

**DIAGRAMAS NASSI-SHNEIDERMAN Y DIAGRAMAS DE FLUJO COMO
TÉCNICAS PARA EL DESARROLLO DEL PENSAMIENTO ALGORÍTMICO**



Andrés Mauricio Molina Monguí

Luis Leonardo Orjuela Preciado

Universidad Pedagógica Nacional

Maestría en Tecnologías de la Información Aplicadas a la Educación

Facultad de Tecnología

Bogotá, D.C.

2018

**DIAGRAMAS NASSI-SHNEIDERMAN Y DIAGRAMAS DE FLUJO COMO
TÉCNICAS PARA EL DESARROLLO DEL PENSAMIENTO ALGORÍTMICO**

**Trabajo para optar al título de Magíster en Tecnologías de la Información Aplicadas
a la Educación**

Andrés Mauricio Molina Monguí

Luis Leonardo Orjuela Preciado

Director

David Macías

Magister en Tecnologías de la Información Aplicadas a la Educación

Universidad Pedagógica Nacional

Maestría en Tecnologías de la Información Aplicadas a la Educación

Facultad de Tecnología

Bogotá, D.C.

2018

Derechos de autor

“Para todos los efectos, declaró que el presente trabajo es original y de mi total autoría; en aquellos casos en los cuales he requerido del trabajo de otros autores o investigadores, he dado los respectivos créditos”. (Artículo 42, parágrafo 2, del Acuerdo 031 del 4 de diciembre de 2007 del Consejo Superior de la Universidad Pedagógica Nacional)



Este trabajo de grado se encuentra bajo una Licencia Creative Commons de **Reconocimiento – No comercial – Compartir igual**, por lo que puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

Dedicatoria**A Dios.**

Por permitirnos culminar este proyecto académico y alcanzar esta meta, por darnos la fortaleza y cordura que necesitamos en los momentos más difíciles, además de su infinita bondad y amor.

A nuestras familias

Por habernos apoyado en todo momento, por ser nuestro cronograma parlante, por ser la motivación constante que nos ha permitido ser personas de bien, pero más que nada, por su amor.

Agradecimientos

Agradecemos al profesor David Macías, tutor, asesor y director de tesis, porque gracias a su conocimiento y sugerencias pudimos acercarnos al objetivo.

A todos los docentes, directivos y personal administrativo del programa de Maestría en Tecnologías de la Información Aplicadas a la Educación de la Universidad Pedagógica Nacional, quienes con profesionalismo, vocación y amor por su trabajo aportaron significativamente en mi aprendizaje como educador y como persona.

A la Institución Educativa IED Unión Europea, Bogotá, por permitirnos desarrollar la investigación y prestarnos las instalaciones para realizarla.

A los estudiantes de grado sexto por asistir en contra jornada y los fines de semana, aun sabiendo que no había refrigerio, sino por la motivación de aprender algo nuevo.

A nuestras familias por ser incondicionales y apoyarnos en este viaje para convertirnos en Magísteres de las Tecnologías Aplicadas a la Educación.

Resumen analítico educativo “RAE” de la tesis

1. Información General	
Tipo de documento	Tesis
Acceso al documento	Universidad Pedagógica Nacional. Biblioteca Central
Título del documento	Diagramas Nassi-Shneiderman y Diagramas de Flujo como Técnicas para el desarrollo del Pensamiento Algorítmico
Autor(es)	Andrés Mauricio Molina Monguí, Luis Leonardo Orjuela Preciado
Director	David Macías Mora
Publicación	Bogotá, Universidad Pedagógica Nacional, 2018. p, 95
Unidad Patrocinante	Universidad Pedagógica Nacional, Facultad de Ciencia y Tecnología, Departamento de Tecnología, Maestría en Tecnología de la Educación Aplicada a la Educación.
Palabras Claves	Diagramación, Nassi-Shneiderman, Diagrama de Flujo, resolución de problemas, problemas de razonamiento algorítmico, ambiente digital, aplicación móvil

Descripción
<p>Tesis de maestría en Tecnologías de la información aplicadas a la educación, en la cual se examina la incidencia dos técnicas de diagramación: Diagramas de Nassi-Shneiderman y Diagramas de Flujo, en la solución de problemas de razonamiento algorítmico. La investigación de corte correlacional se desarrolla en la Institución Educativa Unión Europea, Bogotá D. C., con estudiantes de grado sexto entre los 10 y 12 años de edad.</p>

Fuentes
<p>Alfonso Combita, D. J. (2010). Transferencia análoga de estrategias de solución de problemas de una forma de representación a un lenguaje de programación formal utilizando estructuras selectivas dobles. Bogotá: Universidad Pedagógica Nacional.</p> <p>Erro, L. (2016). Pensamiento computacional. [online] Pensamiento Computacional Curso PC. Available at: http://www.pensamientocomputacional.org/index.php/curso-pc [Accessed 7 Mar. 2016]</p>

- Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. ISSEP, 159-168.
- Grover, S. (2009). Computer Science Is Not Just for Big Kids. *Learning & Leading with Technology*, 27-30.
- Guerrero Posada, M., & García Orozco, J. (2016). DESARROLLO DEL PENSAMIENTO ALGORÍTMICO CON EL APOYO DE OBJETOS DE APRENDIZAJE GENERATIVOS. *Pixel-Bit. Revista de Medios y Educación*.
- Katai, Z. (2015). The challenge of promoting algorithmic thinking of both sciences- and humanities-oriented learners. *Journal of Computer Assisted Learning*, 287-299.
- Lamagna, E. (2015). Algorithmic thinking unplugged. *Journal of Computing Sciences in Colleges*, 45-52.
- Legris, J (2012). Visualizar y manipular. Sobre el razonamiento diagramático y la naturaleza de la deducción. En *Visualização nas Ciências Formais*, comp. por Abel Lassalle Cassanave y Frank Thomas Sautter. Londres, College Publications, p. 89-103. ISBN 978-1-84890-069-1. <http://www.collegepublications.co.uk/filosofia/>
- Nassi, I., & Shneiderman, B. (1973). Flowchart Techniques for structured programming. *SIGPLAN NOTICES*, 12-26.
- National Research Council. (1999). The Intellectual Framework of Fluency with Information Technology. En C. o. Literacy, Being Fluent with Information Technology (págs. 15-40). Washington, D.C.: National Academies Press.
- Pérez Pérez, I., & Monzalvo López, C. A. (2010). Análisis del desempeño de los estudiantes en el diseño y construcción de algoritmos secuenciales, selectivos e iterativos. V *Congreso Universitario en Tecnologías de Información y Comunicaciones*.
- Pérez Pérez, I., & Monzalvo López, C. A. (2011). Nivel de utilización de las técnicas de diagramación por parte de los estudiantes, en el diseño de algoritmos secuenciales, selectivos e iterativos. *Congreso Universitario en Tecnologías de Información y Comunicaciones 2011*.
- Pérez Pérez, I. & Monzalvo López, C. A. (2012). Diseño de una propuesta de herramienta de diagramación para el desarrollo de algoritmos para computadora. VII Congreso Universitario en Tecnologías de Información y Comunicaciones 2012. Área Académica de Computación. ICBI-UAEH. México. ISBN: En trámite
- Prado, A., & Lamas, N. (2014). Alternativas para la enseñanza de pseudocódigo y diagrama de flujo. *Revista Electrónica Iberoamericana de Educación en Ciencias y Tecnología*, 102-113.
- Román González, M., Pérez González, J. C., & Jiménez Fernández, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general. *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)*.
- Sánchez Román, G., Guerrero García, J., Collazos Ordóñez, C. A., Tapia Cortes, C., & Mocenchua Mora, D. (2016). Propuesta de arquitectura de Sistema Tutor Inteligente para desarrollar las habilidades algorítmicas. *Revista Iberoamericana de Informática Educativa*, 30-35.
- Shneiderman, B. (27 de 05 de 2003). *A short history of structured flowcharts (Nassi-Shneiderman Diagrams)*. Obtenido de <https://www.cs.umd.edu/hcil/members/bshneiderman/nsd/>
- Vidal, C., Cabezas, C., Parra, J., & López, L. (2015). Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile. *Formación universitaria*, 23-32.

Villalobos Salcedo, J. A. (2009). Cupi2 Una solución integral al problema de Aprender/enseñar a Programar. *XXVII Reunión Nacional de Facultades de Ingeniería y VI Encuentro Iberoamericano de Instituciones de Enseñanza de la Ingeniería*, 1-37.

Wing, J. (2006). Computational Thinking. *Communications of the ACM* 49, 33-35.

Yoder, C. M., & Schrag, M. L. (1978). NASSI-SCHNEIDERMAN CHARTS AN ALTERNATIVE TO FLOWCHARTS FOR DESIGN. *IBM Corporation System Products Division Endicott, Nueva York 13760*, 386-393.

Contenidos

Introducción

Presentación de la investigación (Problema de investigación, Justificación y Objetivos)

Estado del arte

Marco teórico

Descripción del desarrollo tecnológico

Metodología

Pregunta de investigación

Hipótesis

Participantes

Variables

Diseño

Instrumentos

Pruebas Pre test para la medición de la eficiencia en la solución de problemas algorítmicos

Prueba Post test para la medición de la eficiencia en la solución de problemas algorítmicos

Etapas de la investigación

Análisis de datos

Resultados

Discusión de los resultados

Conclusiones y recomendaciones

Referencias

Anexos

Metodología

La investigación es de tipo cuasiexperimental con dos grupos, uno que interactúa con la técnica de diagramación Nassi-Shneiderman y otro que interactúa con la técnica de diagramación Diagramas de flujo. La muestra la componen 60 estudiantes de los grados sextos del Colegio Unión Europea ubicado en la localidad 19 de Ciudad Bolívar. Se consideró como variable dependiente: desarrollo del pensamiento algorítmico y como variable independiente Resolución de problemas utilizando Diagramas de flujo, Resolución de problemas utilizando Diagramas Nassi-Shneiderman. Para obtener el desarrollo del pensamiento algorítmico se midió la eficiencia para resolver problemas de índole algorítmico y el tiempo de resolución de problemas a través de una app desarrollada por los investigadores. Se consideran tres hipótesis: Hipótesis nula. No existe diferencia en el desarrollo del pensamiento algorítmico entre dos grupos de estudiantes que resuelven problemas en un ambiente mediado por computador que interactúa bajo la representación del conocimiento basado en diagramas de flujo y otro bajo la representación de conocimiento basado en diagramas Nassi-Shneiderman. Hipótesis 1. Los estudiantes que resuelven problemas en interacción con un ambiente de aprendizaje que usa representación de conocimiento basada en diagramas de flujo se desempeña mejor, que quienes interactúan con el mismo ambiente pero con representación de conocimiento basado en Nassi-Shneiderman. Hipótesis 2. Los estudiantes que resuelven problemas en interacción con un ambiente de aprendizaje que usa representación de conocimiento basada en diagramas de Nassi-Shneiderman se desempeña mejor, que quienes interactúan con el mismo ambiente pero con representación de conocimiento basado en diagramas de flujo. La recolección de la información se llevó en dos momentos: 1) validación de la app Prest basada en el Test de Pensamiento Computacional: diseño y psicometría general midiendo los niveles de eficiencia y tiempo de resolución en tres problemáticas. 2) validación de la resolución de problemas algorítmicos.

El análisis de la información cuantitativa, recolectada a través de las diferentes pruebas, se hizo mediante métodos de estadística inferencial, los cuales permiten generalizar los resultados y de esta forma comprobarlos respecto a las hipótesis planteadas. Estos análisis se realizaron utilizando el programa de estadística SPSS Statistics 21.

Conclusiones

Se elabora la app “Prest”, diseñada para medir los niveles de eficiencia y tiempo utilizado en la solución de problemas como lo indica Román y Pérez, (2015) en el test diagnóstico de pensamiento computacional.

Al realizar la intervención a grupos con edades entre los 10 y 14 años, se debe presentar ambientes interactivos y llamativos en cuanto al diseño y dinamismo, para lograr que los estudiantes mantengan un alto grado de interés, participación y entusiasmo en la exploración de tareas y problemas para ser resueltos.

Los resultados obtenidos permitieron determinar que el uso de ambientes de trabajo dinámicos fue relevantes en esta investigación. La alta interactividad entre las herramientas utilizadas por estudiantes fue un factor clave en la construcción del conocimiento y de habilidades cognitivas demostradas en la resolución de los problemas durante la intervención.

En los problemas secuenciales (Tipo I) los resultados estadísticos validan la hipótesis nula, que enuncia que el desarrollo del pensamiento algorítmico no está influenciado s por la técnica de diagramación aplicada, cuando se evalúa la eficiencia en el número de eventos. Por otra parte, el tiempo empleado para la elaboración de la estrategia y solución de la tarea dada, en los problemas tipo I fue menor en los grupos que tuvieron un tratamiento con la técnica de diagramación Nassi-Shneiderman, incluyendo que la solución de un problema implica: la interpretación del problema, organización, el planteamiento de la solución y la ejecución.

Para los problemas tipo II y III los resultados estadísticos validan la hipótesis alterna, la cual plantea que los estudiantes que resuelven problemas en interacción con un ambiente de aprendizaje que usa representación de conocimiento basada en diagramas de Nassi-Shneiderman se desempeña mejor, que quienes interactúan con el mismo ambiente pero con representación de conocimiento basado en diagramas de flujo, es decir, el pensamiento algorítmico, aunque implícito en los procesos de resolución de problema, también demuestran el cumplimiento de ésta hipótesis en cuanto a la eficiencia del proceso desarrollado.

Cabe resaltar como lo menciona Forsythe, Skeenan, Organick y Stenberg (1975) para resolver problemas: “La construcción de algoritmos y sus diagramas..... es esencialmente un proceso de solución de problemas. Para enseñar a resolver problemas, tenemos que proceder en forma muy diferente a la que estamos acostumbrados. No basta simplemente presentar el desarrollo directo de un elegante diagrama..... Debemos, por el contrario, mostrar cómo se llega a la solución final. Escoger el lugar para iniciar el problema. Se debe ver también algunos de los falsos comienzos y omisiones, algunos de los algoritmos deficientes obtenidos en nuestros primeros intentos. Sobre todo, usted debe saber que en la construcción de algoritmos primero intentamos obtener alguna clase de solución del problema (buena o mala), después se estudia y se critica la solución y se trata de encontrar formas de mejorarla.” Definitivamente, el conocimiento de cómo utilizar las herramientas de diagramación, no implica el conocimiento de cómo solucionar problemas (Farina, 1983).

Con base en los resultados de la investigación podemos determinar que la técnica adecuada para estimular el pensamiento algorítmico en los estudiantes de grado Sexto del Colegio Unión Europea IED es el diagrama de Nassi-Shneiderman cuando se aborde problemas de iteración y problemas de toma de decisiones, por las mejoras que presenta en la eficiencia en el tiempo y en el número de eventos.

Elaborado por:	Andrés Mauricio Molina Monguí – Luis Leonardo Orjuela Preciado
Revisado por:	David Macías Mora

Fecha de elaboración del Resumen:	03	11	2018
--	----	----	------

Tabla de contenido

Introducción.....	9
1. Presentación de la investigación	11
1.1 Planteamiento del problema	11
1.2 Justificación.....	14
1.3 Objetivos.....	16
1.3.1 Objetivo General.....	16
1.3.2 Objetivos Específicos	16
2 Estado del arte.....	17
3 Marco Teórico.....	27
3.1 Pensamiento Algorítmico	32
3.2 Razonamiento Diagramático	35
3.3 Diagramación.....	37
3.3.1 Diagramas de Flujo.....	38
3.3.2 Diagramas de Nassi-Shneiderman	42
3.4 Clasificación de los problemas	46
3.4.1 Ejercicios de Reconocimiento.	47
3.4.2 Ejercicios algorítmicos o de repetición.....	47

3.4.3	Problemas de traducción simple o compleja.....	48
3.4.4	Problemas de procesos.....	48
3.4.5	Problemas sobre situaciones reales.....	49
3.4.6	Problemas de investigación matemática.....	50
3.4.7	Problemas de puzles.....	50
3.4.8	Historias matemáticas.....	50
3.5	Fases del proceso de resolución de problemas	51
3.6	Dispositivos móviles en la educación.....	54
4	Descripción del desarrollo tecnológico.....	56
4.1	Modelo Pedagógico utilizado en la app.....	56
4.2	Diseño del ambiente digital: comprensión de problemas-app.....	57
5	Metodología	62
5.1	Pregunta de investigación.....	62
5.2	Variables	62
5.2.1	Variables Independientes.....	62
5.2.2	Variable Dependiente	62
5.3	Hipótesis	62
5.3.1	Hipótesis 0.	62
5.3.2	Hipótesis 1.	63
5.3.3	Hipótesis 2	63

5.4	Tipo de Investigación	63
5.5	Población y Muestra	64
5.5.1	Población.	64
5.5.2	Muestra.	64
5.5.3	Grupo 1	64
5.5.4	Grupo 2	64
5.6	Diseño Metodológico	64
6	Análisis de datos	68
7	Resultados	68
7.1	Anova de un factor	72
7.1.1	Prueba Anova de un Factor para Problemas tipo I.	76
7.1.2	Prueba Anova de un Factor para Problemas tipo II.....	78
7.1.3	Prueba Anova de un Factor para Problemas tipo III.....	80
8	Discusión de los resultados.....	91
8.1	Problemas tipo I: Problemas secuenciales.....	91
8.2	Problemas tipo II: Problemas de iteración.....	92
8.3	Problemas tipo III: Problemas de toma de decisiones.....	93
9	Conclusiones y recomendaciones	95
10	Referencias	98

Lista de tablas

Tabla 1 Principales símbolos para elaborar Diagramas de Flujo	40
Tabla 2 Principales símbolos para elaborar Diagramas de Nassi-Shneiderman	45
Tabla 3 Fases y actividades del diagnóstico de los grupos de investigación	57
Tabla 4 Número de Ejercicios por cada problema	61
Tabla 5 Prueba de Kolmogorov-Smirnov para una muestra	69
Tabla 6 Estadísticos descriptivos Eficiencia con base en el tiempo Problemas tipo I	73
Tabla 7 Estadísticos descriptivos Eficiencia con base en el número de eventos Problemas tipo I	73
Tabla 8 Estadísticos descriptivos Eficiencia con base en el tiempo Problemas tipo II.....	74
Tabla 9 Estadísticos descriptivos Eficiencia con base en el número de eventos Problemas tipo II.....	74
Tabla 10 Estadísticos descriptivos Eficiencia con base en el tiempo Problemas tipo III...75	
Tabla 11 Estadísticos descriptivos Eficiencia con base en el número de eventos Problemas tipo III.....	76
Tabla 12 Prueba de Homogeneidad de Varianzas Eficiencia con base en el tiempo problemas tipo I.....	76
Tabla 13 Anova Eficiencia con base en el tiempo. Ejercicios Tipo I.....	77
Tabla 14 Pruebas robustas de igualdad de las medias	77

Tabla 15 Prueba de Homogeneidad de Varianzas Eficiencia con base en el número de eventos problemas tipo I	77
Tabla 16 Anova Eficiencia con base en el número de eventos problemas tipo I	78
Tabla 17 Prueba de Homogeneidad de Varianzas Eficiencia con base en el tiempo problemas tipo II	79
Tabla 18 Anova Eficiencia con base en el tiempo. Ejercicios Tipo II	79
Tabla 19 Prueba de Homogeneidad de Varianzas Eficiencia con base en el número de eventos problemas tipo II	79
Tabla 20 Anova Eficiencia con base en el número de eventos. Ejercicios Tipo II	80
Tabla 21 Prueba de Homogeneidad de Varianzas Eficiencia con base en el tiempo problemas tipo III	80
Tabla 22 Anova Eficiencia con base en el tiempo. Ejercicios Tipo III	81
Tabla 23 Prueba de Homogeneidad de Varianzas Eficiencia con base en el número de eventos problemas tipo III	81
Tabla 24 Anova Eficiencia con base en el número de eventos. Ejercicios Tipo III	82

Lista de Figuras

Figura 1 Clasificación de los problemas según Blanco.....	47
Figura 2 Estrategia de Polya para la resolución de problemas	53
Figura 3 Interfaz para el ingreso a la app	58
Figura 4 Pantalla de Introducción de Prest.....	58
Figura 5 Escenarios Nivel 1 Llegar al objetivo	59
Figura 6 Introducción del nivel 2	59
Figura 7 Escenarios Nivel 2 Objetivos múltiples	60
Figura 8 Introducción del Nivel 3	60
Figura 9 Escenario Nivel 3 Siguiendo Instrucciones	60
Figura 10 Modelo de la metodología.....	67
Figura 11 Histograma y curva normal Eficiencia con base en el tiempo Grupo N-S	70
Figura 12 Histograma y curva normal Eficiencia con base en el tiempo Grupo Flujo	71
Figura 13Histograma y curva normal Eficiencia con base en el número de eventos Grupo N-S	71
Figura 14 Histograma y curva normal Eficiencia con base en el número de eventos Grupo Flujo	72

Lista de anexos

Anexo 1 Screens de la App Prest	104
Anexo 2 Screens Postest	114
Anexo 3 Resultados Pretest y Postest archivos en SPSS	121
Anexo 4 Registro de Usuarios la base de Datos.....	124

Introducción

Acercar a los jóvenes a la generación de Tecnologías para el procesamiento de la información es un tema que interesa a diversas sociedades, ya que en esta época se vivencia una revolución tecnológica que ha modificado la forma en la cual se comunican, interactúan, negocian, se solicitan citas médicas y comercian las personas. Todo ello conduce a actividades que impulsan el desarrollo del país, ya sea, creando tecnología o diseñando objetos que faciliten la vida como se conoce.

Para lograr esta meta primero se debe desarrollar competencias en resolución de problemas específicamente problemas algorítmicos. Al alcanzar esta competencia los jóvenes activarán su propia capacidad mental, ejercitar su creatividad, reflexionar y mejorar sus procesos de pensamiento para afrontar situaciones problemáticas con una actitud crítica (Ferrer, 2000). A su vez, para resolver un problema primero se tiene que comprender el problema, desarrollar un plan, ejecutar el plan, y revisarlo, Polya (1989).

Es por esto, que la presente investigación trata de indagar y profundizar por la segunda fase, al comparar dos técnicas de diagramación y preguntar cuál de ellas potencia mejor la resolución de problemas. La investigación parte del supuesto o hipótesis que la eficiencia en la solución de problemas lógico algorítmicos está relacionada con la técnica de diagramación aprendida.

En consecuencia, esta investigación de corte correlacional, examina la capacidad de análisis mediante pruebas en forma de aplicación móvil basada en el modelo de Román (2015), evaluando la eficiencia en la solución de problemas lógico algorítmico.

La investigación busca encontrar la relación entre la técnica de diagramación y la eficiencia en la solución de problemas algorítmicos.

1. Presentación de la investigación

1.1 Planteamiento del problema

En el año 2010 el IED Unión Europea ubicado en Bogotá, Colombia realiza una propuesta bastante ambiciosa al cambiar su enfoque hacia las TIC, ello lo respalda la impronta del PEI que ha regido desde entonces “*Formadores de Líderes en Tecnologías de la Información y la Comunicación*”. En los siete años que ha regido esta propuesta se observa que la asignatura Informática sigue manejando unas doctrinas orientadas en la enseñanza de tecnicismos como lo son: los componentes de hardware, sistema operativo y, durante varios años escolares, sobre las aplicaciones de oficina (ofimáticas). Se ve como este tipo de enfoque no ayuda a propender el PEI del colegio y mucho menos propicia el desarrollo del pensamiento computacional, ni el pensamiento algorítmico. Una de las propuestas que la National Science Foundation (NFS), International Society for Technology in Education (ISTE) y el Computer Science Teacher Association (CSTA) han realizado para que los jóvenes sean líderes en su formación de TIC para ser competitivo en las habilidades para la sociedad del conocimiento desde los diferentes niveles de educación. Esto conduce a pensar sobre la importancia de una asignatura como Informática en el desarrollo del país.

Un primer acercamiento para guiar adecuadamente el rumbo que se pretendió dar a la institución y seguir con las indicaciones internacionales es a través del pensamiento algorítmico. Para fortalecer y entender este campo del conocimiento se han propuesto diferentes técnicas de diagramación. Las cuales buscan que las personas puedan estructurar su pensamiento por medio de gráficos que describen el proceso para resolver un problema. La técnica más famosa son los

diagramas de flujo que se propuso en el año 1921 por Frank Gilbreth. Esta técnica es la más usada en las universidades en los niveles para la enseñanza e introducción a la programación. Aunque existen otras técnicas como es el caso de los Diagramas de Chapin o Diagramas Mudos, que en Alemania son ampliamente utilizados, conocidos como Struktogramme y estandarizados por la norma DIN 66261 (noviembre de 1985). En Colombia, los organigrama NASSI-Shneiderman son poco conocidos y usados, y por la desinformación que existe o por la falta de evaluación de dicha técnica se sigue enseñando los diagramas de flujo como técnica de diagramación para representar los procesos de solución de problemas.

A los procesos de pensamiento implicados en la formulación de problemas y representación de sus soluciones, se les conoce como pensamiento computacional, siempre y cuando dichas soluciones pueden ser efectivamente ejecutadas por un agente de procesamiento de información (humano, computadora o combinaciones de humanos y computadoras). Wing (2006)

Para llegar a un pensamiento computacional hay que abordar cuatro habilidades (Pensamiento algorítmico, descomposición, abstracción y reconocimiento de patrones), así como un conjunto de actitudes esenciales (Resolución de problemas, experimentación, facilidad de comunicación, curiosidad, etc.). Para este proyecto nos enfocaremos en la habilidad del pensamiento algorítmico, contemplando el desarrollo de un ambiente computacional en el tema de Estructuras de Decisión cuyo objeto es medir la eficiencia de dos técnicas de diagramación en la solución de problemas potenciando el pensamiento algorítmico.

En consecuencia, el trabajo de investigación busca responder a la pregunta:

¿Existe diferencias significativas en el desarrollo del pensamiento algorítmico entre un grupo que interactúa con un ambiente con técnicas de diagramación Diagramas de Nassi-Schneiderman frente a otro grupo que trabaja con técnicas de diagramación Diagramas de Flujo?

