

Propedéutico como estrategia de apoyo en una asignatura de la Licenciatura en Ingeniería de Software de la Unidad Multidisciplinaria Tizimín

Propaedeutic as a support strategy for a course of the Software Engineering bachelor's degree at the Tizimin Multidisciplinary Unit

Narváez Díaz Lizzie Edmea

Universidad Autónoma de Yucatán, México

lendiaz@correo.uady.mx

López Martínez Rocío Edith

Universidad Autónoma de Querétaro, México

rocio.edith.lopez@uaq.mx

Resumen

La enseñanza de la programación de computadoras es un tema complejo que requiere esfuerzo y un enfoque especial en cómo se aprende y enseña, el alto nivel de abstracción y la complejidad de los conceptos requeridos para programar pueden ser un impedimento para muchos alumnos, por lo que constituye un reto no solo para ellos, sino también para los docentes. La introducción a los conceptos básicos de esta área es un tema que ocupa en muchos centros escolares, ya que su claro entendimiento se verá reflejado en asignaturas relacionadas. En el primer semestre de la Licenciatura en Ingeniería de Software de la Unidad Multidisciplinaria Tizimín, se cursa la asignatura de Algoritmia que está relacionada con el tema citado y los resultados obtenidos en los últimos años no han sido satisfactorios, ante esta situación se implementó un curso propedéutico de Algoritmia para apoyar a los alumnos, en el ingreso de agosto 2019. El objetivo fue nivelar a los estudiantes admitidos a la licenciatura antes de su inicio formal y repercutir en el rendimiento académico de la citada materia, enfocado en temas de matemática básica, introducción a los conceptos elementales de programación de computadoras y el lenguaje de programación Scratch. En la metodología seguida en la investigación primero, los alumnos trabajaron con una ficha de datos generales y una prueba diagnóstica para conocer cuánto sabían respecto del área antes del inicio del curso; posteriormente, los resultados fueron empleados durante el propedéutico con sesiones en el salón y en el laboratorio de cómputo, trabajando los temas mencionados y, por último, los alumnos respondieron una prueba de desempeño. Como resultado, concluido el proceso se evaluaron los datos obtenidos de donde se generó mediante un análisis de contenido los

principales problemas que aquejan a los alumnos los cuales dieron pie para impartir la asignatura de Algoritmia el siguiente semestre, de igual modo, del curso propedéutico se obtuvo como resultado final un promedio de 88.5 puntos y en Algoritmia impartida después del propedéutico se obtuvo un promedio de 85 puntos, lo que en ambos casos nos refiere un nivel de dominio *Satisfactorio* de acuerdo a los lineamientos de la Unidad Multidisciplinaria Tizimín. Este trabajo presenta los resultados del proceso y concluye que el curso propedéutico es un buen punto de partida para iniciar con la asignatura de Algoritmia; en función de lo obtenido, así como del resultado de la materia impartida en agosto – diciembre 2019.

Palabras clave: Curso propedéutico, algoritmia, lenguaje de programación Scratch, diagramas de flujo.

Abstract

The teaching of computer programming is a complex topic that requires effort and a special focus on how learning and teaching are done. The high level of abstraction and the complexity of the concepts required for programming can be an impediment for many students, so it becomes a challenge not only for them, but also for teachers. The introduction to the basic concepts of this area is a topic addressed in many schools since their clear understanding will be reflected in related subjects. At the Tizimín Multidisciplinary Unit, during the first semester of the Software Engineering Bachelor's degree it is coursed the Algorithms subject, which is related to the area of programming, and the results obtained in recent years have not been satisfactory. To face this situation, for the August 2019 admission, a propaedeutic course of algorithmics was implemented to support students. The objective was to level the students admitted into the degree before their formal start and to have in impact on their academic performance on the aforementioned subject, focused on topics of basic mathematics, elementary concepts of computer programming and the Scratch programming language. In the methodology followed in the research, the students first filled out a general data sheet and took a diagnostic test to find out how much they knew about the area before the start of the propaedeutic course, then this data were used during the propaedeutic with sessions in the classroom and in the computer lab, working on the aforementioned topics and, finally, the students answered a performance test. As a result, once concluded this process and by means of a content analysis the obtained data were evaluated and the main problems that afflict the students were identified, which served as a guide to teach the subject of Algorithmics in the following semester. Likewise, from the propaedeutic course it was obtained as result an average score of 88.5 points, and from the subject of Algorithmics imparted after the propaedeutic an average score of 85 points was obtained, which in both cases refers to a satisfactory level according

to the Tizimín Multidisciplinary Unit records. This work presents the results of the process and concludes that the propaedeutic course is a good starting point for the subject of Algorithmics, based on what has been obtained.

Keywords: Propaedeutic course, algorithmics, Scratch programming language, flowcharts.

