

# Identificación de errores en conceptos básicos de principios de programación

*Identification of errors in basic concepts of programming principles*

Lizzie Edmea Narváez Díaz

Rocío Edith López Martínez

## RESUMEN

Crear algoritmos en la universidad es el primer paso para introducir al alumno a la programación de computadoras y es crucial considerarlo ya que diversas investigaciones han reportado que muchas veces el estudiante manifiesta un deficiente rendimiento académico en estos cursos por la complejidad de los contenidos, por no contar con algo tangible para contrastar el algoritmo y porque resolver problemas requiere esfuerzo, creatividad y otras habilidades. Esta investigación se centró en la materia de Algoritmia de la Unidad Multidisciplinaria Tizimín en la cual muchos alumnos no logran alcanzar los conocimientos requeridos, siendo el rendimiento académico al final del curso inadecuado. Se decidió encontrar exactamente en qué temas los estudiantes tienen problemas al crear sus algoritmos, situación no analizada, y que una vez determinados estos pudieran servir de apoyo a una metodología de aprendizaje mediante el lenguaje de programación Scratch, la cual se viene empleando para ayudar al alumno en su rendimiento académico. La investigación fue de tipo cualitativo, sujeta al análisis de contenido aplicado a las pruebas de desempeño de los cursos. Se analizaron 1037 reactivos obteniendo un listado de errores y se probó la efectividad de su uso en los cursos impartidos. Se arrojaron resultados satisfactorios considerando tres aspectos: promedio, cantidad de aprobados y nivel de dominio.

*Palabras clave:* algoritmia, análisis de contenido, diagramas de flujo, educación superior, pruebas de desempeño.

## ABSTRACT

Creating algorithms at the university is the first step to introduce the student to computer programming. A skill crucial to consider since various investigations have reported that many times the student has shown poor academic performance in these courses due to the complexity of the contents, because they do not have something tangible to test the algorithm and because solving problems requires effort, creativity, and other skills. This research focused on the subject of Algorithms of the Tizimín Multidisciplinary Unit in which many students fail to achieve the required knowledge, being the academic performance at the end of the course inadequate. It was decided to find out exactly in which subjects the students have problems when creating their algorithms, a situation not analyzed, and once these were determined, they could serve as support for a learning methodology through the Scratch programming language, which has been used to help the student in their academic performance. The research was qualitative, subject to content analysis applied to course performance tests. 1037 items were analyzed obtaining a list of errors and the effectiveness of their use in the courses taught was tested. Satisfactory results were obtained considering three aspects: average, number of passes, and mastery level.

*Keywords:* algorithms, content analysis, flowcharts, higher education, performance tests.

## INTRODUCCIÓN

Desde hace varias décadas ha existido interés en conocer los factores que se involucran en el aprendizaje de la programación y los retos que existen en el proceso de desarrollo de habilidades algorítmicas, así como también se han hecho múltiples esfuerzos por parte de diversos investigadores y grupos colegiados por estudiar los elementos para mejorar la enseñanza-aprendizaje de materias iniciales del área con el fin de minimizar las estadísticas de reprobación, por lo que se han propuesto varios enfoques de enseñanza, metodologías y han sido diseñadas diversas herramientas para tal efecto (Sánchez et al., 2018; Sánchez et al., 2019).

En el ámbito de la programación, antes de resolver un problema mediante el uso de una computadora se recomienda realizar primero el algoritmo, ya que es aquí donde se encuentra la solución universal de la problemática en cuestión (Vázquez, 2012). En este sentido el conocimiento de conceptos básicos para la creación de algoritmos es un proceso fundamental durante el aprendizaje y parece ser algo difícil para muchos estudiantes, siendo que la mayor parte de sus problemas son originados por la complejidad de los conceptos relacionados con los principios de programación (Xinogalos et al., 2015; López-Escribano y Sánchez-Montoya, 2012; Gomes y Medes, 2007).

Partiendo del hecho de que la creación de algoritmos es clave durante el aprendizaje, es relevante iniciar con su definición: un algoritmo es un método para resolver un problema mediante una serie de pasos, procedimientos o acciones, el cual debe ser preciso, definido y finito; en otras palabras, es un conjunto de reglas para abordar una cierta clase de situación o una forma de describir una solución (Cairó, 2005; Joyanes, 2008). Pensar algorítmicamente requiere un fuerte aspecto creativo, sin embargo, esto no es suficiente ya que existe un conjunto de habilidades que están conectadas con la creación y comprensión de los algoritmos, las cuales se deben tener en cuenta; estas son las siguientes (Futschek, 2006):

**Lizzie Edmea Narváez Díaz.** Profesora Titular C de la Unidad Multidisciplinaria Tizimín de la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, México. Es doctora en Innovación en Tecnología Educativa por la Universidad Autónoma de Querétaro y cuenta con una maestría en Ciencias de la Computación por el Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Cuernavaca. Tiene reconocimiento al perfil PRODEP. Entre sus publicaciones recientes se encuentran los artículos “Creación de videojuegos como estrategia educativa en algoritmia” (2021) e “Incorporación de la herramienta Scratch para el aprendizaje de conceptos de algoritmia” (2020). Correo electrónico: lendiaz@correo.uady.mx. ID: <https://orcid.org/0000-0003-0595-1932>.

**Rocío Edith López Martínez.** Profesora-investigadora de la Facultad de Informática de la Universidad Autónoma de Querétaro, México. Cuenta con tres doctorados en Alta Dirección, Negocios y en Educación. Cuenta con reconocimiento al perfil PRODEP y pertenece al Sistema Nacional de Investigadores. Ha desarrollado diversos proyectos de investigación, con y sin financiamiento, además de publicaciones en revistas nacionales e internacionales. Ha participado en diversos capítulos de libros; es coordinadora de dos libros de texto *TIC: avances en la investigación e innovación educativa* I y II. Posee tres certificados de propiedad intelectual. Correo electrónico: rocio.edith.lopez@uaq.mx. ID: <https://orcid.org/0000-0002-5209-3523>.

