Muñoz Guerrero \*\*

<http://orcid.org/0000-0002-9414-6187>

UNIVERSIDAD TECNOLÓGICA DE PEREIRA, COLOMBIA

Recibido: 10 de septiembre de 2019

Aceptado: 20 de diciembre de 2019

DOI: <https://doi.org/10.29097/2011-639X.257>

## Resumen

El presente artículo expone los resultados comparativos y paralelos de un estudio realizado en la carrera de Ingeniería de Sistemas con grupos simultáneos de programación (por semestre). Con un grupo se adoptó el aprendizaje de códigos ya publicados y en funcionamiento; con el otro, se aplicó la metodología tradicional de enseñanza de la programación mediante exposición magistral. Este estudio se realizó con dieciséis grupos de programación durante los años 2016, 2017, 2018 y 2019. La metodología se ajustó a un estudio comparativo de caso, de manera que en cada semestre se analizaron los resultados del trabajo con códigos ya escritos frente a los resultados de grupos que hicieron lo correspondiente, pero con códigos escritos por los mismos alumnos. Los cursos paralelos fueron de temáticas diferentes. Los resultados cuantitativos permitieron que se hicieran inferencias cualitativas en relación con el aprendizaje a través de la observación y la interacción directa con los estudiantes. Se concluye que cuando el estudiante cuenta con las bases apropiadas, el apoyo que le brindan programas debidamente codificados pareciera acelerar y solidificar el proceso de aprendizaje con un conocimiento con sentido, significado y facilidades de conectividad.

**Palabras clave:** aprendizaje, imitación, ingeniería de sistemas, nuevas tecnologías, programación, significado.

## Abstract

This article describes the parallel comparative results of a study conducted in the Systems Engineering degree with simultaneous Programming groups (per semester). With one group, learning was adopted from codes already published and in operation; with the other, the traditional methodology of teaching programming through lectures was applied. This study was implemented with 16 programming groups during the years 2016, 2017, 2018 and 2019. The methodology was adjusted to a comparative case study,

---

\* PhD en Ciencias de la Educación, Docente Titular de Planta Universidad Tecnológica de Pereira, Investigador Senior Colciencias, miembro del Grupo de Investigación en Informática.

✉ [omartrejos@utp.edu.co](mailto:omartrejos@utp.edu.co)

\*\* PhD en Ciencia de la Educación, Docente de Planta Universidad Tecnológica de Pereira, miembro del Grupo de Investigación en Informática.

so that in each semester the results of the work with codes already written were analyzed in comparison with the results of groups that did the same but with codes written by the same students. The parallel courses were of different subjects. The quantitative results allowed qualitative inferences to be made in relation to learning through observation and direct interaction with students. The conclusion is that when the student has the appropriate knowledge base, the support provided by properly codified programs seems to accelerate and solidify the learning process with meaning, relevance and connectivity facilities.

**Keywords:** anthocyanins, anthocyanin extract, extraction, blackberry, ultrasound.

## INTRODUCCIÓN

Investigar los propósitos trazados desde lo curricular en relación con los objetivos de aprendizaje en una asignatura integrada a un plan de estudios universitarios corresponde a una de las labores más necesarias en el perfil del docente universitario de hoy (Díaz y Hernández, 2002). La búsqueda de nuevos caminos para asimilar la lógica computacional con la que se programan los computadores es uno de los propósitos pretendidos en la educación superior, sabiendo de antemano que no existe un camino excelso para asimilarla (Trejos, 2012), pero que es posible construir uno más expedito para aprender a programar cuando se combinan diferentes estrategias basadas en teorías y modelos de aprendizaje que propenden a transformar la base cognitiva del estudiante en favor de su papel en la escena académica.

El problema para resolver se resume en la búsqueda de una estrategia para simplificar el modo que permita aprender a programar a estudiantes de primer y segundo semestre de Ingeniería de Sistemas en una universidad pública. Esta vez, la mirada investigativa se ha posado en el aprendizaje de la programación de computadores por el camino de la imitación, desde la inmersión bibliográfica y la ejemplificación tecnológica moderna que puede obtenerse de la consulta de fuentes basadas en las nuevas tecnologías de información y comunicación.

El propósito de esta investigación radica en realizar un experimento investigativo con grupos paralelos, de manera que se puedan comparar los resultados de aprendizaje de un grupo, tras emplear la estrategia de la programación por imitación, con los resultados de otro, utilizando la estrategia tradicional de enseñanza con exposición magistral y codificación por iniciativa del estudiante.

