



ESCUELA UNIVERSITARIA DE POSGRADO

“MODELO HOLÍSTICO DE CÓDIGO-ALFABETIZACIÓN EN EL
DESARROLLO DEL PENSAMIENTO COMPUTACIONAL EN
EDUCACIÓN PRIMARIA”.

TESIS PARA OPTAR EL GRADO ACADÉMICO DE:
DOCTOR EN INGENIERÍA DE SISTEMAS

AUTOR:

EDELFRÉ FLORES VELÁSQUEZ

ASESOR:

DRA. ÁNGELES LAZO, ANA MARÍA

JURADO:

DR. RODRÍGUEZ RODRÍGUEZ, CIRO

DRA. VILLAVICENCIO RAMIREZ, ILSE JANINE

DR. MAYHUSCA GUERRA, JORGE VÍCTOR

Lima – Perú

2019

ÍNDICE

	Pág.
RESUMEN	7
ABSTRACT	8
CAPÍTULO I	9
INTRODUCCIÓN	9
1.1 PLANTEAMIENTO DEL PROBLEMA	11
1.2 DESCRIPCIÓN DEL PROBLEMA	12
1.3 FORMULACIÓN DEL PROBLEMA.....	13
1.3.1 Problema general	13
1.3.2 Problemas específicos	13
1.4 ANTECEDENTES.....	14
1.5 JUSTIFICACIÓN DE LA INVESTIGACIÓN	19
1.6 LIMITACIONES DE LA INVESTIGACIÓN.....	21
1.7 OBJETIVOS.....	21
1.7.1 Objetivo General	21
1.7.2 Objetivos Específicos	21
1.8 HIPÓTESIS	22
1.8.1 Hipótesis General	22
1.8.2 Hipótesis específicas.....	22
CAPÍTULO II	23
MARCO TEÓRICO	23
2.1 TEORÍAS GENERALES.....	23
2.1.1 Cibernética de 2do orden	23
2.1.2 Autopoiesis	24
2.1.3 Modelos Educativos.....	25
2.1.4 Holismo y educación.....	29
2.2 BASES TEÓRICAS	31
2.2.1 MODELOS HOLÍSTICOS	31
2.2.2 EL HOLISMO.....	33
2.2.3 CODIGO-ALFABETIZACIÓN.....	35
2.2.4 LA HORA DEL CÓDIGO, EVENTO MUNDIAL.....	38
2.2.5 PENSAMIENTO COMPUTACIONAL	40
2.3 MARCO CONCEPTUAL.....	48
CAPÍTULO III	52
MÉTODO	52

3.1	TIPO DE INVESTIGACIÓN	52
3.2	POBLACIÓN Y MUESTRA	53
3.3	OPERACIONALIZACIÓN DE VARIABLES	54
3.4	INSTRUMENTOS.....	56
3.5	PROCEDIMIENTOS.....	57
3.6	ANÁLISIS DE DATOS.....	57
CAPÍTULO IV		59
RESULTADOS.....		59
4.1	MODELO HOLÍSTICO DE CÓDIGO-ALFABETIZACIÓN PARA NIÑOS DE EDUCACIÓN PRIMARIA.....	59
4.2	PROTOTIPO DE SOFTWARE BASADO EN AL MODELO HOLÍSTICO DE CÓDIGO-ALFABETIZACIÓN.....	68
4.3	EVALUACIÓN DEL NIVEL DE DESARROLLO DEL PENSAMIENTO COMPUTACIONAL	87
CAPÍTULO V.....		97
DISCUSIÓN DE RESULTADOS		97
5.1	DISCUSIÓN	97
CAPÍTULO VI		100
CONCLUSIONES.....		100
CAPÍTULO VII		101
RECOMENDACIONES		101
REFERENCIAS BIBLIOGRÁFICAS		102
ANEXOS.....		108

ÍNDICE DE FIGURAS

Figura 1:	Comunicación y control cibernético	23
Figura 2:	Enfoques constructivistas de la educación	26
Figura 3:	Triángulo instruccional	27
Figura 4:	Modelo holístico del aprendizaje humano	30
Figura 5:	Modelo holístico de aprendizaje – Mendoza Vera, Eugenio	31
Figura 6:	Los planos constitutivos del ser humano	34
Figura 7:	Alfabetización digital, Alfabetización web y Código-alfabetización	36
Figura 8:	Marco-modelo de pensamiento computacional en el aula propuesto desde ‘CAS Barefoot’	43
Figura 9:	Representación de un sistema	60
Figura 10:	Modelo holista de Código-alfabetización propuesto	64
Figura 11:	Tablero Kanban en la Herramienta Kanbanize	73
Figura 12:	Tablero Kanban con procesos iniciales	74
Figura 13:	Tablero Kanban del flujo de trabajo 01	75
Figura 14:	Interfaz del portal web principal	76
Figura 15:	Diseño de la Base de datos	77
Figura 16:	Tablero Kanban del flujo de trabajo 02	78
Figura 17:	Interfaz de inicio de sesión del proyecto	79
Figura 18:	Interfaz de registro de usuario del proyecto	79
Figura 19:	Interfaz de trabajo en grupo del proyecto	80
Figura 20:	Tablero Kanban del flujo de trabajo 03	81
Figura 21:	Interfaz de actividades interactivas con el entorno A	82
Figura 22:	Interfaz de actividades interactivas con el entorno B	83
Figura 23:	Tablero Kanban del flujo de trabajo 04	84

Figura 24:	Tablero Kanban del flujo de trabajo 05 de Programadores Kid's	85
Figura 25:	Interfaz de desarrollo del examen de Programadores Kid's	86
Figura 26:	Ejemplo de pregunta del Pre-test de Programadores Kid's	88
Figura 27:	Distribución de calificaciones obtenidos en el Pre-test por los estudiantes de la I.E.P. "71013 Glorioso San Carlos" - Puno	89
Figura 28:	Distribución de calificaciones obtenidos en el Post-test por los estudiantes de la I.E.P. "71013 Glorioso San Carlos" - Puno	93

ÍNDICE DE TABLAS

Tabla 1:	Criterios que aplica Dr. Scratch para medir el nivel de competencia en las distintas dimensiones del PC	46
Tabla 2:	Distribución de Estudiantes de la Institución Educativa 71013 Glorioso San Carlos – Puno 2018	53
Tabla 3:	Operacionalización de las variables	54
Tabla 4:	HS1, Portal web principal	75
Tabla 5:	HS2, Base de datos del proyecto	76
Tabla 6:	HS3, Logueo de usuario	78
Tabla 7:	HS4, Actividades grupales en aula	80
Tabla 8:	HS5, Actividades interactivas con el entorno	82
Tabla 9:	HS6, Actividades de construcción creativa	84
Tabla 10:	HS7, Desarrollo del examen de módulo	86
Tabla 11:	HS8, Elaboración de reportes	87
Tabla 12:	Distribución de calificativos obtenidos en el Pre-test por los estudiantes de la I.E.P. “71013 Glorioso San Carlos” - Puno	88
Tabla 13:	Distribución de calificativos obtenidos en el Post-test por los estudiantes de la I.E.P. “71013 Glorioso San Carlos” - Puno	92

RESUMEN

La presente investigación tiene como objetivo, determinar de qué manera el modelo holístico de código-alfabetización permite el desarrollo del pensamiento computacional en niños de educación primaria. Para el proceso de experimentación se ha seleccionado a los estudiantes de tercer a quinto grado de primaria de la I.E.P. “71013 Glorioso San Carlos” de la ciudad de Puno; la investigación es de tipo experimental.

Para la construcción del modelo holístico de código-alfabetización se ha tomado en cuenta conceptos de sistemas, modelando los componentes e interacciones de un sistema; cibernética de segundo orden, considerando la comunicación, control y la retroalimentación; autopoiesis, y principios pedagógicos del socio constructivismo y construccionismo. Se ha desarrollado y aplicado un prototipo de software en entorno web, basado en el modelo holístico de código-alfabetización utilizando la metodología ágil de desarrollo de software “Kanban”, lenguaje de programación PHP y bases de datos en MySQL.

Llegando a la conclusión de que, el modelo holístico de código-alfabetización construido en base a conceptos de sistemas y fundamentos pedagógicos y el prototipo de software aplicado, ha permitido el desarrollo del pensamiento computacional, en un nivel medio a alto, en niños de educación primaria de la I.E.P. “71013 Glorioso San Carlos” de la ciudad de Puno, tal como se demuestra en la prueba de hipótesis Z normal con un nivel de significancia del 0,05.

Palabras claves:

Modelo holístico, código-alfabetización, pensamiento computacional.

ABSTRACT

The objective of this research is to determine how the holistic model of code-literacy will allow the development of computational thinking in primary school children. For the experimentation process, students from the third to fifth grade of primary school of the IEP Glorioso San Carlos in the city of Puno have been selected; the research is of an experimental type.

For the construction of the holistic model of code-literacy, system concepts have been taken into account, modeling the components and interactions of a system; second-order cybernetics, considering communication, control and feedback; autopoiesis, and pedagogical principles of socio constructivism and constructionism. A software prototype has been developed and applied in a web environment, based on the holistic model of code-literacy using the agile software development methodology "Kanban", PHP programming language and MySql databases.

Reaching the conclusion that, the holistic model of code-literacy built on concepts of systems and pedagogical foundations and the prototype of applied software, has allowed the development of computational thinking, at a medium to high level, in primary school children of the IEP Glorioso San Carlos in the city of Puno, as demonstrated in the normal Z hypothesis test with a level of significance of 0.05.

Key words:

Holistic model, code-literacy, computational thinking.

CAPÍTULO I

INTRODUCCIÓN

Es importante el estudio y desarrollo de las competencias y capacidades computacionales en el nivel básico, lo cual ya es considerado por instituciones educativas en el mundo, en países como Inglaterra, que incluye el estudio de la programación de computadores a partir del año 2014, dentro de su plan curricular de estudios en educación primaria y secundaria; en nuestro continente, Chile es uno de los países pioneros que también sigue el mismo camino; es decir, existe un interés por el desarrollo del pensamiento computacional desde los niveles básicos de la educación.

Existe también el interés de desarrollar estas habilidades desde las empresas privadas mundiales, es el caso del proyecto code.org, que cuenta con un portal web, donde se ofrece gratuitamente el acceso, para poner en práctica los conceptos de programación de un computador, desarrollando el pensamiento computacional; éste proyecto cuenta con el patrocinio de empresas mundiales de tecnología como: Microsoft, Google, Facebook, Amazon y muchos más. En su portal web se muestra que el proyecto code.org es una organización que se dedica a promover el acceso a la computación en las escuelas de nivel básico y aumentar el actuar de las mujeres y las minorías subrepresentadas; indicando que su visión es que: todo estudiante de nivel escolar primario y secundario tenga acceso a aprender ciencias de la computación, de la misma forma que ahora tienen la posibilidad de aprender otras asignaturas como biología, química o álgebra, para lo cual propone un currículo de estudios para los distritos más grandes de los Estados Unidos.

En el presente trabajo de investigación se pretende proponer un modelo holístico de código-alfabetización, basado en la teoría de sistemas, y éste aplicarlo en la implementación de un prototipo de software acorde con las características propias de la región Puno, en donde se plantea experimentarlo, con lo cual se busca el desarrollo del pensamiento computacional en niños de tercero a quinto grado de primaria.

El presente trabajo de investigación está dividido en cinco capítulos, los mismo que describimos a continuación:

El primer capítulo describe el problema, los antecedentes, los objetivos, la justificación, alcances y limitaciones y la definición de las variables.

El segundo capítulo, contiene el marco teórico de la investigación, donde se considera al Modelo holístico de código-alfabetización y al pensamiento computacional, también se presenta el marco conceptual y la hipótesis.

El tercer capítulo, contiene el método considerando todos los elementos propios de la metodología de la investigación.

El cuarto capítulo, considera la presentación de resultados, dentro de ello se presenta el proceso de la construcción del modelo holístico de código-alfabetización, la implementación del prototipo de software y la aplicación del prototipo de software en el desarrollo del pensamiento computacional en niños de nivel primario en la ciudad de Puno.

El quinto capítulo contiene la discusión, conclusiones y recomendaciones.

1.1 PLANTEAMIENTO DEL PROBLEMA

El Perú en su Currículo Nacional y su Programa Curricular de educación primaria para el año 2017, no muestra el desarrollo del pensamiento computacional en ningún sentido, solo toca aspectos generales de habilidades computacionales referidas al uso de las TIC, esto demuestra que aún no se tiene claro los conceptos referidos a pensamiento computacional, mientras que en el mundo, ya hace muchos años atrás se viene desarrollando estos conceptos como parte de sus currículos oficiales de estudio, países como Rusia, Sudáfrica, Nueva Zelanda y Australia han realizado cambios en sus currículos oficiales de educación primaria; Reino Unido inició en 2012 programas pilotos para llevar la programación a todos los niños y niñas, al mismo tiempo Estonia y Finlandia promueven el conocimiento comprensivo que permite al estudiante construir y comprender las tecnologías digitales y ser capaces de crear sus propias aplicaciones.