- Analizar problemas.
- Especificar un problema en forma precisa.
- Encontrar los pasos básicos adecuados para resolver un problema.
- Construir el algoritmo correcto para un problema dado utilizando los pasos básicos.
- Pensar en todos los posibles casos particulares y normales de un problema.
- Mejorar la eficiencia del algoritmo.

En común acuerdo con las habilidades citadas, se puede mencionar que en mayor o menor medida todas las personas poseen cierto grado de pensamiento algorítmico puesto que constantemente todos están aplicando algoritmos en la vida cotidiana para resolver toda clase de problemas (no únicamente computacionales), aunque esto ocurre muchas veces sin darnos cuenta, de manera inadvertida, inconsciente o automáticamente. Este tipo de escenarios se presenta cuando la situación que se tiene en mente se ha resuelto con anterioridad un gran número de veces y en consecuencia no se piensa en el “paso a paso” para encontrar una solución, como por ejemplo: hacer un emparedado, preparar un café, manejar un auto, andar en bicicleta, trasladarse de un lugar a otro, etc. Con base en este planteamiento se puede afirmar que en la vida diaria las personas están continuamente resolviendo problemas, lo que en otras palabras quiere decir que están aplicando algoritmos.

Complementando lo anterior, la resolución de problemas está fundamentada en activar la curiosidad, el descubrimiento y la exploración; esta resolución se define como la adquisición de modelos de pensamiento a partir de construcciones declarativas y procedimentales (Sánchez et al., 2019).

En el contexto de la programación se vuelve importante mencionar que cuando se generan los algoritmos se busca que el resultado se pueda aplicar mediante el uso de una computadora, ya que por medio de ella se adaptarán cada uno de sus elementos al lenguaje de programación en el que se desee implementar, por lo tanto, el algoritmo que se cree debe tener como característica final la posibilidad de ser transcrito fácilmente a un lenguaje, para lo cual se usarían herramientas de programación (Pinales y Velázquez, 2014).

De acuerdo con la literatura revisada, la enseñanza y el aprendizaje de los principios de programación o bien de los conceptos algorítmicos fundamentales constituyen un reto tanto para los maestros en el proceso que deben seguir para enseñar, como para los estudiantes en su aprendizaje. La creación de algoritmos en el nivel universitario es un primer paso para iniciar al alumno en la programación de computadoras, sin embargo, diversas investigaciones han reportado que muchas veces el estudiante manifiesta un deficiente rendimiento académico a la hora de llevar estos cursos introductorios; sobre este particular se ha encontrado lo siguiente:

- Desde el año 2005 la Facultad de Informática de la Universidad de Tecnología de Viena empezó a ofertar cursos preuniversitarios o propedéuticos de manera

similar a otras universidades para todos aquellos solicitantes que intentaran iniciar una licenciatura en estudios informáticos, esto fue porque encontraron falta de conocimientos previos sobre algoritmos y programación, así como insuficiencias en temas matemáticos (Futschek, 2006).

- En todas las carreras de informática existe alguna materia de programación en el primer año, en el caso particular de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco en la ciudad de Comodoro Rivadavia en Argentina esta materia es Algorítmica y Programación, y tal como acontece en la mayoría de las primeras materias de una carrera universitaria, confluyen diferentes situaciones que dificultan su aprendizaje, razón por la cual en esta materia los docentes se enfocaron en buscar los principales problemas que aquejan a los estudiantes cuando se encuentran en la fase de programación (Rosanigo y Paur, 2006).
- Existe consenso entre investigadores y educadores del área de programación, y en ella de la algoritmia, en lo difícil que resulta a los estudiantes de los primeros años de carreras de ingeniería superar los mismos exitosamente con base en los aprendizajes complejos requeridos. En la Universidad del Centro Educativo Latinoamericano en Argentina identificaron las preferencias cognitivas predominantes de los alumnos de Ingeniería en Sistemas de Información, con el fin de proponer innovaciones metodológicas para afrontar con éxito las dificultades de la enseñanza de la programación teniendo en cuenta las diferencias individuales de los educandos (Dania y Marchisio, 2013).
- Los cursos en los primeros años de las carreras informáticas relacionados con fundamentos de programación tienen altas tasas de reprobación y proponen iniciativas para mejorar asociadas con las herramientas tecnológicas Scratch y Lego (Muñoz et al., 2012).
- Por último, en el trabajo desarrollado en la carrera de Ingeniería Informática de la Universidad de Buenos Aires se menciona que los alumnos que llegan a un primer curso de algoritmia e integran el mismo, normalmente son un grupo heterogéneo de personas en cuanto a conocimientos previos se refiere y esta situación se refleja en los resultados finales (Lage y Cataldi, 2019).

El desarrollo de esta investigación se centra en la materia de Algoritmia, la cual se imparte en la Licenciatura en Ingeniería de Software (LIS) en la Unidad Multidisciplinaria Tizimín (UMT) de la Universidad Autónoma de Yucatán (UADY), y es una asignatura que presenta la situación que aqueja en las investigaciones analizadas, es decir, muchos alumnos no logran alcanzar los conocimientos que se requieren en principios de programación y en consecuencia el nivel de dominio que tienen al final del curso no es el adecuado. Con base en el problema anterior se decidió trabajar en encontrar exactamente en qué temas los estudiantes tienen más dificultades cuando

crean sus algoritmos (fase previa a la programación), la cual es información que no se encuentra analizada, y que una vez determinados estos contenidos pudieran servir de apoyo a una metodología de aprendizaje con el empleo del lenguaje de programación Scratch que se viene aplicando en la UMT para ayudar a elevar el rendimiento académicos de los alumnos; es fundamental señalar que la metodología citada ha sido creada expresamente para tal efecto.

