

# 27

## DESARROLLO DE LA LÓGICA DE PROGRAMACIÓN EN ESTUDIANTES DE SISTEMAS DE UNIANDES SANTO DOMINGO

### DEVELOPMENT OF PROGRAMMING LOGIC IN STUDENTS OF UNIANDES SANTO DOMINGO SYSTEMS

Silvio Amable Machuca Vivar<sup>1</sup>

E-mail: [us.silviomachuca@uniandes.edu.ec](mailto:us.silviomachuca@uniandes.edu.ec)

ORCID: <https://orcid.org/0000-0002-4681-3045>

Carlos Roberto Sampedro Guamán<sup>1</sup>

E-mail: [us.carlossampedro@uniandes.edu.ec](mailto:us.carlossampedro@uniandes.edu.ec)

ORCID: <https://orcid.org/0000-0002-1937-100X>

Diego Paúl Palma Rivera<sup>1</sup>

E-mail: [us.diegopalma@uniandes.edu.ec](mailto:us.diegopalma@uniandes.edu.ec)

ORCID: <https://orcid.org/0000-0002-7684-7721>

Fredy Pablo Cañizares Galarza<sup>1</sup>

E-mail: [us.fredycanizares@uniandes.edu.ec](mailto:us.fredycanizares@uniandes.edu.ec)

ORCID: <https://orcid.org/0000-0003-4854-6996>

<sup>1</sup> Universidad Regional Autónoma de Los Andes. Ecuador.

#### Cita sugerida (APA, séptima edición)

Machuca Vivar, S. A., Sampedro Guamán, C. R., Palma Rivera, D. P., & Cañizares Galarza, F. P. (2021). Desarrollo de la lógica de programación en estudiantes de sistemas de Uniandes Santo Domingo. *Revista Conrado*, 17(79), 214-224.

#### RESUMEN

La enseñanza de algoritmos es una materia básica de las carreras de ingeniería de Sistemas y Software, requiere de los estudiantes un desarrollo del razonamiento lógico, por lo que la deserción inicial es elevada. Los estudiantes piensan que programar un computador es un proceso de ensayo-error en un lenguaje de programación. Para evitar esto se debe contar con elementos específicos y una metodología apropiada que motiven a los estudiantes a visualizar en forma concreta los resultados de los programas escritos. Se presenta un diagnóstico sobre el desarrollo de la lógica de programación de los estudiantes de las carreras de Sistemas y software de las modalidades presencial y semipresencial de la extensión de la Universidad Regional Autónoma de Los Andes, en Santo Domingo, los hábitos de aprendizaje de los estudiantes, métodos y técnicas de enseñanza del docente. Se aplicó una encuesta a una muestra de estudiantes de las carreras de Sistemas y Software, se entrevistó a los docentes, una encuesta piloto a estudiantes de la Pontificia Universidad Católica del Ecuador-PUCE en Santo Domingo. Obteniéndose resultados que demuestran las necesidades de mejorar la estrategia de enseñanza aprendizaje de algoritmos para lograr un mejor desarrollo de la lógica de programación, como base para un estudiante de ingeniería de sistemas y de software.

#### Palabras clave:

Enseñanza de la programación, lógica de programación, lenguaje de programación, Ingeniería de sistemas, algoritmos.

#### ABSTRACT

The teaching of algorithms is a basic subject in the careers of systems and software engineering, requires students to develop logical reasoning, so the initial dropout is high. Students think that programming a computer is a trial-and-error process in a programming language. To avoid this, there must be specific elements and an appropriate methodology to motivate students to visualize in a concrete way the results of written programs. A diagnosis is presented on the development of the programming logic of the students of the Systems and Software degrees of the face-to-face and blended learning modalities of the Universidad Regional Autónoma de Los Andes, in Santo Domingo, the learning habits of the students, methods and techniques of teaching of the teacher. A survey was applied to a sample of students in the Systems and Software careers, teachers were interviewed, and a pilot survey was conducted with students from the Pontificia Universidad Católica del Ecuador-PUCE in Santo Domingo. Results were obtained that demonstrate the need to improve the teaching strategy and learning algorithms to achieve a better development of programming logic, as a basis for a student of systems and software engineering.

#### Keywords:

Teaching of programming, programming logic, programming language, systems engineering, algorithms.

## INTRODUCCIÓN

La mayoría de usuarios de un computador o dispositivo inteligente está familiarizado con la ejecución de programas, de forma general cuando se quiere ejecutar un programa se lo indica al sistema haciendo doble clic sobre él, algunos usuarios más expertos ejecutan un programa al introducir comandos en una consola, pero si el programa se lo permite y puede abrir el código fuente en un editor de texto se encontrará con una serie de símbolos, números u otros caracteres que hace que el usuario se pregunte ¿Cómo y quienes lograron escribir el código?. (Moreno Pérez, 2013), es este caso son los ingenieros de software los responsables de escribir el código, previo proceso de análisis y diseño.