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

Se tiene conocimiento de la existencia de varios entornos para iniciar con el aprendizaje de la programación de computadoras, uno de los más conocidos y aplicados es el Scratch, en sus diferentes versiones que muy esporádicamente son desarrollados en los centros educativos de nuestra localidad, pero como aportes aislados, nada estructurados y menos formalizados. También se tiene la existencia de otro entorno muy utilizado en el mundo como es el caso del entorno code.org, en donde se puede iniciar con el desarrollo de los conceptos de pensamiento computacional, pero este es un entorno basado con formas, personajes y diseños extranjeros, nada relacionados con el medio del estudiante ni su idiosincrasia. Siendo el Perú y la región Puno en particular un medio socioculturalmente diverso, con muchas lenguas, costumbres arraigadas no pueden adecuarse a estos entornos basados para realidades distintas a la nuestra.

1.2 DESCRIPCIÓN DEL PROBLEMA

Los modelos holísticos en el mundo se aplican en diferentes formas, es necesario tomar en cuenta que en la educación se tienen la aplicación de los conceptos de la sistémica para el proceso de enseñanza aprendizaje y por tanto, es necesario también trabajar estos conceptos a nivel del Perú y más aún en la región de Puno, donde es necesario tomar un nuevo modelo para alfabetizar en el conocimiento del código de las máquinas.

Se tiene que varios organismos mundiales vienen promoviendo el concepto de pensamiento computacional como el uso de la Hora del Código, movimiento mundial que busca incursionar a los estudiantes de nivel inicial, primaria y secundario en el desarrollo de habilidades relacionadas con la programación de computadoras, esto

desarrollado a nivel mundial, se viene ejecutando progresivamente en diferentes instituciones educativas del mundo, de igual manera se está iniciando este proceso en el Perú y en la ciudad de Puno, donde se tiene registrado dos organismo que llevan a cabo esta jornada de trabajo.

En la ciudad de Puno es evidente la falta de aplicación de los conceptos de código-alfabetización pues se observa que en un 80% las instituciones educativas de nivel primario, no desarrollan los conceptos de pensamiento computacional y el desarrollo de actividades académicas relacionadas con la código-alfabetización, sólo se desarrollan actividades académicas relacionadas al uso del computador con programas como: sistemas operativos, internet, ofimática y graficadores, al cual le denominan curso de computación.

1.3 FORMULACIÓN DEL PROBLEMA

1.3.1 Problema general

¿De qué manera el modelo holístico de código-alfabetización permite el desarrollo del pensamiento computacional en niños de educación primaria de la región Puno?

1.3.2 Problemas específicos

- ¿Será posible implementar un modelo holístico de código-alfabetización para los niños de educación primaria de la región Puno?
- ¿El modelo holístico de código-alfabetización será suficiente y necesario para desarrollar un prototipo de software que desarrolle el pensamiento computacional?

- ¿En qué medida el uso del prototipo de software, basado en el modelo holístico de código-alfabetización permite desarrollar el pensamiento computacional en los niños de educación primaria de la región Puno?

1.4 ANTECEDENTES

Hitschfeld Nancy, Pérez Jorge y Simmonds Jocelyn en su trabajo: “Pensamiento computacional y programación a nivel escolar en Chile: el valor de formar a los innovadores tecnológicos del futuro”, plantean que “el pensamiento computacional y la programación son una disciplina que nos sirve para resolver problemas haciendo uso de la tecnología. Además, aclaran que, esto no reemplaza la creatividad, el razonamiento y el pensamiento crítico, sino que refuerza estas competencias. Para formular problemas de manera que puedan ser resueltos usando computadores”.(Hitschfeld, Pérez, & Simmonds, 2015)

El desarrollo del PCyP permite además desarrollar y fortalecer una serie de competencias transversales que son indispensables para formar líderes e innovadores. Estas competencias incluyen: “confianza en el manejo de la complejidad, persistencia al trabajar con problemas difíciles, tolerancia a la ambigüedad, habilidad para lidiar con problemas abiertos, y habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común”.(Hitschfeld et al., 2015)

Román Gonzales, Marco (2016) en su Tesis Doctoral: “Código-alfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas”, plantea que, en el contexto digital en que

nos encontramos, la capacidad de conocer y utilizar el lenguaje de las computadoras “es una habilidad básica, que nos permite participar de manera plena y efectiva en la realidad computarizada que nos rodea: se trata de 'programar o ser programado, de ser 'application capacitado o application dependiente'. El término ‘código-alfabetización’ (del inglés ‘code-literacy’) ha sido acuñado recientemente para referirse al proceso de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación”. (Román, 2016) Por tanto, se considera que un niño o niña está código-alfabetizada cuando es capaz de leer y escribir en el mismo lenguaje de los computadores y de otras máquinas, y de pensar computacionalmente.

Se plantea como problema: “si es posible (adecuado y viable) y deseable (relevante) introducir la código-alfabetización en el sistema educativo español, específicamente en las etapas de Educación Primaria y Secundaria”.(Román, 2016) Para poner en práctica esto se realizan tres estudios relacionados: A) El primero es un estudio descriptivo sobre el evento mundial “La hora del código”, realizado en España en los años 2013 y 2014, considerando 29 instituciones educativas de todo el país, llegando a 23 profesores y 1662 estudiantes. B) El segundo, se realiza un estudio experimental en donde se diseña, construye y valida un instrumento como es el “Test de Pensamiento Computacional” (TPC), dicho test para su validación se apoya en la opinión de 20 expertos y para su aplicación lo realiza en estudiantes de primer ciclo de la ESO, en un total de 1251 estudiantes y 24 instituciones educativas. C) El tercero, consiste en realizar una evaluación de programas de código-alfabetización y verificar el desarrollo del pensamiento computacional.

Se tiene como resultado: “A) El evento ‘La Hora del Código’, que se realiza en todo el mundo, logra el objetivo de introducir la código-alfabetización, eliminando el mito de que es difícil programar y haciendo que estudiantes de todo el país, se interesen por el estudio de la programación de computadoras. B) Se logra diseñar, construir y validar el instrumento denominado ‘Test de Pensamiento Computacional’, que muestra propiedades psicométricas convenientes: como es fiable: esto considera que tiene consistencia interna y estabilidad; además se demuestra su validez, considerando que es discriminante, convergente y factorial”.(Román, 2016)

Alonso Urbano, David (2017) en su trabajo de investigación: “Scratch como herramienta para la enseñanza de la programación en la Educación Primaria”, se plantea la hipótesis que: “Scratch es una herramienta que facilita el aprendizaje de la programación desde el punto de vista de la usabilidad”(Alonso, 2017). Las conclusiones a las que llega en su trabajo son: A) Al realizar un análisis heurístico de Scratch, se corrobora el cumplimiento de pautas que son evaluadas a través de encuestas, en donde se confirma el análisis de usuario, concluyendo que el Scratch es una herramienta usable. B) Se hace notar con mucho énfasis, que no es lo mismo saber manejar la herramienta Scratch con saber programar el computador, esto lo demuestra al plantea situaciones que incluye programación, donde los estudiantes tienen mayores problemas. C) El papel preponderante del docente en el proceso de enseñanza – aprendizaje de la programación, donde los estudiantes requieren la actuación del docente, por lo cual es imprescindible poner mayor énfasis en la formación del docente con conocimientos de programación de computadoras. (Alonso, 2017)

Capot, Rosa y Espinoza, Rosa (2015), en su trabajo de Investigación titulado: “Desarrollo del Pensamiento Computacional con Scratch”, en su trabajo de investigación plantean sesiones de aprendizaje que les permita desarrollar el pensamiento computacional en dos muestras diferentes; el primero en un curso taller a 19 niños de enseñanza básica de segundo ciclo y el segundo a 53 estudiantes ingresantes a la universidad en el curso de Computación I, principalmente en la Carrera de Ciencias de la computación. Se diseñan 11 sesiones de aprendizaje donde se considera como aspecto importante, el nivel de complejidad, siendo éstos de lo más básico, manejo de datos, hasta lo más complejo, automatización y simulación de problemas de la vida real. Arribando a la conclusión de que, el diseño y aplicación de sesiones de aprendizaje con contenidos interactivos, lúdicos y basados en la herramienta Scratch, logran desarrollar adecuadamente el pensamiento computacional, poniendo hincapié en el desarrollo de la creatividad de los estudiantes quienes en sesiones superiores logran desarrollar su propios juegos y socializarlos entre sus compañeros, haciendo que su producto sea tangible y ver que otros utilicen su creación, logra una mayor motivación. Se resalta también la importancia de aprender a resolver problemas que debe llevarse luego a un ámbito más científico. (Capot & Espinoza, 2015)

Taborda, Hernando y Medina, Diego (2014) en su trabajo: “Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares”, se plantea como objetivo: “Examinar y describir la forma como la estructura del programa informático SCRATCH y la demanda de las tareas propuestas en clase, promueven el pensamiento computacional y el desarrollo de habilidades de programación”.

(Taborda & Medina, 2014) Se emplea la herramienta del “Análisis de Tareas”, que se utiliza para examinar a profundidad una situación problemática desde su estructura, también se utilizó en el trabajo de investigación, para verificar el nivel de aprendizaje de los individuos en situaciones reales. Las situaciones problemáticas que se plantean son: “Mi acuario y yo”, y “El ciclo de vida”; actividades que se desarrollan por los profesores del nivel primario y se selecciona estas actividades debido a que son actividades introductorias del programa informático en tercero de primaria y el tiempo de duración no es muy largo. Llegando a la conclusión de que: la herramienta del “Análisis de Tareas” muestra que el uso del aplicativo de software Scratch, utilizado como parte de las actividades educativas diseñadas, promueven el pensamiento computacional, mejorando la adquisición de conocimiento y las habilidades cognitivas. (Taborda & Medina, 2014)

Domínguez de León, Guillermo Rafael (2013) en su trabajo de investigación Doctoral titulado: “Desarrollo de un modelo holístico y un método sistémico para representar y estimular la metacognición”, El objetivo que se plantea es: “desarrollar un modelo holístico y un método sistémico para representar y estimular la actividad metacognitiva de los estudiantes a efecto de enriquecer sus habilidades metacognitivas para mejorar su desempeño en el aprendizaje”(Domínguez, 2013) para esto se emplea un enfoque sistémico, aplicando los principios de la cibernética de segundo orden y también el aprendizaje autorregulado. En el trabajo se presenta dos productos principales: a) un modelo, basado en un enfoque sistémico, mediante el cual se abstrae la actividad cognitiva, siendo definido éste como un proceso cibernético de segundo orden, que se desenvuelve como parte de un subsistema cognitivo, incorporando además el elemento de la autopoiesis. b) un método,

mediante el cual se determina los procesos para desarrollar y estimular la actividad metacognitiva. Teniendo como resultado del trabajo que: el modelo holístico propuesto de la actividad metacognitiva, conjuntamente con el método sistémico y el proceso de trabajo que se propone, han logrado estimular la actividad cognitiva de los estudiantes, lo cual queda demostrado en los resultados del grupo experimental y control. (Domínguez, 2013)

1.5 JUSTIFICACIÓN DE LA INVESTIGACIÓN

La tecnologías de la información y la comunicación en estos tiempos es un factor ineludible a considerar en los diferentes aspectos de nuestros quehaceres y mucho más en la educación en sus diferentes niveles, en donde aún en nuestro país no se cuenta con un desarrollo de los conceptos denominados, pensamiento computacional y código-alfabetización, que está siendo promovido por figuras mundiales del mundo de la tecnología como Billi Gates, Mark Zuckerberg y otros, sobre todo con la plataforma code.org y "La hora del código" , que busca llevar a los más pequeños en un entorno de programación de computadoras basado en bloques, el incursionar en el mundo de escribir código fuente, que puede ser entendido por un computador y ejecutar dichas instrucciones. Estos conceptos se desarrollan en el mundo como parte de las currículas oficiales en países de Europa, América del norte, de América del sur y el África.

Es necesario por tanto, contar con un propio modelo basado en conceptos holísticos, o sea con un enfoque de sistemas, que busque la aplicación de estos conceptos en nuestro medio, más en específico en nuestra región de Puno, en donde podamos desarrollar la código-alfabetización considerando factores de integridad, y conceptos educativos

promoviendo el desarrollo de la programación de computadoras en niños desde el nivel primario.

