

Desarrollo Del Pensamiento Computacional Con LEGO Mindstorm Dirigido A Estudiantes

De Grados 10 Y 11

Autor

Alberto Rodriguez Davila

Proyecto presentado Para Optar Al Título De Ingeniero En Sistemas

Director

Edgar Alexis Albornoz Espinel

Ingeniero De Sistemas



Universidad De Pamplona

Facultad De Ingeniería Y Arquitectura

Ingeniería De Sistemas

Pamplona - Colombia

2018

Dedicatoria

A Dios

A mis padres

Y mi hermano Haider Rodriguez D

Agradecimientos

Agradezco a Dios primeramente, por darme fortaleza para afrontar todas las dificultades en mi vida, a mi familia, especialmente a mis padres y a mí hermano Haider que con tanto sacrificio me dieron la oportunidad de culminar este proceso.

A mi cuñada Inés Pérez Lerma, que ha sido un apoyo importante en mi proceso de formación profesional, además de ser mi psicóloga personal en los momentos difíciles de mi vida. Agradezco a Karen Liceth Martínez que ha sido mi confidente, en la buenas y malas me ha soportado, es la vara que sostiene mi caminar.

Gracias a mi tutor Edgar Albornoz que ha estado apoyándome en este proceso de formación, le estoy agradecido por las asesorías y la formación integral que me ha brindado.

Gracias a todos

Índice general

Desarrollo Del Pensamiento Computacional Con LEGO Mindstorm Dirigido A Estudiantes De Grados 10 Y 11	1
Resumen.....	1
Abstract.....	2
1. DESCRIPCION DEL PROYECTO	4
1.1. Planteamiento del problema.	4
1.2. Justificación.....	5
1.3. Delimitaciones.....	6
1.3.1. Objetivo general.	6
1.3.2. Objetivos específicos.....	6
1.4. Acotaciones.	7
2. MARCO CONCEPTUAL DEL PENSAMIENTO COMPUTACIONAL	8
2.1. Generalidades.	8
2.2. Introducción al Pensamiento Computacional.....	9
2.3. Pensamiento Computacional En La Escuela	11
2.3.1. Lenguajes de programación, plataformas educativas e iniciativas orientadas al pensamiento computacional.....	13
2.3.2. LEGO	21
2.4. Países que han incluido en sus currículos materias relacionadas al pensamiento computacional o las ciencias computacionales.....	29
2.4.1. Reino Unido.....	29
2.4.2. Inglaterra.....	29
2.4.3. Autónoma de Madrid (España).	30
2.4.4. Estados Unidos de América.....	32

3.	TEORÍAS DE APRENDIZAJE	35
3.1.	Teorías Del Aprendizaje En El Contexto Educativo.	35
3.2.	¿Qué es el aprendizaje?	35
3.3.	El Modelo De Pedagogía Tradicional.	36
3.4.	Enfoques más importantes de las teorías de aprendizaje.....	37
3.4.1.	El constructivismo	37
3.4.2.	El conductismo	42
3.4.3.	El cognoscitivismo	44
3.4.4.	El conectivismo	45
4.	ESTADO DEL ARTE	49
4.1.	Pensamiento computacional: Una nueva alfabetización digital	49
4.1.1.	¿Cuál es el problema del pensamiento computacional?	50
4.1.2.	Definición del pensamiento computacional	51
4.1.3.	Un dominio teórico específico del pensamiento computacional en las teorías del aprendizaje y un currículo.....	52
4.2.	Desarrollo del pensamiento computacional con Scratch.....	61
4.2.1.	Diseño de estrategia.....	63
4.2.2.	Resultados.....	64
4.2.3.	Conclusiones.....	64
4.3.	El desarrollo del pensamiento computacional para la resolución de problemas en la enseñanza inicial de la programación	66
4.3.1.	Introducción.....	66
4.3.2.	Líneas de investigación y desarrollo	68
4.3.3.	Resultados obtenidos / esperados	69
4.4.	Enseñando a programar: un camino directo para desarrollar el pensamiento computacional.....	70

4.4.1.	Introducción.....	70
4.4.2.	Oír versus escuchar.....	72
4.4.3.	Ver versus observar	73
4.4.4.	Hacer equivale a practicar	74
4.4.5.	Infundiendo buenos hábitos.....	76
4.4.6.	Conclusiones.....	77
4.5.	Pensamiento computacional a través de la programación: paradigma de aprendizaje.	78
5.	METODOLOGÍA.....	81
5.1.	Tipo de investigación	81
5.2.	Diseño de investigación.....	83
5.3.	Población.....	83
5.4.	Técnicas e instrumentos	84
6.	DISEÑO ESTRATÉGICO	85
6.1.	Generalidades	85
6.2.	Desarrollo de la estrategia	90
6.2.1.	GUIA 1. (Primeros Pasos En LEGO Mindstorm Ev3).....	97
6.2.2.	GUIA 2. (Uso De Operadores Con LEGO Mindstorm Ev3 Y Estructuras De Selección) 99	
6.2.3.	GUIA 3. (Ciclo Repetitivo While Con LEGO Mindstorm Ev3).....	103
6.2.4.	GUIA 4. (Bloques Avanzados En LEGO Home Edition).....	108
6.2.5.	GUIA 5. (Bloques Avanzados En LEGO Home Edition, Uso De Arrays).....	110
6.2.6.	GUIA 6. (Bloques Avanzados En LEGO Home Edition, Uso Del Condicional Switch) 112	
7.	RESULTADOS	116
7.1.	Guía 1. Primeros pasos con LEGO Mindstorm EV3	116
7.1.1.	Sesión #1	116

7.2.	Guía 2. Uso de operadores con LEGO Mindstorm EV3	120
7.2.1.	Sesión #2	120
7.2.2.	Sesión #3	122
7.2.3.	Sesión #4	123
7.3.	Guía 3. Ciclo repetitivo while con LEGO Mindstorm EV3	125
7.3.1.	Sesión #5	126
7.3.2.	Sesión #6	128
7.4.	Guía 4. Bloques Avanzados en LEGO Home Edition	130
7.4.1.	Sesión #7	130
7.5.	Guía 5. Bloques Avanzados en LEGO Home Edition, Uso de Arrays	133
7.5.1.	Sesión #8	133
7.6.	Guía 6. Bloques Avanzados en LEGO Home Edition, Uso del Condicional Switch	136
7.6.1.	Sesión #9	136
7.7.	Diagnostico Test Inicial.....	141
7.8.	Diagnostico Test Final.....	146
7.9.	Seguimiento de estudiantes en curso programación estructurada; Error! Marcador no definido.	
8.	CONCLUSIONES.....	150
9.	RECOMENDACIONES	152
10.	REFERENCIAS	154

Lista de figuras

Figura 1.Lenguaje de programación logo	13
Figura 2.programación en Scratch	17
Figura 3 Lenguaje SmallBasic	18
Figura 4 Programación en Alice	19
Figura 5.Juego hecho en code.org	20
Figura 6. Aspectos generales del Kit Lego Mindstorm Ev3	22
Figura 7.Descripción general de la página de inicio del software LEGO Home Edition	24
Figura 8. Pestañan proyectos y programas	25
Figura 9 Interfaz de programación.....	26
Figura 10.Contenidos de la asignatura (Tecnología, Programación y Robótica)	31
Figura 11.Educación tradicional	36
Figura 12 Método constructivista	38
Figura 13 Rendimiento(Actividades Evaluativas) –Guía #1	119
Figura 14 Rendimiento (Actividades propuestas)-guía 2	124
Figura 15 Rendimiento (Actividades de análisis y retos)-Guía 2	125
Figura 16 Rendimiento (Actividades Evaluativas)-Guía 3	129
Figura 17 Rendimiento (Actividades Evaluativas)-Guía 5	132
Figura 18 Rendimiento (Actividades Evaluativas)-Guía 5	135
Figura 19 Rendimiento (Actividades Evaluativas)-Guia6	138
Figura 20 Resultados del test-inicial.....	143
Figura 21. Resultados del test final.....	147
Figura 22 Cotejamiento de los test aplicados	149

Listas de tablas

Tabla 1 Cuadro comparativo de los artículos más relevantes del estado del arte.....	79
Tabla 2 Descripción de los recursos del curso pensamiento computacional con LEGO Mindstorm	95
Tabla 3 Resultados del test inicial	142
Tabla 4 Resultados Test Final.....	146

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

Resumen

Los países con alto índice de desarrollo han visto una decadencia creciente en los profesionales que laboran en el ámbito computacional, las empresas y organizaciones se quedan sin cubrir puestos como desarrollador y digitador, esto lleva a los colegios y universidades a plantearse una posible solución al problema, algunas de estas instituciones optan por una reorganización curricular, esto no siempre es la solución más adecuada ya que depende de muchos factores, uno de ellos es el interés de la población en estudiar carreras relacionadas a la informática. El programa de ingeniería de sistemas no es inherente a esta situación, el índice de personas que desertan de la carrera es alto, una de las principales razones son dificultades en materias como programación lo que lleva al estudiante a tomar decisiones como cambiarse de carrera o abandonar sus estudios, estas falencias vienen desde la etapa escolástica, carecen de un pensamiento computacional generando dificultades en áreas como ciencias, matemáticas y lógica. Los colegios no incluyen en sus currículos asignaturas relacionadas a la enseñanza del pensamiento computacional.

El objetivo principal de este trabajo es el desarrollo del pensamiento computacional con LEGO Mindstorm Ev3, en estudiantes de primer semestre de ingeniería de sistemas de la Universidad de Pamplona, tomando una muestra poblacional de seis estudiantes, se procede a la aplicación de una prueba piloto, que consiste en la realización de un curso para el desarrollo del pensamiento computacional, donde con técnicas de recolección de datos, se sabrá el desempeño del estudiante al inicio del curso y al finalizar el proceso, esto para saber si ha tenido mejoras en

las competencias del pensamiento computacional. Este curso fue diseñado siguiendo la teoría constructivista, donde se le enseña al estudiante a organizar, construir e implementar soluciones a problemas, enriqueciendo su conocimiento y desarrollando habilidades referentes al pensamiento computacional. Esta teoría se fundamenta en la construcción de soluciones a situaciones o problemas a través de un proceso mental que implica la adquisición de conocimiento nuevo, así como la contingencia de edificarlo y adquirir las capacidades que le permitirá al estudiante aplicar dicho conocimiento en la solución del problema. (Berardinelli, Padrino, Ramírez & Páez, 2010).

Abstract

Countries with high development rates have seen a growing decline in professionals working in the computing field, companies and organizations are left without filling positions as a developer and digitizer, this leads colleges and universities to consider a possible solution to the problem, some of these institutions opt for a curricular reorganization, this is not always the most appropriate solution since it depends on many factors, one of them is the interest of the population in studying careers related to computer science. The systems engineering program is not inherent in this situation, the rate of people dropping out of the career is high, one of the main reasons are difficulties in matters such as programming which leads the student to make decisions such as changing careers or dropping out. their studies, these shortcomings come from the scholastic stage, lack a computational thinking generating difficulties in areas such as science, mathematics and logic. The schools do not include in their curricula subjects related to the teaching of computational thinking.

The main objective of this work is the development of computational thinking with LEGO Mindstorm Ev3, in first semester systems engineering students at the University of Pamplona,

taking a population sample of six students, we proceed to the application of a pilot test, which consists in the realization of a course for the development of computational thinking, where with data collection techniques, the student's performance will be known at the beginning of the course and at the end of the process, this to know if he has had improvements in the skills of the computational thinking This course was designed following the constructivist theory, where the student is taught to organize, build and implement solutions to problems, enriching their knowledge and developing skills related to computational thinking. This theory is based on the construction of solutions to situations or problems through a mental process that involves the acquisition of new knowledge, as well as the contingency of building it and acquiring the skills that will allow the student to apply said knowledge in solving the problem.

1. DESCRIPCION DEL PROYECTO

1.1. Planteamiento del problema.

Las instituciones educativas de primaria y secundaria latinoamericanas no tienen contemplado implementar en sus currículos asignaturas relacionadas a la enseñanza de la computación (ciencias computacionales, pensamiento computacional y programación).

Los estudiantes de primaria y secundaria, carecen de un pensamiento computacional generando unas falencias en áreas como ciencias, matemáticas y lógica, generando dificultades para aprobar los diferentes cursos que tienen durante su vida escolar. (Álvarez, 2017)

La deserción de carreras con fines tecnológicos, se presenta debido a que los estudiantes en los primeros años de vida escolar no reciben formación que les permita adquirir habilidades encaminadas al pensamiento computacional.

1.2. Justificación

Países como Inglaterra, Estado Unidos, Finlandia, Estonia, Japón y Singapur se han enfocado en el desarrollo del pensamiento computacional en la edad escolar, estos países lideran la revolución tecnológica.

El desarrollo del pensamiento computacional desde la edad escolar, es de gran beneficio para el estudiante ya que este refuerza los conocimientos básicos de los cursos escolares en materias como matemáticas, ciencias y lógica. Usando el concepto del constructivismo con las herramientas de acompañamiento necesarias que genera una dinámica de aprendizaje.

El desarrollo del pensamiento computacional en tempranas edades logra que los estudiantes adquieran nociones básicas de análisis y lógica orientadas a la programación, incrementando de forma significativa sus competencias en resolución de problemas, pensamiento crítico y desarrolla la abstracción, entre otros.

La incorporación del pensamiento computacional al ámbito escolar, busca que los estudiantes de las instituciones educativas no sean simples usuarios de las tecnologías actuales, sino que estos desarrollen habilidades potenciales como el razonamiento lógico-matemático, la capacidad cognitiva y la creatividad que le serán de gran utilidad en el desarrollo y solución de los problemas que se les presentan en su vida cotidiana, cabe señalar que este proyecto tiene un componente lúdico de aprendizaje que ayuda al proceso de enseñanza, es una metodología de aprendizaje diferente a la tradicional donde el estudiante es autónomo en su aprendizaje y el tutor esta para asesorar y solucionar inquietudes.

1.3. Delimitaciones.

1.3.1. 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.

1.3.2. Objetivos específicos.

- Proponer un procedimiento de aprendizaje para el desarrollo del pensamiento computacional con LEGO Mindstorm.
- Diseñar una serie de guías prácticas orientadas al desarrollo del pensamiento computacional utilizando la herramienta LEGO Mindstorm.
- Diseñar y aplicar una prueba piloto en estudiantes de grados 10 u 11 con las guías de aprendizaje, para recopilar información y analizar el desarrollo del pensamiento computacional, apoyados con la herramienta LEGO Mindstorm

1.4. Acotaciones.

La prueba piloto será aplicada solo a estudiantes de grado 10 o 11 seleccionando un pequeño grupo de máximo 6 estudiantes, se realizaran 6 guías de pensamiento computacional apoyadas con la herramienta LEGO Mindstorm donde se abarcaran los conceptos más importantes, las acotaciones establecidas buscan un mejor seguimiento al proceso de aprendizaje, analizando el desempeño de los estudiantes en el desarrollo de las correspondientes guías.

La anterior acotación fue diseñada el 2 de septiembre de 2018 donde se tenía pensado aplicar el desarrollo del proyecto a la población de estudiantes de grado 10 o 11, pero realizando un análisis más profundo y real de los objetivos que busca este proyecto se vio la oportunidad de contactar a estudiantes matriculados para el programa de ingeniería de sistemas en el primer semestre del 2019 y que estos contaban con un periodo de vacaciones, en el cual se podría iniciar la aplicación del curso, brindando la posibilidad de abordar estudiantes recién graduados de bachiller y que realmente acceden al programa, para poder así medir como afecta el desarrollo del pensamiento computacional en la retención y la deserción al interior del programa.

2. MARCO CONCEPTUAL DEL PENSAMIENTO COMPUTACIONAL

2.1. Generalidades.

El referente teórico que se desarrolla a continuación, presenta la conceptualización del pensamiento computacional el cual sustenta el estudio de la enseñanza y el aprendizaje de los conceptos computacionales esenciales para que un estudiante desarrolle las habilidades de formulación y solución de problemas.

Se abordará el pensamiento computacional a través de herramientas robóticas, donde los estudiantes encuentran la solución a un problema.

Como menciona Starger, (2003) “Los materiales LEGO ofrecen posibilidades de improvisación que permiten incluso a niños pequeños construir una máquina, someter a prueba una hipótesis, cacharrear, corregir errores y superar sus expectativas”

2.2. Introducción al Pensamiento Computacional

En el año 2006 la doctora Jeannette M. Wing, profesora del Departamento de Computación de la Universidad de Carnegie Mellon, utiliza el termino Pensamiento Computacional en un artículo, donde describe cómo piensa una persona relacionada al área de la informática y los beneficios de esta manera de pensar. El pensamiento computacional según (Wing, 2006) se define, como el proceso de pensamiento envuelto en formular un problema y sus soluciones de manera que las soluciones son representadas de forma que pueden ser llevadas a un agente de procesamiento de información.

Un agente de procesamiento de información, se refiere a los diferentes métodos que se utilizan para resolver un problema particular llegando a una solución que puede ser llevada a cabo por un humano, por una maquina o conjuntamente.

Los intelectuales que destacaron en el campo del pensamiento computacional incluyen a Papert, Wing y Wólfram. Papert narra el valor de emplear primitivas cognitivas humanas a problemas orientados a objetos, poniendo como ejemplo la relación de los componentes en un sistema complejo, que es el proceso de dividir una tarea compleja en un conjunto de tareas más simples. (Xabier Basogain et al., sf)

Otras definiciones del pensamiento computacional que han surgido de la literatura científica son las de Aho y la de la Royal Society.

El pensamiento computacional es el proceso que permite formular problemas de forma que sus soluciones pueden ser representadas como secuencias de instrucciones y algoritmos. El pensamiento computacional es el proceso de reconocimiento de aspectos de la informática en el mundo que nos rodea, y aplicar herramientas y técnicas de la informática para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales. (Moreno, 2014)

El apartado expuesto por vilanova (2018) informa sobre las iniciativas que están siendo promovidas para desarrollar una definición clara, precisa, eficaz acerca del pensamiento computacional. Esta definición tiene como objeto ser un referente, un vocabulario para el uso académico, así otorgarle un significado global a este tema, esta iniciativa está siendo apoyada por la Sociedad Internacional de la Tecnología en la Educación (ISTE) y la Asociación de Profesores de Informática (CSTA).

El pensamiento computacional tiene unas características importantes en el proceso de solución de un problema, aunque no se limita:

- Permite el uso de computadoras y otros instrumentos para solucionar un problema formulado.
- Organiza y analiza datos de manera lógica
- Representa datos mediante abstracciones, como modelos y simulaciones.
- Automatiza soluciones mediante pensamiento algorítmico.
- Identifica, analiza e implementa posibles soluciones con el objeto de encontrar la combinación de pasos y recursos más eficiente y efectiva.

Es un error pensar que el Pensamiento Computacional es asunto exclusivo para personas del ámbito de la ingeniería informática y la computación, debido a esto hay un gran interés y esfuerzo creciente en incorporar el Pensamiento Computacional a través de proyectos, juegos, entornos de programación en el currículo de escuelas y universidades (Basogain X, Olabe M.Á, Olabe J.C, 2015). Las personas que desarrollen el pensamiento computacional estarán en disposición de resolver problemas complejos, en la capacidad de describir sistemáticamente un problema en varias capas de abstracción y de describir la interface entre dichas capas, además de encontrar una solución buena y eficiente a los problemas que surjan.

2.3. Pensamiento Computacional En La Escuela

La enseñanza y el aprendizaje en la actualidad no son diferente a aquella utilizada en el pasado. Sin embargo, los alumnos que asisten a los cursos escolares o universitarios forman parte de una nueva generación, acostumbrada al uso de las tecnologías y el manejo de la TIC de una manera innata, su aprovechamiento modifica la forma en la cual se relacionan con la sociedad (Giunti, 2010). Los organismos gubernamentales, agencias competentes, expertos, autores de informes de tendencias, se han sorprendido ya que la sociedad y los sistemas de producción, servicios y consumo, solicitan profesionales calificados en la industria de la información, países y regiones desarrolladas, se quedan sin cubrir puestos de trabajos de ingenieros de software, desarrollador, documentalistas digitales por falta de personal capacitado, esta situación ha llevado al sistema educativo de los países , a abordar el problema desde la reorganización curricular, se ha visto que se trata de una nueva alfabetización, una alfabetización digital , por consiguiente hay que comenzar desde las primeras etapas del desarrollo individual, como sucede con otras habilidades como lo es la lectura, la escritura y las habilidades matemáticas. Es de gran

importancia cotejar la nueva alfabetización con las competencias tradicionales, la respuesta habitual, favorece la enseñanza de la programación y de sus lenguajes de forma gradual, consistiría en plantear a los niños tareas de programación desde las primeras etapas, y a medida que prosperan en el aprendizaje, aumentar el nivel de dificultad, teniendo en cuenta el carácter motivador y lúdico desde la más sencilla hasta la más compleja de las tareas (Zapata, 2015).

Programar computadoras demanda en los estudiantes encontrar diferentes maneras de abordar problemas y de trazar soluciones. Conjuntamente, desarrolla habilidades para visualizar rutas de razonamiento divergentes, anticipar errores y evaluar rápidamente los diferentes escenarios mentales (Stager, 2003). Encontrar una metodología que explore la capacidad de aprendizaje en el estudiante es de gran importancia, para que así el tutor pueda determinar la mejor forma de explotar las destrezas del estudiante en el área de la programación, esto se debe a que no todas las personas aprenden del mismo modo.

Concluyendo, el desarrollo de habilidades en el estudiante deben ir más allá del razonamiento o codificación de un programa, involucra entender un problema (abstraer, modelar, analizar), buscar una solución efectiva (reflexión sobre la abstracción, definir estrategias, seguir un proceso, aplicar una metodología, descomponer en problema más simple), usar un lenguaje para plantear la solución, usar herramientas que compilen esos lenguajes, ejecutar la solución, explicar las decisiones tomadas, entre otras. En contexto, la resolución de problemas supone el desarrollo de habilidades genéricas que forman parte del llamado pensamiento computacional. (Wing, 2006)

2.3.1. Lenguajes de programación, plataformas educativas e iniciativas orientadas al pensamiento computacional

La programación promueve la elaboración de conocimientos por medio de su apropiación, de modo que los resultados de los ejercicios cognitivos son considerados como propios, personales, conjuntamente los objetos y los artefactos (programas, videojuegos, robots) son de gran importancia en estos procesos, ya que materializan el pensamiento y por consiguiente, lo modifican y enriquecen (Valverde, Fernández y Garrido, 2015).

2.3.1.1. Lenguaje logo

El lenguaje logo fue creado en 1967, es un lenguaje derivado del lenguaje Lisp, el objeto de logo es educacional, se considera una herramienta para la enseñanza, aprendizaje y el pensamiento, es usado mayormente en la exploración de las matemáticas. Desarrollado por un equipo del MIT en la década de los 80 se presenta LEGO LOGO un juguete de construcción que permite a los niños programar en la computadora algoritmos que pueden ser ejecutados en el robot. (Morales, s.f.).



Figura 1. Lenguaje de programación logo

El propósito de logo en los estudiantes es:

- El análisis y descomposición de problemas en partes simples
- La solución total de un problema a partir de pequeñas soluciones.
- Corregir errores, reestructurando los métodos utilizados.
- Desarrollo de competencias comunicativas y afectivas.
- Desarrollo de la autonomía, al construir su propio conocimiento.

2.3.1.2. *The Intel Computer Clubhouse Network.*

El Computer Clubhouse es un entorno de aprendizaje fuera de la escuela, fue creado en 1993 por el Museum of Science en Boston, entrena a sus usuarios en programación, brindado apoyo para personas con escasos recursos, los miembros tienen acceso a tecnologías de video, gráficos, diseño web y recursos para manufactura musical, con la tutoría continua de personal capacitado y voluntarios. (The Computer Clubhouse Network, 2008).

Principios de clubhouse:

- Motiva a los miembros a ser inventores, creadores, diseñadores, a través de actividades.
- Los miembros trabajan en proyectos de su escogencia y acordes a sus propios intereses.
- Trabajo en colaboración, donde todos con apoyo de sus mentores logran objetivos.
- La clubhouse está abierta a personas de escasos recursos, en un ambiente de respeto y colaboración.

El entorno de clubhouse promueve los siguientes resultados:

- Permite alcanzar objetivos a través de la tecnología.
- Desarrolla la comunicación y el trabajar en equipo.
- Desarrolla habilidades en la solución de problemas complejos.
- Desarrolla la planeación y la ejecución de proyectos complejos.
- Mejora de la autoestima y la confianza.

2.3.1.3. Scratch

Scratch es un entorno de programación desarrollado por un grupo de investigadores del Instituto de Tecnología de Massachusetts (MIT) aprovecha los avances tecnológicos en el área de diseño gráfico y presenta la programación de manera atractiva y accesible para todo aquel que se enfrente por primera vez a programar. Atendiendo las exigencias, competencias para el siglo XXI definidas por la OCDE y la UNESCO, el uso de Scratch desarrolla en los estudiantes las siguientes competencias (Travieso, s.f.).

- Pensamiento creativo: Scratch favorece la curiosidad intelectual y el pensamiento creativo
- Razonamiento abstracto: como resultado de las clases, el alumno desarrolla el razonamiento divergente, interpreta errores y evalúa escenarios mentales.

- **Pensamiento computacional:** Desarrollan el pensamiento crítico y computacional. Para construir los proyectos, los alumnos necesitan coordinar el tiempo y las interacciones entre los diferentes objetos programables.
- **Resolución de problemas:** Ayuda la identificación de problemas y su solución en un contexto significativo, innovador y atractivo.
- **Aprendizaje autónomo:** Tener una idea y pensar en cómo programarla requiere persistencia y práctica, los desafíos en el proceso de diseño y resolución del problema forman al estudiante.
- **Trabajo colaborativo:** Debido a que los programas de Scratch son construidos con bloques gráficos, el código programado es más fácil de leer y de compartir.
- **Comunicación:** Una comunicación efectiva requiere, hoy en día, algo más que saber leer y escribir textos. Scratch permite a los jóvenes la manipulación e integración de distintos elementos multimedia para expresar su creatividad y poner en práctica su capacidad de persuasión.
- **Manejo de TIC:** Es habitual referirse a las personas jóvenes como nativos digitales debido a su aparente fluidez con las TIC. Pero, aunque interactúan con los medios digitales todo el tiempo, pocos son capaces de crear sus propios juegos, animaciones o simulaciones.
- **Life Long Learning (Aprendizaje a largo plazo):** Con el uso de Scratch, el alumno es quien dirige su aprendizaje. De esa forma, se desarrolla la competencia de aprender a aprender y los estudiantes, al conseguir sus objetivos se percatan de que realmente son capaces de aumentar sus conocimientos por sí mismos, sin depender de instrucción externa.

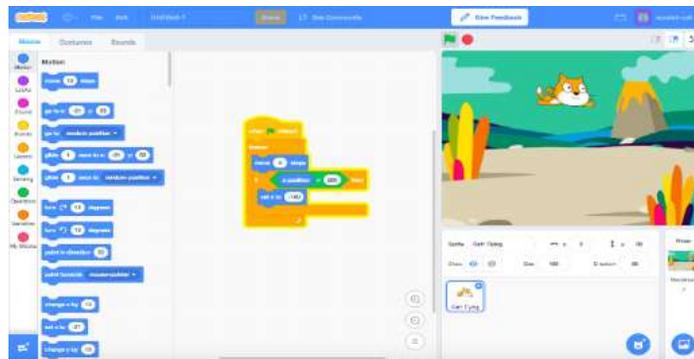


Figura 2. programación en Scratch

2.3.1.4. SmallBasic

SmallBasic es un lenguaje diseñado para ayudar a los estudiantes a pasar de la codificación basada en bloques a la codificación basada en texto, al presentar los elementos fundamentales con una sintaxis, Small Basic les ofrece a los estudiantes las destrezas y la confianza para abordar lenguajes de programación más complejos como Java y C #. En SmallBasic se pueden crear aplicaciones para Kinect, LEGO Mindstorm, Raspberry Pi, Arduino, Oculus Rift (Microsoft, s.f.). Smallbasic es un lenguaje para la enseñanza de la programación en niños, cuenta con funciones de dibujo bidimensionales, además tener funciones de reproducción de sonidos y la realización de cálculos aritméticos, esta herramienta está desarrollada para que el estudiante se motive por la programación, su diseño cuenta con transparencias y grandes botones. Al escribir código en SmallBasic, esta cuenta con un gestor de sugerencias para funciones y variables declaradas, la sencillez de esta herramienta no impide hacer cosas extraordinarias como, formas geométricas que son generadas a través de elementos o formas simples.

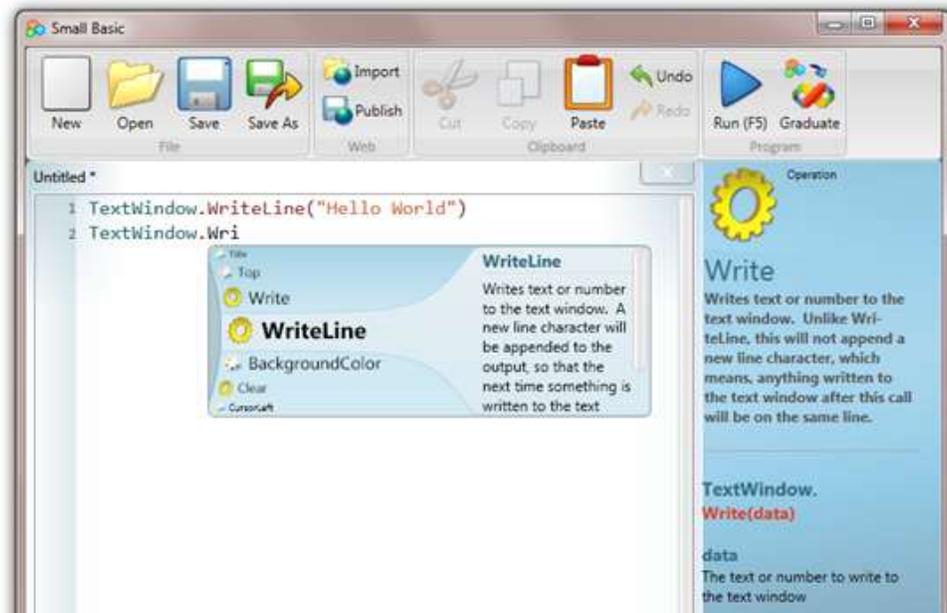


Figura 3 Lenguaje SmallBasic

2.3.1.5. Alice.

Alice es un entorno de programación basado en bloques, es innovador y facilita la creación de animaciones, narrativas interactivas y la programación de juegos 3D. A diferencia de muchas aplicaciones de codificación fundamentadas en rompecabezas, motiva el aprendizaje a través de la exploración creativa. Con Alice se desarrolla el pensamiento lógico, computacional y los principios fundamentales de la programación, además de proporcionar los primeros pasos a los estudiantes hacia la programación orientada a objetos (Alice, s.f.).