Fecha Recepción: Diciembre 2019

Fecha Aceptación: Junio 2020

Introducción

La complejidad que caracteriza a la programación de computadoras y las actividades que se llevan a cabo relacionadas con ésta, hacen que cuando los alumnos cursen asignaturas relacionadas confronten serias dificultades, siendo una de las principales, la falta de éxito que tienen los mismos en el análisis y la construcción de algoritmos de situaciones problemáticas, en relación con su posterior implementación en un lenguaje de programación determinado (Salgado, Alonzo, Gorina, y Tardo, 2013). A muchos estudiantes se les dificulta aprender a programar; el alto nivel de abstracción y complejidad de los conceptos que deben ser aprendidos es un impedimento serio para muchas personas, para las cuales es una actividad compleja y aburrida (López-Escribano y Sánchez-Montoya, 2012; Gomes y Medes, 2007).

Se vuelve importante mencionar que el bajo rendimiento académico en asignaturas afines con el aprendizaje de principios de programación no es exclusivo de una institución en particular y muchas veces la introducción del estudiante es mediada por el uso de lenguajes que requieren el dominio de la sintaxis y la semántica de los mismos (Joyanes Aguilar, 2003), y esto dificulta el acceso y uso de sistemas computacionales aún más; por esta razón, es que la enseñanza-aprendizaje de estas materias constituye un reto para maestros y estudiantes. En las últimas décadas, debido a las altas tasas de abandono y fracaso en estos cursos, se han hecho extensas investigaciones sobre el tema proponiéndose varios enfoques de enseñanza y metodologías y de igual modo se han diseñado diversas herramientas relacionadas con las Tecnologías de la Información y la Comunicación (TIC) las cuales sirven de ayuda a los alumnos (Xinogalos, Satratzemi, y Malliarakis, 2015; López-Escribano y Sánchez-Montoya, 2012; Gomes y Medes, 2007).

En el contexto de este trabajo, la asignatura de Algoritmia se imparte en el primer semestre de la Licenciatura en Ingeniería de Software, en la Unidad Multidisciplinaria Tizimín, de la Universidad Autónoma de Yucatán y es una materia que tiene las características descritas anteriormente. La asignatura de Algoritmia le permite al alumno analizar un problema y plantear soluciones mediante la descripción ordenada, precisa y finita de una secuencia de instrucciones, aportando al alumno, los elementos básicos para desarrollar un pensamiento lógico y ordenado; de igual modo, lo ayuda a diseñar estrategias en la solución de problemas, facilitando el proceso de la escritura de programas

en cualquier lenguaje de programación (Gómez, Narváez, Rejón, y Reyes, 2016); estos conceptos repercuten en toda la licenciatura, por lo que es crucial en el aprendizaje de los estudiantes. Sin embargo, se ha venido presentando el problema del bajo rendimiento académico originado por la complejidad de los conceptos relacionados que se estudian como son: las variables, los operadores, las secuencias, las decisiones y los ciclos (Rosanigo y Paur, 2006), los cuales son clave durante la introducción de los alumnos a la programación y muchas veces el estudiante se enfrenta a ellos por primera vez en la universidad.

En la Unidad Multidisciplinaria Tizimín el problema de bajo rendimiento académico en Algoritmia se presentó desde la primera vez que se impartió la asignatura en la Licenciatura en Ingeniería de Software, en ese momento se detectó la siguiente situación; en 2016 el 58.33% de los alumnos tuvo un desempeño entre *Satisfactorio* y *Sobresaliente* y el 41.66% entre *Suficiente* y *No aprobado*; en 2017 el 46.15% tuvo un desempeño entre *Satisfactorio* y *Sobresaliente* y el 53.84% entre *Suficiente* y *No aprobado* (cabe hacer mención que a partir de 2018 se hicieron cambios en el modo de impartir la materia y por esa razón no se incluye información de 2018 a la fecha); sin embargo, con los datos presentados se puede observar que el rendimiento general de los grupos no es el óptimo y aunque pudiera ser que el índice de reprobación no pareciera ser tan alto si se expresa un desconocimiento fundamental de los alumnos con los temas, lo que podría acarrear problemas con materias relacionadas que se estudian en los siguientes semestres. Según lo obtenido y lo que se ha venido observado, los alumnos no logran entender el correcto resultado de los problemas planteados; es decir, los algoritmos.