La novedad destacable de esta investigación consiste en que pocas veces se realiza en el aula, en asignaturas tecnológicas dentro de un programa de Ingeniería de Sistemas de manera comparativa, con la metodología tradicional de enseñanza con exposición magistral. Igualmente, se está acudiendo a un camino de aprendizaje aún inexplorado que permitiría aprovechar esa fuente de consultas casi inagotable como es el internet, en donde los estudiantes encontrarían códigos que, en más de una oportunidad, les pueden iluminar el camino de la lógica de programación para la implementación de soluciones a problemas conceptuales o técnicos de la sociedad. También es novedoso

que se incluya al ambiente académico la teoría del conectivismo (Aznawarian, 2017) con participación en tiempo real; es decir, propiciando las condiciones para que el estudiante acceda a fuentes de conocimiento virtuales, allende lo real.

Justifica la presente investigación el cada vez más necesario acceso a nuevas fuentes de información, el despliegue y la penetración de las nuevas tecnologías de la información y la comunicación, así como las capacidades de interacción que han desarrollado los jóvenes de hoy, bien llamados nativos digitales (Small, 2009). Se suma el hecho de que la irrupción de dispositivos electrónicos destinados a la conectividad está presente de manera permanente en las aulas de clase de todas las universidades, independientemente de que sean públicas o privadas (MinTic, 2018).

El objetivo de la investigación que inspira el presente artículo consistió en determinar si la adopción de la teoría de aprendizaje por imitación, dentro del contexto de las primeras asignaturas de programación de computadores en «Ingeniería de Sistemas y Computación», facilita y simplifica el aprendizaje con sentido y significado de un determinado paradigma de programación, o si, por el contrario, dicha teoría es irrelevante dentro del mismo contexto.

Para la concepción y el desarrollo del presente artículo se acudió a una bibliografía especializada en las teorías de aprendizaje, que conforman el corpus de esta propuesta, y a los elementos de orden tecnológico, específicamente a la luz del paradigma funcional e imperativo en los lenguajes de programación DrRacket (Felleisen, et al., 2005) y C++, respectivamente (Trejos, 2017). El estudio llegó a recolectar los datos cuantitativos, compararlos con la percepción cualitativa, tabular y organizar los datos, así como a analizarlos desde una perspectiva metodológica científica.

### MARCO TEÓRICO

Aprender implica aproximarse a un conjunto de conceptos, teorías, modelos, estrategias y prácticas para apropiarse de ellos: evaluarlos, aplicarlos, asimilarlos, realimentarlos y afinarlos, de forma que se cambie la base cognitiva que ha fundamentado el conocimiento previo, buscando que se modifique, perdure y sea flexible, según cambien las condiciones en donde puede utilizarse (Bruner, 2009). Aprender programación de computadores implica asimilar, como propia, una lógica que, por ser diferente de la lógica deliberativa humana, es la que permite capitalizar todas las ventajas de las tecnologías computacionales modernas a través de lenguajes de programación (Basogain, Olabe, M y Olabe J., 2015). Estos se resumen en conjuntos de instrucciones que los computadores interpretan y ejecutan como órdenes según la secuencia indicada por un programador.

El aprendizaje de la programación de computadores se ha ido convirtiendo poco a poco en una necesidad en el mundo moderno, dado que con dicho aprendizaje emerge el concepto de pensamiento computacional (Denning y Tedré, 2019), cuyo objetivo consiste en desarrollar sistemáticamente las habilidades de pensamiento crítico y resolución de problemas con base en los conceptos de la computación. Estos implican

estrategias metodológicas paramétricas para resolver situaciones problema, el aprovechamiento de las nuevas tecnologías de información y comunicación, y la capacidad de contemplar, desde una perspectiva evaluativa, las ventajas y desventajas que proveen las dos primeras implicaciones.

El pensamiento computacional se convierte, en tiempos modernos, en un insumo de alta importancia si se quieren abordar y, eventualmente, resolver las dificultades que surgen en la enseñanza y, especialmente, en el aprendizaje de la programación de computadores (Wing, 2006), pues de esta forma se articula el aprovechamiento de las TIC (Tecnologías de la Información y la Comunicación) con las necesidades y soluciones del mundo moderno. Tal es el propósito de la iniciativa mundial *Code.Org*, que busca socializar estos conceptos en todos los procesos de la formación básica secundaria y en los programas de formación profesional universitaria en el mundo.