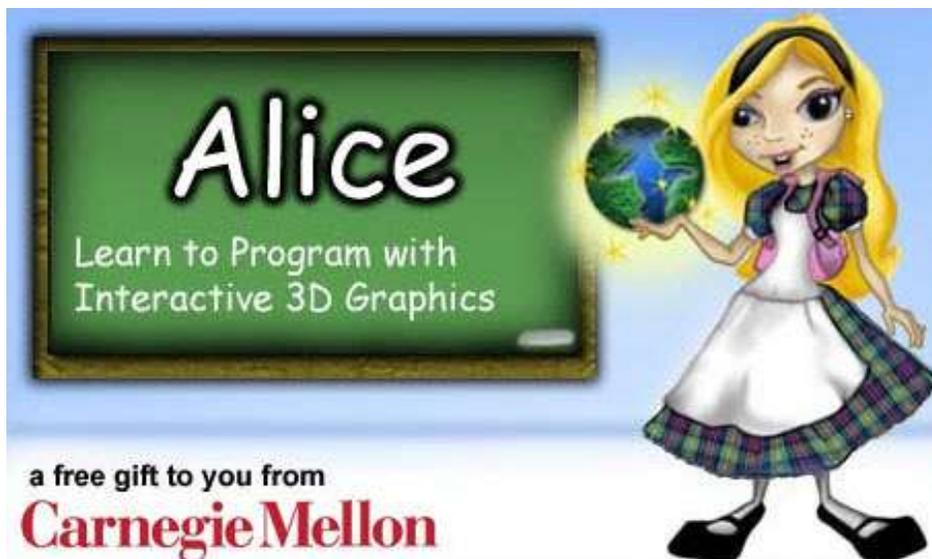


Figura 4 Programación en Alice

2.3.1.6. Code.org.

Tiene como objetivo, difundir la programación como parte de la educación básica de los jóvenes, es una organización sin ánimo de lucro que surge porque en Estados Unidos hay una gran demanda de desarrolladores, según datos de la organización, es tan grande la demanda que se calcula que para 2020 habrá 1,4 millones de empleos afines con la informática frente a tan solo 400.000 estudiantes. En la web están las sesiones educativas donde se pueden probar técnicas de aprendizaje de la programación y practicar directamente desde el propio navegador, el sitio recomiendan academias y cursos, software educativo y formas de aprender mediante juegos y sistemas tales como los robots de LEGO o juegos para teléfonos móviles (Marin, 2013).



Figura 5. Juego hecho en code.org

2.3.1.7. *Scratched.*

Es una comunidad en línea donde los educadores de Scratch intercambian recursos, realizan preguntas relacionadas al entorno Scratch, encuentran personas y descubren eventos (Scratched, s.f.).

El sitio web posee muchos recursos para educadores de todo el mundo que enseñan Scratch. Los recursos incluyen:

- Videos con lecciones del entorno.
- Artículos explicando cómo configurar un Scratch Workshop
- Historias de otros miembros que describen sus experiencias y dan consejos.
- Ejemplos de proyectos Scratch para ayudar a los educadores.

2.3.2.LEGO

2.3.2.1. *Introducción*

La robótica se ha transformado en una herramienta educativa donde los estudiantes de distintos curso educativos, practican los conocimientos recibidos en clase. Promoviendo el desarrollo de nuevo conocimiento en las áreas como el diseño, la construcción y la programación de sistemas robóticos.

La robótica educativa ayuda a relacionar al estudiante con las tecnologías de la información y comunicaciones (TIC), estas tienen influencia en casi todos los aspectos de la vida moderna. En el área de la robótica educativa, existen diversas empresas dedicadas a la fabricación de juguetes, kits educativos y software. Entre ellas, se encuentra la marca LEGO, reconocida por ser una de las principales empresas de juguetes a nivel internacional, se ha caracterizado por crear juguetes que ayudan a la imaginación, creatividad y aprendizaje de sus usuarios. (Cacique Borrego & Camarillo Gómez, 2013)

2.3.2.2. *LEGO Mindstorm EV3*

Es el sucesor de Lego Mindstorm NXT, el kit de Lego EV3 recomendado para enseñar y aprender robótica, permite a los estudiantes construir, programar y testear soluciones a problemas. Este kit posee un bloque EV3 o brick inteligente, se trata de una pequeña y poderosa computadora que permite el control de los motores y captar la información de los sensores. Este bloque admite la comunicación con el ordenador por cable USB, Bluetooth o WIFI. De esta forma

desde el ordenador se puede crear y ejecutar un programa para el control de un robot. (Posada Prieto, 2019)

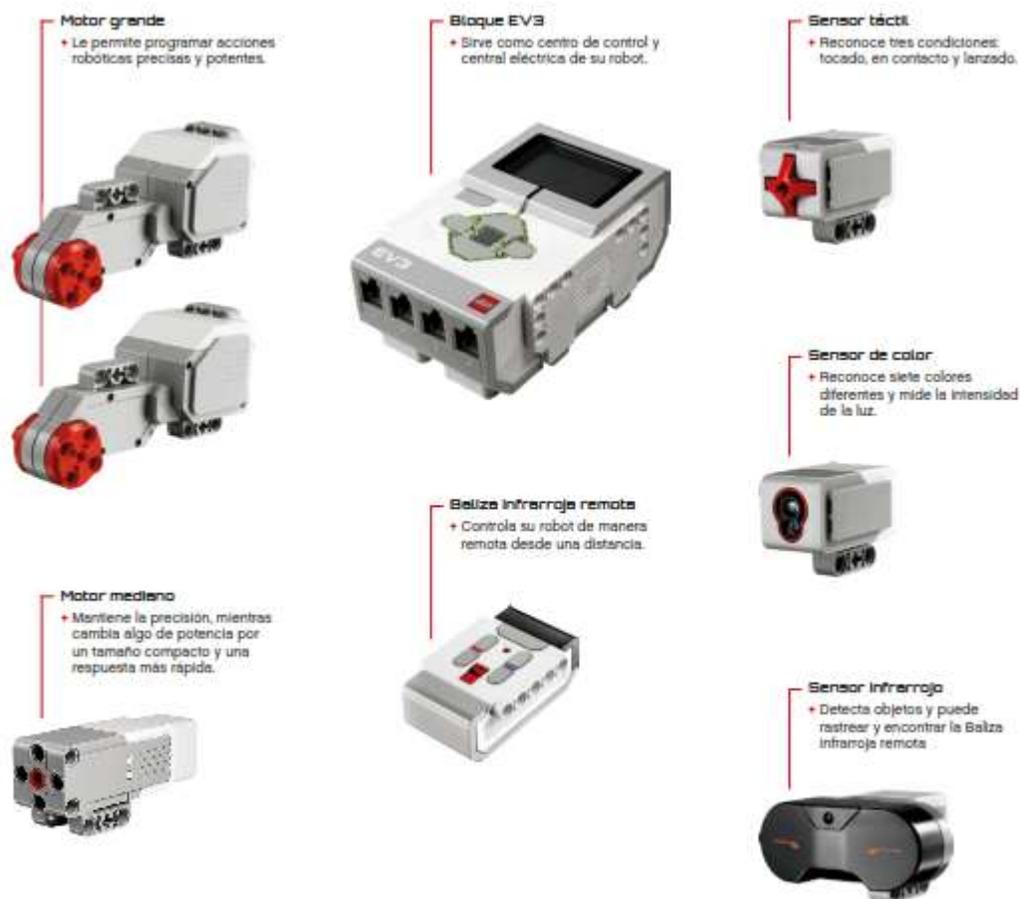


Figura 6. Aspectos generales del Kit Lego Mindstorm Ev3

Como menciona Posada Prieto (2019), el kit LEGO Mindstorm viene en una caja de plástico, donde se almacenan las piezas de construcción por seguridad, cuenta con una bandeja para guardar piezas y clasificarla para su empleo en el aula de clase. Además de las piezas de construcción el kit contiene:

- Bloque o brick inteligente
- Dos servo motores, llamados motores grandes
- Un motor normal llamado motor mediano
- Sensor ultrasónico para detectar distancias y presencia
- Sensor de rotación
- Sensor de color y luminosidad
- Dos sensores táctiles
- Cables de conexión
- Batería de ionLitio recargable
- Ruedas y rueda-bola
- Manual de instrucciones de construcción del robot educador

2.3.2.2.1. LEGO Home Edition

Además de los diferentes componentes que posee el kit LEGO, cuenta con un software llamado LEGO Home Edition, está disponible gratuitamente y es fácil de usar, ya que cuenta con divertidas misiones y una interfaz de programación que te permite experimentar la magia de conseguir que tu robot haga lo que tú quieras("Mindstorm LEGO", 2015)

2.3.2.2.2. Página de inicio

Como menciona Mindstorm LEGO (2015), al abrir el Software de EV3, ingresará automáticamente a la página principal, está diseñada para facilitar la localización de las

herramientas y el trabajo con el software, permite acceder a todo lo necesario, a opciones y recursos como:

- 1) Pestaña Página de inicio: este botón siempre lo hace regresar a la Página de inicio.
- 2) Agregar proyecto: aquí puede agregar un proyecto nuevo para comenzar a programar su propio robot.
- 3) Misiones del Robot: aquí puede comenzar a construir y programar los modelos animados principales.



Figura 7. Descripción general de la página de inicio del software LEGO Home Edition

2.3.2.2.3. *Propiedades y estructura del proyecto*

Al adicionar un programa, este creara una carpeta y dentro de ella un archivo sin un nombre asignado. Todo lo realizado en la carpeta, las imágenes, los sonidos, los videos, las instrucciones y otros recursos que se utilizaron se almacenarán automáticamente en la carpeta del proyecto. Esto facilita el almacenamiento de su proyecto y su uso compartido con otras personas. Cada proyecto aparecerá en la parte superior de la pantalla, vera las pestañas de los programas que está elaborando. Para agregar un nuevo proyecto o programa, haga clic en el botón + que se encuentra a la derecha de las otras pestañas. Para cerrar la pestaña, haga clic en una X. ("Mindstorm LEGO", 2015)

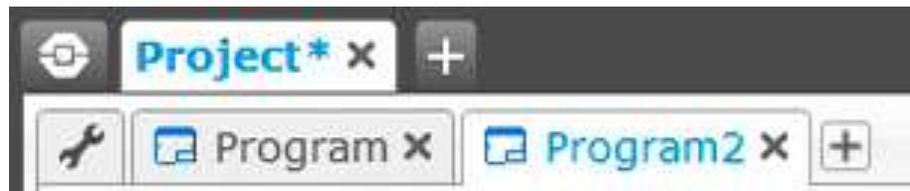


Figura 8. Pestañas proyectos y programas

2.3.2.2.4. *Programación*

Programa su robot en la interfaz de desarrollo intuitiva, arrastre y suelte las acciones que desee en la ventana de programación y ajústelas para que se adapten al comportamiento de su robot ("Mindstorm LEGO", 2015). La interfaz de programación de EV3 consta de las siguientes áreas principales:

- 1) Área de documento de programación: diseñe su programa aquí.
- 2) Paletas de programación: Busque los bloques de construcción para su programa aquí.
- 3) Página de Hardware: Establezca y administre su comunicación con el Bloque EV3 aquí y vea qué sensores y motores están conectados y dónde están conectados. Aquí también descarga los programas al Bloque EV3.
- 4) Editor de contenidos: Un cuaderno de ejercicios digitales integrado al software. Documente su proyecto mediante texto, imágenes y videos.
- 5) Barra de herramientas de programación: Busque aquí las herramientas básicas para trabajar con su programa.

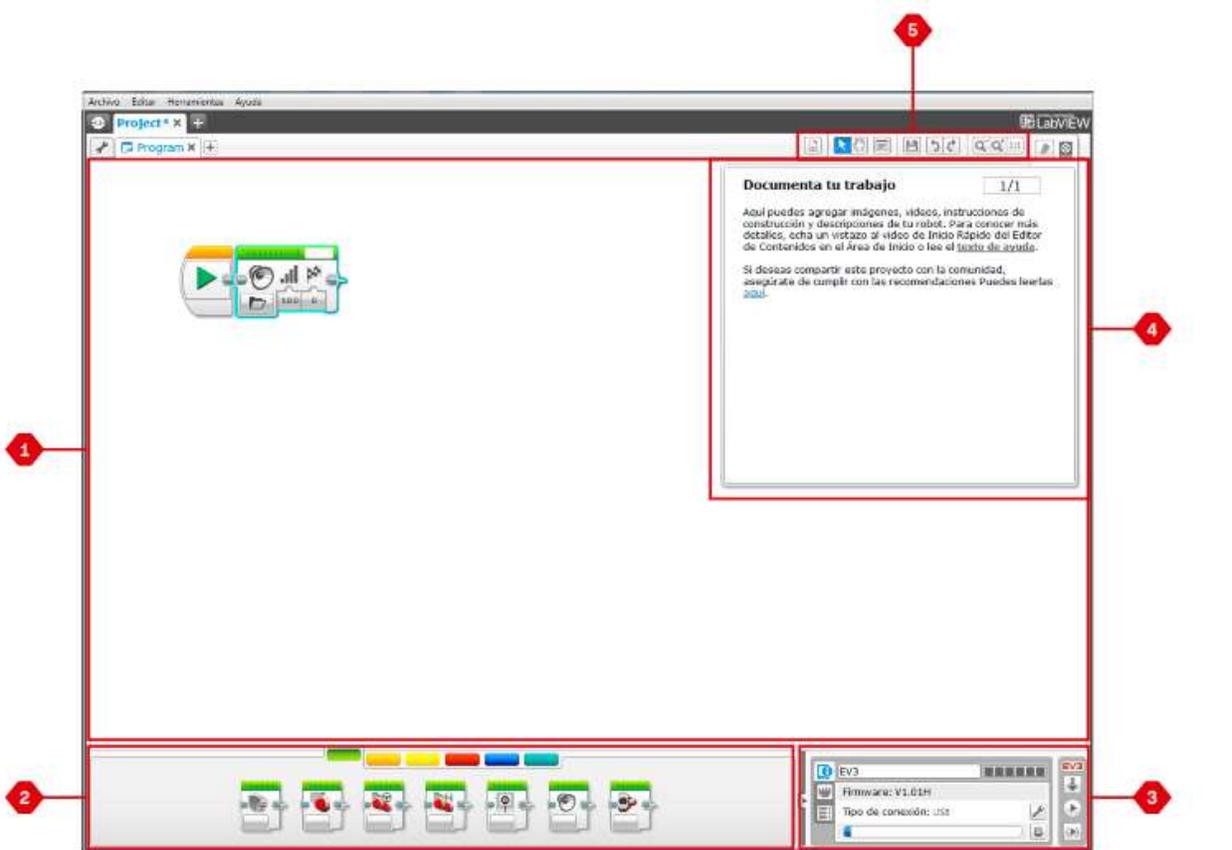


Figura 9 Interfaz de programación

2.3.2.2.5. Paletas y bloques de programación

Todos los bloques utilizados para programar un robot se encuentran en la paleta de programación ubicada en la parte inferior de la Interfaz del software LEGO home edition. Los bloques se dividen en categorías, según su tipo y naturaleza, facilitando su búsqueda, al momento de necesitarlos.

2.3.2.3. Software De Simulación Para Lego Mindstorm

2.3.2.3.1. LEGO Digital Designer (LDD)

Es un programa gratuito diseñado por The LEGO Group, donde se puede diseñar virtualmente cualquier construcción con piezas LEGO en 3D, cuenta con muchas piezas de todo tipo y se encuentra disponibles en diferentes colores.

El software LDD ofrece tres posibilidades para iniciar la construcción de un nuevo modelo, el primero es LEGO Digital Designer, que permite construir modelos de una gran selección de ladrillos LEGO. El segundo LEGO Mindstorm, que permite trabajar con todos los ladrillos de la línea de robots Mindstorms, el tercero es LEGO Digital Designer Extended, permite construir modelos combinando cualquier tipo de ladrillo o piezas de LEGO sin límites.

(Cacique Borrego & Camarillo Gómez, 2013)

2.3.2.3.2. *LeoCAD*

Como menciona el software gratuito, desarrollado por Leonardo Zide, que es usado en el diseño asistido por ordenadores (CAD), en él se pueden crear construcciones tipo LEGO, de manera fácil, rápida y con acabados profesionales, está alineada a estándares relacionados al modelado de construcciones tipo LEGO, su uso es cómodo y flexible, para la creación de modelo basta con arrastrar y soltar las piezas LEGO, que luego son orientadas y configuradas según el gusto del diseñador. (Toro, 2017)

Dentro de las características de LEOCAD se destaca:

- Interfaz intuitiva, con funcionalidades para usuarios noveles como avanzados.
- Línea de aprendizaje corta.
- Múltiples opciones para el trabajo de cada pieza (rotación, coloreado, texturas, entre otros).
- Compatible con LDraw, que es el formato de archivo estándar para la creación de modelos de Lego.
- Permite el acceso a las librerías de LDraw, la cual cuenta con más de 10000 piezas de Lego, que son actualizadas constantemente.
- Importa y exporta archivos LDR y MPD, lo que permite compartir fácilmente tus proyectos en internet o editar los que otros han distribuido.
- Posibilidad de imprimir los modelos en impresoras 3D.
- Se encuentra en los repositorios de las distros Linux más importantes.
- Es multiplataforma (Linux, Windows y OSX).
- Libre y de código abierto.

2.4. Países que han incluido en sus currículos materias relacionadas al pensamiento computacional o las ciencias computacionales.

2.4.1. Reino Unido.

El Reino Unido introdujo un currículo en el año 2014, una asignatura nombrada como Computing que reemplazó a la anterior asignatura tecnologías de la Información y la comunicación para los niveles Primaria y educación Secundaria. El Departamento de Educación del gobierno británico sostiene que la introducción de la programación en el currículo se cimienta en la relevancia del pensamiento computacional, la creatividad para comprender y cambiar el mundo. En este tipo de conocimiento computacional involucra múltiples disciplinas tales como las matemáticas, las ciencias experimentales, la tecnología o el diseño. Estas disciplinas hacen parte de las ciencias de la computación que estudia lo que puede ser computado, codificado, para emplearlo en la solución de problemas; las tecnologías de la información que se ocupan de los dispositivos digitales y cómo utilizarlos para el almacenamiento, recuperación, transmisión y análisis de datos y, por último, la alfabetización digital o capacidad para navegar eficaz, responsable, segura y críticamente, así como crear productos digitales usando diversas tecnologías digitales (Valverde, Fernández y Garrido, 2015).

2.4.2. Inglaterra.

Comenzando el año académico 2014-15, Inglaterra ha incluido explícitamente el estudio del Pensamiento Computacional y programación de computadoras como parte del plan curricular

de la educación primaria y secundaria, como se describe en el currículo nacional en Inglaterra: Estudio de Programa de Computación. (Basogain, Olabe, y Olabe, 2015).

2.4.3. España

Toma como referencia el modelo británico, incluyendo una asignatura llamada Tecnología, Programación y Robótica, que deben cursar todos los estudiantes durante el primer periodo de la Educación Secundaria Obligatoria (tres cursos). Los estudiantes deben dedicar a los cursos dos horas semanales (de un total de 30 horas de clase a la semana) (Valverde, Fernández y Garrido, 2015).

Ejes temáticos:

- Programación y pensamiento computacional.
- Robótica y la conexión con el mundo real.
- Tecnología y el desarrollo del aprendizaje basado en proyectos.
- Internet, su uso seguro y responsable.
- Técnicas de diseño e impresión 3D.

Con el conocimiento adquirido el estudiante estaría en la capacidad de crear una página web, desarrollar aplicaciones móviles, diseñar un videojuego, manipular impresoras 3d y tener conocimientos de robótica. El software utilizado es gratuito, se utiliza Scratch en la programación por bloque, AppInventor para el desarrollo de aplicaciones móviles y Arduino para el desarrollo de robots.

1º ESO	2º ESO	3º ESO
<ol style="list-style-type: none"> 1. Internet: páginas Web, aplicaciones que intercambian datos. <ul style="list-style-type: none"> ▶ Uso seguro de Internet. 2. Privacidad y responsabilidad digital. 3. Herramientas de programación por bloques. 4. Aplicaciones para dispositivos móviles. 5. Proyectos tecnológicos. <ul style="list-style-type: none"> ▶ Fases del proyecto tecnológico y su documentación. ▶ Representación gráfica en proyectos tecnológicos. ▶ Innovación y creatividad tecnológica. ▶ Proyectos de desarrollo de aplicaciones informáticas. 6. Materiales de uso tecnológico. 7. Electricidad y circuitos eléctricos en continua. <ul style="list-style-type: none"> ▶ Análisis, simulación, montaje y medida de circuitos eléctricos. 	<ol style="list-style-type: none"> 1. Análisis y resolución de problemas mediante algoritmos. 2. Internet: arquitectura y protocolos. 3. Seguridad en Internet. 4. Aplicaciones y servicios para internet y nuevas tendencias en la red. 5. Páginas Web. Gestores de contenidos (CMS) y herramientas de publicación. 6. Estructuras y mecanismos. 7. Diseño e impresión 3D. 8. Conceptos básicos de señales y sistemas de comunicaciones. 9. Sistemas electrónicos analógicos y digitales. <ul style="list-style-type: none"> ▶ Componentes eléctricos y electrónicos. ▶ Análisis, simulación, montaje y medida en circuitos electrónicos. 10. Programación de sistemas electrónicos (robótica) 	<ol style="list-style-type: none"> 1. Formulación de un proyecto tecnológico. Identificación del problema. Análisis de su naturaleza. 2. Innovación y creatividad para la búsqueda de soluciones tecnológicas. 3. Diseño y representación gráfica de los elementos de un proyecto tecnológico 4. Documentación de un proyecto para la elaboración de un prototipo tecnológico. 5. Divulgación de la evolución de un proyecto tecnológico a través de la Web. 6. Diseño y fabricación de los elementos mecánicos de un proyecto tecnológico mediante impresión 3D. 7. Diseño, montaje y medida de los circuitos electrónicos de un proyecto tecnológico. 8. Programación de los circuitos electrónicos de un proyecto tecnológico. 9. Documentación de un prototipo desarrollado a través de un proyecto tecnológico.

*Figura 10.*Contenidos de la asignatura (Tecnología, Programación y Robótica)

2.4.4. Estados Unidos

Las escuelas Quest to Learn(Q2L) en Nueva York y Chicago brotan de la necesidad de diseñar entornos de aprendizaje que reconozcan las necesidades de niños y niñas que crecen en un era digital, rica en información y globalmente compleja. Las instituciones que participaron en el diseño educativo para los años 2006 y 2007 de esta escuela, utilizan el juego como aprendizaje, la Fundación MacArthur, New Visions for Public Schools, Center for Transformative Media, Parsons the New School for Design y el Institute of Play. Lo innovador de estas escuelas es el uso de un aprendizaje situado y basado en juegos pedagógicos, todos ellos conducen a un aprendizaje significativo, son dinámicos e inmersivos, interactivos y que exigen la intervención del jugador (Valverde, Fernández y Garrido, 2015).

Según Valverde, Fernández y Garrido (2015), las prácticas definidas por quest to learn son:

- Identidad: Implica la pertenencia a una comunidad de práctica y el ejercicio de roles tales como escritor, diseñador, lector, productor, profesor, alumno y jugador.
- Uso del diseño de juegos y sistemas de pensamiento: todo lo que se hace en el centro educativo se conecta con la vida fuera de la escuela a través del juego y el conocimiento.
- Prácticas contextualizadas: la escuela es un espacio de práctica donde los sistemas en los que vivimos son modelados, diseñados, examinados, reinventados y jugados, como formas de conocimiento.
- Juego y reflexión: usar el juego y reflexionar sobre mi aprendizaje con ellos.

- Teorizar y evaluar: el aprendizaje me permite validar teorías sobre el mundo.
- Respuesta a la necesidad de conocer: la motivación conduce a hacerse preguntas difíciles, buscar respuestas complejas e imaginar soluciones con otros.
- Interactuar con otros: los juegos no solo son un modelo para ayudar a pensar cómo funciona el mundo, sino también un medio dinámico a través del cual comprometerse socialmente y desarrollar una comprensión más profunda de uno mismo en el mundo.
- Experimentar e imaginar posibilidades: tomar riesgos, generar significados y actuar de modo creativo.
- Dar y recibir feedback: hacer visible el aprendizaje, saber cómo anticipar lo que necesitaremos aprender después.
- Inventar soluciones: resolver problemas usando el diseño de juegos (identificar reglas, inventar un proceso, ejecutar y evaluar). La arquitectura interna de los juegos (reglas, componentes, mecánicas, objetivos, conflictos, opciones y espacio) guían el diseño de las experiencias de aprendizaje. A través del currículo Q2L el diseño de juegos, que requiere un alto nivel de pensamiento complejo para asegurar que todos los elementos interactúan de modo significativo, es una estrategia de aprendizaje para los estudiantes.

El progreso de las capacidades que se espera del estudiante, atienden a habilidades y capacidades que vinculan al pensamiento computacional desde el punto de vista de la persona donde interviene su entorno social y cultural. Al aprender estas destrezas el estudiante gozará de la capacidad de identificar relaciones causales entre cosas e ideas, pensar eficazmente causas y efectos, instituir patrones y su relación en el tiempo y el espacio, clasificación de datos y su

organización en categorías , explicar el conocimiento según su punto de vista, aplicar el conocimientos en circunstancias y situaciones, etc.

3. TEORÍAS DE APRENDIZAJE

3.1. Teorías Del Aprendizaje En El Contexto Educativo.

El proceso de la generación del conocimiento se empieza dentro del aula, con la elección de un modelo de transmisión del mismo. Esta elección la hace el tutor de acuerdo al tipo de aprendizaje que desea que se origine en los estudiantes. La elección de un modelo pedagógico cumple también una lógica presente en la transferencia de los conocimientos (Pérez , 2004).las teorías educativas y de aprendizaje, han sido adoptadas y adaptadas en el medio educativo actual, intervienen en la educación y se han desarrollado a lo largo de los últimos cincuenta años. En la educación moderna se formulan una serie de teorías que buscan explicar el proceso de la enseñanza y el aprendizaje en un individuo, centrándose en como adquiere el conocimiento un estudiante, estas teorías se apoyan en investigaciones del aprendizaje cognitivo y otros estudios relacionado al contexto de la enseñanza y aprendizaje. A continuación se define el concepto de aprendizaje, que es de gran importancia para las teorías expuestas, este concepto se considera la base de estas teorías. (Zapata, 2015)

3.2. ¿Qué es el aprendizaje?

Entre las dificultades que sufre la definición de aprendizaje es la gran variedad de acciones que pueden ser definidas como tal, cada escuela psicológica a entregado su propia definición del término (Heredia & Sánchez, 2013), las definiciones más aceptadas o acertadas son las siguientes.

- Reforma de la conducta como consecuencia de la experiencia.
- Transformación permanente en las asociaciones o representaciones mentales como resultado de sus interacciones.

Las definiciones aluden a un cambio, sin embargo no se estipula un tiempo para este, o si este es permanente, no menciona si habrá más de estos cambios, se atribuyen a la experiencia de la persona, se exteriorizan cuando hay fenómenos que transmutan la vida del individuo, cuando somos bebés o infantes y adquirimos el habla, se está sometido constantemente a experiencias que se adquiere con el tiempo esto enriquece la capacidad para aprender un lenguaje. (Heredia & Sánchez, 2013)

3.3. El Modelo De Pedagogía Tradicional.



Figura 11. Educación tradicional

Este modelo pedagógico considera al estudiante como un ser sumiso, al que hay que certificarle el conocimiento. Este modelo empírico lleva a la memorización de conceptos sin que se promueva el conocimiento. Los conceptos se asimilan sin que se reflexione o interpreten. Se despliega un pensamiento empírico que tiene un carácter clasificador, ordenador, el alumno se orienta por las condiciones externas del objeto y por propiedades aisladas, este estilo de pedagogía no reflexiona sobre los procesos de aprendizaje en los estudiantes, consecuentemente no disponen de las acciones que el estudiante debe realizar, ni controla la adquisición del conocimiento. El resultado es lo más importante al evaluar al estudiante, el énfasis no conlleva un análisis o razonamiento. (Pérez, 2014)

3.4. Enfoques más importantes de las teorías de aprendizaje

3.4.1. El constructivismo

Constructivismo es una teoría muy discutida entre los teóricos de la psicología y la educación, sus principales representantes son: Jean Piaget (1952), Lev Vygotsky (1978), David Ausubel (1963) y Jerome Bruner (1960). Esta teoría menciona que todo conocimiento previo puede originar uno nuevo, indica que el aprendizaje es un proceso activo, que cada persona altera constantemente de acuerdo a sus experiencias. Es decir, todo estudiante construye soluciones a situaciones a través de un proceso mental que implica la adquisición de conocimiento nuevo, así como la contingencia de edificarlo y adquirir las capacidades que le permitirá aplicar dicho conocimiento a situaciones nuevas. (Berardinelli, Padrino, Ramírez & Páez, 2010)



Figura 12 Método constructivista

El Constructivismo se ha convertido en referencia y metodología que guía la gran mayoría de las investigaciones en la enseñanza de las ciencias a nivel mundial. Se comienza a hablar de esta metodología en la década de los 70, sin embargo se desarrolla formalmente en la década de los 80, en el constructivismo se aprecian diferentes corrientes y concepciones. (González, 2002)

Como Menciona Trujillo (2017), el constructivismo ayuda al estudiante a interiorizar un pensamiento, re amoldándolo y transformando la información. Esta transformación ocurre gracias a la adquisición de nuevo conocimiento provocando el levantamiento de nuevas estructuras cognitivas que permiten enfrentarse a situaciones iguales o parecidas en la realidad.