La asignatura de Algoritmia prepara al estudiante promoviendo el pensamiento complejo y el desarrollo de diferentes competencias claves al posicionarlo ante procesos de autocorrección y búsqueda de errores (Vázquez-Cano y Ferrer, 2015). Es importante mencionar que, para resolver un problema de modo algorítmico, no existen reglas específicas a seguir, sin embargo, existen técnicas y herramientas que permiten estructurar el razonamiento; para esto, las etapas de resolución de un problema son el análisis, el diseño y las pruebas, las cuales Cairó (2006) las define de la siguiente manera:

- Análisis, es importante determinar cuál es el problema a resolver, obteniendo los datos que servirán de entrada, determinando qué se espera de salida y teniendo una idea clara de cómo transformar la entrada en la salida esperada.
- Diseño, con la información recabada se elabora o construye el algoritmo (un diagrama de flujo o un pseudocódigo, en el contexto de Algoritmia).
- Pruebas, donde se verifica el resultado y se observa si el algoritmo generado produce la salida esperada para todas las entradas.

En la revisión de la literatura, las estrategias que han usado los docentes en el proceso de enseñanza-aprendizaje en la introducción de sus alumnos a temas que pueden resultar complejos podemos citar los siguientes: en Muñoz et al. (2015), Rizvi et al (2011) y en Ulugad, Karakus y Turner (2011) se observa la impartición de cursos con la herramienta TIC Scratch la cual les sirvió para que el alumno pueda madurar mejor los conceptos relacionados con principios de programación así como también para que el paso a un lenguaje de programación formal sea más fácil, por otro lado, otra estrategia que algunos han empleado aunque no necesariamente en el área computacional es el aprendizaje basado en proyectos mediante el cual los estudiantes estimulan sus habilidades más fuertes y desarrollan nuevas (ITESM, 2010; Delibera, 2015), por último, algunos docentes han implementado cursos propedéuticos previos al inicio del curso formal a modo de apoyo para los temas que se estudiarán posteriormente (Castro, Piñon, y Avilés, 2016; Méndez , Vázquez, y López, 2016).

En función de lo antes expuesto, en la Licenciatura en Ingeniería de Software se han venido planteando diversas soluciones de orientación y ayuda para apoyar al educando en su paso por la asignatura de Algoritmia; algunas de las que se han ido implementando son las siguientes:

- Diseño e impartición de un curso propedéutico para nivelar a los alumnos, antes del inicio formal del semestre escolar.
- Diseño de un curso empleando una herramienta TIC, en este caso el lenguaje de programación Scratch, llevándose a cabo con una intervención en el aula.
- Se aplicó el aprendizaje basado en proyectos durante el curso escolar.
- Se desarrollaron procesos tutoriales individualizados.

En particular, en este trabajo nos centramos en la primera de las opciones planteadas, es decir, en impartir un curso propedéutico para nivelar a los alumnos antes del inicio formal del semestre escolar; cabe hacer mención que otros investigadores también han utilizado este método como estrategia para resolver sus problemas educativos, y al respecto se puede decir que tanto en Castro et al. (2016) como en Méndez et al. (2016), encontraron que los cursos propedéuticos que desarrollaron han cumplido con su objetivo que era robustecer las habilidades y conocimientos previos que se requieren de los estudiantes; en este sentido un curso propedéutico es una buena opción en el proceso educativo y además, se adapta al interés que se tiene por parte del docente de la asignatura de brindar ayuda al estudiante.

El curso propedéutico propuesto fue planteado con la intención de nivelar los conocimientos básicos de los alumnos, promover la solución lógica y ordenada de problemas algorítmicos simples e introducirlos a la programación por bloques mediante el lenguaje de programación Scratch, para que al cursar la asignatura de Algoritmia los alumnos obtengan un mejor rendimiento académico;

todo esto relacionado con los conceptos elementales que deberían saber para cursar la asignatura de modo satisfactorio. Cabe hacer mención, que los alumnos que acuden a la Unidad Multidisciplinaria Tizimín provienen de una diversidad de bachilleratos de la región y suelen llegar con bases distintas en el área de interés en función de los enfoques de sus escuelas de procedencia, de aquí que un propedéutico para unificar conocimientos sobre especial relevancia, en este contexto.

En este trabajo se presenta la metodología seguida durante el proceso de investigación, posteriormente se plantean los resultados obtenidos de los instrumentos empleados, así como del mismo propedéutico y se finaliza con las conclusiones a las que se han llegado luego del análisis de los datos recabados.

Metodología

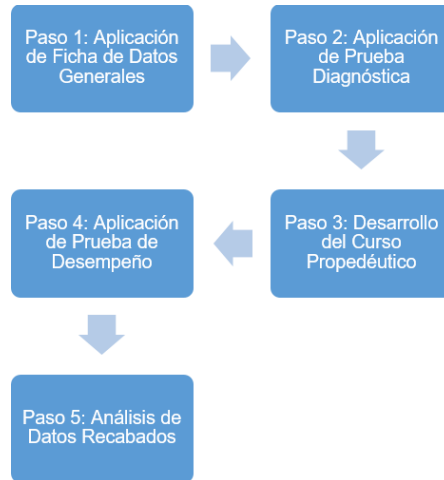
El curso propedéutico propuesto se impartió a los alumnos seleccionados a ingresar a la Licenciatura en Ingeniería de Software antes del inicio de su primer semestre formal en la licenciatura.

