

# ESTUDIO DE DIFICULTADES EN LA ENSEÑANZA Y APRENDIZAJE EN LOS CURSOS BÁSICOS DE PROGRAMACIÓN DE COMPUTADORAS EN PANAMÁ

## STUDY OF DIFFICULTIES IN TEACHING AND LEARNING IN THE BASIC COURSES OF COMPUTER PROGRAMMING IN PANAMA

*Diego Santimateo, Giannina Núñez, Ediviel González*

Facultad de Informática, Electrónica y Comunicación,  
Universidad de Panamá, Centro Regional Universitario de Veraguas  
E-mail: [diego.santimateo, giannina.nunez]@up.ac.pa

(Enviado Marzo 11, 2018; Aceptado Abril 16, 2018)

### Resumen

El objetivo de este estudio es analizar la percepción de los estudiantes respecto a la dificultad para aprender a programar, la utilidad de los recursos y ambientes de aprendizaje. La población objeto son estudiantes de las carreras de licenciaturas en informática de la Universidad de Panamá, Universidad Tecnológica de Panamá, Universidad Latina de Panamá. Se concluye que las actividades educativas que menos facilitan el logro de nuevos aprendizajes son las conferencias (38.4%) y el estudio en grupo (21.5%). Por otra parte, son actividades útiles, el uso de ejemplos para modificar (77.6%) y las explicaciones del docente (62.2%). Los vídeos, ejemplos de programas y tutoriales web son los recursos de mayor utilidad para los estudiantes. Los recursos que requieren de mayor lectura son considerados menos útiles. Los conceptos más difíciles de aprender son: punteros, objetos, registros, recursividad, dividir programas en módulos y crear funciones o subprogramas, aspectos coincidentes con otras investigaciones.

**Palabras clave:** Enseñanza-Aprendizaje de Programación, Conceptos de Programación, Recursos para Programar, Actividades de Programación, Ambientes de Aprendizaje, Programación.

### Abstract

The objective of this research is to know the perception of students with respect to the difficulty of learning programming, the usefulness of the resources and learning environments. The target population are students of the bachelor's degrees in computer science from the University of Panama, the Technological University of Panama and the Latina University of Panama. It is concluded that least helpful activities in obtaining new knowledge are conferences (38.4%) and study groups (21.5%). On the other hand, there are useful activities, the use of examples to be modified (77.6%) and classroom lectures (62.2%). The videos, examples programs and web tutorials are the most useful resources for students. Resources that require more reading are considered less useful. The most difficult programming concepts to learn are: pointers, objects, registers, recursion, dividing programs into module and creating functions or subprograms, that is coincident with others research.

**Keywords:** Teaching-Learning Programming, Programming Concepts, Resources for Programming, Programming Activities, Learning Environments, Programming.

## 1 INTRODUCCIÓN

El problema de la enseñanza y aprendizaje de la programación es el tema de esta investigación, debido al alto nivel de deserción que se da en las asignaturas de programación, no solo en el Centro Regional Universitario de Veraguas de la Universidad de Panamá [1], sino también a nivel internacional como lo señalan Coppo *et al.* [2] quien en función de ello propone una metodología que consiste en visualizar de forma concreta los resultados de los programas escritos por los estudiantes, introduciendo como elemento didáctico una

placa electrónica basada en principios de hardware y software.

Shuhidan *et al.* [3] ponen de manifiesto que aprender a programar es difícil, una situación que es en gran parte responsable de altas tasas de deserción en las escuelas de informática. Indica que el problema ha generado interés en una serie de investigaciones y ha dado impulso a la necesidad de un nexo de enseñanza-investigación para una mejor comprensión de los problemas de programación de los principiantes. Derus [4] se refiere a la importancia de la programación en la Ciencia, Matemática e Ingeniería en diferentes niveles, considera

el campo de la programación como difícil, complejo y caracterizado como parte de los siete grandes retos en la educación computacional.

Por otra parte, Schulte y Bennedsen [5] exploran lo que los docentes consideran importante enseñar, lo que realmente enseñan y lo que los alumnos encuentran más difícil según ellos en un curso introductorio de programación.

