

UNIVERSIDAD NACIONAL HERMILIO VALDIZÁN
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TESIS

**APLICACIÓN DEL LENGUAJE DE PROGRAMACIÓN SCRATCH
PARA EL DESARROLLO DEL PENSAMIENTO ALGORÍTMICO EN
LOS ALUMNOS DEL 6TO GRADO DEL NIVEL PRIMARIO EN LA
I.E.P AUGUSTO CARDICH – PILLCO MARCA.**

TESIS PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS

TESISTA:

Bach. RIVERA ILLATOPA, Romer Ramiro

ASESOR:

Dr. FRANCISCO PAREDES ,Adam.

HUÁNUCO – PERÚ

2020

DEDICATORIA

A Dios. Por haberme permitido llegar hasta este punto, darme la vida para lograr mis objetivos y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de mi vida. Además de su infinita bondad y amor.

A mis hijos Kazzuo y Khalessi. Porque simplemente y sencillamente son mi vida.

A mi esposa Amparo Kanashiro. Por creer en mí, por sus palabras de aliento que me permitieron culminar esta tesis y por ser un motivo más en mi vida.

A mi mamá Marilú. Por haberme apoyado en todo momento y ser el pilar fundamental en todo lo que soy, por sus consejos, por sus valores, por la motivación constante, pero más que nada, por su amor y sacrificio incondicional.

A mi papá Nilo (QEPD). Por los ejemplos de perseverancia, constancia que lo caracterizaron y que me ha infundado siempre, por el valor que me mostró para salir adelante y por su amor.

A mi hermano Gean. Por ser un motor importante para mi superación personal, porque siempre me ha apoyado incondicionalmente.

AGRADECIMIENTO

A los docentes de la FIIS (Facultad de ingeniería Industrial y de sistemas).

Aquellos que marcaron cada etapa de nuestro camino universitario vertiéndonos sus conocimientos y experiencias.

A los docentes del PROCAP.

Por el apoyo, sugerencias y guía para el desarrollo de la presente tesis que sin sus conocimientos no se hubiera podido culminar.

Al promotor, director, coordinadores, docentes y alumnos de la I.E.P Augusto Cardich.

Por sus valiosas informaciones y permitir que se ejecute la presente tesis.

RESUMEN

El trabajo realizado tuvo como objetivo determinar el desarrollo del pensamiento algorítmico de los alumnos del sexto grado de la I.E.P Augusto Cardich a través de la aplicación del lenguaje de programación SCRATCH. Es una investigación de tipo Aplicada donde se contó con dos grupos: uno experimental que cuenta con 23 alumnos y el otro de control que cuenta con 19 alumnos, ambos grupos del sexto grado del nivel primario; a éstos dos grupos se toma dos pruebas de conocimientos una pretest y la otra postest. En el grupo experimental se aplica el lenguaje de programación SCRATCH para que los alumnos desarrollen el pensamiento algorítmico y luego compararlas con el grupo de control. Después del desarrollo de las sesiones que fueron cinco, el primero y el último de las sesiones fueron considerados para tomar el examen de conocimiento del pretest y postest pero las otras tres sesiones fueron llevados al SPSS y se obtuvo los siguientes resultados: Utilizando la Prueba T para muestras relacionadas y comparando la variable dependiente del pretest y la variable dependiente de postest, (la variable dependiente es el desarrollo del pensamiento algorítmico) nos determina que la significancia o el P-valor es $0.000 \leq 0.05$ por lo que se acepta la hipótesis alterna que dice: “la aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019”; con esto podemos afirmar que la aplicación del lenguaje de programación SCRATCH mejora la capacidad analítica, ayuda a ordenar en pasos lógicos un problema e influye en la mejor toma de decisiones de los alumnos, por ende se debe

utilizar de forma adecuada el software SCRATCH para que los alumnos aprendan de manera divertida a programar e influir en su desarrollo del pensamiento algorítmico .

PALABRAS CLAVES: SCRATCH, lenguaje de programación, algoritmo y pensamiento algorítmico.

SUMMARY

The objective of the work carried out was to determine the development of the algorithmic thinking of the sixth grade students of the I.E.P Augusto Cardich through the application of the SCRATCH programming language. It is an Applied type investigation where there were two groups: an experimental one with 23 students and the other with a control group with 19 students, both groups from the sixth grade of the primary level; these two groups are given two knowledge tests, one pretest and the other posttest. In the experimental group, the SCRATCH programming language is applied so that the students develop algorithmic thinking and then compare them with the control group. After the development of the five sessions, the first and last of the sessions were considered to take the pretest and posttest knowledge exam, but the other three sessions were taken to the SPSS and the following results were obtained: Using the T Test For related samples and comparing the dependent variable of the pre-test and the dependent variable of the post-test, (the dependent variable is the development of algorithmic thinking), it determines that the significance or the P-value is $0.000 \leq 0.05$, which is why the alternative hypothesis that says: “the application of the SCRATCH programming language positively influences the development of algorithmic thinking of 6th grade students of the primary level of the Augusto Cardich-Huánuco 2019 IEP”; With this we can affirm that the application of the SCRATCH programming language improves analytical capacity, helps to order a problem into logical steps and influences the better decision-making of students, therefore the

SCRATCH software must be used appropriately so that students learn in a fun way to program and influence their development of algorithmic thinking.

KEY WORDS: SCRATCH, programming language, algorithm and algorithmic thinking

CONTENIDO	pg.
INTRODUCCIÓN	12
I. PLANTEAMIENTO DEL PROBLEMA	14
1.1 ANTECEDENTES Y FUNDAMENTACIÓN DEL PROBLEMA	14
1.1.1 ANTECEDENTES:	14
1.1.2 FUNDAMENTACIÓN DEL PROBLEMA:	16
1.2 FORMULACIÓN DEL PROBLEMA.	19
1.3 OBJETIVOS: GENERALES Y ESPECÍFICOS.	20
1.4 HIPÓTESIS: GENERAL Y ESPECÍFICOS	21
1.4.1 HIPÓTESIS GENERAL	21
1.4.2 HIPÓTESIS ESPECÍFICOS	21
1.5 VARIABLES	22
1.5.1 VARIABLE INDEPENDIENTE	22
1.5.2 VARIABLE DEPENDIENTE	22
1.6 DEFINICIÓN OPERACIONAL DE VARIABLES, DIMENSIONES E INDICADORES	23
1.7 JUSTIFICACIÓN E IMPORTANCIA	24
1.8 LIMITACIONES	26
II. MARCO TEÓRICO	27
2.1 REVISIÓN DE ESTUDIOS REALIZADOS (ANTECEDENTES)	27
2.1.1 ANTECEDENTES INTERNACIONALES:	27
2.1.2 ANTECEDENTE NACIONAL:	36
2.1.3 ANTECEDENTE REGIONAL	37
2.2 PRINCIPALES LEYES, DEFINICIONES Y CONCEPTOS FUNDAMENTALES	37
2.2.1 TEMAS EJES DE LA VARIABLE INDEPENDIENTE- LENGUAJE DE PROGRAMACIÓN SCRATCH.	38
2.2.2 TEMAS EJES DEL PENSAMIENTO ALGORÍTMICO (VARIABLE INDEPENDIENTE).	59
2.3 MARCO SITUACIONAL	92

2.4	DEFINICIÓN DE TÉRMINOS BÁSICOS	96
III.	MARCO METODOLÓGICO	98
3.1	NIVEL Y TIPO DE INVESTIGACIÓN.....	98
3.2	DISEÑO DE LA INVESTIGACIÓN	98
3.3	DETERMINACIÓN DE LA POBLACIÓN	99
3.4	SELECCIÓN DE LA MUESTRA	99
3.5	TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS	100
3.5.1	TÉCNICA DE RECOLECCIÓN DE DATOS	100
3.5.2	MEDIOS E INSTRUMENTO DE RECOLECCIÓN DE DATOS.....	101
3.6	PROCESAMIENTO Y PRESENTACIÓN DE DATOS	103
IV.	RESULTADOS DE LA INVESTIGACIÓN.....	104
4.1.1.	ORDENAMIENTO DE CUESTIONARIOS PARA LA VALIDACIÓN DE MI INSTRUMENTO DE INVESTIGACIÓN	104
4.1.2.	ORDENAMIENTO Y CLASIFICACIÓN DE LAS PRUEBAS DE CONOCIMIENTOS	106
4.1.3.	PRUEBA DE HIPÓTESIS	106
V.	DISCUSIÓN DE LOS RESULTADOS	122
	CONCLUSIONES.....	124
	RECOMENDACIONES.....	126
	BIBLIOGRAFÍA.....	127
	ANEXOS.....	130

TABLAS

Tabla 1: DEFINICIÓN OPERACIONAL DE LA VARIABLE INDEPENDIENTE	23
Tabla 2: DEFINICIÓN OPERACIONAL DE LA VARIABLE DEPENDIENTE	23
Tabla 3: PRINCIPALES LEYES, DEFINICIONES Y CONCEPTOS FUNDAMENTALES	38
Tabla 4: DATOS DEL COLEGIO AUGUSTO CARDICH	92
Tabla 5: DISTRIBUCIÓN DE LA POBLACIÓN	99
Tabla 6: DISTRIBUCIÓN DE LOS GRUPOS Y LA MUESTRA	100
Tabla 7: ESQUEMA DE LA TESIS	100
Tabla 8: PRUEBA DE CONOCIMIENTO DEL PRETEST Y POSTEST	101
Tabla 9: VALIDACIÓN DEL INSTRUMENTO	102
Tabla 10: CALIFICACIÓN DEL INSTRUMENTO DE INVESTIGACIÓN	105
Tabla 11: CUADRO COMPARATIVO PARA ELECCIÓN DE LA PRUEBA	108
Tabla 12: PRUEBA DE NORMALIDAD	110
Tabla 13: CRITERIO PARA LA NORMALIDAD	110
Tabla 14: PRUEBA DE MUESTRAS RELACIONADAS	111
Tabla 15: PRUEBA UTILIZANDO ANOVA	112
Tabla 16: PRUEBA DE MUESTRAS RELACIONADAS PARA LA HE1	114
Tabla 17: PRUEBA UTILIZANDO ANOVA PARA HE1	115
Tabla 18: PRUEBA DE MUESTRAS RELACIONADAS PARA LA HE2	116
Tabla 19: PRUEBA UTILIZANDO ANOVA PARA LA HE 2	117
Tabla 20: PRUEBA DE MUESTRAS RELACIONADAS PARA LA HE 3	119
Tabla 21: PRUEBA UTILIZANDO ANOVA PARA LA HE 3	120

ILUSTRACIONES

Ilustración 1: ZONAS DEL SCRATCH	50
Ilustración 2: ETAPAS PARA RESOLVER PROBLEMAS.....	68
Ilustración 3: FASES PARA ELBAORAR UN PROGRAMA DE COMPUTADOR... 	71
Ilustración 4: ETAPAS A DESARROLLAR EN LA FASE DE ANÁLISIS DE UN PROBLEMA	72
Ilustración 5: DIAGRAMA DE FLUJO PARA HALLAR EL ÁREA DE UN TRIÁNGULO RECTÁNGULO	81

INTRODUCCIÓN

Vivimos en una era digital en donde los niños de hoy son nativos digitales porque han crecido con la tecnología, sin embargo el uso inadecuado no contribuye en el desarrollo de sus capacidades y habilidades, por ello la presente investigación es importante porque permite al alumno desarrollar su pensamiento algorítmico utilizando el lenguaje de programación SCRATCH.

A través del SCRATCH los niños experimentan la solución de un problema, animaciones y juegos con secuencias de pasos o instrucciones a la cual en programación llamamos algoritmos. La experiencia como docente del área de EPT (Educación para el Trabajo)/Computación ha permitido observar que los estudiantes no lleven programación en el nivel primario ni secundario porque se considera que es muy complejo, solo aprenden herramientas básicas como lo son Microsoft Office (Word, Power Point, y Excel) y repiten año tras año las mismas temáticas dejando de lado los estándares de aprendizaje que se encuentra plasmado en el DCN.

Siendo la tecnología hoy en día el eje transversal que se puede aplicar en todas las áreas curriculares, la programación SCRATCH permitirá que el estudiante sea creativo, analítico y desarrolle su pensamiento algorítmico la misma que le permitirá resolver problemas y tomar la mejor alternativa en su toma de decisiones y plasmarla en su vida cotidiana. El lenguaje de programación Scratch es importante también porque nos dará los inicios a la programación ya que enseñar programación es importante en este mundo tecnológico. La consecuencia de entrelazar las zonas del Scratch y las fases de resolución de problemas de Polya que están muy ligadas a las

fases de la programación , nos da las bases de un pensamiento algorítmico y por ende conlleva a la resolución de un determinado problema y su posterior evaluación-depuración que es vital para una mejor toma de decisiones.

EL AUTOR

I. PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES Y FUNDAMENTACIÓN DEL PROBLEMA

1.1.1 ANTECEDENTES:

Antecedente Internacional:

- ❖ En esta tesis el licenciado afirma que: el SCRATCH contribuye no solo al uso y apropiación del pensamiento algorítmico, sino que se sirve como mediadora tanto para implementar la estrategia de solución de problemas basada en el enfoque de Polya, como para posibilitar que los estudiantes tengan la oportunidad de utilizar conceptos del pensamiento algorítmico. (López García, 2014).
- ❖ En este artículo nos da sus conclusiones que: a partir del estudio efectuado, se considera que el software Scratch posibilita la participación colaborativa y la interactividad ofreciendo un entorno online amigable y robusto para aprender programación de forma exploratoria y creativa a través de proyectos colaborativos, pudiendo a su vez habilitar el desarrollo de habilidades cognitivas. Así mismo se observa que las barreras para la co-construcción de TIS en el contexto educativo no se relacionan con limitaciones de Scratch en sí, sino con la orientación instrumental de las prácticas de enseñanza y aprendizaje que se han puesto en obra. (Monjelat & San Martin, 2016, págs. 61-71).

Antecedente Nacional:

❖ En esta tesis se afirma que: Para medir el pensamiento algorítmico de los alumnos se realizó un primer test donde se muestra los siguientes resultados:

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

Asimismo, se hace una evaluación de salida donde se utilizó el Scratch y Wedo donde el resultado es el siguiente:

El Post Test. Solo un 8,33% obtuvieron un puntaje catalogado como bajo o con muy poco logro. Un 33,33% obtuvieron puntaje catalogado como regular. Mientras un 58,33% lograron una valoración alta en por lo menos en dos conceptos algorítmicos del pensamiento computacional. Finalmente, como corolario de este primer avance, se ha logrado una aproximación a la definición de los componentes del pensamiento computacional, midiéndose el aspecto de conceptos algorítmicos. (Cadillo León J. R., 2017)

Antecedente Regional:

En el repositorio de la UNHEVAL no se encontraron investigaciones donde se usa el lenguaje de programación Scratch o el desarrollo del Pensamiento Algorítmico, pero si había trabajos en donde se usan las fases del método Polya

que se usa para la resolución de problemas y en la presente investigación se usará como parte del desarrollo del pensamiento algorítmico.

❖ Aquí presento las conclusiones de la investigación del repositorio:

Los niños en un principio no entendían, pero luego de aplicar el método Polya a partir de las sesiones 2 comenzaban a familiarizarse e identificaban los pasos para la comprensión del problema y tenían mayor dominio del desarrollo del problema para así dar mejores respuestas y se aseguraban de no equivocarse comprobando y volviendo a revisar los pasos anteriores. (Mingos Echevarría, 2018).

1.1.2 FUNDAMENTACIÓN DEL PROBLEMA:

Se observa que no se desarrolla el pensamiento algorítmico en los estudiantes de Educación Básica Regular; en la actualidad la tecnología es parte de nuestra vida cotidiana en todos los escenarios ya sean políticos, económicos, sociales, culturales, educativos y se utiliza poco o nada los lenguajes de programación en ambientes escolares; las estrategias, metodologías y técnicas de la enseñanza aprendizaje entre el profesor y estudiante no garantizan el desarrollo de capacidades y habilidades tecnológicas que requieren los niños y jóvenes en la actualidad, para desenvolverse en un escenario diferente y resolver problemas; pues existe una falta o desconocimiento de políticas y capacitaciones para mejorar las competencias digitales de los profesores en programación y uso de herramientas tecnológicas que requieren en este mundo digital (Alfabetización

digital). El impacto de la tecnología en la prestación de la educación sigue siendo menos que óptimo, porque podemos estar sobreestimando las competencias digitales de los profesores y estudiantes, debido a estrategias de diseño e implementación de políticas ingenuas, a causa de una mala comprensión de la pedagogía, o debido a la generalmente pobre calidad del software educativo y didáctico. (Schleider, 2015).

El proceso de enseñanza-aprendizaje en programación resulta complejo para los estudiantes de EBR ya que no existe una metodología didáctica e innovadora para comenzar a enseñar programación y esto provocará que los estudiantes no sean competentes en esta era digital. Si se da un inicio en programación en la etapa escolar estaremos dejando a los estudiantes un análisis crítico para la resolución de algoritmos y por ende la forma correcta de secuencializar y dar respuesta a un determinado problema; si los estudiantes no reciben una buena orientación en programación tendrán poca habilidad para desarrollar algoritmos de mediana o alta complejidad, tampoco lograrán desarrollar un modelo viable o estructura que permita resolver el problema y poseer suficiente experiencia en el manejo del lenguaje de programación. El proyecto se trabajará con estudiantes del sexto grado de primaria de la I.E.P. Augusto Cardich, estudiantes que tienen una edad promedio entre 10 a 11 años de edad porque es una edad promedio para el uso del lenguaje de programación Scratch ,como preámbulo, el curso de computación se abrió otra vez en el año 2017, debido a las necesidades que los propios padres de familia hicieron llegar al colegio, considerando que sus hijos necesitan

aprender aspectos básicos de la computación, muchos de los estudiantes consideraban al curso de computación como sinónimo de juegos online , offline , entretenimiento y carecían de orientación pedagógica y metodológica para usar los programas de computación de manera adecuada. El Diseño Curricular Nacional (DCN) tiene el área de Educación por el Trabajo en donde parte de esa área se da el curso de computación; en estos dos años sólo se usaron programas como: Microsoft Office (Word, Power Point, Excel y visio), Gimp, Xmind y Filmora de manera general pero sólo para los alumnos del sexto grado se utilizaron los programas de Microsoft Office (Word, Power Point y Excel) y Gimp. . Es por ello que nace el proyecto de investigación que intenta aplicar el lenguaje de programación Scratch para el desarrollo del pensamiento algorítmico en la Institución, muy aparte de los programas básicos. El uso del lenguaje de programación SCRATCH se adapta al desarrollo de habilidades mentales mediante el aprendizaje de la programación ya que sirve como inicio-punto cero para programar de manera interactiva en donde se puede crear juegos, animaciones, etc. La creación de juegos y aplicaciones tienen un efecto positivo en el desarrollo cognitivo del niño, aprendiendo a programar y mejorando su tiempo de respuesta a la resolución de un problema mediante secuencias de algoritmos (desarrollo del pensamiento algorítmico).

1.2 FORMULACIÓN DEL PROBLEMA.

1.2.1 PROBLEMA GENERAL

¿En qué medida la aplicación del lenguaje de programación SCRATCH influye en el desarrollo del pensamiento algorítmico en los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019?

1.2.2 PROBLEMAS ESPECÍFICOS

✚ ¿En qué medida la aplicación del lenguaje de programación SCRATCH influye en la *fase de análisis del problema* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019?

✚ ¿En qué medida la aplicación del lenguaje de programación SCRATCH influye en la *fase de diseño y traducción del algoritmo* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019?

✚ En qué medida la aplicación del lenguaje de programación SCRATCH influye en la *fase de depuración del programa* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019?

1.3 OBJETIVOS: GENERALES Y ESPECÍFICOS.

1.3.1 OBJETIVO GENERAL

Determinar de qué manera la aplicación del lenguaje de programación SCRATCH influye en el pensamiento algorítmico de los alumnos del 6to grado de primaria del colegio Augusto Cardich-Huánuco 2019.

1.3.2 OBJETIVOS ESPECÍFICOS

- ✚ Determinar de qué manera la aplicación del lenguaje de programación SCRATCH influye en la *fase de análisis del problema* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.
- ✚ Determinar de qué manera la aplicación del lenguaje de programación SCRATCH influye en la *fase de diseño y traducción del algoritmo* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.
- ✚ Determinar de qué manera la aplicación del lenguaje de programación SCRATCH influye en la *fase de depuración del programa* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

1.4 HIPÓTESIS: GENERAL Y ESPECÍFICOS

1.4.1 HIPÓTESIS GENERAL

Ha: La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

Ho: La aplicación del lenguaje de programación SCRATCH No influye positivamente en el desarrollo del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

1.4.2 HIPÓTESIS ESPECÍFICOS

✚ ***HE1a:*** La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo de la *fase de análisis del problema* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

✚ ***HE1o:*** La aplicación del lenguaje de programación SCRATCH No influye positivamente en el desarrollo de la *fase de análisis del problema* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

✚ ***HE2a:*** La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo de la *fase de diseño y traducción del algoritmo* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

- ✚ **HE2o:** La aplicación del lenguaje de programación SCRATCH No influye positivamente en el desarrollo de la *fase de diseño y traducción del algoritmo* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.
- ✚ **HE3a:** La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo de la *fase de depuración del programa* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.
- ✚ **HE3o:** La aplicación del lenguaje de programación SCRATCH No influye positivamente en el desarrollo de la *fase de depuración del programa* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

1.5 VARIABLES

1.5.1 VARIABLE INDEPENDIENTE

Lenguaje de Programación SCRATCH.

1.5.2 VARIABLE DEPENDIENTE

Pensamiento Algorítmico.

1.6 DEFINICIÓN OPERACIONAL DE VARIABLES, DIMENSIONES E INDICADORES

1.6.1 VARIABLE INDEPENDIENTE: LENGUAJE DE PROGRAMACIÓN SCRATCH

Tabla 1: DEFINICIÓN OPERACIONAL DE LA VARIABLE INDEPENDIENTE

VARIABLE INDEPENDIENTE O V1	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
LENGUAJE DE PROGRAMACIÓN SCRATCH	SCRATCH, es un lenguaje de programación educativo gratuito que fue desarrollado por el Lifelong Kindergarten Group en la MIT. Está dirigido a niños de 8-16 años. Scratch está diseñado para ser divertido, educativo, y fácil de aprender. Tiene las herramientas para crear historias interactivas, juegos, arte, simulaciones, y mucho más. Scratch incluso tiene su propio editor de imágenes y editor de sonido incorporado.	Operacionalmente el lenguaje de programación SCRATCH nos permitirá conocer conceptos previos sobre la programación de manera fácil y entretenida.	PLANIFICACIÓN	Elaboración de las sesiones de aprendizaje aplicando el SCRATCH a las dimensiones de la variable Dependiente.
			EJECUCIÓN	Desarrollo de las sesiones de aprendizaje aplicando el SCRATCH a las dimensiones de la variable Dependiente.
				Desarrollo de un juego interactivo

FUENTE: Elaborado por Tesista

1.6.2 VARIABLE DEPENDIENTE: PENSAMIENTO ALGORÍTMICO

Tabla 2: DEFINICIÓN OPERACIONAL DE LA VARIABLE DEPENDIENTE

VARIABLE INDEPENDIENTE O V2	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES	
PENSAMIENTO ALGORÍTMICO	El Pensamiento Algorítmico, se refiere al desarrollo y uso de algoritmos que puedan ayudar a resolver un tipo específico de problema o a	Operacionalmente el pensamiento algorítmico nos permitirá resolver problemas de manera secuencial.	ANALIZA EL PROBLEMA	Comprende el problema (escala de 0 a 2)	PRETEST Y POSTEST
				Define los datos disponibles. (escala de 0 a 2)	
				Define los procesos del problema. (escala de 0 a 2)	

	realizar un tipo específico de tarea.		DISEÑA Y TRADUCE EL ALGORITMO	Representa gráficamente el problema. (escala de 0 a 2)	
				Secuencializa el desarrollo del problema. (escala de 0 a 2)	
				Ordena lógicamente las instrucciones a seguir. (escala de 0 a 2)	
			DEPURA EL PROGRAMA (Validación de resultados)	Verifica la respuesta del problema. (escala de 0 a 2)	
				Propone otras alternativas de solución.	
				Corrige fallas. (escala de 0 a 2)	

FUENTE: Elaborado por Tesista

1.7 JUSTIFICACIÓN E IMPORTANCIA

La presente investigación se **justifica** por cuán importante es desarrollar las competencias, capacidades, desempeños y habilidades informáticas de los estudiantes en edad promedio de 10 a 11 años de edad, ya que los niños de estas edades de las diferentes Instituciones públicas o privadas del Perú, de la Región y del Distrito de Pillco Marca llevan computación como parte de su plan de estudios, pero no utilizan la tecnología de manera adecuada, debido a diferentes factores como el entorno familiar, su propia comunidad educativa; la experiencia como docente de aula de computación ha permitido observar que los estudiantes no llevan programación en el nivel primario ni secundario porque se considera que es muy complejo, solo aprenden aspectos básicos y repiten año tras año las mismas temáticas dejando de lado los indicadores de desempeño, las capacidades, las

competencias y los estándares de aprendizaje que se encuentra plasmado en el DCN - Diseño Curricular Nacional.

Siendo la tecnología hoy en día el eje transversal que se puede aplicar en todas las áreas curriculares la programación SCRATCH permitirá que el estudiante sea creativo, analítico y desarrolle su pensamiento algorítmico la misma que le permitirá resolver problemas y tomar la mejor alternativa en su toma de decisiones y plasmarla en su vida cotidiana. Pues como sugiere el profesor alemán Sven Vosseler recomienda que la introducción a la tecnología empiece con la alfabetización digital. La programación SCRATCH incluso se puede programar teniendo en cuenta los enfoques transversales que se impregnan en las competencias que orienta el trabajo pedagógico de los docentes. Las competencias del área transversal de Educación para el trabajo (Computación), es *se desenvuelve en entornos virtuales generados por las TIC* y sus capacidades son; Personaliza entornos virtuales, Gestiona información del entorno virtual, Interactúa en entornos virtuales, Crea objetos virtuales en diversos formatos. El lenguaje de programación Scratch es **importante** porque nos dará los inicios a la programación ya que enseñar programación es importante en este mundo tecnológico. La consecuencia de entrelazar las zonas del Scratch y las fases de resolución de problemas de Polya que están muy ligadas a las fases de la programación nos dará las bases de un pensamiento algorítmico y esto nos conlleva a una mejor toma de decisiones.

1.8 LIMITACIONES

En el plan de estudios de computación solo se trabaja dos horas pedagógicas que equivale a una hora y 45 minutos por semana, lo que dificulta las actividades pedagógicas y metodológicas para el desarrollo de los contenidos que se realizan para sentar las bases de un pensamiento algorítmico y trabajar programación Scratch en articulación con las competencias, capacidades y desempeños que exige el DCN.

II. MARCO TEÓRICO

2.1 REVISIÓN DE ESTUDIOS REALIZADOS (ANTECEDENTES)

2.1.1 ANTECEDENTES INTERNACIONALES:

❖ El análisis de los resultados permite concluir que, aunque en los datos cuantitativos aparece un claro indicador de cambio en los desempeños de los estudiantes, estos desempeños no explican suficientemente el uso de conceptos del pensamiento algorítmico. Por otra parte, los datos cualitativos sí permitieron establecer un modelo categorial empírico que explica cómo los estudiantes de grado 3° del INSA usan y se apropian de los conceptos del pensamiento algorítmico en un entorno educativo que tiene como eje articulador la metodología de solución de problemas propuesta por Polya.

Se identificó que programar con Scratch no es lo mismo que resolver problemas con Scratch. En este último caso, los estudiantes deben activar estrategias cognitivas, así como usar recursos y conceptos del pensamiento computacional para poder resolverlos. Además, se determinó que Scratch se enmarca en la categoría de “herramientas de la mente” o en la de “auxiliares exteriores”, por cuanto contribuye no solo al uso y apropiación del pensamiento algorítmico, sino que se erige como mediadora tanto para implementar la estrategia de solución de problemas basada en el enfoque de Polya, como para posibilitar que los estudiantes tengan la oportunidad de utilizar conceptos del pensamiento algorítmico. (López García, 2014).

