

Episteme vs. Tekné: una aproximación al aprendizaje de la programación de computadores en ingeniería de sistemas

Omar Iván Trejos-Buriticá

Facultad de Ingenierías, Universidad Tecnológica de Pereira, Pereira, Colombia. omartrejos@utp.edu.co

Resumen— El presente artículo expone los resultados de una investigación realizada en el aula cuyo propósito consistía en comparar dos estrategias de enseñanza y aprendizaje de la programación de computadores en Ingeniería de Sistemas. Las dos estrategias a comparar son: la enseñanza conceptual de la lógica de programación (episteme) y la enseñanza tecnológica a través de un lenguaje de programación (tekné). Para medir el impacto de las dos estrategias en el aprendizaje de la programación se acudió a dos grupos diferentes de Programación I. Con cada grupo se desarrolló la misma temática, pero desde una perspectiva diferente. Los resultados se midieron desde lo cuantitativo a partir de evaluaciones escritas tal que su formulación fuera similar en los dos grupos. También se realizó una aproximación cualitativa a partir de la opinión de los estudiantes al final del curso. Se concluye que pareciera ser más relevante la enseñanza de la *episteme* de la programación que de su *tekné* puesto que el mismo estudiante puede inferir mucho yendo de aquella a éste. Como caso especial, también se encontró que algunos estudiantes, a partir del estudio del lenguaje de programación, pudieron inferir los conceptos subyacentes a la lógica computacional.

Palabras Clave— aprendizaje; computadores; episteme; programación; tekné.

Recibido: 26 de enero de 2019. Revisado: 20 de marzo de 2019. Aceptado: 24 de abril de 2019.

Episteme vs. Tekné: an approximation for learning computer programming in engineering programs

Abstract— This article exposes the results of a research made in the classroom looking for establish elements of comparative judgment that would allow comparing two teaching strategies, and their impact on learning, of computer programming in a Systems Engineering program. The two strategies to be compared consist in the teaching of the logic of programming (episteme) or the teaching of its application reflected in a programming language (tekné). In order to measure the impact of the two strategies on the learning of programming we did with two Programming I courses. The process was carried out with the 1st semester of Systems Engineering of a public university. The results were measured quantitatively from identical written evaluations and under highly similar conditions for their resolution. A qualitative approach was also made based on the opinion of the students at the end of the course. It is concluded that the teaching of the episteme of programming seems to be much more relevant than of its application since the student can go from the 1st to the other. However, there were cases in which some students, from the study of the programming language, were able to infer the underlying concepts of computational logic.

Keywords— computers; episteme; learning; programming; tekné.

1. Introducción

Una de las inquietudes más recurrentes en el proceso de enseñanza y aprendizaje de la programación de computadores

incluye la discusión acerca de la conveniencia de enfatizar más en la parte puramente conceptual y su fundamento teórico – matemático o llevar al estudiante directamente al computador y articularlo, desde el principio con un lenguaje de programación, con el ánimo de que el mismo descubra los elementos teóricos que le subyacen [1]. Surge esta inquietud debido a que un docente ingeniero está capacitado, por su formación disciplinar, para ser ingeniero y pocas veces (...) está capacitado para ser docente [2] y la programación (de computadores) es parte esencial e integral de cualquier programa de ingeniería [3].

En cualquier otro tiempo la respuesta pudiera ser más fácil de encontrar dado que, veinte años atrás o más, no se contaba con los medios tecnológicos ni con el acceso a ellos tan simple, sencillo y masivo como se cuenta hoy. La aparición de los computadores personales y otros dispositivos, la accesibilidad y el cambio de mentalidad que permitió que cada hogar tuviera en la actualidad, por lo menos, un computador (o intentara tenerlo) adicional a la aparición de cibercafés, o café-Internet, ha puesto a la generación actual de estudiantes de la programación en una situación en la cual encontrar la respuesta invita procesos investigativos [4] como el que se expone en este artículo que sugieren la necesidad de que el tutor reflexione en torno a su práctica [5].

La inquietud fundamental gira en torno al hecho de pensar que, a partir del aprendizaje y apropiación de los conceptos fundamentales que subyacen a los lenguajes de programación, es decir, la lógica de programación, se pueda llegar a tener la proficiencia requerida para entender y utilizar los diferentes lenguajes de programación que existen en el mercado informático de los desarrolladores de software, en contraposición con la inquietud que surge si al acompañar al estudiante en un proceso de aprendizaje profundo del manejo de un lenguaje de programación es posible que él llegue a inferir los conceptos teóricos que subyacen a la lógica de programación y, por lo tanto, pueda llegar a extrapolarlos hacia otros lenguajes o hacia otros paradigmas de programación.