Salleth *et al.* [6] hacen una revisión de 45 investigaciones de la base de datos de ACM en el periodo 2005-2011 de aspectos relevantes en la enseñanza y aprendizaje de la programación, destacando que 31 de ellas atienden particularmente las técnicas, herramientas de enseñanza y la evaluación.

Soloway [7] afirma que los docentes de los cursos introductorios de programación son conscientes que enseñar sintaxis y semántica de un lenguaje de programación no es suficiente para el aprendizaje de la programación, se considera de mayor importancia fortalecer habilidades para solucionar problemas, la idea es lograr que el aprendiz pueda transferir lo aprendido en programación a otras situaciones de resolución de problemas, no sólo como una habilidad vocacional, sino como un vehículo para aprender la resolución efectiva de problemas.

En la investigación de Rodríguez [8] se procuran estrategias didácticas adecuadas para los cursos de programación de computadoras precisamente por los altos índices de deserción y pérdida académica registradas en dichos cursos. Para el desarrollo de ese trabajo se analizaron artículos de investigación e informes referenciados desde las bases de datos Scopus y DOAJ, empleando las palabras clave *teaching programming*, considerados desde el año 2000.

Teague [9] también atiende las deficiencias en los cursos de programación y presenta resultados sobre el modelo de aprendizaje por pares para proponer soluciones.

Cuando se dan deficiencias en el aprendizaje es importante determinar las posibles causas para proponer correctivos que puedan ayudar a eliminar o reducir las deficiencias, en ese sentido el trabajo de Milne y Rowe [10] determina los obstáculos más comunes encontrado por estudiantes matriculados en un primer curso de programación orientada a objetos. Para ello consultó tanto a estudiantes como a tutores de la asignatura sobre los conceptos que se enseñan y que los estudiantes se esfuerzan por aprender. En ese mismo orden de ideas encontramos el trabajo de Muñoz *et al.* [11] donde se indaga sobre los tópicos que provocaron más problemas en el proceso de enseñanza y aprendizaje de la programación, además de obtener las preferencias de los encuestados por el estudio. Los resultados se asocian con los estilos de aprendizajes detectados en esa investigación.

Con el interés de conocer con mayores detalles la situación actual en la enseñanza y aprendizaje de la programación en Panamá, se realiza esta investigación donde se analiza la percepción de los estudiantes sobre los aspectos incidentes; recursos, ambientes de aprendizaje, actividades y dominio de conceptos de manera que se identifiquen en cada uno de ellos las debilidades y fortalezas, en consecuencia, se espera que los resultados presentados en esta investigación sirvan de fundamento para diseñar estrategias didácticas, recursos didácticos, experiencias y ambientes de aprendizaje que incidan favorablemente en el desempeño de los estudiantes con la consecuente disminución de las deficiencias y deserciones.

## 2 MATERIALES Y MÉTODO

Esta investigación se basa en el estudio de Essi Lahtinen, Kirsti Ala-Mutka y Hannu-Matti Järvinen titulado “*A Study of the Difficulties of Novice Programmers*” [12] con la diferencia de que no se confrontan opiniones de estudiantes y docentes, pero adicionando aspectos de interés en las fases de programación, situaciones de aprendizaje y de recursos utilizados.

El instrumento utilizado se aplicó en el periodo 2015-2016 a estudiantes que han cursado por lo menos un semestre de programación de computadoras en la Universidad Latina de Panamá (UL) sedes Azuero y Veraguas, en la Universidad de Panamá (UP) sedes Campus Central, Panamá Oeste, Coclé, Aguadulce, Azuero, Veraguas y en la Universidad Tecnológica de Panamá (UTP) sedes Coclé, Azuero y Veraguas. Se procesaron 177 encuestas utilizando el software SPSS versión 20.

El cuestionario tiene cinco secciones; la primera sobre generalidades de los encuestados, las dos siguientes evalúan las actividades y conceptos usualmente utilizados en un curso de programación con la escala (muy difícil=1, difícil=2, normal=3, fácil=4, muy fácil=5) y las dos últimas tratan de la utilidad de los ambientes de aprendizajes evaluadas con la escala (nunca=1, algunas veces=2, normal=3, casi siempre=4, siempre=5) y recursos para aprender con la escala (nada útil=1, algo útil=2, normal=3, útil=4, muy útil=5).

