

**DESARROLLO DE GUÍAS DE PENSAMIENTO COMPUTACIONAL CON
ARDUINO ORIENTADO A ESTUDIANTES DE PRIMER SEMESTRE DE
INGENIERÍA DE SISTEMAS**

Jholman Duván Rojas Bautista

**Facultad de Ingeniería y arquitectura
Departamento de Electrónica, Eléctrica, Telecomunicaciones y Sistemas
Programa de ingeniería de sistemas
Pamplona
2019**

**DESARROLLO DE GUÍAS DE PENSAMIENTO COMPUTACIONAL CON
ARDUINO ORIENTADO A ESTUDIANTES DE PRIMER SEMESTRE DE
INGENIERÍA DE SISTEMAS**

Jholman Duván Rojas Bautista

Trabajo de grado para obtener el título de Ingeniero de sistemas

Director:

**Edgar Alexis Albornoz Espinel
Mg. En Ciencias de la computación**

**Facultad de Ingeniería y arquitectura
Departamento de Electrónica, Eléctrica, Telecomunicaciones y Sistemas
Programa de ingeniería de sistemas**

Pamplona

2019

CITAS

“Todo el mundo debería aprender a programar, porque te enseña a pensar”

Steve Jobs.

DEDICATORIA

Primero al Padre Creador por darme la vida y los sueños, a mi Madre y mi Tía por el apoyo tanto emocional como monetario, a mis docente por la enseñanza impartida en cada uno de estos años tanto académico como personal, a mi director de tesis por su orientación y consejos.

RESUMEN

Este trabajo trata de la realización y los resultados obtenidos de un curso de pensamiento computacional, apoyado en la plataforma arduino con el propósito de ser una estrategia para evitar la deserción y mortalidad académica presente en el programa de ingeniería de sistemas de la Universidad de Pamplona.

Estos cursos se realizaron con las teorías de los aprendizajes basado en proyectos y basado en problemas y la metodología de aprendizaje de aprender haciendo, los cuales son aplicados para la enseñanza de conceptos de ciencia y tecnología y están tomando importancia en la enseñanza no solo a nivel universitario, sino en la educación primaria y secundaria a nivel mundial.

En este curso se realizaron dos test de diagnóstico, los cuales fueron aplicados antes y después de la realización de las sesiones del curso, en la cual se aplicaron un total de 10 guías elaboradas por el investigador. Estas guías contenían temarios que ayudarían al estudiante a realizar habilidades propias del pensamiento computacional tales como algoritmia, resolución de problemas, pensamiento lógico, matemático entre otros.

Para garantizar la efectividad de la investigación, se tomaron datos de cada una de las actividades realizadas en el curso, adicionalmente se hizo un seguimiento a las notas de los estudiantes participantes en las materias de programación estructurada y programación orientada a objetos, materias del componente profesional del programa.

ABSTRACT

This work deals with the realization and the results obtained from a course of computational thinking, supported by the arduino platform with the purpose of being a strategy to avoid academic dropout and mortality present in the systems engineering program of the University of Pamplona.

These courses were conducted with project-based and problem-based learning theories and learning-learning methodology, which are applied to the teaching of science and technology concepts and are taking importance in teaching not only at the level university, but in primary and secondary education worldwide.

In this course two diagnostic tests were carried out, which were applied before and after the course sessions were carried out, in which a total of 10 guides developed by the researcher were applied. These guides contained agendas that would help the student to realize their own computational thinking skills such as algorithm, problem solving, logical, mathematical thinking among others.

To guarantee the effectiveness of the research, data were taken from each of the activities carried out in the course, in addition to the notes of the students participating in the subjects of structured programming and object-oriented programming, subjects of the professional component of the program

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	13
1.1. PLANTEAMIENTO DEL PROBLEMA.....	13
1.2. JUSTIFICACIÓN.....	13
1.3. DELIMITACIONES	14
1.3.1. Objetivos	14
1.3.2. Acotaciones.....	15
2. MARCO TEÓRICO	16
2.1. PENSAMIENTO COMPUTACIONAL.....	16
2.1.1. Definiciones.....	16
2.1.2. Características.....	17
2.1.3. Fases del pensamiento computacional.....	18
2.1.4. Pensamiento computacional en la academia.....	19
2.2. MODELOS DE APRENDIZAJE.	19
2.2.1. Aprendizaje tradicional.	19
2.2.2. Aprendizaje basado en proyectos.	20
2.2.3. Aprendizaje basado en problemas.....	20
2.2.4. Similitudes y diferencias entre el aprendizaje basado en proyectos y el aprendizaje basado en problemas.	21
2.2.5. Metodología de learning by doing (aprender haciendo).	22
2.3. ENTORNOS ESPECIALIZADOS EN EL PENSAMIENTO COMPUTACIONAL.....	22
2.3.1. Logos	22
2.3.2. SmallBasic.	23
2.3.3. Alice.	24
2.3.4. Scratch.	25
2.3.5. Lego Mindstorms.....	26
2.3.6. Raspberry pi.....	27
2.4. ARDUINO	29
2.4.1. Características.	30
2.4.2. Tipología.	31
2.4.3. Dispositivos que se conectan a arduino.	33
2.4.4. Lenguaje de Programación.	33
2.4.5. Tabla comparativa de las distintas formas.	33
3. ESTADO DEL ARTE.	35

3.1.	TRABAJOS DEL AMBITO INTERNACIONAL	35
3.2.	TRABAJOS DE AMBITO NACIONAL	39
3.3.	TRABAJOS DE AMBITO LOCAL	41
3.4.	APRECIACION GENERAL	44
4.	METODOLOGÍA	45
4.1.	PLAN DE DESARROLLO	45
4.2.	MUESTRA Y POBLACIÓN OBJETIVO.	46
4.3.	HERRAMIENTAS Y RECOLECCIÓN DE DATOS.	46
4.3.1.	Herramientas para recolección de datos.....	46
4.3.2.	Recolección de datos.	47
4.4.	ANÁLISIS DE DATOS.	47
5.	DISEÑO ESTRATÉGICO	49
5.1.	GENERALIDADES.	49
5.2.	ELABORACIÓN DE LOS TEST DE PENSAMIENTO COMPUTACIONAL.	49
5.2.1.	Estructura del test de diagnóstico inicial.	50
5.2.2.	Estructura del test de diagnóstico final.....	51
5.3.	ELABORACIÓN DE LAS guías DE PENSAMIENTO COMPUTACIONAL.	51
5.3.1.	Guía 01: Introducción al mundo arduino.	55
5.3.2.	Guía 02: Estructuras de selección.....	56
5.3.3.	Guía 03: Estructuras repetitivas.	58
5.3.4.	Guía 04: Funciones y vectores.....	59
5.3.5.	Guía 05: Librerías.....	61
5.3.6.	Guía 06: Aplicación de las librerías: Conectividad por bluetooth. 62	
5.3.7.	Guía 07: Introducción a la programación orientada a objetos... 63	
5.3.8.	Guía 08: Programación orienta a Objetos: Herencia.....	65
5.3.9.	Guías 09 y 10: Retroalimentación de arduino.	66
6.	ANÁLISIS DE RESULTADOS	68
6.1.	REALIZACIÓN DEL CURSO DE PENSAMIENTO COMPUTACIONAL. 68	
6.1.1.	Introducción.....	68
6.1.2.	Sesión 01.....	69
6.1.3.	Sesión 02.....	77

6.1.4. Sesión 03.....	82
6.1.5. Sesión 04.....	86
6.1.6. Sesión 05.....	89
6.1.7. Sesión 06.....	91
6.1.8. Sesión 07.....	93
6.1.9. Sesión 08.....	97
6.2. TEST DE DIAGNOSTICO DE PENSAMIENTO COMPUTACIONAL.	98
6.2.1. Resultados test inicial.	98
6.1.1. Diagnóstico del test final.	103
6.1.2. Comparativo de los dos test.....	107
6.2. SEGUIMIENTO A LOS ESTUDIANTES.	108
6.3.1. Seguimiento de la materia de programación estructurada.....	108
6.3.2. Seguimiento en programación orientada a objetos.....	112
7. CONCLUSIONES	114
8. RECOMENDACIONES Y TRABAJOS FUTUROS	116
BIBLIOGRAFIA.....	117

LISTA DE TABLAS.

Tabla 1: Componentes de arduino.....	30
Tabla 2: Tipos de arduino	32
Tabla 3: Tabla de comparativo de las plataformas estudia.....	34
Tabla 4: Análisis del contenido de las guías de pensamiento computacional	54
Tabla 5: Inscritos al curso de pensamiento computacional con arduino	68
Tabla 6: Horarios desarrollados en los talleres de pensamiento computacional	69
Tabla 7: Observaciones de la actividad de algoritmos.....	70
Tabla 8: Observaciones de la actividad de datos.....	71
Tabla 9: Aciertos de los estudiantes en el actividad 2 inciso a:	72
Tabla 10: Observaciones de la actividad lúdica de la guía 01	74
Tabla 11: Observaciones de los retos de led parpadeante de la guía 01	76
Tabla 12: Respuesta de la autoevaluación de la primera guía	77
Tabla 13: Observaciones de la actividad de operadores lógicos de la guía 02.	78
Tabla 14: Observaciones de la actividad de operadores relacionales	79
Tabla 15: Observaciones del montaje de led con pulsador de la guía 02.....	81
Tabla 16: Observaciones de la actividad lúdica de la guía 02.	81
Tabla 17: Observaciones del montaje del ultrasonido de la guía 02.....	81
Tabla 18: observaciones de los retos de ultrasonido de la guía 02.	82
Tabla 19: Autoevaluación guía 02.....	82
Tabla 20: Opiniones de las estructuras repetitiva de la guía 03.	84
Tabla 21: Observaciones de la actividad de la guía 03.....	84
Tabla 22: Observaciones de la actividad lúdica de la guía 03.	85
Tabla 23: Observaciones de los retos de la guía 03.	86
Tabla 24: Autoevaluación guía 03.....	86
Tabla 25: Observaciones de las practicas con led de la guía 04.	87
Tabla 26: Observaciones del montaje del sensor de movimiento de la guía 04	88
Tabla 27: Observaciones del montaje del semáforo de la guía 04.	88
Tabla 28: Resultados de la autoevaluación de la guía 04.....	88
Tabla 29: Actividad de librería de la guía 05.	89
Tabla 30: Observaciones de la actividad lúdica de la guía 05.	91
Tabla 31: Resultados de la autoevaluación de la guía 05.....	91
Tabla 32: Observaciones de la actividad lúdica de la guía 06.	92
Tabla 33: Resultados de la autoevaluación de la guía 05.....	93
Tabla 34: Observaciones de la actividad de clases.	94
Tabla 35: Observaciones de la actividad de herencia.....	95
Tabla 36: Observaciones de la actividad lúdica de la guía 07.	96
Tabla 37: Autoevaluación de la guía 07 y guía 08.	96
Tabla 38: Observaciones de los ejercicios realizados en la sesión.	98
Tabla 39: Resultados del test inicial.....	99

Tabla 40: Rangos de calificación de los test de diagnóstico.....	100
Tabla 41: Resultados del segundo test.....	104
Tabla 42: Notas del primer corte de los estudiantes del grupo b.....	109
Tabla 43: Notas de los dos cortes del grupo b de programación estructurada.	110
Tabla 44: Notas definitivas de los estudiantes del grupo b.....	111
Tabla 45: Notas del primer corte de programación estructurada del grupo a.	112
Tabla 46: Notas del primer corte de programación estructurada del grupo b.	113

LISTA DE FIGURAS

Figura 1: Ejemplo de un programa en logos.....	23
Figura 2: Entorno de desarrollo de SmallBasic.....	24
Figura 3: IDE alice.....	25
Figura 4: Entorno de desarrollo de scratch.....	26
Figura 5: Ejemplo de robot hecho con lego mindstorms.....	27
Figura 6: Placa de Raspberry pi.....	28
Figura 7: Computadora armada con una placa raspberry pi.....	29
Figura 8: Placa e IDE de arduino.....	29
Figura 9: Montaje del led pulsador.....	56
Figura 10: Practicas de la guía 2, a) led con pulsador b) sensor ultrasonido	57
Figura 11: Montajes de la guía 03.....	59
Figura 12: Fotografía del montaje explicado en la practica.....	60
Figura 13: Sensor de temperatura y humedad con arduino.....	62
Figura 14: Practicas de la guía 06.....	63
Figura 15: Montaje de la guía 08.....	66
Figura 16: Montajes de las guías 09 y 10.....	67

LISTA DE GRÁFICAS

Gráfica 1: Resultados de las actividades de algoritmia de la guía 01.....	71
Gráfica 2: Resultados de las actividades de datos de la guía 01.....	72
Gráfica 3: Resultados de la actividad de constantes y variables de la guía 01	73
Gráfica 4: Resultados de la actividad de operadores lógicos de la guía 02..	78
Gráfica 5: Resultados de la actividad de operadores relacionales de la guía 02.	80
Gráfica 6: Resultados de la actividad de estructuras repetitivas.....	85
Gráfica 7: resultados de la actividad de librerías.....	90
Gráfica 8: Resultados de la actividad de clases de la guía 07.....	94
Gráfica 9: Resultados de las preguntas relacionadas con pensamiento lógico- matemático del primer test.....	100
Gráfica 10: Resultados de las preguntas relacionadas con reconocimiento de patrones del primer test.....	101

Gráfica 11: Resultados de las preguntas de comprensión lectora del primer test.	102
Gráfica 12: Resultados de las preguntas de pensamiento matemático del primer test.	103
Gráfica 13: Resultados de las preguntas de pensamiento lógico matemático del segundo test.	104
Gráfica 14: Resultados de las preguntas de reconocimiento de patrones del segundo test.	105
Gráfica 15: Resultados de la preguntas de comprensión lectora del segundo test.	106
Gráfica 16: Resultados de las preguntas de pensamiento matemático del segundo test.	106
Gráfica 17: Curva de rendimiento de los test de pensamiento computacional.	107
Gráfica 18: Comparativo por categorías evaluadas.	108

1. INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

El programa de ingeniería de sistemas en la Universidad de Pamplona, presenta un alto grado de deserción y mortalidad académica, debido a que un gran número de estudiantes que ingresan no son capaces de aprobar las materias del componente profesional como programación estructurada y del componente básico como calculo diferencial, la cual son base del programa.

Una de las primeras causas de la deserción del programa, viene desde la educación básica y secundaria con la cual el estudiante llega al programa, ya que este no cuenta con los conocimientos suficientes en áreas como matemáticas y ciencias, las cuales son fundamentales para el desarrollo del programa de ingeniería de sistemas, lo cual lo lleva a cancelar o perder las materias en las cuales requiera emplear dichos fundamentos, especialmente la asignatura de programación estructurada, la cual se cursa en el primer semestre, y como consecuencia el estudiante queda atrasado en su línea de pensum, generando esto perdida de motivación para la continuidad del programa.

El primer semestre es donde se presenta la mayor grado de deserción del programa, debido a que el estudiante, al no aprobar o cancelar materias como programación estructurada y calculo diferencial,. Prefiere cambiar de carrera ya que no encuentra las herramientas necesarias para su aprobación.

Surge el interrogante ¿existen métodos que permitirán evitar la deserción y mortalidad académica en el programa de ingeniería de sistemas? Este proyecto planteó ser una alternativa de solución a dicho interrogante, el cual busco que el estudiante desarrolle un pensamiento con el cual le permita enfrentarse a los retos que demanda el programa, permitiéndole continuarlo y concluirlo para así, ser de gran aporte para la sociedad.

1.2. JUSTIFICACIÓN

Para un estudiante de ingeniería de sistemas, el pensamiento computacional abarca diferentes aspectos, ya que gracias a este puede desarrollar un mayor análisis, razonamiento, extracción de información y solución de problemas, el cual se convertirán en una habilidad indispensable para su desarrollo profesional y garantice la permanencia y conclusión del programa por parte del mismo.

Gracias al pensamiento computacional, se puede encontrar soluciones a problemas cotidianos aplicando técnicas computacionales, como, por ejemplo, identificar los colores del más claro al más oscuro en un cuadro. Para esto no

se requiere de un equipo de cómputo, sino que el individuo desarrolle un nivel de abstracción e identificación de patrones y de forma mental desarrolle una serie de pasos (algoritmo) para poder ejecutar la tarea. Por esto es importante que un estudiante de ingeniería de sistemas aprenda a desarrollar dicha habilidad, ya que le permite triunfar no solo en el ámbito profesional, sino en cualquier situación cotidiana que se le presente durante su vida.

Este proyecto buscó aparte de introducir al estudiante al mundo del pensamiento computacional, incentivarlos para continuar en el programa de ingeniería de sistemas, mediante el desarrollo de una programación más didáctica y entretenida, que aparte de crear su interés, fomentó el desarrollo en su forma de pensar que le permita avanzar en sus estudios, principalmente en las áreas donde involucre tecnologías de la información e informática.

Existen diferentes herramientas para el desarrollo del pensamiento computacional, dispositivos como arduino, lego mindstrom, raspberry y software como scratch, logo, etc., los cuales son muy utilizados. Arduino es un dispositivo electrónico que brinda características y posibilidades interesantes para desarrollar de forma didáctica proyectos, gracias a su simplicidad y manejabilidad. Este lenguaje, al ser multiparadigma, permitió no solo realizar prácticas con programación estructurada, sino con otros paradigmas, en específico el paradigma orientado a objetos, asignatura que precede a esta. Otra ventaja de arduino, es que fue creado con fines educativos, lo que permitió la opción de utilizarlo para desarrollar el pensamiento computacional con los estudiantes.

1.3. DELIMITACIONES

1.3.1. Objetivos

- **Objetivo general.**
 - Desarrollar el pensamiento computacional apoyado por dispositivos electrónicos y arduino en estudiantes de primer semestre de ingeniería de sistemas.
- **Objetivos específicos**
 - Realizar un estudio del desarrollo del pensamiento computacional utilizando arduino.
 - Diseñar talleres orientados al desarrollo del pensamiento computacional apoyado en dispositivos electrónicos y arduino.

- Validar el curso generando una prueba piloto a un grupo de estudiantes analizando los resultados.

1.3.2. Acotaciones

- Se generará una serie de guías de trabajo, para al menos 10 sesiones donde se aborde el pensamiento computacional orientado al trabajo con arduino.
- La validación se aplicará a un grupo de estudiantes de la universidad de pamplona.
- En los talleres no se desarrollará conceptos de electrónica, solo conceptos del pensamiento computacional y programación.

2. MARCO TEÓRICO

Este proyecto tiene como eje central, el pensamiento computacional el cual incursionó en el ámbito profesional en 2006, cuando una investigadora lo mencionó en una de sus investigaciones, y desde ese entonces se ha visto como una alternativa para revolucionar la educación de la sociedad, en la cual la tecnología y la información se ha convertido en un “habito” no solo para el consumidor, sino para los desarrolladores, programadores y personas afines a estos campos.

2.1. PENSAMIENTO COMPUTACIONAL.

2.1.1. Definiciones.

Para entender mejor el pensamiento computacional, se debe conocer el significado de las palabras que la componen. La RAE (real academia española), define pensamiento como el conjunto de ideas propias de una persona, de una colectividad o de una época y computación como el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras.

Por el momento no hay una definición concreta de lo que es el pensamiento computacional, sin embargo, una de las más utilizadas es la que planteó Jannette Wing (Wing, 2006) esta investigadora de la universidad de Columbia la cual lo define como “un enfoque para resolver problemas, diseñar sistemas y comprender el comportamiento humano que se basa en conceptos fundamentales para la computación”. Wing, una de las impulsoras del movimiento del pensamiento computacional, clarificó que el pensamiento computacional debía incluir procesos de pensamiento para formular y solucionar problemas de tal forma que sean asimilados por una computadora. (Wing,2008).

Otra definición de pensamiento computacional, la aporta la asociación de maestros de la ciencias computacionales (CSTA) en conjunto con la sociedad internacional de tecnología de la educación (ISTE), la cual lo definen como “un enfoque para resolver un determinado problema que empodera la integración de tecnologías digitales con ideas humanas. No reemplaza el énfasis en creatividad, razonamiento o pensamiento crítico, pero refuerza esas habilidades al tiempo que realiza formas de organizar el problema de manera que el computador pueda ayudar” (Boasing Olave, 2015).

Román, Pérez y Jiménez (2015), cita la definición que da Google, esta compañía define el pensamiento computacional como “un conjunto de habilidades y técnicas de solución de problemas, que los ingenieros de

software usan para escribir los programas informáticos que subyacen a las aplicaciones que usamos a diario”.

Para Zapata-Ríos (2015) “al igual que sucede con la música, con la danza o con la práctica de deportes, es clave que se fomente una práctica formativa del pensamiento computacional desde las primeras etapas de desarrollo”. Para tener un buen desempeño del pensamiento computacional, hay que entrenar, así como un futbolista se prepara para un partido, el pensamiento computacional se debe entrenar para poder desarrollarlo.

Desde su postulación, el pensamiento computacional se ha buscado incluirlo en la educación básica primaria y secundaria, como lo argumentan algunos expertos (Blikstein y Wilensky, 2009; Eisenberg, 2002; Rubin y Nemirovsky, 1991; Sáez y Cózar, 2016) “El pensamiento computacional puede integrarse transversalmente en asignatura como matemáticas, lenguaje historia, arte o ciencias”.

Estas definiciones coinciden en que el pensamiento computacional es una alternativa para resolver cualquier tipo de problemas, con la utilización de habilidades propias de los informáticos. Sin embargo, el pensamiento computacional no es algo exclusivo de aquellos profesionales que estén relacionado con la informática, sino que cualquier persona puede aplicar el pensamiento computacional (Basogain, et al, 2015).

2.1.2. Características

En su investigación, Bassogain (2015), citando a Wing (2011), el pensamiento computacional presenta 19 características, las cuales son:

1. Reformular un problema a uno parecido que sepamos resolver por reducción, encuadrarlo, transformar, simular.
2. Pensar recursivamente.
3. Procesar en paralelo.
4. Interpretar códigos como datos y datos como códigos.
5. Generalizar análisis dimensional.
6. Reconocer ventajas y desventajas del solapamiento.
7. Reconocer coste y potencia del tratamiento indirecto y llamada a proceso.
8. Juzgar un programa por simplicidad de diseño.
9. Utilizar abstracción y descomposición en un problema complejo o diseño de sistemas complejos.
10. Elegir una correcta representación o modelo para hacer tratable el problema.
11. Seguridad en utilizarlo, modificarlo en un problema complejo sin conocer cada detalle.
12. Modularizar ante múltiples usuarios.
13. Prefetching y catching anticipadamente para el futuro.
14. Prevención, protección recuperarse de escenario de peor caso.
15. Utilizar el razonamiento heurístico para encontrar la solución.
16. Planificar y aprender en presencia de la incertidumbre.

17. Buscar, buscar y buscar más.
18. Utilizar muchos datos para acelerar la computación.
19. Límite de tiempo/espacio y memoria/potencia de procesado.

Para Berrocoso, Sánchez y Arroyo (2015), el pensamiento computacional al redactar un código de programación, se aprende a organizar un proceso, reconocer las rutinas y/o repeticiones e identificar fallos en su pensamiento computacional, principalmente cuando el programa no cumple las expectativas del creador. Estos autores también sugieren que la “codificación se pueden construir aprendizajes significativos desde un punto de vista individual, social, cultural y tangible que conduzcan a una participación computacional” (Berrocoso, et al, 2015).

2.1.3. Fases del pensamiento computacional.

El pensamiento computacional, para su desarrollo presenta una serie de pasos o fases, López Nerea (2016) expone las siguientes fases en su orden de ejecución:

- Recopilación de datos.
- Análisis de datos.
- Representación de datos.
- Descomposición de datos.
- Abstracción.
- Algoritmos y procedimientos.
- Automatización.
- Simulación.

Román, Pérez y Jiménez (2015), denotan como fases del pensamiento computacional 4, las cuales son las más utilizadas por la mayoría. Estas fases en su orden de ejecución son:

1. Descomposición: El proceso de reducir el problema en subproblemas los cuales sean más fáciles de solucionar, es decir reducir la complejidad del problema para encontrar soluciones optimas.
2. Reconocimiento de patrones: En esta fase, se busca encontrar elementos similares que sean útiles que permitan encontrar uno o varios patrones de solución del problema.
3. Abstracción: El proceso en el cual se selecciona la información útil para la solución del problema. Para Pérez Angulo (2019), la diferencia con respecto al reconocimiento de patrones recae en “la capacidad para filtrar e ignorar toda la información que no es necesaria para resolver un problema determinado, y de generalizar la que sí es necesaria”.

4. Algoritmia: también conocida como diseño algorítmico, es la capacidad de realizar una serie de pasos ordenados y finitos que permitan la solución del problema.

2.1.4. Pensamiento computacional en la academia.

Pérez Angulo (2019) resalta que desde el momento que se dio a conocer el concepto de pensamiento computacional, este se ha popularizado convirtiéndose en un “concepto relevante porque la resolución de problemas es una de las habilidades necesarias del siglo XXI”. Esto se evidencia principalmente en los centros de educación, llámese escuelas, colegios, universidades ya que esta buscando “revolucionar” la educación tradicional en la cual el estudiante actúa como una “esponja” ya que solo recibe conocimiento de un docente o tutor sin cuestionarse dicho conocimiento.

Para Basogain, citando a Brown y otros, afirma que “la creación y puesta en marcha de un curso sobre pensamiento computacional, en el que el maestro va a realizar su labor docente en el aula apoyado por una plataforma educativa, y el estudiante va a desarrollar su conocimiento computacional a través de tecnología y metodología educativa”. Este mismo autor, hace énfasis en la colaboración de la real academia de tecnologías avanzadas (RENATA) para introducir el pensamiento computacional en las escuelas, teniendo como prototipo a Bogotá, con el curso PC-01(introducción al pensamiento computacional) y en la actualidad ya se implementa en al menos 11 instituciones educativas.

Muchos países alrededor del mundo han empezado a incluir entre su plan de estudios obligatorio el desarrollo del pensamiento computacional (Román, Pérez y Jiménez,2015). Esto a raíz que con este se desarrollan nuevas habilidades que le permiten al estudiante adaptarse a un mundo más pensante, por el auge de la expansión del conocimiento que se adquiere.

2.2. MODELOS DE APRENDIZAJE.

2.2.1. Aprendizaje tradicional.

El modelo tradicional o memorista, el cual se caracteriza por “la conservación del orden de cosas y para ello el profesor asume el poder y la autoridad como transmisor esencial de conocimientos, quien exige disciplina y obediencia, apropiándose de una imagen impositiva, coercitiva, paternalista, autoritaria, que ha trascendido más allá de un siglo y subsiste hoy día” (Van Arcken).

Sin embargo, este método como lo afirma Benítez y Mora, los cuales exponen que muchos expertos en la materia “han encontrado que la enseñanza tradicional tiene escasa efectividad en lograr un cambio conceptual aceptable de los conceptos de la Física, estas deficiencias han promovido la necesidad de un cambio en el tipo de enseñanza de la misma y se han propuesto diversas metodologías de enseñanza para incrementar la ganancia conceptual en el aprendizaje” (Benítez y Mora, 2010). En caso de ingeniería, donde las áreas como la matemática y la física son sus bases, lo cual la enseñanza tradicional es poco recomendable.

2.2.2. Aprendizaje basado en proyectos.

El aprendizaje basado en proyectos (ABPr) como la define Martí y otros (2019) es un “modelo de aprendizaje en el cual los estudiantes trabajan de manera activa, planean, implementan y evalúan proyectos que tienen aplicación en el mundo real más allá del aula de clase”. Otra definición la ofrece Alcober, Ruiz y Valero (2003), la cual definen la ABPr como el “aprendizaje que se produce como resultado del esfuerzo que realiza el alumno al resolver un problema”.

Otra definición del ABPr, la enunciada por Sánchez, la cual la define como un “conjunto de tareas basadas en la resolución de preguntas o problemas a través de la implicación del alumno en procesos de investigación de manera relativamente autónoma que culmina con un producto final presentado ante los demás.” El aprendizaje basado en proyecto, requiere de un gran compromiso por parte tanto del estudiante como del maestro, ya que esta requiere de una retroalimentación por parte de los involucrados en ella.

El ABPr busca cambiar la forma de enseñanza, como expone Labra Gayo (2006), ya impartir una clase con un docente que expone una temática y al final del curso, se aplique un examen es muy cuestionado en muchas ocasiones. Con el ABPr, se busca que el estudiante y el docente asuman retos que le ayuden a aprender a las dos partes.

2.2.3. Aprendizaje basado en problemas.

Este aprendizaje, del cual se deriva el ABPr, ya que fue implementado por primera vez en la Universidad de McMaster en Canadá, en estudiantes de medicina con el propósito conseguir mejores profesionales. El aprendizaje basado en problemas (ABP) según Barrow consiste en “un método de aprendizaje basado en el principio de usar problemas como punto de partida para la adquisición e integración de los nuevos conocimientos” (Barrow, 1996).

Para McGrath, se define como “método de aprendizaje en grupo que usa problemas reales como estímulo para desarrollar habilidades de solución de problemas y adquirir conocimientos específicos” (McGrath,2012),

Para Dueñas, “el enfoque de ABP, el estudiante decide cuáles contenidos o temas deberá abordar o estudiar para resolver los problemas o casos objeto de estudio el propio estudiante se propone objetivos instruccionales al identificar las necesidades de aprendizaje y evalúa permanentemente su aprendizaje y la adquisición de habilidades, competencias y actitudes”, es decir, el estudiante tiene la autonomía de determinar que conceptos son útiles para la solución del problema planteado.

2.2.4. Similitudes y diferencias entre el aprendizaje basado en proyectos y el aprendizaje basado en problemas.

El aprendizaje basado en proyectos, por lo general suele confundirse con el aprendizaje basado en problemas (ABP), comenzando por sus abreviaturas, ya que muchos textos suelen asociarlo como similares y a pesar de que comparten muchas cosas en común, tienen ciertas diferencias que los diferencian.

Para Rodríguez-Sandoval (2010) el aprendizaje basado en problemas, se encarga de solucionar un problema en específico, mediante la investigación y la retroalimentación de la forma de solucionarlo, mientras que el aprendizaje basado en proyectos, busca mediante estrategias creativas para solucionar un problema específico que cree un producto final. Con esta aclaración, se podría decir que el ABP es más aplicable en casos de investigación y generación de nuevos conocimientos mientras que el ABPr para aquellos que generen prototipos e innovaciones.

El ABP, es muy utilizado en áreas como la medicina, ya que este nació en la escuela de medicina de la Universidad de McMaster (Canadá), con el objetivo de motivar a los estudiantes. En cuanto al ABPr, surge como una variante del ABP, aplicable al campo de la ingeniería y más adelante en la informática (Labra Gayo, 2006).

Tanto el ABP, como el ABPr, se encarga de solucionar problemas, aplicando métodos que permiten realizar una retroalimentación, requieren del trabajo en equipo, el docente es más una guía que alguien que imparte conocimiento. También en estos métodos de aprendizaje requieren de una planificación ordenada para encontrar soluciones lógicas y válidas.

