

# ESTRATEGIAS PEDAGÓGICAS Y DIDACTICAS PARA SUPERAR DIFICULTADES EN EL APRENDIZAJE DE LA PROGRAMACION EN LOS NIVELES INTRODUCTORIOS

**MSc. Edilma Leonor Díaz \*, MSc. Clara Patricia Avella Ibañez \*\*, PhD John A. Bohada \***

\* **Fundación Universitaria Juan de Castellanos**, Facultad de Ingeniería y Ciencias Básicas, grupo de investigación CIyT, Tunja, Boyacá, Colombia.  
{ediaz, jbohada}@jdc.edu.co

\*\* **Universidad de Boyacá**, Tunja, Boyacá, Colombia.  
cpavella@gmail.com

**Resumen.** Una de las principales competencias exigida por la industria del software, sin importar el área de especialización es la programación en algún lenguaje de programación. Razón por la cual las instituciones de Educación Superior, ponen su mayor empeño en desarrollar bases sólidas en esta disciplina del conocimiento. Sin embargo, se ha podido establecer que un alto porcentaje de estudiantes de programas relacionados con Ingeniería de Sistemas, sienten apatía por la programación de computadores, y manifiestan su deseo de desempeñarse laboralmente en 'cualquier área que no implique construcción de software' y en ocasiones abandonan el programa. Esto se da principalmente por malas prácticas didácticas empleadas en la enseñanza de la programación que a su vez incentivan la deserción estudiantil, debida principalmente a la pérdida de asignaturas pertenecientes a las áreas de matemáticas y programación de computadores.

El objetivo de este documento es la identificación de las dificultades que influyen negativamente a la hora de aprender a programar con el fin de establecer estrategias efectivas que garanticen la aprehensión de dichos conocimientos para reducir los índices de deserción en la carrera de Ingeniería de Sistemas y afines. En este sentido, el proyecto busca identificar estrategias que aporten a la reducción de la deserción de estudiantes que adelantan estudios de Ingeniería de Sistemas y afines, a la apatía que algunos estudiantes sienten por el área de programación, al considerarla compleja. Esto se espera, ya que *la investigación busca* identificar estrategias pedagógicas y recursos didácticos a ser creados de acuerdo con los intereses de los nativos digitales, con el fin de convertirlos en herramientas atractivas y útiles para el aprendizaje de la programación. Los resultados aquí presentados se obtuvieron a partir de la aplicación de instrumentos de recolección con las opiniones de docentes y estudiantes universitarios, en torno al tema en estudio.

**Palabras claves:** Ingeniería de Sistemas, Lenguaje de programación, Herramientas didácticas, Ambientes de aprendizaje.

## **Abstract**

One of the main competencies required by the software industry, regardless of the area of expertise, is programming in a programming language. Reason why Higher Education institutions put their greatest effort into developing solid foundations in this discipline of knowledge. However, it has been established that a high percentage of students of programs related to Systems Engineering feel apathy towards computer programming, and express their desire to work in 'any area that does not involve software construction' and sometimes they leave the program. This is mainly due to bad didactic practices used in the teaching of programming, which in turn encourage student dropout, mainly due to the loss of subjects belonging to the areas of mathematics and computer programming.

The objective of this document is to identify the difficulties that negatively influence when learning to program in order to establish effective strategies that guarantee the apprehension of said knowledge to reduce dropout rates in the Systems Engineering career and related. In this sense, the project seeks to identify strategies that contribute to the reduction of the desertion of students who carry out studies in Systems Engineering and related, to the apathy that some students feel towards the programming area, considering it complex. This is expected, since the research seeks to identify pedagogical strategies and didactic resources to be created according to the interests of digital natives, in order to turn them into attractive and useful tools for learning programming. The results presented here were obtained from the application of collection instruments with the opinions of teachers and university students, regarding the subject under study.

## 1. INTRODUCCIÓN

El principal problema que se ha evidenciado en los primeros semestres de programas de Ingeniería de Sistemas, es el aprendizaje de la programación, situación que, aunque preocupa a los docentes que orientan la asignatura no se ha reconocido su importancia ya que se atribuye a la falta de habilidades cognitivas para la solución de problemas y que con el tiempo se convierte en causa de deserción, situación que se evidencia tanto a nivel nacional como internacional. Es así, que uno de los estudios adelantados a nivel nacional, es el trabajo exploratorio realizado por Insuasti J. (2016) frente a dicha problemática, el cual indica que las estrategias de enseñanza y aprendizaje han experimentado cierto retraso ya que, a pesar de existir múltiples escenarios, herramientas tecnológicas para complementar la formación impartida en el aula de clase, hay una marcada tendencia hacia el uso de ejemplos y el desarrollo posterior de ejercicios. Indica también, que el principal aporte del estudio es la identificación de esfuerzos aislados, para lo que plantea la creación de un “repositorio de experiencias exitosas” que debería conformarse como solución frente a la problemática planteada. Sin embargo, resalta que sin una adecuada apropiación didáctica de estos recursos se podría desperdiciar el potencial de los materiales creados. Por tal razón, propone integrar estos recursos donde se narre la vida dentro del aula y de forma explícita se pueda evidenciar cómo el profesor realiza el ejercicio de docencia, para así enriquecer el conocimiento sobre la enseñanza de la programación, a partir de las experiencias exitosas.

Por otra parte, los manifiestos de la Association for Computing Machinery (ACM por su siglas en inglés) en materia de reformas curriculares y planteamientos de didácticas (ACM, IEEE, AIS, 2020) hacen un amplio reconocimiento entre académicos en Ciencias Computacionales, sobre la necesidad de hacer algo para contrarrestar este problema; varios programas a nivel mundial han adoptado medidas energéticas para compensar estos puntos de dificultad.