Debe pensarse en diseñar una metodología que permita llegar a conclusiones que, sin ser absolutas, pueden orientar en cuanto a la relevancia que, en tiempos modernos, tiene la

Como citar este artículo: Trejos-Buriticá, O.I., Episteme vs. Tekné: una aproximación al aprendizaje de la programación de computadores en ingeniería de sistemas Educación en Ingeniería, 14(28), pp. 39-44, Marzo - Julio de 2019.

enseñanza y su impacto en el aprendizaje de la programación de computadores cuando se desarrollan procesos de formación profesional con estudiantes que son nativos digitales [6]. En la enseñanza de las matemáticas, por ejemplo, para una mejora en el aprendizaje debe apartarse de la abstracción y acercarse a los cursos (...) concretos, por ejemplo, basado en matrices [7].

El propósito de esta investigación estriba en aportar elementos de juicio que le permitan a los docentes de programación de computadores, de formación básica secundaria o formación superior, enfatizar más en lo puramente conceptual (*episteme*), reflejado en la lógica de programación y en su fundamentación matemática, o en lo enteramente aplicativo (*tekné*) reflejado en un lenguaje de programación desde donde el estudiante pudiera inferir los conceptos que subyacen [8] para que, de esta forma, se aprovechen las características de los nativos digitales [9] y se pueda avanzar mucho más en los procesos de apropiación, asimilación, aplicación, retroalimentación, validación y evaluación del conocimiento [10].

La novedad del presente artículo consiste en que la pregunta ha sido recurrente, motivo de discusión en las salas de docentes de programación, pero pocos estudios se han realizado desde la perspectiva de la investigación educativa tal que se puedan realizar inferencias que conduzcan a conclusiones científicas y menos intuitivas dado que una clase de ingeniería depende, entre otros, de factores como el ambiente, los recursos, las estrategias didácticas y los mecanismos de evaluación para un logro efectivo de los objetivos de formación [11].

Esta investigación se justifica porque tanto en los colegios y universidades se considera, per se, que la enseñanza de la programación de computadores, y sus procesos de aprendizaje asociados, requieren (sin discusión) de la utilización e incorporación de computadores, de lenguajes de programación, de compiladores y de ambientes integrados (IDE por sus siglas en inglés) que posibiliten el aprendizaje [12] olvidando que las oportunidades de mejora para seguir brindando calidad en la educación a los estudiantes constituye uno de los compromisos que adquiere una universidad socialmente responsable [13].

Para el desarrollo de esta investigación, además del trabajo de campo, también se contó con bibliografía especializada tanto en lo teórico investigativo, en lo práctico y en lo metodológico además de la participación de diferentes cursos a partir de los cuales se obtuvieron los resultados y se realizaron las inferencias pertinentes teniendo en cuenta que los individuos que acogen dichos procesos deben permanecer en continuo aprendizaje [14] para poder articular habilidades y cualidades para que se (...) complementen sus habilidades en el logro de diferentes objetivos [15]. Esta investigación es producto del proyecto de investigación código 6-16-13 autorizado por la Vicerrectoría de Investigaciones, Extensión e Innovación de la Universidad Tecnológica de Pereira. La investigación se realizó con cursos de 1er semestre de Ingeniería de Sistemas durante los años 2016, 2017 y 2018 en la asignatura Programación I.

Para los efectos descriptivos necesarios, el presente artículo está organizado de acuerdo al estándar IMRYD [16] que comienza con una Introducción, continúa con unos elementos teóricos para pasar a una metodología, posteriormente se

exponen los resultados obtenidos para finalizar en una discusión de la cual se llega a unas conclusiones. En la última parte se relacionan las referencias bibliográficas que sirvieron de soporte para el desarrollo tanto del artículo como de la investigación.

