



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE CIENCIAS DE LA EDUCACIÓN
ESCUELA PROFESIONAL DE EDUCACIÓN PRIMARIA



**APLICACIÓN DEL SOFTWARE DE PROGRAMACIÓN SCRATCH
EN EL DESARROLLO DEL PENSAMIENTO COMPUTACIONAL
DE LOS ESTUDIANTES DEL QUINTO GRADO DE LA
INSTITUCIÓN EDUCATIVA PRIMARIA N° 70 116
CARITAMAYA, 2020**

TESIS

PRESENTADA POR:

Bach. LIZBETH CONDE MAYTA CUTIPA

**PARA OPTAR EL TÍTULO PROFESIONAL DE:
LICENCIADA EN EDUCACIÓN PRIMARIA**

PUNO – PERÚ

2022



DEDICATORIA

Dedico esta tesis principalmente a Dios, por permitirme llegar hasta este momento tan importante de mi formación profesional. A mis queridos padres Humberto y Eulalia por el gran apoyo que me brindaron, porque son ejemplo de trabajo, esfuerzo y amor; a través de estas líneas les expreso mi profunda admiración. A mi hermana Ruth Karina por todo su apoyo moral durante mi formación profesional.

Lizbeth Condemayta Cutipa



AGRADECIMIENTOS

A Dios por haberme acompañado espiritualmente y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes y experiencias.

Expreso mi más profundo y sincero agradecimiento a la Universidad Nacional del Altiplano, a la Escuela Profesional de Educación Primaria, por haberme conducido en el camino del conocimiento. En especial a la plana docente por todo los conocimientos impartidos y experiencias compartidas sobre la labor docente.

Mi agradecimiento sincero a mis queridos padres por su apoyo incondicional, por siempre motivarme a sobresalir.

A mi asesora D.sc. Danitza Luisa Sardón Ari, por su dedicación, sus conocimientos brindados, por sus valiosas observaciones y sugerencias, ya que han sido fundamentales para la culminación de mi tesis.

A los miembros del jurado: Dra. Zaida Esther Callata Gallegos, Lic. Milciades Conrado Suaña Calsin, y Dra. Natali Ardiles Cáceres, quienes encaminaron y fortalecieron la redacción de esta tesis.



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ACRÓNIMOS

RESUMEN.....11

ABSTRACT..... 12

CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA..... 15

1.2. FORMULACIÓN DEL PROBLEMA 21

1.2.1. Problema general..... 21

1.2.2. Problemas específicos 21

1.3. HIPÓTESIS DE LA INVESTIGACIÓN 21

1.3.1. Hipótesis general 21

1.3.2. Hipótesis específicas 22

1.4. JUSTIFICACIÓN DEL ESTUDIO 22

1.5. OBJETIVOS DE LA INVESTIGACIÓN 25

1.5.1. Objetivo general 25

1.5.2. Objetivos específicos 25

CAPÍTULO II

REVISIÓN DE LA LITERATURA

2.1. ANTECEDENTES 26

2.2. MARCO TEÓRICO 33

2.2.1. Software de programación Scratch 33

2.2.1.1. Características del software de programación Scratch..... 35



2.2.1.2.	Entorno del software de programación Scratch	36
2.2.1.3.	Dimensiones del software de programación Scratch	38
2.2.2.	Pensamiento computacional.....	41
2.2.2.1.	Dimensiones del pensamiento computacional	44
2.3.	MARCO CONCEPTUAL	46

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1.	UBICACIÓN GEOGRÁFICA DEL ESTUDIO.....	48
3.2.	PERIODO DE DURACIÓN DEL ESTUDIO	48
3.3.	PROCEDENCIA DEL MATERIAL UTILIZADO	49
3.4.	POBLACIÓN Y MUESTRA DEL ESTUDIO	50
3.5.	DISEÑO ESTADÍSTICO	52
3.6.	PROCEDIMIENTO.....	53
3.7.	VARIABLES	55
3.8.	ANÁLISIS DE LOS RESULTADOS	57
3.9.	METODOLOGÍA DE LA INVESTIGACIÓN	57

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1.	RESULTADOS.....	60
4.2.	DISCUSIÓN.....	76
V.	CONCLUSIONES.....	80
VI.	RECOMENDACIONES.....	82
VII.	REFERENCIAS BIBLIOGRÁFICAS	84
	ANEXOS.....	94

Área : Perspectivas teóricas de la educación

Tema : Pensamiento Computacional

Fecha de sustentación: 03/Febrero/2022



ÍNDICE DE FIGURAS

Figura 1.	Área de programación Scratch.....	37
Figura 2.	Área de escenario de Scratch.....	38
Figura 3.	Ubicación Satelital del Centro Poblado de Caritamaya.....	48
Figura 4.	Resultados del Pre Test y Post Test sobre el desarrollo de conceptos computacionales en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.....	60
Figura 5.	Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de conceptos computacionales.....	62
Figura 6.	Resultados del Pre Test y Post Test sobre el desarrollo de la dimensión prácticas computacionales en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.....	64
Figura 7.	Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de prácticas computacionales.....	66
Figura 8.	Resultados del Pre Test y Post Test sobre el desarrollo de Perspectivas Computacionales en estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.....	68
Figura 9.	Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de perspectivas computacionales.....	70
Figura 10.	Comparación de resultados del Pre Test y Post Test sobre el desarrollo del pensamiento computacional en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.....	72



Figura 11. Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post- test sobre el desarrollo del pensamiento computacional..... 74



ÍNDICE DE TABLAS

Tabla 1	Bloques de órdenes de Scratch	36
Tabla 2	Población de estudio	51
Tabla 3	Muestra de estudio	51
Tabla 4	Dimensiones del software de programación Scratch.....	55
Tabla 5	Dimensiones del pensamiento computacional.....	56
Tabla 6	Diseño de la investigación	58
Tabla 7	Estadísticos de fiabilidad	59
Tabla 8	Resultados del Pre Test y Post Test sobre el desarrollo de conceptos computacionales en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.	60
Tabla 9	Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de conceptos computacionales.....	62
Tabla 10	Prueba t de Student de muestras emparejadas para el objetivo específico 1	63
Tabla 11	Resultados del Pre Test y Post Test sobre el desarrollo de prácticas computacionales en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.	64
Tabla 12	Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de prácticas computacionales.....	66
Tabla 13	Prueba t de Student de muestras emparejadas para el objetivo específico 2	67
Tabla 14	Resultados del Pre Test y Post Test sobre el desarrollo de Perspectivas Computacionales en estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.	68



Tabla 15	Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de perspectivas computacionales.....	70
Tabla 16	Prueba t de Student de muestras emparejadas para el objetivo específico 371	
Tabla 17	Comparación de resultados del Pre Test y Post Test sobre el desarrollo del pensamiento computacional en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.	72
Tabla 18	Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post- test sobre el desarrollo del pensamiento computacional.....	73
Tabla 19	Prueba t de student de muestras emparejadas sobre el desarrollo del pensamiento computacional.....	75
Tabla 20	Resumen de valores estadísticos calculados.....	75



ÍNDICE DE ACRÓNIMOS

ISTE:	Instituto Tecnológico Superior España
CSTA:	Computer Science Teachers Association
TIC:	Tecnologías de la Información y la Comunicación
MIT:	Instituto de Tecnología de Massachusetts
IE:	Institución Educativa
SPSS:	Statistical Package for the Social Sciences
TPC:	Test de Pensamiento Computacional
EBR	Educación Básica Regular
JEC	Jornada Escolar Completa
JER	Jornada Escolar Regular
COAR	Colegio de Alto Rendimiento
STEM	Science, Technology, Engineering and Mathematics



RESUMEN

La investigación realizada tuvo como propósito determinar la eficacia de la aplicación del software de programación “Scratch” en el desarrollo del pensamiento computacional en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya. Para ello, se estableció una metodología de tipo aplicada con un enfoque cuantitativo y diseño pre experimental con corte longitudinal. La muestra fue determinada por el muestreo no probabilístico intencional y estuvo conformada por 13 estudiantes del 5° grado de educación primaria, a los cuales les fue aplicado un test de pensamiento computacional diseñado por Marcos Román, el cual fue validado por criterio de jueces y obtuvo la aprobación necesaria. Para lograr los objetivos propuestos se aplicó el software de programación Scratch a través de diez sesiones de aprendizaje como propuesta pedagógica. El contraste de la hipótesis se ha realizado mediante la aplicación de la prueba estadística t de Student para muestras relacionadas. Los resultados fueron analizados en el nivel descriptivo y en el nivel inferencial, de esta manera se puede destacar una diferencia significativa de 7.15 en las medias de los puntajes obtenidos por los estudiantes en la aplicación del Pre Test y Post Test. Además, el P-valor (0.000) resultó menor que el nivel de significancia (0.05) y el valor de la prueba t calculada (11.024) resultó mayor que el valor de la prueba t tabulada (1.782). Concluyendo que la aplicación del software de programación Scratch es eficaz en el desarrollo del pensamiento computacional de los estudiantes.

Palabras Clave: Alfabetización digital, Pensamiento computacional, Primaria, Scratch.



ABSTRACT

The purpose of the research carried out was to determine the effectiveness of the application of the "Scratch" programming software in the development of computational thinking in fifth grade students of the Primary Educational Institution No. 70 116 Caritamaya. For this, an applied type methodology was established with a quantitative approach and a pre-experimental design with longitudinal section. The sample was determined by intentional non-probabilistic sampling and was made up of 13 students of the 5th grade of primary education, to whom a computational thinking test designed by Marcos Román was applied, which was validated by judges' criteria and obtained the necessary approval. To achieve the proposed objectives, the Scratch programming software was applied through ten learning sessions as a pedagogical proposal. The hypothesis contrast was carried out by applying the student's t test for related samples. The results were analyzed at the descriptive level and at the inferential level, in this way a significant difference of 7.15 can be highlighted in the means of the scores obtained by the students in the application of the Pretest and Post Test. In addition, the P-value (0.000) was lower than the significance level (0.05) and the value of the calculated t-test (11.024) was higher than the value of the tabulated t-test (1.782). Concluding that the application of Scratch programming software is effective in the development of computational thinking of students.

Key Words: Digital Literacy, Computational Thinking, Primary, Scratch.



CAPÍTULO I

INTRODUCCIÓN

En la actualidad, se está impulsando un proceso de pensamiento orientado a la formulación y solución de problemas que puede ser aplicado a la vida cotidiana: El pensamiento Computacional. Dicho termino fue acuñado por la ingeniera estadounidense Jeannette Wing, quien indica que este pensamiento permite adaptar la computación a nuestras necesidades, por ello se han ido desarrollando programas informáticos que ofrecen soluciones simples a necesidades cotidianas.

En el sistema educativo, según Zapata (2015) desarrollar el pensamiento computacional desde edades tempranas resulta imprescindible para mejorar la competitividad e innovación y formar a los estudiantes en las competencias que les permita pasar de consumidores a productores de tecnología. Adicionalmente, refiere que el Pensamiento Computacional en este momento es una materia obligatoria en los planes curriculares escolares de países desarrollados y se ha ido fortaleciendo a través de la programación. Del mismo modo, Universidad Católica San Pablo (UCSP, 2021) resalta que las pruebas PISA del 2022, han propuesto evaluar por primera vez el pensamiento computacional en los estudiantes a nivel mundial. En consecuencia, el Perú tiene una oportunidad de oro para hacer un cambio generacional e implementar el pensamiento computacional en las instituciones a través de la programación.

De esta manera, surgió el interés de realizar una investigación para determinar la eficacia de la aplicación del software de programación Scratch en el desarrollo del pensamiento computacional, así mismo determinar la eficacia de la aplicación del software en el desarrollo de conceptos computacionales, practicas computacionales y perspectivas computacionales.



El informe comprende cuatro capítulos según los aspectos formales establecidos por la Unidad de Investigación de la Facultad de Ciencias de la Educación. Estos capítulos son los siguientes:

El Capítulo I hace referencia al planteamiento del problema de investigación en forma de preguntas; se justifica la realización de la investigación utilizando referencias, así como también se consideran el objetivo general y los objetivos específicos planteados en el estudio realizado.

En el Capítulo II se aborda el marco teórico del estudio, los antecedentes de investigación y el marco conceptual.

El Capítulo III comprende la parte metodológica de la investigación, las técnicas e instrumentos, el proceso de recojo de datos y el proceso de tratamiento de datos.

Finalmente, en el Capítulo IV se dan a conocer los resultados de la investigación y se realiza la discusión.

El informe de investigación termina con las conclusiones que se plantean en función de los objetivos; las recomendaciones se alcanzan en función de las conclusiones, posteriormente se encuentra las referencias bibliográficas y los anexos pertinentes.



1.1. PLANTEAMIENTO DEL PROBLEMA

El desarrollo de las Tecnologías de Información y Comunicación (TIC) exige un constante uso de la informática (software) y computación (hardware) en la vida cotidiana. “Se trata de una nueva alfabetización, la alfabetización digital, y que como tal hay que comenzar desde las primeras etapas del desarrollo individual, así como sucede con otras habilidades clave: la lectura, la escritura y las habilidades matemáticas” (Zapata, 2015, p. 1).

Frente a ello, según Mantilla y Negre (2021), el desarrollo del pensamiento computacional se consolida como una estrategia efectiva para favorecer la alfabetización digital representando una tendencia estratégica educativa que permite incrementar las habilidades en los estudiantes para resolver problemas.

A nivel Mundial, Bocconi et al. (2016) sostienen que en diversos países europeos (Inglaterra, Francia, Finlandia, Italia, Portugal) se ha incluido a la programación y pensamiento computacional en educación primaria y secundaria, ya sea como un curso requerido o una actividad extra curricular. Así mismo, Duarte et al. (2020) destacan que estas iniciativas son en gran parte posibles debido al alto nivel de acceso a computadores y a conexión a internet de los países europeos.

Valverde et al. (2015) manifiestan que Reino Unido y Madrid han elaborado propuestas curriculares, sin embargo, surgieron con importantes limitaciones porque su introducción no va acompañada de una reforma sustancial de la organización escolar ni tampoco de los principios metodológicos que debería asumir el aula del siglo XXI porque no han tomado en cuenta evidencias de trabajos de investigación educativa.



Del mismo modo, los autores refieren que la forma más frecuente de enseñar a programar se basa solo en la comprensión de procedimientos puramente de programación y esto excluye a los que no tienen el don para programar directamente, creando con ello el estereotipo de que la programación es solo cosa de los programadores. Por último, agregan que, el pensamiento computacional, como cualquier otra competencia básica en el siglo XXI, no debería estar limitada dentro de una asignatura de un currículo.

A nivel de Latinoamérica, Vásquez et al. (2019) mencionan que el Pensamiento Computacional comienza a ser una preocupación de los gobiernos, siendo obligatoria en Argentina, desde 2018. En Uruguay no tiene la obligatoriedad, sin embargo, al ser de carácter voluntario es difícil tener una dimensión clara de su avance. Además, según García (2020) no existen investigaciones que sistematicen los trabajos realizados en esta área, al realizarse una encuesta nacional en el 2019, los docentes señalaron no tener capacitación previa en Pensamiento Computacional y sólo el 44% de 155 instituciones encuestadas de ámbito urbano desarrollan trabajos relativos a PC, pese a que el Plan Ceibal implementado en Uruguay provee de tabletas o computadoras para todos los estudiantes y docentes de la educación primaria, además de conectividad a internet en los centros educativos, y planes de conectividad gratuita tanto en lugares públicos como en hogares de bajos recursos.

Así mismo, Vásquez, et al. (2019) mencionan que en Chile se ha desarrollado diversas iniciativas en relación con el PC, estos son el proyecto “Programa tus ideas” desde 2015, la Red de colegios SIP implementando cursos de programación utilizando plataformas como Code.org, Scratch y Khan Academy. También, firmó un acuerdo con Google for education en 2017, con el objetivo de incentivar la programación y adicionalmente se propuso abrir el primer colegio enfocado en la programación.



A pesar de los intentos, los autores mencionan que dichas iniciativas no han logrado desarrollarse a nivel transversal, debido a que los indicadores del país muestran que los conocimientos digitales son excesivamente bajos, pese a que se han puesto en marcha recientes políticas públicas para el desarrollo del PC en el aula, aún resulta prematuro poder vislumbrar los resultados de dicha implementación por lo que aún se considera un desafío para el Gobierno y el sector educativo chileno.