❖ La experiencia muestra que Scratch constituye una herramienta propicia para el desarrollo del pensamiento lógico y algorítmico para niños y estudiantes de Chile, y presenta un ambiente en el cual los estudiantes se motivan y participan en la propuesta de soluciones a las situaciones planteadas sin temor al error, posibilita el análisis de problemas y la propuesta, desarrollo y aplicación de soluciones lógicas y algorítmicas, las que se pueden probar y mejorar. Es decir, mediante pruebas de ensayo y error, los estudiantes pueden desarrollar y mejorar un pensamiento algorítmico. Mediante el uso de objetos, es posible el desarrollo de razonamiento algorítmico, y Scratch permite trabajar directamente con las propiedades y acciones de objetos. De esta forma, Scratch es una herramienta adecuada para la enseñanza de algoritmos y programación. (Cooper, 2003). Justamente, en la realización de este experimento, fue posible comprobar que las alternativas de respuesta a una pregunta lógica, generan el razonamiento del estudiante sobre cuál es la opción correcta así como validar sus soluciones. Es relevante destacar que, para la enseñanza de algoritmos también existen herramientas útiles como PseInt (PseInt, 2015), la cual es muy usada en cursos iniciales de programación ya que se permite definir elementos algorítmicos en español. Primero, gracias a los objetos gráficos, segundo gracias a la nula posibilidad de errores sintácticos, salvo con la inclusión de variables, y por el efecto acción-reacción, este trabajo afirma que Scratch es más adecuado para el desarrollo de

pensamiento algorítmico en niños. Después de este trabajo, que representa una experiencia práctica de enseñanza, aprendizaje y uso de Scratch, para el desarrollo del pensamiento algorítmico, se puede afirmar que, a la luz de los resultados obtenidos, Scratch es una herramienta adecuada para la educación escolar por su versatilidad y posibilidades que ofrece a través del uso de recursos multimedia como sonido, gráfica y movimiento. Desde este punto de vista, permite el uso de técnicas de visualización para conjeturar y experimentar soluciones a problemas de diversa índole. En efecto, el contexto presentado en el experimento que se reporta en este artículo permite, con algunas modificaciones, plantear problemas de tipo geométrico relativos a triángulos u otras figuras planas. (Vidal, Cabezas, Parra, & López, 2015).

❖ (Moreno León, 2016) **nos dice:**

Recientemente, con la aparición de lenguajes de programación visuales basados en bloques, entre los que destaca Scratch, hemos sido testigos de un resurgimiento del uso de la programación en las escuelas. Por todo el mundo, también en España, vemos docentes que han comenzado a utilizar la programación en sus clases. Desde Programamos, convencidos de la potencia de la programación como herramienta para desarrollar el pensamiento computacional y adquirir otras habilidades y competencias, vemos este movimiento con una inmensa alegría. Sin embargo, venimos

detectando desde hace un tiempo algunas señales que nos hacen temer que la historia pueda repetirse y que, tras unos años de protagonismo, el uso de la programación en las escuelas vuelva a desaparecer.

Por ejemplo, escuchamos de boca de docentes que han tenido un cierto contacto con Scratch frases como las siguientes:

- ✓ Scratch es para niños pequeños.
- ✓ Scratch es demasiado simple.
- ✓ Los proyectos que se pueden hacer en Scratch son una chorrada.
- ✓ Scratch sólo sirve para pasear muñecos por la pantalla. Así no se aprende nada.
- ✓ Para trabajar las matemáticas no tiene sentido usa Scratch.

Es evidente que nosotros no estamos de acuerdo con ninguno de estos “mitos”:

- Scratch no es (solamente) para niños pequeños. En muchas universidades se usa como lenguaje de introducción a la programación en los primeros cursos. Por ejemplo, en el curso CS50 de la Universidad de Harvard.
- Scratch no es demasiado simple. Aunque está diseñado para que dar los primeros pasos sea muy sencillo, Scratch permite realizar proyectos realmente avanzados, como esta réplica de Súper Mario, esta de Tetris, o este videojuego de Star Wars por plantear tan solo unos ejemplos.

- “Paseando” personajes por la pantalla se pueden aprender muchas cosas. Por ejemplo, para programar esta versión de Angry Birds se trabajan conceptos como aceleración, gravedad, trayectorias. En nuestro blog hemos hablado muchas veces de cómo incorporar la programación y el pensamiento computacional a las artes o a la física, por ejemplo. Y existen miles de estudios (literalmente) que recopilan proyectos para trabajar conceptos matemáticos o de representaciones gráficas de funciones. Pero además existe evidencia científica de que el uso de la programación como herramienta educativa mejora el aprendizaje de materias como las matemáticas, los idiomas, las ciencias o la narración, entre otras.

Quizás el hecho de que se ha puesto de moda usar la programación y la robótica en la escuela haga que existan centros educativos y administraciones públicas que quieran apuntarse al movimiento, pero sin realizar la inversión necesaria en formación para que los docentes aprendan a programar y conozcan estrategias de introducción de la programación en diferentes niveles y disciplinas.

El problema es que, como es lógico y como se ha visto en estudios científicos, cuando los docentes no están bien formados y se lanzan a usar la programación en sus clases, no se consiguen los objetivos previstos, e incluso se produce un impacto negativo en los resultados de aprendizaje. Y por supuesto, los estudiantes se aburren muy pronto de mover al gatito de

Scratch por la pantalla de un lado a otro y de mostrar bocadillos de texto. Desde Programamos consideramos fundamental, por tanto, concienciar a las administraciones públicas de que, si se desea que los docentes puedan incorporar esta herramienta a sus clases con garantías, no hay otra alternativa que ofrecerles una formación de calidad. De este modo estaremos sentando las bases para tratar de que, al contrario de lo que ocurrió hace 20 años, la programación y el pensamiento computacional no desaparezcan en esta ocasión de la agenda educativa tras unos años de protagonismo. EL Scratch rompe algunos paradigmas que los docentes tenían en España y define como Scratch como una herramienta importante para las estrategias creativas en clases y sus proyectos pueden ser grandes para que así la programación y el pensamiento computacional tengan mayor demanda.

❖ (Coronel, 2013) nos comenta que : **Una experiencia con Scratch en clase de 1º de primaria se da de la siguiente manera:**

En esta entrada vamos a contaros una experiencia que hemos tenido esta semana en la que hemos puesto en marcha este proyecto con un curso de 1º de primaria. Cuando el alumnado con el que se trabaja es muy pequeño y no tiene experiencia en el diseño y la programación con Scratch, comprender la diferencia entre crear y jugar no es tan fácil. Se observa que la experiencia que tienen con los aparatos electrónicos es

fundamentalmente pasiva, reciben información o siguen unas instrucciones para poder jugar. Es por eso que en sus primeros contactos con Scratch tienden a confundirlo con un juego y su primer impulso es intentar averiguar cómo se juega, qué hace el gato, a qué botón hay que darle para empezar. Cuando descubren que lo que realmente pueden hacer es crear sus propios juegos, el interés, la emoción y la curiosidad se multiplican: ¡están ante un juego que hace juegos!

En esta experiencia se ha trabajado con un grupo pequeño, seis niños de seis años. Hemos partido del interés que tenían por hacer algo con Scratch. Lo habían visto en alguna ocasión pero tan solo como observadores. Ahora se les planteaba la posibilidad de crear. El primer paso fue darnos cuenta de que con los medios que teníamos (un ordenador para todos) y la ayuda que iban a necesitar (para leer y comprender algunos bloques) no podíamos hacer lo que cada uno quisiera, por lo que nuestro primer paso fue diseñar una idea y tratar de convencer a los compañeros para que la eligieran. Estos fueron los resultados:

1. “Un avión que vuela bien y derecho por la noche.”
2. “Un murciélago que mueve las alas”.
3. “Dos murciélagos jugando a la pelota en la playa”
4. “Dos mariposas jugando en la playa a la pelota con un grano de arena”
5. “Una mariposa paseando por el cielo”

6. “Un monstruo que anda”

Para la toma de decisiones fue necesaria una votación en la que cada candidato podía escoger los dos proyectos que más le gustasen (por supuesto, el suyo propio y otro más). Lo decidimos a través de un peculiar sistema de puntos: llenando de lunares los números de nuestros proyectos. Ya teníamos ganador: ¡el número 4! Aunque con una condición, es mejor que el grano de arena sea una pelota.

El siguiente paso es ponernos manos a la obra. Abrimos el programa y analizamos su estructura. Vemos que está separado en tres partes: una donde tenemos disponibles todos los bloques organizados por colores y que cada color contiene bloques de contenido similar. Otra donde colocamos los bloques que van a controlar a nuestro personaje y, por último, la pantalla donde vemos lo que pasa. Ahora lo más importante es seleccionar a nuestros personajes, dos mariposas y una pelota:

Nos llevamos una gran sorpresa al seleccionar la pelota. ¡Llegó con un código que la hacía moverse por toda la pantalla! Y entonces empezaron a demandar lo que le faltaba para que fuera más real:

- Las mariposas tienen que ser más pequeñas y tienen que mover las alas.
- Hay que darle la vuelta a una de ellas porque se tienen que mirar.

- La pelota tiene que ser más pequeña y no puede ir sin rumbo, se la tienen que pasar.
- Y nos hace falta un fondo de una playa. Alguien incluso añade que lo podríamos dibujar.
- Como el tiempo se acababa, elegimos el fondo de la playa que encontramos en la carpeta, adaptamos el tamaño de los objetos y conseguimos que una de las mariposas moviese las alas “como si volara”, aunque aún nos falta que se mueva.
- Ha sido una sesión de trabajo que nos ha hecho pensar, aceptar las ideas de los compañeros y ser un poco más conscientes de lo difícil que es hacer que nuestros personajes cumplan nuestras órdenes. Nos queda mucho que pensar para el próximo día, pero ya hemos construido un poco de nuestro primer trabajo, que ya hemos compartido con la comunidad Scratch.

Esta experiencia que narra la autora se llega a la conclusión que los niños consiguen en trabajar en equipo, les interesa el programa pero jugar y crear sus propias animaciones son cosas muy diferentes.

2.1.2 ANTECEDENTE NACIONAL:

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

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

El Post Test. Solo un 8,33% obtuvieron un puntaje catalogado como bajo o con muy poco logro. Un 33,33% obtuvieron puntaje catalogado como regular. Mientras un 58,33% logaron una valoración alta en por lo menos en dos conceptos algorítmicos del pensamiento computacional. Finalmente, como corolario de este primer avance, se ha logrado una aproximación a la definición de los componentes del pensamiento computacional, midiéndose el aspecto de conceptos algorítmicos. Respecto al marco de trabajo de Scratch y WeDo se ha realizado un esfuerzo por marcar futuras tendencias que podrán ser tomadas en otros estudios como son la consideración de los entornos de trabajo y los niveles de reelaboración de contenidos a partir de las herramientas trabajadas. (Cadillo León J. , 2017).

2.1.3 ANTECEDENTE REGIONAL

En el repositorio de la UNHEVAL no se encontraron investigaciones donde se usa el lenguaje de programación Scratch o el desarrollo del Pensamiento Algorítmico, pero si había trabajos en donde se usan las fases del método Polya que se usa para la resolución de problemas y en la presente investigación y se usará como parte del desarrollo del pensamiento algorítmico.

Aquí presento las conclusiones de la investigación del repositorio:

- ❖ Los niños en un principio no entendían, pero luego de aplicar el método Polya a partir de las sesiones 2 comenzaban a familiarizarse e identificaban los pasos para la comprensión del problema y tenían mayor dominio del desarrollo del problema para así dar mejores respuestas y se aseguraban de no equivocarse comprobando y volviendo a revisar los pasos anteriores.

(Mingos Echevarría, 2018).

2.2 PRINCIPALES LEYES, DEFINICIONES Y CONCEPTOS FUNDAMENTALES

La determinación de los temas ejes o teorías generales que formarán parte de la estructura del marco teórico, es una tarea primordial que el investigador debe realizar con sumo cuidado y fina habilidad. (Carrasco Díaz, 2005)

Tabla 3: PRINCIPALES LEYES, DEFINICIONES Y CONCEPTOS FUNDAMENTALES

VARIABLES	TEMAS EJES
Variable Independiente: LENGUAJE DE PROGRAMACIÓN SCRATCH	Historia del lenguaje de programación Scratch.
	Concepto y usabilidad del lenguaje de programación Scratch.
	Interfaz y zonas del lenguaje de programación Scratch.
	Scratch en la Educación.
	Ventajas y Desventajas del Scratch.
Variable Dependiente: PENSAMIENTO ALGORÍTMICO	Desarrollo de habilidades de Pensamiento de Orden Superior.
	Solución de Problemas.
	Operaciones mentales para la solución de problemas.
	Solución de problemas y programación.
	Pensamiento algorítmico.
	Importancia y trascendencia del pensamiento algorítmico.
	Herramientas para promover el Pensamiento Algorítmico.

FUENTE: Elaborado por Tesista

2.2.1 TEMAS EJES DE LA VARIABLE INDEPENDIENTE- LENGUAJE DE PROGRAMACIÓN SCRATCH.

2.2.1.1 Historia del Lenguaje de Programación Scratch.

- ❖ Según (Resnick, y otros, 2009); nos dice que el Scratch es un proyecto del Grupo Lifelong Kindergarten del MIT Media Lab. Su inicio nace con una pregunta:

¿Por qué programar?

Todos consideran a los niños de hoy en día como “Nativos Digitales” debido a su aparente fluidez con lo digital. Aunque se interactúa con medios digitales todo el tiempo, pocos son capaces de crear sus propios juegos, animaciones o simulaciones. Es como si pudiera leer pero no escribir. No sólo se trata de chatear y navegar sino también de modificar y crear. La programación soporta el pensamiento computacional, ayudándote a aprender la resolución de un problema y estrategias de diseño. Cuando las computadoras personales aparecieron, a finales de los años 70 y 80 hubo un entusiasmo inicial por enseñar a todos los niños a programar, miles de escuelas enseñaron a millones de estudiantes a escribir programas simples. Hoy en día la mayoría de la gente ve a la programación informática como un estrecho.

¿Pero porque no funcionó el entusiasmo inicial para introducir a la programación?

Factores:

- ✓ Lenguaje de Programación Temprana, demasiado difícil de usar, muchos niños no podían dominar la sintaxis.

- ✓ La programación a menudo se introdujo con actividades que no eran interesantes a los alumnos (generar listas de números primos y hacer dibujos lineales).
- ✓ Orientación inadecuada.

Papert (2007), argumentó que la programación debe tener un piso bajo (fácil de comenzar), techo alto (oportunidades para crear proyectos más complejos y paredes anchas (apoyando diferentes proyectos, estados de aprendizaje y estilos).

No aceptaban las opciones de ese entonces así que Resnick y su equipo querían hacer el piso aún más bajo y las paredes aún más ancha.

❖ **Principios básicos para el diseño del Scratch:**

En palabras de Resnick (2009), los tres principios clave del diseño de Scratch son:

- Make it more tinkerable (Hacerlo más experimentable).
- More meaningful (más significativo).
- More social than other programming environments (más social que otros entornos de programación).

El diseño de la interfaz de usuario de Scratch surgió de un deseo de hacer sus conceptos clave tan tangibles y manifiestos como fuera posible. (Maloney, Peppler, Kafai, Resnick, & Rusk, 2008)

Scratch y lego han trabajado estrechamente. El Lego consistía en una cajita de pequeños ladrillos donde los niños inmediatamente comenzaban a jugar encajando y con muchas posibilidades de nuevas ideas de construcción. El proceso de jugar, construir, planes, metas y que evolucionen lo quería ver reflejado en Scratch. El nombre de Scratch viene de la técnica de Rascado de los disc Jockeys de hip hop que juegan con la música, mezclando clips musicales de manera creativa. La programación de Scratch es similar, mezclando gráficos, animaciones, fotos, música y sonido; de ahí nace lo que hoy es el lenguaje de programación Scratch.

Un investigador perteneciente a esta institución Nael Gershenfeld dijo lo siguiente: "Cuando la gente puede crear su propia tecnología es cuando se despierta la pasión". Cabe mencionar que esta aplicación fue desarrollada con fondos de la Fundación Nacional para la Ciencia.

❖ **Kindergarten es el modelo para el aprendizaje de por vida**

Sigamos enseñando creatividad a lo largo de la escuela y la edad adulta. En palabras de (Resnick M. , 2009) desde que abrió el primer jardín de infantes en 1837, ha sido un lugar para contar historias, construir castillos, pintar cuadros, hacer amigos y aprender a compartir. Pero el jardín de infancia está experimentando un cambio dramático.

En los jardines de infancia de hoy, los niños pasan cada vez más tiempo llenando hojas de trabajo y perforando tarjetas de memoria flash. En resumen, el jardín de infancia se está pareciendo más al resto de la escuela. Exactamente lo contrario debe suceder: debemos hacer que el resto de la escuela (de hecho, el resto de la vida) se parezca más al jardín de infancia. ¿Qué tiene de especial el jardín de infancia? A medida que los estudiantes de kinder crean con alegría cuentos, castillos y pinturas, desarrollan y refinan sus habilidades para pensar creativamente y trabajar en colaboración, precisamente las habilidades más necesarias para lograr el éxito y la satisfacción en el siglo XXI. Las actividades subyacentes del jardín de infancia tradicional son un proceso de aprendizaje en espiral en el que los niños se imaginan lo que quieren hacer, crean un proyecto basado en sus ideas (utilizando bloques, pintura con los dedos u otros materiales), juegan con sus creaciones, comparten sus ideas y creaciones con otros. Reflexionar sobre sus experiencias, todo lo cual los lleva a imaginar nuevas ideas y nuevos proyectos. Este proceso de aprendizaje iterativo es la preparación ideal para la sociedad actual que cambia rápidamente, en la que las personas deben encontrar continuamente soluciones innovadoras para situaciones inesperadas en sus vidas.

Si este enfoque está tan bien alineado con las necesidades actuales de la sociedad, ¿por qué rara vez lo apoyamos en las aulas? Una razón es

que nuestra sociedad y nuestro sistema educativo no dan suficiente valor al pensamiento creativo. Otra razón es la falta de medios y tecnologías apropiados: los bloques de madera y la pintura para dedos son excelentes para aprender conceptos de jardín de infantes (como números, formas, tamaños y colores). Pero a medida que los niños crecen, quieren y necesitan trabajar en proyectos más avanzados y aprender conceptos más avanzados. Para hacer eso, necesitan diferentes tipos de herramientas, medios y materiales. Aquí es donde creo que las tecnologías digitales pueden desempeñar su papel más importante. Si se diseñan y usan adecuadamente, las nuevas tecnologías pueden extender el enfoque de kindergarten, permitiendo a "estudiantes" de todas las edades continuar aprendiendo al estilo de kindergarten y, en el proceso, seguir creciendo como pensadores creativos. En mi grupo de investigación en el Laboratorio de Medios en el Instituto de Tecnología de Massachusetts, hemos estado desarrollando nuevas tecnologías específicamente para apoyar el enfoque de aprendizaje de kindergarten. Por ejemplo, hemos colaborado con el Grupo Lego desde 1985 en una colección de kits de construcción de robótica que permiten a los niños imaginar y crear invenciones interactivas con el mismo espíritu con que los alumnos de kindergarten construyen torres con bloques. Recientemente, hemos desarrollado un nuevo lenguaje de programación llamado Scratch, que trae el enfoque de aprendizaje de

kindergarten a la pantalla de la computadora. Con Scratch (disponible como descarga gratuita), los niños pueden crear sus propias historias, juegos y animaciones interactivas y luego compartir sus creaciones en la Web. Una participante activa en la comunidad es una niña de 13 años con el nombre de BalaBethany (no es su nombre real). Como su primer proyecto Scratch, BalaBethany creó y compartió una historia animada con personajes de anime. Otros miembros de la comunidad en línea respondieron positivamente, publicando comentarios brillantes bajo su proyecto. Alentada, BalaBethany comenzó a crear y compartir nuevas historias de anime de forma regular, como episodios de una serie de televisión. BalaBethany periódicamente agregó nuevos personajes a sus historias. En un momento dado, tuvo una idea: ¿por qué no involucrar a la comunidad en el proceso? Creó y subió un nuevo proyecto Scratch que anunció un concurso: pidió a otros miembros de la comunidad que diseñaran una hermana para uno de los personajes. El proyecto recibió más de 100 comentarios. Uno era de un miembro de la comunidad que quería participar en el concurso, pero no sabía cómo dibujar personajes de anime. Así que BalaBethany produjo otro proyecto Scratch: un tutorial paso a paso que muestra un proceso de 13 etapas para dibujar y colorear un personaje de anime. Veo a BalaBethany como un estudio de caso de un jardín de infantes de por vida. Ella está usando Scratch para imaginar, crear, jugar, compartir y reflexionar. En el proceso, ella

está refinando sus habilidades de programación y artísticas, y se está desarrollando como un pensador creativo. Nuestro objetivo final es un mundo lleno de gente lúdicamente creativa que, como BalaBethany, continúe aprendiendo como lo hacen los estudiantes de kindergarten.

❖ **Versiones del Scratch:**

El día 18 de mayo de 2007 aparece por primera vez en el mundo cibernético Scratch, considerada en ese entonces como una herramienta que permite hacer “animaciones fáciles a base de ladrillos”. Luego de eso se da **Scratch 1.4** (comúnmente conocida como la era 1.4) fue la última versión de la serie 1.x. Se lanzó oficialmente el 2 de julio de 2009. Así mismo el **Scratch 2.0** fue anunciado por Andresmh en los foros de Scratch en enero de 2010. El primer experimento lanzado por el Scratch Team como parte del desarrollo de 2.0 fue el Visor Experimental , en agosto de 2010. Más tarde, en 2011, se lanzó un Flash Player beta. Proyectos, que los usuarios registrados podrían optar por utilizar. En 2012, esto fue reemplazado por una versión del editor alfa; Esta versión se hizo por defecto para todos los usuarios ese mes de octubre. En mayo de 2011, la primera versión conocida del editor del proyecto, prealpha, se lanzó a una audiencia limitada en Scratch Day @ MIT . Poco después, el Equipo de Scratch comenzó a publicar actualizaciones en su blog, llamadas Scratch 2.0

Progress Reports , en su blog. El nuevo sitio web y el rediseñado editor de proyectos, para entonces en la etapa alfa (en alpha.scratch.mit.edu), se estrenaron durante unos días al público para Scratch Day 2012, un año después. Durante el resto de 2012, se invitó a personas a probar esta versión: moderadores comunitarios y educadores seleccionados; Collab Consejeros , ex curadores , comisarios de Scratch Design Studio , TBGModeradores, y un grupo de 500 voluntarios. Algunos usuarios también pudieron infiltrarse y usar el programa debido a un problema técnico. En diciembre de 2012, se anunció que la versión beta pública comenzaría el 28 de enero de 2013. Estaba disponible en beta.scratch.mit.edu desde entonces hasta su lanzamiento completo, en el que reemplazó el sitio web anterior. El 13 de mayo de 2014, se lanzó el código fuente de Scratch 2.0. **Scratch 3.0** es la tercera y actual versión principal de Scratch . Es un completo rediseño y re implementación de Scratch escrito en HTML5 y JavaScript. Cuenta con un aspecto y diseño nuevos y modernos con el escenario movido hacia la derecha como Scratch 1.4 y anteriores. Fue lanzado el 2 de enero de 2019. Además, **Scratch 3.0** cambia los mapas de bits por los gráficos vectoriales para que los objetos no se pixelen si se amplían. Esta función, que aporta nitidez a los proyectos nuevos, cobra aún más sentido con la ampliación de los

personajes, algo que otorga creatividad al programa y que lo hace aún más atractivo para los estudiantes de los primeros niveles educativos.

En la presente investigación se usará en lenguaje de programación **Scratch 2.0**. (Wikipedia, s.f.).

2.2.1.2 Concepto y usabilidad del Lenguaje de Programación Scratch

Respecto al concepto de Scratch en su página principal que es oficial vía internet se considera que el Scratch es un lenguaje de programación educativo gratuito que fue desarrollado por el Lifelong Kindergarten Group en el Instituto de Tecnología de Massachusetts (MIT) con más de 9 millones de usuarios registrados y 12 millones de proyectos compartidos. Está dirigido a niños de 8-16 años. Scratch está diseñado para ser divertido, educativo, y fácil de aprender. Tiene las herramientas para crear historias interactivas, juegos, arte, simulaciones, y mucho más. Scratch incluso tiene su propio editor de imágenes y editor de sonido incorporado. Los usuarios del programa en Scratch deben arrastrar los bloques de la paleta de la izquierda y unirlos a otros bloques como un rompecabezas. Estructuras de varios bloques se llaman scripts. Este método de programación (código de construcción con bloques) se conoce como "Programación agarrar y soltar" (drag-and-drop programming).

❖ Usabilidad del Scratch

El concepto que nos da la misma página web oficial de scratch es la siguiente:

Con Scratch puedes programar tus propias historias interactivas, juegos y animaciones y compartir tus creaciones con otros en la comunidad online. Scratch ayuda a los jóvenes a aprender a pensar de forma creativa, a razonar sistemáticamente, y a trabajar de forma colaborativa y mejorar las habilidades esenciales para la vida en el siglo 21. Scratch es utilizado en más de 150 países y está disponible en más de 40 idiomas. Scratch es usado en escuelas de todo el mundo para aprender programación básica a niños, pero también es usado fuera de escuelas. Los niños y adultos ganan el conocimiento de la programación fundamental para Scratch, y después aprenden otros lenguajes de programación. Mientras usen Scratch, la gente puede crear, reinventar y colaborar con otros con proyectos de Scratch.

❖ **¿Quién utiliza Scratch?**

El concepto que nos da la misma página web oficial de scratch es la siguiente:

Scratch es utilizado por personas de todos los orígenes, en todos los países del mundo, en todo tipo de entornos: hogares, escuelas, bibliotecas, museos y más. Scratch está diseñado especialmente para jóvenes de 8 a 16 años, pero personas de todas las edades crean y

comparten con Scratch. Los niños más pequeños pueden querer probar ScratchJr, una versión simplificada de Scratch diseñada para las edades de 5 a 7 años.

¿Cuánto cuesta Scratch? ¿Necesito una licencia?

El concepto que nos da la misma página web oficial de scratch es la siguiente:

Scratch es y siempre será gratis. No necesita una licencia para usar Scratch en su escuela, su hogar o en cualquier otro lugar. El desarrollo y mantenimiento de Scratch se paga mediante subvenciones y donaciones.