De acuerdo con Hernández Sampieri *et al.* [13], este trabajo es exploratorio, porque aunque existen muchas referencias al respecto, en Panamá no es así, es decir, es un tema poco estudiado en este país. Por otra parte, esta investigación también es descriptiva, ya que se busca identificar características y rasgos importantes de la problemática del proceso enseñanza-aprendizaje de la programación de computadoras en Panamá. Se selecciona una serie de aspectos involucrados en la enseñanza y aprendizaje de la programación, algunos resultantes de investigaciones previas como la de Lahtinen *et al.* [12] y se recolectan datos sobre cada uno de ellos, aplicando la escala de Likert para luego describir lo que se investiga. Al no realizar manipulación de variables esta investigación no es experimental.

El índice de consistencia interna se calcula mediante coeficiente alfa de Cronbach dando como resultado 0.92. La medida de fiabilidad mediante el alfa de Cronbach asume que los ítems medidos en escala de Likert miden un mismo constructo y que están altamente correlacionados entre sí [14]. Se utilizan los porcentajes relativos a las frecuencias de las respuestas de cada una de las preguntas, para identificar el nivel de dificultad según la escala.

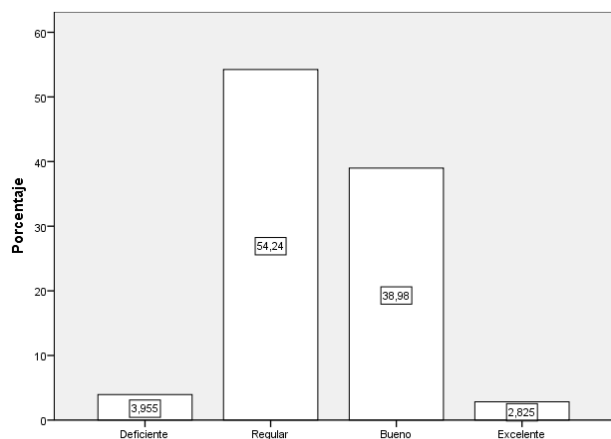
El cuestionario se diseñó agrupando las preguntas de manera que combinadas miden un rasgo particular, en consecuencia, se considera escala de intervalos y es aplicable la suma o media aritmética para su análisis [15]. Además del análisis a partir de la media de los componentes de cada rasgo bajo estudio, se determinan mediante el coeficiente de correlación de Pearson, con significación estadística, las posibles relaciones de las variables involucradas en las dificultades para el dominio de actividades de programación, aprendizaje y uso de conceptos de programación, recursos que facilitan el aprendizaje de la programación y ambientes de aprendizaje.

### 3 RESULTADOS Y DISCUSIÓN

Se aplicaron 177 encuestas, la distribución se registra en la Tabla 1.

**Tabla 1** Distribución de la muestra por Centro de Estudio y Sexo.

| Centro de Estudio | Sexo del encuestado |           | Total |
|-------------------|---------------------|-----------|-------|
|                   | Femenino            | Masculino |       |
| UL Veraguas       | 3                   | 11        | 14    |
| UL Azuero         | 3                   | 6         | 9     |
| UP Veraguas       | 11                  | 13        | 24    |
| UP Aguadulce      | 2                   | 7         | 9     |
| UP Coclé          | 8                   | 8         | 16    |
| UP Azuero         | 6                   | 8         | 14    |
| UP Campus Central | 1                   | 7         | 8     |
| UP Panamá Oeste   | 9                   | 9         | 18    |
| UTP Veraguas      | 12                  | 25        | 37    |
| UTP Coclé         | 5                   | 7         | 12    |
| UTP Azuero        | 4                   | 12        | 16    |
| Total             | 64                  | 113       | 177   |



**Figura 1** Los estudiantes valoran sus conocimientos en programación.

Poco más del 54% de los estudiantes consideran sus conocimientos en programación como regular, seguido de un 38.98% que lo consideran bueno. Los estudiantes que perciben sus conocimientos como excelentes o deficientes no superan el 4% (Fig. 1).