“Desde otra posición se encuentra México, un país que ha fracasado en sus intentos de introducir el Pensamiento Computacional al currículo, a consecuencia de su baja inversión en infraestructura tecnológica” (Fundación Telefónica, 2016, como se citó en Vásquez et al., 2019, p. 44).

Ante la preocupación sobre el tema, surge “La Mesa Latinoamericana de Pensamiento Computacional” en el año 2019, para proponer y desarrollar papers y proyectos, todo con el gran objetivo de aportar al fortalecimiento de la educación según las demandas y necesidades de los contextos locales y regionales. Los países integrados fueron Chile, Colombia, Argentina, México y en Perú se encuentra la mesa denominada STEM y Pensamiento Computacional liderada por la Universidad de San Pablo de Arequipa (Red STEM Latinoamérica, 2019).

En el año 2020, se realiza una conferencia que congregó a ponentes nacionales e internacionales, en la cual se ha expuesto los desafíos que afronta la implementación del PC: Alex Cuadros (Perú), mencionó la necesidad de realizar investigación aplicada entre los actores académicos con la sociedad civil, Eugenia Cortona (Argentina) hizo referencia al desafío de como incorporar el contenido del pensamiento computacional dentro del currículo y hacerlo transversal en la enseñanza de las disciplinas de distintos niveles, y Emiliano Pereiro (Uruguay) mencionó que no se tiene recursos humanos suficientes para masificar el desarrollo de pensamiento computacional, no hay cantidad de docentes



especializados y sólo se proveen de docentes remotos y, por último, aún está en construcción el currículo que busca integrar el pensamiento computacional. (Code En Mi Cole, 2020).

A nivel Nacional, Acurio (2017) menciona que, para el Ministerio de Educación aún es un reto migrar del concepto tradicional de las TIC hacia el de la inteligencia digital porque se ha trabajado al revés, sin entender la lógica del pensamiento computacional, se compraron muchos equipos, hasta de robótica, pero no estaban enmarcados en el currículo nacional que tenía una definición de competencia digital. Además, menciona que se debe priorizar el cierre de brechas, bajar la desigualdad al máximo y en el ámbito de educación esto significa, priorizar la educación rural frente a la urbana; es necesario la transformación del docente, quien no solo debe familiarizarse con estas tecnologías, sino debe apropiarse de ellas como un aliado que lo ayuda a hacer cosas que no podía hacer bajo un sistema tradicional.

Actualmente, según Ministerio de Educación, (2019) hay una organización denominada “code en mi cole” que trabaja el pensamiento computacional con niños del Programa de Atención Educativa para Niños con Facultades Talentosas Sobresalientes (PAENFTS) del Ministerio de Educación. Sin embargo, el resto de las instituciones a nivel nacional aún no han incorporado el pensamiento computacional en sus aulas debido a que no han encontrado las estrategias adecuadas ni los medios necesarios para realizarlo.

Así mismo, el Ministerio de Educación en coordinación con la organización “code,org” ha incorporado una campaña denominada “La hora del código” para desarrollar la competencia transversal 28 del currículo Nacional. Según Hourofcode (2020) la campaña mencionada fue diseñada para romper las barreras de la programación y mostrar que las ciencias de la computación no son difíciles y cualquier persona puede aprender los conceptos básicos.



Ministerio de Educación (2019) refiere que, en la campaña desarrollada en el 2019 participaron 40 estudiantes del nivel primario y 160 del nivel secundario pertenecientes a 6 instituciones educativas de Lima Metropolitana. Así mismo, se conectaron 20 estudiantes del nivel primario y 180 del nivel secundario, pertenecientes a instituciones de seis (6) regiones: Ayacucho, Arequipa, Cajamarca, Huancavelica, Lambayeque, Tumbes. Entre dichas Instituciones Educativas se encuentran COAR, JEC y JER

Sin embargo, es preocupante conocer que sólo seis de veinticuatro regiones participaron de esta campaña tan importante que genera espacios de innovación para el desarrollo de competencias digitales basadas en el pensamiento computacional, donde tampoco se resaltó la participación de la región de Puno y mucho menos la participación de Instituciones Educativas de zonas rurales. En ese sentido, según Code learn (2020), todo apunta a un momento de cambio bajo la necesidad de repensar el sistema educativo actual y definir el espacio que debe ocupar el pensamiento computacional en las aulas.

A nivel regional, Flores (2019) refiere lo siguiente:

En la región de Puno, al igual que en el resto del Perú, la alfabetización digital es un término que aún no incursionan en el diccionario de las escuelas, se usa el computador para ser usuarios de aplicaciones generales como sistemas operativos, procesadores de texto, hojas de cálculo, juegos etc.; a lo cual le denominan cursos de computación, dejándose de lado el uso del computador para la creación de sus propias aplicaciones, generación de conocimiento, discusión y análisis de información, y otros conceptos presentes en los nuevos paradigmas computacionales para las escuelas; que nos incursionarían en un nuevo mundo para formar parte de la sociedad activa digital, dejando de ser meros consumidores de productos tecnológicos digitales. (p. 11)



En la institución Educativa intervenida se ha identificado barreras que dificultan la integración eficaz de las TIC al aula, estas barreras están relacionadas con el profesor, con su capacitación, con la infraestructura tecnológica, con el contexto, y con el soporte técnico. Existe falta de capacitación en estrategias metodológicas con el uso del portátil y cómo integrarlo a las áreas curriculares, falta de manejo de habilidades tecnológicas en los profesores, debido a ello se limitan a desarrollar las actividades preestablecidas en las computadoras y no se enfocan en consolidar la creatividad de los estudiantes para que se entusiasmen con la tecnología desde una perspectiva diferente. En términos de Muñoz (2019) “ya no es suficiente que los estudiantes aprendan a usar la tecnología solo como fuente de información y comunicación (las famosas TIC); deben comprenderla, aprender a hacer y crear. (Párr. 8)

Respecto a la infraestructura tecnológica, la cantidad de portátiles disponibles en la escuela no es proporcional al número de estudiantes, y en ocasiones se presenta problemas técnicos con los portátiles. Como consecuencia de ello, los estudiantes no han tenido la oportunidad de aprender a programar y no pueden desarrollar habilidades del pensamiento computacional, en ese sentido, se está perdiendo la oportunidad de aprender y desarrollar estas habilidades fundamentales para su futuro, “ya que de no hacerlo se seguirá formando simples consumidores y no productores de cambios sustanciales, donde la tecnología es un gran aliado” (Lazo, 2019, párr. 4).

Este panorama descrito se ha reflejado en una brecha digital muy grande, en el cuál debe incentivarse la innovación, para hacer un país menos desigual, ayudado por las tecnologías. El aporte frente a este problema consiste en trabajar el desarrollo del pensamiento computacional a través del software de programación Scratch, un lenguaje de programación visual recomendado para introducir a los niños en el mundo de la programación.



1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. Problema general

¿Será eficaz la aplicación del software de programación “Scratch” en el desarrollo del pensamiento computacional de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya?

1.2.2. Problemas específicos

¿Cuál es el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de conceptos computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya?

¿Cuál es el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de las prácticas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya?

¿Cuál es el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de las perspectivas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya?

1.3. HIPÓTESIS DE LA INVESTIGACIÓN

1.3.1. Hipótesis general

La aplicación del Software de programación “Scratch” es eficaz en el desarrollo del pensamiento computacional de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya 2020.



1.3.2. Hipótesis específicas

La aplicación del software de programación “Scratch” es significativamente eficaz en el desarrollo de conceptos computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

La aplicación del software de programación “Scratch” es significativamente eficaz en el desarrollo de las prácticas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

La aplicación del software de programación “Scratch” es significativamente eficaz en el desarrollo de las perspectivas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

1.4. JUSTIFICACIÓN DEL ESTUDIO

En la actualidad, según Román (2016), el pensamiento Computacional se viene situando en el foco de innovación educativa como un conjunto de habilidades de solución de problemas que debe ser adquirido por las nuevas generaciones de estudiantes. Además, Wing (2006) una teórica informática, precursora del pensamiento computacional, indica que esas son habilidades útiles para todo el mundo, no sólo para los científicos de la computación. En este contexto, Rushkoff (2010) manifiesta que, ser capaz de manejar el lenguaje de las computadoras emerge como una habilidad indispensable, un nuevo alfabetismo que nos permite participar de manera plena y efectiva en la realidad digital que nos rodea.



Es así que, el sistema educativo no puede mantenerse al margen de esta realidad digital, donde desarrollar el pensamiento computacional es cada vez más necesario por el rumbo que está tomando, y el protagonismo que está asumiendo el desarrollo tecnológico en la sociedad.

Además, Soria y Rivero (2019) manifiestan que se deben sumar esfuerzos en la sociedad para incluir el pensamiento computacional en los currículos educativos como una habilidad imprescindible en la formación de los estudiantes. Este nuevo escenario, exige que los profesores conozcan las diversas herramientas y lenguajes que permitan el desarrollo del PC.

Para desarrollar el pensamiento computacional, se requiere que los estudiantes aprendan a amar y descubrir la tecnología de una forma fácil y lúdica, por ello se propone trabajar “la programación”, esta idea se sustenta en el aporte de Vilanova (2018) quien sostiene que: “Aprender a programar computadores en la educación escolar constituye una buena alternativa para ayudar a los estudiantes a desarrollar habilidades de pensamiento de orden superior, especialmente pensamiento computacional” (p. 27).

Por ello, la finalidad de la presente investigación es aplicar el software de programación Scratch para el desarrollo del pensamiento computacional en los estudiantes de la IEP N° 70016 Caritamaya, puesto que es un lenguaje de programación visual recomendado para introducir a los niños en el mundo de la programación y resulta necesario, porque llegará un momento en el que saber programar será tan importante como saber leer o escribir. Además, “el Pensamiento Computacional es una habilidad transversal, no es una forma de pensamiento exclusiva de los ingenieros ni de los científicos computacionales” (Muñoz, 2018, párr. 6).



La investigación sobre el desarrollo del pensamiento computacional es relevante porque se trata de un proceso mental y según Code learn (2020), los beneficios educativos de poder pensar de manera computacional son varios, empezando por el uso de abstracciones que mejoran y refuerzan las habilidades intelectuales, y que por tanto pueden ser transferidos a cualquier otro ámbito. También ayuda a los estudiantes en la descomposición de un problema en otros más pequeños y más manejables, reconocer patrones en los problemas más sencillos, y diseñar pasos (algoritmos) que permitirán resolver el problema.

En ese sentido, es oportuno desarrollar el tema pensamiento computacional porque es un asunto de carácter social e involucra el quehacer educativo, además ayuda a cubrir un espacio de desarrollo formativo inexistente en el panorama regional. Por ello, se espera que sea de gran relevancia social, para que, en los próximos años, los estudiantes de Instituciones de ámbito urbano y rural estén familiarizados con la programación y los docentes sean capaces de integrarlas de manera progresiva en su práctica profesional para desarrollar el Pensamiento Computacional.

El tema es trascendental porque será el punto de partida para futuras investigaciones de profesionales de educación en la Universidad Nacional del Altiplano Puno, lo que beneficiará al ámbito regional conduciendo a la mejora de los aprendizajes en las instituciones educativas. Así mismo, tiene un alcance tecnológico porque se pretende innovar las prácticas pedagógicas utilizando el lenguaje de programación Scratch como recurso TIC, convirtiendo a los estudiantes en agentes activos y creadores de su propio conocimiento. Finalmente, la investigación realizada contribuirá con el enriquecimiento de la literatura científica (aspecto teórico) del pensamiento computacional para futuras investigaciones, puesto que los resultados presentados serán de gran utilidad para conocer la realidad al aplicarse a una nueva población.



1.5. OBJETIVOS DE LA INVESTIGACIÓN

1.5.1. Objetivo general

Determinar la eficacia de la aplicación del software de programación “Scratch” en el desarrollo del pensamiento computacional en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya, 2020.

1.5.2. Objetivos específicos

Identificar el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de conceptos computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Identificar el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de las prácticas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya

Identificar el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de las perspectivas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.



CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES

En el ámbito Internacional se ha consultado los siguientes trabajos de investigación:

Cabra y Ramírez (2022), en su estudio titulado “Desarrollo del pensamiento computacional y las competencias matemáticas en análisis y solución de problemas: una experiencia de aprendizaje con Scratch en la plataforma Moodle”, tuvieron como objetivo demostrar el fortalecimiento de las competencias matemáticas y el desarrollo del pensamiento computacional en 200 estudiantes entre los 7 y 10 años. Concluyeron que la aplicación de la estrategia pedagógica Scratch en la plataforma Moodle mejoró significativamente el desarrollo de las competencias matemáticas y el pensamiento computacional y la aplicación del curso denominado “Pensamiento computacional y resolución de problemas matemáticos con Scratch en Moodle” tuvo un efecto positivo, ya que generó en el estudiantado bastante motivación y entusiasmo.

Mantilla y Negre (2021), en su estudio titulado “Pensamiento computacional, una estrategia educativa en épocas de pandemia”, tuvieron como objetivo diagnosticar el nivel de formación de competencias del PC en 133 estudiantes de último grado en educación media de una institución pública de Colombia. Concluyeron que existe una relación representativa con una significancia de 0.000 entre el desarrollo de perspectivas computacionales (saber ser) con las prácticas y conceptos computacionales (Saber y saber hacer), ya que la práctica permite obtener un mejor desempeño en la toma de decisiones y evaluación de situaciones, a sí mismo afirman que la teoría fomenta la capacidad para tomar decisiones fundamentadas, administrando conocimiento.



Sáez et al. (2021) en su estudio titulado “aplicación de la robótica y programación por bloques (Scratch) en la enseñanza elemental”, tuvieron como objetivo evaluar la asimilación de conceptos computacionales a través de la programación visual en 107 estudiantes de quinto grado de educación primaria, concluyeron que la aplicación de Scratch a través de la interacción didáctica con los estudiantes logra mejoras estadísticamente significativas en la comprensión de conceptos computacionales como secuencias, bucles, paralelismo y manejo de eventos. Además, promueve en los estudiantes, motivación compromiso e interés en el proceso.

Daza y Lizarazo (2020) en su estudio titulado “Promover habilidades de pensamiento computacional en la era de la inequidad tecnológica” tuvieron como objetivo identificar habilidades de pensamiento computacional en 224 estudiantes de una institución educativa oficial de Colombia que desarrollaron ejercicios básicos de programación en Scratch. Concluyeron que el 42% de los estudiantes participantes desarrolla practicas computacionales y mas en concreto el indicador de “depurar” donde el grupo en general alcanzó mejor desempeño, por ello resaltan la necesidad de seguir fortaleciendo acciones didácticas y la formación de docentes.

Zarza y Holgado (2020), en su estudio titulado “Competencia del pensamiento computacional en la educación no formal”, tuvieron como objetivo determinar la correlación de dos variables: Participación en actividades de programación con Scratch en un entorno de educación no curricular y el desarrollo de la Competencia de Pensamiento Computacional desde la investigación Acción, para ello trabajó con una muestra de 9 educandos de quinto y sexto de primaria. Concluyeron en que los datos obtenidos demostraron una mejora en el grado de conocimiento de los contenidos computacionales, tanto conceptuales como procedimentales.



Caballero y García (2019), en su estudio titulado “Fortaleciendo habilidades de Pensamiento Computacional en Educación Infantil: Experiencia de aprendizaje mediante interfaces tangible y gráfica”, tuvieron como objetivo evaluar el desempeño de 44 estudiantes que participaron en el desarrollo de una experiencia de aprendizaje sobre habilidades de pensamiento computacional y programación utilizando interfaces tangibles y gráficas. Concluyeron que se generó un avance en referencia al aprendizaje del pensamiento computacional mediante la característica explorada. Además, se evidenció una aceptación positiva de las actividades entre estudiantes y profesores.

