



Universidad Autónoma de Querétaro

Facultad de Informática

Metodología de aprendizaje usando Scratch para mejorar el
rendimiento académico de los alumnos de la asignatura de
algoritmia

Tesis

Que como parte de los requisitos
para obtener el Grado de

Doctora en Innovación en Tecnología Educativa

Presenta

Lizzie Edmea Narváez Díaz

Dirigido por:

Dra. Rocío Edith López Martínez

Querétaro, Qro. a 14 de junio de 2021



Universidad Autónoma de Querétaro
Facultad de Informática
Doctorado en Innovación en Tecnología Educativa

Metodología de aprendizaje usando Scratch para mejorar el rendimiento académico de los alumnos de la asignatura de algoritmia

Tesis

Que como parte de los requisitos para obtener el Grado
Doctora en Innovación en Tecnología Educativa

Presenta

Lizzie Edmea Narváez Díaz

Dirigido por:

Dra. Rocío Edith López Martínez

Dra. Rocío Edith López Martínez
Presidente

Dra. Ma. Teresa García Ramírez
Secretario

Dr. Alexandro Escudero Nahón
Vocal

Dr. Ricardo Chaparro Sánchez
Suplente

Dr. José Luis López Martínez
Suplente

Centro Universitario, Querétaro, Qro.
Junio 2021
México

DEDICATORIAS

A Dios, quien me regala la vida para continuar.

A Rafael y Andrea, por ser el principal motor de mi vida.

A mi madre, por su apoyo incondicional siempre.

A mis hermanos y toda mi familia.

AGRADECIMIENTOS

Al Programa para el Desarrollo Profesional Docente (PRODEP) por la asignación de la beca *Apoyo para estudios de posgrado de alta calidad* con la que he podido continuar estudiando y facilitar el ingreso al Doctorado en Innovación en Tecnología Educativa que hoy es parte de mis conocimientos y fortalece mi práctica profesional.

A mi directora de tesis, Dra. Rocío Edith López Martínez, por su valioso apoyo y motivación constante en todo el desarrollo de este trabajo de investigación.

A la Dra. María Teresa García Ramírez y al Dr. Alexandro Escudero Nahón, integrantes de mi comité tutorial, por sus invaluable comentarios en el desarrollo de esta tesis.

Al Dr. Ricardo Chaparro Sánchez y al Dr. José Luis López Martínez, integrantes del sínodo revisor del trabajo desarrollado, gracias por sus valiosos comentarios.

A la Facultad de Informática de la Universidad Autónoma de Querétaro, docentes y administrativos por haberme recibido y darme la oportunidad de continuar y contribuir con mi preparación académica en las aulas virtuales.

A mi familia y amigos que nunca dudaron que lo lograría y siempre tuvieron esas palabras que me motivaban a seguir adelante.

ÍNDICE

| | |
|---|------|
| DEDICATORIAS..... | iii |
| AGRADECIMIENTOS..... | iv |
| ÍNDICE..... | v |
| ÍNDICE DE TABLAS..... | vii |
| ÍNDICE DE FIGURAS..... | viii |
| ABREVIATURAS Y SIGLAS..... | ix |
| RESUMEN..... | x |
| ABSTRACT..... | xii |
| 1. INTRODUCCIÓN..... | 13 |
| 2. LAS TIC EN LA EDUCACIÓN, LA ALGORITMIA Y SCRATCH..... | 22 |
| 2.1 Las TIC en la educación..... | 22 |
| 2.1.1 Conceptualización..... | 25 |
| 2.1.2 Aplicación de herramientas TIC..... | 28 |
| 2.1.3 Perspectiva del docente en la incorporación de TIC..... | 31 |
| 2.2 Algoritmia..... | 34 |
| 2.2.1 Conceptualización..... | 35 |
| 2.2.2 Situación actual de la algoritmia en instituciones de nivel superior..... | 38 |
| 2.2.3 Problemas comunes y herramientas usadas para mejorar..... | 42 |
| 2.3 Scratch..... | 45 |
| 2.3.1 Conceptualización..... | 49 |
| 2.3.2 Uso de Scratch en la educación..... | 52 |

| | | |
|-------|---|-----|
| 2.3.3 | Uso de Scratch en la universidad..... | 54 |
| 3. | METODOLOGÍA..... | 58 |
| 3.1 | Problemática en estudio | 58 |
| 3.1.1 | Sujeto experimental..... | 59 |
| 3.2 | Justificación del problema..... | 60 |
| 3.3 | Establecimiento de la hipótesis..... | 65 |
| 3.4 | Objetivos del estudio..... | 66 |
| 3.5 | Diseño del estudio | 66 |
| 3.6 | Población y muestra | 71 |
| 3.7 | Datos requeridos | 72 |
| 3.8 | Instrumentos..... | 72 |
| 3.9 | Procedimiento para la obtención y análisis de datos recabados | 73 |
| 3.10 | Limitaciones y delimitaciones del estudio | 74 |
| 3.11 | Propuesta: Metodología de aprendizaje usando Scratch..... | 75 |
| 4. | RESULTADOS Y DISCUSIÓN..... | 83 |
| 4.1 | Resultados..... | 83 |
| 4.2 | Discusión | 104 |
| 5. | CONCLUSIONES..... | 119 |
| | REFERENCIAS BIBLIOGRÁFICAS..... | 124 |
| | ANEXOS..... | 136 |
| | Anexo A Cuestionario | 136 |

ÍNDICE DE TABLAS

| | |
|---|-----|
| Tabla 2.1 Perspectivas docentes sobre TIC. | 32 |
| Tabla 2.2 Niveles de dominio para la acreditación de una asignatura en la UADY. | 41 |
| Tabla 4.1 Bachillerato de procedencia de los alumnos de nuevo ingreso a la LIS. | 85 |
| Tabla 4.2 Respuestas a la segunda sección de la prueba diagnóstica..... | 87 |
| Tabla 4.3 Respuestas a la tercera sección de la prueba diagnóstica. | 87 |
| Tabla 4.4 Respuestas al apartado resolución de operaciones aritméticas de la prueba de desempeño final. | 90 |
| Tabla 4.5 Respuestas al apartado de identificación de conceptos básicos de principios de programación de la prueba de desempeño final..... | 90 |
| Tabla 4.6 Codificación del análisis de contenido de las pruebas de desempeño del propedéutico. | 92 |
| Tabla 4.7 Pruebas de desempeño analizadas..... | 93 |
| Tabla 4.8 Errores en las pruebas de desempeño, agosto - diciembre de 2018 y 2019. | 94 |
| Tabla 4.9 Alumnos aprobados-reprobados en algoritmia 2016 a 2021..... | 96 |
| Tabla 4.10 Estadísticos del cuestionario de actitud hacia el uso de la TIC Scratch. | 102 |
| Tabla 4.11 Estadísticos del cuestionario de opinión sobre las TIC..... | 103 |
| Tabla 4.12 Estadísticos del cuestionario de opinión de los alumnos hacia Scratch. | 104 |

ÍNDICE DE FIGURAS

| | |
|--|-----|
| Figura 2.1 Etapas en la resolución de un problema..... | 37 |
| Figura 2.2 Primera versión al público Scratch 1.0 en 2007..... | 46 |
| Figura 2.3 Scratch versión 2.0 en el 2013. | 48 |
| Figura 2.4 Scratch versión 3.0 enero de 2019. | 49 |
| Figura 2.5 Bloques de Scratch 2.0 y 3.0. | 51 |
| Figura 3.1 Fases generales de la IBD..... | 68 |
| Figura 3.2 Metodología de aprendizaje usando Scratch..... | 77 |
| Figura 4.1 Gráfica de las edades de los alumnos de nuevo ingreso a la LIS. | 86 |
| Figura 4.2 Respuestas a la sección cuatro de la prueba diagnóstica (análisis de problema)..... | 88 |
| Figura 4.3 Respuestas a la sección cinco de la prueba diagnóstica (DF)..... | 89 |
| Figura 4.4 Prueba Shapiro-Wilk para residuales..... | 98 |
| Figura 4.5 Prueba Kruskal-Wallis a los promedios de las calificaciones con los 2 métodos. | 98 |
| Figura 4.6 Prueba Fisher's Exacta a los aprobados-reprobados con los 2 métodos. | 99 |
| Figura 4.7 Prueba Fisher-Exacta de acuerdo con el nivel de dominio..... | 100 |
| Figura 4.8 Prueba Shapiro-Wilk con las calificaciones finales de algoritmia..... | 101 |
| Figura 4.9 Prueba Wilcoxon con las calificaciones finales de algoritmia. | 101 |

ABREVIATURAS Y SIGLAS

| | |
|---------|--|
| ABP | Aprendizaje Basado en Proyectos |
| CBTA | Centro de Bachillerato Tecnológico Agropecuario |
| CECyTEY | Colegio de Estudios Científicos y Tecnológicos del Estado de Yucatán |
| Cobay | Colegio de Bachilleres del Estado de Yucatán |
| Conalep | Colegio Nacional de Educación Profesional Técnica |
| DF | Diagrama de Flujo |
| IBD | Investigación Basada en Diseño |
| LIS | Licenciatura en Ingeniería de Software |
| MEFI | Modelo Educativo para la Formación Integral |
| NSF | National Science Foundation |
| SICEI | Sistema de Información y Control Escolar Institucional |
| TIC | Tecnologías de la Información y la Comunicación |
| UADY | Universidad Autónoma de Yucatán |
| UMT | Unidad Multidisciplinaria Tizimín |

RESUMEN

La enseñanza y el aprendizaje de los principios de programación constituyen un reto tanto para maestros como para los estudiantes; debido a las tasas de abandono y fracaso en estos cursos, se han llevado a cabo extensas investigaciones sobre el tema, en las cuales se han propuesto varios enfoques de enseñanza, metodologías, y han sido diseñadas diversas herramientas para tal efecto. La aplicación de conceptos básicos o el diseño de algoritmos es difícil para muchos estudiantes, siendo que la mayor parte de sus problemas son originados por la complejidad de los conceptos que se estudian como las variables, secuencias, decisiones y ciclos. En la Unidad Multidisciplinaria Tizimín de la Universidad Autónoma de Yucatán se imparte la materia de algoritmia la cual presenta la situación anteriormente descrita. Partiendo de la revisión de la literatura científica y de lo que hasta ahora se ha hecho en el área, este trabajo expone los resultados de una investigación realizada con la metodología de la investigación basada en diseño, para brindar ayuda a los alumnos que tienen problemas con este tema, el objetivo consistió en crear una metodología de aprendizaje que incorpore el lenguaje de programación *Scratch* en la asignatura de algoritmia para mejorar el rendimiento académico de los alumnos. Entre los principales resultados que se tienen se encuentran los siguientes: *Scratch* ha sido bien recibido por los alumnos, se determinaron los principales errores que tienen los estudiantes cuando trabajan con fundamentos de programación, se elevó el promedio de aprovechamiento de los grupos, se verificó que los índices de aprobación mejoran con la implementación de esta metodología, se incrementó el índice de alumnos con calificaciones en los niveles de dominio sobresaliente y satisfactorio, así como también se comprobó que el promedio de aprovechamiento de los alumnos se encuentra en un nivel satisfactorio. En general los resultados obtenidos arrojan una diferencia significativa cuando se emplea el lenguaje de programación *Scratch* en la materia de algoritmia, en contraste con el método

tradicional, se concluye que la intervención tuvo un mayor impacto en el rendimiento académico de los alumnos.

Palabras Clave: Algoritmia, *Scratch*, Principios de programación, Investigación Basada en Diseño.

Dirección General de Bibliotecas UAQ

ABSTRACT

The teaching and learning of programming principles is challenging for both teachers and students; due to the dropout and failure rates in these courses, extensive research has been carried out on the subject, in which various teaching approaches and methodologies have been proposed, and many tools have been designed for this purpose. The application of basic concepts or the design of algorithms seems to be something difficult for many students, and most of their problems are caused by the complexity of related concepts to be programming principles such as variables, sequences, decisions, and cycles. At the Tizimin Multidisciplinary Unit of the Autonomous University of Yucatan, the subject of algorithmics is taught, which presents the situation described above. This work presents the results of an investigation carried out with the methodology of design-based research, using as a starting point a review of the scientific literature and what has been done so far in the area to provide help to students who have problems with this topic; according to this, the objective was to create a learning methodology that incorporates the *Scratch* programming language in the algorithmics subject to improve the academic results of the students. Among the main results we have are the following: Scratch has been well received by students, the main mistakes that students make when working with programming fundamentals were determined, the average level of achievement of the groups was raised, it was verified that the approval rates improve with the implementation of this methodology, the index of students with grades in the levels of outstanding and satisfactory domain was increased, and it was also found that the average student achievement is at a satisfactory level. In general, the results obtained show a significant difference when using the Scratch programming language in the field of algorithmics; in contrast to the traditional method, it is concluded that the intervention had a greater impact on the academic performance of students.

Keywords: Algorithmics, *Scratch*, Programming principles, Design-based research.

1. INTRODUCCIÓN

En este capítulo se describe el problema al que se enfrentan los alumnos en el área de programación de computadoras, de igual modo se cita lo que sucede al respecto en el contexto de esta investigación y para dar sustento a la problemática se mencionan algunos estudios relacionados, asimismo, se plantea a las tecnologías de la información y la comunicación (TIC) como una opción viable de soporte y apoyo en este tipo de situaciones haciendo referencia tanto a sus ventajas como a sus desventajas.

La enseñanza y el aprendizaje de los principios de programación constituyen un reto tanto para los maestros como para los estudiantes respectivamente. En las últimas décadas, debido a las tasas de abandono y fracaso en estos cursos, se han llevado a cabo extensas investigaciones sobre el tema, en las cuales se han propuesto varios enfoques de enseñanza y metodologías, así como también han sido diseñadas diversas herramientas para tal efecto (Xinogalos, Satratzemi, & Malliarakis, 2015; López-Escribano & Sánchez-Montoya, 2012; Gomes & Medes, 2007). La programación de computadoras en contextos escolares es una temática ampliamente estudiada a nivel internacional y actualmente está siendo reconsiderada a través de propuestas innovadoras (Monjelat, Cenacchi, & San Martín, 2018).

Muchos estudiantes tienen dificultades para aprender a programar; la programación por sí misma es un tema complejo que requiere esfuerzo y un enfoque especial en la forma en que se aprende y se enseña (Gomes y Medes, 2007). El alto nivel de abstracción y la complejidad de los conceptos que deben ser aprendidos para esta actividad son un impedimento serio para muchas personas, para las cuales es una actividad compleja y aburrida, destinada a un grupo reducido de personas (López-Escribano & Sánchez-Montoya, 2012). Por otro lado, cuando se introduce al estudiante a los principios básicos del área, esta tarea usualmente es mediada por el uso de lenguajes cuyo aprendizaje requiere el dominio de la sintaxis y semántica del lenguaje en cuestión (Joyanes Aguilar, 2003), este hecho

a menudo dificulta aún más el acceso y el uso de los sistemas computacionales por parte del alumnado y obstaculiza el desarrollo del pensamiento lógico y algorítmico el cual es indispensable en estos procesos para dar solución a los ejercicios que se plantean.

Asimismo, la aplicación de conceptos elementales o el diseño de algoritmos como los diagramas de flujo que son relativamente simples para los docentes parece ser algo difícil para muchos estudiantes. La mayor parte de esos problemas son originados por la complejidad de los conceptos relacionados con los principios de programación como las variables, secuencias, decisiones y ciclos, los cuales son temas clave durante la introducción de los alumnos al área y a los que muchas veces el estudiante se enfrenta por primera vez en la universidad. Es importante considerar, que ésta es una tarea complicada por su misma naturaleza, ya que tratar de hacer entender al alumno algo que es abstracto y de lo que no se tiene un referente físico, es una tarea compleja.

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), es una asignatura que presenta la situación anteriormente descrita. Algoritmia, está formada de 3 unidades y los contenidos clave que se le complican al alumno se ubican en la unidad 2 la cual se enfoca al análisis, diseño y construcción de los diagramas de flujo; cabe hacer mención que en la unidad 1 solo se ven conceptos teóricos y en la unidad 3 se analizan los mismos temas de la unidad 2 pero enfocados al pseudocódigo (Gómez, Narváez, Rejón, & Reyes, 2016), no obstante, se consideran cruciales los diagramas de flujo porque se ha visto en los cursos implementados que si el alumno lo genera adecuadamente, entonces el pseudocódigo no le implica mayor problema, pero si el diagramas de flujo no es adecuado entonces el pseudocódigo es incorrecto.

Para coadyuvar con este problema en la LIS, la UADY en su Modelo Educativo de Formación Integral (MEFI) menciona que uno de los ejes que orientan el trabajo académico es la innovación, concebida para solucionar situaciones

problemáticas y mejora continua de la educación, incorporando recursos/medios vanguardistas, promoviendo el uso intensivo de laboratorios y talleres para recrear la realidad así como también diversificando el uso de las TIC (UADY, 2012); cómo se puede apreciar la misma casa de estudios promueve entre sus docentes el hecho de buscar nuevas rutas y/o alternativas de acción que den apoyo a la enseñanza para ayudar al alumnado en su proceso escolar, hecho que brinda una puerta abierta para encontrar soluciones a problemas como el que se describe.

Para dar sustento a la problemática planteada, existen estudios que se han llevado a cabo al respecto como el que se encuentra en Xinogalos et al. (2015), donde partiendo de los problemas existentes en principios de programación, investigaron las percepciones que un grupo de estudiantes de pregrado inscritos a un curso de entornos y técnicas de programación tenían respecto de cinco entornos que son ampliamente utilizados en el ámbito educativo, de igual modo el estudio también incluyó la ubicación de las características que a su juicio debe tener cualquier entorno introductorio. Los programas investigados fueron: *BlueJ* (desarrollado por el equipo de BlueJ en la Universidad de Monash y la Universidad de Southern Denmark, liderado por Michael Kölling), *ObjectKarel* (desarrollado por Stelios Xinogalos, Maya Satratzemi y Vassilios Dagdilelis, de la Universidad de Macedonia), *Scratch* (desarrollado por Lifelong Kindergarten Group en el MIT Media Lab, liderado por Mitch Resnick, John Maeda y Yasmin Kafai), *Alice* (desarrollado por investigadores de la Universidad Carnegie Mellon, entre los que destaca Randy Pausch) y *App inventor* (desarrollada inicialmente por Google, el Programa de Formación de Maestros Scheller y el MIT); después de su uso durante un semestre, los estudiantes evaluaron: el contexto pedagógico en el cual resultaron más adecuados, las ventajas y las desventajas de cada uno y las características que debe tener cada entorno dirigido a los principiantes para ayudarles en su proceso de aprendizaje (BlueJ, s.f.; Xinogalos, Satratzemi, & Dagdilelis, 2006; MIT Scratch Team, s.f.; Ramírez, Lucio, Garza, García, & Vargas, 2011; Fuentes A. , 2017).

El grupo de estudiantes que participó en la investigación contaba con experiencia en lenguajes y entornos de programación, así como también conocían de las dificultades de aprender a codificar, esto fue un aspecto muy importante porque permitió que los participantes se centraran en cuál de los ambientes estudiados sería mejor para novatos.

Entre las conclusiones consideradas de importancia en este contexto, se puede mencionar que en cuanto a la evaluación de la parte pedagógica se obtuvo que todos los ambientes en mayor o menor medida ayudaron a comprender o profundizar los temas; sin embargo, respecto al entorno que los alumnos consideran el más eficaz para una introducción a la programación, *Scratch* fue el más elegido para todos los niveles educativos. También encontraron que tanto *Scratch* como *App inventor* atraen a más estudiantes probablemente debido a que estos dos ambientes brindan al alumno la oportunidad de escribir programas relacionados con sus intereses como son los videojuegos.

Otro estudio que refleja la problemática de la retención de alumnos de nuevo ingreso a carreras de ciencias de la computación es el realizado por Rizvi, Humphries, Major, Jones y Lauzun (2011), su planteamiento propone impartir un curso (denominado CS0) antes del primer curso introductorio a la programación que se lleva en las carreras de ciencias de la computación (denominado CS1), para enfrentar los problemas de retención que tienen.

El curso CS0 contempla el uso de *Scratch* para mejorar la retención y rendimiento de los estudiantes y consistió en clases en salón y tareas de laboratorio para cubrir los temas, al final el alumno entregó un proyecto que incluía todo lo estudiado. Cabe destacar que los objetivos del curso CS0 son similares a nuestro entorno en la materia de algoritmia.

El grupo participante de este estudio fueron los alumnos que ingresaron a la universidad con escasos antecedentes matemáticos y que fueron obligados a cursar álgebra como su primer curso de matemáticas habiendo elegido ciencias de la computación como su especialidad, estos estudiantes fueron asignados a llevar el

nuevo curso CS0 y los resultados se compararon con los obtenidos por alumnos que fueron inscritos directamente al curso CS1 los cuales no se consideraron como estudiantes en riesgo.

Como instrumentos de evaluación se recopilaron las calificaciones de los grupos que cursaron CS0 y CS1, así como también las respuestas de una encuesta en línea que midió el compromiso de los participantes, la capacidad de programación y las matemáticas. Las escalas para cada una de estas categorías fueron adaptadas de varios instrumentos.

Rizvi et al. (2011) concluyen que los alumnos que están usando *Scratch* tienen un alto grado de autoeficacia respecto a sus habilidades de programación. Además, el rendimiento de los estudiantes que tomaron el curso CS0 y que después se inscribieron en CS1 supera el de los alumnos en años anteriores. Respecto a *Scratch* mencionan que es un entorno rico en el manejo de una diversidad de medios, libre de las complejidades de sintaxis y sirve de apoyo en la implementación de proyectos como los juegos y las animaciones.

Como se ha detallado en las investigaciones citadas y desde esta óptica se considera que usar *Scratch* como una TIC en algoritmia es una opción factible que permitirá construir un programa, animación o juego en 2D el cual servirá al alumno como referencia para entender los resultados de los problemas; esto le permitirá comprobar los efectos de las soluciones encontradas y tendrá un referente con el cual contrastará sus resultados. *Scratch* permite al estudiante el aprendizaje autónomo, dejándolo investigar e introducirse a la programación mediante juegos lo que le ayuda a madurar el proceso de análisis y razonamiento de la lógica (Juárez, López y Villareal, 2016; Pajares, 2014).

Por otro lado, es importante mencionar que a pesar de las bondades que se pueden encontrar en muchas investigaciones donde han empleado las TIC para solucionar problemas educativos, también resulta conveniente reconocer que no todo es bondadoso con su empleo. En Díaz (2013), se citan algunas desventajas

de su uso en el proceso de enseñanza y aprendizaje centrados en el profesor y en los alumnos, a continuación se enumeran las más relevantes.

Desventajas centradas en el profesor:

- Exige un perfeccionamiento constante, así como una inversión de tiempo y dinero.
- La existencia de mucha información demanda que los profesores dediquen tiempo para analizar la pertinencia de las TIC de interés al momento de tomar decisiones sobre su uso.
- En ocasiones, el método clásico exige menos compromiso, tiempo y esfuerzo.
- Puede suceder que los profesores se vuelvan dependientes de la tecnología y si algo falla no pueden desarrollar una clase.

Adicionalmente, más que una desventaja es importante considerar que en un mundo centrado en las TIC, como docentes hay que ser competentes digitales, a este respecto el proyecto europeo *DIGCOMP: A Framework for Developing and Understanding Digital Competence in Europe* (2013, citado en Viñals y Cuenca, 2016) menciona que ser competente digital significa adoptar 5 dimensiones:

- **Información:** Conlleva a identificar, localizar, recuperar, almacenar, organizar y analizar los contenidos digitales, evaluando su finalidad y relevancia.
- **Comunicación:** Implica comunicar en entornos digitales, compartir recursos en línea, conectar y colaborar con herramientas digitales, interactuar y participar en comunidades y redes, así como también hace referencia a la conciencia e interculturalidad.
- **Creación de contenido:** Involucra crear y editar, integrar y reelaborar conocimientos, hacer producciones artísticas, multimedia y programación informática, aplicar derechos de propiedad intelectual y licencias de uso.

- **Seguridad:** Implica protección personal, protección de datos, protección de la identidad digital, uso seguro y sostenible.
- **Resolución de problemas:** Incluye identificar necesidades y recursos digitales, tomar decisiones a la hora de elegir herramientas apropiadas, resolver problemas con medios digitales y técnicos, uso creativo de la tecnología y actualización de la competencia propia y la de otros.

Desventajas centradas en el alumno:

- Ya que se tiene acceso a mucha información, es muy fácil que los estudiantes se distraigan en la búsqueda usando diversas aplicaciones y en algunas ocasiones lo obtenido sea incorrecto.
- Puede dar lugar a que los estudiantes se aprovechen del aprendizaje colaborativo y no trabajen, también puede haber suplantaciones de identidad.
- Se tiene acceso a muchas distracciones, ya que se sabe que cuando un estudiante está trabajando al mismo tiempo está conectado a diferentes redes sociales, por ejemplo.
- El acceder a mucha información puede llevar al plagio, es decir, el uso común de cortar y pegar lo cual puede ocasionar problemas graves.

Complementado lo anterior, Plaza (2018) cita algunas influencias negativas de las TIC desde el mismo punto de vista de los educandos:

- Destaca que las TIC pueden ser un distractor de las cosas importantes ocasionando pérdida de tiempo dedicada al estudio y poca concentración.
- Reconoce que se puede llegar a una dependencia y uso excesivo de las TIC.
- Uso solo por motivos de ocio.

- La inseguridad que provoca la apertura de internet es un punto que bien vale la pena considerar.

Estos aspectos negativos del uso de TIC son algunos a los que los docentes y los estudiantes tienen que enfrentarse si deciden incluirlos en las aulas de clase, sin embargo, hay que destacar que los beneficios citados anteriormente son satisfactorios y pueden sobrepasar estas desventajas o bien se pueden buscar maneras de cómo reducir su impacto.

La transformación que han sufrido las TIC hace que hoy puedan ser instrumentos educativos capaces de mejorar la calidad educativa del estudiante, revolucionando la forma en que se obtiene, se maneja y se interpreta la información, (Aguilar, 2012), lo cual hasta hace unos años atrás era impensable. La diversidad de escenarios, contextos y tendencias en la educación actualmente impone nuevos roles al proceso formativo, implica retos para el profesional del futuro, para las instituciones y para los agentes encargados de la formación (Prieto et al., 2011, citado en Hernández, 2017). Hoy más que nunca ante la pandemia que se vive en todo el mundo esto es una realidad.

En función de lo que se ha venido planteando el interés de esta propuesta se centró en crear una metodología de aprendizaje que incorporó el lenguaje de programación *Scratch* a la asignatura de algoritmia de la LIS que se imparte en la UMT la cual sirvió para mejorar la comprensión de la solución de problemas computacionales por medio de una TIC haciendo que el alumno logre palpar los efectos de los algoritmos con el propósito de elevar el rendimiento académico. Para lograr lo anterior se volvió relevante examinar el nivel de conocimientos que respecto al área de algoritmia tenían los alumnos de nuevo ingreso, definir ejercicios y actividades a desarrollar en *Scratch* enfocadas en sus principales problemas (los cuales se volvieron cruciales de detectar), aplicar el material generado en el laboratorio y por último evaluar los resultados. El interés de la investigación va de la

mano con el de la UADY que es la institución contexto y con las tendencias de inclusión de TIC en los países de la región.

Esta propuesta deriva de la investigación basada en diseño (IBD) siguiendo las fases citadas en De Benito y Salinas (2016), siendo estas el análisis de la situación, diseño de soluciones, implementación de solución y evaluación de resultados.

Dirección General de Bibliotecas UADY

2. LAS TIC EN LA EDUCACIÓN, LA ALGORITMIA Y SCRATCH

En este capítulo se describen las teorías que sustentan la presente investigación, también se detallan las investigaciones previas y los antecedentes en general que se consideraron válidos para la correcta delimitación.