Respecto a las actividades propias del proceso de aprender a programar, se presentan los porcentajes de frecuencias acumuladas para los valores “muy difícil” y “difícil” (Tabla 2). Los estudiantes perciben que la actividad de crear módulos asociada con el uso de funciones o subprogramas es una de las más difíciles de aprender. Se observa la tendencia de la media hacia la alternativa “fácil de aprender” indicando que en general los estudiantes perciben que estas actividades se aprenden con relativa facilidad.

**Tabla 2** Percepción del aprendizaje de las actividades del proceso de aprender a programar.

| Actividades   | Media | Porcentaje de dificultad |
|---|-------|--------------------------|
| Dividir el programa en módulos                        | 3,14  | 27.7                     |
| Encontrar errores en el programa                      | 3,31  | 15.3                     |
| Crear y usar funciones o subprogramas                 | 3,36  | 22.6                     |
| Usar la sintaxis del lenguaje de programación         | 3,50  | 11.9                     |
| Conocer como la computadora ejecuta/corre el programa | 3,52  | 11.9                     |
| Usar el entorno de desarrollo de programas            | 3,63  | 11.3                     |
| Crear diagrama de flujo o lógica del programa         | 3,69  | 10.7                     |
| Seleccionar datos para probar el programa             | 3,69  | 8.5                      |

En el caso de los conceptos de programación de computadoras, también se muestran los porcentajes de frecuencias acumuladas para los valores “muy difícil” y “difícil” reflejando que los punteros, arreglos, estructuras, objetos y la recursividad se perciben como los conceptos más difíciles de aprender (Tabla 3). Según la media, el concepto de variable, las instrucciones de control del flujo del programa y el manejo de entrada / salida de datos son los conceptos moderadamente más fáciles de aprender.

**Tabla 3** Percepción del aprendizaje de los conceptos.

| Concepto   | Media | Porcentaje de dificultad |
|--|-------|--------------------------|
| Punteros o Apuntadores                           | 2,73  | 45.2                     |
| Tipos abstractos de datos (objetos)              | 2,86  | 28.2                     |
| Recursividad                                     | 2,98  | 30.5                     |
| Tipos de datos estructurados (struct, registros) | 3,03  | 25.4                     |
| Parámetros / Argumentos                          | 3,11  | 22.6                     |
| Funciones / Subprogramas / Métodos               | 3,11  | 18.6                     |
| Librerías del lenguaje                           | 3,22  | 16.9                     |
| Arreglos   | 3,24  | 24.9                     |
| Manejo de errores / Excepciones                  | 3,27  | 17.5                     |
| Entrada / Salida                                 | 3,55  | 11.3                     |
| Ciclos   | 3,80  | 11.9                     |
| Decisiones (if, case)                            | 3,95  | 9.0                      |
| Uso de las variables                             | 4,14  | 7.9                      |

En la Tabla 4 se observa que los estudiantes consideran que el aprendizaje se les facilita en las sesiones de laboratorios, se resalta la débil valoración del trabajo en grupos y los aportes de las conferencias.

Respecto a los recursos utilizados para aprender, se observa que los vídeos, los ejemplos de programas y los tutoriales web son los de mayor utilidad. Los recursos que requieren de mayor lectura son considerados menos útiles (Tabla 5).

En la Tabla 6 se presentan las correlaciones entre actividades y entre ellas y los conceptos, resultando fuertes, positivas y significativas al nivel 0.01 ( $p=0.01$ ). Se han descartado las correlaciones menores que 0.4, para un mayor significado de los resultados.

Al analizar las correlaciones entre las actividades evaluadas resalta la importancia del dominio de la sintaxis del lenguaje, así como el entorno de programación y la creación de diagramas de flujo cuando se manejan funciones o subprogramas, además es evidente la relación entre la estrategia de dividir el programa en módulos con la creación de funciones que refleja una correlación fuerte y positiva con un coeficiente de correlación de 0.687, en muchos casos la división del programa en módulos se considera parte de la creación de funciones.