1.2 Justificación

El pensamiento computacional es una competencia compleja relacionada con un modelo de conceptualización específica de los seres humanos que desarrolla ideas y vinculada con el pensamiento abstracto-matemático y con el pragmático-ingenieril que se aplica en múltiples aspectos de nuestra vida diaria. (Dapozo, 2017). El pensamiento computacional no es sinónimo de capacidad para programar un ordenador, se puede desarrollar pensamiento computacional sin utilizar ordenadores (basta papel y lápiz), es una competencia básica que todo ciudadano debería conocer para desenvolverse en la sociedad digital, pero no es una habilidad rutinaria o mecánica, ya que es una forma de resolver problemas de manera inteligente e imaginativa (cualidades humanas que no poseen los ordenadores). Los conceptos computacionales se utilizan para enfocar y resolver problemas reales, comunicarnos con otras personas y gestionar múltiples aspectos de nuestra vida cotidiana (Wing, 2006).

Para llegar a desarrollar un pensamiento computacional hay que abordar cuatro habilidades (Pensamiento algorítmico, descomposición, abstracción y reconocimiento de patrones), así como un conjunto de actitudes esenciales (Resolución de problemas, experimentación, facilidad de comunicación, curiosidad, etc.). En este trabajo abordamos las temáticas relacionadas con la habilidad del Pensamiento algorítmico evaluando la actitud de resolución de problemas.

Esta tesis se propone evaluar dos técnicas de diagramación (Diagramas de Flujo y Diagramas de Nassi Shneiderman) aplicando su estructura en la resolución de problemas que involucran el pensamiento algorítmico para su solución. A tal fin se parte de la pregunta ¿Existe diferencias significativas en el desarrollo del pensamiento algorítmico entre un grupo que

interactúa con un ambiente con técnicas de diagramación Nassi-Schneiderman frente a otro grupo que lo hace con técnicas de diagramación de Flujo? Con el propósito de responder ésta pregunta revisamos, en primer lugar, el concepto de resolución problemas, para ello se hace necesario distinguir por lo que se entiende comúnmente por problema y por su resolución.

Según (Perales, 1998), el problema podría ser definido genéricamente como cualquier situación prevista o espontánea que produce, por un lado, un cierto grado de incertidumbre y por el otro, una conducta tendente a la búsqueda de su solución. En la vida ordinaria se resuelve un problema para obtener un resultado; por el contrario, en el contexto escolar el resultado importa poco (a menudo es conocido) y sí lo hace la propia resolución (Dumas-Carré 1987).

La resolución de problemas» se utilizaría para referirse al proceso mediante el cual la situación incierta es clarificada e implica, en mayor o menor medida, la aplicación de conocimientos y procedimientos por parte del solucionador (Gagné 1965, Ashmore et al. 1979) así como la reorganización de la información almacenada en la estructura cognitiva (Novak 1977), es decir, un aprendizaje. La palabra «resolución» sirve para designar la actividad que consiste en resolver el problema desde la lectura del enunciado, pudiendo establecerse una distinción entre el tratamiento lógico-matemático y la propia actividad de resolución, analizada a menudo en términos de encadenamiento de procesos, y la solución o respuesta, producto de dicha actividad (Dumas-Carré, 1987)

Según (Ferrer 2000) la resolución de problemas es una actividad básica del pensamiento, lo que permite al ser humano activar su capacidad mental, ejercitando su creatividad, reflexionar y mejorar sus procesos de pensamiento para afrontar situaciones problemáticas con una actitud crítica.

Para alcanzar el objetivo de esta tesis se determina el nivel de resolución de tres tipos de problemáticas mediante una aplicación móvil midiendo el tiempo de solución y la cantidad de instrucciones usadas para llegar a la meta. El trabajo de investigación permitirá concluir cuál de las dos técnicas de diagramación fortalecen la resolución de problemas que implican el pensamiento algorítmico.

Por último, “El impacto de las Tecnologías de la Información y Comunicación (TIC) sobre la educación, propicia posiblemente uno de los mayores cambios en el ámbito de la Educación” (Belloch, 2012), en este sentido, las herramientas tecnológicas se convierten en un insumo importante del trabajo de investigación, de manera concreta, en la aplicación de la prueba pretest, la cual se realiza por medio de una app (aplicación móvil) diseñada por el investigador, por medio de la cual se evalúan la resolución de problemas que cuenta el participante; de igual forma, dicha aplicación se proyecta como un insumo de importancia dentro del contexto pedagógico en escenarios académicos a la hora de diagnosticar el nivel de resolución de problemas de estudiantes de básica secundaria y media.

1.3 Objetivos

1.3.1 Objetivo General

Comparar la incidencia de las técnicas Diagramas Nassi-Shneiderman y Diagramas de flujo en el desarrollo del pensamiento algorítmico en la resolución de problemas.

1.3.2 Objetivos Específicos

1. Caracterizar los sistemas de representación como forma de solución de problemas.
2. Diseñar dos ambientes computacionales según las condiciones de la investigación.

3. Identificar a través de los registros almacenados en el ambiente de aprendizaje cuál de las dos técnicas presenta una mejor eficiencia en la solución de problemas.
4. Con base en los resultados de la investigación determinar la técnica adecuada para estimular el pensamiento algorítmico en los estudiantes de grado Sexto del Colegio Union Europea IED.

2 Estado del arte

La revisión bibliográfica de las investigaciones realizadas a la fecha tiene en cuenta estudios que abordan las temáticas de diagramas de flujo, diagramas de Chapin y su implicación en la solución de problemas, como aquellos en los que se aplica para potenciar el pensamiento algorítmico, mismos que se exponen a continuación y brindan un panorama general del objeto de estudio.

La investigación realizada por Nassi y Shneiderman (1973), tuvo como objetivo crear un método para modelar el cálculo en estructuras simplemente ordenadas, evitando las transferencias de control sin restricciones, teniendo una estructura de control más cercana a los lenguajes de programación. Este planteamiento surge a partir de los problemas que se generan en la transferencia analógica de diagramas de flujo a un lenguaje de programación con las estructuras de control. Los resultados obtenidos fueron que los estudiantes que primero aprenden a diseñar programas con estos símbolos nunca desarrollan los malos hábitos que otros sistemas de diagramación permiten, adicional le da orden al programa. Se elimina la tentación de usar conectores fuera de página que sólo conducen a confusión. Por último, la facilidad con que un diagrama estructurado se puede traducir en un programa estructurado. Con el fin de hacer más eficiente la transición del diagrama al programa de computadora, se necesita expresar

construcciones de lenguaje más potentes en este modelo. Aunque no han facilitado el trabajo del programador y, de hecho, se puede requerir mayor previsión, creen que los beneficios en la depuración, la auto documentación y el mantenimiento superan en gran medida el costo adicional.

López (2014) investigó sobre cuáles son las características de las actividades de aula bajo el modelo de resolución de problemas, del entorno de programación Scratch y de las interacciones en el aula, que favorecen el uso de conceptos del pensamiento algorítmico por parte de los estudiantes de grado 3° de educación básica primaria del INSA. Para ello, trabajó con dos grupos de 33 estudiantes implementado la metodología ABP (Aprendizaje basado en Proyectos), e implementó cinco instrumentos para la recolección de la información, como resultados encontró que los desempeños de los estudiantes mejoraron el rendimiento y la eficacia en la resolución de tareas aumentando la categoría superiores y disminuyendo el nivel básico. También concluyó que si los estudiantes usan y se apropian de los conceptos del pensamiento algorítmico en un entorno educativo que tiene como eje articulador la metodología de solución de problemas propuesta por Polya pueden movilizarse fácilmente entre los ejes del pensamiento algorítmico.

En el campo del desarrollo del pensamiento lógico y algorítmico, Vidal, Cabezas, Parra y López (2015), desarrollaron una investigación donde realizaron experimentos de enseñanza y aprendizaje del uso de un lenguaje de programación orientado a objetos, mediante la herramienta Scratch. Esta investigación se aplicó a 27 estudiantes de los grados 6° y 7° de dos colegios de Linares y de Viña del Mar, utilizando la metodología de Resolución de problemas; concluyendo que Scratch motiva a la participación de los estudiantes en la resolución de las problemáticas

planteadas, es una herramienta adecuada y propicia para el acercamiento a la programación, posibilitando el análisis del problema y la propuesta, la aplicación de soluciones lógicas y algorítmicas, desarrollarlas y probarlas para poder ser mejoradas, además que permite utilizar recursos de sonido, gráficas y movimiento. Los estudiantes pueden crear diferentes proyectos utilizando recursos multimedia, desarrollando el pensamiento creativo, estructurado, lógico y obtener verdaderos aprendizajes significativos.

Villalobos (2009) expuso su experiencia de investigación "Cupi2: Una solución integral al problema de Aprender/enseñar a Programar", dirigido a estudiantes de la Universidad de los Andes de pregrado de ingeniería de sistemas y programas afines que toman en su pensum académico la clase de programación, en este proyecto el objetivo de esta investigación es encontrar las razones por las que los estudiantes no comprenden la programación con la metodología implementada y cómo estimular en los estudiantes la comprensión y aplicación del conocimiento de la programación. El proyecto se aborda desde las metodologías de aprendizaje activo, aprendizaje basado en problemas y proyectos (PBL), aprendizaje incremental y aprendizaje basado en ejemplos. Para lograr este objetivo se ha dividido en metas específicas como: Diseñar una herramienta de estudio e identificar el proceso de aprendizaje que no debería aplicarse para enseñar la materia. Donde se concluyó que el problema de enseñar a programar es complejo y cualquier solución que se proponga debe contemplar distintos componentes, logrando construir soluciones exitosas para enfrentar el problema de enseñar a programar; permitiendo que los cursos de programación soportan mejor al resto de cursos del currículo, dando al estudiante una visión global de la problemática de construcción de software y obteniendo resultados que antes se encontraban restringidos a cursos más avanzados.

Alfonso (2010) investigó sobre la presencia de transferencia análoga de estrategias de solución de problemas relacionados con la temática de estructuras selectivas dobles planteadas en la etapa de entrenamiento a la solución de problemas de programación. Este proyecto surge porque los estudiantes de los primeros semestres de ingeniería presentan dificultades para comprender y manejar acertadamente estructuras básicas de programación.

Para el desarrollo de la investigación se compararon 3 grupos de estudiantes de la Universidad Central de Colombia, cada grupo representa la solución de problemas en tres formas diferentes: Diagramas de flujo, Pseudocódigo, lenguaje de programación java.

Los grupos experimentales utilizaron un ambiente computacional, el cual proporciona los enunciados de los problemas a resolver, dependiendo del grupo el ambiente computacional automáticamente habilita las herramientas para diseñar la representación: grupo 1 representa primero en diagrama de flujo y posteriormente transfiere esta solución a la sintaxis de un lenguaje, el siguiente grupo primero lo representa en pseudocódigo y luego lo transfiere a la sintaxis y el tercero lo representa en diagramas de flujo lo transfiere a pseudocódigo y por último a la sintaxis. Alfonso encuentra que en el grupo 3 existe un porcentaje alto de transferencia análoga al evaluar los resultados desde la eficiencia (número de líneas utilizadas en el desarrollo del problema).

Ortegate (2010) compara los diagramas de flujo con los diagramas de Chapin al representar problemas con arreglos, para desarrollar esta propuesta construye un tutor inteligente donde se enseña que es un arreglo y como se usan. La investigación lleva a la conclusión que no hay diferencia significativa en la enseñanza de las dos propuestas. Esto contradice los resultados de (Pérez & Monzalvo, 2010) donde llegan a la conclusión que si existen diferencias en la herramienta que se use para el diseño de algoritmos.

Pérez y Monzalvo (2010) investigaron sobre la problemática presente en el diseño de los algoritmos secuenciales, selectivos e iterativos, concebidos dentro del paradigma de la programación estructurada, por parte de los estudiantes de las asignaturas de programación estructurada en la UAEH (Universidad Autónoma del Estado de Hidalgo). Se diseñó, aplicó y se llevó a cabo el análisis de un instrumento de evaluación, que se centra en abordar problemas donde se desarrollan algoritmos con sentencias secuenciales, selectivas e iterativas, con el propósito de identificar en los estudiantes su desempeño en el diseño de estos tipos de algoritmos. La metodología para desarrollar esta investigación se divide en 4 fases (Selección de dos problemas tipo a plantear, planteamiento de los enunciados de los problemas, recolección de los resultados obtenidos y análisis y generación de las conclusiones).

Los dos enunciados de los problemas seleccionados, se aplicaron a 21 estudiantes de la materia de programación estructurada, de primer semestre, de la carrera en Ingeniería en Electrónica y Telecomunicaciones, impartida en la Universidad Autónoma del Estado de Hidalgo. De este estudio se obtuvieron los siguientes resultados, en el caso del diseño del algoritmo secuencial y selectivo, los estudiantes tuvieron un éxito promedio de 69.8%, se puede observar que en el desarrollo de los pasos que comprende la fase de diseño de algoritmos, queda evidente que conforme se avanza desde el Diseño Descendente hasta la Representación de Algoritmos, el desempeño de los estudiantes va disminuyendo significativamente.

En la subetapa del Diseño Descendente, en donde se definen los subproblemas de manera verbal, la determinación de sus entradas y salidas, se logra alcanzar resultados de más del 90%, lo que hace pensar que los estudiantes tienen éxito en expresar de manera verbal, el diseño del algoritmo a construir. En la subetapa del Refinamiento de los Subproblemas, en donde se trata de formalizar los planteamientos verbales de la subetapa anterior, los estudiantes muestran dos

aspectos relevantes: aunque desciende su éxito en lograr el objetivo de esta parte, en poco más de 23% en promedio, el otro hecho es que pueden mejorar en poco más del 16%, el diseño del proceso que realizará el algoritmo, con respecto a la determinación de las entradas y salidas de este. Finalmente, en la subetapa de Representación del Algoritmo, sucede algo similar a la fase de diseño anterior, ya que disminuye el desempeño de los estudiantes en aproximadamente 28% en promedio, pero a pesar de esto, mejora el diseño del proceso que realizará el algoritmo, es poco más del 16%, con respecto a la determinación de las entradas y salidas. Además en esta subetapa, el uso de las diversas herramientas de diagramación, presenta una utilización de poco más del 61% en promedio, por parte de los encuestados.

En el caso del diseño del algoritmo cíclico, que utiliza los tres tipos de sentencias comprendidas dentro de la programación estructurada (secuenciales, selectivas e iterativas), el desempeño de los estudiantes es significativamente menor que en el algoritmo secuencial y selectivo. El promedio de éxito en el desarrollo de los pasos para el diseño del algoritmo fue apenas del 21.7% en promedio, poco más de tres veces menos que el desempeño que tuvieron los encuestados, en el desarrollo del algoritmo secuencial y selectivo.

Se puede observar que en la subetapa del Diseño Descendente, la determinación de las entradas y salidas, alcanza un éxito de poco más del 40% en promedio, menos de la mitad de lo que logran en el problema secuencial-selectivo (90%), lo que hace pensar que los estudiantes tienen desde un principio, dificultad en expresar de manera verbal el diseño del algoritmo a construir. En la subetapa del Refinamiento de los Subproblemas, en donde se trata de formalizar los planteamientos verbales de la subetapa anterior, los estudiantes muestran una tendencia a la baja bastante marcada, en promedio del 22%, en comparación al desempeño que se tuvo en el problema secuencial-selectivo (74%) en esta misma subfase. Finalmente, en la subetapa de

Representación del Algoritmo, el desempeño de los estudiantes en el programa cíclico, disminuye hasta el 3% en promedio, siendo que en el problema secuencial-selectivo era del 46% en promedio, lo que indica un grave problema de cómo desarrollar el producto de esta subfase del diseño de algoritmos. Además, el uso de las diversas herramientas de diagramación en esta subetapa, es casi nula, ya que poco más del 90% de los encuestados no las utilizan, indicando que las técnicas y herramientas de diagramación no parecen aportar elementos importantes para el diseño de algoritmos cíclicos.

Pérez & Monzalvo (2012) hace una comparación global del desempeño de los estudiantes en el diseño de dos tipos de problemas, el secuencial-selectivo (P-SS) y cíclicos. Por una parte, el diseño relativo al problema secuencial-selectivo (P-SS) muestra una tendencia de alto a medio, lo cual deja entrever que los estudiantes pueden concebir a buen nivel, los algoritmos a construir de manera inicial, pero conforme se va avanzando en su especificación, esta es más difícil de que la logren. En cuanto al diseño de los problemas cíclicos, el problema es mucho más crítico, ya que el desempeño de los estudiantes se caracteriza por su bajo nivel de éxito, conforme se avanza en tratar de concebir y especificar las diversas subetapas del diseño de algoritmos.

Los resultados obtenidos, muestran que los algoritmos secuenciales-selectivos son relativamente fáciles de concebir por los estudiantes. Esto posiblemente se puede deber a que las personas piensan de manera muy similar a como estas sentencias se estructuran; no así en el caso de los problemas cíclicos, en donde los resultados son poco alentadores. El presente estudio hace evidente que en el uso de sentencias repetitivas es en donde los estudiantes tienen más problemas al tratar de implementarlas en el diseño de algoritmos. Esto es debido probablemente a que las personas no tienden a pensar en forma iterativa.

Por otra parte, las técnicas y herramientas de diagramación clásicas para el diseño de algoritmos (diagramas de flujo, tablas N-S, pseudocódigo), parecen adecuarse adecuadamente a la forma de concebir los programas secuenciales y selectivos, en un poco más del 60% de los casos, pero parecen no aportar beneficios sustanciales en el diseño de programas repetitivos, ya que más del 90% de los estudiantes no las utilizaron para tal fin. Debido a todo lo anterior, es necesario realizar más estudios sobre la problemática presente en el diseño de algoritmos cíclicos en investigaciones posteriores, ya que al parecer, los programas secuenciales y selectivos presentan un aspecto diferente en su concepción por las personas, en comparación con la forma de abordar programas iterativos o repetitivos.

Pérez & Monzalvo (2011) en su investigación “Nivel de utilización de las técnicas de diagramación por parte de los estudiantes, en el diseño de algoritmos secuenciales, selectivos e iterativos” exploró la problemática presente en el diseño de los algoritmos concebidos dentro del paradigma de la programación estructurada. Para ello, realizó un análisis del nivel de utilización de las técnicas de diagramación, en el diseño de algoritmos con sentencias secuenciales, selectivas e iterativas, con el propósito de observar que tan importantes son estas técnicas en el diseño de algoritmos para los estudiantes de las asignaturas de programación estructurada. La metodología en esta investigación tuvo 4 etapas (Selección de dos problemas tipo a plantear, planteamiento de los enunciados de los problemas, recolección de los resultados obtenidos y Análisis y generación de las conclusiones, derivadas de los resultados obtenidos).

Los dos enunciados de los problemas seleccionados, se aplicaron a 23 estudiantes de la materia de programación estructurada, de primer semestre, de la carrera en Ingeniería en

Electrónica y Telecomunicaciones, impartida en la Universidad Autónoma del Estado de Hidalgo. Se encuestaron a los estudiantes participantes en el estudio, abordando el tema del diseño de algoritmos de tipos secuenciales-selectivos e iterativos, con respecto a la utilización de las diversas técnicas de diagramación disponibles. Como lo menciona el estudio de Pérez y Monzalvo (2010), una cuarta parte de los estudiantes (26.1%), se les clasificaba en un nivel de desempeño alto, en cuanto al aspecto de diseñar algoritmos secuenciales-selectivos e iterativos. Poco más de la mitad de los estudiantes (52.2%), se clasificarían en un nivel de desempeño medio, y finalmente, poco menos de la cuarta parte (21.7%), serían considerados en un nivel de desempeño bajo.

A este respecto, el conocimiento y aplicación adecuada de las técnicas de representación de algoritmos, no parece ayudar sustancialmente a resolver el problema que presentan los estudiantes, sobre su deficiente capacidad para desarrollar algoritmos que reflejen correctamente las soluciones solicitadas. Hay que señalar que las técnicas de representación de algoritmos son sólo parte de la compleja estructura metodológica pendiente por definir y ajustar en futuras investigaciones (Pérez, Fuentes y Moreno, 2008).

(Prado & Lamas, 2014) en su investigación “Alternativas para la enseñanza de pseudocódigo y diagrama de flujo”, busca mejorar el proceso de enseñanza aprendizaje de la materia Fundamentos de Informática de la carrera Tecnicatura Industrial de la Facultad de Tecnología y Ciencias Aplicadas, tratan de encontrar nuevas formas en las que los alumnos pueden desarrollar sus habilidades de solución de problemas a través del desarrollo de pseudocódigo y diagramas de flujo. Para ello, se analizan las herramientas de enseñanza aprendizaje Pseint y FreeDfd, basadas en software libre, para la enseñanza de pseudocódigo y diagramas de flujo, determinando cuál de ellas cumple con requisitos de usabilidad y resulta más

amigable a los alumnos para la resolución de problemas, comprobación de errores y explicación paso a paso de cada instrucción ejecutada.

Para evaluar las dos herramientas propuestas en esta investigación se abordan 10 criterios:

1. Que la finalidad de la herramienta sea un recurso didáctico en la enseñanza-aprendizaje de algoritmos.
2. Representar mediante diagramas de flujo las soluciones algorítmicas.
3. Que cumpla con los requisitos de usabilidad (es decir que los alumnos puedan hacer sus tareas de forma rápida y fácil)
4. Que la herramienta sea amigable a los alumnos para la resolución de problemas
5. Que se realicen comprobación de errores.
6. Que se explique paso a paso cada instrucción ejecutada.
7. Ejecución multiplataforma (Linux, Windows y MacOS).
8. Facilidad de instalación en distintas plataformas.
9. Página Web dedicada específicamente al software para proveer de nuevas versiones, documentación y ejemplos.
10. Blog de ayuda para el usuario.

3 Marco Teórico

En esta sección se presentan los referentes teóricos en los que se sustenta el trabajo de investigación, realizando un desarrollo conceptual sobre la importancia y relevancia que tiene el pensamiento algorítmico en la resolución de problemas, y como ha motivado a ministerios de educación a proponer adecuaciones en los currículos donde se de una apropiación del uso de las tecnologías de información, como es el caso en Inglaterra, EEUU, España, etc.

Entidades como la ISTE, CSTA y la National Research Council, en sus publicaciones enuncian la importancia del aprendizaje del pensamiento algorítmico para potenciar el pensamiento computacional, y avanzar en la fluidez con las tecnologías de la información. A esto lo llaman FITness (Fluency with information technology) (National Research Council, 1999; p 2).

Ello conduce a reformular el conocimiento, la manera de expresarse, de producir y generar información en lugar de simplemente comprenderla, ya que la tecnología de la información está más presente en nuestro diario vivir desempeñando un papel cada vez más importante en la vida, tanto a nivel laboral como personal.

Muchas de las personas que utilizan la tecnología de la información sólo tiene una comprensión limitada de las herramientas que utilizan, no se sienten seguros y les gustaría estar más seguros de sí mismos. Hay una simple curiosidad sobre cómo funciona esta potente y penetrante tecnología. Una de las soluciones que se han llevado a cabo es la alfabetización informática, pero la tecnología avanza a pasos agigantados y las habilidades aprendidas se vuelven obsoletas y el cambio a nuevas habilidades se vuelve difícil. Una mejor solución plantea

la adaptación a los cambios en la tecnología sin atarse a un aplicativo o software. Allí las personas son capaces de expresarse creativamente, reformular el conocimiento y sintetizar nueva información.

Este proceso genera la capacidad de aplicar la tecnología de la información en situaciones complejas, encapsulan el pensamiento de nivel superior en el contexto de la tecnología de la información, la gente aprende a manipular el medio a su ventaja para manejar los problemas no deseados e inesperados cuando surgen. Las capacidades intelectuales fomentan un pensamiento más abstracto sobre la información y su manipulación.

Según la National Research Council (1999) aquellos que conocen un conjunto básico de habilidades de tecnología de la información, que entienden los conceptos básicos sobre los cuales se basa la tecnología de la información y que se han involucrado en el pensamiento de nivel superior incorporado en las capacidades intelectuales, deben llegar a trabajar listos para aprender rápidamente nuevos sistemas de negocio y utilizarlos con eficacia, deben ser capaces de aplicar la tecnología de la información a los problemas personalmente relevantes y deben ser capaces de adaptarse a los cambios inevitables como la tecnología de la información evoluciona a lo largo de su vida. Les brindará conocimientos esenciales para el uso de la tecnología de la información ahora y en el futuro.

Una parte para lograr esta meta es fortalecer la enseñanza de la programación de computadores y para ello, se han propuesto diferentes entornos que potencian el aprendizaje de la programación, motivando y atrayendo a los jóvenes a interesarse por este campo desde intereses como: crear sus propias aplicaciones, desarrollar juegos o dar soluciones a problemas que

perciban en su entorno. Algunas de estas aplicaciones con fines educativos son: Appinventor, Scratch (<http://scratch.mit.edu/>), Alice (<http://www.alice.org/>), etc., aquí los estudiantes pueden programar de una manera intuitiva, gracias a la interfaz gráfica que tiene mucha similitud a un rompecabezas, ya que los programadores van enlazando bloques y van interactuando con la funcionalidad de su creación. Otro escenario para que se lleve a cabo este proceso de aprendizaje lo impulsa el sitio web code.org, un movimiento que pretende dar la oportunidad a cada estudiante que cuente con acceso a un computador con internet. La metodología de aprendizaje se basa a partir de juegos y una serie de retos que inducen al aprendiz a solucionar una serie de problemáticas mientras se van familiarizando con las sentencias más famosas en la programación. Code.org tienen como meta introducir o vincular en los currículos escolares la enseñanza de la programación desde diversas áreas del conocimiento.

Para impulsar este desarrollo cognitivo existen una serie de aprendizajes intermedios, pensamiento algorítmico de diagramación, para que los estudiantes no se queden solo con la habilidad de enlazar bloques, sino que aprendan a estructurar su pensamiento y puedan dar alcance a las metas logradas (crear un videojuego, crear la aplicación, etc.) independientemente del ambiente en el cual estén trabajando.

Yoder y Schrag (1978) desarrollan un artículo comparativo entre las técnicas de diagramación convencionales (Diagramas de flujo), pseudocódigo, HIPO (En inglés, Hierarchy-Input-Process-Output) y diagramas de Nassi-Shneiderman. Exponen que un diagrama estructurado busca cumplir los objetivos de la programación: corrección, eficiencia, creatividad, la facilidad de mantenimiento (la facilidad de fijar los errores), la modificabilidad (la facilidad de hacer cambios), la legibilidad (la claridad del programa) y la incomprensibilidad.