Vale la pena tener en cuenta que la aproximación al pensamiento crítico permite que el estudiante evalúe el conocimiento en relación con un marco de referencia, de manera que pueda confrontarlo, revisarlo, evaluarlo y realimentarlo para bien de sus expectativas de aprendizaje y de la evolución de los conceptos teóricos y prácticos que surjan en el campo académico en beneficio de la sociedad. De la misma manera, el aprovechamiento de las nuevas tecnologías de la información y la comunicación facilita establecer un diálogo con los jóvenes en los términos de su lenguaje natural moderno (*sc.*, tecnológico) sobre las posibilidades de conectividad e interacción con otras personas y sus servicios asociados. Con estos elementos se puede pensar que el estudiante planea soluciones algorítmicas realizables, por eficiencia, en sistemas electrónicos computacionales dentro de un marco de referencia que atienda necesidades y resuelva problemas de la sociedad (Trejos, 2017).

Una de las estrategias que hasta el momento se explora es el aprendizaje por imitación (Osa y Pajarinen, 2018), según el cual el cerebro aprende —con significado, sentido y mayor facilidad— aquello que se aproxima a su objeto de estudio y que ya ha sido cristalizado, realizado, resuelto, practicado, evaluado, monitoreado o realimentado con anterioridad en unas condiciones similares, análogas o relativas a aquellas en las que se requiere aplicar dicho conocimiento (Ausubel, 2012). La estrategia de imitación ha sido exitosa en diferentes áreas tales como la música, la pintura, la escultura y, especialmente, en aquellas expresiones que las bellas artes proveen y en las que se toma como base una obra elaborada y difundida ante un determinado público para realizar análisis, estudios e inferencias. Así se pueden heredar de ellas sus características principales, de forma que los artistas puedan aplicar dichos conceptos en sus propias producciones.

El aprendizaje por imitación también ha sido importante y útil en áreas como la lectura de obras literarias; el análisis de guiones de cine, libros y cuentos; la redacción de cuentos, poemas, historias y ensayos, descripción de estilos literarios y escritura de artículos científicos, entre algunas de sus áreas más significativas. Allí los resultados han sido notorios y en donde el estilo, la pluma o el talento ha emergido del análisis y apropiación de lo que los precursores ya habían realizado, y de quienes se han inferido los conceptos pertinentes. No es un concepto nuevo, pues durante el Renacimiento

fueron muchos los artistas dedicados a duplicar las obras de los más virtuosos (Rey y Pastor, 2013) hasta el punto de convertirse en unos ilustres anónimos, ya que sus obras, plagios basados en el análisis de los estilos de los grandes maestros, hoy se exhiben en diferentes museos del mundo, con la misma prestancia y significatividad que las obras originales (Boyer, 2010). Vale la pena tener en cuenta que, en tiempos modernos más que en cualesquiera otros, el sentido y el significado del conocimiento cobran mayor relevancia dado que, por su misma naturaleza, el cerebro descarta todo aquello que, siendo conocimiento, no le es útil ni importante, tampoco los utiliza con una frecuencia que lo haga relevante. Es aquí cuando emerge el gran reto del docente en la actualidad: encontrar el sentido y significado de aquello que constituya el corpus de sus syllabus, y se conviertan en conocimiento relevante para el cerebro (Trejos, 2000).

Conviene recordar que el conductismo, el cognitivismo y el constructivismo son las tres grandes teorías de aprendizaje utilizadas con mayor frecuencia en la forja de ambientes instruccionales. Fueron planteadas, revisadas, aplicadas y refinadas en épocas cuando los entornos de aprendizaje no estaban permeados por las nuevas tecnologías, que, en su más simple interpretación, han reestructurado la manera como nos comunicamos y relacionamos en la sociedad moderna (Trejos, 2017). Esto ha afectado profundamente la manera como aprendemos y transformamos la base cognitiva de nuestros saberes mediante la interacción, con la dinámica de las redes sociales, de los dispositivos electrónicos, de las pantallas y de los servicios derivados. Cabe anotar que las teorías que finalmente describen los procesos de aprendizaje deben reflejar los ambientes sociales sobre los cuales se fundamentan (Coelho, et al, 2018).

En el conectivismo se acepta, por primera vez, que el conocimiento ya no radica en el cerebro, sino en dispositivos que lo almacenan, difunden, socializan y posibilitan su acceso (Campillay y Meléndez, 2015), que existe una necesidad estudiantil para que los docentes los ayuden a crear competencias donde encuentren dicho conocimiento, en ese océano de información, por momentos estéril, llamado internet. De manera que, de paso, encuentren (o le confieran) tal sentido y significado, que dichos dispositivos y los servicios que proveen se conviertan en la gran herramienta de aprendizaje del mundo moderno para capitalizar y aprovechar las capacidades del ser humano en su relación con otras personas y dentro de una sociedad a la que se debe.