Para la UADY, para fines de promoción de un estudiante se define el nivel de dominio como los atributos o características que describen el grado en que el estudiante ha desarrollado determinada(s) competencia(s) de una asignatura, se representa de manera cuantitativa y cualitativa y debe registrarse en un documento validado por la institución. La representación cuantitativa del nivel de dominio se establece con un puntaje del 0 al 100 y este se categoriza en *Sobresaliente*, *Satisfactorio*, *Suficiente* y *No acreditado* (Tabla 1) (UADY, 2012).

**Tabla 1**

*Niveles de dominio para la acreditación de una asignatura en la UADY*

Niveles de dominio	
Puntaje	Categoría
90 – 100	Sobresaliente (SS)
80 – 89	Satisfactorio (SA)
70 – 79	Suficiente (S)
0 – 69	No acreditado (NA)

*Fuente: UADY (2012, p. 71).*

Desde el año 2016 se imparte en la LIS la asignatura de Algoritmia y desde ese primer momento se detectó una situación problemática a partir del análisis estadístico realizado a las actas de examen finales de los cursos dados durante los años 2016 y 2017; de este estudio se han obtenido los siguientes resultados: en el 2016 (se nombra grupo A) el 58.33% de los alumnos tuvo un nivel de dominio entre satisfactorio y sobresaliente y el 41.66% entre suficiente y no aprobado; en el 2017 (se nombra grupo B) el 46.15% tuvo un nivel de dominio entre satisfactorio y sobresaliente y el 53.84% entre suficiente y no aprobado. El grupo A, al cursar Programación Estructurada que se imparte posterior a Algoritmia, con la cual está relacionada, tuvo un nivel de desempeño entre satisfactorio y sobresaliente del 45.45% y el 54.54% entre suficiente y no aprobado; en cuanto al grupo B, tuvo un nivel de desempeño entre satisfactorio y sobresaliente del 50% y el 50% entre suficiente y no aprobado; con estos datos se puede observar que el rendimiento general de los grupos no es el óptimo en ninguna de las dos materias, y aunque pudiera ser que el índice de reprobación no pareciera ser tan alto, sí se expresa un desconocimiento fundamental de los alumnos con los

temas, lo que podría acarrear problemas más adelante con materias relacionadas. Según los resultados obtenidos y lo que se ha observado, de un modo general los alumnos no logran entender el correcto resultado de los problemas planteados, es decir, los algoritmos que se generan, para lo cual se han venido utilizando dos tipos de herramientas: los diagramas de flujo y el pseudocódigo.

En este sentido se vuelve relevante comprender de modo más específico cuáles son los principales problemas a los que se enfrentan los educandos cuando crean los algoritmos con el fin de que el docente pueda apoyarlos desde el inicio de la asignatura. Atendiendo a esta situación en los cursos impartidos de la materia Algoritmia en los semestres agosto-diciembre 2018, agosto-diciembre 2019 y septiembre 2020-febrero 2021, se ha llevado a cabo una investigación para encontrar cuáles son los problemas principales que afronta el alumno en la construcción de los algoritmos en general; cabe hacer mención de que en los semestres agosto-diciembre 2016 y agosto-diciembre 2017, que fueron los dos primeros en que se impartió la citada materia, estos fueron llevados a cabo del modo tradicional (a papel y lápiz), es decir, sin el conocimiento a fondo de los principales temas problema que afrontaron los alumnos y sin el uso de ninguna herramienta tecnológica.

Es relevante indicar que en la literatura revisada se encontraron algunos de los problemas que tienen los alumnos cuando están trabajando en la fase de programación y no en la fase previa que es la creación del algoritmo (etapa de interés), a pesar de esto, algunos de estos errores son completamente aplicables en el desarrollo de este último (Rosanigo y Paur, 2006).

Teniendo en mente el objetivo planteado para la materia de Algoritmia, se consideró de importancia llevar a cabo un análisis de contenido a las pruebas de desempeño presentadas por los alumnos durante el curso, así como también a las pruebas obtenidas de un propedéutico, con el propósito de encontrar los significados, tanto los manifiestos como los latentes, y para eso clasificar y codificar los diferentes elementos en categorías que representen más claramente el sentido de lo encontrado. El análisis de contenido en este contexto se considera de utilidad ya que cumple con la función de tratar de descubrir los significados de un documento, el cual puede ser textual, como la transcripción de una entrevista, una historia de vida, un libro o material audiovisual, entre otros, en este caso serán las pruebas de desempeño.

Analizar un contenido supone estudiar a fondo los temas de un material previamente seleccionado y es ampliamente aceptado considerar que cualquier estudio con espíritu crítico de un mensaje constituye ya un análisis de contenido en sí mismo (Tinto, 2013).

Tinto (2013), luego de una revisión bibliográfica sobre el análisis de contenido, pone de manifiesto la diversidad de puntos de vista de los diferentes autores a este respecto, pero afirma que todos coinciden en el objeto material de estudio, el cual son

las comunicaciones en su conjunto, bien sean estas escritas, orales o audiovisuales, siempre y cuando sean susceptibles de identificar y grabar.

La literatura menciona que el análisis de contenido es una técnica para estudiar cualquier tipo de comunicación de manera objetiva y sistemática sobre contenidos en muy diversos contextos y en múltiples formas (Hernández et al., 2014; Fernández, 2002). Por otro lado, también es considerado como un método de observación y medición, en lugar de estudiar el comportamiento de las personas en forma directa, o de pedirles que respondan a escalas, o aun de entrevistarlas, el investigador toma las comunicaciones que la gente ha producido y pregunta acerca de dichas comunicaciones (Kerlinger, 2002).

En este proceso de estudio, con la ayuda del análisis de contenido aplicado a las pruebas de desempeño seleccionadas se obtuvo la relación de errores más significativos que cometen los alumnos en la creación de sus algoritmos.