Monjelat (2019) en su estudio titulado “programación de tecnologías para la inclusión Social con Scratch: Prácticas sobre el pensamiento computacional” tuvieron como objetivo analizar las prácticas del pensamiento computacional desarrolladas por un grupo de 13 docentes en un taller de formación donde programaron con Scratch. Concluyeron que es posible desarrollar prácticas computacionales a través de una estrategia didáctica innovadora programando con Scratch en el marco de una formación docente significativa que no solo se centra en la apropiación de conceptos computacionales

Martinez y Gomez (2018) en su estudio titulado “programar computadoras en educación infantil” tuvieron como objetivo analizar una experiencia de enseñanza de la programación en un centro de educación infantil con 46 niños, para ello trabajaron una secuencia didáctica junto con las docentes que recibieron previamente una capacitación. Concluyeron que la programación con lenguaje icónico (flechas y dibujos) permite desarrollar prácticas computacionales, es decir estrategias propias de los programadores tales como depurar cuando revisaron el código, ensayar, e iterar en una actitud de cooperación entre los niños, los cuales representan procesos centrales del pensamiento computacional.



Sáez y Cózar (2017), en su estudio titulado “Pensamiento Computacional y programación visual por bloques en el aula de primaria” tuvieron como objetivo abordar el PC en Primaria desde una perspectiva lúdica y activa a través de lenguajes de programación visual utilizando Scratch con un enfoque pedagógico. Trabajó con un grupo experimental de 93 estudiantes que cursan 6° de Educación Primaria y un grupo control de 27 estudiantes. Concluyeron que el Scratch mejora significativamente la comprensión de los conceptos computacionales, tales como secuencias, bucles, paralelismos y eventos.

Álvares (2017), en su estudio titulado “Desarrollo del pensamiento computacional en educación primaria: una experiencia educativa con Scratch”, tuvo como objetivo evaluar la factibilidad del desarrollo del Pensamiento Computacional (PC) de alumnos de 6° de Primaria, que carecen de conocimientos computacionales previos al estudio, a través de proyectos realizados con la herramienta Scratch. Concluyeron que los sujetos han mostrado cierto desarrollo del PC sobre todo en lo referente a la adquisición de conceptos computacionales que establecen la base del PC, tales como la repetición de programas, las direcciones o las funciones simples.

Arranz y Pérez (2017) “Evaluación del Pensamiento Computacional en Educación” tuvieron como objetivo analizar el desarrollo del Pensamiento Computacional en 28 alumnos de Educación Primaria que han trabajado con el software de programación Scratch. Concluyeron que el Scratch afecta favorablemente al desarrollo de las dimensiones del Pensamiento Computacional, además son los conceptos computacionales los que mejor utilizan los estudiantes, y la práctica computacional de abstraer y modular no han sido desarrollado favorablemente por los estudiantes.

Así mismo en el ámbito Nacional se ha consultado los siguientes trabajos de investigación:



Churata et al. (2021) en su investigación titulada “Influencia del Scratch en el desarrollo del pensamiento computacional en los estudiantes del quinto de primaria de la Institución Educativa 40028 Alto Selva Alegre – Arequipa”, tuvieron como objetivo determinar la influencia del uso del Scratch en el desarrollo del Pensamiento Computacional en 24 estudiantes del quinto de Primaria. Concluyeron que el Scratch influye de manera positiva en el desarrollo del pensamiento computacional de los estudiantes de quinto grado de primaria y la aplicación de sesiones de aprendizaje como propuesta pedagógica reflejan un impacto positivo.

Araujo (2021) en su estudio titulado “Lenguaje de programación Scratch y pensamiento computacional en niños de 6 a 12 años, Institución Influence SAC. Los Olivos”, tuvo como objetivo determinar de qué manera el lenguaje de programación Scratch impacta en el pensamiento computacional en niños de 6 a 12 años en el distrito de los Olivos. Concluye que el lenguaje de programación Scratch influye en el pensamiento computacional de los estudiantes, además el dictado de los cursos de programación con Scratch tuvo un impacto positivo en los estudiantes.

Rosales (2019) en su investigación titulada “Pensamiento computacional y la resolución de problemas de matemática en estudiantes de primaria en Lima Cercado”, tuvo como objetivo establecer la relación entre el Pensamiento computacional y la resolución de problemas de matemática en 100 estudiantes de segundo grado de primaria de la I.E. Abraham Zea Carreón- 1150 en Lima Cercado. Concluye que existe una relación significativa entre el Pensamiento computacional y la Resolución de problemas de matemática en estudiantes de 2° grado de primaria, debido a $P < 0.05$ y Rho de Spearman de 0.553, estableciéndose moderada relación entre las variables.

Flores (2019), en su investigación titulada “Modelo holístico de código alfabetización en el desarrollo del pensamiento computacional en educación primaria”,



tuvo como objetivo determinar de qué manera el modelo holístico de código-alfabetización permite el desarrollo del pensamiento computacional en 311 niños de tercero a quinto grado de la I.E.P. “71013 Glorioso San Carlos” de la ciudad de Puno. Concluye que el modelo holístico de código-alfabetización construido en base a conceptos de sistemas y fundamentos pedagógicos y el prototipo de software aplicado, ha permitido el desarrollo del pensamiento computacional, en un nivel medio a alto, en los estudiantes, tal como se demuestra en la prueba de hipótesis Z normal con un nivel de significancia del 0,05.

Taco (2019), en su investigación titulada “Influencia del Programa Scratch en el Pensamiento Computacional en Estudiantes del Nivel Primario de la Institución Educativa de la Policía Nacional del Perú Alférez Mariano Santos Mateos, Tacna”, tuvo como objetivo determinar la influencia del programa Scratch en el pensamiento computacional de 384 estudiantes del nivel primario. Concluye que el programa Scratch influye significativamente sobre el pensamiento computacional de los estudiantes, demostrando de este modo la relevancia del programa como una fuente para mejorar las competencias de los estudiantes en el ámbito de la computación, así mismo destaca que el 100% de estudiantes del sexto grado alcanzaron una calificación alta en la evaluación de prácticas computacionales y perspectivas computacionales, mientras que en el quinto grado alcanzaron una calificación regular en ambas dimensiones.

Mamani (2018), en su investigación titulada “Pensamiento computacional en estudiantes del 4to grado de educación secundaria de la I.E. Morales Delgado de Pampa de Camarones Sachaca Arequipa”, tuvo como objetivo medir el pensamiento computacional que presentan 37 estudiantes del 4º grado de educación secundaria. Concluye que los estudiantes presentan un bajo nivel de pensamiento computacional. Además, el pensamiento computacional se evidencia más en varones que en mujeres. Por



ello, la propuesta de fortalecimiento del pensamiento computacional a abordar debe de poner énfasis las siguientes dimensiones: Condicionales while, condicionales simples, funciones simples, bucles y secuencias básicas o direcciones.

Condo (2017), en su investigación titulada “El pensamiento computacional en estudiantes del VII ciclo de la institución educativa particular “Ricardo Palma” - San Juan de Miraflores”, tuvo como objetivo precisar el grado de desarrollo del pensamiento computacional en 30 estudiantes del VII ciclo. Concluye que el 66.7% de los estudiantes alcanzan un nivel medio considerando que, si se estuviese desarrollando un software de programación dentro de la currícula, la tendencia sería más alta.

Chancolla y Pacori (2017), en su investigación titulada “El uso del software Scratch para mejorar el pensamiento computacional en los estudiantes del quinto grado de primaria de la Institución Educativa N° 40009 San Martín de Porres del distrito de Paucarpata, Arequipa” tuvo como objetivo desarrollar el pensamiento computacional a través del programa “Scratch” en 43 estudiantes de quinto grado. Concluyeron que la aplicación del programa “Scratch” en las sesiones de aprendizaje mejora el pensamiento computacional de los estudiantes de quinto grado de primaria. Además, reportaron una mejora en las puntuaciones del post test referente a los indicadores de secuencia, ciclos y paralelismo correspondientes a los indicadores de conceptos computacionales.



2.2. MARCO TEÓRICO

2.2.1. Software de programación Scratch

Según Pascual (2015), es un lenguaje de programación desarrollado por un grupo de investigación del Instituto de Tecnología de Massachusetts (MIT), su característica principal es su entorno de programación visual y multimedia que permite crear programas de forma fácil e intuitiva mediante una interfaz gráfica.

Scratch es la comunidad de programación para niños y niñas más grande del mundo, tiene una interfaz sencilla que promueve el pensamiento computacional y las habilidades en resolución de problemas. Además, está disponible en más de 70 idiomas, y diseñado especialmente para edades entre los 8 y 16 años (Scratch, 2020).

Resnick et al. (2009) definen a Scratch como el programa ideal para toda persona que desea aprender a programar, independiente de la edad que tenga, los conocimientos acumulados sobre esta materia. Afirman que el uso de tecnología para el beneficio de la sociedad no puede limitarse al uso de mensajes de texto, juegos en línea o navegar en internet, sino que debe llevar a cada individuo a generar soluciones innovadoras desde el punto de vista computacional, hacer uso de los medios digitales es como saber leer y crear nuevos medios digitales es como saber escribir.

López (2020) indica que la herramienta Scratch “hace que la programación sea más divertida para todo aquel que se enfrente por primera vez a aprender a programar” (p. 11). Además, considera que esta herramienta fue diseñada para expresar ideas de forma creativa con el objetivo de incentivar el desarrollo de habilidades de pensamiento computacional.



Según Santoyo (2016):

Es una herramienta gratuita que se puede descargar desde la página oficial de Scratch (<http://scratch.mit.edu>), es una aplicación diseñada para todos los sistemas operativos, su descarga e instalación es muy sencilla y su interfaz es extremadamente interactiva, pensada para que cualquier persona pueda empezar a programar. (p. 22)

En ese sentido, Flores y Maldonado (2007) manifiestan que el programa Scratch fue creado teniendo en cuenta las bases pedagógicas y principios teóricos del constructivismo y del construccionismo, con la finalidad de que el programa sea utilizado como una herramienta educativa, para ayudar a los estudiantes a desarrollar su creatividad, es decir, el estudiante deja de ser un simple consumidor a ser creador de su propio conocimiento, elevando así sus posibilidades de diseñar y crear programas con la computadora.

Es así que, Santoyo (2016) afirma que se prefiere utilizar Scratch porque permite programar de forma gráfica, por su portabilidad, su fácil manejo, la exigencia mínima de recursos tecnológicos, la creación de sus propios elementos multimediales y la posibilidad de trabajarlo de forma local hacen de este programa una herramienta que cumple las características principales para ser trabajada en las zonas rurales de nuestro país.

De acuerdo a todo lo mencionado, Scratch es un lenguaje de programación que permite programar de manera interactiva encajando bloques gráficos, sin tomar en cuenta la sintaxis y puntuación de los sistemas de programación tradicional, lo que permite que sea accesible y significativo para los niños.



2.2.1.1. Características del software de programación Scratch

Resnick et al. (2009) han introducido tres características básicas en el diseño de este lenguaje de programación, a continuación, se explica brevemente:

El lenguaje de programación debe ser lúdico: El interfaz del programa se proyecta como un escritorio real, busca la comodidad del usuario, se basa en la programación a través de bloques de diferentes colores, con conectores que permiten que se puedan encajar unos con otros.

La experiencia debe ser significativa: Prioriza el trabajo individual, promoviendo que cada usuario trabaje en el desarrollo de las soluciones personalmente significativos. De este modo en el diseño de Scratch, sus creadores, han dado prioridad a dos criterios del diseño: diversidad (que pueda soportar diferentes tipos de proyectos: historias, juegos, animaciones, simulaciones) y personalización (que los proyectos se puedan personalizar importando fotos, voces, gráficos, etc.)

Su uso debe propiciar la interacción social: El desarrollo de Scratch va muy unido al desarrollo de su página Web y necesita que una gran comunidad de personas comparta, apoye, critique, colabore y pueda construir sobre el trabajo de otros. Haciendo un clic sobre “compartir” los proyectos suben a la página Web de Scratch para ser compartidos y finalmente esto resulta una experiencia de aprendizaje interactiva y enriquecedora para todos.

2.2.1.2. Entorno del software de programación Scratch

Según Scratch (2020) se puede visualizar tres áreas principales que componen el entorno del software de programación Scratch:

Área de bloques

Scratch tiene bloques, que se arrastran hasta el área de programación. De modo que, se puede dar órdenes a los objetos para que efectúen distintas acciones.

Los bloques están divididos en nueve categorías:

Tabla 1

Bloques de órdenes de Scratch

Bloque	Descripción
<i>Movimiento</i>	Controlan la posición, dirección, rotación y movimiento de los objetos.
<i>Apariencia</i>	Controlan la apariencia de objetos y escenario
<i>Sonido</i>	Controlan la reproducción y volumen de notas musicales y archivos de audio
<i>Control</i>	Se utilizan para disparar la ejecución de los programas y cambiar la secuencia de los mismos.
<i>Lápiz</i>	Permite controlar el trazado en la pantalla como si trabajáramos con un lápiz.
<i>Sensores</i>	Permite detectar señales y reaccionar frente a ellas.
<i>Operadores</i>	Permite realizar operaciones y establecer relaciones.
<i>Variables</i>	Permite incluir elementos que varían a medida que avanza el programa.

Nota: Adaptado de Evangelista (2018)

Área de programación

Programas: Se arrastra las instrucciones de los bloques al área de programas. Para ejecutar un bloque se realiza doble clic.

Sonidos: Permite ver la selección de sonidos que se realizó para los objetos. Para ello se pueden grabar nuevos sonidos o importar archivos de sonidos que trae Scratch.

Disfraces: Permite ver y editar los disfraces del objeto. Además, se pueden crear nuevos disfraces dibujando, importando un archivo de imagen y tomando una fotografía para insertarlo en el escenario.

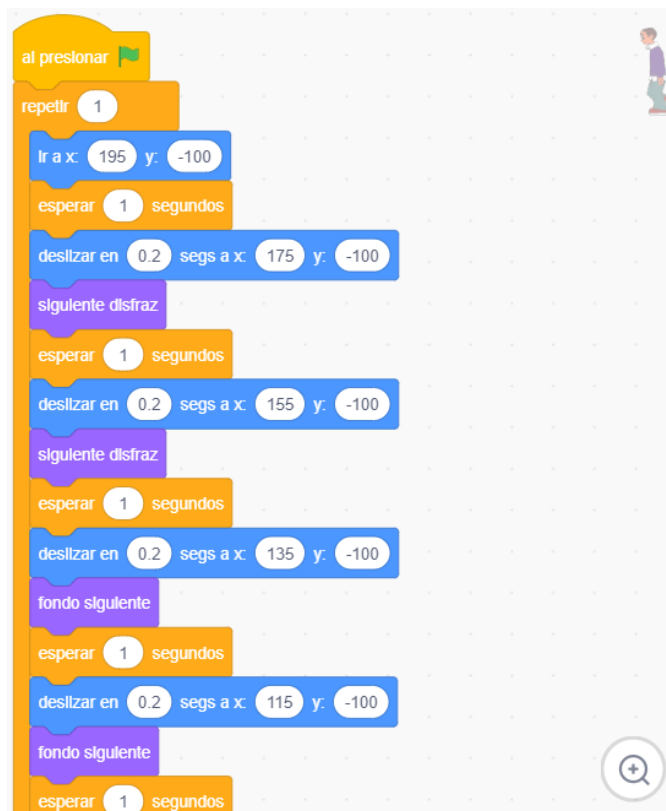


Figura 1. Área de programación Scratch.

Área de Escenario

Es el área dónde se puede visualizar cómo las historias, juegos y animaciones van cobrando vida. Además, es el área donde los objetos se mueven e interactúan unos con otros.



Figura 2. Área de escenario de Scratch.

En el escenario se puede observar también:

Herramientas: Permite duplicar los objetos, disminuir y aumentar su tamaño.

Bandera verde: Permite ejecutar los programas simultáneamente.

Detener: Permite finalizar toda la animación que se está ejecutando.

Objetos: Se puede importar e insertar objetos con los cuáles se desea trabajar.

Fondo de escenario: Se puede modificar y editar el fondo del escenario donde se realiza la animación de los objetos.