Un ingeniero en sistemas o de software identifica problemas y propone soluciones en diversas áreas del conocimiento, por tal motivo su trabajo va más allá de ser capaz de manipular diversas aplicaciones de ofimática, de navegar en internet, de reparar una computadora o de programar una computadora. Un ingeniero en sistemas crea y hace uso de diferentes niveles de abstracción, para entender y resolver problemas con mayor eficacia. La esencia de la programación es la abstracción y que las abstracciones para la informática son las herramientas intelectuales y la computadora una herramienta que automatizan las abstracciones.

Para un estudiante de secundaria le es bastante llamativa el área de la informática, las posibilidades de aprender a navegar en internet y utilizar todos sus servicios, de instalar programas, descargar juegos e incluso prácticas de hacking, todo esto le hace pensar que es una carrera muy interesante y nace su interés también por aprender a programar. Cuando un estudiante se interesa por aprender programación se enfrenta con el primer problema de no saber exactamente por dónde comenzar a estudiar. Como primeros requerimientos se destaca la dedicación en practicar y aprender de los errores, la afinidad con matemática e inglés. Una estrategia valedera es comenzar a enseñar programación utilizando los algoritmos como recursos esquemáticos para plasmar el modelo de la resolución de un problema.

El proceso de escribir los algoritmos hasta ponerlos a punto es operativamente complicado, más aún si se trabaja solo el lápiz y papel, tiza y pizarrón, o una herramienta de pseudolenguaje. El principal inconveniente es comprobar la validez del algoritmo, es difícil, mental o gráficamente, llevar a cabo las acciones del algoritmo en ejecución de manera totalmente objetiva sin dejarse llevar por la subjetividad de su especificación, es decir, tratando de olvidar el pensamiento que llevó al desarrollo y concentrándose

exclusivamente en lo que se encuentra escrito. (Alonso & Fernández, 2013).

En los países de América, especialmente en EE. UU., la asignatura de Programación está integrada dentro de la cultura educativa. Usar un lenguaje de programación como soporte a las Matemáticas, Física o Química, o incluso áreas que no son exclusivas de Ciencias, es algo habitual en muchas escuelas de secundaria o las titulaciones universitarias. Muchos centros educativos en esos países trabajan con herramientas que ayudan a una mejor didáctica de la Programación, alejándose de habituales entornos y lenguajes profesionales de Programación (Pérez Narváez & Roig, 2015).

En 2006 Ecuador empieza a incorporar formalmente las TIC dentro de los diferentes sectores. “En el sector educativo del país se ha apuntado a la dotación de infraestructuras, equipamiento de aulas con ordenadores y recursos informáticos, dotación de software educativo, capacitación al profesorado, creación de portales educativos, soporte técnico a las escuelas, entre otros” (Peñaherrera León, 2012). Ahora bien, son escasos los proyectos desarrollados basados en herramientas tecnológicas del tipo Scratch para desarrollar el pensamiento computacional. Únicamente se ha llevado a cabo un proyecto implementado por la Escuela Politécnica del Litoral (ESPOL) a nivel de educación básica y otro por la Universidad Yachay en colaboración con Clear mindit organizado con niños de entre 8 a 12 años, pero no hay evidencias de su uso en la educación superior y los resultados logrados con el uso de la herramienta (Pérez Narváez & Roig, 2015).

La palabra lógica viene del griego antiguo *logike* y significa razón, tratado o ciencia. En matemáticas es la ciencia que estudia los métodos de razonamiento, proporciona reglas y técnicas para determinar si un argumento es válido o no, indica la forma correcta de obtener conclusiones y los métodos adecuados para llegar a ellas. El razonamiento lógico se emplea en las matemáticas y la computación para demostrar teoremas y para resolver una multitud de situaciones problemáticas.

La lógica provee los elementos necesarios para representar el conocimiento a través de métodos de formalización de las frases declarativas (Cardona Torres, et al., 2010). Cuando un sistema de razonamiento utiliza un lenguaje simbólico y un conjunto de reglas de inferencia recibe el nombre de lógica simbólica, esta lógica es la que a través de los algoritmos conformados por estructuras lógicas ha permitido el desarrollo de la informática, principalmente del componente de software.

Un algoritmo es un conjunto finito de reglas bien definidas en su lógica de control que permiten la solución de

un problema en una cantidad finita de tiempo (Mancilla Herrera, et al., 2015). La programación es parte esencial e integral de cualquier programa de ingeniería. Los estudiantes de ingeniería en sus últimos semestres tienen que enfrentar tareas de solución de problemas empleando esta competencia. El contar con buenas habilidades de desarrollo les ayudará a dar solución a estos problemas fácilmente. Es importante que los estudiantes de ingeniería y tecnología aprendan programación básica en sus primeros años de su preparación universitaria (Fuentes Rosado & Moo Medina, 2017).