2.2.1.3 Interfaz y Zonas del Lenguaje de Programación Scratch

ConectaCode-La Fundación Esplai (**ConectaCODE** es un proyecto que se ejecuta desde la línea de innovación Tecnologías de la Información y la Comunicación (TIC) de **Red Conecta** y a través del cual se promueve la introducción de la programación y la creación de código en edades tempranas en el telecentro, apoyando así usos más creativos de la tecnología, y potenciando que niños y jóvenes desarrollen estrategias y competencias para la resolución de problemas y la creación de sus propios programas informáticos). Nos dice que:

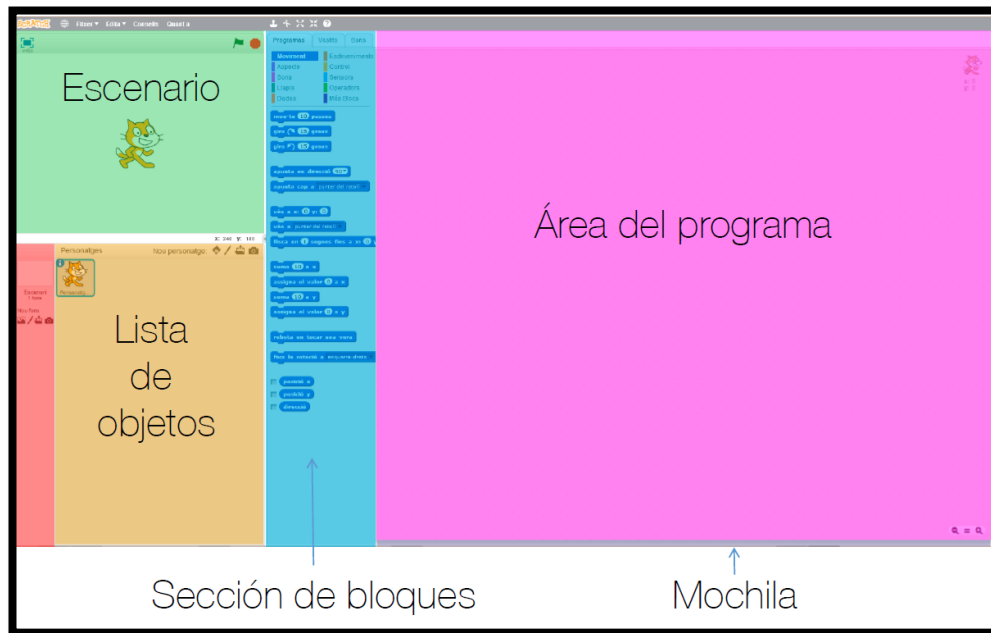
La Fundación Esplai ha desarrollado una metodología y contenidos formativos adecuados para, primero, capacitar a los dinamizadores de los

telecentros, con formaciones que van desde cómo organizar el club de código e introducir los aspectos básicos de la programación, o aprender herramientas actuales para el desarrollo de programas sin necesidad de teclear código, como son Scratch, App Inventor o Windows App Studio.

Clasificó el Scratch en 5 zonas:

- ❖ **Escenario:** Zona del entorno donde se ejecutan los proyectos.
- ❖ **Lista de objetos:** Zona del entorno donde se muestran los objetos añadidos al proyecto.
- ❖ **Sección de Bloques:** Los bloques que representan las instrucciones se encuentran en la paleta o sección de bloques (pestañas programas).
- ❖ **Área del Programa:** Zona donde se arrastran los bloques para que los objetos ejecuten dichas órdenes.
- ❖ **Mochila:** Esta Zona se utiliza para copiar programas, objetos, sonidos... entre otros.

Ilustración 1: ZONAS DEL SCRATCH



Elaborado por: ConectaCode-La Fundación Esplai (2017)

2.2.1.4 Scratch en la Educación

❖ Aprendiendo Con Scratch

¿Qué aprenden los jóvenes a medida que crean historias interactivas, animaciones, juegos, música y arte con Scratch? Por un lado, aprenden ideas matemáticas y computacionales que están integradas en la experiencia Scratch. A medida que los estudiantes crean programas en Scratch, aprenden conceptos computacionales básicos como iteración y condicionales. También obtienen una comprensión de conceptos matemáticos importantes como coordenadas, variables y números aleatorios. Significativamente, los estudiantes aprenden estos conceptos

en un contexto significativo y motivador. Cuando los estudiantes aprenden sobre las variables en las clases de álgebra tradicionales, generalmente sienten poca conexión personal con el concepto. Pero cuando aprenden acerca de las variables en el contexto de Scratch, pueden usarlas de manera muy significativa: para controlar la velocidad de una animación o para realizar un seguimiento de la puntuación en un juego que están creando. A medida que los alumnos trabajan en proyectos Scratch, también aprenden sobre el proceso de diseño. Por lo general, un estudiante comenzará con una idea, creará un prototipo funcional, experimentará con él, lo depurará cuando las cosas salgan mal, obtendrá comentarios de otros, luego lo revisará y lo rediseñará. Es una espiral continua: crea una idea, crea un proyecto, que conduce a nuevas ideas, que conducen a nuevos proyectos, y así sucesivamente. Este proceso de diseño de proyecto combina muchas de las habilidades de aprendizaje del siglo XXI que serán críticas para el éxito en el futuro: pensar creativamente, comunicarse claramente, analizar de manera sistemática, colaborar de manera efectiva, diseñar de manera iterativa, aprender continuamente. La creación de proyectos en Scratch también ayuda a los estudiantes a desarrollar un nivel más profundo de fluidez con la tecnología digital. ¿Qué entendemos por fluidez? Para ser considerado fluido en inglés, español u otro idioma, debe aprender no solo a leer, sino también a escribir, es decir, a expresarse con el idioma.

De manera similar, para dominar la tecnología digital, debe aprender no solo a interactuar con la computadora, sino también a crear con ella. Por supuesto, la mayoría de los estudiantes no crecerán para convertirse en programadores profesionales, al igual que la mayoría no se convertirá en escritores profesionales. Pero aprender a programar ofrece beneficios para todos: permite a los estudiantes expresarse de manera más completa y creativa, les ayuda a desarrollarse como pensadores lógicos y les ayuda a comprender el funcionamiento de las nuevas tecnologías que encuentran en todas partes en su vida cotidiana.

❖ **Programación y educación**

Partiendo de los fundamentos teóricos del enfoque socio-técnico (Fressoli & Santos, 2012) y, el marco teórico-metodológico del Dispositivo Hipermedial Dinámico refiere que:

Los discursos y directrices de los desarrolladores de Scratch para la enseñanza y aprendizaje de la programación, posibilitan promover Prácticas Educativas Mediatizadas (PEM) que viabilicen la co-construcción de Tecnologías para la Inclusión Social (TIS) en contextos tanto formales como no formales. Por otra parte, en el contexto educativo formal iberoamericano, diferentes países están priorizando la introducción de la programación en sus programas educativos: España incluyó por Decreto 48/2015 la asignatura Programación dentro del

currículo del nivel secundario, Uruguay ha puesto en Marcha en Plan Ceibal (Kereki y Oliveira, 2013).

Papert (1980) proponía con su proyecto “Logo” acercar a los niños y jóvenes a la programación justificando su potencial para el desarrollo cognitivo. Según Brennan (2013), los lenguajes inicialmente utilizados resultaban de difícil comprensión para los niños y jóvenes por su sintaxis, y la programación era por lo general introducida con actividades desconectadas de los intereses de los destinatarios, en contextos donde no se ofrecía guía frente a los errores, ni se promovía una mayor exploración. Actualmente, Scratch se utiliza para enseñar programación a personas de todas las edades en más de 30 países en diversas modalidades educativas. Según estadísticas recientes del MIT, se registran casi 10 millones de proyectos y realizados en múltiples idiomas, creados por unos 7 millones de usuarios de entre 4 y 75 años. Sin embargo, se advierte que si bien se plantea una metodología de trabajo por proyectos, un número significativo de cursos de programación con Scratch ofrecidos en el contexto iberoamericano, están articulados sólo en los tutoriales y guías que proporciona la plataforma. Los mismos se centran básicamente en realizar distintas actividades como dibujar un logo, crear juegos de preguntas y respuestas directas de contenido curricular, sonorizar un escenario, descubrir formas u otras producciones similares vinculadas a

secuencias lógicas. En este sentido, cabe interrogar si es posible poner en obra con esta herramienta un proceso activo y dinámico de co-construcción de las tecnologías donde sea posible también atender y ser protagonista de la búsqueda de soluciones a los problemas del propio contexto comunitario.

2.2.1.5 Ventajas y Desventajas del Scratch

❖ Según (Jiménez, 2016) nos dice:

Ventajas

- ✓ Permite el desarrollo de los procesos de pensamientos y habilidades mentales en los educandos
- ✓ Es un programa gratuito y de software libre.
- ✓ Es perfecto para introducirse en la programación.
- ✓ Está disponible para varios sistemas operativos.
- ✓ Permite compartir los proyectos a través del web, se pueden descargar y utilizar. Pudiendo ser descargados y utilizados por otras personas.
- ✓ Es multilinguaje.

Desventajas

Los inconvenientes del uso de Scratch son escasos. El primero de ellos es que es necesario tener instalado Java en el ordenador para poder comenzar a utilizar la aplicación, así que antes de proceder a la descarga

de Scratch y su instalación, hay que asegurarse de que Java está instalado en el ordenador.

❖ Según (SCRATCH, s.f.) nos dice:

Ventajas del scratch

- Las aplicaciones Web 2.0 posibilitan una interacción social nunca antes vista, pero requieren millones de líneas de código para hacerla realidad. Las computadoras son omnipresentes en la sociedad contemporánea, por lo tanto, no es suficiente saber utilizar las herramientas básicas de estos sino que cada vez se hace más necesario tener conocimientos de programación que permitan a las personas comprender cómo funcionan las máquinas y los sistemas que tiene a su alrededor.
- En este contexto, es indudable que herramientas como Scratch contribuyen efectivamente a preparar a los estudiantes desde la primaria, para que puedan insertarse activamente al mundo altamente programado que los espera. Pero va más allá; Scratch promueve el desarrollo de algunas habilidades y capacidades intelectuales de orden superior que, en el nivel escolar, son responsabilidad de cualquier sistema educativo de calidad.

- La programación de computadores posibilita no solo activar una amplia variedad de estilos de aprendizaje sino desarrollar **pensamiento algorítmico**. Además, compromete a los estudiantes en considerar varios aspectos importantes para solucionar problemas: decidir sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo y, monitorear sus propios pensamientos (metacognición) y estrategias de solución utilizadas. Incluso, compromete a los jóvenes en la búsqueda de soluciones innovadoras a problemas inesperados; no se trata solamente de aprender a solucionar problemas de manera predefinida, sino estar preparado para generar nuevas soluciones a medida que se presentan los problemas.
- Otro conjunto de habilidades que se pueden desarrollar con la programación, es el identificado por el Consorcio de Habilidades Indispensables para el Siglo XXI, habilidades de aprendizaje que serán fundamentales para el éxito de los estudiantes en el futuro próximo: pensar creativamente, comunicar claramente, analizar sistemáticamente, colaborar efectivamente, diseñar iterativamente y aprender continuamente.
- Es un programa gratuito, de software libre.
- Es perfecto para enseñar y aprender a programar.

- Está disponible para varios sistemas operativos, Windows, Mac y Linux.
- Permite compartir los proyectos a través de Internet, pudiendo ser descargados y utilizados por otras personas.
- Es multilinguaje.

Desventajas

Hasta la versión 1.4 de este entorno encontramos que no soporta:

- métodos, que permitan pasar el control de la ejecución de una secuencia de bloques a otra;
- parámetros, que permitan influenciar el comportamiento de los métodos;
- valores de retorno, que permitan a una secuencia de bloques “devolver” información a otra;
- herencia y polimorfismo, que permitan la existencia de relaciones entre estructuras de datos.

Por supuesto todas estas fueron solucionadas en la versión 2.0.

- ❖ Según (Wikipedia, s.f.) señala que las:

Ventajas

- Permite el desarrollo de los procesos de pensamientos y habilidades mentales en los educados.
- Es un programa gratuito y de software libre.
- Es perfecto para introducirse en la programación.
- Está disponible para varios sistemas operativos.
(Windows, Ubuntu, Sugar, Mac)
- Permite compartir los proyectos a través del web, se pueden descargar y utilizar. Pudiendo ser descargados y utilizados por otras personas.
- Es multilinguaje.
- Educativamente hablando, los beneficios son una gran cantidad pero se podría destacar el desarrollo del pensamiento lógico en el alumnado, fomenta la creatividad, mejora la habilidad de comprensión de los niños, facilita el pensamiento sistémico y, en general, mejora el rendimiento escolar.

2.2.2 TEMAS EJES DEL PENSAMIENTO ALGORÍTMICO (VARIABLE INDEPENDIENTE).

2.2.1.1 Desarrollo de habilidades de Pensamiento de Orden Superior

Existe actualmente un consenso general dentro de la comunidad educativa mundial sobre la necesidad de superar el tipo de enseñanza basada en la

transmisión de contenidos para apuntarle en su lugar al desarrollo de capacidades. Investigaciones y estudios recientes proponen diversos conjuntos de habilidades que la educación debe fomentar para que los estudiantes puedan tener éxito en el mundo digital y globalizado en el que van a vivir. Este planteamiento exige, sin dilaciones, implementar estrategias que contribuyan efectivamente en el desarrollo de esas habilidades planteadas como fundamentales para la educación en el Siglo XXI (Century Skills, 2004). En la mayoría de conjuntos de habilidades propuestos figuran las habilidades de pensamiento de orden superior entre las que se incluye la destreza para solucionar problemas; por esta razón, se requiere seleccionar estrategias efectivas para ayudar a que los estudiantes las desarrollen. Para atender esta necesidad, la programación de computadores constituye una buena alternativa, siempre y cuando se la enfoque al logro de esta destreza y no a la formación de programadores. Es importante insistir en esta orientación debido a que las metodologías utilizadas en Educación Básica para llevar a cabo cursos de Algoritmos y Programación, son heredadas de la educación superior y muchos de los docentes que las utilizan se dedican principalmente a enseñar los vericuetos de lenguajes de programación profesionales tales como Java, C++, Visual Basic, etc. Hablar hoy de aprender a diseñar y construir aplicaciones (programas) complejas, implica una labor titánica que en la mayoría de los casos está fuera del alcance de la Educación Básica ya que demanda necesariamente enfoques de programación como el orientado a objetos al que

apuntan la mayoría de tendencias en Ingeniería de Sistemas. Por esta razón, en la Educación Básica es altamente recomendable utilizar ambientes de programación basados en Logo, fáciles de utilizar y que permitan realizar procedimientos que contengan estructuras básicas (secuencial, decisión y repetición), pero siempre conducentes al desarrollo de habilidades del Siglo XXI. Solo en los últimos grados de básica secundaria o en la Media Técnica sería aconsejable introducir a los estudiantes a la programación orientada a objetos mediante entornos de programación visuales y amigables como Alice, KPL o Processing. Desde el punto de vista educativo, la programación de computadores posibilita no solo activar una amplia variedad de estilos de aprendizaje (Stager, 2003). Sino desarrollar el pensamiento algorítmico. Adicionalmente, compromete a los estudiantes en la consideración de varios aspectos importantes para la solución de problemas: decidir sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo y, monitorear sus propios pensamientos (metacognición) y estrategias de solución. Este último, es un aspecto que deben desarrollar desde edades tempranas. No debemos olvidar que solucionar problemas con ayuda del computador puede convertirse en una excelente herramienta para adquirir la costumbre de enfrentar problemas predefinidos de manera rigurosa y sistemática, aun, cuando no se utilice un computador para solucionarlo. Esto en cuanto a la solución de problemas, pero hay otra habilidad de pensamiento que también se puede ayudar a desarrollar con unos cursos de Algoritmos y

Programación: La Creatividad. En los últimos años, la creatividad forma parte de las prioridades de los sistemas educativos en varios países, junto a otras habilidades de pensamiento de orden superior. Al punto que los Estándares Nacionales Norteamericanos de TIC para Estudiantes (NETS-S) formulados en 1998, estaban encabezados por “Operaciones y conceptos básicos de las TIC” y la Creatividad no figuraba. Sin embargo, en la nueva versión de estos Estándares, liberada en 2008, la creatividad encabeza los seis grupos de estándares. Otro ejemplo muy dicente es la creación en Inglaterra del Consorcio para la Creatividad que busca promover en la educación el desarrollo de habilidades de pensamiento que conduzcan la formación de personas orientadas a la creatividad y a la innovación. Una de las razones para que la creatividad se hubiese convertido en tema prioritario es que tiene un alto impacto en la generación de riqueza por parte de las empresas de la Sociedad de la Creatividad. En estas empresas, los reconocimientos profesionales se dan gracias al talento, la creatividad y la inteligencia. La creatividad reemplazó las materias primas como fuente fundamental de crecimiento económico. Para tener éxito en esta nueva Sociedad, las regiones deben desarrollar, atraer y retener a personas talentosas y creativas que generen innovaciones (Banaji & Burn, 2006). Cada vez es mayor el número de empresas que fundamentan su modelo de negocio en la creatividad y la innovación; para ellas, son indispensables personas que además de tener los conocimientos requeridos para desempeñarse en los diferentes cargos, tengan habilidad para pensar y

actuar creativamente. Ejemplo tangible de esto es el que la Comisión Europea, consciente de la importancia que tienen la creatividad y a innovación para el desarrollo social y económico de los países del viejo continente, decidiera proclamar el 2009 como el “Año de la Creatividad y la Innovación” El reto enorme que recae hoy sobre los sistemas educativos consiste en lograr que se generen las estrategias adecuadas para que los estudiantes se desarrollen como pensadores creativos. Así como para la sociedad griega en tiempos de Alejandro el Grande era prioridad que las personas desarrollaran su cuerpo como preparación para los quehaceres del campo de batalla, para la sociedad actual es prioritario que las personas desarrollen sus habilidades de pensamiento de orden superior para que pueden desempeñarse con éxito en ella. Pero, dado que el desarrollo de estas habilidades se debe iniciar desde edad temprana, la educación debe asumir su cuota de responsabilidad en esta importante tarea. Desde este punto de vista, la presente “Guía de Algoritmos y Programación”, dirigida a docentes de Educación Básica, se concentra en el desarrollo de la creatividad y de habilidades para solucionar problemas predefinidos. Para facilitar a los docentes su utilización en el aula, los ejemplos que se proponen corresponden a temas de Matemáticas y Ciencias Naturales para grados cuarto y quinto de Básica Primaria. (López García, ALGORITMOS Y PROGRAMACIÓN, 2009).

2.2.1.2 Solución de Problemas

Una de las acepciones que trae el Diccionario de Real Academia de la Lengua Española (RAE) respecto a la palabra Problema es “Planteamiento de una situación cuya respuesta desconocida debe obtenerse a través de métodos científicos”. Con miras a lograr esa respuesta, un problema se puede definir como una situación en la cual se trata de alcanzar una meta y para lograrlo se deben hallar y utilizar unos medios y unas estrategias. La mayoría de problemas tienen algunos elementos en común: un estado inicial; una meta, lo que se pretende lograr; un conjunto de recursos, lo que está permitido hacer y/o utilizar; y un dominio, el estado actual de conocimientos, habilidades y energía de quien va a resolverlo (Moursund, 1999). Casi todos los problemas requieren, que quien los resuelve, los divida en submetas que, cuando son dominadas (por lo regular en orden), llevan a alcanzar el objetivo. La solución de problemas también requiere que se realicen operaciones durante el estado inicial y las submetas, actividades (conductuales, cognoscitivas) que alteran la naturaleza de tales estados (Schunk, 1997). Cada disciplina dispone de estrategias específicas para resolver problemas de su ámbito; por ejemplo, resolver problemas matemáticos implica utilizar estrategias propias de las matemáticas. Sin embargo, algunos psicólogos opinan que es posible utilizar con éxito estrategias generales, útiles para resolver problemas en muchas áreas. A través del tiempo, la humanidad ha utilizado diversas estrategias generales para resolver problemas. Schunk (et al.), destacan los siguientes métodos o estrategias de tipo general:

- **Ensayo y error:** Consiste en actuar hasta que algo funcione. Puede tomar mucho tiempo y no es seguro que se llegue a una solución. Es una estrategia apropiada cuando las soluciones posibles son pocas y se pueden probar todas, empezando por la que ofrece mayor probabilidad de resolver el problema. Por ejemplo, una bombilla que no prende: revisar la bombilla, verificar la corriente eléctrica, verificar el interruptor.

- **Iluminación:** Implica la súbita conciencia de una solución que sea viable. Es muy utilizado el modelo de cuatro pasos formulado por Wallas (1921): preparación, incubación, iluminación y verificación. Estos cuatro momentos también se conocen como proceso creativo. Algunas investigaciones han determinado que cuando en el periodo de incubación se incluye una interrupción en el trabajo sobre un problema se logran mejores resultados desde el punto de vista de la creatividad. La incubación ayuda a "olvidar" falsas pistas, mientras que no hacer interrupciones o descansos puede hacer que la persona que trata de encontrar una solución creativa se estanque en estrategias inapropiadas.

- **Heurística:** Se basa en la utilización de reglas empíricas para llegar a una solución. El método heurístico conocido como "IDEAL", formulado por Bransford y Stein (1984), incluye cinco pasos: Identificar el problema; definir y presentar el problema; explorar las estrategias viables; avanzar en las estrategias; y lograr la solución y volver para evaluar los efectos de las actividades (Bransford & Stein, 1984). El matemático Polya (1957) también

formuló un método heurístico para resolver problemas que se aproxima mucho al ciclo utilizado para programar computadores. A lo largo de esta Guía se utilizará este método propuesto por Polya.

- **Algoritmos:** Consiste en aplicar adecuadamente una serie de pasos detallados que aseguran una solución correcta. Por lo general, cada algoritmo es específico de un dominio del conocimiento. La programación de computadores se apoya en este método.

- **Modelo de procesamiento de información:** El modelo propuesto por Newell y Simon (1972) se basa en plantear varios momentos para un problema (estado inicial, estado final y vías de solución). Las posibles soluciones avanzan por subtemas y requieren que se realicen operaciones en cada uno de ellos.

- **Análisis de medios y fines:** Se funda en la comparación del estado inicial con la meta que se pretende alcanzar para identificar las diferencias. Luego se establecen submetas y se aplican las operaciones necesarias para alcanzar cada submeta hasta que se alcance la meta global. Con este método se puede proceder en retrospectiva (desde la meta hacia el estado inicial) o en prospectiva (desde el estado inicial hacia la meta).

- **Razonamiento analógico:** Se apoya en el establecimiento de una analogía entre una situación que resulte familiar y la situación problema. Requiere conocimientos suficientes de ambas situaciones.

- **Lluvia de ideas:** Consiste en formular soluciones viables a un problema. El modelo propuesto por Mayer (1992) plantea: definir el problema; generar muchas soluciones (sin evaluarlas); decidir los criterios para estimar las soluciones generadas; y emplear esos criterios para seleccionar la mejor solución. Requiere que los estudiantes no emitan juicios con respecto a las posibles soluciones hasta que terminen de formularlas.
- **Sistemas de producción:** Se basa en la aplicación de una red de secuencias de condición y acción (Anderson, 1990).
- **Pensamiento lateral:** Se apoya en el pensamiento creativo, formulado por Edwar de Bono (1970), el cual difiere completamente del pensamiento lineal (lógico). El pensamiento lateral requiere que se exploren y consideren la mayor cantidad posible de alternativas para solucionar un problema. Su importancia para la educación radica en permitir que el estudiante: explore (escuche y acepte puntos de vista diferentes, busque alternativas); avive (promueva el uso de la fantasía y del humor); libere (use la discontinuidad y escape de ideas preestablecidas); y contrarreste la rigidez (vea las cosas desde diferentes ángulos y evite dogmatismos). Este es un método adecuado cuando el problema que se desea resolver no requiere información adicional, sino un reordenamiento de la información disponible; cuando hay ausencia del problema y es necesario apercibirse de que hay un problema; o cuando se debe reconocer la posibilidad de perfeccionamiento y redefinir esa posibilidad como un problema (De Bono, 1970).

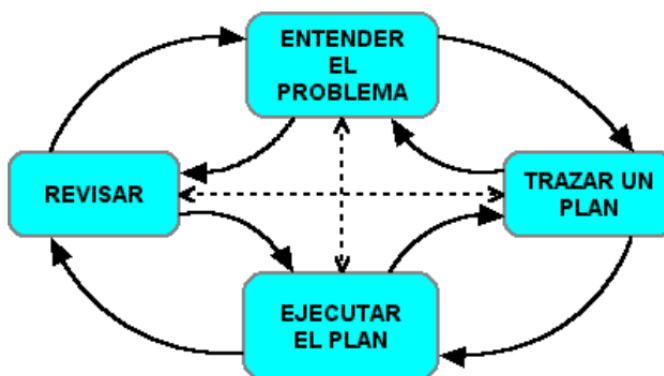
2.2.1.3 Operaciones Mentales para la Resolución de Problemas

Según Polya (1957), cuando se resuelven problemas, intervienen cuatro operaciones mentales:

- Entender el problema.
- Trazar un plan.
- Ejecutar el plan (resolver).
- Revisar.

Numerosos autores de textos escolares de matemáticas hacen referencia a estas cuatro etapas planteadas por Polya. Sin embargo, es importante notar que estas son flexibles y no una simple lista de pasos como a menudo se plantea en muchos de esos textos (Wilson, Fernández & Hadaway, 1993). Cuando estas etapas se siguen como un modelo lineal, resulta contraproducente para cualquier actividad encaminada a resolver problemas.

Ilustración 2: ETAPAS PARA RESOLVER PROBLEMAS



FUENTE: Elaborado Por Polya-Interpretación dinámica y cíclica de las etapas para resolver problemas.

Es necesario hacer énfasis en la naturaleza dinámica y cíclica de la solución de problemas. En el intento de trazar un plan, los estudiantes pueden concluir que necesitan entender mejor el problema y deben regresar a la etapa anterior; o cuando han trazado un plan y tratan de ejecutarlo, no encuentran cómo hacerlo; entonces, la actividad siguiente puede ser intentar con un nuevo plan o regresar y desarrollar una nueva comprensión del problema. (Fernández, et al.).

Ejemplo, en “Recreo Matemático 5” (Díaz, 1993) y en “Dominios 5” (Melo, 2001) se pueden identificar las siguientes sugerencias propuestas a los estudiantes para llegar a la solución de un problema matemático:

1. Comprender el problema.

- Leer el problema varias veces
- Establecer los datos del problema
- Aclarar lo que se va a resolver (¿Cuál es la pregunta?)
- Precisar el resultado que se desea lograr
- Determinar la incógnita del problema
- Organizar la información
- Agrupar los datos en categorías
- Trazar una figura o diagrama.

2. Hacer el plan.

- Escoger y decidir las operaciones a efectuar.

- Eliminar los datos inútiles.
- Descomponer el problema en otros más pequeños.

3. Ejecutar el plan (resolver).

- Ejecutar en detalle cada operación.
- Simplificar antes de calcular.
- Realizar un dibujo o diagrama

4. Analizar la solución (revisar).

- Dar una respuesta completa
- Hallar el mismo resultado de otra manera.
- Verificar por apreciación que la respuesta es adecuada

2.2.1.4 Solución de Problemas y Programación

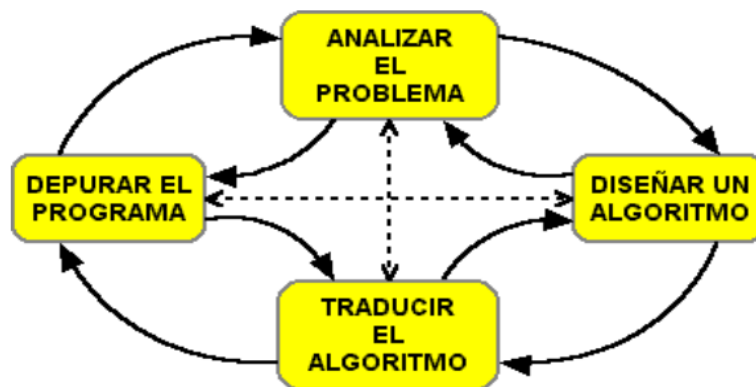
De hecho, para muchos educadores, el uso apropiado de la tecnología en la educación tiene un significado similar a la solución de problemas matemáticos. La programación de computadores para llevar a cabo tareas matemáticas retadoras puede mejorar la comprensión del estudiante “programador” sobre las matemáticas relacionadas con una solución. Esto implica abrirle un espacio a la programación en el estudio de las matemáticas, pero enfocándose en los problemas matemáticos y en el uso del computador como una herramienta para solucionar problemas de esta área (Wilson, Fernández & Hadaway, 1993). Numerosos autores de libros sobre programación, plantean cuatro fases para elaborar un procedimiento que realice una tarea específica. Estas fases

conducen con las operaciones mentales descritas por Polya para resolver problemas:

1. Analizar el problema (Entender el problema).
2. Diseñar un algoritmo (Trazar un plan).
3. Traducir el algoritmo a un lenguaje de programación (Ejecutar el plan).
4. Depurar el programa (Revisar).

Como se puede apreciar, hay una similitud entre las metodologías propuestas para solucionar problemas matemáticos (Clements & Meredith, 1992; Díaz, 1993; Melo, 2001; NAP, 2004) y las cuatro fases para solucionar problemas específicos de áreas diversas, mediante la programación de computadores.