**Tabla 0**

Teorías que inspiran el presente artículo

Ítem	Teoría o modelo	Aplicación
1	El concepto de aprendizaje	Es la base de la presente investigación, pues se pretende mejorar, por caminos más expeditos, el proceso de aprendizaje y efectivizarlo en el tiempo, en su adquisición y su perdurabilidad

2	Teoría de aprendizaje por imitación	Constituye la teoría base sobre la cual se inspira el presente artículo, y los detalles de su aplicación en esta investigación se explican en el ítem metodología
3	Aprendizaje de la programación de computadores	Es el corpus objeto de investigación, el aprendizaje y contenido de la asignatura sobre la que se realizó esa investigación
4	El sentido y significado de un proceso de aprendizaje	Concepto heredado de la teoría de aprendizaje significativo que cristaliza, hoy más que nunca, el conocimiento en favor de su perdurabilidad y hace efectivo el concepto de competencia
5	El pensamiento computacional	Es la base teórica que el mundo de hoy exige en términos de la apropiación del conocimiento tecnológico en favor de los procesos de formación tanto en básica secundaria como en programas de Ingeniería
6	Teoría del Conectivismo (George Siemens)	Constituye la otra teoría de aprendizaje formulada hace muy poco, que plantea elementos de juicio nuevos en la relación de los estudiantes con las nuevas tecnologías y el papel de estas dentro de un contexto de conocimiento

*Nota:* La tabla presenta la relación de los antecedentes científicos y las teorías que los fundamentan, también la manera como se han utilizado para el desarrollo de esta investigación y del artículo presentado.

## METODOLOGÍA

El estudio se realizó durante los períodos académicos comprendidos entre el segundo semestre de 2016, primer y segundo semestre de 2017, primer y segundo semestre de 2018 y primer semestre de 2019 en los cursos de Programación I y Programación II del programa «Ingeniería de Sistemas y Computación» de la Universidad Tecnológica de Pereira. Con el ánimo de conseguir mayor objetividad a los resultados obtenidos, se tomaron como base dos cursos en paralelo de Programación I (cuyo contenido versa sobre el paradigma de programación funcional desde la perspectiva del lenguaje DrRacket) y otros dos cursos de Programación II (cuyo contenido es el paradigma de programación imperativa desde la perspectiva de la arista estructurada del lenguaje C++ entorno DevC++). Se realizó de esta manera para que los resultados de un curso, por su contenido, no permearan —y eventualmente malogaran— los resultados que arrojara el mismo estudio en el otro curso.

Con uno de los grupos se realizaron los trabajos y talleres en clase, aprovechando las TIC, y con el otro se realizaron la exposición magistral y el desarrollo de la asignatura en los términos y bajo las estrategias tradicionales. Se recurrió a esta forma de trabajo para que se pudieran realizar análisis comparativos *in situ*, en tiempo real, y de manera que se observaran los posibles aportes de la teoría de aprendizaje por imitación, a la luz de los contenidos de asignaturas de programación de computadores.

En el plan de estudios de «Ingeniería de Sistemas y Computación» de la Universidad Tecnológica de Pereira estos son los dos cursos donde el estudiante tiene su primer contacto con el amplio mundo de la programación que, según el planteamiento curricular, corresponde a la espina dorsal del programa desde el enfoque que se le ha querido conferir en el fortalecimiento de las competencias para el desarrollo de *software*.

La investigación se ha realizado con tres estrategias básicamente: a) entregar a los estudiantes programas ya realizados que logran objetivos establecidos para socializar antes de conocer la solución y de los que se cita el enlace de descarga por parte del docente de la asignatura, b) plantear enunciados por resolver, con los respectivos paradigmas y lenguajes de programación, y entregando a los estudiantes el enlace respectivo para que descarguen, estudien y analicen una posible solución disponible, debidamente documentada, y c) construir enunciados, de forma que los estudiantes tengan que analizar la posible solución y buscarla por internet, con ayuda de un buscador; que esté bien escrita (según los lineamientos del paradigma y del lenguaje de programación), bien documentada, estéticamente presentada, y que cumpla plenamente y de manera satisfactoria con el objetivo.