2. Marco teórico

La esencia del conocimiento se ha contrapuesto, desde épocas inmemoriales cuando brillaba la filosofía griega como faro iluminador del pensamiento universal, a la aplicación práctica del mismo [17]. La discusión se ha centrado, desde hace más de veinte siglos, en la pugna entre el camino de buscar dicha esencia y su apropiación o aferrarse a la aplicación del mismo y, a partir de allí, encontrar los elementos teóricos que le subyacen por la vía de la razón heredada de los procesos aplicativos del saber [18]. No ha sido una discusión nueva y por ello los antiguos griegos llamaron a la esencia del conocimiento la *episteme* y a la aplicación del mismo la *tekné* que puso en escena dos corrientes que, con el tiempo, han rivalizado en ese anhelo del ser humano de apropiar y asimilar el conocimiento en la búsqueda de mejorar como seres evolucionados [19].

Esta discusión ha trascendido diferentes épocas del conocimiento, así como diferentes áreas tal como se reflejó en el siglo VIII cuando los árabes difunden los fundamentos del álgebra [20], en el siglo XVI cuando se plantean las bases de la geometría analítica [21], en el siglo XVII cuando se propone el cálculo diferencial [22] y en el siglo XIX cuando la matemática toma nuevos rumbos posibles aún no descubiertos [23]. En estos ejemplos la discusión siempre giraba alrededor de la necesidad de definir si eran más importantes los fundamentos del conocimiento matemático o las situaciones de la vida del ser humano donde éstos eran aplicables.

Hacia la mitad del siglo XX aparecen los computadores como las herramientas que permiten realizar tareas que para el ser humano son posibles, pero altamente agotadoras y, por ello, altamente proclives a generar errores tanto en los resultados como en la interpretación [24]. Las computadoras básicamente permiten realizar grandes cálculos con un margen de error establecido por los cánones que dictan las matemáticas y no por la naturaleza misma de los seres humanos. Eso hizo pensar que era posible llegar a incluir las computadoras en una inmensa cantidad de tareas que, en sus albores, eran todavía inimaginables.

A partir de la necesidad de encontrar nuevas aplicaciones para los computadores se desarrolla la lógica de programación como fundamento matemático para capitalizar al máximo sus capacidades de procesamiento y cálculo [25]. Al mismo tiempo se comienzan a desarrollar los lenguajes de programación que son el vehículo para que se instrumentalicen las soluciones que la matemática provee usando esa nueva lógica computacional requerida.

Las aplicaciones de los computadores, de la lógica de programación y de los lenguajes de programación fueron cada vez más numerosas y necesarias para la sociedad y entonces se empezó a especializar el conocimiento en esta materia al punto de llegar a configurar una nueva profesión que en principio fue un nuevo oficio: el de programador de computadores [26]. Fue

allí en donde apareció, de nuevo, la discusión acerca de lo que necesitaba aprender el programador como base para sus posteriores desarrollos.

De una parte estaban los que defendían el aprendizaje de la lógica de programación representada en sus fundamentos matemáticos pensando en que si se conoce la esencia del conocimiento es posible articularse con sus posibles aplicaciones y desarrollos y, de otra parte, estaban los que defendían el aprendizaje de los lenguajes de programación dado que era una nueva era, una nueva sociedad y un nuevo ser humano y se consideraba posible, y aún se considera, que por el camino del aprendizaje de la *tekne* se puede llegar a los fundamentos que caracterizan su *episteme* [27].

Nunca se pensó, eso sí, que los computadores llegarían a formar parte de la vida cotidiana de los seres humanos al punto que aquellas personas que han nacido en medio de la tecnología se conocen como nativos digitales, como sucede en el presente siglo XXI y para quienes ésta se ha convertido en su lenguaje natural. Todos aquellos que conocieron el mundo sin la alta penetración tecnológica, se conocen como inmigrantes digitales a raíz de la gran influencia que en el día de hoy se experimenta [28] y de su gran incidencia a través de redes sociales en la toma de decisiones y en la manipulación de diferentes momentos de su vida.

El cerebro de los jóvenes de hoy, y de aquellos que no son tan jóvenes, está permanentemente bombardeado por pantallas [29] que en todo momento están trayendo información, bien bajo los parámetros que provee la objetividad o bien como producto, y este es tal vez el más común, de la subjetividad que persigue determinados intereses tanto en lo social, en lo político, en lo comercial, en lo económico e, incluso, en la esfera puramente personal.