Independientemente de la técnica de diagramación que se use, esta debe tener unos mínimos para realizar la codificación estructurada, como son: Un punto de partida y un punto de salida, tener unas reglas para la segmentación y anidación de estructuras. Además deben contar con un conjunto mínimo de estructuras de programa (secuencias, estructuras de decisión (Si y No), Bucles (Do While, Do Until, Repeat) y estructuras de selección (Case)). Cada estructura debe contar con su convención para su posterior codificación. Los autores llegan a la conclusión que ningún método es mejor que otro, depende el uso que se les dé. Los diagramas HIPO son usados en las especificaciones funcionales, mientras que los diagramas de flujo son usados para la comprensión de la codificación, los de Nassi-Shneiderman permiten el paso a los pseudocódigos y posteriormente a la codificación y tienen capacidad de estructuración incorporada en la técnica. Como semejanza los Diagramas de flujo y los de Nassi-Shneiderman pueden ser fácilmente traducidos directamente en código. El pseudocódigo tiene la ventaja de representar gráficamente la lógica y también identificar claramente y visualmente procesos dentro de condicionales compuestos.

Los usuarios pueden combinar las técnicas para obtener mejores resultados, como por ejemplo: gráficos de HIPO y N-S son muy útiles en el flujo lógico. Los diagramas de flujo y pseudocódigo están demasiado orientados a la programación para ser utilizados por los programadores.

La documentación del programa tradicionalmente ha estado separada de los programas. Uno de los beneficios esperados del código estructurado eran los programas de auto-documentación. Hasta cierto punto, este beneficio se ha realizado; Sin embargo, en muchos casos se requiere documentación adicional. El pseudocódigo proporciona una excelente manera

de incluir esta explicación suplementaria de código dentro del programa como comentarios. Sin embargo, si se requiere documentación externa del programa, una representación gráfica del código (algo imposible de codificar en el programa) puede ser significativamente mejor. El éxito de los diagramas HIPO ha demostrado este hecho para la visión general y la documentación de la función. Para mostrar la lógica, los gráficos N-S son mucho mejores que los gráficos HIPO y mucho mejores que los diagramas de flujo.

Las gráficas de Nassi-Shneiderman han demostrado ser útiles en casi todas las fases del desarrollo del programa, desde el diseño temprano hasta el paso a través de la codificación, las pruebas y la educación del usuario. Es una excelente técnica gráfica, los diagramas N-S proporcionan un lenguaje sencillo pero elegante que, intencionalmente, es compatible con los objetivos y métodos de programación estructurada. Como escribieron Nassi y Shneiderman,

"Los programadores que primero aprenden a diseñar programas con estos símbolos nunca desarrollan los malos hábitos que otros sistemas de notación de diagramas de flujo permiten. Ya que no se pueden dibujar más de quince o veinte símbolos en una sola hoja de papel, el programador debe modularizar su programa en significativas Secciones. Se elimina la tentación de usar conectores fuera de página, que sólo conducen a confusión. Por último, la facilidad con que se puede traducir un diagrama de flujo puede traducirse en un programa estructurado agradablemente sorprendente" [1]

Para el año de 1978 los autores pensaban: "Debido a que las cartas de Nassi-Shneiderman sólo se están conociendo ahora, el método no ha sido totalmente explotado. Hay potencial en muchas áreas para expandir el uso de estos gráficos estructurados y si el pensamiento lógico que

ahora estamos insistiendo en la programación se puede extender a otras disciplinas, no podemos perder.”

3.1 Pensamiento Algorítmico

El pensamiento algorítmico es una de las etapas para llegar a un pensamiento computacional, como lo confirma la ISTE y CSTA. Para abordar el tema de pensamiento algorítmico surge el interrogante qué es un algoritmo, Futschek (2006) lo define como: “método para resolver un problema que consiste en instrucciones exactamente definidas”, López (2009) como: “una herramienta que permite describir claramente un conjunto finito de instrucciones, ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible”. La RAE lo define como: “un grupo finito de operaciones organizadas de manera lógica y ordenada que permite solucionar un determinado problema”.

Se trata de una serie de instrucciones o reglas establecidas que, por medio de una sucesión de pasos, permiten arribar a un resultado o solución. Inicialmente, los algoritmos se suelen representar en forma narrativa, pero cuando tienen aplicación más directa, como es en el caso de la programación de computadoras, éstos se convierten en diagramas o gráficos de programación, que son la representación gráfica de la solución del problema que se desea mecanizar (Joyanes Aguilar, 1988).

El pensamiento algorítmico ayuda a los estudiantes a pasar de un problema a un programa o itinerario de su solución, esto es, a una solución como una secuencia finita y determinística de pasos (Joyanes Aguilar, 2003). Esto implica la capacidad de definir y enunciar con claridad un problema; descomponerlo en subproblemas más pequeños y manejables y, describir una solución

a lograr en un conjunto de pasos bien definido. La capacidad para analizar y dar solución a problemas, es una habilidad importante para todo ser humano, y en especial para los estudiantes (Polya, 1945).

Grover (2009) menciona que los estudiantes deben exponerse a algunos objetivos generales del aprendizaje del siglo XXI para tener un mayor conocimiento de la Ciencia de la Computación para tener una mayor comprensión. Dentro de los objetivos está el pensamiento algorítmico, el cual tiene como finalidad comprender, cómo y por qué funcionan, de la manera en que lo hacen, los sistemas de las tecnologías de la información.

Según Grover (2009): “El pensamiento algorítmico ayuda a los estudiantes a pasar del problema al programa. Esto implica estar en capacidad de definir y enunciar con claridad un problema; descomponerlo en problemas más pequeños y manejables; y, describir una solución en un conjunto de pasos bien definido. Esta es una habilidad importante que los estudiantes pueden transferir a otras asignaturas cuando enfrenten en estas situaciones en las que deban solucionar problemas”. Para la National Research Council (1999) las personas que poseen un conocimiento en el pensamiento algorítmico, avanzará en el camino hacia fluidez en el uso de las Tecnologías de la Información y la Comunicación (TIC) porque es uno de los conceptos claves en el FITness. (Fluidez en el uso de las tecnologías de la información y las comunicaciones).

Tal como lo menciona Wing (2006), “el pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática”, y utiliza algunos elementos para hacer frente a una variedad de problemas, como son: la descomposición, el reconocimiento de patrones, la

abstracción, el pensamiento algorítmico, la depuración y la presentación de una solución en una forma utilizable.

La National Research Council (1999) establece que en el pensamiento algorítmico se deben dar los conceptos: “descomposición funcional, repetición (iteración y/o recursión), organización de datos básicos (registro, matriz, lista), generalización y parametrización, algoritmo vs. Programa, diseño de arriba hacia abajo y refinamiento”.

Según Artopoulos (2013), “El pensamiento algorítmico es una habilidad genérica que se aplica para resolver problemas complejos de procesamiento de información mediante el método formal de explicitar los pasos en forma ordenada. Es la aplicación de métodos algorítmicos y modelos computacionales para aprovechar la inteligencia de las máquinas electrónicas y resolver problemas que los humanos no podemos resolver solos.” Para lograr resolver los problemas usando el pensamiento algorítmico Futschek (2006) plantea que este pensamiento debe incluir las capacidades de: a) analizar problemas dados; b) especificar un problema de manera precisa; c) encontrar las acciones básicas que son adecuadas para resolver el problema dado; d) construir un algoritmo correcto para resolver un problema determinado, utilizando las acciones básicas; e) pensar en todos los posibles casos tanto especiales como normales de un problema; y, f) mejorar la eficiencia de un algoritmo. “El pensamiento algorítmico posee un elemento creativo fuerte: la construcción de nuevos algoritmos que resuelvan problemas dados. Si alguien quiere hacer esto, necesita pensar algorítmicamente” (Futschek, 2006). Para Katai (2015) y Lamagna (2015) el término pensamiento algorítmico es la habilidad de ejecutar, evaluar, entender y crear procedimientos computacionales, para poder desarrollar la competencia del pensamiento

algorítmico se debe de entender y ejecutar un procedimiento paso por paso y poder crear nuevos algoritmos.

López (2014) integra las propuestas relacionadas con el pensamiento algorítmico hechas por Futschek (2006), ISTE (2011) e INSA (2005), determinando que se podrían enmarcar claramente en la dimensión “formular problemas” y “solucionar problemas”. Para tal efecto se puede utilizar la plantilla propuesta por López (2014). Esta plantilla tiene como objetivo contribuir al desarrollo de la habilidad para solucionar problemas, a partir del pensamiento algorítmico. En esta matriz se pide al estudiante que a partir de un enunciado de un problema (en diferentes dominios), identifica cinco elementos: formulación del problema, datos disponibles, restricciones, resultados esperados y procesos necesarios. Este análisis implica una actividad cognitiva que involucra planificación, formulación de hipótesis, abstracción, comprensión lingüística; es decir, las mismas habilidades identificadas en los estudios sobre pensamiento computacional (Wing, 2006; ISTE, 2011).

3.2 Razonamiento Diagramático

Según Legris (2012) un diagrama es una representación de relaciones entre entidades. Estas relaciones implican necesariamente el concepto de espacio y tienen un carácter topológico. Para mayor exactitud se encuentra la siguiente caracterización:

“Un diagrama es un conjunto de objetos en el plano que denotan objetos en una situación [una estructura], cuyas mutuas relaciones espaciales y gráficas denotan relaciones en aquella estructura.” (Como se cita en Legris, 2012).

Teniendo un concepto más claro de diagrama se puede afrontar el concepto de razonamiento diagramático como lo aborda Legris según Pierce, en la conclusión de su trabajo determina que:

“Por razonamiento diagramático entiendo el razonamiento que construye un diagrama de acuerdo con un precepto expresado en términos generales, realiza experimentos sobre este diagrama, toma nota de sus resultados, se asegura de que experimentos realizados sobre cualquier razonamiento que sea construido con el mismo precepto tengan los mismos resultados, y expresa esto en términos generales.” (como se cita en Legris, 2012).

Basándose en esta definición, se puede decir que el razonamiento diagramático es un método de deducción. Y este método es el que representa mejor la “trayectoria del razonamiento”, esto es, la estructura de la deducción. (Legris, 2012)

El Razonamiento Diagramático es actualmente un área de investigación integrada en la lógica simbólica de creciente interés interdisciplinario (Filosofía, Ciencias de la Computación, Ciencias cognitivas, etc.). Los diagramas como una forma de representación semiótica muestran que una cierta característica de un conjunto de objetos es considerada como un nuevo objeto; se asigna un nombre a un predicado concreto creándose de ese modo un objeto abstracto (Bakker & Hoffmann, 2005), El razonamiento diagramático es una manipulación, inventiva y constructiva, de diagramas basada en reglas para investigar sus propiedades y relaciones, Dörfler (2003, p. 41)

identifica algunas de las características de los diagramas y el razonamiento diagramático: basadas en esta estructura diagramática se realizan operaciones guiadas por reglas sobre y con las inscripciones mediante transformación, composición, descomposición y combinación de las mismas.

El razonamiento diagramático eficiente y exitoso presupone una experiencia intensa y extensa con la manipulación de diagramas. Un amplio “inventario” de diagramas, sus propiedades y relaciones apoya y ocasiona el uso creativo e inventivo de diagramas. El razonamiento diagramático implica tres pasos (Bakker y Hoffmann, 2005, p. 340): 1) Construir uno o varios diagramas mediante un sistema de representación. 2) Experimentar con los diagramas. 3) Observar los resultados de la experimentación y reflexionar sobre ellos. Cualquier experimentación con un diagrama se está ejecutando dentro de un sistema de representación y es una regla o actividad, situado dentro de una práctica.

3.3 Diagramación

También se le conoce como diagramar u organizadores gráficos, son técnicas utilizadas para representar visualmente y esquemáticamente bien sea la secuencia de instrucciones de un algoritmo o los pasos de un proceso, lo que permite obtener una información preliminar sobre la amplitud de las actividades. Un diagrama de programación (o también llamado técnica o herramienta de diagramación), es la representación gráfica de los procedimientos y de la secuencia u orden en que deben ejecutarse éstos; en resumen, es la representación gráfica de la solución de un problema o de un procedimiento (Joyanes Aguilar, 1990).

La diagramación tiene como finalidad representar la solución de un problema, esto conlleva a identificar todos los pasos de una solución de forma clara y ordenada. Con esta representación se puede verificar si se ha tenido en cuenta todas las posibilidades de solución del problema, si hay procesos repetidos o duplicados, para así establecer alternativas que hagan más eficiente la solución planteada por medio de la discusión y socialización. Estas técnicas permiten que otras personas pueden comprender la lógica propuesta y documentar el camino que se siguió para lograr el objetivo. En esta investigación se hará uso de dos técnicas de diagramación, diagramas de flujo y diagramas de Nassi-Schneiderman.

3.3.1 Diagramas de Flujo

Existen diferentes definiciones y usos que se le ha dado a los diagramas de flujo, una de estas definiciones está dada desde el campo administrativo, como lo enuncia Gómez (1997) el cual le dio el nombre de Flujogramas o Fluxograma, y los definió como un diagrama que expresa las distintas operaciones que componen un procedimiento o parte de este, además pueden representar secuencias de rutina simple y tienen la ventaja de indicar los responsables.

En el campo de la ingeniería un diagrama de flujo es una serie de símbolos, cada uno de los cuales representa una función en el programa que se encuentra conectada a la siguiente en dirección descendente por las líneas de flujo. Es un medio de presentación visual y gráfica del flujo de datos, a través de un algoritmo.


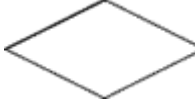

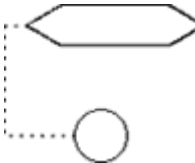
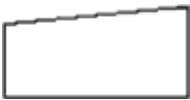





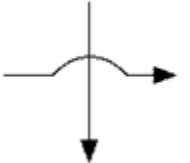

Los diagramas de flujo han sido la herramienta de programación por excelencia y, aún hoy, siguen siendo muy empleados. Son fáciles de diseñar porque el flujo lógico del algoritmo se muestra en un dibujo en lugar de con palabras. Sin embargo, los diagramas de flujo tienen una

limitación grande: son difíciles de actualizar y complejos de leer si ocupan varias páginas; cuando se han de modificar los diagramas de flujo resultan difíciles precisamente por su carácter gráfico (Joyanes Aguilar, 1988, 1990).

El diagrama de flujo ofrece una descripción visual de las actividades implicadas en un proceso mostrando la relación secuencial entre ellas, facilitando la rápida comprensión de cada actividad y su relación con las demás, el flujo de la información y los materiales, las ramas en el proceso, la existencia de bucles repetitivos, el número de pasos del proceso, las operaciones de interdepartamentales.

Con el fin de evitar la utilización de símbolos diferentes para representar procesos iguales, la Organización Internacional para la Estandarización (ISO, por su sigla en inglés) y el Instituto Nacional Americano de Estandarización (ANSI, por su sigla en inglés), estandarizaron los símbolos que mayor aceptación tenían en 1985. En la siguiente Tabla se muestran los principales símbolos para elaborar Diagramas de Flujo:

Tabla 1 Principales símbolos para elaborar Diagramas de Flujo

Símbolo	Descripción	Símbolo	Descripción
	Inicio/Final Se utiliza para indicar el inicio y el final de un diagrama; del Inicio sólo puede salir una línea de flujo y al Final sólo debe llegar una línea.		Decisión Indica la comparación de dos datos y dependiendo del resultado lógico (falso o verdadero) se toma la decisión de seguir un camino del diagrama u otro.
	Entrada General Entrada/Salida de datos en General (en esta guía, solo la usaremos para la Entrada).		Iteración Indica que una instrucción o grupo de instrucciones deben ejecutarse varias veces.
	Entrada por Teclado Instrucción de entrada de datos por Teclado. Indica que el computador debe esperar a que el usuario teclee un dato que se guardará en una variable o constante.		Salida Impresa Indica la presentación de uno o varios resultados en forma impresa.
	Llamada a subrutina Indica la llamada a una subrutina o procedimiento determinado.		Salida en Pantalla Instrucción de presentación de mensajes o resultados en pantalla.
	Acción/Proceso General Indica una acción o instrucción general que debe realizar el computador (cambios de valores de variables, asignaciones, operaciones Aritméticas, etc.).		Conector Indica el enlace de dos partes de un diagrama dentro de la misma página.
	Flujo Indica el seguimiento lógico del diagrama. También indica el sentido de ejecución de las operaciones.		Conector Indica el enlace de dos partes de un diagrama en páginas diferentes.

Como lo indica Talavera (1999) al usar diagramas de flujo para representar la solución de un problema o proceso, estos traen ciertos beneficios:

- Facilita la obtención de una visión transparente del proceso, mejorando su comprensión.
- El conjunto de actividades, relaciones e incidencias de un proceso no es fácilmente discernible a priori. La diagramación hace posible aprehender ese conjunto e ir más allá, centrándose en aspectos específicos del mismo, apreciando las interrelaciones que forman parte del proceso así como las que se dan con otros procesos y subprocesos. Permiten definir los límites de un proceso. A veces estos límites no son tan evidentes, no estando definidos los distintos actores (internos y externos) involucrados.
- El diagrama de flujo facilita la identificación de actores, es más sencillo determinar sus necesidades y ajustar el proceso hacia la satisfacción de sus necesidades y expectativas.
- Estimula el pensamiento analítico en el momento de estudiar un proceso, haciendo más factible generar alternativas útiles.
- Proporciona un método de comunicación más eficaz, al introducir un lenguaje común, si bien es cierto que para ello se hace preciso la capacitación de aquellas personas que entrarán en contacto con la diagramación.

- Un diagrama de flujo ayuda a establecer el valor agregado de cada una de las actividades que componen el proceso.
- Constituye una excelente referencia para establecer mecanismos de control y medición de los procesos, así como de los objetivos concretos para las distintas operaciones llevadas a cabo.
- Facilita el estudio y aplicación de acciones que redunden en la mejora de las variables tiempo y costes de actividad e incidir, por consiguiente, en la mejora de la eficacia y la eficiencia.
- Constituye el punto de comienzo indispensable para acciones de mejora o reingeniería.

3.3.2 Diagramas de Nassi-Shneiderman

La fascinante historia y la evolución de los organigramas estructurados (generalmente llamado Nassi-Shneiderman diagramas o structograms) se remonta a 1972. En una conferencia organizada por Michael Jackson de ACM en Nueva York, en una de las ponencias sobre programación estructurada surgió la pregunta si los GOTO debían evitarse, entonces no deberían las líneas en los viejos diagramas de flujo evitarse también? Quince minutos del dibujo condujeron a las primeras ideas de la secuencia, condicionales e iteración.

Allí fue donde surgió la idea de crear una nueva técnica de diagramación llamada diagrama Nassi-Shneiderman, la cual es una representación gráfica de un algoritmo para programación estructurada. Desarrollados en 1972 por Isaac Nassi y Ben Shneiderman, estos

diagramas también son conocidos como estructogramas debido a que muestran las estructuras de un programa.

Siguen un diseño de arriba a abajo, el problema en cuestión es reducido en subproblemas cada vez menores, hasta que sólo comandos y estructuras de control permanecen. Los diagramas Nassi-Shneiderman (N-S) reflejan esta descomposición de una forma clara y simple, usando cajas anidadas para representar subproblemas.

El diagrama estructurado N-S es una técnica híbrida (una mezcla) entre Diagramas de Flujo y Pseudocódigo. Esta técnica, también conocida como Diagrama de Chapín, utiliza una serie de cajas, similar a los diagramas de flujos, pero no requiere la utilización de flechas, debido a que su flujo siempre es descendente. Y han sido aplicados en diversos países como herramienta de diagramación.

En Alemania, son ampliamente utilizados y conocidos como Struktogramme, estandarizada por la norma DIN 66261 (noviembre de 1985), que aborda el tema de Tratamiento de la Información. Esto ha llevado a ver cómo se han extendido, como lo han ampliado y cuál es la influencia que ha tenido en la solución de problemas y el acercamiento a la programación. Un ejemplo claro es la plataforma Scratch, creada por el MIT, donde se evidencia que su estructura es muy similar a los diagramas de N-S.

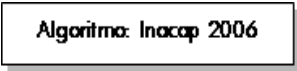




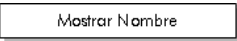
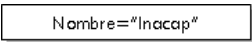
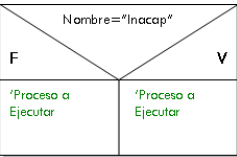
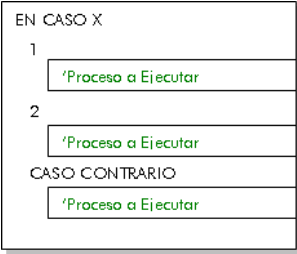
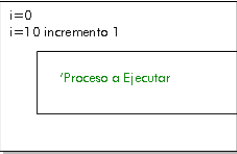
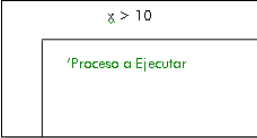
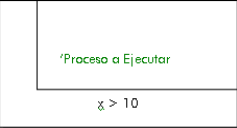
El atractivo de esta técnica es su sencillez, originalidad y facilidad de ampliación e incrustación. Para los usuarios se orientan visualmente y esto proporciona una visión general compacta de un programa que puede mostrar algunas relaciones muy bien. Por otro lado, puede consumir más papel que el código y ser más difícil para buscar algunos detalles.

En las primeras etapas de diseño es muy útil para esbozar el diseño modular de alto nivel. Para algunas personas es una ayuda en la orientación de pensar acerca de las estructuras condicionales anidadas, de una manera más visual que una tabla de decisión.

Existen una serie de similitudes entre los diagramas de flujo y los diagramas de Nassi-Schneiderman, una de ellas es que debe ir el nombre o título del algoritmo, en el recuadro siguiente debe ir la palabra Inicio. La última instrucción de un Diagrama NS debe ser el recuadro Fin. Todas las declaraciones de variables, funciones deben estar especificadas bajo el recuadro que contiene la palabra Inicio.

En la siguiente tabla se puede ver los símbolos más usados para este tipo de diagramación:

Tabla 2 Principales símbolos para elaborar Diagramas de Nassi-Shneiderman

Simbología utilizada en Técnica Nassi-Schneiderman			
Símbolo	Descripción	Símbolo	Descripción
	Título del Algoritmo		Indica el inicio de un algoritmo.
	Indica el Fin de un algoritmo.		Declaración de Variables.
	Indica la entrada de información		Indica la salida de información
	Indica un proceso, una ejecución de una línea de código, que puede ser una asignación.		Estructura de Decisión.
	Estructura de Decisión Múltiple		Estructura Cíclica de Ciclo Desde-Hasta
	Estructura Cíclica de Ciclo Hacer – Mientras		Estructura cíclica de Repetir - Hasta

[1] I. Nassi and B. Shneiderman, "Flowchart Techniques for Structured Programming," SIGPLAN Notices

3.4 Clasificación de los problemas

En muchos estudios se ha tratado la cuestión de la definición de problema y sus más diversas perspectivas. El contenido del término viene determinado, generalmente, a partir de la actividad que implica, más que en la forma que se propone. Haciendo un repaso por la bibliografía, Borasi (1986), reclama la clarificación del concepto, por no ser usado siempre de la misma manera en contextos diferentes. Parece ser, que en lo que sí existe un acuerdo generalizado es el de considerar un problema como una situación que presenta dificultades y para las cuales no hay soluciones evidentes. Así, Krulik y Rudnik (1980) definen el problema como: “una situación cuantitativa o no, que pide una solución, para la cual los individuos implicados, no conocen medios o caminos evidentes para obtenerla”.

Según (Perales, 1998), el problema podría ser definido genéricamente como cualquier situación prevista o espontánea que produce, por un lado, un cierto grado de incertidumbre y por el otro, una conducta tendente a la búsqueda de su solución.

Perales (1998) realiza una clasificación de los problemas estableciendo unos criterios y, a partir de ellos, agrupa los problemas como se muestra en la siguiente figura.

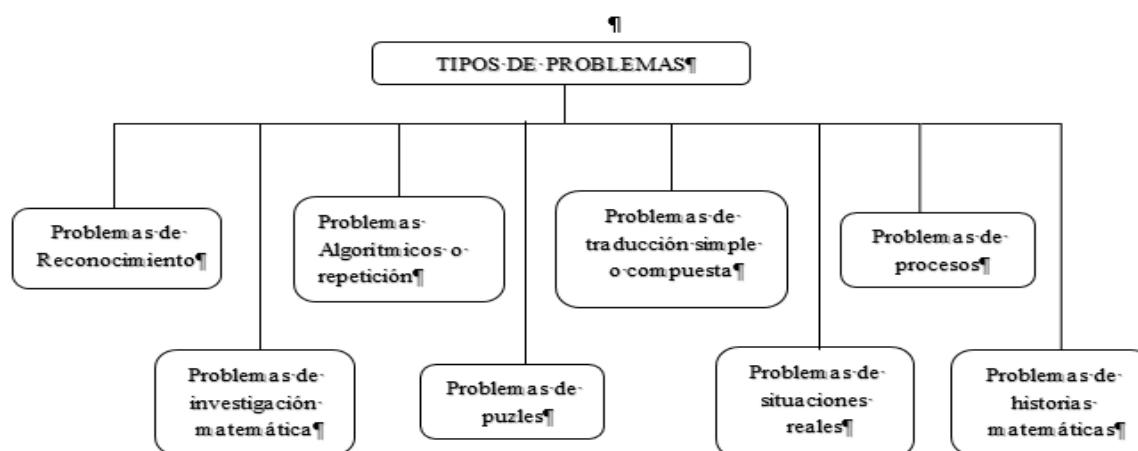


Figura 1 Clasificación de los problemas según Blanco

No existe un criterio único ni una sola clasificación de problemas matemáticos. Existen diferentes clasificaciones que pueden servir de ayuda para recordar la variedad de problemas que deben ser tratados en las aulas de Matemáticas de los distintos niveles educativos, Muñoz (2011)

Según la clasificación que hace Blanco (1993), los problemas matemáticos pueden ser:

3.4.1 Ejercicios de Reconocimiento.

Con este ejercicio se pretende resolver, reconocer o recordar un factor específico, una definición o una proposición de un teorema.

3.4.2 Ejercicios algorítmicos o de repetición

Son ejercicios que pueden ser resueltos con un proceso algorítmico, en estos ejercicios se trata de reforzar alguna expresión matemática determinada o potenciar las habilidades de cálculo.