A partir de estos estudios y viendo la necesidad de profundizar en dicha problemática, se ha planteado adelantar una investigación que permita identificar las principales dificultades que tienen o que presentan los estudiantes en los cursos introductorios de programación, y, además, tomando como base las experiencias de profesionales de la disciplina, buscar acciones para contrarrestar las dificultades a la hora de aprender a programar y a partir de estas, formular una metodología que contribuya a superar dichas dificultades y a la vez que garanticen la culminación

de sus estudios, contribuyendo en la disminución de la deserción que por esta situación se puede presentar en la formación profesional. Para lograr este cometido, se llevó a cabo análisis a nivel nacional sobre la percepción de docentes y estudiantes de programación respecto a las dificultades en el proceso de enseñanza-aprendizaje de la programación, respecto al material complementario ofrecido por los profesores y consultado por los estudiantes, el aporte de los video tutoriales, de los blogs o sitios web, de los libros, deficiencias del material didáctico existente (video tutoriales, blogs, cartillas didácticas, libros, etc.), recursos didácticos que deberían ofrecer los profesores como material complementario, para así, identificar las debilidades y fortalezas, y en consecuencia, proponer acciones que 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 de la deserciones.

## 2. MATERIALES Y MÉTODOS

Para el desarrollo de esta investigación, se tomó como base una metodología descriptiva-explicativa dado que se pretende mostrar a fondo la realidad de la enseñanza-aprendizaje de la programación a nivel universitario. Para ello, en primer lugar, se realiza una indagación bibliográfica en bases de datos científicas, bases de datos académicas e incluso, publicaciones realizadas en revistas no necesariamente de tinte tecnológico que, dado la novedad del tema, realizan análisis interesantes respecto a los procesos y falencias de la enseñanza-aprendizaje de la programación. En segundo lugar, se realiza un estudio de percepción entre docentes y estudiantes en el área objeto de estudio, para ello, en el segundo semestre del año 2018 se realizó un estudio cuantitativo, a través de la aplicación de encuestas online dirigidas a profesores de programación de computadores y estudiantes de Ingeniería que cursaban como mínimo una asignatura de programación en alguna de las universidades colombianas, con modalidad presencial y virtual.

Las encuestas de docentes fueron revisadas y validadas por cuatro profesores universitarios del área disciplinar de la Ingeniería de Sistemas de universidades de la ciudad de Tunja y un profesor de licenciatura en Informática, quienes emitieron sus observaciones y sugerencias para mejorar las preguntas formuladas (18 en total relacionadas a las dificultades que se presentan en los estudiantes para la enseñanza-aprendizaje de la programación y el material educativo complementario utilizado en el proceso de enseñanza-aprendizaje). Para la aplicación de las encuestas se contó con el apoyo de REDIS (Red

de Directores de Ingeniería de Sistemas) Colombia, quienes compartieron las encuestas con profesores y estudiantes de cursos de programación de las universidades a las cuales se encuentra vinculados laboralmente, las cuales fueron respondidas por 36 docentes y 101 estudiantes.

Finalmente, con el análisis realizado y el estudio obtenido, se plantean una serie de estrategias pedagógicas para mejorar el proceso de enseñanza-aprendizaje de la programación en los programas profesionales que tienen en su pensum dichas asignaturas.

### **3. RESULTADOS Y DISCUSIÓN**

#### **3.1 Análisis del proceso de enseñanza-aprendizaje de la programación**

Uno de los temas contemplados dentro de esta investigación es la identificación de los factores que inciden a la hora de aprender a programar ya que esta es una de las asignaturas que influye significativamente en la deserción, no solo a nivel de las instituciones de educación superior de la región sino también a nivel internacional.

Es por ello que el primer contacto con la programación es clave en la formación de profesionales en Ingeniería de Sistemas, Fuentes & Moo (2017) exponen que un ingeniero debe ser capaz de codificar sus ideas ya sea para hacer experimentos y simulaciones de su propuestas de solución, así como crear soluciones de software razón por la cual si el estudiante no desarrolla las competencias debidas se verá afectado en el aprovechamiento de las materias subsecuentes, convirtiéndose en una de las causas de las altas tasas de deserción a nivel de educación superior. Prueba de esta afirmación se encuentra en la Investigación Doctoral realizada por Bosse & Gerosa (2016), la cual permitió establecer que los estudiantes se enfrentan a muchas dificultades mientras aprenden la lógica de la programación, mediante la identificación de patrones de dificultad relacionados con el aprendizaje de cómo programar, una parte crucial de la formación de ingenieros de software y que con frecuencia los conduce a la deserción.

De igual forma, Qian & Lehman (2017) expresan que, aunque se utilizan diferentes nombres para representar las dificultades de los estudiantes en el aprendizaje de la programación, en los cursos de programación introductoria, las dificultades experimentadas por los estudiantes están relacionadas con muchos factores, resaltando que las fuentes de dificultades de los

estudiantes tienen conexiones con los conocimientos previos.

Gomes & Mendes (2014) expresan que el aprendizaje de la programación en novatos exige esfuerzo y motivación, sin embargo, la desmotivación es común en muchos estudiantes que no pueden hacer frente a las dificultades naturales asociadas con el aprendizaje de programación. Dado que muchos estudiantes carecen de motivación propia, depende del maestro encontrar estrategias para ayudar a los estudiantes y mantenerlos motivados durante el curso, razón por la cual su investigación se centró en conocer las estrategias pedagógicas y motivacionales utilizadas por los maestros en las instituciones para promover la motivación de los estudiantes en programación.

Por otra parte, Shuchi & Satabdi (2017) a través de su estudio resaltan que los estudiantes generalmente no están familiarizados con el uso de variables, y albergan conceptos erróneos sobre ellas y que también tienen problemas con otros aspectos de la programación introductoria, cómo funcionan los bucles y cómo funcionan los operadores booleanos. Estos hallazgos evidencian la necesidad de una pedagogía que combine actividades constructoras populares con las que se integre el aprendizaje conceptual, junto con un mejor desarrollo profesional para apoyar el aprendizaje conceptual por parte de los maestros.