A partir de estos elementos de juicio, la investigación en educación está llamada a intentar buscar respuestas a la inquietud inicial considerando, eso sí, las características de los jóvenes de hoy, las bondades que las nuevas tecnologías han incorporado a sus vidas y las limitaciones que ellas mismas proveen dentro de un contexto de interacción social que se mueve más por canales digitales que por canales directos de interrelación personal [30]. Eso es lo que se persigue en esta investigación: intentar encontrar respuestas, o por lo menos poder tener acceso a elementos de juicio, que posibiliten una reflexión más objetiva al respecto del aprendizaje de la programación, que propenda por zanjar la discusión acerca de la relevancia que puede tener la *episteme* sobre la *tekne* pertinente o viceversa si fuere el caso.

3. Metodología

La investigación se realizó durante los años 2016, 2017 y 2018 con grupos paralelos de la asignatura Programación I de Ingeniería de Sistemas y Computación de la Facultad de Ingenierías de la Universidad Tecnológica de Pereira. Se escogió el contenido de la asignatura Programación I (que corresponde al paradigma de programación funcional) debido a que sus conceptos innovadores permitían posicionar los grupos en un mismo partidito a diferencia de otros cursos cuyo contenido es el paradigma imperativo o el paradigma orientado

a objetos, pues estas temáticas se reciben como asignaturas durante el proceso de formación en básica secundaria. Lo innovador del paradigma funcional, para los estudiantes de I semestre que recién llegan a la vida universitaria, facilita algunas de las actividades prevista para el desarrollo de la investigación.

Cada grupo se dividió en dos subgrupos de manera que con uno de ellos se pudiera conducir la asignatura basado en el desarrollo del contenido establecido curricularmente, pero acudiendo exclusivamente a su fundamentación teórica y al modelo matemático que subyace al paradigma funcional, objetivo principal de la asignatura. Con el otro subgrupo se condujo la asignatura interactuando, desde el primer día del curso, con el lenguaje de programación que se ha establecido para tal fin y que corresponde al lenguaje. La Tabla 1 presenta la cantidad de estudiantes que participaron y la forma como se estructuraron los subgrupos.

El contenido, durante el semestre, se dividió temáticamente en 4 módulos que, siendo interdependientes, fueran también independientes en sus fronteras conceptuales y que permitieran una evaluación claramente delimitada. De esta forma, cada módulo tomaría 4 semanas y así se completarían las 16 semanas que dura el semestre académico. La división conceptual que se realizó fue la siguiente: 1º módulo - Fundamentos, el concepto de función y la estrategia “divide y vencerás”, 2º módulo - Condicionales y Menús, 3º módulo – Recursividad y procesos cíclicos, 4º módulo – Conjuntos de datos (listas y vectores).

Las sesiones de cada grupo E se realizó bajo una estrategia de clase magistral haciendo énfasis en la fundamentación matemática asociada al paradigma de programación funcional que corresponde a la asignatura. Las sesiones de cada grupo T se realizó algunas veces en sala de computadores y otras veces con la utilización de computador y videobeam en el aula, pero siempre llevando los conceptos hasta las posibilidades que provee el lenguaje de programación.

Los talleres a realizar durante el curso tuvieron una orientación diferente para cada grupo. Los talleres que realizaron los estudiantes de los grupos E siempre estaban orientados a la resolución de problemas desde una perspectiva matemática general y a la luz de la algoritmia conceptual funcional. Los talleres que realizaron los estudiantes de los grupos T siempre estuvieron orientados hacia la resolución de problemas utilizando directamente el lenguaje de programación.

Se realizaron 4 evaluaciones parciales, todas ellas escritas, y cada una tenía un peso de 15% sobre la nota final, lo cual daría un 60%. El 40% restante correspondió a una evaluación final

Tabla 1.
Estudiantes participantes en la investigación

Año	Sem	Tot. Est.	Grp E*	Grp T*
2016	I	45	23	22
	II	40	19	21
2017	I	42	21	21
	II	42	20	22
2018	I	40	20	20
	II	43	22	21
Total		252	125	127
%		100%	49,6%	50,4%

E = Con orientación a la Episteme,

T = Con orientación a la Tekné

Fuente: Elaboración propia

escrita. Esta evaluación se planteó de la siguiente forma: a los estudiantes de los grupos E, se les plantearon enunciados para ser resueltos con el lenguaje de programación (se le proporcionaron algunos elementos básicos en la misma hoja de la evaluación); a los estudiantes de los grupos T, se les plantearon enunciados para que fueran resueltos desde lo puramente conceptual. De esta manera, se podía conocer el nivel de aprendizaje de cada grupo en función de la posibilidad de resolver problemas conceptuales con una preparación tecnológica o de resolverlos a través de un lenguaje de programación con una preparación enteramente conceptual.