2.2.5. Metodología de learning by doing (aprender haciendo).

Desde Aristóteles, el cual decía que lo que “tenemos que aprender a hacer, lo hacemos aprendiendo”, acciones como caminar, respirar, hablar etc. Aunque no hay un método de enseñanza, las aprendemos ya sea intuitivamente u observando al entorno como se realizan. De esta forma funciona la metodología de aprender haciendo (learning by doing en ingles).

La metodología de aprender fue impulsada por Schank (2002), esta metodología consiste en “sumergir al alumno en un ambiente de situaciones controladas, una simulación interactiva, sin ningún tipo de castigo ni reprobaciones, sólo mostrándole las consecuencias de sus errores y cómo prevenirlos, dando la libertad al alumno para que pueda”. Con esto, el estudiante aprende de las experiencias vividas más que de un texto o de un docente que imparta ciertos conocimientos. Otra definición, la aporta Llorente Barroso (2013), la cual “mediante un planteamiento del problema, se propone a los estudiantes un aprendizaje a través de la aplicación práctica de los conocimientos que adquieren en las clases regulares”.

El aprender haciendo, busca una forma de revolución de la enseñanza tradicional, ya que la enseñanza en la cual un docente imparte una cátedra, esta es absorbida por el estudiante, principalmente por el uso de las tecnologías de la información y las comunicación (TIC) en la sociedad actual (Rodríguez García *et at*, 2014).

2.3. ENTORNOS ESPECIALIZADOS EN EL PENSAMIENTO COMPUTACIONAL.

2.3.1. Logos

El lenguaje pionero en la enseñanza del pensamiento computacional es Logos, un lenguaje que se creó en la década de los 60, por el matemático Seymond Paper. Fue inventado con fines educativos, principalmente para niños. Su objetivo era que el estudiante aprendiera de una forma más eficaz determinadas materias y a su vez, desarrollara un pensamiento computacional.

Logos era un lenguaje basado en Lisp, (un lenguaje orientado a la matemática) con el propósito de la enseñanza de la matemáticas en los niños. Logos era un lenguaje de programación sencillo, poseía pocas ordenes como forward, *left* o *right*, acompañado de distancias numéricas que tenía que recorra una tortuga (la tortuga era la imagen de logos), esta realizaba trazos en la pantalla. En logos, con otras ordenes se podía cambiar el trazo, el color de la ruta, la velocidad que avanza la tortuga. Logos era el lenguaje apropiado para la enseñanza de la programación en los niños como lo muestra la figura 1.

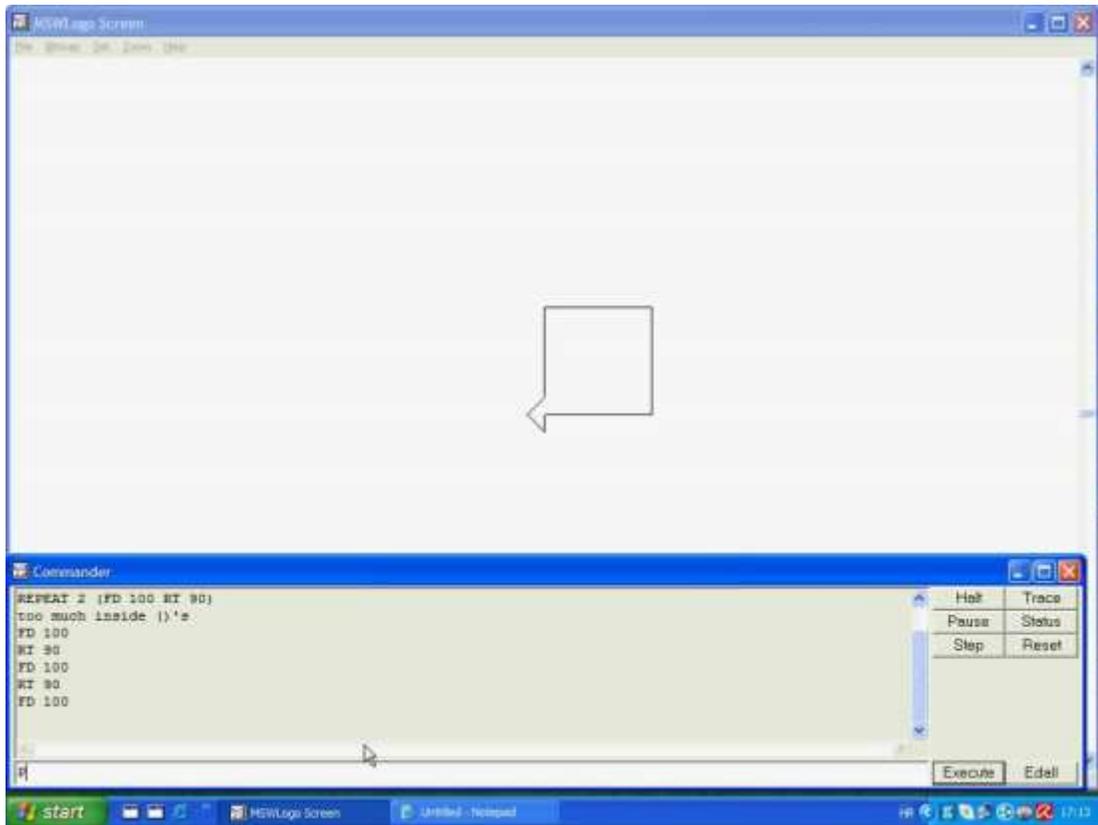


Figura 1: Ejemplo de un programa en logos¹.

Logos también tenía un pequeño robot, que ayudaba a ver lo que se estaba programando, siendo uno de los primeros en combinar la robótica con la programación. De ahí, empiezan a surgir otros lenguajes de programación especializadas en la enseñanza del pensamiento computacional tales como lego y scratch. Sin embargo, logos desapareció quedando el recuerdo de lo que fue un lenguaje de programación orientado a la enseñanza de los niños, siendo entendible y didáctico.

2.3.2. SmallBasic.

Es un lenguaje de programación sencillo, diseñado por Microsoft para hacer la programación sencilla y divertida, ya que está dirigido principalmente a niños. SmallBasic está inspirado en BASIC original, con tan solo 15 palabras reservadas, las que evidencian la sencillez del lenguaje.

Su entorno de desarrollo, es muy simple ya que, al abrirse su IDE, el cual colorea el código (como otros IDE como dev c++, NetBeans, Matlab etc.) y con

¹ Fuente: <https://brungumacho.gq/photo/603692.jpg>.

una barra de tareas similar a la de office en la cual contiene los botones necesarios para su funcionamiento como se muestra en la figura 2.

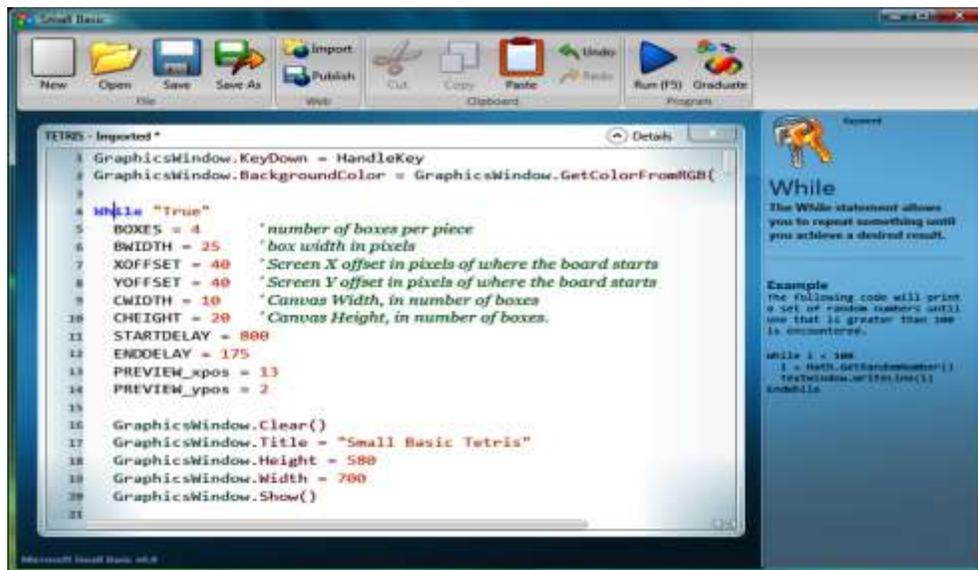


Figura 2: Entorno de desarrollo de SmallBasic²

SmallBasic funciona en plataformas .NET y posee el mismo paradigma de BASIC, y presenta conceptos minimalistas con el propósito de mantener su complejidad, para que las personas logren comprenderlo.

2.3.3. Alice.

Alice es un entorno de desarrollo orientado a animaciones, orientado a la realización de juegos que maneja el paradigma orientado a objetos. Fue creado por Randy Pausch, profesor de la universidad de Carnegie Mellon en Estados Unidos, aunque el software fue desarrollado primeramente en la Universidad de Virginia. Alice se creó con el objetivo de solucionar tres problemas fundamentales de los software educativos, los cuales eran la enseñanza a la programación, está ligado a su entorno de desarrollo y está diseñado a usuarios que no se enfrentan al mundo de la programación.

Con alicé se puede aprender conceptos fundamentales de la programación mediante la animación y los videojuegos sencillos. Alice no es un lenguaje como tal, ya que para crear un aplicación solo hay que arrastrar los objetos al entorno.

² Fuente:
https://www.techworld.com/cmsdata/downloads/4104/largeImg_thumb800.png?version=1%2E2



Figura 3: IDE Alice³.

La interfaz de alice es muy intuitiva, ya que el usuario solo debe arrastrar y soltar los bloques que necesita para crear su animación (ver figura 3). Alice está en su versión 3.0.0.1 Beta, disponible para Windows y Mac OS, y no tiene ningún costo, lo que lo hace accesible a todo el mundo. Con alice se han hecho proyectos como el pavo enamorado, el encuentro en el mar, feliz hipopótama y feliz Halloween. Estos proyectos se encuentran explicados en la web oficial de alice.

2.3.4. Scratch.

Uno de los lenguajes más conocidos cuando se habla de pensamiento computacional, Scratch es un lenguaje de programación, dirigido principalmente a niños ya que, mediante historias interactivas, juegos y animaciones, los niños aprenden a programar.

Este lenguaje, creado por MIT media lab. (Instituto de tecnología de Massachusetts), bajo la política de software libre, compatible con todas las plataformas (Windows, Mac OS, Linux). Scratch posee una interfaz intuitiva, la cual permite que el usuario comprenda de una forma sencilla su código, también le permite la manipulación de algunos archivos multimedia (caso de imágenes, fotos, música) para crear una programación visual con estos elementos.

Con este lenguaje, se pueden crear un sinnúmero de historias, mediante la programación de personajes, esto lo lleva a convertirse en un lenguaje de programación ideal para la creación de videojuegos. Scratch realiza acciones y comportamientos tales como movimiento de personajes, cambio de apariencia, agregar sonidos.

³ Fuente: <https://image.slideserve.com/49961/the-ide-l.jpg>

Scratch, maneja programación basada por bloques, basado en logos lo cual lo hace un lenguaje orientado a la enseñanza de la programación en tempranas edades, ya que los principales usuarios suelen ser los niños entre 8 y 16 años. Su interfaz es muy intuitiva, la cual posee varios tipos de movimientos, personajes y efectos que el usuario solo debe llamar como se muestra en la figura:

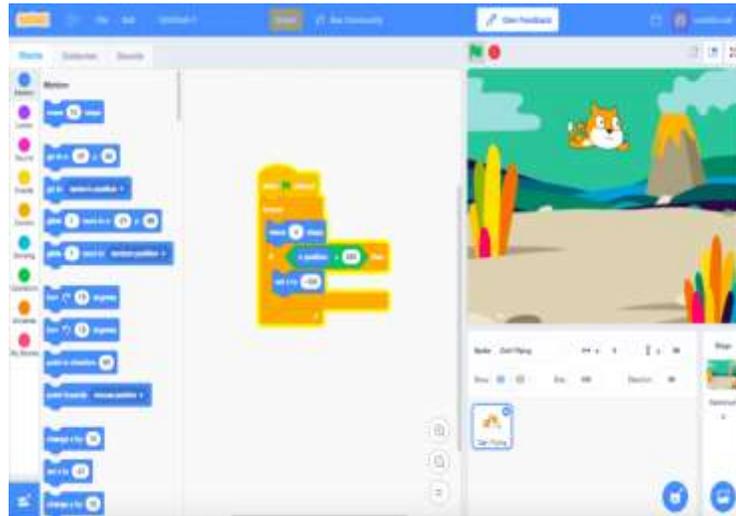


Figura 4: Entorno de desarrollo de scratch⁴

Con Scratch, el usuario aprende a desarrollar habilidades de pensamiento, ayudándoles a pensar metodológicamente la solución de problemas, aprender conceptos matemáticos y algorítmicos, comprendiendo el mundo de la programación.

El programa de scratch, ha recibido apoyo financiero de grandes compañías como Google, Lego, Intel, Cartoon Network, la National Science foundation, entre otras. Algunos proyectos más destacados que se encuentran en la web oficial son: HANABI, Fireworks, Rainbow striker entre otros.

2.3.5. Lego Mindstorms.

Es una línea de Juguetes de lego especializada en la enseñanza de la robótica. Fue lanzada en el año 1998, cuando la compañía creo un microcontrolador de fácil programación y que permite la conexión con sensores y actuadores. Estos robot son de fácil armado y no requieren ningún tipo de soldadura ya que con estas piezas se pueden realizar diferentes modelos como lo muestra la figura 5, y es ampliable con las piezas y componentes LEGO disponibles en el mercado. Sin embargo, por su estructura, la cual son piezas que ensamblan con presión, no se les puede recibir golpes y caídas, ya que se debilitan, no se

⁴ Fuente: https://miro.medium.com/max/2560/0*356zWDYXZyGbmQxa.

pueden construir estructuras circulares debido a la forma de los bloques y, además, que no son accesibles a todo el mundo debido a su alto costo en el mercado (un kit de lego mindstorms puede costar alrededor de los US\$ 800 y US\$ 1000).

El lenguaje de programación propio de mindstorms es NQS (cuyas sigas significa Not quite c), el cual en su sintaxis es parecido a C, aunque estos robot pueden ser programados por otros lenguajes tales como C, java e incluso scratch. Está dirigido principalmente para niños de 10 años en adelante, aunque también es utilizado por adultos ya que está dirigido a personas que desean aprender robótica y programación y su software es operable de una forma sencilla, en el cual el usuario aprende a realizar juegos o aplicaciones visuales de una forma sencilla, es compatible con sistemas operativos Windows y Mac os.



5

Figura 5: Ejemplo de robot hecho con lego mindstorms.

2.3.6. Raspberry pi.

Es una pequeña placa de micro ordenador, con el tamaño de una tarjeta de crédito que es conectado a cualquier periférico de entrada o de salida de bajo costo diseñado en 2006 en Cambridge, Reino Unido (ver figura 6) y comenzado a comercializarse en 2012, su objetivo es la estimulación de la enseñanza de la programación y la robótica en las instituciones educativas.

⁵ Fuente: https://www.zonaoutdoor.es/sites/default/files/lego_mindstorms_ev3_ii.jpg



6

Figura 6: Placa de Raspberry pi

Es considerado un “ordenador desnudo”, ya que se le pueden conectar distintos sensores y periféricos, que no afectan su funciones básicas. Estas tarjetas cuentan con una tarjeta de red, un procesador gráfico, una memoria RAM de 512 GB, conector de red, puertos USB, salidas de video y audio, puertos analógicos conector de alimentación y lector de tarjetas SD.

Raspberry utiliza sistema operativo de base Linux (código abierto) y es de bajo costo, lo que lo hace accesible a todo el público. Este dispositivo es multipropósito, ya que permite una amplia variedad de conexiones. Esta placa, debido a su tamaño y versatilidad, es muy útil para la fabricación y programación de futuros dispositivos móviles como se muestra en la figura 7.

Raspberry al ser una especie de ordenador, no tiene un lenguaje de programación y es instalado como si fuera un sistema operativo, aunque la mayoría de sus aplicaciones pueden ser programadas en Python, aunque se pueden utilizar otros lenguajes como c, java, scratch etc. Actualmente, está la versión 4.0 de la placa la cual se consigue en el mercado por un valor de USD\$ 35.00.

⁶ Fuente:

<https://cdn.computerhoy.com/sites/navi.axelspringer.es/public/styles/480/public/media/image/2014/01/28413-que-es-raspberry-pi.jpg?itok=GOQkVsT7>



Figura 7: Computadora armada con una placa raspberry pi.

2.4. ARDUINO

Arduino es una plataforma que sigue la filosofía de hardware libre y software libre (ver figura 8) en la cual se pueden conectar distintos microcontroladores. Al ser de hardware y software libre, permite desarrollar y aportar ideas para nuevos proyectos que se pueden realizar con dicha placa.

Arduino surgió como un proyecto, de unos estudiantes del instituto de Ivrea, en Italia, recibió el nombre por el bar donde estos estudiantes frecuentaban. Ellos, buscaban la forma de integrar la electrónica con la programación, además de una forma económica para realizar los trabajos, ya que al utilizar las BASIC Stamp, que eran costosas y de poco acceso para el estudiantado.



Figura 8: Placa e IDE de arduino.

Arduino fue creado con los componentes necesarios para conectar dispositivos de entrada y de salida, que se puede utilizar con cualquier plataforma (Windows, Mac OS, Linux). Arduino sigue la filosofía de “learning by doing” (Aprender haciendo).

2.4.1. Características.

La placa de arduino consta de microcontroladores, con puertos de comunicación y de salida. Arduino posee puertos analógicos y digitales, que permiten crear variedad de dispositivos, ya sean medidores, sensores, montajes robóticos, etc. La placa de arduino tiene incorporado un puerto, que mediante USB este conector puede variar dependiendo del tipo de arduino.

Arduino no soporta más de 20 voltios, lo que lo hace una placa para sensores sencillos y su principal fin es de introducir al usuario al mundo de los microcontroladores, la programación y la robótica. La mayoría de los arduinos manejan tensiones entre 6 y 12 Voltios.

Los puertos que maneja arduino (también denominados pines) son por donde se conectan los distintos dispositivos los cuales están debidamente marcados en la plataforma. Estos pines están explicados en la tabla 1.

NOMBRE	CARACTERÍSTICAS	NUMERO DE PINES (ARDUINO ONE)
Puertos GND	También son conocidos como puertos a tierra, los cuales se utilizan para la conexiones que requieran una corriente negativa	Posee 3 pines
Puertos de Alimentación	Son puertos por los cuales se transmiten la corriente positiva del arduino. Estos son los que alimentan los dispositivos conectados.	Posee dos puertos, uno de 3.3 V y otro de 5 V.
Pines analógicos	Son pines que leen las señales analógicas y los convierten de tal forma que puedan ser leídos por el usuario	Posee 6 pines
Pines digitales	Son pines utilizados como conectores para entradas y salidas digitales	Posee 13 pines
Pines PWM	Dentro de los pines digitales, hay unos identificados con el símbolo ~, estos pines son utilizados ya sea como pines normales o como pines de modulación de ancho de pulso (PWM por sus siglas en ingles).	Son 6 pines
Pin AREF (referencia analógica)	Es un pin que se utiliza con sensores de extrema precisión (por ejemplo, un sensor de temperatura) en el cual comprueba la entrada de voltaje que este recibe.	Solo un pin

Tabla 1: Componentes de arduino

Además de estos pines, arduino posee un botón de reinicio, unos led RXTX (recepción, emisión), microcontroladores y un regulador de voltaje.

Arduino trabaja con microcontroladores ATMEL, el cual es el encargado de guardar las instrucciones, que se le dan al arduino mediante su programación. Arduino también maneja una comunicación serial, la cual se produce mediante la comunicación que establece el arduino con cualquier dispositivo al que este enlazado (pc, dispositivos de salida).

En cuanto a su lenguaje de programación, su lenguaje está basado en C ++, lo que le permite ser un lenguaje multiparadigma, capaz de adaptarse a cualquier estilo del programador. La estructura de un programa en arduino, para que funcione correctamente, tiene dos instrucciones esenciales, el setup, donde se configura cada uno de los dispositivos que se conectan además de las instrucciones iniciales al encender el arduino y el loop, el cual es el cuerpo principal del programa, donde se aplican cada una de las instrucciones que tienen que realizar el montaje realizado.

Un programa en arduino es conocido como sketch, su tipo de dato es. ino. Arduino también permite mezclarse con otros lenguajes de programación, tales como java, C ++, Python, MATHLAB, Scratch entre otros. Esto, en caso que el usuario quiera crear una aplicación con interfaz gráfica, dado que arduino no permite la creación de la misma.

2.4.2. Tipología.

En el mercado, existen distintos tipos de arduino, los cuales deben ser seleccionado según el tipo de montaje que se desee realizar, ya que hay algunos con mayor o menor procesamiento. algunos de los arduino más utilizados son explicados en la tabla 2.

NOMBRE	DESCRIPCION	MICROCONTROLADOR	NUMERO DE PINES	VOLTAJE RECOMENDADO	TIPO DE PUERTO USB
Arduino uno (one)	Es el más conocido, documentado y utilizado. Recomendado para principiantes.	Atmega328	14 pines digitales de entrada y salida, de ellos 6 PWM, 6 pines analógicos,	7 – 12 V	B
Arduino Leonardo	Posee un diseño similar al arduino uno, con la diferencia de su puerto USB, el cual está incorporado, lo que le permite funcionar como si fuera un mouse o teclado conectado a una computadora.	Atmega32U4	14 pines digitales de entrada y de salida, de ellos 7 son PWM, 6 pines analógicos	7 -12 V	Micro B
Arduino Mega	Es una placa más potente, ideal para trabajos complejos. Posee un nivel de procesamiento mayor.	Atmega2560	54 pines digitales, de los cuales 15 son PWM, 16 pines analógicos	7 - 12 V	B
Arduino Pro	Es un tipo de arduino que se puede usar de forma permanente, posee las mismas características que el arduino uno (en cuanto a pines, procesador). Pero con un costo menor al uno.	Atmega328	14 pines digitales de entrada y salida, de los cuales 6 son PWM, 6 pines analógicos.	5 – 12 V	Micro B
Arduino nano	Es un pequeño tablero que, aunque pequeño, posee las mismas características de un arduino uno. No tiene conector de alimentación, solo se alimenta por USB.	Atmega328	14 pines digitales de entrada y salida, 6 de ellos de PWM. 8 pines analógicos.	7 - 9 V	AB
Arduino Micro	Posee la potencia de un arduino Leonardo, aunque posee un tamaño más pequeño.	ATmega32u4	20 pines digitales de entrada y de salida, de los cuales 7 son PWM. 12 pines analógicos.	7 -12 V	Micro B
Arduino Mini	Es una placa aún más pequeña que la nano, No posee una entrada USB y se usa principalmente para montajes permanentes.	Atmega328	14 pines digitales de entrada y salida, de los cuales 6 PWM. 8 pines analógicos	7 - 9 V	Puerto USB nativo.
Arduino YUN	Es un arduino especializado para conexiones que requieran internet, ya que posee dos procesadores, además posee lector de tarjetas micro SD,	Atmega 32U4 y Atheros AR9331	20 pines digitales de entrada y salida, de los cuales 7 son PWM, 12 pines analógicos	7 – 12 V	USB

Tabla 2: Tipos de arduino

2.4.3. Dispositivos que se conectan a arduino.

Arduino como tal, es solo una placa de microcontroladores, para realizar cualquier tipo de proyecto, se necesita conectarle ciertos dispositivos, ya sea que reciban o envíen información. Estos dispositivos son de los siguientes tipos:

- **Sensores:** Son dispositivos capaces de detectar magnitudes físicas o químicas, llamadas variables de instrumentación. En el mercado, existen múltiples sensores especializados para una tarea en específico, algunos de los más comunes son los sensores de proximidad, de temperatura, de conectividad, de presión atmosférica, sensores de luz.
- **Actuadores:** Es un dispositivo capaz de formar energía hidráulica, neumática o eléctrica en la activación de un proceso con el objetivo de generar un efecto sobre un agente externo. Existen varios tipos de actuadores: Eléctricos, neumáticos, hidráulicos, eléctricos, motores y bombas.
- **Periféricos:** son los aparatos con los cuales el arduino se comunica con el exterior. Estos periféricos pueden ser de entrada (Cámaras, teclados, botones) o de salida (pantallas, teclados, bombillos).

2.4.4. Lenguaje de Programación.

Como se había mencionado anteriormente, arduino posee un lenguaje de programación propio, basado en C++, aunque la placa puede ser programada en otros lenguajes de programación como scratch. También posee librerías propias, utilizadas para ahorrar la programación de algunos dispositivos que se conectan con el arduino. El lenguaje de arduino al tener como base c, es multiparadigma, lo que lo hace versátil para los usuarios exigentes en cuanto a forma de programar.

2.4.5. Tabla comparativa de las distintas formas.

En la tabla 3 se hace un pequeño resumen de los entornos expuestos en este capítulo, destacando algunos aspectos que permiten que estos desarrollen el pensamiento computacional.

PLATAFORMA	Orientadas al pensamiento computacional	Necesario hardware	Multipataforma (SO)	Tiene soporte (actualidad)	Open Source	Desarrolla Pensamiento matemático	Desarrolla reconocimiento de patrones	Desarrolla pensamiento algorítmico
LOGOS	✓	✓	✓		✓	✓	✓	✓
SMALLBASIC	✓				✓	✓	✓	✓
ALICE	✓		✓	✓	✓	✓	✓	✓
SCRATCH	✓		✓	✓	✓	✓	✓	✓
LEGO MINDSTORM	✓	✓	✓	✓		✓	✓	✓
RASBERRY PI	✓	✓		✓	✓	✓	✓	✓
ARDUINO	✓	✓	✓	✓	✓	✓	✓	✓

Tabla 3: Tabla de comparativo de las plataformas estudiadas

Al realizar el correcto análisis, se llegó a la conclusión de utilizar arduino, porque es una plataforma especializada para el desarrollo del pensamiento computacional, además de ser una alternativa innovadora, ya que anteriormente se había trabajado la enseñanza del pensamiento computacional con las plataformas lego y scratch.

Arduino es una plataforma de bajo costo, comparada con lego y raspberry, lo que lo hace más accesible para el estudiante, además que el programa de ingeniería de sistemas cuenta con kits especializados en el desarrollo de la programación con arduino, evitando que el estudiante invierta dinero obligatoriamente (en adquirir los materiales) y no encuentre excusas en el aprendizaje por falta de los mismos.

Otra razón es que, al manejar tanto software y hardware, llama la atención del estudiante, principalmente de aquellos indecisos que no encuentran una motivación para la continuidad de sus estudios, principalmente en las áreas básicas y profesionales, además de servir de “distracción sana” ya que mientras realizan y programan un determinado prototipo, se divierten en su aprendizaje y recopilan y retienen la información de una forma rápida y segura.

3. ESTADO DEL ARTE.

En este capítulo se abordará las investigaciones que se han realizado en cuanto al pensamiento computacional, en el ámbito internacional, nacional y local. Estas investigaciones, fueron la base para la documentación, elaboración y ejecución de este proyecto de investigación.

3.1. TRABAJOS DEL AMBITO INTERNACIONAL.

Título: Aprendizaje basado en proyectos y arduino en tecnología de 4°ESO

Autor(es): Silvana Victoria Silva Laurelillo.

Año: 2018

Objetivo general: Diseñar una intervención educativa para el alumbrado de tecnología de 4° de ESO que emplee la metodología del aprendizaje basado en proyectos para solucionar problemas de la vida real, mediante la herramienta arduino.

Descripción: Esta investigación propone una unidad didáctica aplicando el aprendizaje basado en proyectos con la plataforma arduino orientado a estudiantes de 4° (14 y 15 años) de la ESO (educación secundaria obligatoria). Para esta se realizaron un total de ocho actividades enfocadas en la resolución de proyectos, de una institución educativa de Almería, España, de la materia de tecnología. El investigador, apoyado con el docente del área realiza una serie de evaluaciones diarias para evaluar el desempeño de los estudiantes y el impacto de la aplicación de esta estrategia.

Resultados: Fue una estrategia que fue implementada, en la cual se encontraron deficiencias como el tiempo de realización de las actividades, el conocimiento que debe tener el estudiante, el compromiso del estudiante entre otras. Fortalezas como la fomentación del trabajo en equipo, el learning by doing, la creatividad entre otros aspectos.

Apreciación: Esta investigación fue importante ya que sirvió para conocer los conceptos del aprendizaje basado en proyectos, el aprender haciendo y de los posibles aciertos y riesgos que pudo tener este proyecto al momento de ser ejecutado.

Título: Aprendizaje Basado en Proyectos con la plataforma Arduino en el desarrollo de las competencias de programación en la Universidad Privada del Norte-2017.

Autor(es): Miguel Liévano.

Año: 2017

Objetivo general: evaluar la influencia del Aprendizaje Basado en Proyectos con la plataforma arduino en el desarrollo de competencias de programación de los estudiantes de Ingeniería de Sistemas computacionales.

Descripción: Esta investigación busca demostrar el impacto que genera la enseñanza basada en proyectos en los estudiantes de ingeniería de sistemas y computación de la universidad privada del norte en el Perú. Este proyecto se hizo con arduino y fue de forma experimental. Se midieron sus resultados mediante la aplicación de encuestas y test a dos grupos de estudiantes, unos que habían trabajado en un montaje robótico con arduino y otros a los que no.

Resultados: El aprendizaje basado en proyectos es una influencia positiva para el estudiante que cursaba la carrera de ingeniería de sistemas computacionales.

Apreciación: Con esta investigación se pudo observar el impacto que podría generar el aprendizaje basado en problemas, estrategia para que un estudiante desarrolle el pensamiento computacional y arduino como herramienta que garantizaría el éxito del proyecto.

Título: **Planificación anual del taller de informática 2017. Programación y resolución de problemas Scratch para Arduino.**

Autor(es): Corinne Schoefer y Carolina Alonso.

Año: 2017.

Objetivo general: Desarrollar un plan de estudios para la enseñanza del pensamiento computacional.

Descripción: En este trabajo de investigación realizado en Argentina, muestran la forma de organizar los proyectos para la enseñanza del pensamiento computacional, teniendo como herramientas arduino y scratch. La enseñanza, se hace mediante proyectos sencillos que abordan los conceptos básicos de programación. La evaluación de la efectividad, fue mediante una autoevaluación y una coevaluación que acompaña cada una de las practicas.

Resultados: El resultado fue una investigación de la temática adecuada para la enseñanza del pensamiento computacional con diferentes herramientas, terminando con Scratch para arduino.

Apreciación: Fue importante para plantear las temática, dificultad, evaluación y organización del material de trabajo que se pretendía realizar, ya que una buena planificación garantizaba que el estudiante se sintiera cómodo con la temática, ya que la idea era no sobrecargarlos de información y que esta se tornada confusa e impidiera el desarrollo de sus actividades académicas.