Se exponen tres temas de importancia en este entorno, en primer lugar se hace una descripción de las TIC en la educación, la implementación de herramientas y la perspectiva del docente en este ámbito; posteriormente se enfoca a la algoritmia, dando un encuadre de la materia en la que se desarrolló el proceso investigativo, así como también su situación en instituciones de nivel superior, los problemas más comunes a los que se enfrenta el educando y algunas herramientas de las que se vale para afrontarlos; por último, se centra en el lenguaje de programación *Scratch*, que es la herramienta TIC que se empleó en esta investigación, se hace una descripción de su uso a nivel educación en general y en la universidad de modo específico.

2.1 Las TIC en la educación

La llegada de la economía del conocimiento y de la competencia económica global plantea la necesidad de dar mayor prioridad a la calidad de la educación, al aprendizaje a lo largo de la vida y a la igualdad de oportunidades para todos. Los formuladores de políticas educativas han adoptado una postura común en el sentido de que un mejor acceso a las TIC en la educación brinda a las personas una mejor oportunidad de competir en la economía global (UNESCO, 2013b).

El uso de las TIC se detonó en forma exponencial en los últimos años y se ha consolidado de tal manera que se han vuelto indispensables en la vida cotidiana (Calvo, 2011). La razón de lo anterior es porque las TIC permiten generar, almacenar, transmitir, recuperar y procesar información en dimensiones espaciales y temporales hasta ahora inéditas. No cabe duda, de que es urgente incorporar las TIC a la educación formal, dado que es el modo más rápido, económico y extendido

de reducir la brecha digital entre un país y otro e inclusive dentro de ellos (Sunkel, Trucco, & Espejo, 2014).

Sin duda, el ámbito educativo ha sido tocado por las TIC y partiendo de que el estudiante actual genera su propio conocimiento, se identifica por ser más interactivo, crítico, espontáneo y su rol en el aula no es de un simple espectador sino de un estudiante activo y decidido, entonces se necesitan cambios (Prieto, Mijares, & Llorent, 2014), en este sentido Edel, Aguirre y Balderrama (2016), mencionan que la enseñanza tradicional requiere transformarse y reconocer el potencial didáctico de la tecnología para la formación académica, los docentes deben estimular las competencias digitales de los alumnos, que conducen a la usabilidad de TIC y gestión del conocimiento; dado que el empleo de éstas representa la oportunidad para incrementar:

- El nivel de desarrollo personal de los estudiantes.
- El crecimiento en su cultura computacional y algorítmica.
- El desarrollo de su razonamiento espacial y cultura gráfica.
- La expansión del espectro cognitivo a través de estimular sus procesos de pensamiento.
- La percepción, comprensión y representación, entre otros.

Del mismo modo, desde hace más de una década, en toda América Latina se ha venido produciendo una acelerada irrupción de las TIC en el ámbito educativo, a partir de muchas iniciativas que procuran dar respuesta a la exigencia de generar condiciones para garantizar más y mejor educación para la población en todas las áreas de las ciencias (UNESCO, 2014).

La introducción de las TIC al ámbito educativo aporta una nueva dimensión, la posibilidad de compartir, de transferir información y conocimientos básicos,

facilitan el acceder a nuevas fuentes de saber y aumentan la capacidad de aprender (Delgado, Arrieta, & Riveros, 2009).

Por otro lado, las TIC en las aulas ponen en evidencia la necesidad de una nueva definición de roles, especialmente, para los alumnos ya que deben adquirir mayor autonomía y responsabilidad en el proceso de aprendizaje y para los docentes a los que las TIC los obliga a salir de su desempeño tradicional como única fuente de conocimiento (UNESCO, 2013a); el clásico yo enseño, tú aprendes comienza a ceder espacios a un nuevo paradigma, las nuevas tecnologías, donde específicamente los sistemas telemáticos son medios interesantes para introducir pedagogías alternativas y potenciar cambios en las estructuras educativas. Los educadores, ante la acelerada producción de conocimiento que se vive, deben reflexionar sobre el quehacer académico y sobre la responsabilidad que tienen con los estudiantes de hoy y de mañana si se quiere dotarlos de las capacidades necesarias para enfrentarse con éxito a la vida.

Es un hecho que el aporte de las TIC a la educación y a la sociedad como tal, es la flexibilidad y la adaptación a un entorno cada vez más cambiante, al principio el trabajo fue el más afectado en este proceso, sin embargo, con el paso del tiempo se ha evidenciado que la sociedad depende de un enfoque tecnológico que lo ayude a construir y adquirir conocimiento (Hernández, 2017).

Según Ávila Díaz (2013), las TIC cubren diversas áreas incluyendo el sector educativo al emplear herramientas informáticas como principal motivante para el aprendizaje de los alumnos e incluso de los docentes. En su trabajo, Duart (2008) concluye que el potencial de la tecnología en los procesos de enseñanza y aprendizaje en educación superior es elevado y contribuye al valor global de la institución que la use de forma claramente destacada.

Por su parte, Villarruel (2012) señala que en la última década las políticas de educación superior en América Latina han visto como una necesidad insoslayable incorporar tecnología e iniciativas que fomenten el uso de TIC por parte de los docentes para dar apoyo al alumno. En las políticas TIC en los sistemas educativos

de América Latina de la UNESCO (2014), es palpable cómo las tecnologías se han posicionado en nuestras vidas y muestran la importancia de éstas en los procesos educativos.

Es crucial tener en cuenta quiénes son estos nuevos estudiantes para avanzar hacia una escuela que garantice plenamente el derecho a la educación y que esté a la vanguardia con los desarrollos tecnológicos dotando a los educandos de la capacidad para enfrentarse con éxito a la vida.

Las TIC se han situado como una herramienta imprescindible en nuestras vidas no siendo exclusivas de un área en particular y lo han hecho aportando procesos y procedimientos. De acuerdo con Quesada, Sobrado y Solís (2015), las TIC se revisten de un valor especial que potencia el desarrollo de las capacidades cognitivas de los estudiantes y de la sociedad en general, con el fin de que las personas puedan:

- Exponer sus criterios.
- Se les facilite la resolución de los problemas.
- Produzcan ideas.
- Innoven mecanismos.
- Trabajen colaborativamente.
- Aumenten el grado de comunicación.
- Participen en las áreas locales y globales.

2.1.1 Conceptualización

Existen múltiples definiciones de TIC que van de acuerdo con el entorno y la visión que los autores han tenido en determinado momento; en esta investigación se presentan algunas que se consideran relevantes en un contexto general y en el ámbito educativo.

En líneas coloquiales Cabero (1998) menciona que las nuevas tecnologías de la información y comunicación son las que giran en torno a tres medios básicos: la informática, la microelectrónica y las telecomunicaciones; pero giran no sólo de forma aislada, sino lo que es más significativo de manera interactiva e interconectadas, lo que permite conseguir nuevas realidades comunicativas. De igual modo, cita una serie de características que poseen como son:

- La inmaterialidad.
- La interactividad e interconexión.
- La instantaneidad.
- Elevados parámetros de calidad de imagen y sonido.
- La digitalización.
- La influencia sobre los procesos más que sobre los productos.
- La penetración en todos los sectores: culturales, económicos, educativos e industriales, entre otros.
- La innovación.
- La tendencia hacia la automatización.
- La diversidad.

Por su parte, para Marqués (2000) el concepto de tecnologías de la información y comunicación viene formado de tres importantes palabras las cuales son tecnología, información y comunicación y las define como sigue:

- **Tecnología:** Al referirse a ella habla de la aplicación de los conocimientos científicos para facilitar la realización de las actividades humanas, supone la creación de productos, instrumentos, lenguajes y métodos al servicio de las personas.

- **Información:** La cual resulta fundamental para las personas, ya que a partir del proceso cognitivo de la información que se obtiene continuamente por medio de los sentidos, los seres humanos pueden ir tomando decisiones que dan lugar a acciones.
- **Comunicación:** Utilizada para la transmisión de mensajes entre personas ya que como seres sociales además de recibir información necesitamos comunicarnos, expresar nuestros pensamientos y sentimientos, entre otros.

Al unir tecnología, información y comunicación se hace referencia al conjunto de avances que proporciona la informática, las telecomunicaciones y los medios audiovisuales que comprenden los desarrollos relacionados con las computadoras, Internet, la telefonía, los medios de comunicación masiva, las aplicaciones multimedia y la realidad virtual. Concluye diciendo que estas TIC básicamente proporcionan información, herramientas para su proceso y canales de comunicación.

Asimismo, Ávila Díaz (2013) después de efectuar una amplia lectura de los diferentes conceptos dados por diversos autores, presenta la siguiente definición de las TIC: es el conjunto de herramientas, soportes y canales desarrollados y sustentados por las tecnologías (telecomunicaciones, informática, programas, computadoras e internet) que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de información, en forma de voz, imágenes y datos, contenidos en señales de naturaleza acústica, óptica o electromagnética a fin de mejorar la calidad de vida de las personas.

Por último, aterrizando los conceptos al área educativa, Villegas (2015) cita que las TIC son herramientas capaces de permitir la mejora en el ámbito académico tanto en la parte pedagógica como administrativa; estas tecnologías mediante un enfoque constructivista ayudan al educando en su proceso de generación del conocimiento, de igual modo pueden crear competencias que se demandan en la

sociedad actual, sin embargo, se deben de usar con una estricta planeación acorde a los conceptos temáticos y al nivel educativo. A su vez, Prieto et al. (2011, citado en Hernández, 2017) señalan que si se añaden herramientas TIC a los modelos pedagógicos, éstas pueden convertirse en recursos valiosos para el aprendizaje logrando formar estudiantes con competencias personales y profesionales idóneas para el desarrollo de un país.

2.1.2 Aplicación de herramientas TIC

Para integrar eficazmente las TIC a la enseñanza y el aprendizaje se precisa de una redefinición de la función de los docentes en la planificación y aplicación de estas tecnologías, a fin de cambiar y mejorar la formación de los estudiantes. Los sistemas educativos deben actualizar y mejorar regularmente la preparación y la formación profesional de los educadores y velar por que todos los profesores puedan sacar partido de la tecnología con fines educativos (UNESCO, 2019), del mismo modo, ya ha sido citada la importancia de las competencias digitales que los maestros deben de tener las cuales son de vital importancia en este contexto.

Las TIC proporcionan una nueva visión educativa en el mundo entero, sin embargo, la presencia de éstas en las aulas ha mostrado ser insuficiente para la transformación de las prácticas pedagógicas de los maestros, el hecho de disponer de una computadora o un pizarrón electrónico puede contribuir a ello pero no es suficiente para transformar el quehacer en el aula, es decir, su mera incorporación a los procesos de formación no garantiza por ningún motivo su efectividad (Chocó, s.f.), la clave está en la forma en que utilicen los docentes estas TIC para contribuir al logro de los aprendizajes esperados (Santiago, Caballero, Gómez, & Domínguez, 2013), su uso requiere de análisis, reflexión y trabajo para obtener buenos resultados (Garza & Solís, 2012).

Para apoyar lo anterior, varios investigadores han desarrollado propuestas que ayudan a los educadores a incorporar adecuadamente las TIC a sus aulas de

clase, como ejemplo se cita a Delgado et al. (2009) los cuales plantean una propuesta del uso óptimo de las TIC y exponen que su empleo permite crear ambientes motivantes y retadores para adquirir conocimientos, por lo que es importante que los maestros conozcan su influencia, sus consecuencias e implicaciones ante la situación educativa, para usarlas de forma óptima y eficiente; hay que estar conscientes de que las TIC no son la solución a todos los problemas educativos, por lo que es importante tomar en consideración diversos factores que intervienen en el proceso.

De la mano con lo expuesto, es importante mencionar que el uso de las TIC si puede ayudar al docente a mejorar la calidad de la enseñanza y a aprovechar mejor el tiempo libre de los alumnos por lo que es importante que exista una oportunidad de formación permanente y actualizada, para que tengan las bases para despertar valores en los jóvenes para el uso adecuado de las TIC de manera responsable y de modo crítico para que éstos puedan discernir y tomar buenas decisiones sobre lo que buscan y lo que encuentran.

Por otro lado, atendiendo al problema de interés, es sabido que el aprendizaje de conceptos complejos por parte del alumno puede ser una tarea difícil para la mayoría de los educandos y generarles un deficiente rendimiento académico, sin embargo, siguiendo el llamado de la incorporación de las TIC en los procesos educativos, muchos docentes han desarrollado soluciones a las situaciones problemáticas que se les han presentado mediante la inclusión de éstas, en los siguientes párrafos se describen algunos ejemplos.

Según Huertas y Pantoja (2016), aplicar TIC favorece el aprendizaje de los alumnos, aumenta su motivación, interés y creatividad, potencia el trabajo en grupo y la capacidad para resolver problemas reforzando su autoestima; esto lo determinaron empleando las TIC en la asignatura de tecnología en secundaria y comprobaron que el rendimiento académico y la motivación del alumnado mejoraron significativamente.

En Mirete y García (2014), se concluye que existe una relación positiva con los resultados académicos obtenidos y el uso de las TIC, esto fue logrado mediante la inclusión de webs didácticas. Los alumnos señalaron su utilidad en el proceso de aprendizaje e identificaron que no es solo el uso de las webs la causa de la calificación final, sino también los enfoques de aprendizaje de los estudiantes, expectativas y motivación que consigue el recurso empleado, Mirete y García remarcan que la inclusión de las TIC requiere de una rigurosa planificación, atendiendo a las características, usuarios y objetivos del proceso educativo donde se insertarán, llevando a seleccionar la tecnología que más se adecúe; es importante mencionar que este aspecto va de la mano con lo que anteriormente se citaba en Delgado et al. (2009).

Autores como Botello y Guerrero (2014), muestran que usar TIC en el aprendizaje escolar mediante actividades, afecta positivamente el desempeño académico; en Botello y López (2014) usaron las TIC como una herramienta para lograr mayores puntajes en una prueba de lectura; para Dávila (2017) en la clase de matemáticas, el uso de las TIC motiva el aprendizaje al contextualizar los conocimientos mejor que con el modo tradicional y para García, Revuelta y Esnaola (2017) los alumnos ejercitaron múltiples habilidades cognitivas para resolver problemas teniendo mejoras significativas mediante la programación de computadoras.

En las investigaciones analizadas de los distintos autores consultados, los alumnos han mejorado su rendimiento académico, el cual es probablemente una de las dimensiones más importantes en el proceso de enseñanza-aprendizaje y por esa razón es que se ataca desde distintos frentes. De acuerdo con Edel (2003), cuando se trata de evaluar el rendimiento académico y cómo mejorarlo, se analizan en mayor o menor grado los factores que influyen en él, como los socioeconómicos, la amplitud de los programas, las metodologías de enseñanza, los conceptos previos del alumnado, el nivel de pensamiento formal de los mismos, entre otros; se puede decir que el rendimiento académico es un fenómeno multifactorial. Por su

parte, Fuentes (2004) considera que el rendimiento académico es la correspondencia entre la conducta del alumno y los comportamientos especificados que se deben aprender en su momento escolar, en esta conceptualización encuentra una relación tríadica: lo que se pretende que se aprenda, lo que facilita el aprendizaje y lo que se aprende.

Atendiendo a todo lo anteriormente expuesto, se concluye que la implementación de herramientas TIC durante el proceso de enseñanza y aprendizaje es uno de varios factores que resultan determinantes ya que pueden modificar positivamente el rendimiento académico de los alumnos.

2.1.3 Perspectiva del docente en la incorporación de TIC

Son muchos los factores que intervienen para poder realizar una inserción exitosa de las TIC en la educación, pero sin duda alguna uno de los agentes más significativos e importantes es el docente, pues en él descansa en gran medida, la integración de éstas en los procesos educativos (Cervantes, 2015).

El impacto de las TIC dentro de la sociedad del conocimiento ha tenido un papel decisivo en el cambio del dinamismo social, cultural y económico, respecto a forma y contenido, inclusive el efecto ha sido masivo y multiplicador, de tal manera que el sentido del conocimiento ha calado en la sociedad en general y una de las grandes implicancias y modificaciones es la educación, ya que se ha planteado un entorno distinto al que se vivía antes del siglo XX (Ávila Díaz, 2013; Hernández, 2017). En este sentido, la tabla 2.1 presenta el pensamiento de algunos autores relacionado con su forma de percibir la incorporación de las TIC al ámbito educativo.

Tabla 2.1 Perspectivas docentes sobre TIC.

| Autor | Aporte |
|-------------------------------|--|
| Salinas, 2004 | Es importante tener en cuenta que la institución educativa y el profesor dejan de ser fuentes de todo conocimiento, el docente debe actuar como guía de los alumnos convirtiéndose en gestor de los recursos de aprendizaje y acentuando su rol de orientador, facilitándoles el uso de los recursos y las herramientas que necesitan para explorar y elaborar nuevos conocimientos y destrezas. |
| Khvilon y Patru, 2004 | El desarrollo profesional que implica incorporar las TIC a la enseñanza y el aprendizaje debe verse como un proceso continuo, en el entendido de que las TIC evolucionan de la mano con la tecnología imperante; por esta razón los docentes deben actualizar sus habilidades y conocimientos; actividades que deben ir de la mano con el plan de estudios y con los medios tecnológicos con los que se cuenta. Se requiere tener docentes comprometidos que se desarrollen y estén acompañados por la organización y el compromiso de las escuelas, centros de formación y universidades. |
| Díaz-Barriga, 2009 | No se puede concebir una transformación de sistemas y procesos educativos sin la vinculación de la educación misma con las TIC, desde esta perspectiva cuesta trabajo pensar en alguna innovación educativa que no esté ligada a los desarrollos tecnológicos, sin embargo, no es en las TIC en sí, sino en las actividades que se llevan a cabo con posibilidades de comunicación, intercambio, acceso y procesamiento de la información que ofrecen, donde se debe buscar el medio para comprender y valorar el alcance de su impacto en la educación escolar. |
| Campos, Brenes y Solano, 2010 | <p>Destacan un perfil de competencias que debería tener el personal académico de educación superior para el desarrollo de programas en línea.</p> <ul style="list-style-type: none"> • Competencias tecnológicas, son las habilidades para la aplicación de herramientas tecnológicas asociadas con el proceso educativo. • Competencias investigativas, habilidades, conocimientos y actitudes que le permiten la problematización de la realidad educativa y la búsqueda sistemática de respuestas. • Competencias pedagógicas, implica el dominio de teorías de enseñanza y aprendizaje y de diseño instruccional, que permiten el planteamiento, desarrollo y evaluación de propuestas pedagógicas efectivas. • Competencias de liderazgo académico, saberes, actitudes y prácticas orientadas hacia la búsqueda permanente de la excelencia académica de sí mismo, de sus estudiantes y de la organización. • Competencias sociales, crear estrategias que favorezcan la comunicación con los estudiantes manteniendo altos niveles de motivación en el aprendizaje. |

| | |
|---|---|
| Aguilar, 2012 | El reto docente consiste en tener la competencia adecuada en el manejo y uso pedagógico de las TIC, ser capaz de transformar y/o abandonar las metodologías tradicionales por estrategias innovadoras que tiendan a la construcción de aprendizajes. Los sistemas tradicionales entran en conflicto con las nuevas tendencias en donde es el propio estudiante quien de forma progresiva, desarrolla estrategias para planear, gestionar los recursos de las TIC y valorar su propio aprendizaje contando con el docente como guía. |
| Fuentes y Careaga, 2012 | En la universidad, independientemente de la materia, el profesor debe considerar espacios que le permitan a los estudiantes practicar y acercar sus habilidades tecnológicas a su quehacer académico, de igual modo debe centrarse en preguntar, orientar y guiar a los estudiantes garantizando el rigor y el sentido de lograr resultados de calidad en sus aprendizajes. |
| Cervantes, 2015 | Se observa de modo general que los educadores muestran una actitud favorable ante el cambio tecnológico a pesar de que en algunos casos no se tiene un buen dominio de las tecnologías, los docentes reconocen que las TIC pueden ser una herramienta de ayuda para la mejora educativa e incluso mantienen buenas expectativas de éstas en los procesos de enseñanza-aprendizaje; sin embargo, esta buena disposición dista mucho de lo que en la práctica concreta se realiza para poder introducir tecnología al aula. |
| Gairín, Castro, Silva, y Mercader, 2016 | Para el profesorado, las TIC motivan y promueven la autonomía del estudiante y lo sitúan como protagonista de su formación, refuerzan el rol del profesor como facilitador e impulsan la innovación. Paralelamente, hay profesores que valoran su impacto motivador, la mejor calidad de la formación, la aproximación a la realidad sociocultural y la facilitación de tareas administrativas y de evaluación. |
| Hernández, 2017 | El proceso de enseñanza-aprendizaje en el aula haciendo uso de las TIC, requiere de un conjunto de competencias que el maestro debe adquirir con la lógica de sumar una metodología capaz de aprovechar las herramientas tecnológicas, donde la capacitación deberá considerarse una de las primeras opciones antes de afrontar nuevos retos educativos. |

Fuente: Salinas (2004), Khvilon y Patru (2004), Díaz-Barriga (2009), Campos, Brenés y Solano (2010), Aguilar (2012), Fuentes y Careaga (2012), Cervantes (2015), Gairín, Castro, Silva y Mercader (2016) y Hernández (2017).

En síntesis, se requieren nuevos escenarios educativos con propuestas innovadoras que ayuden al estudiante a tener un aprendizaje significativo; se habla de la integración de las TIC a la educación cuando el sistema educativo puede brindar ese aprendizaje significativo producto de vivencias y de trabajo en modo reflexivo. Estas condiciones han llevado a pensar en la necesidad de una adecuada

formación y actualización profesional en TIC que permita a los docentes afrontar estos retos con el desarrollo de prácticas innovadoras, el manejo de herramientas que puedan fortalecer su labor en el aula y el desarrollo de habilidades y competencias tecnológicas (Cervantes, 2015). Paralelamente, la institución educativa receptora debe proporcionar los recursos que se necesiten para el logro de los objetivos; en el entorno de esta investigación la UADY se encuentra ampliamente comprometida como bien se aprecia en su Modelo Educativo de Formación Integral (UADY, 2012) dado que en su eje de innovación, habla sobre la planeación deliberada y sistemática de nuevas propuestas para la solución de situaciones problemáticas para la mejora continua de la práctica educativa lo cual implica un cambio en el contexto y en la misma realidad mediante la incorporación de recursos y medios educativos vanguardistas.

2.2 Algoritmia

En las carreras relacionadas con la computación y/o informática suele existir una materia de introducción a la programación en los primeros semestres que tiene como objetivo darle al alumno los conceptos básicos de la misma. El inicio de estudiantes a esta área es un tema que preocupa y ocupa a los docentes en muchos centros escolares ya que su correcto entendimiento será necesario para que después éste se pueda adentrar a la programación de computadoras de modo adecuado y sus resultados se vean reflejados en asignaturas relacionadas. En el caso que se presenta, la materia de algoritmia se imparte en el primer semestre de la licenciatura en ingeniería de software y es una asignatura introductoria a los principios de programación.

Algunos de los problemas que presentan los alumnos universitarios en los primeros cursos que llevan de su especialidad son: el completo desconocimiento por la materia y la carencia de hábitos de estudio. De la mano con esto, en el caso particular que se plantea dado el contexto en que se imparte la materia de algoritmia suele suceder que los alumnos se inscriben a ingeniería de software porque no

tuvieron oportunidad de ingresar en alguna otra carrera y esta situación particular que se presenta dificulta aún más el proceso de aprendizaje.

Por otro lado, la materia algorítmica suele presentarse como una asignatura difícil, enseñar a un alumno los principios básicos de programación es complejo, en esencia porque es una habilidad abstracta, suele ser difícil de entender en los primeros cursos y lo que produce no es un referente físico que le permita ver al alumno el producto final de su trabajo, esto acorde con que muchas veces se sigue enseñando con el método tradicional utilizando papel y lápiz. Este punto, sumado con lo descrito en el párrafo anterior provoca un bajo rendimiento académico lo cual se ve reflejado en las calificaciones finales que obtienen los alumnos en la materia.

En Sleeman (1986, citado en Alania, Diez y Pinglo, 2012), se menciona que el primer objetivo de un programador es descomponer la tarea para especificar un plan detallado y realizable, es decir un algoritmo que solucione el problema, el segundo es implementar este plan en un lenguaje de programación y el tercero es depurar el programa resultante (este proceso puede ser tan complejo como los anteriores). En la materia de algorítmica se cumple con el primer objetivo únicamente, es decir, se trata de construir el algoritmo sin que este se implemente y depure, de aquí que el alumno no cuenta con un referente concreto que le permita ver la ejecución de su trabajo ya que su producto final queda plasmado solamente a papel y lápiz y esto no les representa mucho.

2.2.1 Conceptualización

La palabra algoritmo tiene su origen en el nombre de un famoso matemático y erudito árabe del siglo IX, Al-Khwarizmi (Juarismi o Khorezmi), a quien también se le deben las palabras guarismo y álgebra (Puig, 2008). La misma se puede definir como un conjunto de pasos, procedimientos o acciones que permiten alcanzar un resultado o resolver un problema, en otras palabras, es un grupo de reglas para

abordar una cierta clase de situación o una forma de describir su solución (Joyanes Aguilar, 2003; Cairó, 2005).

La definición anterior la complementan Pinales y Velázquez (2014) y Sleeman (1986, citado en Alania et al., 2012), diciendo que el resultado que se busca es mediante el uso de una computadora, en este sentido el algoritmo debe tener como característica final la posibilidad de ser transcrito fácilmente a un lenguaje de programación para lo cual se usarían herramientas de programación.

Aterrizando lo anterior en el marco de esta investigación, la algoritmia implica la capacidad de resolver problemas haciendo uso de los conceptos fundamentales de los principios de programación, también es importante considerar lo que diversos estudios demuestran al decir que las personas entrenadas en esta disciplina desarrollan distintas habilidades que le ayudan a resolver las dificultades de su vida diaria, además de desarrollar pensamiento crítico y la capacidad de análisis (Rodríguez, 2020).

Muchas veces las personas aplican algoritmos en su vida cotidiana para resolver dificultades de manera inadvertida, inconsciente o automáticamente, esto generalmente se produce cuando la situación que se tiene en mente se ha resuelto con anterioridad un gran número de veces, por ejemplo: cepillarse los dientes, vestirse, andar en bicicleta, ir a la escuela desde la casa, prepararse un café, etc. Entonces, se puede decir que en la vida diaria se están aplicando algoritmos continuamente sin darnos cuenta.

En algoritmia, no existen reglas específicas para resolver un problema, no obstante, se han desarrollado técnicas y herramientas que permiten estructurar el razonamiento, por lo que es conveniente seguir las etapas de resolución las cuales se pueden definir como: el análisis, el diseño y las pruebas (figura 2.1), Cairó (2005) las define de la siguiente manera:

- **Análisis:** Se determina cuál es exactamente el problema por resolver, es un examen a fondo sobre qué datos forman la entrada del algoritmo, cuáles deberán

obtenerse como salida, así como el proceso de transformar esa entrada en la salida esperada.

- **Diseño:** Es la elaboración o construcción del algoritmo, usando los datos obtenidos en el análisis, por medio de una herramienta, por ejemplo, un diagrama de flujo (DF).
- **Pruebas:** Comprobación del resultado, se observa si el algoritmo generado obtiene la salida esperada para todas las entradas.

Figura 2.1 Etapas en la resolución de un problema.



Fuente: Elaboración propia (2019).

Como se citó anteriormente, un algoritmo debe ser fácil de poder transcribirlo a un lenguaje de programación, por lo que es importante que cuente para tal efecto con las siguientes características obligatoriamente (Joyanes Aguilar, 2003):

- **Preciso:** Debe indicar el orden en el cual se realizarán cada uno de los pasos que conducen a la solución.
- **Definido:** Esto implica que el resultado nunca debe cambiar, bajo las mismas condiciones este siempre será el mismo.
- **Finito:** No se debe caer en repeticiones de procesos de manera innecesaria y obligatoriamente terminará en algún momento tras un número determinado de pasos.