Es de anotar que para ambos grupos las evaluaciones escritas parciales estaban en sintonía con el perfil de los talleres que habían resuelto. Esto solo fue cambiado cuando se realizó la prueba escrita final. Al finalizar el curso, se le solicitó a cada estudiante una opinión acerca de la experiencia investigativa de forma que ellos pudieran expresarse anónimamente y con libertad para detectar elementos cualitativos que fortalecieran las inferencias realizadas al respecto.

En el desarrollo de este artículo se han omitido intencionalmente ejemplos de talleres y evaluaciones parciales realizadas, así como enunciados finales en virtud de la extensión posible del artículo como tal y como una forma de respetar el espacio que para este tipo de publicaciones dispone la revista. Los detalles están disponibles para las verificaciones a que haya lugar.

4. Resultados

Para efectos de simplificar la presentación de los resultados que se obtuvieron durante toda la investigación, se calculó el promedio de las notas obtenidas por parte de los estudiantes en cada una de sus pruebas, tal como se presenta en la Tabla 2.

La evaluación (trabajo) final, que condensaba el resultado compilado de todo el proceso pues permitía la oportunidad de que el estudiante pusiera a prueba los conocimientos aprendidos en programación funcional, arrojó los resultados que se exponen en la Tabla 3.

En este caso también se acudió al cálculo del promedio de las notas para simplificar la presentación de los resultados.

Al finalizar el curso se realizó una encuesta anónima, para garantizar mayor libertad y sinceridad en la respuesta, en la cual se les hicieron a los estudiantes las siguientes 3 preguntas:

Pregunta 1E. ¿Considera usted que es posible, a partir del estudio de los conceptos básicos que subyacen a la programación funcional y a su modelo matemático, llegar a manejar con total libertad y proficiencia el lenguaje de programación?

Pregunta 1T. ¿Considera usted que es posible, a partir del estudio del lenguaje de programación y de las utilidades que provee, llega a inferir y apropiarse los conceptos que subyacen a su modelo matemático y sus respectivos conceptos básicos?

Pregunta 2. ¿Estuvo a gusto en el grupo en el cual le correspondió?

Pregunta 3. ¿De 1 a 5, en cuánto calificaría su propio nivel de aprendizaje de la programación funcional, de su modelo matemático, de la teoría que le subyace y de sus aplicaciones en el computador?

Tabla 2.

Notas obtenidas en las evaluaciones parciales

Año	Sem	Promedio de notas (Eval. Parciales)	
		Grp E	Grp T
2016	I	4.1	3.7
	II	4.0	3.9
2017	I	4.2	3.8
	II	4.0	3.9
2018	I	4.1	3.6
	II	4.0	3.8
Prom de prom		4.066	3.783
Diferencia		0.283	

Fuente: Elaboración propia

Tabla 3.

Notas obtenidas en el programa final

Año	Sem	Promedio de notas (Trab. Final)	
		Grp E	Grp T
2016	I	4.6	3.4
	II	4.5	3.5
2017	I	4.3	3.5
	II	4.5	3.6
2018	I	4.7	3.9
	II	4.4	3.6
Prom de prom		4.500	3.583
Diferencia		0.917	

Fuente: Elaboración propia

Tabla 4.

Resultados Opinión de estudiantes

Año	Sem	Pregunta 1				Pregunta 2		Pregunta 3	
		1E		1T		Si	No	GrE	Gr T
		Si	No	Si	No				
2016	I	20	3	20	2	42	3	4.5	3.5
	II	18	1	19	2	36	4	4.0	4.0
2017	I	18	3	17	4	38	4	4.5	3.5
	II	19	1	20	2	39	3	4.0	3.5
2018	I	18	2	18	2	35	5	4.4	3.9
	II	21	1	18	3	40	3	4.6	3.5
Totales		114	11	112	15	230	22	4.6	3.7
% Relativo		92%	8%	88%	12%	91%	9%	0.9	
% Absoluto		45%	4%	44%	6%				

Fuente: Elaboración propia

A los estudiantes que participaron en grupos con orientación a la episteme se les realizó la pregunta 1E y a los que formaron parte de los grupos con orientación a la tekne se les formuló la pregunta 1T. Las otras dos preguntas se les hicieron a todos los estudiantes, independiente del subgrupo en el cual les correspondió. Los resultados obtenidos con estas preguntas se presentan en la Tabla 4.