Título: Primera exposición a Arduino a través del entrenamiento entre pares: Impacto en los estudiantes actitudes hacia la programación.

Autor(es): Pablo Martín-Ramos, Maria João Lopes, M. Margarida Lima da Silva, Pedro E.B. Gomes, Pedro S. Pereira da Silva, José P.P. Domingues, Manuela Ramos Silva.

Año: 2017.

Objetivo general: Enseñar programación aplicando arduino y la metodología basada en proyectos.

Descripción: Esta iniciativa realizada por los estudiantes de la universidad de Combra, consistía en la enseñanza de la programación y la robótica aplicando el aprendizaje basado en problemas, mediante la plataforma arduino, dirigido a estudiantes que tenían o no contacto con este mundo. Para esto se utilizó un portal, en el cual el estudiante descargaba la práctica las cuales estaban relacionadas con la programación. Después de realizado estos cursos, el estudiante debía llenar una encuesta la cual serviría como referencia para medir el aprendizaje de los estudiantes participantes. Las clases se impartían en forma de tutorías.

Resultados: La encuesta fue realizada a 26 estudiantes de grado 11 (entre los 16 y 17 años) y a 18 estudiantes de educación superior (universidad). Los cuales respondieron la encuesta de forma voluntaria, estos encuestados fueron separados por género y edad. Por cada sesión se originó una serie de preguntas en las cuales el estudiante dio su punto de vista la cual serviría para analizar la aceptabilidad. En la encuesta se dejó un espacio para que diera su punto de vista, el cual ellos daban su opiniones.

Apreciación: Con este proyecto, dio la base para los temas de programación incluidos en las guías, no con las practicas ya que se buscó explotar al máximo los kits de arduino con los que cuenta el programa. Además de cómo se aplica el aprendizaje basado en proyectos y el pensamiento computacional en dos tipos de estudiante, teniendo en cuenta que, en los cursos de pensamiento computacional, podría llegar personas que hubieran tenido ya contacto con la programación. Además de la forma de abordar la enseñanza de la programación cuando se realiza en forma de tutorías.

Título: Diseño e implementación de un laboratorio remoto para la enseñanza de la programación y la robótica.

Autor(es): João Paulo Cardoso de Lima, Lucas Mellos Carlos, José Pedro Schardosim Simão, José Pereira, Paulo Manoel Mafra, Juarez Benito Da Silva.

Año: 2016

Objetivo general: Enseñar la robótica y programación mediante la utilización de un laboratorio remoto apoyado en arduino.

Descripción La presente investigación tenía de problema la falta de laboratorios modernos para la enseñanza de la robótica y la programación, a causa que en las regiones que están en desarrollo no poseen un laboratorio para dicha enseñanza, además de incrementar el interés por su enseñanza en los establecimientos educativos. En este proyecto, se hace todo de manera virtual, ya que el usuario ingresa al laboratorio mediante un navegador web o una aplicación móvil, por lo que es compatible con dispositivos Android. El usuario crea el código, en lenguaje arduino, el usuario sube el código y de forma remota. En cuanto al montaje, existe uno el cual tiene conectado un arduino uno, led, sensor de temperatura, sensor ultrasónico, servomotor, sensor paso a paso, potenciómetro, sensor Hall y sensor fotoeléctrico.

Resultados: La prueba se aplicó a 18 personas, de las cuales 13 eran estudiantes entre los 14 a 16 años y los restantes fueron maestros cuyas edades están entre los 25 a 38 años, de los cuales el 38.9%, ya tenía un conocimiento previo en áreas de la programación. Se les aplicó un cuestionario el cual pretendía medir la aplicabilidad de este laboratorio en cuanto a la app utilizada, al desarrollo de la interfaz y a la compilación del programa, los cuales presentaron una sobresaliente aceptación por parte de los indagados.

Apreciación: Esta investigación fue importante ya que permitió observar una estrategia para la enseñanza del pensamiento computacional con arduino. Con esta se podía apreciar que la enseñanza con arduino podría traer buenos resultados a esta investigación, principalmente en la forma de despertar el interés no solo por la enseñanza con arduino, sino con la programación en sí, problema que muchos estudiantes que cursan la materia de programación estructurada presentan y de reforzar los conceptos de pensamiento computacional.

Título: Test de Pensamiento Computacional: diseño y psicometría general.

Autor(es): Marcos Ramón-González, Juan Carlos Pérez-González, Carmen Jiménez-Fernández.

Año: 2015.

Objetivo general:

Descripción: Este trabajo, es reconocido por la implementación de un test de pensamiento computacional, el cual en su principio contenía 40 preguntas que fueron disminuidas a 28 por expertos en el área. Esto se hizo a raíz que no existía un método para medir el pensamiento computacional en sus distintas etapas y que el autor considero que se debía abordar. Este test tenía como objetivo la aptitud – desarrollo del pensamiento computacional en los sujetos a los que se les fuera implementado, los cuales fueron los cursos 1° y 2° de la ESO (12 y 13 años de edad). El desarrollo en cuanto a tiempo era de 45 minutos. El test consistía en tres tipos de preguntas, unas preguntas de bucles

repetir veces, en el cual debía responder la dirección en la que Pacman (videojuego muy famoso de los años 80) se desplaza hacia donde está el fantasma. El segundo estilo de preguntas es similar al anterior, pero en esta ocasión se hace con programación por bloques, es decir el encuestado debía responder la(s) orden(es) exactas del Pacman para llegar al fantasma. El tercer tipo de pregunta era con un lienzo, el cual, mediante la programación por bloques, se debía determinar las ordenes exactas para formar cierta figura. Este test se encuentra de manera virtual, ya que se hizo como una encuesta de Google. Con este trabajo, se planteó la forma de medir al estudiante que ingrese al taller de pensamiento computacional, además de la forma de plantearse las preguntas que permitirían la realización de dicho test.

Resultados: Se aplicó a una muestra de 400 estudiantes de la población objetivo, de nueve instituciones de las tres provincias de la comunidad Valenciana, que están previamente integrados en la red de innovación IT Teaching, En las tablas de resultados manejo dos tablas, en las cuales diferencio entre el curso (ya sean 1° y 2° de la ESO), y por género (chicos y chicas), en la primera tuvieron mejor resultados los de la 2° de la ESO, lo que significa que estos estudiantes desarrollan mejor el test de pensamiento computacional y en cuanto a la segunda los chicos mostraron mejor desempeño.

Apreciación: Este proyecto fue importante ya que daba una idea de como realizar un test de pensamiento computacional, y de un posible modelo de pregunta que permitiera de una forma sencilla pero certera, medir el nivel del pensamiento computacional en los estudiantes que asistieran a este taller.

3.2. TRABAJOS DE AMBITO NACIONAL

Título: “Evolución”: Diseño e implementación de material educativo digital para fortalecer habilidades del pensamiento computacional

Autor(es): Mauricio Javier Rico Lugo, Xabier Basogain Olabe, Nancy Moreno Niño

Año: 2018.

Objetivo general: Diseñar e implementar un material educativo digital para fortalecer el pensamiento computacional.

Descripción: Abordando la problemática del bajo desempeño con que los estudiantes del Instituto Colombiano de Aprendizaje en matemáticas y comprensión lectora. Para combatir esta problemática, la enseñanza del pensamiento computacional es una alternativa para reforzar estos conceptos. Este trabajo consistió en la recolección de información referente al pensamiento computacional, además de los materiales educativos digitales dirigidos al estudio. Se diseñó y se aplicó un MED (Material educativo digital)

el cual consistía en un storyboard (Guion gráfico) el cual en su primera versión tenía siete retos que orientaban al estudiante a fortalecer diferentes tipos de pensamientos relacionados con el pensamiento computacional.

Resultados: Se realizó una prueba de entrada, para medir el desempeño de los estudiantes que realizarían el MED, esta prueba constó de 10 preguntas abiertas relacionadas a la matemáticas, lógica y algoritmia. Una vez resuelta, el estudiante pasaba a realizar el MED, el cual les tomó en tiempo 67 minutos completarlo y por último desarrollo otra prueba con 10 preguntas relacionadas con la prueba de entrada y con los retos del MED, tanto en esta prueba como en la de entrada contaban con 40 minutos para su solución. Adicional a esto, se llevaron diarios de campo y registro de audio y video. Una vez interactuado, se les entregó un cuestionario relacionado con el beneficio, agrado y mejoras al MED, este constó con una duración de 20 minutos. Esta investigación arrojó que se debe reforzar en los estudiantes habilidades en matemáticas y comprensión lectora, además que fue una actividad que se realizó en un espacio de 3 horas, lo que imposibilitó la participación de más estudiantes.

Apreciación: A pesar que esta investigación fue de orden cualitativo, permitió observar la relevancia de aplicar estrategias que refuercen las habilidades propias del pensamiento computacional en los estudiantes, ya que al momento de realizar las guías, se busca que no solo se aprenda programación en otro lenguaje, sino que el estudiante mediante esta refuerce o despierte las habilidades propias del pensamiento computacional tales como pensamiento matemático, comprensión lectora, análisis algorítmico etc.

Título: Entorno de aprendizaje para la enseñanza de programación en Arduino mediado por una mano robótica didáctica

Autor(es): Jorge Armando Niño Vega. Lina Yormary Martínez Díaz, Flavio Humberto Fernández Morales, Julio Enrique Duarte, Faustino Reyes Caballero, Guerly José Gutiérrez Barrios.

Año: 2017.

Objetivo general: Enseñar robótica y programación mediante la construcción de una mano robótica en arduino.

Descripción: Este proyecto consistía en la elaboración de una mano robótica en arduino con el fin de motivar al estudiante para el aprendizaje de la robótica y la programación en el colegio La Nueva Familia de Duitama, a los estudiantes del grado 11° los cuales tenían un promedio de edad de 17 años. Se realizaron un total de 6 guías de aprendizaje para su validación, las cuales contenían los temas relacionados para su construcción. La dificultad de las guías era de forma creciente, es decir a medida que se avanzaba en la temática, mayor era la dificultad de las guías. Otra implementación que tuvo este proyecto, fue la de la plataforma Edmodo, un entorno de interacción docente-estudiante para facilitar la comunicación y solución de dudas.

Resultados: Según el autor, sus resultados fueron satisfactorios ya que la aceptación por parte del estudiantado fue buena (dado que dos estudiantes comentaron que no estaban satisfechos). En cuanto a la realización de un prototipo para la enseñanza de la robótica y la programación, puede ser complementaria con otras materias y herramientas que hagan uso de las tecnologías de la información.

Apreciación: Esta investigación es importante, ya que da una perspectiva del grado de dificultad que debían llevar las guías de pensamiento computacional en cuanto a la programación con arduino, ya que la idea era realizar las guías de tal forma que fueran complemento de la temática manejada en la materia de programación estructurada.

Título: ABP (aprendizaje basado en problemas) para la enseñanza y el desarrollo de proyectos tecnológicos interdisciplinarios basados en arduino.

Autor(es): Yair Rivera Julio, Luis Turizo

Año: 2014.

Objetivo general: Implementar el aprendizaje basado en problemas en la enseñanza y desarrollo de procesos tecnológicos interdisciplinarios.

Descripción: Este proyecto aborda el proceso de enseñanza-aprendizaje para la generación de procesos tecnológicos aplicando la metodología del aprendizaje basado en problemas y arduino, apoyado con plataformas como Fritzing para el diseño de prototipos electrónicos.

Resultados: Esta investigación, de forma parcial demostró que la implementación de prototipos electrónicos apoyados en el aprendizaje basado en problemas, fomenta el trabajo en equipo y la generación cognitiva como el aprendizaje dentro del aula.

Apreciación: Con esta investigación, se tuvo el primer acercamiento al aprendizaje basado en problemas y el impacto que se podría general en la enseñanza con prototipos electrónicos.

3.3. TRABAJOS DE AMBITO LOCAL

Título: Desarrollo Del Pensamiento Computacional Con LEGO Mindstorm Dirigido A Estudiantes De Grados 10 Y 11.

Autor(es): Alberto Rodríguez Dávila.

Año: 2019.

Objetivo general: Desarrollar el pensamiento computacional en estudiantes de grado 10 y 11 con LEGO Mindstorm, para incentivar el interés de esta población en carreras con fines tecnológicos.

Descripción: Este proyecto, el cual en principio estaba planteado para estudiantes de 10 y 11 de un establecimiento educativo en Pamplona, se replanteo dirigirlo a los estudiantes de primer semestre de la Universidad de Pamplona. Este proyecto consto de guías orientadas a desarrollar el pensamiento computacional apoyado en lego mindstorm. Este curso se hizo previo al inicio del periodo académico 2019-01 el cual conto con 6 estudiantes, organizados en dos grupos ya que el programa solo cuenta con dos set de LEGO mindstorm, adicional se realizaron dos test para medir la capacidad antes y después de iniciar el taller.

Resultados: Se realizaron un total de 6 guías, distribuidas en 9 sesiones las cuales fueron impartidas entre el 09 de abril hasta el 11 de mayo. Algunas de las guías fueron repartidas en varias sesiones por la extensión y en el tiempo programado (2 horas) no se podían completar. En estas guías el estudiante aprendía programación con LEGO mindstorm a la par que desarrollaban el pensamiento computacional, ya que las guías contenían actividades que ayudaban al estudiante a su desarrollo. Los test de pensamiento computacional, los cuales servían de diagnóstico, arrojaron que el desarrollo de las guías demostraba un desarrollo favorable en el desempeño del pensamiento computacional y en cuanto a su registro de calificaciones a las cuales se le hizo seguimiento, los seis estudiantes tuvieron un desempeño sobresaliente en sus calificaciones.

Apreciación: Con este proyecto, el cual se realizó anterior a este sirvió para limitar el contenido de las guías, ya que se buscó que las guías realizadas en este proyecto se ejecutaran en las dos horas planteadas, para garantizar que el estudiante captara todo lo que ha desarrollado e las guías y no tuviera algún tipo de hueco.

Título: Desarrollo del pensamiento computacional mediante Scratch utilizando una herramienta e-Learning

Autor(es): Alba Beatriz Jáuregui Jaimes

Año: 2018.

Objetivo General: Determinar el impacto obtenido con el desarrollo de un curso de introducción a la programación en Scratch apoyado en Moodle, en el desarrollo del pensamiento computacional en estudiantes de educación media.

Descripción: Este proyecto se basa en la enseñanza del pensamiento computacional, orientado a estudiantes de 11° grado, utilizando como lenguaje de programación scratch, y la plataforma e-learning Moodle. Este se les aplico

a 25 estudiantes, y las clases eran de forma virtual repartidas en 5 sesiones con supervisión del investigador, estas se realizaron en un periodo de 2 horas durante 6 semanas (1 sesión por semana). Además, se implementaron dos test de diagnóstico.

Resultados: Se diseñó cada uno de los materiales necesarios para la ejecución del curso, además de pedir los permisos necesarios para la utilización del aula Moodle de la universidad de Pamplona. En esta aula contaba con un foro, en el cual los estudiantes podían compartir sus dudas y experiencia a la hora de interactuar con las guías. En los test, se observó una mejoría en cuanto notoria, ya que los estudiantes participantes manifestaron no tener ningún tipo de conocimiento acerca del pensamiento computacional y scratch.

Apreciación: Con esta investigación, con la cual se basó para la realización de algunas de las guías realizadas en este proyecto, además de manejar una modalidad de enseñanza completamente diferente a la trabajada en este proyecto (virtual) la cual podría afectar los posibles resultados, ya que al aplicar otra forma de enseñanza el estudiante podría estar más motivado o no a recibir los conocimientos impartidos.

Título: Desarrollo del pensamiento computacional mediante Scratch en estudiantes de educación media del municipio de Pamplona

Autor(es): Nelson Andrés Acevedo Mora.

Año: 2018

Objetivo general: Determinar el impacto obtenido con el desarrollo de un curso de Scratch en el desarrollo del pensamiento computacional, en estudiantes del grado 11°, mediante la aplicación de pruebas técnicas y observaciones sobre campo.

Descripción: Este proyecto consistía en la enseñanza del pensamiento computacional en estudiantes de 11° grado utilizando scratch, el cual buscaba ser un estímulo para que los estudiantes ingresaran a carreras relacionadas con las tecnologías de la información, concretamente al programa de ingeniería de sistemas de la Universidad de Pamplona. Este curso contó con una muestra de 25 personas pertenecientes al grado 11° del colegio Brighton, a los cuales se les aplicó una encuesta para identificar el interés y conocimiento acerca de las áreas de las tecnologías de información, dos test de diagnósticos para medir las habilidades de los estudiantes antes y después de la realización de las guías, las cuales fueron un total de 6 impartidas en siete sesiones, en esta última los estudiantes debían presentar un proyecto final de propia inspiración.

Resultados: En el desarrollo de las pruebas técnicas (test), se evidenció que los estudiantes se evidenció un incremento en el desempeño de las

habilidades concerniente al pensamiento computacional, y el autor resalta que los estudiantes alcanzaron el nivel esperado, evidenciado en el desarrollo del proyecto final.

Apreciación: Esta investigación fue importante ya que con esta se pudo documentar acerca del pensamiento computacional, además del modo correcto de la realización de un curso orientado al desarrollo del pensamiento computacional con cualquier tipo de lenguaje teniendo en cuenta el promedio de edad de los estudiantes (grado 11°) son los estudiantes de primer semestre de una institución de educación superior.

3.4. APRECIACION GENERAL

De los trabajos investigados anteriormente, se puede concluir que la enseñanza del pensamiento computacional, aplicando diferentes metodologías de aprendizaje, genera buenos índices en la mejora de habilidades relacionadas con la ciencia y la tecnologías, en la cual muchos expertos consultados trataron como su situación problema.

Estos artículos, uno solo enfocados a la enseñanza de la programación en si demuestra que al programar también se desarrollan habilidades propias del pensamiento computacional a través de este medio, ya que, al enfrentarse a un determinado problema, se genera en su conciencia un sin fin de opciones lógicas para encontrar soluciones acordes.

También se pudo apreciar en estos trabajos, que la mejor forma de enseñanza del pensamiento computacional es mediante una metodología de aprendizaje que sea bidireccional, es decir que haya un continuo aprendizaje-enseñanza no solo por parte del tutor o maestro, sino que el estudiante también aporte, apoyando la inventiva y el interés por parte de los involucrados.

4. METODOLOGÍA

Para la RAE (real academia española), la metodología es el conjunto de métodos que siguen una investigación científica o una exposición doctrinal, y se entiende por método la forma de ordenar y estructurar un trabajo. Con base en esta definición, se realizó este capítulo que busca mostrar los aspectos que se utilizaron para la realización del proyecto, la población objetivo y la manera e instrumentos con los que se recopilaron los datos proporcionados de la realización de este proyecto.

Esta investigación está enmarcada en el enfoque cuantitativo, ya que los datos que se recolectaron, tabularon y analizaron provienen de guías, test orientados a desarrollar el pensamiento computacional, adicional a los registros de calificaciones del periodo académico, con los resultados obtenidos de estos se podrá dictaminar el efecto de la propuesta de investigación.

4.1. PLAN DE DESARROLLO.

Esta investigación busca encontrar estrategias que frenen los altos índices de deserción y mortalidad académica del programa de ingeniería de sistemas, específicamente en la materia de programación estructurada, correspondiente al primer semestre académico del programa. Para el desarrollo del proyecto se desarrolló una serie de guías orientadas al desarrollo del pensamiento computacional apoyado en arduino, las cuales serían impartidas mediante un curso el cual su participación fue abierta y libre.

Para el inicio del curso del pensamiento computacional se debía abordar aspectos sobre la concertación de horarios, el cual debía ser cómodo para los participantes de tal manera que no interfiriera con sus actividades académicas. Así que antes de comenzar y en común acuerdo con los participantes, se concertó un horario.

Al inicio y al final del curso, se realizaron dos test de pensamiento computacional, con el propósito de medir sus capacidades antes y después del curso. Esto sirvió, para hacer un diagnóstico de sus capacidades y evolución, así como de observar el nivel de aprendizaje que el estudiante desarrolla al estar expuesto al pensamiento computacional.

Finalmente se realizó el curso, en el cual se realizaron en 2 sesiones semanales de 2 horas, en esas sesiones se ejecutaron los distintos talleres realizados por parte del investigador, aplicando los aprendizajes basados en problemas y proyectos, ya que con estos se potencia el desarrollo del pensamiento computacional, al ser tipos de aprendizajes ideales para la enseñanza de la ciencia y la ingeniería, dado que dan un enfoque más práctico

y motivador para estudiante. El PIS cuenta con 5 kit de desarrollo arduino que cuenta con la placa y diferentes componentes.

El análisis de resultados integrará las notas de los estudiantes correspondientes al segundo y tercer corte, tomando como grupo de investigación los estudiantes que participaron en el curso contra los compañeros de curso de programación estructurada, siendo esto este el grupo de control. Con esto se podrá que tan efectiva es la propuesta investigativa.

4.2. MUESTRA Y POBLACIÓN OBJETIVO.

La población fueron los estudiantes de ingeniería de sistemas de la Universidad de Pamplona, los cuales estuvieran matriculados en la materia de programación estructurada en el periodo académico 2019-01. Sin embargo, no se podía tomar todos, ya que los kit de arduino con los que cuenta el programa son 5, con cual no sería efectivo y retrasaría el desarrollo del proyecto.

Para ello se lanzó una convocatoria, en la cual fueron colocadas en los salones donde los estudiantes reciben clase. En la cual se dio plazo de dos semanas para que los interesados enviaran un correo manifestando su interés y con base en los que se manifestaron su interés por asistir al curso se conformó el grupo de investigación.

Después del tiempo trazado, se recibieron ocho correos de los estudiantes que cursan programación estructurada, los cuales serían los participantes del curso, y se convertirían en la muestra de la investigación y por ende, el grupo experimental y como grupo de control sería el resto de los estudiantes que cursan la materia de programación estructurada.

4.3. HERRAMIENTAS Y RECOLECCIÓN DE DATOS.

4.3.1. Herramientas para recolección de datos.

- Test de pensamiento computacional: Se diseñaron dos test con los que Con esta se busca medir las aptitudes con las que el estudiante ingresa y finaliza el curso de pensamiento computacional.
- Guías de pensamiento computacional orientadas a arduino: Se diseñaron 10 guías que buscan que el estudiante desarrolle el pensamiento computacional, aplicando conceptos básicos de programación y temas afines.

- Diario de campo: Agenda llevada por el investigador, para registrar los comportamientos de los participantes del curso y su evolución en el desarrollo del pensamiento computacional con el abordaje y aplicación de los diferentes talleres.
- Evidencia y muestras fotográficas: Se llevó un registro de asistencia el cual debía ser llenado por el estudiantado, además evidencias fotográficas que la realización de las actividades propuestas en las guías de desarrollo.

4.3.2. Recolección de datos.

- Participación: El proyecto presentó una investigación pasiva, ya que el investigador solo se ocupó de ser un observador y la mayoría del trabajo fue desarrollada por el estudiantado.
- Sesiones: En cada sesión se ejecutó una guía del pensamiento computacional, la cual tuvo la participación de los estudiantes interesados de la asignatura de programación estructurada.
- Lista de asistencia: En cada sesión, el estudiante debió firmar una lista en la cual se afirmaba que el estudiante asistió a cada sesión de los talleres de pensamiento computacional con arduino.
- Pruebas y resultados: A cada una de las actividades propuestas, se le realizó su análisis correspondiente para verificar la eficacia del aprendizaje y el desarrollo del pensamiento computacional en el estudiante.

4.4. ANÁLISIS DE DATOS.

Los datos obtenidos en la realización de este proyecto fueron de las siguientes fuentes:

- Guías de pensamiento computacional, las cuales tenían enfoque a fomentar el pensamiento computacional en los estudiantes, estas se corrigieron y se tabularon para medir el progreso general de la muestra.
- Test de pensamiento computacional los cuales fueron aplicados y tabulados para su posterior análisis.

- Se indago en la evolución académica del curso programación estructurada tomando los estudiantes que asistieron al curso como grupo de prueba y aquellos que no asistieron al curso como grupo de control para determinar si hay una diferencia significativa en el rendimiento académico.
- Adicional se analizó los resultados de la continuidad académica del grupo en el siguiente curso correspondiente a la asignatura de programación orientada a objetos donde se hará un análisis entre ambos grupos.

5. DISEÑO ESTRATÉGICO

En este capítulo se aborda el diseño del contenido de cada uno de los materiales de trabajo empleados en el curso de pensamiento computacional utilizando arduino.

5.1. GENERALIDADES.

Analizando la investigación consignada en capítulos anteriores y buscando estrategias para disminuir la deserción y mortalidad académica del programa de ingeniería de sistemas de la universidad de Pamplona, se decidió implementar el desarrollo del pensamiento computacional en dichos estudiantes

Para ello lo más recomendable, era realizar un curso en los cuales los estudiantes pudieran desarrollar el pensamiento computacional, que, a su vez fuera complementarios con sus estudios, principalmente en la materia de programación estructurada. Analizando múltiples opciones, la mejor forma era realizarlo de forma magistral, aplicando la metodología de aprendizajes como el basado en problemas, e basado en proyectos y la metodología de aprender haciendo, con la cual el estudiante podría acomodarse a su propio ritmo y asimilar los conceptos planteados.

El curso se dividió en tres momentos, el primero un diagnóstico inicial del estudiante que fue medido mediante la aplicación de un test inicial, el segundo que consistió en el desarrollo del curso, en el cual se aplicaron una serie de guías orientadas a desarrollar el pensamiento computacional apoyado por la plataforma arduino, y el ultimo el cual consistía en un diagnóstico final del estudiante, el cual fue medido mediante un test.

La elaboración y desarrollo de los diferentes momentos se a bordo de la siguiente manera.

5.2. ELABORACIÓN DE LOS TEST DE PENSAMIENTO COMPUTACIONAL.

Los test que permitieron medir el nivel de los estudiantes a la hora de comenzar y finalizar el curso de pensamiento computacional con arduino, constaban de 10 preguntas de selección múltiple, tipo pruebas de estado para facilitar la comprensión de cada uno de los incisos por parte del estudiante. Los test abordaron algunos de los aspectos del pensamiento computacional

tales como comprensión lectora, razonamiento matemático, razonamiento lógico y reconocimiento de patrones.

Para la elaboración del test, también se consultaron algunas fuentes, principalmente el test de pensamiento computacional presentado por Román, Pérez y Jiménez (2016), el cual constaba de 28 preguntas relacionadas con cuatro temáticas de computación: Direcciones, Bucles (repetir veces, repetir hasta), condicionales (simple, compuesto, hasta que) y funciones simples..

Sin embargo, se quería hacer un test que permitiera observar algunos aspectos del pensamiento computacional con que el estudiante ingresaba al claustro universitario, es por eso se decidió no incluir directamente temas relacionados con programación, sino con asignaturas tales como física, matemáticas y español.

Las preguntas fueron agrupadas en 4 categorías: Comprensión lectora Reconocimiento de patrones, análisis matemático y análisis lógico-matemático, de las cuales habría un mínimo de dos preguntas y máximo de tres por categoría. Para el diseño de las diferentes preguntas se tomaron preguntas de simulacros de las pruebas de estados, textos escolares de grados 10 y 11 y algunos test encontrados en la internet, estas preguntas fueron adaptadas y otras con base en estas, inventadas por el investigador.

A continuación, se expondrán el contenido de cada uno de los test, al igual que su respectiva respuesta.

5.2.1. Estructura del test de diagnóstico inicial.

Las preguntas 01, 02, y 06, están enmarcadas en el análisis lógico matemático, el cual le ayuda al estudiante debe demostrar sus habilidades para encontrar soluciones acordes a problema planteado, mediante la utilización de métodos matemáticos capaces de resolverlos.

Las preguntas 03 y 10 están dentro del ámbito de la comprensión de lectura, actividad que es muy reacia en los estudiantes de educación secundaria y media, ya que con esta se observa el nivel de interpretación del estudiante al momento de enfrentarse a un problema y la búsqueda de una solución lógica adecuada.

Las preguntas 04, 05 y 09 están dirigidas al reconocimiento de patrones, la cual permite encontrar diferentes patrones que permiten encontrar el orden lógico de cada una de las situaciones problema que se plantean en el entorno al que están expuestos.

Las preguntas 07 y 08, las cuales corresponden al pensamiento matemático, esencial para la solución de problemas que requieren un análisis cuantitativo de situaciones que son medibles en el ambiente laboral y personal del

individuo, además de pieza clave en las áreas de las tecnologías e información.

Las preguntas correspondiente se encuentran en el anexo A1.

5.2.2. Estructura del test de diagnóstico final.

Las preguntas 01 y 02, correspondían a la comprensión de lectura, la cual el estudiante se enfrentó en la mayor parte del curso, y que llevaría a que mejorara su interpretación, búsqueda de palabras claves para una acorde solución a la problemáticas que se enfrente en su entorno.

Las preguntas 03 y 10, en las cuales se evalúa el reconocimiento de patrones, la cual el estudiante mejoraría en la identificación de detalles claves y clasificarlos por relevancia para una solución a los desafíos que se enfrente en su entorno.

Las preguntas 04, 08 y 09, correspondiente al análisis de pensamiento lógico-matemático, en la cual, mediante la aplicación de la programación, desarrolla técnicas algorítmicas capaces de identificar detalles coherentes y una serie de pasos ordenados que le permitan realizar las soluciones adecuadas a las problemáticas planteadas.

Las preguntas 05, 06 y 07, con un enfoque matemático, en el cual, a través del curso reforzo sus habilidades en la identificación y realización de problemas que requieren un enfoque numérico y una solución cuantificable de los mismo.

Las preguntas de este test se encuentran en el anexo A4.

5.3. ELABORACIÓN DE LAS GUÍAS DE PENSAMIENTO COMPUTACIONAL.

Para el desarrollo de las guías se hizo una planeación, donde se identificaron las distintas habilidades del pensamiento computacional y su acoplamiento en la temática de la programación con arduino. Generando un total de 10 guías, de las cuales 8 estaban estructuradas con temáticas relacionadas a la programación y las dos restantes, más practicas ya que contenían una serie de ejercicios que permitirían observar las habilidades adquiridas por el estudiante.

Los temas abordados en cada guía avanzaban incrementalmente, así el conocimiento impartido seguiría en constante evolución permitiendo una experiencia enriquecedora para el estudiante.

Las guía se componen de tres ejes: el primero era una conceptualización teórica, que tiene como objetivo orientar al estudiante sobre un tema en

cuestión, apoyado por actividades relacionadas, combinado con aspectos que le ayudarían a desarrollar el pensamiento computacional, el segundo eje consistía en actividades lúdicas, en esta parte se buscaba que el estudiante mediante arduino, aborde un problema en específico y encuentre una solución apoyado en esta plataforma, y el último eje presenta una tercera una pequeña evaluación autocrítica, para que el estudiante se hiciera un examen de conciencia, si lo visto le había servido para su crecimiento académico.