Un aporte significativo dentro de la investigación lo realiza ACM (ACM, IEEE, AIS, 2020) quien en materia de reformas curriculares y planteamientos de didácticas ha propuesto a toda la comunidad mundial un conjunto de áreas del conocimiento que reflejan una visión actual de la informática, incluida la incorporación de nuevas sub-disciplinas emergentes como la ciberseguridad y la ciencia de datos y propone una caracterización de la informática basada en competencias, así como una organización ascendente de la informática en sus sub-disciplinas basadas en un análisis de competencias.

Rubiano & Lopez (2015), a través de su investigación realizada en universidades de Bogotá, permite afirmar que las dificultades en la enseñanza de la programación y el no logro de los objetivos en el proceso de instrucción, son el resultado del diseño de cursos, que no tienen en cuenta la didáctica orientada al desarrollo de habilidades de resolución de problemas, la comprensión lógica, y abstracción del mundo real, expresión escrita y/o uso adecuado del lenguaje natural y que la evaluación de los estudiantes debe dejar de utilizarse como instrumento coercitivo para convertirse en un verdadero método de evaluación de la enseñanza y el aprendizaje.

Rodríguez (2014) en su investigación, expresa que una metodología adecuada para la enseñanza de los cursos iniciales de programación de computadoras, debe incorporar varios factores desde la pertinencia de la temática ofertada: objetivos de aprendizaje y pre-conceptos del estudiante, hasta el contexto de los programas académicos para los que se oferta el curso.

Alturki (2016) a través de su artículo expone que los resultados de los estudiantes en los cursos de programación introductoria presentan muchas variaciones las cuales son el resultado de los métodos de enseñanza, competencia del profesorado en la materia, antecedentes y habilidades de codificación de los estudiantes, autodisciplina de los estudiantes, el entorno de enseñanza, y los recursos disponibles para los estudiantes, todos los cuales pueden afectar el rendimiento y los resultados, factores que contribuyen a la deserción. Razón por la cual a través de su estudio expone que los estudiantes toman los exámenes en serio, pero dedican menos atención a las tareas y al trabajo de laboratorio para corregir dicha situación propone simplificar la estructura del curso y reducir su complejidad, además de ofrecer a los estudiantes menos métodos de evaluación, reduciendo el peso de los laboratorios, tareas y cuestionarios, y aumentar el de los exámenes finales y de mitad de período.

Medeiros, Ramalho & Falcão (2019) a través de su investigación encontraron que las habilidades más necesarias para el aprendizaje de la programación estaban relacionadas con la resolución de problemas y la habilidad matemática, de igual manera se citó como un desafío de aprendizaje, seguido de motivación y compromiso, y dificultades para aprender la sintaxis de los lenguajes de programación; los principales desafíos de enseñanza se refieren a la falta de métodos y herramientas apropiados.

Por otra parte, Carbonaro & Ravaioli (2017) proponen que para mejorar el aprendizaje en los cursos de programación es necesario desarrollar nuevas herramientas proponiendo la evaluación por pares basada en la web, ya que puede estimular los intereses de los estudiantes en el aprendizaje y mejorar su conocimiento sobre el aprendizaje activo.

Gomes & Ke (2017) consideran que aprender a programar no es una tarea fácil para muchos estudiantes y que contribuye a encontrar altas tasas de deserción y fracaso en los cursos de programación introductorios. El objetivo de su investigación era conocer más sobre las estrategias utilizadas por los profesores de diferentes instituciones y culturas para promover el aprendizaje de programación y la

motivación de los alumnos, encontrando que la relación alumno-profesor, las estrategias motivacionales utilizadas, los métodos de evaluación, los materiales presentados en clase son cruciales a la hora de atraer a los estudiantes y animarlos hasta alcanzar un buen nivel de programación.

Santimateo (2018) presenta un estudio en el cual analiza la percepción de los estudiantes de licenciatura en informática de varias universidades de Panamá, respecto a la dificultad para aprender a programar, la utilidad de los recursos y ambientes de aprendizaje, identificando actividades educativas que menos facilitan el logro de nuevos aprendizajes en su respectivo orden son: las conferencias y el estudio en grupo. Por otra parte, son actividades útiles, el uso de ejemplos para modificar y las explicaciones del docente. Los vídeos, ejemplos de programas y tutoriales web son los recursos de mayor utilidad para los estudiantes.

Por otra parte es posible encontrar en los EVA, un recurso útil en el proceso enseñanza aprendizaje ya que cuenta con diversas herramientas y metodologías que permiten la transmisión de conocimientos, entre las que se menciona e-learning, dejando de ser simplemente, depósitos de un curso en un computador; sino que también se han de considerar los diversos recursos de apoyo que facilitarán el proceso de enseñanza y en especial el soporte informático o plataforma sobre la cual se implantan recursos de aprendizaje que deben contar con un alto grado de usabilidad para garantizar su uso para lo cual se han adelantado investigaciones dentro de la cual vale la pena mencionar la guía para la evaluación de usabilidad en entornos virtuales de aprendizaje Alarcon (2014), a partir de la identificación de algunos criterios que permiten cuantificar el grado de satisfacción de los usuarios de este tipo de herramientas que podrían ser otra opción para la enseñanza de la programación.

Muchas de las dificultades presentes en el proceso de enseñanza y aprendizaje de la programación de computadores que existen en Colombia se hacen visibles en estudios como el de Rahim & Badioze (2018) de Indonesia quienes presenta resultados similares con estudiantes de Malasia, resaltando la necesidad de establecer buenas estrategias y herramientas didácticas para proporcionar al estudiante guías óptimas y apoyo de aprendizaje. Es importante resaltar que los resultados obtenidos corresponden a la percepción de expertos, permitiendo identificar que la resolución de problemas es un problema importante que se experimenta al enseñar y aprender lenguajes de programación, proporcionando

información sobre los métodos actuales utilizados para enseñar lenguaje de programación y recomiendan programas de visualización y juegos como herramientas de aprendizaje para mejorar los métodos actuales de enseñanza-aprendizaje.