En consecuencia y a modo de resumen, el algoritmo es una serie de operaciones detalladas y no ambiguas que se ejecutan paso a paso y conducen a la resolución de un problema, las cuales se representan mediante una herramienta o técnica (Pinales y Velázquez, 2014). Las herramientas o técnicas comúnmente usadas en la etapa de diseño y que se emplean en la materia de algoritmia que se imparte en la LIS son los diagramas de flujo y el pseudocódigo.

Por otro lado, existe un conjunto de características aconsejables que los algoritmos deben cumplir y se definen como sigue (Desarrollo de Lógica de Programación, s.f.):

- **Validez:** Es válido si carece de errores, hay que notar que puede resolver el problema para el que se planteó y sin embargo, no ser válido debido a que tiene errores.
- **Eficiencia:** Es eficiente si obtiene una solución en poco tiempo y no lo es si es lento en obtener el resultado.
- **Óptimo:** Es óptimo si es el más eficiente posible y no contiene errores, la búsqueda de este algoritmo es el objetivo prioritario del programador y no siempre se puede garantizar que el resultado hallado cumpla con esta característica.

2.2.2 Situación actual de la algoritmia en instituciones de nivel superior

En esta sección, se exponen algunas situaciones que se han presentado en el nivel superior relacionadas con la implementación de asignaturas introductorias a los principios de programación.

Según Muñoz et al. (2015b) en la carrera de ingeniería en informática de la Universidad de Valparaíso en Chile, los cursos de pregrado relacionados con fundamentos de programación de computadoras tienen frecuentemente altas tasas

de reprobación. Los estudiantes de esta carrera reciben en su formación inicial tres cursos de programación:

- Fundamentos de programación, que aborda tipos de datos, estructuras de control y la definición y uso de algoritmos para la resolución de problemas.
- Previa aprobación de fundamentos de programación, los alumnos cursan programación I, en donde se estudia el paradigma imperativo de programación usando el lenguaje C.
- Por último, cursan programación II donde se estudia el paradigma orientado a objetos y suele pasar que a esta materia llega un número considerablemente menor de estudiantes de los que inician fundamentos de programación (cerca del 30%), esto debido a la alta tasa de deserción.

Por su parte Rosanigo y Paur (2006) trabajando en la materia de algorítmica y programación que se imparte en la facultad de ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco en la ciudad de Comodoro Rivadavia en Argentina determinaron que en ella confluyen diferentes situaciones que dificultan el aprendizaje y sumado a las inherentes de la asignatura provocan alta deserción y bajo rendimiento que se refleja en una relación aprobados/inscritos muy baja. Algorítmica y programación es una materia de vital importancia en la formación del alumno, el éxito o fracaso en ella influye decisivamente en las restantes ya que no quedan casi posibilidades de seguir avanzando en la carrera si no se logran las competencias requeridas.

Para apoyar lo anterior, es imperativo buscar las mejores estrategias que ayuden a efectivizar el proceso de enseñanza-aprendizaje y la calidad de los resultados. En este sentido, los docentes del centro de estudios mencionado a lo largo de muchos años han observado, investigado, propuesto y probado estrategias que den mejor resultado para la enseñanza-aprendizaje de la programación, estas estrategias abarcan desde cambios en la forma didáctica de impartir los

conocimientos hasta modificaciones en la distribución de las horas de clase y han implementado una serie de medidas que les dieron buenos resultados.

Lage y Cataldi (2019) trabajando en la carrera de ingeniería informática de la Universidad de Buenos Aires, mencionan que los alumnos que llegan a un primer curso de algoritmia e integran el mismo normalmente son un grupo heterogéneo en cuanto a conocimientos previos, lo cual queda evidenciado en las evaluaciones diagnósticas iniciales que consisten en la resolución de ejercicios orientados al manejo de proporciones, relaciones y uso de variables. Los resultados dan información acerca del estadio evolutivo general de los alumnos, desde una óptica piagetiana y permiten tener una referencia para saber cuál ha sido el rendimiento de los estudiantes respecto del estado inicial.

Lage y Cataldi (2019) pusieron en marcha una propuesta en función de los problemas detectados en los alumnos evidenciado por los fracasos en los exámenes en los últimos cuatrimestres, esta propuesta se realizó a través de foros de discusión con seguimiento de los alumnos en la realización de tareas cooperativas y colaborativas, con la intención de que logren el andamiaje necesario en la realización de sus trabajos prácticos. Se determinó que los alumnos sostienen que el trabajo en grupo les facilita los aprendizajes y que la metodología empleada se podría mejorar con sesiones de chat. Este modo de compartir inquietudes comunes y de resolución entre todos les ha dado una visión más amplia en cuanto al modo de encarar la resolución de los problemas, se observó en este curso que todos los grupos llegaron a entregar su trabajo práctico en tiempo y en forma.

Existe consenso entre investigadores y educadores del área de programación y en ella de la algoritmia, en lo dificultoso 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 función de esto Dania y Marchisio (2013) buscaron identificar las preferencias cognitivas predominantes de los estudiantes de ingeniería en sistemas de información de la Universidad del Centro Educativo Latinoamericano en Argentina, con el fin de proponer

innovaciones metodológicas con enfoque multimedia que permitan afrontar con éxito las dificultades de la enseñanza de la programación teniendo en cuenta las diferencias individuales asociadas a la percepción sensorial. Los resultados ofrecen orientaciones válidas, con significación especial al momento del diseño de estrategias didácticas que incluyeron el empleo de materiales multimedia para ayudar al alumno.

La situación de la materia de algoritmia que se imparte en la UMT de la UADY no difiere mucho de lo aquí descrito, en función de este problema se pensó en la búsqueda de una metodología que ayude al alumno en su proceso de enseñanza-aprendizaje, en especial a alcanzar los niveles de dominio más altos. En este sentido, el nivel de dominio se define como los atributos o características que describen el grado en que el estudiante ha desarrollado determinada(s) competencia(s) de una asignatura. Para fines de promoción de un estudiante, el nivel de dominio 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 éste se categoriza en sobresaliente, satisfactorio, suficiente y no acreditado (tabla 2.2) (UADY, 2012).

Tabla 2.2 Niveles de dominio para la acreditación de una asignatura en la UADY.

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

Fuente: UADY (2012).

2.2.3 Problemas comunes y herramientas usadas para mejorar

La complejidad que caracteriza a la programación y las actividades que se llevan a cabo relacionadas con ésta hacen que cuando los alumnos cursan 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 determinado (Salgado, Alonzo, Gorina, & Tardo, 2013). En este sentido, en las instituciones de educación superior, el profesor de programación debe enfrentar y resolver los siguientes problemas de carácter general que presentan la mayoría de los alumnos de primer ingreso Oviedo y Ortiz (2002, citado en Alania et al., 2012):

- Desconocimiento de la materia.
- Carencia de habilidades para programar y de disciplina en programación.
- Falta de conciencia estudiantil.
- Desinterés por sus estudios en general y por la materia en particular.
- La enseñanza de la programación implica la consideración de otras dos áreas importantes:
 - Los lenguajes de programación.
 - Las herramientas de desarrollo.

En la LIS de la UMT, es frecuente encontrarnos con los problemas antes citados, por un lado los alumnos llegan sin un conocimiento sobre programación (a excepción de algunos), por otro el cuerpo académico que respalda la LIS ha venido observando una apatía y falta de interés general en los alumnos de nuevo ingreso.

También de importancia, en Rosanigo y Paur (2006) se presentan una serie de errores que normalmente comete el alumno cuando resuelve problemas de

programación, sin embargo, algunos de ellos son completamente aplicables al desarrollo de algoritmos, entre los que se pueden mencionar como recurrentes y que aquejan a los estudiantes se mencionan:

- **Errores en operadores:** Confunden el uso de los operadores lógicos, por ejemplo, utilizan conjunción en lugar de disyunción o viceversa, o no reconocen bien cual 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 seleccionada.
- **Errores en la solución:** No muestran los resultados obtenidos, no resuelven el problema (suele suceder que resuelven otro), mala interpretación del enunciado (en particular este es un grave problema con los alumnos de la LIS, se ha observado una pobre comprensión lectora), falta de casos para comprobar la solución (comprobando solo el que funciona), establecen la solución general pero no la refinan, resuelven puntos parciales, pero no arman la solución global.

En particular Rosanigo y Paur (2006) adoptaron una serie de estrategias para resolver los principales problemas que enfrentaban, en primer lugar hicieron una redistribución de la carga horaria con el grupo, como segundo punto le dieron a los alumnos una guía de trabajos prácticos cuyo objetivo es que el alumno resuelva situaciones diferentes, preste atención a las palabras del enunciado y se familiarice con distintas maneras de expresar en lenguaje natural lo que en lenguaje algorítmico podría representarse de una misma forma, como tercer punto empezaron a evaluar también en forma práctica y por último se le dedica mayor tiempo al análisis de los datos de entrada del problema, las condiciones que deben cumplir y la definición de operaciones especiales para su validación, esto ha

mejorado la visión del alumno para resolver los ejercicios y disminuido los errores frecuentes que ocurrían sobre el tema.

Como apoyo a los planteamientos anteriores, han surgido varias iniciativas tecnológicas que involucran organizaciones y grupos colegiados en busca de ayudar al alumno en sus primeros pasos hacia la programación, sin que se enfrente a las dificultades inherentes de un lenguaje formal que implique el uso de una sintaxis específica. Fuentes (2017) menciona algunos de estos desarrollos entre los que se encuentra *Scratch*, *Snap!* y *App inventor*; mientras que otros autores como Gandy, Bradley, Brookes y Allen (2010) y Muñoz et al. (2015b) apuestan por *Legó Mindstorms*. A continuación, se presenta una breve descripción de las herramientas mencionadas:

Scratch inició a principios de 2003 con Mitch Resnick y John Maeda de MIT y Yasmin Kafai de UCLA como un proyecto del Lifelong Kindergarten Group en el MIT Media Lab y es un lenguaje de programación que permite crear historias interactivas, juegos, animaciones y programas en general y con esto desarrollar habilidades mentales mediante el aprendizaje de la programación sin tener conocimientos profundos sobre el código, todo mediante bloques, permitiendo al educando pensar de forma creativa, razonar sistemáticamente y trabajar en modo colaborativo (MIT Scratch Team, s.f.).

Snap! de la Universidad de California en Berkeley fue desarrollado por Jens Möning en MioSoft Corporation, con aportes de diseño y documentación por Brian Harvey de Berkeley. Es un lenguaje de programación visual que consiste en arrastrar y soltar, es una extensión de *Scratch* que permite construir bloques sin conocimientos previos de programación evitando los problemas que normalmente se presentan con la sintaxis. *Snap!* tiene características que *Scratch* no incluye como son las listas y los procedimientos de primera clase, la definición de procedimientos y la creación de estructuras de control, entre otros. Esto lo hace adecuado para una introducción seria a la informática para estudiantes de universidad (Rodríguez, 2020; Snap! Build your own blocks, 2018).

App inventor es una herramienta diseñada para los principiantes en la programación y creación de aplicaciones Android, desarrollada inicialmente por Google, el Programa de Formación de Maestros Scheller y el MIT. Se basa en la programación visual y en la construcción de bloques con el método de arrastrar y soltar (a semejanza de *Scratch* y *Snap!*), su uso está dirigido a la educación y su misión es fomentar el traspaso de los usuarios de consumidores de aplicaciones a desarrolladores de un modo simple, entre sus objetivos se encuentra la promoción de la educación en ciencias de la computación y la realización y apoyo a la investigación comunitaria (Fuentes A. , 2017).

Legó Mindstorms fue desarrollado por la compañía *Legó* y el MIT, inicialmente se propuso como un juego, pero ha ido adquiriendo interés creciente en entornos educativos a diferentes niveles. Según Gandy et al. (2010) la plataforma se usa para desarrollar competencias en programación tanto a nivel básico como avanzado. La programación con robots *Legó* permite al alumno desarrollar una aplicación tangible y concreta de las habilidades de programación basándose en los principios del aprendizaje activo y constructivo (Muñoz et al., 2015b).

En nuestro contexto, la herramienta TIC que se empleó fue la de *Scratch* por las características que presenta y otras razones de importancia que son tratadas en la sección siguiente.

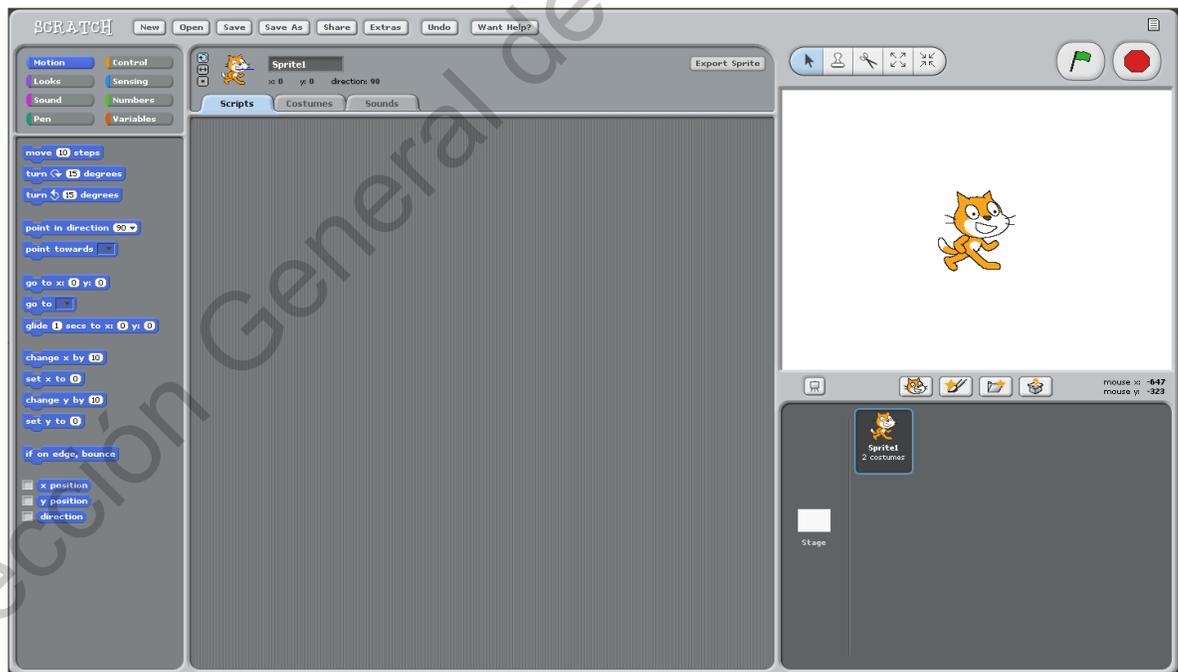
2.3 Scratch

Scratch es un lenguaje de programación que permite crear programas sin tener conocimientos profundos sobre el código (es decir, sin tener que estar sujetos a una sintaxis en particular) ya que es un entorno visual de programación por bloques. El diseño y la programación de proyectos en *Scratch* sirve de apoyo a los estudiantes para que aprendan a pensar de forma creativa, razonar sistemáticamente y a trabajar en modo colaborativo (MIT Scratch Team, s.f.; Monjelat et al., 2018).

Scratch es desarrollado y mantenido por el Equipo de *Scratch* en el grupo Lifelong Kindergarten del MIT Media Lab y una de sus más importantes características es que es un lenguaje multiplataforma y gratuito, por lo que no se necesita una licencia para usarlo, lo que lo hace accesible a una gran cantidad de usuarios e instituciones.

El desarrollo de *Scratch* fue propuesto por primera vez a la NSF (National Science Foundation) a principios de 2003 por Mitch Resnick, John Maeda de MIT y Yasmin Kafai de UCLA, fue aceptado y se les otorgó una subvención para su desarrollo. En su primera versión la 1.0 abarcó un proceso de desarrollo largo desde sus inicios en 2003 hasta el lanzamiento del programa final y el sitio web en enero de 2007 (figura 2.2) (Scratch Wiki, 2008).

Figura 2.2 Primera versión al público Scratch 1.0 en 2007.

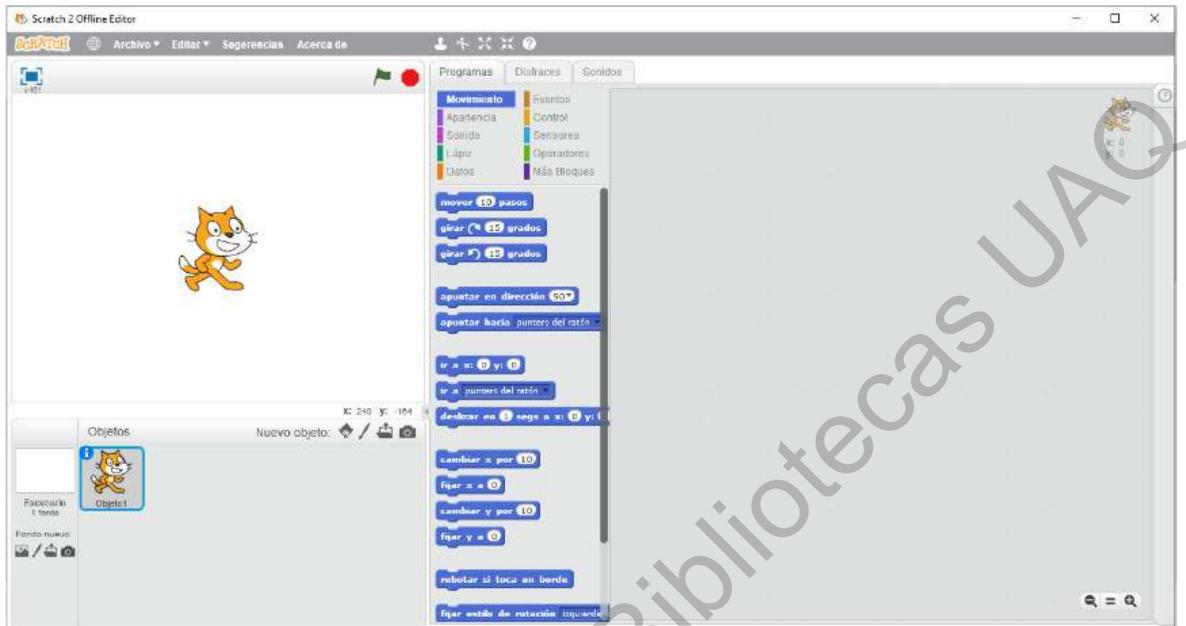


Fuente: Scratch Wiki (2008).

De mayo de 2007 a julio de 2009 se liberaron varias versiones las cuales fueron la 1.1, 1.2, 1.3 y 1.4, todas en apariencia física semejante a la versión 1.0 y entre las principales variaciones entre una versión y otra se puede citar de modo general la inclusión de nuevos bloques, mejoras en los sonidos, sensores y manejo de cadenas. En particular la versión 1.4 tenía otras características como el uso de la cámara web para tomar fotos y emplearlas en sprites y fondos, bloques ocultos en la categoría movimiento para usar con el kit *Legó WeDo Robotics*, un nuevo modo de escenario agregado al modo de presentación y al modo regular (modo de escenario pequeño) y un selector de color revisado en el editor de pintura, entre otros (Scratch Wiki, 2008).

La versión 1.4 perduró vigente hasta el año 2013 que es cuando se lanza una versión beta de *Scratch 2.0* la cual fue desarrollada durante varios años siendo su lanzamiento oficial en mayo de ese mismo año. Entre las principales características que incluyó destacan una nueva interfaz de usuario, procedimientos, clonación, datos en la nube, gráficos vectoriales, una mochila para recopilar scripts y un editor de sonido, entre otros, esto puede apreciarse en la figura 2.3.

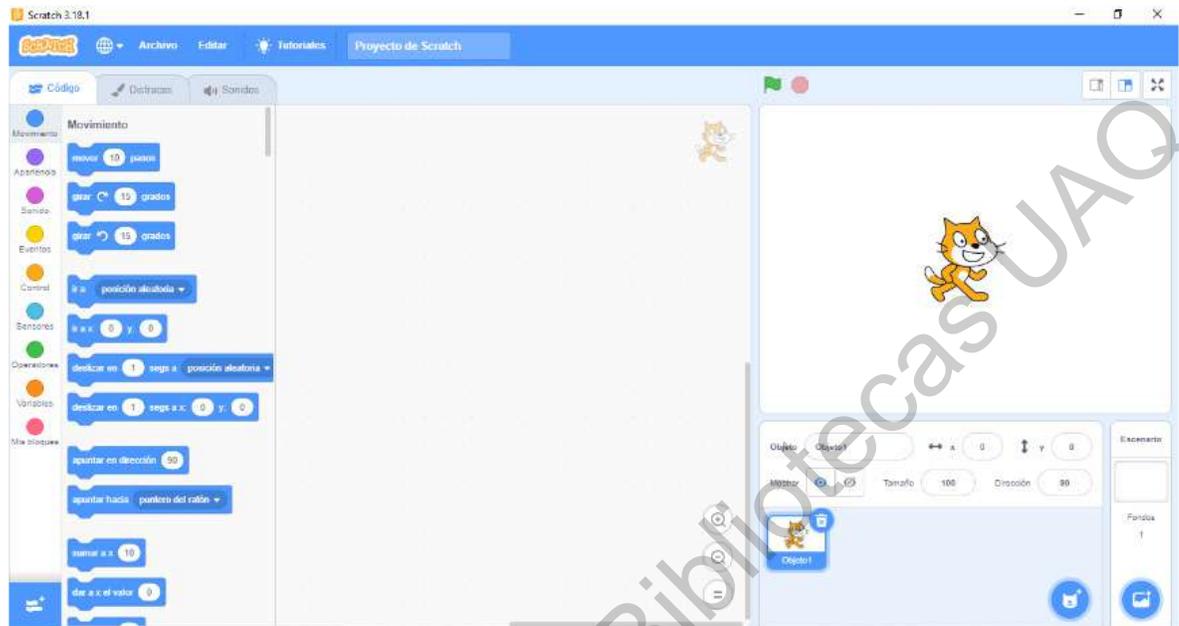
Figura 2.3 Scratch versión 2.0 en el 2013.



Fuente: Scratch Wiki (2008).

La versión 2.0 estuvo vigente desde el 2013 hasta enero de 2019 que fue cuando sale la versión 3.0 en la cual se anunció también que Google se afiliaría a *Scratch* para su lanzamiento y se usaría *Scratch Blocks*, una biblioteca bifurcada de Google's Blockly, la cual permitiría la colocación horizontal de bloques, junto con los bloques verticales. El lanzamiento de la versión 3.0 se había anunciado para finales de 2017, luego se reprogramó para agosto de 2018 y finalmente su lanzamiento oficial se realizó en enero de 2019, figura 2.4 (Scratch Wiki, 2008).

Figura 2.4 Scratch versión 3.0 enero de 2019.



Fuente: Scratch Wiki (2008).

Como se puede apreciar desde la primera versión al público en el año 2007 hasta la versión más actual liberada en enero de 2019, *Scratch* no ha dejado de evolucionar y mejorar, se puede decir que hoy es una herramienta madura. Todo esto para ir a la vanguardia con los desarrollos tecnológicos de muchas áreas y además porque con el paso del tiempo ha demostrado ser una herramienta eficaz para lo que fue diseñada, lograr que el estudiante aprenda, razone, diseñe y sea creativo sin tener que usar estrictas sintaxis de programación, más que pensar en programar el estudiante usa la herramienta jugando.

2.3.1 Conceptualización

Cuando se trabaja con la herramienta *Scratch* se entienden fácilmente conceptos matemáticos e informáticos que están muy bien integrados en el programa, como son los procesos secuenciales, interactivos (bucles), los criterios

condicionales (si, entonces, si - no), las coordenadas en un plano cartesiano, variables, hilos múltiples y eventos, entre otros (Isuri Sormenezko Zerbitzuak Servicios Creativos, 2018; Suárez, 2015).

Una de las principales ventajas de *Scratch* es que presenta una interfaz sencilla e intuitiva de fácil comprensión, donde se pueden manipular imágenes, fotos, sonido y música, en un entorno de programación visual el cual facilita crear nuevos personajes, cambiarles su apariencia y hacerlos interactuar con otros existentes. De igual modo, permite al educador que lo emplee el poder desarrollar en los estudiantes un pensamiento creativo, crítico y reflexivo; adicionalmente, se puede aplicar en diferentes asignaturas de tal manera que posibilita la transversalidad, genera en los estudiantes interés y los motiva planteándoles desafíos y la posibilidad de un aprendizaje diferente, por otro lado, brinda la oportunidad de obtener resultados complejos a partir de ideas simples donde cada estudiante puede trabajar a su ritmo en función de sus competencias.

Asimismo, de acuerdo con Aparicio y Ostos (2018) el lenguaje *Scratch* es ideal para construir pequeños códigos, historias y juegos informáticos, por medio del encaje de sentencias, órdenes y acciones, que hacen posible imaginar, programar, compartir (lema de *Scratch*), investigar, experimentar y familiarizarse con el área, mediante el uso de una interfaz que facilita el aprendizaje autónomo.

Es importante destacar que *Scratch* se encuentra disponible en distintos idiomas (incluyendo el español), la interfaz gráfica permite crear y utilizar un escenario con múltiples fondos y objetos móviles, en lugar de escribir instrucciones (códigos) se puede decir que la programación se da con el mouse, arrastrando los bloques de sus diferentes categorías (señaladas con colores) los cuales son autoencajables; esto implica explorar y adentrarse a los principios básicos de la codificación de problemas sin las complicaciones de sintaxis que tienen otros lenguajes, lo cual muchas veces dificulta el aprendizaje de los mismos (Suárez, 2015; Zamora, 2016).

Esta última característica es lo que lo hace ideal para emplearlo en la materia de algoritmia en la cual no se pretende enseñar al alumno a programar mediante el uso de sintaxis (esto corresponde a materias más avanzadas en LIS), por lo que el empleo de bloques que sustituye este punto es ideal para la materia. En particular *Scratch 2.0* incluye bloques de movimiento, apariencia, sonido, lápiz, datos, eventos, control, sensores, operadores y una sección especial para la creación de nuevos; por su parte la versión 3.0 incluye movimiento, apariencia, sonido, eventos, control, sensores, operadores, variables (semejante a la sección de datos de la versión 2.0) y un apartado para crear nuevos; en la figura 2.5 se pueden apreciar los componentes de estas dos versiones marcadas con colores. Aunque la versión 3.0 es la más reciente (en función de su lanzamiento al mercado), aún se encuentran muchas personas que siguen usando la versión 2.0, que es la empleada en esta investigación, esto unido al hecho de que cuando se inició con este estudio la versión 3.0 todavía se encontraba en desarrollo.

Figura 2.5 Bloques de Scratch 2.0 y 3.0.



Fuente: Elaboración propia (2019).

2.3.2 Uso de Scratch en la educación

Múltiples investigaciones han sido desarrolladas utilizando la herramienta *Scratch* tanto en la educación en general como en el ámbito específico de los principios de programación de computadoras en el nivel universitario; existe un considerable número de docentes que han apostado por esta herramienta para dar solución a sus problemas educativos, en particular, relacionados con ayudar al alumno en su rendimiento académico. En esta sección, se presentan algunas de estas investigaciones que han sido llevadas a cabo.

Según Tec, Uc, González, García, Escalante, y Montañez (2010), la robótica con *Legó* y la animación con *Scratch* se usan en diversas áreas, incluyendo las matemáticas ya que permiten interacción con la tecnología, la cual está ausente en la enseñanza tradicional en esta materia, estos autores explicaron temas matemáticos con apoyo de talleres en bachillerato y concluyen que el uso de estas TIC son una opción atractiva para el alumno.