Estas tres estrategias se han utilizado en el orden presentado en el párrafo anterior y con la frecuencia posibilitada por las tres sesiones semanales de clase.

La primera estrategia se utiliza como preparación para la primera evaluación parcial, que se realiza en la cuarta semana y que tiene un valor del 20 % de toda la asignatura. La segunda estrategia se utiliza como preparación para la segunda evaluación parcial, que se realiza en la octava semana y que tiene un valor del 20 % de toda la asignatura. La tercera estrategia se aprovecha como preparación para la tercera evaluación parcial, que se realiza en la duodécima semana, y cuyo valor también es del 20 % de toda la asignatura. De esta forma solo quedaría pendiente un 40 % de la nota total que se distribuye de la siguiente forma: 30 % que corresponde al valor de la evaluación final y un 10 % motivacional, que le confiere a cada estudiante una nota de 5.0 con el ánimo de generar en ellos una disposición de trabajo desde el primer día de clases. Las evaluaciones parciales se realizaron siempre de manera escrita, y en condiciones de forma y fondo similares para ambos grupos paralelos, según lo establecido por el respectivo paradigma.

Se distribuyeron las estrategias cada cuatro semanas para que, en cada una de ellas, el estudiante lograra tres objetivos: articulación con el pensamiento computacional en lo que corresponde al enfoque crítico, aprovechamiento de las TIC y resolución algorítmica de problemas. En otras palabras: mediante la primera estrategia (de la primera semana a la cuarta) se busca que el estudiante fortalezca su relación natural con

las TIC como base para la resolución de problemas. A través de la segunda estrategia (de la quinta semana a la octava) se pretende que el estudiante se arriesgue a encontrar, con algo de ayuda, la solución algorítmica necesaria, que entienda un buen código y que le sirva como base de imitación para escribir sus propios códigos. Y con la tercera estrategia (de la novena semana a la duodécima) se busca que el estudiante indague, por sus propios medios, sobre las soluciones requeridas, las estudie, analice, imite y construya, con ellas, sus propias soluciones algorítmicas.

Para realizar todo el proceso se usaron dos caminos de comunicación con los estudiantes: el servicio de correo electrónico y el servicio de WhatsApp. En ambos casos se creó un grupo con los correos de cada uno de los estudiantes y otro con sus números de celular para tener una vía de comunicación fluida y expedita entre docentes y estudiantes.

La investigación es cuantitativa por los datos obtenidos que se expondrán en el ítem de resultados, y es también cualitativa, pues, al final de la experiencia, se abre el espacio para que los estudiantes manifiesten de manera anónima y libre si consideran que aprendieron a programar o no dentro de los cánones del paradigma respectivo. Esta información se contrastó con la observación y análisis del docente, quien estuvo atento al desarrollo de los talleres realizados con las técnicas expuestas durante todo el semestre.

Se llegó hasta el punto de determinar si la experiencia fue exitosa o no, considerando el hecho de que la opinión del docente y la del estudiante coincidieran según el desempeño y los resultados obtenidos. Sabiendo que en el primer semestre existen varios grupos de Programación I, y que en el segundo semestre siempre se planea otra buena cantidad de grupos de Programación II, se escogieron al azar, de la manera más aleatoria posible, un grupo de Programación I y un grupo de Programación II, con el ánimo de que fueran los datos recolectados los que, al final de la experiencia, hablaran por sí mismos.

Por esto se consideró la extensión del estudio durante seis semestres (desde el segundo semestre de 2016 hasta el primer semestre de 2019). En cuanto a las preguntas abiertas que se plantearon al finalizar la experiencia, y que nutrieron el análisis cualitativo, debe tenerse en cuenta que fueron totalmente anónimas, libres y espontáneas. Los estudiantes estuvieron solos en el aula y tuvieron una franquicia absolutamente abierta para responder todo lo que quisieran, en la extensión y con los términos que consideraran apropiados, sin restricciones de ningún tipo.

## RESULTADOS

La tabla 1 presenta la relación cuantitativa de estudiantes participantes en la investigación que inspira el presente artículo:

**Tabla 1**

Cantidad de estudiantes participantes

<b>Año</b>	<b>Semestre</b>	<b>Grupo ProgTrad</b>	<b>Grupo ProgImit</b>	<b>Cant Est</b>
2016	II	23	21	44
2017	I	20	21	41
	II	17	19	36
2018	I	19	18	37
	II	21	20	41
2019	I	20	22	42
Total		120	121	251

*Nota:* Tabla de elaboración propia.