A continuación, se describe el objetivo de cada una de las guías, su elaboración y temática, las guías en su contenido se encuentran en el anexo B:

GUÍA	EJE TEMATICO	HABILIDADES A DESARROLLAR EN PENSAMIENTO COMPUTACIONAL
Guía 01	<ul style="list-style-type: none"> • Introducción a la plataforma arduino • Conceptos básicos de programación, Conceptos de algoritmo, dato, constantes y variables 	<ul style="list-style-type: none"> ✓ Comprensión lectora ✓ Pensamiento algorítmico ✓ Pensamiento lógico ✓ Resolución de problemas.
Guía 02	<ul style="list-style-type: none"> • Estructuras de selección • Operadores lógicos. • Operadores relacionales. • Operadores matemáticos. 	<ul style="list-style-type: none"> ✓ Comprensión lectora ✓ Pensamiento lógico. ✓ Pensamiento matemático. ✓ Pensamiento algorítmico. ✓ Resolución de problemas.
Guía 03	<ul style="list-style-type: none"> • Estructura repetitiva para (<i>for</i>). • Estructura repetitiva mientras que (<i>while</i>) • Estructura repetitiva hacer mientras que (<i>do While</i>) 	<ul style="list-style-type: none"> ✓ Comprensión lectora. ✓ Reconocimiento de patrones. ✓ Pensamiento lógico. ✓ Pensamiento algorítmico. ✓ Razonamiento ✓ Resolución de problemas
Guía 04	<ul style="list-style-type: none"> • Funciones • Vectores 	<ul style="list-style-type: none"> ✓ Comprensión lectora. ✓ Pensamiento algorítmico. ✓ Pensamiento lógico. ✓ Pensamiento matemático. ✓ Resolución de problemas.
Guía 05	<ul style="list-style-type: none"> • Librerías 	<ul style="list-style-type: none"> ✓ Comprensión lectora ✓ Reconocimiento de patrones. ✓ Pensamiento lógico. ✓ Solución de problemas
Guía 06	<ul style="list-style-type: none"> • Bluetooth 	<ul style="list-style-type: none"> ✓ Comprensión lectora. ✓ Resolución de problemas.
Guía 07	<ul style="list-style-type: none"> • Clase • Atributos • Objetos • Propiedades de la programación orientada a objetos 	<ul style="list-style-type: none"> ✓ Comprensión lectora ✓ Pensamiento lógico ✓ Pensamiento algorítmico ✓ Resolución de problemas
Guía 08	<ul style="list-style-type: none"> • Herencia 	<ul style="list-style-type: none"> ✓ Comprensión lectora. ✓ Pensamiento algorítmico.

		<ul style="list-style-type: none"> ✓ Pensamiento lógico. ✓ Resolución de problemas.
Guía 09	<ul style="list-style-type: none"> • Ejercicios prácticos 	<ul style="list-style-type: none"> ✓ Comprensión lectora. ✓ Pensamiento algorítmico. ✓ Pensamiento lógico. ✓ Resolución de problemas.
Guía 10	<ul style="list-style-type: none"> • Ejercicios prácticos 	<ul style="list-style-type: none"> ✓ Comprensión lectora. ✓ Pensamiento algorítmico. ✓ Pensamiento lógico. ✓ Resolución de problemas.

Tabla 4: Análisis del contenido de las guías de pensamiento computacional

5.3.1. Guía 01: Introducción al mundo arduino.

El objetivo de esta guía, es comenzar a introducir al estudiante al pensamiento computacional, mediante la programación de arduino. Esta guía tiene como eje central algunos conceptos de la programación tales como algoritmos, datos y constantes y variables, cada una con su respectiva actividad. Además, posee un pequeño resumen de lo que es arduino, esto con el propósito de que el estudiante conociera la plataforma en la cual se va a trabajar.

- **Contenido de la guía.**

Primer eje: Conceptualización teórica.

- *Introducción a Arduino:* Se entregó la conceptualización de la plataforma arduino, la cual contenía su estructura, tipología y fragmentos de su historia y creación.
- *Introducción a Algoritmia:* Se entregó la conceptualización de al algoritmia con conceptos básicos de algoritmo y su forma de elaboración.
- *Concepto de dato:* Se entregó la conceptualización de datos y su tipología, además de su clasificación en constantes y variables con su respectivo ejemplo ilustrativo.

Cada elemento de este eje buscaba desarrollar la habilidad lectora del estudiante y al final se aplicaron actividades de medición de apropiación de conceptos con ejercicios simples, ejemplos e identificación de información.

Segundo eje: Actividad lúdica.

En este eje se plantearon una serie de desafíos los cuales serían resueltos mediante la aplicación de la programación en arduino. En este eje se explica los materiales, el montaje y la forma de programación,

- *Problema planteado 01:* El problema del led pulsador, el cual es ideal para la introducción a la enseñanza de arduino principalmente para aquellos que no han tenido contacto con la plataforma. (ver figura 9)

Esta actividad contenía retos que obligaban al estudiante a aplicar y proponer soluciones acordes basados en lo aprendido en el problema explicado.

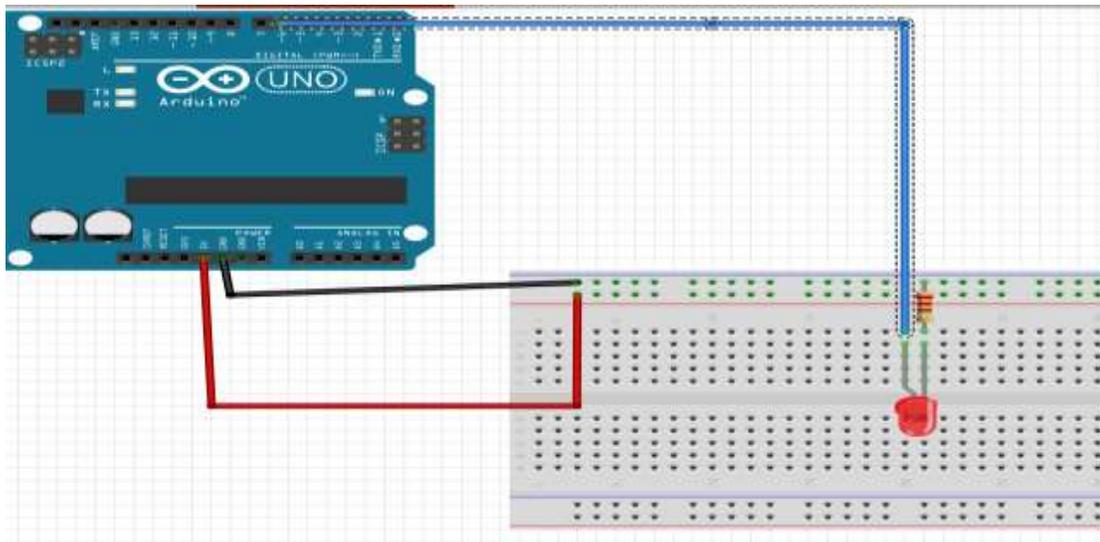


Figura 9: Montaje del led pulsador.

Tercer eje: Evaluación crítica.

En este eje, el estudiante a conciencia debía responder unas preguntas relacionadas con el aprendizaje de los temas planteados. Para analizar el impacto del tema en su vida académica.

5.3.2. Guía 02: Estructuras de selección.

El objetivo de esta guía, era la de que el estudiante comprendiera que las situaciones problema en múltiples ocasiones requieren de tomar decisiones dependiendo de su entorno, además de seguir en el orden lógico de la enseñanza de la programación. Para esta guía se complementó con la enseñanza de los diferentes tipos de operadores que se utilizan en programación (lógicos, relacionales y matemáticos).

- **Contenido de la guía.**

Primer eje: Conceptualización teórica.

- Operadores lógicos: Se le entregó la contextualización de los operadores lógicos de negación, conjunción y disyunción, explicados mediante ejemplos.
- Operadores relacionales: Se le entregó la contextualización de los operadores relacionales mayor y mayor que, menor y menor que, igual y diferente explicados mediante una tabla.

- Operadores matemáticos: Se le entregó la contextualización de los operadores matemáticos más utilizados en programación, explicados en una tabla.
- Conceptualización estructura de selección if-else: Se le entregó la contextualización de la estructura de selección if-else (si-sino), ilustrando su sintaxis.

Cada elemento de este eje buscaba desarrollar la habilidad lectoras y matemáticas del estudiante y al final se aplicaron actividades de medición de apropiación de conceptos con ejercicios simples de pensamiento matemático.

Segundo eje: Actividad lúdica.

En este eje se plantearon una serie de desafíos los cuales serían resueltos mediante la aplicación de la programación en arduino. En este eje se explica los materiales, el montaje y la forma de programación

- *Problema planteado 01*: El problema del led con pulsador, el cual es apropiado para la explicación de la estructura de selección if-else, ya que es ideal para la conceptualización de la sintaxis de la estructura.
- *Problema planteado 02*: Montaje del sensor ultrasonido, el cual demuestra la aplicabilidad de la estructura de selección y la lógica implementada para la solución del problema.

Los montajes se pueden apreciar en la figura 10.

Esta actividad contenía retos que obligaban al estudiante a aplicar y proponer soluciones aplicando la estructura de selección y los conceptos aplicados en esta guía.

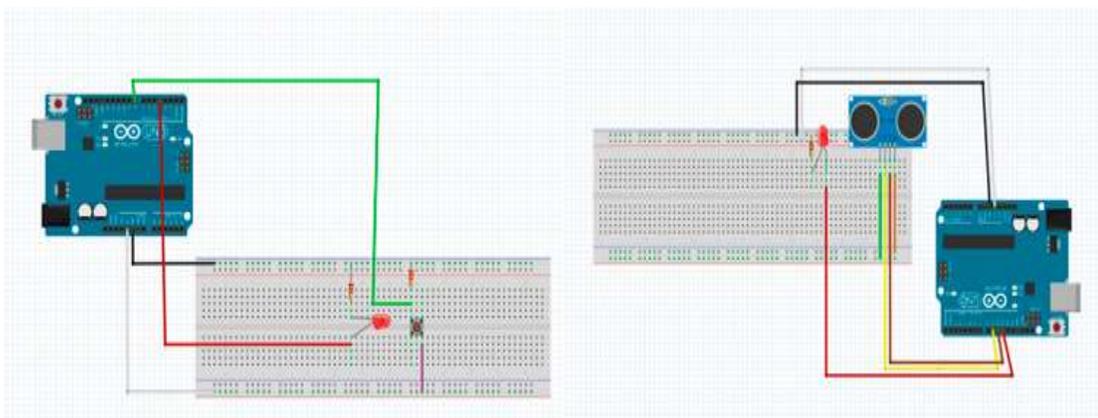


Figura 10: Practicas de la guía 2, a) led con pulsador b) sensor ultrasonido

Tercer eje: Evaluación crítica.

En este eje, el estudiante a conciencia debía responder unas preguntas relacionadas con el aprendizaje de los temas planteados. Para analizar el impacto del tema en su vida académica.

5.3.3. Guía 03: Estructuras repetitivas.

En esta guía el objetivo fue que el estudiante conociera el concepto de las estructuras repetitivas utilizadas en la programación y su aplicación en la vida cotidiana. Esta guía se centra en este tema ya que a la mayoría de los estudiantes no solo de ingeniería de sistemas, sino de las demás ingenierías que ven programación, se les dificulta entenderlo.

- **Contenido de la guía.**

Primer eje: Conceptualización teórica.

- Introducción a las estructuras repetitivas: Se le entregó la contextualización de las estructuras repetitivas mediante una lectura.
- Concepto de la estructura repetitiva for: Se le entregó la contextualización de la estructura repetitiva for (para), explicado su sintaxis mediante un ejemplo.
- Concepto de la estructura de selección mientras while: Se le entregó la contextualización de la estructura repetitiva while (mientras que), explicado su sintaxis mediante ejemplos.
- Concepto de la estructura de selección mientras do while: Se le entregó la contextualización de la estructura repetitiva do while (hacer mientras que), explicado su sintaxis mediante ejemplos.

Cada elemento de este eje buscaba desarrollar la habilidad lectoras y algorítmicas del estudiante y al final se aplicaron actividades de medición de apropiación de conceptos con ejercicios simples de algoritmia.

Segundo eje: Actividad lúdica.

En este eje se plantearon una serie de desafíos los cuales serían resueltos mediante la aplicación de la programación en arduino. En este eje se explica los materiales, el montaje y la forma de programación

- *Problema planteado 01:* El problema del auto fantástico, el cual es ideal para explicar el uso de la estructura de selección for, porque es un claro ejemplo de su funcionamiento en la plataforma arduino
- *Problema planteado 02:* Montaje del sensor ultrasonido con zumbador, el cual demuestra la aplicabilidad de la estructura repetitiva while e impulsa la investigación en cuanto a la tonada del zumbador.

Los montajes se pueden apreciar en la figura 11.

Esta actividad contenía retos que obligaban al estudiante a aplicar y proponer soluciones aplicando la estructuras repetitivas y los conceptos aplicados en esta guía.

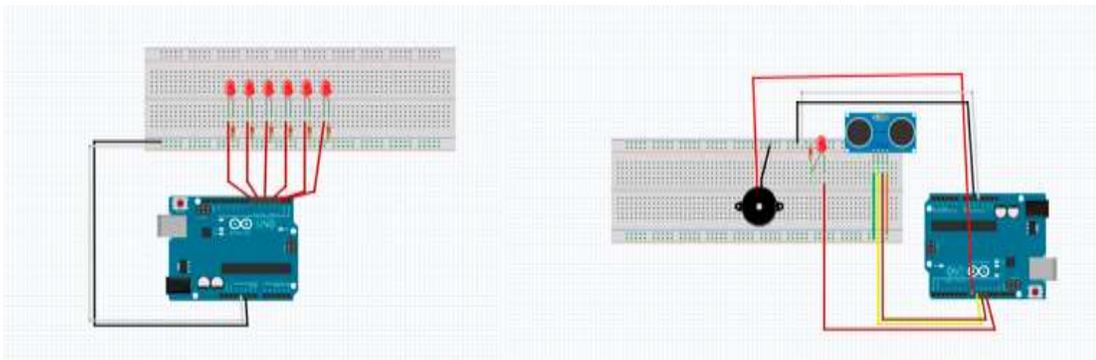


Figura 11: Montajes de la guía 03

Tercer eje: Evaluación crítica.

En este eje, el estudiante a conciencia debía responder unas preguntas relacionadas con el aprendizaje de los temas planteados. Para analizar el impacto del tema en su vida académica.

5.3.4. Guía 04: Funciones y vectores.

Con esta guía se buscó que el estudiante entendiera una forma más optima de programar, además que conociera el concepto de función, el cual es aplicado en otras áreas de su carrera universitaria. También se manejó el concepto de vectores (arreglos unidimensionales) como complemento, se expuso el concepto de vectores, como forma de que el estudiante analizara formas diferentes de almacenar información.

- **Contenido de la guía.**

Primer eje: Conceptualización teórica.

- Concepto de funciones: Se le entregó la contextualización de las funciones, su tipología y su sintaxis.
- Concepto de vectores: Se le entregó la contextualización de vectores, su sintaxis y aplicabilidad en programación.

Cada elemento de este eje buscaba desarrollar la habilidad lectoras, algorítmicas y lógica del estudiante y al final se aplicaron actividades de medición de apropiación de conceptos con ejercicios de algoritmia realizando montajes explicados anteriormente.

Segundo eje: Actividad lúdica.

En este eje se plantearon una serie de desafíos los cuales serían resueltos mediante la aplicación de la programación en arduino. En este eje se explica los materiales, el montaje y la forma de programación

- *Problema planteado 01:* Montaje utilizando sensor de movimiento, En el cual el estudiante, con el conocimiento previo realiza y asimila el uso de funciones (ver figura 12).

Esta actividad contenía retos que obligaban al estudiante a aplicar y proponer soluciones aplicando los conceptos explicados en esta guía.



Figura 12: Fotografía del montaje explicado en la practica

Tercer eje: Evaluación crítica.

En este eje, el estudiante a conciencia debía responder unas preguntas relacionadas con el aprendizaje de los temas planteados. Para analizar el impacto del tema en su vida académica, además de incentivar la lectura e investigación mediante la consulta de temática complementaria al tema.

5.3.5. Guía 05: Librerías.

El objetivo de esta guía, continuando con el concepto de funciones era analizar algunas funciones que tienen los lenguajes de programación, para este caso arduino ya están prediseñadas y su forma de aplicación. Esta guía se centra principalmente en la investigación por parte del estudiante, ya que varias actividades se centran en la búsqueda de la información.

- **Contenido de la guía.**

Primer eje: Conceptualización teórica.

- Concepto de Librerías: Se le entregó la contextualización de las librerías, su forma de instalación y aplicabilidad.

Cada elemento de este eje buscaba desarrollar la habilidad lectoras y algorítmicas del estudiante y al final se aplicaron actividades de medición de apropiación de conceptos con ejercicios de comprensión lectora e investigación.

Segundo eje: Actividad lúdica.

En este eje se plantearon una serie de desafíos los cuales serían resueltos mediante la aplicación de la programación en arduino. En este eje se explica los materiales, el montaje y la forma de programación

- *Problema planteado 01:* Montaje de un sensor de temperatura, En el cual el estudiante aplica los conceptos trabajados en la guía, probando su comprensión lectora y retención de la información (ver figura 13).

Esta actividad contenía retos que obligaban al estudiante a aplicar y proponer soluciones aplicando los conceptos aplicados en esta guía.

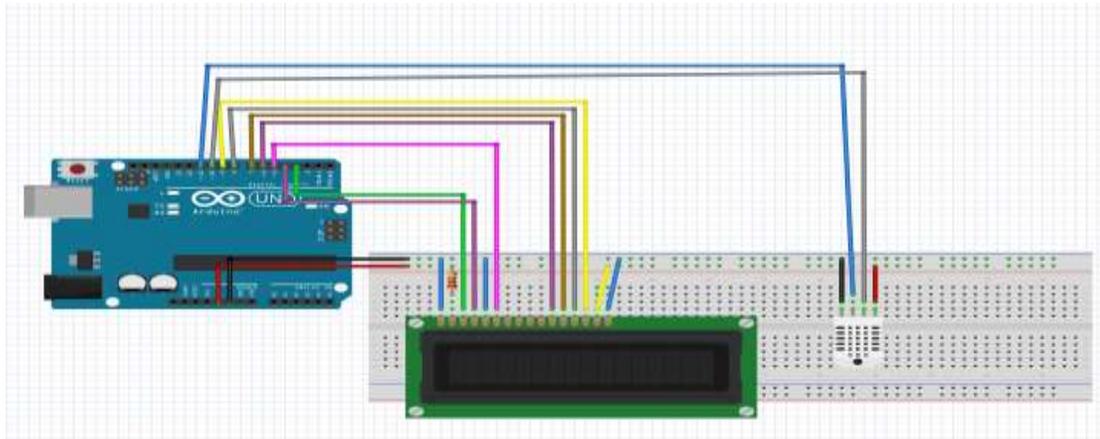


Figura 13: Sensor de temperatura y humedad con arduino

Tercer eje: Evaluación crítica.

En este eje, el estudiante a conciencia debía responder unas preguntas relacionadas con el aprendizaje de los temas planteados. Para analizar el impacto del tema en su vida académica.

-

5.3.6. Guía 06: Aplicación de las librerías: Conectividad por bluetooth.

El objetivo es empezar a probar las capacidades que el estudiante ha desarrollado hasta el momento (ya que se ha pasado al ecuador del curso) y se consideró que la mejor forma era con un tema netamente propio de arduino, además de motivar al estudiante con aplicaciones que en futuras ocasiones podría realizar.

- **Contenido de la guía.**

Primer eje: Conceptualización teórica.

Primer eje: Conceptualización teórica.

- Introducción a la tecnología Bluetooth: Se le entregó la contextualización de la tecnología bluetooth, clase y versiones que se manejan hasta la actual.

Cada elemento de este eje buscaba desarrollar la habilidad lectoras y del estudiante y así fomentarla para su vida diaria..

Segundo eje: Actividad lúdica.

En este eje se plantearon una serie de desafíos los cuales serían resueltos mediante la aplicación de la programación en arduino. En este eje se explica los materiales, el montaje y la forma de programación

- *Problema planteado 01:* Montaje de leds controlados remotamente, En el cual el estudiante observaría la conectividad de esta tecnología, además de utilizar algoritmia para la solución del problema.
- *Problema planteado 02:* Montaje con led 7 segmentos, en el cual el estudiante aplicará y reforzara los conceptos trabajados hasta el momento.

Los montajes se pueden apreciar en la figura 14.

Esta actividad contenía retos que obligaban al estudiante a aplicar y proponer soluciones aplicando los conceptos aplicados en esta guía.

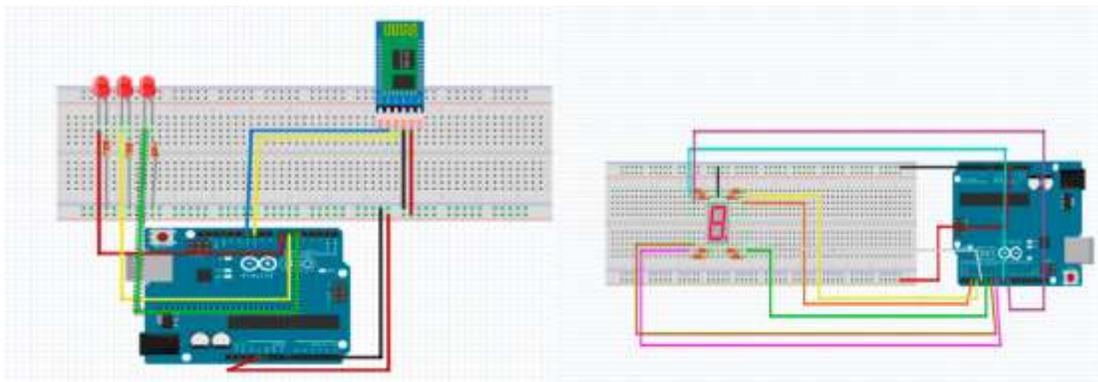


Figura 14: Practicas de la guía 06

Tercer eje: Evaluación crítica.

En este eje, el estudiante a conciencia debía responder unas preguntas relacionadas con el aprendizaje de los temas planteados. Para analizar el impacto del tema en su vida académica.

-

5.3.7. Guía 07: Introducción a la programación orientada a objetos.

El objetivo de esta guía era introducir al estudiante al paradigma orientado a objetos, sin perder la esencia de la enseñanza del pensamiento computacional. Esto a raíz que el estudiante le cuesta la adaptabilidad a este nuevo modelo de programación y después de la programación estructurada,

en la programación orientada a objetos se presenta otro porcentaje alto de deserción del programa. Contando con la ventaja que, arduino es un lenguaje multiparadigma esto se pudo realizar introduciendo al estudiante con este tema.

- **Contenido de la guía.**

Primer eje: Conceptualización teórica.

- Introducción a la programación orientada a objetos: Se le entregó la contextualización de la programación orientada a objetos, mediante la ilustración con un ejemplo.
- Concepto de clases: Se le entregó la contextualización del concepto de clase, explicando sus componentes y su elaboración.
- Concepto de propiedades de programación orientada a objetos: Se le entregó la contextualización de los principios de encapsulamiento, herencia y polimorfismo

Cada elemento de este eje buscaba desarrollar la habilidad lectoras, algorítmicas y lógica del estudiante y al final se aplicaron actividades de medición de apropiación de conceptos con ejercicios simples de algoritmia.

Segundo eje: Actividad lúdica.

En este eje se plantearon una serie de desafíos los cuales serían resueltos mediante la aplicación de la programación en arduino. En este eje se explica los materiales, el montaje y la forma de programación

- *Problema planteado 01:* Led pulsador, en el cual el estudiante aprendería y compararía el cambio de paradigma, además de explicar sintaxis y conceptos propios de la programación orientada a objetos como niveles de visibilidad y constructores..
- *Problema planteado 02:* Montaje del sensor ultrasonido, en el cual el estudiante aplicará y reforzará los conceptos trabajados en esta guía..

Esta actividad contenía retos que obligaban al estudiante a aplicar y proponer soluciones aplicando los conceptos aplicados los conceptos de la programación orientada a objetos.

Tercer eje: Evaluación crítica.

En este eje, el estudiante a conciencia debía responder unas preguntas relacionadas con el aprendizaje de los temas planteados. Para analizar el impacto del tema en su vida académica.

-

5.3.8. Guía 08: Programación orienta a Objetos: Herencia.

Continuando con la temática anterior, como objetivo de la guía era que el estudiante conociera uno de los principios más importante y utilizado en la programación orientada a objetos. Dado que es uno de los temas que más conflicto genera en los estudiantes que ven esta materia.

- **Contenido de la guía.**

Primer eje: Conceptualización teórica.

- Concepto de herencia: Se le entregó la contextualización de herencia mediante un ejemplo.

Cada elemento de este eje buscaba desarrollar la habilidad lectoras, lógicas y algorítmicas del estudiante aplicando los conceptos trabajados en la guía.

Segundo eje: Actividad lúdica.

En este eje se plantearon una serie de desafíos los cuales serían resueltos mediante la aplicación de la programación en arduino. En este eje se explica los materiales, el montaje y la forma de programación

- *Problema planteado 01:* Maqueta de un semáforo con sensores, En el cual el estudiante aplica los conceptos trabajados en la guía, probando su comprensión lectora, adaptabilidad y algoritmia (ver figura 15).

Esta actividad contenía retos que obligaban al estudiante a aplicar y proponer soluciones aplicando los conceptos aplicados en esta guía.

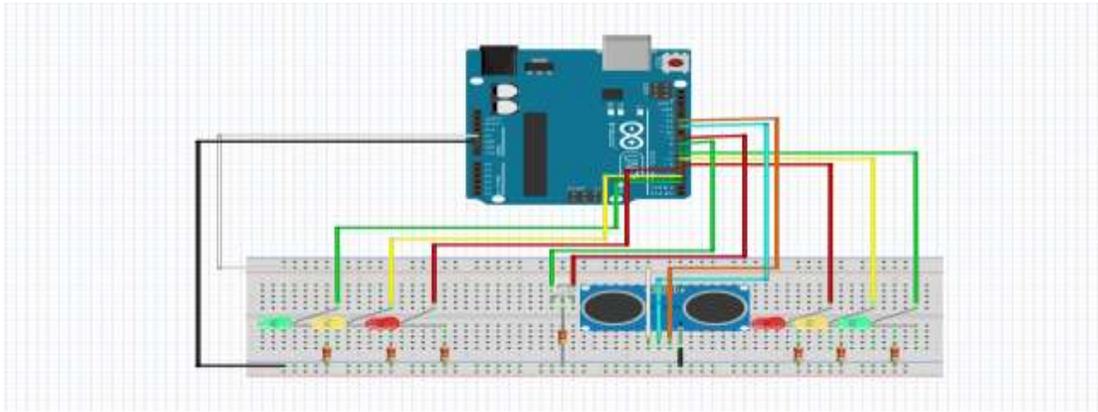


Figura 15: Montaje de la guía 08

Tercer eje: Evaluación crítica.

En este eje, el estudiante a conciencia debía responder unas preguntas relacionadas con el aprendizaje de los temas planteados. Para analizar el impacto del tema en su vida académica.

-

5.3.9. Guías 09 y 10: Retroalimentación de arduino.

En estas dos guías, el principal objetivo era ver el conocimiento que el estudiante adquirió con todo lo realizado en las anteriores talleres, tanto en la programación con arduino como en el desarrollo del pensamiento computacional.

- **Contenido de la guías.**

Primer eje: Conceptualización teórica.

En estas guías no se explica un tema en concreto, solo se fomenta la aplicabilidad de lo aprendido en las guías anteriores, principalmente en aspectos como análisis, algoritmia y lógica.

Segundo eje: Actividad lúdica.

- *Problemas de la guía 09:* Se plantearon dos problemas, el primero la realización de un dado mágico mediante leds, y el segundo correspondiente a la creación de un tester de baterías.

- *Problemas de la guía 10:* Se plantearon dos problemas, el primero la realización de una maquina de morse, utilizando un led y dos pulsadores y el segundo controlando la intensidad de los led mediante un dispositivo móvil.

En la figura 16 se observa los montajes de las dos guías.

En estas guías se fomentó la inventiva e investigación del estudiante para solucionar diferentes problemas que se pueden enfrentar en su vida cotidiana.

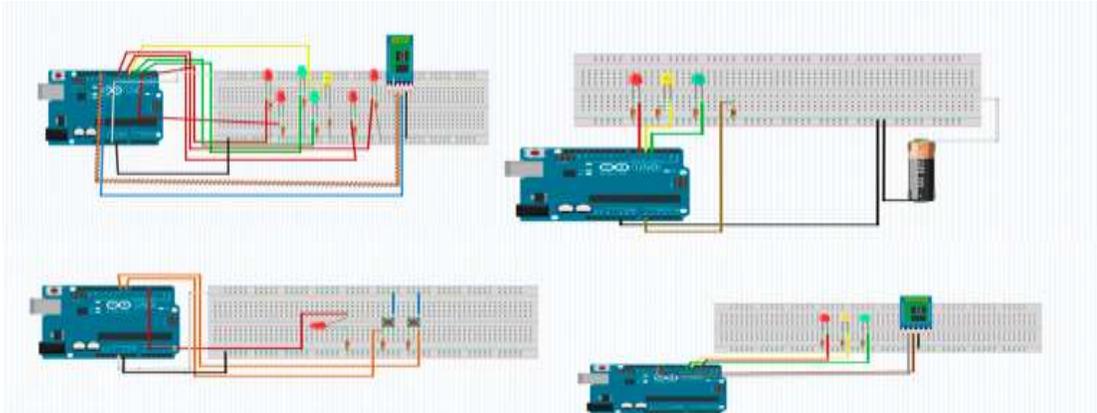


Figura 16: Montajes de las guías 09 y 10.

Tercer eje: Evaluación crítica.

Estas guías no tenían este eje.

6. ANÁLISIS DE RESULTADOS

Este capítulo aborda el análisis de los datos que se recopilaron al desarrollar el curso de pensamiento computacional con arduino, el cual fue efectuado entre el 8 de junio hasta el 9 de agosto. Se abordan tres análisis fundamentales, el primero corresponde a lo desarrollado en las sesiones del curso, en el cual se aplicaron las guías; el segundo se recopilaron y tabularon los resultados correspondientes a los test de diagnóstico que se ejecutaron al principio y al final del curso y el ultimo, corresponde al comparativo del rendimiento académico de los estudiantes que participaron en el curso con aquellos que no, esto mediante las notas de las materias de programación estructurada y programación orientada a objetos.