En cualquier caso, y de acuerdo a referencias de Kanttowki, (1981), Schoenfeld, (1985), estos dos tipos de actividades no deben enmarcarse dentro de lo que entenderíamos por problemas, ya que su capacidad para desarrollar las habilidades de pensamiento y de estrategia de resolución de problemas son muy limitadas.

Para este tipo de actividades la matemática recreativa nos sugiere algunos ejemplos que, al mismo tiempo que ayudan a profundizar en algún algoritmo concreto, pueden ser utilizados como ejercicios de lógica.

3.4.3 Problemas de traducción simple o compleja.

Son problemas formulados en un contexto concreto y cuya resolución supone una traducción del enunciado, oral o escrito, a una expresión matemática. En el enunciado del problema aparece toda la información necesaria para la resolución del mismo y suele, implícitamente, indicar la estrategia a seguir. Se quiere reforzar la comprensión de los conceptos matemáticos y de las habilidades computacionales de los alumnos y conseguir que estos sean capaces de traducir situaciones del mundo real a expresiones matemáticas.

En estas actividades, el trabajo de resolución de un problema es doble: por una parte, traducir el enunciado por una serie de relaciones simbólicas o ecuaciones numéricas para posteriormente, resolver esas ecuaciones sucesivas aplicando las técnicas del cálculo numérico (Leif y Dezaly, 1961, p. 191).

3.4.4 Problemas de procesos.

Son problemas que se diferencian de los anteriores en que la forma de cálculo no aparece claramente delimitada, dándose la posibilidad de conjeturar varios caminos para encontrar la

solución. Este tipo de problemas intenta ejemplificar los procesos inherentes a su solución. Ayudan a desarrollar estrategias generales de comprensión, planificación y de solución de problemas. No aparece una estructura clara en el enunciado que posibilite, la traducción fácil a una expresión matemática. Esto posibilita diferentes formas de abordar el problema ya que al no disponer fácilmente de un algoritmo que lo resuelva le obliga a pensar sobre el mismo, y buscar estrategias de solución.

3.4.5 Problemas sobre situaciones reales

Se trata de plantear actividades lo más cercanas posibles a situaciones reales que requieran el uso de habilidades, conceptos y procesos matemáticos. Aunque no sean típicamente matemáticos al considerar otros tipos de información, las Matemáticas juegan un papel preponderante para encontrar la solución. Es una herramienta que ayuda a organizar, sintetizar y representar los datos, y dándole significado a las decisiones que se tomen.

Estos problemas dan oportunidad a la construcción de diagramas, a la realización de estimaciones, cálculo de las medidas, procesos de análisis y síntesis, pero sobre todo ayudan a comprender el significado de las Matemáticas y su relación con la realidad.

Estas actividades son importantes porque aportan a los alumnos ideas y maneras de aplicar conocimientos matemáticos fuera del aula, y ayuda a los estudiantes a evaluar y resolver mejor distintas situaciones que se les puedan plantear, desarrollando diferentes estrategias en análisis de contextos y creando formulaciones Matemáticas.

3.4.6 Problemas de investigación matemática.

Son problemas directamente relacionados con contenidos matemáticos, cuyas proposiciones pueden no contener ninguna estrategia para representarlos, y sugieren la búsqueda de algún modelo para encontrar la solución. En estas actividades son usuales las expresiones como "Probar que. . . "; "Encontrar todos los. . . "; "Para que . . . es . . . ?", etc.

De otra parte podemos incluir aquí referencias a prueba de una conjetura o la demostración de los teoremas como actividades de solución de problemas. La formulación de un teorema puede ser considerada, en algún contexto, como el planteamiento de un problema de investigación.

3.4.7 Problemas de puzles.

Son problemas en los que se pretende mostrar el potencial recreativo de las Matemáticas. Obliga a flexibilizar la forma de atacar un problema y a considerar varias perspectivas ya que normalmente el contexto y la formulación que se hacen de estos problemas suele ser engañosa.

3.4.8 Historias matemáticas.

Frecuentemente podemos observar en las librerías libros de cuentos, novelas entre los que encontramos algunas propuestas o planteamientos que requieren de nosotros un esfuerzo que impliquen algún concepto matemático.

Esto sucede, por una parte, en la lectura de autores que como Lewis Carrol han puesto al servicio de la literatura parte de su saber matemático. De otra parte, es fácil utilizar ciertas situaciones de la literatura que sin haber sido escritas con esa intencionalidad pueden aprovecharse

como material didáctico en la enseñanza de las Matemáticas. A este respecto es interesante recordar las referencias a los "viajes de Gulliver" que se establecen en el libro publicado por el Grupo Beta (1990, p. 139) a propósito de la proporcionalidad geométrica y la semejanza.

Así mismo, queremos hacer referencia a historias, más o menos completas, en cuyo interior guardan intencionadamente ciertas cuestiones matemáticas que van provocando la curiosidad y participación del lector. A este respecto libros como *Primos* o algunos *dislates* sobre números de S. Thio de Pol (1976) o *Las intrigantes aventuras del Doctor Ecco* de D. Shasha (1989) son una muestra interesante de cómo hacer matemáticas partiendo de la lectura.

3.5 Fases del proceso de resolución de problemas

Es ya clásica, y bien conocida, la formulación que hizo Polya (1989) de las cuatro etapas esenciales para la resolución de un problema, que constituyen el punto de arranque de todos los estudios posteriores:

Comprender el problema. Para la comprensión del problema el alumno tendrá que realizar una lectura detallada, para separar lo dado de lo buscado, lograr hallar alguna palabra clave u otro recurso que permita encontrar una adecuada orientación en el contexto de actuación, expresar el problema con sus palabras, realizar una figura de análisis, establecer analogías entre el problema y otros problemas o entre los conceptos y juicios que aparecen en el texto y otros conceptos y juicios incorporados al saber del individuo, o transferir el problema de un contexto a otro (Polya, 1989, p. 122).

Analizar el problema. Para ello el alumno deberá analizar nuevamente el problema para encontrar relaciones, precisando e interpretando el significado de los elementos dados y

buscados. Relacionará éstos con otros que puedan sustituirse en el contexto de actuación.

Generalizará las propiedades comunes a casos particulares, mediante la comparación de éstos sobre la base de la distinción de las cualidades relevantes y significativas de las que no lo son.

Tomará decisiones, al tener que comparar diferentes estrategias y procedimientos para escoger el más adecuado (Polya, 1989, p. 122).

Solucionar el problema. Para la realización de esta acción el alumno deberá: Aplicar a la solución del mismo los elementos obtenidos en el análisis del problema (Polya, 1989, p. 122).

Evaluar la solución del problema. El sujeto deberá analizar la solución planteada, contemplando diferentes variantes para determinar si es posible encontrar otra solución, verificando si la solución hallada cumple con las exigencias planteadas en el texto del problema. Valorar críticamente el trabajo realizado, determinando cuál solución es (Polya, 1989, p. 123).

Es preciso destacar que estas etapas no se dan separadas, aisladas entre sí, sino muy estrechamente unidas con un carácter de espiral, que se expresa en el hecho de quien resuelve el problema repite en determinados niveles un mismo tipo de actividad que caracteriza una etapa concreta.

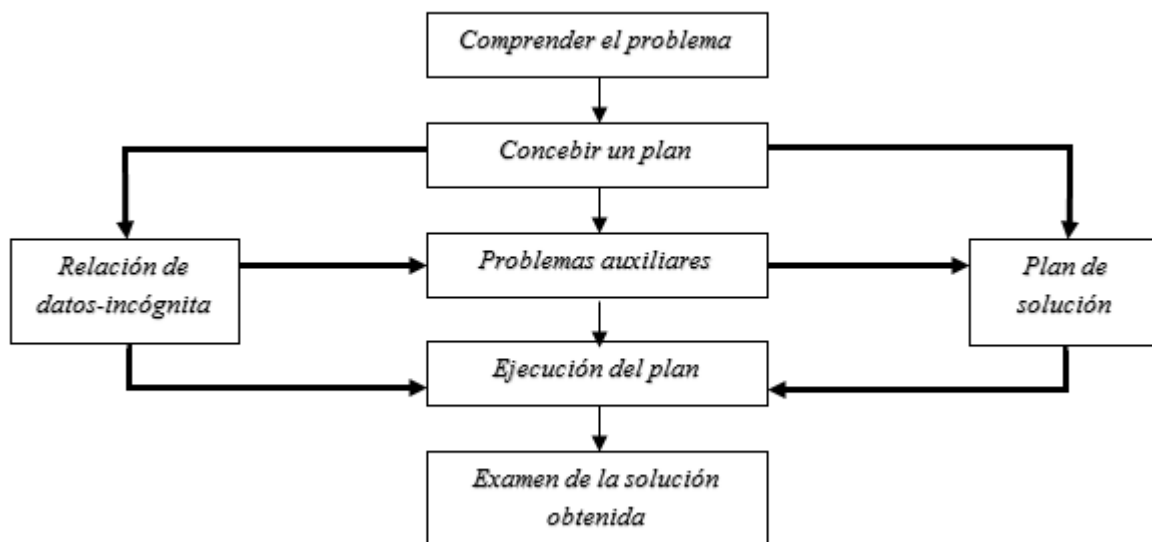


Figura 2 Estrategia de Polya para la resolución de problemas

Maza (1991) reformula el modelo de Polya, y diferencia dos procesos en la fase de Comprensión, en análisis y representación del problema y extendiendo la fase de Revisión-Comprobación de la siguiente forma:

Análisis del problema, lo que implica analizar-descomponer la información que nos da el enunciado (datos, condiciones, etc)

Representación del problema, relacionando los elementos del problema.

Planificación, eligiendo la estrategia más adecuada para su resolución.

Ejecución, o aplicación de la estrategia elegida, donde es conveniente la revisión constante de tal aplicación, detección de errores y corrección.

Generalización, conectándolo con algún principio general que permita resolver ejercicios similares en el futuro.

3.6 Dispositivos móviles en la educación

Es importante destacar las enormes posibilidades de las Tics, -entre ellos los dispositivos móviles- en un entorno de clase como recurso de aprendizaje presentan unas atractivas características como son: versatilidad, portabilidad, funcionalidad, el acceso a diversos materiales educativos y su uso natural, con la simple acción de tocar la pantalla el usuario interactúa con las aplicaciones.

En cuanto a las formas de trabajo, las posibilidades son ilimitadas, desde ejercicios de respuesta múltiple, respuesta corta, actividades de tipo tormenta de ideas, ejercicios de psicomotricidad en alumnado de corta edad o con necesidades educativas específicas, señalar sobre una imagen la parte que corresponda al contenido que se está tratando, seguir instrucciones para completar un gráfico o un diagrama, etc. Se trata de una estrategia que permitiría no sólo al profesor enviar contenidos y corregir actividades sino que también los alumnos podrían interactuar en las correcciones o presentar contenidos multimedia al resto de sus compañeros en clase.

Entre las posibles ventajas de esta tecnología para la clase podríamos destacar:

Trabajar contenidos de manera innovadora

Fomentar de la participación de los alumnos en clase.

Corregir errores con gran inmediatez.

Mantener el nivel de atención del alumno y su interés por seguir los contenidos.

Favorecer el pensamiento crítico y la creatividad permitiéndoles descubrir y participar en la construcción de su conocimiento.

Ayudar al docente a averiguar el grado de comprensión de los contenidos en tiempo real.

Flexibilizar la presentación de contenidos.

Monitorear aprendizajes

4 Descripción del desarrollo tecnológico

El ambiente digital - aplicación móvil se diseña por medio de la plataforma online AppInventor del MIT - Massachusetts Institute of Technology. Dicha aplicación móvil es creada por el investigador con el propósito de evaluar y diagnosticar a los estudiantes participantes de la investigación en la capacidad de resolución de problemas de razonamiento algorítmico.

4.1 Modelo Pedagógico utilizado en la app

En el diseño de las pruebas se utiliza como referencia el modelo Test de Pensamiento Computacional de Román, Pérez y Jiménez para diagnosticar la resolución de problemas. Dichos autores proponen medir el nivel de aptitud desarrollo del pensamiento computacional en el sujeto.

Según Román et al (2015) el pensamiento computacional es la capacidad de formular y solucionar problemas apoyándose en los conceptos fundamentales de la computación, y usando la lógica sintaxis de los lenguajes informáticos de programación: secuencias básicas, bucles, iteraciones, condicionales, funciones y variables.

El modelo está basado en cinco dimensiones (Concepto computacional abordado, Entorno interfaz del ítem, Estilo de las alternativas de la respuesta, Existencia o inexistencia de anidamiento y Tarea requerida). Las dimensiones anteriormente descritas abordan 7 conceptos computacionales, ordenados en dificultad creciente: Direcciones básicas; Bucles-‘repetir veces’; Bucles-‘repetir hasta’; Condicional simple-‘if’; Condicional compuesto-‘if/else’; Mientras que-‘while’; Funciones simples. Estos conceptos están alineados con los estándares que fija la CSTA para la educación en ciencias de la computación en estas edades (CSTA, 2011).

Para el desarrollo de la prueba Pretest se acoge la dimensión Tareas requeridas evaluando la secuenciación el análisis del problema implica analizar-descomponer la información que nos da el enunciado (datos, condiciones, etc) y por su parte, en la representación del problema, el sujeto debe relacionar los elementos del problema.

4.2 Diseño del ambiente digital: comprensión de problemas-app

Esta prueba se ejecuta por medio de 15 tableros dividido en tres grupos, en donde el estudiante debe utilizar su lógica para resolver las tareas planteadas.

Tabla 3 Fases y actividades del diagnóstico de los grupos de investigación

Fases de la prueba	Aspecto a evaluar	Actividad
Fase 0	Ninguno	Presentación de la App Prest
Fase 1	Tiempo para resolver los problemas	Llegar al objetivo.
	Ruta más corta	Tableros del 1 al 5
Fase 2	Secuencialidad	Recoger todos los objetos.
	Seguir instrucciones	Tableros del 6 al 10
Fase 3		Poner objetos en el campo
		Tableros del 11 al 15

Fase 0. Presentación de la App Prest. En esta fase se presenta la aplicación Prest a los estudiantes, se realiza una explicación de su funcionalidad y cómo debe interactuar para cumplir

el objetivo. En la figura 3 se muestra la interfaz de ingreso al aplicativo. Y en la figura 4 se detalla el objetivo al cual debe llegar el usuario y la explicación para interactuar con el aplicativo.



Figura 3 Interfaz para el ingreso a la app



Figura 4 Pantalla de Introducción de Prest

Fase 1. Llegar al objetivo. En esta fase la app evalúa la capacidad del sujeto para llegar a un objetivo en el menor tiempo posible y con la menor cantidad de pasos. El sujeto debe identificar una ruta que cumpla dos condiciones, llegar al objetivo y evadir los obstáculos

presentes en cada desafío. No existe un tiempo límite, el sujeto de investigación pasará a un nuevo desafío cuando llegue al objetivo. A continuación en las figuras 5 se muestran los 5 primeros escenarios.

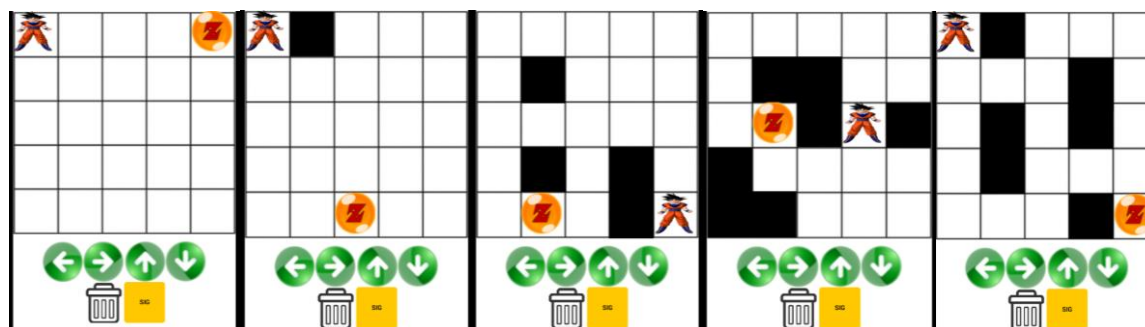


Figura 5 Escenarios Nivel 1 Llegar al objetivo

Fase 2. En esta fase el sujeto de investigación debe afrontar una dificultad un poco mayor, ya que debe ir a múltiples objetos que están ubicados en el mismo escenario. Para ello, el estudiante debe crear una estrategia planificando la ruta más corta y ejecutándola. A continuación se muestran los tableros de la segunda fase.



Figura 6 Introducción del nivel 2

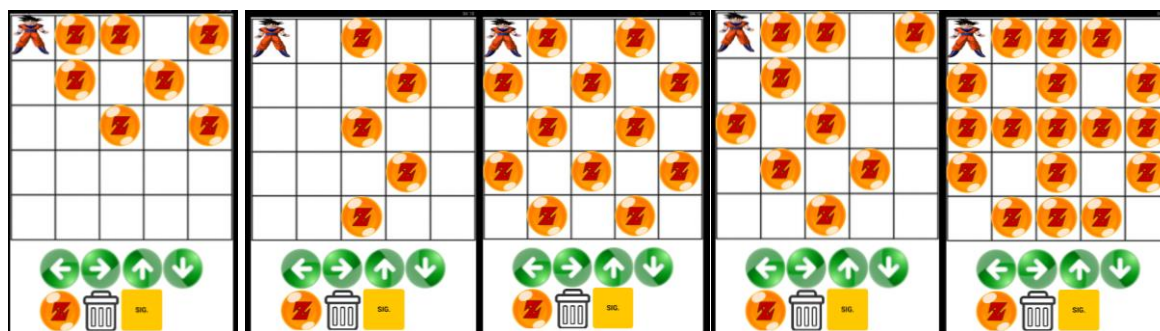


Figura 7 Escenarios Nivel 2 Objetivos múltiples

Fase 3. Siguiendo la ruta. Esta última fase de la app Prest se busca evaluar la secuencialidad o la capacidad que tiene cada sujeto para seguir instrucciones. En cada tablero se le suministrará una serie de instrucciones que deberá seguir. Se evalúa si logra el objetivo y en qué porcentaje. A continuación se puede observar los escenarios con los que interactúan los sujetos en el nivel 3 de la app Prest.

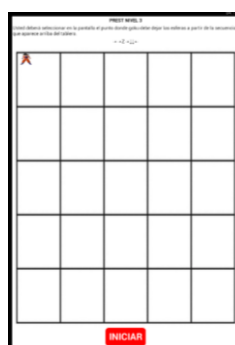


Figura 8 Introducción del Nivel 3

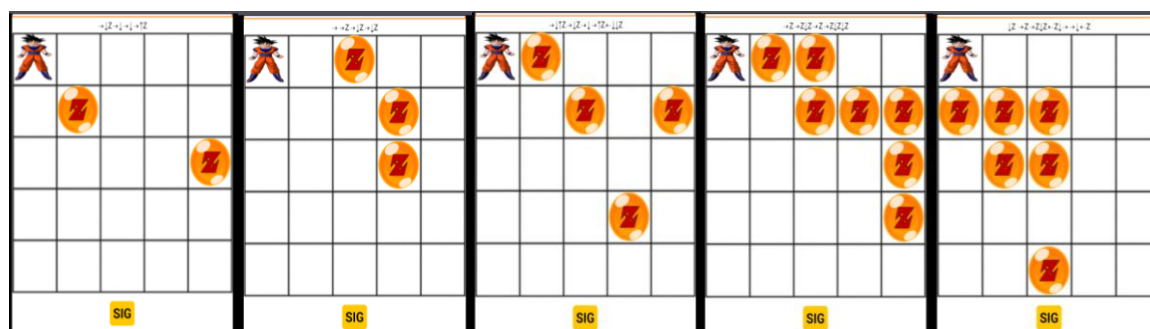


Figura 9 Escenario Nivel 3 Siguiendo Instrucciones

Medición en la app-Prest

La aplicación móvil Prest se diseña para establecer una prueba pretest que tiene como fin validar si los grupos experimentales son homogéneos. Para ello, la aplicación cuenta con conexión a base datos, donde se almacenará el tiempo y la secuencia de pasos, con el fin de evaluar la eficiencia en la solución de problemas comparando la solución que un experto ha utilizado. La cantidad de ejercicios por cada grupo de problema se relacionan en la tabla 3.

Tabla 4 Número de Ejercicios por cada problema

	Descomposición de la Información (identificación de meta y condiciones)	Relación de la Información (identificación de datos necesarios)	Identificación de la Información (memorización de datos)	Total ítems
Problema 1	5 situaciones	0 situaciones	0 situaciones	5 situaciones
Problema 2	0 situaciones	5 situaciones	5 situaciones	5 situaciones
Problema 3	0 situaciones	0 situaciones	5 situaciones	5 situaciones
Total ítems	5 situaciones	5 situaciones	5 situaciones	5 situaciones

De tal forma que al final de la prueba, la app pondera en un rango de 0 a 100 la valoración de la eficiencia para resolver problemas del estudiante en cada una de las fases de la solución de problemas secuenciales tenidas en cuenta para la investigación.

5 Metodología

5.1 Pregunta de investigación

¿Existe diferencias significativas en el desarrollo del pensamiento algorítmico entre un grupo que interactúa con un ambiente con técnicas de diagramación Diagramas de Nassi-Schneiderman frente a otro grupo que trabaja con técnicas de diagramación Diagramas de Flujo?

5.2 Variables

5.2.1 Variables Independientes

Resolución de problemas utilizando Diagramas Nassi-Shneiderman.

Resolución de problemas utilizando Diagramas de flujo.

5.2.2 Variable Dependiente

Desarrollo del pensamiento algorítmico.

5.3 Hipótesis

5.3.1 Hipótesis 0.

No existe diferencia en el desarrollo del pensamiento algorítmico entre dos grupos de estudiantes que resuelven problemas en un ambiente mediado por computador que interactúa bajo

la representación del conocimiento basado en diagramas de flujo y otro bajo la representación de conocimiento basado en diagramas Nassi-Shneiderman.

5.3.2 Hipótesis 1.

Los estudiantes que resuelven problemas en interacción con un ambiente de aprendizaje que usa representación de conocimiento basada en diagramas de flujo se desempeña mejor, que quienes interactúan con el mismo ambiente pero con representación de conocimiento basado en Nassi-Shneiderman.

5.3.3 Hipótesis 2

Los estudiantes que resuelven problemas en interacción con un ambiente de aprendizaje que usa representación de conocimiento basada en diagramas de Nassi-Shneiderman se desempeña mejor, que quienes interactúan con el mismo ambiente pero con representación de conocimiento basado en diagramas de flujo.

5.4 Tipo de Investigación

El enfoque de la investigación será de tipo cuasi experimental debido a que se seleccionarán dos grupos de sexto grado de la IED Unión Europea que están preestablecidos. Los problemas serán seleccionados de un banco de ejercicios que se ha preparado. El propósito en este tipo de investigaciones es determinar relaciones de causa-efecto exponiendo los grupos a estímulos y los comportamientos resultantes se comparan con los del otro grupo.

5.5 Población y Muestra

5.5.1 Población.

La población dentro de la cual se realiza la investigación corresponde a 60 estudiantes cuyas edades oscilan entre los 9 y 14 años aproximadamente. Los estudiantes de grado sexto pertenecen al colegio Unión Europea sede A, ubicado al sur de la ciudad de Bogotá en la localidad de Ciudad Bolívar (19), entre los barrios Gibraltar, Domingo Laín y Lucero Bajo, que cuenta con dos jornadas atendiendo a estudiantes pertenecientes a los estratos 1 y 2.

5.5.2 Muestra.

El trabajo se realiza con 60 estudiantes del grado sexto, jornada tarde divididos en dos grupos, caracterizados así:

5.5.3 Grupo 1

Estudiantes que van a interactuar con el ambiente computacional basado en Diagramas de Nassi-Shneiderman.

5.5.4 Grupo 2

Estudiantes que van a interactuar con el ambiente computacional basado en Diagramas de flujo.

5.6 Diseño Metodológico

Para llevar a cabo la investigación se propone 4 fases, organizadas de la siguiente manera:

La primera fase consiste en la presentación de la técnica de diagramación correspondiente a cada grupo, utilizando como ejemplo situaciones cotidianas a los estudiantes, luego se amplía la explicación y se procede a plantear problemas que son resueltos por los estudiantes. Posterior a

desarrollar esa fase se realimenta de manera grupal para identificar los aciertos y las equivocaciones cometidos en el uso de la técnica de diagramación seleccionada.

La segunda fase consiste en la interacción de la app “Prest”, diseñada para medir los niveles de eficiencia y tiempo utilizado en la solución de los problemas de laberintos, recolección de objetos y seguir instrucciones, para así identificar el grado de homogeneidad de los dos grupos de muestra. Los problemas a desarrollar buscan evidenciar aspectos como seguir instrucciones, gastar la menor cantidad de pasos para llegar al objetivo y recolectar objetos en el menor tiempo posible como lo indica (Guerrero, 2016) y (Román y Pérez, 2015) en el test diagnóstico de pensamiento computacional.

Esta aplicación móvil aborda tres problemáticas anteriormente descritas y en cada una de ellas se manejan 5 tableros, iniciando con el de menor dificultad e irá aumentando con el progreso del participante.

La prueba de laberintos consiste en llegar al objetivo con el menor número de pasos. Cabe resaltar que este tipo de problemáticas se ejecutan en el hemisferio izquierdo y como lo menciona (Morris, Ahmed, Syed & Toone, 1993) la porción dorsal del hemisferio izquierdo se encuentra estrechamente relacionada con los procesos de planeación, memoria de trabajo, fluidez, flexibilidad mental, seriación, secuenciación, estrategias de memoria verbal, codificación de memoria semántica y secuencias inversas. Desde el punto de vista neuropsicológico se ha visto que las pruebas de laberintos incluyen tres tipos de tareas: Planeación, Ejecución viso-espacial y Respeto de reglas.

La planeación es la capacidad para identificar, integrar, organizar y desarrollar una serie de pasos intermedios en una secuencia adecuada para lograr metas a corto, mediano o largo plazo.

En ocasiones trabaja de manera conjunta con la flexibilidad mental, ya que requiere de adaptaciones a nuevas circunstancias en el desarrollo del plan para alcanzar objetivos.

La fluidez es un importante atributo de la corteza prefrontal y consiste en la velocidad y precisión con que se busca y actualiza información, así como la producción de elementos específicos en un tiempo determinado.