Ilustración 3: FASES PARA ELABORAR UN PROGRAMA DE COMPUTADOR



FUENTE: Elaborado por Clements & Meredith, Díaz, Melo, NAP.

Analiza el Problema:

Los programas de computador tienen como finalidad resolver problemas específicos y el primer paso consiste en definir con precisión el problema hasta lograr la mejor comprensión posible. Una forma de realizar esta actividad se basa en formular claramente el problema, especificar los resultados que se desean obtener, identificar la información disponible (datos), determinar las restricciones y definir los procesos necesarios para convertir los datos disponibles (materia prima) en la información requerida (resultados). Estas etapas coinciden parcialmente con los elementos generales que, según Schunk (1997), están presentes en todos los problemas:

1. Especificar claramente los resultados que se desean obtener (meta y submetas)
2. Identificar la información disponible (estado inicial)
3. Definir los procesos que llevan desde los datos disponibles hasta el resultado deseado (operaciones).

Ilustración 4: ETAPAS A DESARROLLAR EN LA FASE DE ANÁLISIS DE UN PROBLEMA



FUENTE: Elaborado por: Schunk (1997).

Para establecer un modelo que los estudiantes puedan utilizar en la fase de análisis del problema, debemos agregar dos temas a los elementos expuestos por Schunk (1997): formular el problema y determinar las restricciones. Ahora veamos con mayor detalle cada una de las etapas del análisis de un problema.

Formula el Problema: La solución de un problema debe iniciar por determinar y comprender exactamente en qué consiste ese problema. La mayoría de los problemas que se resuelven en el aula de clase llegan a manos de los estudiantes perfectamente formulados. Esta etapa es una buena oportunidad para plantear situaciones en forma verbal o escrita que vinculen la enseñanza de las matemáticas con el entorno en el que vive el estudiante y que tengan una variedad de estructuras y de formas de solución (Zemelman (et al). Esta metodología obliga al estudiante a formular el problema a partir de la situación real planteada. De esta manera se contrarresta la costumbre tan común en el aula de que los

problemas sean formulados por el profesor o tomados de los libros de texto (Brown & Walter, 1990).

La comprensión lingüística del problema (entender el significado de cada enunciado) es muy importante. El estudiante debe realizar una lectura previa del problema con el fin de obtener una visión general de lo que se le pide y una segunda lectura para poder responder preguntas como:

- ¿Puedo definir mejor el problema?
- ¿Qué palabras del problema me son desconocidas?
- ¿Cuáles son las palabras clave del problema?
- ¿He resuelto antes algún problema similar?
- ¿Qué información es importante?
- ¿Qué información puedo omitir?

Además, es conveniente que los estudiantes se habitúen a analizar los problemas desde diferentes puntos de vista y a categorizar la información dispersa que reciben como materia prima (Schunk, 1997).

En programación es frecuente que quien programa deba formular el problema a partir de los resultados esperados. Es muy importante que el estudiante sea consciente de que cuando las especificaciones de un programa se comunican mediante lenguaje natural, estas pueden ser

ambiguas, incompletas e incongruentes. En esta etapa se debe hacer una representación precisa del problema (Rumbaugh, 1996); especificar lo más exactamente posible lo que hay que hacer (no cómo hay que hacerlo).

Precisar los resultados esperados (meta y submetas): Para establecer los resultados que se esperan (meta) es necesario identificar la información relevante, ignorar los detalles sin importancia, entender los elementos del problema y activar el esquema correcto que permita comprenderlo en su totalidad (Woolfolk, 1999). Determinar con claridad cuál es el resultado final (producto) que debe devolver el programa es algo que ayuda a establecer la meta. Es necesario analizar qué resultados se solicitan y qué formato deben tener esos resultados (impresos, en pantalla, diagramación, orden, etc). El estudiante debe preguntarse:

- ¿Qué información me solicitan?
- ¿Qué formato debe tener esta información?

Identifica datos disponibles (estado inicial): Otro aspecto muy importante en la etapa de análisis del problema consiste en determinar cuál es la información disponible. El estudiante debe preguntarse:

- ¿Qué información es importante?
- ¿Qué información no es relevante?

- ¿Cuáles son los datos de entrada? (conocidos)
- ¿Cuál es la incógnita?
- ¿Qué información me falta para resolver el problema? (datos desconocidos).
- ¿Puedo agrupar los datos en categorías?

Otro aspecto importante del estado inicial hace referencia al nivel de conocimiento que el estudiante posee en el ámbito del problema que está tratando de resolver. Es conveniente que el estudiante se pregunte a sí mismo:

- ¿Qué conocimientos tengo en el área o áreas del problema?
- ¿Son suficientes esos conocimientos?
- ¿Dónde puedo obtener el conocimiento que necesito para resolver el problema?
- ¿Mis compañeros de estudio me pueden ayudar a clarificar mis dudas?
- ¿Qué expertos en el tema puedo consultar?

En el ámbito de las matemáticas, se conoce como conocimiento condicional a aquel que activan los estudiantes cuando aplican procedimientos matemáticos concretos de manera intencional y consciente a ciertas situaciones. “El conocimiento condicional

proporciona al alumno un sistema de valoración sobre la extensión y las limitaciones de su saber (qué sabe sobre el tema, su capacidad de memoria, etc), a la vez que examina la naturaleza de la demanda del profesor y su objetivo último, y evalúa variables externas como pueden ser el tiempo que tiene o con quién realiza la tarea” (Orubia & Rochera & Barberà, 2001).

Determinar las restricciones: Resulta fundamental que los estudiantes determinen aquello que está permitido o prohibido hacer y/o utilizar para llegar a una solución. En este punto se deben exponer las necesidades y restricciones (no una propuesta de solución). El estudiante debe preguntarse:

- ¿Qué condiciones me plantea el problema?
- ¿Qué está prohibido hacer y/o utilizar?
- ¿Qué está permitido hacer y/o utilizar?
- ¿Cuáles datos puedo considerar fijos (constantes) para simplificar el problema?
- ¿Cuáles datos son variables?
- ¿Cuáles datos debo calcular?

Establecer procesos (operaciones): Consiste en determinar los procesos que permiten llegar a los resultados esperados a partir de los datos disponibles. El estudiante debe preguntarse:

- ¿Qué procesos necesito?
- ¿Qué fórmulas debo emplear?
- ¿Cómo afectan las condiciones a los procesos?
- ¿Qué debo hacer?
- ¿Cuál es el orden de lo que debo hacer?

En la medida de lo posible, es aconsejable dividir el problema original en otros más pequeños y fáciles de solucionar (submetas), hasta que los pasos para alcanzarlas se puedan determinar con bastante precisión (módulos). Esto es lo que en programación se denomina diseño descendente o top-down (Joyanes, 2001). El diseño descendente se utiliza en la programación estructurada de computadores debido a que facilita:

- La comprensión del problema
- Las modificaciones en los módulos
- La verificación de la solución

Al realizar divisiones sucesivas del problema en otros más pequeños y manejables (módulos), hay que tener cuidado para no perder de vista la comprensión de este como un todo. El estudiante, luego de dividir el problema original en submetas (módulos), debe integrar cada parte de tal forma que le permita comprender el problema como un todo (Woolfolk, 1999). Igualmente hay que tener cuidado cuando se utiliza este enfoque para resolver problemas complejos o extensos, en cuyo caso resulta más aconsejable utilizar una metodología orientada a objetos. Especialmente, cuando profesores universitarios manifiestan su preocupación por el aprendizaje de malas prácticas de programación en el colegio. Hay casos en los cuales algunos estudiantes no han podido cambiar su forma de pensar “estructurada” por otra orientada a objetos, la cual hace parte de los programas universitarios modernos en la carrera de Ingeniería de Sistemas. Es aconsejable que los ejemplos y actividades planteados a los estudiantes contengan solo un problema cuya solución sea muy corta (no necesariamente sencillo de resolver). De esta forma ellos podrán enfocarse en aplicar completamente la metodología propuesta para analizar problemas (formular el problema, especificar los resultados, identificar la información disponible, determinar las restricciones y definir los procesos) sin perderse en el laberinto de un problema demasiado complejo.

Diseñar el Algoritmo (trazar un Plan).

Este consiste en la representación gráfica, mediante símbolos geométricos, de la secuencia lógica de las instrucciones (plan) que posteriormente serán traducidas a un lenguaje de programación, como Logo, para ejecutarlas y probarlas en un computador.

Ejemplo

Diseñar un algoritmo (seudocódigo y diagrama de flujo) para hallar el área de un triángulo rectángulo cuya Base mide 3 cm, la Altura 4 cm y la Hipotenusa 5 cm.

Análisis Del Problema

Formular el problema: Ya se encuentra claramente planteado.

Resultados esperados: El área de un triángulo rectángulo.

Datos disponibles: Base, Altura, Hipotenusa, tipo de triángulo. La incógnita es el área y todos los valores son constantes. El valor de la hipotenusa se puede omitir. El estudiante debe preguntarse si sus conocimientos actuales de matemáticas le permiten resolver este problema; de no ser así, debe plantear una estrategia para obtener los conocimientos requeridos.

Determinar las restricciones: Utilizar las medidas dadas.

Procesos necesarios: Guardar en dos variables (BASE y ALTURA) los valores de Base y Altura; Guardar en una constante (DIV) el divisor 2; aplicar

la fórmula $BASE * ALTURA / DIV$ y guardar el resultado en la variable AREA; comunicar el resultado (AREA).

Algoritmo En Seudocódigo

Paso 1: Inicio

Paso 2: Asignar el número 2 a la constante "div"

Paso 3: Asignar el número 3 a la constante "base"

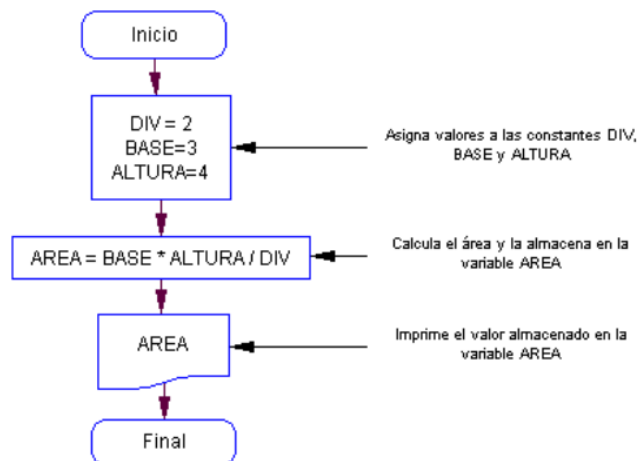
Paso 4: Asignar el número 4 a la constante "altura"

Paso 5: Guardar en la variable "área" el resultado de $base * altura / div$

Paso 6: Imprimir el valor de la variable "área"

Paso 7: Final

Ilustración 5: DIAGRAMA DE FLUJO PARA HALLAR EL ÁREA DE UN TRIÁNGULO RECTÁNGULO



FUENTE: Elaborado por (López García, ALGORITMOS Y PROGRAMACIÓN, 2009).

Traducir el Algoritmo (Ejecutar el Plan).

Una vez que el algoritmo está diseñado y representado gráficamente se pasa a la etapa de traducción a un lenguaje de programación determinado (en nuestro caso será Logo). Cada lenguaje posee sus propias reglas gramaticales, por lo tanto es fundamental que los estudiantes conozcan de antemano la sintaxis de los comandos que deben utilizar para resolver el problema. A mayor dominio del lenguaje de programación, mayor posibilidad de llegar rápidamente a una solución satisfactoria. A esta fase de traducción se le conoce comúnmente como codificación. Esto trata de llevarlo al lenguaje de programación Scratch.

Depurar el Programa (revisar).

Después de traducir el algoritmo en un lenguaje de programación como Logo, el programa resultante debe ser probado y validados los resultados. A este proceso se le conoce como depuración. Depurar programas contribuye a mejorar la capacidad en los estudiantes para resolver problemas; la depuración basada en la retroalimentación es una habilidad útil para toda la vida (Stager, 2003). Quienes han escrito alguna vez un programa de computador, saben de la dificultad que representa elaborar programas perfectos en el primer intento, dificultad que aumenta a medida que el problema a resolver es más complejo. La depuración, afinamiento y documentación de un programa hacen parte fundamental del ciclo de programación y desde el punto de vista educativo estimula en los estudiantes la curiosidad, la perspectiva, la comunicación y

promueve valores como responsabilidad, fortaleza, laboriosidad, paciencia y perseverancia. La programación facilita un diálogo interior en el cual la retroalimentación constante y el éxito gradual empujan a los alumnos a ir más allá de sus expectativas (Stager, 2003). Otras dos actividades relacionadas con esta etapa son la afinación y la documentación. La primera consiste en realizar retoques para lograr una mejor apariencia del programa (en pantalla o en los resultados impresos) o para ofrecer funcionalidades más allá de los resultados esperados (especificados en la fase de análisis del problema). La segunda tiene un carácter eminentemente comunicativo, con la documentación de un programa se pone a prueba la capacidad del estudiante para informar a otras personas cómo funciona su programa y lo que significa cada elemento utilizado.

2.2.1.5 Pensamiento Algorítmico

Cuando se habla de algoritmos, con frecuencia aparecen tres tipos de pensamiento que generalmente se relacionan con ellos y que se utilizan indiscriminadamente como sinónimos: Pensamiento Computacional, Pensamiento Algorítmico y Pensamiento Procedimental. Por lo tanto es importante puntualizar a qué se refiere cada uno de estos pensamientos. Según Moursund (2006), el pensamiento computacional hace referencia a la representación y solución de problemas utilizando inteligencia humana, de máquinas o de otras formas que ayuden a resolver el problema. El pensamiento algorítmico se refiere al desarrollo y uso de algoritmos que

puedan ayudar a resolver un tipo específico de problema o a realizar un tipo específico de tarea. Por su parte, el pensamiento procedimental se ocupa del desarrollo y utilización de procedimientos diseñados para resolver un tipo específico de problema o para realizar un tipo específico de tarea, pero que no necesariamente, siempre resulta exitoso. Por otra parte y de acuerdo con un reporte del Consejo Nacional de Investigación de Estados Unidos (National Research Council, NRC, 2004), conocido como “Being Fluent with Information Technology”, el Pensamiento Algorítmico incluye elementos tales como: descomposición funcional, repetición (iteración y/o recursión), organización de datos (registro, campo, arreglo, lista, etc), generalización y parametrización, diseño por descomposición de un problema en partes más pequeñas y manejables (top-down) y refinamiento. El Pensamiento Algorítmico está fuertemente ligado al pensamiento procedimental requerido en la programación de computadores; sin embargo, su desarrollo puede conducir a los estudiantes a aproximarse guiada y disciplinadamente a los problemas de forma que este pueda transferirse a otros ambientes diferentes a los de la programación. En pocas palabras, la programación de computadores aporta al ámbito escolar un laboratorio para desarrollar habilidades indispensables en la vida real del Siglo XXI. Una diferencia notoria entre un algoritmo y un programa es que el algoritmo incorpora las características estructurales básicas de la computación, independientemente de los detalles de su implementación; mientras que un

programa tiene un conjunto específico de detalles para resolver un problema. Se puede observar que una técnica de solución (correspondiente al algoritmo) se puede utilizar en diferentes situaciones problemáticas (correspondiente a los programas). De manera inversa, se espera que una solución exitosa de problemas incorpore procesos generales que son independientes de las situaciones específicas (NRC, 2004). Esto se conoce como experiencias de vida y los estudiantes deben adquirirlas en su paso por la educación básica y media para desempeñarse adecuadamente en su vida diaria. Este es todo un reto para la educación, reto en el que la programación de computadores puede hacer una contribución positiva. Un programa consiste de uno o más procedimientos con instrucciones paso a paso que pueden ejecutarse en un computador; por lo tanto, utilizar el diseño de procedimientos que solucionen o ayuden a solucionar problemas con diferentes niveles de complejidad es un recurso que puede aprovechar el docente para captar el interés de los estudiantes en actividades de programación. Por ejemplo, asignar la tarea de diseñar un procesador de texto básico (ingreso del texto mediante teclado, mostrarlo en la pantalla y guardarlo en el disco duro) es una tarea relativamente sencilla. Pero el proyecto puede aumentar su complejidad si se añaden funciones para dar formato al texto (fuentes, tamaño y características especiales). Posteriormente el proyecto puede crecer si se agregan funcionalidades para manejar imágenes y tablas. Al igual que en este ejemplo, se pueden diseñar

proyectos de clase interesantes para mantener motivados a los estudiantes y cuyas tareas y retos sean progresivos en complejidad; que cada nuevo reto parta de lo construido con anterioridad. En resumen, los procedimientos son un tipo particular de tarea que busca solucionar problemas específicos y al desarrollarlos se ponen en juego los pensamientos algorítmico y procedimental. David Moursund (2006) se basó en sus propias experimentaciones y en la teoría de los cuatro estados de desarrollo cognitivo planteada por Piaget para proponer un planteamiento que amarra la computación con una escala de desarrollo cognitivo en la que se da bastante protagonismo al desarrollo del pensamiento algorítmico en los niños. Según Moursund (2006) en la etapa de las operaciones concretas los niños empiezan a manipular lógicamente y sistemáticamente símbolos en un computador y aprenden a apoyarse en software para resolver un rango amplio de problemas y tareas de tipo general. De esta manera, ganan habilidad considerable tanto en la utilización de lenguajes como Scratch y MicroMundos, como en la manipulación de ambientes gráficos. Posteriormente, en la etapa de operaciones formales, los estudiantes demuestran su inteligencia por medio del uso lógico de símbolos relacionados con conceptos abstractos.

2.2.1.6 Importancia y Trascendencia del pensamiento algorítmico.

(UNLP) Nos dice:

Conceptos de Algoritmos y su Historia

Es notable como una palabra que nace de los trabajos de un matemático y astrónomo de la Edad Media (muy lejos de las computadoras en el tiempo!) se convirtió con los años en uno de los pilares de la Ciencia Informática. En efecto el nombre se relaciona con el matemático árabe Al-Khwarizmi, quien desarrolló gran parte de su carrera en Bagdad, alrededor del año 800 DC. Allí creó un Centro Superior de Investigaciones Científicas y se dedicó especialmente al Algebra y la Astronomía. Sus procedimientos para resolución de ecuaciones y el tratado traducido al latín sobre números “Algoritmi de número Indorum” lo han dejado como el referente más antiguo de la palabra “Algoritmo”. Nos quedamos entonces con la definición “genérica” de Algoritmo: “Conjunto ordenado de operaciones que tienen como objetivo resolver un problema”. Esta definición excede la Informática e incluso las Ciencias duras y nos transmite un par de atributos conceptuales importantes: el conjunto de operaciones puede no ser único y las soluciones pueden tener mayor o menor calidad, en función de diferentes métricas.

Cuando acotamos el campo de aplicación de la definición de Algoritmo a la Informática o la Matemática, diremos que tendremos una lista ordenada

y finita de pasos que dado un estado inicial nos permite transformarlo en un estado final “solución” en un tiempo también finito. Naturalmente la importancia del concepto en la formación de estudiantes de todo nivel está en la relación de transformación “problema-solución” que significa la elaboración correcta de un Algoritmo.

La capacidad de “pensar algoritmos” y la formación de los Informáticos

El Informático es normalmente un profesional dedicado a analizar y abstraer problemas del mundo real (en los campos más diversos) y sintetizar su posible solución (correcta y eficiente) empleando alguna forma de “computadora”. Es decir, la esencia de su tarea profesional se relaciona directamente con aquella definición genérica de Algoritmo. Por esto a través de los años el inicio de las carreras universitarias de Informática tiene uno o más cursos de Algorítmica. Incluso esta capacidad de abstracción y pensamiento de problemas es deseable introducirla a edades muy tempranas, para ayudar en la formación integral de niños y adolescentes. En los países más desarrollados el tema de elaborar “Algoritmos” en un sentido amplio ha dejado de ser un tema exclusivamente “Informático” para convertirse en un tema formativo de importancia para el desarrollo de las capacidades intelectuales del individuo.

Pensamiento Algorítmico, Pensamiento Computacional y Pensamiento Lógico

Un algoritmo es entonces una secuencia finita, ordenada y lógica de pasos para llegar al objetivo de resolver un problema. El “pensamiento algorítmico” es la capacidad/ aptitud que tenemos para realizar el proceso de abstracción, modelización del problema, deducciones lógicas y síntesis de la solución que conduzca a escribir el algoritmo correcto. Un paso más allá es lo que se denomina “pensamiento computacional” que definió Jeannette Wing en 2006 como una nueva competencia que debiera incluirse en la formación educativa y que implica “resolver problemas, diseñar sistemas y comprender el comportamiento humano haciendo uso de los conceptos fundamentales de la Informática”. Sintéticamente llevar la metodología de un científico de Informática a la formación de niños y adolescentes como una competencia central. Por último el “pensamiento lógico” es aquel que se desprende de las relaciones entre los objetos y procede de la propia elaboración del individuo. Surge a través de la coordinación de las relaciones que previamente ha creado el mismo individuo entre los objetos. El pensamiento lógico sirve para analizar, razonar, validar razonamientos y confrontarlos con la realidad. Los educadores debemos propiciar actividades, experiencias y problemas/proyectos que permitan potenciar el desarrollo del pensamiento lógico, como una faceta central de formación de competencias en los

alumnos. Podemos ver estas tres definiciones como “escalones” en el desarrollo de aptitudes en el proceso formativo... notemos que en todas las etapas está presente aquel concepto genérico de “Algoritmo” que definimos al inicio.

A qué edad debemos formar la capacidad de “pensar algoritmos”?

Existen numerosas experiencias (entre ellas las más importantes en los últimos años en el MIT – USA) respecto de introducir la capacidad de escribir algoritmos y los fundamentos del pensamiento computacional a edades muy tempranas (pre-escolares). En particular herramientas como Scratch y Scratch Jr. y la relación con actividades experimentales con robots (la evolución de la antigua tortuga de LOGO en el MIT) en niños de 5-8 años están teniendo resultados muy significativos. Por qué? Porque la creatividad y capacidad de imaginar soluciones son muy destacadas en los niños e incentivarlas mediante reglas “algorítmicas” y operadores físicos que responden a las “órdenes” o “pasos” que definen los mismos niños (como pueden ser los robots) genera una sinergia altamente positiva. Posiblemente la educación orientada a desarrollar competencias sea el futuro... y en ese camino la formación en “pensar algoritmos” como un paso hacia el pensamiento computacional y el pensamiento lógico sea un componente muy importante.

2.2.1.7 Herramientas para promover el pensamiento algorítmico

(Gutierrez, 2016). Señala que el pensamiento algorítmico en las matemáticas son las siguientes:

Herramientas para promover el pensamiento algorítmico

1. **Diagramas de flujo con *PseInt***. Visualizan las operaciones, condiciones y repeticiones de forma clara. Es muy útil para la comprensión lógica y codificación en un lenguaje como C. Su uso ha trascendido a otros ámbitos profesionales donde se requiere que las personas pueden comprender los flujos de datos e información de forma clara siguiendo los flujos.

2.- ***Scratch***. Es ideal para fomentar el desarrollo de animaciones de forma gráfica con objetos lúdicos, lo cual lo hace atractivo a los niños y a los jóvenes planteándoles objetivos atractivos. Se lo encuentra en: <http://scratch.mit.edu/>

3.- ***Blockly***. Es una herramienta visual de programación básica como una alternativa a *scratch*, se programa en la web por google: <https://developers.google.com/blockly/>

4.- **El lenguaje C.** Una opción ideal para implementar programas desde los más simples hasta los más complejos. Existe una variedad de compiladores para el lenguaje C.

2.3 MARCO SITUACIONAL

La presente tesis se llevó a cabo en la institución educativa particular Augusto Cardich que tiene como datos:

Tabla 4: DATOS DEL COLEGIO AUGUSTO CARDICH

I.E.P AUGUSTO CARDICH	
Dirección	Av. Universitaria km 2.5 – urb. San Diego- Cayhuayna-Pillco Marca.
Ubigeo	100111
Género	Mixto
Nivel	Inicial, Primaria y Secundaria

FUENTE: Elaborado por Tesista.

MISIÓN DE LA IEP “AUGUSTO CARDICH”

Somos Institución Educativa que brinda servicios educativos de calidad en todos sus niveles y modalidades de la EBR, orientado al desarrollo de la formación integral, científica, humanística y holística de los niños y jóvenes, con una mística de trabajo corporativo y en equipo, predispuestos a participar en eventos académicos de competencia; la constante práctica de valores, para desarrollar capacidades sociales y ejercer su ciudadanía con autonomía y seguir aprendiendo

durante toda la vida.

VISIÓN DE LA IEP. "AUGUSTO CARDICH"

Al 2021 la Institución Educativa Privada "Augusto Cardich" seremos una institución líder en la comunidad y la región, que brinda una educación de alta calidad, dirigido a niños, púberes, adolescentes, jóvenes y adultos, basado en un enfoque curricular innovador, centrado en las personas, la sociedad y la investigación; en los modelos pedagógicos y de gestión abiertos al cambio, con propuestas flexibles, dinámicas, democráticas y eficientes, que son los fundamentos del desarrollo integral de la persona, con la mejora continua del proceso enseñanza-aprendizaje para la vida.

VALORES QUE SE DESARROLLARÁN EN EL AUGUSTO CARDICH

a) Respeto a la persona:

Otorgarle el valor que le corresponde a cada uno de las personas que forman parte de la comunidad educativa cardichana, consagrada en la Constitución Política del Estado.

b) Responsabilidad:

Cada uno de los miembros de la familia cardichana asume el rol que le corresponde, como docente, estudiante, padre de familia o directivo.

c) Honestidad:

Desarrollan en forma transparente, bajo el imperio de la verdad, todos los hechos o actividades que se ejecutan.

d) Investigador:

Cultura de valor impulsado por las acciones de buscar la verdad con creatividad y actitud reflexiva, y desarrollar nuevas creativas acciones pedagógicas.

e) Identidad institucional:

Se fortalecen valores de amor a la institución, a la patria y a los símbolos patrios.

f) Crítico-Reflexivo:

Los estudiantes, docentes, y directivos de la IEP “Augusto Cardich”, tienen una alta predisposición por recibir y procesar las críticas, las opiniones constructivas, con las intenciones de corregir lo que es necesario.

g) Democrático:

La comunidad educativa, practica los valores supremos de respetar las ideas de los demás, por muy controversiales que éstas fueran, reconocer las acciones distinguidas de los demás.

h) Autocrítico:

La comunidad educativa desarrolla la capacidad de reconocer los errores individuales y de equipo, y procesan las correcciones de manera asertiva.

En la I.E.P Augusto Cardich en el 2019 se lleva el área de educación por el trabajo/computación en el nivel primario desde el cuarto grado hasta el sexto grado y cada uno de ellos con dos secciones A y B, también se lleva en el nivel

secundario desde el primer año hasta el quinto año distribuidos en dos secciones A y B sólo hasta tercer año ya que los alumnos del cuarto y quinto año están divididos en tres áreas que son negocios, ingeniería y salud.

En el 2017 se vuelve a retomar computación por la insistencia de los padres ya que años anteriores no hubo el área de computación. En un inicio no se contaba con computadoras actuales y el ambiente no era adecuado; el promotor y el director adecuaron de infraestructura y computadoras una sala netamente para computación con 25 computadoras y módulos como se puede ver en el anexo 12.

Muchos de los estudiantes consideraban al curso de computación como sinónimo de juegos online, offline, entretenimiento y carecían de orientación pedagógica y metodológica para usar los programas de computación de manera adecuada.

En un inicio se determinó que los estudiantes de primaria aprendieran a usar los programas básicos como Paint, Microsoft Word, Power point y Excel, para los estudiantes de secundaria los programas básicos más los programas de Microsoft Visio, Gimp, Xmind, Filmora.