## METODOLOGÍA

En función de la situación reflejada en la materia de Algoritmia en la LIS como ya fue mencionada, se decidió crear una metodología de aprendizaje en el año 2018, la cual incorporó el uso del lenguaje de programación Scratch con el objetivo de ayudar a los alumnos a mejorar su rendimiento académico. Al poner en marcha esa metodología se convirtió de relevante importancia conocer exactamente en qué temas se encontraban la mayor cantidad de errores por parte del alumnado cuando creaba sus algoritmos, es por esto que se procedió a realizar un análisis de contenido para identificar los errores más frecuentes y posteriormente usar la información generada para complementar la metodología de aprendizaje que se tenía en marcha, de aquí que el objetivo de este artículo se centró en encontrar esos temas problema.

Inicialmente solo se contaba con el conocimiento que el docente de la materia tenía respecto de esta relación de errores y con el fin de tener evidencia justificada se procedió a hacer el análisis de contenido a las pruebas de desempeño que se aplicaron durante los cursos y los propedéuticos impartidos, siendo estos los siguientes:

- Curso de Algoritmia del semestre agosto-diciembre 2018.
- Curso propedéutico desarrollado en junio-julio 2019.
- Curso de Algoritmia del semestre agosto-diciembre 2019.
- Curso propedéutico desarrollado en septiembre del 2020.

En las dos ocasiones en que se llevó a cabo el propedéutico, al finalizar se aplicó una prueba de desempeño a la cual posteriormente se le realizó el análisis de contenido.

Adicionalmente, a lo largo de los cursos de Algoritmia impartidos en los dos semestres previamente mencionados se fueron aplicando pruebas de desempeño (cuatro en total en cada curso) y a estas también se les realizó el análisis de contenido.

El conjunto de todas las pruebas de desempeño obtenidas fue de 193 y se decidió analizar la totalidad de ellas, de las cuales 34 fueron derivadas de los dos cursos propedéuticos y 159 generadas durante la impartición de los cursos de Algoritmia; esta información traducida al número de reactivos analizados en las pruebas se puede apreciar en las tablas 2 y 3.

**Tabla 2**

*Pruebas de desempeño y reactivos analizados durante el curso propedéutico*

Grupo del propedéutico	Total de pruebas de desempeño	No. de reactivos analizados en la prueba	Total de reactivos analizados
1	14	6	84
2	20	6	120
			<b>Total</b> 204

*Fuente:* Construcción personal.

**Tabla 3**

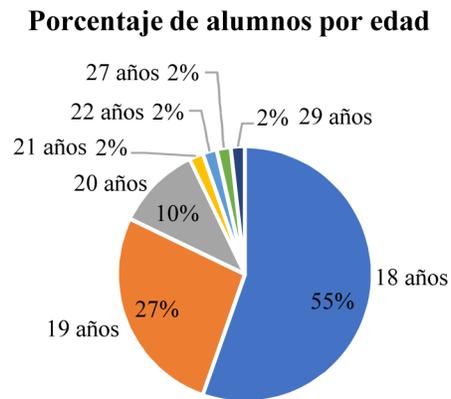
*Pruebas de desempeño y reactivos analizados durante el curso de Algoritmia*

Número de prueba de desempeño	Total de pruebas de desempeño	No. de reactivos analizados en la prueba	Total de reactivos analizados
1	42	4	168
2	41	6	246
3	39	6	234
4	37	5	185
			<b>Total</b> 833

*Fuente:* Construcción personal.

Las pruebas de desempeño fueron obtenidas únicamente del curso de Algoritmia y del propedéutico. En cuanto a los respondientes, estos estuvieron formados por 76.79% de hombres y 23.21% de mujeres. Por otro lado, aun cuando los alumnos que optan por la UMT proceden de un contexto variado de bachilleratos con planes educativos diferentes pertenecientes a diversos sistemas educativos (siete para ser precisos), la mayor parte de ellos llegan de las dos principales escuelas de enseñanza media de Tizimín, las cuales son el CONALEP (Colegio Nacional de Educación Profesional Técnica) y el COBAY (Colegio de Bachilleres del Estado de Yucatán), lo que representa el 53.57% del total de alumnos en la LIS. En cuanto al rango de edades, este fluctúa entre 18 y 29, siendo el promedio de 18.96 años (Figura 1) y, por último, es crucial mencionar que en el tiempo de la investigación el profesor que impartió las clases siempre fue el mismo y los temas incluidos en la planeación didáctica de la materia no han presentado cambios.

**Figura 1**  
*Gráfica de las edades de los alumnos de la LIS*



*Fuente:* Construcción personal.

La investigación fue de tipo cualitativo y se sujetó al proceso de análisis de contenido, el cual se compone de cuatro etapas: el análisis previo, la preparación del material, la selección de unidades de análisis y la explotación de los resultados; estas fases se desglosan a continuación (Gómez, 2000):

- *Análisis previo.* En esta etapa se definió el problema que se presenta en Algoritmia en su contexto general y la situación actual de la materia determinando los objetivos; se describieron las condiciones iniciales tanto en el contexto educativo como con los alumnos participantes y se definió una lista inicial de errores, que se obtuvo mediante entrevistas llevadas a cabo con el maestro de la materia, el cual la ha obtenido en función de su propio análisis, observaciones y su experiencia en el área.
- *Preparación del material.* El corpus de documentos seleccionados fueron las pruebas de desempeño que se obtuvieron luego de dos sucesos diferentes, las primeras al término del curso propedéutico y las segundas se fueron obteniendo poco a poco a lo largo del curso de Algoritmia (ambos momentos en el tiempo definido con anterioridad). Teniendo en mente la identificación de los tópicos problema se procedió a tener cuidado en el diseño de las pruebas de desempeño que se aplicaron tanto durante el propedéutico como durante el curso escolar, en las cuales se incluyeron problemas que los alumnos debían de resolver y a la par el maestro tenía una guía sobre qué tipo de temas se referían en cada uno de los problemas; el total de los documentos sobre los que se trabajó se fue recolectando a lo largo de tres años y se analizaron en la medida en que se obtuvieron. Se llevó a cabo la transcripción de las pruebas, se revisó cada una y se fue anotando en ellas lo que se consideró relevante,

haciendo señalamientos precisos en los diagramas de flujo que se estudiaron, así como consideraciones del revisor. Para finalizar, en esta etapa se optó por el procedimiento de tratamiento tradicional que consistió en desglosar el contenido y hacer agrupaciones con el mismo para identificar las categorías de importancia.