**Tabla 4** Percepción de los ambientes en los que se aprende.

| Ambiente                                | Media | Menor incidencia en aprendizaje (%) |
|---|-------|-------------------------------------|
| En conferencias                         | 3,05  | 38.4                                |
| Cuando estudio en grupos                | 3,58  | 21.5                                |
| Cuando estudio solo                     | 3,62  | 18.1                                |
| Mientras tenga problemas que resolver   | 3,62  | 12.4                                |
| Cuando el profesor explica              | 3,69  | 11.9                                |
| Mientras tenga un ejemplo que modificar | 3,72  | 12.4                                |
| En sesiones prácticas de laboratorios   | 4,03  | 5.6                                 |

**Tabla 5** Percepción de la utilidad de los recursos.

| Recursos   | Media | Menor utilidad en el aprendizaje (%) |
|--|-------|--------------------------------------|
| Clase virtual o folleto del docente o transparencias | 3,36  | 22.0                                 |
| Libros del curso de programación                     | 3,37  | 26.6                                 |
| Ejercicios de preguntas y respuestas                 | 3,53  | 15.8                                 |
| Listado de problemas por resolver                    | 3,66  | 11.3                                 |
| Imágenes de las estructuras de programación          | 3,68  | 12.4                                 |
| Mis apuntes de la clase                              | 3,92  | 9.6                                  |
| Videos de programación                               | 4,15  | 9.0                                  |
| Ejemplos de programas                                | 4,25  | 2.3                                  |
| Tutoriales de la web en internet                     | 4,37  | 6.2                                  |

**Tabla 6** Actividades del proceso de aprender a programar con mayor correlación.

| Actividades                                   | Usar el entorno de desarrollo de programas | Crear diagrama de flujo o lógica del programa | Crear y usar funciones o subprogramas | Usar la sintaxis del lenguaje de programación |
|---|--|---|---------------------------------------|---|
| Crear diagrama de flujo o lógica del programa | ,446**                                     |   |                                       |   |
| Crear y usar funciones o subprogramas         | ,616**                                     | ,598**  |                                       |   |
| Usar la sintaxis del lenguaje de programación | ,492**                                     | ,532**  | ,667**                                |   |
| Dividir el programa en módulos                | ,492**                                     |   | ,687**                                | ,579**  |

\*\* La correlación es significativa al nivel 0,01 (bilateral).

Al momento de relacionar las actividades con los conceptos, se percibe la división del programa en módulos como la más afín a la mayoría de los conceptos tratados en la programación de computadoras, enfatizando así la importancia que los estudiantes le dan a la misma. Además, presenta una correlación de 0.61 ( $p=0.01$ ) con los conceptos involucrados con las funciones o subprogramas (Tabla 7).

**Tabla 7** Actividades del proceso de aprender a programar y conceptos con mayor correlación.

| Conceptos                           | Usar el entorno de desarrollo de programas | Crear y usar funciones o subprogramas | Usar la sintaxis del lenguaje de programación | Dividir el programa en módulos |
|-------------------------------------|--|---------------------------------------|---|--------------------------------|
| Decisiones (if /case)               |  |                                       | ,524**  | ,572**                         |
| Ciclos                              |  | ,563**                                | ,500**  | ,537**                         |
| Recursividad                        |  | ,550**                                |   | ,511**                         |
| Arreglos                            |  | ,554**                                |   | ,559**                         |
| Punteros                            |  | ,459**                                |   | ,487**                         |
| Parámetros / Argumentos             |  | ,532**                                |   | ,557**                         |
| Tipos de datos estructurados        | ,545**                                     | ,553**                                |   | ,567**                         |
| Tipos abstractos de datos (objetos) | ,505**                                     | ,565**                                |   | ,554**                         |
| Funciones / Subprogramas / Métodos  |  |                                       |   | ,610**                         |

\*\* La correlación es significativa al nivel 0,01 (bilateral).