El tercer grupo de problemas seguir instrucciones está orientado a evaluar la capacidad de los participantes a interpretar un algoritmo.

La fase 3, se dan instrucciones a cada grupo frente a la técnica de diagramación correspondiente mediante el uso de una plataforma educativa “MILAULAS”, que funciona como aula virtual, tutoriales, canal de retroalimentación y repositorio de ejercicios; y también la herramienta “PSeINT”, ya que en ella se puede estructurar, construir y graficar cada una de las técnicas, además al ser combinadas también hacen parte del proceso de afianzamiento de ellas.



Figura 10 Modelo de la metodología

En la última etapa los participantes interactúan con dos aplicaciones Code.org y Plastelina.net, que presentan diferentes problemas programáticos, utilizando los modelos iniciales de resolución de problemas, secuenciales, cíclicos y problemas de toma de decisiones.

6 Análisis de datos

Se realizan diferentes pruebas ANOVA de un solo factor con el fin de estudiar las relaciones significativas entre las variables, de acuerdo al tipo de problema planteado; centrando el análisis en la eficiencia con base en el número de eventos y de la eficiencia con base en el tiempo de ejecución.

De esta forma se pretende cumplir con los objetivos específicos y por ende con el objetivo general de la investigación que es comparar la incidencia de las técnicas Diagramas Nassi-Shneiderman y Diagramas de flujo en el desarrollo del pensamiento algorítmico en la resolución de problemas.

7 Resultados

A partir de los resultado obtenidos en los test finales con la ayuda del software Statistical Package for the Social Sciences (SPSS) versión 21 como se ilustra en este apartado.

La pregunta de investigación a la que se le dio respuesta fue: ¿Existe diferencias significativas en el desarrollo del pensamiento algorítmico entre un grupo que interactúa con un ambiente con técnicas de diagramación Diagramas de Nassi-Schneiderman frente a otro grupo que trabaja con técnicas de diagramación Diagramas de Flujo?

Los ambientes usados para dar respuesta a la pregunta de investigación, nos brindaron las variables “Eficiencia con base en el tiempo y eficiencia con base en el número de eventos”, en la solución de los problemas planteados.

Para el desarrollo de la investigación se utilizó un diseño cuasi experimental, con dos grupos de estudiantes, uno con énfasis en diagramas de N-S y el otro con diagramas de Flujo para explorar las competencias del pensamiento algorítmico entre ellos.

Prueba de normalidad en el Pretest

Para verificar el supuesto de normalidad en el pretest, se utiliza la prueba de Kolmogorov-Smirnov, esta prueba permite contrastar la hipótesis que los datos provienen de una distribución normal. Para determinar la normalidad se tomaron los resultados totales de la eficiencia con base al tiempo empleado para resolver los problemas y la eficiencia con base en el número de eventos de los 60 estudiantes que participaron en la investigación. Como la muestra es mayor o igual a 30 se aplicó la prueba de Kolmogorov-Smirnov para normalidad, como se muestra en la Tabla 5:

Tabla 5 Prueba de Kolmogorov-Smirnov para una muestra

		PreEfiTiempoT	PreEficiNuET
N		60	60
Parámetros normales ^{a,b}	Media	466,32	65,7467
	Desviación típica	131,513	9,46221
Z de Kolmogorov-Smirnov		,759	,683
Sig. asintót. (bilateral)		,613	,740

Para la eficiencia con base en el Tiempo en el Pretest (PreEfiTiempoT) la significancia bilateral es 0,613 mayor que 0,05; por tanto, se deduce que ambos grupos tienen distribución normal, lo que abre el camino para aplicar la prueba T.

Para la eficiencia con base en el número de eventos en el Pretest (PreEficiNuET) la significancia bilateral es 0,740 mayor que 0,05, por tanto, se deduce que ambos grupos tienen distribución normal, lo que abre el camino para aplicar la prueba Anova.

Se acepta la hipótesis que los datos provienen de una distribución normal, se da validez a la hipótesis nula que enuncia que los grupos experimentales no tienen diferencias significativas. Estos resultados implican que se puede aplicar la prueba Anova de un factor. A continuación, en las figuras 11, 12, 13 y 14 se muestran los histogramas y las curvas de normalidad en la prueba posttest de las variables Eficiencia con base en el tiempo y Eficiencia con base en el número de eventos.

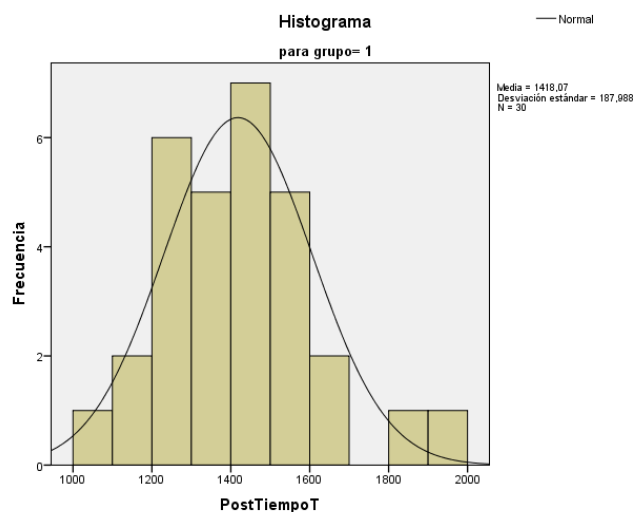


Figura 11 Histograma y curva normal Eficiencia con base en el tiempo Grupo N-S

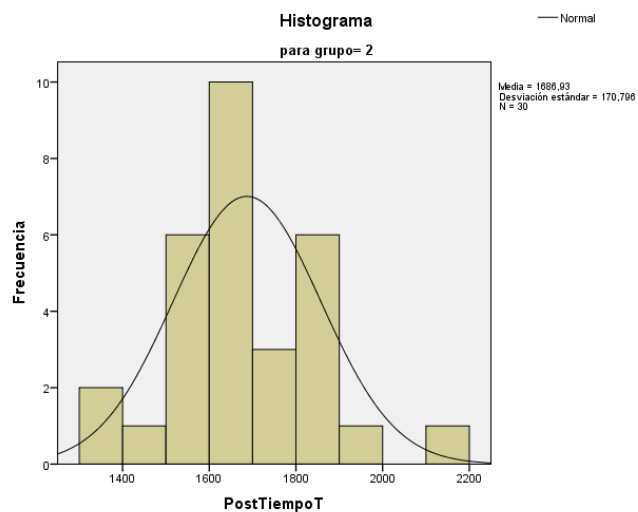


Figura 12 Histograma y curva normal Eficiencia con base en el tiempo Grupo Flujo

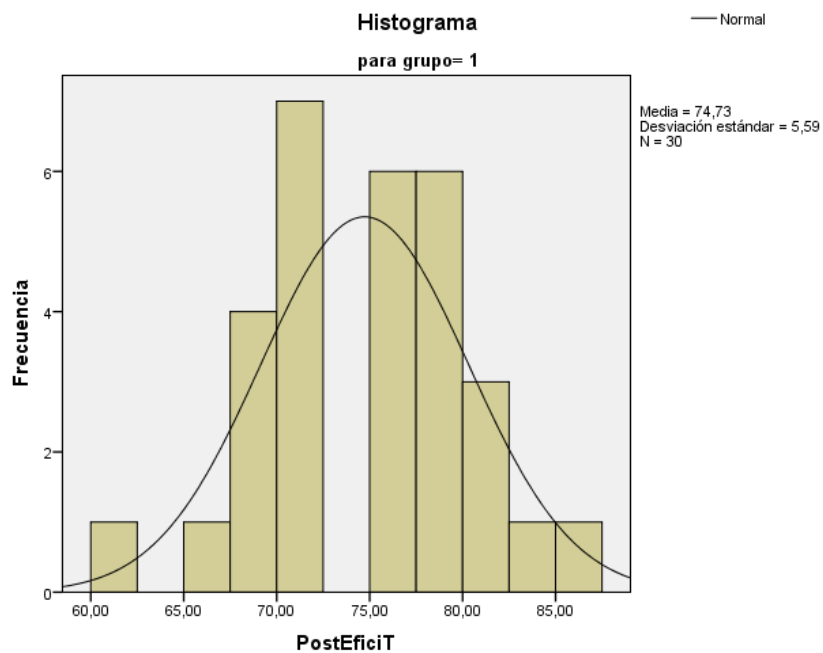


Figura 13 Histograma y curva normal Eficiencia con base en el número de eventos Grupo

N-S

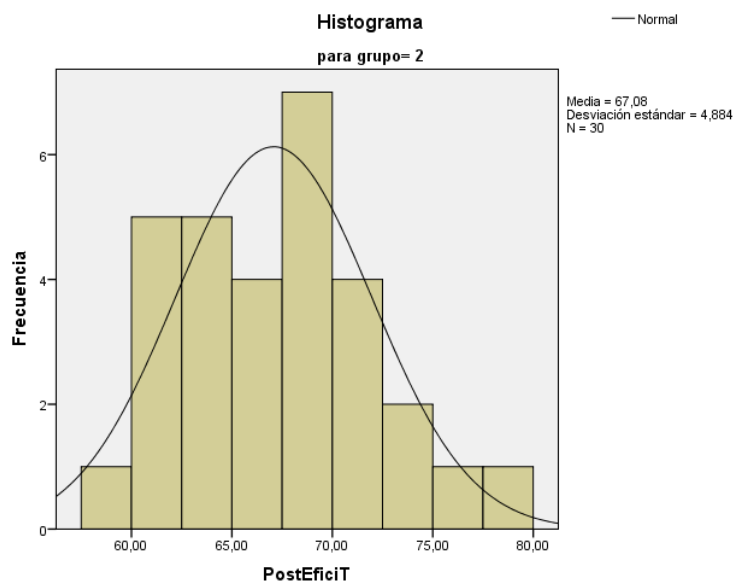


Figura 14 Histograma y curva normal Eficiencia con base en el número de eventos Grupo Flujo

7.1 Anova de un factor

Para estudiar la relación de la eficiencia con base en el tiempo y la eficiencia con base en el número de eventos, se aplica la prueba ANOVA de un factor para cada una de los tipos de problemas de manera separada (Tipo I: Problemas secuenciales, Tipo II: Problemas de iteraciones y Tipo III: Problemas de toma de decisiones).

La tabla 6 muestra los resultados estadísticos de la eficiencia basada en el tiempo para los problemas tipo I. Por otra parte se observa que la media de los resultados obtenidos en el tratamiento Nassi-Shneiderman para los problemas tipo I es de 128.33 segundos y para el tratamiento Diagramas de Flujo es de 139.13 segundos. Además, el resultado para el grupo Nassi-Shneiderman más alto fue de 157 segundos y el más bajo fue de 109 segundos, en cambio para el grupo Diagramas de Flujo el resultado más alto fue de 172 segundos y el más bajo fue de 111 segundos.

Tabla 6 Estadísticos descriptivos Eficiencia con base en el tiempo Problemas tipo I

N	Media	Desviación estándar	Error estándar	Mínimo	Máximo	
N-S	30	128,33	14,435	2,635	109	157
DF	30	139,13	11,461	2,093	111	172
Total	60	133,73	14,023	1,810	109	172

La tabla 7 muestra los resultados estadísticos de la eficiencia basada en el número de eventos para los problemas tipo I. Por otra parte se observa que la media de los resultados obtenidos en el tratamiento Nassi-Shneiderman para los problemas tipo I es de 84.431% y para el tratamiento Diagramas de Flujo es de 83.6608%. Además, el resultado para el grupo Nassi-Shneiderman más alto fue de 100% y el más bajo fue de 58.97%, en cambio para el grupo Diagramas de Flujo el resultado más alto fue de 100% y el más bajo fue de 60.91%.

Tabla 7 Estadísticos descriptivos Eficiencia con base en el número de eventos Problemas tipo I

	N	Media	Desviación típica	Error típico	Mínimo	Máximo
Nassi-Shneiderman	30	84,4310	12,83779	2,34385	58,97	100,00
Flujo	30	83,6608	10,68344	1,95052	60,91	100,00
Total	60	84,0459	11,71577	1,51250	58,97	100,00

La tabla 8 muestra los resultados estadísticos de la eficiencia basada en el tiempo para los problemas tipo II. Por otra parte se observa que la media de los resultados obtenidos en el tratamiento Nassi-Shneiderman para los problemas tipo II es de 174.97 segundos y para el

tratamiento Diagramas de Flujo es de 185.10 segundos. Además, el resultado para el grupo Nassi-Shneiderman más alto fue de 197 segundos y el más bajo fue de 155 segundos, en cambio para el grupo Diagramas de Flujo el resultado más alto fue de 203 segundos y el más bajo fue de 162 segundos.

Tabla 8 Estadísticos descriptivos Eficiencia con base en el tiempo Problemas tipo II

	N	Media	Desviación estándar	Error estándar	Mínimo	Máximo
N-S	30	174,97	10,972	2,003	155	197
DF	30	185,10	11,978	2,187	162	203
Total	60	180,03	12,482	1,611	155	203

La tabla 9 muestra los resultados estadísticos de la eficiencia basada en el número de eventos para los problemas tipo II. Por otra parte se observa que la media de los resultados obtenidos en el tratamiento Nassi-Shneiderman para los problemas tipo II es de 75.0667% y para el tratamiento Diagramas de Flujo es de 67.9444%. Además, el resultado para el grupo Nassi-Shneiderman más alto fue de 86% y el más bajo fue de 60.50%, en cambio para el grupo Diagramas de Flujo el resultado más alto fue de 84.33% y el más bajo fue de 53.83%.

Tabla 9 Estadísticos descriptivos Eficiencia con base en el número de eventos Problemas tipo II

	N	Media	Desviación estándar	Error estándar	Mínimo	Máximo
1	30	75,0667	6,91306	1,26215	60,50	86,00
2	30	67,9444	7,30734	1,33413	53,83	84,33
Total	60	71,5056	7,91408	1,02170	53,83	86,00

La tabla 10 muestra los resultados estadísticos de la eficiencia basada en el tiempo para los problemas tipo III. Por otra parte se observa que la media de los resultados obtenidos en el tratamiento Nassi-Shneiderman para los problemas tipo III es de 1114,77 segundos y para el tratamiento Diagramas de Flujo es de 1362,70 segundos. Además, el resultado para el grupo Nassi-Shneiderman más alto fue de 1594 segundos y el más bajo fue de 757 segundos, en cambio para el grupo Diagramas de Flujo el resultado más alto fue de 1821 segundos y el más bajo fue de 1024 segundos.

Tabla 10 Estadísticos descriptivos Eficiencia con base en el tiempo Problemas tipo III

	N	Media	Desviación estándar	Error estándar	Mínimo	Máximo
1	30	1114,77	181,345	33,109	757	1594
2	30	1362,70	171,784	31,363	1024	1821
Total	60	1238,73	215,168	27,778	757	1821

La tabla 11 muestra los resultados estadísticos de la eficiencia basada en el número de eventos para los problemas tipo III. Por otra parte se observa que la media de los resultados obtenidos en el tratamiento Nassi-Shneiderman para los problemas tipo III es de 57,2333 % y para el tratamiento Diagramas de Flujo es de 50,3444%. Además, el resultado para el grupo Nassi-Shneiderman más alto fue de 70% y el más bajo fue de 41.67%, en cambio para el grupo Diagramas de Flujo el resultado más alto fue de 70% y el más bajo fue de 33.33%.

Tabla 11 Estadísticos descriptivos Eficiencia con base en el número de eventos Problemas tipo III

	N	Media	Desviación estándar	Error estándar	Mínimo	Máximo
1	30	57,2333	7,55828	1,37995	41,67	70,00
2	30	50,3444	9,14652	1,66992	33,33	70,00
Total	60	53,7889	9,01473	1,16380	33,33	70,00

7.1.1 Prueba Anova de un Factor para Problemas tipo I.

En los siguientes dos apartados se realiza el análisis estadístico en los problemas secuenciales evaluando la relación en los tratamientos realizados con respecto a la Eficiencia. La eficiencia se evalúa de las maneras: Eficiencia con base en el tiempo y eficiencia con base en el número de eventos.

7.1.1.1 Anova Eficiencia con base en el tiempo problemas tipo I

Al aplicar la prueba Anova de un factor para la Eficiencia con base en el tiempo para los problemas tipo I se observan los resultados de la prueba de homogeneidad de varianzas en la tabla 12, señalando que las varianzas no son homogéneas ($Levene_{(1, 58)}=7.900$, $p=0.007$). Por lo tanto, no se cumple el supuesto de homogeneidad de las varianzas y no se puede tener en cuenta los resultados de la tabla 13. Por ende, se procede a realizar unas pruebas robustas de igualdad de medias (análisis Anova con el estadístico Welch) como se muestra en la tabla 14.

Tabla 12 Prueba de Homogeneidad de Varianzas Eficiencia con base en el tiempo problemas tipo I

Estadístico de Levene	gl1	gl2	Sig.
7,900	1	58	,007

Tabla 13 Anova Eficiencia con base en el tiempo. Ejercicios Tipo I

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Entre grupos	1749,600	1	1749,600	10,300	,002
Dentro de grupos	9852,133	58	169,864		
Total	11601,733	59			

Los resultados de la prueba robusta de igualdad de medias nos señala que hay un nivel de significancia de 0.001 menor que 0.05 ($p < 0.05$), donde se puede concluir que existe diferencias significativas en las medias de los grupos N-S y Flujo.

Tabla 14 Pruebas robustas de igualdad de las medias

	Estadístico	gl1	gl2	Sig.
Welch	11,731	1	52,488	,001
Brown-Forsythe	11,731	1	52,488	,001

7.1.1.2 Anova Eficiencia con base en el número de eventos problemas tipo I

Al aplicar la prueba Anova de un factor para la Eficiencia con base en el número de eventos para los problemas tipo I se observan los resultados de la prueba de homogeneidad de varianzas en la tabla 15, señalando que las varianzas son homogéneas ($Levene_{(1, 58)}=1.735$, $p=0.193$). Por lo tanto, se cumple el supuesto de homogeneidad de las varianzas.

Tabla 15 Prueba de Homogeneidad de Varianzas Eficiencia con base en el número de eventos problemas tipo I

Estadístico de Levene	gl1	gl2	Sig.
1,735	1	58	,193

Al realizar la prueba Anova de un factor para la eficiencia con base en el número de eventos, tabla 16, se encuentra que el nivel de significancia es 0.845 mayor que 0.05 se acepta la hipótesis nula, no existen diferencias significativas en los grupos.

Tabla 16 Anova Eficiencia con base en el número de eventos problemas tipo I

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Inter-grupos	,017	1	,017	,038	,845
Intra-grupos	24,981	58	,431		
Total	24,997	59			

7.1.2 Prueba Anova de un Factor para Problemas tipo II

En los siguientes dos apartados se realiza el análisis estadístico en los problemas de iteración evaluando la relación en los tratamientos realizados con respecto a la Eficiencia. La eficiencia se evalúa de las maneras: Eficiencia con base en el tiempo y eficiencia con base en el número de evento.

7.1.2.1 Anova Eficiencia con base en el tiempo problemas tipo II

Al aplicar la prueba Anova de un factor para la Eficiencia con base en el tiempo para los problemas tipo II se observan los resultados de la prueba de homogeneidad de varianzas en la tabla 17, señalando que las varianzas son homogéneas ($Levene_{(1, 58)}=1.133, p=0.291$). Por lo tanto, se cumple el supuesto de homogeneidad de las varianzas y se puede proceder hacer la lectura de la prueba Anova.

Tabla 17 Prueba de Homogeneidad de Varianzas Eficiencia con base en el tiempo problemas tipo II

Estadístico de Levene	gl1	gl2	Sig.
1,133	1	58	,291

Al realizar la prueba Anova de un factor para la eficiencia con base en el número de eventos, tabla 18, se encuentra que el nivel de significancia es 0.001 menor que 0.05 se acepta la hipótesis alterna, existen diferencias significativas en los grupos.

Tabla 18 Anova Eficiencia con base en el tiempo. Ejercicios Tipo II

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Inter-grupos	1540,267	1	1540,267	11,675	,001
Intra-grupos	7651,667	58	131,925		
Total	9191,933	59			

7.1.2.2 Anova Eficiencia con base en el número de eventos problemas tipo II

Al aplicar la prueba Anova de un factor para la Eficiencia con base en el número de eventos para los problemas tipo II se observan los resultados de la prueba de homogeneidad de varianzas en la tabla 19, señalando que las varianzas son homogéneas ($Levene_{(1, 58)}=0.020$, $p=0.889$). Por lo tanto, se cumple el supuesto de homogeneidad de las varianzas.

Tabla 19 Prueba de Homogeneidad de Varianzas Eficiencia con base en el número de eventos problemas tipo II

Estadístico de Levene	gl1	gl2	Sig.
,020	1	58	,889

Al realizar la prueba Anova de un factor para la eficiencia con base en el número de eventos, tabla 20, se encuentra que el nivel de significancia es 0.000 menor que 0.05 se acepta la hipótesis alterna, existen diferencias significativas en los grupos.

Tabla 20 Anova Eficiencia con base en el número de eventos. Ejercicios Tipo II

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Inter-grupos	760,891	1	760,891	15,039	,000
Intra-grupos	2934,441	58	50,594		
Total	3695,331	59			

7.1.3 Prueba Anova de un Factor para Problemas tipo III

En los siguientes dos apartados se realiza el análisis estadístico en los problemas de toma de decisiones evaluando la relación en los tratamientos realizados con respecto a la Eficiencia. La eficiencia se evalúa de las maneras: Eficiencia con base en el tiempo y eficiencia con base en el número de eventos para los problemas tipo III.

7.1.3.1 Anova Eficiencia con base en el tiempo problemas tipo III

Al aplicar la prueba Anova de un factor para la Eficiencia con base en el tiempo para los problemas tipo III se observan los resultados de la prueba de homogeneidad de varianzas en la tabla 21, señalando que las varianzas son homogéneas ($Levene_{(1, 58)}=0.031$, $p=0.860$). Por lo tanto, se cumple el supuesto de homogeneidad de las varianzas

Tabla 21 Prueba de Homogeneidad de Varianzas Eficiencia con base en el tiempo problemas tipo III

Estadístico de Levene	gl1	gl2	Sig.
,031	1	58	,860

Al realizar la prueba Anova de un factor para la eficiencia con base en el número de eventos, tabla 22, se encuentra que el nivel de significancia es 0.000 menor que 0.05 se acepta la hipótesis alterna, existen diferencias significativas en los grupos.

Tabla 22 Anova Eficiencia con base en el tiempo. Ejercicios Tipo III

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Inter-grupos	922064,067	1	922064,067	29,555	,000
Intra-grupos	1809481,667	58	31197,960		
Total	2731545,733	59			

7.1.3.2 Anova Eficiencia con base en el número de eventos problemas tipo III

Al aplicar la prueba Anova de un factor para la Eficiencia con base en el número de eventos para los problemas tipo III se observan los resultados de la prueba de homogeneidad de varianzas en la tabla 23, señalando que las varianzas son homogéneas ($Levene_{(1, 58)}=0.173$, $p=0.679$). Por lo tanto, se cumple el supuesto de homogeneidad de las varianzas.

Tabla 23 Prueba de Homogeneidad de Varianzas Eficiencia con base en el número de eventos problemas tipo III

Estadístico de Levene	gl1	gl2	Sig.
,173	1	58	,679

Al realizar la prueba Anova de un factor para la eficiencia con base en el número de eventos, tabla 24, se encuentra que el nivel de significancia es 0.002 menor que 0.05 se acepta la hipótesis alterna, existen diferencias significativas en los grupos.

Tabla 24 Anova Eficiencia con base en el número de eventos. Ejercicios Tipo III

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Inter-grupos	711,852	1	711,852	10,113	,002
Intra-grupos	4082,807	58	70,393		
Total	4794,659	59			

7.2 Análisis de Varianza Factorial diseño mixto Medidas Repetidas

Se realizó un análisis de medidas repetidas para ver la evolución de los grupos. Como factores para este análisis se seleccionaron Momentos, tipos de problemas y las variables de eficiencia. Teniendo como momentos el pretest (1) y el posttest (2). En el factor ejercicios se tiene secuenciales (1), iterativos (2) y de toma de decisión (3). Y las variables de la eficiencia fueron tiempo (1) y número de eventos (2). En las tablas 25 y 26 e encuentra los factores intra-sujetos para este análisis.

Tabla 25 Factores intra-sujetos

Momentos	ejercicios	Variables	Variable dependiente
1	1	1	PretTiempo1
		2	PreEfici1
	2	1	PretTiempo2
		2	PreEfici2
	3	1	PretTiempo3
		2	PreEfici3
2	1	1	PostTiempo1
		2	PostEfici1
	2	1	PostTiempo2
		2	PostEfici2
	3	1	PostTiempo3
		2	PostEfici3

Tabla 26 Factores intra-sujeto II

	Etiqueta del valor	N
grupo	1 Nassi-Shneiderman	30
	2 Flujo	30

A continuación en la tabla 27 se encuentran los estadísticos descriptivos media, desviación típica, y el tamaño de la muestra usada para esta investigación.

Tabla 27 Estadísticos descriptivos

	Grupo	Media	Desviación típica	N
PretTiempo1	Nassi-Shneiderman	69,9000	19,58773	30
	Flujo	54,5333	19,94775	30
	Total	62,2167	21,07621	60
PreEfici1	Nassi-Shneiderman	84,4310	12,83779	30
	Flujo	83,6608	10,68344	30
	Total	84,0459	11,71577	60
PretTiempo2	Nassi-Shneiderman	282,9667	102,36765	30
	Flujo	205,2333	83,62135	30
	Total	244,1000	100,61804	60
PreEfici2	Nassi-Shneiderman	78,6808	13,58539	30
	Flujo	84,7811	8,28142	30
	Total	81,7310	11,57100	60
PretTiempo3	Nassi-Shneiderman	168,8000	58,81907	30
	Flujo	149,8667	56,82929	30
	Total	159,3333	58,12980	60
PreEfici3	Nassi-Shneiderman	29,5533	15,16336	30
	Flujo	33,3730	23,38581	30
	Total	31,4632	19,63512	60
PostTiempo1	Nassi-Shneiderman	128,33	14,435	30
	Flujo	139,13	11,461	30
	Total	133,73	14,023	60
PostEfici1	Nassi-Shneiderman	91,8889	9,04155	30
	Flujo	82,9444	7,92304	30
	Total	87,4167	9,55914	60
PostTiempo2	Nassi-Shneiderman	174,97	10,972	30
	Flujo	185,10	11,978	30
	Total	180,03	12,482	60
PostEfici2	Nassi-Shneiderman	75,0667	6,91306	30
	Flujo	67,9444	7,30734	30
	Total	71,5056	7,91408	60
PostTiempo3	Nassi-Shneiderman	1114,77	181,345	30
	Flujo	1362,70	171,784	30
	Total	1238,73	215,168	60
PostEfici3	Nassi-Shneiderman	57,2333	7,55828	30
	Flujo	50,3444	9,14652	30
	Total	53,7889	9,01473	60

En la tabla 28 se encuentra los resultados de la Prueba box sobre la igualdad de las matrices de covarianzas aceptando la hipótesis nula de que las matrices de covarianza observadas de las variables dependientes son iguales en todos los grupos (Diseño: Intersección + grupo

Diseño intra-sujetos: momentos + ejercicios + variables + momentos * ejercicios + momentos * variables + ejercicios * variables + momentos * ejercicios * variables), ya que, el nivel de significancia es 0.103 mayor que 0.05 ($p > 0.05$).