ProgTrad: grupo de programación con metodología de aprendizaje tradicional

ProgImit: grupo de programación con metodología de aprendizaje por imitación

Las tablas 2a y 2b presentan el promedio de los resultados (para facilitar los análisis) de las evaluaciones escritas en los tres parciales durante el semestre.

**Tabla 2a**

Resultados cuantitativos – Grupos ProgTrad

<b>Año</b>	<b>Sem</b>	<b>I P</b>	<b>II P</b>	<b>III P</b>	<b>Prom</b>
2016	II	3,5	3,3	3,4	3,4
2017	I	3,4	3,2	3,1	3,2
	II	3,5	3,4	3,6	3,5
2018	I	3,3	3,2	3,3	3,3

	II	3,1	3,3	3,2	3,2
2019	I	3,4	3,4	3,2	3,3
Promedios		3,4	3,3	3,3	3,3

Nota: Tabla de elaboración propia.

**Tabla 2b**

Resultados cuantitativos – Grupos ProgImit

Año	Sem	I P	II P	III P	Prom
2016	II	4,2	4,1	4,3	4,2
2017	I	4,4	4,3	4,4	4,4
	II	4,2	4,2	4,3	4,2
2018	I	4,3	4,2	4,5	4,3
	II	4,4	4,3	4,1	4,3
2019	I	4,3	4,2	4,1	4,2
Promedios		4,3	4,2	4,3	4,3

Nota: Tabla de elaboración propia.

Las tablas 3a y 3b presentan el resultado de la encuesta breve realizada al final del curso en la cual se les hicieron tres preguntas a los estudiantes para conocer su percepción final de la experiencia, tanto para aquellos que recibieron el contenido por los métodos tradicionales, como para aquellos otros que lo recibieron a través de la estrategia de aprendizaje por imitación.

**Tabla 3a**

Resultados cualitativos – Opinión de estudiantes – ProgTrad

Opinión	Sí	%	No	%	Q Total	% Total
¿Considera que aprendió a programar en el curso?	180	71 %	71	29 %	251	100 %

¿Adquirió fundamentos para construir soluciones algorítmicas?	191	76 %	60	24 %	251	100 %
¿Aprendió a aprovechar académicamente las TIC?	211	84 %	40	16 %	251	100 %

Nota: Tabla de elaboración propia.  
Q = cantidad de estudiantes

**Tabla 3b**

Resultados cualitativos – Opinión de estudiantes – ProgÍmit

Opinión	Sí	%	No	%	Q Total	% Total
¿Considera que aprendió a programar en el curso?	249	99 %	2	1 %	251	100 %
¿Adquirió fundamentos para construir soluciones algorítmicas?	247	98 %	4	2 %	251	100 %
¿Aprendió a aprovechar académicamente las TIC?	250	99 %	1	1 %	251	100 %

Nota: Tabla de elaboración propia.  
Q = cantidad de estudiantes

Cabe anotar que las respuestas abiertas hechas por los estudiantes, aprovechando la franqueza que se les concedió de mantenerse como anónimos, libres y espontáneos, corresponden a un análisis posterior que se incluye en otro artículo de los mismos autores.

## ANÁLISIS DE RESULTADOS Y DISCUSIÓN

Tal como se muestra en la tabla 1 se puede observar que, a lo largo del período designado para realizar la investigación correspondiente a seis semestres, la muestra seleccionada es significativa frente a la población de estudiantes de primer y de segundo semestre en ingeniería de sistemas donde se realizó dicha investigación. Se intentó mantener una cantidad aproximada de estudiantes por semestre de forma que se diera, desde la perspectiva cuantitativa, una aproximación objetiva de los resultados.

Es de anotar que, en cada semestre, la cantidad de estudiantes que participó en la investigación correspondió, por lo menos, a un 20 % como máximo, de la población total de estudiantes y a un 15 % como mínimo, hecho que, estadísticamente, deja entrever una muestra seleccionada con los estándares que para estos fines se establece desde lo paramétrico. Lo mismo sucede con los cursos de Programación II, que fueron seleccionados alternadamente para el desarrollo de la investigación. Estos elementos

técnicos permiten garantizar un nivel de confianza en los resultados a través de los cuales se pueden extrapolar las inferencias obtenidas a toda la población que cumpla con las características seleccionadas (estudiantes de primer o segundo semestre).