Para la enseñanza-aprendizaje de la programación se encuentra una serie de recursos en internet, tales como: manuales, video tutoriales, código ya realizado, ejemplos de algoritmos en distintos lenguajes de programación, que pueden aplicarse a la didáctica del docente y en la organización de sus clases, pero se encuentra poca atención a los factores cognitivos presentes en el proceso de aprendizaje. El estilo de aprendizaje y la motivación se destacan como factores cognitivos que afectan el desempeño del estudiante.

El estudiante aplica método de refinamientos sucesivos: este componente busca diagnosticar si el estudiante ha adquirido la habilidad analítica de descomponer un problema complejo en partes más simples (estrategia divide y vencerás). Las respuestas tabuladas pueden ser: <sí, no, en forma incompleta>.

El estudiante describe el significado de variables: este componente busca detectar si el estudiante tiene la capacidad de explicar para qué define los recursos que usa en el desarrollo del programa. Las respuestas tabuladas pueden ser: <sí, no>.

El estudiante tiene estilo de escritura: este componente busca detectar si el estudiante tiene en cuenta el mantenimiento de un programa, es decir, que el programa puede ser leído y/o utilizado por otros programadores o incluso por sí mismo en un futuro cercano, por lo cual debe escribirse en forma clara e inteligible. Las respuestas tabuladas pueden ser: <sí, no>.

El estudiante usa enunciados de documentación interna: este componente busca detectar si el estudiante ha adquirido la noción de programa autodocumentado, es decir, auto explicativo. Las respuestas tabuladas pueden ser: <sí, no, en forma incompleta> (Jiménez Reyes, et al., 2008).

Luego de varios años la comunidad académica se puso de acuerdo sobre cuál debe ser la forma correcta como los estudiantes deben aprender a programar computadoras. Primero, deben aprender la metodología de la

programación, usando técnicas de diseño algorítmico o pseudocódigo, después, deben aprender cómo implementarla usando un lenguaje de programación.

Con el desarrollo del lenguaje Java y la difusión que ha tenido como el primer lenguaje que muchos estudiantes están aprendiendo, y debido a la falta de una metodología apropiada, se está cayendo en el error de enseñar a programar directamente con el lenguaje Java, dejando de lado el desarrollo de la lógica, y se están formando programadores con poca o sin lógica (López, 2013).

Para la mayoría de los usuarios de un computador le he desconocido todo el proceso que llevó al funcionamiento de un programa en el computador, hay un trabajo intelectual muy exigente que se resume en por lo menor siete etapas o pasos:

1. Planteamiento del problema, entender el problema.
2. Análisis del problema, planear la lógica.
3. Codificar el programa, transcribir el algoritmo a un lenguaje de programación.
4. Depurar el programa, usar software (un compilador o intérprete) para traducir el programa a lenguaje de máquina.
5. Probar el programa.
6. Poner el programa en producción.
7. Mantener el programa.

El resultado de ordenar los pasos de forma lógica para realizar un proceso específico se conoce como algoritmo, luego el algoritmo requiere de la codificación utilizando un lenguaje de programación.

Un programa con errores de sintaxis no puede traducirse por completo ni ejecutarse. Uno que no los tenga es traducible y puede ejecutarse, pero todavía podría contener errores lógicos y dar como resultado una salida incorrecta. Para que un programa funcione en forma apropiada usted debe desarrollar una lógica correcta, es decir, escribir las instrucciones en una secuencia específica, no dejar fuera ninguna instrucción y no agregar instrucciones ajenas (Farrell, 2013).

#### Técnicas de diseños de programas:

**Organigrama.** - Es una técnica gráfica que sirve para expresar el orden en que deben ejecutarse las instrucciones.

**Pseudocódigo.** - Es una técnica alternativa a la anterior que sirve para realizar programas. Emplea palabras clave (sentencias) que indican el tipo de operación a realizar. Ejemplo (García Sánchez, et al., 2002):

DO WHILE Sentencias;

## DO END

La construcción lógica de programas (CLP) es una concepción que estimula al programador a diseñar programas, no a codificar.

Se aplican los principios:

Estructuras básicas (Secuencia, alternativas, repetitivas), recursos abstractos y razonamiento top-down.

Cualquier sistema informático basado en software que opere sobre un conjunto de máquinas de procesamiento electrónico depende de los algoritmos para su funcionamiento (Mancilla Herrera, et al., 2015)

Los sistemas de información al servicio de la sociedad del conocimiento se basan tanto en la Ciencia de la Computación como en la Ingeniería de Software. La resolución de problemas del mundo real de cualquier disciplina y a cualquier nivel en las escalas económicas nacional e internacional ha utilizado un sistema informático. Es así como el Software y la informática han estado y estarán presentes en la solución de problemas en todas las áreas.