Tabla 28 Prueba de Box

M de Box	121,048
F	1,208
gl1	78
gl2	10622,951
Sig.	,103

La tabla 29 contiene los resultados de cuatro estadísticos multivariados: la tasa de Pillai, la lambda de Wilks, la tasa de Hotelling y la raíz mayor de Roy. Puesto que el nivel crítico (Sig.) asociado a cada uno de ellos es menor que 0.05 ($p < 0.05$), se rechaza la hipótesis nula de igualdad de medias y concluir que la eficiencia no es la misma en los dos momentos, ni en los 3 tipos de ejercicios para los tipos de eficiencia evaluados.

Tabla 29 Contrastes Multivaridos

Efecto		Valor	F	GI de la hipótesis	GI del error	Sig.	Eta al cuadrado parcial	Parámetro de no centralidad	Potencia observada ^c
momentos * ejercicios	Traza de Pillai	,967	839,767 ^b	2,000	57,000	,000	,967	1679,534	1,000
	Lambda de Wilks	,033	839,767 ^b	2,000	57,000	,000	,967	1679,534	1,000
	Traza de Hotelling	29,466	839,767 ^b	2,000	57,000	,000	,967	1679,534	1,000
	Raíz mayor de Roy	29,466	839,767 ^b	2,000	57,000	,000	,967	1679,534	1,000
	Traza de Pillai	,315	13,089 ^b	2,000	57,000	,000	,315	26,178	,996
	Lambda de Wilks	,685	13,089 ^b	2,000	57,000	,000	,315	26,178	,996
momentos * ejercicios * grupo	Traza de Hotelling	,459	13,089 ^b	2,000	57,000	,000	,315	26,178	,996
	Raíz mayor de Roy	,459	13,089 ^b	2,000	57,000	,000	,315	26,178	,996
	Traza de Pillai	,299	12,181 ^b	2,000	57,000	,000	,299	24,362	,994
momentos * ejercicios * variables * grupo	Lambda de Wilks	,701	12,181 ^b	2,000	57,000	,000	,299	24,362	,994
	Traza de Hotelling	,427	12,181 ^b	2,000	57,000	,000	,299	24,362	,994
	Raíz mayor de Roy	,427	12,181 ^b	2,000	57,000	,000	,299	24,362	,994

En el modelo de medidas repetidas es necesario suponer que las varianzas de las diferencias de los niveles son iguales, se toma el modelo de esfericidad de Mauchly. Puesto que el nivel crítico asociado al estadístico W (sig. =0.00) es menor que 0.05, se rechaza la hipótesis de esfericidad. Basamos la decisión en los estadísticos multivariados llamado épsilon, pueden ser el de Greenhouse-Geisser (1959) y el de Huynh-Feldt- (1976) siendo la primera más conservadora. Observando los resultados de la tabla, vemos que el nivel crítico es menor a 0,05, rechazando la hipótesis de igualdad de medias y concluir que entre los dos momentos (pretest y posttest) efectuados no es el mismo. Concluyendo que la intervención realizada sí da efectos en el proceso de desarrollo del pensamiento algorítmico.

Se elige el índice corrector Epsilon de Huynh-Feldt. Como se observa en la tabla 30

Prueba de esfericidad de Mauchly.

Tabla 30 Prueba de esfericidad de Mauchly

Efecto intra-sujetos	W de Mauchly	Chi-cuadrado aprox.	gl	Sig.	Epsilon ^b Greenhouse-Geisser	Epsilon Huynh-Feldt	Límite-inferior
momentos	1,000	,000	0	.	1,000	1,000	1,000
ejercicios	,573	31,726	2	,000	,701	,725	,500
variables	1,000	,000	0	.	1,000	1,000	1,000
momentos *	,397	52,706	2	,000	,624	,642	,500
ejercicios							
momentos *	1,000	,000	0	.	1,000	1,000	1,000
variables							
ejercicios *	,670	22,810	2	,000	,752	,781	,500
variables							
momentos *	,442	46,511	2	,000	,642	,661	,500
ejercicios *							
variables							

Observando los resultados de la tabla 31 Pruebas de efectos intra-sujetos. Se observa las cuatro versiones del estadístico F (corregidas y no corregidas) conducen a la misma conclusión, que a su vez coincide con la ya alcanzada utilizando la aproximación multivariada: puesto que el

nivel crítico (Sig=0.00.) es menor que 0.05 ($p < 0.05$), se puede rechazar la hipótesis de igualdad de medias y concluir que la eficiencia no es la misma en los dos momentos.

Tabla 31 Pruebas de efectos intra-sujetos.

Origen		Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.	Eta al cuadrado parcial	Parámetro de no centralidad	Potencia observada ^a
momentos * ejercicios	Esfericidad asumida	12298345,662	2	6149172,831	1507,105	,000	,963	3014,210	1,000
	Greenhouse-Geisser	12298345,662	1,247	9859178,826	1507,105	,000	,963	1879,964	1,000
	Huynh-Feldt	12298345,662	1,284	9581609,124	1507,105	,000	,963	1934,424	1,000
	Límite-inferior	12298345,662	1,000	12298345,662	1507,105	,000	,963	1507,105	1,000
momentos * ejercicios * grupo	Esfericidad asumida	116099,446	2	58049,723	14,227	,000	,197	28,455	,998
	Greenhouse-Geisser	116099,446	1,247	93073,104	14,227	,000	,197	17,747	,982
	Huynh-Feldt	116099,446	1,284	90452,777	14,227	,000	,197	18,261	,984
	Límite-inferior	116099,446	1,000	116099,446	14,227	,000	,197	14,227	,960
Error(momentos * ejercicios)	Esfericidad asumida	473294,197	116	4080,122					
	Greenhouse-Geisser	473294,197	72,349	6541,800					
	Huynh-Feldt	473294,197	74,445	6357,625					
	Límite-inferior	473294,197	58,000	8160,245					
momentos * ejercicios * variables * grupo	Esfericidad asumida	118457,076	2	59228,538	12,058	,000	,172	24,116	,995
	Greenhouse-Geisser	118457,076	1,284	92265,873	12,058	,000	,172	15,481	,964
	Huynh-Feldt	118457,076	1,323	89540,829	12,058	,000	,172	15,952	,968
	Límite-inferior	118457,076	1,000	118457,076	12,058	,000	,172	12,058	,927

Existe una interacción significativa de los factores momentos x ejercicios en la eficiencia $F_{(2, 58)} = 1507.10$, $p = 0.00$ y $\eta^2 = 0.963$. El comportamiento de la eficiencia en la solución de problemas no es el mismo en los ejercicios según el momento en el que se ha aplicado.

Existe una interacción significativa de los factores momentos x ejercicios x grupo en la eficiencia $F_{(2, 58)} = 12.058$, $p = 0.000$ y $\eta^2 = 0.197$. El comportamiento de la eficiencia en la solución de problemas no es el mismo en los ejercicios según el momento en el que se ha aplicado y

además no se comporta tampoco de la misma forma en Nassi-Sneiderman que en Diagramas de Flujo.

Existe una interacción significativa de los factores momentos x ejercicios x variable x grupo en la eficiencia $F_{(2, 58)} = 1507.10$, $p = 0.001$ y $\eta^2 = 0.172$. El comportamiento de la eficiencia con base en el tiempo y en el número de eventos en la solución de problemas no es el mismo en los ejercicios según el momento en el que se ha aplicado y además no se comporta tampoco de la misma forma en Nassi-Sneiderman que en Diagramas de Flujo.

La tabla 32 muestra que existe un efecto principal significativo del factor grupo en el desarrollo del pensamiento algorítmico $F_{(1,58)} = 8.14$, $p = 0.006$ y $\eta^2 = 0.123$.

Tabla 32 Pruebas de los efectos inter-sujetos

Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.	Eta al cuadrado parcial
Intersección	29478376,021	1	29478376,021	9384,309	,000	,994
Grupo	25571,090	1	25571,090	8,140	,006	,123
Error	182191,982	58	3141,241			

En la tabla 33 se encuentran las estimaciones de la agrupación Momentos x Ejercicios x Eficiencia. En la tabla 34 se encuentran los resultados de las comparaciones por pares de Momentos (Pretest vs Postest) en los tipos de ejercicios y eficiencia sin tener en cuenta el efecto de los grupos.

Tabla 33 Estimaciones Momentos x Ejercicios x Eficiencia

Momentos	Ejercicios	Eficiencia	Media	Error típ.	Intervalo de confianza 95%	
					Límite inferior	Límite superior
1	1	1	62,217	2,552	57,108	67,325
		2	84,046	1,525	80,994	87,098
	2	1	244,100	12,066	219,947	268,253
		2	81,731	1,452	78,824	84,638
	3	1	159,333	7,466	144,388	174,279
		2	31,463	2,544	26,370	36,556
2	1	1	133,733	1,683	130,365	137,101
		2	87,417	1,097	85,220	89,613
	2	1	180,033	1,483	177,065	183,002
		2	71,506	,918	69,667	73,344
	3	1	1238,733	22,803	1193,089	1284,378
		2	53,789	1,083	51,621	55,957

Al realizar la lectura de los resultados se concluye que en el ejercicio 1 no existe diferencia significativa cuando se evalúa la eficiencia por número de eventos pero sí existe diferencias en la eficiencia con base en el tiempo.

En el ejercicio 2 la eficiencia con base en el tiempo y la eficiencia con base en el número de eventos presenta diferencia estadísticamente significativa ($p < 0.05$), porque tienen un nivel de significancia de 0.00. Al comparar el momento 1 (Pretest) del ejercicio 2 con el momento 2 (Posttest) se observa que la eficiencia con base en el número de eventos en el momento 1 (Media=81.73, DT=1.45) y en el momento 2 (Media=71.50, DT= 0.918) Y la eficiencia con base en el tiempo en el momento 1 (Media =244.1 seg, DT= 12.06) y en el momento 2 (Media=180.033 seg, DT= 1.483)

En el ejercicio tipo 3 la eficiencia con base en el número de eventos en el momento 1 tenía una media de 31.46 con una desviación típica de 2.54 y en el momento 2 tiene una media de 53.789 con una desviación típica de 1.083. Y para la eficiencia con base en el tiempo se observa un aumento en las medias, momento 1 (media=159.33, DT=7.466) y en el momento 2 (media=1238.733, DT=22.803).

Tabla 34 Comparaciones por pares de Momentos (Pretest vs Posttest) en Ejercicios y eficiencia

Ejercicios	Eficiencia	(I)momentos	(J)momentos	Diferencia de medias (I-J)	Error típ.	Sig. ^b	Intervalo de confianza al 95 % para la diferencia	
							Límite inferior	Límite superior
1	1	1	2	-71,517*	3,211	,000	-77,943*	-65,090
		2	1	71,517*	3,211	,000	65,090*	77,943
	2	1	2	-3,371	1,727	,056	-6,828	,086
		2	1	3,371	1,727	,056	-,086	6,828
2	1	1	2	64,067*	12,072	,000	39,902*	88,231
		2	1	-64,067*	12,072	,000	-88,231*	-39,902
	2	1	2	10,225*	1,610	,000	7,003*	13,448
		2	1	-10,225*	1,610	,000	-13,448*	-7,003
3	1	1	2	-1079,400*	26,884	,000	-1133,214*	-1025,586
		2	1	1079,400*	26,884	,000	1025,586*	1133,214
	2	1	2	-22,326*	2,826	,000	-27,982*	-16,670
		2	1	22,326*	2,826	,000	16,670*	27,982

La tabla 35 se encuentran las estimaciones de la agrupación Grupos x Momentos x Ejercicios x Eficiencia. La tabla 36 se encuentran los resultados de las comparaciones por pares de Momentos (Pretest vs Postest) en Grupos, Ejercicios y eficiencia.

Tabla 35 Estimaciones Grupos x Momentos x Ejercicios x Eficiencia

Grupo	momentos	ejercicios	eficiencia	Media	Error típ.	Intervalo de confianza 95%	
						Límite inferior	Límite superior
Nassi-Shneiderman	1	1	1	69,900	3,609	62,675	77,125
			2	84,431	2,156	80,115	88,747
		2	1	282,967	17,064	248,809	317,125
			2	78,681	2,054	74,569	82,792
		3	1	168,800	10,559	147,664	189,936
			2	29,553	3,598	22,351	36,756
	2	1	1	128,333	2,380	123,570	133,096
			2	91,889	1,552	88,782	94,996
		2	1	174,967	2,097	170,769	179,164
			2	75,067	1,299	72,467	77,666
		3	1	1114,767	32,248	1050,215	1179,318
			2	57,233	1,532	54,167	60,300
Flujo	1	1	1	54,533	3,609	47,309	61,758
			2	83,661	2,156	79,345	87,977
		2	1	205,233	17,064	171,075	239,391
			2	84,781	2,054	80,669	88,893
		3	1	149,867	10,559	128,731	171,002
			2	33,373	3,598	26,170	40,576
	2	1	1	139,133	2,380	134,370	143,896
			2	82,944	1,552	79,838	86,051
		2	1	185,100	2,097	180,902	189,298
			2	67,944	1,299	65,345	70,544
		3	1	1362,700	32,248	1298,149	1427,251
			2	50,344	1,532	47,278	53,411

Al tener presente el efecto del factor grupo se observa que para el grupo Nassi-Shneiderman no existen diferencias estadísticamente significativas en el ejercicio 2 evaluando la eficiencia con base en el número de eventos, ya que su nivel de significancia es de 0.118. En los ejercicios 2 y 3, en las eficiencias con base en el tiempo y con base en el número de eventos existe diferencias significativas ($p > 0.05$).

Para el grupo Diagramas de Flujo en el ejercicio 1 cuando se evalúa la eficiencia por número de eventos y en el ejercicio 2 cuando se evalúa la eficiencia con base en el tiempo no existe diferencia estadísticamente significativa. Para los otros casos si existe diferencia estadísticamente significativa ($p < 0.05$).

Tabla 36 Comparaciones por pares de Momentos (Pretest vs Postest) en Grupos, Ejercicios y eficiencia

Grupo	ejercicios	eficiencia	(I)momentos	(J)momentos	Diferencia de medias (I-J)	Error tip.	Sig.	Intervalo de confianza al 95 % para la diferencia Límite inferior	Intervalo de confianza al 95 % para la diferencia Límite superior	
Nassi-Shneiderman	1	1	1	2	-58,433 [*]	4,541 [*]	,000	-67,522 [*]	-49,344 [*]	
			2	1	58,433 [*]	4,541 [*]	,000	49,344 [*]	67,522 [*]	
		2	1	1	2	-7,458 [*]	2,442 [*]	,003	-12,347 [*]	-2,569 [*]
				2	1	7,458 [*]	2,442 [*]	,003	2,569 [*]	12,347 [*]
		2	2	1	2	108,000 [*]	17,072 [*]	,000	73,826 [*]	142,174 [*]
				2	1	-108,000 [*]	17,072 [*]	,000	-142,174 [*]	-73,826 [*]
	3	1	1	1	2	-3,614	2,277	,118	-9,944	8,172
				2	1	3,614	2,277	,118	8,172	-9,944
			2	1	-945,967 [*]	38,020 [*]	,000	-1022,071 [*]	-869,862 [*]	
		2	1	1	2	945,967 [*]	38,020 [*]	,000	1022,071 [*]	869,862 [*]
				2	1	-27,680 [*]	3,996 [*]	,000	-35,679 [*]	-19,681 [*]
			2	1	27,680 [*]	3,996 [*]	,000	19,681 [*]	35,679 [*]	
Flujo	1	1	1	2	-84,600 [*]	4,541 [*]	,000	-93,689 [*]	-75,511 [*]	
			2	1	84,600 [*]	4,541 [*]	,000	75,511 [*]	93,689 [*]	
		2	1	1	2	,716	2,442	,770	-4,172	5,605
				2	1	-,716	2,442	,770	4,172	-5,605
		2	1	1	2	20,133	17,072	,243	-14,040	54,307
				2	1	-20,133	17,072	,243	14,040	-54,307
	3	1	1	1	2	16,837 [*]	2,277 [*]	,000	12,279 [*]	21,394 [*]
				2	1	-16,837 [*]	2,277 [*]	,000	-12,279 [*]	21,394 [*]
			2	1	-1212,833 [*]	38,020 [*]	,000	-1288,938 [*]	-1136,729 [*]	
		2	1	1	2	1212,833 [*]	38,020 [*]	,000	1288,938 [*]	1136,729 [*]
				2	1	-16,971 [*]	3,996 [*]	,000	-24,970 [*]	-8,973 [*]
			2	1	16,971 [*]	3,996 [*]	,000	8,973 [*]	24,970 [*]	

8 Discusión de los resultados

El propósito de esta investigación fue comparar el grado de efectividad de dos técnicas de diagramación para el desarrollo de las competencias del pensamiento algorítmico, para lograr el objetivo, se evaluó la eficiencia de estas dos técnicas (diagramas N-S y diagramas de flujo) basados en el número de eventos (pasos o líneas de código usadas) y también del tiempo utilizado en la resolución de los problemas. De la misma manera, se analizó esta eficiencia de acuerdo a tres tipos de problemas planteados separadamente (Tipo I: Problemas secuenciales, Tipo II: Problemas de iteraciones y Tipo III: Problemas de toma de decisiones).

En este apartado se presentan y discuten los resultados obtenidos a la luz del marco teórico.

8.1 Problemas tipo I: Problemas secuenciales

Para determinar cuál de las dos técnicas utilizadas en el proceso para la adquisición de las competencias del pensamiento algorítmico se utilizó la prueba ANOVA comparando las medias de los dos grupos. El resultado de esta prueba concluyó que al evaluar la eficiencia con base en el número de eventos para este tipo de problemas no hay diferencia significativa, como se observa en la tabla 16 ($p=0.845$ y $p>0.05$), en la utilización de alguna de las dos técnicas. Este resultado apoya la idea de Ortegata (2010), quien compara los diagramas de flujo con los diagramas de Chapin (Diagramas de Nassi-Shneiderman) al representar problemas al ingresar y extraer información en arreglos lineales

Estos resultados difieren de los obtenidos por Pérez Pérez, Fuentes Gálvez y Moreno Gutiérrez (2008), quienes concluyen que el conocimiento y aplicación adecuada de las técnicas de representación de algoritmos, no parece ayudar totalmente a resolver el problema que

presentan los estudiantes, de su deficiente capacidad para desarrollar algoritmos que reflejen correctamente las soluciones solicitadas; ya que se observa que el uso de los diagramas N-S mejoran la eficiencia con base en el tiempo de solución de problemas secuenciales, tabla 14.

Pero estamos de acuerdo con Pérez Pérez, Fuentes Gálvez y Moreno Gutiérrez (2008) en la sugerencia de buscar mejoras o nuevas propuestas para las técnicas de representación con el objetivo de reformar el proceso del desarrollo de algoritmos por computadora y mejorar los tiempos en la solución de problemas.

A este respecto, el conocimiento y aplicación adecuada de las técnicas de representación de algoritmos, no parece ayudar sustancialmente a resolver el problema que presentan los estudiantes, sobre su deficiente capacidad para desarrollar algoritmos que reflejen correctamente las soluciones solicitadas. Hay que señalar que las técnicas de representación de algoritmos son sólo parte de la compleja estructura metodológica pendiente por definir y ajustar en futuras investigaciones (Pérez Pérez, Fuentes Gálvez y Moreno Gutiérrez, 2008)

8.2 Problemas tipo II: Problemas de iteración

La prueba ANOVA de un factor tanto para la eficiencia con base en el tiempo como la eficiencia con base en el número de eventos para los problemas tipo II, cuyos resultados se presentan en la tabla 18 y 20, evalúa de manera estadística la hipótesis de existencia de diferencias significativas entre la solución de problemas de iteración. En ellos se puede observar que entre los estudiantes a los cuales se les aplicó el tratamiento de Nassi-Shneiderman presentan mejores resultados que los que fueron tratados con Diagramas de Flujo, porque el nivel de significancia en la eficiencia con base en el tiempo es de 0.001 y en el caso de la eficiencia con base en el número de eventos es 0.00. Estos resultados muestran diferencias significativas entre

los grupos, lo cual indica que los estudiantes que fueron tratados con Nassi-Shneiderman tienen una mejor eficiencia tanto en el tiempo de solución como en el uso de recursos para resolver problemas iterativos.

Los resultados para los problemas de iteración difieren de los encontrados por Pérez & Monzalvo (2012) quien determina que las técnicas de diagramación parecen no aportar beneficios sustanciales en el diseño de programas repetitivos, para nuestro caso las técnicas de diagramación sí influyen en la solución de problemas cíclicos. Adicional, Pérez & Monzalvo (2012) sugirieron realizar estudios sobre la problemática presente en el diseño de algoritmos cíclicos.

8.3 Problemas tipo III: Problemas de toma de decisiones

La prueba ANOVA de un factor tanto para la eficiencia con base en el tiempo como la eficiencia con base en el número de eventos para los problemas tipo III, cuyos resultados se presentan en la tabla 22 y 24, evalúa de manera estadística la hipótesis de existencia de diferencias significativas entre la solución de problemas de iteración. En ellos se puede observar que entre los estudiantes a los cuales se les aplicó el tratamiento de Nassi-Shneiderman presentan mejores resultados que los que fueron tratados con Diagramas de Flujo, porque el nivel de significancia en la eficiencia con base en el tiempo es de 0.00 y en el caso de la eficiencia con base en el número de eventos es 0.002. Estos resultados muestran diferencias significativas entre los grupos, lo cual indica que los estudiantes que fueron tratados con Nassi-Shneiderman tienen una mejor eficiencia tanto en el tiempo de solución como en el uso de recursos para resolver problemas en toma de decisiones.

Los resultados para los problemas de toma de decisiones complementan los resultados de Pérez & Monzalvo (2011) quien determina que los estudiantes utilizan en un 61% las técnicas de diagramación para resolver y plantear algoritmos que resuelvan problemas selectivos.

8.4 Análisis de Medidas Repetidas

Al revisar las estimaciones generadas, podemos notar que en las eficiencias (1 basado en el tiempo – 2 basado en el número de pasos) entre el pretest y el posttest, del ejercicio tipo I, y teniendo en cuenta que estos ejercicios tenían un grado de dificultad mucho más alto en la prueba de contraste se observa una mejoría en los procesos de pensamiento algorítmico para resolución de problemas, aunque, y el tiempo utilizado es inversamente proporcional a la calidad del ejercicio planteado. Es decir, al aumentar la dificultad del ejercicio, después de la intervención, la eficiencia de los estudiantes es más sólida, pero el tiempo empleado, en el análisis y planteamiento de la solución es más alto.

Siendo diferente en los ejercicios tipo II, ya que en ellos el nivel de eficiencia no aumenta, significativamente ni en cuanto al tiempo utilizado, como en la cantidad de pasos, esto nos lleva a concluir que para los test de repeticiones, o ciclos, los estudiantes no tienen muy bien estructurados los procesos para generar acciones de agrupación y repetición.

Para los problemas tipo III, el índice de eficiencia en cuanto a los pasos utilizados se incrementa de una 31,46% a una 53,78%, es decir, mejora; pero el tiempo también se incrementa significativamente.

A la luz del desarrollo del pensamiento algorítmico, concluimos que a mayor calidad del planteamiento y efectividad, el tiempo que los estudiantes objeto del intervención, también aumentó.

9 Conclusiones y recomendaciones

Se elabora la app “Prest”, diseñada para medir los niveles de eficiencia y tiempo utilizado en la solución de problemas como lo indica Román y Pérez, (2015) en el test diagnóstico de pensamiento computacional.

Al realizar la intervención a grupos con edades entre los 10 y 14 años, se debe presentar ambientes interactivos y llamativos en cuanto al diseño y dinamismo, para lograr que los estudiantes mantengan un alto grado de interés, participación y entusiasmo en la exploración de tareas y problemas para ser resueltos.

Los resultados obtenidos permitieron determinar que el uso de ambientes de trabajo dinámicos fue relevantes en esta investigación. La alta interactividad entre las herramientas utilizadas por estudiantes fue un factor clave en la construcción del conocimiento y de habilidades cognitivas demostradas en la resolución de los problemas durante la intervención.

En los problemas secuenciales (Tipo I) los resultados estadísticos validan la hipótesis nula, que enuncia que el desarrollo del pensamiento algorítmico no está influenciado s por la técnica de diagramación aplicada, cuando se evalúa la eficiencia en el número de eventos. Por otra parte, el tiempo empleado para la elaboración de la estrategia y solución de la tarea dada, en los problemas tipo I fue menor en los grupos que tuvieron un tratamiento con la técnica de diagramación Nassi-Shneiderman, incluyendo que la solución de un problema implica: la interpretación del problema, organización, el planteamiento de la solución y la ejecución.

Para los problemas tipo II y III los resultados estadísticos validan la hipótesis alterna, la cual plantea que los estudiantes que resuelven problemas en interacción con un ambiente de

aprendizaje que usa representación de conocimiento basada en diagramas de Nassi-Shneiderman se desempeña mejor, que quienes interactúan con el mismo ambiente pero con representación de conocimiento basado en diagramas de flujo, es decir, el pensamiento algorítmico, aunque implícito en los procesos de resolución de problema, también demuestran el cumplimiento de ésta hipótesis en cuanto a la eficiencia del proceso desarrollado.