## 4 DISCUSIÓN

Muchas de las dificultades en el proceso de enseñanza y aprendizaje de la programación de computadoras que existen en Panamá, no son exclusivas de nuestro medio, Lahtinen *et al.* [12] presenta resultados similares con

estudiantes de Alemania, Finlandia, Rumania, Islandia y Letonia. Es importante resaltar que los datos obtenidos corresponden a las percepciones de los estudiantes consultados y que en algunos aspectos la experiencia indica que no siempre tienen claridad en los conceptos, ni el dominio profundo de los procesos, ni de la aplicación de los conceptos. Esto claramente se ve cuando el estudiante decide que ya ha terminado un programa y no ha hecho un análisis de los datos de prueba corroborando por lo menos el seguimiento de cada rama de la lógica y verificando los resultados con un modelo previamente preparado para ello.

Según la percepción general de los estudiantes, los recursos y los ambientes de aprendizajes representan un nivel aceptable de utilidad, se distinguen los ejemplos de programas, los materiales visuales y los tutoriales web como los recursos de mayor utilidad para los estudiantes, resultados similares a los obtenidos por Muñoz *et al.* [11]. Los recursos que requieren de mayor lectura son considerados menos útiles.

Respecto a las actividades de programación, los estudiantes perciben que lo más difícil de aprender se centra en la división de un programa en módulos, muy asociado con la creación y uso de funciones. Es posible que la comprensión del proceso de comunicación entre el programa invocador y el invocado no se haya logrado satisfactoriamente, particularmente el pase de parámetros a argumentos y el retorno de resultados. Por otra parte, poco se trabaja en el desarrollo en equipo donde cada estudiante se encarga de un módulo, sus pruebas e integración [16]. En cuanto a los conceptos propios de la programación de computadoras, los más difíciles de aprender son: los punteros, la recursividad y el manejo de objetos. Pero en general, más del 69% de los estudiantes consideran que el aprendizaje de los conceptos es fácil o muy fácil cuando se utiliza la sumatoria de los ítems, no obstante, al analizar los valores promedios la mayoría cae en el rango entre 3.03 y 3.80 lo que refleja un nivel de conocimiento moderado, similar al resultado del estudio de Derus [4].

Las correlaciones entre conceptos y actividades resultaron ser las más significativas corroborando la importancia de la creación de funciones o subprogramas que correlaciona fuerte y significativamente con la mayoría de los conceptos tratados en un curso de programación.

Se coincide con lo reportado por Lahtinen *et al.* [12] respecto a que el manejo de las librerías resultó poco correlacionado con las actividades, en consecuencia, debe prestarse mayor interés en la enseñanza y aprendizaje de ese tópico.

Es recomendable que la teoría asociada a los conceptos sea transferida a la práctica y posteriormente a la solución de problemas reales avanzando de lo simple a lo complejo conjugando lo aprendido con los nuevos retos de aprendizaje, como se desprende del aporte de Díaz-Barriga y Hernández [17].

El estudio y participación en la creación de objetos de aprendizajes interactivos como los que promueve el proyecto CODEWITZ ([www.codewitz.net](http://www.codewitz.net)) puede rendir beneficios al proceso de enseñanza y aprendizaje de la programación. Por otra parte, comprender cómo los estudiantes aprenden apoya la creación de materiales didácticos adecuados y personalizados que favorecen el aprendizaje. Queda como reto investigar, experimentar y evaluar las estrategias de enseñanza y aprendizaje propuestas por muchos investigadores en esta área, a fin de lograr la reducción de los fracasos y la deserción en asignaturas de programación.

## 5 CONCLUSIÓN

Se identifican de manera coincidente con otras investigaciones los conceptos de programación de mayor dificultad para aprender. Con la investigación se tiene una idea clara de los recursos, actividades y ambientes de aprendizaje que favorecen el aprendizaje de los conceptos y contenidos de un curso de programación básico. Se debe profundizar en los orígenes del poco valor que los estudiantes le dan al trabajo en grupo por las bondades conocidas en la interacción e intercambio de conocimientos del ser humano, así como en la variedad de metodologías que diversos autores proponen para reducir los fracasos y deserción en los cursos de programación. En ese sentido corresponde realizar experimentos con estrategias sugeridas como por ejemplo, gamificación, programación visual, robótica educativa, uso de software diseñado para la enseñanza de la programación, pensamiento computacional, aplicaciones móviles, efectos de la zona de confort, enfoque constructivista de desarrollo de proyectos, aspectos motivacionales y otras, de manera que se puedan obtener conclusiones sobre sus respectivas eficiencia y efectividad en el aprendizaje de la programación.