En López (2012) en una Institución Educativa de Cali, Colombia se presenta una interesante propuesta, el uso de la programación mediante el lenguaje *Scratch* para identificar procesos emocionales con miras a aprender a regularlos. El investigador piensa que programar computadoras es una actividad que hace que afloren con facilidad en los estudiantes ciertas emociones tales como alegría, tristeza, ira, interés, preocupación, envidia, admiración, miedo, incapacidad y sorpresa y decidió enfocarse a *Scratch* y no otros lenguajes por ser sencillo y no usar complicadas sintaxis. El proceso fue llevado a cabo con niños de educación primaria y secundaria mediante el uso de emoticones que fueron incluidos a la galería de *Scratch* y como conclusión los docentes participantes han reconocido empíricamente el potencial que tiene la programación de computadoras para desencadenar en los estudiantes diversas emociones, esto se ha evidenciado cuando programan con *Scratch* y también han observado de manera experimental cambios positivos en los estudiantes.

Analizando la tendencia mundial Vázquez-Cano y Ferrer (2015) consideran que la programación en el aula es una actividad de presente y futuro para el desarrollo de competencias relacionadas con la realidad del mundo laboral y personal de los estudiantes; razón por la cual desarrollaron una experiencia educativa en la que alumnos de bachillerato crearon videojuegos en el aula mediante el software de programación *Scratch* con la intención de promover la creatividad en el proceso de enseñanza-aprendizaje ya que piensan que el analfabetismo digital vendrá representado por aquellas personas que no son capaces de ser creadores digitales. Como resultado de la experiencia propuesta los alumnos lograron trabajar en forma colaborativa, mejorando principalmente sus competencias de expresión y creatividad; también en el desarrollo de sus proyectos pudieron afrontar situaciones problemáticas y buscaron la manera de solucionarlas usando la técnica de ensayo-error y la crítica constructiva, conceptos que forman parte del pensamiento sistemático, por último, los alumnos pudieron controlar desde una mayor autonomía su proceso y su propio aprendizaje.

En Janeiro (2016), realizaron una intervención didáctica en alumnos de primaria enfocándose entre otros a fundamentos de la programación en *Scratch* y su aplicación en matemáticas y geometría, los niños aprendieron mediante juegos y la programación de sus personajes, lo cual añadió el aprendizaje significativo a través de aprender haciendo.

Sáez, Miyata y Domínguez-Garrido (2016) presentan un estudio donde analizan conceptos, actitudes y prácticas de 113 alumnos de tres importantes universidades de diferentes países: la Universidad Chukyo en Japón, la Universidad Cuauhtémoc en Aguascalientes, México y la Universidad Nacional de Educación a Distancia en España la cual tiene estudiantes provenientes de diferentes países del mundo; el objetivo se centró en la codificación mediante *Scratch*, para crear presentaciones multimedia en un contexto intercultural. El proyecto estuvo enmarcado en dos grupos de investigación y fue desarrollado para mejorar habilidades de codificación, los resultados muestran que el fomento de este tipo de

actividades tiene varias ventajas con respecto a las competencias TIC a pesar de las limitaciones de *Scratch*, a lo cual los estudiantes destacaron que es fácil de usar, divertido y perfecto para presentaciones y animaciones. El proyecto ha mejorado la eficacia con respecto a la capacidad de los estudiantes para comprender y usar contenido multimedia a través de la programación en bloques lo cual permite mejorar las presentaciones. Los datos obtenidos mediante pruebas, cuestionarios y entrevistas muestran actitudes positivas y favorables hacia el uso de *Scratch*.

Por último, se destaca un desarrollo que se presenta en Collí (2017), donde crearon un software utilizando la herramienta *Scratch* para ser empleado en estudiantes de nivel bachillerato con el objetivo de poder ayudarlos en el aprendizaje del inglés y la maya, con apoyo del Kinect. El software ha sido aplicado de manera experimental a un grupo de alumnos y hasta ahora ha arrojado buenos resultados.

2.3.3 Uso de Scratch en la universidad

En esta sección se presentan casos específicos de la aplicación de *Scratch* en el nivel universitario, relacionados con los problemas que tienen los docentes con las materias introductorias a los principios de programación.

Según Dania y Marchisio (2014) la enseñanza de la programación y la algoritmia ha sufrido muchos cambios, coexistiendo varios enfoques didácticos y tendencias como los que aquí se describen. Existe consenso entre investigadores y educadores, en lo difícil que resulta a los estudiantes de los primeros años de estas carreras el superar los mismos exitosamente con base en los aprendizajes complejos que se requieren.

En este contexto, es importante en primera instancia traer a colación las investigaciones realizadas por Stuart (2009) el cual plantea el uso de *Scratch* desde los primeros años de la educación hasta introducir a universitarios a la programación y concluye que su uso es una buena alternativa para que luego el estudiante dé el salto a otros lenguajes con sintaxis y entornos de desarrollo más avanzados.

Por otro lado, en el trabajo de Uludag, Karakus y Turner (2011) se resume la experiencia en la enseñanza en un curso introductorio de computación, con los fundamentos pedagógicos en materia de aprendizaje constructivista y educación informática contextualizada, presentando la motivación y los detalles de un curso que utiliza *Scratch*, *App inventor* y *Legó Mindstorms*.

Por su parte Muñoz et al. (2015b) menciona que los cursos en los primeros años de las carreras informáticas relacionados con fundamentos de programación tienen altas tasas de reprobación y propusieron iniciativas para mejorar asociadas con *Scratch* y *Legó* desarrollando un taller de programación de juegos para apoyar la asignatura, al final encontraron un incremento considerable en la tasa de aprobación de los estudiantes a raíz de su empleo.

En el trabajo realizado por Muñoz, Barcelos, Villarroel y Silveira (2015a), diseñaron un taller de fundamentos de programación a nivel universitario y con la creación de juegos en *Scratch* los alumnos aumentaron su motivación para aprender, aplicando la programación de manera tangible (lo cual es difícil de lograr únicamente con el diseño de los algoritmos), los resultados indican un aumento en las tasas de aprobación.

Por su parte, Vidal, Cabezas, Parra y López (2015) realizaron experimentos para enseñar pensamiento computacional y algoritmos en la universidad usando *Scratch* en vez de otros lenguajes cuya sintaxis dificulta el acceso y uso de sistemas computacionales, los resultados fueron satisfactorios. *Scratch* es una herramienta propicia para el desarrollo del pensamiento lógico y algorítmico, los alumnos se motivan y participan encontrando respuestas y creando aplicaciones que pueden probar y mejorar.

Como se expone no son pocos los docentes que mediante la inclusión de *Scratch* han logrado ayudar a sus estudiantes en los procesos de programación de computadoras, de acuerdo con la literatura encontrada *Scratch* es una herramienta ampliamente utilizada y se palpa su efectividad en las situaciones en donde la

aplicaron (Durak, 2018; Muñoz et al., 2015b; Rizvi et al., 2011; Stuart, 2009; Uludag et al., 2011; Vidal et al., 2015).

Se puede concluir que la inclusión de una TIC en un problema educativo es un punto favorable y una necesidad en el ámbito educativo actual, los docentes ven la incorporación de éstas como algo motivante y de mejora en el rendimiento académico, siempre y cuando exista una planeación adecuada y esté acorde al nivel educativo.

En particular, la inclusión de *Scratch* en el área de principios de programación sería algo benéfico, la literatura permite ver una variedad de casos donde se ha usado exitosamente con la metodología de la construcción de talleres, los cuales se aplican siguiendo la teoría del aprendizaje constructorista; con esto el alumno aprende haciendo y logra aterrizar los conceptos, todo enfocado a las fases introductorias a la programación.

El interés de esta investigación se centró en apoyar al alumno en la construcción de algoritmos mediante diagramas de flujo en un curso introductorio a la programación, lo cual es un paso previo a la programación, aquí es donde se empleó *Scratch* aplicándolo en actividades de la planeación didáctica de algoritmia. La intervención se hizo mediante la creación de una metodología que involucró el diseño de un taller que empleó del lenguaje de programación *Scratch* aquellas características que se acoplaron a los temas problema y se diseñaron ejercicios y animaciones, todo esto siguiendo la teoría del aprendizaje constructorista.

La teoría del constructorismo dice que el conocimiento es una construcción humana realizada mediante estructuras previas a partir de las cuales se construye, para Ackermann (2001, citado en Aparicio y Ostos, 2018) el constructorismo de Papert se centra en el arte de aprender o aprender a aprender, se interesa en cómo los estudiantes se involucran en una conversación (propia o ajena) con artefactos tangibles y cómo estas conversaciones fomentan el aprendizaje autodirigido para facilitar la construcción de nuevos conocimientos; esto resultó útil para desarrollar el pensamiento complejo en los estudiantes, los cuales crearon programas,

animaciones y juegos básicos con la herramienta y de esta manera el alumno pudo aterrizar en algo tangible lo que trabaja del modo tradicional entendiendo los conceptos que le resultan complejos. Con este proceso se obtuvieron resultados positivos los cuales se han alcanzado en ambientes similares de acuerdo con la literatura revisada.

Dirección General de Bibliotecas UAQ

3. METODOLOGÍA

En este capítulo se describe el escenario general de la investigación, exponiendo en primera instancia el problema de interés, así como los elementos que sirven de sustento para la justificación de ésta, de igual modo, se plantea la hipótesis, los objetivos de la investigación y el diseño desarrollado, así como también, se especifica la población sobre la que se recolectaron todos los datos necesarios para el desarrollo del trabajo que se presenta. Finalmente se establecen los instrumentos, los procedimientos seguidos, las limitaciones del alcance de la investigación y se detalla la metodología de aprendizaje que se propone.

3.1 Problemática en estudio

El interés de este estudio se centra en la creación de una metodología que emplee el lenguaje de programación *Scratch* en la materia de algoritmia que se imparte en la licenciatura en ingeniería de software en la Unidad Multidisciplinaria Tizimín, para ayudar a los alumnos en su rendimiento académico.

La intención que se persigue en la creación de esta metodología es que por medio de ella el alumno pueda desarrollar referentes de tipo visual (programas generados en *Scratch*) que le permitan entender mejor la resolución de ejercicios que se hacen en la materia mediante los diagramas de flujo los cuales utilizan las estructuras básicas de programación y como consecuencia de esto pueda elevar su rendimiento académico. El éxito de esta metodología hará que los alumnos cuenten con bases más sólidas para las materias siguientes del área.

La herramienta TIC *Scratch* es la seleccionada a ser aplicada en este contexto ya que favorece el aprendizaje activo y constructivo mediante la creación de proyectos multimedia interactivos y es una herramienta de uso libre (Pajares, 2014).

3.1.1 Sujeto experimental

El contexto geográfico de este estudio se encuentra situado en la UMT que depende de la UADY, la cual se encuentra ubicada en la ciudad de Tizimín, en el estado de Yucatán, aproximadamente a 160 km de distancia de Mérida, la capital. La UMT cuenta con 400 alumnos aproximadamente distribuidos en 4 licenciaturas las cuales son: ingeniería de software, educación, enfermería y contador público; todas ellas reciben a sus alumnos provenientes de los bachilleratos de la región.

En particular el grupo de estudio sobre el que se llevó a cabo la intervención lo componen los alumnos de primer semestre de la licenciatura en ingeniería de software, la cual tiene un ingreso al año en el mes de agosto y sus estudiantes provienen en su mayoría de los siguientes bachilleratos:

- Colegio Nacional de Educación Profesional Técnica plantel Tizimín (Conalep).
- Centro de Bachillerato Tecnológico Agropecuario (CBTA) #14 de Tizimín.
- Colegios de Bachilleres de Yucatán (Cobay) como son el de Tizimín, Temozón, Sucilá, Calotmul, Colonia Yucatán, entre otros.
- Preparatorias incorporadas al sistema UADY.
- Colegio de Estudios Científicos y Tecnológicos del Estado de Yucatán (CECyTEY) de los municipios de Panabá y Espita.
- Otros.

Como se observa se reciben alumnos de una gran variedad de sistemas lo que implica que cuenten con diversos perfiles y conocimientos en función de sus planes y programas de estudio, esto conlleva muchas veces tener que unificar conocimientos, además, es importante destacar que son pocos los alumnos que llegan con algunos conocimientos sobre el área de algoritmia, en su mayoría son recién egresados, tienen en promedio 18.96 años y un alto porcentaje son de origen maya.

Por otro lado, también es importante mencionar que la UMT cuenta con un aula especial (laboratorio de cómputo) para poder llevar a cabo la intervención, la cual fue diseñada para que los estudiantes realicen la parte práctica de sus asignaturas. El lenguaje *Scratch* a usar es gratuito y multiplataforma, por lo que no existe problema alguno para descargarlo desde internet e instalarlo en los equipos del laboratorio y para tal fin se tiene el apoyo de un grupo de soporte tecnológico el cual se encargó de esta labor. Asimismo, resulta indispensable que el estudiante lleve a cabo prácticas fuera del contexto escolar, en este caso tampoco existe impedimento alguno para que se les proporcione el software o bien el mismo lo descargue e instale en su equipo personal.

Es relevante mencionar que la intervención propuesta se pudo llevar a cabo gracias a que la UADY brinda a sus docentes las facilidades para introducir cambios a las materias de manera libre para mejorarlas (UADY, 2012).

3.2 Justificación del problema

Las materias que se cursan en la universidad donde los alumnos no logran alcanzar los niveles requeridos para aprobarlas no son propias de una carrera o área en particular. Esta situación es común que se presente cuando se introduce al alumno en el razonamiento de problemas para posteriormente buscarles una solución en el área computacional, lo que habitualmente sucede en algoritmia.

Partiendo del planteamiento anterior se reflexiona sobre la forma en cómo se ha venido dando la citada materia, el contenido se suele trabajar a papel y lápiz del modo tradicional y el alumno con esta forma de trabajo con frecuencia no aterriza los conceptos estudiados en el aula porque el método empleado no le permite ver cómo es que el algoritmo generado es la solución a su problema, al no tener algo tangible como punto de referencia que le permita ver y seguir los resultados, en consecuencia su rendimiento al final del curso no siempre es el deseado.

En la UMT de la UADY se imparte en la LIS la materia de algoritmia desde el año 2016, en ella se detecta una situación problemática de la que se han obtenido los siguientes resultados, en 2016 el 58.33% de los alumnos tuvo un nivel de desempeño entre satisfactorio y sobresaliente y el 41.66% entre suficiente y no acreditado; en 2017 el 46.15% tuvo un nivel de desempeño entre satisfactorio y sobresaliente y el 53.84% entre suficiente y no acreditado; con estos datos se puede observar que el rendimiento general de los grupos no es el óptimo en ninguno de los dos cursos 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 más adelante con materias relacionadas. Según los resultados obtenidos y lo que se ha observado, los alumnos no logran entender el correcto resultado de los problemas planteados, es decir los diagramas de flujo y el pseudocódigo.

Ante esta situación, se ha encontrado que las nuevas TIC aplicadas a la educación son un apoyo en el proceso de enseñanza de diversos temas en el área de la programación y un procedimiento que se ha empleado es el uso del lenguaje de programación *Scratch* el cual por medio de la construcción de programas mediante el uso de bloques ha permitido que los docentes puedan ayudar a sus estudiantes (Janeiro, 2016; Muñoz et al., 2015b; Stuart, 2009; Tec et al., 2010; Uludag et al., 2011; Vidal et al., 2015). Partiendo de este punto, surge el interés de llevar a cabo una intervención educativa en el aula con el uso de *Scratch* la cual va más allá del simple empleo de la herramienta en el salón de clase porque así se indique en el programa de estudio o por la parte administrativa del centro, se encamina al apoyo del educando ante diversas situaciones conflictivas que pueden presentarse en el aula escolar (incluyendo el bajo rendimiento), las cuales el docente pudiera detectar y a partir de esto, hacer una profunda y adecuada planeación para intervenir en el grupo. Es un proceso que parte de una intención claramente identificada, es bien definido, conlleva un conjunto de acciones a realizar y va dirigido a la mejora y cambio de la situación; para poder intervenir se plantea

la creación de una metodología guiada en la investigación basada en diseño (IBD).

Ante el panorama que se presenta, resulta posible incorporar *Scratch* a la materia de algoritmia como herramienta tecnológica de innovación y de apoyo, de igual modo se tiene el antecedente de que otros docentes en la UMT la han usado en algunos temas de sus asignaturas con buenos resultados. Adicionalmente, emplear *Scratch* en un aula incide en la teoría del construccionismo la cual según Vicario (2009) constituye un referente obligado para todos aquellos que trabajan en el campo de la innovación educativa y se han comprometido a transformar la realidad.

El construccionismo tiene sus orígenes en lo que hoy se conoce como informática educativa, su creador es el autor de *Logo*, que marcó un parteaguas en la producción del software para la educación desde la década de 1960, *Logo* fue diseñado por Seymour Papert y algunos de sus colaboradores y fue concebido con la finalidad de usarlo para enseñar los conceptos básicos de la programación así como para presentar retos intelectuales a los estudiantes y puedan resolverlos mediante un lenguaje de programación de tipo procedimental y recursivo, potenciador de la creatividad y la heurística (Aparicio & Ostos, 2018). La arquitectura de *Logo* está basada en el enfoque construccionista del propio autor y a la vez esta herramienta es la antecesora de tecnologías como micromundos, *Lego Mindstorms* y hoy en día *Scratch* (Vicario, 2009).

El construccionismo como teoría del aprendizaje contemporáneo pone acento en el valor de las TIC como poderosas herramientas de construcción mental, útiles para desarrollar el pensamiento complejo (Fernández, Calderón, Méndez y Rolim, 2014). La premisa básica del aprendizaje del enfoque construccionista supone la existencia de una habilidad natural en los seres humanos para aprender a través de la experiencia y para crear estructuras mentales que organicen y sinteticen la información y vivencias (Vicario, 2009). Papert y Harel (1991, citados en Aparicio y Ostos, 2018) dicen que el construccionismo tiene la misma

connotación del constructivismo del aprendizaje como creación de estructuras de conocimiento, independientemente de las circunstancias del aprendizaje y además agregan que esto ocurre en un contexto donde la persona que aprende está conscientemente dedicada a construir una entidad pública, ya sea un castillo de arena en la playa o una teoría del universo.

El aprendizaje construccionista implica a los estudiantes en su propio proceso educativo para que ellos mismos saquen sus propias conclusiones mediante la experimentación creativa y la elaboración de sus propios productos (Aparicio & Ostos, 2018), entendiendo por producto la elaboración de un material tangible como citaron Papert y Harel, por ejemplo: elaborar un informe, hacer una conferencia, crear un juego con *Scratch*, etc.

Partiendo de estos conceptos y vinculando *Scratch* con el construccionismo se logra que el estudiante construya sus propios productos los cuales son un referente tangible para aterrizar sus algoritmos y le permiten entender las soluciones encontradas, la retroalimentación visual que los alumnos obtienen de *Scratch* les permite relacionar el algoritmo que realizan con la acción que ven en pantalla y esto les ayuda a entenderlo, parte esencial tanto de la solución de problemas como de la programación. El construccionismo afirma que el aprendizaje es mejor cuando los estudiantes se comprometen en la construcción de un producto significativo y de acuerdo con Pérez (2013, citado en Aparicio y Ostos, 2018) el maestro aparece como facilitador que actúa mediando entre el estudiante y su propio proceso. Según Papert, el aprendizaje debe ser autodirigido, siguiendo un proceso iterativo para que los estudiantes puedan descubrir por sí mismos las mediaciones que mejor apoyen la exploración de su curiosidad (Aparicio & Ostos, 2018), lo cual es un aspecto clave con el uso de *Scratch*.

Asimismo, otra razón de peso de incluir *Scratch* a la asignatura de algoritmia, es que es un lenguaje de programación visual, mediante bloques y su principal característica consiste en que permite el desarrollo de habilidades mentales mediante el aprendizaje de la programación, sin tener conocimientos profundos

sobre el código, por otro lado, gracias al uso de este lenguaje los alumnos pueden aprender fácilmente conceptos matemáticos e informáticos como son las variables, constantes, ciclos, estructuras de decisión y selección, entre otros (Posada Prieto, 2017); siendo estos algunos de los aspectos donde se encontró la mayor cantidad de errores por parte de los alumnos. Existe evidencia de que esta herramienta es benéfica, para formar percepciones positivas sobre la programación ya que se introduce al estudiante a este mundo de una manera amena, motivadora y eficiente.

Igualmente, es importante mencionar que *Scratch* es una herramienta que está en constante evolución y mejora continua desde sus inicios (Calderón, 2020), lo que indica que no pasará de moda como otras; esto habla del respaldo que tiene por la institución que lo desarrolla y por otro lado cuenta con una gran comunidad de usuarios a nivel mundial, foros de discusión y soporte, una wiki y una página web que sirven de apoyo para aclarar dudas.

Con esta intervención educativa se creó una metodología la cual ha estado sirviendo para mejorar la comprensión de los alumnos en problemas del área y por ende el rendimiento académico en algoritmia; posteriormente, este conocimiento adquirido se espera lo lleve a tener un mejor desempeño en otras materias relacionadas.

En cuanto a la organización y gestión del centro escolar, en la UMT de la UADY trabajan profesores capacitados en el uso de la TIC propuesta y profesionales de la educación que estuvieron dispuestos a colaborar en el proceso, se contó con la infraestructura adecuada para la implementación y los directivos no limitaron su uso.

Esta propuesta considera las buenas prácticas con TIC, que según Sosa, Peligros y Díaz (2010) es todo aquel proceso educativo que con el uso de TIC supone una potencialización del proceso de enseñanza-aprendizaje y de sus resultados, esto implica pensar nuevas formas de hacer la labor docente marcada con el uso de la tecnología.

3.3 Establecimiento de la hipótesis

El planteamiento teórico consistió en determinar si incluir el lenguaje *Scratch* aplicando una metodología de aprendizaje, ayudaría en su rendimiento académico a los alumnos de la asignatura de algoritmia la cual se imparte en la LIS en la UMT. Con esto se formuló la pregunta de investigación:

- ¿El uso del lenguaje de programación *Scratch* mediante la creación de programas, juegos y/o animaciones básicas elevará el rendimiento académico de los alumnos que cursan la asignatura de algoritmia que se imparte en el primer semestre de la LIS en la UMT de la UADY?

Sobre la pregunta general propuesta, se formularon las siguientes preguntas específicas:

- ¿Cuántos y qué programas, juegos y/o animaciones básicas deben diseñarse?
- ¿Cuándo deben ser aplicados los programas, juegos y/o animaciones diseñadas?
- ¿El uso de la herramienta *Scratch* ayudará al alumno a entender los diagramas de flujo y tener un mejor rendimiento en la materia?
- ¿*Scratch* será bien recibida por los alumnos, será útil y atractiva durante el proceso?

Partiendo de las preguntas anteriores se formula la siguiente hipótesis:

Utilizar una metodología de aprendizaje que incorpore el lenguaje de programación *Scratch* en la asignatura de algoritmia que se imparte en la LIS de la UMT, mejora la comprensión de problemas del área y eleva el rendimiento académico de los alumnos.

Con base a lo anterior se analizó la situación que se tenía, se creó el diseño y desarrollo de los temas y materiales de apoyo que sirvieron de guía al estudiante

durante un curso propedéutico y un taller; y por último, se procedió con la evaluación.

3.4 Objetivos del estudio

Objetivo General:

El objetivo general propuesto es el siguiente:

Crear una metodología de aprendizaje que incorpore el lenguaje de programación *Scratch* en la asignatura de algoritmia de la LIS que se imparte en la UMT para mejorar el rendimiento académico de los alumnos.

Objetivo Particular:

Con base al objetivo general, se proponen los siguientes objetivos específicos:

- Examinar el nivel de conocimientos que, respecto al área de algoritmia, tienen los alumnos de nuevo ingreso de la LIS mediante un curso de inducción.
- Definir ejercicios y actividades a desarrollar en *Scratch* a lo largo del curso atendiendo a los temas donde los estudiantes presentan dificultades, considerando la experiencia.
- Aplicar el material generado en el laboratorio con el lenguaje de programación *Scratch*.
- Analizar los resultados de la implementación del lenguaje de programación *Scratch*, mediante métodos estadísticos.

3.5 Diseño del estudio

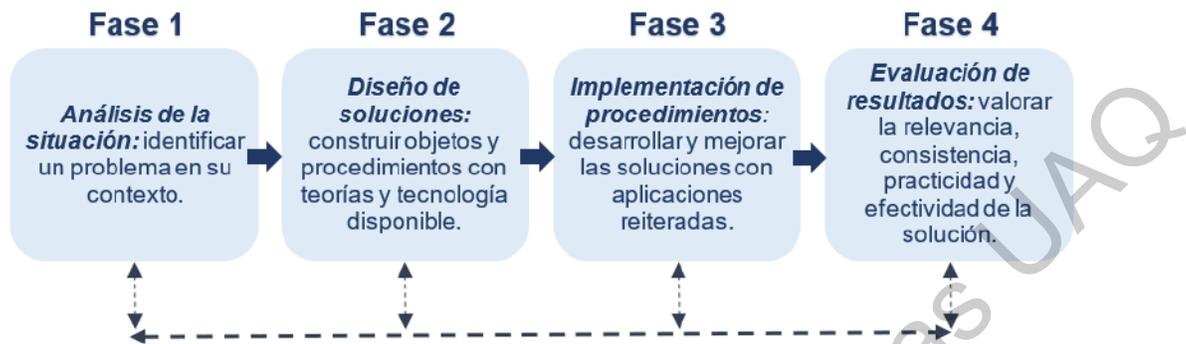
El trabajo de investigación que se planteó incluyó el análisis, diseño, implementación y evaluación de una propuesta de intervención pedagógica que

incorporó el uso de *Scratch* el cual es un lenguaje de programación mediante bloques para promover la adquisición de conocimientos de la materia algoritmia que se imparte en la LIS de la UMT con el fin de mejorar el rendimiento académico de los alumnos.

Se consideró apropiado utilizar el enfoque de investigación basada en diseño (IBD), entre cuyas características principales se encuentra la decisión de ubicar la investigación en el contexto natural, el propósito de producir cambios específicos en ese contexto y el carácter cíclico e iterativo de los diseños (Gibelli, 2014). La IBD está orientada a la innovación educativa introduciendo un elemento nuevo para transformar una situación (De Benito y Salinas, 2016), se centra en el diseño y exploración de todo tipo de innovaciones educativas a nivel didáctico y organizativo, considerando también posibles artefactos (el software, en este caso *Scratch*) como núcleos de esas innovaciones y contribuyendo a una mejor comprensión de la naturaleza y condiciones del aprendizaje (Bell, 2004). La IBD es el estudio sistemático y reflexivo de las fases que se aplican para solucionar un problema educativo complejo, teniendo un doble objetivo: primero resolver el problema educativo en cuestión, segundo al finalizar el estudio ofrecer principios de investigación para que estudios similares cuenten con referencias metodológicas Easterday, Lewis y Gerber (2014, citado en Escudero, 2018).

Para conducir la IBD, generalmente se admiten cuatro fases flexibles y recursivas: análisis, diseño, implementación y evaluación. El término flexibilidad hace referencia a que las etapas de la investigación se conducen de modo versátil, siendo sensible al ambiente y a las condiciones donde se realiza la investigación; es decir, si es necesario éstas se realizan sin un orden estricto pero garantizando una disposición rigurosa al presentar los resultados (figura 3.1) (Escudero, 2018).

Figura 3.1 Fases generales de la IBD.



Fuente: De Benito y Salinas (2016, citado en Escudero, 2018).