- **Concepción del estudiante:** El proceso de aprendizaje es responsabilidad del estudiante, la participación y colaboración de sus compañeros es de gran importancia para llevar el ámbito teórico a lo práctico.

Hay dos autores que han aportado al constructivismo: Piaget con el constructivismo psicológico y Vygotsky con el constructivismo social. (Trujillo, 2017)

- **Constructivismo psicológico:** El proceso de aprendizaje es personal, estimula el área cognitiva del ser humano, donde se genera nuevo conocimiento y se incentiva el deseo de saber, encontrando explicaciones al mundo que nos rodea. Se afirma que toda actividad constructivista debe existir una idea tal que el conocimiento previo de ella se vea afectado haciendo un reacomodo del viejo conocimiento para asimilar el nuevo. El aprendizaje frecuentemente puede ser trabajado por descubrimiento, experimentación y manipulación de realidades concretas, pensamiento crítico, diálogo y cuestionamiento continuo.
- **Constructivismo social:** El aprendizaje sólo es significativo en un contexto social, este paradigma no niega al constructivismo psicológico, no obstante considera que está incompleto. El origen de todo conocimiento no es la mente humana, sino una sociedad dentro de una cultura dentro de una época histórica. El lenguaje es la herramienta cultural de aprendizaje. El sujeto es participe de su conocimiento no porque sea una función natural de su cerebro, más bien porque se le ha enseñado a construir a través de un dialogo continuo con otros seres humanos.

En el artículo escrito por Valdez (2012), menciona los principios de un aprendizaje significativo en los que se basa el constructivismo son:

- *Activo*: Los alumnos son responsables de su proceso de aprendizaje donde captan la información de manera consiente, de cuyo resultado son responsables.
- *Constructivo*: Los alumnos acogen nuevas ideas a un conocimiento previo para dar significado o reconciliar disconformidades o incertidumbres.
- *Colaborativo*: Los estudiantes trabajan en equipo y construyen nuevo conocimiento, aprovechando las habilidades de sus compañeros y aportando apoyo social.
- *Intencional*: Los estudiantes tienen objetivos a nivel cognitivo de forma activa e intencional.
- *Conversacional*: Aprender es parte de un proceso social, en el cual el estudiante capta conocimientos de las personas que lo rodea, tanto dentro de clase como fuera.
- *Contextualizado*: Las actividades están enfocadas a vivencias de la vida diaria o simulada mediante un entorno de aprendizaje basado en algún caso o problema.
- *Reflexivo*: los alumnos articulan lo que han aprendido y reflexionan sobre los procesos y decisiones implicadas.

El constructivismo se enfoca en ayudar al estudiante a resolver problemas, realizar labores en función de un conocimiento obtenido a partir de los conocimientos encaminados en clases y herramientas usadas por el tutor. Las vivencias y conocimientos previos del alumno son claves para lograr mejores aprendizajes (Valdez, 2012). En esta teoría se plantean diferentes conceptos que están relacionados con la función del alumno:

- *Explorador*: Los estudiantes tienen la capacidad de explorar nuevas ideas, herramientas que lo impulsan a considerar ideas y exploraciones.
- *Aprehensión Cognitiva*: El aprendizaje es orientado por un mentor quien dirige a los alumnos para el desarrollo de ideas y habilidades y estimulando la práctica a nivel profesional.
- *Enseñanza*: Los alumnos aprenden con argumentos formales e informales.
- *Producción*: Los estudiantes desarrollan productos para su beneficio y el de otras personas.

3.4.2. El conductismo

Al hablar del conductismo, se hace referencia al comportamiento humano, lo que sirve de eje de trabajo a esta corriente psicológica, sus máximos exponentes son: Iván Petrovich Pavlov, John Broadus Watson, Edward Thorndike y Burrhus Frederic Skinner. Los inicios de la teoría se remontan a los primeros años del siglo XX, donde surge como una teoría psicológica y posteriormente se adapta su uso a la educación, esta teoría es la primera en influenciar en gran medida la manera en que se entiende el aprendizaje humano, desde el ámbito educativo el conductismo establece que el aprendizaje es un cambio en el comportamiento en función del entorno, por consiguiente el aprendizaje es el resultado de la asociación de estímulos y respuestas, el conductismo ve al aprendiz o alumno como un sujeto cuyo desempeño, aprendizaje puede ser transformado, basta con implementar buenos insumos educativos para lograr un cometido y una conducta deseable. El maestro ve al estudiante conductista como una hoja en blanco que, cumple órdenes y obedece, requiere constante aprobación, es dependiente del tutor, su existencia es pasiva en el proceso de enseñanza- aprendizaje, realiza tareas en las cuales el comportamiento pueda ser observado, medido, evaluado directamente. El conductismo pretende que el alumno responda a los estímulos ambientales y que se convierta en un ser auto-disciplinado. (Valdez, 2012)

En el artículo de Trujillo (2017), se menciona que la teoría conductista es importante en la captación de conocimientos memorísticos, ya que suponen niveles primarios de comprensión donde la preparación estímulo-respuesta, es útil.

- **Concepción del estudiante:** El estudiante es un receptor de conocimientos, su ambición debe ser aprender lo que se le enseña, el desempeño escolar del aprendiz puede ser modificado desde el exterior, para ello se debe planear adecuadamente los insumos educativos. La motivación es externa y debe ser reforzada con premios o castigos para mejorar el aprendizaje. El aprendiz es como un robot que es posible predecir, conocer su estado y las conductas de interacción.
- **Concepción del maestro:** En este paradigma el tutor es un ser omnipotente que está dotado con el conocimiento, que pone en práctica dependiendo de las necesidades en el aula, esto garantizando al estudiante el aprendizaje. Su trabajo consiste en desarrollar estrategias de reforzamiento y control en los estímulos que utilice para enseñar.

El trabajo del profesor reside en desarrollar una apropiada serie conocimiento para el refuerzo y control de estímulos para enseñar. El conductismo es uno de los paradigmas de mayor tradición, no obstante se queda corto comparado con los nuevos paradigmas de aprendizaje, es fuertemente criticado porque percibe al aprendizaje como algo mecánico, deshumanizado y reduccionista, sin embargo, aún tiene gran vigencia en nuestra cultura dejando, una amplia gama de prácticas que todavía se utilizan en muchos sistemas escolares. (Valdez, 2012)

3.4.3.El cognoscitivismo

La importancia de este paradigma reside en las capacidades y los valores, esto indica procesos cognitivos afectivos de los aprendices. Las temáticas y metodologías están enfocados a desarrollar capacidades y valores (Trujillo, 2017). Este paradigma está orientado a varias dimensiones de lo cognitivo (atención, percepción, memoria, inteligencia, lenguaje, pensamiento, etc.)

Entre los representantes de este paradigma se encuentran, J. Novak , Avram Noam Chomsky, Ulric Neisser y Albert Bandura que aportaron conocimientos significativos a la teoría del aprendizaje cognitivo. Esta teoría asume la mente como un agente activo en el proceso de aprendizaje, levantando y acomodando los esquemas mentales. El cognitivismo pretende adicionar nueva información al esquema cognitivo existente, se centra en estudiar como el ser humano construye su pensamiento a través de sus estructuras organizativas y funciones adaptativas al convivir con el medio que lo rodea, se centra en la construcción de esquemas que expliquen el comportamiento humano y los procesos según los cuales se resuelven los problemas. (Valdez, 2012)

La teoría cognitiva afirma que el aprendizaje depende de lo que el sujeto observe de acuerdo a sus competencias y percepciones previas o antecedentes, las cuales tiene un comportamiento dependiendo del individuo, influenciado por sus actitudes y motivaciones. En el paradigma cognitivo el sujeto es un organismo cognitivo-afectivo. Esta nueva percepción facilita el procesamiento de la información, reconociendo la importancia de cómo las personas organizan, filtran, codifican, categorizan, y evalúan la información, tomando en cuenta, la forma en que sus

estructuras o esquemas mentales son empleadas para acceder e interpretar la realidad. (Trujillo, 2017)

La concepción de la teoría cognitiva, según los entes involucrados se divide en la:

- **Concepción del estudiante:** Es un ente activo que es capaz de procesar el conocimiento, además de tener la capacidad de aprender y solucionar problemas; estas habilidades evolucionan con los nuevos aprendizajes, desarrollan estrategias.
- **Concepción del maestro:** Teniendo en cuenta que el estudiante es un ente activo en el proceso de aprendizaje, el tutor se enfoca en la confección y organización de experiencias didácticas para lograr una adecuada enseñanza. No desempeña el papel protagonista, donde tiene la verdad absoluta.

3.4.4. El conectivismo

Una teoría del aprendizaje de la era digital, que toma como referente las teorías de aprendizaje, analiza sus limitaciones, el conductismo, el cognitivismo y el constructivismo, para ilustrar el efecto de las tecnologías sobre la manera en que actualmente vivimos, nos comunicamos y aprendemos. Es la unión de los principios abordados en las teorías del caos, redes neuronales, complejidad y auto organización. Éste se enfoca en la inclusión de tecnología como parte de nuestra distribución de cognición y conocimiento (Barón, sf)

Cuando se menciona el conectivismo se hace referencia a enlazar ideas, vincular conocimiento, crear nuevas conexiones, formar patrones y mantener e incrementar las redes personales de aprendizaje (Altamirano, Becerra & Nava, 2010), el conectivismo puede entenderse como un punto de vista del aprendizaje que afirma:

- El conocimiento existe distribuido y en red.
- El aprendizaje es el proceso de conformar y podar conexiones en las redes sociales y tecnológicas.

Lo que distingue a una red conectivista es que produce precisamente conocimiento conectivo, factible del aprendizaje, se mencionan cuatro elementos necesarios para conformar un conocimiento conectivo: autonomía, diversidad, apertura e interactividad/conectividad.

- Autonomía: Cada individuo conectado debe tomar decisiones propias en todo cuanto le compete a su participación en la red, incluyendo las plataformas y herramientas que utilizará para participar.
- Diversidad: La diversidad de opiniones, culturas, lenguajes, antecedentes, espacios físicos, intereses personales, es un elemento deseable para que la interactividad sea realmente productiva.
- Apertura: Una tendencia importante en Internet es la apertura. Contenidos de conocimiento abiertos y participación de los individuos en la discusión y generación de conocimiento libre.

- Interactividad y conectividad: La interacción puede generar conocimiento nuevo, útil. Este conocimiento es producido por la comunidad, no es poseído por una sola persona, es comunitario.

La concepción de la teoría conectivista, según los entes involucrados:

- ***Concepción del tutor:*** El papel del tutor en esta teoría del aprendizaje, es orientar a los estudiantes a elegir fuentes confiables de información y a su vez seleccionar la información más importante, es decir, tener la habilidad para discernir entre la información que es importante y la que es trivial.
- ***Concepción del estudiante:*** el papel del estudiante se centra en adquirir la habilidad para seleccionar entre tantas formas y medios de información y de comunicación. Por tanto, el punto de inicio del conectivismo es el individuo, el estudiante en nuestro caso.

4. ESTADO DEL ARTE

4.1. Pensamiento computacional: Una nueva alfabetización digital

La sociedad y los sistemas de producción, de servicio y consumo se han impresionado por el hecho de que hay poco personal capacitado en la industria de la informática, entidades o empresas que tiene puestos vacantes en desarrollador de software, digitalizadores, etc. se quedan sin cubrirlos por este problema. Ante esta situación los sistemas educativos de los países más sensibles a la problemática, lo han abordado desde la perspectiva de una reorganización curricular. (Zapata-Ros, 2015)

Las sociedades más conscientes se han dado cuenta que una posible solución del problema radica en la implementación de una alfabetización digital (El conocimiento y la capacidad de utilizar las computadoras y la tecnología de manera eficiente), por lo tanto hay que aplicarla desde las primeras etapas del desarrollo individual, las habilidades que favorecen ciertas actividades, con entornos de aprendizaje, desarrollan el pensamiento específico, de un pensamiento computacional.

Por otro lado al igual que el estudiante se forma en competencias tales como la pre-lectura, pre-escritura o pre-cálculos que son bases importantes, es necesario hablar de pre-codificación, para referirnos a las competencias previas necesarias antes de la fase del desarrollo para la codificación, esto no corresponde con el propósito más amplio que es el pensamiento computacional, no solo es prepararse para la programación y para la codificación, sino dotarse de

claves de comprensión y representaciones de los objetos de conocimiento en general que en esencia constituye el concepto de pensamiento computacional. (Zapata-Ros, 2015)

4.1.1. ¿Cuál es el problema del pensamiento computacional?

Enseñamos a los niños a escribir líneas de código, sin pensar en el problema a resolver, sin diagramas, sin diseños, es la idea que plantean los concursos de programación, enseñar a través de los videojuegos, etc. Es un planteamiento competitivo que excluye a muchos niños y que posiblemente haga poco deseable ser programador. (Zapata-Ros, 2015)

Volvamos al principio, antes de escribir líneas de código de forma compulsiva, pensemos en la representación de la realidad, las expectativas, las personas exitosas en el mundo de la programación, no es importante el software en que se desarrolla un programa, si no la manera en que se piensa la solución a un problema, conocer el mundo de ideas y de representaciones, es parte importante de los principios que rigen el pensamiento computacional.

Si se aprende a programar en un lenguaje de programación es difícil que se produzca una transferencia a otro en futuras ocasiones, las asociaciones profesionales se quejan de las empresas que contratan informáticos de forma temporal, no inspeccionan la forma en que estos han aprendido el lenguaje de programación, cuando cambian el programa, el lenguaje o la versión, no tendrán la flexibilidad mental requerida para adaptarse a nuevos entornos, no solo de programación si no de problemas, esto es sabido por las empresas. Si se contratan profesionales relacionados a elementos del pensamiento heurístico, como los matemáticos y físicos que poseen estas competencias de resolución de problemas en entornos cambiantes. (Zapata-Ros, 2015)

Existen investigaciones que prueban que cuando se empieza con la enseñanza del pensamiento computacional en vez de la elaboración de código, se evita el principio de discriminación que hace que los niños o niñas se desmotiven por aprender un lenguaje de programación, dando paso así a la democratización del aprendizaje , esto supone que los niños y niñas que en el futuro tiene profesiones de bibliotecarios, médicos o artistas , puedan ser buenos programadores, ampliando así la base del conocimiento de la computación, lo que constituye una sociedad del conocimiento.

4.1.2. Definición del pensamiento computacional

Se propone la construcción de ideas del pensamiento computacional a partir de elementos o formas específicas del pensamiento para la solución de problemas, relacionándolo con la alfabetización digital que constituye competencias claves en los procesos de aprender, comprender ideas y fenómenos no solo en el ámbito de la programación de ordenadores, el internet o la sociedad del conocimiento, sino que es útil para aprender operaciones cognitivas y elaboración compleja, sin estos elementos es más difícil la solución de ciertos problemas, estas habilidades son necesarias para la vida y se consideran talento especial. (Zapata-Ros, 2015)

El enfoque computacional toma una visión del mundo como un rompecabezas, que se pueden desarmar en partes pequeñas y armarlo poco a poco usando la lógica y el razonamiento deductivos, el rompecabeza representa un problema que se divide en partes pequeñas (problemas) para así ir resolviéndolos, pero hay situaciones en las que hay tareas complejas que son consideradas partes del pensamiento, como es el análisis ascendente, la heurística, pensamiento

divergente o lateral, la creatividad, la resolución de problemas, pensamiento abstracto, la recursividad, la iteración, los métodos de aproximaciones sucesivas, el ensayo y error, los métodos colaborativos, etc.

4.1.3. Un dominio teórico específico del pensamiento computacional en las teorías del aprendizaje y un currículo

Hasta el momento se ha hablado de establecer un currículo sobre el pensamiento computacionales, importante realizar un análisis del contexto y de implementar esta clase de pensamientos de manera interdisciplinar y las implicaciones que tiene para modificar el dominio teórico específico dentro de las teorías de aprendizaje , así describir un primer contacto o acercamiento a un currículo que cumpla los dominios conceptuales para las etapas educativas y para la capacitación de maestros y profesores. (Zapata-Ros, 2015)

Se encuentra así los siguientes componentes del pensamiento computacional

4.1.3.1. Análisis descendente

El análisis descendente involucra, un método general de solución de algoritmos, que es posible llevar al diseño de submétodos o módulos de resolución de problemas distintos y auxiliares. (Zapata-Ros, 2015)

4.1.3.2. *Análisis ascendente*

Una de las formas de abordar un problema complejo, es solucionar los problemas más concretos, luego se solucionan los problemas abstractos, esta forma de solución de problemas de lo concreto a lo abstracto es denominada análisis ascendente. (Zapata-Ros, 2015)

4.1.3.3. *Heurística*

La heurística se considera un arte, una técnica o conjunto de procedimientos prácticos o informales para la solución de problemas, los métodos heurísticos no tienen garantía de que la solución que se obtiene es única ni la más óptima, este tipo de saberes se obtiene a través de la observación, el análisis y el registro.

Encontramos que la heurística tiene como base la sistematización de la experiencia al solucionar problemas, hay una serie de pasos que son importantes seguir: Analizar el problema, imaginar un plan, ejecutar el plan y utilizar técnicas de descomposición de los problemas en problemas más sencillos. (Zapata-Ros, 2015)

4.1.3.4. *Pensamiento lateral y pensamiento divergente*

El pensamiento lateral es el uso de la información de tal forma que esta genere ideas creativas reestructurando los conceptos ya existentes en la mente, se cultiva con el estudio y se desarrolla en la práctica en la solución de problemas

Por otro lado el pensamiento divergente es aquel que usa los recursos empleados en la educación, estrategias, métodos educativos, tiene sus orígenes en las artes. (Zapata-Ros, 2015)

4.1.3.5. *Creatividad*

Uno de los principales rasgos de la creatividad es el uso del pensamiento convergente y divergente, en un medio orientado a captar la creatividad, los alumnos cuyo pensamiento es fluido, flexible y generador de ideas innovadoras, originales tiene mayor probabilidad de obtener creaciones, por esta razón es importante el desarrollo del pensamiento divergente en las aulas, talleres y laboratorios , igualmente la presencia de un profesor es de gran importancia ya que tiene como objetivo, orientar las ideas más apropiadas que generen los estudiantes. (Zapata-Ros, 2015)

4.1.3.6. *Resolución de problemas*

La resolución de problema es una variante metodológica del pensamiento computacional, a aquellos problemas cuya solución está ligada a un ordenador, la abstracción, la recursividad y la iteración se utilizan para analizar y procesar datos para crear métodos de solución de problemas, y crear artilugios reales y virtuales para resolverlos.(Zapata-Ros, 2015)

4.1.3.7. *Pensamiento abstracto*

Se considera la capacidad de una persona de manejar modelos ideales de la realidad, abstrayendo los objetos relevantes, se estudian sus propiedades al obtener el modelo abstracto, se

obtienen conclusiones o reglas que permitan saber el comportamiento de los objetos, se considera el pensamiento matemático como un pensamiento abstracto por excelencia. (Zapata-Ros, 2015)

En el diseño curricular es importante el desarrollo intelectual, teniendo en cuenta el paso de lo concreto a lo abstracto, es fundamental en la presentación de contenidos a desarrollar en un currículo.

El estudiante considera lo complejo como fácil y lo abstracto como difícil, esto se debe a que el alumno se enfrenta a una habilidad o un concepto para lo cual no está aún preparado, no por la edad sino por las condiciones que se produce en el aprendizaje. (Zapata-Ros, 2015)

4.1.3.8. *Recursividad*

La recursividad es un concepto que va más allá de las matemáticas o la computación, es una forma de pensar, cuando un problema es muy difícil de resolver por los medios convencionales, puede ser remitido a otro problema de las mismas características o naturaleza pero más pequeño que puede ser resuelto o nos dé una pista de una regla a usar (regla de recurrencia), en ambos casos nos permite resolver el problema, a estos métodos de resolución de problemas se les llama recursividad o recurrencia. (Zapata-Ros, 2015)

4.1.3.9. *Iteración*

El concepto es asociado al uso de los ciclos repetitivos o bucles, For To, while, do-while, repeat y a diagramas de flujo, es complicado hablar de iteración sin mencionar la construcción de

algoritmos repetitivos, no obstante pocas veces exploramos la esencia de este concepto, lo básico, en las primeras etapas del desarrollo y el pensamiento de este tipo.(Zapata-Ros, 2015)

Un ejemplo de iteración es cuando el niño aprende las fracciones, números racionales, numero reales, su notación, representación son procedimientos iterativos. También el sistema de medición, de magnitudes, de peso, masa, volumen, superficie son procesos de representación conceptual iterativas.

La iteración es un componente esencial del pensamiento computacional con una amplia influencia en otras representaciones cognitivas en procedimientos que son la base de importantes actividades y tareas. (Zapata-Ros, 2015)

4.1.3.10. Método por aproximaciones sucesivas (Ensayo - error)

Es un método que utilizamos para confrontar las ideas, que se forman a partir de la realidad percibida. Son acciones, percepciones en la formación de modelos cognitivos, de ideas, desde que nacemos utilizamos el ensayo- error, los sentidos, la experimentación y la representación de las ideas obtenidas de las experiencias.

El método por aproximaciones es además un procedimiento muy usado por los programadores de forma espontánea y profunda. (Zapata-Ros, 2015)

4.1.3.11. Contingencia e inmediatez

Las aproximaciones sucesivas juegan un papel importante en los objetivos educativos, es decir la acción tutorial proporciona instrucciones para la resolución de problemas, el estudiante puede ensayar públicamente en el ambiente de clase métodos de resolución y puede pedir asesoría al tutor, profesor y a los propios alumnos, el cuándo y el donde el tutor imparta la tutoría es clave importante en el proceso. Es necesario que se hagan en el momento pertinente de manera contingente ya que el estudiante puede ser sensible a la pérdida de tiempo y perder la motivación, la respuesta debe ser inmediata a los errores, para que no se pierda la inmediatez. (Zapata-Ros, 2015)

El papel del tutor es detectar, en el lugar y en el momento que se produzca, la dificultad del aprendiz que comprenda escasamente el tema que es objeto de aprendizaje, de tal manera que él pueda intervenir continuamente para reparar el error y mostrar al alumno que hacer. Así el objetivo es apoyar la resolución de problemas con aproximaciones sucesivas hasta un desempeño competente.

4.1.3.12. Métodos colaborativos ¿Hacer cosas juntos o entender cosas juntos?

El aprendizaje colaborativo y el trabajo colaborativo son muy importantes en la enseñanza, en un mundo computacional la dificultad de los desarrollos y arquitecturas hacen inconcebible el trabajo aislado, tiende a darse un fuerte flujo de trabajo y de comunicación que hagan posibles proyectos comunes en equipos amplios, la disposición a compartir y al trabajo colaborativo son elementos esenciales del pensamiento computacional. (Zapata-Ros, 2015)

El trabajo en grupos pequeños o grandes dan lugar a situaciones de aprendizaje diferentes, cada una de las cuales lleva aparejado un análisis diverso, estos entornos que se mencionan permiten un trabajo eficiente que accede a procesar la información que genera el resto.

4.1.3.13. Patrones

Son una herramienta para el análisis de la programación con dos características importantes: impiden el trabajo molesto como lo es el repetir partes del código, diagramas de flujo o procedimientos, también permiten la capacidad de distinguir lo que tienen en común situaciones desiguales. Un patrón puede entenderse como una plantilla, una guía, conjunto de directrices o de normas de diseño que permite la adquisición de buenas prácticas, de esta forma un patrón resuelve un problema. (Zapata-Ros, 2015)

4.1.3.14. Sinéctica

La sinéctica es un punto donde se agrupan las teorías que tratan de explicar y estudian la creatividad, estudia cómo se organiza e integran sujetos que componen un grupo, para la solución de problemas.

La sinéctica se presenta como una metodología de resolución de problemas, que estimula los procesos de pensamiento de los cuales la persona puede no ser consciente, la teoría sinéctica considera que el entendimiento de los elementos emocionales e irracionales de un problema o idea, un grupo puede llegar a la solución de un problema de manera más eficiente, se considera un componente del pensamiento computacional. (Zapata-Ros, 2015)

4.1.3.15. Metacognición

Es el estudio de los estilos de aprendizaje, es la conciencia de los recursos cognitivos con lo que cuenta un aprendiz.

Los psicólogos del aprendizaje probaron que los estudiantes tenían diferentes formas de aprender ya que estos tenían distintas estructuras cognitivas, los escritores de los estudios de los estilos de aprendizaje no están preocupados por lo que el alumno aprenda, sino más bien la forma en que estos prefieren aprender, esto da lugar a un factor intencional. (Zapata-Ros, 2015)

Lo importante es indagar como el alumno puede adquirir conocimiento, como fórmalo para que desarrolle habilidades cognitivas.

Con respecto al pensamiento computacional la idea es investigar como los estudiantes pueden adquirir las habilidades metacognitivas específicas, indagar sobre las mejores estrategias de aprendizaje, detectar debilidades y fortalezas de los estilos, así cambiarlas y potenciarlas(Zapata-Ros, 2015)

4.1.3.16. Fases de creación de un código y componentes de pensamiento computacional

La creación de código está muy estudiada desde el punto de vista informático, se estudiara ahora desde el punto de vista del pensamiento computacional, que elementos del pensamiento presentan cada uno de los conceptos desarrollados anteriormente. (Zapata-Ros, 2015)

Las fases diferenciadas en el proceso de creación de un código son:

- Detección, delimitación del problema y su naturaleza.
- Delimitación de métodos y disciplinas en la resolución del problema.
- Organización de la resolución, feed back e investigación formativa.
- Diseño de la resolución.
- Algoritmia/ diagrama de flujo.- Incluye la discusión.
- Elaboración del código (programa).- Incluye codificación, ejecuciones e implementación, documentación, etiquetas, modularización.
- Prueba/ Validación.- Incluye implementación y depuraciones de errores, se pueden considerar otras competencias en cada una de la fases.

4.2. Desarrollo del pensamiento computacional con Scratch

Barrera y Montaña (2015) expresan que la primera vez que se utilizó el término pensamiento computacional fue en 1996 por Seymour Papert, se entiende como un método de resolución de problemas, además son procesos del pensamiento involucrados en la formulación y sus soluciones de modo que las salidas puedan ser ejecutadas por un agente de procesamiento.

El pensamiento computacional representa una ayuda al esquema educativo en todas las asignaturas, ya que el estudiante desarrolla las habilidades necesarias para la resolución de problemas.

El pensamiento computacional se conjuga con otros tipos de pensamiento como lo es el pensamiento lógico, analítico, algorítmico, abstracto, divergente y crítico.

Según Barrera y Montaña (2015), el pensamiento computacional posee las siguientes características:

- Formular problemas de tal manera que al solucionarlos permita el uso de un ordenador o herramientas computacionales.
- Permite ordenar analizar datos de manera lógica.
- Personifica datos como abstracciones, modelos y simulaciones.
- Usa el pensamiento algorítmico (Serie de pasos ordenados).
- Identifica, implementa, y analiza soluciones para encontrar la más eficiente y eficaz.

Algunos de los principales procesos del pensamiento computacional, identificados por Computer Science Teachers Associations (CSTA) e International Society for Technology in Education (ISTE), corresponden a:

- Recopilar datos: Recoger información de manera adecuada.
- Analizar datos: Instituir patrones y extraer conclusiones.
- Representar datos: Simbolizar y organizar en gráficos, cuadro, palabras o imágenes.
- Descomponer problemas: tomar un problema y dividirlos en problemas más pequeños.
- Abstraer: Disminuir la dificultad de un problema, para así instaurar las ideas principales.
- Algoritmos y procedimientos: series de pasos que se usan en la solución de un problema logrando un objetivo.
- Automatización: Programar la computadora para que haga las tareas tediosas o repetitivas.
- Simulación: Representación o modelado de un proceso.
- Paralelismo: Programar recursos simultáneamente para la realización de una tarea con el fin de dar solución a un problema.

Estos procesos son útiles en la creación de algoritmos, supone una tarea abstracta, los estudiosos en el área plantean desarrollarla de forma lúdica, por medio del juego, incentivando el aprendizaje y haciéndola asequible. (Barrera & Montaña, 2015)

Los juegos de computadoras son una forma de enseñar a las personas, ya que estos permiten desarrollar el lenguaje, despiertan el ingenio, la observación, afianzan la voluntad y afinan la paciencia, además de desarrollar la agudeza visual, táctil y auditiva.

Gracias a los beneficios de los juegos en el aprendizaje, se está optando de manera frecuente herramientas para que los estudiantes aprendan de una manera entretenida, rápida y clara. Actualmente existen muchas plataformas donde se desarrollan juegos o actividades, algunas de ellas permiten la enseñanza de la programación o formarse en el pensamiento computacional. (Barrera & Montaña, 2015)

4.2.1. Diseño de estrategia

El desarrollo de recursos, para generar habilidades lógicas de programación en Scratch, estos recursos poseen varias actividades donde cada una de ella se centra en el desarrollo del pensamiento computacional, en diferentes niveles. Estas actividades se usan luego para visualizar el progreso de cada estudiante estadísticamente, además de ser graduadas en los niveles del pensamiento computacional. (Barrera & Montaña, 2015)

El objetivo de los recursos es que sean usados por alguien con competencias en el área de la computación o educación para dar clases a los estudiantes interesados en la temática, los recursos son actividades en Scratch a realizar por los estudiantes y estas estarán supervisadas por un tutor a cargo, o que las actividades sean realizadas por el estudiante de manera autónoma siguiendo las instrucciones planteadas y diseñadas. (Barrera & Montaña, 2015)

Los recursos tienen un total de 11 sesiones, cada una con un tiempo estimado de 1 hora y 30 minutos a 2 horas.

