

Scratch y WeDo como herramientas para desarrollar el pensamiento computacional en niños de 11 y 12 años

Cadillo León, Juan Raúl

Universidad Nacional Santiago Antúnez de Mayolo

I.E. 87003-1 “Jesús Nazareno” - Perú

Resumen

En el ámbito educativo peruano, se viene promoviendo la utilización de herramientas tecnológicas como Scratch y WEDO como medios que pueden posibilitar la mejora de calidad educativa, pero no se tienen definidos los tipos de pensamiento o capacidades de orden superior que se deben lograr en los estudiantes al emplear estos medios. Frente a ello se postula la presente investigación siguiendo la línea propuesta por investigadores del Instituto de Tecnología de Massachusetts (MIT) y del Carnegie Mellon University respecto al desarrollo del Pensamiento Computacional en los niños, buscando entender y adaptar esta propuesta de manera sistemática; familiarizando conceptos dentro de un fenómeno poco documentado, marcando posibles tendencias de estudio que se pueden llevar a cabo en nuestro ámbito educativo y señalando las potenciales relaciones entre la variable marco de trabajo con Scratch y WEDO y pensamiento computacional.

Se aborda el problema de cómo enseñar el pensamiento computacional a los niños de 11 y 12 años en las escuelas, para lo cual se han desarrollado un conjunto de tareas de aprendizaje basado en desafíos que alienta al estudiante a probar soluciones a través de la instrucción directa, la simulación y los videojuegos usando Scratch y WEDO.

Palabras clave:

Pensamiento computacional, Scratch, WeDo, marco de trabajo, algoritmos, TIC en la Educación.

Scratch and WeDo as tools for developing computational thinking in children 11 to 12 years

Abstract

In Peruvian education, has promoted the use of technological tools such as WEDO and Scratch as means which can make possible the improvement of educational quality, but

have not defined the types of thought or capabilities of superior order that must be accomplished in the students to employ these means. In response it is postulated this research along the lines proposed by researchers from the Massachusetts Institute of technology (MIT) and the Carnegie Mellon University on the development of computational thinking in children, seeking to understand and adapt this proposal in a systematic way; familiar concepts within a phenomenon little-documented, possible trendsetter of study that can be performed in our educational environment and pointing out the potential relationship between the variable framework with Scratch, WEDO and computational thinking.

Addresses the problem of how teaching computational thinking children of 11 and 12 years old in schools, which has been developing a set of learning tasks based on challenges that encourages the student to test solutions through direct instruction, simulation and video games using Scratch and WEDO.

Keywords:

Computational thinking, Scratch, WeDo, framework, algorithms, ICT in Education.

I. Introducción

Existe una enorme corriente de opinión que manifiesta de manera abierta las bondades de aplicar las computadoras como herramientas que permiten el desarrollo de diversos tipos de pensamiento, de las habilidades y competencias del siglo XXI. Pero pocas investigaciones apoyan esta corriente de opinión. En este sentido en el ámbito educativo peruano, se viene promoviendo la utilización de herramientas de programación como Scratch y recientemente kits WEDO como agentes que pueden posibilitar la mejora de calidad educativa, pero no contamos con sustento probado en las escuelas peruanas de sus resultados; es más no se cuenta con una estrategia de aplicación que permita identificar qué tipos de pensamientos o capacidades superiores estamos logrando desarrollar en nuestros estudiantes con la aplicación de estas herramientas tecnológicas.

Frente a esta enorme carencia se postula la presente investigación exploratoria que sigue la línea propuesta por investigadores del Instituto de Tecnología de Massachusetts (MIT) y del Carnegie Mellon University respecto al desarrollo en niños menores de 12 años de un tipo de Pensamiento denominado Computacional, en base a la utilización de herramientas como Scratch y WEDO. Este primer avance busca examinar un tema poco estudiado de manera sistemática, buscando familiarizar conceptos dentro de un fenómeno poco documentado, marcando posibles tendencias de estudio que se pueden llevar a cabo en nuestro ámbito educativo, señalando las potenciales relaciones entre la variable marco de trabajo con Scratch y WEDO y pensamiento computacional.

II. Marco teórico

Se partió de la operacionalización del marco de trabajo que está definido por el entorno de aprendizaje, la estrategia de aprendizaje y los contenidos a desarrollar en Scratch y WEDO para lograr desarrollar el pensamiento computacional.