Las fuentes de consulta especializadas enfocan la enseñanza de la programación en las nuevas tecnologías (web, Programación Orientada a Objetos, modelo vista controlador, y otros) que pueden aplicarse a la didáctica del docente y en la organización de sus clases. Pero poca información se detalla referente a los factores cognitivos presentes en el proceso de aprendizaje. El estilo de aprendizaje y la motivación se destacan como factores cognitivos que afectan el desempeño del estudiante.

La motivación, que por definición es “la dirección y magnitud del comportamiento humano”, tiene como factores indispensables la elección del curso de acción, la persistencia en el mismo y el esfuerzo que el estudiante decidirá realizar. Una vez decidido el curso de acción el factor emocional/afectivo tiene un rol preponderante. El estudiante se relaciona no solo cognitivamente con su proyecto sino también en forma afectiva; característica que es fundamental para determinar el tiempo y esfuerzo que luego dedicará a la conclusión de este. Tradicionalmente la motivación se ha considerado bajo una perspectiva individualista. Sin embargo, como toda acción humana se realiza en un contexto, este ejerce una influencia indiscutible en todo el proceso de aprendizaje (Coppo, et al., 2011).

Las computadoras y los ahora denominados dispositivos inteligentes (teléfono, Tablet y otros) son máquinas electrónicas que pueden ejecutar un gran número de operaciones a muy altas velocidades, de forma precisa y automática. Pero para que esto ocurra, el componente

electrónico requiere del componente lógico (software), el cual debe indicarle específicamente qué operaciones debe ejecutar y en qué orden, esta serie de instrucciones denominadas programas de computadora son elaborados por los programadores.

*“Un programa es una serie de instrucciones escritas en forma codificada que la computadora puede traducir a su propio lenguaje (binario)”*. (Oviedo Regino, 2004). Para construir un programa o modificarlo se debe partir de un análisis de sistemas, el cual consiste generalmente en ubicarnos en lugar del computador y especificar qué datos de entrada debo ingresar, qué procesos se deberán realizar con esos datos de entrada (expresiones, sentencias, formular) y finalmente general los datos de salida en el formato y cantidad requerida por el usuario del computador, concluido este proceso es necesario seleccionar el lenguaje de programación que permitirá traducir de lenguaje natural a lenguaje de máquina.

En la evolución de la programación de computadoras, un punto de inflexión lo marca la aparición PASCAL, un lenguaje de programación para la enseñanza de técnicas de programación estructurada que se convirtió en un estándar de facto en el mundo de la programación. Han pasado más de 45 años y sin embargo la programación estructurada sigue teniendo enorme importancia en el campo de la enseñanza. Aunque ha habido muchos cambios y avances desde ese entonces, por ejemplo, la aparición y consolidación de la programación orientada a objetos, coexisten varios enfoques y tendencias, pero sin consenso sobre cuál es la mejor manera de enseñar los conceptos básicos de la programación (Ferreira Szpiniak & Rojo, 2006).

Al citar dos grandes mentes del siglo XXI: *“Aprender a escribir programas exprime tu mente, y ayuda a pensar mejor, crea una manera de pensar sobre cosas que creo es útil en todos los dominios”*-Bill Gates. *“Nuestra política en Facebook es literalmente contratar tantos ingenieros talentosos que podamos encontrar. No hay suficiente gente entrenada y que tengan estas habilidades hoy”*-Mark Zuckerberg. En primera instancia Bill Gates, Fundador de Microsoft, menciona los beneficios que tiene el aprender a programar, incluso sin mencionar el área de ingeniería, Bill Gates dice que los beneficios de programar crean nuevas formas de pensar y solucionar problemas. Por su parte Mark Zuckerberg, fundador de Facebook, reclama la necesidad de buenos y talentosos programadores, haciendo notoria la falta de ellos (Fuentes Rosado & Moo Medina, 2017).

El mercado laboral exige que todo ingeniero tenga la competencia de crear aplicaciones, tal vez, no al grado de



un ingeniero en software o en sistemas, pero sí que pueda realizar rutinas que le apoyen en su trabajo, por tal motivo todo ingeniero debe programar y para ello debe desarrollar su lógica de programación. De acuerdo al U.S. Bureau of Labor Statistics (BLS) para el 2024 los trabajos en Tecnologías de la Información (TI), en comparación a 2014, se incrementarán en un 17%.

En este trabajo se analizaron los resultados en cuanto al nivel de desarrollo de la lógica de programación de los estudiantes de las carreras de Sistemas y software de las modalidades presencial y semipresencial de la extensión de la Universidad Regional Autónoma de los Andes en Santo Domingo, a los hábitos de aprendizaje de los estudiantes y métodos y técnicas de enseñanza del docente. El objetivo general es realizar un diagnóstico del nivel de desarrollo de la lógica de programación de los estudiantes de la carrera de sistemas y software de UNIANDES extensión Santo Domingo para proponer estrategias de enseñanza aprendizaje que permitan mejorar el proceso.