Los objetivos específicos del curso propedéutico fueron los siguientes:

1. Nivelar los conocimientos básicos de los alumnos.
2. Desarrollar habilidades y competencias que permitan a los estudiantes prepararse para el curso de Algoritmia en el primer semestre de la Licenciatura en Ingeniería de Software, promoviendo la solución lógica y ordenada de problemas.
3. Introducirlos al uso de los lenguajes de programación, sin el manejo de sintaxis específicas.

La metodología que se describe se aplicó durante los meses de junio y julio de 2019 a la muestra la cual estuvo formada por el total de los 15 alumnos que resultaron seleccionados del proceso de admisión para estudiar la Licenciatura en Ingeniería de Software en la Unidad Multidisciplinaria Tizimín durante el período de agosto de 2019 a mayo de 2023. Los instrumentos de recolección de datos fueron una ficha de datos generales para recabar información de los alumnos, una prueba diagnóstica para saber cómo estaban los alumnos antes de empezar con el propedéutico y una prueba de desempeño la cual fue presentada por los alumnos al término del curso. Con base en lo citado, el curso propedéutico que fue impartido se planteó con la estructura que se aprecia en la figura 1.

Figura 1. Metodología del curso propedéutico



Fuente: Elaboración propia (2020)

La ficha de datos generales que se aplicó fue con el propósito de recabar información general del alumno en cuanto a su escuela de procedencia y estudios sobre algoritmia; posteriormente, se procedió a conocer los antecedentes de los 15 alumnos de nuevo ingreso, para lo cual se les aplicó una prueba diagnóstica que evaluó los conocimientos matemáticos elementales que deben de saber los egresados de bachillerato así como sus antecedentes sobre algoritmia; cabe hacer mención que dado que varios de nuestros alumnos desconocen la palabra algoritmia, se les plantearon los cuestionamientos en la prueba diagnóstica en lenguaje coloquial el cual sí manejan.

El curso propedéutico tuvo una duración de 10 sesiones de 3 horas cada una, siendo un total de 30 horas distribuidas en dos semanas; se impartió en el aula de clase y se empleó el laboratorio para la parte práctica. El último día del curso, los estudiantes fueron evaluados mediante una prueba de desempeño, la cual sirvió para verificar su grado de avance y concluido el proceso descrito se procedió con el análisis de toda la información recabada durante el curso propedéutico.

Resultados

Ficha de Datos Generales

Los estudiantes que participaron del curso propedéutico provienen de diversos bachilleratos de la ciudad de Tizimín y sus comisarías, un alumno del estado de Quintana Roo y uno del estado de Veracruz; los cuales llegan a la Unidad Multidisciplinaria Tizimín con diversos perfiles y conocimientos dado el tipo de bachillerato del cual proceden; se observa en la tabla 1 que provienen de 5 sistemas diferentes. Cabe hacer mención, que sólo los alumnos del Conalep llegan con algunos conocimientos sobre programación, a diferencia de los que llegan de otros bachilleratos.

Tabla 1. Bachillerato de procedencia de los alumnos de nuevo ingreso

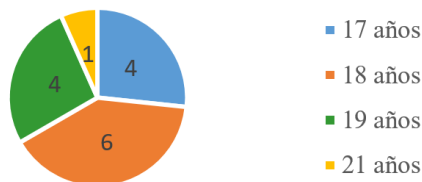
Escuela	Alumnos de Algoritmia Semestre Ago - Dic 2019
	Cantidad de alumnos
Conalep Tizimín	2
Conalep Cancún	1
CBTA No. 36	1
Cecyte	1
Cobay Tixcancal	2
Cobay Tizimín	4
Cobay Sucilá	2
Cobay Calotmul	1
Prepa abierta	1

Fuente: Elaboración propia (2019)

El rango de edades de los alumnos que asistieron fluctúa entre 17 y 21 años, la figura 2 representa este dato.

Figura 2. Gráfica de edades de los alumnos de nuevo ingreso

Cantidad de Alumnos por Edad



Fuente: Elaboración propia (2019)

Es importante también mencionar que de los 15 alumnos a los que se les impartió el curso propedéutico solamente 2 manifestaron haber llevado algún curso relacionado con la algoritmia anteriormente y 4 habían utilizado Scratch.

Prueba Diagnóstica

La prueba diagnóstica aplicada a los alumnos se dividió en cinco partes, las cuales se describen a continuación:

Parte I: Se les solicitó a los alumnos que elaboren un algoritmo de una actividad simple de la vida cotidiana empleando lenguaje coloquial, a este respecto todos lograron hacerlo, aunque con diferentes niveles de detalle; cabe mencionar que en Algoritmia el nivel de detalle es un punto importante, de aquí el interés en este ejercicio. Solo una persona logró hacer el algoritmo al 100%.