6.1. REALIZACIÓN DEL CURSO DE PENSAMIENTO COMPUTACIONAL.

6.1.1. Introducción

El curso de pensamiento computacional apoyado en la plataforma arduino, el cual contó con encuentros semanales de 4 horas, distribuidos en dos sesiones de 2 horas. Para la participación del estudiantado, donde participaron un total de 8 estudiante que respondieron a la convocatoria los cuales fueron:

	PRIMER APELLIDO	SEGUNDO APELLIDO	NOMBRE	GRUPO PROGRAMACION ESTRUCTURADA
1	Carrillo	Parada	Volkmar Adyya	B
2	Isidro	López	Sergio Andrés	B
3	Martínez		Daniel	A
4	Mendoza	Parra	Mónica Yasveyli	B
5	Peñaloza	Gelves	Sergio Alejandro	B
6	Pérez	Mojica	Oxiber Wilfran	B
7	Suarez	Hernández	Carolina	B
8	Vera	Carrillo	Fabian Hernando	B

Tabla 5: Inscritos al curso de pensamiento computacional con arduino

Una vez se conformó el grupo, se acordó un horario, el cual fueron los días martes de 7:00 AM a 8:59 AM y los sábados de 10:00 AM a 11:59 AM.. En la siguiente tabla se resume la fecha y lugar de cada una de las sesiones:

FECHA	LUGAR	Hora entrada	Hora Salida
09-06-2019	Salón JG 103	10: 00 AM	12:00 PM
11-06-2019	Salón JG 103	07:00 AM	09:00 AM
15-06-2019	Salón JG 103	10:00 AM	12:00 PM
22-06-2019	Salón JG 103	07:00 AM	09:00 AM
19-07-2019	Salón JG 103	02:00 AM	04:50 PM
02-08-2019	Salón JG 103	02:00 PM	09:00 PM
08-08-2019	Salón FJ 103	02:00 PM	06:00 PM
09-08-2019	Salón FJ 103	08:00 AM	12:00 PM

Tabla 6: Horarios desarrollados en los talleres de pensamiento computacional

A continuación, se expondrá cada una de las incidencias ocurridas en las diferentes sesiones que componen el curso de pensamiento computacional. para comprenderlas claramente, se han dividido en tres momentos, presaberes, saberes y validación.

6.1.2. Sesión 01.

- **Presaberes**

Como esta sesión marcaba el primer encuentro, se debía medir las habilidades de los participantes, por consiguiente, se aplicó el primero de los dos test pactados. El test contenía un total de 10 preguntas de selección múltiple, el cual estuvo diseñado para una duración de 30 minutos, los estudiantes lo desarrollaron en 35 minutos, arrojando una primera impresión en cuanto a su comprensión lectora, que se encuentra dentro del rango aceptable para la presentación de la prueba.

Una vez finalizado el test se organizó el grupo en equipos de trabajo de 2 y 3 integrantes, a los cuales se les asignó el kit de arduino, donde se solicitó al estudiante las diferentes partes que componen el kit para generar interés en este curso.

Para continuar con la sesión se entregó en forma electrónica la guía 01-introducción a arduino, la cual correspondía a los temas de introducción a arduino, algoritmo, dato, constantes y variables. Es importante destacar que se van a desarrollar habilidades como comprensión lectora, pensamiento algorítmico, pensamiento lógico y resolución de problemas.

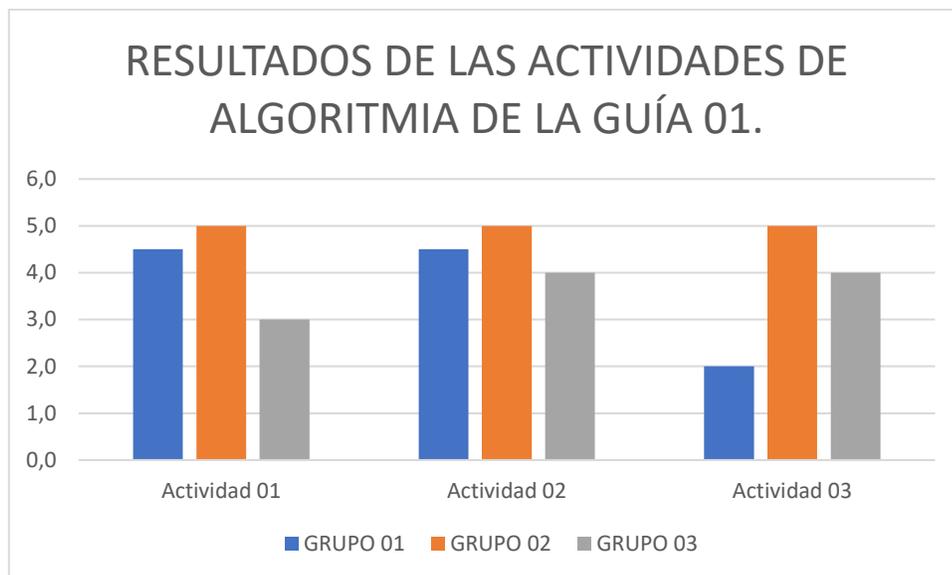
La primera parte de la guía pretende desarrollar la habilidad de comprensión lectora ya que contiene la información básica de arduino, el comportamiento de los estudiantes en esta etapa fue fluido ya que no presentaron preguntas y avanzaron sobre el tema. El siguiente tema que aborda la guía corresponde a algoritmia, donde se maneja los conceptos fundamentales y se plantean ejemplos para la validación de la apropiación del conocimiento sobre este tema

se solicita el desarrollo de una actividad dividida en tres secciones. Como se observa en la tabla 7.

	OBSERVACIONES		
	ACTIVIDAD 01	ACTIVIDAD 02	ACTIVIDAD 03
OBJETIVO DE LA ACTIVIDAD	Enuncie algoritmos (rutinas) que utilizamos en nuestra vida cotidiana	5 que en vida	Escoja una de las que enuncio anteriormente y usando un lenguaje natural es decir con sus propias palabras realice su respectivo algoritmo.
RESULTADO GRUPO 01	A pesar que no colocaron el nombre de los algoritmos, realizaron 5 ejemplos detallados de algoritmos sin ningún error.		No lograron identificar el orden correcto del algoritmo, el cual tenía ese error.
RESULTADO GRUPO 02	Enunciaron correctamente algoritmos	5	Los estudiantes realizaron un ejemplo de forma impecable, sin ningún error
RESULTADO GRUPO 03	Enunciaron ejemplos correctamente.	3	A pesar de realizar bien el algoritmo, les faltó ordenarlo de forma entendible.

Tabla 7: Observaciones de la actividad de algoritmos.

Las actividades se corrigieron con valor numérico de 1.0 a 5.0. en esta primera actividad el grupo 02 sobresalió ya que fue el único que logro la calificación perfecta en las tres actividades relacionadas al tema. En la gráfica 1 se observa el comportamiento de cada actividad por grupo.



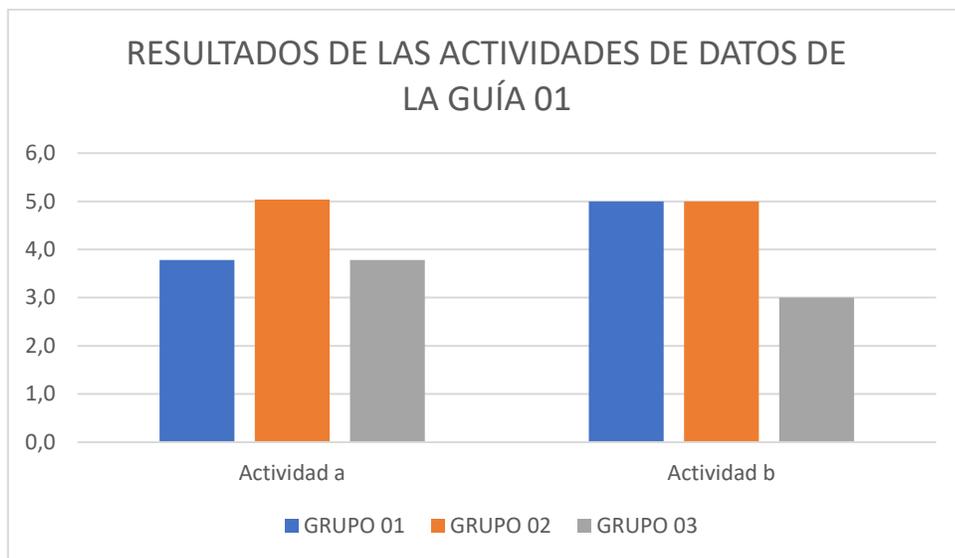
Gráfica 1: Resultados de las actividades de algoritmia de la guía 01.

El siguiente tema que se manejó el concepto de dato, variables y constantes y variables. La primera parte consistía en una lectura conceptual del tema, enmarcado en una serie de actividades, las cuales se resumen en la tabla 8.

OBSERVACIONES		
	ACTIVIDAD 01	ACTIVIDAD 02
OBJETIVO DE LA ACTIVIDAD	Determine el tipo de dato utilizado como respuesta en los siguientes casos.	En el texto, identifique los datos y clasifíquelos.
RESULTADO GRUPO 01	De 12 ejemplos, identificaron 9.	Identificaron correctamente los datos presentes en el enunciado.
RESULTADO GRUPO 02	De 12 ejemplos, identificaron 12.	Identificaron correctamente los datos presentes en el enunciado.
RESULTADO GRUPO 03	De 12 ejemplos, identificaron 9.	Identificaron correctamente algunos de los datos presentes en el enunciado.

Tabla 8: Observaciones de la actividad de datos.

En la gráfica 2, se resumen los resultados de la actividad relacionada con la temática de dato:



Gráfica 2: Resultados de las actividades de datos de la guía 01

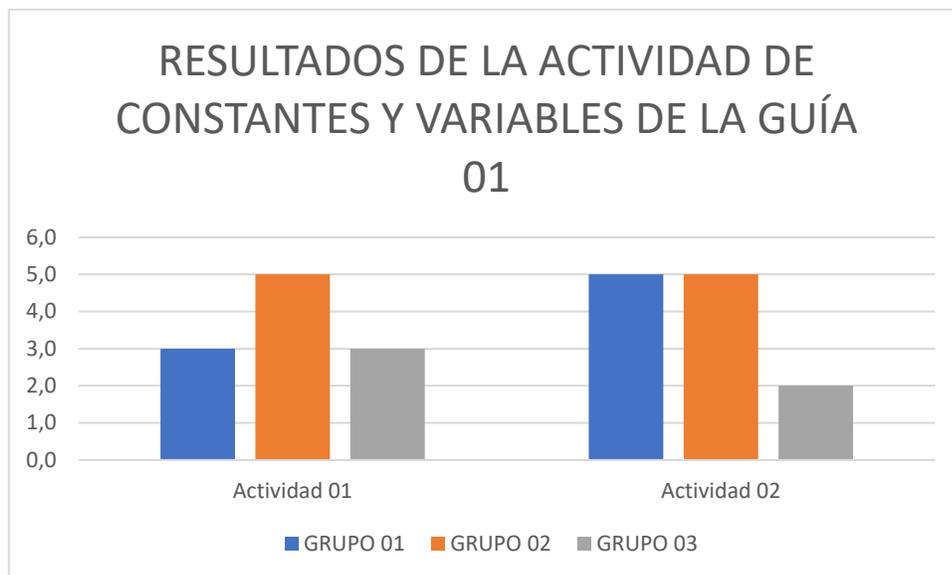
En esta actividad, el grupo 02 volvió a tener un desempeño excelente, seguido por el grupo 01 el cual se destacó en la actividad 02. y el grupo 03 aun demuestra dificultades.

Continuando con la temática de dato, se le expuso el concepto de constantes y variables. Para saber la retención de este temario, se enmarco una actividad, la cual se resume en la tabla 9.

OBJETIVO DE LA ACTIVIDAD	OBSERVACIONES	
	ACTIVIDAD 01	ACTIVIDAD 02
	De acuerdo a la siguiente situaciones, identifique constantes y variables (2 situaciones).	
RESULTADO GRUPO 01	Identifica de forma aceptable algunas situaciones del enunciado planteado.	Identifica excelentemente las situaciones del enunciado planteado.
RESULTADO GRUPO 02	Identifica excelentemente las situaciones del enunciado planteado.	Identifica excelentemente las situaciones del enunciado planteado.
RESULTADO GRUPO 03	Identifica de forma aceptable algunas situaciones del enunciado planteado.	Le cuesta trabajo identificar las situaciones del enunciado planteado

Tabla 9: Aciertos de los estudiantes en el actividad 2 inciso a:

El gráfico 3 muestra los resultados numéricos de la actividad relacionada con constantes y variables:



Gráfica 3: Resultados de la actividad de constantes y variables de la guía 01

El grupo 02 seguía demostrando un gran dominio en la temática, identificando correctamente en los dos ejemplos, seguido por el grupo 01 y en cuanto al grupo 03 el desempeño no fue el esperado, ya que solo acertó regularmente el primer ejercicio y en cuanto al segundo parece que presentó dificultades.

En esta parte como observación general, el grupo presentó un desempeño sobresaliente, dado que esta parte de la guía se estima un tiempo de realización de 45 a 60 minutos, el cual fue concluido entre 55 y 60 minutos, llenando las expectativas del investigador.

- **Saberes**

Continuando con el desarrollo de la guía, la siguiente parte correspondía a la actividad lúdica en la cual el estudiante comenzaría a desarrollar la programación en arduino, cabe resaltar que esta guía al ser la primera se le explico aspectos técnicos en cuanto a la sintaxis, palabras claves del lenguaje de arduino. El problema planteado no debía ser tan complejo, por eso se escogió el “hola mundo” de arduino, es decir el montaje del led parpadeante. En la guía se planteó en forma de problema, para comenzar a desarrollar la habilidad de resolución de problemas del pensamiento computacional. Se les explico cada uno de los materiales utilizados brevemente. Se le explica paso a paso como hacer tanto el montaje como el código y al final, se le plantea una serie de retos el cual el estudiante debía realizarlos, estos retos serian el verdadero desafío para probar las habilidades del estudiante.

En la tabla 10 se resume las observaciones de la actividad lúdica explicada en la guía:

OBSERVACIONES DE LA ACTIVIDAD: LED PARPADEANTE.		
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
OBSERVACIONES GRUPO 01	El grupo dominó ampliamente, siendo el primero en terminar el montaje explicado	EL código estaba acorde a las indicaciones de la guía
OBSERVACIONES GRUPO 02	Al principio presentaron dificultad en cuanto a la realización del montaje y con un poco de ayuda lograron que funcionara correctamente.	Realizaron el código según las indicaciones de la guía
OBSERVACIONES GRUPO 03	Fue el grupo que más se retrasó en el montaje, además de que sus compañeros, principalmente los del grupo 01 les asesoraron para poder realizar el montaje.	Realizaron el código según las indicaciones de la guía

Tabla 10: Observaciones de la actividad lúdica de la guía 01

Para garantizar que el estudiante aprendiera acerca del tema, se planteó una serie de desafíos relacionados con la práctica anterior. En la tabla 11 se resume el comportamiento presentado por cada uno de los grupos:

OBSERVACIONES DE LOS RETOS RELACIONADOS A LED PARPADANTE

RETOS	RETO1: Modificar para que el led se encienda cada 5 segundos		Reto 2: Modificar los pines del montaje de arduino donde está el led		Reto 3: Agregar otro led		Reto 4: Montaje de un semáforo.	
	Aspectos Montaje	Aspectos código	Aspectos Montaje	Aspectos código	Aspectos Montaje	Aspectos código	Aspectos Montaje	Aspectos código
Observaciones grupo 01	Entendió que no se debía modificar el montaje de led parpadeante	Analizó la función encargada de controlar el tiempo en arduino y modificarla para cumplir los requerimientos del reto.	Cambio adecuadamente el pin del montaje entendiendo su funcionalidad.	Modificó el código en el sitio correcto y probó su montaje, el cual cumplió con lo pactado en el reto.	Agregó correctamente el led y lo conectó adecuadamente a la placa de arduino.	Entendió cada una de las funciones orientadas al encendido y apagado de los led, modificando el código adecuadamente.	Aplicó todo lo aprendido durante la práctica, realizando el montaje de forma correcta.	Realizaron el código que cumple con las expectativas del reto, sin embargo, se les dificultó la declaración de variables.
Observaciones grupo 02	Entendió que no se debía modificar el montaje de led parpadeante	Analizó la función encargada de controlar el tiempo en arduino y modificarla para cumplir los requerimientos del reto.	Cambio adecuadamente el pin del montaje entendiendo su funcionalidad.	Modificó el código en el sitio correcto y probó su montaje, el cual cumplió con lo pactado en el reto.	Agregó correctamente el led y lo conectó adecuadamente a la placa de arduino.	Entendió cada una de las funciones orientadas al encendido y apagado de los led, modificando el código adecuadamente.	Hubo dificultades en la conexión del tercer led, que lograron solucionar adecuadamente.	Entendieron la lógica del enunciado, y a pesar de tener problemas con el montaje, realizaron un código que satisface las necesidades del enunciado.
Observaciones grupo 03	Entendió que no se debía modificar el montaje de led parpadeante	Analizó la función encargada de controlar el tiempo en arduino y modificarla para cumplir los requerimientos del reto.	Cambio adecuadamente el pin del montaje entendiendo su funcionalidad.	Modificó el código en el sitio correcto y probó su montaje, el cual cumplió con lo pactado en el reto.	Tuvo dificultades principalmente por el encendido del segundo led, ya que no se conectó adecuadamente.	Entendió cada una de las funciones orientadas al encendido y apagado de los led, modificando el código adecuadamente.	Hubo dificultades en la conexión de los led, que lograron solucionar adecuadamente.	Realizaron el código que cumple con las expectativas del reto.

	led parpadeante	modificarla para cumplir los requerimiento s del reto.	entendiendo su funcionalidad.	con lo pactado en el reto.	que tenía problemas en la continuidad de la energía	led, modificando el código adecuadamente .	adecuadamente .	
--	--------------------	--	----------------------------------	----------------------------------	--	---	--------------------	--

Tabla 11: Observaciones de los retos de led parpadeante de la guía 01

- **Validación**

Para finalizar el trabajo realizado durante la sesión, cada grupo debía llenar una autoevaluación, la cual consistía en 5 preguntas que buscaban una autocrítica al proceso realizado. Los resultados se muestran en la tabla 12.

PREGUNTA	Grupo 01	Grupo 02	Grupo 03
1. Con el desarrollo de esta guía reafirmó el concepto de programa.	SI	SI	SI
2. Con el desarrollo de esta guía reafirmó el concepto de algoritmo.	SI	SI	SI
3. Con el desarrollo de esta guía reafirmó el concepto de dato informáticos	SI	SI	NO
4. Con el desarrollo de esta guía reafirmó el concepto de constante y variable	SI	SI	SI
5. Había escuchado previamente la existencia de la plataforma Arduino	NO	SI	SI

Tabla 12: Respuesta de la autoevaluación de la primera guía

En esta sesión se observó un ambiente de expectativa, ya que para los integrantes del curso era una forma de programar novedosa, también la afinidad del grupo, ya que la mayoría de los integrantes pertenecían a un mismo grupo de programación estructurada.

En cuanto al nivel general, era un grupo trabajador, se notaba el interés por cada uno de los temas planteados, y a pesar de que el nivel de aprendizaje no fue el mismo para cada individuo, se notó un esfuerzo por hacer el mejor trabajo posible.

De los ocho estudiantes inscritos, solo participaron siete, ya que un estudiante llegó tarde y por pena y no interrumpir el proceso que llevaban sus compañeros no ingresó al salón.

6.1.3. Sesión 02.

- **Presaberes**

Comenzando la sesión, se tuvo que reacomodar los grupos, medida que se planteaba de forma temporal, ya que el estudiante que llegó tarde en la sesión anterior se integró al grupo, con el compromiso de adelantarse en la temática planteada en la guía anterior, y un estudiante perteneciente al grupo 03 no asistió a la sesión, llenando ese hueco.

Rearmado los grupos, de forma electrónica se entregó la guía 02: estructuras de selección, la cual explora los diferentes tipos de operadores utilizados en la programación, además de la estructura de selección *if-e/se* (si, sino). en la cual desarrolla habilidades de comprensión de lectura, algoritmia, pensamiento matemático, pensamiento lógico y resolución de problemas.

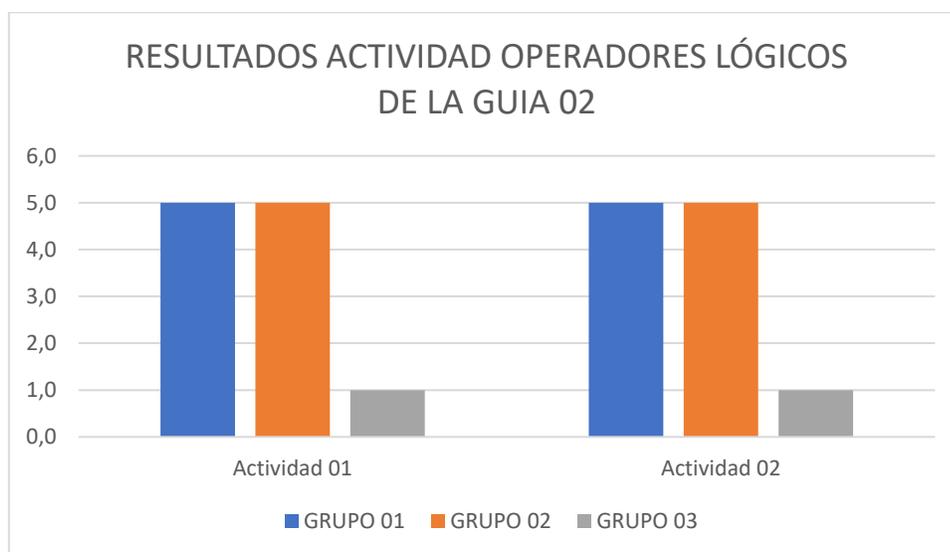
En cuanto a la lectura, se notó que el estudiante captó el tema, apoyado en el hecho que, en la materia de programación, se estaba empezando a trabajar dicha temática lo cual facilitó la comprensión.

El primer tema explicado consistía de los operadores lógicos más utilizados en la programación (negación, conjunción y disyunción). Para la temática, se realizó una actividad la cual contenía dos partes, el comportamiento de la actividad se resume en tabla 13.

OBJETIVO DE LA ACTIVIDAD	OBSERVACIONES	
	ACTIVIDAD 01	ACTIVIDAD 02
	De acuerdo con la lectura responder falso o verdadero.	De los ejemplos planteados, realizar las tablas de verdad
GRUPO 01	Acertaron correctamente las 5 preguntas de la lectura	Realizaron los ejemplos de forma correcta y ordenada.
GRUPO 02	Acertaron correctamente las 5 preguntas de la lectura	Realizaron los ejemplos de forma correcta y ordenada.
GRUPO 03	Se observó que trabajaron, sin embargo, no entregaron una guía para su respectiva corrección.	

Tabla 13: Observaciones de la actividad de operadores lógicos de la guía 02.

Los resultados obtenidos por cada uno de los grupos en la actividad se observan en la gráfica 4:



Gráfica 4: Resultados de la actividad de operadores lógicos de la guía 02.

Cabe aclarar, en la actividad a había 5 afirmaciones, es decir que el acierto tenía un valor de 1.0, en cuanto a la actividad 02, el cual eran 4 ejemplos, cada

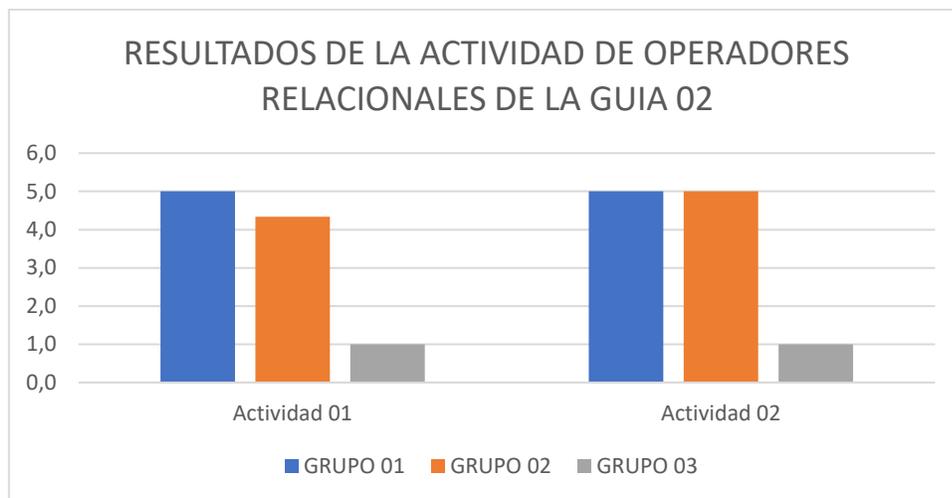
uno con un valor de 1.25 por cada punto. Como se puede apreciar, los grupos 01 y 02 dominaron el tema, en cuanto al grupo 03 argumentó que debía hacer unas correcciones y se comprometió a enviar la guía más tarde, la cual nunca fue recibida (ya que en sesiones posteriores desertaron del curso). Es por eso que solo por el trabajo observado durante la sesión, se les puso una calificación de 1.0.

El siguiente tema de la guía fue el de operadores relacionales, al igual que el anterior se explicó de forma breve y concisa. Esta tenía una actividad la cual estaba dividida en dos: la primera se enunciaban expresiones matemáticas en el cual el estudiante debía determinar si estaba correcta y en caso que no, debía corregirla por una expresión válida y la segunda debía dar 4 ejemplos donde el estudiante debía tomar decisiones en su vida diaria. La tabla 14 muestra las observaciones de la actividad:

OBSERVACIONES		
	ACTIVIDAD 01	ACTIVIDAD 02
OBJETIVO DE LA ACTIVIDAD	Determine si el símbolo utilizado es correcto, en caso contrario, escriba la expresión correcta	De ejemplo de situaciones (máximo 4) en donde tenga que tomar decisiones.
GRUPO 01	De 6 ítem, identificaron 6. Corrigieron correctamente aquellos en los que la expresión era invalida.	Plantearon 4 ejemplos válidos.
GRUPO 02	De 6 ítem, identificaron 6. No corrigieron correctamente aquellos en los que la expresión era invalida, solo cambiaron el signo.	Plantearon 4 ejemplos válidos.
GRUPO 03	Se observó que trabajaron, sin embargo, no entregaron una guía para su respectiva corrección.	

Tabla 14: Observaciones de la actividad de operadores relacionales

Para la calificación de las actividades se tuvieron en cuenta el número de incisos que poseía, para determinar su valor. En la actividad 01 cada acierto tenía un valor de 0.83, en cuanto a la actividad 02, la cual por cada ejemplo válido se sumaba 1.25. en la gráfica 5 se observa las notas obtenidas por cada grupo.



Gráfica 5: Resultados de la actividad de operadores relacionales de la guía 02.

En esta actividad, grupo 01 fue el mejor grupo, ya que logro un buen desempeño en las dos actividades. En el siguiente cuadro se resume las observaciones de la actividad:

El siguiente tema fue el de operadores matemáticos, en la cual se explicaron los operadores más utilizados en la programación. en este caso no hay una actividad, ya que se consideró que los estudiantes dominaban el tema.

El Otro tema expuesto fue la sintaxis de la estructura de selección *if-else*, principalmente su sintaxis para empezar a introducir al estudiantado en la programación de esta estructura.

En este momento de la sesión, tuvo una duración de 35 minutos, el tiempo presupuestado dentro de la planificación, factores como que el tema estaba siendo trabajado en la materia, facilitó la comprensión. En cuanto al trabajo general del grupo, fue un trabajo ordenado y cada grupo estaba sumergido en la lectura y la solución de las actividades planteadas.

- **Saberes**

El siguiente momento está enmarcado por la actividad lúdica, la cual contenía dos prácticas. La primera consistía en el led parpadeante, en la cual se explicó una serie de pasos para resolver problemas al igual que su respectivo montaje y retos. El segundo montaje corresponde a crear un sensor de proximidad, ayudado con el sensor de ultrasonido, en esta también se explicó brevemente su funcionamiento y su respectivo montaje. En la explicación del primer montaje, en la tabla 15 se resume el comportamiento del trabajo por grupo:

OBSERVACIONES DE LA ACTIVIDAD: LED CON PULSADOR.		
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
OBSERVACIONES GRUPO 01	El grupo realizó el montaje según las indicaciones, a pesar de un pequeño retraso en cuanto a la ubicación del pulsador.	EL código estaba acorde a las indicaciones de la guía
OBSERVACIONES GRUPO 02	El grupo realizó el montaje según las indicaciones, a pesar de un pequeño retraso en cuanto a la ubicación del pulsador.	Realizaron el código según las indicaciones de la guía
OBSERVACIONES GRUPO 03	Presentó dificultades en la ubicación de los led y pulsador, ya que no siguió las instrucciones de la guía.	No pudo probar el código en su montaje, el cual fue corregido en varias ocasiones hasta lograr el resultado esperado.

Tabla 15: Observaciones del montaje de led con pulsador de la guía 02

Para comprobar el aprendizaje, este montaje debía ser modificado para cumplir ciertas necesidades, en la tabla 16 se resume el comportamiento observado por cada grupo en cada uno de los retos:

Tabla 16: Observaciones de la actividad lúdica de la guía 02.

Esta guía, a diferencia de la anterior tenía un segundo montaje, el cual correspondía al sensor ultrasonido. La guía tenía consignado tanto el montaje como fragmentos del código. en la tabla 17 se muestra los comportamientos observados.

OBSERVACIONES DE LA ACTIVIDAD: MONTAJE ULTRASONIDO.		
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
OBSERVACIONES GRUPO 01	Realizo el correcto montaje, siguiendo cada una de las indicaciones de la guía.	Siguió cada una de las indicaciones de la guía en cuanto a la realización del código.
OBSERVACIONES GRUPO 02	Realizo el correcto montaje, siguiendo cada una de las indicaciones de la guía.	Siguió cada una de las indicaciones de la guía en cuanto a la realización del código.
OBSERVACIONES GRUPO 03	No realizo el correcto montaje, presentando problemas en la continuidad del montaje.	Siguió cada una de las indicaciones de la guía en cuanto a la realización del código.

Tabla 17: Observaciones del montaje del ultrasonido de la guía 02

Los retos de esta montaje, el cual se pretendía medir el nivel de abstracción y reconocimiento de patrones adquiridos hasta el momento. En la tabla 17 se resume el comportamiento durante los retos de cada grupo.

OBSERVACIONES DE LOS RETOS PLANTEADOS DE LA GUÍA 02				
RETOS	RETO 1: agregue um led y modifique para que encenada um led a 10 cm y dos led a 5 cm.		RETO 2: Agregar un led de diferente color, el cual se debe encender entre 0 y 15 cm y otro entre 15 y 25 cm.	
	Aspectos Montaje	Aspecto Código	Aspecto Montaje	Aspecto Código
OBSERVACIONES GRUPO 01	Insertaron correctamente el led sin	Modificaron correctamente las	Insertaron correctamente el led sin	Aplicó correctamente la estructura de

	interferir con el montaje ya realizado.	modificaciones del código.	interferir con el montaje ya realizado.	selección explicada para la solución del reto.
OBSERVACIONES GRUPO 02	Insertaron correctamente el led sin interferir con el montaje ya realizado.	Modificaron correctamente las modificaciones del código.	Insertaron correctamente el led sin interferir con el montaje ya realizado.	Aplicó correctamente la estructura de selección explicada para la solución del reto.
OBSERVACIONES GRUPO 03	Insertaron correctamente el led sin interferir con el montaje ya realizado.	Les costo dificultades, las cuales con ayuda de sus compañeros fueron solucionadas.	Insertaron correctamente el led sin interferir con el montaje ya realizado.	Aplicó correctamente la estructura de selección explicada para la solución del reto.