### MATERIALES Y MÉTODOS

Los métodos de trabajo contemplan la revisión bibliográfica principalmente en el repositorio digital de la universidad Nacional de La Plata, sobre temas referentes al desarrollo de la lógica de programación, libros referentes a la enseñanza de algoritmos y métodos de aprendizaje basado en problemas.

La investigación de campo con estudiantes de UNIANDES y PUCE SD permitió conocer de manera directa la problemática que tienen los estudiantes al momento de desarrollar un programa aplicando lógica de programación y las principales estrategias didácticas de los docentes. De igual modo se indagó sobre los hábitos de estudio y resolución de problemas por parte de los estudiantes, así como también el análisis de la malla curricular y sílabo de las asignaturas de algoritmos, lógica difusa y calculo proposicional y predicado.

Las características de esta investigación son las siguientes:

1. Tiene un alcance exploratorio: presenta informe estadístico del nivel de desarrollo de la lógica de programación, hábitos de estudio y estrategias de los docentes.

2. Tiene un alcance descriptivo, describe la realidad del proceso de enseñanza aprendizaje de algoritmos.
3. Con una modalidad cuali-cuantitativa se presentan valores cualitativos de cuantitativos que resumen los problemas a los que se enfrentan los estudiantes de sistemas y de software.

Los métodos y técnicas del nivel empírico de los conocimientos empleados son los siguientes:

1. Observación científica: se observa cual es el proceso planteamiento que emplean los estudiantes durante el proceso de enseñanza-aprendizaje.
2. Para desarrollar esta investigación se encuestó a una muestra de los estudiantes de las carreras de sistemas y software, modalidad presencial y semipresencial de UNIANDES Santo Domingo. También se encuestó a una muestra no probabilística de estudiantes de la Pontificia universidad Católica del Ecuador, extensión Santo Domingo.

Según datos proporcionados por la secretaría académica, en UNIANDES Santo Domingo, en el periodo académico abril – agosto 2018, se matricularon (Tabla 1):

Tabla 1. Población de estudiantes de UNIANDES SD, carreras de sistemas y software.

Carrera de Sistemas		Carrera de Software	
Presencial Nocturna	61	Presencial Nocturna	24
Semipresencial	32	Semipresencial	5

Dando un total de 85 estudiantes de la carrera de Sistemas y 37 estudiantes de la carrera de software. En total 122 estudiantes de UNIANDES.

### RESULTADOS Y DISCUSIÓN

El nivel de desarrollo de la lógica de programación está entre un poco, A medias y Casi totalmente, en los niveles superiores de UNIANDES SD no indican un desarrollo total, en tanto que en un 40% del sexto semestre de la PUCE SD manifiestan tener un desarrollo total de la lógica de programación en comparación con un 25% del sexto semestre de UNIANDES (Tabla 2 y 3, Figura 1).

Tabla 2. Porcentaje de Nivel de desarrollado de la lógica de programación.

Sem.	Totalmente		Casi totalmente		A medias		Un poco		Nada		% Encues.	
	P	SP	P	SP	P	SP	P	SP	P	SP	P	SP
1	7,1%		42,9%		50%		0%		0		17,5	0
2	0%	66,7%	0%	0%	50%	33,3%	50%	0%	0	0	2,5	3,8
3	0%		0%		100%		0%		0		1,3	0
5	0%	33,3%	38,5%	33,3%	46,2%	33,3%	15,4%		0	0	16,3	3,8
6	25%		25%		50%				0		5	0
7	0%	11,1%	46,2%	44,4%	53,8%	33,3%	0%	11,1%	0	0	16,3	11,3
8	0%	0%	33,3%	75%	66,7%	0%	0%	25%	0	0	3,8	5
9	0%	50%	71,4%	50%	14,3%	0%	14,3%	0%	0	0	8,8	5
<b>Total</b>	<b>3,5%</b>	<b>26,1%</b>	<b>42,1%</b>	<b>43,5%</b>	<b>47,4%</b>	<b>21,7%</b>	<b>7%</b>	<b>8,7%</b>	<b>0</b>	<b>0</b>	<b>71,3</b>	<b>28,8</b>
	10%		42,5%		40%		7,5%		0%		100%	

Tabla 3. Porcentaje de Nivel de desarrollado de la lógica de programación.