Considerando a Wing afirma que “el Pensamiento Computacional (PC) implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, basándose en los conceptos fundamentales de la ciencia de la Computación. El PC incluye una amplia variedad de herramientas mentales que reflejan la amplitud del campo de la Computación”(Bocconi, Chiocciariello, & Dettori, 2016), y busca desarrollar capacidades y habilidades propias de estos tiempos, donde los ciudadanos estén preparados para afrontar problemas globales como el calentamiento global, la biodiversidad, la pobreza y las enfermedades infecciosas, entre otras.

Con la presente investigación se busca también la implementación de un prototipo de software que en un entorno web y también en un entorno para dispositivos móviles, aplique el modelo desarrollado, para contar con un medio donde aplicar dichos conceptos. Los niños de nivel primario de nuestra región tendrán un entorno en donde poner en práctica la código-alfabetización, generando en base al aplicativo código fuente, programar al computador jugando, con personajes y escenario acorde a su contexto y se desarrolle las habilidades y actitudes propias del pensamiento computacional, que lleven a generar personas capaces de solucionar en un futuro cercano problemas de complejidad global de su medio y el mundo.

1.6 LIMITACIONES DE LA INVESTIGACIÓN

Se considera limitaciones de la investigación los siguientes:

- La implementación de laboratorios de cómputo y la línea de internet de las Instituciones Educativas de la ciudad de Puno, ya que es necesario contar con internet y equipos de cómputo para verificar el pensamiento computacional en los niños.
- El tiempo disponible con los niños para el uso del aplicativo de software diseñado e implementado, ya que se tendrá que realizar la experimentación fuera de las horas de clase regular programadas en las Instituciones Educativas.
- El costo del desarrollo del software que no permite la implementación íntegra del sistema, sino un prototipo.

1.7 OBJETIVOS

1.7.1 Objetivo General

Determinar de qué manera el modelo holístico de código-alfabetización permite el desarrollo del pensamiento computacional en niños de educación primaria de la región Puno.

1.7.2 Objetivos Específicos

- Diseñar un modelo holístico de código-alfabetización para los niños de educación primaria de la región Puno.
- Desarrollar un prototipo de software en base al modelo holístico de código-alfabetización.
- Evaluar el nivel de desarrollo del pensamiento computacional en los niños de educación primaria de la región Puno, con el uso del prototipo de software.

1.8 HIPÓTESIS

1.8.1 Hipótesis General

El modelo holístico de código-alfabetización permite el desarrollo del pensamiento computacional en niños de educación primaria de la región Puno.

1.8.2 Hipótesis específicas

- La implementación de un modelo holístico de código-alfabetización para los niños de educación primaria de la región Puno, se fundamenta en teorías holísticas y pedagógicas.
- El modelo holístico de código-alfabetización es una base necesaria y suficiente para el desarrollo de un prototipo de software que permita generar el pensamiento computacional.
- Mediante el uso de la aplicación de software se desarrolla el pensamiento computacional en los niños de educación primaria de la región Puno en un nivel medio a alto.

CAPÍTULO II

MARCO TEÓRICO

2.1 TEORÍAS GENERALES

2.1.1 Cibernética de 2do orden

“La cibernética puede entenderse como la ciencia que analiza racionalmente lo que significa gobernar”. (Acosta, 2001)

La cibernética es la ciencia que estudia los flujos de energía vinculados a la teoría de control y a la teoría de sistemas, y se aplica también a los sistemas físicos y sociales.

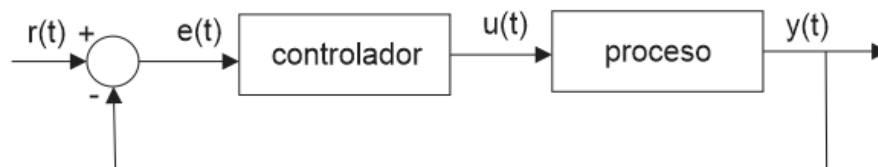


Figura 1: Comunicación y control cibernético

Donde:

$u(t)$ = función de Entrada

$y(t)$ = función de salida

$e(t) = r(t) - y(t)$

“La cibernética de segundo orden se ocupa del observador como parte de lo observado. Así, según Heinz Von Foerster, el objetivo de la segunda cibernética es explicar el observador a sí mismo, es decir la cibernética de la cibernética, y se refiere a los sistemas que son capaces de modificar su objetivo o finalidad (o su camino) por

sí mismos, sin necesidad de ser guiados por alguien o algo desde fuera del sistema”.
(Heylighen & Joslyn, 2001)

2.1.2 Autopoiesis

En el libro “De Máquinas y Seres Vivos” publicado en 1973, Maturana y Varela definen a un sistema autopoietico de la siguiente manera:

“Una máquina autopoietica es una máquina organizada como un sistema de procesos de producción de componentes concatenados de tal manera que producen componentes que: i) generan los procesos (relaciones) de producción que los producen a través de sus continuas interacciones y transformaciones, y ii) constituyen a la máquina como una unidad en el espacio físico”.(Maturana & Varela, 1994)

Entonces una máquina autopoietica “constantemente especifica y produce su propia organización a través de la producción de sus propios componentes, bajo condiciones de continua perturbación y compensación de esas perturbaciones (producción de componentes). Entonces se afirma que una maquina autopoietica es un sistema homeostático”.(Razeto & Ramos, 2013)

Considerando el caso educativo un sujeto que se ubica dentro de un sistema educativo, es importante primero pasar del concepto biológico de autopoiesis a conceptos sociológicos y antropológicos, en donde basándose en el concepto el componente en el sistema educativo debe cambiar, transformarse, mutar constantemente, pero tener muy en cuenta, que tiene que reflexionar y volver a evaluar lo sucedido los cambios dados y esto repetirse en círculos. (Daza, 2013)

Considerando a Mendoza y Godoy (2016) es necesario tener en cuenta el concepto de autopoiesis como un elemento primordial que permita explicar y comprender desde un enfoque sistémico las características principales del aprendizaje y estos son.

(a) **Organización:** El aprendizaje se da mediante la interacción de los elementos que lo constituyen: estudiantes, docentes, medio ambiente, etc. y un proceso de realimentación positiva o negativa, considerando su complejidad estructural.

(b) **Determinismo estructural:** “son los cambios que tienen lugar dentro de cualquier organismo o sistema que pueden ser observables de modo directo o indirecto”. (Maturana & Porsksen, 2004)

(c) **Acoplamiento estructural:** “dado que ningún organismo o sistema está realmente aislado siempre deberá interactuar con el complemento de sí, es decir con todo lo que no le define como tal, lo cual aplica tanto para seres vivos como no vivos”. (Maturana & Varela, 1994)

(d) **Clausura operativa:** “el aprendizaje ocurre para cada individuo atendiendo a sus propias estructuras y el modo como esta se da dependerá de esas estructuras. El rango dentro del cual puede darse el aprendizaje es muy amplio y estocástico, a diferencia de otros organismos y sistemas simples donde tienen un carácter más bien determinístico”. (Mendoza & Godoy, 2016)

2.1.3 Modelos Educativos

Constructivismo

Esta teoría sostiene que el Conocimiento no se descubre, se construye; “es un proceso activo donde el alumno elabora y construye sus propios conocimientos a

partir de su experiencia previa y de las interacciones que establece con el maestro y con el entorno”.(Coloma & Tafur, 1999) El enfoque constructivista tiene muchos autores importantes, entre los cuales se encuentran Ausubel, Bruner, Jean Piaget y Vygotsky.

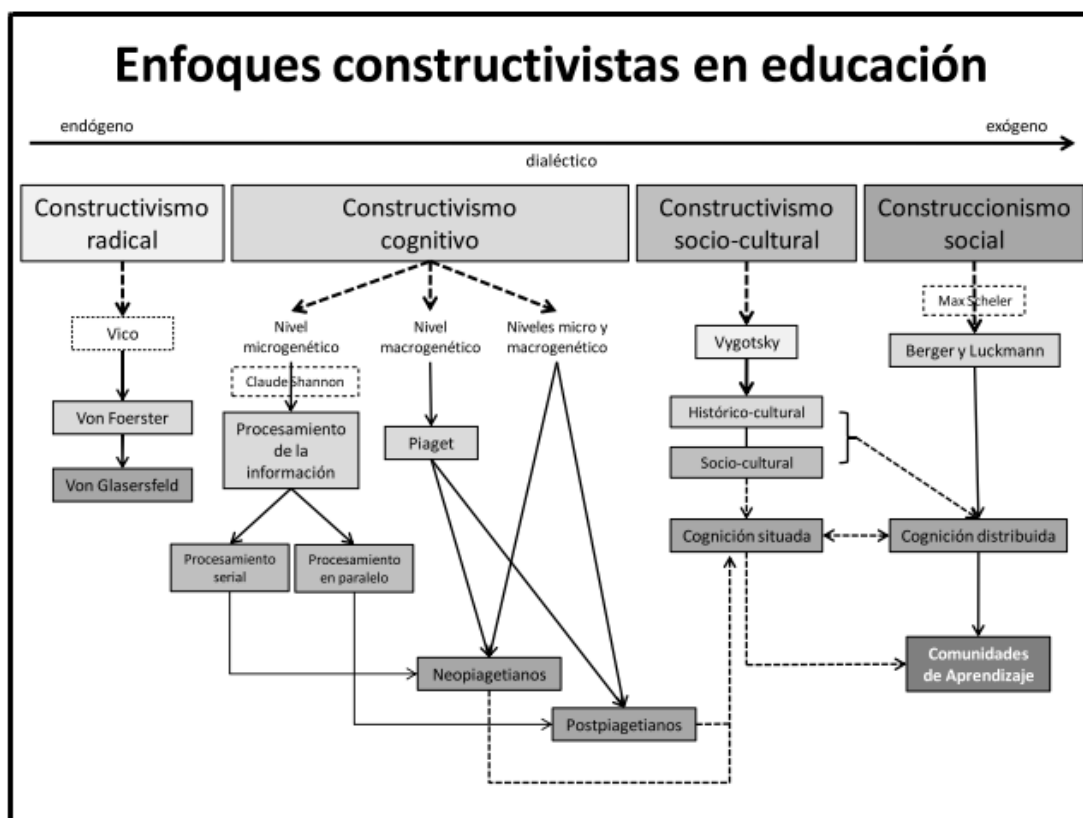


Figura 2: Enfoques constructivistas de la educación

Este enfoque considera que “el conocimiento es una construcción del ser humano: cada persona percibe la realidad, la organiza y le da sentido en forma de constructos, gracias a la actividad de su sistema nervioso central, lo que contribuye a la edificación de un todo coherente que da sentido y unicidad a la realidad”.(Ortiz, 2015)

En un proceso de enseñanza-aprendizaje en este enfoque, José Manuel Serrano plantea la existencia del **Triángulo Instruccional**, en el cual se observa la interacción

de cuatro elementos: el alumno, el profesor, los contenidos y los objetivos; que se ubican en los vértices. Además, en los lados uniendo los vértices se ubican, la construcción de significados, uniendo al alumno con el contenido, la atribución de sentido relacionando al alumno con los objetivos y las competencias uniendo los contenidos con los objetivos. Se tiene también que el profesor se ubica en el baricentro del triángulo. (Serrano & Pons, 2008)

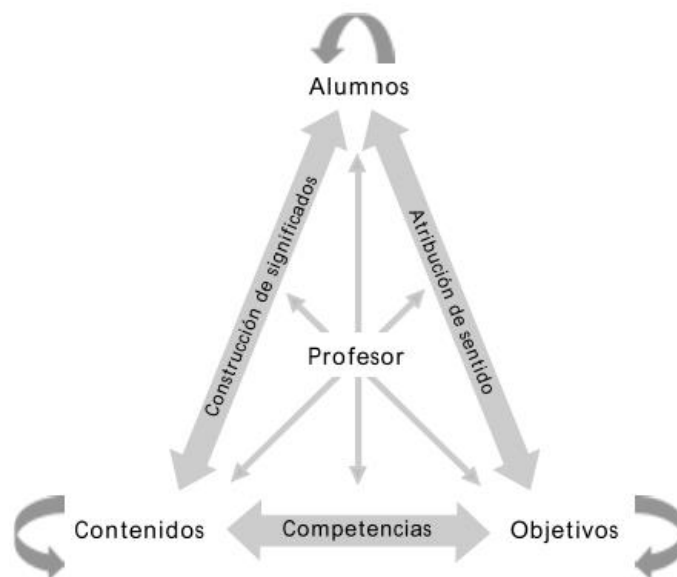


Figura 3: Triángulo instruccional

Construccionismo

Es una teoría de la educación “desarrollada por Seymour Papert del Instituto Tecnológico de Massachussets. Está basada en la teoría del aprendizaje creada por el psicólogo Suizo Jean Piaget (1896-1990) Papert, trabajó con Piaget en Ginebra a finales de los años 50 y principios de los 60”. (Falbel, 2001)

La teoría del construccionismo indica que “el aprendizaje es mucho mejor cuando los niños se comprometen en la construcción de un producto significativo, tal como un castillo de arena, un poema, una máquina, un cuento, un programa o una canción. De esta forma el construccionismo involucra dos tipos de construcción:

cuando los niños **construyen cosas en el mundo externo**, simultáneamente **construyen conocimiento al interior de sus mentes**". (Falbel, 2001) Este conocimiento que se genera construyendo un objeto con significado social, se retroalimenta en un ciclo auto reforzante, donde al construir un elemento significativo también está construyendo su aprendizaje.