4.2.2. Resultados

El material, se aplicó a dos muestras diferentes. La primera es un taller complementario a segundo ciclo de enseñanza básica, donde un grupo de 19 alumnos (9 de séptimo y 10 de octavo) de un colegio particular subvencionado de hombres, tomaron en forma voluntaria el taller durante un periodo de dos meses, con dos sesiones a la semana de dos horas pedagógicas, las actividades 10 y 11 se trabajaron en más de una sesión de trabajo, ya que les resulto complejo terminar la actividad en la sesión programada. La segunda muestra corresponde a 53 estudiantes que ingresan a la universidad, en particular a la Carrera Licenciatura en Ciencia de la Computación. En la asignatura de Computación I, se genera una unidad 0, donde los estudiantes tienen que desarrollar las actividades propuestas, esta unidad se ve en 4 sesiones de dos horas pedagógicas, en un comienzo se avanza en más de una actividad por sesión, y se enfatiza en las cuatro últimas que apuntan a niveles más profundos del pensamiento computacional. (Barrera & Montaña, 2015)

4.2.3. Conclusiones

En este trabajo se diseñó una estrategia para desarrollar el pensamiento computacional en personas, de los medios y formas revisadas, lo más recomendado fue por medio de juegos interactivos, pero más que jugar también creando estos juegos. La revisión de plataformas que permiten realizar juegos de forma lúdica nos llevó a Scratch, herramienta altamente utilizada hoy en día en los procesos de enseñanza interactiva, esta permite la generación de contenidos mediante una programación basada en bloques y en base a las funcionalidades que ofrece, se planifican los recursos y se desarrollan las actividades, se triangula con las componentes del pensamiento computacional. (Barrera & Montaña, 2015)

Los resultados obtenidos son alentadores y nos indican que estamos en el camino correcto, ya que se logra el objetivo propuesto y además las actividades interactivas que dejan una ventana abierta a la creatividad del aprendiz. El estudiante puede llegar a tener un pensamiento computacional avanzado con las actividades diseñadas, en forma lúdica y entretenida.

Quisiéramos destacar la creatividad que tienen los estudiantes y como esta emerge en forma natural en las últimas sesiones, el concepto de hacer un juego para un estudiante es un desafío y las reglas que ellos se imponen superan las expectativas de lo que se pretende. En este contexto, el juego una vez realizado es socializado y puesto a prueba por sus compañeros, resulta gratificante para el estudiante que su producto sea algo tangible y que otra persona lo quiera utilizar. También presenta un desafío su mejora para hacerlo lo más atractivo. (Barrera & Montaña, 2015)

En suma, las actividades interactivas para generar pensamiento computacional de alto nivel son una herramienta efectiva que ayuda a fortalecer los procesos lógicos que permiten la modelación correcta de solución de problemas, además de fomentar habilidades como la creatividad. Hay que seguir incursionando para establecer el efecto que provoca transversalmente en la resolución de problemas, en un ámbito más científico. (Barrera & Montaña, 2015)

4.3. El desarrollo del pensamiento computacional para la resolución de problemas en la enseñanza inicial de la programación

4.3.1. Introducción

En computación el hecho de programar involucra que a partir de un lenguaje como el natural, trazando límites, diseñando o modelando una solución a un problema y convertir dicha solución en un programa de computadora.

La programación de computadoras exige a los estudiantes diferentes formas de abordar un problema y plantear soluciones, ayuda a los aprendices a desarrollar habilidades para imaginar rutas de razonamiento divergente, anticipar errores y evaluar rápidamente los diferentes escenarios mentales. (Zúñiga, Rosas, Fernández & Guerrero, 2014)

Es importante que los estudiantes desarrollen destrezas, que van más allá de la simple codificación de un programa involucra aprender a entender un problema (abstraer, modelar y analizar), plantear soluciones certeras (pensar y considera sobre la abstracción, concretar estrategias, seguir un proceso, aplicar una metodología, descomponer el problema en problemas más simples), aprovechar un lenguaje en la solución del problema (codificarlo, entenderlo y respetar su sintaxis), emplear una herramienta que entienda el lenguaje(programar, compilar, ejecutar, depurar), ensayar que la solución sea acertada(comprender el concepto de ensayo-error).En este contexto, la resolución de problemas computacionales, presume el desarrollo de destrezas que forman parte del pensamiento computacional. (Zúñiga, Rosas, Fernández & Guerrero, 2014)

La complejidad no está en lo duro de un lenguaje de programación o en formular un algoritmo en ese lenguaje, sino en escribir un algoritmo que solucione el problema. El reto está en desarrollar el pensamiento computacional para solucionar problemas utilizando una computadora.

Zúñiga, Rosas, Fernández & Guerrero, (2014).Manifiestan que:

Como tutor de asignaturas del primer año en carreras universitarias debemos contribuir a la adaptación de los estudiantes a la vida universitaria. Ésta supone la apropiación de nuevas responsabilidades, normativas y hábitos, todo un desafío para muchos ingresantes, constituyéndose así en un factor que incide en su rendimiento académico. A partir de nuestra experiencia tutor y de los resultados académicos obtenidos en los últimos periodos, fue posible identificar algunos aspectos críticos de los alumnos ingresantes que pudieron influir de forma negativa en su desempeño (p. 2).

- Las habilidades académicas que caracterizan a la mayoría de estudiantes, es la falta de hábitos de estudio, problema en comprensión lectora, y dificultad en la resolución de problemas.
- Poca motivación e interés de parte del estudiante.
- Baja tolerancia al fracaso

Estos y otros factores hacen que el estudiante deserte de la carrera por bajo rendimiento académico. No obstante, es posible pensar en estas nuevas generaciones como nativos digitales que presentan características que rigen el comportamiento, habilidades y pensamientos.

4.3.2. Líneas de investigación y desarrollo

Las líneas de investigación y desarrollo según Zúñiga, Rosas, Fernández & Guerrero, (2014), son :

- *La resolución de problemas*: empleando nuevos modelos de enseñanza aprendizaje, algunos métodos como la programación por pares y el trabajo colaborativo, para favorecer el desarrollo del pensamiento computacional con el objetivo de plantear soluciones viables.
- *El uso de las TICs como soporte en el proceso de aprendizaje de la programación*: el uso de los recursos tecnológicos que permita y faciliten a los estudiantes la experiencia de la programación como complementos y soporte de la resolución de problemas

4.3.3. Resultados obtenidos / esperados

Los estudios previos y la realidad en las aulas donde se ofrece la asignatura, se ha descubierto que el rendimiento académico de los estudiantes, no es satisfactorio. Es de suma urgencia llevar a cabo tareas específicas con la intención de atenuar esta situación (Zúñiga, Rosas, Fernández & Guerrero, 2014). Es así que se proponen los siguientes objetivos a corto, mediano y largo plazo:

- Indagar los factores académicos que influyen en el desempeño de los alumnos ingresantes a la UNSL en lo que respecta al aprendizaje de la programación en carreras de Ciencias e Ingeniería.
- Promover nuevas estrategias didácticas para abordar la resolución de problemas en la enseñanza de la programación.
- Contribuir con la producción de nuevos recursos educativos para el aprendizaje de la programación potenciando el desarrollo de las habilidades y competencias que forman parte del pensamiento computacional.
- Desarrollo y adecuación de un entorno integrado de desarrollo flexible y apropiado para la enseñanza inicial de la programación basado en la resolución de problemas y que incorpore soporte al aprendizaje colaborativo.

4.4. Enseñando a programar: un camino directo para desarrollar el pensamiento computacional

4.4.1. Introducción

La enseñanza y el aprendizaje son dos las caras de una misma moneda, por el lado de la enseñanza como actor tenemos al profesor y por el otro lado tenemos al alumno, es claro que los términos no se asemejan pero se complementan el uno con el otro. (Compañ Rosique, et al, 2015)

Todo tutor tiene como objetivo el aprendizaje de sus estudiantes y para eso existen numerosas metodologías que son puestas en práctica, esto para conseguir que los estudiantes adquieran conocimientos, habilidades, destrezas.

La programación de computadoras es parte esencial de cualquier currículo de informática, es una disciplina en la que muy pocos estudiantes han tenido un contacto previo por ende le causa dificultad, entonces es importante reflexionar acerca de ¿Cómo se enseña?, ¿Por dónde se empieza?, ¿Qué paradigma se debe utilizar?, ¿Qué lenguaje de programación se debe emplear?, ¿Cómo se deben orientar los ejercicios? y ¿Cómo se imparten las clases teóricas?, aparte del paradigma utilizado, los estudiantes deben desarrollar algoritmos para resolución de problemas. (Compañ Rosique, et al, 2015)

La UNESCO en 1996 realiza un informe, coordinado por Jacques Delors, que determina los cuatro pilares de la educación, los cuales son:

- Aprender a conocer, la actividad más habitual de la enseñanza en la que el tutor le trasmite al estudiante sus conocimientos.
- Aprender a hacer, el estudiante afronta determinadas tareas académicas.
- Aprender a vivir juntos, aprendiendo de sus colegas y desarrollando valores sociales, además ampliar la percepción con sus colegas de forma mutua, sin perder las ideas propias.
- Aprender a ser, se basa en el desarrollo de la personalidad, la madurez y el compromiso.

Es enriquecedor para un estudiante de informática, que en la asignatura como introducción a la programación adquiera estas destrezas. La forma de impartir la asignatura, resultará en el desarrollo del pensamiento computacional.

El carecer el estudiante de hábitos de estudio y aprendizajes adecuados hace mucho más difícil la tarea del tutor en cualquier área, algunas dificultades son, la disponibilidad de tiempo para impartir los saberes, otra es la dificultad de la materia, ya que el estudiante no posee pre-saberes, desconociendo la dinámica para su aprendizaje, además es una asignatura que se aborda desde la objetividad del pensamiento computacional, donde las soluciones a los problemas se representa como una serie de pasos(algoritmo). (Compañ Rosique, et al, 2015)

Como lo menciona Compañ Rosique, et al, (2015), “hay etapas que se deben tener en cuenta en el aprendizaje y la enseñanza, hay que aprovecharlas para lograr que el estudiante aprenda. Estas fases o etapas se han concretado en”:

- Oír: Asistir a la definición que el tutor hace de una facultad para aprenderla.
- Ver: Percibir algo con cualquier sentido o con la inteligencia.
- Hacer: Ejecutar, poner por obra una acción o trabajo.

4.4.2. Oír versus escuchar

Una de las características de la enseñanza es impartir clases, las impartidas por un maestro o tutor sigue siendo el procedimiento más usado en educación universitaria a pesar de ser muy criticada por: sujetarse a lo que el tutor conoce del tema, favorece el ritmo de aprendizaje del estudiante, etc. Posee ventajas como información rápida y sustancial, brinda confianza al alumno, ayuda a comprender temas complejos, reúne información precisa de diferentes fuentes, etc. (Compañ Rosique, et al, 2015)

Según Compañ Rosique, et al, (2015), el desarrollo de una buena lección consta de cuatro fases:

- Preparación y diseño: Formulación de objetivos, organización de los contenidos, preparación de actividades para el alumnado. Esta fase sería previa a la actuación en el aula.
- Introducción: Ganar la atención de la audiencia, establecer relaciones con el grupo, despertar el interés, motivar hacia la tarea, presentación de objetivos, resumen general introductorio, etc.

- **Cuerpo:** Estructuración del contenido, mantenimiento de la atención y del interés, velocidad y ritmo adecuados, expresividad, etc.
- **Conclusión:** El objetivo es intensificar la retención, énfasis en las ideas principales, preguntas, resumen, etc.

Es esencial que el alumno este motivado, esta motivación puede ser motivación interna (el estudiante tiene interés propio por aprender), motivación externa (es obligado a estudiar). Aunque el tutor siga cada una de las fase esto no garantiza la asimilación del conocimiento por parte de los estudiantes, ya que como se sabe oír no significa escuchar. En las clases interactivas se brinda gran variedad de ejercicios, trucos, comparaciones de situaciones de la vida diaria, buscando analogías entre concepto de programación. (Compañ Rosique, et al, 2015)

Un método para la reflexión de cualquier estructura o tipo de datos, es mostrarles a los estudiantes una sintaxis de la estructura, y esperar una reflexión del cómo funciona y para que, luego explicar cómo y dónde se utiliza. Un ejemplo de esta metodología es enseñarles una pequeña parte de un lenguaje algorítmico, que tenga cercanía al lenguaje natural y preguntar ¿Qué hace? ese fragmento de código, la respuesta más común sería que los estudiantes explicaran el cómo, en la mayoría de los casos el estudiante interpreta y describe el fragmento, pero con otras palabras.

4.4.3. Ver versus observar

La visualización de programas son metodologías gráficas para ayudar a entender el funcionamiento de un programa de computadora, es de gran ayuda al alumnado a mejorar su

conocimiento acerca de la realización de un programa, no hay que pensar que este tipo de métodos va ser solución a todos los problemas y que el aprendizaje será más eficiente. (Compañ Rosique, et al, 2015)

Se entiende que una representación gráfica de un programa, especialmente los que constan con animaciones, es mejor que el código escrito, aunque no siempre es así, depende de los estilos de aprendizaje.

¿Es lo mismo observar que ver? , si se busca en el diccionario los terminos, vemos que su definición se asemeja en su significado. No obstante se considera que tienen diferencias, ver es una cualidad, donde no se hace ningún esfuerzo, por otro lado el termino observar, es una actividad en la que hay un esfuerzo en asimilar lo que se está viendo. Los métodos de visualización son de observar, por lo tanto son útiles en el proceso del aprendizaje, se consideran eficaces para el estudiante con poco conocimiento de una asignatura. (Compañ Rosique, et al, 2015)

4.4.4. Hacer equivale a practicar

Practicar es realizar ejercicios o actividades para el enriquecimiento del aprendizaje y la mejor manera de aprender es compartir los conocimientos con otra persona, De hecho cuando un estudiante está realizando un algoritmo le está enseñando al ordenador como realizar una tarea, lo que quiere decir que el estudiante tiene claridad en lo que intenta explicar.

Es de gran ayuda entender que la práctica hace al maestro, el tiempo utilizado en analizar un problema es esencial, para no resolverlo de manera incorrecta y para no aceptar la solución propuesta por el tutor como única y la más certera. El proceso de razonamiento y de búsqueda nunca es lineal. (Compañ Rosique, et al, 2015)

El objetivo de una asignatura como introducción a la programación no es que los algoritmos, estén perfectamente escrito o que la sintaxis sea impecable, el objetivo es que consigan estructurar la solución a un problema.

Lo importante es identificar, analizar e implementar soluciones con el objeto de encontrar más eficiente y efectiva a un algoritmo, tanto en el número de pasos como en los recursos empleados, una manera de cumplir este objetivo son los diagramas de flujo mudo, estos indican de manera gráfica la estructura lógica en la solución de un problema determinado. (Compañ Rosique, et al, 2015)

No obstante en la práctica los estudiantes prefieren desarrollar la solución de un problema en lenguaje de programación para posteriormente dibujar el diagrama mudo, no se ha podido modificar este mal hábito.

Para aumentar el atractivo de las prácticas se plantea el aprendizaje mediante el juego, no es una técnica nueva, pero es muy útil para motivar a los estudiantes mediante la creación, confesión o utilización de estos mismos, otra competencia de gran importancia, que deben desarrollar los estudiantes, es el trabajo en equipo. (Compañ Rosique, et al, 2015)

4.4.5. Infundiendo buenos hábitos

Es importante entender que al desarrollar un algoritmo el objetivo principal no es que funcione, sino que tenga una estructura correcta, un despeje de parámetros adecuado, emplear de manera correcta las variables.

Al escribir la frase la ventana hesta havierta, al leerla se comprende el mensaje transmitido, pero ortográficamente no es correcta, si se tratase de una asignatura como lengua, no se admitiría. Esto mismo procuramos que los estudiantes hagan con sus algoritmos, haciendo algo similar, la gramática en lengua es como la estructura en un programa, a nivel de algoritmia, como a nivel de datos. (Compañ Rosique, et al, 2015)

Los estudiantes poseen costumbres que muchas veces son difíciles de erradicar, en ocasiones escriben el código sin sangrar, justifican que es para ir más rápido, luego lo sangran pero no siempre de manera correcta. No entienden que un correcto sangrado en el código, les ayuda a legibilidad del mismo.

En este tipo de prácticas los estudiantes aprenden que cumplir las especificaciones es fundamental, no antes de ponerse a codificar. Ya finalizado el programa se dan cuenta que no funciona debido a que no siguieron de manera rigurosa las instrucciones, conlleva a no poder enlazar su trabajo con el del resto del equipo, y hay que tener en mente que los proyectos informáticos son de tal complejidad que es indispensable un equipo de desarrollo. (Compañ Rosique, et al, 2015)

4.4.6. Conclusiones

Una asignatura de introducción a la programación de ordenadores es el vehículo perfecto para desarrollar habilidades de pensamiento computacional ya que implica la resolución de problemas haciendo uso de conceptos informáticos.

A nuestro entender, enseñar programación no consiste en enumerar una serie de estructuras de programación indicando para que sirve cada una de ellas. Es mucho más que eso, se trata de que el estudiante aprenda a pensar, a analizar una situación y a diseñar el método de resolución más adecuado, dejando al margen el lenguaje de programación. Se trata de un objetivo muy complejo. Para cualquier persona diseñar la solución a un problema requiere de un esfuerzo importante de abstracción, aún más si tiene que expresarla en forma de un algoritmo.

Es fundamental que los estudiantes comprendan que la habilidad de escribir programas correctos, eficientes, bien organizados y adecuadamente documentados es un requisito esencial para cualquier titulado en informática, porque programar es mucho más que conocer un determinado lenguaje. Hemos presentado en este trabajo nuestras reflexiones acerca de cómo iniciar con éxito a un estudiante en el campo de la programación de ordenadores. Aunque nuestra experiencia se basa en enseñar programación a futuros ingenieros en informática, la misma es válida para cursos de introducción a la programación tanto de niveles no universitarios como de universitarios de otras ramas (Compañ Rosique, et al, 2015, p 12)

4.5. Pensamiento computacional a través de la programación: paradigma de aprendizaje.

En este artículo diseñan e implementan proyectos de programación en el lenguaje Scratch, se describe la necesidad y el propósito del pensamiento computacional en la escuela y nombran algunos recursos educativos usados para el desarrollo del pensamiento, además de brindar una alternativa a la implementación de este tema en la aulas de primaria y secundaria., la metodología usada en este proyecto implementa el construccionismo para el desarrollo del pensamiento computacional en Scratch , este lenguaje tiene sus raíces en la enseñanza del lenguaje LOGO, el rol que desempeña el maestro en clase usando esta metodología es apoyar al estudiante , promocionando la colaboración, el debate, motivando la creatividad en el estudiante creando nuevo conocimiento. (Basogain Olabe, et al , 2015)

Tabla 1 Cuadro comparativo de los artículos más relevantes del estado del arte.

Título del artículos	Objetivos	Componentes Del Pensamiento Computacional Tratados En El Artículo	Resultados Y Discusiones
Pensamiento computacional: Una nueva alfabetización digital	Este trabajo tiene como objetivo una recensión de las formas de pensamiento que se han manifestado y han sido estudiadas como útiles en la resolución de problemas en el ámbito cognitivo	Metacognición, Sinéctica, Análisis descendente, Análisis ascendente, Recursividad, Método por aproximaciones sucesivas(Ensayo – error),Heurística, Iteración, Pensamiento divergente, Creatividad, Resolución de problemas, Pensamiento abstracto, Métodos colaborativos Patrones	La codificación es una competencia compleja o un conjunto de competencias. Faltaría definir qué es codificación en un sentido pluridisciplinar que implique a profesionales de la psicología del Aprendizaje y del Desarrollo, los especialistas en Educación
Desarrollo del pensamiento computacional con Scratch	Diseñar una estrategia para desarrollar el pensamiento computacional, utilizando actividades interactivas en Scratch.	Recopilar Datos, Analizar Datos, Representar Datos, Descomponer Problemas, Abstracción, Algoritmos, Procedimientos, Automatización, Simulación y Paralelismo	Los resultados muestran que los recursos de actividades interactivos con Scratch, logran el objetivo planteado, motivando al usuario, y alcanzando al término de la experiencia niveles más altos del pensamiento computacional.
El desarrollo del pensamiento computacional para la resolución de problemas en la enseñanza inicial de la programación	El objetivo de esta investigación es promover nuevas propuestas didácticas que permitan afrontar la problemática recurrente en relación con el desempeño académico de los alumnos ingresantes a la UNSL en lo que respecta al aprendizaje de la programación en carreras que la	Abstraer, Modelar, Analizar, Reflexionar, Codificar, Medir y Argumentar	Factores académicos que influyen en el desempeño de los alumnos ingresantes a la UNSL en lo que respecta al aprendizaje de la programación en carreras de Ciencias e Ingeniería Estrategias didácticas para abordar la resolución de problemas en la enseñanza de la programación.

	incluyan como asignatura en el bloque de las ciencias básicas de sus correspondientes planes de estudio.		
Enseñando a programar: un camino directo para desarrollar el pensamiento computacional	Este trabajo tiene como objetivo es orientar al estudiante en el campo de la programación de computadores	Resolución de problemas, algoritmia, abstracción, modelado, simulación.	Una asignatura de introducción a la programación de ordenadores es el vehículo perfecto para desarrollar habilidades de pensamiento computacional ya que implica la resolución de problemas haciendo uso de conceptos informáticos Es fundamental que los estudiantes comprendan que la habilidad de escribir, programas correctos, eficientes, bien organizados y adecuadamente documentados es un requisito esencial para cualquier titulado en informática, porque programar es mucho más que conocer un determinado lenguaje.
Pensamiento computacional a través de la programación: paradigma de aprendizaje	Presentar el concepto del Pensamiento Computacional y su integración las aulas a través del diseño e implementación de proyectos de programación.	Reformulación de un Problema , Pensamiento Recursivo, Paralelismo, Razonamiento Heurístico, Abstracción y Descomposición	El 57,88% sabía programar antes de iniciar el curso, y el 42,16% no sabía programar. Las dos terceras partes del alumnado del curso realizan su profesión en el ámbito de la docencia, mientras que el otro tercio (30,14%) su profesión no tiene que ver con la docencia.

Elaboración:

propia

5. METODOLOGÍA

La presente investigación se desarrollará a través de la metodología holística que hace referencia a una actitud del investigador hacia el proceso de generación del conocimiento. Una actitud de apertura y de búsqueda de una comprensión integradora de su evento de estudio (Hurtado, 2000), la holística permite al investigador acercarse y comprender la realidad, ayudando a la adquisición del conocimiento, para así estipular nuevas hipótesis, ideas y valores, es por esto que la investigación holística representa una oportunidad para la innovación y la generación de más conocimiento. Es importante mencionar que en holística se trasciende del concepto de paradigma al de sintagma, puesto que las propuestas que surgen del paradigma tienden a ser parciales, mientras el sintagma “integra un conjunto de eventos en un todo con sentido unitario” (Hurtado, 2000, p. 11), es así como un sintagma orienta la acción hacia la simultaneidad. La investigación holística:

La holística presenta la investigación como un proceso global, evolutivo, integrador, concatenado y organizado. Trabaja los procesos que tienen que ver con la invención, con la formulación de propuestas novedosas, con la descripción y la clasificación, considera la creación de teorías y modelos, la indagación acerca del futuro, la aplicación práctica de soluciones y la evaluación de proyectos, programas y acciones sociales, entre otras cosas (Hurtado, 2000, p. 14).

5.1. Tipo de investigación

El tipo de investigación en el presente trabajo es analítico, ya que ésta conlleva un “procesamiento reflexivo, lógico, cognitivo que implica abstraer pautas de relación internas de un

evento, situación, fenómeno, etc.” (Hurtado, 2000, p. 269), es decir, se pretende implementar una serie de guías que permitan construir e influir en el pensamiento de los estudiantes de los grados 10° y 11° de una Institución Educativa previamente seleccionada, con el fin, de aportar conceptos a cerca del pensamiento computacional, de manera que, los estudiantes se apropien y asimilen dichos conocimientos, modificando y moldeando así, su estructura mental, enriqueciendo su manera de reflexionar, pensar y de actuar frente a las problemáticas que se le presentan en su vida cotidiana.

Por medio de la herramienta robótica LEGO, se pretende que el niño adquiera los conceptos necesarios para incentivar el pensamiento lógico, algorítmico, cognitivo, que son fundamentales para el desarrollo del pensamiento computacional, lo anterior, podrá ser observado y analizado haciendo uso de una evaluación inicial y una evaluación final, que permitirá determinar los avances logrados a través de la implementación de esta estrategia, es por ello que Hurtado (2000) manifiesta que “cuando se analiza un objeto, se puede agrupar sus cualidades con base en el seguimiento de una función específica. Analizar algunas veces implica corrientes de acción” (p. 270). La finalidad es lograr que los estudiantes desarrolle el pensamiento computacional, que les facilitará buscar soluciones a los problemas, esto estimulará su forma de pensar y hará que refuerce sus funciones cognitivas.

Para comprender un poco más en qué consiste el tipo de investigación analítica, se encuentra que:

Toda investigación analítica requiere de una descripción previa sobre la cual se hace el análisis; es por eso que muchas investigaciones analíticas son de carácter documental. Sin embargo cuando no hay descripciones previamente realizadas, es

posible hacer investigaciones analíticas situacionales, cubriendo los estadios previos, particularmente el estado descriptivo (Hurtado 2000, p. 273).

Lo anterior, se refiere al hecho de que la investigación puede ser de tipo documental o situacional, en el presente caso es de carácter situacional, ya que, conlleva a analizar a los estudiantes en el desarrollo de las actividades propuestas y dependiendo de la situación expuesta en las guías de aprendizaje ellos buscarán una solución a un problema, desarrollando así, nuevos conocimientos necesarios para resolver el resto de las actividades.

5.2. Diseño de investigación

El diseño de investigación, es el estudio analítico situacional, que permite “analizar los elementos que intervienen en una situación determinada” (Hurtado, 2000, p 279). Pero en este caso específicamente, se obtendrán los datos de manera directa a través de la observación directa, ya que, será el investigador, quien utilizando un instrumento de observación, registrará los hechos que se desarrollarán en el lugar en que se lleven a cabo las actividades, y es así, como Hurtado (2000) comprende el estudio analítico situacional.

5.3. Población

La población de la presente investigación corresponde a estudiantes del primer semestre de ingeniería de sistemas de la Universidad de Pamplona, para ello, se tomó una muestra de seis estudiantes.

5.4. Técnicas e instrumentos

En la presente investigación, se considera que las técnicas e instrumentos a usar para estimular el rendimiento, desarrollo, y afianzamiento de los conocimientos correspondiente al pensamiento computacional son las siguientes:

- Un test inicial que valora el nivel de conocimiento que tiene el estudiante en los conceptos referentes al pensamiento computacional.
- 6 guías metodológicas que ayudaran al aprendiz a afianzar los conceptos correspondientes al pensamiento computacional.
- Un test final que permite identificar el nivel de conocimiento adquirido por el estudiante a lo largo de la implementación de las guías.

Se hará uso de la observación y el análisis en la aplicación de las guías metodológicas, prestando atención a las dificultades que tienen los estudiantes en el desarrollo de la estrategia, anotando y sugiriendo maneras de resolver los problemas, para así lograr consolidar el aprendizaje y una retroalimentación tutor-aprendiz, además de tomar fotografías, algunos apuntes que facilite el estudiante del desarrollo de las actividades propuestas en las guías, estas serán las evidencias de la aplicación de la estrategia.

Dentro de los instrumentos se cuenta con 2 set LEGO Mindstorm proporcionado por el programa de ingeniería de sistemas de la Universidad de Pamplona, este juguete es usado para el aprendizaje de la programación en estudiantes, donde el set permite construir, programar y testear soluciones a problemas en la vida real con la tecnología LEGO.

6. DISEÑO ESTRATÉGICO

El presente capítulo de diseño estratégico permite tener una concepción de cada uno de los recursos estipulados en las guías metodológicas, utilizadas para impartir y orientar a los estudiantes en los conceptos referentes al pensamiento computacional.

6.1. Generalidades

Teniendo en cuenta el marco conceptual, y tomando como referencia las investigaciones mencionadas en el estado del arte acerca del pensamiento computacional y las metodologías de aprendizaje, se concluye que la gran mayoría de estudiantes no captan el conocimiento o la información de la misma forma. Un estudiante al cursar una materia donde la comprensión lectora, el análisis de problemas y la lógica juegan un papel trascendental en el éxito de lograr buen aprendizaje, enfocando un curso como programación donde el estudiantes se enfrenta a dificultades de un lenguaje, que difiere del lenguaje natural al que están acostumbrado, donde apropiarse de la sintaxis, la lógica, estructurar la forma en que se aborda un problema para resolverlo y la interacción con entornos de desarrollo integrados IDE, es un proceso agotador para el estudiante, por no decir que muchas veces conlleva a abandonar el aprendizaje. El pensamiento computacional es interpretado como la resolución de problemas usando los recursos y conceptos que ofrece la computación, el análisis, la interpretación, son las diferentes líneas que se abordan bajo este concepto, orientando el pensamiento para que cualquier persona sea capaz de dar solución a diferentes situaciones, ya sea en el ámbito de la vida diaria y/o laboral, aplicando herramientas informáticas.

6.1.1. Teoría de Aprendizaje Escogida para esta Investigación

Las teorías de aprendizaje constructivista, conductistas, cognitivista, conectivista, son de gran importancia para el aprendizaje, cada una de ellas fue o son aplicadas en escuelas y universidades en la formación y enseñanza de sus estudiantes. Recordemos que mencionan cada una de estas teorías.

El constructivismo, es una teoría que ayuda al estudiante a interiorizar un pensamiento, reamoldar, o transformar la información. Esta transformación ocurre gracias a la adquisición de nuevo conocimiento provocando el levantamiento de nuevas estructuras cognitivas que permiten enfrentarse a situaciones iguales o parecidas en la realidad (Trujillo, 2017).

El conductismo, el aprendizaje es un cambio en el comportamiento en función del entorno, por consiguiente el aprendizaje es el resultado de la asociación de estímulos y respuestas, el conductismo ve al aprendiz como un sujeto cuyo desempeño, aprendizaje puede ser transformado, basta con implementar buenos insumos educativos para lograr un cometido y una conducta deseable. El maestro ve al estudiante conductista como una hoja en blanco que, cumple órdenes y obedece, requiere constante aprobación, es dependiente del tutor, su existencia es pasiva en el proceso de enseñanza- aprendizaje, realiza tareas en las cuales el comportamiento pueda ser observado, medido, evaluado directamente. (Valdez, 2012)

El cognitivismo, este paradigma está orientado a varias dimensiones de lo cognitivo (atención, percepción, memoria, inteligencia, lenguaje, pensamiento, etc.), esta teoría afirma que el aprendizaje depende de lo que el sujeto observe de acuerdo a sus competencias y percepciones previas o antecedentes, las cuales tiene un comportamiento dependiendo del individuo, influenciado por sus actitudes y motivaciones. (Trujillo, 2017)

El conectivismo, es una teoría del aprendizaje de la era digital, que toma como referente, el conductismo, el cognitivismo y el constructivismo, analizando sus limitaciones para ilustrar el efecto de las tecnologías sobre la manera en que actualmente vivimos, nos comunicamos y aprendemos. El conectivismo enlaza ideas, vincula conocimiento, crear nuevas conexiones, forma patrones y mantiene e incrementa las redes personales de aprendizaje. (Altamirano, Becerra & Nava, 2010)

Analizando cada de las teorías expuestas podemos establecer que la teoría constructivista se convierte en la más acertada para aplicar en el desarrollo del pensamiento computacional, ya que ofrece una forma en que el aprendiz construye estructuras mentales propias, esto generando nuevos conocimientos, analizando un problema, entendiéndolo y generando posibles soluciones, además de implementar una herramienta informática, que pueda apoyar el proceso de aplicación de la solución.