2.2.1.3. Dimensiones del software de programación Scratch

Según Santoyo (2016) existe cuatro dimensiones:



Dimensión de Proceso: Según Santoyo (2016) hace referencia a la ejecución de una secuencia de instrucciones, que junto a una correcta planificación y en un tiempo determinado permitirá crear con Scratch historias interactivas, juegos y animaciones. Esta dimensión tiene en cuenta los siguientes indicadores:

- **El análisis de Problema:** Formula el problema, procesos necesarios, restricciones, datos disponibles y resultados esperados.
- **El tiempo:** Utiliza productivamente el tiempo asignado para realizar el proyecto de una manera eficaz.
- **El trabajo colaborativo:** Los estudiantes al crear su proyecto lo comparten y lo publican, de esta manera queda al alcance de sus compañeros, los cuales podrán realizar modificaciones a su gusto, mejorando el Proyecto inicial. Así mismo colabora con sus compañeros, incluso en la depuración del programa.

Dimensión de Funcionamiento: Según Santoyo (2016) hace referencia al funcionamiento correcto de los proyectos y si cumple con los criterios planteados por el docente. Esta dimensión tiene en cuenta los siguientes indicadores:

- **Funcionamiento:** El programa realizado funciona correctamente.
- **Cumple Criterios:** Cumple con lo planteado por el docente en su proyecto realizado en clase.

Dimensión de Interfaz Gráfica: Según Santoyo (2016) es el diseño de interfaces de usuarios interactivas con Scratch, que se diseñan teniendo en cuenta niveles de complejidad, los cuales presentan claridad y fácil interacción con el usuario final. Esta dimensión debe estar conformado por los siguientes indicadores:

- **Organizado:** El programa realizado esta organizado, tiene varios niveles y su diseño es complejo.



- ***Coherente:*** La interfaz gráfica es clara, tiene estructura y se adapta tanto al contenido como al diseño del programa.
- ***Interactivo:*** Es fácil de interactuar con el programa.

Dimensión de Programación: Según Santoyo (2016) se refiere al uso de los elementos y lógica de programación. En esta dimensión se tiene en cuenta los siguientes indicadores:

- ***Dominio de Lógica:*** El programa evidencia comprensión avanzada de bloques y procedimientos.
- ***Manejo de estructuras:*** Utiliza apropiadamente las estructuras de control como secuencial, condicional e iterativa.
- ***Programación Organizada:*** Los hilos de programación son lógicos y están bien organizados.
- ***Depuración:*** Depurar significa identificar y corregir los errores que pueda contener un proyecto. Es una de las partes más complejas de la programación. Si corre el programa está correctamente depurado.



2.2.2. Pensamiento computacional

La definición formal del pensamiento computacional fue introducida por primera vez dentro de la comunidad científica en computación, por Wing (2006), quién define el término como una forma de pensar que no se limita en exclusiva hacia programadores ni científicos en computación, sino como un conjunto de habilidades útiles para todas las personas, que implica resolver los problemas, diseñar sistemas y entender el comportamiento humano, aprovechando los conceptos fundamentales para las ciencias informáticas.

Otro concepto que proviene del mundo computacional es el señalado por Raja (2014) que indica “el enfoque computacional se basa en ver el mundo como una serie de puzzles, a los que se puede romper en trozos más pequeños y resolver poco a poco a través de la lógica y el razonamiento deductivo” (p. 1), es así que, el enfoque computacional es una forma intuitiva de abordar varios de los métodos existentes de la psicología del aprendizaje.

“El Pensamiento Computacional es un proceso mental utilizado para formular problemas y sus soluciones, de forma que las soluciones se representan en una forma que puede ser llevada a cabo por un agente de proceso de información” (Cuny et al., como se citó en Wing, 2010, p. 1)

Así mismo, Denning (2017) afirma que:

El pensamiento computacional es el proceso de pensamiento involucrado en la formulación de los problemas de tal manera que sus soluciones se presentan como pasos computacionales y algoritmos que pueden ser efectivamente llevados a cabo por un agente de procesamiento de la información. (párr. 11)



Vera et al. (2015) agregan que:

El Pensamiento Computacional permite que las personas puedan cultivar la capacidad de resolución de problemas, haciendo abstracciones y división de los problemas en otros de menor complejidad para plantear la mejor solución; es aplicada en diferentes áreas del conocimiento; siendo más aplicado en unas áreas que en otras, con base en los conceptos de la computación, tales como abstracción, algoritmos, programación, entre otros. (p. 730)

Román (2016) afirma que, el pensamiento computacional sería un término de corte cognitivo; que se puede definir como la capacidad de formular y solucionar problemas apoyándose en conceptos computacionales (secuencias, bucles, condicionales, funciones, variables), y siguiendo la lógica algorítmica inherente a los lenguajes informáticos de programación.

“El pensamiento computacional es una forma de pensar para resolver problemas de manera efectiva y eficiente (es decir, algorítmicamente, con o sin la ayuda de computadoras) con soluciones que son reutilizables en diferentes contextos” (Shute et al., 2017, p. 144)

Además, Code learn (2020) define el pensamiento computacional como un proceso mental que lleva al individuo a buscar soluciones óptimas, eficientes y abiertas. No solamente aplicable al mundo informático.

Según Basogain (2015), es un método comprobado por investigadores de nivel, fácil de enseñar, divertido de aprender, y debe incorporarse a la educación para que los niños se conviertan en pensadores computacionales que resuelven problemas con métodos innovadores, siendo más creadores que consumidores.



Wing (2006) describe una serie de rasgos, los cuáles pueden ser muy útiles para establecer un corpus curricular para el aprendizaje basado en el pensamiento computacional:

En el pensamiento computacional son fundamentales las habilidades no memorísticas o no mecánicas: Memoria significa mecánico, aburrido, rutinario. Para programar los computadores hace falta una mente imaginativa e inteligente.

En el pensamiento computacional lo importante son las ideas, no los artefactos. Quedan descartados por tanto la fascinación y los espejismos por las novedades tecnológicas. Y mucho menos estos factores como elementos determinantes de la resolución de problemas o de la elección de caminos para resolverlos.

Según UCSP (2021):

El pensamiento computacional incluye 4 habilidades principales

- (1) Descomposición: consiste en el procedimiento por el cual un problema de mayor complejidad se desarticula en pequeñas series más manejables.
- (2) Reconocimiento de patrones: luego de la desarticulación del problema complejo, las pequeñas series son enfrentadas de forma individual de manera que puedan ser resueltas de forma similar a problemas frecuentados anteriormente.
- (3) Abstracción: Consiste en la omisión de información irrelevante al problema propuesto.
- (4) Algoritmos: se presentan pasos para la resolución de cada problema. (Párr. 2)



2.2.2.1. Dimensiones del pensamiento computacional

Brenan y Resnick (2012) señalan que, el pensamiento computacional está basado en el lenguaje de programación visual Scratch, como un entorno que permite a los estudiantes crear juegos o simulaciones empleando los conceptos de programación de una forma más creativa y divertida y las experiencias de talleres con Scratch y su comunidad en línea, desarrollaron una definición de Pensamiento Computacional que incluye tres dimensiones clave: Conceptos computacionales, prácticas computacionales y perspectivas computacionales:

Conceptos computacionales: Según Brenan y Resnick (2012) hace referencia a los conceptos de los que deben ocuparse los diseñadores a medida que programan. Sus indicadores son los siguientes:

- **Secuencias:** Un concepto clave en programación, es que una tarea o actividad particular se expresa como una serie de pasos o de instrucciones individuales, que puede ejecutar el computador. Tal como en una receta, la secuencia de instrucciones de programación indica el comportamiento o acción que se debe producir.
- **Ciclos:** Son mecanismos que ejecutan la misma secuencia, múltiples veces.
- **Paralelismo:** Los lenguajes de computador más modernos soportan paralelismo; esto es, secuencias de instrucciones que se realizan simultáneamente. Scratch permite paralelismo entre objetos. Por ejemplo, la escena del baile de una fiesta en Scratch puede incluir varios personajes que bailan simultáneamente, cada uno con su propia secuencia de instrucciones de baile. Scratch permite también paralelismo dentro del mismo objeto.



Prácticas computacionales: Según Brenan y Resnick (2012) hace referencia a las prácticas que los diseñadores desarrollan a medida que se ocupan de los conceptos; tales como, depuración de proyectos o remezclas. Sus indicadores son los siguientes:

- **Ensayo y Depuración:** Muy rara vez, las cosas funcionan como se habían imaginado; por lo que es crítico para los diseñadores desarrollar estrategias para manejar y anticipar problemas. En las entrevistas, los Scratchers describen las diversas prácticas de ensayo y depuración que han desarrollado, ya sea por ensayo y error, transfiriéndolas de otras actividades.
- **Abstraer y Modularizar:** Se puede caracterizar como construir algo grande uniendo colecciones de partes más pequeñas, es una práctica siempre importante para el diseño y la solución de problemas. En Scratch, los diseñadores emplean la abstracción y la modularización en múltiples niveles, desde el trabajo inicial de conceptualizar el problema hasta trasladar el concepto a los objetos individuales y a las pilas de código.

Perspectivas computacionales: Según Brenan y Resnick (2012) hace referencia a las perspectivas que los diseñadores construyen sobre el mundo que los rodea y sobre ellos mismos: Sus indicadores son los siguientes:

- **Expresar:** Las personas están rodeadas de medios interactivos, pero la mayor parte de nuestras experiencias con estos, es como consumidores. Invertimos tiempo en señalar, hacer clic, buscar y chatear, actividades estas importantes para aprender a usar las Tecnologías de la Información y la Comunicación (TIC), pero no suficientes para desarrollarnos como pensadores computacionales.



- *Un pensador computacional ve la computación como algo más que un producto de consumo; la computación es algo que ellos pueden usar para diseñar y auto expresarse. Un pensador computacional ve la computación como medio y piensa Yo puedo crear y expresar mis ideas usando este nuevo medio.*
- **Conectar:** La creatividad y el aprendizaje son prácticas profundamente sociales, por lo que no sorprende que diseñar medios computacionales con Scratch se enriquezca mucho mediante la interacción con otros.
- **Preguntar:** Con la perspectiva computacional de preguntar, buscamos indicadores de que los jóvenes no sientan esa desconexión entre las tecnologías que los rodean y sus habilidades para negociar las realidades del mundo tecnológico.

2.3. MARCO CONCEPTUAL

Programa Scratch

Es un lenguaje de programación que facilita crear historias interactivas, juegos y animaciones y compartir sus creaciones con otras personas en la Web. Es un entorno de programación visual y multimedia basado en Squeak destinado a la realización y difusión de secuencias animadas con o sin sonido y al aprendizaje de programación” (Cano & Delgado, 2015, p. 66)

Pensamiento Computacional

"El Pensamiento Computacional son los procesos de pensamiento implicados en la formulación de problemas y sus soluciones para que estas últimas estén representadas de forma que puedan llevarse a cabo de manera efectiva por un procesador de información" (Wing, 2010, pág. 1)



Alfabetización digital

Se entiende como las acciones educativas desarrolladas para capacitar en el manejo crítico, efectivo y eficaz de las herramientas digitales y su apropiación en contextos determinados que permitan al alfabetizado solucionar problemas cotidianos y mejorar su desempeño en el uso de Tecnologías de Información y Comunicación (TIC). (Lamothe et al., 2020, p. 196)

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. UBICACIÓN GEOGRÁFICA DEL ESTUDIO

El presente trabajo de investigación se realizó en el centro poblado de Caritamaya, distrito de Acora, provincia y departamento de Puno. La Institución Educativa Primaria N° 70116 de Caritamaya se ubica en el mismo centro poblado, en la Carretera Panamericana Km 38 Puno – Ilave.

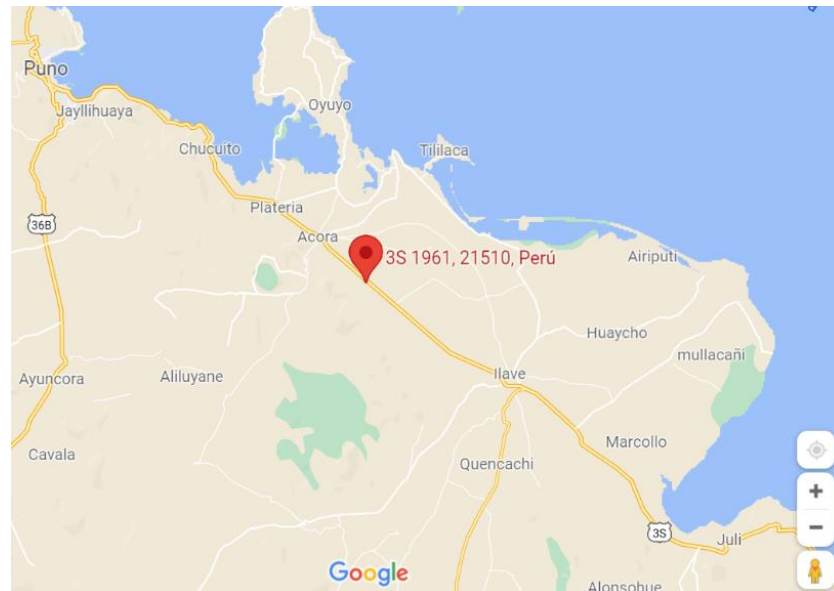


Figura 3. Ubicación Satelital del Centro Poblado de Caritamaya.

Localización:

Latitud: -16.00980592000

Longitud: -69.73470589000)

Altitud: 3832 m s. n. m.

3.2. PERIODO DE DURACIÓN DEL ESTUDIO

La investigación inició en el año 2020, para la ejecución del proyecto y recolección de datos se planteó una propuesta pedagógica denominada “Aprendemos a



programar con Scratch”, la cual constó de 10 sesiones de aprendizaje con sus respectivas guías prácticas, para ello se ocupó dos semanas más dos días adicionales en los cuales se aplicó el pre test y post test. De esta manera, se procedió al análisis e interpretación de los resultados obtenidos para posteriormente presentar el informe final del trabajo de investigación realizado.

3.3. PROCEDENCIA DEL MATERIAL UTILIZADO

Para la recopilación de datos de la investigación se ha utilizado el instrumento denominado "Test de Pensamiento Computacional”, el cuál fue creado por el Dr. Marcos Román Gonzales del departamento de Métodos de Investigación y Diagnóstico en Educación I (MIDE I) de la facultad de educación de la Universidad Nacional de Educación a Distancia (UNED).

El instrumento ha sido debidamente validado con 20 expertos, quiénes colaboraron con la emisión de juicio y valoración. Obteniendo como resultados: +70% de los expertos valoraron la versión inicial como “muy larga” o “algo larga” debido a ello se determinó que la longitud ideal sería 28 – 30 ítems y el tiempo ideal de 45 – 50 minutos, y de esta manera surge la versión 2.0 del TPC. Además, la fiabilidad, como consistencia interna, del TPC en su conjunto arroja un valor de $\alpha = 0,74$ que puede considerarse un valor aceptable. Según Román & Pérez (2015) la fiabilidad del TPC es suficiente para ser una prueba de aptitud. A continuación, se menciona los artículos que avalan su fiabilidad y validez, los cuales fueron facilitados por el mismo autor junto a una carta de autorización para utilizar el instrumento en la presente investigación:

- “Computational thinking test: design guidelines and content validation” Román, (2015), en dicho artículo se encuentra la validez de contenido por juicio de expertos.



- “Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test” Román, Pérez, & Jiménez (2017) en este artículo se encuentra la fiabilidad y validez criterial del instrumento.
- “Can computational talent be detected? Predictive validity of the Computational Thinking Test” Román, Pérez, & Moreno (2018) en este artículo se encuentra la validez predictiva del test de pensamiento computacional.

A su vez, el instrumento ha sido utilizado, validado y adaptado en Perú en los trabajos de investigación de Chancolla y Pacori (2017) y Taco (2019)

3.4. POBLACIÓN Y MUESTRA DEL ESTUDIO

La población de estudio estuvo conformada por los estudiantes de la Institución Educativa Primaria N° 70 116 Caritamaya, cuyo número según las nóminas de matrícula del 2020 es de 52 estudiantes, todos ellos cursan estudios de Educación Básica Regular.

Flores (2011) afirma que una población “es el conjunto de unidades de observación que posee características específicas determinadas por el investigador según los objetivos que persigue”. (p. 123) Y añade que “el número de elementos de una población no es el factor clave para la obtención de una muestra representativa (p. 212)”



Tabla 2

Población de estudio

Grado	Número de estudiantes
1er grado	7
2do grado	6
3er grado	13
4to grado	5
5to grado	13
6to grado	8
Total	52

Nota: De Nóminas de Matrícula 2020 de la Institución Educativa Primaria 70 116 Caritamaya

Mejía (2013) expresa que “no existe un criterio definido acerca del tamaño adecuado que debe tener la muestra. La experiencia del investigador, los recursos que dispone o las facilidades técnicas que se le presenten, serán los determinantes para fijar el tamaño de su muestra” (p. 173).