“La institución educativa debe apoyar las iniciativas de los docentes que quieren transformar sus clases, y brindar el respaldo que posibilite aplicar la creatividad en el proceso de enseñanza”.(Pinto, 2013)

Papert pone algunas consideraciones en la aplicación del construccionismo en las aulas:

- Docentes formados y capacitados en el uso de técnicas para la acción, creación de objetos reales o abstractos, sean estos con o sin el uso del computador.
- Un ambiente de aprendizaje idóneo y a fin con el estudiante, es decir debe ser amigable, acogedor y estimulante.
- La frustración o el error y la satisfacción son factores importantes, pues se construye algún producto realizando muchas pruebas, que no funcionarán en un primer intento.
- Unos requerimientos relacionados con equipamiento con tecnología y un currículo flexible.
- Un ambiente de aprendizaje con estudiantes de distintas niveles de destreza, significado ello que en algunas ocasiones se convienen estudiantes de diferentes edades.(Falbel, 2001)

2.1.4 Holismo y educación

Considerando a Ramón Gallegos, nos plantea como principios de una visión holística de la educación los siguientes: a) Unidad, donde el conocimiento es unitario conjuncionando sentimientos, discernimientos, intuición y cognición, b) Totalidad, donde el todo es más que la suma de las partes, c) Desarrollo cualitativo, pues existen interrelaciones circulares y dinámicos, d) Transdisciplinariedad, donde todas las disciplinas científicas se integran, e) espiritualidad, entendida esta como la experiencia directa con la totalidad y f) aprendizaje, que ocurre en todos los niveles: físico, espiritual, racional, cognitivo, artísticos y espacial. (Gallegos, 2018)

Eugenio Enrique Mendoza, en su investigación: “El aprendizaje desde un enfoque holístico e integrador”, define el **aprendizaje** como: “un proceso complejo y sistémico, cuyo origen está en la necesidad que tiene todo ser vivo de alcanzar un equilibrio con su complemento que le permita mantener su existencia y de este modo lograr sus fines.”(Mendoza & Godoy, 2016) Además, toma en cuenta tres elementos o componentes que se interrelacionan, compartiendo información y energía como son: el componente biológico, donde se considera el sistema nervioso, las relaciones de autopoiesis, la filogenia y otras características; El componentes psicológico y cognitivo, tomando en cuenta los diferentes elementos cognitivos y psicológicos como los procesos cognitivos, la metacognición, las emociones; El componente ecológico, que incluye el contexto cognitivo donde se desarrolla el aprendizaje.

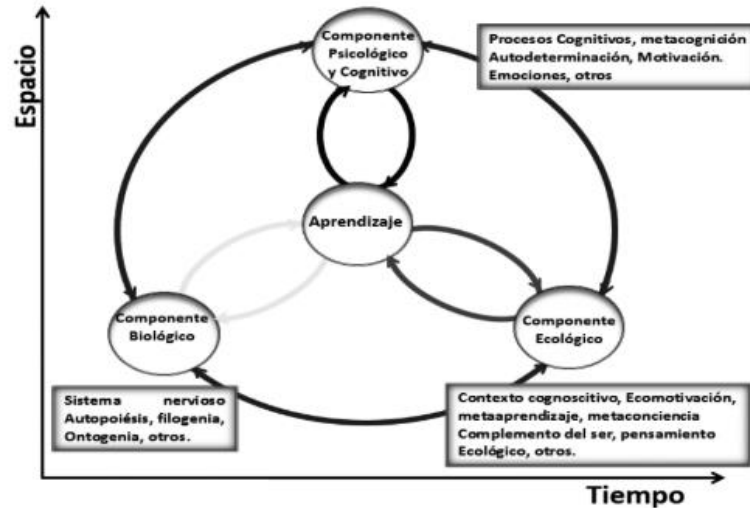


Figura 4: Modelo holístico del aprendizaje humano según Mendoza(2016)

Considerando a Mendoza Vera, Eugenio (2015); quien realiza una definición ampliada de aprendizaje, indicando que “es un proceso complejo y sistémico, cuyo origen está en la necesidad que tiene todo organismo vivo de alcanzar un equilibrio con el entorno, que le permita mantenerse vivo y lograr sus fines”.(Mendoza, 2015) El aprendizaje en el ser humano lo considera como un sistema, cuyos componentes son el docente, las estrategias docentes, el alumno cuyo proceso central es el aprendizaje y cuenta con variables intervinientes endógenas y exógenas, mostrando además, el proceso de retroalimentación mediante la evaluación en sus diferentes formas.

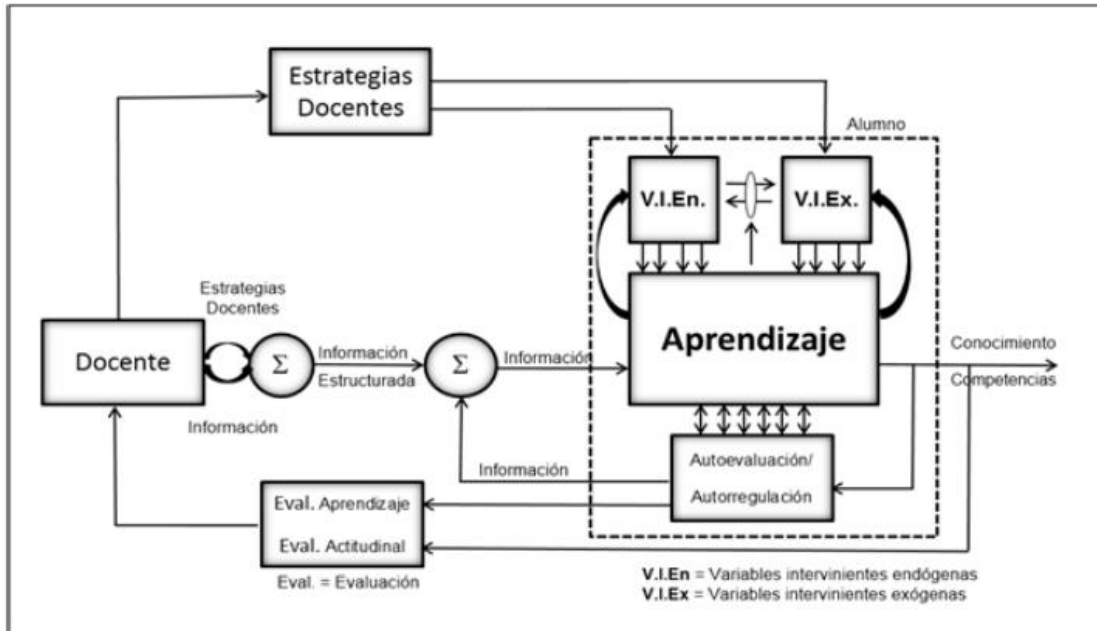


Figura 5: Modelo holístico de aprendizaje – Mendoza Vera, Eugenio (2015)

2.2 BASES TEÓRICAS

2.2.1 MODELOS HOLÍSTICOS

2.2.1.1 Definición de modelo

Considerando a Carlos Tapia sobre modelo: “Es una simplificación de la realidad, surgen de un análisis de todas las variables intervinientes en el sistema y de las relaciones que se descubren existen entre ellas”.(Tapia, 1971)

Mohammad Azarang, nos indica sobre modelo: “Un modelo es una representación de la realidad desarrollado con el propósito de estudiarlo. En la mayoría de los análisis no es necesario considerar todos los detalles de la realidad, entonces, el modelo no sólo es un sustituto de la realidad sino también una simplificación de ella”.(Azarang, 1996)

Preguntándonos sobre el por qué construir modelos de la realidad, nos podemos responder, desde el punto de vista de la utilidad, que los modelos son útiles para: a) el pensamiento, se ordena y completa el conocimiento de la realidad; b) la comunicación, evita la ambigüedad; c) el entrenamiento y la instrucción, permite la experimentación sin riesgos físicos y en costos; d) la predicción, se puede predecir el comportamiento del sistema real y e) experimentación, pues es barato y seguro experimentar con un modelo que en la realidad. (Tarifa, 1998)

2.2.1.2 Clasificación de los modelos

Los modelos según Mohammad Azarang pueden clasificarse en:

- Modelos icónicos.
- Modelos análogos.
- Modelos simbólicos:
 - Modelos determinísticos.
 - Modelos estocásticos o probabilísticos.
 - Modelos dinámicos.
 - Modelos estáticos.
 - Modelos continuos.
 - Modelos discretos. (Azarang, 1996)

Características de los modelos

La siguiente lista muestra las características principales que debe tener todo modelo:

- “Confiabilidad

- Sencillez
- Bajo costo de desarrollo y operación
- Manejabilidad
- De fácil entendimiento, tanto el modelo como los resultados.
- La relación costo-beneficio debe ser positiva”. (Azarang, 1996)

Considerando a Enrique Tarifa, el modelo debe ser:

- “Fácil de entender por parte del usuario.
- Dirigido a metas u objetivos.
- Sensato, en cuanto no de respuestas absurdas.
- Fácil de manipular y controlar por parte del usuario. Es decir, debe ser sencillo comunicarse con el modelo.
- Completo, en lo referente a asuntos importantes.
- Adaptable, con un sencillo procedimiento para modificar o actualizar el modelo.
- Evolutivo, debe ser sencillo al principio y volverse más complejo en el tiempo”. (Tarifa, 1998)

2.2.2 EL HOLISMO

“El holismo es una doctrina filosófica contemporánea que tiene su origen en los planteamientos del filósofo sudafricano Smuts (1926), quien fue el primero en utilizar el término en su Libro ‘Holismo y evolución’. Sin embargo, su raíz ‘holos’, procede del griego y significa ‘todo’, ‘íntegro’, ‘entero’,

‘completo’, y el sufijo ‘ismo’ se emplea para designar una doctrina o práctica de la globalidad o de la integralidad”. (Barrera, 2000)

Considerando a Carlos Wernicke, en su artículo sobre la educación holística, afirma que: “El ser humano es observable desde diversas perspectivas. Todos estaremos de acuerdo en que es posible dispensar nuestra atención al ser humano, por caso, en cuanto ser biológico o en cuanto ser emocional. El holismo distingue en el ser humano cinco planos, esto es, cinco caras de un mismo todo”. (Wernicke, 1999)



Figura 6: Los planos constitutivos del ser humano

Los planos que considera son: a) el plano físico, compuesto de materia y energía; b) el plano biológico, el concepto central es la vida y la interacción con su entorno intercambiando energía, c) el plano emocional, los sentimientos, y los estímulos del entorno que ingresan a través de sensaciones, d) el plano intelectual, donde se tiene el procesamiento de la información con entradas y formas de responder a estas informaciones salidas y e) el plano

espiritual, considera la filosofía y la religiosidad tomando en cuenta su relación espiritual con el todo.(Wernicke, 1999)

2.2.3 CODIGO-ALFABETIZACIÓN

2.2.3.1 ¿Qué es código-alfabetización?

Considerando a Román Gonzales, define la código-alfabetización como el “proceso de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación”.(Román, 2013)

Además, este mismo autor se pregunta sobre, ¿dónde ubicar exactamente la código-alfabetización respecto de otros nuevos alfabetismos afines como la “alfabetización digital” y la “alfabetización web”? y realizando una descripción sobre estos tipos de alfabetización podemos observar:

- **Alfabetización digital (digital literacy)**

Se ubica en el nivel 1.0 y es la alfabetización de textos digitales, refiriéndose a texto todo material que puede transmitirse en el espacio y el tiempo y digital cuando se representa mediante 1 (unos) y 0 (ceros). En esta alfabetización entonces se incluye el aprendizaje de procesadores de textos, hojas de cálculo, retoques fotográficos, etc. que pueden tener su almacenamiento online o offline.