Lo anterior evidencia la necesidad de implementar una iniciativa pedagógica en los cursos del área de la programación como se evidencia en la Universidades de Granada, Almería y Murcia (España), que enfaticen en subsanar este aspecto, para lo cual Hernández & Fernández (2014) expresan que las dificultades en la enseñanza de la programación y su aprendizaje es un problema grave no solo por los conceptos importantes que subyacen y estructuran el curso, sino también por la falta de motivación, el fracaso y el abandono que tal frustración puede implicar para el estudiante. Para lo cual describe un sistema que permite sugerir ejercicios y evaluar los resultados automáticamente a través del seguimiento inmediato y personalizado de cada alumno con el fin de construir un perfil de las competencias de cada estudiante que contribuirá a desarrollar habilidades en la programación introductoria.

En este mismo sentido Ramirez-Lopez & Muñoz (2015) en su investigación realizada en el Instituto Tecnológico Autónomo de México. Río Hondo, demostró que los ejemplos relacionados con las carreras de ingeniería tienen un impacto muy positivo en el aprendizaje y despiertan el interés de los estudiantes en los cursos de programación; en comparación con los estudiantes que siguen ejemplos de libro de texto como un curso convencional.

También se tiene la percepción, por parte de los profesores, de que el material educativo existente para apoyar los cursos de programación hace énfasis en la sintaxis de los lenguajes de programación, más que en el desarrollo de la lógica, así que para solucionar esto es necesario definir una metodología apropiada para abordar el problema a solucionar desde el inicio de los cursos, que estimule en el estudiante la capacidad de seguir un esquema ordenado y lógico para alcanzar los objetivos esperados, con mayor rapidez, efectividad y eficiencia, sustentado en procesos formales de ingeniería. Para sustentar esta afirmación Xinogalos (2016) de Macedonia Grecia, plantea que los cursos de introducción a la programación deben utilizar:

- Pseudo-lenguaje, pues son menos estrictos que los lenguajes de programación convencionales y ayudan a los estudiantes a concentrarse en los aspectos más importantes, que son sin duda los conceptos/construcciones algorítmicos/programación y no su sintaxis.

- La transición de programación procedimental a la programación orientada a objetos no es fácil para los estudiantes, ya que les resulta difícil cambiar su mentalidad y utilizar clases y objetos en lugar de funciones como los conceptos principales para idear una solución a un problema.

- Los estudiantes evalúan positivamente la secuencia de técnicas e idiomas enseñados y las medidas tomadas para apoyarlos en su transición de una técnica a la otra.

- Conocer las dificultades de los estudiantes es importante para el instructor con el fin de distribuir adecuadamente el tiempo disponible a las diversas unidades de aprendizaje e idear situaciones didácticas especiales para apoyar estudiantes a hacer frente a las dificultades subyacentes.

- Prestar especial atención en el diseño de la enseñanza y aprendizaje de un curso de programación.

- El material utilizado por los estudiantes para estudiar, debe estar disponible para ellos, ya que dependen en gran medida del material preparado por los instructores razón por la cual los instructores deben preparar presentaciones de alta calidad, ejercicios de laboratorio, tareas y soluciones ejemplares con el fin de apoyar a los estudiantes en el aprendizaje de programación.

Por otra parte, es importante mencionar que los estudios han demostrado que el aprendizaje de la programación no tiene un rango especial de edad, y aseguran que entre más temprano se enseñe, es mejor, para lo cual, las áreas de matemáticas, estadística y geometría se podrían abordar utilizando estructuras de control a nivel básico, complementadas con proyectos que planteen situaciones particulares, juego con objetivos, reglas, niveles definidos relacionados a través de una historia como lo expresa Palma (2015) de México. En ese mismo sentido, los lenguajes de programación textual, la inclusión de un pensamiento crítico, creativo e innovador, debe motivar a los sistemas educativos para preparar a los jóvenes en el dominio de un nuevo lenguaje que les permita desenvolverse en el mundo digital, proporcionando las herramientas cognitivas necesarias para desenvolverse en un mundo digital que se encuentra en constante cambio como lo propone Llorens (2017) de la Universidad de Salamanca España.

Por otra parte Gallego & Villagrà (2017) de la Universidad de Alicante España, a través de su artículo concluyeron que el problema principal era un cambio en los intereses de los estudiantes. Los estudiantes crecieron jugando juegos y estaban acostumbrados a la retroalimentación inmediata y a un sentido preciso de progresión, las actividades que carecen de ambas se vuelven menos interesantes para ellos la solución

produjo el juego PLMan y un sistema de evaluación automatizado que transformó las actividades del aula en jugar PLMan., evidenciando que los juegos de computadora pueden ser una herramienta útil para el aprendizaje de la programación, debido a que el juego de computadora puede guiar el aprendizaje paso a paso para lograr el objetivo del proceso de aprendizaje basado en la habilidad del alumno y sus preferencias. Lo anterior, hace ver la necesidad de contar con material educativo que permita a los estudiantes crear, imaginar, inventar, y que la vez contribuyan a estimular las actividades grupales, el trabajo colaborativo, así como valores básicos de convivencia (Díaz, Banchoff, Queiruga, & Martin, 2015) de la Plata Argentina. En ese mismo sentido se pueden utilizar herramientas lúdicas (juegos) que ponen a prueba las habilidades, tanto de programación, como de lógica, que resultan más atractivas para los estudiantes como lo propone Dapozo (2016) de Argentina.

Como fruto de estos procesos de investigación vale la pena mencionar algunas propuestas como la de Zatarain & Barrón (2018) de México, presentan un entorno basado en la web para aprender programación Java que tiene como objetivo proporcionar instrucción de programación adaptadas e individualizadas como un sistema de recomendación y minería, un reconocedor de afecto, un analizador de sentimientos y una herramienta de autoría. Los resultados de las evaluaciones muestran que los estudiantes percibieron el disfrute y están dispuestos a usar la herramienta y también muestra que los estudiantes que usan la herramienta tienen una mayor ganancia de aprendizaje que aquellos que aprenden usando un método tradicional.