En ese sentido, a través del muestreo no probabilístico intencional se determinó como muestra a un total de 13 estudiantes que pertenecen al quinto grado, los mismos que conformaron el grupo experimental.

Tabla 3

Muestra de estudio

Estudiantes de 5to grado	Número de estudiantes
Varones	6
Mujeres	7
Total	13

Nota: De Nóminas de Matrícula 2020 del quinto grado de la Institución Educativa Primaria 70 116 Caritamaya.



3.5. DISEÑO ESTADÍSTICO

La información obtenida de los sujetos de investigación fue procesada utilizando el software estadístico SPSS versión 25, programa ideal para el análisis estadístico en investigaciones de ciencias sociales, la información de los instrumentos aplicados fue codificada adecuadamente para ser sometida a un análisis descriptivo y posteriormente a un análisis explicativo, el cual tuvo como finalidad definir la eficacia del software de programación Scratch. Los resultados se presentan mediante tablas de frecuencia y gráficos de barra.

Prueba estadística

Para probar la hipótesis se usó la prueba estadística t de Student para muestras relacionadas, la cual permitió comparar las medias de dos series de mediciones realizadas sobre las mismas unidades estadísticas. Se estructura de la siguiente manera:

a. Establecimiento de hipótesis estadística

Ha: La aplicación del Software de programación “Scratch” es eficaz en el desarrollo del pensamiento computacional de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya 2020.

Ho: La aplicación del Software de programación “Scratch” no es eficaz en el desarrollo del pensamiento computacional de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya 2020.

b. Nivel de significancia

El nivel de significancia o error que se asume es $= 0.05$, que es igual al 5% margen de error, con un nivel de confianza del 95%.

c. Prueba estadística a usar:

$$t = \frac{\bar{x} - \mu}{\frac{S_x}{\sqrt{\eta}}}$$

Grados de libertad para el caso es $G_l = n - 1$

Sabiendo que:

\bar{x} = Media

μ = Valor a analizar

s_x = Desviación estandar

η = Tamaño de muestra

d. Toma de decisión y conclusión

Regla de decisión:

Si $P < 0.05$, entonces se rechaza la hipótesis nula

Si $T_c > T_t$, entonces se acepta la hipótesis alterna

3.6. PROCEDIMIENTO

El experimento se realizó de la siguiente manera:

Primero. Se realizó el análisis de las dimensiones y los indicadores de la variable dependiente para seleccionar el instrumento de investigación y validarlo por juicio de expertos.

Segundo. Se presentó una solicitud al director de la Institución Educativa Primaria N° 70116 de Caritamaya, con la finalidad de que se autorice la realización de la investigación. Además, se solicitó el préstamo de equipos para aplicar las sesiones con el software de programación Scratch.



Tercero. Se coordinó con el docente titular y los padres de familia del quinto grado para realizar la investigación y aplicar las sesiones de la propuesta pedagógica. Así mismo, los padres de familia firmaron una carta de consentimiento para autorizar la ejecución del proyecto.

Cuarto. Se aplicó a los estudiantes el test de pensamiento computacional (Pre test) para obtener información sobre el desarrollo del pensamiento computacional.

Quinto. Se aplicó el software de programación Scratch a través de 10 sesiones de aprendizaje, las cuales se encuentran en la propuesta pedagógica.

Sexto. Se aplicó el test de pensamiento computacional (post test), para poder comprobar los resultados; se ubicaron también los datos en las tablas de frecuencia para analizarlos e interpretarlos.

Séptimo. Se analizó e interpretó los resultados, finalmente se comprobó la eficacia de la aplicación del software de programación Scratch en el desarrollo del pensamiento computacional.



3.7. VARIABLES

Variable independiente:

Software de programación Scratch

Tabla 4

Dimensiones del software de programación Scratch

DIMENSIONES	INDICADORES
<i>Dimensión de proceso</i>	Análisis del problema
	Tiempo
	Trabajo colaborativo
<i>Dimensión de funcionamiento</i>	Funcionamiento
	Cumple criterios
<i>Dimensión de interfaz gráfica</i>	Organizado
	Coherente
	Interactivo
<i>Dimensión de programación</i>	Dominio de lógica
	Programación organizada
	Depuración

Nota: De elaboración propia.



Variable dependiente:

Pensamiento computacional

Tabla 5

Dimensiones del pensamiento computacional

DIMENSIONES	INDICADORES	ÍTEMS
Conceptos computacionales	Secuencias: Analiza y reconoce la secuencia correcta haciendo uso de las variables presentadas y utiliza la direccionalidad para llegar al objeto deseado.	1, 4 y 5
	Ciclos: Reconoce el bucle “repetir veces” y “repetir hasta”	6, 8, 12, 13, 14 y 23
	Paralelismo: Selecciona la opción correcta identificando el condicional mientras	15, 21 22 y 24
Prácticas computacionales	Ensayo y depuración: Analiza y reconoce el error de la secuencia.	7, 11 y 25
	Abstraer y modularizar: Utiliza la abstracción y modularización para la depuración	3, 16 y 19
Perspectivas computacionales	Expresar: Resuelve un problema complejo mediante la descomposición	2, 26, 27 y 28
	Conectar: Identifica y completa la condición que falta en el ciclo repetir hasta.	9, 10 y 20
	Preguntar: Elige el orden correcto identificando las señales de peligro.	17 y 18

Nota: De elaboración propia.



3.8. ANÁLISIS DE LOS RESULTADOS

De manera inicial para el procesamiento de datos se tuvo que realizar una base de datos para el pre test y otra para el post test, posteriormente se identificaron los niveles que obtuvo cada participante en el pre y post test, de manera general y por objetivos específicos, se realizaron las tablas y gráficos correspondientes, todo este procedimiento inicial se realizó con ayuda del programa Microsoft Excel. Para identificar el nivel de eficacia del software de programación Scratch fue empleada la T de student, debido a que los resultados fueron sometidos a prueba de normalidad, con la ayuda de estadístico Shapiro-Wilk porque la muestra era inferior a 50, y se obtuvo como resultado que los datos tenían distribución normal.

3.9. METODOLOGÍA DE LA INVESTIGACIÓN

Tipo de investigación

De acuerdo con el propósito de la investigación, este estudio es de tipo aplicativo. Según Hernández, et al. (2010) “la investigación de tipo o nivel aplicativo se caracteriza porque se preparan las condiciones de investigación y se manipula la variable independiente” (p. 117). Así mismo, Carrasco (2005) manifiesta que este tipo de investigación se distingue por tener propósitos inmediatos bien definidos, es decir, se investiga para actuar, transformar, modificar o producir cambios en un determinado sector de la realidad. (p. 43)

Por su naturaleza, viene a ser un estudio de enfoque cuantitativo. Según Hernández et al. (2010) este tipo de estudio utiliza la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin de establecer pautas de comportamiento y probar teorías.

Diseño de investigación

Teniendo en consideración las características de la muestra, el diseño de investigación se ha definido como experimental de tipo pre experimental, con corte longitudinal. Es decir, solo cuenta con un grupo experimental, en este diseño se aplica una prueba Pre – test al inicio de la investigación y una prueba Post – test luego del tratamiento que se aplica sobre la muestra; de tal manera que, para la comprobación de la hipótesis se compara los resultados de ambas pruebas. Así mismo, sobre el corte longitudinal, Hernández et al. (2010) señalan: Es longitudinal cuando el interés del investigador es analizar cambios a través del tiempo en determinadas variables o en relaciones entre estas.

Tabla 6

Diseño de la investigación

Grupo	Aplicación Del Pre-Test	Aplicación Del Tratamiento	Aplicación Del Post-Test
Ge	O1	X	O2

Nota: De “el proyecto de investigación” por Arias (2012).

Técnicas e instrumentos

La recolección de datos permitió establecer la eficacia del software de programación “Scratch” en el desarrollo del pensamiento computacional, es decir, medir para posteriormente interpretar los resultados. Para ello, se empleó la técnica de la evaluación y como instrumento una prueba estandarizada que consta de 28 preguntas cerradas, con 4 alternativas.



En ese sentido, se asume que la puntuación total es de 28 puntos y cada ítem contiene un criterio numérico para el análisis estadístico: Sí el estudiante responde correctamente, se le asigna el dígito 1 y en el caso que su respuesta sea incorrecta le corresponde el dígito 0.

Validez: Para aplicar el instrumento al contexto concreto de la presente investigación se realizó la validación por dos juicios de experto: Dr. Alfredo Carlos Castro Quispe y Dr. Heber Nehemias Chui Betancur.

De acuerdo al juicio de valoración del Dr. Alfredo Carlos Castro Quispe, se obtuvo como promedio final 17, por lo que el instrumento se considera APTO para su aplicación.

De igual manera, con respecto al juicio de valoración del Dr. Heber Nehemias Chui Betancur, se obtuvo como promedio final 17, por lo que el instrumento se considera APTO para su aplicación.

Confiabilidad: El instrumento ha sido sometido a la prueba alfa de Cronbach, con la finalidad de establecer confiabilidad a los resultados obtenidos y de esta manera puedan ser tomados como referencia para posteriores trabajos de investigación, siendo estos los resultados:

Tabla 7

Estadísticos de fiabilidad

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,724	28

Nota: De SPSS Statistics v25

De este resultado observado, se asume que el instrumento es aceptable según los valores que consideran George & Mallery (2003, pág. 231)

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

Objetivo específico 1

Determinar el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de conceptos computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Tabla 8

Resultados del Pre Test y Post Test sobre el desarrollo de conceptos computacionales en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.

<i>Nivel</i>	<i>Pre Test</i>		<i>Post Test</i>	
	<i>F</i>	<i>%</i>	<i>F</i>	<i>%</i>
Básico	9	69%	1	7,7%
Medio	4	31%	11	84,6%
Alto	0	0%	1	7,7%
<i>Total</i>	13	100%	13	100%

Nota: De base de datos de la aplicación del instrumento de Pensamiento Computacional.

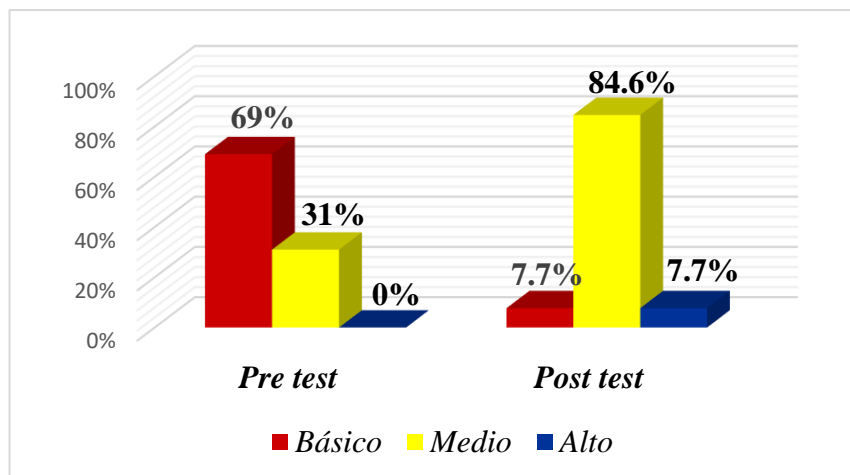


Figura 4. Resultados del Pre Test y Post Test sobre el desarrollo de conceptos computacionales en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.



En la **tabla 8** y **figura 4**, se pueden observar los resultados que obtuvieron los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya, se puede apreciar que, durante el pre test, 9 estudiantes manifiestan un nivel Básico lo que equivale a un 69%, en el nivel Medio se ubicaron 4 estudiantes que conforman un 31%. Ninguno se encuentra en el nivel Alto.

En el post test, 11 estudiantes manifiestan un nivel Medio lo que equivale a un 84.6%. En el nivel Alto se ubicó sólo un estudiante que representa un 7.7%. Finalmente, solo un estudiante se encuentra en el nivel básico, lo que representa un 7.7% del total.

Estos resultados indican que la propuesta pedagógica desarrollada con los estudiantes de quinto grado de la Institución Educativa Primaria N° 70 116 aplicando el software de programación Scratch tuvo eficacia en el desarrollo de Conceptos Computacionales, lo cual evidencia que los estudiantes utilizan las direcciones para llevar al pac-man al objeto deseado, identifican las ordenes que llevan al artista para dibujar el cuadrado, analizan y seleccionan la opción correcta para llegar hacia el fantasma, haciendo uso de la direccionalidad y reconociendo el ciclo “repetir veces”. Además, identifican la secuencia correcta del ciclo “repetir veces” utilizando su sentido de la reflexión. Eligieron el orden correcto, identificando las señales de peligro, puesto que tomaron decisiones basados en la situación presentada y eligen la secuencia correcta del ciclo “repetir hasta” reconociendo las señales de peligro.

Tabla 9

Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de conceptos computacionales

Pruebas	Medidas De Tendencia Central		
	X	S	S ²
Pre Test	3,85	1,34	1,81
Post Test	6,77	1,96	3,86
Diferencia	2,92	0,62	2,05

Nota: De elaboración propia

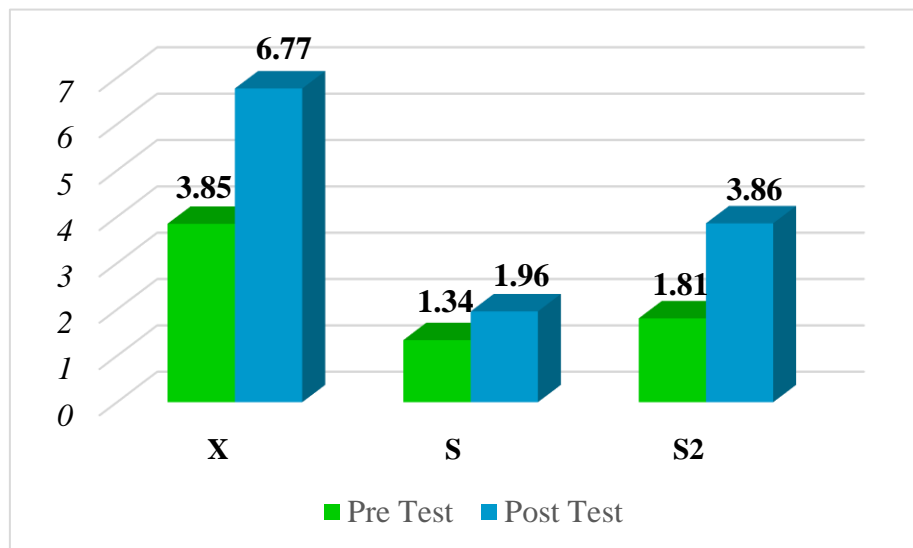


Figura 5. Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de conceptos computacionales.

De la **tabla 9** y **figura 5**, en las comparaciones diferenciadas entre las medidas de tendencia central y dispersión del Pre y Post test sobre el desarrollo de conceptos computacionales, se observa que en el Pre-Test la media aritmética es de 3.85 puntos y por otro lado la media aritmética del post test es de 6.77 puntos, habiendo una diferencia de 2.92 puntos entre ambos promedios, y se demuestra que antes de la aplicación del Scratch, la media fue menor con relación al post test.

Contrastación de la hipótesis específica 1

Ha: La aplicación del software de programación “Scratch” es significativamente eficaz en el desarrollo de conceptos computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Ho: La aplicación del software de programación “Scratch” no es eficaz en el desarrollo de conceptos computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Regla de decisión: Si $P < 0.05$, entonces se rechaza la hipótesis nula.

Tabla 10

Prueba t de Student de muestras emparejadas para el objetivo específico 1

	<i>t</i>	<i>gl</i>	<i>Sig. (Bilateral)</i>
<i>Pre test y post test sobre el desarrollo de conceptos computacionales</i>	6,178	12	,000

Nota: De elaboración propia en SPSS v25

En la **tabla 10**, se observa que $P < 0.05$, de esta manera se rechaza la hipótesis nula y se acepta la hipótesis alterna, por ello se concluye que la aplicación del Software de programación “Scratch” es significativamente eficaz en el desarrollo de conceptos computacionales de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya 2020.