Respecto al entorno de aprendizaje como explican Loughlin y Suina (1987) el profesor tiene cuatro tareas principales en la disposición de la estructura básica del entorno de aprendizaje: organización espacial, dotación del aprendizaje, disposición de los materiales y organización para propósitos especiales. La organización espacial es la tarea de disponer los muebles y equipos para actividad de aprendizaje. La dotación es la tarea de seleccionar y colocar en el contexto los equipos, materiales de robótica, fuentes de información, los programas y elementos auxiliares al alcance de los niños. La disposición de los materiales es el proceso de decidir cómo y cuándo combinar y exhibir esta dotación. La organización para propósitos especiales implica disponer el entorno para promover y lograr los resultados deseados respecto al aprendizaje.

Toda estrategia debe estar orientada de manera general para lograr un objetivo; en el ámbito educativo nos guiamos por los objetivos educativos que se miden mediante indicadores. Últimamente en el ámbito peruano se va avanzando de un modelo regido por objetivos a uno por competencias, que implican habilidades o capacidades, conocimiento y actitudes; donde sus indicadores de logro se formulan entorno a verbos. Donde el interés resaltante se centra profundamente en las capacidades, por ello se seleccionó dentro de la investigación como guía, la taxonomía de Bloom actualizada para la era digital por Anderson y Krathwohl (2000) que se enfoca en el uso de los medios digitales para lograr describir procesos de aprendizaje en niveles de pensamiento de orden superior como: recordar, comprender, aplicar, analizar, evaluar y crear.

Respecto a los contenidos se propuso una acción activa sobre ellos, según Dale (1969) se considera de naturaleza activa, aquellas como la participación en una discusión, dar una plática, hacer una presentación dramática, una simulación de la actividad real y la propia actividad real. Por ello apuntamos a organizar la apropiación de contenidos mediante proyectos de realización como trabajar con lenguaje de señas para niños sordos, elaboración de animaciones, videojuegos, simulaciones y prototipos robóticos.

López (2011) afirma que Scratch es un entorno de programación desarrollado por un grupo de investigadores del Instituto Tecnológico de Massachusetts (MIT), bajo la dirección de Mitchel Resnick. El propio Resnick (2009) indica que Scratch se basa en un enfoque que atrae a la programación a las personas que no son programadores con el objetivo de que se obtenga fluidez en las nuevas tecnologías, teniendo en cuenta que hasta hora muchos envían textos, juegan en línea o navegan por la Web, pero pocos son capaces de crear sus propios juegos, animaciones o simulaciones. Es como si se puede “leer”, pero no “escribir”. Por ello Scratch es más como un bricolaje con bloques armables e intercambiables, significativo porque apoya la realización de diversos tipos de proyectos y la personalización de los mismos y social porque está ligado a una comunidad que colabora y comparte.

MED Perú (2011) afirma que el Kit de robótica WeDo ha sido diseñado para el nivel primario, permitiendo construir prototipos de diversa complejidad con motores y sensores pudiéndose usar con las XO. Las actividades con este material permiten construir un modelo funcional que pueda complementar y articular actividades de las áreas de matemáticas, ciencia y ambiente o comunicación.

Según Wing (2012) el pensamiento computacional será una habilidad fundamental utilizada por todos en el mundo. A la lectura, escritura y aritmética, hay que añadir el pensamiento computacional a la capacidad analítica de cada niño. El pensamiento computacional implica la resolución de problemas, diseño de sistemas, y la comprensión de la conducta humana, sobre la base de los conceptos que son fundamentales para la informática. Pensar como un Informático significa más que ser capaz de programar una computadora. Se requiere la capacidad de abstracción y por lo tanto pensar en múltiples niveles de abstracción. Para la Sociedad Internacional para la Tecnología en Educación (ISTE) y la Asociación de Docentes en Ciencias de la Computación (CSTA) el pensamiento computacional es un proceso de solución de problemas que incluyen entre otros la formulación de problemas, la organización y análisis de datos, modelos y simulaciones, cierta automatización mediante pensamiento algorítmico, la formulación de soluciones, selección de la mejor solución, aplicación de la solución y la generalización de la solución a otros contextos. Esto a su vez requiere de una serie de actitudes como confianza en el manejo de la complejidad, la constancia, tolerancia a la ambigüedad, mente abierta para trabajar con problemas no estructurados, la comunicación y el trabajo en equipo.