Sem.	Totalmente	Casi totalmente	A medias	Un poco	Nada	Total, enc.
1	5%	20%	30%	45%	0	71,4%
3	0	0	66,7%	33,3%	0	10,7%
6	40%	40%	20%	0	0	17,9%
<b>Total</b>	<b>10,7%</b>	<b>21,4%</b>	<b>32,1%</b>	<b>35,7%</b>	<b>0</b>	<b>100%</b>

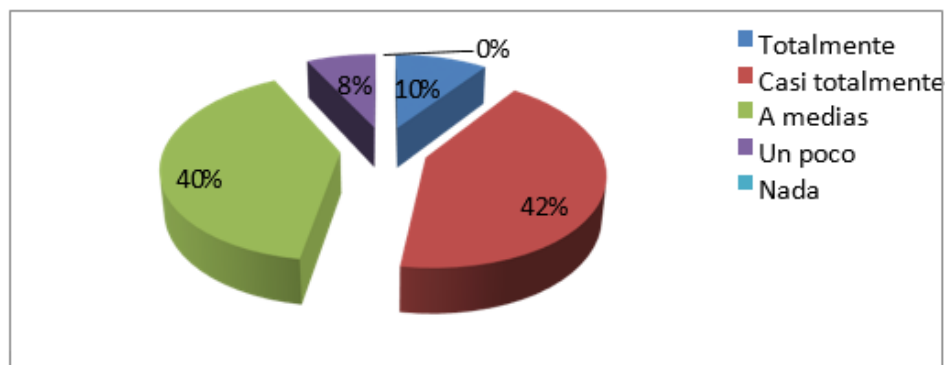


Figura 1. Por ciento del nivel de desarrollado de la lógica de programación estudiantes de UNIANDES SD.

Los estudiantes tanto de UNIANDES SD como de la PUCE SD, piensan de forma estructurada o de manera secuencial al resolver un problema de programación con un porcentaje mayor de frecuencia casi siempre y A veces (Tabla 4 y 5, Figura 2).

Tabla 4. Frecuencia de pensamiento estructural de manera secuencial.

Semestre	Siempre		Casi siempre		A veces		Rara vez		Nunca	
	%P	%SP	%P	%SP	%P	%SP	%P	%SP	%P	%SP
1	7,1		50		42,9		0		0	
2	50	0	0	66,7	50	33,3	0	0	0	0
3	0		100		0		0		0	
5	15,4	33,3	23,1	66,7	53,8	0	7,7	0	0	0
<b>6</b>	<b>0</b>		<b>25</b>		<b>75</b>		<b>0</b>		<b>0</b>	

7	23,1	0	46,2	77,8	30,8	22,2	0	0	0	0
8	0	25	66,7	50	33,3	25	0	0	0	0
9	0	50	42,9	25	57,1	25	0	0	0	0
Total	12,3	17,4	40,4	60,9	45,6	21,7	1,8	0	0	0
	13,8%		46,3%		38,8%		1,3%		0%	

Tabla 5. Frecuencia de pensamiento estructural de manera secuencial.

Sem.	Siempre	Casi siempre	A veces	Rara vez	Nunca
1	10	25	55	0	10
3	0	33,3	33,3	33,3	0
6	20	60	20	0	0
Total	10,7%	32,1%	46,4%	3,6%	7,1%

Para los estudiantes de UNIANDES SD, el aprendizaje de lógica de programación les es más fácil mediante un lenguaje de programación, mientras que los estudiantes de la PUCE SD indican que prefieren el Pseudocódigo, en la opción de otros indicaron el nombre de un lenguaje de programación, Pselnt y diagramas de flujo (Tabla 6 y 7).

Tabla 6. Aprendizaje de lógica de programación mediante.

Semestre	Lenguaje de programación		Pseudocódigo		Otro	
	%P	%SP	%P	%SP	%P	%SP
1	50		35,7		14,3	
2	0	33,3	100	66,7	0	0
3	0		100		0	
5	46,2	100	53,8	0	0	0
6	25		75		0	
7	53,8	77,8	46,2	22,2	0	0
8	33,3	75	66,7	25	0	0
9	57,1	25	42,9	75	0	0
Total	45,6	65,2	50,9	34,8	3,5	0
	51,25%		46,25%		2,5%	

Tabla 7. Aprendizaje de lógica de programación mediante.

Sem.	Lenguaje de programación	Pseudocódigo	otro
1	5	90	5
3	66,7	33,3	0
6	40	40	20
Total	17,9%	75%	7,1%

Para el desarrollo de la lógica de programación los estudiantes de UNIANDES SD y PUCE SD, recurren mayormente a la actividad de: Analizar el código de otros desarrolladores, Intenta modificar la estructura de un código fuente desarrollado en clase y Transcribe lo observado en videos de otros desarrolladores (Figuras 2 y 3).

- Transcribe lo observado en vi... 31
- Intenta modificar la estructura... 46
- Analiza el código de otros des... 45
- Copia el código de otros desa... 13



Figura 2. Actividades que realizan los estudiantes de UNIANDÉS SD para desarrollar la lógica de programación.

- Transcribe lo observado en vi... 8
- Intenta modificar la estructura... 13
- Analiza el código de otros des... 17
- Copia el código de otros desa... 3



Figura 3. Actividades que realizan los estudiantes de PUCE SD para desarrollar la lógica de programación.