Parte II: Se les solicitó resolver operaciones aritméticas en las que debían tener presente la precedencia de operadores, así como las reglas de asociatividad (conceptos básicos del nivel de bachillerato), los resultados se aprecian en la tabla 2.

Tabla 2. Respuestas a la Parte II del examen de diagnóstico

Aciertos correctos	Examen diagnóstico
	Porcentaje de alumnos
0	20%
1	26.67%
2	53.33%

Fuente: Elaboración propia (2019)

Parte III: Se les solicitó resolver cinco ejercicios los cuales empleaban los operadores relacionales, estos se estudian en bachillerato y son relevantes al hacer comparaciones en Algoritmia, los resultados se pueden apreciar en la tabla 3.

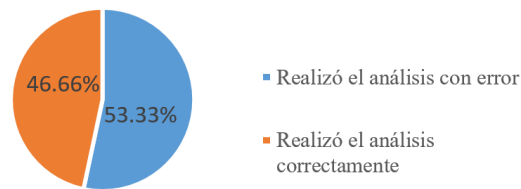
Tabla 3. Respuestas a la Parte III del examen de diagnóstico

Aciertos correctos	Examen diagnóstico
	Porcentaje de alumnos
0	0%
1	0%
2	6.67%
3	53.33%
4	20%
5	20%

Fuente: Elaboración propia (2019)

Parte IV: Se les solicitó que analizaran el problema siguiente *encuentra el promedio de las calificaciones de un alumno que cursa cinco materias durante un semestre*, un ejercicio simple donde el análisis lo tenían que hacer en función de los datos de entrada, los datos de salida y el proceso. Como resultado de este ejercicio, un 53.33% de los alumnos hizo un planteamiento al que le faltó detallar algún concepto, por ejemplo: no identificaban que tenían el número de materias y en consecuencia trataban de obtenerlo, intentaron obtener el nombre de las materias (lo cual era irrelevante), el resultado lo daban en porcentaje cuando lo que se solicitó fue un promedio o bien no especificaron la suma de las calificaciones. El 46.66% de los alumnos lo realizaron correctamente. Cabe destacar que el ejercicio que se solicitó es de los más simples de Algoritmia, se optó por este porque es un problema básico de la vida cotidiana que aun cuando no tengan conocimientos del área deberían poder resolverlo sin ningún problema (figura 3).

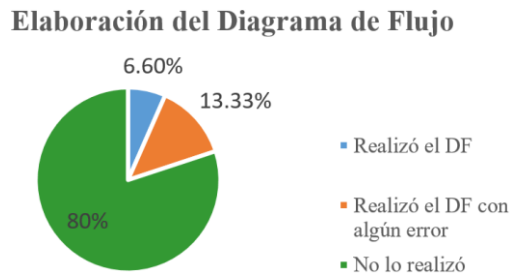
Figura 3. Respuestas a la Parte IV del examen de diagnóstico
Análisis de Problema



Fuente: Elaboración propia (2019)

Parte V: Como punto final se les pidió obtener el diagrama de flujo del ejercicio desarrollado en la parte IV, es importante decir que este ejercicio fue incluido para tener una idea precisa de cuántos tenían conocimientos más exactos de la materia, de antemano no se esperaba que fuera resuelto por la mayoría ya que se sabe que en varios bachilleratos no se estudian cursos relacionados con esta área. En este sentido, solo una persona lo logró realizar correctamente es decir el 6.67% de los alumnos y dos personas tuvieron una idea bastante clara, pero con varios errores en el proceso, esto nos representó el 13.33% de los alumnos. El 80% restante no lo pudo resolver (ver figura 4).

Figura 4. Respuestas a la Parte V del examen de diagnóstico



Fuente: Elaboración propia (2019)

El resultado promedio de la prueba diagnóstica fue de 58.9 puntos lo cual nos represente un nivel de dominio de *No Acreditado* de acuerdo a la Universidad Autónoma de Yucatán, la prueba presentada y el análisis de la misma afirman la importancia de que los alumnos de nuevo ingreso lleven el curso propedéutico para nivelar conocimientos antes de empezar con el curso de Algoritmia formalmente ya que existen claras deficiencias en la parte que es de interés en el contexto.

Curso Propedéutico

El curso se estructuró teniendo como referencia los conceptos básicos e introductorios que se estudian en el contenido de la materia Algoritmia (Gómez et al., 2016), mismo que estuvo dividido en tres temas de especial importancia.