En ese mismo sentido Colucci & Kostandy (2015) de New York, presentan RadStax que es una aplicación basada en la Web desarrollada utilizando software gratuito, de código abierto y ubicuo. Es una aplicación novedosa para cursos introductorios que desean crear un atlas de estudios radiológicos MPR etiquetados y diseñados para satisfacer las necesidades específicas de su plan de estudios. Simple y enfocado, proporciona una experiencia interactiva para estudiantes similar a la práctica de radiólogos. Este programa es una herramienta sólida de enseñanza de anatomía que ayuda eficazmente a educar al estudiante de medicina preclínica.

### 3.2 Estudio de Percepción respecto a la enseñanza de la programación

Teniendo en cuenta el análisis realizado en la sección anterior sobre las dificultades que los estudiantes

tienen respecto a aprender a programar, podemos deducir que las principales causas están en el desarrollo de la lógica de programación, solucionar un ejercicio diferente al explicado por el profesor, entender la programación orientada a objetos y/o entender la sintaxis de un lenguaje de programación, se realizó un estudio sobre la percepción entre estudiantes y docentes respecto a estas dificultades. Los resultados obtenidos son los que se muestran en la tabla 1.

Tabla 1 Percepción de los profesores y estudiantes sobre las dificultades de los estudiantes para aprender a programar

Dificultad	Porcentaje de respuestas	
	Profesores	Estudiantes
El desarrollo de la lógica de programación.	91.7 %	45.5%
Solucionar un ejercicio diferente al explicado por el profesor.	47.2 %	ON
Entender la programación orientada a objetos.	38.9 %	22.8 %
Entender la sintaxis de un lenguaje de programación.	19.4 %	32.7 %
Ninguna, me ha parecido fácil aprender a programar.	ONP	22.8 %
Otras	2.8 %	1%

ONE: Opción no incluida en encuesta a estudiantes;  
ONP: No incluida en encuesta a profesores

Fuente: elaboración propia a partir de encuestas aplicadas a profesores y estudiantes

En dicho estudio se evidencio que los docentes y estudiantes perciben con un 91,7% y 45,5% respectivamente, que el desarrollo de la lógica de programación es una de las principales dificultades que se enfrentan a la hora de aprender a programar. De igual forma, los docentes perciben con un 47,2% que cuando se hace cambios en los enunciados de los ejercicios desarrollados en clase, los estudiantes se confunden o no saben que realizar. Esta característica va de la mano con problemas de lectura y comprensión que en los primeros semestres de la carrera se evidencia por falencias que traen en este aspecto desde la educación básica y media.

Adicional al estudio anterior, se realizaron dos estudios uno a docentes (ver tabla 2) y otro a estudiantes (ver tabla 3) respecto a los tipos de material complementario que se usa para mejorar el proceso de enseñanza-aprendizaje de la programación. En el primer estudio se evidenció por parte de los docentes que el 69% de los video tutoriales es el material que mayormente usan para la enseñanza de la programación, seguido del uso de libros de programación y la producción de material de autoría del docente, ambos, con un 66,7%.

Tabla 2 Material complementario ofrecido por los profesores para el aprendizaje de la programación

Material complementario	Porcentaje de respuestas de los profesores
Video tutoriales	69.4%
Libros de programación	66.7%
Material didáctico de autoría propia	66.7%
Blog o sitios web que muestran ejemplos	41.7%
Video Conferencias	16.7%
Otros	2.8%

Fuente: elaboración propia a partir de encuestas aplicadas a profesores

Para el caso de los estudiantes, éstos indican que prefieren para el proceso de aprendizaje de la programación el uso de video tutoriales en la web, con un 73.3%, seguido de blogs o sitios web que publican el desarrollo de ejercicios, con un 44,6% o la solicitud de explicación a sus compañeros con un 40,6%. Esta situación refleja el rompimiento entre la relación enseñanza-aprendizaje y su relación docente-estudiante por cuanto estos últimos prefieren fuentes externas y no, seguramente, las indicadas o producidas por los docentes.

Tabla 3 Material complementario empleado por los estudiantes en el aprendizaje de la programación

Material complementario	Porcentaje de respuestas de los estudiantes
Video tutoriales en la Web	73.3%
Blogs o sitios Web que muestran ejercicios	44.6%
Pido explicación a compañeros o monitores	40.6%
Libros de programación	24.8%

No complemento, me quedo con lo explicado por el profesor	2%
---	----

Fuente: elaboración propia a partir de encuestas aplicadas a estudiantes

Para entender un poco más la percepción que tienen tanto docentes como estudiantes respecto a los materiales complementarios usados, se tomó como base los materiales relevantes y se les consulto para determinar el grado de satisfacción respecto a ellos. Los resultados obtenidos son los que se presentan desde la tabla 4 hasta la tabla 6.

Respecto al aporte de los videos tutoriales para la enseñanza-aprendizaje de la programación, se observa que para estudiantes y docentes con un 41,7% y 60,4%, respectivamente, son buenos, y en un menor grado los consideran deficientes tal y como se muestra en la tabla 4.

Tabla 4 Percepción de los profesores y estudiantes sobre el aporte de los video tutoriales para el aprendizaje programación

Percepción	Porcentaje de respuestas	
	Profesores	Estudiantes
Excelentes, son una gran ayuda para el aprendizaje de la programación	16.7%	17.8%
Buenos, aportan en gran medida al aprendizaje de la programación	41.7%	60.4%
Regulares, poco aportan al aprendizaje de la programación	22.2%	10.9%
Deficientes, no sirven para el aprendizaje de la programación	8.3%	1%
No he visto video tutoriales sobre programación	11.1%	9.9%

Fuente: elaboración propia a partir de encuestas aplicadas a profesores

Respecto al aporte de los blogs o sitios web para la enseñanza-aprendizaje de la programación, se observa que para estudiantes y docentes con un 44,4% y 56,4%, respectivamente, son buenos, y en un menor grado los consideran deficientes por parte de los docentes tal y como se muestra en la tabla 5.