En el 2019 se tenía establecido ya un sílabus para el nivel primario y secundario según los programas que se optaron por enseñar pero se tuvo que incluir dentro del sílabus el programa SCRATCH sugerido por mi persona porque se pretendía iniciar en la programación de manera sencilla y que los alumnos mejoren en sus desempeños académicos; se desarrollaron cinco sesiones de aprendizaje donde la primera y última sesión son para la evaluación de pretest y postest, las restantes

sesiones son para la aplicación de la presente investigación en total fueron de cinco semanas ya que cada semana se tiene dos horas pedagógicas con el grupo experimental y el grupo de control. Se realizó la cuarta sesión fuera del horario de clase porque cada mes (4 semanas) se toma un examen mensual que evidencia si los alumnos están aprendiendo o no según las políticas del colegio Augusto Cardich.

2.4 DEFINICIÓN DE TÉRMINOS BÁSICOS

Pensamiento computacional: En palabras de la propia Wing “el pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática”. Es decir, que la esencia del pensamiento computacional es pensar como lo haría un científico informático cuando nos enfrentamos a un problema.

SRATCH: Es un lenguaje de programación desarrollado por el Massachusetts Institute of Technology Media Lab (MIT), con la finalidad de que las personas puedan programar en forma sencilla, sin necesidad de complejos algoritmos y largas horas de cursos de programación. Es una evolución del famoso lenguaje de programación LOGO, que permite en muy corto tiempo la creación interactiva de animaciones, historias, juegos música, arte, etc. (Resnick, 2007).

El lenguaje está basado bloques gráficos y con una interface fácil para su aprendizaje. Consiste en la unión de bloques que pueden ser eventos, movimientos de gráficos y sonidos.

Pensamiento algorítmico: Es la capacidad de entender, ejecutar, evaluar y crear algoritmos.

Desarrollo del pensamiento algorítmico: Es una de las dificultades que los estudiantes confrontan cuando aprenden a programar, utilizar la estructura de selección y de control correcta es un gran reto.

Algoritmo: Es un conjunto de instrucciones paso a paso precisas para la solución de una tarea

III. MARCO METODOLÓGICO

3.1 NIVEL Y TIPO DE INVESTIGACIÓN

El nivel de investigación es **experimental**.

“Responde a las preguntas ¿qué cambios y modificaciones se han producido? ¿Qué mejoras se han logrado? ¿Cuál es la eficiencia del nuevo sistema? (Carrasco Díaz S. , 2005).

El tipo de investigación es **aplicada**.

“Refiere; que se enseña para actuar, transformar, modificar o producir cambios en un determinado sector de la realidad”. (Carrasco Díaz S. , 2005).

3.2 DISEÑO DE LA INVESTIGACIÓN

En el presente trabajo de investigación se ha utilizado el diseño **cuasi experimental** con dos grupos: grupo experimental y grupo de control, con pre test y pos test.

Refiere que “en los diseños cuasi experimentales los grupos de trabajo ya están tomados; es decir, ya existen previamente al experimento”. (Carrasco Díaz, 2005)

Se da el siguiente esquema:

G 1 O1 X O2

G 2 O1 --- O2

G1 = grupo experimental

G2 = grupo de control

O1= Observación (pretest)

O2= observación (postest)

X= Uso del lenguaje de programación SCRATCH.

3.3 DETERMINACIÓN DE LA POBLACIÓN

Población

La población estará conformada por 440 estudiantes del nivel primario y secundario de la Institución Educativa Particular Augusto Cardich.

Tabla 5: DISTRIBUCIÓN DE LA POBLACIÓN

LUGAR	I.E.P	CANTIDAD ESTUDIANTES
Pillco Marca	Institución Educativa Particular Augusto Cardich	440

FUENTE: Elaborado por Tesista

3.4 SELECCIÓN DE LA MUESTRA

Muestra

La muestra es **No Probabilística intencionada**, (Carrasco Díaz, 2005). “donde el investigador selecciona según su propio criterio, sin ninguna regla matemática o estadística”.

La muestra estará constituida por 42 estudiantes distribuidos de la siguiente manera del sexto grado A de educación primaria 19 estudiantes (grupo de control) y del sexto grado B de educación primaria 23 estudiantes (grupo experimental) de la Institución Educativa Particular Augusto Cardich.

Tabla 6: DISTRIBUCIÓN DE LOS GRUPOS Y LA MUESTRA

ESTUDIANTES	GRUPO EXPERIMENTAL	GRUPO CONTROL
SECCIÓN	Aula: B	Aula: A
N° DE ESTUDIANTES	23	19
TOTAL = 42		

FUENTE: Elaborado por Tesista

La conformación de estudiantes por aula ha sido seleccionada al azar.

Tabla 7: ESQUEMA DE LA TESIS

Grupo	Pretest	Proceso	Postest
Experimental	O1	X	O2
Control	O1	O2

FUENTE: Elaborado por Tesista

3.5 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

3.5.1 TÉCNICA DE RECOLECCIÓN DE DATOS

Las técnicas para la recolección de datos en esta investigación serán:

Observación Participante. Se refiere que “este tipo de observación supone la integración del observador al grupo que observa, es decir, forma parte de él y convive con ellos para conocer desde dentro lo que desea investigar”. (Carrasco Díaz, 2005).

La encuesta. Se refiere que “es una técnica para la investigación social para indagación, exploración y recolección de datos, mediante preguntas formuladas directa o indirecta a los sujetos que constituyen la unidad de análisis”. (Carrasco Díaz, 2005).

3.5.2 MEDIOS E INSTRUMENTO DE RECOLECCIÓN DE DATOS

Tabla 8: PRUEBA DE CONOCIMIENTO DEL PRETEST Y POSTEST

ANALIZA EL PROBLEMA	Comprende el problema. (escala DE 0 A 2)	Prueba de conocimiento. PRETEST Y POSTEST
	Define los datos disponibles (escala DE 0 A 2)	
	Define los procesos del problema. (escala DE 0 A 2)	
DISEÑA Y TRADUCE EL ALGORITMO	Representa gráficamente el problema. (escala DE 0 A 2)	
	Secuencializa el desarrollo del problema. (escala DE 0 A 2)	
	Ordena lógicamente las instrucciones a seguir. (escala DE 0 A 2)	
DEPURA EL PROGRAMA (validación de resultados)	Verifica la respuesta del problema. (escala DE 0 A 2)	
	Propone otras alternativas de solución. (escala DE 0 A 2)	

	Corrige fallas. (escala DE 0 A 2)	
--	--------------------------------------	--

FUENTE: Elaborado por Tesista

Tabla 9: VALIDACIÓN DEL INSTRUMENTO

VARIABLE - DIMENSIÓN	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD
Variable Dependiente:				
PENSAMIENTO ALGORÍTMICO				
Dimensión 1:				
FASE DE ANALIZAR EL PROBLEMA				
VARIABLE - DIMENSIÓN	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD
Variable Dependiente:				
PENSAMIENTO ALGORÍTMICO				
Dimensión 2:				
FASE DE DISEÑAR Y TRADUCIR EL ALGORITMO				
VARIABLE - DIMENSIÓN	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD
Variable Dependiente:				
PENSAMIENTO ALGORÍTMICO				
Dimensión 3:				
FASE DE DEPURAR EL PROGRAMA				

FUENTE: Elaborado por Tesista

Los medios que se utilizarán en la investigación son:

- ✓ **Celular:** Cámara fotográfica, filmadora y grabadora. (Anexo 12).

Los instrumentos que se utilizarán en la investigación son:

- ✓ **Pruebas de conocimiento (pre test y post test):** Las pruebas de conocimiento serán dos: de inicio (pre test) y otro al final (post test)

cada una constará de 10 preguntas y se aplicará la misma prueba de conocimiento tanto al inicio como el final. (Anexo 5). .

- ✓ **Cuestionario:** Este instrumento permitirá analizar y validar nuestro instrumento de investigación. (Anexo 1; 2; 3 y 4).
- ✓ **Registro de incidencias.** Éste instrumento ha permitido observar la conducta de los estudiantes en las actividades programadas como parte del desarrollo de la investigación. (Anexo 6).

3.6 PROCESAMIENTO Y PRESENTACIÓN DE DATOS

Para el procesamiento y análisis de datos

- a) Ordenamiento de cuestionarios para la validación de mi instrumento.
- b) Ordenamiento y clasificación de pruebas de conocimiento.
- c) Ordenamiento y clasificación de registro de incidencias
- d) SPSS.

Para la presentación de resultados

- a) Tablas.

IV. RESULTADOS DE LA INVESTIGACIÓN

4.1 PROCESAMIENTO Y PRESENTACIÓN DE DATOS

4.1.1. ORDENAMIENTO DE CUESTIONARIOS PARA LA

VALIDACIÓN DE MI INSTRUMENTO DE INVESTIGACIÓN

- ✓ El anexo 2 nos muestra que para la validación de nuestro instrumento fue realizado por el coordinador de primaria David Esquivel Márquez de la Institución Educativa Particular Augusto Cardich da como calificación del instrumento cuatro, quiere decir que pertenece a una calificación de nivel Alto, por lo tanto queda validado nuestro instrumento por el coordinador David Esquivel Márquez.
- ✓ El anexo 3 nos muestra que para la validación de nuestro instrumento fue realizado por el doctor en educación Grover Vicente Marchan , director de la Institución Educativa Particular Augusto Cardich y da como calificación del instrumento cuatro, quiere decir que pertenece a una calificación de nivel Alto, por lo tanto queda validado nuestro instrumento por el Director Grover Vicente Marchan .
- ✓ El anexo 4 nos muestra que para la validación de nuestro instrumento fue realizado por el coordinador de Matemáticas Bequer Santos Mallqui de la institución educativa Particular Augusto Cardich y da como calificación del instrumento, cuatro, quiere decir que pertenece a una

calificación de nivel Alto, por lo tanto queda validado nuestro instrumento por el coordinador de Matemáticas Beker Santos Mallqui.

El cuestionario para la validación del instrumento de investigación está dividido en tres fases:

- a) Fase de Analizar el problema: Éstas a su vez están sub divididas en cuatro problemas como lo dice el Anexo 1 y cada una de ellas posee los criterios de relevancia, coherencia, suficiencia y claridad.
- b) Fase de Diseñar y Traducir el algoritmo: Éstas a su vez están sub divididas en tres problemas como lo dice el Anexo 1 y cada una de ellas posee los criterios de relevancia, coherencia, suficiencia y claridad.
- c) Fase de Depurar el programa: Éstas a su vez están sub divididas en tres problemas como lo dice el Anexo 1 y cada una de ellas posee los criterios de relevancia, coherencia, suficiencia y claridad.

La calificación del instrumento corresponde a:

Tabla 10: CALIFICACIÓN DEL INSTRUMENTO DE INVESTIGACIÓN

Nº	CALIFICACIÓN
1	No cumple
2	Nivel Bajo
3	Nivel Moderado
4	Nivel Alto

FUENTE: Elaborado por el Tesista

4.1.2. ORDENAMIENTO Y CLASIFICACIÓN DE LAS PRUEBAS DE CONOCIMIENTOS

Se tomaron diez preguntas para la prueba de conocimiento que sirvieron para demostrar los problemas específicos, objetivos específicos e hipótesis específicos; a su vez estas preguntas responden al desarrollo del pensamiento algorítmico a través de sus fases como se puede ver en el Anexo 5.

4.1.3. PRUEBA DE HIPÓTESIS

4.1.3.1. PASO 1 (Redactar Las Hipótesis)

HIPÓTESIS ESPECIFICOS

Hipótesis Específico 1:

HE1o: La aplicación del lenguaje de programación SCRATCH NO influye positivamente en el *desarrollo de la fase de análisis del problema* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

HE1a: La aplicación del lenguaje de programación SCRATCH si influye positivamente en el *desarrollo de la fase de análisis del problema* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

Hipótesis Específico 2:

HE2o: La aplicación del lenguaje de programación SCRATCH NO influye positivamente en el *desarrollo de la fase de diseño y traducción del algoritmo* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

HE2a: La aplicación del lenguaje de programación SCRATCH influye positivamente en el *desarrollo de la fase de diseño y traducción del algoritmo* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

Hipótesis Específico 3:

HE3o: La aplicación del lenguaje de programación SCRATCH NO influye positivamente en el *desarrollo de la fase de depuración del programa* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

HE3a: La aplicación del lenguaje de programación SCRATCH influye positivamente en el *desarrollo de la fase de depuración del programa* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

HIPÓTESIS GENERAL

Ho: La aplicación del lenguaje de programación SCRATCH NO INFLUYE POSITIVAMENTE en el desarrollo del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

Ha: La aplicación del lenguaje de programación SCRATCH INFLUYE POSITIVAMENTE en el desarrollo del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

4.1.3.2. PASO 2 (Definir Alfa)

$$\text{Alfa} = 0.05 = 5\%$$

4.1.3.3. PASO 3 (Elección De La Prueba)

Tabla 11: CUADRO COMPARATIVO PARA ELECCIÓN DE LA PRUEBA

		PRUEBAS NO PARAMÉTRICAS			PRUEBAS PARAMÉTRICAS
VARIABLE ALEATORIA →		NOMINAL DICOTÓMICA	NOMINAL POLITÓMICA	ORDINAL	NUMÉRICA
VARIABLE FIJA ↓					
Estudio Transversal Muestras Independientes	Un grupo	χ^2 Bondad de Ajuste Binomial	χ^2 Bondad de Ajuste	χ^2 Bondad de Ajuste	T de Student (Una muestra)
	Dos grupos	χ^2 Bondad de Ajuste Corrección de Yates. Test exacto de Fisher	χ^2 de Homogeneidad	U Mann-Withney	T de Student (Muestras Independientes)
	Más de dos grupos	χ^2 Bondad de Ajuste	χ^2 Bondad de Ajuste	H Kruskal-Wallis	ANOVA con un factor (INTERsujetos)

Estudio Longitudinal Muestras Relacionadas	Dos medidas	Mc Nemar	Q de Cochran	Wilcoxon	T de Student (Muestras Relacionadas)
	Más de dos Medias	Q de Cochran	Q de Cochran	Friedman	ANOVA para medidas repetidas (INTRAsujetos)

FUENTE: Elaborado por Atanael Varela Lopez

Para nuestra variable fija o dependiente: A un mismo grupo se le aplica dos medidas en momentos diferentes de tiempo entonces se refiere a un estudio longitudinal.

Primero se toma una medida (examen pre test) para evaluar si tienen pensamiento algorítmico y luego se toma otra medida (examen post test) usando la aplicación del lenguaje de programación Scratch para el desarrollo del pensamiento algorítmico.

Para nuestra variable aleatoria o independiente: se usa una variable numérica.

En ese caso nuestra elección de la prueba sería **T de Student** (muestras relacionadas), pero para una mejor división de las medidas y grupos también analizaremos con la prueba de **ANOVA**.

4.1.3.4. PASO 4 (Calcular El P- Valor)

NORMALIDAD

Antes de calcular el valor de la prueba es necesario corroborar si la variable dependiente se comporta normalmente, es decir verificar el

supuesto de normalidad. Existen dos pruebas para revisar la normalidad:

Kolmogorov-Smirnov para muestras grandes (>30 individuos)

Chapiro wilk para muestras pequeñas (<30 individuos).

Tabla 12: PRUEBA DE NORMALIDAD

Pruebas de normalidad						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Variable Dependiente del pretest	,125	42	,097	,958	42	,126
Variable Dependiente del postest	,124	42	,108	,972	42	,373

a. Corrección de significación de Lilliefors

FUENTE: Elaborado por Tesista utilizando SPSS

Criterio para determinar NORMALIDAD:

P-valor => alfa aceptar Ho = Los datos provienen de una distribución normal.

P-valor < alfa aceptar Ha = Los datos NO provienen de una distribución normal.

Para nuestro trabajo:

Tabla 13: CRITERIO PARA LA NORMALIDAD

NORMALIDAD		
P-Valor (pre test) =0.097	>	Alfa = 0.05
P- Valor (post test) = 0.108	>	Alfa = 0.05
CONCLUSIÓN:		

Los datos del pensamiento algorítmico provienen de una distribución normal tanto del pre como del postest.

FUENTE: Elaborado por Tesista

4.1.3.5. PASO 5 (Decisión Estadística)

Si la probabilidad obtenida P-Valor \leq alfa, rechace la H_0 (Se acepta la H_a).

EN LA HIPÓTESIS GENERAL:

Utilizando el SPSS con la Prueba T para muestras relacionadas nos muestra:

Tabla 14: PRUEBA DE MUESTRAS RELACIONADAS

Estadísticas de muestras emparejadas					
		Media	N	Desv. Desviación	Desv. Error promedio
Par 1	Variable Dependiente del pretest	6,45	42	3,358	,518
	Variable Dependiente del postest	9,90	42	4,107	,634

Correlaciones de muestras emparejadas				
		N	Correlación	Sig.
Par 1	Variable Dependiente del pretest & Variable Dependiente del postest	42	,608	,000

Prueba de muestras emparejadas									
		Diferencias emparejadas							
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	Variable Dependiente del pretest - Variable Dependiente del postest	-3,452	3,373	,520	-4,503	-2,401	-6,634	41	,000

FUENTE: Elaborado por Tesista utilizando SPSS

Entonces se dice que nuestra significancia o nuestro P-valor es 0.000

\leq a 0.05 por lo tanto se acepta la hipótesis alterna que dice:

Ha: La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

Usando El Anova Podemos Afirmar Lo Siguiente:

Tabla 15: PRUEBA UTILIZANDO ANOVA

Descriptivos									
		N	Media	Desv. Desviación	Desv. Error	95% del intervalo de confianza para la media		Mínimo	Máximo
						Límite inferior	Límite superior		
Variable Dependiente del pretest	A	19	5,84	3,671	,842	4,07	7,61	0	13
	B	23	6,96	3,067	,640	5,63	8,28	2	14
	Total	42	6,45	3,358	,518	5,41	7,50	0	14
Variable Dependiente del postest	A	19	6,79	3,119	,716	5,29	8,29	1	13
	B	23	12,48	2,858	,596	11,24	13,71	7	18
	Total	42	9,90	4,107	,634	8,62	11,18	1	18

ANOVA						
		Suma de cuadrados	gl	Media cuadrática	F	Sig.
	Dentro de grupos	449,483	40	11,237		
	Total	462,405	41			
Variable Dependiente del postest	Entre grupos	336,722	1	336,722	37,952	,000
	Dentro de grupos	354,897	40	8,872		
	Total	691,619	41			

FUENTE: Elaborado por Tesista utilizando SPSS

La prueba estadística entre las notas de la variable dependiente del pretest de los grupos del sexto grado A y B existe una significancia o P- valor de 0.290 en donde nos dice que No existe una influencia positiva en el desarrollo del pensamiento algorítmico de los estudiantes

del Augusto Cardich entre el sexto grado A y sexto grado B en la prueba del pre test. Obviamente no existe ningún desarrollo del pensamiento algorítmico ya que aún no se ha aplicado el lenguaje de programación SCRATCH al sexto grado B.

Nos dice también que en las notas de la variable dependiente del postest de los grupos del sexto grado A y sexto grado B existe una significancia o P-valor de 0.000 en donde nos demuestra que luego de aplicar el lenguaje de programación SCRATCH al sexto grado B, sí existe dicha influencia en el desarrollo del pensamiento algorítmico entre los alumnos del sexto grado A y B.

EN LAS HIPÓTESIS ESPECÍFICAS:

Hipótesis Específica 1:

Utilizando el SPSS con la Prueba T para muestras relacionadas nos muestra:

Tabla 16: PRUEBA DE MUESTRAS RELACIONADAS PARA LA HE1

FUENTE: Elaborado por Tesista utilizando SPSS

Estadísticas de muestras emparejadas					
		Media	N	Desv. Desviación	Desv. Error promedio
Par 1	D1PRE= Fase de analizar el problema en el pre test	1,38	42	1,847	,285
	D1POST= Fase de analizar el problema en el post test	3,36	42	2,304	,356

Correlaciones de muestras emparejadas				
		N	Correlación	Sig.
Par 1	D1PRE= Fase de analizar el problema en el pre test & D1POST= Fase de analizar el problema en el post test	42	,529	,000

Prueba de muestras emparejadas									
		Diferencias emparejadas							
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	D1PRE= Fase de analizar el problema en el pre test - D1POST= Fase de analizar el problema en el post test	-1,976	2,054	,317	-2,616	-1,336	-6,235	41	,000

Entonces se dice que nuestra significancia o nuestro P-valor es 0.000 <= a 0.05 por lo tanto se acepta la hipótesis alterna que dice:

HE1a: La aplicación del lenguaje de programación SCRATCH si influye positivamente en el *desarrollo de la fase de análisis del problema* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

Usando el ANOVA podemos afirmar lo siguiente:

Tabla 17: PRUEBA UTILIZANDO ANOVA PARA HE1

Descriptivos									
		N	Media	Desv. Desviación	Desv. Error	95% del intervalo de confianza para la media		Mínimo	Máximo
						Límite inferior	Límite superior		
D1PRE= Fase de analizar el problema en el pre test	A	19	1,26	1,695	,389	,45	2,08	0	5
	B	23	1,48	1,997	,416	,61	2,34	0	6
	Total	42	1,38	1,847	,285	,81	1,96	0	6
D1POST= Fase de analizar el problema en el post test	A	19	1,89	1,595	,366	1,13	2,66	0	5
	B	23	4,57	2,107	,439	3,65	5,48	0	8
	Total	42	3,36	2,304	,356	2,64	4,08	0	8

ANOVA						
		Suma de cuadrados	gl	Media cuadrática	F	Sig.
D1PRE= Fase de analizar el problema en el pre test	Entre grupos	,481	1	,481	,138	,712
	Dentro de grupos	139,423	40	3,486		
	Total	139,905	41			
D1POST= Fase de analizar el problema en el post test	Entre grupos	74,201	1	74,201	20,692	,000
	Dentro de grupos	143,442	40	3,586		
	Total	217,643	41			

FUENTE: Elaborado por Tesista utilizando SPSS

La prueba estadística entre las notas de la dimensión 1 que es la fase de analizar el problema en el pre test de los grupos del sexto grado A y B existe una significancia o P- valor de 0.712 en donde nos dice que: No existe una influencia positiva en el desarrollo de la fase de análisis del problema del pensamiento algorítmico de los estudiantes del Augusto Cardich entre el sexto grado A y sexto grado B en la prueba del pre test. Obviamente no existe ningún desarrollo de esa fase del pensamiento algorítmico ya que aún no se ha aplicado el lenguaje de programación SCRATCH al sexto grado B para el desarrollo de dicha fase. Nos dice también que en las notas de la variable dependiente del postest de los grupos del sexto grado A y sexto grado B existe una

significancia o P-valor de 0.000 en donde nos demuestra que luego de aplicar el lenguaje de programación SCRATCH al sexto grado B, SÍ existe dicha influencia en el desarrollo de la fase de análisis del problema del pensamiento algorítmico entre los alumnos del sexto grado A y B.

Hipótesis Específica 2:

Utilizando el SPSS con la Prueba T para muestras relacionadas nos muestra:

Tabla 18: PRUEBA DE MUESTRAS RELACIONADAS PARA LA HE2

Estadísticas de muestras emparejadas					
		Media	N	Desv. Desviación	Desv. Error promedio
Par 1	D2PRE= Fase de diseñar y traducir el algoritmo en el pre test	2,10	42	1,605	,248
	D2POST= Fase de diseñar y traducir el algoritmo en el post test	3,14	42	1,523	,235

Correlaciones de muestras emparejadas				
		N	Correlación	Sig.
Par 1	D2PRE= Fase de diseñar y traducir el algoritmo en el pre test & D2POST= Fase de diseñar y traducir el algoritmo en el post test	42	,593	,000

Prueba de muestras emparejadas									
		Diferencias emparejadas			95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
		Media	Desv. Desviación	Desv. Error promedio	Inferior	Superior			
Par 1	D2PRE= Fase de diseñar y traducir el algoritmo en el pre test - D2POST= Fase de diseñar y traducir el algoritmo en el post test	-1,048	1,413	,218	-1,488	-,607	-4,804	41	,000

FUENTE: Elaborado por Tesista utilizando SPSS

Entonces se dice que nuestra significancia o nuestro P-valor es $0.000 \leq a$ 0.05 por lo tanto se acepta la hipótesis alterna que dice:

HE2a: La aplicación del lenguaje de programación SCRATCH influye positivamente en el *desarrollo de la fase de diseño y traducción del algoritmo* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

Usando el ANOVA podemos afirmar lo siguiente:

Tabla 19: PRUEBA UTILIZANDO ANOVA PARA LA HE 2

Descriptivos								
		N	Media	Desv. Desviación	Desv. Error	95% del intervalo de confianza para la media		
						Límite inferior	Límite superior	Mínimo Máximo
D2PRE= Fase de diseñar y traducir el algoritmo en el pre test	A	19	2,21	1,751	,402	1,37	3,05	0 5
	B	23	2,00	1,508	,314	1,35	2,65	0 5
	Total	42	2,10	1,605	,248	1,60	2,60	0 5
D2POST= Fase de diseñar y traducir el algoritmo en el post test	A	19	2,53	1,541	,353	1,78	3,27	0 5
	B	23	3,65	1,335	,278	3,07	4,23	2 6
	Total	42	3,14	1,523	,235	2,67	3,62	0 6

ANOVA						
		Suma de cuadrados	gl	Media cuadrática	F	Sig.
D2PRE= Fase de diseñar y traducir el algoritmo en el pre test	Entre grupos	,461	1	,461	,175	,678
	Dentro de grupos	105,158	40	2,629		
	Total	105,619	41			
D2POST= Fase de diseñar y traducir el algoritmo en el post test	Entre grupos	13,189	1	13,189	6,437	,015
	Dentro de grupos	81,954	40	2,049		
	Total	95,143	41			

FUENTE: Elaborado por Tesista utilizando SPSS

La prueba estadística entre las notas de la dimensión 2 que es la fase de diseño y traducción del algoritmo en el pre test de los grupos del sexto

grado A y B existe una significancia o P- valor de 0.678 en donde nos dice que: No existe una influencia positiva en el desarrollo de la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los estudiantes del Augusto Cardich entre el sexto grado A y sexto grado B en la prueba del pre test. Obviamente no existe ningún desarrollo de esa fase del pensamiento algorítmico ya que aún no se ha aplicado el lenguaje de programación SCRATCH al sexto grado B para el desarrollo de dicha fase.

Nos dice también que en las notas de la variable dependiente del postest de los grupos del sexto grado A y sexto grado B existe una significancia o P-valor de 0.015 en donde nos demuestra que luego de aplicar el lenguaje de programación SCRATCH al sexto grado B, SÍ existe dicha influencia en el desarrollo de diseño y traducción del algoritmo del pensamiento algorítmico entre los alumnos del sexto grado A y B.

Hipótesis Específica 3:

Utilizando el SPSS con la Prueba T para muestras relacionadas nos muestra:

Tabla 20: PRUEBA DE MUESTRAS RELACIONADAS PARA LA HE 3

Estadísticas de muestras emparejadas					
		Media	N	Desv. Desviación	Desv. Error promedio
Par 1	D2PRE= Fase de depurar el programa en el pretest	2,98	42	1,630	,252
	D3POST= Fase de depurar el programa en el post test	3,40	42	1,781	,275

Correlaciones de muestras emparejadas				
		N	Correlación	Sig.
Par 1	D2PRE= Fase de depurar el programa en el pretest & D3POST= Fase de depurar el programa en el post test	42	,885	,000

Prueba de muestras emparejadas									
		Diferencias emparejadas							
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	D2PRE= Fase de depurar el programa en el pretest - D3POST= Fase de depurar el programa en el post test	- ,429	,331	,128	-,687	-,170	-3,344	41	,002

FUENTE: Elaborado por Tesista utilizando SPSS

Entonces se dice que nuestra significancia o nuestro P-valor es $0.002 \leq \alpha$ a 0.05 por lo tanto se acepta la hipótesis alterna que dice:

HE3a: La aplicación del lenguaje de programación SCRATCH influye positivamente en el *desarrollo de la fase de depuración del programa* del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.