Cada una de las fases citadas de la IBD se detallan en breve en los siguientes párrafos:

Análisis de la situación: 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 de la intervención tanto en el contexto educativo como con los alumnos participantes, se determinó el nivel de conocimientos de los alumnos de nuevo ingreso de la LIS mediante un curso propedéutico que se les impartió previo al inicio del semestre y los resultados arrojados de éste sirvieron para definir con claridad la situación de los alumnos, también el curso ayudó para que los estudiantes tengan un primer acercamiento con algoritmia y con el lenguaje de programación *Scratch* para unificar conocimientos, lo cual fue un aspecto muy importante ya que como se comentó en apartado anterior los alumnos que estudian la LIS llegan a la UMT de una gran variedad de sistemas educativos por lo que resulta importante partir de un punto común, por último, se finalizó con la descripción de la situación y definición de necesidades.

Con el fin de tener una evidencia justificada de dónde los alumnos presentan más problemas en la materia de algoritmia, al término de la implementación de los cursos propedéuticos y de la primera y la segunda intervención de la herramienta,

se procedió a realizar un análisis de contenido de las pruebas de desempeño para identificar los errores más frecuentes y posteriormente se usó esta información en las siguientes intervenciones.

Diseño de soluciones: En primera instancia fue importante conocer lo que otros docentes-investigadores han hecho frente a situaciones similares; de lo encontrado y lo obtenido en la fase de análisis se generó el planteamiento de un taller de *Scratch* para la intervención con el objetivo de solucionar el problema señalado, se especificaron los temas a tratar así como sus extensiones para que sean implementadas mediante el uso de la herramienta *Scratch*, se determinaron ejercicios y animaciones a desarrollar y con base en ellos se crearon compendios específicos de trabajo los cuales aterrizaron los contenidos y los principales errores encontrados del análisis de contenidos, estos compendios sirvieron de apoyo para la implementación de la herramienta e incluían el detalle paso a paso de los problemas, se definió el modo de trabajo en el aula y se detalló el proyecto final a entregar el cual se desarrolló aplicando el aprendizaje basado en proyectos (ABP). De igual forma, en esta etapa se diseñó el cuestionario de evaluación que se utilizó al término de cada uno de los procesos de intervención.

Implementación de procedimientos: Se puso en marcha todo lo generado en las fases anteriores en el entorno adecuado (salón/laboratorio) con la participación de los alumnos. En esta etapa se incorporó *Scratch* a algoritmia, este proceso se llevó a cabo en tres ocasiones haciendo los refinamientos necesarios en cada uno de ellos según el caso y se implementó con alumnos de primer semestre de la LIS en la UMT en los períodos siguientes:

- Primera implementación, semestre agosto – diciembre de 2018.
- Segunda implementación, semestre agosto – diciembre de 2019.
- Tercera implementación, semestre septiembre 2020 – febrero de 2021.

En cada intervención primero se procedió con la implementación de un curso propedéutico, sin embargo, cabe mencionar que de las tres intervenciones el propedéutico solamente se dio en la segunda y tercera ocasión, dado que al inicio no se tenía el encuadre adecuado de la investigación.

Posterior al curso propedéutico, se intervino la asignatura de algoritmia con la herramienta *Scratch*, proceso que se planteó formado de 4 secciones:

- Sección 1, estructura secuencial y variables.
- Sección 2, estructura selectiva.
- Sección 3, estructura iterativa.
- Sección 4, integración enfocada al desarrollo de una animación o juego.

Cabe señalar que las unidades incluidas en la planeación didáctica de la materia de algoritmia son 3, siendo éstas las siguientes (Gómez et al., 2016):

- Unidad I Introducción a la algoritmia.
- Unidad II Diagramas de flujo.
- Unidad III Pseudocódigo.

En la primera unidad no se incluyó el uso de la herramienta *Scratch* dado que solo se ven conceptos introductorios, la unidad 2 que se refiere a los diagramas de flujo es la unidad que se estudió con apoyo de la herramienta *Scratch*, en cuanto a la unidad 3 se ve a la par de la unidad 2 ya que los temas son coincidentes (solo cambia el algoritmo con el que se presenta la solución), por lo que los problemas que se resuelven con *Scratch* inciden también en ella.

En el aula de clase se definen las estructuras del desagregado de contenidos de la planeación didáctica y se realizan ejercicios a papel y lápiz, posteriormente, y una vez dadas las bases se procedió con el empleo de *Scratch* en el laboratorio, el objetivo de éste consistió en pasar los ejercicios desarrollados en el aula a la

programación. Por último, el alumno añadió conceptos de animación a sus programas para hacerlos más dinámicos, por consiguiente, de modo paulatino y de forma activa, aprendieron el manejo de *Scratch* y adquirieron y construyeron los conocimientos básicos y necesarios de la materia algoritmia. Es importante mencionar que antes de esta propuesta el material de trabajo se estudiaba al 100% con papel y lápiz.

Evaluación de resultados: Durante todo el proceso investigativo se fueron recabando los datos necesarios para llevar a cabo la evaluación, estos datos se obtuvieron del curso propedéutico, de las actas de examen final de cada una de las ocasiones en que se hizo la intervención, de los resultados de los proyectos, así como también del cuestionario aplicado, es importante mencionar que durante el proceso el docente ha analizados las pruebas de desempeño y las actividades de aprendizaje y de igual modo ha observado lo que sucede en cada una de las sesiones impartidas, con esta información se logró determinar cuáles son los principales problemas que aquejan a los alumnos. Al final de cada proceso de intervención, teniendo recopilados todos los datos se ha llevado a cabo el estudio de estos, mediante análisis y métodos estadísticos. Se verificó que los contenidos seleccionados y presentados sean los adecuados.

3.6 Población y muestra

En cuanto a los participantes del estudio, se trabajó con 3 grupos de alumnos los cuales se correspondieron con los semestres de agosto – diciembre 2018, agosto – diciembre 2019 y septiembre 2020 – febrero 2021 con la totalidad de los estudiantes inscritos, ya que en función de los datos históricos de la suma de alumnos de nuevo ingreso, este valor en los tres grupos no superaría los 100. La cantidad con la que se trabajó fue de 66 quedando distribuidos de la siguiente manera:

- Grupo 1 del semestre agosto – diciembre 2018, 20 alumnos.

- Grupo 2 del semestre agosto – diciembre 2019, 19 alumnos.
- Grupo 3 del semestre septiembre 2020 – febrero 2021, 27 alumnos.

Partiendo de esta situación, se propuso que la población sea la muestra también, es decir, todos los alumnos inscritos a algoritmia en el primer semestre de la LIS en la UMT durante los períodos semestrales citados.

3.7 Datos requeridos

Para llevar a cabo el estudio que se ha descrito se requirieron datos para que sean analizados y revisados. A continuación, se citan los que se consideraron necesarios:

- Datos relacionados con los conocimientos previos de los alumnos de nuevo ingreso a la licenciatura en ingeniería de software.
- Resultados de las calificaciones finales de cada uno de los tres grupos con los que se trabajó con *Scratch* (2018, 2019 y 2020-2021) y de los grupos anteriores que fueron impartidos sin el uso de *Scratch* (2016 y 2017).
- Datos relacionados con la aplicación de un cuestionario de evaluación sobre la manera en que fue percibida la herramienta empleada durante el proceso de estudio.
- Análisis de las pruebas de desempeño y las actividades de aprendizaje para detectar los principales temas problema que tienen los alumnos.

3.8 Instrumentos

Los instrumentos que se emplearon para la recolección de la información de este estudio fueron los siguientes:

- Una prueba exploratoria sobre antecedentes académicos para conocer el nivel de conocimientos que tienen los alumnos al ingresar a la LIS relacionados con

la materia algoritmia, esta prueba evaluó conceptos matemáticos elementales que deben de saber los egresados de bachillerato, así como sus antecedentes de algoritmia y se aplicó los 3 años de la intervención.

- Las actas de examen generadas al finalizar los semestres en los que se llevó a cabo el estudio (2018, 2019 y 2020-2021) y las actas de examen de los años anteriores (2016 y 2017) para comparar los resultados, estas se obtuvieron del Sistema de Información y Control Escolar Institucional (SICEI) de la UADY.
- Cuestionario para evaluar aspectos sobre cómo percibieron la herramienta *Scratch*, su impacto y si los ayudó a entender los conceptos y un mejor puntaje en la materia.
- Las pruebas de desempeño para detectar los principales temas problema de algoritmia.

3.9 Procedimiento para la obtención y análisis de datos recabados

El manejo de la información recabada en los instrumentos mencionados se llevó a cabo de la siguiente manera:

- Las pruebas de diagnóstico se calificaron y generaron resultados que dieron una idea sobre los antecedentes de los alumnos de nuevo ingreso que cursan la LIS, de igual modo se hizo un análisis de cada una.
- Las actas de examen se obtuvieron del SICEI al finalizar el curso de algoritmia en los semestres agosto – diciembre 2018 y 2019 y septiembre 2020 – febrero 2021, también se obtuvieron las actas de los cursos impartidos en agosto – diciembre 2016 y 2017 (en los que las clases fueron impartidas de manera tradicional), igualmente se les aplicaron métodos estadísticos a los datos obtenidos.
- En cuanto a los datos del cuestionario, fueron respondidos por los alumnos de algoritmia al término del curso en los semestres de agosto – diciembre 2018 y 2019 y septiembre 2020 – febrero 2021; posteriormente éstos fueron

condensados mediante tablas y se les aplicaron métodos estadísticos para obtener resultados.

- Las pruebas de desempeño obtenidas durante el proceso de intervención fueron examinadas mediante análisis de contenidos para determinar los temas principales donde los estudiantes tenían problemas, este proceso se llevó a cabo al término de cada uno de los cursos impartidos de agosto – diciembre 2018 y 2019. Asimismo, también fueron analizadas las pruebas de desempeño final del curso propedéutico.

3.10 Limitaciones y delimitaciones del estudio

Limitaciones

La manera en que fue percibida la herramienta *Scratch* empleada durante el proceso de estudio fue evaluada por medio de opiniones dadas por los alumnos mediante un cuestionario, por lo que esto es solo un indicador de sus opiniones.

Las conclusiones del estudio fueron validadas únicamente para la población que se estudió.

El curso propedéutico solo fue posible llevarlo a cabo en los semestres correspondientes con agosto a diciembre 2019 y septiembre 2020 a febrero 2021. En 2018 no pudo realizarse dado que este proceso investigativo todavía se encontraba en la etapa de diseño.

Algunas de las fechas consideradas para la tercera intervención tuvieron que ser modificadas durante el desarrollo en función de la pandemia de COVID-19 que nos afectó a nivel mundial.

Delimitaciones

La investigación se llevó a cabo en los semestres escolares de agosto a diciembre de los años 2018 y 2019, y de septiembre 2020 a febrero de 2021, la

variación que se tuvo en el último semestre fue debido a la pandemia del COVID-19 la cual afectó los tiempos académicos en la universidad. En todos los casos la población estuvo formada por los alumnos de nuevo ingreso que se inscribieron a la materia de algoritmia que se imparte en la LIS en la UMT, los cuales proceden de los distintos bachilleratos de la ciudad de Tizimín y sus alrededores en alto porcentaje.

Para la recolección de la información se utilizaron los siguientes instrumentos: una prueba diagnóstica la cual fue aplicada a los alumnos de nuevo ingreso en los años 2019 y 2020, la prueba de desempeño final del curso propedéutico obtenida al finalizar el mismo en los años 2019 y 2020, los resultados finales de los alumnos en la materia fueron obtenidos del SICEI al término del semestre escolar en el mes de diciembre de 2018 y 2019 y febrero de 2021, de igual modo se obtuvieron los resultados de 2016 y 2017 (años en los que no se aplicó ningún proceso adicional a la materia), las pruebas de desempeño obtenidas durante la intervención de los años 2018 y 2019 y un cuestionario tipo Likert el cual fue aplicado a los alumnos al término del semestre, es decir, se aplicó tres veces en diciembre de 2018 y 2019 y febrero de 2021.

Por último, es importante mencionar que el cuestionario que se utilizó fue validado y piloteado, al instrumento se le realizó un análisis de confiabilidad que arrojó un coeficiente de Alfa de Cronbach de 0.902 el cual es un valor adecuado según lo expuesto en George y Mallery (2003); por otra parte la validez de este fue avalada por un grupo de expertos del área de educación y de ingeniería de software de la UMT, los cuales lo revisaron y adecuaron.

3.11 Propuesta: Metodología de aprendizaje usando Scratch

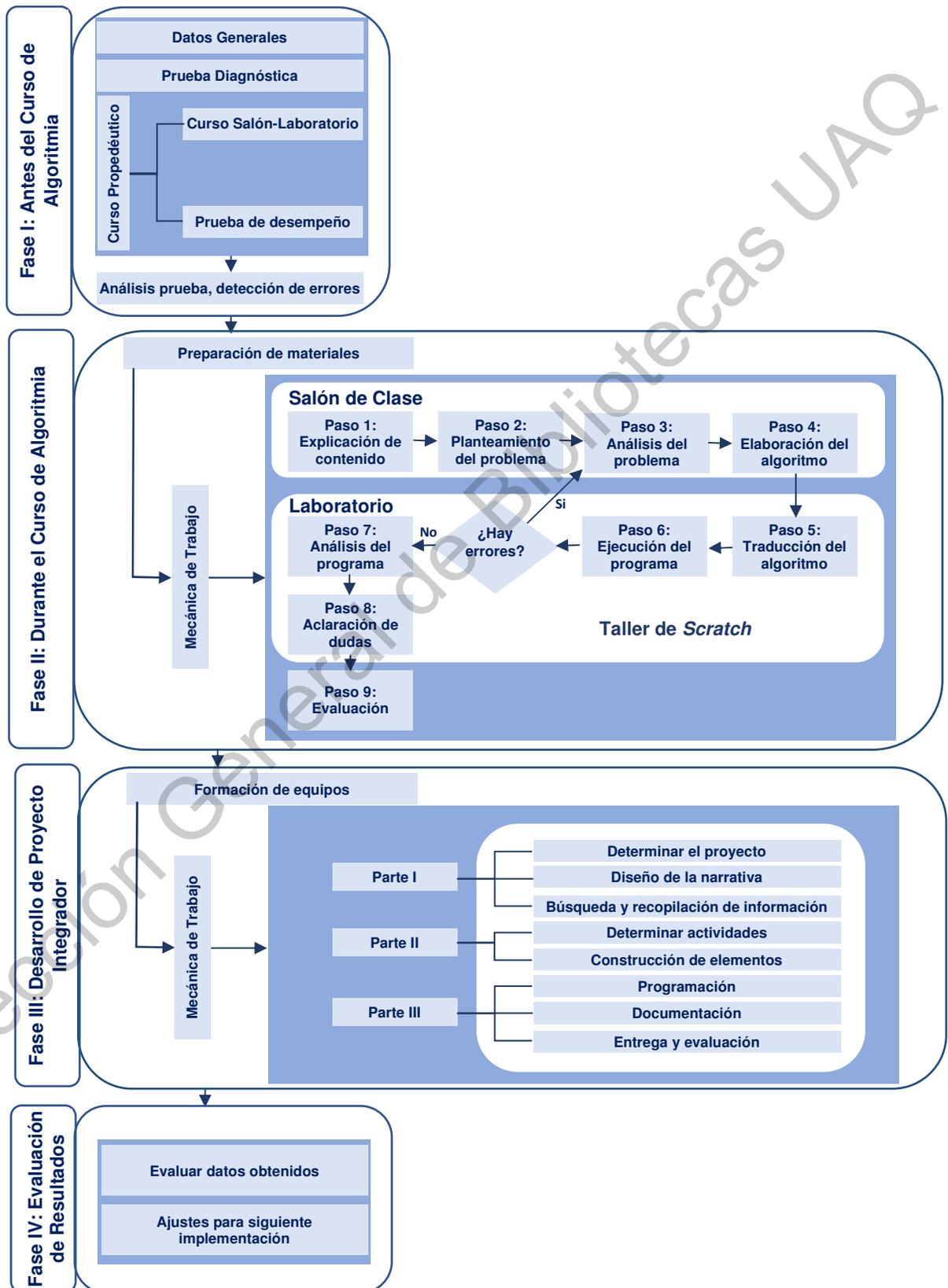
Teniendo en cuenta el objetivo planteado que fue buscar estrategias de apoyo para los alumnos que cursan la asignatura de algoritmia para mejorar su rendimiento académico, se creó y se presenta la siguiente metodología de

aprendizaje usando el lenguaje de programación *Scratch*. Esta metodología se implementó y perfeccionó durante el lapso de tres períodos semestrales, siendo estos los semestres de agosto a diciembre de 2018 y 2019 y septiembre 2020 a febrero 2021.

La metodología propuesta está formada de 4 fases y las 3 primeras a su vez se dividen en varias etapas, el planteamiento desarrollado se puede observar a groso modo en la figura 3.2 y posteriormente se describe con detalle cada una de las partes incluidas.

Dirección General de Bibliotecas UAQ

Figura 3.2 Metodología de aprendizaje usando Scratch.



Fase I: *Antes del curso formal de algoritmia.*

- Previo al inicio del curso formal es deseable conocer los datos generales de los alumnos así como también sus antecedentes en el área de algoritmia y en conceptos básicos del área matemática por medio de una prueba diagnóstica. En la prueba se puede evaluar el uso de operadores aritméticos, relacionales y lógicos, aplicación de la regla de tres, obtención de porcentajes, así como también análisis de problemas; se proponen estos temas que son algunos de los que se han detectado como importantes durante el proceso investigativo.
 - En caso de encontrar deficiencias, una buena estrategia es impartir un curso propedéutico para unificar conocimientos.
 - Se recomienda un curso de 25 horas aproximadamente donde se presenten los conceptos introductorios de principios de programación, una introducción al lenguaje de programación *Scratch* y análisis de problemas.
 - Durante el propedéutico el maestro puede detectar a los alumnos con mayores dificultades para entender la materia, en este caso se les puede apoyar con sesiones de tutorías personalizadas.
 - Al término del propedéutico se recomienda aplicar una prueba de desempeño para verificar el grado de avance que ha tenido el grupo y que estos mismos resultados puedan servir de base para el curso formal de la materia obteniendo de ella los principales errores de los alumnos mediante un análisis de contenido aplicado a la prueba.

Fase II: *Durante el curso de algoritmia.*

- En esta fase la propuesta es seguir dos caminos en simultáneo, por un lado, estar impartiendo el curso de algoritmia en sus tiempos y horarios asignados y

por otro a la par, el alumno debe llevar un taller en el laboratorio de cómputo con el lenguaje de programación *Scratch* considerando tiempo adicional para cursarlo. La propuesta es que los temas vistos en el aula de clase se refuercen en el laboratorio en donde el alumno pueda transcribir los problemas desarrollados en el salón mediante un algoritmo, al lenguaje de programación *Scratch*. Esto además requiere:

- Detección de los temas problemas de los alumnos.
- Creación de ejercicios relacionados con los problemas encontrados, se recomienda la elaboración de compendios de ejercicios y de animaciones que pueden servir de apoyo y guía al estudiante.
- Cada tema estudiado será con el siguiente tratamiento:
 - **Explicación de contenido:** En esta parte el profesor debe explicar las estructuras con la ayuda de una presentación.
 - **Planteamiento del problema:** En el aula de clase el profesor deberá plantear ejercicios que se resolverán por medio de la creación de un algoritmo.
 - **Análisis del problema:** En este punto, el alumno con la guía del profesor deberá de analizar el ejercicio, determinando cuáles son los datos de entrada, los datos de salida y el proceso que se necesita para convertir los datos de entrada en los de salida. Se sugiere dedicar tiempo suficiente a esta etapa de análisis.
 - **Elaboración del algoritmo:** Posteriormente, el alumno con la información generada del punto anterior deberá de construir el algoritmo.
 - **Traducción del algoritmo:** Esta etapa se trabaja en el laboratorio, partiendo del algoritmo generado éste se traduce a un programa utilizando el lenguaje de programación *Scratch* al cual el alumno le puede añadir conceptos de animación para hacerlo más dinámico, permitiéndole explorar la herramienta en busca de mejorar su actividad; por consiguiente, de modo paulatino y de forma activa

puede ir aprendiendo el manejo de la TIC e ir adquiriendo y construyendo los conocimientos básicos necesarios. El profesor debe actuar como un guía en el proceso.

- **Ejecución del programa:** Cuando se finalice el programa el alumno debe proceder con la ejecución, donde el resultado puede ser correr correctamente o con errores (en el caso que en el paso anterior se hubieran codificado mal algunos elementos). Si se detectan errores se regresa al análisis del problema.
- **Análisis del programa:** Se procede a analizar el programa generado junto con el algoritmo con la intención de poder comprender cabalmente cada una de las estructuras empleadas y su proceso de traducción al lenguaje de programación *Scratch*.
- **Aclaración de dudas:** El profesor debe hacer la retroalimentación del programa y el algoritmo generado, aclarando las dudas surgidas en el proceso.
- **Evaluación:** Con base en una rúbrica establecida por el profesor se procede a la evaluación de los resultados.

Fase III: *Desarrollo de proyecto integrador.*

- En esta fase los alumnos organizados mediante grupos de trabajo deberán de desarrollar un proyecto integrador (dividido en tres partes) con la herramienta *Scratch* aplicando el ABP, el cual contemplará todas las estructuras que han sido estudiadas durante la intervención, es importante mencionar que el docente debe de considerar en qué momento empezar con el mismo en función del grado de avance que se tenga con el grupo. El proceso que se propone seguir es el siguiente:
 - Formación de grupos de trabajo, el profesor no debe intervenir, se deja que se integren de acuerdo con su afinidad.

- **Parte I:**
 - **Determinar el proyecto:** El grupo de trabajo seleccionará el tema a desarrollar en el proyecto y presentará un bosquejo de este, el cual debe incluir el uso de las estructuras básicas de algoritmia y adicionalmente dado el contexto se pidió incluir elementos de la cultura maya.
 - **Diseño de la narrativa:** Determinar el tipo de proyecto a desarrollar el cual puede ser una invitación a algún lugar, una tarjeta de felicitación o bien pueden optar por un juego simple; después de la selección se bosqueja la narrativa, es decir la historia a contar. Los equipos deben recibir asesoría por parte del profesor el cual también debe asignar la rúbrica a seguir.
 - **Búsqueda y recopilación de información:** El grupo de desarrollo debe buscar información relacionada con el software a crear de acuerdo con la temática seleccionada, el profesor actuará como orientador y tutor.
- **Parte II:**
 - **Determinar actividades:** Se deben determinar todas las actividades necesarias a desarrollar durante el proyecto, así como también generar un cronograma para vigilar el avance de estas.
 - **Construcción de elementos:** Es el momento de construir todos los objetos y los fondos, así como también de seleccionar los sonidos que se usarán en el desarrollo del proyecto los cuales ya fueron previamente definidos en la parte I.
- **Parte III:**
 - **Programación:** Con toda la información recabada y los elementos creados, en esta parte se debe poner en marcha la creatividad de los estudiantes para el proceso de programación en *Scratch* y el profesor debe dar apoyo con sesiones de supervisión.

- **Documentación:** Concluido el proceso de programación el grupo de desarrollo debe elaborar la documentación necesaria del proyecto donde se explicará con detalle todo lo desarrollado apegado a una rúbrica.
- **Entrega y evaluación:** La entrega final del proyecto debe ser mediante una exposición y defensa de este, durante la cual el alumno será evaluado con apego a las indicaciones preestablecidas.

Fase IV: *Evaluación de resultados.*

- En esta fase el docente deberá de evaluar todos los resultados obtenidos y analizarlos, posteriormente deberá de hacer los ajustes necesarios para implementarlos la siguiente vez que se imparta la asignatura.

4. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados encontrados y se expone la discusión de estos a partir de toda la información obtenida de los instrumentos empleados durante los tres años en los que se llevó a cabo la intervención en los que fue aplicada la metodología de aprendizaje propuesta usando el lenguaje de programación *Scratch*.

4.1 Resultados

A lo largo del proceso de investigación partiendo de la metodología anteriormente descrita se recabó información de diversas fuentes, todas encaminadas a responder la hipótesis planteada para determinar si el hecho de incluir el lenguaje de programación *Scratch* a la asignatura de algoritmia siguiendo una metodología hecha para tal efecto podría elevar el rendimiento académico de los alumnos que cursan la asignatura, de aquí que se consideró importante obtener información del curso propedéutico impartido antes del inicio formal de la asignatura, de las diversas pruebas de desempeño aplicadas a los alumnos durante todo el proceso de intervención, de las actas de examen y del cuestionario de evaluación final. A continuación, se presenta toda la información vinculada con cada una de estas fuentes.

Curso propedéutico

El curso propedéutico se planeó para llevarse a cabo en dos ocasiones, una antes del inicio del semestre agosto – diciembre 2019 y la otra antes del semestre septiembre 2020 – febrero 2021; en ambos casos se contó con la participación de 35 estudiantes del total de 46 aceptados en estos dos ingresos.

El curso se organizó de tal manera para que al inicio los alumnos presentaran una ficha de datos generales y una prueba diagnóstica y al finalizar todas las sesiones proyectadas respondieran una prueba de desempeño.

A pesar de que el propedéutico solo se llevó a cabo el segundo y el tercer año de la intervención, la ficha de datos generales se pudo aplicar durante los tres años de esta. Como se mencionó con anterioridad el total de alumnos aceptados en la LIS durante este tiempo fue de 66 y de este total 56 respondieron la ficha de datos generales (unos el primer día del propedéutico y otros el primer día del curso de algoritmia), en cualquier caso, la información recabada sirvió para conocer las características básicas de los alumnos que se inscribieron a la licenciatura en ingeniería de software.

De los resultados generados de la ficha de datos generales aplicada (independientemente de cuando la presentaron), se determinó que los estudiantes provienen de diversos bachilleratos de la ciudad de Tizimín y sus comisarías, pueblos circunvecinos y otros de lugares más distantes como son uno de Mérida, tres de Valladolid, tres del estado de Quintana Roo, uno del estado de Campeche, uno del estado de Veracruz y uno del Estado de México (tabla 4.1); cabe señalar que todos llegan a la UMT con diversos perfiles y conocimientos procedentes de 7 sistemas de bachillerato diferentes, asimismo, también se pudo precisar que son pocos los alumnos que ingresan a la LIS con conocimientos básicos sobre principios de programación.

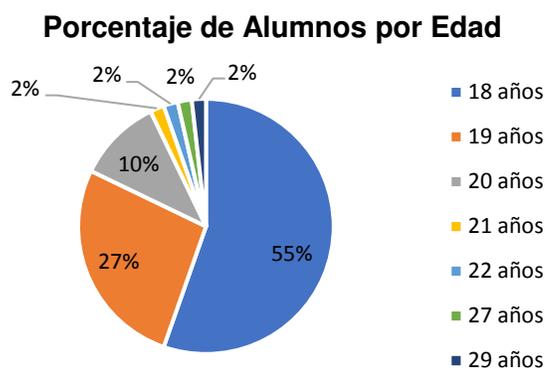
Tabla 4.1 Bachillerato de procedencia de los alumnos de nuevo ingreso a la LIS.

| Escuela | Cantidad total de alumnos de algoritmia | |
|------------------------------------|--|-----------|
| | Períodos: Ago – Dic 2018 / Ago – Dic 2019 / Sep 2020 – Feb 2021 | |
| <i>Conalep</i> | | |
| Conalep Tizimín | | 12 |
| Conalep Cancún | | 2 |
| Conalep Champotón | | 1 |
| <i>CBTA</i> | | |
| CBTA No. 14 Tizimín | | 3 |
| CBTA No. 36 Veracruz | | 1 |
| <i>CECyTEY</i> | | |
| CECyTEY Panabá | | 2 |
| CECyTEY Espita | | 1 |
| <i>Colegio de Bachilleres</i> | | |
| Cobach QRoo | | 1 |
| Cobay Dzonot Aké | | 1 |
| Cobay Tixcancal | | 2 |
| Cobay Tizimín | | 16 |
| Cobay Sucilá | | 6 |
| Cobay Calotmul | | 2 |
| Cobay Valladolid | | 2 |
| <i>Incorporadas a Sistema UADY</i> | | |
| Incorporada UADY Mérida | | 1 |
| Incorporada UADY Valladolid | | 1 |
| <i>UVM Lago de Guadalupe</i> | | 1 |
| <i>Prepa abierta</i> | | 1 |
| Total | | 56 |

Fuente: Elaboración propia (2020).