Tabla 5 Percepción de los profesores y estudiantes sobre el aporte de los blogs o sitios web para el aprendizaje programación

Percepción	Porcentaje de respuestas	
	Profesores	Estudiantes
Excelentes, son una gran ayuda para el aprendizaje de la programación.	11.1%	12.9%
Buenos, aportan en gran medida al aprendizaje de la programación	44.4%	56.4%
Regulares, poco aportan al aprendizaje de la programación	22.2%	6.9%
Deficientes, no sirven para el aprendizaje de la programación	8.3%	0%
No he visto blogs o sitios web sobre programación	14%	23.8%

Fuente: elaboración propia a partir de encuestas aplicadas a profesores

Respecto al aporte de los libros para la enseñanza-aprendizaje de la programación, los docentes y estudiantes consideran que éstos son buenos en un 55,6% y 32,7%, respectivamente. Sin embargo y paradójicamente, los estudiantes manifiestan en un 38,6%, reforzado la situación que prefieren los materiales web o digitales, tal y como se muestra en la tabla 6.

Tabla 6 Percepción de los profesores y estudiantes sobre el aporte de los libros para el aprendizaje de la programación

Percepción	Porcentaje de respuestas	
	Profesores	Estudiantes
Excelentes, son una gran ayuda para el aprendizaje de la programación	27.7%	19.8%
Buenos, aportan en gran medida al aprendizaje de la programación	55.6%	32.7%

Regulares, poco aportan al aprendizaje de la programación	16.7%	8.9%
Deficientes, no sirven para el aprendizaje de la programación	0%	0%
No he consultado libros de programación	0%	38.6%

Fuente: elaboración propia a partir de encuestas aplicadas a profesores

De igual forma, es importante conocer la percepción de los profesores con respecto a las deficiencias del material didáctico existente, en este sentido, consideran que la principal deficiencia del material existente para el enseñanza-aprendizaje de la programación, con un 58,3%, es el enfoque que se le da a la sintaxis del lenguaje y no en la identificación del problema, en un porcentaje similar, 55,6%, está que los ejercicios propuestos no solucionan problemas de la vida real o que no se enfocan en el desarrollo de la lógica de programación con un 52,8%, tal y como se muestra en la tabla 7.

Tabla 7 Percepción de los profesores sobre las deficiencias del material didáctico existente

Percepción	Porcentaje de respuestas de los profesores
Se enfocan en la sintaxis del lenguaje y no en la identificación del problema.	58.3%
Los ejercicios no solucionan problemas de la vida real	55.6%
No se enfocan en el desarrollo de la lógica de programación.	52.8%
Los ejercicios propuestos son muy fragmentados	36.1%
Otros aspectos	2.8%

Fuente: elaboración propia a partir de encuestas aplicadas a profesores

Finalmente, se muestra la percepción de los estudiantes sobre los recursos didácticos que deberían ofrecer los profesores como material complementario. En este aspecto, ellos manifestaron con un 54.5%, que les gustaría se les ofreciera como material complementario para el aprendizaje de la programación los video juegos educativos. De igual forma, se destaca con un 49,5% los video tutoriales, con un 46.5% las bibliotecas de retos de programación y los simuladores educativos con un 44.4%. Vale la pena destacar que consideran importante en un menor



grado que los docentes deberían ofrecer: ejercicios de programación orientados al ámbito empresarial, desarrollo conjunto de ejercicios tipo Mob programming, manuales y guías, tal y como se evidencia en la tabla 8.

Tabla 8 Percepción de los estudiantes sobre el material educativo complementario que deberían ofrecer los profesores

Recurso didáctico	Porcentaje de respuestas de los estudiantes
Video juegos educativos	54.5%
Video tutoriales	49.5%
Bibliotecas con retos de programación	46.5%
Simuladores educativos	44.4%
Ejercicios resueltos	35.4%
Libros y cartillas digitales	25.3
Otros	1%

Fuente: elaboración propia a partir de encuestas aplicadas a estudiantes

### 3.3 Análisis de los resultados

Las dificultades identificadas en el proceso de enseñanza y aprendizaje de la programación que existe en las universidades de Boyacá, no son exclusivas de nuestra región, Santimateo (2018), presenta resultados similares con estudiantes de Panamá. Es importante destacar que la información obtenida corresponde a la percepción de estudiantes y estudiantes consultados. Según la percepción general de docentes y estudiantes, la principal dificultad que presentan los estudiantes es el desarrollo de la lógica de programación, y que los materiales didácticos existentes son considerados deficiencias ya que se centran en la sintaxis del lenguaje y no en la identificación del problema. Resultados similares a los obtenidos por Rubiano & Lopez (2015) y Xinogalos (2016).

Respecto al material complementario ofrecido por los docentes y empleado por los estudiantes estos perciben que los videos tutoriales y blogs son los más utilizados ya que aportan en gran medida al aprendizaje de la programación.

En cuanto al aporte de los libros el 55,6% de los docentes los consideran recursos útiles para el aprendizaje de la programación y el 38,6% de los estudiantes no los utilizan evidenciado que para los

últimos los recursos que requieren mayor lectura son considerados menos útiles.

Los estudiantes consideran que el material educativo complementario que deberían ofrecer los profesores en su orden son: 54,5% Video juegos educativos, 49,5% Video tutoriales, 46,5% Bibliotecas con retos de programación, 44,4% Simuladores educativos, 35,4% Ejercicios resueltos evidenciado la necesidad de desarrollar nuevas herramientas basadas en la Web, resueltos similar al resultado del estudio de Carbonaro & Ravaoli (2017).

El estudio y la participación en el proyecto Programium, será una solución que complementará las herramientas y sitios Web existentes para la enseñanza de la programación, los cuales, en su mayoría, no se enfocan en procesos de análisis de problemas, ni diseño de soluciones, sino en la sintaxis del lenguaje de programación, dejando de lado aspectos de suma importancia como pedagogía, diseño de aplicaciones y buenas prácticas de programación.