Usando el ANOVA podemos afirmar lo siguiente:

Tabla 21: PRUEBA UTILIZANDO ANOVA PARA LA HE 3

FUENTE: Elaborado por Tesista utilizando SPSS

Descriptivos									
		N	Media	Desv. Desviación	Desv. Error	95% del intervalo de confianza para la media		Mínimo	Máximo
						Límite inferior	Límite superior		
D2PRE= Fase de depurar el programa en el pretest	A	19	2,37	1,499	,344	1,65	3,09	0	4
	B	23	3,48	1,592	,332	2,79	4,17	0	6
	Total	42	2,98	1,630	,252	2,47	3,48	0	6
D3POST= Fase de depurar el programa en el posttest	A	19	2,37	1,499	,344	1,65	3,09	0	4
	B	23	4,26	1,544	,322	3,59	4,93	1	6
	Total	42	3,40	1,781	,275	2,85	3,96	0	6

ANOVA						
		Suma de cuadrados	gl	Media cuadrática	F	Sig.
D2PRE= Fase de depurar el programa en el pretest	Entre grupos	12,816	1	12,816	5,331	,026
	Dentro de grupos	96,160	40	2,404		
	Total	108,976	41			
D3POST= Fase de depurar el programa en el posttest	Entre grupos	37,263	1	37,263	16,052	,000
	Dentro de grupos	92,856	40	2,321		
	Total	130,119	41			

La prueba estadística entre las notas de la dimensión 3 que es la fase de depuración del programa en el pre test de los grupos del sexto grado A y B existe una significancia o P- valor de 0.026 en donde nos dice que: SI existe una influencia positiva en el desarrollo de la fase de depuración del programa del pensamiento algorítmico de los estudiantes del Augusto Cardich entre el sexto grado A y sexto grado B en la prueba del pre test. El resultado es positivo en el pretest a pesar de que aún no se aplicó el lenguaje de programación SCRATCH en el pretest porque los estudiantes si tienen un indicio de desarrollo en esta fase.

Nos dice también que en las notas de la variable dependiente del postest de los grupos del sexto grado A y sexto grado B existe una significancia o P-valor de 0.000 en donde nos demuestra que luego de aplicar el

lenguaje de programación SCRATCH al sexto grado B, SÍ existe dicha influencia en el desarrollo de la fase de depuración del programa del pensamiento algorítmico entre los alumnos del sexto grado A y B.

V. DISCUSIÓN DE LOS RESULTADOS

Para la realización de la presente investigación fue indispensable hacer una revisión de tesis y artículos realizados anteriormente con similitudes al tema abordado en esta investigación, se analiza los enfoques tomados por los autores y al resultado que deseaban llegar cada involucrado.

En la presente tesis (López García 2014) se concuerda que sí existe un claro indicador de cambio en el desempeño del estudiante pero esos desempeños no explican suficientemente el uso de conceptos específicos del pensamiento algorítmico ya que se necesita de mucho más tiempo para realizar las actividades que conllevan al pensamiento algorítmico, a pesar que la investigación de López García se dio en 4 unidades que conllevan a maso menos (entre 4 a 6 meses) el presente trabajo de investigación sólo se dio en 1 unidad (1 mes y medio) pero a pesar del corto periodo si muestra evidencia del desarrollo del pensamiento algorítmico en ese periodo en todas sus fases.

En la presente tesis (Vidal, Cabezas, Parra López, 2015) se concuerda que el programa SCRATCH si es una herramienta propicia para el desarrollo del pensamiento algorítmico, donde se presenta un ambiente en el cual los estudiantes se motivan, participan en la propuesta de soluciones y colaboran entre ellos porque el lenguaje de programación SCRATCH es muy interactivo y fácil de usar.

En la presente tesis (Juan cadillo león, 2017) llega a la conclusión que se ha logrado una aproximación a la definición de los componentes del pensamiento computacional, midiéndose el aspecto de conceptos algorítmicos pero en su investigación él tomó una muestra de 12 estudiantes del sexto grado del nivel primario y que no tienen un grupo de control, sólo toma datos de pretest y postest de esos 12 estudiantes; en cambio en esta investigación la muestra es de 42 estudiantes divididos en 23 estudiantes para el grupo experimental y 19 estudiantes para el grupo de control por consiguiente se concluye que tomar ambas muestras permite analizar y comparar mejor los resultados sobre la influencia del lenguaje de programación SCRATCH para el desarrollo del pensamiento algorítmico.

CONCLUSIONES

- ✓ Al aplicar el lenguaje de programación SCRATCH al grupo experimental (sexto grado B) sí existe una mejora en su desarrollo de pensamiento algorítmico en comparación del grupo de control (sexto grado A) ya que los alumnos pueden secuencializar mejor los pasos a seguir para la resolución de un problema.
- ✓ En la fase de Análisis del problema los alumnos del grupo experimental (sexto grado B) analizan e identifican mejor el problema en comparación con los alumnos del grupo de control (sexto grado A).
- ✓ En la fase de Diseño y Traducción del algoritmo los alumnos del grupo experimental (sexto grado B) representan gráficamente de mejor manera las instrucciones a seguir en un problema que los alumnos del grupo de control (sexto grado A).
- ✓ En la fase de Depuración del programa los alumnos del grupo experimental (sexto grado B) revisan y corrigen errores que podrían suscitar en un problema en comparación con los alumnos del grupo de control (sexto grado A).
- ✓ Acercar a los niños y jóvenes a la programación ya no es un DESAFÍO, porque el lenguaje de programación SCRATCH es muy interactivo y fácil de usar debido a su programación de bloques “agarrar y soltar”.
- ✓ Se evidencia que los alumnos del sexto grado B que es el grupo experimental donde se aplicó el lenguaje de programación SCRATCH no sólo mejoraron el

desarrollo de su pensamiento algorítmico en el área de COMPUTACIÓN sino que también mejoraron en las demás áreas que llevan en la I.E.P Augusto Cardich.

RECOMENDACIONES

- ✓ Se sugiere a la I.E.P Augusto Cardich programar talleres sobre el Lenguaje de programación SCRATCH en todos los grados y así desarrollar el pensamiento algorítmico de los estudiantes además nos conlleva a tener una base en programación y consecuentemente entrar en el ámbito de la Robótica.
- ✓ Es evidente que en la presente investigación sí existe una clara influencia del lenguaje de programación SCRATCH en el desarrollo del pensamiento algorítmico, pero se sugiere que la aplicación y desarrollo de las sesiones y actividades sean mucho más prolongados ya que para saber conceptos específicos del pensamiento algorítmico y que los alumnos puedan aplicarlo a su vida diaria conlleva de mucho más tiempo.
- ✓ Se sugiere a los bachilleres y profesionales tomar en cuenta la presente investigación como un inicio al desarrollo del pensamiento computacional para su posterior investigación.
- ✓ Se sugiere a las instituciones educativas públicas y privadas que consideren al lenguaje de programación Scratch como parte de su contenido en la elaboración de sus diseños metodológicos y pedagógicos, ya que permite fortalecer las habilidades de resolución de problemas y así mismo mejorar la toma de decisiones.

BIBLIOGRAFÍA

- Carrasco, E. (2005). *Metodología de la Investigación científica*. Lima: Editorial San Marcos.
- Claudia, A (2013). *Un lenguaje para aprender a programar jugando*. Recuperado de: http://www0.unsl.edu.ar/~agaris/slides/t2_scratchAula.pdf.
- Monjelat, N. San Martín, P. (2016). *Programar con Scratch en contextos educativos*. Recuperado de: <https://www.redalyc.org/pdf/1531/153146047006.pdf>
- Cadillo, J. (2012). *Scratch y Wedo como herramientas para desarrollar el pensamiento computacional en niños de 11 a 12 años – I.E. 87003-1 “Jesús Nazareno”*. (Tesis de maestría). Universidad Nacional Santiago Antúnez de Mayolo, Perú.
- Echevarría, D. (2018). *Mi Problema No es Problema con el Método de Polya* (título de segunda especialidad en Didáctica de la Educación Primaria). Universidad Nacional Hermilio Valdizán. Huánuco.
- López, J. (2014). *Actividades de aula con scratch que favorecen el uso del pensamiento algorítmico*. (Tesis de postgrado para optar la Maestría en Educación). Universidad ICESI. Cali – Colombia.
- Cegarra, M. Caballos, L (2012). *Planteamiento del problema*.
- EDUTEKA. (2013). *Guía de referencia de Scratch. 2.0*. Recuperado de: <http://eduteka.icesi.edu.co/articulos/Scratch20>

- Garis, A. Albornoz, C. (2013). *Un lenguaje para aprender a programar jugando*. Proyecto de Extensión “Estrategias creativas de resolución de problemas. Pp. 17-25.
- López, J. *Actividades de aula con scratch que favorecen el uso del pensamiento algorítmico*. (tesis de maestría) Cali. Universidad ICESI.
- Mingos, D. (2018). *Mi problema no es problema con el método de Polya*. (tesis de Especialidad en Didáctica de la Educación Primaria). Huánuco. Universidad Nacional Hermilio Valdizán.
- Monjelat, N. San Martín, P. (2016). *Programar con Scratch en contextos educativos: ¿Asimilar directrices o co-construir Tecnologías para la Inclusión Social?* Recuperado de: researchgate.net/publication/301648926_Programar_con_Scratch_en_contextos_educativos_Asimilar_directrices_o_co-construir_Tecnologias_para_la_Inclusion_Social
- López, J. (2011). *Algoritmos y programación*. Recuperado de:
- Revista Institucional UNLP. (s/f). *Por qué “pensar Algoritmos” es tan importante en Informática?*. Volumen (Nº 4). P.21
- Instituto Técnico Superior Industrial Tecnología e Informática. (s/f). *Programación de computadoras*.
- Artecona, F., Boneti, E., Darino, C., Mello, F. Rosa, M., Scòpise, M. (2017). *Pensamiento computacional*. Recuperado de:

- López, J. (2019). *Pensamiento con Scratch*. Recuperado de: <http://eduteka.icesi.edu.co/pdfdir/tesis-juan-carlos-lopez.pdf>
- Camilo, J. (2017). *Scratch como herramienta para la enseñanza de la programación en la Educación Primaria* (Tesis doctoral). Madrid. Facultad de Educación. Universidad Camilo José Vega.
- Pascual, D. (2017). *Scratch como herramienta para la enseñanza de la programación en la Educación Primaria*. (tesis de maestría). Rioja. Universidad de la Rioja.
- León, C. *Scratch y WeDo como herramientas para desarrollar el pensamiento computacional en niños de 11 y 12 años*. Recuperado de: <https://recursos.portaleducoas.org/publicaciones/scratch-y-wedo-como-herramientas-para-desarrollar-el-pensamiento-computacional-en-ni?audiencia=2&area=25&country=>
- Hervás, R. (2012). *Método para el aprendizaje de entornos y lenguajes de programación basado en prototipado ágil*. Recuperado de: <https://core.ac.uk/download/pdf/41791857.pdf>
- Pérez, A. (2017). Evaluación del Pensamiento Computacional en Educación. *Revista Interuniversitaria de Investigación en Tecnología Educativa (RIITE)*. N^o 3. Diciembre 2017 pp. 25-39 ISSN.



ANEXOS

ANEXO 1: CUESTIONARIO DE VALIDACIÓN DEL INSTRUMENTO	131
ANEXO 2: VALIDACIÓN DEL INSTRUMENTO EXPERTO 1	132
ANEXO 3: VALIDACIÓN DEL INSTRUMENTO EXPERTO 2	133
ANEXO 4: VALIDACIÓN DEL INSTRUMENTO EXPERTO 3	134
ANEXO 5: PRUEBA DE CONOCIMIENTO	135
ANEXO 6: REGISTRO DE INCIDENCIAS	141
ANEXO 7: SESIONES DE APRENDIZAJE	142
ANEXO 8: DATOS PARA EL SPSS	151
ANEXO 9: MATRIZ DE CONSISTENCIA	152
ANEXO 10: PRUEBA DE CONOCIMIENTO PRETEST-EJEMPLO1	153
ANEXO 11: PRUEBA DE CONOCIMIENTO POSTEST-EJEMPLO 1	159
ANEXO 12: FOTOGRAFÍAS	165



ANEXO 1: CUESTIONARIO DE VALIDACIÓN DEL INSTRUMENTO

VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	PROM	C. PARCIAL
Variable Dependiente:	problema N° 1; plantea y resuelve el problema.						
PENSAMIENTO ALGORÍTMICO	problema N° 2; Cuáles son los datos de entrada,proceso y salida.?						
Dimensión 1:	problema N° 3; ... Cuáles son los datos de entrada,proceso y salida.?						
FASE DE ANALIZAR EL PROBLEMA	problema N° 4;... Cuáles son los datos de entrada,proceso y salida.?						
	TOTAL						
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	TOTAL	C. PARCIAL
Variable Dependiente:	Problema N° 6; Dibuja y ordena los objetos.						
PENSAMIENTO ALGORÍTMICO	Problema N° 8; Cuáles son los pasos a seguir para lav arse las manos.						
Dimensión 2:	Problema N° 7; Enumera en orden lógico los pasos siguientes (para lav arse los dientes)						
FASE DE DE DISEÑAR Y TRADUCIR EL ALGORITMO							
	TOTAL						
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	TOTAL	C. PARCIAL
Variable Dependiente:	problema N° 9; Completa el ejercicio y comenta si el resultado es verdadero o falso, porqué?						
PENSAMIENTO ALGORÍTMICO	problema N° 10; Señala en el dibujo y marca la alternativa correcta para que pacman llegue a comer al fantasma.						
Dimensión 3:	problema N° 5; Poner verdadero (V) o falso (F) en los siguientes ejercicios.						
FASE DE DE DEPURAR EL PROGRAMA							
	TOTAL						



ANEXO 2: VALIDACIÓN DEL INSTRUMENTO EXPERTO 1

	UNIVERSIDAD NACIONAL HERMILIO VALDIZÁN HUÁNUCO - PERÚ ESCUELA DE POSGRADO							
VALIDACIÓN DE INSTRUMENTO								
Título de la Investigación:	APLICACIÓN DEL LENGUAJE DE PROGRAMACIÓN SCRATCH PARA EL DESARROLLO DEL PENSAMIENTO ALGORÍTMICO EN LOS ALUMNOS DEL SEXTO GRADO DEL NIVEL PRIMARIO EN LA I.E.P AUGUSTO CARDICH.							
Nombre del Tesista:	ROMER RAMIRO RIVERA ILLATOPA	Asesor:	DR. ABIMAEEL ADAM FRANCISCO PAREDES					
Nombre del experto:	DAVID ESQUIVEL MARQUEZ	Especialidad:	INGENIERÍA DE SISTEMAS					
Calificar con 1, 2, 3, ó 4 cada ítem respecto a los criterios de relevancia, coherencia, suficiencia y claridad								
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	PROM	C. PARCIAL	
Variable Dependiente:	problema N° 1; plantea y resuelve el problema.	3	4	4	4	3,75	4	
PENSAMIENTO ALGORÍTMICO	problema N° 2; Cuáles son los datos de entrada, proceso y salida.?	4	4	4	4	4,00	4	
Dimensión:	problema N° 3; ... Cuáles son los datos de entrada, proceso y salida.?	4	3	4	4	3,75	4	
FASE DE ANALIZAR EL PROBLEMA	problema N° 4;... Cuáles son los datos de entrada, proceso y salida.?	4	4	3	4	3,75	4	
TOTAL		3,75	3,75	3,75	4	3,81	4,00	
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	TOTAL	C. PARCIAL	
Variable Dependiente:	Problema N° 6; Dibuja y ordena los objetos.	4	3	4	3	3,50	4	
PENSAMIENTO ALGORÍTMICO	Problema N° 8; Cuáles son los pasos a seguir para lavarse las manos.	3	4	4	4	3,75	4	
Dimensión:	Problema N° 7; Enumera en orden lógico los pasos siguientes (para lavarse los dientes)	3	4	4	3	3,50	4	
FASE DE DISEÑAR Y TRADUCIR EL ALGORITMO								
TOTAL		3,33	3,67	4,00	3,33	3,58	4,00	
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	TOTAL	C. PARCIAL	
Variable Dependiente:	problema N° 9; Completa el ejercicio y comenta si el resultado es verdadero o falso, porqué?	4	4	4	4	4,00	4	
PENSAMIENTO ALGORÍTMICO	problema N° 10; Señala en el dibujo y marca la alternativa correcta para que pacman llegue a comer al fantasma.	4	4	4	4	4,00	4	
Dimensión:	problema N° 5; Poner verdadero (V) o falso (F) en los siguientes ejercicios.	4	4	4	4	4,00	4	
FASE DE DEPURAR EL PROGRAMA								
TOTAL		4,00	4,00	4,00	4,00	4,00	4	
CALIFICACIÓN: 1=No cumple; 2=Nivel Bajo; 3=Nivel moderado; 4= Nivel Alto								
¿Hay alguna dimensión o ítem que no ha sido evaluada? SI () NO ().		CALIFICACIÓN DEL INSTRUMENTO				4		
En caso de SI ¿Qué dimensión o ítem falta?		DECISIÓN DEL EXPERTO						
		CALIFICACIÓN:				NIVEL ALTO		
		EL INSTRUMENTO DEBE SER APLICADO SI () NO ()						
FIRMA Y SELLO DEL JUEZ								

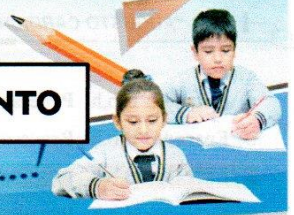


ANEXO 3: VALIDACIÓN DEL INSTRUMENTO EXPERTO 2

	UNIVERSIDAD NACIONAL HERMILO VALDIZÁN HUÁNUCO - PERÚ ESCUELA DE POSGRADO						
VALIDACIÓN DE INSTRUMENTO							
Título de la Investigación:	APLICACIÓN DEL LENGUAJE DE PROGRAMACIÓN SCRATCH PARA EL DESARROLLO DEL PENSAMIENTO ALGORÍTMICO EN LOS ALUMNOS DEL SEXTO GRADO DEL NIVEL PRIMARIO EN LA I.E.P AUGUSTO CARDICH.						
Nombre del Tesista:	ROMER RAMIRO RIVERA ILLATOPA	Asesor:	DR. ABIMAEEL ADAM FRANCISCO PAREDES				
Nombre del experto:	GROVER VICENTE MARCHAN	Especialidad:	INGENIERÍA DE SISTEMAS				
Calificar con 1, 2, 3, ó 4 cada ítem respecto a los criterios de relevancia, coherencia, suficiencia y claridad							
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	PROM	C. PARCIAL
Variable Dependiente:	problema N° 1; plantea y resuelve el problema.	4	4	4	4	4,00	4
PENSAMIENTO ALGORÍTMICO	problema N° 2; Cuáles son los datos de entrada, proceso y salida.?	4	4	4	4	4,00	4
Dimensión:	problema N° 3; ... Cuáles son los datos de entrada, proceso y salida.?	4	4	4	4	4,00	4
FASE DE ANALIZAR EL PROBLEMA	problema N° 4;... Cuáles son los datos de entrada, proceso y salida.?	4	4	4	4	4,00	4
TOTAL		4	4	4	4	4,00	4,00
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	TOTAL	C. PARCIAL
Variable Dependiente:	Problema N° 6; Dibuja y ordena los objetos.	4	4	4	4	4,00	4
PENSAMIENTO ALGORÍTMICO	Problema N° 8; Cuáles son los pasos a seguir para lavarse las manos.	4	4	4	4	4,00	4
Dimensión:	Problema N° 7; Enumera en orden lógico los pasos siguientes (para lavarse los dientes)	4	4	4	4	4,00	4
FASE DE DE DISEÑAR Y TRADUCIR EL ALGORITMO							
TOTAL		4,00	4,00	4,00	4,00	4,00	4,00
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	TOTAL	C. PARCIAL
Variable Dependiente:	problema N° 9; Completa el ejercicio y comenta si el resultado es verdadero o falso, porqué?	4	4	4	4	4,00	4
PENSAMIENTO ALGORÍTMICO	problema N° 10; Señala en el dibujo y marca la alternativa correcta para que pacman llegue a comer al fantasma.	4	4	4	4	4,00	4
Dimensión:	problema N° 5; Poner verdadero (V) o falso (F) en los siguientes ejercicios.	4	4	4	4	4,00	4
FASE DE DE DEPURAR EL PROGRAMA							
TOTAL		4,00	4,00	4,00	4,00	4,00	4
CALIFICACIÓN: 1=No cumple; 2=Nivel Bajo; 3=Nivel moderado; 4= Nivel Alto							
¿Hay alguna dimensión o ítem que no ha sido evaluada? SI () NO ().		CALIFICACIÓN DEL INSTRUMENTO				4	
En caso de SI ¿Qué dimensión o ítem falta?		DECISIÓN DEL EXPERTO					
		CALIFICACIÓN:				NIVEL ALTO	
		EL INSTRUMENTO DEBE SER APLICADO SI() NO()					
FIRMA Y SELLO DEL JUEZ							

ANEXO 4: VALIDACIÓN DEL INSTRUMENTO EXPERTO 3

	UNIVERSIDAD NACIONAL HERMILIO VALDIZÁN HUÁNUCO - PERÚ ESCUELA DE POSGRADO						
VALIDACIÓN DE INSTRUMENTO							
Título de la Investigación:	APLICACIÓN DEL LENGUAJE DE PROGRAMACIÓN SCRATCH PARA EL DESARROLLO DEL PENSAMIENTO ALGORÍTMICO EN LOS ALUMNOS DEL SEXTO GRADO DEL NIVEL PRIMARIO EN LA I.E.P AUGUSTO CARDICH.						
Nombre del Tesista:	ROMER RAMIRO RIVERA ILLATOPA	Asesor:	DR. ABIMAEEL ADAM FRANCISCO PAREDES				
Nombre del experto:	BEKER SANTOS	Especialidad:	INGENIERÍA DE SISTEMAS				
Calificar con 1, 2, 3, ó 4 cada ítem respecto a los criterios de relevancia, coherencia, suficiencia y claridad							
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	PROM	C. PARCIAL
Variable Dependiente:	problema N° 1; plantea y resuelve el problema.	3	3	4	3	3,25	3
PENSAMIENTO ALGORÍTMICO	problema N° 2; Cuáles son los datos de entrada, proceso y salida.?	2	3	3	3	2,75	3
Dimensión 1:	problema N° 3; ... Cuáles son los datos de entrada, proceso y salida.?	3	4	3	3	3,25	3
FASE DE ANALIZAR EL PROBLEMA	problema N° 4; ... Cuáles son los datos de entrada, proceso y salida.?	3	4	3	3	3,25	3
TOTAL		2,75	3,5	3,25	3	3,13	3,00
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	TOTAL	C. PARCIAL
Variable Dependiente:	Problema N° 6; Dibuja y ordena los objetos.	4	4	4	4	4,00	4
PENSAMIENTO ALGORÍTMICO	Problema N° 8; Cuáles son los pasos a seguir para lavarse las manos.	3	4	4	4	3,75	4
Dimensión 2:	Problema N° 7; Enumera en orden lógico los pasos siguientes (para lavarse los dientes)	4	3	4	4	3,75	4
FASE DE DISEÑAR Y TRADUCIR EL ALGORITMO							
TOTAL		3,67	3,67	4,00	4,00	3,83	4
VARIABLE - DIMENSIÓN	ÍTEM	RELEVANCIA	COHERENCIA	SUFICIENCIA	CLARIDAD	TOTAL	C. PARCIAL
Variable Dependiente:	problema N° 9; Completa el ejercicio y comenta si el resultado es verdadero o falso, por qué?	3	4	4	3	3,50	4
PENSAMIENTO ALGORÍTMICO	problema N° 10; Señala en el dibujo y marca la alternativa correcta para que pacman llegue a comer al fantasma.	3	3	4	4	3,50	4
Dimensión 3:	problema N° 5; Poner verdadero (V) o falso (F) en los siguientes ejercicios.	4	4	4	4	4,00	4
FASE DE DEPURAR EL PROGRAMA							
TOTAL		3,33	3,67	4,00	3,67	3,67	4
CALIFICACIÓN: 1=No cumple; 2=Nivel Bajo; 3=Nivel moderado; 4= Nivel Alto							
¿Hay alguna dimensión o ítem que no ha sido evaluada? SI() NO().		CALIFICACIÓN DEL INSTRUMENTO					4
En caso de SI ¿Qué dimensión o ítem falta?		DECISIÓN DEL EXPERTO					
		CALIFICACIÓN: NIVEL ALTO					
		EL INSTRUMENTO DEBE SER APLICADO SI() NO()					
FIRMA Y SELLO DEL JUEZ							

ANEXO 5: PRUEBA DE CONOCIMIENTO



INSTITUCIÓN EDUCATIVA PRIVADA
AUGUSTO CARDICH

PRUEBA DE CONOCIMIENTO

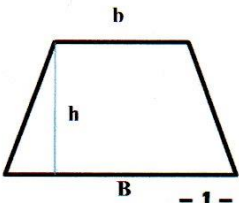
2019 *Un colegio diferente, para niños y jóvenes diferentes....!*

6TO GRADO

1. Juan Felipe es jefe de bodega en una fábrica de pañales desechables y una de las tareas del día consiste en llamar al proveedor de los empaques y ordenarle la cantidad suficiente de cajas para empacar los pañales fabricados en la semana próxima. El jefe de producción le informó ayer a Juan Felipe que la producción diaria será de 744 pañales y en cada caja cabe una docena de ellos. ¿Qué debe hacer Felipe?

En el siguiente problema plantea y resuelve el problema.

2. El perímetro de un trapecio es igual a la suma de todos sus lados y su área es igual a la suma de la base mayor (B) más la base menor (b) dividido todo entre 2 para después multiplicar este resultado por la altura (h). Considerando el trapecio de la siguiente figura y de acuerdo al Modelo de solución, identifique los elementos de entrada, proceso y salida necesarios para resolver el problema del cálculo del área del Trapecio.



B **h** **b**

— 1 —

INSTITUCIÓN EDUCATIVA PRIVADA
Augusto Cardich

Un colegio diferente, para niños y jóvenes diferentes....!

- a) **Entrada:** B,b,h **Proceso:** $A=b+B/2*h$ **Salida:** Valor de A
- b) **Entrada:** B,b,h **Proceso:** $A=b+(B/2)*h$ **Salida:** Valor de A
- c) **Entrada:** B,b,h **Proceso:** $A=(b+B)/2*h$ **Salida:** Valor de A
- d) **Entrada:** B,b,h **Proceso:** $A=b+B/(2*h)$ **Salida:** Valor de A

3. Se necesita obtener el puntaje final de un estudiante en un examen considerando el número de respuestas correctas, incorrectas y en blancos, si se sabe que por cada respuesta correcta tendrá 4 puntos, respuestas incorrectas tendrá -1 y respuestas en blanco tendrá 0. ¿cuáles son sus datos de entrada, proceso y salida?

RC = Número de respuestas correctas.

RI = Número de respuestas incorrectas.

RB = Número de respuestas en blanco

PF = PUNTAJE FINAL

- a) **Entrada:** RC,RI,RB,PF **Proceso:** $RB=(4*RC)+(-1*RI)+PF$ **Salida:** Valor de RB
- b) **Entrada:** RC,RI,RB **Proceso:** $PF=(RC*4)+(RI*-1)$ **Salida:** Valor de PF
- c) **Entrada:** RC,RI,RB **Proceso:** $PF=(RC*-1)+(RI*-4)$ **Salida:** Valor de PF
- d) **Entrada:** RC,RI,RB,PF **Proceso:** $PF=(RC*4)+(RI*-1)$ **Salida:** Valor de PF

4. Se necesita obtener el puntaje total de un equipo, considerando el número de partidos ganados, número de partidos perdidos y empatados; se debe saber también que por cada partido ganado obtendrá 3 puntos, empatado 1 punto y perdido 0 puntos. ¿cuáles son sus datos de entrada, proceso y salida?