El pensamiento computacional podría aportar como beneficios a los educando la posibilidad de:

- Facilitar nuevas formas de ver los problemas existentes
- Según Phillips (2008):
 - Destacar la creación de conocimiento en lugar de utilizar la información.
 - Presentar posibilidades para resolver creativamente problemas.
- Facilitar la innovación

III. Objetivos y Metodología

Objetivo:

El termino pensamiento computacional aún no está definido a cabalidad por ello partimos de los siguientes objetivos:

- Mejorar el conocimiento de lo que se denomina pensamiento computacional para identificarlo como una habilidad superior.
- Adaptar un marco de trabajo con Scratch y WeDo para desarrollar el pensamiento computacional.

Muestra de Estudio:

La muestra de estudio seleccionada comprendió a 12 alumnos que asistieron en el 2013 al 6to grado de Educación Primaria de la I.E. “Jesús Nazareno”, Huaraz – Ancash Perú, en el periodo marzo – mayo.

Métodos:

- Nivel y tipo de Estudio: Aplicado
- Sistema de variables:
 - Variable Independiente: marco de trabajo con Scratch y WeDo
 - Variable Dependiente: procesamiento computacional.
 - Variables Intervienes: Edad, sexo, aptitudes de los niños.

- Operacionalización de variables:

Tabla 01: Variables de Investigación

| | |
|--|---|
| V.I. Marco de trabajo con Scratch y WeDo | VD. pensamiento computacional |
| Entorno de aprendizaje | Conceptos algorítmicos |
| Estrategias de aprendizaje | Prácticas comunes de programación |
| Nivel de reelaboración de contenidos | Generalización, colaboración y comunicación de resultados |

- Diseño de la Investigación: pre-experimental con pre y post test, con manipulación deliberada de la variable independiente.
- Técnicas e Instrumento de recolección de datos:
Técnicas: Observación, Cuestionario, análisis documental, matrices.
Instrumentos: Test, Rubrica de Evaluación, ficha de análisis, ficha de registro bibliográfico, matriz estadística.

Estructura y Elementos del Marco de Trabajo

a. La oportunidad detectada

Hablar de pensamiento computacional es nuevo y fresco en ámbito educativo, además da la oportunidad de usar las herramientas con que se cuenta en las escuelas con un sentido estratégico y poder reforzar con la misma herramienta los conceptos que se enseñan. Parafraseando a Wing (2008), usar la herramienta para que nuestras abstracciones cobren vida ya sea a través de visualizaciones o animaciones, es decir se da posibilidad a los estudiantes a través de la programación automatizar sus propias abstracciones aprovechando además el sentido exploratorio de los niños y tomar ventaja de la exposición a los dispositivos de cómputo a los que están expuestos los niños en el hogar y la escuela.

Recogiendo la última afirmación, también se puede explorar su influencia en el ámbito informal así como el aprendizaje formal; ya que el aprendizaje tiene lugar en muchos ámbitos fuera del aula: ya que los niños se enseñan unos a otros; aprenden de sus padres, familia y amigos; aprender en casa, en las cabinas públicas de internet o en los museos; y aprender a través de sus aficiones.

En otro sentido esta oportunidad puede ser limitada, a decir de Wing, por las múltiples barreras culturales, económicas, políticas y sociales en la realización de

una visión en la que el pensamiento computacional no tenga una relevancia central, sobre todo en países en los que el sistema educativo no es un tema principal. Pero aun así, actuar en favor de esta visión aún podría reportar beneficios que valen la pena.