- *Tema 1:* Introducción a la algoritmia; en donde se detalló la definición y características de los algoritmos, los identificadores y en cuanto a operadores matemáticos se enfocó en los aritméticos y relacionales.
- *Tema 2:* Diagramas de flujo; en este apartado se definieron los diagramas de flujo y se presentó su simbología básica, en cuanto a la creación de algoritmos se enfocó a la estructura secuencial básica y la estructura alternativa simple, doble y múltiple.
- *Tema 3:* Lenguaje de programación Scratch; en esta última parte se siguió trabajando con las estructuras secuenciales y alternativas, transcribiendo los diagramas de flujo que se tenían desarrollados en el tema 2 al lenguaje de programación Scratch.

Estos contenidos fueron diseñados sin perder de vista el objetivo de aumentar el rendimiento académico de los alumnos que cursan Algoritmia en el primer semestre de la Licenciatura en Ingeniería de Software, de aquí la importancia de apegarse a la planeación didáctica de la materia. Durante la impartición del curso se promovió la participación activa de los estudiantes, que implica el compromiso de las partes para alcanzar los propósitos planteados, los ejercicios fueron presentados y analizados en conjunto, posteriormente cada alumno hizo una propuesta de resolución del problema para después discutir las diversas soluciones planteadas, de este modo se estimuló el trabajo responsable de cada uno de los participantes al analizar y extraer las características más relevantes de las situaciones problemáticas; discutir y encontrar formas de solución de los problemas y elegir, entre ellas, las más eficaces, así como fundamentar en todo momento, el porqué de la estrategia de solución empleada.

Prueba de desempeño

El último día del curso los estudiantes fueron evaluados mediante una prueba de desempeño, de los 15 que empezaron el curso solamente 14 presentaron la prueba y ésta estuvo compuesta de las siguientes secciones:

Sección 1: Se presentaron ejercicios con operaciones aritméticas.

Sección 2: Preguntas de opción múltiple, relacionadas a conceptos clave de la asignatura de Algoritmia.

Sección 3: Problemas a resolver mediante el diseño de un diagrama de flujo.

Al término de la prueba se procedió a realizar el análisis de las mismas y los resultados encontrados fueron los siguientes:

En la sección 1, se les pidió resolver dos ejercicios con operadores aritméticos en los que se debía cuidar la precedencia de los mismos y las reglas de asociatividad; el 57.14% de los alumnos logró hacer los dos ejercicios correctamente, el 28.57% realizó uno de modo correcto y en el otro pudo

hacer parte del desarrollo (lo cual se consideró como la mitad), por último el 14.29% realizó un ejercicio correctamente, estos resultados manifiestan un ligero avance con respecto a la prueba diagnóstica, se destaca que ningún alumno obtuvo incorrecto en los dos ejercicios planteados (ver tabla 4).

Tabla 4. Respuestas a la Sección 1 de la prueba de desempeño final

Aciertos correctos	Prueba de desempeño
	Porcentaje de alumnos
1	14.29%
1.5	28.57%
2	57.14%

Fuente: Elaboración propia (2019)

En la sección 2, se plantearon 4 preguntas de opción múltiple, teniendo como resultado que el 85.71% de los alumnos logró responder correctamente las 4, el 7.14% obtuvo 3 aciertos correctos y el 7.14% obtuvo 2 aciertos correctos, nadie obtuvo 1 o cero aciertos incorrectos, se concluye que en esta sección se obtuvieron buenos resultados (ver tabla 5).

Tabla 5. Respuestas a la Sección 2 de la prueba de desempeño final

Aciertos correctos	Examen diagnóstico
	Porcentaje de alumnos
0	0%
1	0%
2	7.14%
3	7.14%
4	85.71%

Fuente: Elaboración propia (2019)

En la sección 3, se les pidió a los alumnos que resolvieran 4 problemas mediante un diagrama de flujo, haciendo primero el análisis de los mismos. El grado de complejidad de los problemas iba de menor a mayor y se enfocaron a los siguientes temas:

Ejercicio 1: Estructura secuencial.

Ejercicio 2: Estructura alternativa simple.

Ejercicio 3: Estructura alternativa doble.

Ejercicio 4: Estructura alternativa múltiple.

El ejercicio 1 el 93% de los alumnos lo tuvo correctamente, el ejercicio 2 el 100% lo tuvo bien, el ejercicio 3 el 86% lo tuvo bien y por último el ejercicio 4 el 71 % lo realizó adecuadamente.

En esta prueba de desempeño final el promedio de las calificaciones fue de 88.5 puntos, lo cual nos representa un nivel de dominio *Satisfactorio* (de acuerdo con los niveles de dominio que emplea la

Universidad Autónoma de Yucatán en su proceso de evaluación) e indica un resultado adecuado en este curso propedéutico de nivelación (UADY, 2012). Por otra parte, se puede apreciar un incremento de 28.6 puntos respecto de la prueba diagnóstica presentada antes del curso propedéutico.