- **Alfabetización web (web literacy)**

Se ubica en el nivel 2.0 e incluye actividades que se desarrollan en la web. En esta alfabetización se considera la representación mediante hipertexto y material

multimedial en la web y las redes de computadoras, las comunicaciones con las redes sociales, gestionar perfiles y blogs, diseñar páginas web.

- **La código-alfabetización**

Se sitúa en un nivel 3.0 superior a las otras, ya que nos permite ingresar al mismo código que generan las imágenes, las páginas, las aplicaciones. Es manejar los lenguajes de programación y escribir los códigos que hacen posible la existencia de las aplicaciones, juegos, sistemas.(Román, 2016)

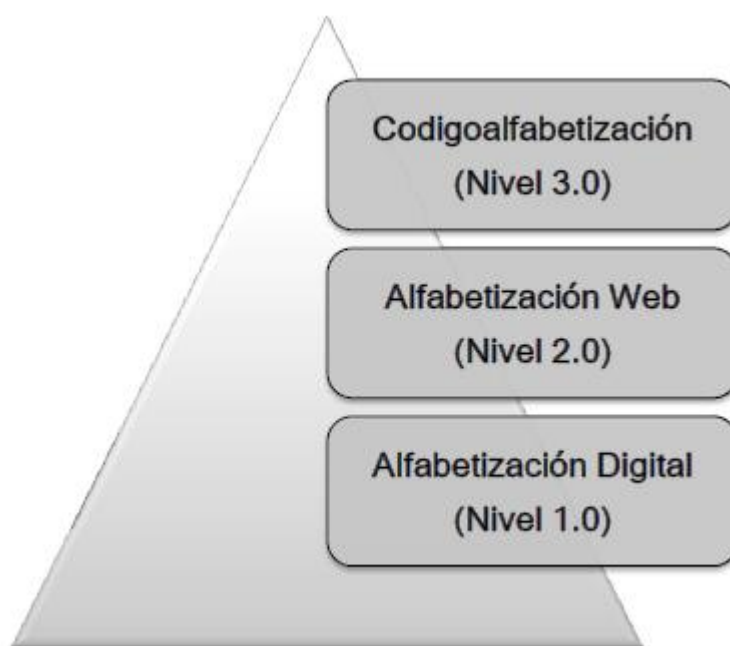


Figura 7: Alfabetización digital, Alfabetización web y Código-alfabetización.

Román Gonzales (2016)

Otros autores, la consideran como el “...ser capaz de leer y escribir en el lenguaje de los ordenadores y otras máquinas, y pensar computacionalmente”. (Belshaw, 2013)

Casanova amplia este concepto indicando: “Sin embargo, nosotros añadiríamos que, en profundidad, la código-alfabetización iría enfocada hacia

el **proceso de enseñanza-aprendizaje** en sí mismo y no sólo de la lectoescritura. El aprendizaje basado en computación o también pensar computacionalmente”. (Casanova, 2014)

2.2.3.2 El movimiento código-alfabetización o code- literacy

El movimiento ‘code-literacy’ tiene su origen en los Estados Unidos alrededor del año 2010 y tiene como axioma fundamental el considerar, y promover, la programación informática (‘coding’) como el alfabetismo clave en el siglo XXI. Consideramos que son tres sus figuras fundamentales, a saber: Marc Prensky, Douglas Rushkoff, y Mitch Resnick. Describiremos brevemente a cada uno.

Prensky Mark, define los términos: “nativos digitales e inmigrantes digitales; los primeros son los nacidos en la era digital y tienen interiorizados los textos digitales y los manejan como una ‘lengua materna’; frente a los segundos, nacidos y educados en la era analógica, y que han debido aprender a manejar los textos digitales como si se tratara de una lengua extranjera. Además, indica que existen rasgos distintivos entre ambos”. (Prensky, 2010)

Indica sobre código-alfabetización: “Creo que la habilidad única que, por encima de todo, distinguirá a una persona alfabetizada es la **alfabetización de programación**, la capacidad de hacer que la tecnología digital haga lo que sea, dentro de lo posible que se quiera: doblar la tecnología digital a las necesidades, propósitos y Así como en el presente doblamos palabras e imágenes. Algunos

llaman a esta habilidad la interacción hombre-máquina; Algunos la llaman alfabetización procesal. Otros sólo lo llaman programación”. (Prensky, 2008)

Douglas Rushkoff , por otra parte indica: “Como yo lo veo, la ‘code literacy’ (código-alfabetización) es un requisito para la participación en un mundo digital. Cuando adquirimos el lenguaje, no sólo aprendimos a escuchar, sino también cómo hablar. Cuando adquirimos el texto, no sólo aprendimos a leer, sino también cómo escribir. Ahora que tenemos computadoras, estamos aprendiendo a usarlas pero no cómo programarlas. Cuando no estamos alfabetizados en el código, debemos aceptar los dispositivos y el software que usamos con las limitaciones y agendas que sus creadores han incorporado en ellos”. (Rushkoff, 2012)

Mitch Resnick también sobre código-alfabetización, indica: “es un cambio radical para la informática... para muchos padres la codificación se parece menos a una actividad extracurricular y más a una habilidad básica de la vida, que podría algún día llevar a un gran trabajo o incluso a riquezas instantáneas. La difusión de la instrucción de codificación, aunque todavía naciente, es sin precedentes - nunca ha habido un movimiento tan rápido en la educación”. (Richtel, 2014)

2.2.4 LA HORA DEL CÓDIGO, EVENTO MUNDIAL

(‘The Hour of Code’) es el evento mundial más grande cuyo objetivo es la promoción del aprendizaje de la programación en las escuelas de nivel básico. El evento es promovido por la fundación de estadounidense Code.org, cuyo

lema es: “Every student in every school should have the opportunity to learn Computer Science” (“Cada estudiante, en cada escuela, debería tener la oportunidad de aprender Ciencias de la Computación”).

“La programación de computadoras debe ser enseñada en todas las escuelas, dijo Hadi Partovi, fundador de Code.org y ex ejecutivo de Microsoft. Lo llamó tan esencial como aprender sobre la gravedad o las moléculas, la electricidad o la fotosíntesis”.(Richtel, 2014)

Al acceder a su portal, encontramos algunas preguntas frecuentes como:

¿Qué es la hora del código?

“La Hora del Código es una introducción de una hora de duración a las Ciencias de la Computación, diseñada para mostrar que todo el mundo puede aprender a programar y así comprender los fundamentos básicos de la disciplina”.(Code.org, 2013)

¿Quién está detrás de la Hora de Programación?

“Está organizada por Code.org, una organización pública sin fines de lucro dedicada a promover la participación en escuelas e institutos de las Ciencias de la Computación, con una especial atención en incrementar el número de mujeres y estudiantes de colectivos minoritarios que aprenden a programar. Una coalición sin precedentes de socios que se ha reunido para apoyar la Hora del Código, también — incluyendo Microsoft, Apple, Amazon, Boys y Girls Clubs of America y College Board”.(Code.org, 2013)

2.2.5 PENSAMIENTO COMPUTACIONAL

2.2.5.1 ¿Qué es pensamiento computacional?

Existen muchos autores que desarrollan el concepto, pero principalmente se tiene a Jeannette Wing quien, lo define como: “El pensamiento computacional está adoptando un enfoque para resolver problemas, diseñar sistemas y comprender el comportamiento humano que se basa en conceptos fundamentales para la computación. El pensamiento computacional es una especie de pensamiento analítico. Comparte con el pensamiento matemático las maneras generales en las que podríamos acercarnos a la solución de un problema”. (Wing, 2006)

Considera Jeannette Wing que el pensamiento computacional tiene los siguientes factores a considerar:

- a) **Computación: abstracción y automatización.**- La esencia del pensamiento computacional es la abstracción. En computación, abstraemos las nociones más allá de las dimensiones físicas del tiempo y el espacio.
- b) **El pensamiento computacional en todas partes.**- El pensamiento computacional está influyendo en la investigación en casi todas las disciplinas, tanto en las ciencias como en las humanidades.
- c) **Pensamiento computacional para todos.**- Si el pensamiento computacional se añade al repertorio de habilidades de pensamiento, ¿cómo y cuándo deberían las personas aprender este tipo de pensamiento y cómo y cuándo debemos enseñarlo? Supongamos que la tendencia de utilizar el pensamiento computacional en la investigación en todos los campos ya está

ocurriendo, influyendo así en la formación de los estudiantes de posgrado.
(Wing, 2008)

El equipo Scratch del MIT define el Pensamiento Computacional como “un conjunto de conceptos, prácticas y perspectivas que se basan en las ideas del mundo de la informática. Los estudiantes al programar y compartir proyectos de Scratch, comienzan a desarrollarse como pensadores computacionales: aprenden conceptos básicos de computación y matemáticas, y a la vez también aprenden estrategias de diseño, resolución de problemas, y otras formas de colaboración”.(ScratchEd, 2011)

El objetivo del Pensamiento Computacional es “desarrollar sistemáticamente las habilidades de pensamiento crítico y resolución de problemas con base en los conceptos de la computación. En suma, potenciar el aprovechamiento del poder de cálculo que tienen las computadoras actualmente. Los estudiantes y profesionistas tendrán la necesidad de aprender y practicar las habilidades del PC para poder utilizar las nuevas tecnologías y confrontar los desafíos del Siglo XXI”.(Zapotecatl, 2014)

‘Computing at School’ (CAS) es una alianza estratégica formada por el Ministerio de Educación del Reino Unido, Barefoot’. En dicha comunidad ante la pregunta ¿Qué es el pensamiento computacional? Se responde:

“El pensamiento computacional consiste en mirar un problema de una manera que una computadora puede ayudarnos a resolverlo. Este es un proceso de dos pasos:

1. Primero, pensamos en los pasos necesarios para resolver un problema.
2. Luego, utilizamos nuestras habilidades técnicas para que la computadora trabaje en el problema.” (Barefoot, 2014)

Considerando el trabajo de Jorge Luis Zapotecatl, (Zapotecatl, 2014) “el pensamiento computacional ejercita sistemáticamente las habilidades de pensamiento crítico y resolución de problemas con base en los conceptos de la computación. La fórmula que nos permite recordar e identificar los componentes del PC es la siguiente”:

$$\text{pensamiento computacional} = \text{pensamiento crítico} + \text{poder de la computación}$$

“El primer componente del PC es el pensamiento crítico. La Fundación para el Pensamiento Crítico define al pensamiento crítico como: el modo de pensar (sobre cualquier tema, contenido o problema) en el cual el pensante mejora la calidad de su pensamiento al apoderarse de las estructuras inherentes del acto de pensar y al someterlas a estándares intelectuales”.(Zapotecatl, 2014)

(Barefoot, 2014) Indica sobre las características del pensamiento computacional que: “el pensamiento computacional no es pensar en computadoras. Las computadoras no piensan por sí mismas. ¡Todavía no, por lo menos! El pensamiento computacional implica 6 diferentes conceptos y 5 enfoques de trabajo: Este organismo plantea que el pensamiento computacional incluye 6 conceptos (‘lógica’, ‘algoritmos’, ‘descomposición’, ‘patrones’, ‘abstracción’, y ‘evaluación sistemática’) y 5 aproximaciones

(‘experimentación’, ‘creación’, ‘depuración’, ‘perseverancia’, y ‘colaboración’). Se detalla en la Figura N° 8.