De la misma ficha de datos generales se obtuvo que el 76.79% de los alumnos que se inscribieron en el lapso de estos tres años fueron hombres y el 23.21% mujeres, otro dato importante es que el 53.57% llegan a la UMT procedentes de los dos principales bachilleratos de Tizimín los cuales son el Conalep y el Cobay y en cuanto al rango de edades (figura 4.1) este fluctúa entre 18 y 29 años siendo el promedio de 18.96 años.

Figura 4.1 Gráfica de las edades de los alumnos de nuevo ingreso a la LIS.



Fuente: Elaboración propia (2020).

El curso propedéutico solo fue cursado por 35 alumnos en las dos ocasiones en que se llevó a cabo, sin embargo, la prueba diagnóstica que fue presentada el primer día del curso la resolvieron un total de 40 estudiantes, lo que nos indica que 5 de ellos no continuaron con el curso por diversos motivos.

La prueba diagnóstica estuvo dividida en cinco secciones las cuales se detallan como sigue:

- **Sección I:** Elaboración de un algoritmo con palabras.
- **Sección II:** Resolución de operaciones aritméticas.
- **Sección III:** Resolución de operaciones relacionales.
- **Sección IV:** Análisis de un problema.
- **Sección V:** Construcción de un diagrama de flujo.

La primera sección consistió en la elaboración de un algoritmo de una actividad simple de la vida cotidiana, a este respecto todos lograron hacerlo, aunque con diferentes niveles de detalle, cabe mencionar que en algoritmia el nivel de

detalle es un punto muy importante y es precisamente lo que se quería evaluar al incluir este ejercicio.

En la segunda sección se resolvieron dos operaciones aritméticas en las que los alumnos debían tener presente el orden de precedencia de los operadores, así como las reglas de asociatividad, esta sección fue incluida dada la relevancia de estos temas matemáticos en algoritmia, los resultados se pueden apreciar en la tabla 4.2.

Tabla 4.2 Respuestas a la segunda sección de la prueba diagnóstica.

| <i>Aciertos</i> | Prueba diagnóstica |
|-----------------|------------------------------|
| | <i>Porcentaje de alumnos</i> |
| 0 | 7.89% |
| 1 | 15.79% |
| 2 | 76.32% |

Fuente: Elaboración propia (2020).

En lo que respecta a la tercera sección se resolvieron cinco ejercicios los cuales empleaban los operadores relacionales, al igual que el apartado anterior esta sección fue incluida dada la relevancia de ese tema matemático en algoritmia, los resultados se pueden apreciar en la tabla 4.3.

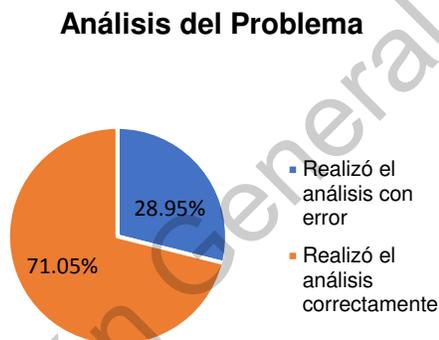
Tabla 4.3 Respuestas a la tercera sección de la prueba diagnóstica.

| <i>Aciertos</i> | Prueba diagnóstica |
|-----------------|------------------------------|
| | <i>Porcentaje de alumnos</i> |
| 0 | 0% |
| 1 | 0% |
| 2 | 2.63% |
| 3 | 42.11% |
| 4 | 36.84% |
| 5 | 18.42% |

Fuente: Elaboración propia (2020).

En la cuarta sección se les pidió realizar el análisis de un problema simple en función de tres partes de importancia: sus datos de entrada, sus datos de salida y el proceso. En específico se les pidió a los alumnos encontrar el promedio de un grupo de cinco materias y como resultado de este ejercicio, aproximadamente una tercera parte 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 hacer la lectura del nombre de las materias (lo cual resultó irrelevante en el ejercicio planteado), el resultado lo daban en porcentaje cuando lo que se solicitó fue un promedio o bien no especificaron el resultado; el resto de los alumnos lo realizaron correctamente (figura 4.2).

Figura 4.2 Respuestas a la sección cuatro de la prueba diagnóstica (análisis de problema).



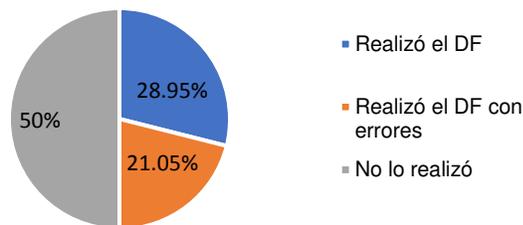
Fuente: Elaboración propia (2020).

En la última sección se les pidió obtener el diagrama de flujo del ejercicio de la sección cuatro, cabe mencionar que éste 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 el área de programación de computadoras, en la figura 4.3 se puede ver el resultado que se obtuvo.

Figura 4.3 Respuestas a la sección cinco de la prueba diagnóstica (DF).

Elaboración del Diagrama de Flujo



Fuente: Elaboración propia (2020).

La segunda prueba aplicada se llevó a cabo al finalizar el curso propedéutico, fue presentada por 34 de los 35 alumnos que lo cursaron (el alumno que no presentó no acudió a la sesión el día en que se realizó) y consistió en una prueba de desempeño la cual se utilizó para contrastar resultados con el diagnóstico y para medir el avance de los alumnos, esta prueba estuvo dividida en tres partes como sigue:

- Resolución de operaciones aritméticas.
- Identificación de conceptos básicos de principios de programación.
- Construcción de diagramas de flujo.

En el apartado de resolución de operaciones aritméticas se les pidió resolver dos ejercicios en los que se debía cuidar el orden de precedencia de los operadores y las reglas de asociatividad; la tabla 4.4 presenta los resultados obtenidos y es

importante decir que en esta sección fue considerado el proceso de desarrollo del ejercicio y no solo el resultado final.

Tabla 4.4 Respuestas al apartado resolución de operaciones aritméticas de la prueba de desempeño final.

| <i>Aciertos</i> | Prueba de desempeño |
|-----------------|------------------------------|
| | <i>Porcentaje de alumnos</i> |
| 1 | 6.25% |
| 1.5 | 12.50% |
| 2 | 81.25% |

Fuente: Elaboración propia (2020).

En el apartado de identificación de conceptos básicos de principios de programación se plantearon 4 preguntas de opción múltiple y la tabla 4.5 nos resume los resultados obtenidos.

Tabla 4.5 Respuestas al apartado de identificación de conceptos básicos de principios de programación de la prueba de desempeño final.

| <i>Aciertos</i> | Prueba de desempeño |
|-----------------|------------------------------|
| | <i>Porcentaje de alumnos</i> |
| 0 | 0% |
| 1 | 0% |
| 2 | 3.125% |
| 3 | 18.75% |
| 4 | 78.125% |

Fuente: Elaboración propia (2020).

En el apartado de construcción de diagramas de flujo se les pidió a los alumnos que resolvieran 4 problemas en los que debían hacer primero el análisis de los mismos.

El ejercicio 1 el 96.88% lo tuvo correctamente y el 3.12% no lo pudo resolver, el ejercicio 2 el 100% de los alumnos lo realizó sin errores, el ejercicio 3 el 89.06% lo tuvo bien y el 10.94% no lo hizo adecuadamente y por último el ejercicio 4 el 82.81% lo tuvo perfecto y el 17.19% lo realizó con diversos fallos.

Análisis de contenido

Uno de los aspectos relevantes durante este proceso investigativo fue conocer con exactitud en qué temas los estudiantes presentaban la mayor cantidad de errores en el proceso de análisis y construcción de los diagramas de flujo y esta información se desconocía al inicio ya que el docente que imparte algoritmia solo tenía una idea vaga sobre en qué aspectos fallaban los alumnos; dada esta situación se decidió llevar a cabo un análisis de contenido aplicado a las pruebas de desempeño generadas durante el propedéutico en los períodos en que fue impartido (los cuales se citaron con anterioridad) y a las pruebas obtenidas durante el curso formal de algoritmia de las dos primeras intervenciones es decir de los semestres agosto – diciembre 2018 y 2019.

En primera instancia se aplicó el análisis de contenido a las pruebas de desempeño finales del curso propedéutico y se detectaron cuáles fueron los errores más frecuentes. Se evaluaron un total de 34 pruebas y de cada una se analizaron los resultados de 6 ejercicios, esto nos generó un total de 204 ejercicios revisados, la codificación de errores obtenida se puede ver en la tabla 4.6.

Tabla 4.6 Codificación del análisis de contenido de las pruebas de desempeño del propedéutico.

| 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 leerlas | | | | | | | | | | | | | | | | | | | | |
| Nombres de variables incorrectos | | | | | | | | | | | | | | | | | | | | |
| Falta de resultados en datos de salida | | | | | | | | | | | | | | | | | | | | |
| Lógica general del problema | | | | | | | | | | | | | | | | | | | | |
| Con operadores aritméticos | | | | | | | | | | | | | | | | | | | | |
| Con operadores relacionales | | | | | | | | | | | | | | | | | | | | |

Fuente: Elaboración propia (2020).

Posteriormente, se realizó el análisis de contenido de las pruebas de desempeño, generadas durante la implementación del curso formal de algoritmia y los resultados ayudaron a tener una evidencia justificada más a fondo sobre dónde los alumnos presentaban la mayor cantidad de problemas en la materia de algoritmia, en este análisis se revisaron un total 833 ejercicios distribuidos en 4 pruebas de desempeño; esta información se puede apreciar en la tabla 4.7.

Tabla 4.7 Pruebas de desempeño analizadas.

| Número de prueba de desempeño | Total de alumnos que presentaron | 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 |
| Total | | | 833 |

Fuente: Elaboración propia (2020).

La tabla 4.8, presenta una relación de las veces que ocurrieron cada uno de los errores detectados; es importante indicar que no se contabilizaron como error los ejercicios no resueltos completamente y se determinó que la cantidad de estos corresponde a 49 ejercicios del total de 833.

Tabla 4.8 Errores en las pruebas de desempeño, agosto - diciembre de 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 desición | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 erróneo del DF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Fuente: Elaboración propia (2020).

Actas de examen

Otro aspecto relevante en este proceso investigativo fue tratar de elevar el rendimiento académico de los alumnos en la materia de algoritmia, razón por la cual

se consideró necesario analizar la información recogida con las actas de examen de la asignatura.

Las actas de examen fueron obtenidas del SICEI de la UADY al término de los períodos escolares y se analizaron los resultados de los semestres agosto – diciembre 2018, 2019 y septiembre 2020 – febrero 2021, los cuales fueron los ciclos en los que se realizó la intervención con la metodología propuesta. También es importante mencionar que para establecer comparaciones se obtuvieron las actas de la misma asignatura de los cursos donde fue impartida algoritmia del modo tradicional sin la inclusión de ninguna herramienta tecnológica siendo estos los semestres de agosto – diciembre 2016 y 2017.

En esta parte del documento se describe lo encontrado en las actas de examen recabadas durante estos 5 cursos y es importante mencionar que en este tiempo en que se impartió algoritmia el profesor que dio las clases siempre fue el mismo, los temas incluidos en la planeación didáctica de la materia no han presentado cambios y los alumnos han procedido de un contexto variado de bachilleratos con planes educativos diferentes pertenecientes a diversos sistemas educativos.

Los resultados que se presentan se centran en tres aspectos (medidas o estadísticos) fundamentales para poder medir el efecto que tiene esta propuesta en el proceso enseñanza aprendizaje de la asignatura, el primer aspecto 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 y con la implementación de *Scratch*.

Los resultados obtenidos en cuanto al promedio de aprovechamiento final de los 5 grupos son los siguientes: para los cursos impartidos en 2016 y 2017 con el modo tradicional se obtuvo un promedio de 75.75 puntos y para los cursos impartidos en 2018, 2019 y 2020 - 2021 con el uso de *Scratch* se obtuvo un

promedio de 85.12 puntos; con estos dos resultados se observa un incremento de 9.37 puntos en los grupos en los que se empleó *Scratch* respecto del método tradicional.

En cuanto al índice de aprobados y reprobados, en la tabla 4.9 se encuentran los resultados obtenidos en estos 5 cursos identificando los porcentajes de los alumnos con la aplicación de *Scratch* y con el método tradicional.

Tabla 4.9 Alumnos aprobados-reprobados en algoritmia 2016 a 2021.

| Estrategia usada | Total de alumnos | Aprobados | Reprobados |
|-------------------------|-------------------------|------------------|-------------------|
| Tradicional | 24 | 70.83% | 29.17% |
| <i>Scratch</i> | 66 | 92.42% | 7.58% |

Fuente: Elaboración propia (2021).

Se observó que se obtuvo un mayor índice de aprobados usando *Scratch* (92.42%), en comparación con el método tradicional (70.83%); teniendo un incremento del 21.59%.

Por último, en cuanto al nivel de dominio, usando el método tradicional 54.17% tuvieron un nivel entre satisfactorio y sobresaliente mientras que con el uso de *Scratch* este valor fue del 78.79%; al igual que con los dos resultados anteriores se ve una mejoría, en este caso del 24.62%.

De acuerdo con el análisis anterior, en los tres aspectos considerados para poder medir el impacto que tiene la metodología de aprendizaje propuesta se observó un incremento cuando se usa la estrategia *Scratch* sobre el método tradicional.

A continuación, para corroborar que los incrementos observados en los aspectos estudiados son estadísticamente significativos, se presentan diferentes pruebas estadísticas que fueron implementadas, en todos los casos que se detallan

el software utilizado para realizar estas pruebas fue RStudio y se consideró adecuado trabajar con un nivel de significancia del 5% ($\alpha = 0.05$).

Primero se trabajó en comprobar si el promedio de aprovechamiento cuando se usa *Scratch* es significativamente diferente de cuando no se usa, en función de esto se planteó la siguiente hipótesis:

$$H01: \mu_1 = \mu_2$$

$$H11: \mu_1 \neq \mu_2$$

Donde:

μ_1 = Promedio de calificaciones cuando se usa el método tradicional.

μ_2 = Promedio de calificaciones cuando se usa *Scratch*.

Se aplicó el diseño experimental análisis de varianza de una vía (anova1) y se analizaron los residuales para determinar si los datos de estudio seguían un comportamiento normal como dicta la metodología, un resultado satisfactorio permitiría usar la teoría de la estadística paramétrica para hacer inferencia con base a los resultados. Desafortunadamente al usar el test de Shapiro-Wilk a los residuales del modelo de análisis de varianza de una vía se obtuvieron resultados que indicaron que los residuales no son normales, esto se puede observar en la figura 4.4.

Figura 4.4 Prueba Shapiro-Wilk para residuales.

```
          Df Sum Sq Mean Sq F value Pr(>F)
as.factor(metodos) 1    1546   1545.6   7.791 0.00644 **
Residuals        88   17458    198.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> shapiro.test(anova$residuals)

      Shapiro-Wilk normality test

data:  anova$residuals
W = 0.89223, p-value = 1.849e-06
```

Fuente: Elaboración propia (2021).

Por lo tanto, el diseño experimental análisis de varianza de una vía (anova1) no resultó adecuado, de esta forma se tuvo que recurrir a la estadística no paramétrica para comprobar la hipótesis, para esto la prueba más adecuada fue el test no paramétrico Kruskal-Wallis, en la figura 4.5 se muestran los resultados.

Figura 4.5 Prueba Kruskal-Wallis a los promedios de las calificaciones con los 2 métodos.

```
> kw<-kruskal.test(calificaciones,metodos)
> kw

      Kruskal-Wallis rank sum test

data:  calificaciones and metodos
Kruskal-Wallis chi-squared = 5.729, df = 1, p-value = 0.01669
```

Fuente: Elaboración propia (2021).

En segundo lugar, se procedió a analizar otro aspecto que se consideró importante que fue el rendimiento de los alumnos con base al índice de aprobados y reprobados, en función de esto se planteó la siguiente hipótesis:

H02: Los índices de aprobados y reprobados son independientes del método aplicado.

H12: Los índices de aprobados y reprobados no son independientes del método aplicado.

Para ello, se consideró adecuado la implementación de la prueba no paramétrica de Fisher's Exacta y se usó la información ya presentada en la tabla 4.9. Los resultados de la aplicación de esta prueba se pueden apreciar en la figura 4.6.

Figura 4.6 Prueba Fisher's Exacta a los aprobados-reprobados con los 2 métodos.

```
                indice_aprobacion
metodos         A  R
con scratch    61  5
sin scratch    17  7
> fisher.test(Table.est)

Fisher's Exact Test for Count Data

data: Table.est
p-value = 0.01346
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.175573 22.344740
sample estimates:
odds ratio
 4.911605
```

Fuente: Elaboración propia (2021).

Por último, se analizó otro aspecto que se consideró importante que es el rendimiento de los alumnos atendiendo al nivel de dominio con base al modelo que sigue la UADY, considerando en este aspecto el porcentaje de estudiantes con nivel de dominio satisfactorio o sobresaliente; de acuerdo con esto se planteó la siguiente hipótesis.

H03: El índice de alumnos con un nivel de dominio satisfactorio o sobresaliente y el índice de alumnos con nivel de dominio suficiente o no acreditado son independientes del método aplicado.

H13: El índice de alumnos con un nivel de dominio satisfactorio o sobresaliente y el índice de alumnos con nivel de dominio suficiente o no acreditado no son independientes del método aplicado.

Para lo anterior se usó de nuevo la prueba no paramétrica Fisher's Exacta, usando la información que se presentó en la tabla 4.9 y se obtuvieron los resultados que se aprecian en la figura 4.7.

Figura 4.7 Prueba Fisher-Exacta de acuerdo con el nivel de dominio.

```
      niveles_mefi
metodos satisfactorio suficiente
con scratch      52      14
sin scratch      13      11
> fisher.test(Table2.est)

Fisher's Exact Test for Count Data

data: Table2.est
p-value = 0.03233
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.021394 9.490332
sample estimates:
odds ratio
 3.097883
```

Fuente: Elaboración propia (2021).

Adicionalmente y como beneficio de la implementación de esta estrategia se observó que el promedio de calificación de los cursos impartidos usando *Scratch* fue de 85.12 puntos y bajo los estándares de la UADY esta calificación se encuentra en el nivel satisfactorio, en función de esto se planteó la siguiente hipótesis:

$$H04: \mu_2=80$$

$$H14: \mu_2>80$$

Donde:

μ_2 = Promedio de calificaciones cuando se usa *Scratch*.

Primeramente, se procedió a verificar si los datos tenían una distribución normal, para luego decidir sobre el tipo de test a aplicar para probar la hipótesis, para esto se aplicó el test de Shapiro-Wilk al conjunto de calificaciones obtenidas con la implementación de *Scratch* la figura 4.8 presenta los resultados.

Figura 4.8 Prueba Shapiro-Wilk con las calificaciones finales de algoritmia.

```
shapiro-wilk normality test
data: calificaciones_con_scratch
W = 0.88382, p-value = 1.589e-05
```

Fuente: Elaboración propia (2021).

En función de lo obtenido de la prueba de Shapiro-Wilk, se puede concluir que las observaciones a estudiar no tienen un comportamiento normal, por lo que se decidió emplear la prueba no paramétrica del test de signos de Wilcoxon para probar la hipótesis planteada, los resultados se muestran en la figura 4.9.

Figura 4.9 Prueba Wilcoxon con las calificaciones finales de algoritmia.

```
wilcoxon signed rank test with continuity correction
data: calificaciones_con_scratch
V = 1687.5, p-value = 7.488e-06
alternative hypothesis: true location is greater than 79
```

Fuente: Elaboración propia (2021).

Cuestionario

En cuanto al instrumento del cuestionario (anexo 1), este estuvo dividido en tres secciones y sirvió para evaluar aspectos sobre la actitud de los estudiantes ante la herramienta *Scratch*, el impacto de esta y si ayudó a los alumnos a entender los conceptos básicos de programación y en consecuencia un mejor puntaje en la materia. Se analizaron los resultados generados de la aplicación de este a los grupos de los semestres impartidos en los periodos de agosto – diciembre 2018, agosto – diciembre 2019 y septiembre 2020 – febrero 2021; fue cumplimentado por un total de 59 alumnos de los cuales 15 corresponden al curso de 2018, 18 al curso

de 2019 y 26 al curso 2020 - 2021; a 33 estudiantes de la primera y la segunda intervención se les presentó el cuestionario en formato físico y a los otros 26 en formato digital por medio de un formulario de Google, es importante mencionar que este cambio se originó a raíz de la pandemia que obligó a realizarlo en forma no presencial; en cualquier caso fue respondido en forma individual, de modo anónimo y el proceso se llevó a cabo al término del semestre el último día de clase de la materia algoritmia. El cuestionario tuvo como opciones de respuesta: Muy en desacuerdo, En Desacuerdo, Indiferente, De Acuerdo y Muy de acuerdo, para todos los casos que se presentan se destacan las opciones de respuestas de Muy de Acuerdo y De Acuerdo, como las adecuadas en este proceso investigativo.

En la tabla 4.10 se presentan los estadísticos descriptivos para las 12 interrogantes del cuestionario que determina la actitud de los alumnos frente al uso de *Scratch* obtenidos después de la inclusión de esta herramienta. Para efectos de esta investigación se consideraron las respuestas Muy de Acuerdo y De Acuerdo actitudes favorables en el uso de *Scratch* en algoritmia por esta razón también se incluye en la tabla 4.10 el porcentaje de los alumnos que seleccionaron estas opciones en cada reactivo.

Tabla 4.10 Estadísticos del cuestionario de actitud hacia el uso de la TIC Scratch.

| Actitud ante el uso de la TIC Scratch | | | | | | | |
|--|------------|------------|-------------|----------------|------------------------|-----------------------|-------------------------|
| <i>Reactivo</i> | <i>Min</i> | <i>Max</i> | <i>Moda</i> | <i>Mediana</i> | <i>MDA^a</i> | <i>DA^a</i> | <i>Suma^a</i> |
| 1 | 3 | 5 | 5 | 5 | 54.2% | 40.7% | 94.9% |
| 2 | 3 | 5 | 5 | 4 | 45.8% | 44.1% | 89.8% |
| 3 | 2 | 5 | 5 | 5 | 64.4% | 32.2% | 96.6% |
| 4 | 3 | 5 | 5 | 4 | 47.5% | 33.9% | 81.4% |
| 5 | 4 | 5 | 5 | 5 | 67.8% | 32.2% | 100% |
| 6 | 3 | 5 | 4 | 4 | 44.1% | 54.2% | 98.3% |
| 7 | 3 | 5 | 5 | 5 | 66.1% | 28.8% | 94.9% |
| 8 | 3 | 5 | 5 | 5 | 50.8% | 35.6% | 86.4% |
| 9 | 2 | 5 | 4 | 4 | 42.4% | 44.1% | 86.4% |
| 10 | 3 | 5 | 5 | 5 | 55.9% | 37.3% | 93.2% |
| 11 | 2 | 5 | 5 | 5 | 59.3% | 37.3% | 96.6% |
| 12 | 2 | 5 | 5 | 5 | 67.8% | 27.1% | 94.9% |

^aMDA = Muy de Acuerdo, DA = De Acuerdo, Suma = Porcentaje de MDA más DA

Fuente: Elaboración propia (2021).

En la tabla 4.11 se presentan los estadísticos descriptivos para las 10 interrogantes del cuestionario que determina la opinión de los alumnos respecto a las TIC después de la inclusión de la herramienta *Scratch*; de igual modo se destacan las respuestas en las opciones de Muy de Acuerdo y De Acuerdo como las favorables para efectos de esta investigación, por esta razón también se incluye en la tabla 4.11 el porcentaje de los alumnos que seleccionaron estas opciones en cada interrogante.

Tabla 4.11 Estadísticos del cuestionario de opinión sobre las TIC.

| Opinión sobre las TIC | | | | | | | |
|-----------------------|-----|-----|------|---------|------------------|-----------------|-------------------|
| Reactivo | Min | Max | Moda | Mediana | MDA ^a | DA ^a | Suma ^a |
| 1 | 3 | 5 | 5 | 5 | 55.9% | 40.7% | 96.6% |
| 2 | 3 | 5 | 5 | 5 | 64.4% | 32.2% | 96.6% |
| 3 | 3 | 5 | 4 | 4 | 42.4% | 44.1% | 86.4% |
| 4 | 2 | 5 | 4 | 4 | 40.7% | 44.1% | 84.7% |
| 5 | 3 | 5 | 4 | 4 | 45.8% | 45.8% | 91.5% |
| 6 | 2 | 5 | 5 | 4 | 40.7% | 35.6% | 76.3% |
| 7 | 2 | 5 | 5 | 4 | 42.4% | 35.6% | 78.0% |
| 8 | 2 | 5 | 5 | 5 | 50.8% | 35.6% | 86.4% |
| 9 | 3 | 5 | 5 | 5 | 57.6% | 40.7% | 98.3% |
| 10 | 2 | 5 | 5 | 4 | 49.2% | 32.2% | 81.4% |

^aMDA = Muy de Acuerdo, DA = De Acuerdo, Suma = Porcentaje de MDA más DA

Fuente: Elaboración propia (2021).

En la tabla 4.12 se presentan los estadísticos descriptivos para las 11 interrogantes del cuestionario que determina la opinión de los alumnos respecto al uso de *Scratch* obtenidos después de la inclusión de esta herramienta; de igual modo se destacan las respuestas en las opciones de Muy de Acuerdo y De Acuerdo, por esta razón también se incluye en la tabla 4.12 el porcentaje de los alumnos que las seleccionaron en cada interrogante.

Tabla 4.12 Estadísticos del cuestionario de opinión de los alumnos hacia Scratch.

| Opinión sobre la TIC Scratch | | | | | | | |
|-------------------------------------|------------|------------|-------------|----------------|------------------------|-----------------------|-------------------------|
| <i>Reactivo</i> | <i>Min</i> | <i>Max</i> | <i>Moda</i> | <i>Mediana</i> | <i>MDA^a</i> | <i>DA^a</i> | <i>Suma^a</i> |
| 1 | 2 | 5 | 4 | 4 | 44.1% | 47.5% | 91.5% |
| 2 | 2 | 5 | 5 | 4 | 44.1% | 40.7% | 84.7% |
| 3 | 4 | 5 | 5 | 5 | 72.9% | 27.1% | 100% |
| 4 | 3 | 5 | 5 | 5 | 83.1% | 15.3% | 98.3% |
| 5 | 3 | 5 | 5 | 5 | 69.5% | 25.4% | 94.9% |
| 6 | 2 | 5 | 5 | 5 | 52.5% | 37.3% | 89.8% |
| 7 | 3 | 5 | 4 | 4 | 45.8% | 47.5% | 93.2% |
| 8 | 3 | 5 | 5 | 4 | 49.2% | 37.3% | 86.4% |
| 9 | 2 | 5 | 5 | 4 | 39.0% | 37.3% | 76.3% |
| 10 | 3 | 5 | 5 | 5 | 64.4% | 33.9% | 98.3% |
| 11 | 3 | 5 | 5 | 5 | 62.7% | 28.8% | 91.5% |

^aMDA = Muy de Acuerdo, DA = De Acuerdo, Suma = Porcentaje de MDA más DA

Fuente: Elaboración propia (2021).

4.2 Discusión

En esta sección se discuten los datos encontrados, los cuales ya fueron presentados en la sección anterior y de nuevo se hace una diferenciación en función de las partes ya consideradas anteriormente.