Discusión

La asignatura Algoritmia suele resultar una materia complicada a los alumnos que recién inician en el campo de la programación de computadoras y este hecho se ha podido constatar a lo largo del tiempo con los alumnos de la Licenciatura en Ingeniería de Software; concordando con Salgado et al. (2012) quienes señalan que el carácter del proceso de resolución de problemas de programación, junto a la complejidad y dificultades detectadas en su enseñanza, han contribuido a despertar la preocupación por el aprendizaje de la programación y a interesar a una amplia comunidad científica, en este sentido es que resulta importante este estudio el cual pretende solventar de cierto modo la situación.

En este caso se procedió a brindar apoyo a los estudiantes de nuevo ingreso por medio de la implementación obligatoria de un curso propedéutico para igualar el nivel de conocimientos de los alumnos dado que proceden de 5 sistemas diferentes y partiendo de la idea de que otros investigadores como Méndez et al. (2016), así como también Castro et al. (2016) han presentado buenos resultados en la implementación de cursos propedéuticos en situaciones similares a la de interés.

El curso propedéutico logró el objetivo planteado que fue nivelar a los estudiantes admitidos, esto fue comprobado al contrastar los resultados obtenidos en la prueba diagnóstica versus la prueba de desempeño final, sin embargo, estos resultados no quedaron aquí dado que al término del semestre agosto – diciembre 2019, los resultados alcanzados por el grupo al que se le impartió el propedéutico resultó *Satisfactorio*, lo cual es relevante para efectos de esta investigación, por lo que en concordancia con Castro et al. (2016) se lograron robustecer las habilidades y conocimientos previos que se requieren.

Limitantes del estudio

La población estudiada es pequeña, aunque es un número que no se puede modificar, ya que hay que ajustarse a la cantidad de alumnos que solicitan inscribirse a la Licenciatura en Ingeniería de Software, en este sentido los resultados solo son aplicables al grupo de estudio.

La presente investigación representa solo un proceso implementado en el hecho de brindar apoyo a los alumnos de nuevo ingreso, en este sentido, se vuelve necesario desarrollar otros métodos para mejorar aún más los resultados que se encuentren.

Conclusiones

Como se citó al inicio, una de las razones de impartir el curso propedéutico es que éste le sirva de base a los alumnos de nuevo ingreso para que al cursar la materia de Algoritmia puedan obtener un mejor rendimiento en la materia, por lo que fue importante el análisis de los resultados para conocer en qué aspectos presentan más carencias los estudiantes, en este sentido, en función de un análisis de contenido aplicado a la prueba de desempeño final y a todo lo observado durante la impartición del curso, los errores más frecuentemente detectados en el proceso de análisis y construcción de los diagramas de flujo y en los que se puso especial interés durante el curso de Algoritmia, fueron los siguientes:

- Errores matemáticos.
- Deficiente comprensión lectora.
- Problemas de asignación de resultados a variables.
- Leer variables en vez de calcularlas.
- Cerrar inadecuadamente las estructuras alternativas.
- Diversos detalles generales como:
 - Falta de lectura de una variable que posteriormente se emplea.
 - No imprimir una variable que se pedía de resultado final.
 - No terminar (cerrar) adecuadamente el diagrama de flujo.
 - Otros en menor importancia.

En cuanto al uso de Scratch en el laboratorio, la experiencia fue satisfactoria para los estudiantes ya que se observó que no solo se esforzaban por acabar el ejercicio, sino que también lo mejoraban añadiéndoles conceptos de animación, esto se logró determinar en los ejercicios que entregaron al final del curso. De igual modo es importante destacar que el uso de Scratch los ayudó a fomentar su creatividad lo cual se pudo observar en sus programas generados.

Por último, se considera que de modo general se obtuvieron buenos resultados en el curso y de alguna forma ya arrojó resultados en la asignatura de Algoritmia que se impartió durante el semestre agosto – diciembre 2019, ya que al término del curso se obtuvo un promedio de calificaciones de 85 puntos lo que representa un resultado *Satisfactorio* de acuerdo a los lineamientos de la Universidad Autónoma de Yucatán y por otro lado solo hubieron 2 reprobados

de un total de 19 alumnos que llevaron el curso, cabe mencionar que estos resultados marcan diferencia respecto de los años en los que no se ha impartido un curso propedéutico como el que se ha descrito. De acuerdo con lo que aquí se ha presentado la nivelación de los alumnos por medio del curso propedéutico ha sido la esperada y ha cumplido con el objetivo planteado.

Es importante mencionar que, aunque se ha tenido un primer acercamiento al problema que nos aqueja, es importante hacer estudios estadísticos más profundos a los datos recabados y en función de lo que se obtenga seguir avanzado en el área.