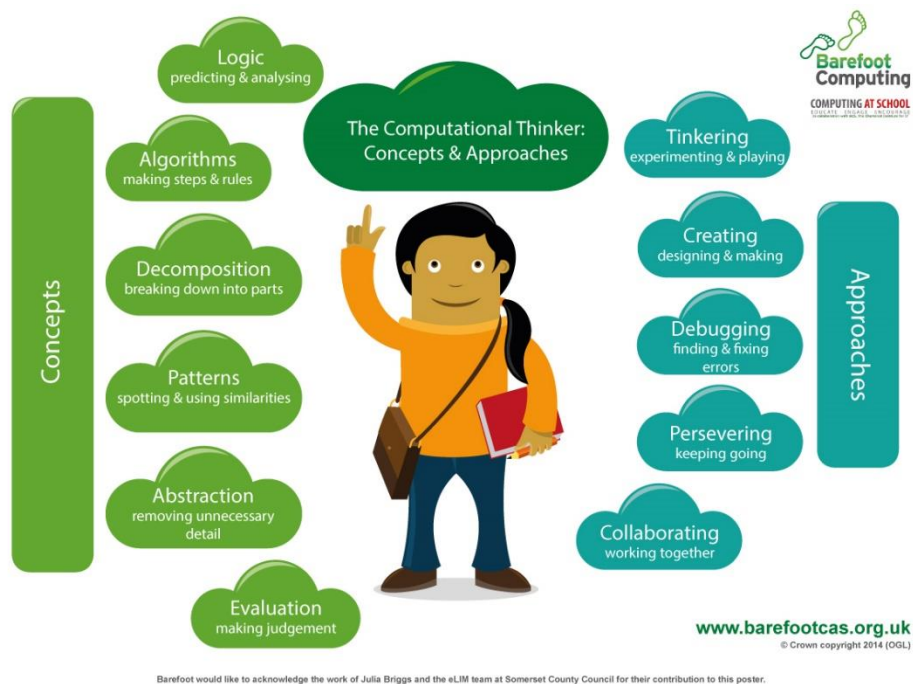


Figura 8: Marco-modelo de pensamiento computacional en el aula propuesto desde ‘CAS Barefoot’ (Barefoot, 2014)

Conceptos:

- **Lógica** (‘Logic’): el razonamiento lógico nos ayuda a explicar por qué algo sucede. Los ordenadores no hacen las cosas según les apetezca, ni trabajan diferente en función de cómo se sientan ese día. Esto significa que son previsibles.
- **Algoritmos** (‘Algorithms’): un algoritmo es una secuencia de instrucciones o conjunto de reglas para llevar algo a cabo.
- **Descomposición** (‘Decomposition’): el proceso de fraccionar un problema en sus partes constitutivas, más pequeñas y manejables, se conoce como

descomposición. La descomposición nos ayuda a resolver problemas complejos y a gestionar proyectos grandes.

- **Patrones** ('Patterns'): los patrones están por todas partes. Al identificar patrones podemos hacer predicciones, crear reglas y resolver problemas más generales.
- **Abstracción** ('Abstraction'): la abstracción se sitúa en el corazón del pensamiento computacional. La abstracción tiene que ver con simplificar las cosas; identificando qué es importante sin preocuparnos demasiado por lo anecdótico o irrelevante.
- **Evaluación sistemática** ('Evaluation'): la evaluación tiene que ver con hacer juicios, de una manera objetiva y sistemática cuando sea posible, hacemos juicios sobre qué hacer y qué pensar basados en una variedad de factores y criterios. En Ciencias de la Computación, la evaluación es sistemática y rigurosa; tiene que ver con juzgar la calidad, la efectividad y la eficiencia de las soluciones, sistemas, productos y procesos.

Aproximaciones:

- **Experimentación** ('Tinkering'): Significa probar a hacer (activamente) las cosas. Para los niños pequeños es la forma natural de aprender sobre algo: el juego espontáneo, la exploración y la experimentación.
- **Creación** ('Creating'): crear tiene que ver con planificar, hacer y evaluar cosas (p.e. animaciones, juegos o robots). Programar es un proceso creativo. El trabajo creativo implica tanto originalidad como la generación de un producto valioso: típicamente algo que es útil o que encaja con el propósito previsto.
- **Depuración** ('Debugging'): los errores en un algoritmo, programa o código se denominan 'bugs', y el proceso de encontrarlos y arreglarlos se denomina

‘debugging’ (puede traducirse al español como ‘depuración’). En la vida cotidiana, depuramos todo el tiempo: sencillamente es darse cuenta de los errores y solucionarlos.

- **Perseverancia** (‘Persevering’): la programación informática es una tarea dura. Esto es parte de su encanto: escribir código elegante y eficaz es un reto intelectual que requiere, no sólo una comprensión algorítmica del problema a resolver y codificar (y un conocimiento del lenguaje de programación utilizado para ello), sino también la disposición a perseverar sobre algo que es a menudo difícil y frustrante.
- **Colaboración** (‘Collaborating’): colaborar significa trabajar con otros para asegurar un mejor resultado. Es complicado pensar en cualquier trabajo o actividad de ocio en la sociedad digital actual que no involucre la colaboración. El trabajo colaborativo tiene además una larga tradición en nuestro sistema educativo (especialmente en la Educación Primaria), y la computación no debería suponer un cambio al respecto. (Román, 2016)

2.2.5.2 Criterios para evaluar el pensamiento computacional

Al ser el pensamiento computacional un tema nuevo, no existe un marco teórico consensuado donde se considere los criterios para medirlo, por lo cual mostramos las experiencias existentes en este punto.

El portal Dr. Scratch es una aplicación web libre y de código abierto que permite analizar de forma simple y automática proyectos programados con Scratch, así como obtener retroalimentación. “Dr. Scratch es una herramienta

desarrollada por el grupo de investigación GSyC/LibreSoft de la Universidad Rey Juan Carlos. Para analizar un proyecto con Dr. Scratch puede subirse a la web de la herramienta un archivo de extensión .sb o .sb2, ya que la herramienta soporta tanto proyectos de la versión Scratch 1.4 como 2.0, o bien se puede copiar-pegar directamente la URL del proyecto Scratch a analizar”. (Scratch, 2017)

Tabla 1: Criterios que aplica Dr. Scratch para medir el nivel de competencia en las distintas dimensiones del PC

Dimensión del PC	Nivel de competencia			
	Ninguno (0)	Básico (1 punto)	Intermedio (2 puntos)	Avanzado (3puntos)
Abstracción y descomposición de problemas	-	Más de un programa y más de un objeto	Definición de bloques propios (funciones)	Uso de clones
Paralelismo	-	Dos programas en bandera verde	Dos programas en “teclas presionada”, dos programas en “al presionar” el mismo objeto	Dos programas en “cuando reciba mensaje”, crear clon, dos programas en “cuando %s es > %”, dos programas en “cuando el escenario cambia a”
Pensamiento lógico	-	Si (if)	Si/Sino (If/else)	Operaciones lógicas
Sincronización	-	Esperar	Enviar, cuando reciba mensaje, parar todos, parar programas, parar	Esperar hasta, cuando el escenario cambia a, enviar y esperar

			programas del objeto	
Control de flujo	-	Secuencia de bloques	Repetir, por siempre	Repetir hasta
Interactividad con el usuario	-	Bandera verde	Tecla presionada, objeto presionado, preguntar y esperar, bloques de operaciones con ratón	Cuando %s es > %s, video, audio
Representación de la información	-	Modificadores de propiedades de objetos	Operaciones con variables	Operaciones con listas

Considerando a Fernando Posada en su Blog CanalTIC.com, Uso educativo de las TIC, sobre el pensamiento computacional y que criterios tener para evaluarlo considera que:

- “Formular problemas de manera que permitan usar computadoras y otras herramientas para solucionarlos.
- Organizar datos de manera lógica y darle sentido a los datos para establecer patrones y sacar conclusiones.
- Representar datos mediante la descomposición del problema en partes más pequeñas y la definición de abstracciones: modelos y simulaciones.
- Automatizar soluciones mediante el diseño creativo de un algoritmo (secuencia ordenada de pasos para resolver el problema.)
- Codificar el algoritmo para aplicarlo: implementación, documentación, etiquetas, módulos, etc.
- Depurar y resolver los errores que pudiera contener.

- Identificar, analizar e implementar posibles soluciones con el objeto de encontrar la combinación de pasos y recursos más eficiente y efectiva.
- Generalizar y transferir ese proceso de solución de problemas a una diversidad de estos”. (Posada, 2016)

Dichas habilidades se pueden potenciar si se trabajan además una serie de actitudes:

- “Confianza en el manejo de la complejidad.
- Persistencia al trabajar en problemas difíciles.
- Tolerancia a la ambigüedad.
- Habilidad para enfrentarse a problemas no estructurados.
- Habilidad para comunicarse y trabajar con otros para alcanzar una solución común”.(Posada, 2016)

2.3 MARCO CONCEPTUAL

Modelo

Es una representación de la realidad implementado con el fin de estudiarlo, es una abstracción de la realidad por lo cual, no es necesario considerar todos los detalles, entonces, el modelo no sólo es un reemplazo de la realidad sino también una simplificación de ella.

Holismo

Teoría filosófica, basada en una posición metodológica y epistemológica, que plantea que todo sistema ya sea físico, biológico, social, etc. y sus propiedades deben ser estudiados tomando en cuenta todos sus elementos y no solamente a través del estudio de las partes que lo componen. Se debe considerar una visión global que parte del todo, para luego ver las partes en sus interacciones entre los componentes y su entorno.

Simulación

Es el proceso de generar modelos de un sistema real y realizar experimentos sobre este modelo, con la finalidad de buscar comprender su comportamiento en cierto contexto y evaluar estrategias para realizar operaciones sobre ella.

Enfoque sistémico

Es la aplicación de la teoría de sistemas en el estudio de un fenómeno u objeto considerándolo como sistema, por lo cual, se establece los componentes que se interrelacionan, el entorno del sistema, los límites que definen las entradas y salidas, y la identificación de los subsistemas y suprasistemas.

Código-alfabetización

Es ser capaz de leer y escribir en el lenguaje de los ordenadores y otras máquinas, y pensar computacionalmente. Aprender cómo se desarrolla el software y cómo funciona realmente el ecosistema de la tecnología informática nos ayuda a entender los nuevos modelos a través de los cuales trabajaremos y viviremos como sociedad.

Pensamiento computacional

Es una competencia transversal en la actualidad, que debe impartirse en todas las personas, considera un enfoque para plantear problemas cuyas soluciones sean representadas mediante una secuencia de instrucciones y algoritmos.

Aplicativo de Software

Es el elemento lógico de un sistema de cómputo, que comprende las instrucciones o algoritmos implementadas en un lenguaje de programación y que le da la funcionalidad al elemento físico llamado también hardware en un sistema de cómputo.

Educación primaria

Es el segundo nivel de la Educación Básica Regular en el Perú, comprende desde los 6 hasta los 11 años de edad, estructurado del primer al sexto grado; tiene como finalidad la formación integral del educando, lo cual incluye el desarrollo de la lecto-escritura, el pensamiento lógico matemático, desarrollo personal y espiritual, etc.

Evaluación

“La evaluación pedagógica consiste en la valoración, a partir de criterios y referencias preespecificados, de la formación técnicamente diseñada y sistemáticamente recogida y organizada, sobre cuantos factores relevantes integran los procesos educativos para facilitar la toma de decisiones de mejora”.(Pérez, 2002)

Prototipo

Es un modelo inicial básico de un software o sistema final, sirve para proporcionar indicadores de la funcionalidad y permiten tener una retroalimentación para su implementación final, es un modelo a escala del software final, pero no lleva a cabo la totalidad de funciones necesarias del software.

CAPÍTULO III

MÉTODO

3.1 TIPO DE INVESTIGACIÓN

El presente trabajo de investigación, “de acuerdo a sus características se enmarca dentro del tipo de investigación cuantitativo”(Hernández, Fernández, & Baptista, 2014) cuya característica principal es probar la hipótesis en base a la medición numérica y el análisis estadístico; además, corresponde al tipo de estudio experimental, puesto que se busca experimentar el material diseñado. El nivel de investigación es el explicativo, ya que se desea explicar el comportamiento de una variable en función de otra.

DISEÑO DE INVESTIGACIÓN

El diseño del presente proyecto es cuasi – experimental.

La fórmula es la siguiente:

G1: O1 X O2

Donde:

G1: Grupo experimental.

X: Tratamiento con el aplicativo de software basado en el modelo de código-alfabetización.

O1: Prueba (PRETEST) antes del experimento.

O2: Prueba (POSTEST) con el experimento

3.2 POBLACIÓN Y MUESTRA

Para la experimentación se considera como población a los estudiantes del nivel primario de la I.E.P. “71013 Glorioso San Carlos” de la ciudad de Puno.

Tabla 2: Distribución de Estudiantes de la I.E.P. “71013 Glorioso San Carlos”

– Puno 2018

Secciones	Número de estudiantes	Porcentaje (%)
Primer grado	150	18
Segundo Grado	148	17
Tercer Grado	146	16
Cuarto Grado	147	17
Quinto Grado	145	16
Sexto Grado	144	16
Total	880	100

Fuente: Reporte UGEL Puno

Elaboración: El ejecutor

MUESTRA

Para la muestra se considera los siguientes criterios:

Criterios de Inclusión:

- Alumnos matriculados durante el año académico 2018.
- Alumnos que tienen desarrollado la habilidad de leer y escribir.
- Alumnos que difieren de edad en 3 años.

Criterios de exclusión:

- Alumnos con limitaciones senso-perceptivas que les impiden utilizar computadoras.
- Alumnos que no cuenten con la autorización de los apoderados, por ser una actividad extracurricular.