Como se ha citado, durante el curso propedéutico se aplicaron dos pruebas una diagnóstica al inicio de este y una de desempeño al final. En la información presentada se determinó que los alumnos que llegan a la UMT para estudiar la LIS provienen de siete bachilleratos diferentes, los cuales tienen diversos programas de estudios, por lo que, el hecho de impartir un curso propedéutico como punto de partida en la licenciatura, específicamente para el área de algoritmia se consideró una buena estrategia para indagar en los conocimientos previos de los alumnos, así como también para unificarlos. En cuanto a la discusión de los resultados después de la aplicación de la prueba diagnóstica llevada a cabo en los cursos de agosto – diciembre 2019 y septiembre 2020 – febrero 2021 se obtuvo lo siguiente.

- A pesar de que todos los seres humanos tienen una idea general de cómo poder resolver un problema simple de la vida cotidiana, por ejemplo cepillarse los dientes, es común para los principiantes en el área algorítmica no considerar con

precisión el nivel de detalle y obviar algunos pasos dándolos por sentado, lo cual en un entorno físico puede ser perfectamente comprensible por todos, no obstante, en el ámbito computacional no se podría llegar a la solución de un problema adecuadamente, esto se vio reflejado en la primera sección de la prueba donde solamente dos personas lograron concluir el ejercicio al 100%.

- En cuanto a la resolución de operaciones aritméticas un alto porcentaje de los alumnos lograron responder correctamente los ejercicios, lo cual fue un punto importante ya que les permitió emplear este conocimiento con el que la mayoría ya contaba en la resolución de los problemas, sin embargo, fue necesario considerar durante el curso ese aproximadamente 25% de alumnos que no lograron las competencias deseadas.
- En la sección de ejercicios con operadores relacionales, si bien es cierto, solo un 18% de los alumnos logró resolver correctamente todos los ejercicios, por los resultados arrojados se observa que la mayoría de ellos tiene una idea bastante clara de cómo proceder, a pesar de esto, el resultado de esta sección fue un referente al momento de estudiar el tema tanto en el propedéutico como durante el curso de algoritmia.
- En cuanto al análisis de un problema, cabe destacar que el ejercicio que se propuso es de los más simples que se ven en algoritmia, se optó por este porque es un problema básico de la vida cotidiana, que aun cuando no tengan conocimientos del área algorítmica deberían poder resolverlo sin problema. Del resultado se deduce que aun cuando la mayoría de los alumnos lo pudo resolver, se considera necesario dedicar más tiempo a la sección de análisis del problema enfocado al desglose en datos de entrada, proceso y salida para ayudar a los alumnos a entender adecuadamente los planteamientos que en ellos se incluyan, como se pudo apreciar en el análisis de contenido de las pruebas la falta de comprensión lectora es un problema que los aqueja y en consecuencia se puso especial interés tanto durante el propedéutico como con el curso de algoritmia.

- En la última sección los alumnos debieron elaborar un diagrama de flujo y a este respecto el 71.05% no lo logró hacer, este hecho confirma el planteamiento inicial que llevó a incluir este ejercicio en la prueba, los alumnos que llegan a la UMT vienen con escasos y/o nulos conocimientos del área algorítmica.

En síntesis, como se puede observar los resultados de la prueba diagnóstica presentada y analizada de modo global, nos lleva a afirmar 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, lo cual será favorable en nuestro contexto. Es importante observar que los ejercicios de las primeras secciones se resolvieron en un nivel medio y en lo que respecta al ejercicio del análisis aun cuando lo obtuvo bien la mayoría de los estudiantes, solo el 28.95% de los mismos lo pudo transformar al diagrama de flujo considerando que lo que se pidió fue uno de los ejercicios más sencillos de realizar.

Después de que los estudiantes concluyeron el curso propedéutico, se les aplicó una prueba de desempeño. En cuanto a la discusión de los resultados de esta parte se expone lo siguiente.

- En la primera parte se les pidió evaluar ejercicios con operaciones aritméticas, los resultados obtenidos manifiestan un ligero avance con respecto a la prueba diagnóstica y se destaca que ningún alumno obtuvo todo incorrecto en los ejercicios planteados. En resumen, se observan resultados positivos situación que puede ser aún mejor con más tiempo y práctica lo que se buscó fortalecer durante el curso de algoritmia.
- En el apartado de conceptos básicos, se concluye que en esta sección se obtuvieron buenos resultados ya que previo a este curso los alumnos tenían un desconocimiento general de la asignatura, el 78.13% de los alumnos respondió correctamente todas las interrogantes.
- En la sección final, se presentaron ejercicios de construcción de diagramas de flujo a los cuales los estudiantes les tenían que hacer primero el análisis, el grado

de complejidad fue de menor a mayor y se enfocaron a los temas de estructura secuencial y alternativa simple, doble y múltiple, además incluían el uso de los operadores aritméticos y relacionales. Con los resultados obtenidos se aprecia el avance de los alumnos ya que las calificaciones de cada prueba se ubicaron en los niveles satisfactorio y sobresaliente de la UADY lo cual resultó adecuado para fines de esta investigación.

El promedio general de calificación de la prueba de desempeño final fue de 92.39 puntos y esto es equivalente a un resultado sobresaliente de acuerdo con los niveles de dominio de la UADY en su proceso de evaluación (UADY, 2012) e indica un resultado adecuado en este curso propedéutico de nivelación, por lo que resultó ser una buena estrategia con los alumnos de nuevo ingreso de la LIS.

Por otro lado, al inicio de la intervención se tenía una idea vaga de los principales errores que tienen los alumnos durante el proceso de creación de sus diagramas de flujo, sin embargo, después del 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 los siguientes:

- 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.
- Mal elaboración de mensajes en los datos de salida.
- Falta de datos de salida (este error junto con el anterior nos lleva a ver que se hizo un análisis incorrecto).
- 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 curso de *Scratch* ya que se usó como evidencia para saber hacia qué aspectos el docente debía enfocarse con más detalle. Posteriormente, este conocimiento se complementó con los resultados del análisis de contenido aplicado a las pruebas de desempeño obtenidas luego de llevar a cabo las intervenciones; los resultados en algunos casos sirvieron para confirmar lo que se tenía inicialmente, pero también se detectaron nuevos fallos que fue importante considerar. Se destaca la siguiente lista de errores que son los más recurrentes en éstas, de igual modo son los que el docente ha detectado en el aula de clase en función de las observaciones hechas y de 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, se detectó una deficiente comprensión lectora.
- Dificultad en el manejo de operadores aritméticos (+, -, *, /, **, div, mod, ()).
- Elaboración de las condiciones de ciclo de modo incorrecto (ciclos while, do..while y for).
- Condición mal elaborada en estructuras de decisión 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, principalmente 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 presenta al inicializar sumadores, multiplicadores, contadores u 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 el 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.

En resumen, el conocimiento de los errores revelados durante el curso propedéutico y durante el curso de algoritmia fue de especial interés en el proceso investigativo, ya que esto permitió que el docente se centre en los principales temas problema. 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 muy importante porque el maestro ahora cuenta con un punto de partida sobre el cual poder enfocarse en las actividades de la materia. Por otro lado, resulta relevante mencionar que en la revisión hecha a la literatura la información encontrada al respecto es escasa.

En cuanto a lo expuesto respecto de las actas de examen, inicialmente se trabajó en hacer comparaciones entre los resultados obtenidos con el método tradicional versus el uso de *Scratch*, como ya ha sido citado se consideraron tres aspectos, el promedio de aprovechamiento de los grupos, el índice de aprobados y atendiendo al porcentaje de estudiantes con nivel de dominio satisfactorio o sobresaliente que maneja la UADY, en todos los casos el uso de *Scratch* presentó diferencias significativas a favor versus el método tradicional como se observó en el apartado de resultados, por lo que posteriormente para tener una evidencia más justificada se llevó a cabo la aplicación de estadísticos específicos para corroborar lo obtenido; a continuación se presenta información detallada de lo encontrado.

Primero se trabajó en comprobar si el promedio de aprovechamiento cuando se usa *Scratch* es significativamente diferente de cuando no se usa, los datos arrojados fueron los siguientes:

- Después de aplicar la prueba de Shapiro-Wilk para residuales (figura 4.4) se observó que éstos no tienen una distribución normal dado que el p-valor arrojado por el test de Shapiro-wilk fue de .000001849 el cual es menor al $\alpha = 0.05$ con el que se trabajó, esto nos conduce a que la prueba o test anova¹ de una vía no es la más adecuada para determinar si el promedio de aprovechamiento cuando se usa *Scratch* de cuando no se usa es significativamente diferente.
- El test de Shapiro-wilk también fue aplicado a los datos del primer y segundo año de la intervención, en el caso del primer año se tuvo un p-valor de 0.000589 y en el segundo fue de 0.0000221 por lo que en ambos años los datos no presentaron un comportamiento normal.
- Como siguiente paso y en función de lo obtenido se decide emplear el test no paramétrico Kruskal-Wallis (figura 4.5), de lo anterior se observó que el p-valor fue 0.01669 el cual es menor que el $\alpha = 0.05$ que se empleó, por lo que se rechaza H₀₁ y se concluye que el promedio de las calificaciones cuando se usa el método tradicional es significativamente diferente de cuando se utiliza *Scratch*.
- Es importante mencionar, que este indicador fue mejorando año con año, en Narváez y López (2020) se encuentra la descripción de lo que se tenía cuando se evaluó un solo año donde se tuvo un p-valor de 0.6623 y cuando se evaluó el segundo año los datos mejoraron con relación al primero ya que el p-valor en este caso fue de 0.1713, sin embargo, en ninguno de esos dos resultados se pudo rechazar H₀₁ ya que los p-valor eran mayores que α , de acuerdo a la evolución presentada se espera que el p-valor siga siendo menor en los próximos años.

- De esta prueba se concluye que el promedio de aprovechamiento cuando se usa *Scratch* si es significativamente mejor de cuando no se usa, esto con un nivel de significancia del 5%.

Aunado a los resultados favorables que se observaron con el promedio de aprovechamiento de los grupos, el segundo aspecto que se consideró importante revisar fue el rendimiento de los alumnos con base al índice de aprobados y reprobados, el cual también sumó a favor de la propuesta como se describe a continuación:

- Se usó la prueba no paramétrica de Fisher's Exacta con la tabla de análisis que fue presentada en la tabla 4.9 y del resultado de la figura 4.6 se observa que el p-valor fue de 0.01346 el cual es menor que $\alpha = 0.05$ por lo que se rechaza H_0 es decir, se verifica que los índices de aprobados y reprobados son dependientes del método aplicado, la estrategia de usar *Scratch* muestra una mejora significativa versus el método tradicional.
- Es importante mencionar que este indicador fue mejorando año con año, en Narváez y López (2020) se encuentra la descripción de lo que se tenía cuando se evaluó un solo año, donde se tuvo un p-valor de 0.3056 y cuando se evaluó el segundo año los datos mejoraron con relación al primero ya que el p-valor en este caso fue de 0.1847, sin embargo, en ninguno de esos dos resultados se pudo rechazar H_0 ya que los p-valor eran mayores que α , de acuerdo a la evolución presentada con esta tercera evaluación se espera que el p-valor siga siendo menor en los próximos años.
- Para efectos de esta prueba se concluye que los índices de aprobación y reprobación son dependientes del método aplicado, es decir el método cuando se usa *Scratch* mejora el índice de aprobación.

Por último, se analizó el rendimiento de los alumnos atendiendo al nivel de dominio con base al modelo que sigue la UADY, el cual también sumó a favor de esta propuesta, como se muestra a continuación:

- Se empleó la prueba no paramétrica de Fisher's Exacta (figura 4.7) y se observó que el p-valor fue de 0.03233 el cual es menor que $\alpha = 0.05$ por lo de nuevo se tiene evidencia suficiente para rechazar H_0 es decir, la estrategia de usar *Scratch* muestra una diferencia significativa versus el método tradicional viendo el problema de acuerdo con el nivel de dominio según el modelo que sigue la UADY.
- Es importante mencionar que este indicador también fue mejorando año con año, al evaluar la primera ocasión los datos arrojaron un p-valor de 0.5467, en cuanto al segundo año el p-valor fue de 0.11, sin embargo, en estos resultados no se pudo rechazar H_0 ya que el p-valor era mayor que α , de acuerdo con la evolución presentada con este tercer año, se espera que el p-valor siga siendo menor en los próximos cursos.
- Con el empleo de este estadístico, se concluye que los índices de alumnos con un nivel de dominio satisfactorio o sobresaliente y el índice de alumnos con nivel de dominio suficiente o no acreditado son dependientes del método aplicado, es decir, el método cuando se usa *Scratch* mejora el porcentaje de alumnos con un nivel de dominio satisfactorio o sobresaliente.

Como beneficio adicional de la implementación de esta estrategia, se planteó una hipótesis para verificar si el promedio de calificación de los cursos impartidos usando *Scratch* realmente se encuentra en el nivel satisfactorio, en cuanto al resultado se obtuvo lo siguiente:

- De acuerdo con la aplicación de la Prueba Shapiro-Wilk (figura 4.8). Se observa que el p-valor es de 0.00001589 el cual es menor al $\alpha = 0.05$, por lo tanto, se

concluye que los datos no tienen una distribución normal y en consecuencia la prueba no paramétrica que se usó fue el test de signos de Wilcoxon (figura 4.9).

- El test de Shapiro-wilk también fue aplicado a los datos del primer y segundo año de la intervención, en el caso del primer año se tuvo un p-valor de 0.007 y en el segundo fue de 0.0125 por lo que en ambos años, los datos no presentaron un comportamiento normal.
- En los resultados del test de signos de Wilcoxon, se observa un p-valor de 0.000007488 menor al $\alpha = 0.05$ es decir, en esta prueba se tiene evidencia para rechazar H_0 , por tanto, se puede afirmar que el promedio de aprovechamiento cuando se usa *Scratch* es superior a 80 puntos y este resultado es óptimo de acuerdo con el planteamiento de esta investigación.
- Cabe hacer mención, que esta prueba del test de signos de Wilcoxon también fue aplicada con los resultados encontrados en el primero y el segundo año de la intervención. En el primer año se obtuvo un p-valor de 0.25 y en este caso fue mayor que el valor de α propuesto por lo que en este año, no se pudo rechazar H_0 ; sin embargo, a partir del segundo año se tuvo evidencia suficiente para rechazar H_0 , ya que se tuvo un p-valor de 0.0125 el cual como se aprecia fue menor al $\alpha = 0.05$. De igual modo, se espera que el p-valor siga siendo menor en los próximos cursos.

Con todos los datos recabados en el transcurso de las tres intervenciones realizadas, se ha encontrado en las pruebas empleadas cómo los datos arrojados fueron mejorando año con año, conduciéndonos a confirmar en algunas pruebas a partir del tercer año y en otras desde el segundo, que la metodología de aprendizaje desarrollada con la implementación del lenguaje de programación *Scratch* es diferenciadora. Vale la pena citar, que suponiendo que los datos preserven el mismo comportamiento con el paso de los cursos, se procedió a realizar una simulación

(cuarto año de la intervención) y se observó que con el correr del tiempo se seguirá marcando una diferencia significativa entre las dos estrategias.

Por último, en cuanto al instrumento del cuestionario, se presentan los resultados en función de los tres aspectos que fueron evaluados. En primera instancia la actitud de los alumnos frente al uso de *Scratch* después de la inclusión de esta herramienta, de lo presentado en la tabla 4.10 se desprende que:

- La puntuación más alta se revela en los reactivos *“Es positivo ir integrando progresivamente la TIC en los estudios”*, *“¿Qué tan adecuado es usar una metodología que incorpora esta TIC?”*, *“Favorece un aprendizaje activo como alumno”* y *“Su uso ayuda a realizar mejor las tareas de algoritmos”*, donde por sobre el 96.6% de los respondientes seleccionaron las respuestas de Muy de Acuerdo y De Acuerdo y la puntuación más baja se ha obtenido en el reactivo *“Los profesores tienen que hacer un esfuerzo de actualización para aprovechar las posibilidades didácticas de Scratch”*, donde únicamente el 81.4% de los respondientes seleccionó las opciones de Muy de Acuerdo y De Acuerdo.
- Del análisis de los resultados, destaca el hecho de que en las doce interrogantes los valores más seleccionados fueron las opciones de Muy de Acuerdo y De Acuerdo y los respondientes nunca eligieron la opción de respuesta Muy en desacuerdo en ninguno de los 12 reactivos del cuestionario.
- Se observa que la tercera parte del cuestionario, tienen más del 96.6% de las selecciones en Muy de Acuerdo y De Acuerdo (columna suma de la tabla 4.10).
- En general, el porcentaje total de la elección de las dos opciones de respuesta de interés para todos los reactivos fue de 92.8% lo que nos indica que la actitud de los estudiantes frente al uso de la TIC *Scratch* durante la asignatura de algoritmia es favorable.

Es importante citar, que la actitud encontrada no es suficiente, solo se podrán integrar las TIC con un verdadero objetivo pedagógico, que no se reduzcan sólo a la habilitación tecnológica, ya que estas no pueden por sí mismas elevar los niveles

educativos, desarrollar una actitud positiva en el alumnado, ni fomentar su autonomía, se requieren cambios profundos en un contexto que ya no admite el método tradicional en un entorno innovador y digitalizado y esto no puede ser a la ligera, conlleva por parte del docente una estricta planeación del proceso, de modo reflexivo, pensado y crítico.

En la opinión de los alumnos respecto a las TIC después de haber incluido la herramienta *Scratch*, del análisis de la tabla 4.11 se desprende que:

- La puntuación más alta se revela en los reactivos "*Favorece la búsqueda de recursos que ayudan en las tareas académicas*", "*Favorece las tareas de repaso de aprendizajes*" y "*Mejora la elaboración de trabajos académicos*", donde sobre el 96.6% de los respondientes seleccionaron los valores de Muy de Acuerdo y De Acuerdo y la puntuación más baja, se ha obtenido en el reactivo "*Permite optimizar el manejo del tiempo que se dedica al estudio*", donde el 76.3% de los respondientes eligió los valores de Muy de Acuerdo y De Acuerdo.
- Del análisis de los resultados, destaca el hecho de que en las diez interrogantes los valores preferidos fueron las opciones de Muy de Acuerdo y De Acuerdo, por otro lado, los respondientes nunca escogieron la opción de respuesta Muy en desacuerdo en ninguno de los 10 reactivos del cuestionario.
- En general, el porcentaje total de la elección de estas dos opciones de respuesta para todos los reactivos fue de 87.6% lo que nos indica que los estudiantes están de acuerdo con el uso de las TIC como estrategias de aprendizaje.

Por último, en cuanto a la opinión de los alumnos respecto al uso de *Scratch*, del análisis de la tabla 4.12 se desprende que:

- La puntuación más alta se revela en las cuestiones de "*Es útil para entender la lógica de los algoritmos que se crean*", "*Es útil para probar los algoritmos creados*" y "*Es útil para entender conceptos relacionados con principios de programación*", donde por sobre el 98.3% de los respondientes seleccionaron

Muy de Acuerdo y De Acuerdo y la puntuación más baja se ha obtenido en el reactivo “*Proporciona flexibilidad de espacio y tiempo para la comunicación entre los miembros de la clase*”, donde el 76.3% de los respondientes optó por los valores de Muy de Acuerdo y De Acuerdo.

- Del análisis de los resultados, destaca el hecho de que en las once interrogantes los valores preferidos fueron las opciones de Muy de Acuerdo y De Acuerdo, por otro lado, los respondientes nunca eligieron Muy en desacuerdo en ninguna de las 11 cuestiones.
- En general, el porcentaje total de la elección de estas dos opciones de respuesta para todas las cuestiones fue de 91.4% lo que nos indica que los estudiantes están de acuerdo con el uso de *Scratch* durante las clases de algoritmia.

El cuestionario arrojó resultados positivos, en las tres secciones analizadas, lo cual es adecuado y se aprecia que se adapta a los intereses de este proceso investigativo, el uso de TIC y en especial de *Scratch* es bien recibido por los estudiantes, por lo que no representa una carga al momento de emplearlo como soporte y apoyo en los temas que resultan complejos y si resulta importante para poder entender mejor y dominar los temas de algoritmia, llevándolos a obtener mejores resultados finales

En resumen y como conclusión de esta sección se tienen los siguientes datos:

- El curso propedéutico resultó ser un buen punto de partida, para unificar conocimientos y para conocer a los estudiantes de un modo general, esto dado la diversidad de sistemas educativos de los cuales provienen.
- El saber cuáles son los principales problemas que tienen los alumnos cuando construyen sus diagramas de flujo es de vital importancia durante la implementación de algoritmia, por lo que tener este conocimiento resultó muy útil para hacer la planeación del curso ya que el docente supo desde el principio en qué temas poner especial cuidado.

- Con los estadísticos aplicados a las actas de examen, se logra apreciar como con el correr de las diferentes intervenciones los datos fueron mejorando y siendo cada vez más positivos en beneficio de la metodología de aprendizaje propuesta, luego de 3 intervenciones en algunas pruebas y en otras luego de 2 se ha visto cómo las hipótesis que nos interesaban han sido aceptadas, esto es indicador de que la metodología está funcionando en beneficio de los estudiantes.
- Se tenía duda si los alumnos estaban dispuestos a aceptar la TIC *Scratch* como herramienta de apoyo durante el semestre escolar, ya que implicaba una carga adicional, sin embargo, por los resultados arrojados en el cuestionario se puede ver como *Scratch* ha sido aceptada y algo que es importante es que el alumnado piensa que es una herramienta que les está sirviendo para entender los temas complejos y elevar su rendimiento académico, además se ha observado que la actitud de los estudiantes hacia las TIC en general y *Scratch* en particular es favorable.
- Cuando se emplea *Scratch* en el aula se está haciendo uso de la teoría del construccionismo y en función de los resultados encontrados donde los alumnos han elevado su rendimiento académico, esto nos indica que los estudiantes realmente se encuentran implicados en su proceso educativo mediante la elaboración de sus programas los cuales demuestran ser un referente tangible para entender los algoritmos.
- La investigación llevada a cabo ha respondido a la hipótesis establecida ya que el planteamiento ha demostrado una relación inequívoca, los resultados del experimento se han recopilado e interpretado adecuadamente y son válidos, asimismo, se ha detectado una relación causal entre las variables.
- Con lo encontrado en la literatura analizada y conjuntando esta información con los resultados obtenidos, se considera posible la aplicación de la herramienta *Scratch* para explicar temas complicados en otras asignaturas no únicamente en la universidad si no también en otros niveles educativos. En esta investigación se resolvió el problema educativo en cuestión y se ofrecen principios como la

metodología creada para que estudios similares cuenten con referencias metodológicas.

- Los resultados obtenidos son similares a los encontrados en otras investigaciones analizadas en los que han empleado herramientas tecnológicas de apoyo al alumno para elevar su rendimiento académico, sin embargo, esas investigaciones se encuentran enfocadas a la programación en sí misma y no al análisis, diseño y construcción de algoritmos como los diagramas de flujo los cuales son los de interés en la materia de algoritmia.
- En función del impacto que se está teniendo con los grupos evaluados dado que los resultados de todo el proceso investigativo han sido positivos y que además la simulación realizada indica que a futuro los resultados seguirán por este camino, se puede decir que el alcance de esta investigación al aplicar la metodología de aprendizaje creada ha sido adecuada, es viable y favorecedora ya que el rendimiento académico de los estudiantes ha ido subiendo año con año, por lo que es conveniente integrar la herramienta *Scratch* a la materia de algoritmia.

5. CONCLUSIONES

En este capítulo, se presentan las conclusiones más importantes que han sido obtenidas a lo largo de los tres años en los que se ha llevado a cabo el proceso investigativo.

Partiendo de la información recopilada en los años en que se impartió la asignatura de algoritmia sin la implementación de ningún tipo de software y de la revisión de la literatura científica, fue posible llegar a una propuesta metodológica de aprendizaje la cual se fue perfeccionando durante tres períodos consecutivos, en los que se implementó y ha estado sirviendo para ayudar al alumno a obtener un mejor rendimiento académico en la citada materia, la cual es impartida en la licenciatura en ingeniería de software de la Unidad Multidisciplinaria Tizimín.

De común acuerdo con la literatura científica y siguiendo la metodología de la investigación basada en diseño, se trabajó primero en determinar los antecedentes de los alumnos cuando se inscriben a la LIS, así como también de unificar sus conocimientos mediante un curso propedéutico; desde el inicio surgió la necesidad de identificar los principales errores que cometen los estudiantes durante el proceso de análisis y creación de algoritmos mediante un análisis de contenido el cual fue un proceso que se fue desarrollando poco a poco.

Posteriormente, con ayuda de la información encontrada, con el apoyo de la experiencia del docente de la materia y como resultado de la revisión teórica de lo que otros docentes han hecho respecto a problemas similares al que aquí se presenta, es que se decidió atacar el problema mediante cursos complementarios con el empleo de la TIC *Scratch* que le pudieran servir de ayuda al alumno para tener un referente de tipo visual de los resultados de los problemas que tiene que resolver, por esta razón se generó un taller de *Scratch* que se impartió a la par del curso de algoritmia durante los 3 años en que se hizo la intervención.

El taller implementado, incluyó el detalle de los ejercicios y animaciones necesarias con el objetivo de resolver la problemática planteada, enfocado a la

enseñanza de los conceptos fundamentales de principios de programación, éste tomó como base la teoría del construccionismo y se perfeccionó año con año en función de los resultados que se fueron obteniendo, lo cual fue crucial para la mejora de la metodología propuesta inicialmente. Al final de cada intervención se le aplicó al alumno un cuestionario y se obtuvieron las actas de examen del SICEI de la universidad.

Como respuesta a la pregunta de investigación, ¿El uso del lenguaje de programación *Scratch* mediante la creación de programas, juegos y/o animaciones básicas elevará el rendimiento académico de los alumnos que cursan la asignatura de algoritmia que se imparte en el primer semestre de la LIS en la UMT de la UADY? se presenta la propuesta de la metodología de aprendizaje, que tomó en cuenta el análisis de los problemas y su posterior aterrizaje en algo tangible por parte del alumno haciendo uso de la teoría del construccionismo con la inclusión de un trabajo en el laboratorio, donde el educando pudo construir sus propios artefactos para entender su realidad pasando por el análisis, diseño y posterior construcción de animaciones y/o juegos básicos en 2D donde fue capaz de hacer, probar y comprobar sus creaciones mediante un espacio que resultó ser un referente visual de la solución de los problemas estudiados; lo cual le sirvió para entender mejor los conceptos y por ende tener un mejor rendimiento académico siendo acompañado en todo momento por la guía del docente.

Atendiendo a lo planteado en las preguntas específicas y complementando la pregunta de investigación, se desarrollaron dos compendios unos de ejercicios y otro de animaciones básicas los cuales se fueron aplicando a lo largo de los cursos y sirvieron como punto de referencia para los alumnos, concluyendo que el uso de estos compendios con el empleo de *Scratch* ayudó al alumno en su rendimiento y la herramienta fue bien recibida.

Con el objetivo de verificar la hipótesis propuesta, para comprobar si el uso de una metodología de aprendizaje que incorpore el lenguaje de programación *Scratch* a la asignatura de algoritmia que se imparte en la LIS de la UMT, mejora la

comprensión de problemas del área y eleva el rendimiento académico de los alumnos, se llevó a cabo el estudio de los datos recabados y de la investigación realizada se concluye lo siguiente:

- 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.
- No se puede negar, que partiendo del marco teórico se ha puesto de manifiesto la influencia de las herramientas tecnológicas en el ámbito educativo para dar solución al bajo rendimiento académico y la herramienta *Scratch* se ha empleado con este fin en una diversidad de temas y niveles educativos.
- Resultó de importancia conocer a detalle en qué temas los alumnos de la LIS tienen más dificultades al momento de construir sus algoritmos, información que sirvió de base para la planeación de los cursos.
- En cuanto al rendimiento académico, cuando se introduce al alumno a los principios básicos de programación, este es un problema que no es exclusivo de la UMT y es un punto que preocupa y ocupa a los docentes. Para verificar que la metodología propuesta funciona se evaluaron los resultados obtenidos de las actas de examen, tanto las de aquellos cursos que fueron impartidos con el método tradicional, como aquellos en los que se implementó *Scratch*. Para esto fueron considerados tres aspectos:
 - Primero: Se comprobó que el promedio de aprovechamiento cuando se usa *Scratch* es significativamente mejor de cuando no se usa.
 - Segundo: Se evaluó el rendimiento de los alumnos con base al índice de aprobados y reprobados, y se determinó que este resultado depende del método aplicado, encontrando una mejora significativa cuando se usa *Scratch*.