### 3.4 Estrategias pedagógicas para la enseñanza-aprendizaje de la programación

Tomando como base los estudios realizados a la forma como ven los investigadores los procesos de enseñanza-aprendizaje de la programación, el estudio de sus falencias y las diferentes percepciones que tienen docentes y estudiantes respecto a los materiales usados en este aspecto, se propone una metodología que permita ayudar a mejorar dicho proceso y, por ende, disminuir los problemas que se tienen a la hora de abordar los temas de programación. En este sentido se dan las siguientes recomendaciones:

- Ofertar temáticas pertinentes en las cuales se contemplen los objetivos de aprendizaje y pre-conceptos del estudiante según el programa académico.
- Conocer las dificultades de los estudiantes, con el fin de distribuir adecuadamente el tiempo disponible a las diversas unidades de aprendizaje e idear situaciones didácticas especiales para apoyar a los estudiantes a hacer frente a las dificultades subyacentes.
- La relación alumno-profesor, las estrategias motivacionales utilizadas, los métodos de evaluación, los materiales presentados en clase son cruciales a la hora de atraer a los estudiantes y animarlos hasta alcanzar un buen nivel de programación.
- Desarrollar ejercicios y evaluar los resultados automáticamente a través del seguimiento inmediato y personalizado de cada alumno con el fin de construir un perfil de las competencias que promuevan el

aprovechamiento de las materias siguientes en la carrera.

- Definir estrategias pedagógicas que integre el aprendizaje conceptual y motivacional que los mantenga motivados y comprometidos durante el curso a los estudiantes en programación.
- Establecer una didáctica orientada al desarrollo de habilidades para la resolución de problemas, la comprensión lógica, y abstracción del mundo real, expresión escrita y/o uso adecuado del lenguaje natural.
- Incluir actividades educativas como el uso de ejemplos para modificar y que complementados con las explicaciones de docentes creen un pensamiento crítico, creativo e innovador en los estudiantes, proporcionando las herramientas cognitivas necesarias para desenvolverse en un mundo digital que se encuentra en constante cambio.
- Recursos que permite sugerir ejercicios y evaluar los resultados automáticamente a través del seguimiento inmediato y personalizado de cada alumno con el fin de construir un perfil de las competencias de cada estudiante que contribuirá a desarrollar habilidades en la programación introductoria.
- Desarrollar nuevas herramientas basadas en la web (videos, tutoriales), ya que puede estimular el interés de los estudiantes en el aprendizaje y mejorar su conocimiento sobre el aprendizaje activo.
- Se recomiendan utilizar programas de visualización y juegos como herramientas de aprendizaje para mejorar los métodos actuales de enseñanza-aprendizaje.

#### 4. CONCLUSIONES

El estudio adelantado se identifica con otras investigaciones en lo relacionado a las principales dificultades a que se enfrentan los estudiantes a la hora de aprender a programar y en este caso el desarrollo de la lógica, así como los recursos con que se puede apoyar el proceso de enseñanza-aprendizaje para garantizar el aprendizaje de la programación. Con la investigación se tiene claridad respecto a que la motivación juega un papel clave dentro del proceso ya que es el punto de partida sobre el cual el estudiante se anima a cursar la asignatura y que se complementa con los recursos, actividades y ambientes de aprendizaje que presenta el docente como apoyo dentro del proceso y que contribuye a la enseñanza de la programación básica pudiéndose establecer que los más apropiados son los recursos digitales, los cuales requieren la integración de diversas áreas del conocimiento como: matemáticas, idiomas, lógica matemática, entre otras.

En ese sentido corresponde realizar experimentos con estrategias sugeridas como por ejemplo, mejorar las habilidades matemáticas de los estudiantes utilizando estructuras de programación a nivel básico, definir una metodología apropiada para abordar el problema a solucionar desde el inicio de los cursos, estimular en el estudiante la capacidad de seguir un esquema ordenado y lógico para alcanzar los objetivos esperados, apoyados en juegos con objetivos, reglas, niveles definidos, los cuales contribuyen a desarrollar habilidades necesarias para el curso de programación el cual a su vez se complementara con video tutoriales que aporte al aprendizaje de programación, libros y cartillas digitales en los cuales se incluya una buena cantidad de ejercicios resueltos y retos a desarrollar por parte de los estudiantes, simuladores educativos, material educativo que permita a los estudiantes crear, imaginar, inventar, y que a la vez contribuyan a estimular las actividades grupales, el trabajo colaborativo, de manera que se pueda obtener conclusiones sobre su eficiencia y efectividad en el aprendizaje de la programación.

#### REFERENCIAS

- Insuasti, J. (2016) Problemas de enseñanza y aprendizaje de los fundamentos de programación. Revista educación y desarrollo social, 10 (2), 234-246. DOI: [org/10/18359/reds.1701](https://doi.org/10.18359/reds.1701)
- Fuentes, J., Iván and Moo, M. Melquizedec (2017). Dificultades de aprender a programar. Revista Educacion en Ingenieria, Febrero 2017, Volumen 12, 73-81.
- Bosse Y., Gerosa M. Aurelio (2016). Why is programming so difficult to learn? Patterns f Difficulties Related to Programming Learning. ACM SIGSOFT. Software Notas de Ingenieria. Volumen 41, No. 6, 1-6. from [https://www.researchgate.net/publication/312142448\\_Why\\_is\\_programming\\_so\\_difficult\\_to\\_learn\\_Patterns\\_of\\_Difficulties\\_Related\\_to\\_Programming\\_Learning\\_Mid-Stage](https://www.researchgate.net/publication/312142448_Why_is_programming_so_difficult_to_learn_Patterns_of_Difficulties_Related_to_Programming_Learning_Mid-Stage)
- Qian, Y and Lehman J. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. Transacciones de ACM sobre educación en computación Octubre de 2017 Artículo No 1: 1- 24 Recuperado: <https://doi.org/10.1145/3077618>
- Gomes A. and Mendes A. (2014). "A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations,"

2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, 2014, pp. 1-8, doi: 10.1109/FIE.2014.7044086.