Teniendo en cuenta las investigaciones realizadas acerca de las teorías de aprendizaje existentes, se considera el paradigma constructivista como el más adecuado para realización de los recursos usados en el desarrollo e implementación de la estrategia, en pro de desarrollar el pensamiento computacional en los estudiantes. Se considera esta teoría ya que el aprendiz es capaz de adquirir el conocimiento a través del juego, construyendo soluciones a problemas planteados y adquiriendo nuevas experiencias de aprendizaje, reorganizando conceptos y reafirman saberes.

Tomando en cuenta el objetivo principal de este proyecto, que es el desarrollo del pensamiento computacional con LEGO Mindstorm, se puede mencionar que la teoría conductista es una teoría que no se ajusta a las necesidades de este proyecto, ya que esta ve al aprendiz como un sujeto al que hay que garantizarle el conocimiento, sin tener aspectos importantes del

entendimiento y saberes que son esenciales para el proceso de un buen aprendizaje, por otro lado tenemos la teoría conectivista que menciona que es la formación de conexiones en una red, explican el efecto de las tecnología en la forma que, nos comunicamos y aprendemos. La red tiene dos componentes: nodos y conectores. Un nodo es una entidad externa (personas, bibliotecas, organizaciones o cualquier tipo de información), el cual es el encargado de suministrar los conocimientos, los conectores son los individuos interesados en estos conocimientos. Esta red es dinámica e inteligente, el aprendizaje se fundamenta en la relevancia del conocimiento y el saber buscar la información.

La teoría conectivista no es tomada en cuenta en esta investigación debido a que los conectores o individuos son máximo 6 estudiantes y la única fuente de conocimiento es el tutor no tomando en cuenta el internet, el tutor suministra recursos para el desarrollo del aprendizaje, además que el recurso físico usado en esta investigación (kit LEGO Mindstorm EV3) no es asequible a toda persona debido al costo monetario, esto impide que los conocimientos y la información sean impartidos de manera virtual.

El papel del tutor es orientar y asesorar al estudiante en las diferentes actividades durante el desarrollo e implementación del proyecto. El conectivismo es teoría de aprendizaje que es útil a la hora de impartir conocimientos a través de plataformas donde el docente y el estudiante interactúan de manera similar a la formación presencial usando programas informáticos, como es el caso de Skype y Facetime, además de compartir información y trabajar en línea con herramientas como Google Drive, Wikispaces y Dropbox entre otras.

En el caso del cognoscitivismo, se considera esta teoría complementaria a la teoría constructivista, esta menciona que el individuo es un ser socio-afectivo, dependiente de sus emociones, competencias y percepciones previas o antecedentes, las cuales tiene un comportamiento dependiendo del individuo, influenciado por sus actitudes y motivaciones.

6.1.2. ¿Qué es LEGO Mindstorm EV3?

Lego Mindstorm EV3 es un kit de desarrollo para realizar montajes robóticos donde el usuario puede diseñar rutinas e implementarlas en el robot para solucionar problemas, en esta investigación este recurso servirá de puente entre lo teórico y lo práctico, facilitando el aprendizaje de los conceptos fundamentales del pensamiento computacional.

6.2. Desarrollo de la estrategia

El desafío de implementar la presente estrategia en estudiantes, diseñando unas guías taller dirigidas al desarrollo de los componentes del pensamiento computacional, complementando el aprendizaje de estos con el kit de Lego Mindstorm realizando montajes robóticos que apliquen los conceptos y soluciones planteadas por los estudiantes. La aplicación de las guías será a un grupo de estudiantes pequeño en una prueba piloto, la cual espera despertar, desarrollar, fortalecer y consolidar los conceptos a bordados en cada guía. El director del programa de Ingeniería de Sistemas realizo la gestión para obtener la información de contacto de los estudiantes inscritos y matriculados para el primer semestre de 2019, se contactaran mínimo 4 y máximo 6 estudiantes, se considera esta cantidad a causa de que solo se cuenta con dos set robóticos LEGO Mindstorm, consecuentemente el estudiantado se dividirá en dos grupos donde cada grupo tendrá un set LEGO Mindstorm, esto facilita proceso enseñanza-aprendizaje de los aprendices que interactúan con la herramienta..

Los aprendices fueron seleccionados de acuerdo a las condiciones de localización, se contacta vía telefónica únicamente a los estudiantes que se encuentran radicados en la ciudad de Pamplona, ya que para el mes de abril del 2019 la universidad se encuentra en periodo de

vacaciones. Una vez contactados los estudiantes se les ofrece participar en un curso libre de pensamiento computacional con lego Mindstorm.

La primera fase para el desarrollo del curso es el diseño de una test inicial que tiene con objetivo medir a los estudiantes participantes en sus aptitudes de razonamiento matemático, lógica, comprensión lectora, solución de problemas entre otros.

En la segunda fase se diseñaran los recursos (guías) que se aplicaran en cada encuentro con los estudiantes donde se irán integrando de forma ordenada y sincronizada los diferentes conceptos y componentes que permitan despertar y desarrollar el pensamiento computacional. Para el desarrollo de las diferentes prácticas, con el set de Lego Mindstorm se diseñaran dos robots genéricos que cubrirán las diferentes necesidades de cada práctica.

Los recursos diseñados se dividen en 3 segmentos, cada uno de ellos está orientado a despertar, fortalecer, desarrollar el pensamiento computacional en el estudiante, a continuación se describen:

Componentes Teóricos: Cada recurso se diseña para abordar un conjunto de conceptos fundamentales del pensamiento computacional, en este ítem se abordaran, desarrollaran y se buscara que cada estudiante construya su pensamiento de forma tal que, reorganice, aprenda los conocimiento relacionados al pensamiento computacional.

Recursos Lego Mindstorms: El apoderamiento y comprensión de los conceptos teóricos serán reforzados con los diferentes bloques de programación que ofrece Lego, en este ítem serán analizados en su estructura y funcionamiento.

Actividades Evaluativas: Se refieren a las actividades propuestas por el tutor, para la enseñanza y aprendizaje, de los conceptos pertinentes al pensamiento computacional, entre ellas se encuentran:

- **Talleres Prácticos:** Ayudan a afianzar los conocimientos pertinentes al componente teórico, su puesta en práctica requiere del uso del kit LEGO Mindstorm Ev3, en la enseñanza para despertar, desarrollar, fortalecer y consolidar los conceptos relacionados al pensamiento computacional.
- **Actividades Propuestas:** Refuerzan los conceptos asimilados en la teoría y los talleres prácticos explicados en las guías, incluyen las competencias enfocadas al pensamiento computacional.
- **Actividades de análisis:** Se realiza con el propósito de estudiar, ponderar, valorar y concluir respecto al desempeño del estudiante en la elaboración y realización de los talleres prácticos y actividades propuestas.
- **Retos:** Se plantean actividades extras, donde los estudiantes deben resolver una serie de nuevos problemas aplicando lo aprendido, afianzando, reforzando y construyendo el conocimiento con base al pensamiento computacional.

6.2.1. Desarrollo del pensamiento computacional

Las áreas que componen el pensamiento computacional deben ser abordadas desde diferentes puntos de vista, cada guía contiene apartados que le permiten al estudiante la asimilación y aplicación de conceptos relacionados al pensamiento computacional, esto le permite analizar problemas planteados y hallar la respuesta indicada.

Las competencias y conceptos relacionados al desarrollo del pensamiento computacional en las personas consiste en la resolución de problemas, el diseño de los sistemas, y la comprensión de la conducta y de las actividades humanas haciendo uso de conceptos y

procedimientos básicos para el trabajo y la elaboración de programas y algoritmos en la informática, tomando en cuenta las habilidades específicas y técnicas necesarias para estos objetivos, que constituyen la base de la cultura digital. Estas habilidades y métodos son: análisis ascendente, análisis descendente, heurística, pensamiento divergente, creatividad, resolución de problemas, pensamiento abstracto, recursividad, iteración, métodos por aproximaciones sucesivas (Ensayo – error), métodos colaborativos, patrones, sinéctica, metacognición y cinestesia, entre otros. (Ros, 2019). En este proyecto estos componentes se encuentran incluidos en las actividades evaluativas, donde los estudiantes a través de los ejercicios y problemas planteados reflexionan, refuerzan y organiza sus conocimientos, además de estimular sus habilidades en las diferentes áreas del pensamiento. Los ejercicios se orientaran a que el estudiante de manera divertida use el kit LEGO Mindstorm, para aprender y practicar los conceptos establecidos en el componente teórico, los cuales son fundamentales para el desarrollo del pensamiento computacional. El diseño de las guías tiene como objetivo estimular en los estudiantes el pensamiento algoritmo, pensamiento abstracto, pensamiento lógico matemático, entre otros, por medio de los talleres prácticos, actividades propuestas, actividades de análisis y los retos.

Las actividades de cada guía, estimulan el pensamiento computacional, debido a que los estudiantes a través de los problemas, analizan, reflexionan y razonan posibles soluciones estimulando y fortaleciendo el pensamiento algorítmico (Considerando que el estudiante debe buscar soluciones a problemas postulando una secuencia de pasos organizados, que al final le permiten hallar la solución al problema planteado), la comprensión lectora (Las guías tienen componentes teóricos que requieren que el estudiante las interprete, a través de la comprensión él puede ejecutarlas.), pensamiento divergente (El estudiante debe desarrollar ideas creativas mediante la exploración de muchas posibles soluciones a un problema planteado), pensamiento convergente(De todas las posibles soluciones que al estudiante se le ocurran, seleccionara una de

ellas como la correcta y la más indicada, dando solución al problema), pensamiento lógico matemático,(La manera en que cada uno de los estudiantes resuelva los problema determinara la lógica y la utilización de los recursos), el trabajo colaborativo (El estudiante debe interactuar con sus compañeros de grupo para la solución de los problemas planteados en las guías). Los métodos de Ensayo y error, Colaborativos, Aproximaciones sucesivas son de gran importancia en la aplicación de la solución al problema, porque el estudiante analiza, reflexiona y corrige, llegando a la repuesta indicada.

A continuación se describe el objetivo de cada uno de los recursos, para lograr el desarrollo del pensamiento computacional. Se consideró abordar los siguientes temas y prácticas para lograr culminar el curso, en un máximo de 9 sesiones.

Tabla 2 Descripción de los recursos del curso pensamiento computacional con LEGO Mindstorm

Guías	Descripción Temática	Desarrollo Pensamiento Computacional
Guía #1	<ul style="list-style-type: none">  Algoritmia, variables y contantes  Introducción al uso y manejo del Bloque Ev3  Bloques Programables en el software LEGO Home Edition.  Taller práctico I: Introducción a Software LEGO Home Edition  Taller práctico II: Movimientos del robot L1  Taller práctico III: Introducción a la programación interna del Bloque Ev3 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Desarrollo comprensión lectora <input checked="" type="checkbox"/> Desarrollo pensamiento algoritmo <input checked="" type="checkbox"/> Desarrollo resolución de problemas. <input checked="" type="checkbox"/> Desarrollo del pensamiento convergente y divergente. <input checked="" type="checkbox"/> Trabajo en equipo
Guía #2	<ul style="list-style-type: none">  Operadores, Estructuras de selección  Bloques Programables en el software LEGO Home Edition.  Taller práctico I: Uso operadores aritméticos en LEGO Home Edition  Taller práctico II: Uso operadores relacional en LEGO Home Edition  Taller práctico III: Uso operadores lógicos en LEGO Home Edition  Taller práctico IV: Estructura de selección IF con LEGO Home Edition.  Taller práctico V: Uso del bloque texto  Taller práctico VI: Armando el robot L1 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Desarrollo comprensión lectora <input checked="" type="checkbox"/> Desarrollo motricidad fina. <input checked="" type="checkbox"/> Desarrollo del pensamiento algorítmico. <input checked="" type="checkbox"/> Desarrollo del pensamiento lógico matemático. <input checked="" type="checkbox"/> Desarrollo polimorfismo <input checked="" type="checkbox"/> Desarrollo descomposición <input checked="" type="checkbox"/> Desarrollo pensamiento divergente y convergente <input checked="" type="checkbox"/> Desarrollo resolución de problemas. <input checked="" type="checkbox"/> Desarrollo trabajo colaborativo
Guía #3	<ul style="list-style-type: none">  Ciclo repetitivos  Bloques Programables en el software LEGO Home Edition.  Taller práctico I: Ciclo while con LEGO Home Edition  Taller práctico II: Seguidor de color usando el ciclo while.  Taller práctico III: Seguidor de línea 1  Taller práctico IV: Contador de pulsaciones, usando el bloque variable. 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Desarrollo comprensión lectora <input checked="" type="checkbox"/> Desarrollo automatización <input checked="" type="checkbox"/> Desarrollo pensamiento algorítmico. <input checked="" type="checkbox"/> Desarrollo pensamiento lógico matemático. <input checked="" type="checkbox"/> Desarrollo pensamiento abstracto. <input checked="" type="checkbox"/> Desarrollo pensamiento divergente y convergente <input checked="" type="checkbox"/> Desarrollo resolución de problemas <input checked="" type="checkbox"/> Desarrollo trabajo colaborativo

Guías	Descripción Temática	Desarrollo Pensamiento Computacional
Guía #4	<ul style="list-style-type: none">  Bloque avanzados con LEGO Home Edition  Taller práctico I: Seguidor de línea 2.  Taller práctico II: Uso del bloque Acceso al archivo. 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Desarrollo comprensión lectora <input checked="" type="checkbox"/> Desarrollo algorítmico. <input checked="" type="checkbox"/> Desarrollo pensamiento lógico matemático. <input checked="" type="checkbox"/> Desarrollo divergente y convergente <input checked="" type="checkbox"/> Desarrollo resolución de problemas <input checked="" type="checkbox"/> Desarrollo trabajo colaborativo
Guía #5	<ul style="list-style-type: none">  Bloque avanzados con LEGO Home Edition, uso de arrays  Taller práctico I: Arreglo numérico  Taller práctico II: Estructura para el sensor táctil y el sensor de color.  Taller práctico III: Arrays de colores. 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Desarrollo comprensión lectora <input checked="" type="checkbox"/> Desarrollo pensamiento algorítmico. <input checked="" type="checkbox"/> Desarrollo pensamiento lógico matemático. <input checked="" type="checkbox"/> Desarrollo pensamiento abstracto <input checked="" type="checkbox"/> Desarrollo pensamiento divergente y convergente. <input checked="" type="checkbox"/> Desarrollo resolución de problemas <input checked="" type="checkbox"/> Desarrollo trabajo colaborativo
Guía #6	<ul style="list-style-type: none">  Bloques avanzados en LEGO Home Edition, uso del condicional switch.  Taller práctico I: Armando el robot L2  Taller práctico II: Contador y clasificador de colores 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Desarrollo comprensión lectora <input checked="" type="checkbox"/> Desarrollo pensamiento lógico matemático. <input checked="" type="checkbox"/> Desarrollo pensamiento abstracto <input checked="" type="checkbox"/> Desarrollo motricidad fina <input checked="" type="checkbox"/> Desarrollo resolución de problemas <input checked="" type="checkbox"/> Desarrollo trabajo colaborativo

Elaboración: Propia

6.2.2. GUIA 1. (Primeros Pasos En LEGO Mindstorm Ev3)

El objetivo de la guía es acercar al estudiante a los primeros conceptos fundamentales de algoritmia y a desarrollar una primera practica usando el set de Lego Mindstorms, desarrollando los conceptos pertinentes al pensamiento computacional. A continuación se explicara, el componente teórico, los talleres prácticos, actividades evaluativas, para esta guía

6.2.2.1. *Componente Teórico*

En su parte inicial se abordara la conceptualización del set de Lego Mindstorms donde los estudiantes tendrán una clara idea del porqué, el que y el para que de esta herramienta, posteriormente se entrara en materia con el tema de algoritmia, además se explicaran los conceptos de variables y constantes. Después de la definición de cada concepto se plantearan ejemplos ilustrativos de su uso y actividades propuestas que ayudaran al estudiante a apropiarse de la temática, se realizan actividades de reflexión, donde el estudiante piensa y razona sobre los conceptos aprendidos, con estas actividades se analiza la teoría con detenimiento, estudiándola, comprendiéndola y formando una opinión sobre lo aprendido, estas actividades ayudan al proceso de aprendizaje. Se les realizara a los estudiantes unas preguntas de análisis, para evaluar la comprensión de los conceptos explicados con anterioridad y saber su desempeño.

Recursos LEGO Mindstorms: En este ítem los estudiantes realizaran una lectura donde se describen los bloques programables utilizados para este taller, se definirá la funcionalidad y los modos de programar cada bloque usando el software LEGO Home Edition. Estos bloques se usaran en la solución de problemas planteados en los talleres prácticos, las actividades propuestas,

las actividades de reflexión y de análisis, con este apartado se integrara la parte teórica con la práctica.

6.2.2.2. Talleres prácticos

Taller práctico 1: Se describen los elementos con los que cuenta el bloque EV3, la interfaz de usuario y sus diferentes pestañas, botones e iconos, se presenta el robot L1 que el tutor debe entregar al estudiante correctamente armado y se hace una primera interacción entre el robot y el estudiante donde se familiarizan con la herramienta de trabajo . El robot tiene en su memoria un programa que fue diseñado con anterioridad por el tutor, con la ayuda de las instrucciones en la guía se le enseña a los estudiantes como navegar por la interfaz del bloque EV3 y ejecutar el programa.

Actividades Evaluativas: En este primer taller se desarrollara una actividad de análisis, la cual evalúa el nivel de comprensión de los estudiantes respecto a lo desarrollado en el taller práctico.

Taller práctico 2: En este taller se le enseñara al estudiante el manejo del software Lego Home Edition, con la lectura y el desarrollo de las instrucciones de la guía, el aprendiz será capaz de crear un proyecto, diseñarlo, programar una rutina y probarla en el robot L1.

Actividades Evaluativas :Al finalizar el taller práctico, los estudiantes deberán realizar las actividades de análisis, donde se evaluara su comprensión de los conceptos desarrollados en el taller práctico, para este taller se realizara un reto que consiste en programar un algoritmo diseñado por el tutor, a través del software LEGO Home Edition.

Taller práctico 3: En este taller se utilizará el bloque Ev3, programando directamente sobre él, la rutina o algoritmo, además los estudiantes aprenderán a navegar por las diferentes pestañas del bloque siguiendo los pasos propuestos en la guía. El algoritmo programado hace que el robot gire 360 grados por un determinado tiempo.

Actividades Evaluativas: Terminado el desarrollo del taller práctico los estudiantes deberán realizar las actividades de análisis, donde se le pregunta al estudiante sobre los conceptos vistos en el taller práctico.

6.2.3. GUIA 2. (Uso De Operadores Con LEGO Mindstorm Ev3 Y Estructuras De Selección)

En esta guía se presenta un acercamiento a los operadores y las estructuras condicionales, la clasificación y características de cada uno de ellos. A continuación se describirán, el componente teórico, los talleres prácticos, las actividades propuestas, actividades evaluativas, para esta guía

6.2.3.1. *Componente teórico*

En la parte inicial de la guía se abordará el componente teórico donde se les enseñará a los estudiantes la definición y el uso de los operadores, además se proporcionara ejemplos ilustrativos. Se les enseñara que los operadores se clasifican en tres, operadores aritméticos, relacionales y lógicos, se explicara cada uno de ellos con ejemplos, se diseñaran actividades propuestas y de análisis, donde el estudiante practicara el uso de los operadores, posteriormente se les enseñara el concepto de estructuras condicionales y se describirán sus tres variantes, la

estructura condicional simple, dobles y anidada, cada una de ellas tendrá ejemplos ilustrativos de su uso.

Recursos LEGO Mindstorms: Se realizara una explicación de los bloques programables usados en el taller práctico, para esta sesión los bloques que se usaran son el bloque matemática (los operadores aritméticos), bloque comparar (operadores relacionales), bloque operaciones lógicas (operadores lógicos), el bloque texto, bloque pantalla, bloque interruptor (condicional IF), bloque espera, bloque sensor táctil. La mayoría de bloques mencionados representan conceptos que los estudiantes leyeron al iniciar la guía, todos ellos son utilizados en el desarrollo de los algoritmos planteados en los talleres prácticos y demás actividades de la guía.

6.2.3.2. Talleres prácticos

Taller práctico 1: En este taller se explicara el uso de los operadores aritméticos, el estudiante ya debe tener un manejo básico en la programación de algoritmos desde el software LEGO Home Edition. Siguiendo las instrucciones los estudiantes harán un programa que suma dos números definidos, y el resultado se imprimirá en la pantalla del Bloque EV3.

Actividades Evaluativas: En este apartado los estudiantes realizaran las actividades propuestas, donde se evaluara la comprensión de los conceptos vistos al inicio de la guía, en este taller se solicitará a los estudiantes realizar un algoritmo en el software Lego home Edition para algunos operadores aritméticos como la multiplicación, la división y la resta. Además se realizarán actividades de análisis donde se observará si el estudiante tiene los conceptos claros, correspondiente a este taller.

Taller práctico 2: En esta práctica se explicará el uso de los operadores relacionales. Usando el bloque comparar y el software Lego Home Edition, los estudiantes realizarán un algoritmo que compara dos números para averiguar si son iguales o para saber qué número es mayor. Los estudiantes pueden elegir una de las seis comparaciones diferentes en la pestaña modo del bloque comparar. El resultado de las operaciones es verdadero o falso y se visualiza en la pantalla del bloque EV3

Actividades Evaluativas: En las actividades propuestas correspondientes a este taller, el alumno realizará cambios al algoritmo del taller práctico para entender su funcionamiento, además responderá unas preguntas de análisis, las cuales ayudarán a entender el nivel de comprensión de los conceptos referentes a este taller.

Taller práctico 3: En este taller se le adicionará y conectará al robot L1, el sensor táctil y el sensor de color, con ayuda de las instrucciones de la guía los estudiantes desarrollarán un algoritmo que usa el bloque sensor táctil y bloque sensor de color conectados a las entradas del bloque operaciones lógicas, mostrará como resultado verdadero o falso. El algoritmo programado en este taller usa el operador AND que solo arroja verdadero si y solo si el sensor táctil y el sensor de color son presionados. Las operaciones lógicas disponibles para el bloque operaciones secuenciales son AND, OR, XOR, NOT y el resultado mostrado en el bloque EV3 es, uno para verdadero y cero para falso.

Actividades Evaluativas: Terminado el taller práctico, los estudiantes realizarán actividades de análisis, donde se evaluará la comprensión del uso del bloque operaciones lógicas.

Taller práctico 4: En este taller se evidencia el uso de las estructuras condicionales, con la ayuda del sensor táctil y las instrucciones de la guía, el estudiante realizará un algoritmo que

permite saber cuándo el sensor táctil está presionado o no, el resultado será evidenciado cuando el Bloque Ev3 en caso verdadero diga (yes) y (No) en caso falso.

Actividades Evaluativas: Las actividad propuesta para este taller consistirá en que los estudiantes en compañía del tutor, desarrollaran un algoritmo utilizando el sensor de color que será capaz de mover en línea recta el robot L1 hasta encontrar una franja u obstáculo de color negro en la superficie que hará que el robot se detenga, esto usando el bloque interruptor.

En este apartado también se plantean actividades de análisis que ayudaran al tutor a entender el nivel de comprensión de los estudiantes con respecto a la conceptualización desarrollada en el taller.

Taller práctico 5: En esta práctica se destacara la importancia del bloque texto, en la impresión de mensajes en la pantalla del bloque EV3, los estudiantes realizaran un algoritmo que compara dos operaciones aritmética una que suma dos números y otra que resta dos números, definidos en las instrucciones de la guía, los resultados de dichas operaciones, serán comparados por un operador relacional (mayor que), el cual con la ayuda de una estructura condicional IF mostrara el resultado mayor , la respuesta será visualizada en forma de texto en la pantalla del Bloque EV3

Actividades Evaluativas: En este apartado el estudiante realizara primeramente una actividad práctica donde fortalecerá los conceptos explicados acerca del bloque texto y el bloque interruptor.

Se realizaran actividades de análisis donde se indagara acerca de los conceptos aprendidos, el nivel de comprensión de los estudiantes.

Taller práctico 6: En este taller se proporcionara al estudiante partes del robot L1, que el aprendiz con la ayuda de las instrucciones de la guía, terminara de armar.

Actividades Evaluativas: En las actividades propuestas para este taller, se desarrollara inicialmente un algoritmo ilustrado en una imagen, este debe ser programado desde la interfaz el bloque Ev3. Se realizaran preguntas de análisis acerca del algoritmo, que el estudiante debe resolver, para fortalecer los conceptos aprendidos.

Como una segunda actividad práctica se propone al estudiante desarrollar un algoritmo capaz de usar el robot L1, programando en el software LEGO Home Edition una secuencia algorítmica propuesta en una imagen ilustrativa. Se realizaran preguntas referentes a la ejecución del algoritmo, para saber el nivel de comprensión de los estudiantes.

Además en este taller se propone el desarrollo de un reto que ayudara a los estudiantes a fortalecer sus conocimientos de programación a través del desarrollo de un algoritmo desde la interfaz del bloque Ev3

6.2.4. GUIA 3. (Ciclo Repetitivo While Con LEGO Mindstorm Ev3)

En esta guía se enseñara el uso del ciclo repetitivo while o bucle, con el fin de que los estudiantes observen su funcionamiento y determinen los posibles usos en la solución de problemas, no obstante se pretende explicar los conceptos de contador y acumulador , con ejemplos de su uso. A continuación se describirán, el componente teórico, los talleres prácticos, actividades evaluativas, para esta guía

6.2.4.1. Componente Teórico

Para el inicio de la guía se pretende explicar el concepto de ciclo repetitivo planteando cuestiones o situaciones de la vida real, posteriormente se definirá ¿Que es un ciclo repetitivos while?, se explicara la importancia de los conceptos contador y acumulador, su relación con los ciclos repetitivos.

Recursos LEGO Mindstorms: En este apartado se explicara los bloques a usar en el desarrollo de la guía, estos son el bloque bucle (Representa al ciclo repetitivo while), el bloque interruptor, el bloque motor grande, el bloque variable.

En el desarrollo de la gran mayoría de talleres prácticos correspondientes a esta guía, se usan unas pistas de carreras diseñadas por el tutor, para la aplicación y puesta en marcha de los algoritmos que usan el sensor de color y el bloque bucle en su ejecución.

6.2.4.2. Talleres prácticos

Taller práctico 1: En esta práctica se realizara un primer acercamiento al Bloque Bucle que representa el ciclo repetitivo while, se realizara un algoritmo capaz de mover el robot L1 en forma rectangular indefinidamente.

Actividades Evaluativas: En las actividades propuestas correspondiente a este taller, el estudiante realizara dos algoritmos, en el primero se le pedirá al estudiante modificar el taller práctico 1, usando el selector de modo del bloque bucle y dando clic en la opción conteo. Para la segunda actividad propuesta los estudiantes modificaran el algoritmo del taller práctico I para que realice un movimiento descrito en una imagen.

En las actividades de análisis propuestas para este taller, se realizan preguntas con el fin de evaluar la comprensión de los conceptos y bloques programables, nombrados en el taller práctico 1.

Taller práctico 2: En este taller práctico se le enseñara al estudiante el uso de del bloque bucle unido con bloque interruptor ,con la ayuda de las instrucciones de la guía escribirá un algoritmo, que hará que el robot L1, a través del sensor de color y los motores grandes siga una línea de color roja .

Actividades Evaluativas: En las actividades propuestas, se les pide a los estudiantes que modifiquen el taller práctico 2, cambiando la entrada de color de rojo a negro, luego se le pedirá que dibuje con un marcador una línea negra y la posicione a una distancia de un metro del robot L1. Los estudiantes deben observar la ejecución del algoritmo y responder unas preguntas de análisis referentes al movimiento del robot.

En una segunda actividad, se les pide a los estudiantes que desarrollen un algoritmo que sea capaz de detectar el color azul. Si el color detectado es azul el robotL1 deberá girar hacia la izquierda, en caso contrario el robot girara hacia la derecha

En las actividades de análisis el estudiante debe responder unas preguntas, diseñadas para saber el nivel de comprensión del uso del bloque bucle, además se le pedirá diseñar y desarrollar un algoritmo que desplace el robot L1 de un punto A a un punto B, atravesando unos obstáculos en el trayecto, como se describe en una imagen ilustrativa.

Taller práctico 3: En este taller los estudiantes diseñaran un algoritmo capaz de seguir una línea de color negro (seguidor de línea 1), a través de sensor de color programado en el modo detección de intensidad de luz, el robot L1 será capaz de seguir la línea negra detectando la

variaciones de intensidad de luz y accionando los motores grandes con la ayuda del bloque interruptor que representa la estructura condicional IF.

En este taller se usan las pistas diseñadas por el tutor para la ejecución del seguidor de línea 1, el estudiante deberá observar la ejecución para posteriormente responder las preguntas de análisis.

Actividades Evaluativas: En las actividades prácticas el estudiante realizara cambios en el algoritmo de taller práctico 3, deberá modificarlo para que cuando el robot detecte una línea roja en alguna de la pistas diseñadas, este se detenga al instante.

En las actividades de análisis los estudiante deberán responder unas preguntas, relacionadas al desarrollo del taller práctico y las actividades propuestas, esto con el fin de saber cómo se desempeñaron y si comprendieron el desarrollo de los algoritmos.