- *Selección de unidades de análisis.* La unidad de análisis es el fragmento de documento o comunicación que se toma como elemento que sirve de base para la investigación, en este sentido se consideraron las categorías establecidas anteriormente como puntos centrales sobre las cuales se analizó el corpus de documentos que se tenía (Balcells, 1994).
- *Explotación de los resultados.* Se construyeron tablas en las cuales se posicionaron marcas que se relacionaban con los temas en los que fallaron cada uno de los estudiantes en función de las categorías seleccionadas, al final estas marcas se tradujeron en la cantidad de veces que se comete una equivocación, fue condensada esta información y se obtuvo la relación de errores que se esperaba.

Apegados a las etapas anteriores el proceso se fue llevando a cabo poco a poco, empezando con el curso de agosto-diciembre 2018 y finalizando con el propedéutico en septiembre del 2020, para lo cual la primera actividad que se realizó para obtener los documentos para el análisis fue el curso de Algoritmia en el semestre escolar de agosto-diciembre 2018, de donde se recabaron cuatro pruebas de desempeño y, como se citó, a estas pruebas se les aplicó el análisis de contenido; acorde con los resultados y con todo lo observado durante la impartición del mismo por parte del docente del curso se detectó cuáles fueron los errores más frecuentes en el proceso de análisis y construcción de los diagramas de flujo. Estos resultados sirvieron para adecuar el curso propedéutico el cual se llevó a cabo por primera vez en los meses de junio-julio 2019, mismo que tuvo como actividad final la aplicación de una prueba de desempeño a la cual se le hizo el análisis de contenido encontrando algunas semejanzas y diferencias con el análisis hecho a las pruebas del semestre agosto-diciembre 2018, de cualquier modo la información generada sirvió de base para ajustar el inicio del siguiente semestre en que se impartió Algoritmia, que fue durante agosto-diciembre 2019, en este periodo de igual manera se procedió como en el semestre previo, y por último en septiembre del 2020 se llevó a cabo otro curso propedéutico siguiendo lo anteriormente descrito. Es importante mencionar que al inicio del proceso en el primer curso que se investigó se empleó una lista de errores generada por el maestro de la materia y esta misma se fue refinando con el paso del tiempo. Al término de todo este proceso se obtuvo un listado final de los principales problemas que afrontaron los estudiantes durante el curso de Algoritmia.

## RESULTADOS

Los resultados conseguidos de la aplicación del análisis de contenido se fueron produciendo poco a poco, de aquí que la relación de los principales errores tuvo varios resultados parciales, los cuales se fueron acumulando hasta llegar a la relación final con la que ahora se cuenta.

De todo el proceso descrito en la metodología anterior se obtuvieron dos tablas de codificación de resultados con base en las categorías seleccionadas; la Tabla 4 que se presenta corresponde a las pruebas de desempeño realizadas durante los dos cursos propedéuticos.

**Tabla 4**

*Codificación del análisis de contenido de las pruebas de desempeño de los propedéuticos*

Casos de error/ Descripción del error	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Condición incorrecta o mala comprensión lectora																			
Término erróneo del DF																			
Asignación de variable																			
Mal elaboración de mensajes																			
Operaciones con porcentajes																			
Leer variables que se calculan o no leerías																			
Nombres de variables incorrectos																			
Falta de resultados en datos de salida																			
Lógica general del problema																			
Con operadores aritméticos																			
Con operadores relacionales																			

*Fuente:* Construcción personal.

La Tabla 5 corresponde al análisis de contenido realizado a las pruebas de desempeño de los dos cursos de Algoritmia citados previamente y presenta una relación de las veces que ocurrió cada uno de los errores detectados; cabe hacer mención de que no se contabilizaron como error los ejercicios no resueltos completamente, y se determinó que la cantidad de estos corresponde a 49 del total de 833.

Tabla 5

Errores en las pruebas de desempeño del curso de Algoritmia de los semestres agosto- diciembre 2018 y 2019

Casos de error/ Descripción del error	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33					
Lógica general del problema																																						
Con operadores aritméticos																																						
Condición de ciclo incorrecta																																						
Condición incorrecta o comprensión lectora																																						
Con operadores relacionales																																						
Lógica incorrecta en proceso																																						
Inicializaciones incorrectas																																						
Error calculo promedio, falta de análisis																																						
Falta de resultados en datos de salida																																						
Asignación incorrecta																																						
Estructura de decisión																																						
Leer vars que calculan/ no leerlas /leer de más																																						
Operaciones con porcentajes																																						
Mal uso de multiplicador																																						
Mal elaboración de mensajes																																						
Falta lectura de valores en ciclo																																						
Con operadores lógicos																																						
Incorrecto uso de variables en arreglos																																						
Arreglos imprimir posición-no valor y vv																																						
Falta operación que se usa después																																						
Fin del DF																																						
Cierre de ciclo incorrecto																																						
No modificó variable de control de ciclo																																						
Arreglo fuera de rango																																						
Mal uso de sumador																																						
Hacer asignaciones que nunca se usan																																						
Término erroneo del DF																																						

Fuente: Construcción personal.

Una vez que se contó con la lista de errores frecuentes, para comprobar la diferencia en el rendimiento académico de cuando se tenía desconocimiento de estos errores con lo que ahora se tiene se presentan tres aspectos o estadísticos fundamentales para poder medir el efecto de su uso en la asignatura, el primero se refiere al promedio de aprovechamiento de los grupos, el segundo al índice de aprobados y

el tercero atendiendo al porcentaje de estudiantes con nivel de dominio satisfactorio o sobresaliente; en los tres casos se hace una distinción en función de la estrategia usada, es decir, empleando el método tradicional (con los cursos impartidos en el 2016 y el 2017) en el cual el docente no contó con una relación de los principales temas problema que aquejan a los alumnos dado que no se conocían, y posteriormente con la implementación de los principales errores (después del análisis de contenido) mediante el apoyo de la herramienta Scratch (con los cursos impartidos en los años 2018, 2019 y 2020-2021).