Objetivo específico 2

Determinar el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de prácticas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Tabla 11

Resultados del Pre Test y Post Test sobre el desarrollo de prácticas computacionales en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.

Nivel	Pre Test		Post Test	
	F	%	F	%
Básico	10	77%	7	54%
Medio	3	23%	5	38%
Alto	0	0%	1	8%
<i>Total</i>	13	100%	13	100%

Nota: De base de datos de la aplicación del instrumento del Pensamiento Computacional

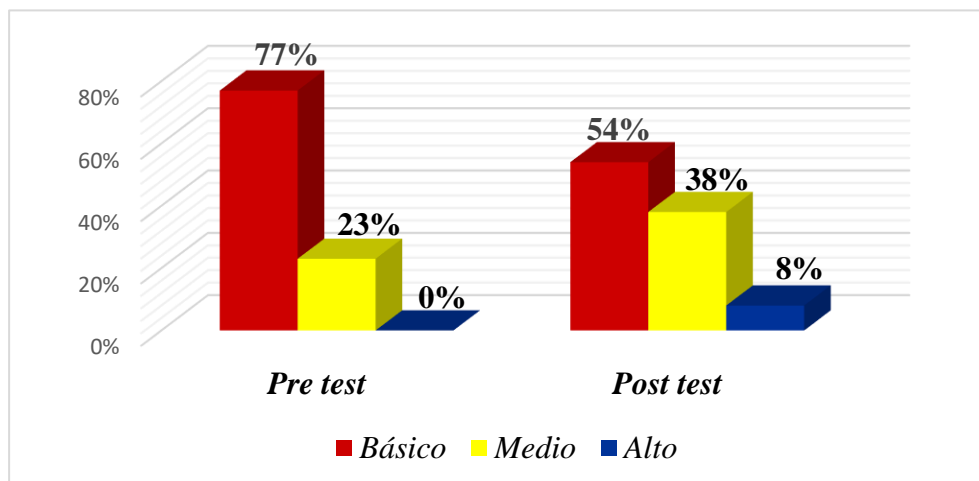


Figura 6. Resultados del Pre Test y Post Test sobre el desarrollo de la dimensión prácticas computacionales en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.



En la **tabla 11** y **figura 6**, se pueden observar los resultados que obtuvieron los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya, se puede apreciar que, durante el pre test, 10 estudiantes manifiestan un nivel Básico lo que equivale a un 77%, en el nivel Medio se ubicaron 3 estudiantes que conforman un 23%. Ninguno se encuentra en el nivel Alto.

En el post test, 7 estudiantes manifiestan un nivel Básico lo que equivale a un 54%, en el nivel Medio se ubicaron 5 estudiantes que representan un 38%. Finalmente, solo un estudiante se encuentra en el nivel Alto, lo que representa un 8% del total.

Estos resultados indican que la propuesta pedagógica desarrollada con los estudiantes de quinto grado de la Institución Educativa Primaria N° 70 116 aplicando el software de programación Scratch tuvo eficacia en el desarrollo de Prácticas Computacionales, aunque un 54% de estudiantes aún se encuentran en nivel básico esto es debido a que no lograron mejorar la abstracción y modularización para poder depurar. Mientras que los demás niños que se encuentran en nivel medio y alto, analizaron y reconocieron el error en las secuencias presentadas utilizando la direccionalidad, además encontraron el error en la secuencia analizando el proceso mediante la abstracción y modularización. Además, analizaron el proceso del todo hacia lo específico y así encontrando el error.

Tabla 12

Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de prácticas computacionales.

Pruebas	Medidas de tendencia central		
	X	S	S ²
Pre Test	2,15	1,28	1,64
Post Test	3,54	1,33	1,77
Diferencia	1,39	0,05	0,13

Nota: De elaboración propia

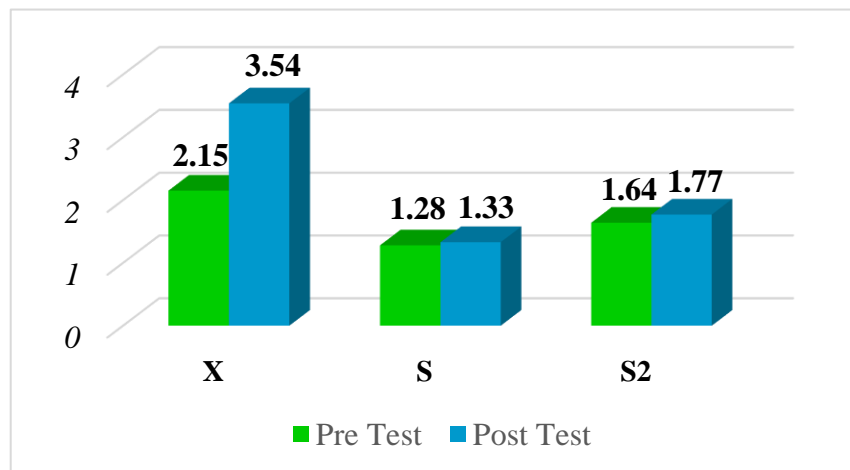


Figura 7. Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de prácticas computacionales.

De la **tabla 12** y **figura 7**, en las comparaciones diferenciadas entre las medidas de tendencia central y dispersión del Pre Test y el Post Test sobre el desarrollo de prácticas computacionales, se observa que en el Pre Test la media aritmética es de 2.15 puntos y por otro lado la media aritmética del Post Test es de 3.54 puntos, habiendo una diferencia de 1.39 puntos entre ambos promedios, lo que demuestra que antes de la aplicación del Scratch, la media fue menor con relación al post test.

Contrastación de la hipótesis específica 2

Ha: La aplicación del software de programación “Scratch” es significativamente eficaz en el desarrollo de prácticas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Ho: La aplicación del software de programación “Scratch” no es eficaz en el desarrollo de prácticas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Regla de decisión: Si $P < 0.05$, entonces se rechaza la hipótesis nula

Tabla 13

Prueba t de Student de muestras emparejadas para el objetivo específico 2

	<i>t</i>	<i>gl</i>	<i>Sig. (Bilateral)</i>
<i>Pre test y post test sobre el desarrollo de prácticas computacionales</i>	8,451	12	,000

Nota: De elaboración propia. SPSS v25

En la **tabla 13**, se observa que $P < 0.05$, de esta manera se rechaza la hipótesis nula y se acepta la hipótesis alterna, por ello se concluye que la aplicación del Software de programación “Scratch” es significativamente eficaz en el desarrollo de las prácticas computacionales de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya 2020.

Objetivo específico 3

Determinar el nivel de eficacia de la aplicación del software de programación “Scratch” en el desarrollo de perspectivas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Tabla 14

Resultados del Pre Test y Post Test sobre el desarrollo de Perspectivas

Computacionales en estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.

Nivel	Pre Proyecto		Post proyecto	
	F	%	F	%
Básico	10	77%	1	7.7%
Medio	3	23%	11	84.6%
Alto	0	0%	1	7.7%
<i>Total</i>	13	100%	13	100%

Nota: De base de datos de la aplicación de instrumento del pensamiento computacional

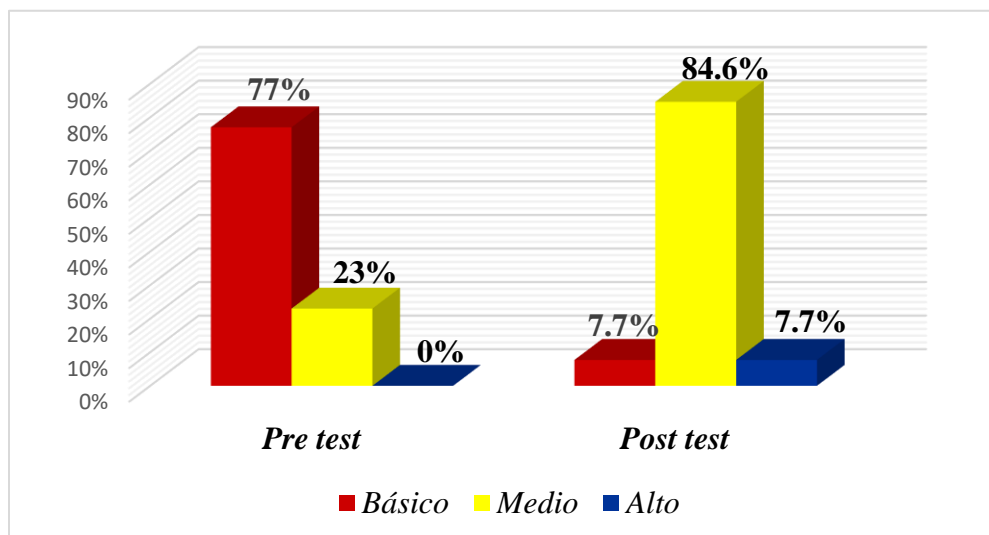


Figura 8. Resultados del Pre Test y Post Test sobre el desarrollo de Perspectivas Computacionales en estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.



En la **tabla 14** y **figura 8**, se pueden observar los resultados que obtuvieron los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya, se puede apreciar que, durante el Pre test, 10 estudiantes manifiestan un nivel Básico lo que equivale a un 77%, en el nivel Medio se ubicaron 3 estudiantes que conforman un 23%. Ninguno se encuentra en el nivel Alto.

Durante el post test, 11 estudiantes manifiestan un nivel Medio lo que equivale a un 84.6%. En el nivel Alto se ubicó un estudiante que representa un 7.7%. Finalmente, solo un estudiante se encuentra en el nivel Bajo, lo que representa un 7.7% del total.

Estos resultados indican que la propuesta pedagógica desarrollada con los estudiantes de quinto grado de la Institución Educativa Primaria N° 70 116 aplicando el software de programación Scratch tuvo eficacia en el desarrollo de Perspectivas Computacionales, ello evidencia que los estudiantes resolvieron problemas complejos mediante la descomposición, lo que les permitió encontrar el componente que faltaba para llegar hacia el camino señalado, además completaron el bloque que faltaba en las secuencias presentadas haciendo uso de las variables presentada.

Tabla 15

Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de perspectivas computacionales.

Pruebas	Medidas de tendencia central		
	X	S	S ²
Pre Test	2,84	,99	,97
Post Test	5,70	1,44	2,06
Diferencia	2,86	0,45	1,09

Nota: De elaboración propia.

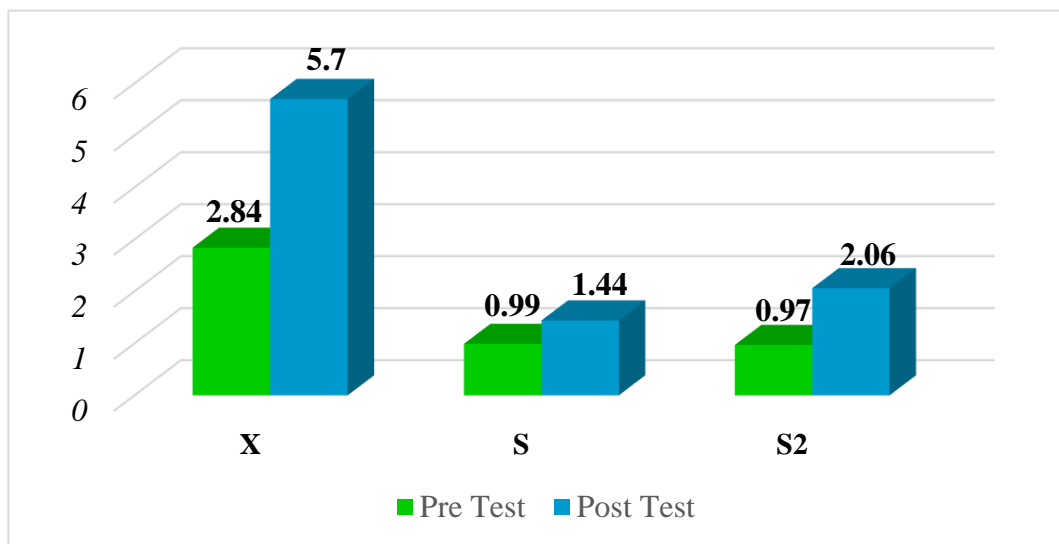


Figura 9. Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post test sobre el desarrollo de perspectivas computacionales.

De la **tabla 15** y **figura 9**, en las comparaciones diferenciadas entre las medidas de tendencia central y dispersión del Pre Test y Post Test sobre el desarrollo de perspectivas computacionales del grupo de estudio experimental, se observa que en el Pre Test la media aritmética es de 2.84 puntos y por otro lado la media aritmética de la prueba de salida es de 5.70 puntos, habiendo una diferencia de 2.86 puntos entre ambos

promedios, lo que demuestra que antes de la aplicación del Scratch, la media fue menor con relación al post test.

Contrastación de la hipótesis específica 3

Ha: La aplicación del software de programación “Scratch” es significativamente eficaz en el desarrollo de perspectivas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

Ho: La aplicación del software de programación “Scratch” no es eficaz en el desarrollo de perspectivas computacionales en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya

Regla de decisión: Si $P < 0.05$, entonces se rechaza la hipótesis nula.

Tabla 16

Prueba t de Student de muestras emparejadas para el objetivo específico 3

	<i>t</i>	<i>gl</i>	<i>Sig. (Bilateral)</i>
<i>Pre test y post test sobre el desarrollo de perspectivas computacionales</i>	8,451	12	,000

Nota: De elaboración propia. SPSS v25

En la **tabla 16**, se observa que $P < 0.05$, de esta manera se rechaza la hipótesis nula y se acepta la hipótesis alterna, la aplicación del Software de programación “Scratch” es significativamente eficaz en el desarrollo de las perspectivas computacionales de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya 2020

Objetivo General

Determinar la eficacia de la aplicación del software de programación “Scratch” en el desarrollo del pensamiento computacional en los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya, 2020.

Tabla 17

Comparación de resultados del Pre Test y Post Test sobre el desarrollo del pensamiento computacional en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.

Nivel	Rango	Pre Proyecto		Post proyecto	
		F	%	F	%
Básico	0 a 9	10	77%	1	8%
Medio	10 a 19	3	23%	10	77%
Alto	20 a 28	0	0%	2	15%
<i>Total</i>		<i>13</i>	<i>100%</i>	<i>13</i>	<i>100%</i>

Nota: De base de datos de la aplicación de instrumento de Pensamiento Computacional

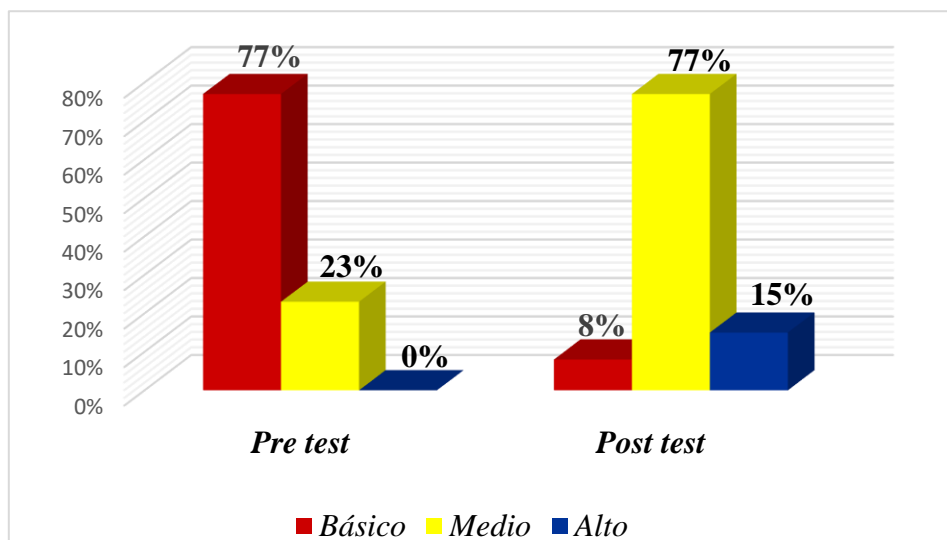


Figura 10. Comparación de resultados del Pre Test y Post Test sobre el desarrollo del pensamiento computacional en los estudiantes del 5° grado de la IEP N° 70 116 Caritamaya.