5. Discusión y análisis

En cuanto a la ventana de tiempo de investigación se consideró que 6 semestres constituían un período apropiado para realizar los análisis pertinentes teniendo en cuenta que los contenidos curriculares de un programa como Ingeniería de Sistemas pueden cambiar y que los jóvenes que vienen de los colegios (pues el estudio se realizó con estudiantes de 1er semestre), con el transcurrir del tiempo, van llegando con más capacidades de aprendizaje debido al uso de las nuevas tecnologías y al acceso a diferentes fuentes de información.

Los grupos seleccionados para este estudio fueron equivalentes tanto en cantidad como en procedencia y nivel de estudio. Se observa en la Tabla 1 que el número de estudiantes en cada grupo fue muy cercano (y algunas veces el mismo) al del grupo paralelo. Todos los estudiantes provenían del mecanismo de selección que tiene destinado la universidad pública según lo establecido en las leyes colombianas. Todos eran recién egresados de colegios tanto públicos como privados. Se presentaron casos muy esporádicos de estudiantes que habían egresado de la formación básica secundaria en tiempo superior a 1 año.

En cuanto al promedio de las notas parciales, como se presenta en la Tabla 2, puede observarse que en todo momento dicho promedio fue superior en los grupos *Episteme* que la orientación con los grupos *Tekné* y que la diferencia entre el promedio de promedios fue de 0.283 a favor de los subgrupos orientados a la *episteme*.

Debe aclararse que las evaluaciones parciales escritas siempre fueron equivalentes, es decir, se procuró que los enunciados tuvieran en común la mayor cantidad de factores que se involucran en su diseño de forma que, aun siendo contenidos diferentes, estuvieran al alcance de los alumnos dentro de un marco coherente de premios y castigos y para que los estudiantes pudieran resolverlos con las herramientas y los conceptos recibidos según la orientación del grupo. Llama la atención de que en todo momento la relación de superioridad en cuanto al promedio de notas de los subgrupos E fue superior al promedio de los subgrupos T.

En cuanto a los resultados que se presentan en la Tabla 3, estas notas reflejan el resultado del trabajo final cuyo objetivo era verificar hasta dónde cada grupo de estudiantes eran capaces de alcanzar el otro conocimiento, esto es, los estudiantes de los grupos orientados a la *Episteme* poder manejar con proficiencia el lenguaje de programación desde su propia autonomía (y con una orientación mínima) y los estudiantes de los grupos orientados a la *Tekné* poder inferir los conceptos que subyacen a la utilización del lenguaje de programación.

De acuerdo a los resultados obtenidos, se observa que el promedio de los promedios de los grupos E es superior (4.5), en resultados cuantitativos, que el promedio de los promedios de los grupos T (3.583) quedando una diferencia de 0.917 (que podría aproximarse a 1 unidad) a favor del promedio de promedios de los subgrupos E. Según los resultados todo parece indicar que resultó mucho más fácil a los estudiantes con orientación a la *episteme* adaptarse y hacer suyo el lenguaje de programación que a los estudiantes con orientación a la *tekné* inferir los conceptos teóricos que subyacen al lenguaje de referencia.

En la Tabla 4 se obtuvieron unos resultados que vale la pena analizar. La pregunta 1 (en sus dos versiones), tuvo una respuesta más favorable con los estudiantes del grupo E que con los estudiantes del grupo T. Según los resultados obtenidos, el 92% de los estudiantes de los E consideraron que era posible ir de la *episteme* a la *tekné* apropiando y utilizando sus facilidades para construir un programa funcional. Por su parte, y en comparación con éstos, un 88% de los estudiantes de los grupos T consideraron que el mismo proceso era posible pero tomando como base la *tekné* para llegar a la *episteme* de la asignatura.

En cuanto a la pregunta 2, puede observarse que el 91% de los estudiantes estuvo a gusto con el grupo en el cual fue ubicado y que solamente el 9% respondió NO a dicha ubicación que, como se explicó en un párrafo anterior, fue completamente aleatoria buscando la mayor objetividad posible y sabiendo que los estudiantes que llegan a 1er semestre traen diferentes niveles de conocimiento en programación de computadores.

Los promedios de las notas obtenidas por los mismos estudiantes en cuanto a su nivel de aprendizaje de la programación, que buscaba responder a la pregunta ¿aprendió a programar?, muestra una valoración mayor hacia los grupos E que hacia los grupos T al punto que la diferencia entre el promedio de los promedios es de 0.9, una diferencia alta considerando que se trata de la tendencia central de un promedio ya calculado.