Las tablas 2a y 2b presentan los resultados cuantitativos de los promedios de los parciales en los dos cursos paralelos (el curso donde se implementó la asignatura, acudiendo a métodos tradicionales, y el curso en donde se implementó el aprendizaje de la programación por imitación). Se observa que en los resultados del grupo con metodología tradicional el promedio de todas las pruebas estuvo por debajo de 0,1 o fue igual o superior a esa cifra, y se mantuvo por debajo de 3,5. El promedio de promedios en cada semestre solamente tuvo un rendimiento académico de 3,5 en el segundo semestre de 2017. El resto fueron inferiores a esta cota. No ha de olvidarse que el promedio es una medida de tendencia central que se afecta por los valores extremos, pero en este caso se ha escogido como métrica de análisis, precisamente debido a eso, pues los resultados cuantitativos indican que, si bien hubo valores superiores al promedio, la mayoría de los valores fueron cercanos (con tendencia a estar por debajo) del promedio.

Por su parte la tabla 2b presenta unos resultados cuantitativos que en todo momento son superiores a 4,0, lo cual advierte unos valores con tendencia a sobrepasar dicha referencia numérica. Los promedios de los tres parciales siempre estuvieron por encima de 4,0 tal como lo presenta el resumen de resultados. Es de anotar que las evaluaciones escritas siempre fueron similares en la forma, aunque, por razones completamente entendibles, con diferencias en el fondo. De la misma forma se procuró que dichas evaluaciones se realizaran en salones contiguos, en condiciones similares de fecha, hora y lugar, de manera que ningún factor adicional pudiera malograr la objetividad de los resultados. Las tablas 3a y 3b presentan la parte cuantitativa de los análisis cualitativos que, por razones de la extensión del presente documento se exponen en otro artículo. Estas tablas presentan los resultados de tres preguntas que se les hicieron a los estudiantes al finalizar la experiencia investigativa, y cuyas respuestas contaban con completa libertad y toda la espontaneidad requerida. Esta información se recolectó de forma anónima y sin ningún tipo de identificación. Los docentes al frente del proceso de investigación dejaron a los estudiantes solos, mientras respondían las preguntas y comentaban lo que consideraran pertinente. Se designó a un estudiante para que recogiera las hojas de respuestas y las entregara en la oficina del docente encargado del curso.

En cuanto al curso en donde se adoptaron metodologías tradicionales de enseñanza y aprendizaje de la programación, se observa una aceptación del 71 %, 76 % y 84 % respectivamente y en referencia a cada una de las respuestas con las cuales el estudiante diligenció las preguntas postuladas. Por su parte, los resultados con preguntas similares en el curso con estrategia de aprendizaje de la programación por imitación arrojaron porcentajes abrumadores (99 %, 98 % y 99 %) sobre las mismas preguntas, lo que permite, por lo menos, inferir que los estudiantes sienten mucha más confianza en sí mismos cuando se les abren espacios de interacción y consulta académica apoyada en su lenguaje natural, como es el uso de las nuevas tecnologías de información y comunicación. No ha de descuidarse que estos porcentajes también están respaldados

por el pensamiento computacional, su difusión, su socialización con los estudiantes y aplicación dentro del contexto del aprendizaje de la programación de computadores (paradigma funcional y paradigma imperativo) en cursos iniciales de Ingeniería de Sistemas en una universidad pública que corresponde al perfil de la experiencia realizada.

### CONCLUSIÓN

Con la metodología adoptada se ha logrado lo principal en un curso de programación, independientemente del paradigma adoptado como contenido: el logro de fortalecer el proceso de aprendizaje y apropiación del conocimiento, según muestran los resultados expuestos en las tablas correspondientes. Se comprueba que, para los jóvenes de hoy, la estrategia de aprendizaje por imitación es una forma innovadora, motivadora y muy asequible para aprender programación. Gracias a esta estrategia se le ha conferido sentido y significado al conocimiento que se hereda de la programación de computadores en lo tocante al paradigma de programación funcional y al paradigma de programación imperativa.

El pensamiento computacional, llevado por los caminos metodológicos con un buen fundamento, ha permitido, en esta experiencia investigativa, presentarse como uno de los puentes entre las expectativas de los estudiantes que ingresan a «Ingeniería de Sistemas y Computación» como programa de formación universitaria, y la teoría que fundamenta su corpus de conocimientos. De manera que es posible lograr un pensamiento crítico, un aprovechamiento de las TIC y una aproximación a la algoritmización de las soluciones que han tenido como marco referencial el aprendizaje por imitación.