Tabla 18: observaciones de los retos de ultrasonido de la guía 02.

- **Validación.**

Como la guía anterior, contenía una serie de preguntas de autoevaluación, lo cual obtuvo las siguientes respuestas por parte de los grupos que entregaron la correspondiente guía.

PREGUNTA	GRUPO 01	GRUPO 02	GRUPO 03
1. Con el desarrollo de esta guía reafirmó el concepto de estructuras condicionales.	SI	SI	
2. Con dos guías realizadas ha reforzado los conceptos de programación vistos hasta ahora.	SI	SI	
3. Con el desarrollo de esta guía ha afianzado los conceptos de la programación trabajados hasta hoy.	NO	SI	

Tabla 19: Autoevaluación guía 02.

Durante la sesión se observó dificultades en cuanto al horario, ya que varios estudiantes manifestaron que, por asistir al curso, dejaban de asistir a asesorías de cálculo diferencial, además se les indagó el rendimiento en dicha materia, la cual los estudiantes manifestaron que tenían dificultades.

Otra observación, recae en el hecho que los estudiantes del grupo 03 se veían distraídos, no prestaban atención, además que el estudiante que suplió al compañero que no asistió, tenía mayor interés hacia el hardware, y era muy apático hacia la programación.

6.1.4. Sesión 03

- **Presaberes**

Para esta sesión, asistieron 6 estudiantes, ya que previamente un estudiante perteneciente al grupo 02 manifestó que no continuaría debido a que requería el tiempo para invertirlo a otras asignaturas (principalmente calculo integral). por consiguiente, para esta práctica se reestructuraron los equipos, quedando solo dos grupos de tres personas. A continuación, se procedió a entregar electrónicamente la guía 03: estructuras repetitivas. La cual busca mejorar su comprensión lectora, reconocimiento de patrones, análisis de problemas, solución de problemas y algoritmia.

En la primera parte de la guía, contenía la fundamentación teórica de las

OBSERVACIONES DE LOS RETOS PLANTEADOS DE LA GUÍA 02				
RETOS	RETO 1: modificar para que, al pulsar el botón, el led se encienda por 5 segundos.		RETO 2: Agregar otro led y otro pulsador que se enciendan al pulsar el botón un determinado tiempo.	
	Aspectos Montaje	Aspecto Código	Aspecto Montaje	Aspecto Código
OBSERVACIONES GRUPO 01	Insertaron correctamente el led sin interferir con el montaje ya realizado.	Analizaron la función, encontrando la función encargada de tiempo de encendido del led.	Al principio presentaron problemas en la inserción del pulsador, la cual pudieron solucionar adecuadamente.	Realizaron la adaptación del código adecuada para el reto planteado.
OBSERVACIONES GRUPO 02	Insertaron correctamente el led sin interferir con el montaje ya realizado.	Analizaron la función, encontrando la función encargada de tiempo de encendido del led.	Presentaron dificultades en la inserción del segundo pulsador.	Realizaron la adaptación del código adecuada para el reto planteado.
OBSERVACIONES GRUPO 03	Insertaron correctamente el led sin interferir con el montaje ya realizado.	Analizaron la función, encontrando la función encargada de tiempo de encendido del led.	Presentaron dificultades en la inserción del segundo pulsador.	Tuvieron dificultades en la lógica del programa, demostrando dificultades en cuanto a comprensión del reto.

estructuras de selección *for* (para), *while* (mientras que) y *do while* (hacer mientras que). Se observó que los estudiantes comprendieron bien el tema, ya que no se presentaron preguntas relacionadas al tema. Para una mejor comprensión del tema, esta se ilustró con un ejemplo, pasando por cada una de las diferentes estructuras de selección planteadas. Durante la práctica, y con el análisis de cada una, se les pidió una opinión personal acerca de las

estructuras de selección explicadas. En la tabla 20 se muestran las opiniones de los diferentes grupos.

GRUPO	OPINION
GRUPO 01	“el for ya que nos permite realizar casi todo tipo de estructuras de repetición y en mi opinión personal es más eficiente.”
GRUPO 02	“Es más eficiente la estructura While porque es más precisa”

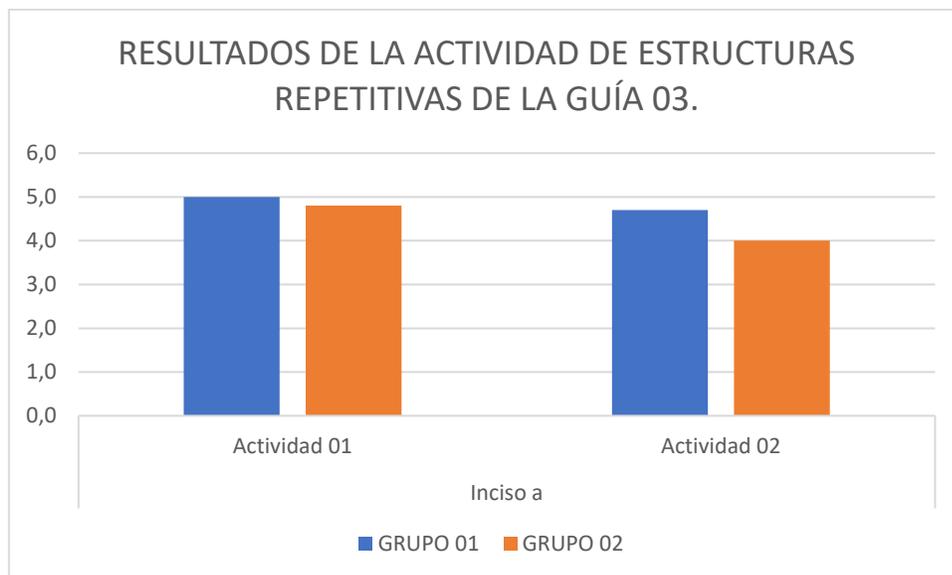
Tabla 20: Opiniones de las estructuras repetitiva de la guía 03.

Para observar el proceso de captación de información, se hizo una actividad, la cual consistía en realizar dos ejercicios, estos ejercicios estaban planteados para tener que utilizar una estructura repetitiva en específico. En la tabla 21 se resume los resultados que dejó esta práctica.

NOTA ACTIVIDAD EST. REPETITIVAS		
GRUPO	Ejercicio 01	Ejercicio 02
OBSERVACIONES GRUPO 01	Fue elaborado correctamente, utilizando las estructura repetitiva <i>for</i> adecuadamente.	En cuanto a lógica, fue correctamente elaborado, sin embargo, este ejercicio se planteó para utilizar la estructura <i>while</i> , la cual no fue aplicada.
OBSERVACIONES GRUPO 02	Fue elaborado correctamente, utilizando las estructura repetitiva <i>for</i> adecuadamente.	Tuvieron dificultades en cuanto a la estructura, y no aplicaron la estructura <i>while</i> .

Tabla 21: Observaciones de la actividad de la guía 03.

En resumen, los dos grupos solo aplicaron una estructura repetitiva para la realización de los dos ejercicios, evidenciando la preferencias de cuanto a estructuras de selección escogida por los estudiantes. En la gráfica 6 se muestra los resultados globales de los dos grupos.



Gráfica 6: Resultados de la actividad de estructuras repetitivas.

En esta actividad, planteada para 30 minutos, los dos grupos terminaron entre 25 y 28 minutos, lo cual demuestra una mejoría en cuanto a comprensión de lectura, además que, durante este momento, no se presentaron preguntas por parte del grupo.

- **Saberes**

Para las actividades lúdicas, en la cual se aplicaron dos montajes, el primero es conocido como el auto fantástico, el cual consiste en iluminar simultáneamente cierta cantidad de leds. En la tabla 22, están consignadas las observaciones durante la realización del trabajo durante la sesión.

OBSERVACIONES DEL MONTAJE DE AUTO FANTASTICO		
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
OBSERVACIONES GRUPO 01	Realizo el montaje correctamente.	Aplicaron excelentemente la estructura <i>for</i> .
OBSERVACIONES GRUPO 02	Aprendieron de los errores cometidos en los montajes anteriores y lograron realizar el montaje en menos tiempo del empleado en los anteriores	Aplicaron excelentemente la estructura <i>for</i> .

Tabla 22: Observaciones de la actividad lúdica de la guía 03.

Los retos planteados, se realizó el montaje del sensor ultrasonido de la guía anterior (guía 02), con la salvedad, que se le adicionó un nuevo periférico, un

zumbador. Las observaciones y resultados correspondiente a los retos se resumen en la tabla 23.

RETOS DE LA GUÍA 03: ALARMA DE MOVIMIENTO		
GRUPO	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
OBSERVACIONES GRUPO 01	Realizaron bien el montaje correctamente, integraron acordemente el zumbador.	Manejaron la estructura <i>while</i> , sin embargo, se les dificultó el código en cuanto al funcionamiento del zumbador.
OBSERVACIONES GRUPO 02	Les costo trabajo realizar el montaje en cuanto a la integración del zumbador.	Fue realizado de forma correcta en cuanto a su lógica, fallaron en la sintaxis el cual fue asesorado por sus compañeros.

Tabla 23: Observaciones de los retos de la guía 03.

- **Validación.**

En la autoevaluación planteada para esta guía, las respuesta de cada uno de los grupos se aprecian en la tabla 24.

PREGUNTA	GRUPO 01	GRUPO 02
1. Con el desarrollo de esta guía reafirmó el concepto de estructuras repetitivas.	SI	SI
2. Ha comprendido semejanzas y diferencias entre las diferentes estructuras repetitivas trabajadas en esta guía	SI	SI
3. Con las guías realizadas anteriormente, ha fortalecido los conceptos que posee de programación	SI	SI
4. Con el desarrollo de esta guía ha comprendido aún más la programación de la plataforma arduino	SI	SI

Tabla 24: Autoevaluación guía 03.

Durante la sesión, se notó un poco de preocupación, principalmente por las asesorías de cálculo diferencial, y a pesar que el trabajo mejoro notoriamente, se empezaron a buscar estrategias para garantizar la permanencia del grupo, principalmente por el aviso del estudiante que manifestó no continuar.

6.1.5. Sesión 04.

- **Presaberes**

Esta sesión está enmarcada en el hecho que se presentó el mayor número de deserción, ya que un estudiante perteneciente al grupo 01 manifestó que no asistiría más al curso, dado que necesitaba mejorar sus registros académicos

en calculo diferencial, y dos estudiantes del grupo 02 sin razón alguna no asistieron sembrando la incertidumbre no solo del tutor sino de sus compañeros, así que por esta práctica los tres estudiantes trabajaron en un único grupo.

En medio electrónico, se entregó la guía 04: funciones y vectores, el cual se ha observado por parte del investigador que a los estudiantes le cuesta trabajo adaptarse a este tema, además que se complementó con otro tema de conflicto para el estudiantado, el cual fue el de vectores. Con esta el estudiante refuerza su comprensión lectora, pensamiento algorítmico, lógico y matemáticos y la solución de problemas. Adicional los participantes al curso, manifestaron que llevaban algo de tiempo manejando las funciones en la materia, lo cual se preveía que no habría dificultades en cuanto a comprensión de lectura y algoritmia.

La primera parte de la guía, el estudiantado debía desarrollar una comprensión de lectura relacionada con la teoría correspondiente a las funciones, sus tipos y forma de declaración en arduino. Para evaluar su desempeño, se propuso una actividad, la cual consistía en tres ejercicios en lenguaje natural, en el cual debían aplicar funciones en lenguaje natural, en la cual el grupo contestó acertadamente todos los ejercicios. Además, se plantearon unas actividades lúdicas en la cual debían reestructurar y probar los montajes de led parpadeante y el semáforo de la guía 01, aplicando funciones en estas. Los resultados se resumen en la siguiente tabla 25:

	LED PARPADEANTE		SEMAFORO	
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
GRUPO UNICO	Realizo el montaje correctamente.	Crearon una función llamada led, en la cual programaron el funcionamiento del periférico.	Realizo el montaje correctamente.	Reutilizaron la función led programando el tiempo correspondiente para cada led.

Tabla 25: Observaciones de las practicas con led de la guía 04.

El siguiente tema trabajado fue el de vectores, en este se explicó brevemente y se ilustro mediante ejemplos su aplicabilidad, como actividad se plantearon tres momentos, el primero consistía en tres ejercicios en lenguaje natural en el cual debían aplicar vectores en ejercicios realizados anteriormente. En este parte se observó unas pequeñas dificultades en el momento de asimilar este concepto que poco a poco fue superado.

- **Saberes**

La actividad lúdica consistía en la realización de un sensor de movimiento, utilizando el sensor de detección de movimiento HCSR501, en el cual se explicó el paso a paso de su montaje. El estudiante, por su cuenta tenía que

realizar el código, en el cual al se evidenció que les costó trabajo, en el cual le llevó más tiempo. En este momento invirtieron alrededor de 95 minutos en su finalización, cuando se había presupuestado que gastarían solo 70 minutos.

En la tabla 26 se resume el desarrollo del montaje por parte de los estudiantes.

OBSERVACIONES DEL MONTAJE SENSOR DE MOVIMIENTO		
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
OBSERVACIONES GRUPO UNICO	Realizo el montaje correctamente, siguiendo las indicaciones de la guía.	Aplicaron correctamente el concepto de funciones en arduino.

Tabla 26: Observaciones del montaje del sensor de movimiento de la guía 04

En cuanto a los vectores, se les solicitó como reto reutilizar el montaje del semáforo, en el cual debían aplicar el concepto de vectores. En la tabla 27 se plasma el comportamiento observado durante la práctica del semáforo.

OBSERVACIONES DEL MONTAJE SEMÁFORO		
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
OBSERVACIONES GRUPO UNICO	Ya no invierten tanto tiempo en la realización del montaje.	Utilizaron el concepto de vectores apropiadamente en el código.

Tabla 27: Observaciones del montaje del semáforo de la guía 04.

Por ultimo y con tiempo de sobra y para completar el proceso de la guía, los estudiantes buscaron el concepto de matrices y se creó una mesa redonda, donde se les explicó y se les aclararon dudas acerca de la temática.

- **Validación**

La autoevaluación correspondiente a la guía, fue llenada de forma oral y los resultados fueron plasmados en una agenda. Los resultados fueron:

PREGUNTA	ESTUDIANT E 01	ESTUDIANT E 02	ESTUDIANT E 03
1. Con el desarrollo de esta guía reafirmó el concepto de función.	SI	SI	SI
2. Con lo desarrollado en esta guía, es capaz de diferenciar los tipos de funciones empleados en programación.	SI	SI	SI
3. Con el desarrollo de esta guía reafirmó el concepto de vectores.	SI	SI	SI
4. Con las guías realizadas anteriormente, ha fortalecido los conceptos que posee de programación	SI	SI	SI
5. Con el desarrollo de esta guía ha comprendido aún más la programación de la plataforma arduino.	SI	SI	SI

Tabla 28: Resultados de la autoevaluación de la guía 04.

Antes de terminar la sesión, los estudiantes participantes manifestaron que no asistirían la siguiente semana, razón que en esa correspondía a la semana de parciales y necesitaban el tiempo para prepararlos. Así que se acordó primero ponerse en contacto con los estudiantes que no habían manifestado dejar de asistir y reanudar las sesiones una vez terminaran los parciales.

En esta sesión, se observó que el trabajo con el grupo restante fluyó más rápido y los estudiantes participantes se acoplaron, sin embargo, había que buscar estrategias para que los compañeros ausentes retornaran. Con el tiempo dado se esperaba que a la siguiente sesión regresaran al menos dos estudiantes al ver sus notas de segundo corte.

6.1.6. Sesión 05.

- **Presaberes**

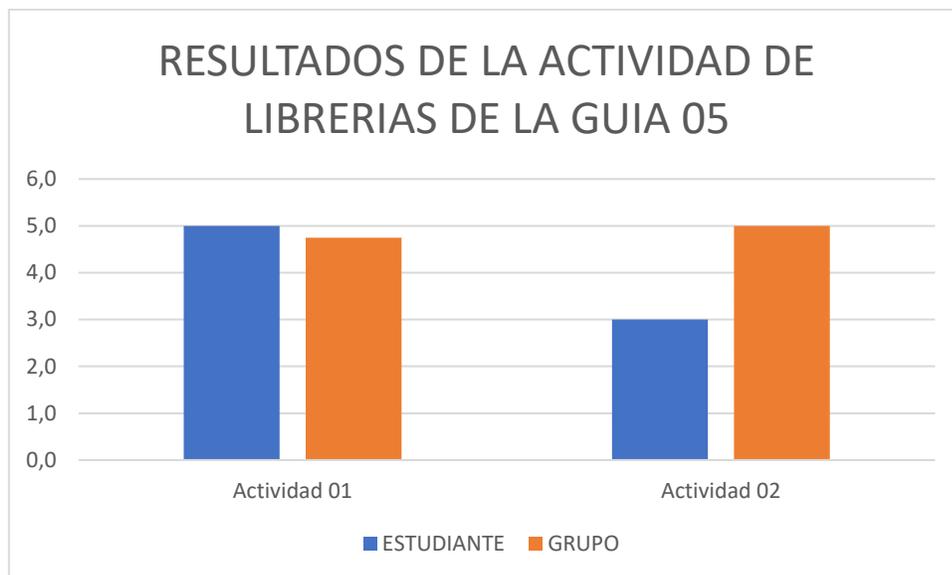
Como se ha venido haciendo en sesiones anteriores, se les entregó la gua 05: Librerías. En esta guía se buscó reforzar habilidades del pensamiento computacional tales como comprensión lectora, reconocimiento de patrones, pensamiento lógico y solución de problemas. Cabe aclarar, que un estudiante que pertenecía al grupo 01 inicial haría la práctica solo y los otros dos los cuales pertenecían a los grupos 01 y 02 se unirían. Esto dio pie a probar si el trabajo en equipo es mejor al de una sola persona.

Los estudiantes leyeron la teoría correspondiente a las librerías, su forma de reconocerlas en arduino y su forma de invocación. De esta lectura se realizó una actividad la cual comprendían en una serie de preguntas, las cuales tenían un valor de 1.0, las 4 primeras eran concerniente a la lectura, la 5 era buscar en internet el concepto y algunos métodos de ciertas librerías. Las observaciones de la actividad se resumen en la tabla 29:

NOTA ACTIVIDAD LIBRERIAS		
	Actividad 01: Comprensión de lectura (preguntas a hasta d)	Actividad 02: Búsqueda de algunas librerías con alguna de las funciones
OBSERVACIONES ESTUDIANTE	Respondió acertadamente tres de las cuatro preguntas.	Realizo la búsqueda de las cinco funciones sugeridas.
OBSERVACIONES GRUPO	Respondió acertadamente las cuatro preguntas sugeridas.	Realizo la búsqueda de tres de las cinco funciones sugeridas.

Tabla 29: Actividad de librería de la guía 05.

Los resultados de la actividad en cuanto a su valor numérico, se muestran en la gráfica 7.



Gráfica 7: resultados de la actividad de librerías.

Como se apreció el trabajo grupal venció al del estudiante en cuanto a búsqueda de información (actividad 02). En esta se observó una buena comprensión lectora por parte del estudiantado, ya que en un tema relacionado con el programa parece mostrar mayor atención por parte del grupo lo que determina que hay temas en los cuales al estudiante le presta más atención que a otros.

- **Saberes**

La actividad lúdica, correspondía a la creación de un sensor de temperatura, la cual se sugirió realizarlo en conjunto, debido a las múltiples conexiones del periférico LCD de 16X2, la cual tiene un número alto de conexiones. Esta práctica tenía la particularidad que debían usar los dos métodos de importación de librerías (una que se encuentra en el IDE de arduino y una que debía ser descargada e instalada). La primera parte consistía en montar la pantalla LCD y mostrar un determinado mensaje, la segunda parte consistía en adicionarle al montaje del LCD el sensor de temperatura. Los detalles se resumen en la siguiente tabla 30:

	MONTAJE LCD 16x2		MONTAJE CON SENSOR DE TEMPERATURA	
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
GRUPO UNICO	Realizo el montaje sin ningún tipo de error.	Importaron correctamente la librería, investigaron e implementaron las funciones	Integraron bien el sensor de temperatura y humedad.	Investigaron, instalaron e implementaron correctamente la librería correspondiente a

		correspondiente correctamente. un estudiante no recordaba correctamente la sintaxis de arduino.		la actividad, no presentaron ninguna dificultad
--	--	---	--	---

Tabla 30: Observaciones de la actividad lúdica de la guía 05.

- **Validaciones.**

Terminada la práctica, cada estudiante contestó la autoevaluación, la cual se resume en la siguiente tabla 31:

PREGUNTA	ESTUDIANT E 01	ESTUDIANT E 02	ESTUDIANT E 03
1. Con su investigación, domina el concepto de matrices.	SI	SI	SI
2. Con el desarrollo de esta guía domina el concepto librería	SI	SI	SI
3. Con lo desarrollado en esta guía, es capaz de importar la librería correspondiente para un caso específico	SI	SI	SI
4. Con las guías realizadas anteriormente, ha fortalecido los conceptos que posee de programación	SI	SI	SI
5. Con el desarrollo de esta guía ha comprendido aún más la programación de la plataforma arduino.	SI	SI	SI

Tabla 31: Resultados de la autoevaluación de la guía 05.

Esta sesión, la cual se realizó casi un mes después de la anterior, se consolidó el grupo con el que se finalizaría el curso (3 estudiantes). Cabe resaltar que, durante la sesión, se produjeron varios cortes de energía los cuales duraron alrededor de 40 minutos, por lo cual tocó extender un poco la sesión para poder completar la guía.

Se observó que el grupo trabajó con mayor fluidez y el grupo que quedó es un grupo trabajador. Los estudiantes participantes tenían el interés por aprender y mejorar sus calificaciones, además que se les había dado un incentivo en las notas correspondientes a la materia de programación estructurada. Además, que los estudiantes quedaron entusiasmados con la práctica realizada, creando un ambiente de optimismo y satisfacción.

6.1.7. Sesión 06.

- **Presaberes.**

Esta sesión comenzó con la entrega de la guía 06: aplicación de librerías: conectividad por bluetooth, la cual se habló de la tecnología bluetooth, manejando como aspecto clave la comprensión de lectura y la solución de problemas. Esta guía solo contenía una lectura la cual los estudiantes realizaron de forma individual, evidenciando que el estudiante 01 tenía una lectura rápida, terminando antes que sus compañeros.

- **Saberes.**

Se realizaron dos montajes para esta guía, el primero consistía en controlar una serie de led mediante un dispositivo móvil, la segunda era simular el comportamiento de un dado con ayuda de un led 7 segmentos. Las observaciones por estudiante se resumen en la siguiente tabla 32:

	MONTAJE LEDS		LED 7 SEGMENTOS	
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
ESTUDIANTE 01	Realizo el montaje del bluetooth correctamente, no presentó dificultades en ningún aspecto y se notó que dominaba la parte electrónica	Realizo bien la configuración del módulo bluetooth, realizo correctamente el código.	Hizo correctamente el montaje del 7 segmento siguiendo cada una de las conexiones.	Investigó el código en cuanto al funcionamiento del 7 segmentos, investigó e implementó la función que le permite generar números aleatoriamente.
ESTUDIANTE 02	Demoró, pero realizo correctamente el montaje.	Realizó el código correctamente, le ha tomado trabajo retomar a la sintaxis de arduino.	Le cuesta trabajo realizar el montaje, ya que recibió ayuda de sus compañeros.	Investigó una forma de ejecutar el 7 segmentos distinta, esta funcionó correctamente.
ESTUDIANTE 03	Realizó el montaje muy rápido, pero tuvo problemas en cuanto a la conexión con el bluetooth, que demoro en solucionar.	En cuanto al código que operaban los leds, no le cuesta trabajo y logra configurar bien tanto el bluetooth como la aplicación móvil.	Demoro más que los demás compañeros, ya que se ocupó de realizar primero el código	Realizo un buen código ya que había investigado como realizarlo. No tuvo contratiempos en configurar el bluetooth.

Tabla 32: Observaciones de la actividad lúdica de la guía 06.

- **Validación**

Una vez terminada la sesión, los estudiantes de forma oral e individual llenaron la autoevaluación, las respuestas corresponden a la siguiente tabla 33:

PREGUNTA	ESTUDIANTE 01	ESTUDIANTE 02	ESTUDIANT E 03
1. Con el desarrollo de esta guía conoció los conceptos de la tecnología bluetooth.	SI	SI	SI
2. Con lo desarrollado en esta guía, reforzo los conceptos de programación vistos hasta ahora	SI	SI	SI
3. Considera que, hasta el momento, ha mejorado sus capacidades en cuanto a extracción y comprensión de información.	SI	SI	SI
4. En escala de 1 a 5, ha visto mejoría en los conceptos de programación, siendo: 1 muy bajo, 2 bajo, 3 medio, 4 bueno, 5 excelente.	4	4	4

Tabla 33: Resultados de la autoevaluación de la guía 05.

Esta sesión se realizó 2 semana después de la anterior, dado que los estudiantes estaban adelantando trabajos y parciales, y no se encontró un momento previo para la reunión. En esta sesión se observó cansancio por parte de los participantes por la conclusión de sus estudios, en cuanto a su trabajo, el estudiante 01 demostraba un mayor dominio de la temática, a los otros dos estudiantes les costó retomar el ritmo de programación en arduino.

6.1.8. Sesión 07.

- **Presaberes**

Para poder terminar el curso, y en vista que solo quedaban dos días para terminar el semestre, se decidió con antelación realizar la sesión con duración de 4 horas, con el propósito de resolver dos guías (07 y 08). Estas guías están enmarcadas en temáticas de programación orientada a objetos, materia la cual los estudiantes verían el siguiente semestre.

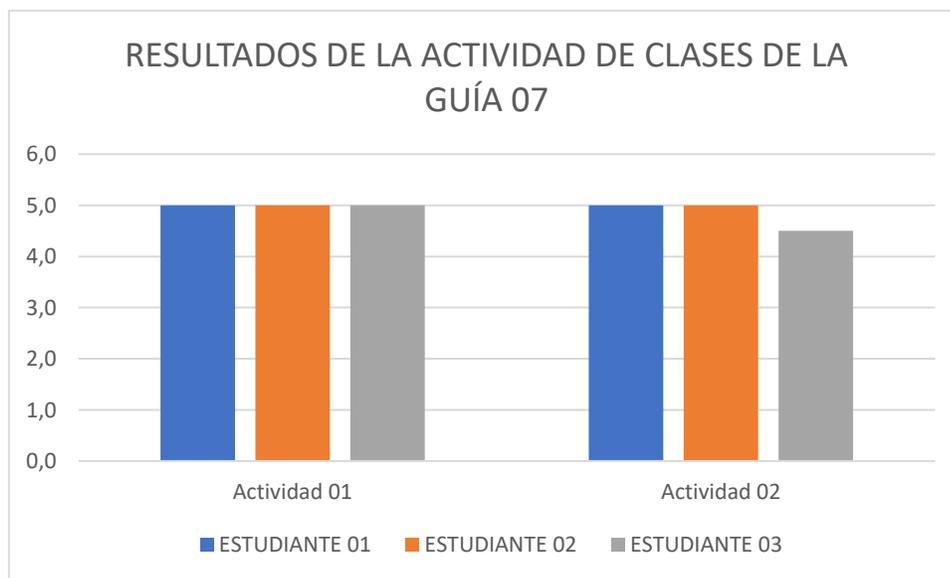
La guía 07: Introducción a la programación orientada a objetos se desarrollarán habilidades tales como comprensión lectora, pensamiento lógico y algorítmico y solución de problemas. En la primera parte de la sesión se desarrolló la guía 07, esta comenzó con la lectura correspondiente a la introducción a la programación orientada a objetos, esta se ilustró mediante un ejemplo, ya que es la forma más sencilla y entendible para el estudiantado. En esta práctica se observó el interés y la expectativa por el tema, ya que en algunas ocasiones los participantes habían consultado con el investigador aspectos relacionados al programa. En esta guía se explicó lo que es la programación orientada a objetos, sus principios y el concepto de clase.

La actividad correspondiente a esta guía, está enmarcada en dos momentos, la primera consistía en completar un ejemplo, en el cual debían llenar con sus atributos y métodos, la segunda consistía en que el estudiante propusiera un ejemplo de clase, con sus atributos y métodos correspondientes. Las observaciones de la actividad se resumen en la tabla 34.

OBJETIVO DE LA ACTIVIDAD	OBSERVACIONES	
	ACTIVIDAD 01	ACTIVIDAD 02
OBJETIVO DE LA ACTIVIDAD	Complete el siguiente cuadro (ejemplo del árbol)	De 1 ejemplo en donde pueda aplicar el uso de clases en la vida cotidiana, encuentre sus atributos y métodos.
ESTUDIANTE 01	Identificó algunos atributos y métodos correctamente.	Realizó el ejemplo correctamente, identificando atributos y métodos.
ESTUDIANTE 02	Identificó algunos atributos y métodos correctamente.	Realizó el ejemplo correctamente, identificando atributos y métodos.
ESTUDIANTE 03	Identificó algunos atributos y métodos correctamente.	Se le dificultó identificar los atributos, muy diferente a los métodos ya que los identificó correctamente.

Tabla 34: Observaciones de la actividad de clases.

Los resultados de la actividad, se evidencia en la gráfica 8:



Gráfica 8: Resultados de la actividad de clases de la guía 07.

Esta actividad fue entregada por parte del estudiante de forma física, dado que para ese momento los salones se encontraban ocupados por una actividad del programa y ninguno de los asistentes había llevado un equipo de cómputo, la guía fue leída en un proyector de manera grupal. Como se puede apreciar, solo el estudiante 03 tuvo un bajo rendimiento con respecto a sus compañeros,

sin embargo, se percibió en el ambiente que los participantes habían comprendido el temario.

La guía 08: Programación orientada a objetos, herencia fue resuelta de manera electrónica, ya que para ese momento ya se disponía de un salón acorde para la realización de la guía. Esta guía pretende reforzar las habilidades en cuanto a Comprensión lectora, Pensamiento algorítmico, Pensamiento lógico y Resolución de problemas.

La comprensión de lectura, referente al temario de herencia, el cual se explicó paso a paso a través de un ejemplo. En la actividad pactada, la cual fue realizada en el tablero de forma conjunta, esto sirvió para medir el nivel de entendimiento de los estudiantes. En la tabla 35 se resume lo observado durante la actividad.

ESTUDIANTE	OBSERVACIÓN
ESTUDIANTE 01	Fue quien más aportó, indicando al compañero que pasó al tablero lo que debía copiar, entendió muy bien el tema
ESTUDIANTE 02	Fue quien paso al tablero, aportando principalmente en las subclases.
ESTUDIANTE 03	Aporto en poca medida, pero se notaba que comprendió el tema ya que, en sus aportes, lo hacía acertadamente.

Tabla 35: Observaciones de la actividad de herencia.