## 6 REFERENCIAS

- [1] Santimateo, D. (2014). Factores que inciden en el rendimiento académico de estudiantes en los cursos iniciales de programación de computadoras. Panamá: Universidad de Panamá.
- [2] Coppo, R., Iparraguirre, J., Feres, G., Ursua, G., Cavallo, A. (2011). Sistema didáctico para la enseñanza de la programación con metodologías de aprendizaje basado en problemas. Conferencia presentada en el XIII Workshop de Investigadores en Ciencias de la Computación, 824-827.
- [3] Shuhidan, S., Hamilton, M., D'Souza, D. (2009). A taxonomic study of novice programming summative assessment. *Conferences in Research and Practice in Information Technology Series*, 95, 147-156.
- [4] Derus, R. (2012). Difficulties in learning programming: Views of students. 1st International Conference on Current Issues in Education, ICCIE2012, At Yogyakarta, Indonesia. doi: 10.13140/2.1.1055.7441.
- [5] Schulte, C., Bennedsen, J. (2006). What do teachers teach in introductory programming? *Proceedings of the second international workshop on computing education research - ICER '06*. doi:10.1145/1151588.1151593.

- [6] Salleh, S., Shukur, Z., Judi, H. (2013). Analysis of Research in Programming Teaching Tools: An Initial Review. *Procedia - Social And Behavioral Sciences*, 103, 127-135. doi:10.1016/j.sbspro.2013.10.317.
- [7] Soloway, E. (1986). Learning to program = learning to construct mechanisms and explanations. *Communications Of The ACM*, 29(9), 850-858. doi:10.1145/6592.6594
- [8] Rodríguez Carrillo, G. (2014). Enseñanza de la programación de computadoras para principiantes: un contexto histórico. *Revista Inventum*, 9(17), 51-61. doi:10.26620/uniminuto.inventum.9.17.2014.51-61
- [9] Teague, D. (2011). *Pedagogy of Introductory Computer Programming: A People-First Approach*. Master of Information Technology. Queensland University of Technology.
- [10] Milne, I., Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, 7(1), 55–66. doi:10.1023/a:1015362608943.
- [11] Muñoz, R., Barría, M., Noël, R., Providel, E., Quiroz, P. (2012). Determinando las Dificultades en el Aprendizaje de la Primera Asignatura de Programación en Estudiantes de Ingeniería Civil Informática. Conferencia presentada en Congreso Internacional de Informática Educativa, XVII, 1-7. doi: 10.13140/2.1.2000.7206.
- [12] Lahtinen, E., Ala-Mutka, K., Järvinen, H. (2005). A study of the difficulties of novice programmers. *Proceedings of the 10th annual SIGCSE conference on Innovation and Technology in Computer Science Education - ITiCSE '05*. 14-18. doi:10.1145/1067445.1067453.
- [13] Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P. (2014). *Metodología de la investigación* (6a ed). México, D.F: McGraw-Hill.
- [14] Frías-Navarro, D. (2014). *Apuntes de SPSS*. Universidad de Valencia. Recuperado de <http://www.uv.es/friasnav/ApuntesSPSS.pdf>.
- [15] Boone, H. N., Boone, D. A. (2012). Analyzing Likert Data. *Journal of Extension*, 50(2), 1–5. Recuperado a partir de [http://wiki.biologysholars.org/@api/deki/files/2002/=Likert\\_Scale\\_Analysis.pdf](http://wiki.biologysholars.org/@api/deki/files/2002/=Likert_Scale_Analysis.pdf).
- [16] Wang, F.L., Kwan, R. and Wong, K. (2012). An Effective Tool to Support Teaching and Learning of Modular Programming. *Communications in Computer and Information Science*, pp.80-90. doi: 10.1007/978-3-642-31398-1\_8.
- [17] Díaz-Barriga, F. and Hernández Rojas, G. (2002). *Constructivismo y Aprendizaje Significativo*. En: *Estrategias docentes para un aprendizaje significativo*, (2ª Ed). México: McGraw-Hill, pp.23-60.