Cabe resaltar como lo menciona Forsythe, Skeenan, Organick y Stenberg (1975) para resolver problemas: “La construcción de algoritmos y sus diagramas..... es esencialmente un proceso de solución de problemas. Para enseñar a resolver problemas, tenemos que proceder en forma muy diferente a la que estamos acostumbrados. No basta simplemente presentar el desarrollo directo de un elegante diagrama..... Debemos, por el contrario, mostrar cómo se llega a la solución final. Escoger el lugar para iniciar el problema. Se debe ver también algunos de los falsos comienzos y omisiones, algunos de los algoritmos deficientes obtenidos en nuestros primeros intentos. Sobre todo, usted debe saber que en la construcción de algoritmos primero intentamos obtener alguna clase de solución del problema (buena o mala), después se estudia y se critica la solución y se trata de encontrar formas de mejorarla.” Definitivamente, el conocimiento de cómo utilizar las herramientas de diagramación, no implica el conocimiento de cómo solucionar problemas (Farina, 1983).

Con base en los resultados de la investigación podemos determinar que la técnica adecuada para estimular el pensamiento algorítmico en los estudiantes de grado Sexto del Colegio Unión Europea IED es el diagrama de Nassi-Shneiderman cuando se aborde problemas

de iteración y problemas de toma de decisiones, por las mejoras que presenta en la eficiencia en el tiempo y en el número de eventos.

10 Referencias

- Alfonso Combita, D. J. (2010). Transferencia análogo de estrategias de solución de problemas de una forma de representación a un lenguaje de programación formal utilizando estructuras selectivas dobles. Bogotá: Universidad Pedagógica Nacional.
- Artopoulos, A. (2013). El docente traductor: claves para la integración de tecnología en la escuela. *Revista Linhas*, 59-82.
- Belloch, C. (2012) Las Tecnologías de la Información y Comunicación en el aprendizaje. Material docente [on-line]. Departamento de Métodos de Investigación y Diagnóstico en Educación. Universidad de Valencia.
- Blanco Nieto, L. (1993). Una clasificación de problemas matemáticos. *Epsilon: Revista de la Sociedad Andaluza de Educación Matemática "Thales"*, (25), 49-60.
- Cole, M., & Consortium, D. L. (2006). *Fifth Dimension, The: An After-School Program Built on Diversity*. Nueva York: Russell Sage Foundation.
- Dapozo, G. N., Greiner, C. L., Petris, R. H., & Espíndola, M. C. (2017, September). Promoción del pensamiento computacional para favorecer la formación de recursos humanos en disciplinas STEM. In XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires).
- Erro, L. (2016). Pensamiento computacional. [Online] Pensamiento Computacional Curso PC. Available at: <http://www.pensamientocomputacional.org/index.php/curso-pc> [Accessed 7 Mar. 2016]
- Farina, M. V. (1983). "Diagramas de flujo". Editorial Diana. Treceava impresión. México. ISBN: 968 -13-0863-8. pp. 15, 33, 56 y 57.

- Forsythe, A.; Skeenan, T.; Organick, E.; Stenberg, W. (1975). "Lenguajes de diagramas de flujo".
Editorial LIMUSA. Segunda reimpresión. México. pp. 66.
- Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science.
ISSEP, 159-168.
- Gómez Ceja, G. (1997). Sistemas administrativos: análisis y diseño. McGraw-Hill.
- Grover, S. (2009). Computer Science Is Not Just for Big Kids. Learning & Leading with
Technology, 27-30.
- Guerrero Posada, M., & García Orozco, J. (2016). DESARROLLO DEL PENSAMIENTO
ALGORÍTMICO CON EL APOYO DE OBJETOS DE APRENDIZAJE
GENERATIVOS. Pixel-Bit. Revista de Medios y Educación.
- Hitt, F. (s.f.). Dificultades en el aprendizaje del cálculo¹. Departamento de Matemática Educativa
del Cinvestav-IPN.
- Katai, Z. (2015). The challenge of promoting algorithmic thinking of both sciences- and
humanities-oriented learners. Journal of Computer Assisted Learning, 287-299.
- Lamagna, E. (2015). Algorithmic thinking unplugged. Journal of Computing Sciences in
Colleges, 45-52.
- Legris, J (2012). Visualizar y manipular. Sobre el razonamiento diagramático y la naturaleza de
la deducción. En Visualização nas Ciências Formais, comp. Por Abel Lassalle Cassanave
y Frank Thomas Sautter. Londres, College Publications, p. 89-103. ISBN 978-1-84890-
069-1. <http://www.collegepublications.co.uk/filosofia/>

- López García, J. C. (2009). Algoritmos y programación, guía para docentes (Segunda Edición). Cali: Eduteka.
- López García, J. C. (2014). Actividades De Aula Con Scratch Que Favorecen El Uso Del Pensamiento Algorítmico. . Calí.
- Martínez Avidad, M. (2012). Comunicación Para El Desarrollo Y La Inclusión Social De Minorías: La Clase Mágica: Un Modelo De Intervención Para El Cambio Social. Madrid: Universidad Complutense de Madrid.
- Morris, R. G., Ahmed, S., Syed, G. M., & Toone, B. K. (1993). Neural correlates of planning ability: frontal lobe activation during the Tower of London test. *Neuropsychologia*, 31(12), 1367-1378
- Nassi, I., & Shneiderman, B. (1973). Flowchart Techniques for structured programming. SIGPLAN NOTICES, 12-26.
- National Research Council. (1999). The Intellectual Framework of Fluency with Information Technology. En C. o. Literacy, Being Fluent with Information Technology (págs. 15-40). Washington, D.C.: National Academies Press.
- Nicolopoulou, A., & Cole, M. (1993). The Generation and Transmission of Shared Knowledge in the Culture of Collaborative Learning. En E. Forman , N. Minick , & C. Adisson Stone , Context for learning, Sociocultural dynamics in children's development (pág. 283). Nueva York: Oxford University Press.
- Olmos, K., Morales, C., Fernández, L., & Fernández, L. F. (s.f.). Objetos de Aprendizaje Enfocados a la Resolución de Problemas para Facilitar la Enseñanza de la Programación.

- Ortegate Becerra, J. (2010). *Los diagramas de flujo y de Chapin: Dos formas gráficas de representación de conocimiento en la solución de problemas*. Bogotá: Universidad Pedagógica Nacional.
- Perales Palacios, F. J. (1998). La resolución de problemas: una revisión estructurada. *Enseñanza de las Ciencias*, 11(2), 170-178.
- Pérez Pérez, I., Moreno Gutiérrez, S. S. (2009). “Estudio de la problemática relativa al uso de las sentencias secuenciales, selectivas e iterativas, en el diseño de algoritmos”. IV Congreso Universitario en Tecnologías de Información y Comunicaciones 2009. Área académica de Computación. ICBI- UAEH. México. ISBN: 978-607-482-058-4.
- Pérez Pérez, I., & Monzalvo López, C. A. (2010). Análisis del desempeño de los estudiantes en el diseño y construcción de algoritmos secuenciales, selectivos e iterativos. V Congreso Universitario en Tecnologías de Información y Comunicaciones.
- Pérez Pérez, I., & Monzalvo López, C. A. (2011). Nivel de utilización de las técnicas de diagramación por parte de los estudiantes, en el diseño de algoritmos secuenciales, selectivos e iterativos. *Congreso Universitario en Tecnologías de Información y Comunicaciones 2011*.
- Pérez Pérez, I. (2012). Análisis del nivel de utilización de las herramientas de diagramación manejadas dentro del diseño de algoritmos.
- Pérez Pérez, I. & Monzalvo López, C. A. (2012). Diseño de una propuesta de herramienta de diagramación para el desarrollo de algoritmos para computadora. VII Congreso

Universitario en Tecnologías de Información y Comunicaciones 2012. Área Académica de Computación. ICBI-UAEH. México. ISBN: En tramite

Prado, A., & Lamas, N. (2014). Alternativas para la enseñanza de pseudocódigo y diagrama de flujo. *Revista Electrónica Iberoamericana de Educación en Ciencias y Tecnología*, 102-113.

Román González, M., Pérez González, J. C., & Jiménez Fernández, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general. *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)*.

Sánchez Román, G., Guerrero García, J., Collazos Ordóñez, C. A., Tapia Cortes, C., & Mocencahua Mora, D. (2016). Propuesta de arquitectura de Sistema Tutor Inteligente para desarrollar las habilidades algorítmicas. *Revista Iberoamericana de Informática Educativa*, 30-35.

Shneiderman, B. (27 de 05 de 2003). *A short history of structured flowcharts (Nassi-Shneiderman Diagrams)*. Obtenido de <https://www.cs.umd.edu/hcil/members/bshneiderman/nsd/>

Stahl, G. (Septiembre de 2010). *Global Introduction to CSCL*. Obtenido de gerrystahl: <http://GerryStahl.net>

Talavera Pleguezuelos, C. (1999). *Calidad Total en la Administración Pública. Granada: Unión Iberoamericana de Municipalistas*.



Vidal, C., Cabezas, C., Parra, J., & López, L. (2015). Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile. *Formación universitaria*, 23-32.

- Villalobos Salcedo, J. A. (2009). Cupi2 Una solución integral al problema de Aprender/enseñar a Programar. *XXVII Reunión Nacional de Facultades de Ingeniería y VI Encuentro Iberoamericano de Instituciones de Enseñanza de la Ingeniería*, 1-37.
- Wing, J. (2006). Computational Thinking. *Communications of the ACM* 49, 33-35.
- Yoder, C. M., & Schrag, M. L. (1978). Nassi-Schneiderman Charts An Alternative To Flowcharts For Design. *IBM Corporation System Products Division Endicott, Nueva York 13760*, 386-393.


Anexos

Anexo 1 Screens de la App Prest






← → ↑ ↓
SIG

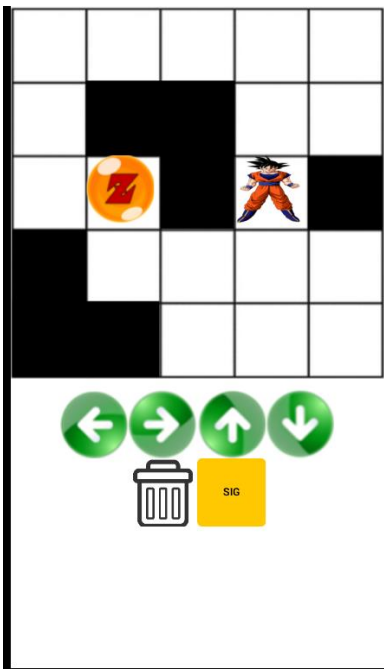
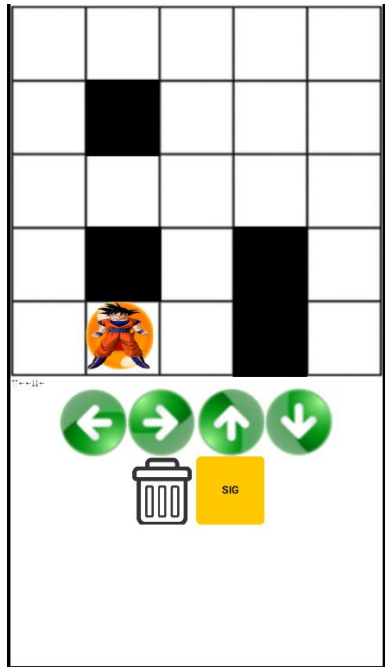
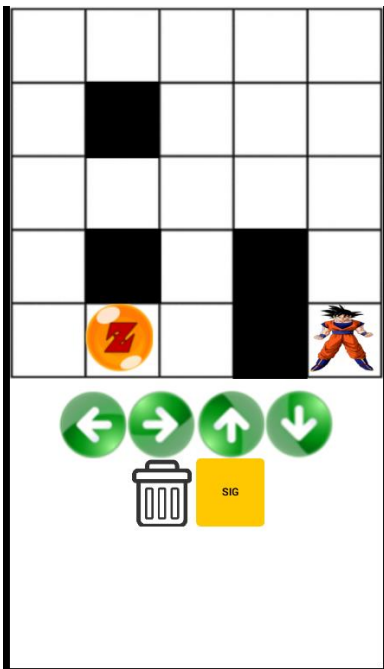
← → ↑ ↓
SIG

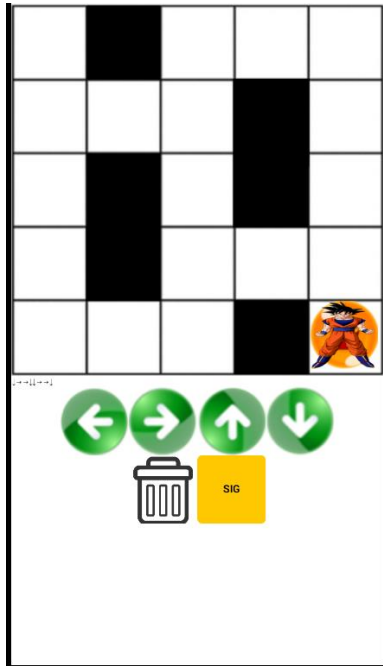
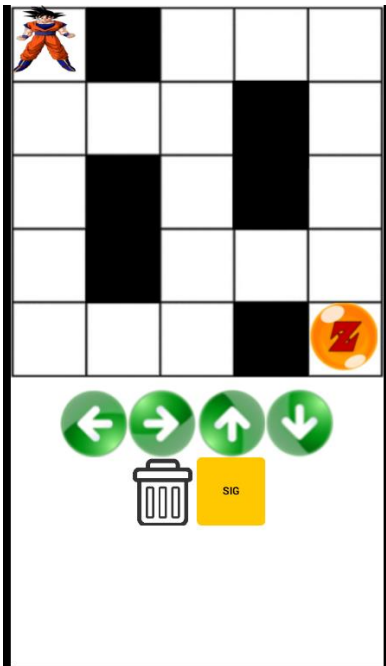
				
				

← → ↑ ↓
SIG

← → ↑ ↓
SIG















PREST NIVEL 2

En este nivel hemos puesto un botón adicional (la esfera del dragón). Este botón le servirá a Goku para levantar la esfera del dragón. En el cuadro de comando aparecerá una Z que identifique la esfera del dragón que ha recogido. El objetivo de este nivel es recoger las esferas del dragón que se encuentran en el tablero.


04-37



← → ↑ ↓

  SIG.







04-37



← → ↑ ↓

  SIG.


04-10



← → ↑ ↓

  SIG.














04-10





				




← → ↑ ↓


  SIG.





04 12




				
				
				
				
				











  SIG. 





				




   

  SIG. 


04 13





				
				
				
				
				




   

  SIG. 



















04 14



   

  SIG. 


04.63



← → ↑ ↓

  SIG.

04.64


← → ↑ ↓

  SIG.

PREST NIVEL 3

Usted deberá seleccionar en la pantalla el punto donde goku debe dejar las esteras a partir de la secuencia que aparece arriba del tablero.


→ → Z → ↓ ↓ ←

INICIAR




Ayuda a Goku a descubrir cuales esferas debe atrapar, sigue las instrucciones:

→ ↓ Z → ↓ → ↓ → ↑ Z


SIG

→ ↓ Z → ↓ → ↓ → ↑ Z

SIG





→ → Z → ↓ Z → ↓ Z

se han guardado los datos con éxito


SIG

→ → Z → ↓ Z → ↓ Z






SIG

→↓↑Z→↓Z→↓→↑Z←↓↑Z


SIG

→↓↑Z→↓Z→↓→↑Z←↓↑Z









SIG

→Z→Z↓Z→Z→Z↓Z↓Z


SIG

→Z→Z↓Z→Z→Z↓Z↓Z








SIG

↓Z → Z → Z↓Z ← Z↓ → → ↓ ← Z

SIG

↓Z → Z → Z↓Z ← Z↓ → → ↓ ← Z

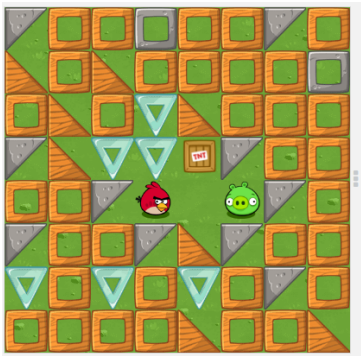
				
				
				
				

SIG

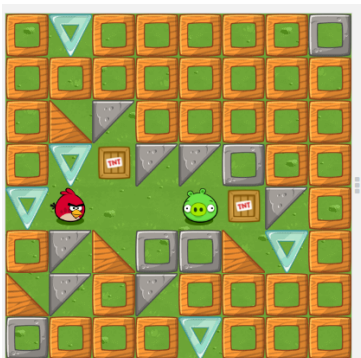
SALIR

Anexo 2 Screens Postest

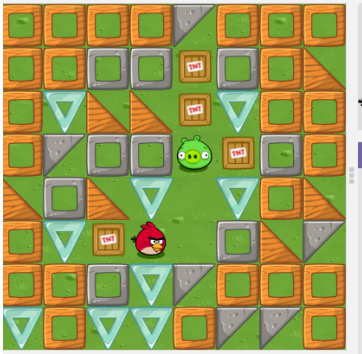
Ejercicio 1



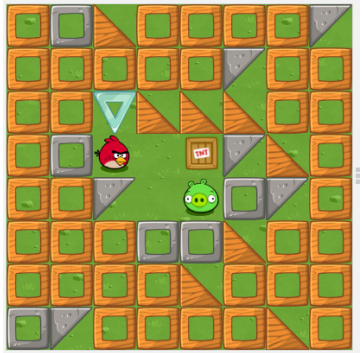
Ejercicio 2



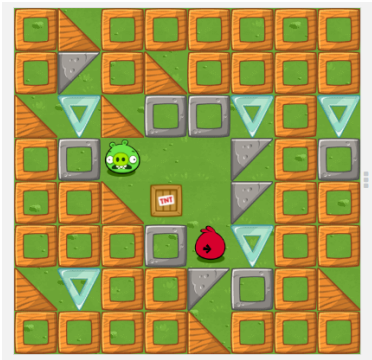
Ejercicio 3



Ejercicio4



Ejercicio 5



Ejercicio 6



Ejercicio 7



Ejercicio 8



Ejercicio 9



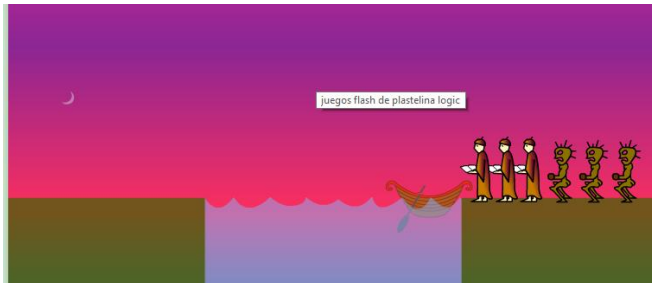
Ejercicio 10



Ejercicio 11



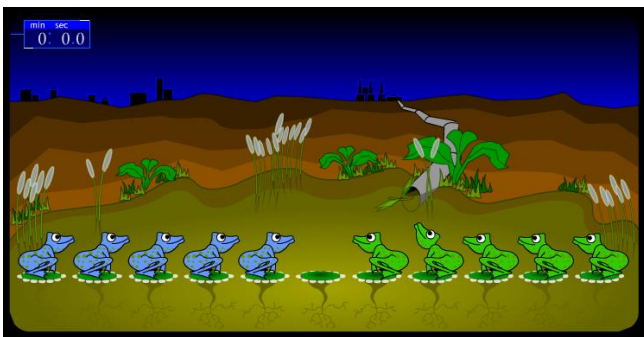
Ejercicio 12



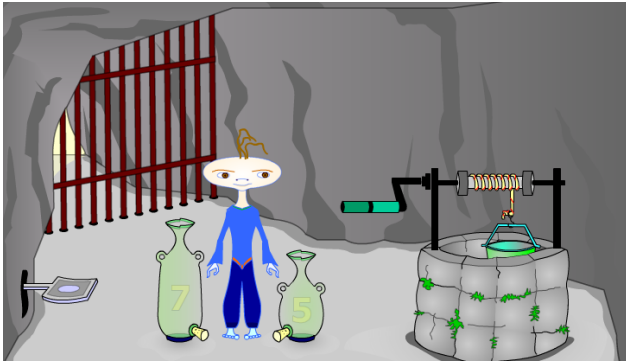
Ejercicio 13



Ejercicio 14



Ejercicio 15



Anexo 3 Resultados Pretest y Postest archivos en SPSS

Pretest

	usuario	grupo	tiempo1	efic1	efic...	tiempo2	efic2	efica2	tiem...	efic3	efica3	tiempo4	efic4	efica4	tiempo5	efic5	efica5	tiempo6	efic6	efica6	tiempo7	efic7
1	13	1	12	100,000	100	19	60,000	100	17	87,500	100	10	40,000	100	14	72,727	100	54	40,426	100	30	71,429
2	14	1	10	100,000	100	5	100,000	100	8	100,000	100	6	100,000	100	13	80,000	100	54	40,426	100	17	71,429
3	15	1	3	80,000	100	7	66,667	100	9	63,636	100	8	100,000	100	12	88,889	100	54	40,426	100	25	71,429
4	17	1	6	100,000	100	12	75,000	100	11	87,500	100	6	100,000	100	18	50,000	100	54	40,426	100	25	71,429
5	20	1	28	100,000	100	12	100,000	100	17	77,778	100	7	100,000	100	21	100,000	100	33	100,000	100	24	100,000
6	21	1	12	80,000	100	14	85,714	100	18	43,750	100	11	66,667	100	21	100,000	100	51	86,364	100	30	88,235
7	26	1	14	80,000	100	14	85,714	100	11	100,000	100	6	100,000	100	18	100,000	100	44	95,000	100	20	100,000
8	27	1	20	100,000	100	14	85,714	100	26	100,000	100	15	66,667	100	24	100,000	100	44	95,000	100	27	93,750
9	30	1	11	100,000	100	13	50,000	100	56	58,333	100	15	66,667	100	15	66,667	100	44	95,000	100	44	65,217
10	31	1	9	100,000	100	18	60,000	100	12	77,778	100	17	26,667	100	14	61,538	100	67	51,351	100	50	51,724
11	39	1	7	100,000	100	10	60,000	100	14	58,333	100	14	50,000	100	23	40,000	100	67	51,351	100	50	51,724
12	40	1	9	100,000	100	10	100,000	100	13	100,000	100	8	100,000	100	12	100,000	100	67	51,351	100	35	88,235
13	40	1	9	100,000	100	10	100,000	100	13	100,000	100	8	100,000	100	12	100,000	100	67	51,351	100	35	88,235
14	42	1	16	40,000	100	9	100,000	100	13	43,750	100	9	66,667	100	22	44,444	100	34	82,609	100	25	83,333
15	43	1	9	100,000	100	14	100,000	100	10	87,500	100	5	100,000	100	35	66,667	100	28	86,364	100	23	100,000
16	44	1	8	100,000	100	13	100,000	100	12	100,000	100	9	100,000	100	15	100,000	100	34	95,000	100	52	71,429
17	46	1	17	100,000	100	27	85,714	100	16	100,000	100	16	44,444	100	44	32,000	100	34	95,000	100	52	71,429
18	48	1	10	100,000	100	20	100,000	100	24	70,000	100	13	100,000	100	21	80,000	100	63	90,476	100	55	44,118
19	50	1	8	100,000	100	17	100,000	100	10	100,000	100	6	100,000	100	10	100,000	100	111	31,667	100	55	44,118
20	55	1	8	66,667	100	5	100,000	100	8	100,000	100	8	100,000	100	13	88,889	100	111	31,667	100	34	50,000
21	56	1	5	100,000	100	26	100,000	100	15	100,000	100	10	100,000	100	24	61,538	100	111	31,667	100	109	38,462
22	57	1	30	40,000	100	15	85,714	100	13	77,778	100	11	50,000	100	15	57,143	100	111	31,667	100	109	38,462
23	59	1	11	66,667	100	11	85,714	100	12	70,000	100	7	100,000	100	15	88,889	100	68	55,882	100	49	60,000

	tiempo8	efic8	efica8	tiempo9	efic9	efica9	tiempo10	efic10	efica10	tiempo11	efic11	efica11	tiempo12	efic12	efica12	tiempo13	efic13	efica13	tiempo14	efic14	efica14
1	56	59,322	100	50	75,758	100	48	90,909	100	23	100,000	100,000	24	0,000	14,000	33	67,000	100,000	13	66,000	100,000
2	41	81,395	100	37	89,286	100	56	100,000	100	23	100,000	100,000	24	0,000	14,000	33	67,000	100,000	13	66,000	100,000
3	44	81,395	100	22	89,286	100	55	80,000	100	101	9,000	80,000	24	0,000	14,000	94	5,000	15,000	21	13,000	33,000
4	44	81,395	100	22	89,286	100	55	80,000	100	101	9,000	80,000	19	5,000	67,000	56	4,000	12,000	11	58,000	8,000
5	54	100,000	100	67	86,207	100	68	88,889	100	168	7,000	11,000	12	0,000	100,000	20	6,000	25,000	13	39,000	100,000
6	56	94,595	100	44	80,645	100	72	72,727	100	39	31,000	80,000	54	5,000	12,000	47	8,000	100,000	33	32,000	8,000
7	53	85,366	100	34	92,593	100	46	95,238	100	49	8,000	80,000	18	6,000	100,000	46	2,000	100,000	18	53,000	100,000
8	53	85,366	100	34	92,593	100	46	95,238	100	13	25,000	20,000	13	29,000	17,000	76	21,000	12,000	19	26,000	33,000
9	53	85,366	100	34	92,593	100	46	95,238	100	39	10,000	80,000	19	12,000	100,000	47	14,000	12,000	19	26,000	33,000
10	56	62,500	100	83	52,083	100	52	83,333	100	81	3,000	17,000	4	15,000	0,000	59	56,000	62,000	36	29,000	33,000
11	56	62,500	100	83	52,083	100	52	83,333	100	81	3,000	17,000	51	0,000	14,000	59	56,000	62,000	29	23,000	25,000
12	43	100,000	100	42	86,207	100	120	83,333	100	36	20,000	100,000	12	77,000	100,000	31	14,000	100,000	19	42,000	100,000
13	43	100,000	100	42	86,207	100	120	83,333	100	36	20,000	100,000	12	77,000	100,000	31	14,000	100,000	19	42,000	100,000
14	46	89,744	100	36	83,333	100	45	100,000	100	35	38,000	17,000	23	62,000	67,000	30	72,000	100,000	23	53,000	100,000
15	42	100,000	100	31	92,593	100	38	100,000	100	22	20,000	17,000	54	2,000	14,000	65	3,000	100,000	42	31,000	100,000
16	71	77,778	100	47	71,429	100	70	76,923	100	22	20,000	17,000	27	2,000	14,000	35	4,000	22,000	72	27,000	8,000
17	71	77,778	100	291	41,667	100	70	76,923	100	14	16,000	20,000	20	21,000	14,000	42	0,000	25,000	20	61,000	42,000
18	96	87,500	100	73	65,789	100	119	81,633	100	19	67,000	60,000	22	100,000	100,000	85	56,000	50,000	105	30,000	8,000
19	96	87,500	100	73	65,789	100	119	81,633	100	10	56,000	20,000	11	47,000	17,000	41	9,000	100,000	30	13,000	100,000
20	31	89,744	100	63	73,529	100	97	66,667	100	31	11,000	100,000	18	27,000	100,000	35	14,000	100,000	14	47,000	79,000
21	112	58,333	100	63	73,529	100	97	66,667	100	31	89,000	80,000	13	43,000	29,000	62	50,000	22,000	34	29,000	25,000
22	112	58,333	100	63	73,529	100	97	66,667	100	3	22,000	0,000	13	43,000	29,000	13	7,000	22,000	34	29,000	25,000
23	102	61,404	100	62	60,976	100	130	66,667	100	3	22,000	0,000	13	43,000	29,000	13	7,000	22,000	34	29,000	25,000