Lo anterior nos deja presente que los estudiantes de los grupos con orientación hacia la *episteme* parecieran haber alcanzado mayor seguridad en la apropiación de los conceptos y que eso les confirió un conocimiento que valoraron cuantitativamente más que aquellos estudiantes que estuvieron en los grupos con orientación a la *tekné*. No debe desconocerse, de todas formas, que son los jóvenes de hoy, y gracias a la penetración de las nuevas tecnologías y de la transformación de su lenguaje natural, los que pueden de manera más fácil inferir los conceptos básicos epistemológicos a partir de la interacción con lenguajes de programación.

Es de anotar que las reflexiones comparativas en relación con otros estudios, por sus características y por los elementos que coinciden y discrepan, corresponden a la temática de otro artículo con el ánimo de que se puedan plantear reflexiones que enriquezcan las discusiones al respecto.

6. Conclusiones

El tamaño de la muestra de estudiantes que participó en la investigación es adecuado, aunque podrían fortalecerse más los resultados involucrando más estudiantes y por ende más cursos por semestre. Debe tenerse en cuenta que entre los factores exógenos que podrían influir en la objetividad de esta investigación está el hecho de que algunos estudiantes tienen conocimiento previo bien fundamentado en relación con la programación de computadores, los grupos tienen comunicación entre sí durante las demás asignaturas y a través de los medios de comunicación modernos posibilitando una interacción inevitable.

Se cumplió con el objetivo fundamental en un curso de programación y es que, bien por el camino de la orientación hacia la *episteme* o bien por el camino de la orientación hacia la *tekné*, los estudiantes que participaron en esta investigación aprendieron a programar aplicando tanto el modelo matemático como sus expresiones tecnológicas (lenguaje de programación).

En relación con el aprendizaje de la programación de computadores, es posible que el estudiante que recibe una orientación puramente epistemológica pueda llegar a apropiarse y aplicar con facilidad los recursos que provee un lenguaje de programación que se ajuste al paradigma estudiado.

Igualmente es posible que, a partir de una orientación enteramente tecnológica de contacto e interacción con un

lenguaje de programación desde el primer día de clases, se puedan llegar a inferir los conceptos básicos y el modelo matemático que subyace al paradigma que se esté estudiando.

Los resultados cuantitativos obtenidos en la investigación indican que es mucho más expedito el camino para apropiarse el uso del lenguaje de programación a partir de una formación basada en los fundamentos teóricos y conceptuales que inferir los fundamentos del modelo matemático que subyace a un paradigma de programación a partir del uso y aplicación de los recursos que provee un lenguaje de programación.