Figura N° 01. Cámara Web accionada por brazo robótica

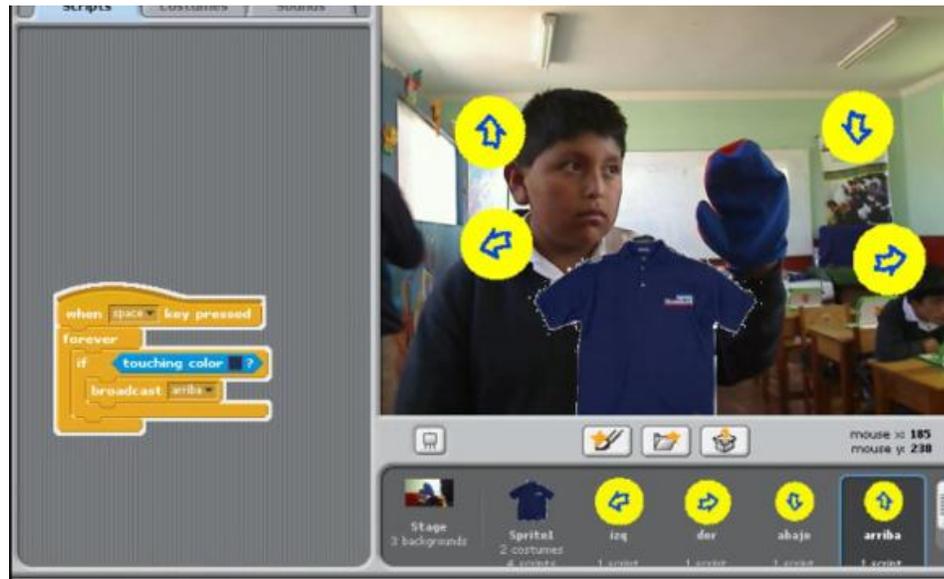


Figura N° 02. Probador de ropa virtual con detección de color

b. Las herramientas usadas

- SRATCH. Es un lenguaje de programación desarrollado por el Massachusetts Institute of Technology Media Lab (MIT), con la finalidad de que las personas puedan programar en forma sencilla, sin necesidad de complejos algoritmos y largas horas de cursos de programación. Es una evolución del famoso lenguaje de programación LOGO, que permite en muy corto tiempo la creación interactiva de animaciones, historias, juegos música, arte, etc. (Resnick, 2007). El lenguaje está basado bloques gráficos y con una interface fácil para su aprendizaje. Consiste en la unión de bloques que pueden ser eventos, movimientos de gráficos y sonidos.
- AR SPOT. Es un entorno de creación de realidad aumentada para los niños. Se basa en el proyecto de Scratch del MIT; este entorno permite a los niños crear experiencias que mezclan elementos reales y virtuales, gracias a que posibilita la integración de imágenes de una cámara de video con el entorno de Scratch, posibilitando interacciones y control de los objetos virtuales a través de las interacciones con los objetos físicos.
- WEDO. El kit de Robótica WeDo ha sido diseñado para el nivel de educación primaria, para estudiantes desde 7 a 11 años. Permite construir y programar prototipos de diversa complejidad, con motores y sensores que son automatizados a través del software de programación iconográfica WeDo; pero también puede ser contralado a través de Scratch.
El set de construcción básico WeDo permite a los niños construir y programar modelos robóticos LEGO sencillos conectados a un equipo informático. Este set contiene más de 150 elementos, incluyendo un motor, sensores de movimiento e inclinación y un Hub USB LEGO.
- ETOYS. Es un lenguaje de programación visual creado por Alan Kay orientado a objetos y basado en prototipos para el uso en educación. En el proyecto se lo ha empleado para complementar aspectos de interacción entre el mundo físico y el virtual y enriquecer la experiencia en los niños.



Figura N° 03. Funcionamiento conjunto de Etoys y Scratch para mover un móvil de WeDo con principios de visión artificial

c. El entorno de aprendizaje

Se organizó el entorno de manera modular dependiendo del trabajo a realizar sea individual o en equipos pequeños de 3 o más integrantes, con una pizarra interactivo como medio de ejemplifican de las estructuras y forma de compartir código elaborado; se contó con 5 laptops y 11 OLPC que tenían instalados Scratch 1.4. Siendo de vital importancia la pizarra digital el medio que permitió captar el funcionamiento de las estructuras y conceptos algorítmicos.

d. Estrategia de aprendizaje.

Se siguió la línea definida por Brennan y Resnick para estudiar el proceso de desarrollo del pensamiento computacional; para la investigación se realizó en dos etapas. La primera consistente en el aprendizaje de los conceptos de secuencia, ciclos, eventos, paralelismo, condicionales, operadores y datos. La segunda etapa consistió en ofrecer retos de como poder usar los conceptos aprendidos para elaborar traductores para comunicarse con niños sordos, aprendizaje de notas musicales, probadores de ropa virtual, simulación del ciclo del agua, conversiones entre sistemas de numeración, elaboración de historias, presentaciones sobre la célula, manejo de una faja transportadora, un automóvil, control de un brazo robótico.