Vale la pena, desde la óptica de la labor docente en programas de formación en ingeniería, estudiar las diferentes vertientes y posibilidades que ofrece la teoría del conectivismo y profundizar en ellas, pues al aceptar, por primera vez, que gran parte del conocimiento ya no está en el ser humano, sino en una red de dispositivos y servicios, se extiende una invitación para que el docente acepte el reto de pensar en el aprovechamiento de las TIC en favor de los logros planteados como objetivos de un curso de programación de computadores. Se ha cumplido con la metodología que la investigación científica exige y se puede inferir que en un curso inicial de programación de computadores es totalmente conveniente la adopción de la teoría de aprendizaje por imitación, como inicio para la transformación de la base cognitiva de los estudiantes que están comenzando un proceso de formación universitaria en ingeniería de sistemas.

En relación con el objetivo de la presente investigación, y tomando como referencia los resultados obtenidos, puede concluirse que, efectivamente, la adopción de la teoría de aprendizaje por imitación dentro del contexto de las primeras asignaturas de programación de computadores en ingeniería de sistemas facilita y simplifica el camino de aprendizaje con el sentido y significado de un determinado paradigma; siempre y cuando se haga bajo las condiciones y con base en las características implementadas, tal como se ha realizado en la presente investigación.

## INVESTIGACIÓN Y FINANCIAMIENTO

Este estudio se realizó en la Universidad Tecnológica de Pereira, programa de «Ingeniería de sistemas y computación», en los cursos Programación I y Programación II (cuyos contenidos corresponden a la programación funcional y programación imperativa, respectivamente, durante los períodos académicos comprendidos entre el segundo semestre de 2016, primero y segundo semestre de 2017, primero y segundo semestre de 2018 y primer semestre de 2019. El presente documento es un producto del proyecto de investigación titulado *Desarrollo de un modelo metodológico para las asignaturas de Programación en Ingeniería de Sistemas basado en la teoría de aprendizaje significativo y el modelo 4Q de preferencias de pensamiento*, tramitado ante la Vicerrectoría de Investigación, Innovación y Extensión de la Universidad Tecnológica de Pereira, y avalado por esa dependencia.

## Referencias

- Ausubel, D. (2012). *The Acquisition and Retention of Knowledge*. Springer.
- Aznawarian, L. (2017). *Conectivismo: una teoría del aprendizaje en la era digital*. Editorial Académica Española.
- Basogain, X., Olabe, M., y Olabe, J. (septiembre de 2015). Pensamiento computacional a través de la programación: paradigma de aprendizaje. *Revista de Educación a Distancia*, 46(6), 2-35.
- Boyer, C. (2010). *Historia de la Matemática*. Madrid, España: Alianza Editorial.
- Bruner, J. (2009). *Actos de significado*. Alianza Editorial.
- Campillay, S., y Meléndez, S. (abril de 2015). Análisis de impacto de metodología activa y aprendizaje heurístico en asignaturas de Ingeniería. *Actualidades Investigativas en Educación*, 15(2).
- Coelho, M., et al (2018). *Teorias da aprendizagem: o conectivismo presente na modalidade de ensino semipresencial*. Novas Edicoes Académicas.
- Denning, P., y Tedré, M. (2019). *Computational Thinking*. The MIT Press.
- Diaz, F., y Hernández, G. (2002). *Estrategias docentes para un aprendizaje significativo*. McGraw Hill.
- Felleisen, M., et al. (2005). *How to Design Programs*. MIT Press.
- Ministerio de Tecnologías de la Información y Comunicaciones. (2018). *Informe de Gestión Año 2018*. MinTIC.
- Osa, T., y Pajarinen, J. (2018). *An Algorithmic Perspective on Imitation Learning*. Now Publishers.

- Rey, J., y Pastor, J. (2013). *Historia de la Matemática* Vol. II. Gedisa.
- Small, G. (2009). *Digital Brain*. Urano.
- Trejos, O. (2000). *La esencia de la lógica de programación*. Centro Editorial Universidad de Caldas.
- Trejos, O. (2012). *Significado y competencias*. Editorial Papiro.
- Trejos, O. (2017). *Lógica de programación*. Ediciones de la U.
- Trejos, O. (febrero de 2017). Metodología para aprender programación funcional en Ingeniería de Sistemas, aplicando teoría de aprendizaje por descubrimiento. *Revista Educación en Ingeniería*, 12(23), 69-75.
- Trejos, O. (2017). *Programación imperativa con lenguaje C*. ECOE Ediciones.
- Wing, J. (Marzo de 2006). Computational Thinking. *Communications on the ACM*, 49(3), 33-35.