- **Saberes**

En la guía 07, no se explicó un montaje en cuanto a hardware, ya que el objetivo planteado era mostrarle al estudiante otra forma de programar y que el siguiente curso académico, la adaptabilidad no fuera tan pesada para ellos. Cabe aclarar, que solo se le explicó la forma de programar en arduino con este paradigma. Adicionalmente, se expusieron en la temática los conceptos den constructor y los niveles de visibilidad, temas propios de la programación orientada a objetos. En la tabla 36 se resume el proceso que se observó durante el montaje.

ESTUDIANTE	MONTAJE LEDS		MONTAJE ULTRASONIDO	
	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO	ASPECTOS DE MONTAJE	ASPECTO DE CÓDIGO
ESTUDIANTE 01	Domina el montaje, ya que no le cuesta trabajo realizarlo.	Asimilo correctamente el concepto de clases, reacomodó el código correctamente.	Domina el montaje, ya que no le cuesta trabajo realizarlo.	Acomodo el código anterior con clases, lo realizo correctamente.

ESTUDIANTE 02	Domina el montaje, ya que no le cuesta trabajo realizarlo.	Asimilo correctamente el concepto de clases, reacomodó el código correctamente.	Domina el montaje, ya que no le cuesta trabajo realizarlo.	Acomodo el código anterior con clases, lo realizo correctamente.
ESTUDIANTE 03	Domina el montaje, ya que no le cuesta trabajo realizarlo.	Asimilo correctamente el concepto de clases, reacomodó el código correctamente.	Domina el montaje, ya que no le cuesta trabajo realizarlo.	Acomodo el código anterior con clases, lo realizo correctamente.

Tabla 36: Observaciones de la actividad lúdica de la guía 07.

Con respecto a la práctica de la guía 08, en la parte lúdica, la cual se pidió montar un semáforo inteligente (con leds y el ultrasonido) para aplicar el concepto de herencia. En esta se hizo de manera grupal ya que el tiempo de la sesión se estaba terminando y los estudiantes manifestaban cansancio. En esta se observó que los estudiantes siguieron el paso a paso contemplado en la guía, integraron el código correctamente.

- **Validación.**

Las dos guías tenían la autoevaluación, la cual debía ser llenada por los estudiantes, a continuación, en la tabla 37 se muestra las respuestas de los estudiantes a dichas preguntas.

PREGUNTAS	ESTUDIANTE 01	ESTUDIANT E 02	ESTUDIANTE 03
GUÍA 07			
1. Comprendió el concepto de Programación orientada a Objetos.	SI	SI	SI
2. Comprendió el concepto de Clases.	SI	SI	SI
3. Identifica semejanzas y diferencias entre la programación estructurada y la programación orientada a objetos.	SI	SI	SI
4. Es capaz de crear una clase en arduino	SI	SI	SI
GUÍA 08			
1. Entiende el principio de Herencia de la programación orientada a objetos	SI	SI	SI
2. Identifica una clase de una subclase.	SI	SI	SI
3. Es capaz de aplicar el concepto de clases en arduino.	SI	SI	SI

Tabla 37: Autoevaluación de la guía 07 y guía 08.

Esta sesión se realizó tiempo después de la anterior, ya que los estudiantes participantes se encontraban en temporada de parciales, y ellos ya estaban disponibles para la conclusión del curso.

En esta sesión, se empezó a ver ya una satisfacción por parte de los participantes, además de un cansancio, que en escasas ocasiones los estudiantes hacían pausas para continuar con el trabajo. En cuanto a lo académico, se les indago acerca de cómo les había ido en el semestre, principalmente en las materias de programación estructurada y calculo diferencial, en la cual solo un estudiante manifestó haber aprobado calculo, en cuanto a programación, ya los tres estudiantes manifestaron haberla aprobado.

6.1.9. Sesión 08.

- **Presaberes.**

Al igual que en la sesión anterior, se iban a realizar dos guías, sin embargo y como las guías 09 y 10 costaban solo de ejercicios, que permitirían al estudiante desarrollar sus habilidades en cuanto a solución de problemas, pensamiento lógico, pensamiento algorítmico, pensamiento matemático y comprensión de lectura, se acordó que, mediante un sorteo, el estudiante realizara un ejercicio de cualquiera de las dos guías. Tenía un plazo de 2 horas para resolver el ejercicio, en la siguiente debía pasar a explicarlo y los compañeros debían aportar sus opiniones.

- **Saberes**

Las guías 09: retroalimentación de arduino 01 y guía 10: retroalimentación de arduino 02 contenían cada una dos ejercicios que buscaban medir el nivel de aprendizaje, ya que debían aplicar todo lo visto no solo en la programación de arduino, sino en las habilidades de pensamiento computacional desarrolladas durante el curso. En la tabla 38 se resume las observaciones del taller.

EJERCICIO CORRESPONDIENTE		OBSERVACIONES
ESTUDIANTE 01	Ejercicio 02 de la guía 09 (medidor de batería)	El estudiante investigó el procedimiento en la web, realizo todos los paso correspondientes y consiguió realizar el montaje correctamente, no tuvo comentarios por parte de sus compañeros.
ESTUDIANTE 02	Ejercicio 02 guía 10 (controlar remotamente la intensidad de un led)	El estudiante realizo correctamente el montaje de los led, tuvo contratiempo en cuanto al montaje del bluetooth, pero encontró la forma de graduar la intensidad de los leds. No recibió opiniones negativas por parte de sus compañeros.

ESTUDIANTE 03	Ejercicio 01 de la guía 09 (dado con leds)	Realizo correctamente el montaje de los leds, le costó algo de trabajo la lógica manejada para el programa, principalmente en el encendido de los leds, el cual sus compañeros le ayudaron a solucionar.
---------------	--	--

Tabla 38: Observaciones de los ejercicios realizados en la sesión.

- **Validación**

Se observó que cada estudiante realizó su respectiva práctica, los estudiantes invirtieron entre 60 y 85 minutos en la realización tanto de los montajes, se les pidió que realizaran el pseudocódigo para observar el desarrollo del pensamiento computacional, ya que en este tenían que aplicar el proceso de descomposición, abstracción, reconocimiento de patrones y algoritmia en el problema planteado.

Como era la última “clase” que los estudiantes tenían, se notaba un cansancio, principalmente en el afán de terminar cada uno sus montajes, sin embargo, a pesar de esto realizaron los montajes adecuados para cada caso.

Como último momento y siendo esta la sesión final, se les aplicó el test final, que al igual que el anterior costaba de 10 preguntas de selección múltiples, el cual los detalles serán expuestos en la siguiente sección.

6.2. TEST DE DIAGNOSTICO DE PENSAMIENTO COMPUTACIONAL.

Los test tenían la finalidad de dar un diagnóstico acerca de cómo el estudiante llega al curso y con que el estudiante sale del mismo. En cada uno de los test había 10 preguntas de opción múltiple tipo pruebas de estado, cada pregunta abordaba un aspecto del pensamiento computacional, el cual el estudiante disponía de 30 minutos para contestarlo (en promedio de 3 minutos por pregunta), estas tenían un valor de 10 puntos cada una. A continuación, se describe el desarrollo y los resultados obtenidos en cada uno de los test.

6.2.1. Resultados test inicial.

Este test se aplicó comenzando el curso (primera sesión), este taller tuvo una duración de 35 minutos (5 más de lo presupuestado), este taller lo resolvieron 7 estudiantes a los cuales se les entregó este test de forma física. El test contenía 10 preguntas de selección múltiple con un valor por pregunta de 10 si acertaba, los resultados se pueden apreciar en la tabla 39 :

Pregunta	Aspecto del pensamiento computacional	Estudiante 01	Estudiante 02	Estudiante 03	Estudiante 04	Estudiante 05	Estudiante 06	Estudiante 07
1	Pensamiento lógico-matemático	10	0	10	10	10	0	0
2	Pensamiento lógico-matemático	10	0	10	10	0	0	0
3	Comprensión lectora	0	0	0	10	0	0	10
4	Reconocimiento de patrones	10	10	0	0	10	0	0
5	Reconocimiento de patrones	10	10	10	10	10	10	10
6	Pensamiento lógico-matemático	10	10	0	10	10	0	0
7	Pensamiento matemático	10	10	10	0	10	0	10
8	Pensamiento matemático	0	0	10	0	10	0	0
9	Reconocimiento de patrones	10	10	10	0	10	10	0
10	Comprensión lectora	0	10	0	0	10	0	0
Total		70	60	60	50	80	20	30

Tabla 39: Resultados del test inicial

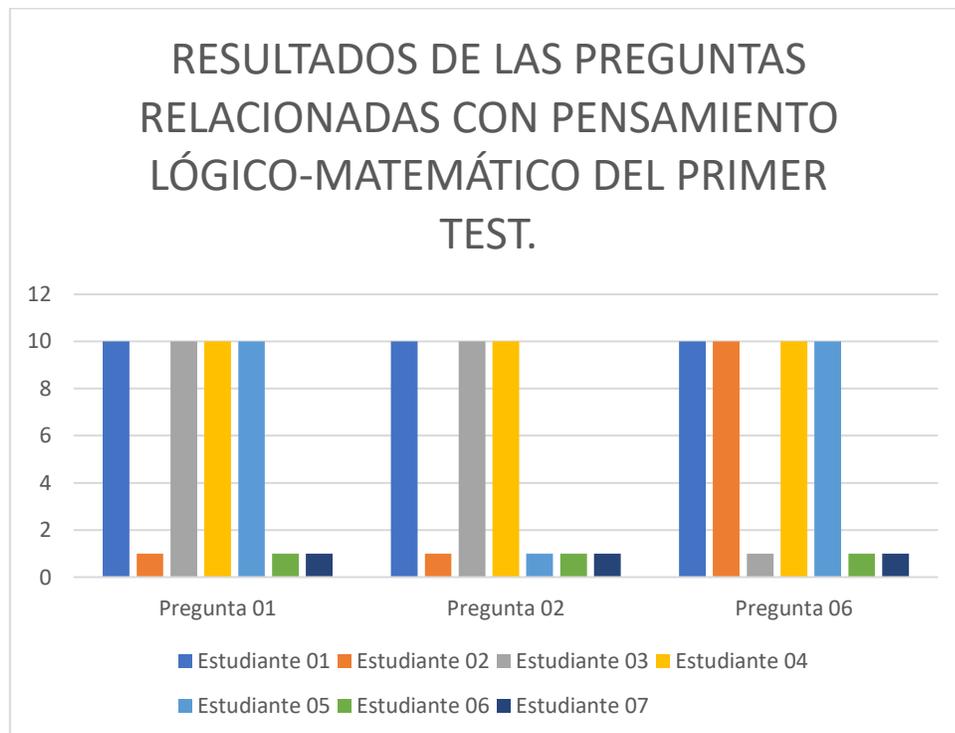
Promediando los resultados generales y teniendo en cuenta la tabla 40, el grupo obtuvo un 52,86 de su nota, lo cual es aceptable. Solo dos estudiantes tuvieron un desempeño inferior al de sus compañeros que sacaron menos de 30, uno estuvo en la mitad de la nota (50), dos en 60 y 1 en 70 y la mayor nota fue del estudiante 5 que tuvo un promedio de 80. A continuación se explicara los resultados de cada una de las subcategorías de las preguntas del test, cabe destacar.

Rango	Calificación
00 - 40	Deficiente
41 - 50	Bajo
51 - 65	Aceptable
65 - 70	Bueno
71 - 90	Sobresaliente
91 - 100	Excelente

Tabla 40: Rangos de calificación de los test de diagnóstico.

- **6.1.2.1. Preguntas de pensamiento lógico-matemático**

En este aspecto, se manejaron tres preguntas, las cuales arrojaron los resultados mostrados en la gráfica 09:



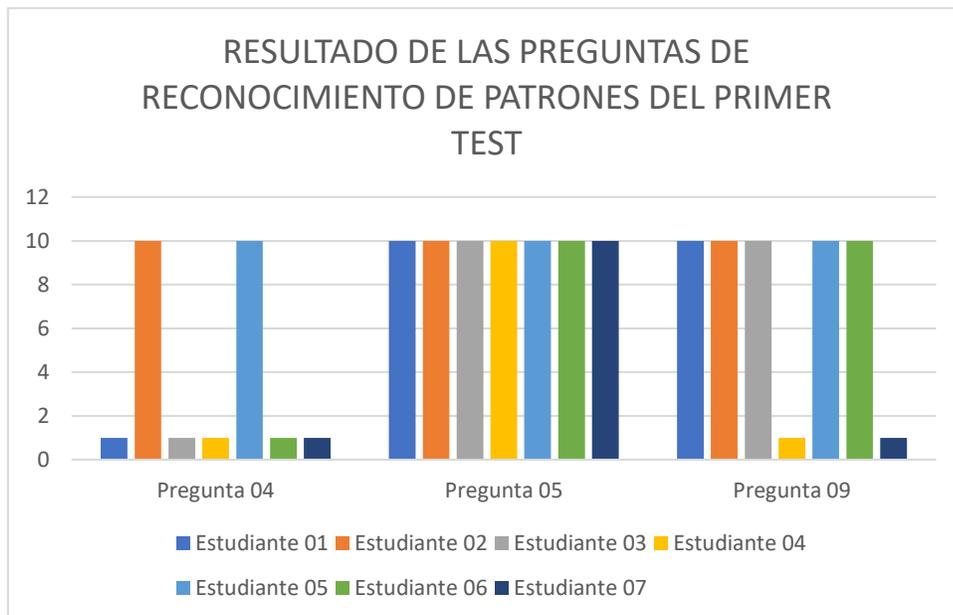
Gráfica 9: Resultados de las preguntas relacionadas con pensamiento lógico-matemático del primer test.

Cada barra representa a cada uno de los estudiantes y su desempeño en la pregunta, cabe aclarar que para la calificación del test se utilizó el valor cero (0) si no se acertaba, para realizar las gráficas se usara el valor de uno (1) para una mejor visualización.

Las preguntas enmarcadas en este aspecto eran la 1, 2 y 6. La pregunta 1 la cual consistía en la torre de Hanói con tres piezas, el cual el estudiante debía seleccionar los mínimos movimientos correctos. Como se aprecia en la gráfica, cuatro estudiantes acertaron (1,3,4 y 5) equivalente al 57.14% del grupo. La pregunta 2, está relacionada en cierto modo con la primera, debido a que, basándose en la torre de Hanói, debía decir cuáles son los tres movimientos de las fichas que garantizara llegar al mínimo de movimientos, tres estudiantes (1,3 y 4) acertaron equivalente al 42,86% del grupo. La pregunta 6, estaba enmarcada en un análisis en el cual se plantea como bajar un bajar un récord olímpico, sin bajar el récord mundial, en esta ocasión suponía un análisis profundo en el cual cuatro estudiantes (1,2,4 y 5) acertaron correctamente (57.14%).

6.2.1.2. Preguntas de reconocimiento de patrones

Para esta ocasión se plantearon tres preguntas las cuales fueron la pregunta 4, 5 y 9 y los resultados obtenidos se representan en la gráfica 10:

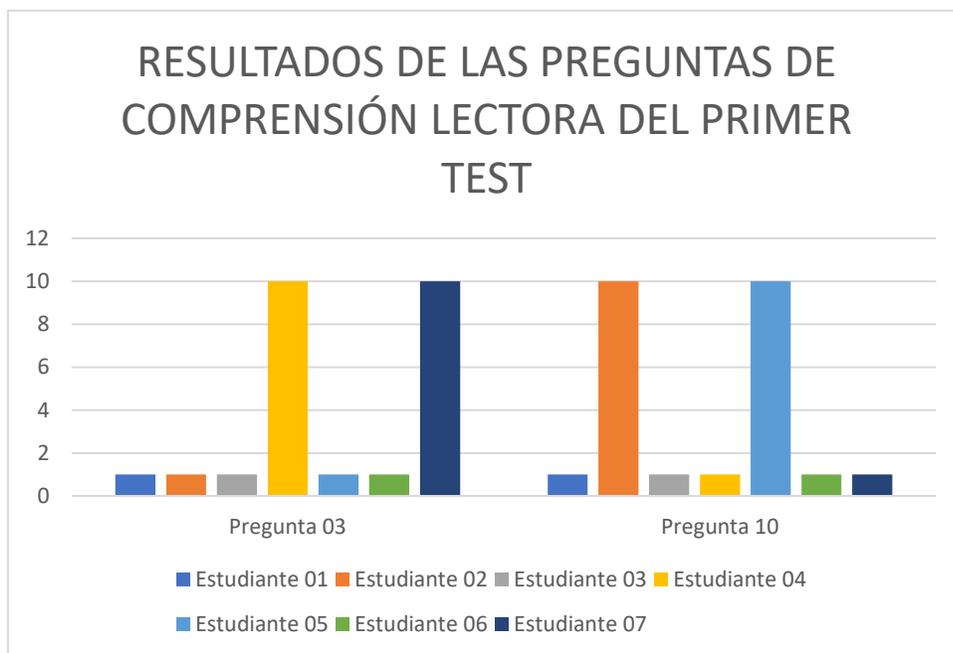


Gráfica 10: Resultados de las preguntas relacionadas con reconocimiento de patrones del primer test.

La pregunta 04, 05 y 09 eran secuencia de figuras, en las cuales el estudiante debía responder cual era la figura correspondiente a la secuencia. La pregunta 04 solo el 28.57% del grupo acertó (dos estudiantes, los cuales fueron el 2 y 5) la 05 fue un acierto del 100% y la pregunta 09 el cual consistía en determinar el orden de la llegada de los trenes a la estación, esta pregunta solo los estudiantes 4 y 7 no acertaron correctamente, lo cual fue un acierto del 71.43%.

- **6.2.1.3. Preguntas de lectura crítica**

Las preguntas correspondientes fueron la 03 y la 10, en la cuales los resultados se evidencian en la siguiente gráfica (gráfica 11):

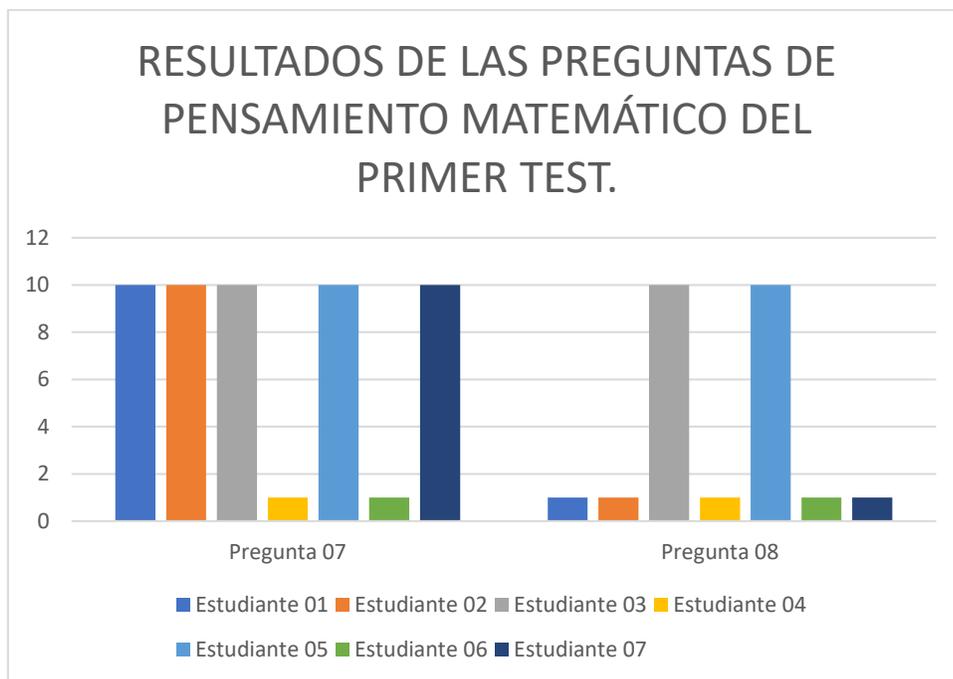


Gráfica 11: Resultados de las preguntas de comprensión lectora del primer test.

La pregunta 03, la cual, en base de un párrafo, que habla del impacto de los centros comerciales en la actualidad, el estudiante en base a ese texto debía seleccionar la respuesta que consideraba correcta, en este punto los estudiantes 4 y 7 acertaron correctamente (28.57%). La pregunta 10, la lectura propuesta habla de la importancia de la puntualidad y la pregunta correspondía a la tesis propuesta por el autor, fue el mismo resultado del punto anterior, es decir solo el 28.57% del grupo acertó (estudiantes 2 y 5).

- **6.2.1.4 Preguntas de pensamiento matemático**

Las preguntas 07 y 08 fueron las encargadas de dar un diagnóstico en este aspecto la cual arrojó los siguientes resultados, observados en la gráfica 12:



Gráfica 12: Resultados de las preguntas de pensamiento matemático del primer test.

Las dos preguntas estaban relacionadas con un problema de física básica, el cual consistía en el recorrido de un bus de un punto a otro, el inciso 07 se debía calcular la velocidad que lleva el autobús en un determinado tiempo, solo los estudiantes 4 y 6 fallaron (71.43% de acierto), en cuanto a la pregunta 08, el cual con base en algunos cálculos debía determinar si el vehículo invertía menos, igual o mayor tiempo del presupuestado, en este análisis solo los estudiantes 03 y 05 acertaron correctamente (28.57%).

6.1.1. Diagnóstico del test final.

Este test se realizó después de terminar la última guía (Octava sesión), al igual que el anterior conto con 10 preguntas agrupadas en los 4 aspectos trabajados en el anterior. En promedio duraron entre 35 y 40 minutos, tiempo presupuestado para la realización del mismo. Debido a que solo terminaron tres estudiantes, a estos se les aplicó y los resultados de estos se ven reflejados en la tabla 41:

Resultados del test final de pensamiento computacional				
Pregunta	Aspecto del pensamiento computacional	Estudiante 01	Estudiante 02	Estudiante 03
1	Comprensión lectora	10	10	10
2	Comprensión lectora	0	0	0

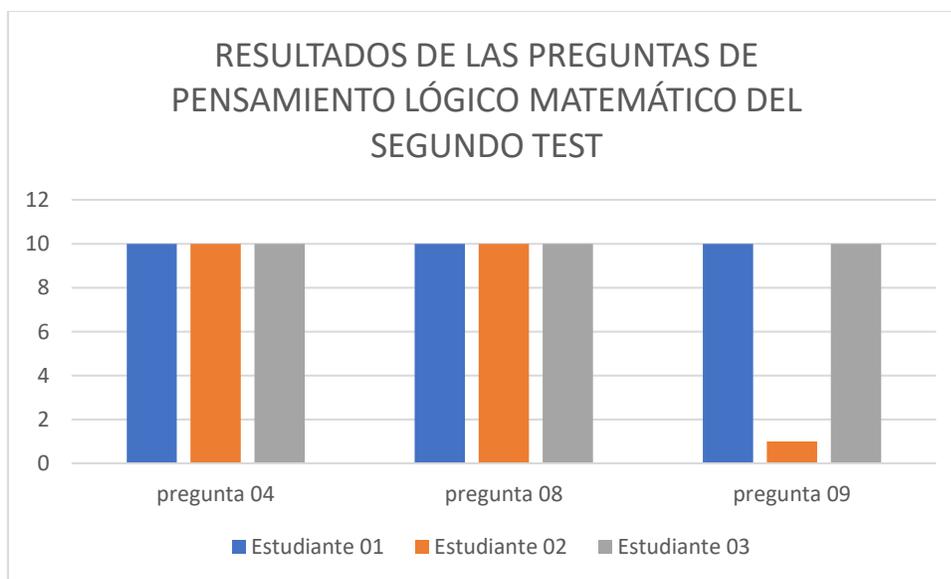
3	Reconocimiento de patrones	10	10	10
4	Pensamiento lógico-matemático	10	10	10
5	Pensamiento matemático	0	10	10
6	Pensamiento matemático	0	0	0
7	Pensamiento matemático	10	10	10
8	Pensamiento lógico-matemático	10	10	10
9	Pensamiento lógico-matemático	10	0	10
10	Reconocimiento de patrones	10	0	10
Total		70	60	80

Tabla 41: Resultados del segundo test.

Los tres estudiantes tuvieron un resultado superior al 50%, lo cual fueron buenas notas. A continuación, se analizará por cada uno de los aspectos evaluados, cabe recordar que, para poder visualizar bien los resultados de las gráficas, las preguntas incorrectas tendrán un valor de 1.0.

6.1.1.1. Preguntas de pensamiento lógico-matemático

Estas preguntas fueron la 04, 08 y 09 las cuales los resultados se resumen en la gráfica 13:



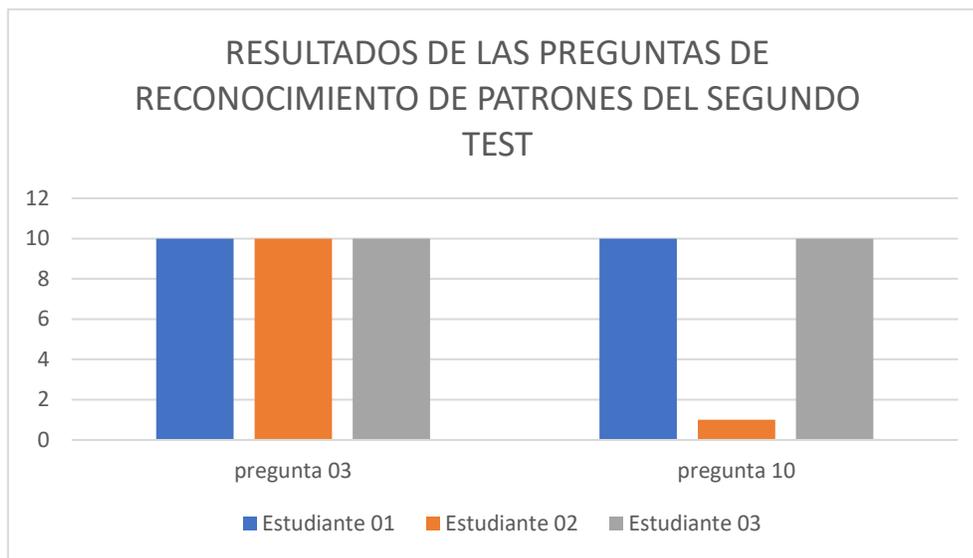
Gráfica 13: Resultados de las preguntas de pensamiento lógico matemático del segundo test.

La pregunta 04, correspondía a un robot, el cual debía seguir un determinado camino y recogía una cierta cantidad de dulces, el acierto fue del 100% la pregunta 08 corresponde a un problema de una empresa de lácteos y debía determinar cuanta materia prima se invertía en su elaboración, el acierto fue del 100%, en cambio en la pregunta 09 tuvo un 66.7% de acierto, este

problema consistía en un análisis del cambio de moneda en el cual el estudiante debía seleccionar la teoría correcta respecto a ese análisis.

6.1.1.2. Preguntas de reconocimiento de patrones

En este aspecto, fueron las preguntas 03 y 10, las cuales arrojaron los resultados mostrados en la gráfica 14.

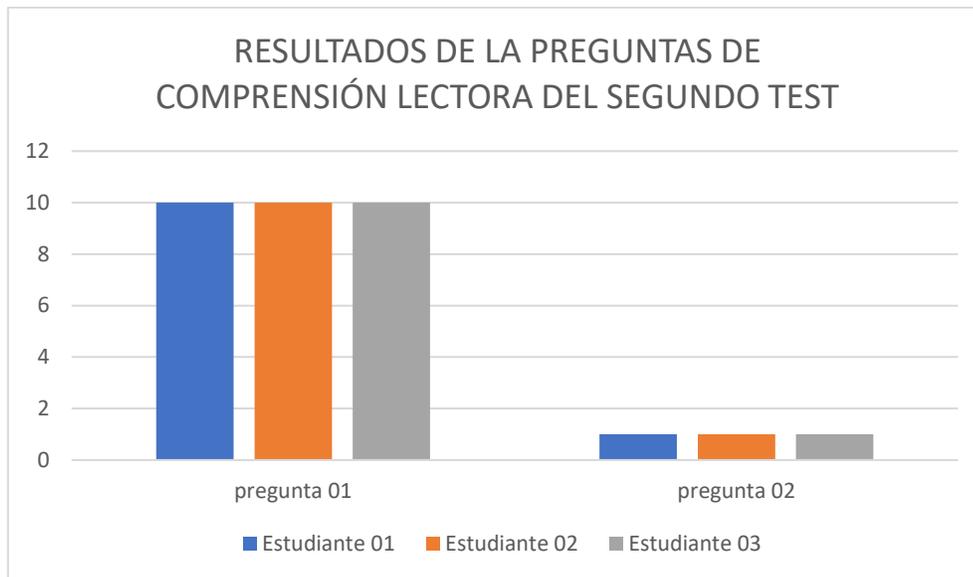


Gráfica 14: Resultados de las preguntas de reconocimiento de patrones del segundo test.

La pregunta 03 consistía en identificar el patrón de un juego de dominó, el cual el acierto fue total por parte del estudiantado, la pregunta 10 correspondía a identificar el movimiento de cierto engrane, solo el estudiante 02 falló en la identificación de este. Con esto demuestra que los estudiantes del grupo dominan este aspecto, solo al estudiante 02 se le dificulta un poco.

6.1.1.3. Preguntas de lectura crítica

Las preguntas correspondientes a esta fueron la 01 y 02, en la gráfica 15 se muestran los resultados de los estudiantes con respecto a esas preguntas:

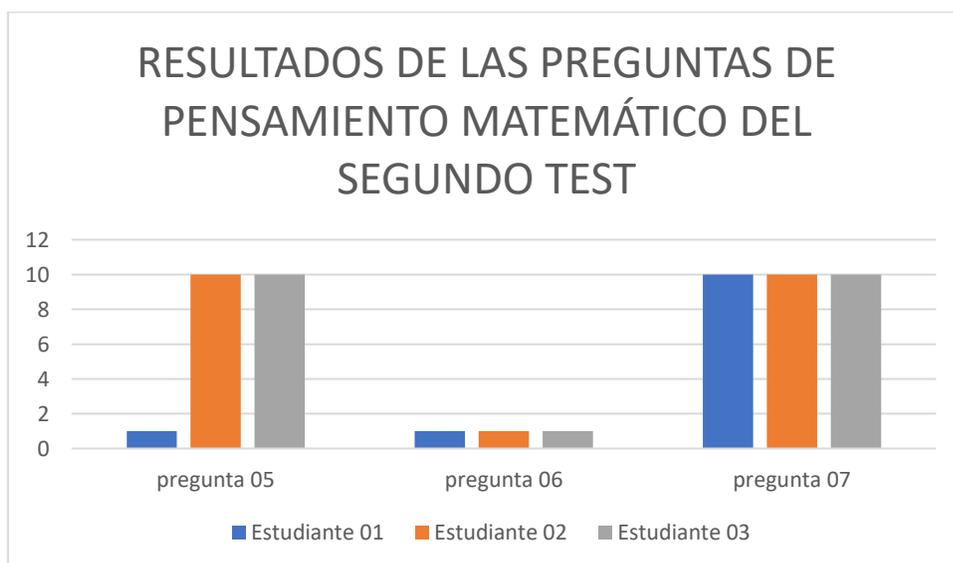


Gráfica 15: Resultados de la preguntas de comprensión lectora del segundo test.