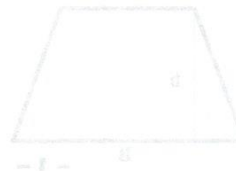
Considerar:

PT = Puntaje total

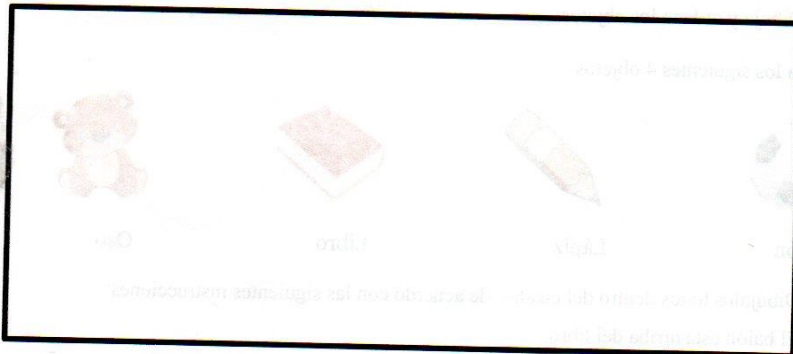
PG = Número de partidos ganados

PE = Número de partidos empatados

PP = Número de partidos perdidos



SOLUCIÓN:



5.

Si

$$\begin{aligned}
 \text{Compass} + \text{Compass} + \text{Compass} &= 60 \\
 \text{Compass} + \text{Eraser} + \text{Eraser} &= 30 \\
 \text{Eraser} - \text{Pencil} &= 3
 \end{aligned}$$

Poner verdadero (V) o falso (F) en los siguientes ejercicios:

$$1^\circ \quad \frac{\text{Compass}}{\text{Eraser}} = 4$$

$$2^\circ \quad \text{Pencil} \times \text{Eraser} = 8$$

$$3^\circ \quad \text{Pencil} \times \text{Compass} \times \text{Eraser} = 180$$

$$4^\circ \quad \text{Pencil} + \text{Compass} \times \text{Eraser} = 101$$

- a) VVFF
- b) VFVF
- c) FVFV
- d) **VFFV**

6. Dibuja y ordena los objetos.

Mira los siguientes 4 objetos



Balón



Lápiz



Libro



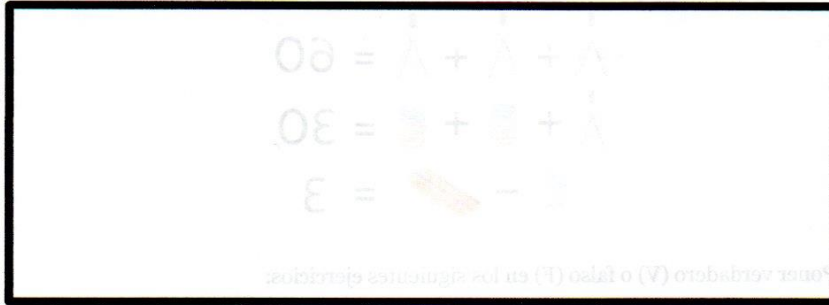
Oso

Dibújalos todos dentro del cuadro, de acuerdo con las siguientes instrucciones:

El balón está arriba del libro.

El libro está al lado del oso.

El balón está abajo del lápiz.

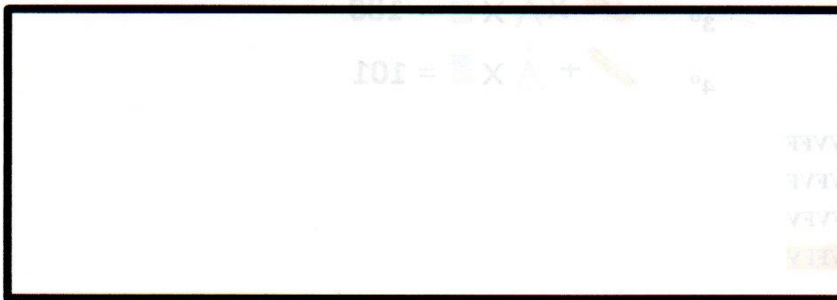


Dibújalos todos dentro del cuadro, de acuerdo con las siguientes instrucciones:

El oso está al lado del lápiz.

El libro está abajo del balón.

El balón está al lado del oso.



7. Enumera en orden lógico los pasos siguientes (para lavarse los dientes):

- Destapar la crema dental.
- Aplicar crema dental al cepillo.
- Remojar el cepillo con la crema dental.
- Tomar la crema dental.
- Tomar el cepillo de dientes.
- Abrir la llave del lavamanos.
- Enjuagar el cepillo.
- Enjuagarse la boca.
- Secarse la cara y las manos con una toalla.
- Cerrar la llave del lavamanos.
- Abrir la llave del lavamanos
- Tapar la crema dental
- Frotar los dientes con el cepillo.
- Cerrar la llave del lavamanos.

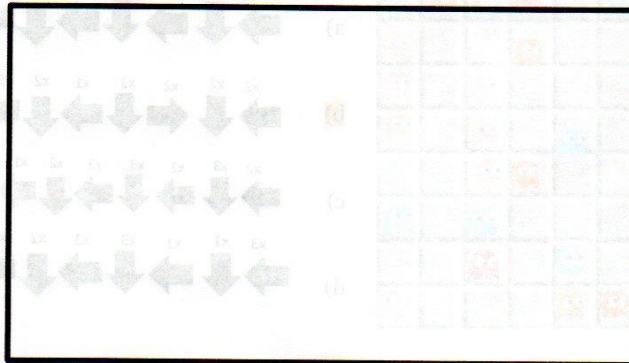
a) 8-2-7-1-3-10-12-11-14-4-6-5-9-13

b) 8-2-5-9-10-14-11-3-1-4-6-7-12-13

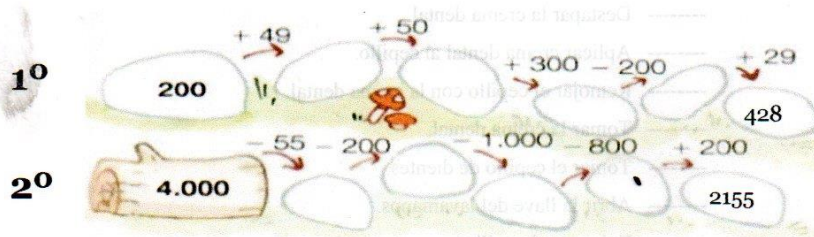
c) 2-4-7-1-3-6-12-11-14-8-10-5-9-13

d) 2-4-7-1-3-10-12-11-14-13-6-5-9-8

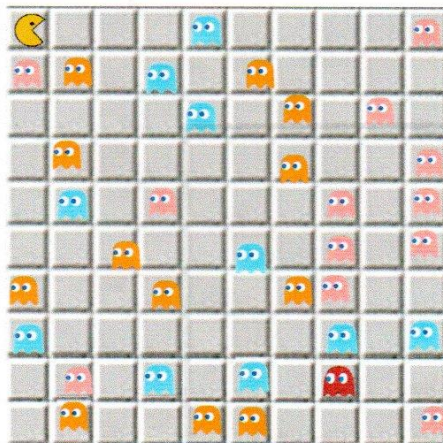
8. Cuáles son los pasos a seguir para lavarse las manos.



9. Completa el ejercicio y comenta si el resultado es verdadero o falso por qué?



10. Señala en el dibujo y marca la alternativa correcta para que Pacman llegue a comer al fantasma.



- a) \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow
- b) \rightarrow \downarrow \leftarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow
- c) \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow
- d) \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow

ANEXO 6: REGISTRO DE INCIDENCIAS



INSTITUCIÓN EDUCATIVA PRIVADA "AUGUSTO CARDICH"

Documentos Técnicos de Planificación

Periodo Académico 2019

FICHA DE REGISTRO DE INCIDENCIAS 2020

DOCENTE:.....

ÁREA DE DESARROLLO:



Fecha	Apellidos y Nombres	Motivo de Observación	Tratamiento/Derivación

ANEXO 7: SESIONES DE APRENDIZAJE

SESIÓN DE APRENDIZAJE N°21

I. TÍTULO DE LA SESIÓN: INTRODUCCIÓN AL SCRATCH

II. Unidad de Aprendizaje: IV UNIDAD

III. DATOS GENERALES1

Área : EPT/COMPUTACIÓN

Grado : 6To Grado Sección: A-B

Docente : Bach. RIVERA ILLATOPA Romer Ramiro.

Duración : 2 horas

IV. APRENDIZAJE ESPERADO:/DESEMPEÑOS DE LOS ESTUDIANTES:

COMPETENCIA	CAPACIDAD	DESEMPEÑO PRECISADO
se envuelve en los entornos digitales generados por las TIC (Tecnología de Información y Comunicación).	Analiza entornos virtuales	Utiliza herramientas de software y plataformas digitales cuando aprende diversas áreas del conocimiento de manera autorregulada y consciente. Por ejemplo: El estudiante accede a un portal educativo y utiliza los recursos digitales.

V. SECUENCIA DE LOS PROCESOS PEDAGÓGICOS: (Fases pedagógicas: Inicio, desarrollo, cierre)

INICIO: 10 minutos (motivación saberes previos, propósitos y organización, problematización)
El docente genera un ambiente grato para el aprendizaje (recepción-saluda) y menciona algunos detalles para tomar de mejor manera el examen pre test para los estudiantes.
DESARROLLO: 80 minutos (Desarrollo de los procesos didáctico del área de COMUNICACIÓN)
El profesor toma el examen de pre test que será de una duración de 50 minutos. Durante los 30 minutos después del examen se hace lo siguiente: Se explica sobre el lenguaje de programación SCRATCH. Se explica sobre las 5 zonas que tiene SCRATCH. Se explica sobre los fondos y objetos que tiene SCRATCH. Se explica cómo se descarga el programa SCRATCH.
CIERRE: 15 minutos (Valoración del aprendizaje)

<p>RE</p> <p>cente motiva la participación de los estudiantes y reflexionan sobre la importancia del tema tratado, elaboran sus resúmenes y llenado de la ficha que se dejó.</p> <p>El docente genera la reflexión con los estudiantes, les pregunta: aprendieron sobre SCRATCH?, ¿Cuántas zonas tiene el SCRATCH?, ¿Que es un SPRITE y fondo? ¿Cómo se descarga el SCRATCH?</p> <p>Extensión para la casa: Descargar el software SCRATCH para realizar los posteriores trabajos.</p>
--

VI. SITUACIONES DE EVALUACIÓN

DESEMPEÑO PRECISADO	Situación de evaluación	Técnicas e instrumentos
<p>la herramientas de software y plataformas digitales cuando aprende rsas áreas del conocimiento de manera autorregulada y consciente. Por plo: El estudiante accede a un portal educativo y utiliza los recursos ales.</p>	<p>en Pre test. untas sobre la parte introductoria CRATCH.</p>	

Huánuco, Agosto del 2019.

SESIÓN DE APRENDIZAJE N°22

TÍTULO DE LA SESIÓN: CREACIÓN DE OBJETOS Y HERRAMIENTAS EN SCRATCH.(FASE DE ANALIZA EL PROBLEMA)

VII. Unidad de Aprendizaje: IV UNIDAD

VIII. DATOS GENERALES1

Área : EPT/COMPUTACIÓN
 Grado : 6To Grado Sección: A-B
 Docente : Bach. RIVERA ILLATOPA Romer Ramiro.
 Duración : 2 horas

IX. APRENDIZAJE ESPERADO:/DESEMPEÑOS DE LOS ESTUDIANTES:

COMPETENCIA	CAPACIDAD	DESEMPEÑO PRECISADO
<p>Se desenvuelve en los entornos virtuales generados por las TIC (tecnología de Información y comunicación).</p>	<p>Crea objetos virtuales en diversos formatos.</p>	<p>Construye objetos virtuales a partir de información seleccionada de diversas fuentes y materiales digitales que respalden sus opiniones o posturas en los diversos trabajos que realiza. Ejemplo: El estudiante hace uso de un presentador visual.</p>

X. SECUENCIA DE LOS PROCESOS PEDAGÓGICOS: (Fases pedagógicas: Inicio, desarrollo, cierre)

INICIO: 20 minutos (motivación saberes previos, propósitos)
<ul style="list-style-type: none"> • El docente genera un ambiente grato para el aprendizaje (recepción-saluda) y realiza algunas preguntas sobre la clase pasada. <ul style="list-style-type: none"> ✓ El profesor muestra un bloque de lego y pregunta ¿qué es? Y para qué sirve? ✓ El profesor obtiene información de los alumnos a través de las preguntas. ✓ El profesor muestra una pieza de rompecabezas y pregunta ¿qué es? ¿para qué sirve? ✓ El profesor obtiene información de los alumnos a través de las preguntas. ✓ Se explica que el código de Scratch es como bloques que tienen que encajar e ir en orden y que si no hay orden no funcionará el programa. ✓ Se anuncia el tema de la sesión y lo que aprenderán. ✓ Se dice a los estudiantes que prendan su computadora y vayan al programa de SCRATCH.
DESARROLLO: 60 minutos (Desarrollo de los procesos didáctico del área de COMPUTACIÓN)
<ul style="list-style-type: none"> ✓ Se explica sobre las herramientas de Apariencia, Movimiento, Sonido, Eventos, Control, Sensores y Operadores que tiene SCRATCH. ✓ Se explica sobre los objetos, fondos y disfraces que ya hay en el sistema de SCRATCH. ✓ Se anuncia a los estudiantes que si se quiere cargar un nuevo objeto, fondo y disfraz se puede crear o descargar de internet. ✓ Se explica que para crear un objeto, fondo y disfraz, tenemos que abrir un Power point y ahí hacer el proceso de diseño (quitar fondo, recortar, iluminar, etc.), luego guardarlo como imagen para después cargarlo al programa de SCRATCH. – ESTE PROCESO DE DISEÑO LO VIMOS EN LA UNIDAD III. ✓ Se anuncia a los estudiantes que realicen un objeto y lo carguen al programa de SCRATCH. (15 minutos).
CIERRE: 25 minutos (Valoración del aprendizaje)
<p>CIERRE</p> <p>El profesor evalúa los objetos creados y puestos al programa.</p> <p>El profesor explica sobre cómo guardar en SCRATCH y anuncia que cierren el programa y apaguen la computadora.</p> <p>El profesor motiva la participación de los estudiantes y reflexionan sobre la importancia del tema tratado.</p> <ul style="list-style-type: none"> ✓ Extensión para la casa: Repasar sobre las herramientas del SCRATCH.

XI. SITUACIONES DE EVALUACIÓN

DESEMPEÑO PRECISADO	Situación de evaluación	Técnicas e instrumentos
Construye objetos virtuales a partir de información seleccionada de diversas fuentes y materiales digitales que respalden sus opiniones o posturas en los diversos trabajos que realiza. Ejemplo: El estudiante hace uso de un presentador visual.	Creación de un objeto para SCRATCH.	Ficha Observación. Registro de Incidencia.

SESIÓN DE APRENDIZAJE N°23

XII. TÍTULO DE LA SESIÓN: DESPLAZAMIENTO EN SCRATCH. (FASE DE DISEÑA Y TRADUCE EL ALGORITMO).

XIII. Unidad de Aprendizaje: IV UNIDAD

XIV. DATOS GENERALES I

Área : EPT/COMPUTACIÓN
 Grado : 6To Grado Sección: A-B
 Docente : Bach. RIVERA ILLATOPA Romer Ramiro.
 Duración : 2 horas

XV. APRENDIZAJE ESPERADO:/DESEMPEÑOS DE LOS ESTUDIANTES:

COMPETENCIA	CAPACIDAD	DESEMPEÑO PRECISADO
Se desenvuelve en los entornos virtuales generados por las TIC (tecnología de Información y comunicación).	<ul style="list-style-type: none"> • Gestiona información del entorno virtual • Interactúa en entornos virtuales. 	<ul style="list-style-type: none"> • Emplea portafolios digitales cuando organiza la información que obtuvo, de manera que esté disponible para actividades frecuentes. Ejemplo: El estudiante crea un blog para difundir las actividades de “El día del logro”. • Accede a entornos virtuales establecidos, mediante credenciales de identificación digital y considerando procedimientos seguros, éticos y responsables; por ejemplo, para ingresar a una red social.

XVI. SECUENCIA DE LOS PROCESOS PEDAGÓGICOS: (Fases pedagógicas: Inicio, desarrollo, cierre)

INICIO: 25 minutos (motivación saberes previos, propósitos)
<ul style="list-style-type: none"> • El docente genera un ambiente grato para el aprendizaje (recepción-saluda) y realiza algunas preguntas sobre la clase pasada. <ul style="list-style-type: none"> ✓ El profesor obtiene información de los alumnos a través de las preguntas realizadas sobre la clase pasada. ✓ El profesor muestra un gráfico de un plano cartesiano en la pizarra con el cañón multimedia y pregunta ¿conocen qué es un plano cartesiano? Qué representa un plano cartesiano? ✓ El profesor obtiene información de los alumnos a través de las preguntas. ✓ El profesor hace un ejemplo sobre un par ordenado dentro del plano cartesiano y pregunta ¿conocen las instrucciones para hallar un par ordenado? ¿qué es un algoritmo? ✓ El profesor obtiene información de los alumnos a través de las preguntas. ✓ Se anuncia el tema de la sesión y lo que aprenderán.
DESARROLLO: 60 minutos (Desarrollo de los procesos didáctico del área de COMPUTACIÓN)

<ul style="list-style-type: none"> ✓ Se explica sobre el plano cartesiano, cómo hallar un par ordenado, cómo secuencializar para hallar un par ordenado. ✓ Se hace otros ejemplos de la vida real para poder crear un orden lógico sobre una acción. ✓ Se explica sobre ¿qué es un algoritmo? ✓ Se da un ejercicio para que puedan resolver, ordenar lógicamente y representarlo gráficamente (15 minutos). ✓ El profesor evalúa los resultados del ejercicio. ✓ Se analiza el procedimiento de desplazamiento de un objeto en SCRATCH junto a los alumnos (pregunta respuesta). ✓ Se representa gráficamente los pasos a seguir. ✓ Se dice a los estudiantes que prendan su computadora y vayan al programa de SCRATCH. ✓ El profesor junto a los alumnos llevan la representación gráfica que se hizo y lo convierten en código bloque del SCRATCH.
<p>CIERRE: 20 minutos (Valoración del aprendizaje)</p>
<p>CIERRE</p> <p>El profesor evalúa el desplazamiento de un objeto en SCRATCH utilizando las teclas de dirección del teclado que se realizó.</p> <p>El profesor explica sobre los bucles en SCRATCH.</p> <p>El profesor anuncia que guarden, cierren el programa y apaguen la computadora.</p> <p>El profesor motiva la participación de los estudiantes y reflexionan sobre la importancia del tema tratado.</p> <p>✓ Extensión para la casa: Realizar una secuencia lógica sobre el cambio de ropa de una persona.</p>

XVII. SITUACIONES DE EVALUACIÓN

DESEMPEÑO PRECISADO	Situación de evaluación	Técnicas e instrumentos
<p>Emplea portafolios digitales cuando organiza la información que obtuvo, de manera que esté disponible para actividades frecuentes. Ejemplo: El estudiante crea un blog para difundir las actividades de “El día del logro”.</p> <p>Accede a entornos virtuales establecidos, mediante credenciales de identificación digital y considerando procedimientos seguros, éticos y responsables; por ejemplo, para ingresar a una red social.</p>	<ul style="list-style-type: none"> ✓ Ordenamiento y representación gráfica sobre despertarse. ✓ Desplazamiento de un objeto utilizando las teclas de dirección del teclado en SCRATCH. 	<p>Ficha Observación. Registro de Incidencia.</p>

SESIÓN DE APRENDIZAJE N°24

XVIII. TÍTULO DE LA SESIÓN: CAMBIO DE DISFRAZ Y POSICIÓN EN SCRATCH. (FASE DE DEPURAR EL PROGRAMA).

XIX. Unidad de Aprendizaje: IV UNIDAD

XX.DATOS GENERALES1

Área : EPT/COMPUTACIÓN
 Grado : 6To Grado Sección: A-B
 Docente : Bach. RIVERA ILLATOPA Romer Ramiro.
 Duración : 2 horas

XXI. APRENDIZAJE ESPERADO:/DESEMPEÑOS DE LOS ESTUDIANTES:

COMPETENCIA	CAPACIDAD	DESEMPEÑO PRECISADO
Se desenvuelve en los entornos virtuales generados por las TIC (tecnología de Información y comunicación).	<ul style="list-style-type: none"> • Gestiona información del entorno virtual • Crea objetos virtuales en diversos formatos 	<ul style="list-style-type: none"> • Emplea portafolios digitales cuando organiza la información que obtuvo, de manera que esté disponible para actividades frecuentes. Ejemplo: El estudiante crea un blog para difundir las actividades de "El día del logro". • Programa secuencias lógicas cuando simula procesos o comportamientos de acuerdo a la construcción de un diseño elaborado para presentar soluciones; por ejemplo, para mostrar una historieta interactiva.

XXII. SECUENCIA DE LOS PROCESOS PEDAGÓGICOS: (Fases pedagógicas: Inicio, desarrollo, cierre)

INICIO: 20 minutos (motivación saberes previos, propósitos)
<ul style="list-style-type: none"> • El docente genera un ambiente grato para el aprendizaje (recepción-saluda) y realiza algunas preguntas sobre la clase pasada. <ul style="list-style-type: none"> ✓ El profesor obtiene información de los alumnos a través de las preguntas realizadas sobre la clase pasada. ✓ El profesor recoge la tarea designada la clase pasada sobre realizar la secuencia lógica sobre el cambio de ropa de una persona. ✓ El profesor desarrolla la tarea designada con ayuda de los alumnos. ✓ El profesor anuncia el tema de la sesión y lo que aprenderán.
DESARROLLO: 65 minutos (Desarrollo de los procesos didáctico del área de COMPUTACIÓN)
<ul style="list-style-type: none"> ✓ El profesor explica el procedimiento de cambio de disfraz de un objeto. ✓ Se dice a los estudiantes que prendan su computadora, vayan al programa SCRATCH y carguen su trabajo guardado. ✓ El profesor deja 10 minutos para que los alumnos puedan realizar el procedimiento de cambio de disfraz. ✓ El profesor muestra el procedimiento de cambio de posición de un objeto aleatoriamente. ✓ El profesor deja 10 minutos para que puedan insertar otro objeto al programa SCRATCH y los alumnos puedan realizar el procedimiento de cambio posición de dicho objeto.

<ul style="list-style-type: none"> ✓ El profesor explica sobre ¿cómo crear una variable en Scratch? ¿cómo utilizar los sensores? Y su uso de éstos en SCRATCH. ✓ El profesor realiza el procedimiento de inserción de la variable y sensor al SCRATCH junto a los estudiantes. ✓ El profesor deja 10 minutos para que los alumnos puedan realizar el procedimiento de inserción de la variable y sensor al programa SCRATCH.
CIERRE: 20 minutos (Valoración del aprendizaje)
<p>CIERRE</p> <p>El profesor deja que los alumnos prueben los resultados esperados jugando el programa que hicieron.</p> <p>El profesor evalúa el 1er juego elaborado por los alumnos.</p> <p>El profesor deja que los alumnos que realizaron bien el trabajo ayuden a corregir a sus compañeros que no lo hicieron bien (cooperación y retroalimentación).</p> <p>El profesor anuncia que guarden, cierren el programa y apaguen la computadora.</p> <p>✓ Extensión para la casa: Realizar una secuencia lógica de cambio de nivel.</p>

XXIII. SITUACIONES DE EVALUACIÓN

DESEMPEÑO PRECISADO	Situación de evaluación	Técnicas e instrumentos
<ul style="list-style-type: none"> • Emplea portafolios digitales cuando organiza la información que obtuvo, de manera que esté disponible para actividades frecuentes. Ejemplo: El estudiante crea un blog para difundir las actividades de “El día del logro”. • Programa secuencias lógicas cuando simula procesos o comportamientos de acuerdo a la construcción de un diseño elaborado para presentar soluciones; por ejemplo, para mostrar una historieta interactiva. 	<ul style="list-style-type: none"> ✓ Secuencia lógica sobre el cambio de ropa de una persona. ✓ 1er JUEGO EN SCRATCH (Desplazamiento de un objeto, Cambio de disfraz de un objeto, Cambio de posición de un objeto e Inserción de variable y sensor en el programa SCRATCH. 	<p>Ficha Observación. Registro de Incidencia.</p>

SESIÓN DE APRENDIZAJE N°25

XXIV. TÍTULO DE LA SESIÓN: EXAMEN MENSUAL

XXV. Unidad de Aprendizaje: IV UNIDAD

XXVI. DATOS GENERALES

Área : EPT/COMPUTACIÓN
Grado : 6To Grado Sección: A-B
Docente : Bach. RIVERA ILLATOPA Romer Ramiro.
Duración : 2 horas

XXVII. APRENDIZAJE ESPERADO:/DESEMPEÑOS DE LOS ESTUDIANTES:

COMPETENCIA	CAPACIDAD	DESEMPEÑO PRECISADO
Se desenvuelve en los entornos virtuales generados por las TIC (tecnología de Información y comunicación).	Personaliza entornos virtuales	Utiliza herramientas de software y plataformas digitales cuando aprende diversas áreas del conocimiento de manera autorregulada y consciente. Por ejemplo: El estudiante accede a un portal educativo y utiliza los recursos digitales.

XXVIII. SECUENCIA DE LOS PROCESOS PEDAGÓGICOS: (Fases pedagógicas: Inicio, desarrollo, cierre)

INICIO: 10 minutos (motivación saberes previos, propósitos y organización, problematización)
<ul style="list-style-type: none">• El docente genera un ambiente grato para el aprendizaje (recepción-saluda) y menciona algunos detalles para tomar de mejor manera el examen post test para los alumnos.
DESARROLLO: 80 minutos (Desarrollo de los procesos didáctico del área de COMUNICACIÓN)
<ul style="list-style-type: none">❖ El profesor toma el examen de pre test que será de una duración de 50 minutos.❖ Durante los 30 minutos después del examen se hace lo siguiente:<ul style="list-style-type: none">✓ El profesor explica sobre la importancia del lenguaje de programación SCRATCH y sugiere a los alumnos ser parte de la comunidad SCRATCH que se tiene en internet.✓ El profesor explica la influencia que tiene el lenguaje de programación SCRATCH para desarrollar el pensamiento algorítmico.✓ El profesor explica que el SCRATCH no sólo sirve para crear juegos y cuentos sino también que indirectamente ayuda a su vida diaria.
CIERRE: 15 minutos (Valoración del aprendizaje)

CIERRE

El docente motiva la participación de los estudiantes para reflexionar sobre el SCRATCH.

- ✓ El docente genera la **reflexión** con los estudiantes, les pregunta:
¿Qué aprendieron sobre SCRATCH?, ¿Les gusta el SCRATCH? ¿Seguirán practicándolo? ¿Quieren aprender más sobre scratch?

XXIX. SITUACIONES DE EVALUACIÓN

DESEMPEÑO PRECISADO	Situación de evaluación	Técnicas e instrumentos
Utiliza herramientas de software y plataformas digitales cuando aprende diversas áreas del conocimiento de manera autorregulada y consciente. Por ejemplo: El estudiante accede a un portal educativo y utiliza los recursos digitales.	Examen Post test. Preguntas sobre el SCRATCH.	Examen de conocimiento. Registro de incidencias.

Huánuco, Agosto del 2019.