Los resultados obtenidos en cuanto al promedio de aprovechamiento final de los cinco grupos son los siguientes: para los cursos impartidos en el 2016 y el 2017 con el método tradicional se obtuvo un promedio de 75.75 puntos y para los cursos impartidos en 2018, 2019 y 2020-2021 con la implementación de los errores mediante el uso de Scratch se obtuvo un promedio de 85.12 puntos.

En cuanto al índice de aprobados y reprobados, en la Tabla 6 se encuentran los resultados obtenidos en estos cinco cursos, identificando los porcentajes de los alumnos con el método tradicional y con la implementación de los errores con apoyo de la herramienta de Scratch.

**Tabla 6**

*Alumnos aprobados-reprobados en Algoritmia 2016 a 2021*

Estrategia usada	Total de alumnos	Aprobados	Reprobados
Tradicional	24	70.83%	29.17%
Errores implementados con Scratch	66	92.42%	7.58%

*Fuente:* Construcción personal.

Por último, en cuanto al nivel de dominio, usando el método tradicional 54.17% tuvo un nivel entre satisfactorio y sobresaliente, mientras que con la implementación de los errores con el uso de Scratch este valor fue de 78.79%.

## DISCUSIÓN

En lo que respecta al análisis de contenido aplicado a las pruebas de desempeño presentadas durante el curso propedéutico, se logró determinar que los errores que presentan un mayor número de ocurrencias y por tanto se consideraron más relevantes fueron:

- Deficiente comprensión lectora y/o incorrecta elaboración de condiciones.
- Problemas con el manejo de operadores aritméticos.
- Problemas con el manejo de operadores relacionales.
- Incorrecta comprensión de la lógica general del problema.
- Mala elaboración de mensajes en los datos de salida.

- Falta de datos de salida (este error junto con el anterior refleja que se hizo un incorrecto análisis).
- Finalización incorrecta del diagrama de flujo.
- Errores al manejar porcentajes.
- Nombrar de modo incorrecto a las variables y/o constantes.
- Leer variables que deben ser calculadas o bien no leerlas.
- Problemas de asignación de resultados a variables.

Inicialmente esta información sirvió para preparar la intervención durante el siguiente curso de Algoritmia ya que se usó como evidencia para saber hacia qué aspectos enfocarse con más detalle. Posteriormente, estos datos se complementaron con los resultados del análisis de contenido aplicado a las pruebas de desempeño obtenidas luego de llevar a cabo los cursos de Algoritmia; los resultados en algunos casos sirvieron para confirmar lo que se tenía inicialmente, pero también se detectaron nuevos fallos que fue necesario considerar. Se destaca la siguiente lista de errores que son los más recurrentes en estas, de igual modo son los que el docente ha detectado en el aula de clase acorde con sus observaciones y su experiencia; los errores que se presentan son aquellos que tienen más del 50% de incidencia en función del caso de error que obtuvo más sucesos, se listan en orden de mayor a menor ocurrencia:

- Mal entendimiento de la lógica general del problema; un aspecto de suma importancia que se logró determinar fue que los alumnos muchas veces no entienden lo que leen por no leer a profundidad el problema.
- Dificultad en el manejo de operadores aritméticos (+, -, \*, /, \*\*, div, mod, ()).
- Elaboración de las condiciones de ciclo de modo incorrecto; les resulta complejo decidir entre usar un ciclo *while*, *do..while* o *for*, por ejemplo.
- Condición mal elaborada en estructuras de decisión, algunas veces asociadas a la falta de análisis del problema a profundidad.
- Dificultad y confusión en el manejo de operadores relacionales (<, <=, >, >=, =, <>).
- Lógica incorrecta en al menos un proceso, por ejemplo, al hacer un cálculo o al asignar un resultado a una variable equivocada.
- Inicializaciones incorrectas que posteriormente afectan al resultado final del problema, frecuentemente se presentan al inicializar sumadores, multiplicadores, contadores y otros.
- Cálculo de promedio; aun cuando esta es una operación que los alumnos saben hacer, en el contexto de los ejercicios este problema fue recurrente, es común ver que muchas veces por no entender a profundidad lo que leen no saben en qué lugar es más adecuado obtener el promedio o bien no saben entre qué valor dividir el total obtenido.

- Falta de resultados en datos de salida consecuencia de un análisis pobre del problema planteado.
- Otros con menor número de ocurrencias.

Ahora se cuenta con una relación de errores bien definidos sobre los cuales se puede partir en este tipo de cursos, lo cual se considera importante, porque el docente contará con un punto de partida sobre el cual poder enfocarse en las actividades de la materia desde el inicio.

Por otro lado, como se mencionó anteriormente en la revisión hecha a la literatura, es escasa la información encontrada al respecto, pero se destaca un listado de errores enfocados básicamente cuando el alumno está en la etapa de programación y no de diseño de un algoritmo, sin embargo, a pesar de esto se encuentran similitudes en algunos aspectos, los cuales se citan a continuación (Rosanigo y Paur, 2006):

- Confunden el uso de los operadores lógicos, por ejemplo utilizan conjunción en lugar de disyunción o viceversa, o no reconocen bien cuál corresponde al problema.
- Errores en las estructuras de control, ciclos sin un fin, ciclos que no avanzan o se vuelven infinitos; no se inicializan variables utilizadas como contadores, acumuladores o multiplicadores; mala selección de la estructura de control más adecuada al problema y mal planteamiento de la condición en la estructura.
- Errores en la solución, es decir, no muestra los resultados obtenidos, no resuelve el problema planteado, mala interpretación del enunciado (en particular este es un grave problema con los alumnos de la LIS, se ha observado una pobre comprensión lectora), mal planteo del problema, falta de casos para comprobar (comprobando solo el que funciona), establece la solución general pero no la refina, resuelve puntos parciales pero no arma la solución global.