Dentro de las estrategias se fueron explorando el desarrollo de las prácticas comunes de programación como el desarrollo incremental en base a la construcción de soluciones cada vez más complejas como traducir más palabras y con mayor variedad, ofrecer simulaciones más complejas para el caso del ciclo de agua o el efecto invernadero. La

formación en grupo permitió la colaboración en la depuración de las secuencias. Los niños aprendieron a reusar código para crear video juegos como los de pingpong o un mata gato; se organizaron secuencias con hilos paralelos y acciones modulares.

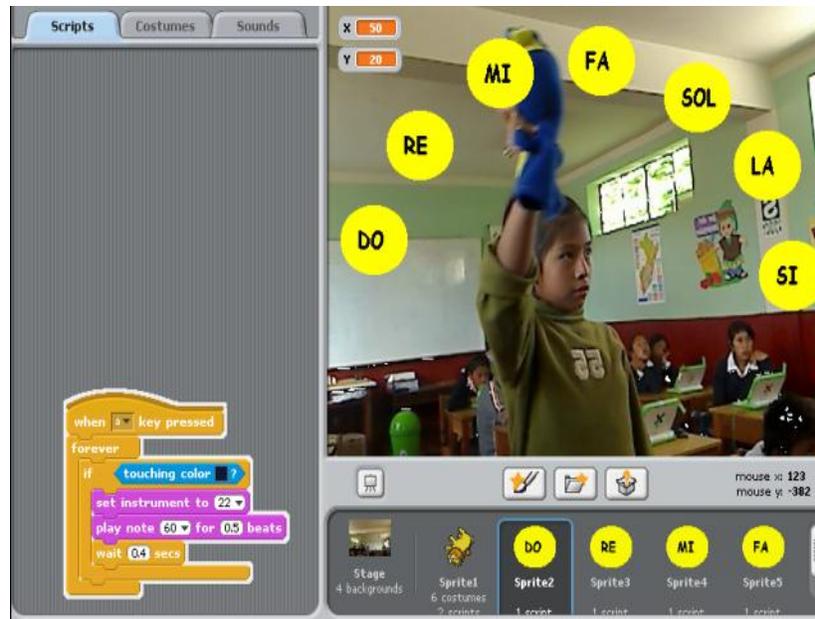


Figura N° 04. Piano virtual, que extiende el uso de las notas musicales en la OLPC a un ambiente de realidad aumentada.

e. Nivel de reelaboración de contenidos

Dentro de nuestro marco de trabajo se contempló la posibilidad de reelaborar contenido, esto referido a poder replicar mediante las herramientas, productos que se pueden observar en diferentes medios; en el camino de comprender el pensamiento computacional se buscó que esto se realizará mediante replicar juegos y animaciones que se podía observar con otras herramientas, así también se estableció jornadas para que los niños elaboren tutoriales y enseñaran a otros niños menores logrando que los niños reelaboren contenido dentro de sus experiencias posibilitando además comunicación y colaboración que son aspecto del pensamiento computacional.

Así también las actividades fueron grabadas por los propios niños, creando tutoriales sobre su trabajo en el aula; los que fueron publicados en una cuenta en YouTube.



Figura N° 05. Escolar elaborando tutorial sobre cómo construir un probador de ropa virtual

f. Actividades desarrolladas

Las actividades se guiaron por el siguiente esquema



Figura N° 06. Esquema de la estrategia de trabajo para el desarrollo del pensamiento computacional propuesto dentro del proyecto

Las actividades desarrolladas se organizaron en sub proyectos.