Por lo cual, se considera a los estudiantes del tercero al quinto grado y aplicando el tamaño de la muestra en base a $N = 438$, nivel de confianza del 95%, entonces $Z = 1,96$; teniendo un error no mayor a 3%, tenemos que $e = 0,03$ y considerando $pq = (0,50)(0,50) = 0,25$.

$$n = \frac{N}{1 + \frac{e^2(N - 1)}{Z^2 pq}}$$

Se tiene que $n = 311$ estudiantes.

3.3 OPERACIONALIZACIÓN DE VARIABLES

Tabla 3: Operacionalización de las variables

VARIABLES	DIMENSIONES	INDICADORES	VALORES
VARIABLE INDEPENDIENTE Modelo holístico de código-alfabetización.	Modelo de código alfabetización	Aplicación de un Enfoque Sistémico	<ul style="list-style-type: none">• Sí• No
		Aplicación de un enfoque cibernético	<ul style="list-style-type: none">• Sí• No
		Aplicación de conceptos de autopoiesis	<ul style="list-style-type: none">• Sí• No
		Aplicación de modelo socio constructivista de la educación	<ul style="list-style-type: none">• Sí• No
		Aplicación de un modelo construccionista de la educación	<ul style="list-style-type: none">• Sí• No

	Prototipo de software con el modelo	<p>Funcionalidad</p> <p>Fiabilidad</p> <p>Usabilidad</p> <p>Eficiencia</p>	<ul style="list-style-type: none"> • Muy Bueno • Bueno • Regular • Malo • Muy malo
<p>VARIABLE DEPENDIENTE</p> <p>Nivel de desarrollo del Pensamiento Computacional</p>	<p>Capacidades a desarrollar en el pensamiento computacional</p>	<p>Formular problemas y darle solución usando el computador o usando otras herramientas.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
		<p>Establecer patrones y obtener conclusiones, mediante la organización lógica de los datos.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
		<p>Descomponer un problema en subproblemas definiendo abstracciones (modelos y simulaciones) para representar datos.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
		<p>Implementar soluciones a los problemas en forma creativa mediante el diseño de algoritmos.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
		<p>Depurar y levantar los errores que se pudieron presentar en su codificación.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
		<p>Optimizar los recursos y los algoritmos mediante la identificación, análisis e implementación de posibles soluciones a los problemas.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
		<p>Generalizar y realizar la transferencia de los procesos encontrados a problemas diversos.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
	<p>Actitudes a desarrollar en el pensamiento computacional</p>	<p>Confianza en el manejo de la complejidad.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
		<p>Persistencia al trabajar en problemas de mayor nivel de complejidad.</p>	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
			<ul style="list-style-type: none"> • Excelente

		Tener tolerancia a la ambigüedad.	<ul style="list-style-type: none"> • Satisfactorio • Puede mejorar • Inadecuado
		Habilidad para afrontar problemas no estructurados.	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado
		Habilidad para realizar trabajo en equipo logrando alcanzar soluciones comunes.	<ul style="list-style-type: none"> • Excelente • Satisfactorio • Puede mejorar • Inadecuado

3.4 INSTRUMENTOS

Técnicas de recolección de datos

Para el recojo de la información se hace uso de la encuesta, para verificar el nivel de aceptación del modelo holístico y del prototipo de software a cargo de los expertos. También se hace uso de la técnica del test, mediante el cual se verifica el nivel de desarrollo del pensamiento computacional por parte de los niños de la I.E.P. “71013 Glorioso San Carlos” de la ciudad de Puno, el test está conformado por dos etapas, la prueba teórica desarrollado en entorno Moodle, donde se tiene preguntas de tipo IBM y de alternativa múltiple, con un tiempo determinado; y la prueba de ejecución, la cual es un caso de estudio a resolver en forma grupal.

3.5 PROCEDIMIENTOS

Se considera el desarrollo del modelo holístico de código –alfabetización basado en conceptos de sistemas y paradigmas educativos como una primera etapa, para luego implementar el aplicativo de software, el cual se aplica en estudiantes considerados en la muestra.

El análisis estadístico se realizó con el programa computacional SPSS (Statistical Package for Social Sciences), los estadísticos que se emplean teniendo en cuenta las características de la muestra y el nivel de las variables son: la Media aritmética, Desviación estándar y la Prueba normal “Z” para realizar la prueba de hipótesis.

3.6 ANÁLISIS DE DATOS

Para la prueba de hipótesis se realiza lo siguiente: se formulan las hipótesis nula y alterna, se establece el nivel de significancia, para nuestro caso es de 5%, luego se realiza el cálculo estadístico, el estadístico es Z normal y finalmente se realiza la toma de decisión en base a la comparación de los datos obtenidos por la Z calculada y la Z crítica.

Como se muestra:

- Hipótesis Nula: H_0 :
- Hipótesis Alterna H_a :
- Nivel de significancia

$$\alpha = 0,05 \text{ ó } 5\%$$

- Región crítica

$$\text{Con } Z_\alpha = Z_{0.05} = 1,64$$

- **Cálculo estadístico**

$$Z_c = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Donde:

Z_c = Z calculada

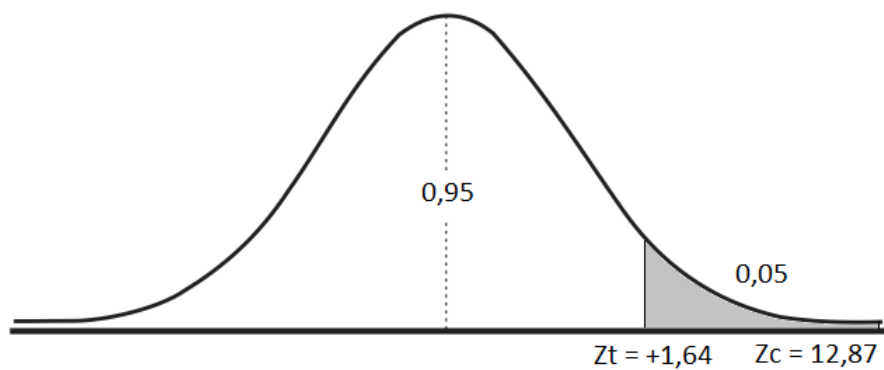
\bar{X} = Media aritmética de la muestra

μ = Valor 15, escogido según los antecedentes

σ = Desviación estándar

n = número de estudiantes de la muestra

- **Toma de decisión**



"Z" calculada > "Z" crítica
SE RECHAZA H_0

CAPÍTULO IV

RESULTADOS

4.1 MODELO HOLÍSTICO DE CÓDIGO-ALFABETIZACIÓN PARA NIÑOS DE EDUCACIÓN PRIMARIA

a) ANTECEDENTES

Conceptos holísticos

Para implementar el modelo basado en un enfoque holístico es necesario tener en cuenta los conceptos centrales en los cuales tiene que fundamentarse, estos son los siguientes:

- **Sistema.** Un sistema considerando a diferentes autores se puede resumir su definición en lo siguiente:

Un sistema es:

- Una unidad compleja conformada por un conjunto de componentes o elementos organizados y relacionados.
- Estos componentes interactúan entre sí para lograr un objetivo o funcionalidad.
- Reciben y/o brindan datos, energía o materia al contexto en el que se encuentran a través de los límites del sistema.

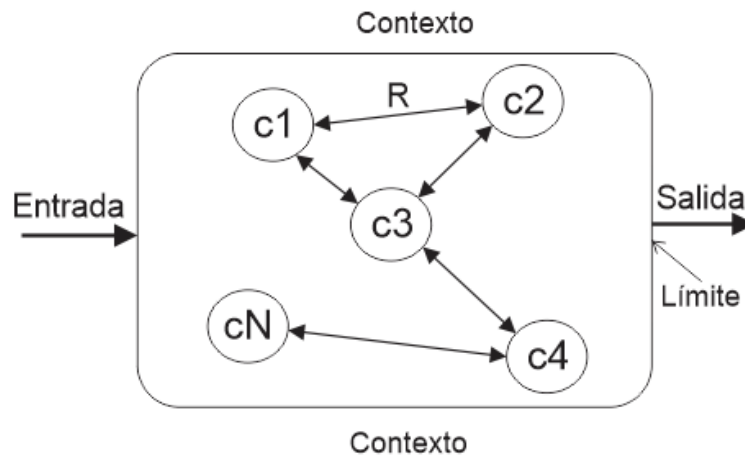


Figura 9: Representación de un sistema

Los elementos que se considera en la definición son:

Los **componentes** o elementos. Constituyen las partes del sistema, pudiendo ser estos objetos o procesos; considerando en el primer caso se consideran las entidades físicas, las unidades espacio-temporales concretas que son consideradas las variables de estado del sistema.

Contexto es el nombre que recibe toda aquella variable que se encuentra fuera del sistema estudiado; en el cual el sistema no influye, pero la variable si puede influir en el funcionamiento u objetivo del sistema.

Los **límites** determinan el sistema abarcado, es aquella frontera que separa el sistema de su contexto, definiendo las variables que forman parte del sistema de las que se quedan fuera.

Las **relaciones** entre componentes son las interacciones entre los elementos que conforman el sistema, compartiendo información, energía o materia, en base a estas interrelaciones se consigue las propiedades únicas del sistema y su carácter de unicidad.

Las **entradas** (inputs) y las **salidas** (outputs) representan el relacionamiento del sistema con el contexto. Corresponden respectivamente a los estímulos y las respuestas procesadas por el sistema. Donde las salidas de un sistema se transforman en la entrada de otro sistema, que realiza el procesamiento en caja negra, para volverla como otra salida, convirtiéndose en un ciclo.

- **Cibernética de 2do orden.**

Es importante para nuestro modelo considerar la cibernética de segundo orden las funciones de comunicación y control entendidos éstos como fenómenos externos e internos del sistema. Además, tener en cuenta el proceso de retroalimentación, que mediante los diferentes tipos de evaluaciones genera la mejora continua del proceso educativo. Otro factor a considerar es que el observador se comporta como parte de lo observado modificando dentro del sistema su objetivo o finalidad.

- **Autopoiesis**

Para el caso de su aplicación en un sujeto dentro de un sistema educativo es importante pasar del concepto biológico de la autopoiesis a lo sociológico y antropológico, donde el proceso autopiético creativo le permite al sujeto cambiar, mutar y transformarse constantemente, pero a su vez volver a reflexionar sobre lo sucedido al respecto.

. Modelos Educativos

Constructivismo

En la construcción de nuestro modelo holístico es importante considerar este modelo educativo, ya que el proceso de enseñanza-aprendizaje de la programación debe

considerar que el estudiante sea el creador de sus experiencias de conocimiento, y en un entorno donde se tiene docentes facilitadores, compañeros con mayor o menor habilidad, donde se aprenda o enseñe de sus pares, pero sobre todo se participa de un proceso activo de creación.

Triángulo Instruccional, en el cual se observa la interacción de cuatro elementos: el alumno, el profesor, los contenidos y los objetivos; que se ubican en los vértices. Además, en los lados uniendo los vértices se ubican, la construcción de significados, uniendo al alumno con el contenido, la atribución de sentido relacionando al alumno con los objetivos y las competencias uniendo los contenidos con los objetivos. Se tiene también que el profesor se ubica en el baricentro del triángulo, es decir, en el centro de gravedad, en el modelo holísticos el profesor es el encargado del proceso de enseñanza, quien equilibra las relaciones entre los otros elementos, considerando también el proceso de retroalimentación mediante la evaluación.

Construccionismo

Es necesario que los estudiantes y docentes dentro de nuestro modelo holístico apliquen un enfoque construccionista, ya que su aprendizaje mejora cuando ellos mismos construyen productos que tenga significado y valor para ellos individual y grupalmente. Por ejemplo, la importancia que tiene que un niño o grupo obtenga un juego implementado con Scratch y Makey Makey, que luego es socializado y utilizado por sus compañeros de su salón y de otros salones. Además, el trabajar en equipos donde hay compañeros un poco mayores o menores en edad o los que tienen mayor habilidad en los programas o no, pero que todos aportan; saber tolerar el error,

ya que no siempre un algoritmo será correcto en la primera implementación y la perseverancia en el logro de un producto.

b) MODELO HOLÍSTICO DE CÓDIGO-ALFABETIZACIÓN

. Consideraciones

El modelo holístico de código-alfabetización, considera en sus fundamentos los conceptos de sistema, general que se desprende de la Teoría de Sistemas, en donde mediante una visión global de unicidad de los entes se modela y plasma los sistemas que corresponden a una abstracción de la realidad enfocándolo desde una visión holística. Todo sistema tiene como características centrales: la emergencia, jerarquía, comunicación y control que lo definen y forman su identidad. Además, dos corrientes desarrolladas juntamente con la Teoría de Sistemas son la cibernética, sobre todo la de segundo orden, que consideran la forma de gobernar los sistemas teniendo como características centrales la comunicación y control de los elementos del sistema y la Autopoiesis, que es la propiedad básica de los seres vivos, donde cuando una variable externa incide sobre ellos, los resultados dependen del sistema mismo, de su estructura en ese momento y no de lo externo; El proceso de enseñanza-aprendizaje es un proceso donde seres vivos son sistemas dinámicos y cíclicos, que en base a influencias de su entorno autoproducen cambios internos y mediante procesos de retroalimentación van mejorando y autorregulando sus resultados.