Con base en lo que ahora se tiene, se considera que el alumno, que es el principal implicado, será beneficiado ya que el docente al conocer dónde el primero tiene más problemas, entonces podrá plantear ejercicios enfocados en estos que le permitan entender mejor los conceptos estudiados. Esta información que se obtuvo fue inmersa en la metodología de aprendizaje planteada y de esta manera, con la práctica de ejercicios y el desarrollo de programas utilizando la herramienta de programación Scratch, el alumno tuvo un referente para mejorar la comprensión en este tipo de problemas.

Asimismo, en función de los estadísticos aplicados se observó que el alumno ha elevado su rendimiento académico, ya que se puede destacar que cuando se implementó la relación de errores encontrados al curso de Algoritmia se encontró un incremento de 9.37 puntos respecto de aquellos cursos en los que no se tenía este listado de errores, esto haciendo énfasis en el promedio de aprovechamiento final.

En cuanto al índice de aprobados y reprobados también se observó que se obtuvo un mayor índice de aprobados en los cursos llevados a cabo usando esta relación de errores, en comparación con los que no, teniendo un incremento del 21.59%.

Por último, evaluando el nivel de dominio el resultado estadístico también sumó, ya que se encontró un incremento de alumnos en los niveles de satisfactorio y sobresaliente, en este caso del 24.62%, de aquellos que atendieron a cursos donde se implementó esta relación de errores en comparación con los que no.

Conforme al análisis anterior, en los tres aspectos considerados para poder medir el impacto que tuvo la implementación del listado de errores en la metodología de aprendizaje que se está utilizando se observó un incremento cuando se usa este listado sobre el método tradicional de impartir la asignatura. Es relevante señalar que, suponiendo que los datos preserven el mismo comportamiento con el paso de los cursos, se procedió a realizar una simulación pensando en otro proceso de implementación y se observó que con el correr del tiempo se seguirá marcando una diferencia significativa entre las dos estrategias.

## CONCLUSIONES

Resolver un problema es un proceso que requiere tiempo, esfuerzo, involucra una gran variedad de habilidades y de igual modo te hace emplear el conocimiento previo que ya tienes sobre el mismo para poder llegar a una solución adecuada. Sin embargo, cuando se llega al resultado final se tiene una recompensa muchas veces intangible, aunque comúnmente en el ámbito educativo esa recompensa se transforma en un número con significado.

Aprender a programar es una tarea compleja pero esencial tanto para la adquisición de conocimiento como para el desarrollo de las capacidades de pensamiento (Rosanigo y Paur, 2006); dado que es un aspecto de relevante importancia en las carreras del área, es crucial que el estudiante participe activamente en el proceso de superación de sus propios errores y no es suficiente decirle al alumno cuál es el camino correcto o cuál es la solución, debe ser el alumno el que reconozca que su saber es insuficiente o inadaptado, pues de lo contrario continuará recurriendo a él. Si el error es descubierto como consecuencia de una interacción o debate entre profesor y alumno, promoverá la superación, puesto que los estudiantes pueden modificar sus viejas ideas cuando están convencidos de que hay otra que es mejor (Pochulu, 2004).

En nuestro contexto, dada la variedad de sistemas de bachillerato de los cuales provienen nuestros alumnos, se consideró el curso propedéutico como un buen punto de partida para unificar conocimientos y a partir de él se empezó a generar una relación de los principales errores que tienen los alumnos cuando trabajan con algoritmos.

Ahora se cuenta en el contexto de la LIS en la UMT con una relación de errores bien definidos que han venido sirviendo al maestro para apoyar al alumno durante el proceso de enseñanza-aprendizaje; anteriormente no se tenía evidencia justificada de esta relación, por lo que resultó de especial interés conocer sobre el tema, lo cual ha detonado en ayuda al alumno para elevar su rendimiento académico, como se

ha reflejado en los resultados, y superar este tipo de cursos con mejores armas para enfrentarse a asignaturas relacionadas en su proceso formativo. La importancia de conocer a detalle en qué temas los alumnos de la LIS tienen más dificultades al momento de construir sus algoritmos sirvió de base para la planeación de los siguientes cursos (ya impartidos y los que se impartirán en un futuro próximo), incluso fue relevante y positiva, esto se puede deducir de los resultados obtenidos de la estadística aplicada, la cual en todos los casos evaluados ha sido significativa y diferenciadora.

Los incrementos observados en los aspectos estudiados fueron de especial importancia, sin embargo, se considera que se puede ahondar más en los resultados, por lo que como trabajo futuro para corroborar que estos son estadísticamente significativos se desarrollarán diferentes pruebas con el software Rstudio.