Sub proyecto: El lenguaje de Señas para personas sordas:

Objetivos:

- Reconocer y superar conflictos que se dan por ser diferentes
- Mejorar las habilidades comunicativas.
- Dominar elementos de la programación en computadora.
- Elaborar escritos funcionales para realizar una invitación

- Difundir los derechos de igualdad y respeto entre los seres humanos, Usar las TIC de manera creativa para resolver problemas y presentar información.
- Fomentar la inclusión

Lecciones

| | |
|------------------------|--|
| Primera | Descubriendo el problema: Los derechos de los peruanos con discapacidad |
| Segunda | Nos organizamos para aprender el lenguaje de señas peruanas usando las TIC |
| Tercera, Cuarta | Trabajamos con los especialistas en el lenguaje de señas |
| Quinta | La tecnología para personas sordas |
| Sexta | Enseñamos el lenguaje de señas a nuestros compañeros menores |
| Séptima | El lenguaje de señas es súper, ¡apréndelo! |



Figura N° 07. Trabajo de los niños con especialistas en el lenguaje de Señas Peruanas

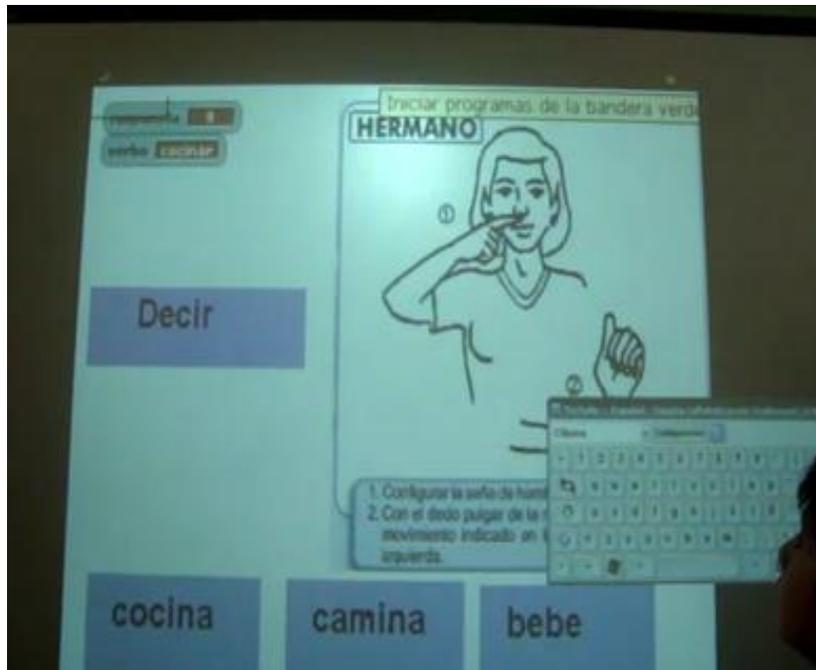


Figura N° 08. Traductor del lenguaje de señas peruanas implementado en Scratch por los niños participantes

Sub proyecto: Dominando poco a poco la Tecnología en la Escuela

Objetivos:

- Establecer vínculos entre causa y efecto.
- Desarrollar la creatividad a través de la construcción de modelos funcionales.
- Desarrollar habilidades de vocabulario y comunicación para explicar cómo funciona un modelo o un juego.
- Buscar respuestas e imaginar nuevas posibilidades de uso de las máquinas y los videojuegos
- Comparar el cambio de factores y medir sus efectos.
- Elabora y presenta tablas de datos
- Proponer soluciones a problemas locales usando máquinas automatizadas o aplicaciones.

Lecciones

| | |
|------------------------|---|
| Primera | La importancia de la robótica y la programación en el mundo |
| Segunda | Controlando un móvil de diferentes maneras |
| Tercera, Cuarta | Las animaciones y videojuegos en Scratch |
| Quinta | Mi automóvil virtual en Etoys |
| Sexta | Usamos WEDO, Etoys y Scratch |
| Séptima, | El brazo robótico accionado a distancia |
| Octava | Las notas musicales virtuales |
| Novena | Musicalizando una poesía |
| Décima | El probador de Ropa Virtual |
| Undécima | La faja transportadora controlada automáticamente |

IV. Resultados obtenidos y comentarios finales

En los datos que se muestra a continuación se ilustra los resultados obtenidos luego de aplicar el marco de trabajo con Scratch y WeDo, en el cual se ha tratado de medir fundamentalmente los conceptos algorítmicos como un primer avance dejando para posteriores intervenciones otros indicadores, obteniéndose los siguientes resultados:

Tabla 02: Distribución de Frecuencias del Pre Test.