Grover S. and Basu S (2017). SIGCSE '17: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education March 2017 Pages 267–272. <https://doi.org/10.1145/3017680.3017723>

ACM, IEEE & AIS (2020). Computing Curricula 2020: Introduction and community engagement. SIGCSE '19: Actas del 50º Simposio Técnico de ACM sobre Educación en Informática febrero de 2019 Páginas 653–654. <https://doi.org/10.1145/3287324.3287517>

Rubiano, S. M. M., López-Cruz, O., & Soto, E. G. (2015). Teaching computer programming: Practices, difficulties and opportunities. In *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

Rodríguez Carrillo, G. (2014). Enseñanza de la programación de computadoras para principiantes: un contexto histórico. *Revista Inventum*, 9(17), 51-61.

Alturki Raad. A. (2016), Measuring and Improving Student Performance in an Introductory Programming Course. *Informatics in Education-An International*, 2, 183-204 <https://files.eric.ed.gov/fulltext/EJ1117149.pdf>.

R. P. Medeiros, G. L. Ramalho and T. P. Falcão, (2019). "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education," in *IEEE Transactions on Education*, vol. 62, no. 2, pp. 77-90, May 2019, doi: 10.1109/TE.2018.2864133..

Carbonaro, A. & Ravaoli, M. (2017). Peer assessment to promote Deep Learning and to reduce a Gender Gap in the Traditional Introductory Programming Course. *Journal of e-Learning and Knowledge Society*, 13(3), Italian e-Learning Association. Retrieved July 13, 2020 from <https://www.learntechlib.org/p/180982/>.

A. Gomes, W. Ke, S. K. Lm, A. Siu, A. J. Mendes and M. J. Marcelino, (2017). "A teacher's view about introductory programming teaching and learning — Portuguese and Macanese perspectives," 2017 IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, 2017, pp. 1-8, doi: 10.1109/FIE.2017.8190493.

Santimateo, D., Núñez, G., Ediviel G. (2018). Estudio de dificultades de la enseñanza y aprendizaje de los cursos básicos de programación de computadores en

Panamá. *Revista de Investigación en Tecnologías de la Información*, 13-18.

Alarcón-Aldana, Andrea C, Díaz, Edilma L, & Callejas-Cuervo, Mauro. (2014). Guía para la evaluación de la Usabilidad en los Entornos Virtuales de Aprendizaje (EVA). *Información tecnológica*, 25(3), 135-144. <https://dx.doi.org/10.4067/S0718-07642014000300016>

Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P. (2014). *Metodología de la investigación* (6a ed). México, D.F: McGraw-Hill.

Rahim H., Badioze Z. Halimah, Ahmad A., Mohamad N. Ali. (2018) Student's Difficultties in Learning Programming. *Advanced Journal of Technical and Vocational Education*, 2 (3): 40-43.

Ramirez-Lopez, A., & Muñoz, A. (2015). Increasing practical lessons and inclusion of applied examples to motivate university students during programming courses. *Procedia - Social and Behavioral Sciences*, 176, 552-564. doi: <https://doi.org/10.1016/j.sbspro.2015.01.510>

Xinogalos, S. (2016). Diseño y despliegue de cursos de programación: estrategias, herramientas, dificultades y pedagogía. *Educ Inf Technol* 21, 559-588 (2016). <https://doi.org/10.1007/s10639-014-9341-9>

Palma-Suárez, C. A., & Sarmiento-Porras, R. E. (2015). Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria. *Revista mexicana de investigación educativa*, 20(65). Obtenido de [http://www.scielo.org.mx/scielo.php?pid=S1405-66662015000200013&script=sci\\_arttext](http://www.scielo.org.mx/scielo.php?pid=S1405-66662015000200013&script=sci_arttext)

Llorens, F., García, F. J., Molero, X., & Vendrel, E. (2017). La enseñanza de la informática, la programación y el pensamiento computacional en los estudios preuniversitarios. *Education in the knowledge society (EKS)*, 18(2). doi: <http://dx.doi.org/10.14201/eks2017182717>

Gallego-Durán, F. J., Villagrà-Arnedo, C. J., Llorens Largo, F., & Molina-Carmona, R. (2017). PLMan: A Game-Based Learning Activity for Teaching Logic Thinking and Programming.

Díaz, J., Banchoff, C., Queiruga, C., & Martin, E. S. (2015). Experiencias de la Facultad de Informática en

la Enseñanza de Programación en Escuelas con Software Libre. La Plata, Argentina. Obtenido de [https://www.researchgate.net/profile/Claudia\\_Queiruga/publication/271506865\\_experiencias\\_de\\_la\\_facultad\\_de\\_informatica\\_en\\_la\\_ensenanza\\_de\\_programacion\\_en\\_escuelas-vf/links/54c948e60cf298fd262461aa.pdf](https://www.researchgate.net/profile/Claudia_Queiruga/publication/271506865_experiencias_de_la_facultad_de_informatica_en_la_ensenanza_de_programacion_en_escuelas-vf/links/54c948e60cf298fd262461aa.pdf)

Dapozo, G. N., Greiner, C., & Petris, R. H. (2016). Herramientas lúdicas como apoyo a la enseñanza de la programación. XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016), 1311-1319. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/56298>

Zatarain Ramón Cabada, María Lucía Barrón Estrada, Francisco González Hernández, Raúl Oramas Bustillos, Carlos Alberto Reyes-García, (2018). An affective and Web 3.0-based learning environment for a programming language, Telematics and Informatics, Volume 35, Issue 3, 2018, 611-628, <http://www.sciencedirect.com/science/article/pii/S0736585316304907>.

Colucci Philip G., Petro Kostandy, William R. Shrauner, Elizabeth Arleo, Michele Fuortes, Andrew S. Griffin, Yun-Han Huang, Krishna Juluru, Apostolos John Tsiouris, (2015) Development and Utilization of a Web-Based Application as a Robust Radiology Teaching Tool (RadStax) for Medical Student Anatomy Teaching, Academic Radiology, Volume 22, Issue 2, 2015, 247-255, <http://www.sciencedirect.com/science/article/pii/S107663321400378X>