Las dos preguntas de esta sección están relacionadas con una lectura, la cual habla de una visión del autor acerca del conocimiento. En la pregunta 01 el estudiante debía de escoger la afirmación que más se adaptaba al texto, fue un acierto total, caso contrario en la pregunta 02 la cual consistía en determinar el contenido del texto con su título, en esta no acertó ninguno de los estudiantes.

6.1.1.4. Preguntas de pensamiento matemático

Las preguntas correspondientes a este aspecto fueron la 05, 06 y 07 y los resultados fueron los apreciados en la gráfica 16:

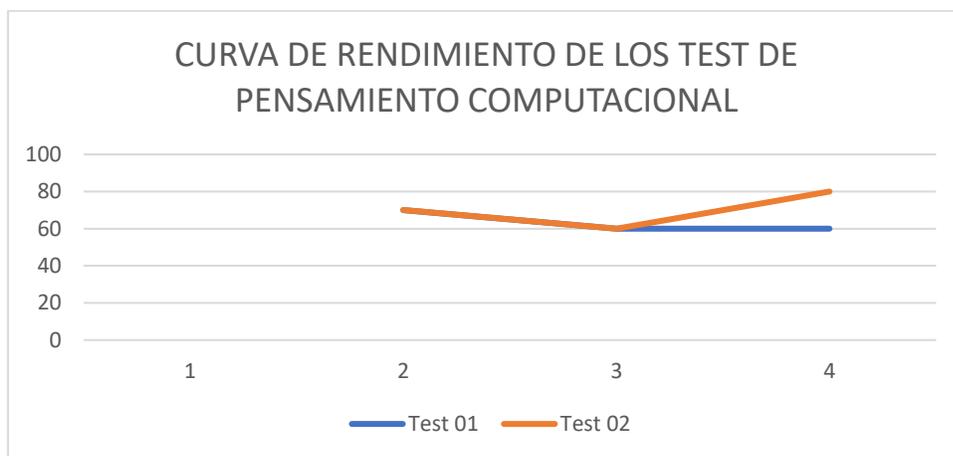


Gráfica 16: Resultados de las preguntas de pensamiento matemático del segundo test.

Las preguntas 05 y 06, tenían relación con un problema, en el cual consistía en el valor de unos cursos de capacitación para una empresa, en la pregunta 05 se debía calcular el valor monetario para cierta capacitación en la cual los estudiantes 2 y 3 acertaron, en la pregunta 6 dependiendo de la información suministrada debían definir el número de participantes del grupo, en este punto no hubo aciertos. El punto 07 consistía en un problema de aceleración de un vehículo, en el cual todos los estudiantes acertaron.

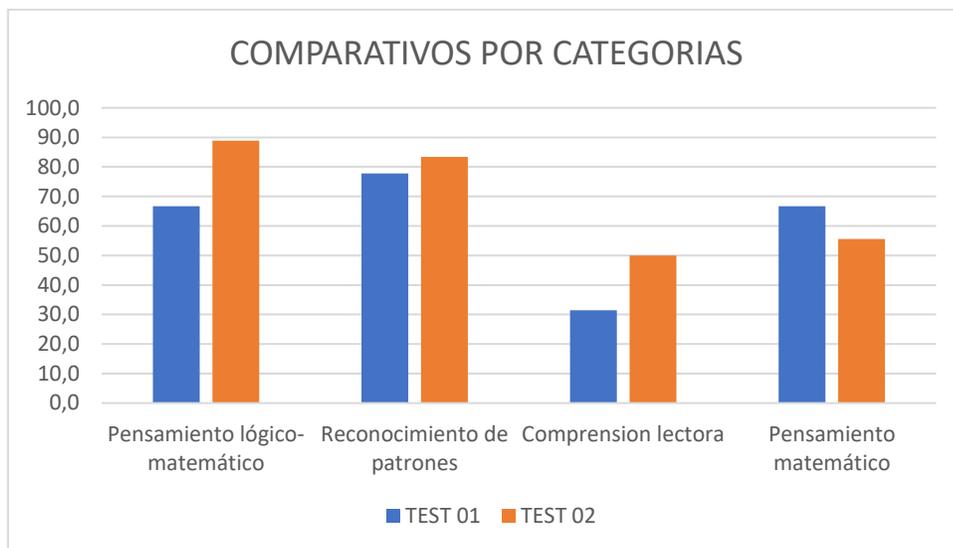
6.1.2. Comparativo de los dos test.

En la gráfica 18 se aprecia el comportamiento de los dos test, los cuales se tuvieron en cuenta solo los estudiantes que concluyeron el curso, los cuales arrojó la siguiente curva de rendimiento.



Gráfica 17: Curva de rendimiento de los test de pensamiento computacional.

Como se puede apreciar, el grupo mantuvo una regularidad, aunque si hubo una pequeña mejoría, lo cual demuestra un impacto positivo, aunque no muy notorio de los test. A continuación, en la gráfica 18 se hará el comparativo por cada una de las categorías evaluadas en cada uno de los test:



Gráfica 18: Comparativo por categorías evaluadas.

Como se puede apreciar, en tres de los 4 aspectos evaluados hubo mejoras en cuanto al test inicial, siendo el pensamiento lógico-matemático el que tuvo un mejor índice (22, 2%), en cuanto al pensamiento matemático, el resultado fue opuesto (-11,1%). A pesar que la comprensión lectora presenta una notoria mejoría (18,6%), sigue siendo un aspecto débil comparado con los demás.

6.2. SEGUIMIENTO A LOS ESTUDIANTES.

Para confirmar que el estudiante ha comenzado a emplear el pensamiento computacional en su vida personal y académica, se le ha hecho un seguimiento con aprobación de los docentes de la materia, para observar en su registro de notas su evolución o involución en este tema a los tres estudiantes que concluyeron satisfactoriamente estos talleres, comparándolos con los resultados obtenidos por sus demás compañeros. También y tomando como base que, de las diez guías implementadas, dos fueron con la temática de programación orientada a objetos, se les hará un seguimiento a sus notas de primer corte de esta materia, así se tendría un dato más exacto de su desempeño.

6.3.1. Seguimiento de la materia de programación estructurada.

En esta materia durante el periodo académico 2019-01 existieron dos grupos, del cual los tres estudiantes que fueron la muestra del proyecto se encontraban en el grupo B. El docente encargado de este grupo aportó el registro de notas. Por confidencialidad, no se darán nombres propios los estudiantes, sino que serán identificados por el orden en el que aparecen en la lista de calificaciones.

Como el curso se comenzó en el segundo corte académico ya los estudiantes venían con un registro de notas previas las cuales se resumen en la tabla 42.

Los códigos 4, 10 y 20 corresponden a los estudiantes que terminaron el curso, los que están en verde, son los estudiantes que comenzaron, pero no lo concluyeron. Como se puede apreciar, fueron estudiantes cuyas notas son superiores a 1.0, lo cual demuestra un interés por aprender conceptos relacionados con la programación y que garantiza su continuidad.

Notas grupo B programación estructurada							
Código	Primer Corte		Segundo Corte		Tercer Corte		TOTAL
	15%	20%	15%	20%	10%	20%	
1	3,0	1,0					0,7
2	4,3	3,8					1,4
3	1,0	2,0					0,6
4	4,8	5,0					1,7
5	0,0	0,0					0,0
6	4,6	4,9					1,7
7	4,2	4,5					1,5
8	3,5	4,5					1,4
9	3,1	2,5					1,0
10	4,1	4,8					1,6
11	1,5	1,0					0,4
12	3,5	3,0					1,1
13	3,1	1,0					0,7
14	2,5	4,8					1,3
15	3,9	4,8					1,5
16	2,7	2,0					0,8
17	2,6	1,8					0,8
18	4,5	5,0					1,7
19	2,9	3,0					1,0
20	4,2	4,9					1,6
21	2,6	4,7					1,3

Tabla 42: Notas del primer corte de los estudiantes del grupo b

Durante el segundo corte, solo se realizaron 4 sesiones, ya que, desde la tercera, se empezó a producir la deserción del grupo, principalmente por argumentos de los participantes que necesitaban tiempo para dedicarle a la materia de cálculo diferencial y la misma programación. En el segundo corte los estudiantes obtuvieron las siguientes notas, observadas en la tabla 43.

Notas grupo B programación estructurada							
Código	Primer Corte		Segundo Corte		Tercer Corte		TOTAL
	15%	20%	15%	20%	10%	20%	
1	3,0	1,0	1,0	1,8			1,2
2	4,3	3,8	3,0	2,3			2,3
3	1,0	2,0	1,8	1,0			1,0
4	4,8	5,0	5,0	4,8			3,4
5	0,0	0,0					0,0
6	4,6	4,9	4,2	3,7			3,0
7	4,2	4,5	3,6	2,4			2,6
8	3,5	4,5	2,8	4,0			2,6
9	3,1	2,5	1,3	0,0			1,2
10	4,1	4,8	3,6	2,9			2,7
11	1,5	1,0	1,0	1,0			0,8
12	3,5	3,0	2,1	2,7			2,0
13	3,1	1,0	1,0	0,0			0,8
14	2,5	4,8	1,0	1,8			1,8
15	3,9	4,8	4,4	3,3			2,9
16	2,7	2,0	4,1	3,4			2,1
17	2,6	1,8	1,0	0,0			0,9
18	4,5	5,0	5,0	4,6			3,3
19	2,9	3,0	2,4	1,9			1,8
20	4,2	4,9	4,9	4,7			3,3
21	2,6	4,7	1,2	0,0			1,5

Tabla 43: Notas de los dos cortes del grupo b de programación estructurada.

Como se puede apreciar de los estudiantes que participaron en el taller de pensamiento computacional, tres de ellos ya aprobaron la materia en segundo corte (ya que la nota mínima para aprobar cualquier materia en la Universidad de Pamplona es 3.0) tres estudiantes necesitan menos de 0.5 en el siguiente corte para aprobar y 1 estudiante tiene una nota de 2.0 lo cual le implica un poco más de esfuerzo para aprobar.

El resto del curso, se concluyó en el tercer corte, este se caracteriza porque tiene el procesos de cancelación de materia, lo cual disminuye el número de estudiantes que terminan la asignatura. La tabla 44 muestra los resultados del tercer corte:

Notas grupo B programación estructurada							
Código	Primer Corte		Segundo Corte		Tercer Corte		TOTAL
	15%	20%	15%	20%	10%	20%	
1	3,0	1,0	1,0	1,8			1,2
2	4,3	3,8	3,0	2,3	3,8	3,3	3,4
3	1,0	2,0	1,8	1,0			1,0
4	4,8	5,0	5,0	4,8	5,0	4,2	4,8
5	0,0	0,0					0,0
6	4,6	4,9	4,2	3,7	4,6	5,0	4,5
7	4,2	4,5	3,6	2,4	5,0	3,0	3,7
8	3,5	4,5	2,8	4,0	4,3	1,8	3,4
9	3,1	2,5	1,3	0,0			1,2
10	4,1	4,8	3,6	2,9	4,5	3,0	3,8
11	1,5	1,0	1,0	1,0			0,8
12	3,5	3,0	2,1	2,7	4,2	2,4	2,9
13	3,1	1,0	1,0	0,0			0,8
14	2,5	4,8	1,0	1,8	3,0	3,5	2,9
15	3,9	4,8	4,4	3,3	4,6	4,2	4,2
16	2,7	2,0	4,1	3,4	3,7	3,0	3,1
17	2,6	1,8	1,0	0,0	0,0	0,0	0,9
18	4,5	5,0	5,0	4,6	5,0	4,5	4,7
19	2,9	3,0	2,4	1,9	4,0	3,0	2,8
20	4,2	4,9	4,9	4,7	4,9	4,2	4,6
21	2,6	4,7	1,2	0,0			1,5

Tabla 44: Notas definitivas de los estudiantes del grupo b.

Como se puede apreciar, de los tres estudiantes que continuaron, dos sacaron notas por encima de 4.5 y el restante saco 3.8, siendo la 6° mejor nota del grupo. De los 4 estudiantes que desertaron un estudiante saco nota de 4.7, dos pasaron la materia con notas de 3.7 y 3.4 respectivamente y uno perdió la asignatura.

El promedio total de grupo en tercer corte es de 3.53, el promedio de los estudiantes que se inscribieron en el curso de pensamiento computacional fue de 3.98, como se puede apreciar que se generó un impacto positivo en los estudiantes que tuvieron algún contacto con el curso. De los siete estudiantes que se inscribieron, el promedio de notas de los que terminaron fue de 4.38 mientras que los estudiantes que desertaron fueron de 3.68, implicando una mejoría notoria en las habilidades de los estudiantes.

6.3.2. Seguimiento en programación orientada a objetos

Para continuar con el seguimiento del proceso del pensamiento computacional, adicional al análisis realizado anteriormente, se consideró seguir con el proceso por lo menos en el primer corte de la materia de programación orientada a objetos, materia la cual se cursa al aprobar programación estructurada. Para el periodo académico 2019-02 se abrieron dos grupos. Los tres estudiantes que terminaron el curso, dos están inscritos en el grupo b y el otro en el grupo a por lo cual fue necesario solicitar las notas de los dos grupos.

Los profesores encargados del área, facilitaron los registros de notas de la materia en la cual están los estudiantes para este análisis. Las notas correspondientes al grupo se muestran en la tabla 45.

Notas Programación orientada a objetos grupo a			
Código	15%	20%	TOTAL
1	4,8	3,4	1,4
2	2,6	2,6	0,9
3	2,4	3,3	1,0
4	0,0	0,0	0,0
5	2,4	3,0	0,9
6	2,3	2,8	0,9
7	3,1	2,6	1,0
8	1,6	1,0	0,0
9	2,7	2,3	0,9
10	2,2	3,0	0,9
11	0,0	1,0	0,2
12	1,8	2,3	0,7
13	4,8	3,3	1,4
14	0,8	3,4	0,8
15	3,6	2,8	1,1

Tabla 45: Notas del primer corte de programación estructurada del grupo a.

En la tabla el registro 12 de color azul, es el estudiante que concluyó el curso de pensamiento computacional. Como se puede apreciar, las notas del grupo estuvieron bajas y el estudiante que realizó el curso tuvo una nota de 0.7. Este estudiante (fue el estudiante 03 del curso) tiene la particularidad, que realizó los dos talleres dirigidos al pensamiento computacional de ese semestre (LEGO y Arduino), lo cual demuestra que algunos estudiantes o no son capaces de asimilar un nuevo paradigma o solo están aprendiendo por el momento.

En cuanto al grupo b, en el cual se encuentran los dos estudiantes restantes que participaron en el curso (estudiante 01 y estudiante 02), las notas del grupo b se reflejan en la tabla 46

Notas Programación orientada a objetos grupo a			
Código	15%	20%	TOTAL
1	4,5	3,9	1,5
2	4,3	3,3	1,3
3	3,9	2,0	1,0
4	3,8	2,5	1,1
5	4,2	4,3	1,5
6	3,9	2,8	1,1
7	4,0	3,0	1,2
8	3,7	2,5	1,1
9	4,4	3,0	1,3
10	3,8	2,7	1,1
11	4,3	3,4	1,3
12	4,5	4,0	1,5

Tabla 46: Notas del primer corte de programación estructurada del grupo b.

Como se puede apreciar, los estudiantes que cursaron el curso, obtuvieron buenas notas, ambos en el promedio acumulado sacaron 1.5. lo cual afirma que los estudiantes asimilaron satisfactoriamente los conceptos manejados principalmente en las guías 07 y 08, y pudieron adaptarse al nuevo paradigma de programación, además que demostraban interés y ganas de aprender conceptos nuevos relacionados con el programa.

7. CONCLUSIONES

- El pensamiento computacional, se está convirtiendo en la actualidad, una revolución a académica en cualquier grado de escolaridad y con esto, el estudiante ve el mundo con otros ojos encontrando soluciones más lógicas a problemas cotidianos y con el auge de la tecnología y la globalización informática, un medio de comprensión de este vasto mundo.
- Durante la realización del curso, una de las principales dificultades observadas radicó en el interés que presentó el estudiante, ya que, al ser un curso libre y no obligatorio, este optaba por abandonarlo o pedir permiso para realizar actividades correspondientes a otras materias, principalmente programación estructurada y cálculo diferencial, las cuales son línea para el estudiante del programa. Esto impidió el cumplimiento del curso en cuanto a su programación, retrasándolo y programando sesiones fuera del horario establecido.
- Mediante la observación por parte del investigador, se evidenció que no todos los estudiantes participantes en el curso de pensamiento computacional aprenden de la misma forma, ya que, unos con solo poca observación e información, empiezan a encontrar interés e investigar por su cuenta aspectos relacionados con el tema, otros que desafortunadamente solo iban por algún interés académico reflejado en sus notas.
- La mayoría de los estudiantes que se inscribieron en curso de pensamiento computacional con arduino, pasaron a la siguiente materia y continuaron en el programa, lo cual prueba que a estos cursos solo asisten personas que les interesa su continuidad en el programa y ampliar sus conocimientos y aptitudes.
- De los tres estudiantes que terminaron el curso, dos estuvieron entre los tres primeros lo cual demuestra la efectividad de desarrollar el pensamiento computacional, complementario a las actividades académicas del semestre, la cual ayuda a la continuidad y mejoramiento de sus estudios académicos dentro del programa.
- Teniendo en cuenta las notas, los test y lo realizado en el curso, se puede llegar a la conclusión que el pensamiento computacional aplicado en los estudiantes influye positivamente en su vida académica, laboral y personal.
- El desarrollo del pensamiento computacional aporta de manera significativa en la retención y la baja mortalidad académica del estudiantado siendo este un componente que debería ser desarrollado en carreras de toda índole.

- Indagando con los estudiantes que concluyeron el curso, se observó que solo 1 estudiante aprobó la materia de cálculo diferencial, y siendo esta la segunda causa de atraso, deserción y mortalidad del programa, y por consiguiente la materia de programación estructurada no es la única culpable que el estudiante decida abandonar el programa de ingeniería de sistemas.

8. RECOMENDACIONES Y TRABAJOS FUTUROS

1. De realizarse un nuevo curso, este debería hacerse como un preuniversitario o un curso de verano, con el fin de garantizar la asistencia de los estudiantes y evitar deserciones masivas.
2. Ampliar los kits de arduino con los que cuenta el programa de Ingeniería de sistemas para poder realizar otras prácticas con un nivel de dificultad mayor y que atrape más al estudiante.
3. Explorar el pensamiento computacional mezclando otros lenguajes y plataformas, como por ejemplo arduino con scratch, lego con arduino, etc.
4. Realizar un trabajo de investigación comparando la enseñanza del pensamiento computacional con Lego Mindstorm y arduino, para determinar las ventajas y desventajas de cada una de estas plataformas.
5. Desarrollar proyectos de extensión social a colegios de grados 11° donde se desarrolle el pensamiento computacional y se motive así a los estudiante para optar por carreras de tecnología.

BIBLIOGRAFIA

- ACEVEDO MORA, Nelson Andrés, Desarrollo del pensamiento computacional mediante Scratch en estudiantes de educación media del municipio de Pamplona, tesis de pregrado, 2018.
- ANGULO, Jesús Alberto Pérez. El pensamiento computacional en la vida cotidiana. *Revista Científic*, 2019, vol. 4, no 13, p. 293-306.
- Anímate con alice, [en línea], [fecha de consulta: 25 de octubre de 2019], disponible en: <https://animateconalice.weebly.com/antecedentes-historicos-de-alice.html>.
- ANONIMO, La revista informática.com [en línea] [Fecha de consulta: 24 de septiembre de 2019] Disponible en <http://www.larevistainformatica.com/Logo.htm>.
- BENITEZ, YOLANDA; MORA, César. Enseñanza tradicional vs aprendizaje activo para alumnos de ingeniería. 2013.
- BERROCOSO, Jesús Valverde; SÁNCHEZ, María Rosa Fernández; ARROYO, María del Carmen Garrido. El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, 2015, no 46.
- Blog aprendiendo arduino, [en línea], [fecha de consulta: 29 de junio de 2019], disponible en: <https://aprendiendoarduino.wordpress.com/2016/12/11/que-es-arduino-2/>.
- Blog aprendiendo arduino, [en línea], [fecha de consulta: 29 de junio de 2019], disponible en: <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>.
- Blog aprendiendo arduino, [en línea], [fecha de consulta: 29 de junio de 2019], disponible en: <https://aprendiendoarduino.wordpress.com/2017/06/19/placas-arduino-2/>.
- Blog aprendiendo arduino, [en línea], [fecha de consulta: 29 de junio de 2019], disponible en: <https://aprendiendoarduino.wordpress.com/category/perifericos/>.

- Blog Historia de la informática, Universidad politécnica de Valencia, [en línea], actualizado: 18 de diciembre de 2013,[fecha de consulta: 26 de octubre de 2019], disponible en: <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>.
- Características de Arduino ONE, [en línea], [fecha de consulta: 29 de junio de 2019], disponible en: <http://www3.gobiernodecanarias.org/medusa/ecoblog/ralvgon/files/2013/05/Caracter%C3%ADsticas-Arduino.pdf>.
- Congreso Hispalinux, [en línea],[fecha de consulta: 28 de octubre de 2019], disponible en: https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200002hispalinux/conf-16/16-html/programacion_LEGO_Mindstorm.html.
- DUEÑAS, Víctor Hugo. El aprendizaje basado en problemas como enfoque pedagógico en la educación en salud. *Colombia médica*, 2001, vol. 32, no 4, p. 189-196.
- El arduino, [en línea], [fecha de consulta: 29 de junio de 2019], disponible en: <https://www.elarduino.com/nano/>.
- Electronica embajadores, [en línea], [fecha de consulta 29 de junio de 2019], disponible en: <https://www.electronicaembajadores.com/es/Productos/Detalle/LCA1016/modulos-electronicos/arduino/arduino-micro-5v-16mhz-original>.
- EUMED, manual de lego mindstorms, [en línea], [fecha de consulta: 26 de octubre del 2019], disponible en: <http://www.eumed.net/libros-gratis/2013/1237/que-es-lego-mindstorms.html>.
- GARCÍA, Arturo Rodríguez; LÓPEZ, Leonardo Ramírez. Aprender haciendo–investigar reflexionando: caso de estudio paralelo en Colombia y Chile. *Revista academia y virtualidad*, 2014, vol. 7, no 2, p. 53-63.
- GAYO, José Emilio Labra, et al. Una experiencia de aprendizaje basado en proyectos utilizando herramientas colaborativas de desarrollo de software libre. *Dpto. de Informática Universidad de Oviedo C/Calvo Sotelo S/N CP*, 2006, vol. 33007.
- Genbeta, [en línea], actualizado: 16 de marzo del 2012, [fecha de consulta, 24 de octubre de 2019], disponible en: <https://www.genbeta.com/herramientas/alice-aprende-programacion-orientada-a-objetos-en-un-entorno-3d-de-forma-divertida>.

- Google sites, [en línea], [fecha de consulta: 25 de octubre de 2019], disponible en: <https://sites.google.com/site/axeltp28/alice>.
- JAUREGUI JAIMES, Alba Beatriz, Desarrollo del pensamiento computacional mediante Scratch utilizando una herramienta e-Learning, tesis de pregrado, 2018.
- João Paulo Cardoso de Lima, Lucas Mellos Carlos, José Pedro Schardosim Simão, Josiel Pereira, Paulo Manoel Mafra, Juarez Bento da Silva, Design and implementation of a remote lab for teaching programming and robotics, Volumen 49, 2016, Pag. 86-91,
- JULIO, Yair Rivera; TURIZO, Luis. ABP (APRENDIZAJE BASADO EN PROBLEMAS) PARA LA ENSEÑANZA Y EL DESARROLLO DE PROYECTOS TECNOLÓGICOS INTERDISCIPLINARES BASADOS EN ARDUINO.
- LIEVANO, Miguel, Aprendizaje Basado en Proyectos con la plataforma Arduino en el desarrollo de las competencias de programación en la Universidad Privada del Norte-2017, Artículo, 2017, pág. 77-83.
- LÓPEZ, José Manuel Sáez; GUTIÉRREZ, Ramón Cózar. Pensamiento computacional y programación visual por bloques en el aula de Primaria. *Educar*, 2017, vol. 53, no 1, p. 129-146.
- LOPEZ, José María, Hipertextual [en línea], Actualizada: 28 de febrero de 2019, [Fecha de consulta: 16 de septiembre de 2019]. Disponible en <https://hipertextual.com/2019/02/logo-tortuga-lenguaje-programacion>.
- LOPEZ, NEREA EGUILUZ. El pensamiento computacional en el aula. *Revista arista digital*, 2016.
- Manual de scratch, [en línea], [fecha de consulta: 28 de junio de 2019], disponible en: <http://lsi.vc.ehu.es/pablogn/docencia/Fdi/Scratch/manual%20scratch.pdf>.
- MARTÍ, José A., et al. Aprendizaje basado en proyectos. *Revista Universidad EAFIT*, 2010, vol. 46, no 158.
- MICROSOFT CORPORATION, download.microsoft.com, manual Microsoft SmallBasic, [en línea], [Fecha de consulta: 24 de Octubre de 2019], disponible en http://download.microsoft.com/download/C/E/6/CE66B602-19F7-4FA6-A4D2-E06F8382A7B9/Introduccion_a_Small_Basic.pdf.

- Mit Scratch, [en línea], [fecha de consulta: 28 de junio de 2019], disponible en: <https://scratch.mit.edu/about>.
- MORALES BUENO, Patricia; LANDA FITZGERALD, Victoria. Aprendizaje basado en problemas. 2004.
- MORENO NIÑO, N.; RICO LUGO, M. J.; BASOGAIN OLABE, X. "Evolución": Diseño e Implementación de Material Educativo Digital para Fortalecer Habilidades del Pensamiento Computacional. 2018.
- NIÑO, Jorge A., et al. Entorno de aprendizaje para la enseñanza de programación en Arduino mediado por una mano robótica didáctica. *Revista Espacios*, 2017, vol. 38, no 60.
- OLABE, Xabier Basogain; BASOGAIN, Miguel Ángel Olabe; BASOGAIN, Juan Carlos Olabe. Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. *Revista de Educación a Distancia*, 2015, no 46.
- Pablo Martín-Ramos, Maria João Lopes, M. Margarida Lima da Silva, Pedro E.B. Gomes, Pedro S. Pereira da Silva, José P.P. Domingues, Manuela Ramos Silva, First exposure to Arduino through peer-coaching: Impact on students' attitudes towards programming, *Computers in Human Behavior*, volumen 76, 2017, Pag. 51-58,
- Página oficial de arduino, [en línea], [fecha de consulta: 29 de junio de 2019], disponible en: <https://www.arduino.cc/en/Main/ArduinoBoardLeonardo>.
- Página oficial de Raspberry pi, [en línea], actualizado: [fecha de consulta: 26 de octubre de 2019]. Disponible en: <https://www.raspberrypi.org/blog/>.
- Revista digital ABC tecnología, [en línea], actualizado: 21 de julio de 2013, [fecha de consulta: 26 de octubre del 2019], disponible en: <https://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html>.
- RODRIGUEA DÁVILA, Alberto, Desarrollo Del Pensamiento Computacional Con LEGO Mindstorm Dirigido A Estudiantes De Grados 10 Y 11, Tesis de pregrado, 2019.
- RODRÍGUEZ, Elías, Wwhat's New [en línea], actualizado: 12 de marzo de 2017,[fecha de consulta: 26 de octubre de 2019], disponible en: <https://www.whatsnew.com/2017/03/12/lego-mindstorms-ev3-historia-detalles-y-precios/>.

- RODRÍGUEZ-SANDOVAL, Eduardo; VARGAS-SOLANO, Édgar Mauricio; LUNA-CORTÉS, Janeth. Evaluación de la estrategia" aprendizaje basado en proyectos". *Educación y educadores*, 2010, vol. 13, no 1, p. 13-25.
- ROMÁN-GONZÁLEZ, Marcos; PÉREZ-GONZÁLEZ, Juan-Carlos; JIMÉNEZ-FERNÁNDEZ, Carmen. Test de pensamiento computacional: diseño y psicometría general. En *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)*. 2015.
- SÁNCHEZ, José. Qué dicen los estudios sobre el Aprendizaje Basado en Proyectos. *Actualidad pedagógica*, 2013.
- SCHANK, Roger C .; BERMAN, Tamara R .; MACPHERSON, Kimberli A. Aprender haciendo. *Teorías y modelos de diseño instruccional: un nuevo paradigma de teoría instruccional*, 1999, vol. 2, no 2, p. 161-181.
- SCHOEPF Corinne, ALONSO Carolina, *Planificación anual del Taller de Informática 2017, Programación y resolución de problemas Scratch para Arduino*, Artículo, Revista de educación Neuquina, 2017.
- Scoliver [en línea], Actualizada: 4 de enero del 2009, [fecha de consulta: 24 de octubre de 2019], disponible en: <https://www.sgoliver.net/blog/microsoft-small-basic/>.
- SILVA-MAURIELLO, Silvana Victoria. *Aprendizaje Basado en Proyectos y Arduino en Tecnología de 4º ESO*. 2018. Tesis de Maestría.
- Teck4kits, [en línea], actualizado: 20 de julio del 2012, [fecha de la consulta: 24 de octubre de 2019], disponible en: <https://tech4kidsmex.wordpress.com/2012/07/20/51/>.
- Tienda oficial de arduino, [en línea], [fecha de consulta: 29 de junio de 2019], disponible en: <https://store.arduino.cc/usa/arduino-yun>.
- Van Arcken, Hernán, La escuela tradicional, Blog pedagogía docente [en línea], [fecha de consulta: 10 de noviembre de 2019], disponible en: <https://pedagogiadocente.wordpress.com/modelos-pedagogicos/la-escuela-tradicional/>
- Varios, Nace la escuela de pensamiento computacional para docentes, Portal educación 3.0 (Madrid, España), Publicado el 25 de octubre del 2018, [Consultado el 11 de septiembre del 2019], disponible en: <https://www.educacionrespuntocero.com/noticias/escuela-de-pensamiento-computacional/93386.html>.

- WING Jeannette M. Computational thinking and thinking about computing366Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences: <http://doi.org/10.1098/rsta.2008.0118>
- Xataka Basic, [en línea], actualizado: 3 de agosto de 2018, [fecha de consulta: 29 de junio de 2019], disponible en: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>.

FUENTE DE LAS IMÁGENES.

Logos:

- <https://brungumacho.gq/photo/603692.jpg>.

SmallBasic:

- https://www.techworld.com/cmsdata/downloads/4104/largeImg_thumb800.png?version=1%2E2.

Alice:

- <https://image.slideserve.com/49961/the-ide-l.jpg>.

Scratch.

- https://miro.medium.com/max/2560/0*356zWDYXZyGbmQxa.

ANEXOS:

<https://drive.google.com/open?id=1TXYhz27KWea101G0iDKzrmVkUTP3bDb7>