	tiempo13	efic13	efica13	tiempo14	efic14	efica14	tiempo15	efic15	efica15	tiempoG1	tiempoG2	tiempoG3	efic1	efic2	efic3	Pefica1	Pefica2	Pefica3	Tiempo
1	33	67,000	100,000	13	66,000	100,000	29	52,000	46,000	72,000	238,000	122,000	72,05	67,57	57,00	100,00	100,00	72,00	432
2	33	67,000	100,000	13	66,000	100,000	29	52,000	46,000	42,000	205,000	122,000	96,00	76,51	57,00	100,00	100,00	72,00	369
3	94	5,000	15,000	21	13,000	33,000	36	13,000	17,000	39,000	200,000	276,000	79,84	72,51	8,00	100,00	100,00	31,80	515
4	56	4,000	12,000	11	58,000	8,000	38	20,000	31,000	53,000	200,000	225,000	82,50	72,51	19,20	100,00	100,00	39,60	478
5	20	6,000	25,000	13	39,000	100,000	22	30,000	37,000	85,000	246,000	235,000	95,56	95,02	16,40	100,00	100,00	54,60	566
6	47	8,000	100,000	33	32,000	8,000	44	47,000	73,000	76,000	253,000	217,000	75,23	84,51	24,60	100,00	100,00	54,60	546
7	46	2,000	100,000	18	53,000	100,000	34	21,000	82,000	63,000	197,000	165,000	93,14	93,64	18,00	100,00	100,00	92,40	425
8	76	21,000	12,000	19	26,000	33,000	34	21,000	82,000	99,000	200,000	155,000	90,48	92,39	24,40	100,00	100,00	32,80	454
9	47	14,000	12,000	19	26,000	33,000	34	21,000	82,000	110,000	221,000	158,000	68,33	86,68	16,60	100,00	100,00	61,40	489
10	59	56,000	62,000	36	29,000	33,000	79	32,000	7,000	70,000	308,000	259,000	65,20	60,20	27,00	100,00	100,00	23,80	637
11	59	56,000	62,000	29	23,000	25,000	26	19,000	7,000	68,000	308,000	246,000	61,67	60,20	20,20	100,00	100,00	25,00	622
12	31	14,000	100,000	19	42,000	100,000	18	100,000	82,000	52,000	307,000	116,000	100,00	81,83	50,60	100,00	100,00	96,40	475
13	31	14,000	100,000	19	42,000	100,000	18	100,000	82,000	52,000	307,000	116,000	100,00	81,83	50,60	100,00	100,00	96,40	475
14	30	72,000	100,000	23	53,000	100,000	24	79,000	82,000	69,000	186,000	135,000	58,97	87,80	60,80	100,00	100,00	73,20	390
15	65	3,000	100,000	42	31,000	100,000	37	49,000	46,000	73,000	162,000	220,000	90,83	95,79	21,00	100,00	100,00	55,40	455
16	35	4,000	22,000	72	27,000	8,000	44	13,000	23,000	57,000	274,000	200,000	100,00	78,51	13,20	100,00	100,00	16,80	531
17	42	,000	25,000	20	61,000	42,000	36	13,000	17,000	120,000	518,000	132,000	72,43	72,56	22,20	100,00	100,00	23,60	770
18	85	56,000	50,000	105	30,000	8,000	73	36,000	8,000	88,000	406,000	304,000	90,00	73,90	57,80	100,00	100,00	45,20	798
19	41	9,000	100,000	30	13,000	100,000	32	48,000	64,000	51,000	454,000	124,000	100,00	62,14	34,60	100,00	100,00	60,20	629
20	35	14,000	100,000	14	47,000	79,000	25	26,000	25,000	42,000	336,000	123,000	91,11	62,32	25,00	100,00	100,00	80,80	501
21	62	50,000	22,000	34	29,000	25,000	34	27,000	18,000	80,000	492,000	174,000	92,31	53,73	47,60	100,00	100,00	34,80	746
22	13	7,000	22,000	34	29,000	25,000	34	27,000	18,000	84,000	492,000	97,000	62,13	53,73	25,60	100,00	100,00	18,80	673
23	13	7,000	22,000	34	29,000	25,000	34	27,000	18,000	56,000	411,000	97,000	82,25	60,99	25,60	100,00	100,00	18,80	564

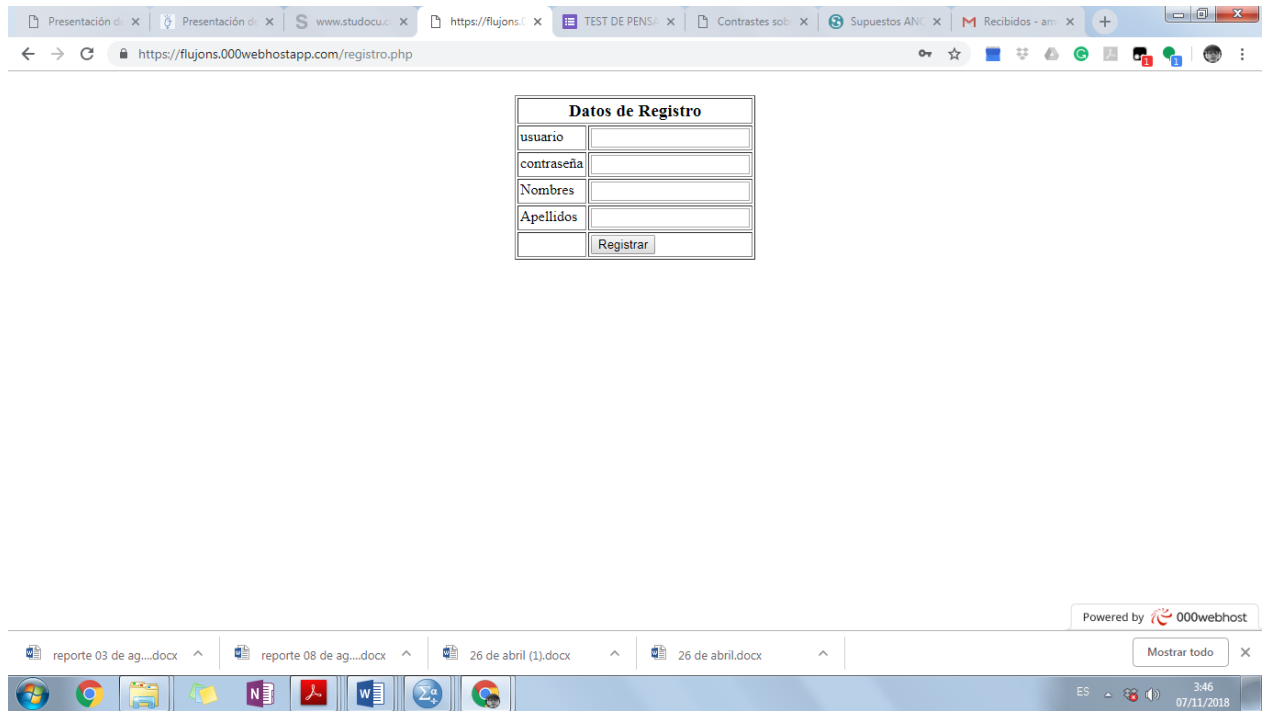
Postest

	nombres	CURSO	REP1	TIEMPO1	LINEAS1	REP2	TIEMPO2	LINEAS2	REP3	TIEMPO3	LINEAS3	REP4	TIEMPO4	LINEAS4	REP5	TIEMPO5	LINEAS5
1	Dilan Alejandro López	1	1,0	0:00:20.00	2,0	1,0	0:00:15.00	3,0	2,0	0:00:20.00	4,0	1,0	0:00:35.00	5,0	1,0	0:00:28.00	5
2	Yeiner Santiago Hernández	1	1,0	0:00:15.00	2,0	1,0	0:00:16.00	3,0	1,0	0:00:21.00	4,0	1,0	0:00:30.00	5,0	1,0	0:00:30.00	5
3	Julian alejandro Molina	1	1,0	0:00:18.00	2,0	1,0	0:00:18.00	3,0	1,0	0:00:25.00	4,0	2,0	0:00:45.00	5,0	1,0	0:00:29.00	5
4	sharith julieih Pulido	1	1,0	0:00:23.00	2,0	1,0	0:00:17.00	3,0	1,0	0:00:30.00	4,0	1,0	0:00:30.00	5,0	1,0	0:00:31.00	5
5	cristian Ascancio	1	1,0	0:00:16.00	2,0	1,0	0:00:15.00	3,0	1,0	0:00:24.00	4,0	1,0	0:00:34.00	5,0	1,0	0:00:30.00	5
6	samuel leandro Cumaco	1	1,0	0:00:15.00	2,0	1,0	0:00:16.00	3,0	1,0	0:00:18.00	4,0	1,0	0:00:35.00	5,0	1,0	0:00:32.00	5
7	fernanda López	1	1,0	0:00:18.00	2,0	1,0	0:00:18.00	3,0	1,0	0:00:19.00	4,0	1,0	0:00:36.00	5,0	1,0	0:00:30.00	5
8	ivan arturo Polania	1	1,0	0:00:16.00	2,0	1,0	0:00:16.00	3,0	1,0	0:00:16.00	4,0	1,0	0:00:38.00	5,0	1,0	0:00:30.00	5
9	Cristian Sánchez	1	1,0	0:00:18.00	2,0	1,0	0:00:15.00	3,0	1,0	0:00:17.00	4,0	2,0	0:00:45.00	5,0	1,0	0:00:31.00	5
10	luis manuel Sanguino	1	1,0	0:00:19.00	2,0	1,0	0:00:14.00	3,0	1,0	0:00:18.00	4,0	2,0	0:00:49.00	5,0	2,0	0:00:30.00	5
11	Jeimmy Avila	1	1,0	0:00:21.00	2,0	1,0	0:00:14.00	3,0	1,0	0:00:19.00	4,0	2,0	0:00:52.00	5,0	1,0	0:00:50.00	5
12	Julian Felipe Vanegas	1	1,0	0:00:14.00	2,0	1,0	0:00:16.00	3,0	1,0	0:00:17.00	4,0	1,0	0:00:35.00	5,0	1,0	0:00:27.00	5
13	marlon andrey Fuentes	1	1,0	0:00:15.00	2,0	1,0	0:00:15.00	3,0	1,0	0:00:17.00	4,0	1,0	0:00:34.00	5,0	1,0	0:00:29.00	5
14	Joseph David Rico	1	1,0	0:00:15.00	2,0	1,0	0:00:15.00	3,0	1,0	0:00:18.00	4,0	1,0	0:00:33.00	5,0	1,0	0:00:30.00	5
15	Samuel Leandro	1	1,0	0:00:16.00	2,0	1,0	0:00:17.00	3,0	1,0	0:00:18.00	4,0	1,0	0:00:30.00	5,0	2,0	0:00:40.00	5
16	Laura Valentina Romero	1	1,0	0:00:16.00	2,0	1,0	0:00:16.00	3,0	1,0	0:00:19.00	4,0	1,0	0:00:39.00	5,0	1,0	0:00:31.00	5
17	Miguel angel Hincapie	1	1,0	0:00:15.00	2,0	1,0	0:00:18.00	3,0	1,0	0:00:15.00	4,0	2,0	0:00:48.00	5,0	2,0	0:00:45.00	5
18	Yurany Pedraza	1	1,0	0:00:17.00	2,0	1,0	0:00:18.00	3,0	1,0	0:00:16.00	4,0	1,0	0:00:35.00	5,0	1,0	0:00:30.00	5
19	Juan Sebastian Galindo	1	1,0	0:00:18.00	2,0	1,0	0:00:16.00	3,0	1,0	0:00:17.00	4,0	1,0	0:00:36.00	5,0	1,0	0:00:32.00	5
20	Maria Camila Quintero	1	1,0	0:00:19.00	2,0	1,0	0:00:15.00	3,0	1,0	0:00:18.00	4,0	1,0	0:00:37.00	5,0	1,0	0:00:30.00	5
21	José David Ramírez	1	1,0	0:00:23.00	2,0	1,0	0:00:18.00	3,0	1,0	0:00:15.00	4,0	1,0	0:00:32.00	5,0	1,0	0:00:30.00	5
22	Laura natalia Beltrán	1	1,0	0:00:25.00	2,0	1,0	0:00:17.00	3,0	1,0	0:00:17.00	4,0	2,0	0:00:50.00	5,0	1,0	0:00:31.00	5
23	Diego Leonardo Lizarazo	1	1,0	0:00:22.00	2,0	1,0	0:00:19.00	3,0	2,0	0:00:19.00	4,0	2,0	0:00:49.00	5,0	2,0	0:00:45.00	5

	LINEAS5	REP6	TIEMPO6	LINEAS6	REP7	TIEMPO7	LINEA...	REP8	TIEMPO8	LINEAS8	REP9	TIEMPO9	LINEAS9	REP10	TIEMPO10	LINEAS10	REP11	TIEMPO11	f
1	5,0	1,0	0:00:23.00	2,0	1,0	0:00:30.00	5,0	2	0:00:30.00	3,0	1,0	0:00:35.00	8,0	1,0	0:00:45.00	8,0	1,0	0:00:45.00	
2	5,0	1,0	0:00:25.00	2,0	1,0	0:00:29.00	5,0	1	0:00:18.00	3,0	1,0	0:00:35.00	8,0	1,0	0:00:48.00	8,0	1,0	0:01:10.00	
3	5,0	1,0	0:00:23.00	2,0	1,0	0:00:27.00	5,0	1	0:00:20.00	3,0	1,0	0:00:36.00	8,0	1,0	0:00:50.00	8,0	1,0	0:00:52.00	
4	5,0	1,0	0:00:24.00	3,0	1,0	0:00:32.00	5,0	1	0:00:18.00	3,0	1,0	0:00:37.00	8,0	1,0	0:00:50.00	8,0	1,0	0:01:03.00	
5	5,0	1,0	0:00:23.00	2,0	1,0	0:00:30.00	5,0	1	0:00:19.00	3,0	2,0	0:00:42.00	8,0	1,0	0:00:49.00	8,0	1,0	0:00:50.00	
6	5,0	1,0	0:00:25.00	2,0	1,0	0:00:38.00	5,0	2	0:00:32.00	3,0	2,0	0:00:45.00	8,0	1,0	0:00:52.00	8,0	1,0	0:00:45.00	
7	5,0	1,0	0:00:23.00	2,0	1,0	0:00:36.00	5,0	2	0:00:34.00	3,0	2,0	0:00:48.00	8,0	1,0	0:00:56.00	8,0	1,0	0:00:59.00	
8	5,0	1,0	0:00:24.00	2,0	1,0	0:00:38.00	5,0	1	0:00:19.00	3,0	1,0	0:00:35.00	8,0	1,0	0:00:58.00	8,0	1,0	0:00:58.00	
9	5,0	1,0	0:00:21.00	2,0	1,0	0:00:37.00	5,0	1	0:00:20.00	3,0	1,0	0:00:34.00	8,0	1,0	0:00:58.00	8,0	1,0	0:01:06.00	
10	5,0	1,0	0:00:20.00	2,0	1,0	0:00:30.00	5,0	1	0:00:22.00	3,0	2,0	0:00:46.00	8,0	1,0	0:00:56.00	8,0	1,0	0:01:00.00	
11	5,0	1,0	0:00:25.00	3,0	2,0	0:00:40.00	5,0	1	0:00:21.00	3,0	2,0	0:00:48.00	8,0	1,0	0:00:54.00	8,0	1,0	0:00:50.00	
12	5,0	1,0	0:00:20.00	2,0	1,0	0:00:32.00	5,0	2	0:00:20.00	3,0	2,0	0:00:49.00	8,0	1,0	0:00:52.00	8,0	1,0	0:00:56.00	
13	5,0	1,0	0:00:23.00	2,0	1,0	0:00:25.00	5,0	2	0:00:32.00	3,0	2,0	0:00:52.00	8,0	1,0	0:00:57.00	8,0	1,0	0:00:58.00	
14	5,0	1,0	0:00:20.00	2,0	1,0	0:00:30.00	5,0	2	0:00:30.00	3,0	1,0	0:00:35.00	8,0	1,0	0:00:58.00	9,0	1,0	0:00:49.00	
15	5,0	1,0	0:00:21.00	3,0	1,0	0:00:32.00	5,0	2	0:00:31.00	3,0	1,0	0:00:35.00	8,0	1,0	0:00:58.00	8,0	1,0	0:00:56.00	
16	5,0	1,0	0:00:22.00	2,0	1,0	0:00:35.00	5,0	2	0:00:30.00	3,0	1,0	0:00:38.00	8,0	1,0	0:00:56.00	8,0	1,0	0:00:58.00	
17	5,0	1,0	0:00:21.00	2,0	1,0	0:00:30.00	5,0	2	0:00:30.00	3,0	1,0	0:00:34.00	8,0	1,0	0:00:50.00	9,0	2,0	0:01:15.00	
18	5,0	1,0	0:00:25.00	2,0	1,0	0:00:35.00	5,0	1	0:00:19.00	3,0	1,0	0:00:36.00	9,0	1,0	0:00:54.00	9,0	1,0	0:00:50.00	
19	5,0	1,0	0:00:23.00	2,0	2,0	0:00:31.00	5,0	1	0:00:17.00	3,0	2,0	0:00:50.00	8,0	1,0	0:00:52.00	8,0	1,0	0:00:56.00	
20	5,0	1,0	0:00:21.00	2,0	2,0	0:00:32.00	5,0	1	0:00:18.00	3,0	2,0	0:00:51.00	8,0	1,0	0:00:52.00	8,0	1,0	0:00:54.00	
21	5,0	1,0	0:00:20.00	2,0	1,0	0:00:30.00	5,0	1	0:00:18.00	3,0	2,0	0:00:46.00	8,0	1,0	0:00:54.00	8,0	1,0	0:00:58.00	
22	5,0	1,0	0:00:20.00	2,0	1,0	0:00:32.00	5,0	1	0:00:18.00	3,0	2,0	0:00:48.00	8,0	1,0	0:00:58.00	8,0	2,0	0:01:20.00	
23	5,0	1,0	0:00:23.00	2,0	2,0	0:00:35.00	5,0	2	0:00:37.00	3,0	2,0	0:00:45.00	8,0	1,0	0:00:57.00	8,0	1,0	0:01:00.00	

	0	TIEMPO10	LINEAS10	REP11	TIEMPO11	REP12	TIEMPO12	REP13	TIEMPO13	REP14	TIEMPO14	REP15	TIEMPO15	tiempobloque1	tiempobloque2	tiempobloque3	tiempototal	efic
1	1,0	0:00:45.00	8,0	1,0	0:00:45.00	1,0	0:01:40.00	3,0	0:05:44.00	2,0	0:03:26.00	2,0	0:05:15.00	0:01:58	0:02:43	0:16:50	0:21:31	
2	1,0	0:00:48.00	8,0	1,0	0:01:10.00	1,0	0:01:30.00	2,0	0:04:30.00	2,0	0:02:40.00	2,0	0:06:12.00	0:01:52	0:02:35	0:16:02	0:20:29	
3	1,0	0:00:50.00	8,0	1,0	0:00:52.00	1,0	0:01:25.00	3,0	0:06:16.00	2,0	0:01:56.00	3,0	0:06:10.00	0:02:15	0:02:36	0:16:39	0:21:30	
4	1,0	0:00:50.00	8,0	1,0	0:01:03.00	1,0	0:01:45.00	2,0	0:03:20.00	3,0	0:03:40.00	2,0	0:04:45.00	0:02:11	0:02:41	0:14:33	0:19:25	
5	1,0	0:00:49.00	8,0	1,0	0:00:50.00	1,0	0:01:55.00	4,0	0:05:16.00	2,0	0:02:10.00	2,0	0:04:30.00	0:01:59	0:02:43	0:14:41	0:19:23	
6	1,0	0:00:52.00	8,0	1,0	0:00:45.00	1,0	0:01:46.00	5,0	0:06:30.00	2,0	0:02:25.00	3,0	0:06:12.00	0:01:56	0:03:12	0:17:38	0:22:46	
7	1,0	0:00:56.00	8,0	1,0	0:00:59.00	1,0	0:01:30.00	2,0	0:04:32.00	2,0	0:02:34.00	2,0	0:05:25.00	0:02:01	0:03:17	0:15:00	0:20:18	
8	1,0	0:00:58.00	8,0	1,0	0:00:58.00	1,0	0:01:45.00	4,0	0:06:23.00	3,0	0:04:00.00	2,0	0:04:28.00	0:01:56	0:02:54	0:17:34	0:22:24	
9	1,0	0:00:58.00	8,0	1,0	0:01:06.00	2,0	0:02:05.00	3,0	0:05:40.00	3,0	0:03:40.00	2,0	0:05:18.00	0:02:06	0:02:50	0:17:49	0:22:45	
10	1,0	0:00:56.00	8,0	1,0	0:01:00.00	2,0	0:02:20.00	3,0	0:05:20.00	2,0	0:01:50.00	3,0	0:05:50.00	0:02:10	0:02:54	0:16:20	0:21:24	
11	1,0	0:00:54.00	8,0	1,0	0:00:50.00	2,0	0:02:50.00	3,0	0:04:50.00	2,0	0:02:35.00	2,0	0:05:46.00	0:02:36	0:03:08	0:16:51	0:22:35	
12	1,0	0:00:52.00	8,0	1,0	0:00:56.00	3,0	0:03:15.00	3,0	0:05:43.00	2,0	0:02:15.00	4,0	0:07:18.00	0:01:49	0:02:53	0:19:27	0:24:09	
13	1,0	0:00:57.00	8,0	1,0	0:00:58.00	1,0	0:01:58.00	4,0	0:04:56.00	2,0	0:02:30.00	2,0	0:05:46.00	0:01:50	0:03:09	0:16:08	0:21:07	
14	1,0	0:00:58.00	9,0	1,0	0:00:49.00	2,0	0:01:00.00	2,0	0:03:10.00	2,0	0:02:40.00	2,0	0:04:58.00	0:01:51	0:02:53	0:12:37	0:17:21	
15	1,0	0:00:58.00	8,0	1,0	0:00:56.00	3,0	0:04:10.00	4,0	0:06:20.00	3,0	0:03:36.00	2,0	0:05:26.00	0:02:01	0:02:57	0:20:28	0:25:26	
16	1,0	0:00:56.00	8,0	1,0	0:00:58.00	2,0	0:02:05.00	2,0	0:04:54.00	2,0	0:03:24.00	4,0	0:08:12.00	0:02:01	0:03:01	0:19:33	0:24:35	
17	1,0	0:00:50.00	9,0	2,0	0:01:15.00	1,0	0:01:36.00	3,0	0:06:34.00	2,0	0:03:35.00	2,0	0:06:30.00	0:02:21	0:02:45	0:19:30	0:24:36	
18	1,0	0:00:54.00	9,0	1,0	0:00:50.00	1,0	0:01:45.00	3,0	0:07:05.00	3,0	0:04:50.00	2,0	0:05:46.00	0:01:56	0:02:49	0:20:16	0:25:01	
19	1,0	0:00:52.00	8,0	1,0	0:00:56.00	1,0	0:01:54.00	3,0	0:05:37.00	2,0	0:03:45.00	3,0	0:06:42.00	0:01:59	0:02:53	0:18:54	0:23:46	
20	1,0	0:00:52.00	8,0	1,0	0:00:54.00	2,0	0:02:35.00	2,0	0:04:10.00	2,0	0:03:27.00	2,0	0:08:16.00	0:01:59	0:02:54	0:19:22	0:24:15	
21	1,0	0:00:54.00	8,0	1,0	0:00:58.00	4,0	0:06:24.00	2,0	0:03:50.00	3,0	0:04:14.00	2,0	0:06:18.00	0:01:58	0:02:48	0:21:44	0:26:30	
22	1,0	0:00:58.00	8,0	2,0	0:01:20.00	3,0	0:04:52.00	2,0	0:03:25.00	2,0	0:03:18.00	4,0	0:08:00.00	0:02:20	0:02:56	0:20:55	0:26:11	

Anexo 4 Registro de Usuarios la base de Datos



The screenshot displays a web browser window with the address bar showing the URL <https://flujons.000webhostapp.com/registro.php>. The browser's tab bar includes several open tabs, such as 'Presentación di...', 'www.studocu...', 'https://flujons...', 'TEST DE PENSA...', 'Contrastes sob...', 'Supuestos AN...', and 'Recibidos - am...'. The main content area features a registration form titled 'Datos de Registro' with the following fields:

Datos de Registro	
usuario	<input type="text"/>
contraseña	<input type="password"/>
Nombres	<input type="text"/>
Apellidos	<input type="text"/>
	<input type="button" value="Registrar"/>

At the bottom of the browser window, a 'Powered by 000webhost' logo is visible. Below the browser, a Windows taskbar shows several open document windows: 'reporte 03 de ag...docx', 'reporte 08 de ag...docx', '26 de abril (1).docx', and '26 de abril.docx'. The taskbar also displays the system tray with the date '07/11/2018' and time '3:46'.