A los estudiantes de UNIANDÉS SD y PUCE SD se les dificulta en: Desarrollar el algoritmo en un lenguaje de programación, Diseñar el algoritmo de solución y Analizar el problema (Figuras 4, 5, 6 y 7).

- Comprender el enunciado del ... 24
- Analizar el problema 25
- Diseñar el algoritmo de soluci... 26
- Desarrollar el algoritmo en un... 40



Figura 4. Dificultades que tienen para aprender a programar los estudiantes de UNIANDÉS SD.





Figura 5. Dificultades que tienen para aprender a programar los estudiantes de PUCE SD.

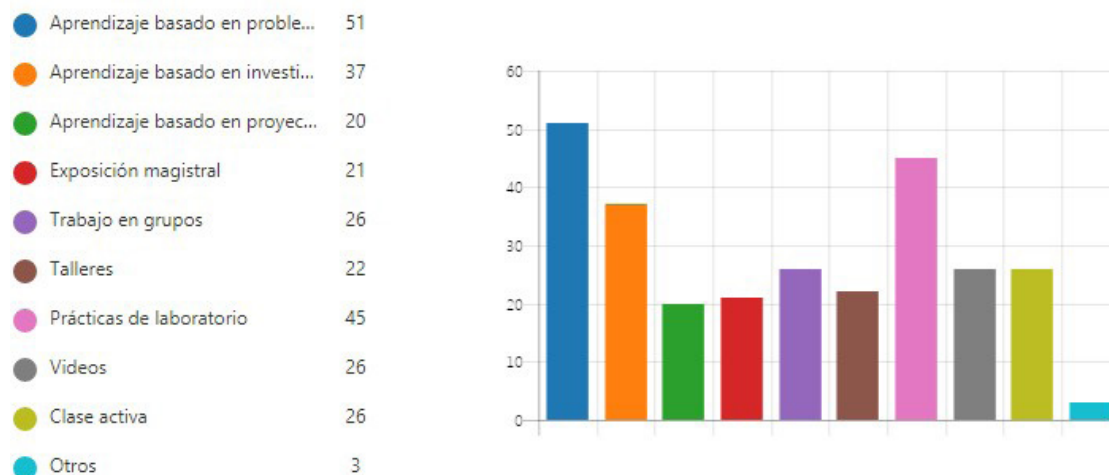


Figura 6. Métodos de enseñanza utilizados por los docentes de UNIANDES SD.

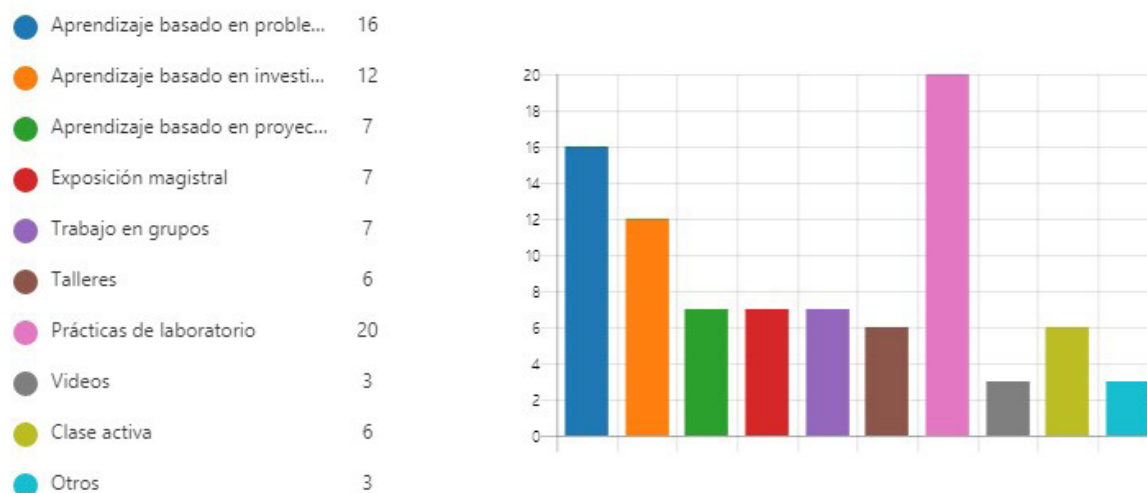


Figura 7. Métodos de enseñanza utilizados por los docentes de PUCE SD.

Demostrada la importancia de la lógica de programación en un estudiante de ingeniería de software y de sistemas no se obtuvo respuesta de un desarrollo completo de la misma por parte de los estudiantes de los niveles superiores

en UNIANDES, evidenciándose esta situación en la actividad docente realizada durante 6 años en la institución por parte del investigador.

Se ha logrado que los estudiantes al resolver un problema de programación piensen estructuradamente, de manera secuencial, una de las habilidades del pensamiento necesarias para desarrollar la lógica de programación.