ANEXO 9: MATRIZ DE CONSISTENCIA

MATRIZ DE CONSISTENCIA PARA PROYECTOS DE INVESTIGACION							
TESISTA: Bach. ROMER RAMIRO RIVERA ILLATOPA			MATRIZ OPERACIONAL DE LAS VARIABLES				
E.A.P DE INGENIERÍA DE SISTEMAS							
PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLES	DIMENSIONES	INDICADORES-ACTIVIDADES	METODOLOGÍA	
GENERAL	GENERAL	GENERAL	VARIABLE INDEPENDIENTE: LENGUAJE DE PROGRAMACIÓN SCRATCH	PLANIFICACIÓN	Elaboración de las sesiones de aprendizaje aplicando el SCRATCH a las dimensiones de la VD.	NIVEL DE INVESTIGACIÓN	
¿En que medida la aplicación del lenguaje de programación SCRATCH influye en el desarrollo del pensamiento algorítmico en los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019?	Determinar de qué manera la aplicación del lenguaje de programación SCRATCH influye en el pensamiento algorítmico de los alumnos del 6to grado de primaria del colegio Augusto Cardich-Huánuco 2019	La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.		EJECUCIÓN	Desarrollo de sesiones de aprendizaje aplicando el SCRATCH a las dimensiones de la VD.	TIPO DE INVESTIGACIÓN	Experimental
					Desarrollo de un juego interactivo	DISEÑO DE LA INVESTIGACIÓN	Aplicada
ESPECÍFICOS	ESPECÍFICOS	ESPECÍFICOS		VARIABLE DEPENDIENTE: PENSAMIENTO ALGORITMICO	ANALIZA EL PROBLEMA	Comprende el problema. (escala DE 0 A 2)	POBLACIÓN
¿En qué medida la aplicación del lenguaje de programación SCRATCH influye en la fase de <i>análisis del problema</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019?	Determinar de qué manera la aplicación del lenguaje de programación SCRATCH influye en la <i>fase de análisis del problema</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.	La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo de la <i>fase de análisis del problema</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.	Define los datos disponibles. (escala DE 0 A 2)			440 Estudiantes del nivel primario y secundario de la I.E.P Augusto Cardich.	
¿En qué medida la aplicación del lenguaje de programación SCRATCH influye en la <i>fase de diseño y traducción del algoritmo</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019?	Determinar de qué manera la aplicación del lenguaje de programación SCRATCH influye en la <i>fase de diseño y traducción del algoritmo</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.	La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo de la <i>fase de diseño y traducción del algoritmo</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.	DISEÑA Y TRADUCE EL ALGORITMO		Define los procesos del problema. (escala DE 0 A 2)	MUESTRA	No probabilística intencionada. Grupo de control (19 alumnos 6to grado A) y Grupo Experimental (23 alumnos 6to)
					Representa gráficamente el problema. (escala DE 0 A 2)	TÉCNICAS DE RECOLECCIÓN DE DATOS	Observación participante
¿En qué medida la aplicación del lenguaje de programación SCRATCH influye en la <i>fase de depuración del programa</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019?	Determinar de qué manera la aplicación del lenguaje de programación SCRATCH influye en la <i>fase de depuración del programa</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.	La aplicación del lenguaje de programación SCRATCH influye positivamente en el desarrollo de la <i>fase de depuración del programa</i> del pensamiento algorítmico de los estudiantes del 6to grado del nivel primario de la I.E.P Augusto Cardich-Huánuco 2019.	DEPURA EL PROGRAMA (validación de resultados)		Ordena lógicamente las instrucciones a seguir. (escala DE 0 A 2)	MEDIOS DE INVESTIGACIÓN	La encuesta.
					Verifica la respuesta del problema	INSTRUMENTO DE INVESTIGACIÓN	Celular
					Propone otras alternativas de solución. (escala DE 0 A 2)	Pruebas de conocimiento (pre test y pos test).	
					Corrige fallas. (escala DE 0 A 2)	Cuestionario y registro de incidencia.	

ANEXO 10: PRUEBA DE CONOCIMIENTO PRETEST-EJEMPLO1

INSTITUCIÓN EDUCATIVA PRIVADA
AUGUSTO CARDICH

PRUEBA DE CONOCIMIENTO

2019 *Un colegio diferente para niños y jóvenes diferentes....!*

6TO GRADO 6TO 1°B 05

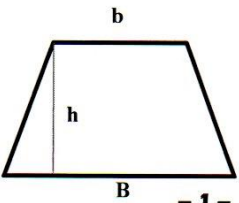
Serrano Ingoroco, Nayeli

1. Juan Felipe es jefe de bodega en una fábrica de pañales desechables y una de las tareas del día consiste en llamar al proveedor de los empaques y ordenarle la cantidad suficiente de cajas para empaquetar los pañales fabricados en la semana próxima. El jefe de producción le informó ayer a Juan Felipe que la producción diaria será de 744 pañales y en cada caja cabe una docena de ellos. ¿Qué debe hacer Felipe?

En el siguiente problema plantea y resuelve el problema.

X

2. El perímetro de un trapecio es igual a la suma de todos sus lados y su área es igual a la suma de la base mayor (B) más la base menor (b) dividido todo entre 2 para después multiplicar este resultado por la altura (h). Considerando el trapecio de la siguiente figura y de acuerdo al Modelo de solución, identifique los elementos de entrada, proceso y salida necesarios para resolver el problema del cálculo del área del Trapecio.



- 1 -

INSTITUCIÓN EDUCATIVA PRIVADA
Augusto Cardich

Un colegio diferente para niños y jóvenes diferentes....!

- a) **Entrada:** B,b,h **Proceso:** $A=b+B/2*h$ **Salida:** Valor de A
 b) **Entrada:** B,b,h **Proceso:** $A=b+(B/2)*h$ **Salida:** Valor de A
 c) **Entrada:** B,b,h **Proceso:** $A=(b+B)/2*h$ **Salida:** Valor de A
 d) **Entrada:** B,b,h **Proceso:** $A=b+B/(2*h)$ **Salida:** Valor de A

3. Se necesita obtener el puntaje final de un estudiante en un examen considerando el número de respuestas correctas, incorrectas y en blancos, si se sabe que por cada respuesta correcta tendrá 4 puntos, respuestas incorrectas tendrá -1 y respuestas en blanco tendrá 0. ¿cuáles son sus datos de entrada, proceso y salida?

RC = Número de respuestas correctas.

RI = Número de respuestas incorrectas.

RB = Número de respuestas en blanco

PF = PUNTAJE FINAL

- a) **Entrada:** RC,RI,RB,PF **Proceso:** $RB=(4*RC)+(-1*RI)+PF$ **Salida:** Valor de RB
 b) **Entrada:** RC,RI,RB **Proceso:** $PF=(RC*4)+(RI*-1)$ **Salida:** Valor de PF
 c) **Entrada:** RC,RI,RB **Proceso:** $PF=(RC*-1)+(RI*-4)$ **Salida:** Valor de PF
 d) **Entrada:** RC,RI,RB,PF **Proceso:** $PF=(RC*4)+(RI*-1)$ **Salida:** Valor de PF

4. Se necesita obtener el puntaje total de un equipo, considerando el número de partidos ganados, número de partidos perdidos y empatados; se debe saber también que por cada partido ganado obtendrá 3 puntos, empatado 1 punto y perdido 0 puntos. ¿cuáles son sus datos de entrada, proceso y salida?

Considerar:

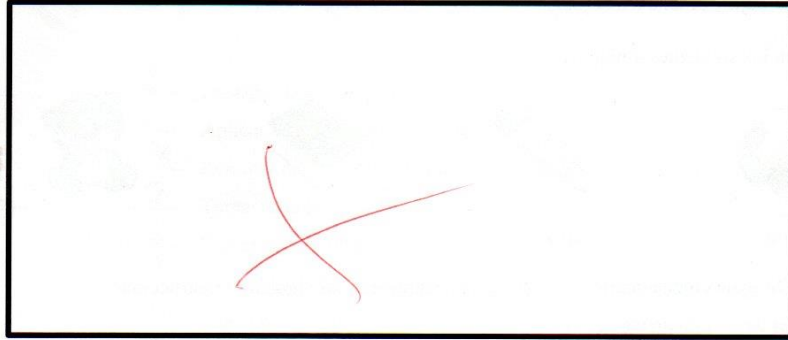
PT = Puntaje total

PG = Número de partidos ganados

PE = Número de partidos empatados

PP = Número de partidos perdidos

SOLUCIÓN:



5.

Si

$$\begin{aligned}
 & \text{Compass} + \text{Compass} + \text{Compass} = 60 \\
 & \text{Compass} + \text{Eraser} + \text{Eraser} = 30 \\
 & \text{Eraser} - \text{Ruler} = 3
 \end{aligned}$$

Poner verdadero (V) o falso (F) en los siguientes ejercicios:

$$\begin{aligned}
 1^\circ & \quad \frac{\text{Compass}}{\text{Eraser}} = 4 \\
 2^\circ & \quad \text{Ruler} \times \text{Eraser} = 8 \\
 3^\circ & \quad \text{Ruler} \times \text{Compass} \times \text{Eraser} = 180 \\
 4^\circ & \quad \text{Ruler} + \text{Compass} \times \text{Eraser} = 101
 \end{aligned}$$

- a) VVFF
- b) VFVF
- c) FVVF
- d) VFFV

6. Dibuja y ordena los objetos.

Mira los siguientes 4 objetos



Balón



Lápiz



Libro



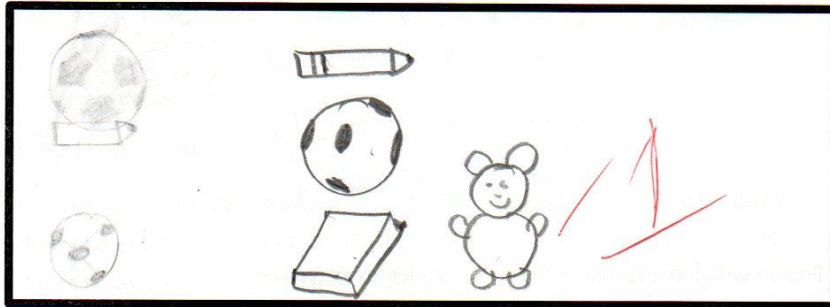
Oso

Dibújalos todos dentro del cuadro, de acuerdo con las siguientes instrucciones:

El balón está arriba del libro.

El libro está al lado del oso.

El balón está abajo del lápiz.

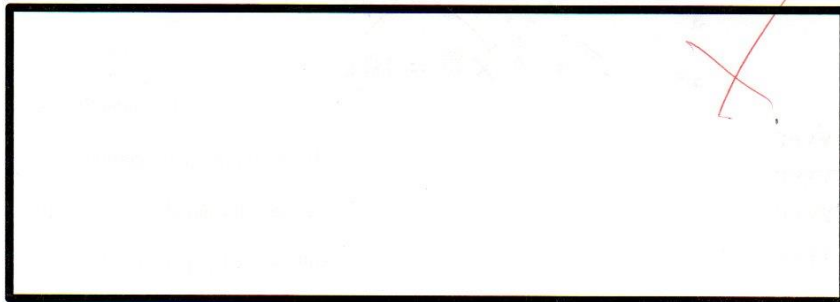


Dibújalos todos dentro del cuadro, de acuerdo con las siguientes instrucciones:

El oso está al lado del lápiz.

El libro está abajo del balón.

El balón está al lado del oso.

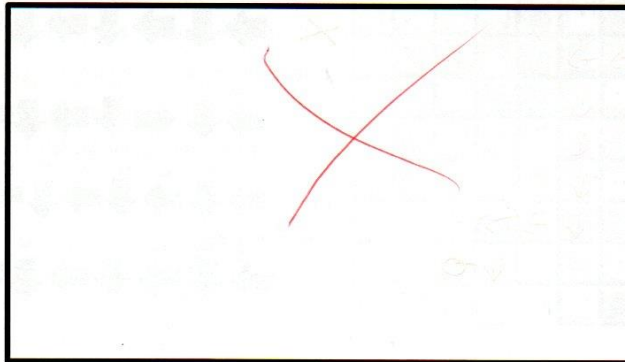


7. Enumera en orden lógico los pasos siguientes (para lavarse los dientes):

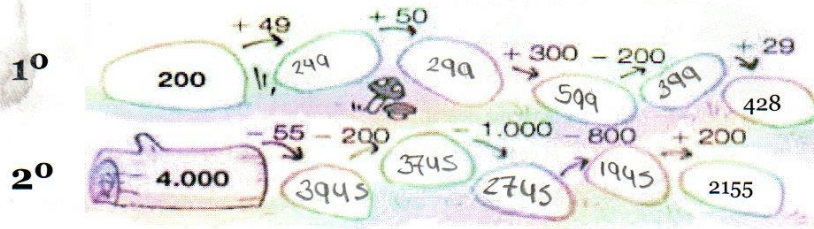
- 3 Destapar la crema dental.
- 4 Aplicar crema dental al cepillo.
- 7 Remojar el cepillo con la crema dental.
- 2 Tomar la crema dental.
- 1 Tomar el cepillo de dientes.
- 6 Abrir la llave del lavamanos.
- 11 Enjuagar el cepillo.
- 12 Enjuagarse la boca.
- 14 Secarse la cara y las manos con una toalla.
- 13 Cerrar la llave del lavamanos.
- 10 Abrir la llave del lavamanos
- 5 Tapar la crema dental
- 9 Frotar los dientes con el cepillo.
- 8 Cerrar la llave del lavamanos.

- a) 8-2-7-1-3-10-12-11-14-4-6-5-9-13
- b) 8-2-5-9-10-14-11-3-1-4-6-7-12-13
- c) 2-4-7-1-3-6-12-11-14-8-10-5-9-13
- d) 2-4-7-1-3-10-12-11-14-13-6-5-9-8

8. Cuáles son los pasos a seguir para lavarse las manos.

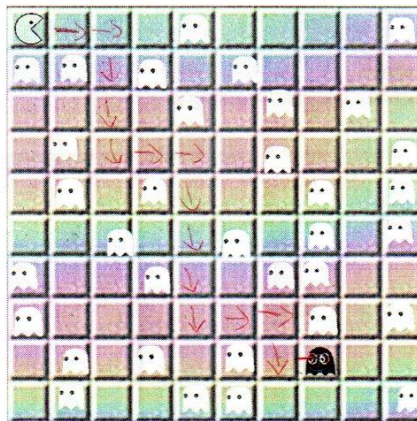


9. Completa el ejercicio y comenta si el resultado es verdadero o falso por qué?




1^o = V
 2^o = F

10. Señala en el dibujo y marca la alternativa correcta para que Pacman llegue a comer al fantasma.



- a) $\rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow \rightarrow$
- b) $\rightarrow \downarrow \leftarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow$
- c) $\rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow$
- d) $\rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow$

ANEXO 11: PRUEBA DE CONOCIMIENTO POSTEST-EJEMPLO 1



INSTITUCIÓN EDUCATIVA PRIVADA
AUGUSTO CARDICH

PRUEBA DE CONOCIMIENTO

2019 *Un colegio diferente, para niños y jóvenes diferentes...!*

6TO GRADO

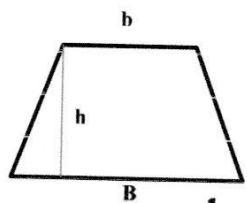
Serrano Ingaroca, Nayeli 14 6^{to} "B"

1. Juan Felipe es jefe de bodega en una fábrica de pañales desechables y una de las tareas del día consiste en llamar al proveedor de los empaques y ordenarle la cantidad suficiente de cajas para empacar los pañales fabricados en la semana próxima. El jefe de producción le informó ayer a Juan Felipe que la producción diaria será de 744 pañales y en cada caja cabe una docena de ellos. ¿Qué debe hacer Felipe?

En el siguiente problema plantea y resuelve el problema.

$$\begin{array}{r}
 744 \text{ } | 12 \\
 \underline{72} \quad 62 \\
 \quad 24 \\
 \quad \underline{24} \\
 \quad \quad \underline{\quad} \\
 \text{Rpt: } 62
 \end{array}$$

2. El perímetro de un trapecio es igual a la suma de todos sus lados y su área es igual a la suma de la base mayor (B) más la base menor (b) dividido todo entre 2 para después multiplicar este resultado por la altura (h). Considerando el trapecio de la siguiente figura y de acuerdo al Modelo de solución, identifique los elementos de entrada, proceso y salida necesarios para resolver el problema del cálculo del área del Trapecio.



- 1 -

INSTITUCIÓN EDUCATIVA PRIVADA
Augusto Cardich

Un colegio diferente, para niños y jóvenes diferentes...!

- a) **Entrada:** B,b,h **Proceso:** $A=b+B/2*h$ **Salida:** Valor de A
 b) **Entrada:** B,b,h **Proceso:** $A=b+(B/2)*h$ **Salida:** Valor de A
 c) **Entrada:** B,b,h **Proceso:** $A=(b+B)/2*h$ **Salida:** Valor de A
 d) **Entrada:** B,b,h **Proceso:** $A=b+B/(2*h)$ **Salida:** Valor de A

3. Se necesita obtener el puntaje final de un estudiante en un examen considerando el número de respuestas correctas, incorrectas y en blancos, si se sabe que por cada respuesta correcta tendrá 4 puntos, respuestas incorrectas tendrá -1 y respuestas en blanco tendrá 0. ¿cuáles son sus datos de entrada, proceso y salida?

RC = Número de respuestas correctas.

RI = Número de respuestas incorrectas.

RB = Número de respuestas en blanco

PF = PUNTAJE FINAL

- a) **Entrada:** RC,RI,RB,PF **Proceso:** $RB=(4*RC)+(-1*RI)+PF$ **Salida:** Valor de RB
 b) **Entrada:** RC,RI,RB **Proceso:** $PF=(RC*4)+(RI*-1)$ **Salida:** Valor de PF
 c) **Entrada:** RC,RI,RB **Proceso:** $PF=(RC*-1)+(RI*-4)$ **Salida:** Valor de PF
 d) **Entrada:** RC,RI,RB,PF **Proceso:** $PF=(RC*4)+(RI*-1)$ **Salida:** Valor de PF

4. Se necesita obtener el puntaje total de un equipo, considerando el número de partidos ganados, número de partidos perdidos y empatados; se debe saber también que por cada partido ganado obtendrá 3 puntos, empatado 1 punto y perdido 0 puntos. ¿cuáles son sus datos de entrada, proceso y salida?

Considerar:

PT = Puntaje total

PG = Número de partidos ganados

PE = Número de partidos empatados




PP = Número de partidos perdidos

SOLUCIÓN:

Entrada = PG; PE; PP; ~~ED~~
 Proceso = PP
 Salida =

5.

Si

 = 20
 = 5
 = ?

$$\begin{aligned}
 \text{Scissors} + \text{Scissors} + \text{Scissors} &= 60 \\
 \text{Scissors} + \text{Eraser} + \text{Eraser} &= 30 \\
 \text{Eraser} - \text{Ruler} &= 3
 \end{aligned}$$

Poner verdadero (V) o falso (F) en los siguientes ejercicios:

- 1º $\frac{\text{Scissors}}{\text{Eraser}} = 4$ V
- 2º $\text{Ruler} \times \text{Eraser} = 8$ F
- 3º $\text{Ruler} \times \text{Scissors} \times \text{Eraser} = 180$ F
- 4º $\text{Ruler} + \text{Scissors} \times \text{Eraser} = 101$ V

- a) VVFF
- b) VFVF
- c) FVFV
- ~~d) VFFV~~

6. Dibuja y ordena los objetos.

Mira los siguientes 4 objetos



Balón



Lápiz



Libro



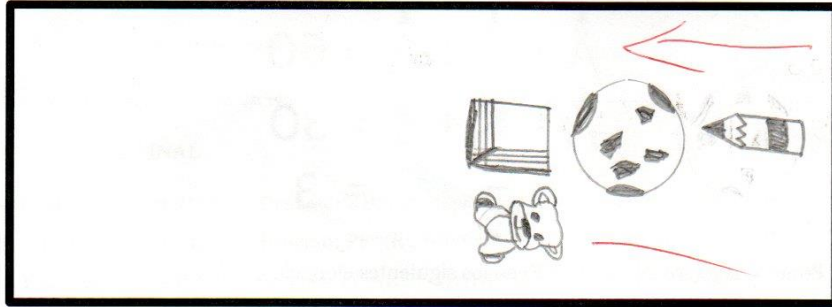
Oso

Dibújalos todos dentro del cuadro, de acuerdo con las siguientes instrucciones:

El balón está arriba del libro.

El libro está al lado del oso.

El balón está abajo del lápiz.

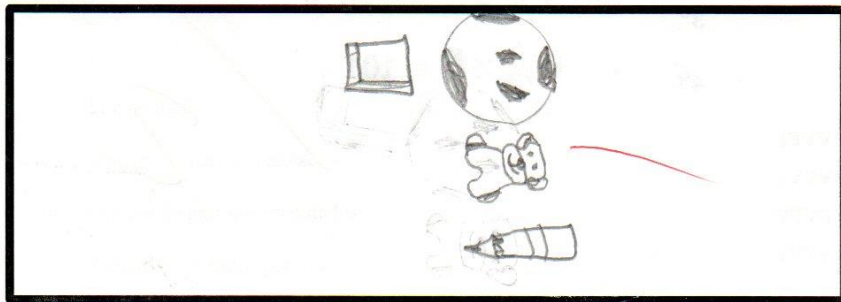


Dibújalos todos dentro del cuadro, de acuerdo con las siguientes instrucciones:

El oso está al lado del lápiz.

El libro está abajo del balón.

El balón está al lado del oso.



7. Enumera en orden lógico los pasos siguientes (para lavarse los dientes):

- Destapar la crema dental.
- Aplicar crema dental al cepillo.
- Remojar el cepillo con la crema dental.
- Tomar la crema dental.
- Tomar el cepillo de dientes.
- Abrir la llave del lavamanos.
- Enjuagar el cepillo.
- Enjuagarse la boca.
- Secarse la cara y las manos con una toalla.
- Cerrar la llave del lavamanos.
- Abrir la llave del lavamanos
- Tapar la crema dental
- Frotar los dientes con el cepillo.
- Cerrar la llave del lavamanos.

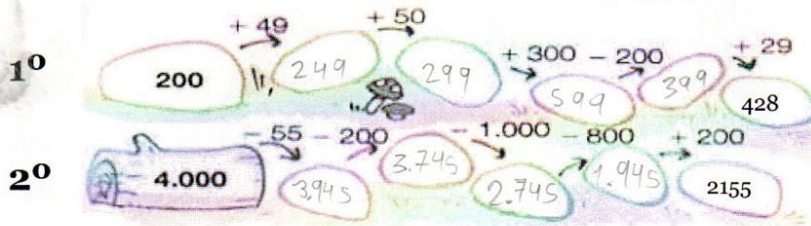
- a) 8-2-7-1-3-10-12-11-14-4-6-5-9-13
- b) 8-2-5-9-10-14-11-3-1-4-6-7-12-13
- c) 2-4-7-1-3-6-12-11-14-8-10-5-9-13
- d) 2-4-7-1-3-10-12-11-14-13-6-5-9-8

12

8. Cuáles son los pasos a seguir para lavarse las manos.

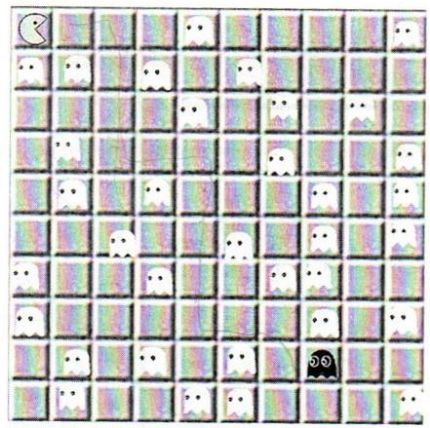
1. ir al Baño
2.

9. Completa el ejercicio y comenta si el resultado es verdadero o falso por qué?



1º	200	+49	249	+50	299	+300	599	-200	399	+29	428	V
2º	4.000	-55	3.945	-200	3.745	-1.000	2.745	-800	1.945	+200	2.145	F

10. Señala en el dibujo y marca la alternativa correcta para que Pacman llegue a comer al fantasma.



- a) $\rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow$
- b) $\rightarrow \downarrow \leftarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow$
- c) $\rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow$
- d) $\rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \downarrow \rightarrow \rightarrow$

ANEXO 12: FOTOGRAFÍAS

AULA DE CONTROL SEXTO GRADO A





AULA EXPERIMENTAL SEXTO GRADO B



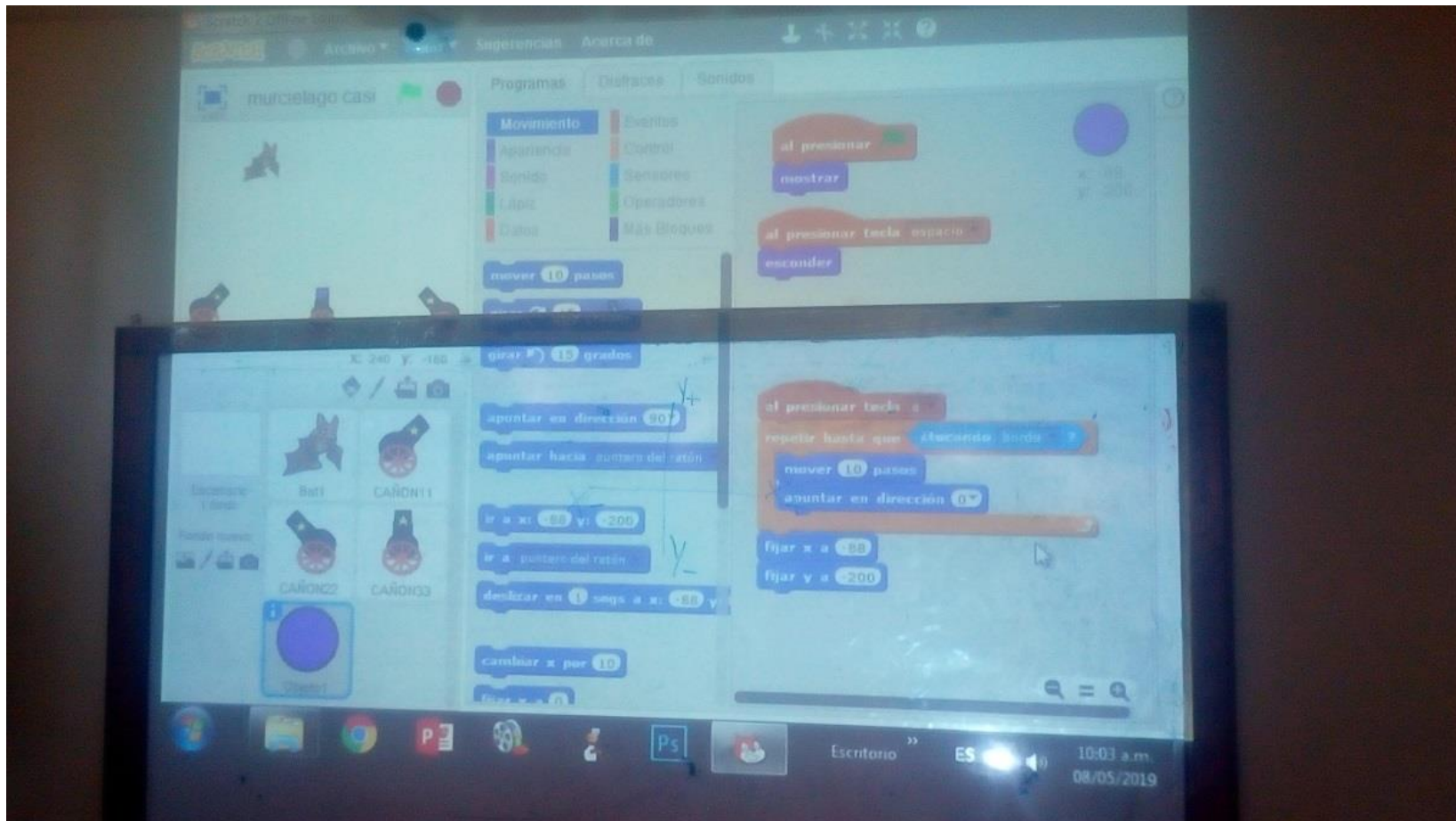













DRAGON



Objetos: Bat, Ballerina, Boy3 Wa..., Dragon

Programas: Disfraces, Sonidos

Movimiento

- mover 10 pasos
- girar 15 grados
- girar 15 grados
- apuntar en dirección 90
- apuntar hacia
- ir a x: -60 y: 126
- ir a puntero del ratón
- deslizar en 1 segs a x: -60 y: 126
- cambiar x por 10
- fijar x a 0
- cambiar y por 10
- fijar y a 0

al presionar

por siempre

- siguiente disfraz
- mover 10 pasos
- esperar 0.1 segundos
- si posición x > 200 entonces
- fijar x a -240