| conceptos | Secuencia y paralelismo | | Condiciones, eventos y ciclos | | Operadores y datos | |
|-----------|-------------------------|--------|-------------------------------|--------|--------------------|--------|
| | N | % | N | % | N | % |
| Bajo | 5 | 41.67% | 6 | 50.00% | 5 | 41.67% |
| Medio | 6 | 50.00% | 5 | 41.67% | 7 | 58.33% |
| Alto | 1 | 8.33% | 1 | 8.33% | 0 | 0.00% |
| TOTAL | 12 | 100 | 12 | 100 | 12 | 100 |

El Pre Test se diseñó de 0 a 20 puntos. Donde el 91, 67% de los alumnos obtuvieron una valoración entre bajo y medio respecto a los conceptos algorítmicos pertenecientes al pensamiento computacional. Solo un alumno poseía un nivel alto en la medición al iniciar la aplicación del marco de trabajo.

Tabla 3: Distribución de Frecuencias del Post Test.

| conceptos | Secuencia y paralelismo | | Condiciones, eventos y ciclos | | Operadores y datos | |
|-----------|-------------------------|--------|-------------------------------|--------|--------------------|--------|
| | N | % | N | % | N | % |
| Bajo | 1 | 8.33% | 1 | 8.33% | 0 | 0.00% |
| Medio | 4 | 33.33% | 5 | 41.67% | 5 | 41.67% |
| Alto | 7 | 58.33% | 6 | 50.00% | 7 | 58.33% |
| TOTAL | 12 | 100 | 12 | 100 | 12 | 100 |

El Post Test, estuvo elaborado de 0 a 20 puntos. Solo un 8,33% obtuvieron un puntaje catalogado como bajo o con muy poco logro. Un 33,33% obtuvieron puntaje catalogado

como regular. Mientras un 58,33% logaron una valoración alta en por lo menos en dos conceptos algorítmicos del pensamiento computacional.

Finalmente como corolario de este primer avance, se ha logrado una aproximación a la definición de los componentes del pensamiento computacional, midiéndose el aspecto de conceptos algorítmicos. Respecto al marco de trabajo de Scratch y WeDo se ha realizado un esfuerzo por marcar futuras tendencias que podrán ser tomadas en otros estudios como son la consideración de los entornos de trabajo y los niveles de reelaboración de contenidos a partir de las herramientas trabajadas.

Referencias Bibliográficas

- Anderson, L., Krathwohl, D. y Bloom, B. (2001). A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. Longman, New York.
- Brennan, K. y Resnick, M. (2012). Nuevas propuestas para estudiar y evaluar el desarrollo del Pensamiento Computacional. Disponible en: <http://www.eduteka.org/modulos/9/284/2120/1>
- Cherches, A. (2009). Taxonomía de Bloom para la Era Digital. Disponible en: <http://www.eduteka.org/TaxonomiaBloomDigital.php>
- Communications of the ACM. (2009). Scratch: Programming for All. Communications of the ACM, 52 (11), 60-67. Disponible en: <http://cacm.acm.org/magazines/2009/11/48421-scratch-programming-for-all/fulltext>
- Cruz, S., Fernández, B., y Vaz, C. (2013). Enfoques Innovadores en Juegos Serios. Disponible en: <http://hal.inria.fr/docs/00/82/03/50/PDF/VAEP-RITA.2013.V1.N1.A4.pdf>
- González, J (2013). Pensamiento Computacional. Disponible en: <http://es.scribd.com/doc/207183916/pensamiento-informatico>
- López, J. (2011). Programación con Scratch. Recuperado de <http://www.eduteka.org/pdfdir/algoritmosprogramacioncuaderno1.pdf>
- Loughlin, C., y Suina, J. (2002). El ambiente de Aprendizaje: Diseño y Organización (5ta Ed). Madrid: Ediciones Morata.
- MIDEDU Perú. (2011). Robótica Educativa WeDo. Lima.
- Phillips, P. (2008). Computational thinking: a problem-solving tool for every classroom. Recuperado de: <http://education.sdsc.edu/resources/CompThinking.pdf>
- Wing, J. (2008). Computational thinking and thinking about computing. Disponible en: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2696102/>
- Wing, J. (2010). Computational Thinking: What and Why?. Disponible en: <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Wing, J. (2012). Computational Thinking. Disponible en: <http://research.microsoft.com/apps/video/default.aspx?id=179285>