De acuerdo a lo señalado en la teoría, el proceso para el aprendizaje inicial de lógica de programación debe empezar por la formulación de Pseudocódigo, en esto coinciden los estudiantes de la PUCE SD, pero los estudiantes de UNIANDES SD consideran que se debe iniciar directamente con un lenguaje de programación, esto a la larga se convierte en un obstáculo para el desarrollo efectivo de la lógica de programación.

Internet ofrece un amplio conjunto de recursos para los programadores, lo cual repercute que entre las actividades que realizan los estudiantes para desarrollar la lógica de programación, las cuales son: Analizar el código de otros desarrolladores, Intenta modificar la estructura de un código fuente desarrollado en clase y transcribir lo observado en videos. Actividades que en su mayoría no fomenta el pensamiento sino la copia de código.

Todo desarrollo de un programa de computadora parte de un algoritmo por lo tanto la principal dificultad que tienen los estudiantes es Diseñar el algoritmo de solución, sin esta etapa no tendrán éxito en el proceso de pasar el algoritmo a un lenguaje de programación.

Para el inicio en el proceso de desarrollo de la lógica de programación es necesario que los docentes utilicen métodos que fomenten el desarrollo del razonamiento y no directamente la práctica como se evidenció en los resultados es necesario la utilización de métodos como el Aprendizaje basado en problemas y aprendizaje basado en investigación, en los cuales el estudiante aprende a inferir.

## CONCLUSIONES

En el estudio se demostró que no se ha logrado un desarrollo efectivo de la lógica de programación en los estudiantes de UNIANDES Santo Domingo. Tampoco los estudiantes de UNIANDES Santo Domingo piensan de forma estructurada y secuencial al intentar resolver un problema.

En el proceso de aprendizaje de algoritmos se presta mayor atención a la práctica en el laboratorio y utilización de un lenguaje de programación, esto de forma contraria al proceso que se debe seguir para un desarrollo efectivo de la lógica de programación.

Los estudiantes aprenden a programar consultando de fuentes en internet ya sea de páginas web o de videos. La práctica de laboratorio es una de las estrategias más utilizadas por los docentes de UNIANDES para la enseñanza de algoritmos.

Es necesario reformular el sílabo de la asignatura de algoritmos incluyendo mayor contenido referente a la lógica y sus diferentes tipos.

## REFERENCIAS BIBLIOGRÁFICAS

- Alonso, L. E., & Fernández Rodríguez, C. J. (2013). *Los discursos del presente: Un análisis de los imaginarios sociales contemporáneos*. Siglo XXI de España Editores.
- Cardona Torres, S. A., Hernández Rodríguez, L. A., & Jaramillo Valbuena, S. (2010). *Lógica Matemática para Ingeniería de Sistemas y Computación*. Elizcom.
- 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. (Ponencia). *XIII Workshop de Investigadores en Ciencias de la Computación*. Rosario, Argentina.
- Farrell, J. (2013). *Introducción a la Programación lógica y diseño*. CENGAGE Learning.
- Ferreira Szpiniak, A., & Rojo, G.A. (2006). Enseñanza de la programación. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, 1(1).
- Fuentes Rosado, J. I., & Moo Medina, M. (2017). Dificultades de aprender a programar. *Educación en Ingeniería*, 12(24), 76-82.
- García Sánchez, L., Cuadrado Gallego, J. J., De Amescua Seco, A., & Velasco de Diego, M. (2002). *Construcción Lógica de programas Teoría y problemas resueltos*. Alfaomega.
- Jiménez Rey, E. M., Rodríguez, D., Britos, P. V., & García Martínez, R. (2008). Identificación de problemas de aprendizaje de programación con explotación de información. (Ponencia). *XIV Congreso Argentino de Ciencias de la Computación*. Universidad de San Luis, Argentina.
- López, L. (2013). Metodología para el Desarrollo de la Lógica de la Programación Orientada a Objetos. *Revista Iberoamericana de Sistemas, Cibernética E Informática*, 10(2), 27-32.

- Mancilla Herrera, A., Ebratt Gómez, R., & Capacho Portilla, J. (2015). *Diseño y construcción de algoritmos*. Universidad del Norte.
- Moreno Pérez, J. C. (2013). *Programación*. Ra-Ma Ediciones de la U.
- Oviedo Regino, E. (2004). *Lógica de programación*. ECOE Ediciones.
- Peñaherrera León, M. (2012). Uso de TIC en escuelas públicas de Ecuador: análisis, reflexiones y valoraciones. *EDUtec. Revista Electrónica de Tecnología Educativa*, 40, 1-16.
- Pérez Narváez, H. O., & Roig, R. (2015). Entornos de programación no mediados simbólicamente para el desarrollo del pensamiento computacional. Una experiencia en la formación de profesores de Informática de la Universidad Central del Ecuador. *RED. Revista de Educación a Distancia*, 46(9), 1-22.