En la **tabla 17** y **figura 10**, se pueden observar los resultados que obtuvieron los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya, se puede apreciar que, durante el pre proyecto, 10 estudiantes manifiestan un nivel Básico lo que equivale a un 77%, esto indica que obtuvieron puntajes directos inferiores a 9. En el nivel Medio se ubicaron 3 estudiantes que conforman un 23%, tuvieron puntajes directos entre 10 y 19. Ninguno se encuentra en el nivel Alto.

Durante el post proyecto, 10 estudiantes manifiestan un nivel Medio lo que equivale a un 77%, esto indica que obtuvieron puntajes directos entre 10 y 19. En el nivel Alto se ubicaron 2 estudiantes que conforman un 15%, tuvieron puntajes superiores a 20. Finalmente, solo un estudiante se encuentra en el nivel básico, lo que representa un 8% del total, obtuvo un puntaje directo de 9.

Estos resultados indican que la propuesta pedagógica desarrollada con los estudiantes de quinto grado de la Institución Educativa Primaria N° 70 116 aplicando el software de programación Scratch tuvo eficacia en el desarrollo del pensamiento computacional de los estudiantes y eso se puede ver reflejado en los resultados presentados anteriormente.

Tabla 18

Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post- test sobre el desarrollo del pensamiento computacional.

Pruebas	Medidas de tendencia central		
	X	S	S ²
Pre Test	8,85	1,63	2,64
Post Test	16,00	3,14	9,83
Diferencia	7,15	1,51	7,19

Nota: De elaboración propia.

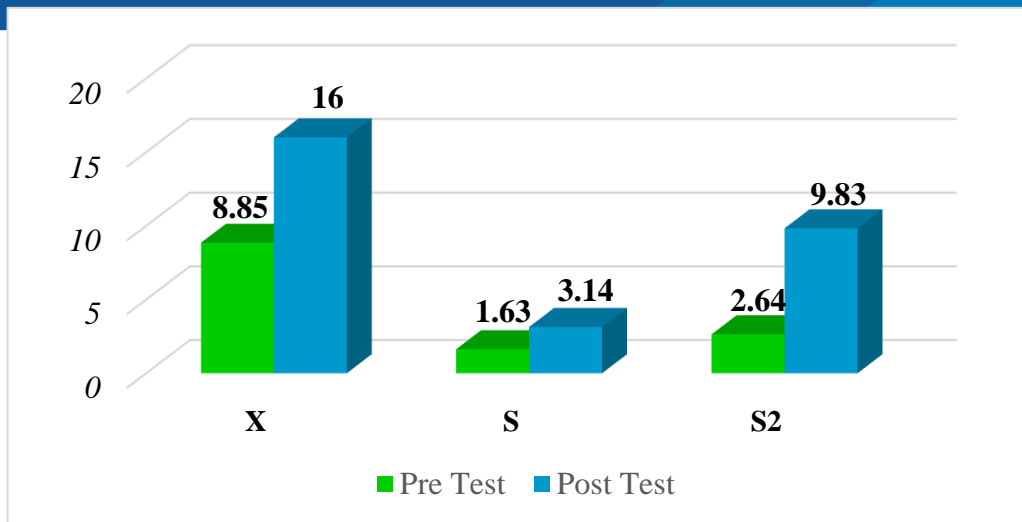


Figura 11. Comparaciones diferenciadas entre las medidas de tendencia central y dispersión del pre test y el post- test sobre el desarrollo del pensamiento computacional.

De la **tabla 18** y **figura 11**, en las comparaciones diferenciadas entre las medidas de tendencia central y dispersión del Pre Test y Post Test sobre el desarrollo del pensamiento computacional, se observa que en el Pre Test la media aritmética es de 8.85 puntos y por otro lado la media aritmética de la Post Test es de 16 puntos, habiendo una diferencia de 7,15 puntos entre ambos promedios, lo que demuestra que antes de la aplicación del Scratch, la media fue menor con relación al post test donde se evidencia una diferencia significativa en el desarrollo del Pensamiento Computacional de los estudiantes del quinto de primaria de la Institución Educativa.

Contrastación de la prueba de hipótesis general

Prueba de la hipótesis general

Ha: La aplicación del Software de programación “Scratch” es eficaz en el desarrollo del pensamiento computacional de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya 2020.

Ho: La aplicación del Software de programación “Scratch” no es eficaz en el desarrollo del pensamiento computacional de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya 2020.

Tabla 19

Prueba t de student de muestras emparejadas sobre el desarrollo del pensamiento computacional.

	<i>t</i>	<i>gl</i>	<i>Sig. (Bilateral)</i>
<i>Pre-Test y Post Test sobre el desarrollo del pensamiento computacional</i>	11,024	12	,000

Nota: De elaboración propia en SPSS v25.

En la **tabla 19**, se reporta los resultados obtenidos sobre la estimación de P-valor de la prueba t de Student. Como se puede observar, el P-valor (0.000) resultó menor que el valor de significancia (0.05).

Tabla 20

Resumen de valores estadísticos calculados.

Prueba T calculada (Tc)	11.024
Prueba T tabulada (Tt)	1.782
Nivel de significancia	0.05
P-valor	0.000

Nota: Datos de la tabla 19.



Según la **tabla 20**, el valor de la prueba t calculada (11.024) resultó mayor que el valor de la prueba t tabulada (1.782). Así mismo, el P-valor (0.000) resultó menor que el valor de nivel de significancia (0.05). Estas premisas enunciadas nos permiten afirmar que se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alterna (H_a). Por lo tanto, se concluye que existe diferencia significativa entre las medias obtenidas del pre test y post test. Lo que indica que la aplicación del software de programación Scratch es eficaz en el desarrollo del pensamiento computacional de los estudiantes del quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya.

4.2. DISCUSIÓN

Los resultados del estudio indican que la aplicación del software de programación Scratch es eficaz en el desarrollo del pensamiento computacional de los estudiantes de quinto grado de la Institución Educativa Primaria N° 70 116 Caritamaya. Esta afirmación se sustenta en la tabla 18 y figura 11, donde se puede observar una diferencia significativa de 7.15 entre la media de las puntuaciones del pre test y post test, así mismo se sustenta en la tabla 20, donde el P-valor (0.000) resultó menor que el nivel de significancia (0.05), y el valor de la prueba t calculada (11.024) resultó mayor que el valor de la prueba t tabulada (1.782). Estos resultados guardan relación con los reportes de Chancolla y Pacori (2017) en su investigación sobre el uso del software Scratch para mejorar el pensamiento computacional en los estudiantes del 5° grado de primaria, donde hallaron una diferencia significativa de 4,68 entre la media de las puntuaciones del Pre Test y Post Test, luego de desarrollar una propuesta pedagógica para aplicar el Scratch. De manera similar se confirma a Churata, et al. (2021) en su investigación sobre la influencia del scratch en el desarrollo del PC en los estudiantes del 5° de primaria, que también encontraron una diferencia significativa en las puntuaciones del Pre y Post Test debido a que la prueba resultó significativa con el P-valor = 0.000 después de aplicar una propuesta pedagógica.



A su vez Caballero y García (2019) en su trabajo de investigación sobre el fortalecimiento de habilidades de pensamiento computacional, también identificaron una diferencia de 1,95 a favor de los resultados del Post-Test después de desarrollar sesiones de intervención con Scratch. Por otro lado, Cabra y Ramírez (2022), en su estudio sobre el desarrollo del pensamiento computacional y las competencias matemáticas en análisis y solución de problemas: una experiencia de aprendizaje con Scratch en la plataforma Moodle, también concluyen que la aplicación de acciones pedagógicas significativas con Scratch favorece en el fortalecimiento de habilidades de pensamiento computacional, a pesar que los estudiantes no tengan experiencia en la creación de actividades por medio de la programación. De igual manera Zarza y Holgado (2020), en su estudio sobre competencia del pensamiento computacional en la educación no formal concluyen que se observa un incremento de nivel en el Post Test respecto a la primera prueba después de haber realizado una secuencia didáctica de sesiones de programación con los estudiantes. A su vez Álvarez (2017), en su estudio sobre el desarrollo del pensamiento computacional en educación primaria: una experiencia educativa con Scratch, concluye que la herramienta Scratch es propicia para desarrollar el Pensamiento Computacional de alumnos de Educación Primaria después de haber desarrollado guías de iniciación con Scratch. Así mismo, se confirma a Araujo (2021) en su estudio titulado sobre lenguaje de programación Scratch y pensamiento computacional que concluye en que el lenguaje de programación Scratch influye en el pensamiento computacional de los estudiantes, y el dictado de los cursos de programación con Scratch tiene un impacto positivo en los estudiantes. Estos hallazgos refuerzan la importancia de aplicar propuestas pedagógicas, que ayudan a desarrollar el pensamiento computacional aplicando el software de programación Scratch en las sesiones de aprendizaje de manera interactiva y colaborativa.



En cuanto al primer objetivo específico, los resultados que se encuentran en la Tabla 8, Figura 4 y número de página 61, muestran que el 92,3% de estudiantes mejoró sus puntajes en el post test, por ello se ubicaron en el nivel medio y uno en el nivel alto. Dicho hallazgo confirma el estudio realizado por Chancolla y Pacori (2017) sobre el uso del software Scratch para mejorar el pensamiento computacional, quienes notaron una mejora en las puntuaciones del post test referente a los indicadores de secuencia, ciclos y paralelismo analizados en los ítems 1, 5, 8, 9, 12, 13, 14, 15 y 16 de su investigación. De esta manera se puede destacar que los estudiantes adaptaron los conceptos computacionales durante el desarrollo de las sesiones aplicando el software de programación Scratch. De la misma forma los resultados guardan relación con los reportes de Arranz y Pérez (2017) en su estudio sobre evaluación del Pensamiento Computacional en educación primaria que concluyen en que el uso de Scratch favorece el desarrollo del PC, y son los conceptos computacionales los que han sido desarrollados en mayor medida por los estudiantes. Así mismo se confirma a Sáez et al. (2021) en su estudio titulado “aplicación de la robótica y programación con Scratch en la enseñanza elemental”, concluyeron que la aplicación de Scratch logra mejoras estadísticamente significativas en la comprensión de conceptos computacionales como secuencias, bucles, paralelismo y manejo de eventos.

Con respecto al segundo objetivo específico, los resultados que se encuentran en la tabla 11, Figura 6 y número de Página 65, muestran la diferencia entre los resultados del Pre y Post test sobre el desarrollo de prácticas computacionales, a pesar que el 54% de estudiantes aún se encuentran en nivel básico esto debido a que no lograron mejorar la abstracción. Estos resultados guardan relación con el estudio realizado por Churata et al. (2019) sobre la influencia del scratch en el desarrollo del pensamiento computacional donde el 50% de los estudiantes manifiestan un nivel Básico en el indicador



“abstracción”, considerándola como una de las habilidades más complicadas de desarrollar prácticas computacionales. De igual forma, Taco (2019) en su estudio sobre la influencia del Scratch en el desarrollo del pensamiento computacional de los estudiantes concluye que los estudiantes de quinto grado presentan un nivel regular de PC esto debido a que suelen encontrar ciertas deficiencias o puntos de quiebre en las prácticas computacionales y las perspectivas computacionales.

En cuanto al tercer objetivo específico, los resultados que se encuentran en la Tabla 14, Figura 8 y número de página 69, muestran una diferencia significativa en los puntajes obtenidos del Pre Test y Post Test sobre el desarrollo de perspectivas computacionales en los estudiantes. Así mismo la estimación de la prueba T de Student que se encuentra en la tabla 16 y número de página 73 lo refleja con un nivel de 8.451, teniendo solo un estudiante en el nivel bajo, puesto que es la dimensión que establece una relación esencial entre teoría y práctica; donde el estudiante tiene la capacidad para administrar su conocimiento y lo que sabe hacer para tomar decisiones y evaluar escenarios. Esto permite confirmar el estudio realizado por Mantilla y Negre (2021) quienes, a pesar de haber usado un diseño distinto, afirman que existe una relación representativa con una significancia de 0.000 entre la dimensión perspectivas computacionales (saber ser) con las dimensiones sobre prácticas y conceptos computacionales (Saber y saber hacer), ya que la práctica permite obtener un mejor desempeño en la toma de decisiones y evaluación de situaciones, a sí mismo afirman que la teoría fomenta la capacidad para tomar decisiones fundamentadas, administrando conocimiento.



V. CONCLUSIONES

PRIMERA: Según los resultados obtenidos, la aplicación del software de programación Scratch es eficaz en el desarrollo del pensamiento computacional de los estudiantes del quinto grado de la IEP N° 70 116 de Caritamaya, 2020. Esta afirmación se sustenta en la tabla 18 y figura 11 donde se puede observar una diferencia significativa de 7.15 entre la media de las puntuaciones del pre test y post test, así mismo se sustenta en la tabla 20, donde el P-valor (0.000) resultó menor que el nivel de significancia (0.05), y el valor de la prueba t calculada (11.024) resultó mayor que el valor de la prueba t tabulada (1.782). Estos resultados obtenidos evidencian el compromiso, la dedicación y perseverancia de los estudiantes durante las sesiones de aprendizaje desarrolladas como parte de la propuesta pedagógica para aprender a programar a través de mecanismo lúdicos como lo presenta el Scratch.

SEGUNDA: La aplicación del software de programación Scratch es significativamente eficaz en el desarrollo de los conceptos computacionales. Esta afirmación se sustenta en la estimación de la prueba t de student, donde el P-valor (0.000) resultó menor que el nivel de significancia (0.05) tal y como se puede observar en la tabla 10. Además, la media de puntuaciones del Post Test es mayor a la del Pre- Test, esto debido a que en cada sesión realizada se buscó obtener un beneficio afectivo, es decir, que las actividades fueran divertidas y llamativas para los niños y que a su vez colaboren al desarrollo cognitivo del pensamiento computacional.



TERCERA: La aplicación del software de programación Scratch es significativamente eficaz en el desarrollo de las prácticas computacionales. Esta afirmación se sustenta en la estimación de la prueba t de student, donde el P-valor (0.000) resultó menor que el nivel de significancia (0.05) tal y como se puede observar en la tabla 13. Además, se ha evidenciado que la media de puntuaciones del Post Test es mayor a la media de las puntuaciones del Pre Test. Sin embargo, el 54% de estudiantes aún se encuentran en nivel básico, esto debido a que no lograron mejorar la abstracción y modularización para poder depurar, el 38% se ubicó en el nivel medio y un estudiante alcanzó el nivel alto esto evidencia que las practicas computacionales afianzan de forma significativa conceptos adquiridos, se puede inferir que los estudiantes practicaron las veces necesarias para utilizar la abstracción en sus proyectos de Scratch.

CUARTA: La aplicación del software de programación Scratch es significativamente eficaz en el desarrollo de las prácticas computacionales. Esta afirmación se sustenta en la estimación de la prueba t de Student donde el P-valor (0.000) resultó menor que el nivel de significancia (0.05) tal y como se puede observar en la tabla 16. Además, se obtuvo una media de puntuaciones del Post Test mayor a la media del Pre Test, al principio el 77% de los estudiantes se encontraban en el nivel Básico y después de la aplicación del software de programación Scratch el 84.6% de los estudiantes se ubicaron en el nivel medio y un estudiante en el nivel alto.



VI. RECOMENDACIONES

PRIMERA: Se recomienda a los docentes de las instituciones educativas a nivel nacional implementar el uso del Scratch en las sesiones de aprendizaje para que los niños aprendan de forma lúdica la programación, lo cual favorecerá en el desarrollo de habilidades del pensamiento computacional. Scratch permite que todo aquel que se enfrente por primera vez a programar, lo realice de manera atractiva y accesible. Además, el pensamiento computacional, como cualquier otra competencia básica en el siglo XXI, no debería estar limitada dentro de una asignatura de un currículo, se puede integrar a otras áreas curriculares que desarrollan en las instituciones educativas.

SEGUNDA: Se recomienda a los docentes que laboran en Instituciones Educativas de ámbito rural iniciar en el desarrollo del pensamiento computacional con sus estudiantes, a través de actividades desenchufadas que permitirán a los niños ir aprendiendo habilidades o conceptos que son parte del pensamiento computacional. Puesto que la realidad nuestro país, presenta instituciones que se ubican en zonas rurales y no cuentan con un laboratorio de computación que permita a los niños aprender a programar.