Taller práctico 4: En este taller se le enseñara a los estudiantes el concepto de contador, usando el bloque bucle y el sensor táctil, se programara un algoritmos en el software LEGO Home Edition capaz de contar el número de veces que es presionado el sensor táctil, mostrando en la pantalla del bloque EV3 en tiempo real el conteo. En esta guía se pondrá en práctica el concepto de variables definido en la guía 1.

Actividades Evaluativas: En las actividades propuestas, se les pedirá a los estudiantes que modifiquen el algoritmo del taller práctico para que cuente de dos en dos, cada vez que se presione el sensor táctil.

En las actividades de análisis se pregunta a los estudiantes, sobre el desarrollo del algoritmo del taller práctico, esto para analizar si comprendieron los bloques usados y su funcionamiento en el algoritmo.

6.2.5. GUIA 4. (Bloques Avanzados En LEGO Home Edition)

En esta guía se enseñara el uso del bloque acceso al archivo y el modo avanzado del bloque matemática, se hará una descripción de cada uno de ellos y se explicara los diferentes modos de programarlos. A continuación se describirá el componente teórico, los talleres prácticos, actividades evaluativas, correspondientes a esta guía.

6.2.5.1. *Componente teórico*

La teoría de esta guía se iniciara con la explicación de los bloques esenciales para el desarrollo de los talleres prácticos, anteriormente en la guía 2 se explicó el uso del bloque matemática en la solución de problemas con operadores aritméticos, en esta guía se usara el modo avanzado de este bloque para la escritura de fórmulas aritméticas complejas. Además se usara el bloque acceso al archivo que es un bloque capaz de guardar datos recogidos de los diferentes sensores y armar un archivo plano con estos datos.

Recursos LEGO Mindstorms: En este apartado se definirán los bloques correspondientes al desarrollo de la guía, con la descripción de su funcionamiento y los modos de programarlos. Los más importantes son el bloque acceso al archivo y el bloque matemática en su modo avanzado.

6.2.5.2. *Talleres prácticos*

Taller práctico I: En este taller se le enseñara a los estudiantes el uso del bloque matemática en el modo avanzado, con ayuda de las instrucciones de la guía y el software LEGO Home Edition los estudiantes serán capaces de programar un algoritmo que sigue una línea de color negro (seguidor de línea 2). A través de sensor de color programado en el modo detección de intensidad de luz, el robot L1 será capaz de seguir una línea negra detectando la variaciones de intensidad de luz y con ayuda del bloque matemática en el modo avanzado será capaz de calcular a través de una formula aritmética la dirección que debe tomar el robot, esta dirección accionará los motores grandes. En esta práctica el robot L1 se ejecutara en una de las pistas diseñadas por el tutor.

Actividades Evaluativas: En las actividades propuestas en este taller, los estudiantes deberán desarrollar un algoritmo que use el bloque matemática en modo avanzado para impulsar al RobotL1 de manera indefinida.

En las actividades de análisis el tutor propone preguntas relacionadas al desarrollo del taller práctico, con el fin de saber si los estudiantes comprendieron el desarrollo del algoritmo.

Taller práctico II: En este taller se le enseñara a los estudiantes el uso del bloque acceso al archivo. Usando el sensor de color, las instrucciones de la guía y el software LEGO Home Edition se programará un algoritmo capaz de capturar cada 3 seg un color, este será guardado en el bloque acceso al archivo y los estudiantes podrán acceder a la información conectando el bloque EV3 a un ordenador.

Actividades Evaluativas: La actividad propuesta para este taller, consistirá en armar una estructura para el sensor de color en el robot L1. Los estudiantes deberán desarrollar un algoritmo para el robot L1, que use el bloque archivo, para almacenar 4 colores cada tres segundos.

En las actividades de análisis se preguntará a los estudiantes sobre el desarrollo del taller práctico y las actividades propuestas, con la finalidad de saber la comprensión de los conceptos desarrollados.

6.2.6. GUIA 5. (Bloques Avanzados En LEGO Home Edition, Uso De Arrays)

En esta guía se enseñará a los estudiantes el concepto de array, se definirá y se realizará una descripción de sus partes y características. A continuación se describe el componente teórico, los talleres prácticos y las actividades evaluativas, correspondientes a esta guía.

6.2.6.1. *Componente teórico*

La teoría que se diseña en esta guía, inicia con la definición de ¿Qué es un array o arreglo?, se realiza una descripción visual del concepto, se definen las partes y componen de un arreglo, se mencionan algunos ejemplos de la vida real donde es usado y se propone una actividad donde los estudiantes se apropian del concepto.

Recursos LEGO Mindstorms: En esta sesión se realizará una explicación de los bloques programables usados en los talleres prácticos, con una descripción de su funcionamiento y modos de programarlos, los estudiantes usarán estos bloques posteriormente en los talleres prácticos, donde se explicará, y se estructurará un programa que llegara a la explicación de un concepto. En esta sesión los bloques que se usarán son el bloque variable en el modo secuencia numérica, el bloque operaciones secuenciales, el bloque pantalla, el bloque espera, entre otros.

6.2.6.2. Talleres prácticos

Taller práctico 1: En este primer taller los estudiantes entenderán el concepto de array estático, con la ayuda de las instrucciones en la guía se desarrollará un algoritmo que imprima los valores de un array en la pantalla del bloque Ev3.

Actividades Evaluativas: En las actividades propuestas, se desarrollará un algoritmo que suma los elementos de un arreglo, como datos de entrada del arreglo el estudiante usará las edades de sus compañeros de curso.

En las actividades de análisis el estudiante resolverá unas preguntas, para evaluar si comprendió el concepto de array.

Taller práctico 2: En esta práctica se armará una estructura para el sensor de color y el sensor táctil, con el objetivo de continuar con los talleres prácticos.

Actividades Evaluativas: En las actividades propuestas se le pedirá a los estudiantes que describan, los posibles usos de la estructura armada en el taller práctico 2 y luego deberán plantear un algoritmo.

Taller práctico 3: En este taller se le enseña al estudiante como realizar un arreglo dinámico con LEGO Mindstorm, se diseñará un programa que detecta cuatro colores, cada color tiene asignado predeterminadamente un número que lo identifica, este número se almacena en un array, que se imprime en la pantalla del bloque EV3.

Actividades Evaluativas: En las actividades propuestas para este taller se desarrollará un algoritmo que tomará dos arreglos numéricos A y B de la misma longitud y multiplicará los datos de A con los de B, guardando el resultado en un arreglo R, que se visualizará en la pantalla del bloque EV3.

En las actividades de análisis el estudiante responderá unas preguntas, que ayudarán al tutor a saber si los conceptos explicados en el taller, quedaron claros o no.

6.2.7. GUIA 6. (Bloques Avanzados En LEGO Home Edition, Uso Del Condicional Switch)

En esta guía se explica el uso del condicional SWITCH, sus características y su modo de uso en la resolución de problema, se describirá un componente teórico, los talleres prácticos y las actividades de evaluación.

6.2.7.1. Componente teórico

En esta guía se le enseñará a los estudiantes el uso del condicional switch, comenzado con la definición y dando un ejemplo ilustrativo de su estructura y su uso en el desarrollo de algoritmos.

Recursos LEGO Mindstorms: En este ítem se realizará una explicación de los bloques programables usados en el taller práctico. Los bloques usados en el desarrollo de la guía son: el bloque interruptor, el bloque variable, el bloque bucle, el bloque espera.

6.2.7.2. Talleres prácticos

Taller Práctico I: En esta práctica los estudiantes tendrán la experiencia de armar el robot L2, que se usará más adelante en el desarrollo de un algoritmo.

Actividades Evaluativas: En las actividades propuestas el estudiante deberá nombrar algunos usos del robot L2.

Taller práctico II: En este taller se le enseñará a los estudiantes el uso del condicional switch, con la ayuda de las instrucciones de la guía, el software Lego Home Edition y el robot L2 desarrollarán un algoritmo que contará el número de piezas de colores que irán pasando por el sensor de color montado en el robot L2, este conteo se visualizara en la pantalla del bloque Ev3, donde se mostrará el color y consecutivamente el número de piezas contadas.

Actividades evaluativas: En las actividades propuestas, los estudiantes modificarán y mejorarán el programa del taller práctico 1 para que identifique más colores. Luego el estudiante realizará un algoritmo que sea capaz de identificar la intensidad de luz de dos colores e imprimirá en la pantalla del bloque EV3 que color es el de mayor intensidad de luz.

En las actividades de análisis el estudiante responderá una serie de preguntas, con el objetivo de evaluar la comprensión de los conceptos abordados en el taller práctico, concepto como el del condicional switch.

En el reto propuesto en este taller los estudiantes desarrollaran un algoritmo que es capaz de detectar 5 colores y almacenarlos en un array, luego los imprima en pantalla y con ayuda del bloque sonido dirá el nombre del color.

Para la fase final del curso se diseñó un test final que tiene el objetivo de medir en los estudiantes participantes, las competencias del pensamiento computacional, tales como comprensión lectora, resolución de problemas, pensamiento lógico matemático, pensamiento algorítmico, entre otras.

Este Test final no pretende ser un examen ni plantear nuevas dificultades. Sólo se les brinda la oportunidad a los estudiantes de repasar y comprobar la adquisición de los nuevos conocimientos y ver su mejoría con respecto al test inicial, y así constatando la efectividad o fracaso del curso.

Estos recursos permitirán que los estudiantes interesados en realizar estudios de pregrados en áreas de la computación desarrollen el pensamiento computacional, adquieran destrezas y habilidades en la resolución de problemas, en el planteamiento de algoritmos, además de contar con la ayuda y tutoría de un tutor con amplio conocimientos en el tema, no obstante las guías están diseñadas para que los estudiantes sean autónomos en el proceso de aprendizaje.

7. RESULTADOS

Como se mencionó en el diseño estratégico (capítulo 6) , del periodo inter semestral del mes de (abril de 2019) se contactaron telefónicamente un grupo de estudiantes matriculados para el primer semestre del programa de ingeniería de sistemas, donde se les explicó la posibilidad de asistir al curso de Desarrollo del Pensamiento Computacional con LEGO Mindstorm EV3, antes de iniciar el semestre, enfatizando su importancia y que les serviría como base introductoria a la materia programación estructurada. Seis estudiantes aceptaron participar en el curso, en común acuerdo con los estudiantes se estableció el horario para el desarrollo del curso, se citó a los estudiantes en los laboratorios de Ingeniería de Sistemas, específicamente, en el salón de Aula TIC, en el edificio Jorge Gaitán Durán (JG) de la Universidad de Pamplona.

7.1. Guía 1. Primeros pasos con LEGO Mindstorm EV3

7.1.1.- Sesión #1

Se dio inicio al curso el día 09 de abril del año 2019 con una duración de 3 horas, en el que se explicó a profundidad a los estudiantes sobre lo que se iba a realizar durante el curso. A continuación, se preguntó a cada estudiante: ¿De qué Institución Educativa fueron graduados? ¿Qué temáticas impartían en las clases de informática en su Institución Educativa? Esto con el fin, de conocer el tipo de formación que recibieron en el área de informática y sus conocimientos en programación.

Una vez obtenida la información, se hizo una introducción al curso en la que se definió el término pensamiento computacional, los objetivos del curso y se mostró la herramienta LEGO Mindstorm EV3, explicando sus componentes. Se continuó con la aplicación de un test inicial, con el objetivo, de medir el desempeño de los estudiantes en las diferentes áreas del pensamiento computacional. Se observa que el tiempo de desarrollo del test inicial fue aproximadamente de 1 hora, evidenciando las dificultades de la población frente a la comprensión lectora, pensamiento lógico matemático y pensamiento algorítmico, entre otras.

Después de la aplicación del test inicial, se procedió a dividir el curso en dos grupos, cada uno de tres integrantes (grupo 1 y grupo2), esto con el fin, de repartir equitativamente, el recurso físico, ya que, solo se contó con dos sets LEGO Mindstorm EV3. Después de esto se procede a trabajar la primera guía, la cual, introdujo a los estudiantes en los conceptos referentes al pensamiento computacional apoyado en el uso de la herramienta LEGO Mindstorm EV3, en esta guía los estudiantes realizaron una lectura preliminar sobre los conceptos: algoritmia, variables y constantes. En esta área los estudiantes desarrollaron las actividades de reflexión y las actividades de análisis de manera óptima.

Luego, se realizó una lectura de los bloques que se implementarán en los talleres prácticos, se recomienda al estudiante prestar atención a las indicaciones de cada bloque, ya que serán esenciales en la programación de los algoritmos planteados en la guía, después de esto se desarrolla el taller práctico I, se presenta un esquema de los elementos que componen el Bloque EV3, se presenta el robot L1 y se le entrega al estudiante para su programación, quien siguiendo una serie de instrucciones es capaz de navegar por el entorno del Bloque EV3, ejecutando un

primer programa, hasta este punto los estudiantes no presenta dificultad alguna en el desarrollo de las guías, ya que, las instrucciones fueron claras y concretas.

En el Taller Práctico II, el estudiante realizó una lectura preliminar en la que se explica cómo navegar en el entorno del software LEGO Home Edition, las diferentes pestañas y luego se procedió a programar un primer algoritmo basado en las instrucciones de la guía, el programa realizado permite que el robot se desplace en línea recta por un determinado tiempo.

Después de guardarlo, almacenarlo y conectar el Bloque a la computadora para descargar el algoritmo realizado al Bloque EV3, se procedió a su ejecución, en este taller práctico los estudiantes tampoco presentaron dudas en cuanto a las instrucciones de la guía, pero sí dificultades en el desarrollo de las actividades prácticas, ya que, se proponía a los estudiantes un reto en el que el robot L1 se desplazaba de forma recta durante un tiempo, luego giraba y hacía un desplazamiento más, además de esto, se pedía a los estudiantes que realizaran cambios en los valores de la velocidad del robot, con ayuda del tutor, los participantes lograron culminar el reto, dicha ayuda no consistió en otorgar las respuestas correctas, sino a través, de una orientación en la que los estudiantes lograron comprender el desarrollo de la actividad.

Por último, se realizó una actividad de análisis en la que se evalúa el nivel de comprensión que tuvieron los estudiantes frente a la temática, en la que se observa que la mayoría de los estudiante comprenden los conceptos explicados.

En el Taller Práctico III se enseñó a los estudiantes la programación interna, usando el Bloque EV3, en esta guía se les enseñan los nombres de las diferentes pestañas y se procede a otorgar las instrucciones de la guía; el algoritmo realizado en el Bloque EV3 directamente,

consistía en que el Robot L1 girara en 360° durante dos minutos. Después, se procede a la realización de una actividad de análisis, donde se evalúa la comprensión de la temática por parte de los estudiantes.

7.1.2. Estadística del desempeño de los grupos

En este apartado se presenta una estadística del desempeño de los grupos en el desarrollo de las actividades propuestas, análisis, reflexión y retos.

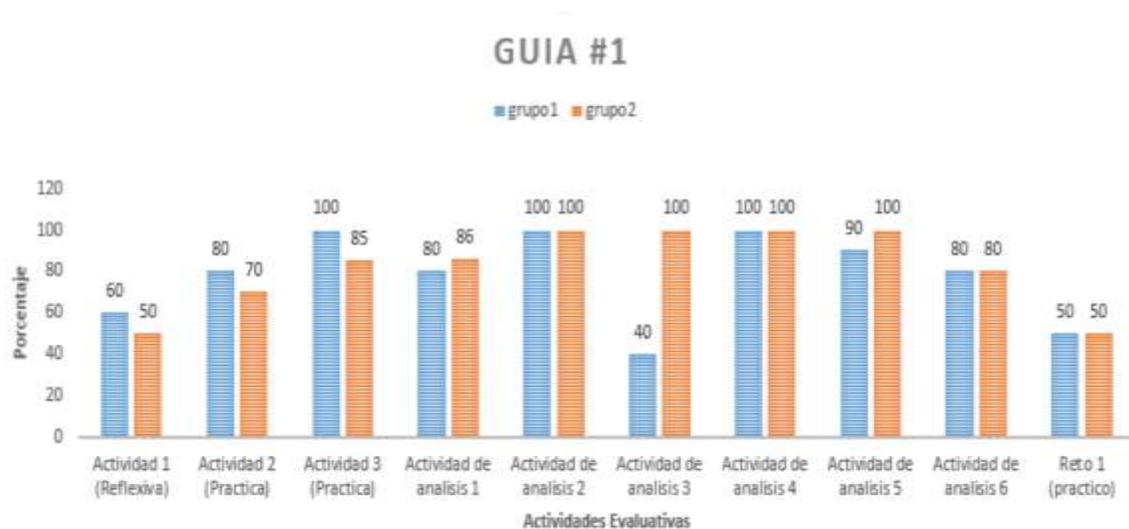


Figura 13 Rendimiento (Actividades Evaluativas) –Guía #1

Como se puede observar en la figura 13, los estudiantes tienen dificultades en el desarrollo de las actividades evaluativas de la guía 1, esto es justificado ya que los conocimientos impartidos son nuevos para ellos y requieren de práctica.

7.2. Guía 2. Uso de operadores con LEGO Mindstorm EV3

7.2.1. Sesión #2

El segundo encuentro se realizó el día 12 del mes de abril del año 2019 con una duración de 2 horas, en esta sesión se dio inicio a la aplicación de la guía número 2. Al comienzo de la guía los estudiantes realizan una lectura introductoria al uso de operadores, donde con ejemplos de la vida diaria, se brinda una explicación al concepto, luego se define concretamente el concepto de operadores, cabe mencionar que se les explica a los estudiantes que los operadores se clasifican en tres, operadores aritméticos, relacionales y lógicos, cada uno de ellos es definido y descrito con ejemplos demostrativos. En el desarrollo de las actividades prácticas y de análisis, se observa que los estudiantes se apropiaron de la temática, logrando comprender el uso de los operadores en la solución de problemas, no obstante los estudiantes no desarrollaron las actividades en los 30 minutos propuestos, debido a que las actividades no toman en cuenta aspectos del aprendizaje de cada estudiante y no consideran que el desarrollo de los recursos en grupo, depende del pensamiento divergente y convergente de cada individuo.

Terminada la teoría y prácticas del uso de operadores , se describen los bloques necesarios para la programación en el software LEGO Home Edition, donde se menciona a los estudiante los bloques que representan conceptos del componente teórico tales como el bloque matemático, que representa a los operadores aritméticos, el bloque comparar, que representa a los operadores relacionales y el bloque lógica, que representa a los operadores lógicos, se explica que cada uno de ellos es usado en los talleres prácticos para la programación de algoritmos.

En el Taller Práctico I, los estudiantes con la ayuda de las instrucciones de la guía desarrollan un algoritmo que suma dos números y el resultado lo imprime en la pantalla del bloque EV3, en esta actividad los grupos no presentan dificultades en el desarrollo del taller práctico, resuelven las actividades propuestas de manera correcta, algunos estudiantes tuvieron dificultades enlazando el bloque matemática con el bloque pantalla, el tutor aclaró estas dudas y hace una explicación del uso de los bloques. En las actividades de análisis se determina la apropiación de los conceptos y se observa una disposición de los estudiantes a seguir aprendiendo e interactuando con el Robot L1, consecuentemente se fortalecen las áreas del pensamiento tales como la comprensión lectora, pensamiento algorítmico, pensamiento lógico matemático, pensamiento convergente y divergente, entre otras.

En el taller práctico II, los estudiantes aprendieron el uso de los operadores relacionales, realizando un algoritmo que compara dos números, usando el operador mayor que se determina el número mayor, mostrando el resultado en la pantalla del bloque EV3. En esta actividad los estudiantes consiguen entender el uso de los operadores relacionales, ya que logra culminar las actividades prácticas correctamente, las actividades de análisis lo confirman.

El desarrollo de la práctica tuvo una duración de dos horas, sin embargo no fue suficiente para culminar el desarrollo de la misma, se indica a los estudiantes que continuara con el desarrollo de esta en la próxima sesión. Se recomienda dividir en 3 sesiones el contenido referente a esta guía, ya que por su contenido teórico y las actividades prácticas, los estudiantes requieren de tiempo suficiente para asimilar correctamente, los conceptos aquí planteados.

7.2.2. Sesión #3

En este encuentro se continúa con el estudio de la guía número 2 el día 25 del mes abril de 2019 con una duración de 2 horas. Antes de seguir con el desarrollando del taller práctico III se le pide a los estudiantes que construyan una estructura para el sensor táctil y el sensor de color, se observa a los estudiantes motivados, emocionados, ya que esta era la primera vez que armarían una estructura de su propia autoría, posteriormente se desarrolla un algoritmo que usa el operador lógico AND el cual es descargado a la estructura anteriormente armada, para su puesta en marcha, se nota que el estudiante no tiene dificultades en seguir la instrucciones de la guía, ya que desarrolla y ejecuta con éxito el algoritmo. En las actividades de análisis se evidencia el progreso de los estudiantes en la temática y se observa que entienden el uso de los operadores lógicos.

Continuando con la guía, se procede con el aprendizaje del condicional IF, los estudiantes realizan una lectura de la definición del concepto y sus clasificaciones, ilustradas con ejemplos demostrativos, luego de esto se procede a realizar el taller práctico IV, donde los estudiantes usando el sensor táctil, programan un algoritmo que usa el bloque interruptor (condicional IF en LEGO Home Edition).

Se observó que el grupo 2 , no tiene disposición en el desarrollo de las actividades de análisis, ya que algunas de ellas se encuentran sin la solución, se observa un bajo interés en algunos integrantes del grupo en seguir realizando las actividades, ya que se encuentran conversando en el salón de clase temas ajenos al curso, por tal razón no se pudo evaluar el entendimiento de la temática de manera objetiva, por otro lado el grupo1 realizó las actividades correctamente y se observó que entendieron el uso del condicional IF.

Éste día no se culminó el desarrollo de esta guía, debido a que los estudiantes se demoran un promedio de 1 hora en la lectura y ejecución de los talleres prácticos y actividades propuestas,

esto es debido a que los algoritmos que se plantean requieren que el estudiante piense una posible solución y no siempre es la correcta, por esta razón se encuentra constantemente en un proceso de ensayo y error para lograr la solución correcta al problema.

7.2.3. Sesión #4

En esta sesión se continúa con la aplicación de los talleres prácticos de la guía número 2, este encuentro se dio el día 27 del mes de abril del año 2019 y se inicia con la realización del taller práctico V, donde el estudiante sigue reforzando el concepto de estructuras condicionales, los estudiantes realizan un algoritmo que compara dos operaciones aritméticas, el resultado se visualiza en la pantalla del bloque EV3. En el desarrollo de las actividades prácticas los estudiantes realizaron un algoritmo que consistía en imprimir en la pantalla del bloque EV3 el nombre, esta actividad tenía como objetivo profundizar el conocimiento en el uso del bloque texto, que será usado posteriormente en las guías. Se desarrollaron las actividades de análisis referente a la temática y se observó que los estudiantes tienen bases en resolución de problemas, pensamiento algorítmico y que está reforzando la comprensión lectora y la creatividad, componentes importantes en el pensamiento computacional.

En el taller práctico VI, se les entrega a los estudiantes partes del robot L1 para que los diferentes grupos lo armen y lo programen. Las actividades de este taller estaban enfocadas en reforzar toda la temática propuesta en la guía número 2, inicialmente se pide a los estudiantes que programen un algoritmo propuesto, usando el bloque Ev3 y luego se les solicita que lo programen esta vez usando el software LEGO Home Edition, en el desarrollo de esta actividad los estudiantes tuvieron dificultades mínimas, relacionadas al manejo de la interfaz del bloque Ev3, el tutor orienta a los estudiantes y resuelve estas dificultades. En este taller práctico se dispone de

un reto donde los estudiantes deben resolver un problema, planteando un algoritmo para la solución. En el desarrollo de esta actividad fue importante la ayuda del tutor, ya que el reto presentaba un nivel de dificultad en su realización, se hizo una lluvia de ideas diseñando posibles soluciones al problema con el fin de lograr el desarrollo de la actividad, efectivamente se observó que la orientación del tutor ayudo a los estudiantes en la solución del problema.

7.2.4. Estadística del desempeño de los grupos

En esta sesión se describe gráficamente, el rendimiento de los estudiantes en las diferentes actividades evaluativas propuestas en la guía 2, como se puede observar en la figura 14.



En la figura 14 se visualiza el avance de los estudiantes en el desarrollo de las actividades evaluativas.

Figura 14 Rendimiento (Actividades propuestas)-Guía 2

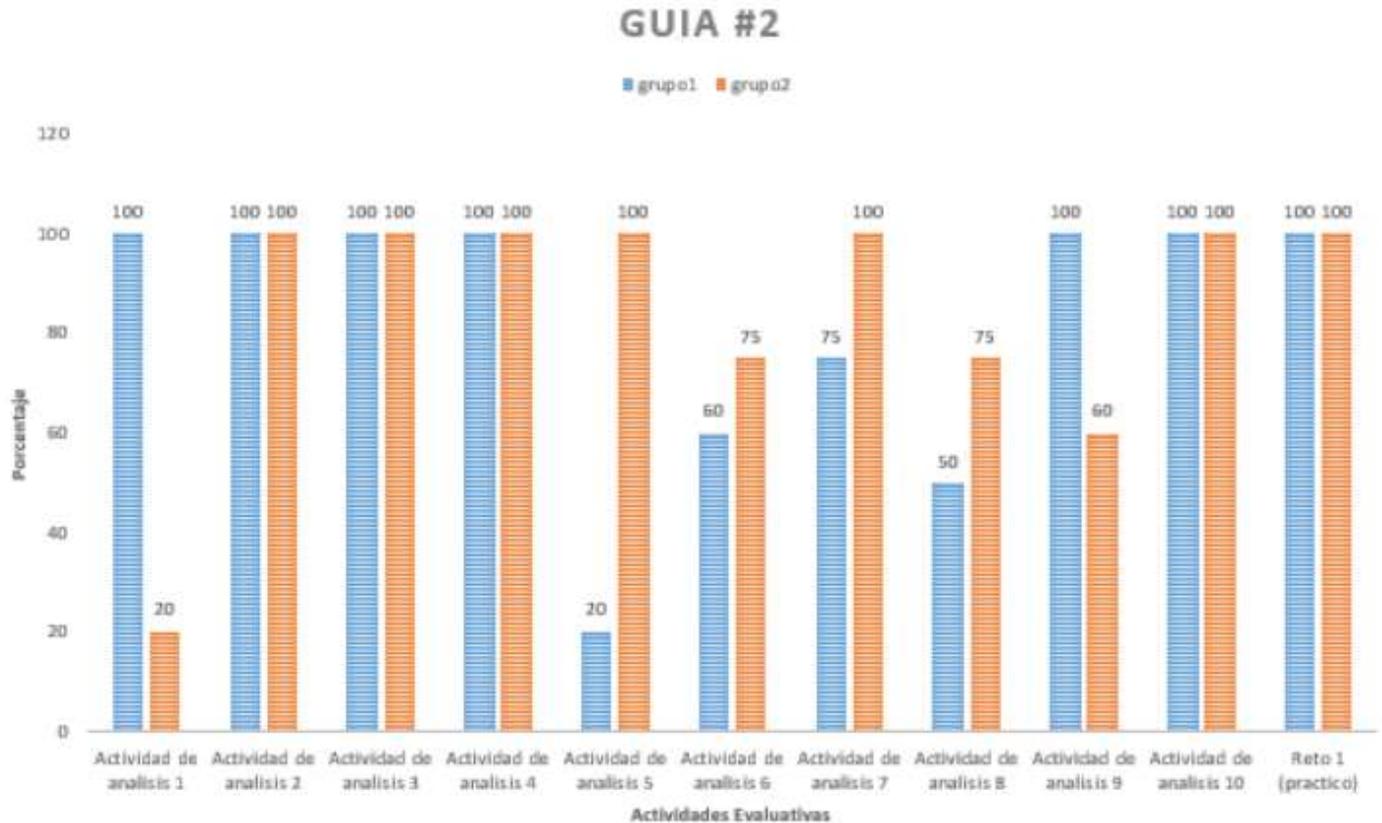


Figura 15 Rendimiento (Actividades de análisis y retos)-Guía 2

Como se describe en el capítulo 6 (Diseño estratégico), las actividades de análisis, son de gran importancia en la formación de los participantes del curso, tienen como objetivo ponderar, valorar y concluir respecto al desempeño de los aprendices en la elaboración de los talleres prácticos y actividades propuestas. Se observa en la figura 15, falencias en la comprensión de los conceptos, esto se debe a que los aprendices aun no asimilan del todo los conceptos propuestos en las guía.

7.3. Guía 3. Ciclo repetitivo while con LEGO Mindstorm EV3

7.3.1.Sesión #5

El desarrollo de la presente guía se llevó a cabo el día 29 del mes de abril del 2019, con una duración de dos horas, se comienza realizando actividades de repaso de los temas tratados en la guía 2, que los estudiantes desarrollaron correctamente, luego se realiza una lectura del referente teórico donde se explica el concepto de ciclos repetitivos, se define que es el ciclo repetitivo while y se dan ejemplos de su uso, posteriormente se describen los bloques usados para programar los talleres prácticos, se invita al estudiante que preste mucha atención a las indicaciones y modos del bloque llamado Bucle que representa al ciclo while.

En el taller práctico I, se les enseña a los estudiantes como programar el bloque bucle en el software LEGO Home Edition, se desarrolla un algoritmo que hace que el robot se mueva en línea recta indefinidamente, con la ayuda de las instrucciones de la guía el estudiante pudo desarrollar el algoritmo y ejecutarlo, se observó al robot L1 moverse por el piso del salón indefinidamente, hasta que se presionó el botón atrás del bloque EV3.

En las actividades propuestas se observó que en alguna de ellas, específicamente en la que el robot L1 debía hacer un recorrido propuesto en una imagen, hubo dificultades, ya que no sabían cómo programar los motores grandes para el movimiento del robot, con ayuda del tutor se pudo resolver las dudas orientado a los estudiantes. Al final los dos grupos resolvieron el problema ensayando las soluciones y corrigiendo los errores. Luego de esto se realizaron actividades de análisis, donde se observó que el estudiante logro comprender el algoritmo y el uso del ciclo repetitivo while.