Bibliografía

- [1] Blanchard, B., System engineering management, 4th ed., Wiley Publish, New York, 2008.
- [2] Trejos-Buriticá, O., Metodología para aprender programación funcional en Ingeniería de Sistemas aplicando teoría de aprendizaje por descubrimiento. *Revista Educación en Ingeniería - ACOFI*, 12(23), pp. 69-75, 2017. DOI: 10.26507/rei.v12n23.719
- [3] Fuentes-Rosado, J. y Moo-Medina, M., Dificultades de aprender a programar. *Revista Educación en Ingeniería - ACOFI*, 12(24), pp. 76-82, 2017. DOI: 10.26507/rei.v12n24.728
- [4] Bisquerra-Alcina, R., Metodología de la investigación educativa. Ed. La Muralla S.A., Madrid, España, 2004.
- [5] Bustos-Gonzalez, R., Tutoría entre pares como estrategia para la formación de ingenieros: desafíos y nudos críticos. *Revista Educación en Ingeniería - ACOFI*, 13(26), pp. 4-10, 2018. DOI: 10.26507/rei.v13n26.887
- [6] Ballester-Valori, A., Meaningful Learning in practice. *Islas Canarias: Universitat de les Illes Balears, Palma, Illes Balears, España*, 2011.
- [7] Costa, V. y Rossignoli, R., Enseñanza del álgebra lineal en una facultad de ingeniería: aspectos metodológicos y didácticos. *Revista Educación en Ingeniería - ACOFI*, 12(23), pp. 49-55, 2017. DOI: 10.26507/rei.v12n23.734
- [8] Campillay, S. y Meléndez, N., Análisis de impacto de metodología activa y aprendizaje heurístico en asignaturas de ingeniería. *Actualidades Investigativas en Educación*, 15(2), pp. 1-16, 2015. DOI: 10.15517/aie.v15i2.18950
- [9] Small, G., Digital Brain. Ed. Urano, Barcelona, España, 2009.
- [10] Van Roy, P. and Haridi, S., Concepts, techniques, and models of computer programming. MIT Press, Boston, USA, 2004.
- [11] Zuluaga-Rendón, Z., Corredor, J., Quintero, J., Ramirez-Echeverry, J. y Olarte, F., ¿Qué es una buena clase en ingeniería desde el punto de vista de los estudiantes? *Revista Educación en Ingeniería - ACOFI*, 12(23), pp. 83-92, 2017. DOI: 10.26507/rei.v12n23.740
- [12] Ministerio de Educación Nacional. Estándares básicos de competencias matemáticas. MEN, Bogotá, 2005.
- [13] Monroy-Mateus, A., Aguirre-Lara, C. y Espitia-Cubillos, A., Propuesta metodológica para identificar el valor agregado de programas de ingeniería a partir del análisis de resultados de pruebas estandarizada. *Revista Educación en Ingeniería - ACOFI*, 13(25), pp. 102-107, 2018. DOI: 10.26507/rei.v13n25.868
- [14] Estrada-Esponda, R., La lectura y la escritura como herramientas pedagógicas para la enseñanza de Ingeniería de Software. *Revista Educación en Ingeniería - ACOFI*, 12(24), pp. 83-88, 2017. DOI: 10.26507/rei.v12n24.815
- [15] Trejos-Buriticá, O., Metodología 4321 de trabajo en grupo para fortalecer el aprendizaje de la programación usando aprendizaje colaborativo y aprendizaje cooperativo. *Revista Educación en Ingeniería - ACOFI*, 13(25), pp. 28-36, 2018. DOI: 10.26507/rei.v13n25.821
- [16] Tlili, A., et al., Role of personality in computer based learning. *Computer in Human Behavior*, 64, pp. 805-813, 2016.
- [17] Kline, M., El pensamiento matemático. De la antigüedad a nuestros días. Alianza Editorial, Madrid, España, 2012.
- [18] Acosta-Flores, J., Ingeniería de Sistemas: un enfoque interdisciplinario, AlfaOmega Grupo Editor, Ciudad de México, México, 2003.
- [19] Boyer, C., Historia de la matemática. Alianza Editorial, Madrid, España, 2010.
- [20] Strogatz, S., The joy of x: from One to Infinity. Mariner Books, London, England, 2013.
- [21] Aczel, A., Cuaderno secreto de Descartes. Biblioteca Buridan, Madrid, España, 2013.
- [22] Rey-Babini, J. y Pastor, J., Historia de la Matemática, Vol. II. Gedisa, Buenos Aires, Argentina, 2013.
- [23] Stewart, I., The story of mathematics. Quercus Publishing, London, England, 2009.
- [24] Jiménez-Murillo, J., Matemáticas para la computación. Alfaomega, Buenos Aires, Argentina, 2014.
- [25] Trejos-Buriticá, O., Lógica de programación. Ediciones de la U., Bogotá, Colombia, 2017.
- [26] Trejos-Buriticá, O., Programación imperativa con lenguaje C. ECOE Ediciones, Bogotá, Colombia, 2017.
- [27] Chukaew, S., et al., A personalized e-learning environment to promote students conceptual learning on basic computer programming. In: 5th World Conference on Educational Sciences, 2014, pp.815-819. DOI: 10.29097/2011-639X.183
- [28] Medina, J., Los 12 principios del cerebro. Grupo Editorial Norma, Bogotá, Colombia, 2010.
- [29] Orozco-Gomez, G., TvMorfosis. México D.F. 2017.
- [30] Levy, P., CyberCulture: electronic mediations. University of Minnesota Press, New York, USA, 2001.

O.I. Trejos-Buriticá, es Ing. de Sistemas en 1992, de la Universidad Incca de Colombia. Esp. en Instrumentación Física en 2000, MSc en Comunicación Educativa en 2004, PhD en Ciencias de la Educación en 2012, todos ellos de la Universidad Tecnológica de Pereira, Colombia. Docente de planta de la Facultad de Ingenierías, Programa Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira, Pereira, Colombia.
ORCID: 0000-0002-3751-6014.