Referencias

- Cairó, O. (2006). Metodología de la programación Algoritmos, Diagramas de Flujo y Programas (3a. ed.). México: Alfaomega.
- Castro, F., Piñon, A., & Avilés, J. (2016). La experiencia de un curso de fortalecimiento académico en el bachillerato universitario para facilitar el ingreso y permanencia a la educación superior; logros y desafíos. Congresos CLABES. Obtenido de <https://revistas.utp.ac.pa/index.php/clabes/article/view/1093>
- Delibera. (2015). Aprendizaje Basado en Proyectos. Recuperado el 10 de Enero de 2019, de <https://www.bcn.cl/obtienearchivo?id=documentos/10221.1/55744/1/Aprendizaje%20basado%20en%20proyectos.pdf>
- Gomes, A., & Medes, A. (2007). Learning to program - difficulties and solutions. International Conference on Engineering Education – ICEE 2007, (págs. 283-287). Coimbra.
- Gómez, J., Narváez, L., Rejón, E., & Reyes, J. (9 de Agosto de 2016). Planeación didáctica de Algoritmia. Mérida, Yucatán, México.
- Huilcapi, A. (2015). Estrategias didácticas basadas en el software educativo Scratch, para la correcta aplicación de las estructuras repetitivas While.
- ITESM. (2010). Investigación e Innovación Educativa. Recuperado el 10 de Enero de 2019, de <http://sitios.itesm.mx/va/dide2>
- Joyanes Aguilar, L. (2003). Fundamentos de Programación: Algoritmos y Estructuras de Datos (Tercera Edición ed.). McGraw-Hill Interamericana.
- Juárez, J., López, M., & Villareal, Y. (2016). Estrategias para Reducir el Índice de Reprobación en Fundamentos de Programación de Sistemas Computacionales del I.T. Mexicali. Revista de Gestión Empresarial y Sustentabilidad, 2(1), 25-41.
- López-Escribano, C., & Sánchez-Montoya, R. (2012). Scratch y Necesidades Educativas Especiales: Programación para todos. RED. Revista de Educación a Distancia. Recuperado el 16 de Abril de 2019, de <https://www.redalyc.org/html/547/54724753001/>

- Méndez, R., Vázquez, E., & López, R. (2016). Efecto de los cursos propedéuticos en la licenciatura en ingeniería de la Facultad de Ingeniería de la UADY. *Ingeniería*, 20(3), 128-136.
- Mirete, A., & García, F. (Enero de 2014). Rendimiento académico y tic. Una experiencia con webs didácticas en la Universidad de Murcia. *Pixel-Bit. Revista de Medios y Educación*(44), 169-183.
- Muñoz, R., Barcelos, T., Villarroel, R., Barría, M., Becerra, C., Noel, R., & Frango, I. (2015). Uso de Scratch y Lego Mindstorms como apoyo a la docencia en Fundamentos de Programación. *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática*, (págs. 248-254). Andorra La Vella.
- Narváez, L., & López, R. (2018). Conceptualización de la Herramienta TIC Scratch y sus Aplicaciones. 2do. Congreso de Computación y Tecnología Educativa, (págs. 174-185). Querétaro.
- Pajares, R. (2014). Estudio del Impacto Educativo de las actividades formativas basadas en ScratchTM en alumnos de 1ºESO del IES Antonio Tovar de Valladolid. Valladolid, España.
- Resnick, M., Maloney, J., Monroy- Hernández, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (2009). *Scratch: Programming for All*.
- Rizvi, M., Humphries, T., Major, D., Jones, M., & Lauzun, H. (Enero de 2011). A CS0 course using Scratch. *J Comput Sci Coll*, 26(3), 19-27.
- Rosanigo, Z., & Paur, A. (2006). Estrategias para la enseñanza de Algorítmica y Programación. I Congreso de Tecnología en Educación y Educación en Tecnología.
- Salgado, A., Alonzo, I., Gorina, A., & Tardo, Y. (2013). Lógica algorítmica para la resolución de problemas de programación computacional: una propuesta didáctica. *Revista Didasc@lia: D&E.*, 4(1), 57-76.
- UADY. (12 de Julio de 2012). Modelo educativo para la formación integral. Mérida, Yucatán, México.
- Uludag, S., Karakus, M., & Turner, S. (2011). Implementing IT0/CS0 with scratch, app inventor for android, and lego mindstorms. *SIGITE Conference*.
- Vázquez-Cano, E., & Ferrer, D. (2015). La creación de videojuegos con Scratch en Educación Secundaria. *Communication Papers Media Literacy & Gender Studies*, 4(6), 63-73.

Xinogalos, S., Satratzemi, M., & 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.
doi:<https://doi.org/10.1007/s10639-015-9433-1>