Para el taller práctico II, se posiciona a cada uno de los grupos en una mesa diferente, se les reparte un pedazo de cinta roja donde cada grupo debe pegarla en un lugar de la mesa lejos de los extremos de la misma, luego con el software LEGO Home Edition se desarrolló un algoritmo que se mueve en línea recta indefinidamente hasta que encuentre una línea de color rojo que hace que el robot L1 se detenga, usando el ciclo repetitivo while , el bloque interruptor, los dos motores grandes y el sensor de color . En la ejecución del algoritmo se observa que el robot, posee mucha velocidad y no deja que el sensor de color detecte la franja roja en la mesa, por consiguiente nunca se detiene. Se le indica al estudiante que reduzcan la velocidad de los motores grandes, con estos el robot debe detenerse cuando detecte la franja roja. Efectivamente el problema era de velocidad y se logró culminar el taller práctico con éxito.

En la actividad propuesta para este taller práctico se observó, que los grupos desarrollaron y ejecutaron la solución al problema de manera correcta, en las actividades de análisis se notó que los grupos tuvieron dificultades con el desarrollo del problema propuesto en una imagen , debido a que los sensores, los motores grandes y las condiciones del piso no dejaban que el robot ejecutara las instrucciones del algoritmo de manera precisa, por esta razón no se logró lo estimado para esta actividad.

A la mayoría de los estudiantes se les observa motivados con los avances del curso, ya que eran participativos en el desarrollo de las actividades, se les indica que la continuidad de la guía se hará en el siguiente encuentro, debido a que las dos horas habían terminado. En esta sesión se observa que tardaron mucho más tiempo en la ejecución de las actividades prácticas, ya que estas requieren más atención y esfuerzo en su desarrollo.

La Universidad de Pamplona dio inicio a sus actividades académicas el día 21 de abril del año 2019 para comenzar con el periodo académico 2019-1, para este tiempo los estudiantes, tenían dos semanas de haber entrado a sus clases universitarias, por lo tanto ya habían dado sus

primeras clases de la materia programación estructurada, algunos de ellos comentaron que los conocimientos que estaban recibiendo en cuanto a variables y constantes les fueron útiles en la materia.

7.3.2. Sesión #6

En esta sesión se da continuidad a la guía número 3, este encuentro se lleva a cabo el día 4 de mayo del 2019, con una duración aproximada de dos horas, se comienza con la realización del taller práctico III donde se desarrolló un algoritmo que seguía una línea de color negra (seguidor de línea 1) de manera indeterminada, con la ayuda del sensor de color, el bloque bucle, el bloque interruptor y las instrucciones de la guía los estudiantes lograron desarrollar este algoritmo. Para esta práctica se diseñaron en cartulina tres pistas de carrera con anterioridad, con la ayuda de los estudiantes se pegaron en las mesas, esto con el objetivo de ejecutar el algoritmo desarrollado en este taller.

En la ejecución del algoritmo se observó como el robot hacia el recorrido en las diferentes pistas usando el sensor de color en el modo intensidad de luz. En la solución de las actividades prácticas el estudiante tuvo dificultades en el desarrollo del algoritmo que permitía que el robot hiciera el recorrido por una pista, que al momento de detectar una franja de color rojo paraba la ejecución del algoritmo, con la ayuda del tutor se plantearon posibles soluciones y se ejecutaron algunas de ellas, se llegó a una solución adecuada al problema, la cual fue programada y probada con éxito. En las actividades de análisis se evaluó si el estudiante comprendió el algoritmo de taller práctico y de las actividades propuestas se concluye que la mayoría de ellos tienen los conceptos claros.

Luego se desarrolla el taller práctico IV, donde los estudiantes con la ayuda de las instrucciones de la guía, fueron capaces de realizar un algoritmo que contaba el número de pulsaciones del sensor táctil, el objetivo de este taller era ilustrar el concepto de contador, visto en la parte teórica de la guía. Se desarrollan las actividades propuestas y de análisis, donde se observa que los estudiantes comprenden el uso de los diferentes bloques programables del algoritmo, entiende el uso del bloque variable como un contador que incrementa su valor cada vez que el bloque Sensor táctil es presionado, además comprende la importancia del bloque Bucle en la ejecución del algoritmo del taller práctico.

7.3.3. Estadística del desempeño de los grupos

En este apartado se observa el rendimiento de los estudiantes en el desarrollo de las actividades evaluativas, presentes en la guía 3.

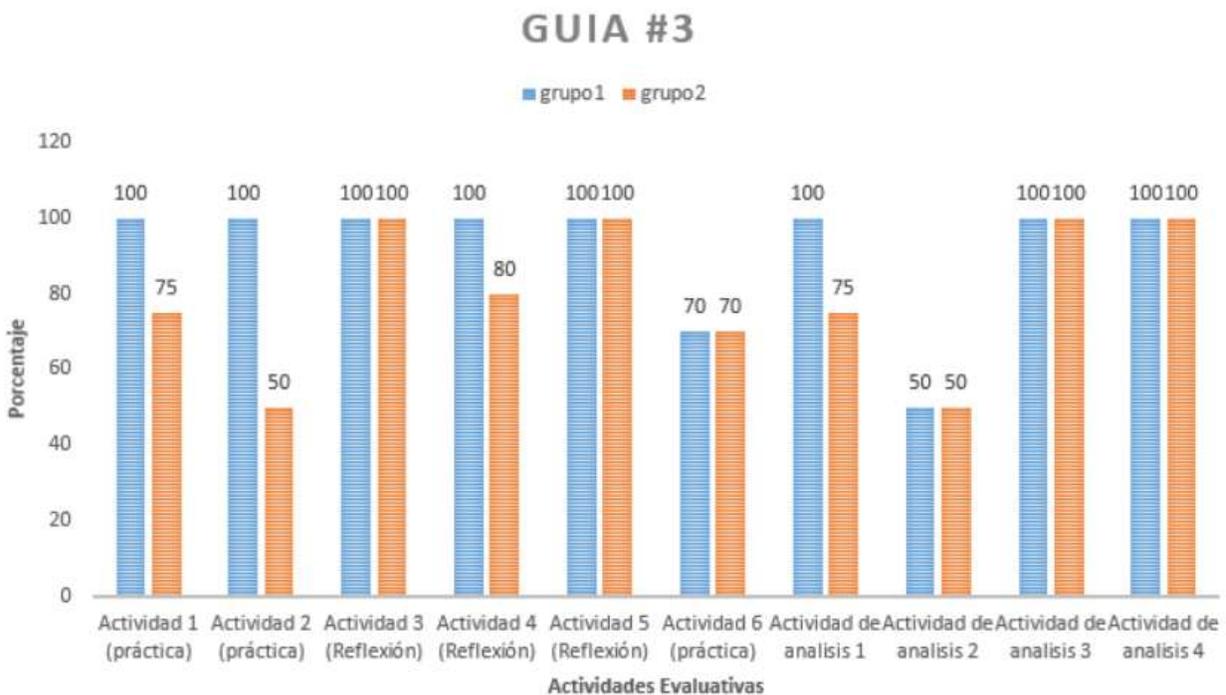


Figura 16 Rendimiento (Actividades Evaluativas)-Guía 3

En la figura 16 se observa que en la gráfica, se muestra el rendimiento de los estudiantes por grupo, se evidencian dificultades en el desarrollo de las actividades evaluativas, en esta guía se enseñó el uso del ciclo repetitivo while, es un tema nuevo que algunos aprendices no conocían y esto dificultó su asimilación.

7.4. Guía 4. Bloques Avanzados en LEGO Home Edition

7.4.1. Sesión #7

En esta guía se le enseña a los estudiantes el uso de los bloques avanzados con el software LEGO Home Edition, se da inicio a su aplicación el día 6 del mes de mayo del año 2019, con una duración de dos horas, se empieza con una lectura de los bloques necesarios para el desarrollo de los talleres prácticos, enfatizando el uso, características del bloque acceso al archivo y el bloque matemática en el modo avanzado. Luego se procede con el desarrollo del taller práctico I, donde los estudiantes desarrollan un algoritmo que sigue una línea de color negra (seguidor de línea 2), este tiene como característica el uso del bloque matemática que por medio de unas fórmulas aritméticas controla la dirección y el movimiento del robot L1. El taller fue desarrollado exitosamente por los estudiantes, las actividades propuestas fueron realizadas satisfactoriamente, en las actividades de análisis se orienta a los estudiantes a comparar el seguidor de línea programado en la guía 3 con el de esta práctica, la mayoría de los estudiantes concluyen que la diferencia radica en la velocidad y el modo en que se programa el algoritmo, ya que este algoritmo tiene un movimiento más fluido según el criterio de los estudiantes. Cabe mencionar que se usaron los recursos disponibles, como cintas de colores y las diferentes pistas para la ejecución del algoritmo.

En el desarrollo de las actividades prácticas los estudiantes no presentaron dificultades, y en las actividades de análisis se observó que entendieron el algoritmo planteado en el taller práctico. Al programar el reto propuesto en este taller, algunos de los estudiantes no fueron capaces de dar una solución al problema, se necesitó de la intervención del tutor, para brindar a los estudiantes una asesoría, donde se resolvieron dudas e inquietudes, con esta orientación lograron finalmente resolver la actividad y probar su ejecución.

En el desarrollo del taller práctico II, se realizó un algoritmo que tomaba lecturas de los colores que detectaba el sensor de color cada tres segundos, estas lecturas eran almacenadas en un archivo plano con el bloque acceso al archivo. Por medio de las instrucciones de la guía el estudiante busco este archivo y logro ver el registro de datos del sensor de color. En el desarrollo de las actividades propuestas los estudiantes analizaron el algoritmo del taller y lo modificaron para que almacenara 4 datos en el archivo plano, desarrollaron y comprendieron el uso del bloque acceso al archivo, esto se afirma gracias a las actividades de análisis y el reto propuesto donde logran culminarlas de manera exitosa, cabe mencionar que alguno de ellos tuvieron dificultades en el desarrollo del reto, con la asesoría del tutor se orientó al estudiante y se resolvieron dudas e inquietudes.

En esta guía no se contó con la asistencia de dos estudiantes, se les escribió por el chat y ellos manifestaron que por falta de tiempo y cambios de horarios no era posible seguir asistiendo al curso. Se les motiva a los estudiantes a continuar el curso proponiéndoles una nota apreciativa en el 15% de la materia programación estructurada, en común acuerdo con el docente de la materia, fue viable proporcionar el incentivo a los participantes.

Debido a la deserción de dos estudiantes fue necesario reorganizar los grupos.

7.4.2. Estadística del desempeño de los grupos

En el presente apartado (figura 17), se visualiza el rendimiento de los estudiantes en las actividades evaluativas.

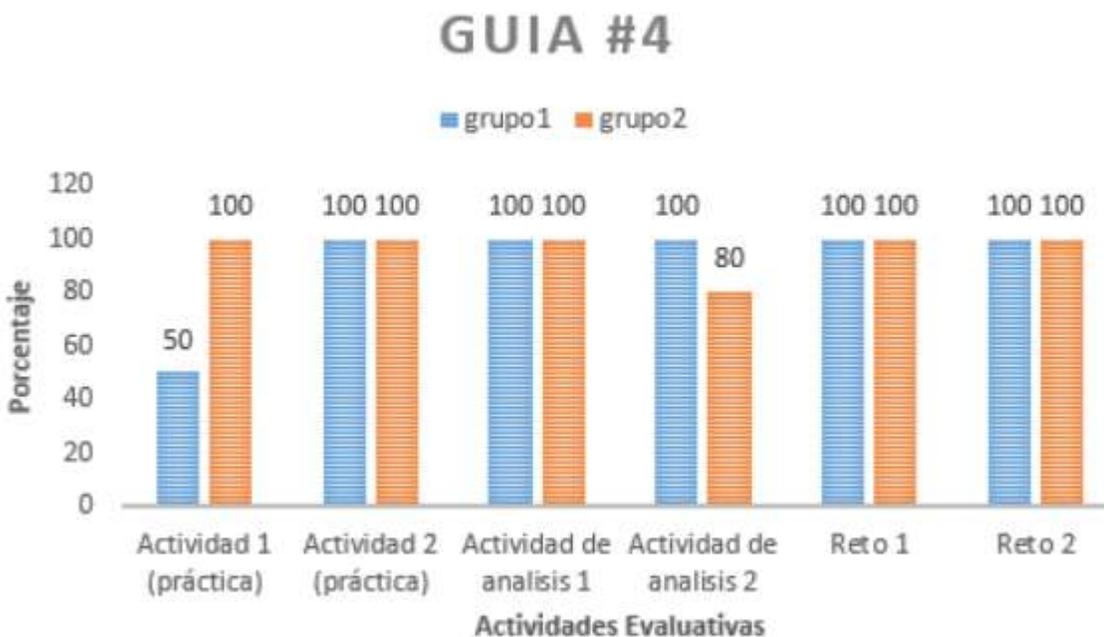


Figura 17 Rendimiento (Actividades Evaluativas)-Guía 5

En esta guía se enseñó la programación de los bloques avanzados en el software LEGO Home Edition, en la figura 17 se puede observar que los estudiantes en el desarrollo de las actividades evaluativas, presentaron un desempeño sobresaliente, lo que indica que lograron aplicar, asimilar los conocimientos correspondientes a esta guía.

7.5. Guía 5. Bloques Avanzados en LEGO Home Edition, Uso de Arrays

7.5.1. Sesión #8

Esta guía fue aplicada el día 10 del mes de mayo del año 2019, con una duración de dos horas, se inició con la definición del concepto de array, explicando sus partes con un ejemplo ilustrativo, esto con el fin de que los estudiantes tengan una mejor comprensión del concepto, se procede a realizar una actividad propuesta donde los estudiantes llenaban un array con las fechas de cumpleaños de los compañeros de curso, esta actividad tenía como objetivo afianzar el concepto, después de esto se describen los bloques programables con el LEGO Home Edition, enfatizando el bloque variable en su modo secuencia numérica, el bloque operaciones secuenciales esencial en la realización de operaciones con arrays, Luego se inicia con el desarrollo del taller práctico I, donde se realiza un algoritmo que imprime en la pantalla del bloque EV3 los números de un array definido en la guía. Con este ejercicio se buscaba que el estudiante se apropiara del concepto de array y como se programaba. En las actividades propuestas se notó dificultades en el desarrollo ya que debían hacer un algoritmo que multiplicara dos arrays, algunos de ellos manifestaban que no sabían por dónde comenzar el problema, mientras que otros modificaban el ejercicio del taller práctico. El tutor explico detalladamente en el tablero como se debía pensar en el problema, haciendo un bosquejo de las interacciones que debían hacerse con el bloque bucle, después de esto los estudiantes intentaron hacer el ejercicio, alguno de ellos lo consiguieron y otros tenían errores en la lógica, rápidamente fueron orientados para llegar a la solución del problema, en las actividades de análisis se observa que el estudiante comprende los conceptos explicados en esta guía.

El taller práctico II, consistía en armar una estructura con el sensor táctil y el sensor de color, con el fin de usarla en el desarrollo de algoritmos posteriormente, se realiza actividades, donde los estudiantes mencionan los posibles usos de la estructura y desarrollan un algoritmo que usa la estructura diseñada en la solución de un problema planteado por el aprendiz.

En el taller práctico III se desarrolló un algoritmo donde se definía un array y se llenaba con los datos de lectura del sensor de color, esos valores del array se imprimían en la pantalla del bloque EV3, con la ayuda de la guía y el material de información suministrado los estudiantes desarrollaron este algoritmo. En las actividades propuestas se programó un algoritmo que multiplicaba dos arrays y almacenaba ese resultado en otro, para la realización del ejercicio se le explico a los estudiantes en el tablero paso a paso lo que debía hacer el algoritmo, los grupos intentaron realizar el ejercicio pero no llegaron a una solución, el tutor ayudo a programar una parte explicando y orientado a los estudiantes, luego ellos se encargaron de terminar el ejercicio. La idea con la aplicación de la guía, es orientar a los estudiantes en la resolución de problemas, logrando que entiendan el concepto de array y sus formas de uso, además de reforzar, desarrollar los conceptos pertinentes al pensamiento computacional.

7.5.2. Estadística del desempeño de los grupos

En este apartado se presenta la estadística del rendimiento de los estudiantes en las actividades evaluativas presentes en la guía 5.

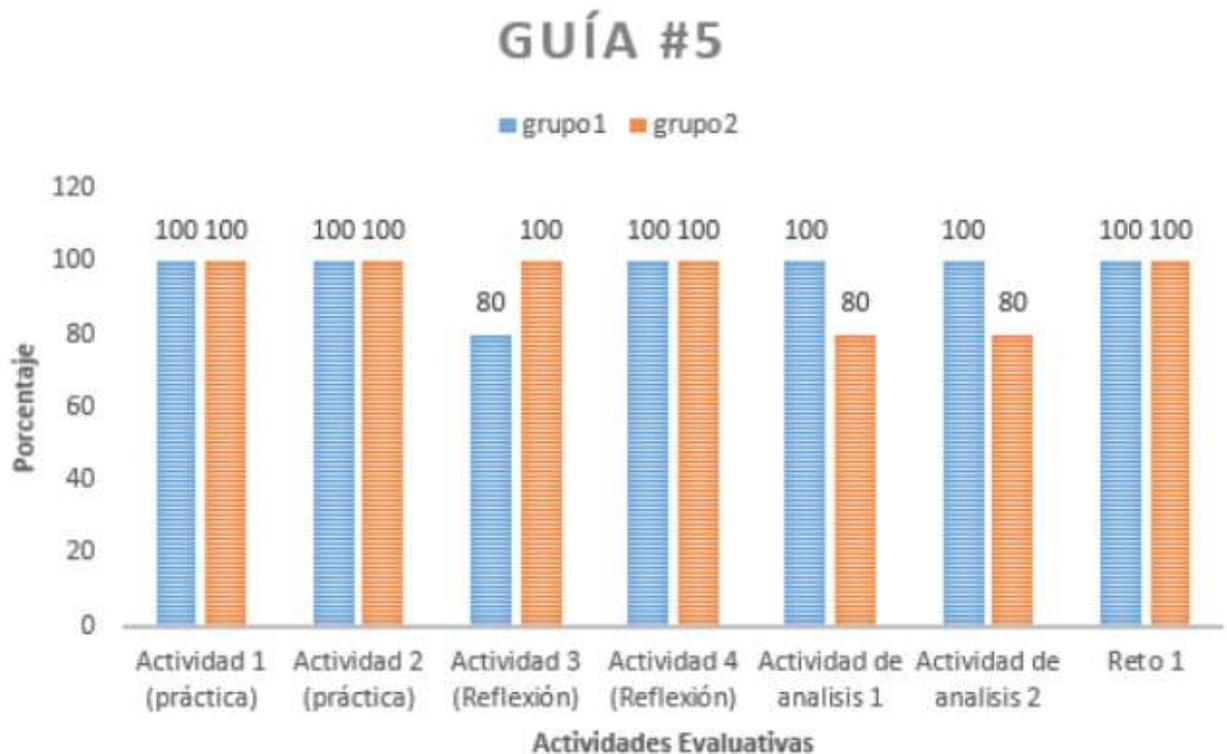


Figura 18 Rendimiento (Actividades Evaluativas)-Guía 5

En la figura 18 se puede observar el rendimiento de los estudiantes en las actividades evaluativas de la guía 5, se observa que los participantes en estas actividades obtuvieron un resultado sobresaliente, notando un avance en la adquisición de los conceptos referentes al pensamiento computacional.

7.6. Guía 6. Bloques Avanzados en LEGO Home Edition, Uso del Condicional Switch

7.6.1. Sesión #9

Este encuentro se llevó a cabo el día 11 del mes de mayo del año 2019, con una duración de 2 horas, en esta sesión se explicó el uso del condicional switch, se definió y se mencionó sus características, además se describen los diferentes bloques programables, se hace mención del bloque interruptor en los diferentes modos, donde cada uno de ellos es un uso del condicional switch.

En el taller práctico I el estudiantes armo el robot L2, que está diseñado para el conteo y clasificación de colores, en esta guía los estudiantes tienen la experiencia de armar el robot desde 0, por consiguiente desarrollan la motricidad fina, la lógica, el razonamiento, el pensamiento divergente y convergente, conceptos que están enlazados al pensamiento computacional. Los estudiantes desarrollan las actividades propuestas, donde menciona los posibles usos del robot L2 en la vida diaria, ellos indicaron algunos usos del robot tales como maquinas que clasifican basura, impresión de circuitería electrónica, entre otros. Con el armado del robot se nota a los estudiantes muy entusiasmados, en esta actividad se emplea una hora y media en la construcción del robot L2, esto es debido a que son más de 100 instrucciones que el estudiante debe realizar para armarlo.

En la realización del taller práctico II, se desarrolla un algoritmo para contar y clasificar colores, los estudiantes lograron desarrollarlo siguiendo las instrucciones de la guía. Se procede con la realización de las actividades propuestas, donde los estudiantes modificaron el taller práctico 2 para que detectara más colores. Después se desarrolla un algoritmo que es capaz de identificar la intensidad de luz de dos colores e imprime el color de mayor intensidad de luz, esto

para fortalecer sus conocimientos en el pensamiento algoritmo, el razonamiento, pensamiento lógico matemático y otras competencias del pensamiento computacional.

Las actividades de análisis fueron resueltas exitosamente, se observa que los estudiantes entendieron lo explicado en el taller práctico 2, el condicional switch y sus diferentes usos, no se presentó ninguna dificultad en el desarrollo de las actividades, debido a que los ejercicios propuestos, se realizaban modificando el algoritmo del taller práctico 2.

En este taller se desarrolló un reto, que consistía en programar un algoritmo con la capacidad de detectar 5 colores, los cuales eran almacenados en un array, que luego se imprimían en la pantalla del bloque Ev3 y con la ayuda del bloque sonido cada color sería nombrado. En el desarrollo de este reto fue importante asesorar a los estudiantes en el uso del bloque sonido y el bloque switch, para que estos pudieran terminar de realizar el algoritmo, ya no daban con la forma relacionar el bloque sonido, para que este nombrara el color, con el acompañamiento del tutor se logró culminar la actividad.

El último encuentro con los estudiantes fue el día 17 de mayo del 2019, en este encuentro se aplicó el test final, donde se evaluaron las competencias referentes al pensamiento computacional, la realización de la prueba tuvo una duración de 2 horas, debido a que algunos estudiantes incumplieron el horario estipulado, para la clase.

7.6.2. Estadística del desempeño de los grupos

En esta sesión se observa el rendimiento de los estudiantes al desarrollar las actividades evaluativas correspondientes a la guía 6.

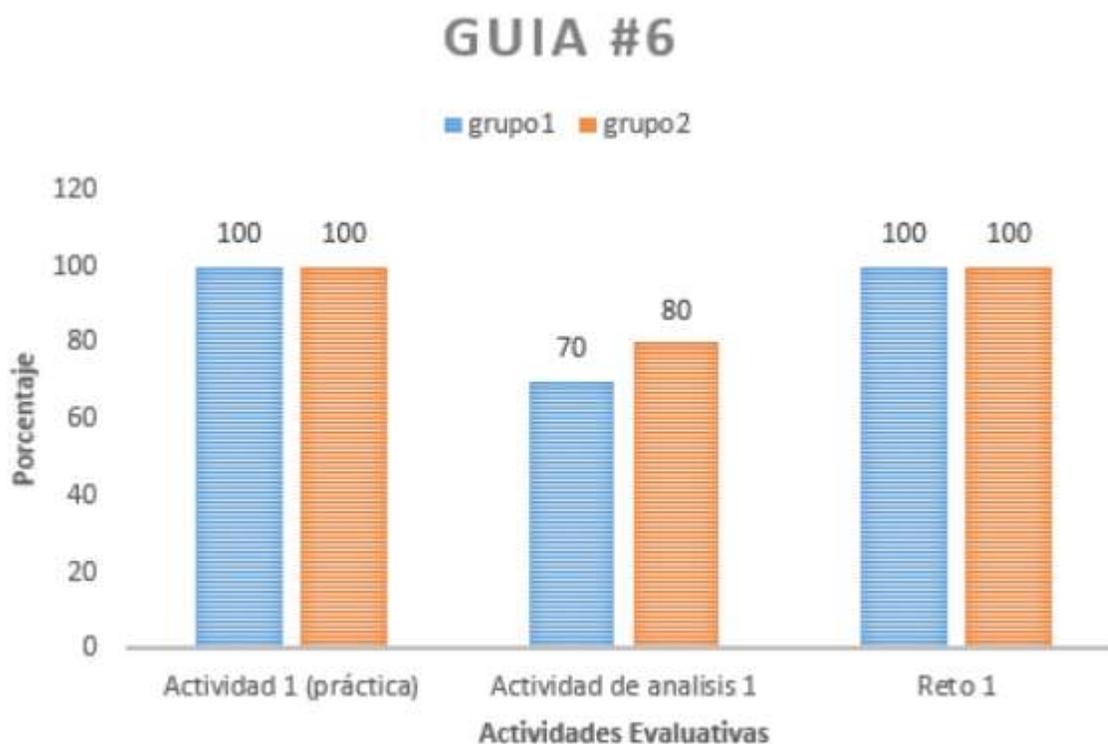


Figura 19 Rendimiento (Actividades Evaluativas)-Guía 6

En la guía 6 se les enseñó a los participantes del curso, el uso del condicional switch, como se puede observar en la figura 19, los estudiantes completaron el 100% de las actividades propuestas de manera satisfactoria.

7.6.3. Seguimiento a estudiantes en curso programación estructurada

Los participantes del curso desarrollo del pensamiento computacional con LEGO Mindstorm, paralelamente asisten a la materia programación estructurada del programa ingeniería de sistemas de la Universidad de Pamplona, donde aprenden a programar e implementar algoritmos en la solución de problemas, el docente de esta materia facilito las calificaciones de los estudiantes asistentes al curso, esto es de gran importancia para este proyecto debido que permite cotejar el desempeño de los estudiantes que asistieron al curso, en comparación al resto de estudiantes de la materia programación estructurada que no asistieron al curso.

Estudiantes	Nota(15%)	Nota(20%)	Definitiva
Est-1	3.0	1.0	2.0
Est-2	4.3	3.8	4.1
Est-3	1	2	1.5
Est-4	4.8	5	4.9
Est-5	4.6	4.9	4.8
Est-6	4.2	4.5	4.4
Est-7	3.5	4.5	4.0
Est-8	3.1	2.5	2.8
Est-9	4.1	4.8	4.5
Est-10	1.5	1	1.3
Est-11	3.5	3	3.3
Est-12	3.1	1	2.1
Est-13	2.5	4.8	3.7
Est-14	3.9	4.8	4.4
Est-15	2.7	2	2.4
Est-16	2.6	1.8	2.2
Est-17	4.5	5	4.8
Est-18	2.9	3	3.0
Est-19	4.2	4.9	4.6
Est-20	2.6	4.7	3.7

Figura 20. Notas primer corte (Programación estructurada)

Los estudiantes resaltados en color amarillo, fueron participantes del curso Desarrollo del pensamiento computacional con LEGO Mindstorm.

Como se puede observar en la figura 20, los estudiantes participantes del curso desarrollo del pensamiento computacional con LEGO Mindstorm Ev3, tiene un buen desempeño académico

en la materia programación estructurada, se puede visualizar que los estudiantes que tienen notas bajas, son estudiantes que no culminaron el aprendizaje del curso.

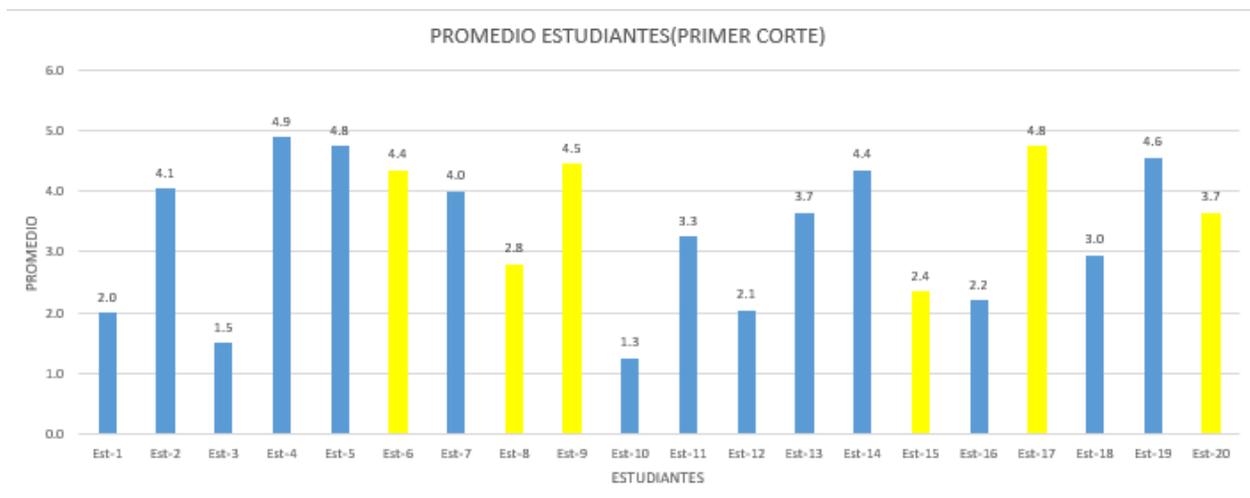


Figura 21. Promedio de notas (corte 1)

Como se puede observar en la figura 21, la mayoría de los estudiantes que asistieron al curso, desarrollo del pensamiento computacional con LEGO Mindstorm, tuvieron un buen desempeño en la materia programación estructurada, demostrando con estos resultados que el curso, permitió la estimulación, fortalecimiento de los conceptos referentes al pensamiento computacional.

7.7. Diagnostico Test Inicial

En la tabla 3 se muestran los resultados referente al desempeño de los estudiantes en la aplicación del test inicial, las cinco primeras preguntas tienen un valor de 16 puntos y la última 20 puntos para completar un total de 100 puntos.

Las preguntas realizadas en el test inicial evalúan las competencias de los estudiantes en pensamiento algorítmico, pensamiento convergente y divergente, resolución de problemas, pensamiento lógico matemático, abstracción, comprensión lectora, entre otras. Estas áreas son parte importante en el desarrollo del pensamiento computacional.

El propósito de aplicar el test inicial es establecer el nivel real de los estudiantes antes de iniciar el proceso de enseñanza-aprendizaje; detectar carencias, lagunas o errores que puedan dificultar el logro de los objetivos planteados, establecer metas razonables a fin de emitir juicios de valor sobre los logros en el desarrollo del curso pensamiento computacional con LEGO Mindstorm y con todo ello adecuar el tratamiento pedagógico a las características y peculiaridades de los estudiantes.