## REFERENCIAS

- Balcells, J. (1994). *La investigación social. Introducción a los métodos y las técnicas*. ESPR-PPU.
- Cairó, O. (2005). *Metodología de la programación. Algoritmos, diagramas de flujo y programas*. Alfaomega.
- Dania, C., y Marchisio, S. (2013). Modalidades de percepción sensorial de estudiantes de ingeniería en sistemas de información. Aportes al diseño de material didáctico para la enseñanza de la algoritmia. *Invenio: Revista de Investigación Académica*, 17(31-32), 215-228. <https://dialnet.unirioja.es/servlet/articulo?codigo=4776739>
- Fernández, F. (2002). El análisis de contenido como ayuda metodológica para la investigación. *Revista de Ciencias Sociales*, 2(96). <https://www.revistaciencias-sociales.ucr.ac.cr/images/revistas/RCS96/03.pdf>
- Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. *Lecture Notes in Computer Science*, (4226), 159-168. [https://link.springer.com/chapter/10.1007/11915355\\_15#:~:text=We%20show%20that%20algorithmic%20thinking,developed%20independently%20from%20learning%20programming.&text=A%20proper%20visualization%20of%20these,%2C%20determinism%2C%20parallelism%2C%20etc](https://link.springer.com/chapter/10.1007/11915355_15#:~:text=We%20show%20that%20algorithmic%20thinking,developed%20independently%20from%20learning%20programming.&text=A%20proper%20visualization%20of%20these,%2C%20determinism%2C%20parallelism%2C%20etc)
- Gomes, A., y Medes, A. (2007). *Learning to program - difficulties and solutions* [Ponencia]. International Conference on Engineering Education. Coimbra, Portugal. <http://icee2007.dei.uc.pt/proceedings/papers/411.pdf>
- Gómez, M. (2000). Análisis de contenido cualitativo y cuantitativo: definición, clasificación y metodología. *Revista de Ciencias Humanas*, (20), 103-113. [https://www.academia.edu/37790794/An%C3%A1lisis\\_de\\_contenido\\_cualitativo\\_y\\_cuantitativo\\_Definici%C3%B3n\\_clasificaci%C3%ADa](https://www.academia.edu/37790794/An%C3%A1lisis_de_contenido_cualitativo_y_cuantitativo_Definici%C3%B3n_clasificaci%C3%ADa)
- Hernández, R., Fernández, C., y Baptista, P. (2014). *Metodología de la investigación* (6a. ed.). McGraw-Hill.
- Joyanes, L. (2008). *Fundamentos de programación algoritmos, estructura de datos y objetos*. McGraw-Hill.
- Kerlinger, F. (2002). *Investigación del comportamiento* (3a. ed.). McGraw-Hill.
- Lage, F., y Cataldi, Z. (2019). *Una experiencia de resolución de problemas a través de modelos cooperativos-colaborativos aplicada a algoritmia usando nuevas tecnologías de comunicación* [Ponencia]. I Congreso Internacional de Matemática Aplicada a la Ingeniería y Enseñanza de la Matemática en Ingeniería. Argentina. <http://dragodsm.com/pdf/resolucion-de-problemas.pdf>
- López-Escribano, C., y Sánchez-Montoya, R. (2012). Scratch y necesidades educativas especiales: programación para todos. *RED. Revista de Educación a Distancia*, (34). <https://revistas.um.es/red/article/view/233521/179471>
- Muñoz, R., Barría, M., Noël, R., Providel, E., y Quiroz, P. (2012). *Determinando las dificultades en el aprendizaje de la primera asignatura de programación en estudiantes de ingeniería civil informática* [Ponencia]. XVII Congreso Internacional de Informática Educativa. Santiago, Chile. <http://www.tise.cl/volumen8/TISE2012/17.pdf>

- Pinales, F., y Velázquez, C. (2014). *Algoritmos resueltos con diagramas de flujo y pseudocódigo*. <https://tecnologosmic.files.wordpress.com/2016/01/algoritmos-y-pseudoc3b3digos.pdf>
- Pochulu, M. (2004). Análisis y categorización de errores en el aprendizaje de la matemática en alumnos que ingresan a la universidad. *Revista Iberoamericana de Educación*. <https://rieoei.org/historico/deloslectores/849Pochulu.pdf>
- Rosanigo, Z., y Paur, A. (2006). *Estrategias para la enseñanza de algorítmica y programación* [Ponencia]. I Congreso de Tecnología en Educación y Educación en Tecnología. Argentina. <http://sedici.unlp.edu.ar/handle/10915/19184>
- Sánchez, G., Guerrero, J., y Martínez, E. (2019). Perfil del alumno de computación para el diseño de un sistema tutor. *Certiuni Journal*, (5), 19-26. <http://www.uajournals.com/ojs/index.php/certiunijournal/article/view/504/386>
- Sánchez, G., Guerrero, J., Mocencagua, D., y Reyes, I. (2018). Catálogo de actividades para desarrollar habilidades algorítmicas para un sistema tutor. *Campus Virtuales*, 7(1), 9-17. <http://uajournals.com/ojs/index.php/campusvirtuales/article/view/276/226>
- Tinto, J. (2013). El análisis de contenido como herramienta de utilidad para la realización de una investigación descriptiva. Un ejemplo de aplicación práctica utilizado para conocer las investigaciones realizadas sobre la imagen de marca de España y el efecto país de origen. *Provincia* (29), 135-173. <https://www.redalyc.org/articulo.oa?id=55530465007>
- UADY [Universidad Autónoma de Yucatán] (2012). *Modelo educativo para la formación integral*. <https://www.dgda.uady.mx/media/file/mefi.pdf>
- Vázquez, J. (2012). *Análisis y diseño de algoritmos*. [http://www.aliat.org.mx/BibliotecasDigitales/sistemas/Analisis\\_y\\_diseño\\_de\\_algoritmos.pdf](http://www.aliat.org.mx/BibliotecasDigitales/sistemas/Analisis_y_diseño_de_algoritmos.pdf)
- Xinogalos, S., Satratzemi, M., y Malliarakis, C. (2015). Microworlds, games, animations, mobile apps, puzzle editors and more: What is important for an introductory programming environment? *Education and Information Technologies*, 22(1), 145-176. [https://www.researchgate.net/publication/281847214\\_Microworlds\\_games\\_animations\\_mobile\\_apps\\_puzzle\\_editors\\_and\\_more\\_What\\_is\\_important\\_for\\_an\\_introduutory\\_programming\\_environment](https://www.researchgate.net/publication/281847214_Microworlds_games_animations_mobile_apps_puzzle_editors_and_more_What_is_important_for_an_introduutory_programming_environment)

Cómo citar este artículo:

Narváez Díaz, L. E., y López Martínez, R. E. (2022). Identificación de errores en conceptos básicos de principios de programación. *IE Revista de Investigación Educativa de la REDIECH*, 13, e1222. [http://dx.doi.org/10.33010/ie\\_rie\\_rediech.v13i0.1222](http://dx.doi.org/10.33010/ie_rie_rediech.v13i0.1222)



Todos los contenidos de *IE Revista de Investigación Educativa de la REDIECH* se publican bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional, y pueden ser usados gratuitamente para fines no comerciales, dando los créditos a los autores y a la revista, como lo establece la licencia.