- Tercero: Se analizó el rendimiento de los alumnos atendiendo al nivel de dominio con base al modelo que sigue la UADY y se encontró que cuando se usa *Scratch* mejora el porcentaje de alumnos con un nivel de aprovechamiento satisfactorio o sobresaliente.
- Como beneficio adicional de la implementación de esta estrategia también se comprobó que el promedio general de calificación de los cursos impartidos usando *Scratch* se encuentra en el nivel satisfactorio de acuerdo con los niveles de dominio de la UADY, lo cual es un resultado adecuado en esta investigación.
- Para finalizar, en cuanto al rendimiento académico, es importante mencionar que los resultados, cuando se emplea *Scratch* han ido mejorando después de cada intervención y se espera que en los cursos futuros esta tendencia continúe con base en los resultados obtenidos y en las simulaciones realizadas.
- Respecto al cuestionario aplicado, se puede concluir que los estudiantes manifiestan una actitud positiva ante la incorporación de la herramienta tecnológica *Scratch* en su curso de algoritmia y esto es coincidente con el análisis de actitudes hacia diversas herramientas tecnológicas realizadas en otras investigaciones (Bullones, Vivas, & Caseres, 2015; Roig, Mengual, Sterrantino, & Quinto, 2015).
- En cuanto a la opinión que manifiestan los alumnos en función del uso de las TIC en general, así como de la incorporación de *Scratch* a algoritmia para el caso particular, el cuestionario aplicado refiere que se encuentran de acuerdo con su uso como una estrategia de apoyo en su proceso de aprendizaje.
- De igual modo, se concluye que los alumnos perciben a *Scratch* como una herramienta que los ayuda a entender los temas donde tienen dificultades así como también a tener un mejor rendimiento académico.
- Se concluye, que partiendo de los datos encontrados en el curso propedéutico y el análisis de contenido a las pruebas de desempeño, se logró la implementación del taller de *Scratch* en algoritmia, el cual en conjunto con los ejercicios resueltos

y los proyectos desarrollados, se verificó que este proceso de intervención tuvo un mayor impacto en el rendimiento académico de los alumnos.

En el ámbito tecnológico, se contará con una metodología al momento de impartir las clases, la cual incorporará el lenguaje de programación *Scratch* que le permitirá al alumno aterrizar en algo tangible lo que solo se tiene en papel, esto será relevante para la LIS en la UMT en beneficio de los estudiantes.

En el enfoque social, se utilizó *Scratch* para complementar los procesos de enseñanza-aprendizaje en un entorno real en algoritmia y en el enfoque cultural los proyectos que desarrollaron los alumnos incluyeron elementos de la cultura maya la cual pertenece al contexto y les permitió a los alumnos acercarse a esta parte tan valiosa que identifica a los yucatecos.

La aportación radica, en un cambio de estrategia en el proceso de enseñanza de la materia, el modelo tradicional que consistía en estudiar la asignatura a papel y lápiz ya no es suficiente para los alumnos de este siglo que están inmersos en la tecnología, razón por la cual en los cursos de algoritmia que se impartirán posteriores a esta investigación, se seguirá empleando la metodología de aprendizaje desarrollada, el taller creado, así como también los compendios de ejercicios y animaciones.

Como trabajo futuro, está abierta la propuesta de investigar, cuál es la situación que se presenta respecto del problema aquí descrito en otro campus de la UADY, con el fin de identificar situaciones específicas y dado el caso poder emplear la metodología desarrollada. Por otra parte, también se está considerando seguir la metodología propuesta empleando otro software semejante a *Scratch* para comparar resultados.

REFERENCIAS BIBLIOGRÁFICAS

- Aguilar, M. (2012). Aprendizaje y Tecnologías de Información y Comunicación: Hacia nuevos escenarios educativos. *Revista Latinoamericana de Ciencias Sociales, Niñez y Juventud*, 10(2), 801-811. Obtenido de <http://www.scielo.org.co/pdf/rlcs/v10n2/v10n2a02.pdf>
- Alania, R., Diez, C., & Pinglo, J. (2012). Estrategias de enseñanza y estilos de aprendizaje en los alumnos del curso Introducción a la Algoritmia: caso Cibertec. doi:10.13140/RG.2.1.3183.0480
- Aparicio, O., & Ostos, O. (2018). El Constructivismo y el Construccinismo. *Revista Interamericana de Investigación, Educación y Pedagogía, RIIEP*, 11(2), 115-120.
- Ávila Díaz, W. (2013). Hacia una reflexión histórica de las TIC. *Hallazgos*, 10(19), 213-233. Obtenido de <https://www.redalyc.org/pdf/4138/413835217013.pdf>
- Bell, P. (2004). On the theoretical breadth of design-based research in Education. *Educational Psychologist*, 4(39), 243-253.
- BlueJ. (s.f.). *BlueJ*. Obtenido de BlueJ: <https://www.bluej.org/>
- Botello, H., & Guerrero, A. (2014). *Portal Educativo de las Américas*. Obtenido de La influencia de las TIC en el desempeño académico de los estudiantes en América Latina: Evidencia de la prueba PISA 2012: <http://recursos.portaleducoas.org/publicaciones/la-influencia-de-las-tic-en-el-desempe-o-acad-mico-de-los-estudiantes-en-am-rica>
- Botello, H., & López, A. (2014). La influencia de las tic en el desempeño académico: evidencia de la prueba Pirls en Colombia 2011. *Revista Academia y Virtualidad*, 15-26.
- Bullones, M., Vivas, M., & Caseres, E. (2015). Actitud de los estudiantes frente al uso de tecnologías educativas para el aprendizaje de la matemática: Una

visión desde los estudiantes de ingeniería de la Universidad Centroccidental Lisandro Alvarado. *Revista Educación en Ingeniería*, 10(20), 143-153.

Cabero, J. (1998). Impacto de las nuevas tecnologías de la información y la comunicación en las organizaciones educativas. *En Lorenzo, M. y otros (coords): Enfoques en la organización y dirección de instituciones educativas formales y no formales*, 197-206.

Cairó, O. (2005). *Metodología de la Programación. Algoritmos, Diagramas de Flujo y Programas*. México: Alfaomega.

Calderón, C. (2020). Arqueología informática: diseño e implementación de calculadoras mecánicas Facit con Scratch (Tesis de Licenciatura). Valencia, España: Escola Tècnica Superior d'Enginyeria Informàtica Universitat Politècnica de València.

Calvo, A. (2011). Educación de competencias informáticas en adolescentes de secundaria del siglo XXI: una responsabilidad compartida entre la escuela y la familia. *Apertura*, 3(2).

Campos, J., Brenes, O., & Solano, A. (2010). Competencias del Docente de Educación Superior en línea. *Revista Electrónica Actualidades Investigativas en Educación*, 10(3), 1-19.

Cervantes, J. (2015). La mediación pedagógica a través de las TIC . Una propuesta para la formación docente en Educación Media Superior. Universidad Pedagógica Nacional Unidad Ajusco.

Chocó, A. (s.f.). *Planeación Estratégica*. Obtenido de Educación, pedagógica y TIC. Un Trinomio imperioso en la nueva perspectiva social: <https://arbercho.jimdo.com/pedagog%C3%ADa-y-tic/>

Collí, D. (2017). Desarrollo de un software educativo para apoyo en inglés y maya mediante el uso del Kinect. Mérida, Yucatán, México.

- Dania, C., & 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*, 17(31-32), 215-228.
- Dávila, H. (2017). *Estrategia pedagógica mediada por las TIC para mejorar el rendimiento académico en el área de matemáticas para estudiantes de grado octavo del colegio Pablo Neruda*. Bogotá.
- De Benito, B., & Salinas, J. (Junio de 2016). La Investigación Basada en Diseño en Tecnología Educativa. *RIITE. Revista Interuniversitaria de Investigación en Tecnología Educativa*(0), 44-59. doi:<http://dx.doi.org/10.6018/riite/2016/260631>
- Delgado, M., Arrieta, X., & Riveros, V. (2009). Uso de las TIC en educación, una propuesta para su optimización. *Omnia*(3), 58-77. Obtenido de <https://www.redalyc.org/comocitar.oa?id=73712297005>
- Desarrollo de Lógica de Programación*. (s.f.). Recuperado el 26 de Septiembre de 2019, de <https://virtual.itca.edu.sv/Mediadores/dlp/>
- Díaz, D. (2013). TIC en Educación Superior: Ventajas y desventajas. *Educación y Tecnología*(4), 44-50.
- Díaz-Barriga, F. (2009). *La innovación en la enseñanza soportada en TIC. Una mirada al futuro desde las condiciones actuales*. Obtenido de <http://www.oei.es/tic/santillana/Barriga.pdf>
- Duart, J. (2008). El valor de las TIC en educación superior. *RUSC. Universities and Knowledge Society Journal*, 5(1).
- Durak, H. (2018). The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving. *Technology, Knowledge and Learning*.

- Edel, R. (2003). El rendimiento académico: concepto, investigación y desarrollo. *REICE. Revista Iberoamericana sobre Calidad, Eficacia y Cambio en Educación*, 1(2).
- Edel, R., Aguirre, G., & Balderrama, J. (2016). La contribución didáctica de las TIC en el desarrollo de competencias para la investigación. *Revista Ideales*, 1(1), 54-59. Obtenido de <http://revistas.ut.edu.co/index.php/Ideales/article/view/1080/844>
- Escudero, A. (2018). Principios de Investigación Basada en Diseño para la creación de un modelo de educación virtual. *Afrontar los retos de la educación en el siglo XXI 2*, (págs. 217-232).
- Fernández, J., Calderón, A., Méndez, A., & Rolim, R. (2014). Teoría Construcionista del Aprendizaje en formación del profesorado. Perspectivas de alumnado y profesorado desde la investigación cuantitativa y cualitativa. *Profesorado. Revista de Currículum y Formación de Profesorado*, 18(3), 213-228.
- Fuentes, A. (2017). Desarrollo y evaluación del pensamiento computacional: una propuesta metodológica y una herramienta de apoyo (Tesis de licenciatura). Universidad de la Laguna. San Cristóbal de La Laguna, España.
- Fuentes, C., & Careaga, M. (2012). *Contexto Mixto de Aprendizaje con Tutoría Virtual*. Editorial Académica Española.
- Fuentes, T. (2004). El estudiante como sujeto del rendimiento académico. *Sinéctica, Revista Electrónica de Educación*(25), 23-27. Obtenido de <http://www.redalyc.org/articulo.oa?id=99815899004>
- Gairín, J., Castro, D., Silva, J., & Mercader, C. (2016). *El impacto de las TIC en el aula desde la perspectiva del profesorado*. Barcelona.
- Gandy, E., Bradley, S., Brookes, D., & Allen, N. (2010). The use of LEGO Mindstorms NXT Robots in the Teaching of Introductory Java Programming to Undergraduate Students. *Innovation in Teaching and Learning in*

Information and Computer Sciences, 9(1).
doi:doi:10.11120/ital.2010.09010002

- García, E., Revuelta, F., & Esnaola, G. (2017). Videojuegos y su potencial para producir desarrollos en los aprendizajes. *Actas del V Congreso Internacional de Videojuegos y Educación*. Puerto de la Cruz.
- Garza, B., & Solís, A. (2012). Uso Pedagógico de las TIC en el Aula. *Revista Iberoamericana de las Ciencias Computacionales e Informática*, 1(2).
Obtenido de <https://www.reci.org.mx/index.php/reci/article/view/13/49>
- George, D., & Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference. 11.0 update (4thed.)*. Boston: Allyn & Bacon.
- Gibelli, T. (2014). La investigación basada en diseño para el estudio de una innovación en educación superior que promueve la autorregulación del aprendizaje utilizando TIC. *Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación*. Buenos Aires.
- 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.
- Hernández, R. (2017). Impacto de las TIC en la educación: Retos y Perspectivas. *Propósitos y Representaciones*, 5(1), 325-347.
doi:<http://dx.doi.org/10.20511/pyr2017.v5n1.149>
- Huertas, A., & Pantoja, A. (2016). Efectos de un programa educativo basado en el uso de las TIC sobre el rendimiento académico y la motivación del alumnado en la asignatura de tecnología de educación secundaria. *Educación XX1*, 19(2), 229.

- Isuri Sormenezko Zerbitzuak Servicios Creativos. (2018). *Isuri*. Obtenido de Guía Didáctica para Profesores: www.isuriarte.com
- Janeiro, E. (2016). Scratch y videojuegos aplicados a la enseñanza de la geometría . (Tesis de Maestría). Del repositorio digital de la Universidad Internacional de la Rioja. Vigo.
- 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.
- Khvilon, E., & Patru, M. (2004). *Las tecnologías de la información y la comunicación en la formación docente. Guía de planificación*. (UNESCO, Ed.) Obtenido de https://unesdoc.unesco.org/in/documentViewer.xhtml?v=2.1.196&id=p::usm arcdef_0000129533_spa&file=/in/rest/annotationSVC/DownloadWatermarkedAttachment/attach_import_11149789-15d5-4202-ab5d-07a35ab85885%3F_%3D129533spa.pdf&locale=es&multi=true&ark=/ark:/482
- Lage, F., & Cataldi, Z. (2019). Una experiencia de resolución de problemas a través de modelos cooperativos-colaborativos aplicada a algoritmia usando nuevas tecnologías de comunicación.
- López, J. (2012). Identificación y regulación de emociones con Scratch. En *Tendencias emergentes en educación con TIC* (págs. 67-81). Barcelona: Asociación Espiral, Educación y Tecnología.
- 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/>

- Marqués, P. (2000). Las TIC y sus aportaciones a la sociedad. Recuperado el Marzo de 2019, de file:///D:/2%20Doctorado%20DITE/2o.%20Semestre/Tesis/Art%C3%ADculos%20para%20la%20Tesis/LAS_TIC_Y_SUS_APORTACIONES_A_LA_SO-CIEDAD.pdf
- Mirete, A., & García, F. (2014). Rendimiento académico y tic. Una experiencia con webs didácticas en la universidad de Murcia. *Pixel-Bit Revista de Medios y Educación*, 169-183. doi:10.12795/pixelbit.2014.i44.12
- MIT Scratch Team. (s.f.). *Scratch*. Obtenido de <https://scratch.mit.edu/>
- Monjelat, N., Cenacchi, M., & San Martín, P. (2018). ¿Programación para Todos? Herramientas y Accesibilidad: Un Estudio de Caso. *Revista Latinoamericana de Educación Inclusiva*, 12(1), 213-227.
- Muñoz, R., Barcelos, T., Villarroel, R., & Silveira, I. (2015a). Diseño e implementación de un taller de programación de juegos digitales con Scratch como apoyo a Fundamentos de Programación. *Anais dos Workshops do IV Congresso Brasileiro de Informática na Educacao (CBIE)*, (págs. 1495-1504). doi:10.5753/cbie.wcbie.2015.1495
- Muñoz, R., Barcelos, T., Villarroel, R., Barría, M., Becerra, C., Noel, R., & Frango, I. (2015b). 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.

- Pinales, F., & Velázquez, C. (2014). *Algoritmos resueltos con diagramas de flujo y pseudocódigo* (Departamento Editorial de la Dirección General de Difusión y Vinculación ed.). Aguascalientes: Universidad Autónoma de Aguascalientes.
- Plaza, J. (2018). Ventajas y desventajas del uso adolescente de las TIC: visión de los estudiantes. *Revista Complutense de Educación*, 29(2), 491-508. doi:<https://doi.org/10.5209/RCED.53428>
- Posada Prieto, F. (Febrero de 2017). *CanalTIC Educación*. Recuperado el 11 de Junio de 2018, de Videojuegos y robots: <http://canaltic.com/vr/manual/index.html>
- Prieto, M., Mijares, B., & Llorent, V. (2014). Roles del docente y del alumno universitario desde las perspectivas de ambos protagonistas del hecho educativo. *Docencia*, 273-293.
- Puig, L. (2008). Historias de al-Khwārizmī (1ª entrega). *Suma*, 125-130.
- Quesada, M., Sobrado, D., & Solís, C. (2 de Julio de 2015). *Implicaciones para la tecnología educativa*. Obtenido de <https://modelosestandaresycompetencias.wordpress.com/2015/07/02/implicaciones-para-la-tecnologia-educativa/>
- Ramírez, M., Lucio, M., Garza, J., García, L., & Vargas, J. (2011). Alice: Un entorno diferente para aprender programación orientada a objetos. *CienciaUAT*, 6(2), 64-68.
- 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.
- Rodríguez, J. (9 de Septiembre de 2020). Sistemas software para monitorizar el entrenamiento personalizado del Pensamiento Computacional (Tesis de Licenciatura). La Laguna, España. Obtenido de <https://riull.ull.es/xmlui/bitstream/handle/915/21319/Sistemas%20software%20para%20monitorizar%20el%20entrenamiento%20personalizado%20del%20pensamiento%20computacional.pdf?sequence=1>

- Roig, R., Mengual, S., Sterrantino, C., & Quinto, P. (2015). Actitudes hacia los recursos tecnológicos en el aula de los futuros docentes. *Revista d'innovació educativa*(15), 12-19. doi:10.7203/attic.15.7220
- Rosanigo, Z., & Paur, A. (2006). Estrategias para la enseñanza de Algorítmica y Programación. *I Congreso de Tecnología en Educación y Educación en Tecnología.*
- Sáez, J., Miyata, Y., & Domínguez-Garrido, M. (2016). Creative Coding and Intercultural Projects in Higher Education: a Case Study in Three Universities. *RIED. Revista Iberoamericana de Educación a Distancia, 19*(2), 145-165. Obtenido de <http://www.redalyc.org/articulo.oa?id=331445859009>
- 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.
- Salinas, J. (2004). Innovación docente y uso de las TIC en la enseñanza universitaria. *Revista Universitaria y Sociedad del Conocimiento*, 1-16. Obtenido de <https://www.raco.cat/index.php/RUSC/article/viewFile/28810/28644>
- Santiago, G., Caballero, R., Gómez, D., & Domínguez, A. (2013). El uso didáctico de las TIC en escuelas de educación básica en México. *Revista Latinoamericana de Estudios Educativos, XLIII*(3), 99-131. Obtenido de <http://cee.edu.mx/home/wp-content/uploads/2015/04/04-Uso-TIC-R3-2013.pdf>
- Scratch Wiki. (2008). *Scratch Versions*. Obtenido de https://en.scratch-wiki.info/wiki/Scratch_Wiki_Home
- Snap! Build your own blocks*. (2018). Recuperado el 30 de Octubre de 2018, de About Snap!: <https://snap.berkeley.edu/index.html>

- Sosa, M., Peligros, S., & Díaz, D. (2010). Buenas prácticas organizativas para la integración de las TIC en el sistema educativo extremeño. *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*.
- Stuart, G. (2009). Learning to program from Scratch. *9th IEEE International Conference on Advanced Learning Technologies, ICALT 2009*, (págs. 451-452). Riga, Latvia. doi:10.1109/ICALT.2009.50
- Suárez, M. L. (2015). *Conectar Igualdad*. Obtenido de https://cdn.educ.ar/dinamico/UnidadHtml__get__54caa67c-5847-4be3-bbb9-5db9342bd51c/pdf/lenguajes_de_programacion_5.pdf
- Sunkel, G., Trucco, D., & Espejo, A. (2014). *La integración de las tecnologías digitales en las escuelas de América Latina y el Caribe. Una mirada multidimensional*. Santiago: CEPAL y Naciones Unidas.
- Tec, B., Uc, J., González, C., García, M., Escalante, M., & Montañez, T. (2010). Análisis Comparativo de dos Formas de Enseñar Matemáticas Básicas: Robots LEGO NXT y Animación con Scratch. *Conferencia Conjunta Iberoamericana sobre Tecnologías de Aprendizaje*. Mérida.
- 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*.
- UNESCO. (2013a). *Enfoque estratégico sobre las TICS en educación en América Latina y el Caribe*. Chile. Obtenido de <http://www.unesco.org/new/fileadmin/MULTIMEDIA/FIELD/Santiago/images/ticsesp.pdf>
- UNESCO. (2013b). *Uso de TIC en educación en América Latina y el Caribe: análisis regional de la integración de las TIC en la educación y de la aptitud digital (e-readiness)*. Montreal, Canadá. Obtenido de <https://unesdoc.unesco.org/ark:/48223/pf0000219369>

- UNESCO. (2014). Políticas TIC en los sistemas educativos de América Latina. En *Informe sobre tendencias sociales y educativas en América Latina*. (págs. 16-31, 217-219, 248-258). Buenos Aires: UNESCO.
- UNESCO. (2019). Marco de Competencias de los Docentes en materia de TIC. París, Francia. Recuperado el 20 de Marzo de 2019, de <https://unesdoc.unesco.org/ark:/48223/pf0000371024>
- 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.
- Vicario, C. (2009). Construccinismo. Referente sociotecnopedagógico para la era digital. *Innovación Educativa*, 9(47), 45-50.
- Vidal, C., Cabezas, C., Parra, J., & López Leopoldo. (2015). Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile. *Formación Universitaria*, 8(4), 23-32.
- Villarruel, M. (2012). Innovar desde las tecnologías de la información y la comunicación. *RIED. Revista Iberoamericana de Educación a Distancia*, 15(1), 37-47.
- Villegas, A. (11 de Abril de 2015). *e-historia*. Obtenido de Aporte de las NTIC a la Educación: <http://www.e-historia.cl/e-historia/articulo-e-historia-aporte-de-las-ntic-a-la-educacion/>
- Viñals, A., & Cuenca, J. (2016). El rol del docente en la era digital. *Revista Interuniversitaria de Formación del Profesorado*, 30(2), 103-114.
- Xinogalos, S., Satratzemi, M., & Dagdilelis, V. (2006). An introduction to object-oriented programming with a didactic microworld: objectKarel. *Computers & Education*(47), 148-171.

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>

Zamora, F. (Junio de 2016). Estrategia pedagógica mediante creación de historietas en Scratch para fortalecer la comprensión lectora en los estudiantes del grado 401 de la IED la Victoria sede rural María Auxiliadora. Girardot, Colombia.

Dirección General de Bibliotecas UAG

ANEXOS

Anexo A Cuestionario

El propósito de este cuestionario es recolectar información que permita mejorar la forma de impartir la materia de Algoritmia.

Para cada cuestión se incluye una escala de valoración, por favor rellene el cuestionario marcando la opción con la que mejor se identifique.

Recuerde responder a todas las preguntas. La información que facilite será de gran utilidad y se tratará **confidencialmente**.

Datos de Identificación (Marque lo que proceda)

1. Género () Hombre () Mujer

2. Edad

3. Escuela de procedencia

4. ¿Es recién egresado de bachillerato? () Si () No

5. Si la respuesta a la pregunta 4 fue NO ¿En qué año egresó?

6. La licenciatura en ingeniería de software ¿fue su primera opción? () Si () No

7. Si la respuesta a la pregunta 6 fue NO ¿Cuál fue su primera opción?

8. ¿Ha llevado cursos de algoritmia antes del curso actual? () Si () No

9. Si la respuesta a la pregunta 6 fue SI ¿Qué curso fue?

Se solicita su opinión en relación con el contenido de las frases que aparecen a continuación, en términos de acuerdo/desacuerdo y respondiendo al siguiente criterio:

| | | |
|----|-----|------------------------------------|
| MD | (1) | Muy en desacuerdo |
| D | (2) | En Desacuerdo |
| I | (3) | Indiferente, sin opinión elaborada |
| A | (4) | De Acuerdo |
| MA | (5) | Muy de acuerdo |

| <i>Actitud ante el uso de la TIC Scratch</i> | Grado de acuerdo/desacuerdo con el contenido de la frase | | | | |
|--|--|--------|--------|--------|---------|
| | MD 1 | D 2 | I 3 | A 4 | MA 5 |
| 1. Es importante para el aprendizaje en principios de programación. | | | | | |
| 2. Considera que los profesores deberían utilizar <i>Scratch</i> para facilitar el aprendizaje de los alumnos en algoritmia. | | | | | |
| 3. Favorece un aprendizaje activo como alumno | | | | | |
| 4. Los profesores tienen que hacer un esfuerzo de actualización para aprovechar las posibilidades didácticas de <i>Scratch</i> | | | | | |
| 5. Es positivo ir integrando progresivamente la TIC en los estudios | | | | | |
| 6. ¿Qué tan adecuado es usar una metodología que incorpora esta TIC? | | | | | |
| 7. El uso en la clases retrasa el aprendizaje | | | | | |
| 8. El aprendizaje perderá eficacia a medida que se vayan incorporando <i>Scratch</i> | | | | | |
| 9. ¿Qué tan relevante es usar <i>Scratch</i> para la enseñanza y el aprendizaje? | | | | | |
| 10. ¿Qué tan importante es la introducción de <i>Scratch</i> en algoritmia? | | | | | |
| 11. Su uso ayuda a realizar mejor las tareas de algoritmos | | | | | |
| 12. El aprendizaje de algoritmia puede mejorar con el uso de <i>Scratch</i> | | | | | |

| <i>Se considera que el uso de las TIC</i> | Grado de acuerdo/desacuerdo con el contenido de la frase | | | | |
|---|--|--------|--------|--------|---------|
| | MD 1 | D 2 | I 3 | A 4 | MA 5 |
| 1. Favorece las tareas de repaso de aprendizajes | | | | | |
| 2. Mejora la elaboración de trabajos académicos | | | | | |
| 3. Organizan mejor el trabajo académico | | | | | |
| 4. Desarrollan un mejor pensamiento crítico | | | | | |
| 5. Posibilita una mejor organización cognitiva | | | | | |
| 6. Permite optimizar el manejo del tiempo que se dedica al estudio | | | | | |
| 7. Favorece la regulación del esfuerzo que se dedica a las tareas académicas. | | | | | |
| 8. Mejora las posibilidades de trabajar con otros compañeros. | | | | | |
| 9. Favorece la búsqueda de recursos que ayudan en las tareas académicas. | | | | | |
| 10. Mejora los resultados académicos | | | | | |

| <i>Uso de la TIC Scratch</i> | Grado de acuerdo/desacuerdo con el contenido de la frase | | | | |
|---|--|--------|--------|--------|---------|
| | MD 1 | D 2 | I 3 | A 4 | MA 5 |
| 1. Fomenta la implicación en los procesos de enseñanza y aprendizaje | | | | | |
| 2. Los profesores del área de algoritmia deben utilizar <i>Scratch</i> para mejorar la calidad de los procesos de aprendizaje | | | | | |
| 3. Es útil para entender la lógica de los algoritmos que se crean | | | | | |
| 4. Es útil para probar los algoritmos creados | | | | | |
| 5. Ha servido para mejorar la comprensión de las estructuras básicas de programación | | | | | |
| 6. Las clases mejoran a medida que se va incorporando <i>Scratch</i> | | | | | |
| 7. Facilita el desarrollo de las clases | | | | | |
| 8. Ayuda a conseguir las competencias | | | | | |
| 9. Proporciona flexibilidad de espacio y tiempo para la comunicación entre los miembros de la clase | | | | | |
| 10. Es útil para entender conceptos relacionados con principios de programación | | | | | |
| 11. ¿Te ha servido para obtener un mejor rendimiento académico? | | | | | |

Dirección General de Bibliotecas UAG