TERCERA: Se recomienda a las autoridades competentes de la Escuela Profesional de Educación Primaria, contemplen en su plan de estudios asignaturas relacionadas a los Entornos Virtuales para el Aprendizaje, de esa manera se podrá atender las necesidades tecnológicas de los estudiantes. Así mismo, se recomienda a futuros investigadores profundizar trabajos de investigación relacionado al tema, tomando como base la presente investigación.



CUARTA: A los docentes de la Institución Educativa Primaria N° 70 116 Caritamaya, recomendarles que promuevan en sus estudiantes actividades productivas con la tecnología, de manera tal que los estudiantes puedan encontrar en las sesiones de aprendizaje, oportunidades de desarrollo precisas para participar creativa y constructivamente en la nueva sociedad digital. En este sentido amplio, el pensamiento computacional permitirá pasar de ser meros usuarios de las TIC a ser creadores en el mundo de las TIC.



VII. REFERENCIAS BIBLIOGRÁFICAS

- Acurio, L. (11 de abril de 2017). Primero la lógica del pensamiento computacional después trabajar con el software y luego, aterrizarlo en robótica [ENTREVISTA]. (Channel news Perú, Entrevistador) Obtenido de <https://channelnewsperu.com/index.php/2017/04/11/primero-la-logica-del-pensamiento-computacional-despues-trabajar-con-el-software-y-luego-aterrizarlo-en-robotica/>
- Álvarez, M. (2017). Desarrollo del pensamiento computacional en educación primaria: una experiencia educativa con Scratch. *Revista de Ciències de l'Educació*(2), 45-64. doi:<https://doi.org/10.17345/ute.2017.2.1820>
- Araujo, L. (2021). *Lenguaje de programación Scratch y pensamiento computacional en niños de 6 a 12 años, Institución Influence SAC. Los Olivos, 2020*". Universidad Científica del Sur. Obtenido de <https://repositorio.cientifica.edu.pe/bitstream/handle/20.500.12805/1902/TB-Araujo%20L-Ext.pdf?sequence=1&isAllowed=y>
- Arias, F. (2012). *El proyecto de investigación*. Caracas, Venezuela: EPISTEME C.A. .
- Arranz, H., & Pérez, A. (2017). Evaluación del Pensamiento Computacional en Educación. *Revista Interuniversitaria de Investigación en Tecnología Educativa (RIITE)*(3), 25-39. doi:<http://dx.doi.org/10.6018/riite/2017/267411>
- Basogain, X., Olabe, M., & Olabe, J. (2015). Pensamiento Computacional a través de la Programación: Paradigma de aprendizaje. *Revista de Educación a Distancia*, 1-33. doi:[10.6018/red/46/6](https://doi.org/10.6018/red/46/6)



- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing Computational Thinking in Compulsory Education. *JRC science for policy report*, 1-66.
- Caballero, Y. A., & García, A. (2019). Fortaleciendo habilidades de pensamiento computacional en Educación Infantil: Experiencia de aprendizaje mediante interfaces tangible y gráfica. *RELATEC. Revista Latinoamericana de Tecnología Educativa*, 18(2), 134-149. Obtenido de <http://hdl.handle.net/10662/10445>
- Cabra, M., & Ramirez, S. (2022). Desarrollo del pensamiento computacional y las competencias matemáticas en análisis y solución de problemas: una experiencia de aprendizaje con Scratch en la plataforma Moodle. *Revista educación*, 46(1), 1-16. Obtenido de <https://revistas.ucr.ac.cr/index.php/educacion/article/view/44970/48400>
- Cano, E., & Delgado, D. (2015). La creación de videojuegos con Scratch en Educación secundaria. *Communication papers: media literacy and gender studies.*, 63-73. Obtenido de *Communication papers: media literacy and gender studies.*
- Carrasco, S. (2005). *Metodología de la investigación científica. Pautas metodológicas para diseñar y elaborar el proyecto de investigación.* (Primera edición ed.). Lima: San Marcos.
- Chancolla, G. L., & Pacori, E. J. (2017). *El uso del software Scratch para mejorar el pensamiento computacional en los estudiantes del quinto grado de primaria de la Institución Educativa N° 40009 San Martín de Porres del distrito de Paucarpata, Arequipa, 2016.* Arequipa: Universidad Nacional de San Agustín de Arequipa.
- Churata, E., Coila, T., & Tintaya, L. M. (2021). *Influencia del SCRATCH en el desarrollo del pensamiento computacional en los estudiantes del quinto de primaria de la*



Institución Educativa 40028 Alto Selva Alegre - Arequipa - 2019. Arequipa: Universidad Católica de Santa María. Obtenido de <http://tesis.ucsm.edu.pe/repositorio/handle/UCSM/10611>

Code En Mi Cole. (16 de diciembre de 2020). Políticas públicas educativas para el desarrollo del pensamiento computacional [VIDEO]. Lima, Lima, Perú. Obtenido de <https://www.youtube.com/watch?v=cgxFd40rwnk>

Code learn. (8 de junio de 2020). *Code learn Computational Thinking*. Obtenido de PISA 2021 evaluará el pensamiento computacional de los alumnos: <https://codelearn.es/blog/pisa-2021-evaluara-el-pensamiento-computacional-de-los-alumnos/>

Condo, A. (2017). *El pensamiento computacional en estudiantes del VII ciclo de la institución educativa particular "Ricardo Palma" - San Juan de Miraflores 2016.* Repositorio Institucional Universidad César Vallejo. Obtenido de https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/13236/Condo_LA.pdf?sequence=6&isAllowed=y

Daza, E., & Lizarazo, F. (2020). *Promover habilidades de pensamiento computacional en la era de la inequidad tecnológica.* Cali: Universidad Santiago de Cali. Obtenido de <https://libros.usc.edu.co/index.php/usc/catalog/download/215/217/3942?inline=1>

Denning, P. (2017). Computational Thinking in Science. *Americas Scientist*, 105(1), 13. doi: 10.1511/2017.124.13

Duarte, V., Carrillo, G., Carrera, Á., & Falcones, C. M. (2020). Experiencias sobre la inducción de tecnologías programables para el desarrollo del pensamiento



computacional en escuelas de zonas rurales y urbano marginales. *Vinculos - Espe*, 5(3), 67-81. doi:10.24133/vinculosespe.v5i3.1720

Evangelista, A. (2018). *Aplicación del Software SCRATCH en el fortalecimiento de capacidades de Pensamiento de Orden Superior de los estudiantes de Educación Primaria de la Universidad Nacional José Faustino Sánchez Carrión Año 2018*. Huacho: Universidad Nacional José Faustino Sánchez Carrión.

Flores, C., & Maldonado, S. (2007). Software Educativo SCRATCH Aplicado como Herramienta Transversal en el Currículo Educativo del Colegio Mayor San Lorenzo. *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*.

Flores, E. (2019). *Modelo holístico de código alfabetización en el desarrollo del pensamiento computacional en educación primaria*. Lima: Universidad Nacional Federico Villarreal. Obtenido de <http://repositorio.unfv.edu.pe/handle/UNFV/3735>

Flores, J. (2011). *Construyendo la tesis universitaria. Guia didáctica*. Lima: Garden Graf S.R.L.

García, J. (30 de Abril de 2020). La expansión del Pensamiento Computacional en Uruguay. *Revista de Educación a Distancia*, 20(63), 1-15. doi:<http://dx.doi.org/10.6018/red.410441>

George, D., & Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference*. Boston: USA:Allyn & Bacon.



- Hernández, R., Fernández, C., & Baptista, P. (2010). *Metodología de la investigación 5ta edición*. Colombia: McGRAW - HILL INTERAMERICANA DE MÉXICO, S.A.
- Hourofcode. (2020). *Hora del código*. Obtenido de Propagandas y estadísticas útiles: <https://hourofcode.com/es/promote/stats>
- Lamoth, Y., Montero, J. L., & García, Y. (8 de octubre de 2020). La Alfabetización Digital en los docentes universitarios: un reto para las universidades contemporáneas. *EduSol*, 20(73), 193-205. Recuperado el Noviembre de 2021, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1729-80912020000400193
- Lazo, K. (2019). Expertos discutirán la importancia de la Ciencia de la Computación en la Educación Escolar [ENTREVISTA]. (B. Empresarial, Entrevistador) Obtenido de <http://www.businessempresarial.com.pe/expertos-discutiran-la-importancia-de-la-ciencia-de-la-computacion-en-la-educacion-escolar/>
- López, J. (30 de Enero de 2020). Programación con scratch cuaderno de trabajo para estudiantes. *Revista Cátedra*, 3(1). doi:<https://doi.org/10.29166/10.29166/catedra.v3i1.2006>
- Mamani, J. L. (2018). *Pensamiento computacional en estudiantes del 4to grado de educación secundaria de la I.E. Morales Delgado de Pampa de Camarones Sachaca Arequipa 2017*. Arequipa: Repositorio UNSA. Obtenido de <http://repositorio.unsa.edu.pe/handle/UNSA/6805>
- Mantilla, R., & Negre, F. (2021). Pensamiento computacional, una estrategia educativa en épocas de pandemia. *Innoeduca*, 7(1), 89-106. doi:<https://doi.org/10.24310/innoeduca.2021.v7i1.10593>



- Martinez, M. C., & Gomez, M. J. (2018). Programar computadoras en educación infantil. *Revista Electrónica de Tecnología Educativa EDUTEC*(65), 40-53. doi:<https://doi.org/10.21556/edutec.2018.65.1103>
- Mejía, E. (2013). *La investigación científica en educación*. Lima: Universidad Nacional Mayor de San Marcos.
- Ministerio de Educación. (Noviembre de 2019). *Perú Educa*. Obtenido de Importancia de la incorporación de ciencia de la computación en la educación escolar: <https://www.perueduca.pe/#/de/docentes/noticias-2019/11/expertos-discutiran-la-importancia-de-la-incorporacion-de-ciencia-de-la-computacion-en-la-educacion-escolar>
- Ministerio de Educación. (25 de Agosto de 2019). *Perú Educa*. Obtenido de La hora del código Perú 2019: http://www.ugel06.gob.pe/portal/images/AM_La_Hora_del_Cdigo.pdf
- Monjelat, N. (2019). Programación de tecnologías para la inclusión social con Scratch: Prácticas sobre el pensamiento computacional en la formación docente. *Revista Electrónica Educare*, 23(3), 182-206. doi:<https://doi.org/10.15359/ree.23-3.9>
- Muñoz, L. (29 de octubre de 2019). Foro: Educar para el futuro incierto. *La Nación*.
- Muñoz, M. (4 de setiembre de 2018). *Generación TEC*. Obtenido de ¿Qué es el pensamiento computacional?: <https://www.generaciontec.com/post/qu%C3%A9-es-el-pensamiento-computacional>
- Pascual, J. (13 de diciembre de 2015). Scratch, programación sencilla y gratis para niños y mayores. *Computer Hoy*. Obtenido de



<https://computerhoy.com/noticias/software/scratch-programacion-sencilla-gratis-ninos-mayores-37925>

Raja, T. (19 de Junio de 2014). *We Can Code It! Why computer literacy is key to winning the 21 st century. Mother Jones*. Obtenido de <https://medium.com/mother-jones/we-can-code-it-e5f64d8b3075>

Red STEM Latinoamérica. (2019). *Mesas Temáticas Regionales: Otro espacio para construir*. Recuperado el 5 de Noviembre de 2021, de <https://educacion.stem.siemens-stiftung.org/mesas-tematicas-regionales-otro-espacio-para-construir/>

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (2009). Scratch: Programming for All. *COMMUNICATIONS OF THE ACM*, 52(11), 60 - 67.

Román, M. (Julio de 2015). COMPUTATIONAL THINKING TEST: DESIGN GUIDELINES AND CONTENT VALIDATION. *Universidad Nacional de Educación a Distancia* , 2436-2444. doi:10.13140/RG.2.1.4203.4329

Román, M. (2016). *Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: Validación de un instrumento y evaluación de programas*. Madrid.

Román, M., Pérez, J., & Jiménez, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 678 - 691.



- Román, M., Pérez, J., & Moreno, J. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction ELSEVIER*, 47-58.
- Rosales, M. E. (2019). *Pensamiento computacional y la resolución de problemas de matemática en estudiantes de primaria en Lima Cercado, 2019*. Universidad César Vallejo. Obtenido de https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/49621/Rosales_BME-SD.pdf?sequence=1&isAllowed=y
- Rushkoff, D. (2010). *Program or be programmed*. Nueva York: OR Books.
- Sáez, J. M., & Cózar, R. (2017). Pensamiento computacional y programación visual por bloques en el aula de Primaria. *Educar*, 53(1), 129-146. Obtenido de <https://raco.cat/index.php/Educar/article/view/317274>
- Sáez, J. M., Buceta, R., & De Lara García, S. (2021). Aplicación de la robótica y programación por bloques en la enseñanza elemental. *Revista Iberoamericana de Educación a Distancia*, 24(1), 95-113. doi:<https://doi.org/10.5944/ried.24.1.27649>
- Santoyo, J. (2016). *“innovación de video juegos con el software scratch para fortalecer las habilidades de pensamiento creativo en estudiantes de tecnología informática del grado noveno del instituto agrícola de alto jordán de vélez santander colombia para el año 2016”*. Universidad Privada Norbert Wiener. Obtenido de <http://repositorio.uwiener.edu.pe/handle/123456789/593>
- Scratch. (2020). *Scratch.mit.edu*. Recuperado el 07 de Marzo de 2021, de Acerca de Scratch: <https://scratch.mit.edu/about>



- Shute, V., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. doi:<https://doi.org/10.1016/j.edurev.2017.09.003>
- Soria, E., & Rivero, C. (2019). Pensamiento computacional: una nueva exigencia para la educación del siglo XXI. *Espacio Pedagógico*, 26(2), 323-337. doi:<https://doi.org/10.5335/rep.v26i2.8702>
- Taco, R. A. (2019). *Influencia del Programa Scratch en el Pensamiento Computacional en Estudiantes del Nivel Primario de la Institución Educativa de la Policía Nacional del Perú Alférez Mariano Santos Mateos, Tacna 2018*. Tacna: Universidad Privada de Tacna. Obtenido de <http://repositorio.upt.edu.pe/handle/20.500.12969/1015>
- Universidad Católica San Pablo. (2021). *Universidad Católica San Pablo*. Recuperado el 2021, de Diplomado en pensamiento computacional: <https://cs.ucsp.edu.pe/postgrado/diplomado-pensamiento-computacional/>
- Valverde, J., Fernández, M. R., & Garrido, M. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, 3(46), 1-18. Obtenido de https://www.um.es/ead/red/46/valverde_et_al.pdf
- Vásquez, E., Bottamedi, J., & Brizuela, M. (2019). Pensamiento computacional en el aula: el desafío en los sistemas educativos de Latinoamérica. *Revista interuniversitaria de investigación en Tecnología Educativa*(7), 36-47. doi:<https://doi.org/10.6018/riite.397901>
- Vera, J., Ocsa, A., & Villalba, K. (2015). Modelo de sistema de recomendación de Objetos de Aprendizaje en dispositivos móviles, caso: Desarrollo del pensamiento computacional. *Nuevas ideas en informática Educativa TISE*, 730-734.



- Vilanova, G. (2018). Tecnología Educativa para el Desarrollo del Pensamiento Computacional. *Sistemas, Cibernética e informática*, 15(3), 25-32. Obtenido de <http://www.iiisci.org/journal/pdv/risci/pdfs/CA074QW17.pdf>
- Wing, J. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35. Obtenido de <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- Wing, J. (2010). *Computational Thinking: What and Why?* . Obtenido de The magazine of Carnegie Mellon University's School of Computer Science.
- Zapata, M. (15 de Setiembre de 2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia*(46), 1-47. Obtenido de <https://revistas.um.es/red/article/view/240321/183001>
- Zarza, M., & Holgado, J. (2020). Competencia de Pensamiento Computacional en la educación no formal. *EDUTEC. Revista Electrónica de Tecnología Educativa*(72), 68 - 87. Obtenido de <https://www.edutec.es/revista/index.php/edutec-e/article/view/1575/755>



ANEXOS