Tabla 3 *Resultados del test inicial*

Tema	Puntajes Estudiantes					
	E1	E2	E3	E4	E5	E6
Pensamiento Algorítmico, Pensamiento Abstracto, Toma De Decisiones, Comprensión Lectora	8	8	8	8	8	8
Pensamiento Algorítmico, Comprensión Lectora, pensamiento divergente y convergente, Resolución de problemas	16	0	0	0	12	16
Pensamiento Algorítmico, Pensamiento Abstracto	16	12	16	16	12	16
Pensamiento Algorítmico, creatividad	16	8	16	16	8	16
Pensamiento lógico matemático y comprensión lectora	16	12	12	12	12	16
Pensamiento lógico matemático	10	15	10	10	15	15
Total	82	55	62	62	67	87
Resultado	13.6	9.17	10	10	11	15

Elaboración: Propia

A continuación, se observan las puntuaciones que obtuvieron los estudiantes en cada una de las actividades planteadas en el test inicial

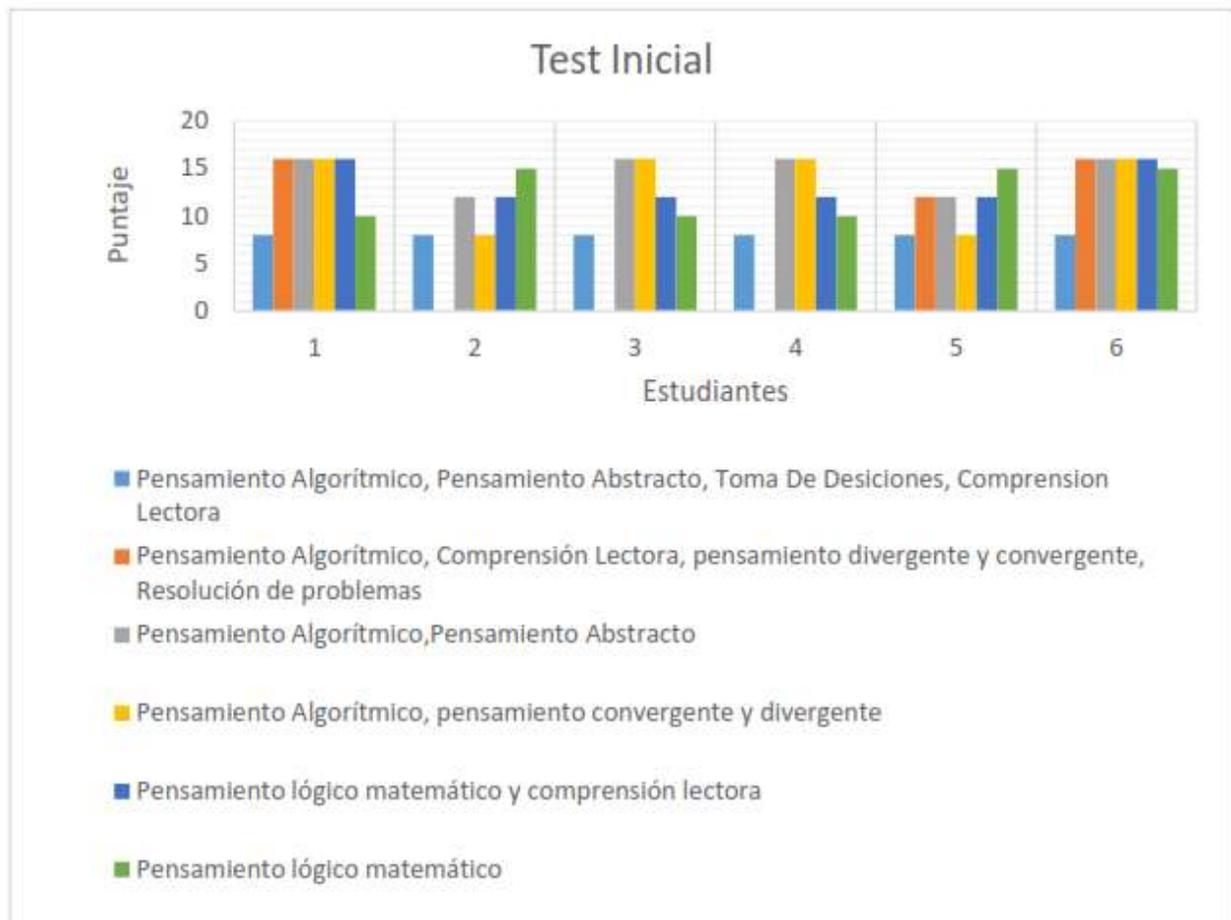


Figura 22 Resultados del test-inicial

En la figura 21, se muestra el rendimiento de los 6 estudiantes que asistieron al curso “Pensamiento computacional con LEGO Mindstorm”, cada barra de la figura representa una actividad y áreas que se evaluaron en el test inicial, que están relacionadas al pensamiento computacional, la comprensión lectora, pensamiento algorítmico, pensamiento convergente, divergente y la resolución de problemas, una vez, calificado el test inicial, se encontró que los estudiantes presentaban un bajo rendimiento en las diferentes áreas evaluadas.

Para la actividad 1, que está orientada al pensamiento algoritmo, pensamiento abstracto, toma de decisiones y comprensión lectora, se encontró que el estudiantado obtuvo en promedio el 50% de aciertos en este punto, demostrando un bajo rendimiento en el desarrollo de la actividad. Esta actividad consistió en un laberinto donde los estudiantes debían encontrar la ruta más corta nombrándola con la letra A y la más larga con la B.

En la actividad 2, donde se evalúan los componentes del pensamiento computacional tales como el pensamiento algorítmico, comprensión lectora, pensamiento divergente y convergente, resolución de problemas, se observó que el 67% de los estudiantes presentan dificultades en el desarrollo de la actividad. Esta actividad consistía en proponer una serie de pasos algorítmicos que llegarían a la solución de un problema.

En el desarrollo de la actividad 3, que está orientada a evaluar el pensamiento algorítmico y pensamiento abstracto, se encontró que el 66% del estudiantado obtuvo el puntaje máximo de 16 puntos, mientras que 34% obtuvo un puntaje de 12 sobre 16. Se observa que los participantes presentan falencias en estas áreas, las cuales deben ser reforzadas. Esta actividad consistía en dibujar en unas cuadrículas de 6x6, letras o caracteres propuestos por el tutor y describir en una serie de pasos algorítmicos el dibujo, cabe mencionar que el estudiante contó con un ejemplo ilustrativo. Sin embargo algunos de ellos no lograron realizar la actividad correctamente, esto puede tratarse de problemas de atención o poca disposición del estudiante en desarrollar la actividad.

En la actividad 4, que evaluaba el pensamiento algorítmico, el pensamiento convergente y divergente de los estudiantes, se presenta que el 34% obtuvo la puntuación máxima que es 16, mientras que el 66% obtuvo 12 puntos, notando dificultades en estas competencias. En esta actividad los estudiantes debían ordenar y enumerar una serie de pasos propuestos para la realización de una receta de cocina. Se observó que las dificultades estaban relacionadas a la

manera en que el estudiante enumeraba y escribía la receta, teniendo en cuenta un orden lógico y la coherencia en los pasos.

En la actividad 5, que estaba orientada al pensamiento lógico matemático y comprensión lectora, se encontró que el estudiantado tiene dificultades en estas competencias, debido a que el 66,66% obtuvo 12 puntos sobre 16, mientras que el 33,33% restante obtuvo un puntaje de 16 sobre 16. En esta actividad se realizó un ejercicio matemático que consistía en completar un cuadro mágico, siguiendo una serie de instrucciones. Se observó dificultades en el desarrollo de la actividad, debido a que algunos estudiantes no leían las instrucciones de manera detallada, lo cual causaba confusión en la resolución de la actividad.

El desarrollo de la actividad 6, se orientaba a evaluar el pensamiento lógico matemático, se observa que el 50% de los estudiantes obtuvo 10 puntos sobre 20 y el otro 50% restante obtuvo 15 sobre 20, notando dificultades en estas competencias. En esta actividad se plantean problemas matemáticos que el estudiante debe resolver, seleccionando la respuesta correcta y argumentándola. Se observa que el estudiantado tiene dificultades en estas áreas, pueden ser lagunas de aprendizaje correspondiente a su periodo académico en la secundaria.

En este análisis se observa el bajo rendimiento de los estudiantes en las diferentes áreas de pensamiento computacional y la importancia de intervenir haciendo uso del Curso de Desarrollo del pensamiento computacional con LEGO Mindstorm.

7.8. Diagnostico Test Final

Una vez finalizado el curso (Desarrollo del pensamiento computacional con LEGO Mindstorm), se aplicó el Test Final, con el fin, de evaluar de qué forma mejoraron los estudiantes en las diversas competencias del pensamiento computacional, mencionadas con anterioridad. A continuación, se ven reflejados los resultados de dicho test, las cuatro primeras preguntas tienen un valor de 18 puntos y la última 28 puntos para completar un total de 100 puntos.

Tabla 4 *Resultados Test Final*

Tema	Puntajes Estudiantes					
	E1	E2	E3	E4	E5	E6
Pensamiento Algorítmico, Pensamiento Abstracto, Toma De Decisiones, Comprensión Lectora	18	0	0	0	18	18
Pensamiento Algorítmico, Comprensión Lectora, pensamiento divergente y convergente, Resolución de problemas	0	18	0	0	0	18
Pensamiento Algorítmico y Pensamiento Abstracto	18	18	0	0	18	9
Pensamiento lógico matemático y comprensión lectora	18	18	0	0	18	18
Pensamiento lógico matemático y comprensión lectora	25	8	0	0	28	28
Total	79	62	0	0	82	91
Resultado	15.8	12	0	0	16	18

Elaboración: Propia.

Como se evidencia en la Tabla 4, los estudiantes mejoraron considerablemente en su rendimiento en las competencias del pensamiento computacional, en la mayoría de las ocasiones lograron el puntaje más alto en cada una de las actividades desarrolladas, cabe mencionar que las actividades de este test se realizaron teniendo en cuenta el test inicial, se puede decir que tienen el mismo nivel de dificultad.

En la siguiente imagen se observan los resultados de la aplicación del Test Final, para analizar estos resultados, no se tendrán en cuenta los estudiantes que desertaron del curso.

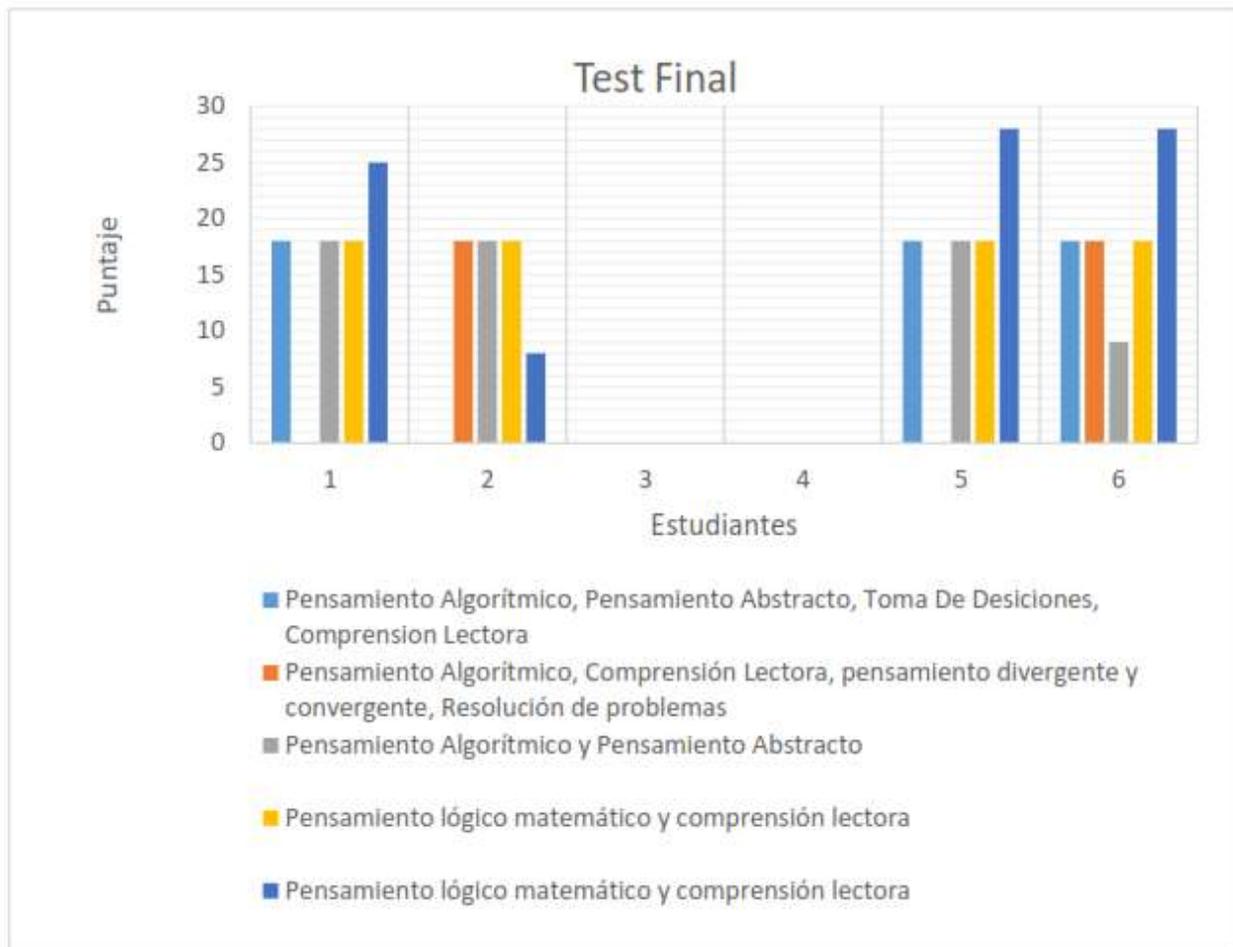


Figura 23. Resultados del test final

En la actividad 1, que estaba orientada al pensamiento algorítmico, pensamiento abstracto, toma de decisiones, comprensión lectora, se puede observar que los estudiantes mejoraron en estas competencias debido a que el 75% logró el puntaje más alto en la actividad. En esta actividad el participante debía desarrollar un algoritmo, usando una simbología propuesta que le ayudaría en el armado de una figura, se observa que el estudiante comprende el concepto de algoritmia, es capaz de aplicarlo en la resolución de problemas.

En la actividad 2, que hace referencia a las áreas del pensamiento computacional tales como el pensamiento algorítmico, comprensión lectora, pensamiento divergente y convergente, resolución de problemas, se observa que el 50% del estudiantado obtuvo el puntaje más alto en la actividad, el otro 50% tuvo dificultades en estas áreas. En esta actividad se le pide al estudiante que determine la ruta más corta desde un punto A hasta un punto B, seleccionando la respuesta correcta.

En la actividad 3, que estaba orientada al pensamiento algorítmico y pensamiento abstracto, se encontró que los estudiantes obtuvieron un promedio del 75% en el desarrollo de la actividad, se observa avances en las competencias referentes al pensamiento computacional. En esta actividad el estudiante debe ordenar, enumerar, lógica y coherentemente los pasos para la realización de un puré de papa.

En la actividad 4, que se orientó al pensamiento lógico matemático y la comprensión lectora, se encontró que los estudiantes obtuvieron un 100% en el desarrollo de la actividad, demostrando un avance en el desarrollo de las habilidades referentes al pensamiento computacional. En esta actividad se le presenta al estudiante un cuadro mágico reversible que resolvería con la ayuda de unas instrucciones, seleccionando la respuesta correcta.

En la actividad 5, se evaluó el pensamiento lógico matemático y la comprensión lectora, se observó que los estudiantes obtuvieron un promedio del 50% en la realización de la actividad,

el otro 50% obtuvo un puntaje promedio de 25 puntos sobre 28, lo que evidencia que los estudiantes mostraron un progreso en los diferentes elementos abordados. En esta actividad el estudiante debía realizar una serie de ejercicios matemáticos, seleccionando la respuesta correcta.

En la siguiente imagen, se muestra la comparación entre el Test Inicial (representado por la línea azul) y el Test final (representado por la línea naranja), donde se ve claramente, un progreso en las competencias del pensamiento computacional, trabajadas en la presente propuesta.

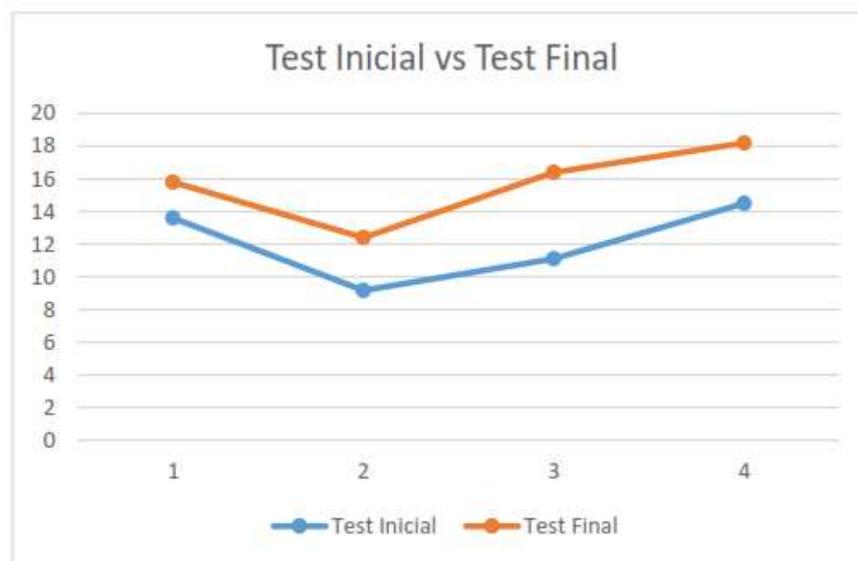


Figura 24 Cotejamiento de los test aplicados

En los resultados del cotejamiento del test inicial con el test final se observa que la implementación de las guías del curso el resultado fue favorable en el proceso de aprendizaje de los estudiantes. Se considera de igual forma, relevante que este tipo de cursos o actividades se realicen de manera periódica o seguida en las aulas de clase, para enriquecer el aprendizaje, debido a que, por la cualidad plástica del cerebro, estos logros alcanzados pueden perderse con el tiempo si se deja de practicar en dichas áreas.

8. CONCLUSIONES

- Se determina que la teoría constructivista es la más acertada para esta investigación, ya que permite al aprendiz construir estructuras mentales propias, generando nuevos conocimientos, analizando problemas, entendiéndolos y generando posibles soluciones, además de implementar una herramienta informática, que pueda apoyar el proceso de aplicación de la solución.
- Se diseñaron 6 guías prácticas enfocadas al desarrollo del pensamiento computacional utilizando la herramienta LEGO Mindstorm. Las guías orientaban al estudiante en la adquisición de los conceptos y habilidades del pensamiento computacional.
- Durante el desarrollo del curso se observó que los estudiantes se encontraban motivados y dispuestos a adquirir los conceptos relacionados al pensamiento computacional, además de que estos aprendizajes fueron bien recibidos, reforzaban las clases de la materia programación estructurada, esencial en el crecimiento profesional de los aprendices en la carrera de ingeniería de sistemas.
- Se encontró que los estudiantes participantes de primer semestre de la carrera ingeniería de sistemas de la Universidad de Pamplona, poseen habilidades y características para el aprendizaje de la programación.
- En el proceso del desarrollo del curso se observa que los estudiantes tienen diversas formas de aprendizaje, en este proyecto se aplicó una metodología constructivista que se enfocaba en construir, ordenar y estructurar el conocimiento de tal forma que el estudiante fuera capaz de asimilarlo, teniendo un tutor que estaba dispuesto a colaborar y asesorar al aprendiz en sus dudas e inquietudes.
- Con la aplicación del test final se vio el resultado del curso, donde se observa un desarrollo y fortalecimiento en los componentes del pensamiento computacional, los participantes del curso fueron capaces de diseñar, estructurar y desarrollar algoritmos para resolver un problema. Además se observa un avance en el área de comprensión lectora, pensamiento lógico matemático y otras habilidades.
- El uso del set robótico LEGO Mindstorm, facilitó el aprendizaje en gran medida, fue un componente donde el estudiante experimentó sus primeros pasos en la programación, además

fortaleció las habilidades en la resolución de problemas, realizando algoritmos de manera divertida y entretenida.

- En los resultados del test final se puede validar el impacto del curso, Desarrollo del Pensamiento Computacional con LEGO Mindstorm, ya que los estudiantes mejoraron su rendimiento en las diferentes áreas del pensamiento computacional, lo que valida la materia de programación como parte importante en el desarrollo del pensamiento computacional.

9. RECOMENDACIONES

- Se recomienda que este curso se integre al pensum del programa ingeniería de sistemas, ya que el estudiante al ser un curso libre y no tener una nota, tienden a desertar del aprendizaje.
- Mejorar el tiempo a la hora de desarrollar las guías.
- Se recomienda integrar la placa Arduino con el kit LEGO Mindstorm, para ampliar el horizonte temático.
- Para futuros trabajos se recomienda estudiar el uso de metodologías, técnicas y herramientas enfocadas a desarrollar el pensamiento computacional en estudiantes, de manera autónoma en un espacio más interactivo y práctico, ya que esto permite la flexibilidad y dinamismo en el proceso de aprendizaje.
- El uso de la internet es un recurso a un sin explorar en estos temas, es posible encontrar inmensidad de información y actividades que ayuden al estudiante en el procesos de formación y desarrollo del pensamiento computacional. Estos recursos pueden ser impartidos en las clases escolásticas o universitarias. El docente juega un papel importante ya que es el encargado de motivar al estudiante en la búsqueda de recursos que ayuden al desarrollo de estos conocimientos, estos conocimientos ayudaran al estudiantado a desempeñarse en el ámbito laboral y de la vida diaria.
- Para dinamizar el proceso de aprendizaje, se propone el uso de otras herramientas o plataformas, como Scratch, donde se puede enseñar los conceptos o habilidades relacionadas al pensamiento computacional. Estas investigaciones pueden compararse y brindar una estadística de la mejor herramienta o plataforma para abordar la enseñanza del pensamiento computacional.

- Se recomienda que para futuras aplicaciones de los set LEGO Mindstorm, en la enseñanza del pensamiento computacional, se deben dividir el estudiantado en grupos de dos, para que los aprendices logren trabajar de manera óptima en la resolución de los recursos propuestos.

REFERENCIAS

- Alice. (s.f.). About Alice. Recuperado de <https://www.alice.org/about/>
- Álvarez Rodríguez, M. (2017). Desarrollo del pensamiento computacional en educación primaria: una experiencia educativa con scratch. 2nd ed. [ebook] Revista de Ciències de l'Educació. Available at: <https://revistes.urv.cat/index.php/ute/article/download/1820/1829> [Accessed 3 Aug. 2019].
- Altamirano Carmona, E., Becerra Correa, N., & Nava Casarrubias, A. (2010). Hacia una educación conectivista. *Revista Alternativa*, (22), 22-24. Recuperado de https://www.researchgate.net/publication/264790115_Hacia_una_educacion_conectivista
- Barón, R. A.(sf). Tendencias educativas con TI Conectivismo [Reseña].*Red CGTI* Recuperado de https://portal.ucol.mx/content/micrositios/260/file/conectivismo_resena.pdf
- Berardinelli, D., Padrino, F., Ramírez, A., & Páez, E. (2010). Teorías Educativas. Recuperado de <https://sites.google.com/site/jhailpalma/proyecto/carta-de-admision>
- Barrera Capot, R., & Montaña Espinoza, R. (2015). Desarrollo del Pensamiento Computacional con Scratch [Ebook]. Santiago de Chile: Nuevas Ideas en Informática Educativa TISE. Recuperado de <http://www.tise.cl/volumen11/TISE2015/616-620.pdf>
- Basogain, X, Olabe M.A, Olabe J.C, Rico M.J, Rodríguez, L, Amórtégui, M. (sf). Pensamiento computacional en las escuelas de Colombia: colaboración internacional de innovación en la educación. Recuperado de <http://recursos.portaleducoas.org/sites/default/files/5188.pdf>
- Basogain, X., Olabe, M. A., y Olabe, J. C. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. *Revista de Educación a Distancia*, 46(6), 1 - 33. Recuperado de <https://www.um.es/ead/red/46/Basogain.pdf>

- Cacique Borrego, V., & Camarillo Gómez, K. (2013). LEGO Digital Designer en la robótica educativa [Ebook] (p. 1). Celaya, Guanajuato, México. Recuperado de https://www.researchgate.net/profile/Karla_Camarillo-Gomez/publication/275343885_LEGO_Digital_Designer_en_la_robotica_educativa/links/553923b30cf226723ab9a398.pdf
- González, D.J. (2002). El Constructivismo: Reseña Del Libro Corrientes Constructivistas De Royman Pérez Miranda Y Rómulo Gallego – Badillo. *Instituto Superior Pedagógico. Enrique José Varona*. (19) Recuperado de <http://pepsic.bvsalud.org/pdf/rcp/v19n2/14.pdf>
- Giunti, G. (2010). Reseña de tecnología educativa: recursos, modelos y metodologías. *Revista del Instituto de Investigaciones Bibliotecológicas*, (22), 128 - 131. Recuperado de www.redalyc.org/articulo.oa?id=263019489008
- Heredia Escorza, Y., & Sánchez Aradillas, A. (2013). Teorías Del Aprendizaje En El Contexto Educativo [Ebook] (pp. 8-9). Monterrey: Editorial Digital Tecnológico de Monterrey. Recuperado de <http://prod77ms.itesm.mx/podcast/EDTM/P231.pdf>
- Hurtado, J. (2000). *Metodología de la investigación holística*. Ciudad, país: editorial.
- Marin, A. (2013). ¿Qué es code.org?. Recuperado de <https://alexismarin.wordpress.com/2013/03/13/que-es-code-org/>
- Microsoft. (s.f.). Welcome to Small Basic!. Recuperado de <https://smallbasic-publicwebsite.azurewebsites.net/Default.aspx>
- Morales, S. L. (s.f.). Programación para niños con logo. Recuperado de virtual.cudi.edu.mx:8080/access/content/group/981e62d6-89ea-4e3e-baa1-96b45cf0a570/2012_03_27/sandra_morales.pdf
- Moreno, J. (2014). ¿Qué es el pensamiento computacional?. Recuperado de <https://programamos.es/que-es-el-pensamiento-computacional/>

- Mindstorms LEGO. (2015). Recuperado de <https://www.lego.com/es-es/mindstorms/downloads>
- Posada Prieto, F. (2019). Robótica Lego EV3. Recuperado de <http://canaltic.com/rb/legoev3/index.html>
- Pérez P.M, (2004). Revisión De Las Teorías Del Aprendizaje Más Sobresalientes Del Siglo XX. *Revista Interinstitucional de Investigación Educativa*. (10), 40 – 41. Recuperado de <http://www.redalyc.org/html/311/31101003/>
- Rodriguez Rodriguez, A., & Molero de Martins, D. (2008). CONECTIVISMO COMO GESTIÓN DEL CONOCIMIENTO [Ebook] (p. 77). Maracaibo-Venezuela. Retrieved from <https://dialnet.unirioja.es/descarga/articulo/2937200.pdf>
- Ros, M. (2019). Pensamiento computacional. Una tercera competencia clave (III): ¿Qué es el Pensamiento Computacional? Una definición. Recuperado 30 Julio 2019, de <https://red.hypotheses.org/1069>
- Scratched. (s.f.). What is Scratch? Recuperado de scratched.gse.harvard.edu/about
- Stager, G. (2003). En pro de los computadores (Parte I). Recuperado de eduteka.icesi.edu.co/articulos/ProComputadores
- The Computer Clubhouse Network. (2008). A Place to Be Your Best: Youth Outcomes in the Computer Clubhouse. Boston, Estados Unidos: Center for Technology in Learning, SRI International. Recuperado de theclubhousenetwork.org/sites/default/files/A%20Place%20to%20Be%20Your%20Best%20FINAL%207-25.pdf
- Travieso, C. (s.f.). ¿Qué es Scratch?. Recuperado de static.esla.com/img/cargadas/2267/Documentación%20Scratch.pdf
- Trujillo L.M. (2017). Teorías pedagógicas contemporáneas. *Fundación Universitaria del Área Andina*. Recuperado de

<http://digitk.areandina.edu.co/repositorio/bitstream/123456789/825/1/Teor%C3%ADas%20pedag%C3%B3gicas%20contempor%C3%A1neas.pdf>

Toro, L. (2017). Crear construcciones tipo LEGO con el renovado LeoCAD. Retrieved from <https://blog.desdelinux.net/crear-construcciones-tipo-lego/>

Valdez Alejandro F.J. (2012). Teorías educativas y su relación con las tecnologías de la información y de la comunicación (TIC). *Universidad Nacional Autónoma de México México*. Recuperado de <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC433653/?tool=pubmed>

Valverde Berrocoso, J., Fernández Sánchez, M., & Garrido Arroyo, M. (2015). Contenidos de la asignatura (Tecnología, Programación y Robótica) por cursos [Imagen]. Retrieved from https://www.um.es/ead/red/46/valverde_et_al.pdf

Valverde, J., Fernández, M. R., y Garrido, M. C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, 46(3), 1-18. Recuperado de https://www.um.es/ead/red/46/valverde_et_al.pdf

Vilanova, G. (2018). Tecnología Educativa para el Desarrollo del Pensamiento Computacional. *Sistemas, Cibernética e informática*, 15(3), 25 - 32. Recuperado de [www.iiisci.org/journal/CV\\$/risci/pdfs/CA074QW17.pdf](http://www.iiisci.org/journal/CV$/risci/pdfs/CA074QW17.pdf)

Wing J.M, (2006), Computational Thinking. *Communications of the Acm* . (49), 33 - 35. Recuperado de <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf>

Zúñiga, M., Rosas, M., Fernández, J., & Guerrero, R. (2014). El Desarrollo del Pensamiento Computacional para la Resolución de Problemas en la Enseñanza Inicial de la Programación [Ebook]. San Luis, Argentina: Workshop de Investigadores en Ciencias de la Computación. Recuperado de <http://hdl.handle.net/10915/41352>

Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia*, (46), 1 - 47. Recuperado de <https://revistas.um.es/red/article/view/240321/183001>