Desde el punto de vista pedagógico, se considera dos corrientes pedagógicas de actualidad y que en sus fundamentos, cuentan con varios aportes de pensadores sistémicos como son el constructivismo, sobre todo el socio constructivismo y el construccionismo. El constructivismo propone un enfoque donde el proceso de enseñanza-aprendizaje se percibe como un proceso dinámico e interactivo del

estudiante, de modo que el conocimiento es un proceso auténtico de construcción operada por la persona que aprende. Considerando además a Vygotsky, indica la importancia de la interacción social, demostrando que el estudiante aprende más eficazmente cuando lo hacen forma cooperativa. El construccionismo, a la par, aporta su afán por que el estudiante construya un producto real pero significativo para él y su entorno, que las actividades de confección o construcción de artefactos, sean estos el diseño de un producto, la construcción de un juego en computadora, son facilitadoras del aprendizaje; pone mucho énfasis en el uso de los recursos computacionales, aprovechando el inmenso bagaje de posibilidades que se tiene con el uso del computador, los software especializados, y la programación de las computadoras.

El modelo propuesto se presenta a continuación:

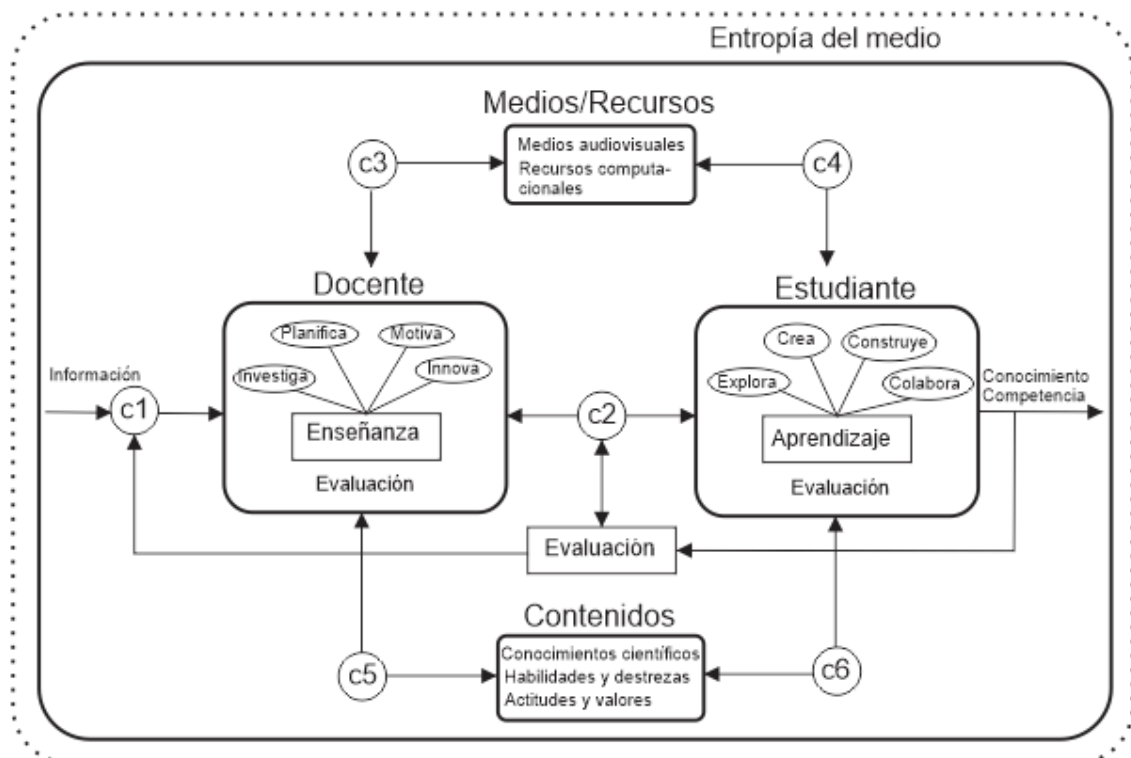


Figura 10: Modelo holista de Código-alfabetización propuesto

Elaboración propia

Descripción del modelo

a) Componentes

Considerando el modelo como un sistema se tiene los componentes del sistema y son los principales elementos del sistema que interactúan entre si compartiendo información y control.

- **Docente**, elemento encargado del proceso de enseñanza, tiene como función principal el de planificar el proceso de enseñanza aprendizaje, partiendo de una investigación profunda de los contenidos, estrategias didácticas, recursos - materiales y de los avances de la tecnología. Para dirigir el proceso de enseñanza es imprescindible tener presente el factor motivacional de los estudiantes y mucho tiene que ver las dotes de innovador en este proceso.
- **Estudiante**, responsable del proceso de aprendizaje, tiene como actividades a realizar la exploración de su medio, basado en un conjunto de contenidos de actualidad que inciten a aprender, aplicar la creatividad en la construcción de objetos reales que tengan sentido para él, sus pares y su medio, que solucionen problemas reales, logrando que sea constructor de su aprendizaje y que tenga significado para él. Además, se debe propiciar en todo momento el trabajo colaborativo, en un medio donde exista diversidad en edades, intereses, horarios y no encerrados en un salón y un colegio.
- **Contenidos**, para el caso de la código-alfabetización, debe ser un contenido estructurado, desde lo simple a lo complejo, que muestre desde el saber arrastrar y soltar, propio de la programación basada en bloques y el armar rompecabezas, que debe ser la forma de llevar al estudiante a escribir código, hasta saber codificar la forma de almacenar la información y luego incursionar en el concepto de algoritmo, donde se debe mostrar las diversas estructuras de la programación como: tipos de

datos, estructuras selectivas, estructuras condicionales, estructuras iterativas y funciones. Tomando en cuenta la edad de los niños debe aplicarse conceptos de la gamificación que motiven a desarrollar proyectos reales.

- **Medios/Recursos**, son necesarios los medios que brindan diferentes posibilidades de codificar la realidad y determinantes del proceso de enseñanza-aprendizaje de calidad, por medio de ellos se les ofrecen a los alumnos un mayor bagaje de experiencias, mayores posibilidades de creatividad y una interacción más directa con la realidad. El hardware y el software es por excelencia el recurso a utilizar en la código-alfabetización, con diferentes aplicaciones de software que existen y el hardware mediante las placas de electrónica que también se tienen toda una variedad, como las placas Makey Makey, Arduino, etc. Pero no solo los audiovisuales como el computador, sino también recursos que permitan construir y ejecutar algoritmos en su medio real mediante: tarjetas, fichas, dibujos, cartulinas, etc.

b) Relaciones

En un modelo de sistemas las interacciones entre los componentes de un sistema son los que permiten la sinergia, entre los elementos y muestran la integridad que existe en la palabra sistema. Las relaciones se hacen visibles mediante dos conceptos comunicación y control, existe intercambio de información y energía entre docente y estudiante, mediante la comunicación de estos dos elementos se observa la capacidad de transmitir mensajes y de reaccionar frente a ellos eso da sentido al proceso de enseñanza – aprendizaje que es el resultado de esta interacción. La comunicación entre docente y estudiantes se visualiza en la función de facilitador y orientador del profesor de acuerdo con las exigencias sociales y significa guiar el proceso de aprendizaje del estudiante hacia la construcción autónoma de conocimientos, habilidades y valores.

El control, el segundo concepto de la cibernética, significa en esencia que un sistema que inicialmente no era estable, puede hacerse asintóticamente estable gracias a la introducción de un controlador, que contrarresta la desviación del sistema con respecto al estado estable. Este factor por tanto es el que genera la homeostasis en un sistema y como tal en nuestro sistema se tienen niveles de controles que permiten en base a la retroalimentación, lograr el equilibrio en el sistema mediante las sumas y restas sinérgicas que se producen entre las entradas y la retroalimentación principalmente los resultados de las evaluaciones.

d) Retroalimentación

Se utiliza en nuestro modelo como el proceso por el cual, el sistema obtiene y procesa información acerca de las funciones que ejecuta para generar acciones correctivas, preventivas o de optimización. Esta retroalimentación permite el control del sistema y que el mismo sistema tome medidas de corrección, en base a la información que se obtiene de las evaluaciones que son: evaluación de entrada, evaluación de proceso, las autoevaluaciones y las sumativas al finalizar los módulos. Además, el docente va realizando una verificación de los contenidos y de los medios y recursos que se utiliza con la finalidad de lograr el objetivo del sistema el lograr un aprendizaje óptimo de código-alfabetización de los estudiantes.

4.2 PROTOTIPO DE SOFTWARE BASADO EN AL MODELO HOLÍSTICO DE CÓDIGO-ALFABETIZACIÓN

Para el desarrollo del aplicativo se consideró la “Metodología Ágil de desarrollo de software”, un proceso es ágil cuando el desarrollo del software es:

- Incremental, es decir se tiene entregas pequeñas de software con ciclos rápidos.
- Cooperativo, donde clientes y desarrolladores trabajan juntos constantemente cara a cara y una comunicación cercana.
- Sencillo, pues el método en sí mismo es fácil de aprender y modificar, permitiendo realizar cambios de último momento.

Sus elementos claves son:

- Poca documentación
- Simplicidad
- Análisis como una actividad constante
- Diseño evolutivo
- Pruebas diarias y constantes

Metodología ágil KANBAN, es una palabra japonesa que significa “tarjeta visual”, En Toyota es el término usado para el sistema de señalización visual y físico que une a todo el sistema de producción Lean.

“Kanban se basa en una idea muy simple: el Trabajo en Curso (Work In Progress, WIP) debería limitarse y solo debería empezarse con algún trabajo nuevo cuando un bloque de trabajo anterior haya sido entregado o ha pasado a otra función posterior de la cadena”.(Kniberg & Skarin, 2010)

Dominio del prototipo de software

El prototipo de software de Código-Alfabetización, está dirigido a niños del tercer al quinto grado de primaria, es decir niños entre las edades de 8 a 10 años de edad, de la Institución Educativa Primaria “71013 Glorioso San Carlos” de la ciudad de Puno. Los usuarios a los que se dirige el aplicativo es a los estudiantes, quienes interactúan con el sistema para poder desarrollar el pensamiento computacional, poniendo en práctica los conceptos de programación tanto en forma grupal como individual, estas actividades van desde conceptos de arrastrar soltar, para practicar la programación de bloques, luego los conceptos almacenamiento de los datos en un computador, conociendo la forma de codificar los bits (1 y 0), luego la implementación de algoritmos, donde se aplica las instrucciones básicas de adelante, atrás, izquierda, derecha en diversas situaciones, para luego hacer más específicas las instrucciones. En un segundo módulo se realiza la práctica de conceptos de programación como estructuras secuenciales, condicionales e iterativas, en el entorno de programación de bloques Scratch. En el módulo tercero se aplica todos los conocimientos anteriores combinándolos con módulos de Makey Makey y Arduino de tal manera que elaboran proyectos reales de acuerdo a su nivel.

Requerimientos funcionales

- **RF-1 Portal Web del proyecto:** El prototipo debe contar con un portal web que contenga la información general del proyecto, tener características de diseño responsive o adaptativo y permitir el acceso al entorno de trabajo.
- **RF-2 Interface de Logueo:** Contar con la opción de logueo, en donde se permite el registro del estudiante mediante un usuario y contraseña, además de información de sus datos personales, para estudiantes nuevos y el acceso para los usuarios

registrados. Se debe realizar las restricciones y controles al momento del llenado de la información en el formulario.

- **RF-3 Módulos del proyecto:** El contenido del desarrollo de las actividades debe ser dividido en 3 módulos, los cuales cuentan con actividades bien diferenciadas que cuentan con una interfaz similar para cada módulo: Actividades grupales en aula, Actividades interactivas con el entorno, Actividades de construcción creativa y Desarrollo del examen de módulo.
- **RF- 4 Actividades grupales en aula:** Interfaz gráfica con instrucciones sencillas y gráficos instructivos, que permiten desarrollar las actividades en el salón, formando grupos de trabajos y con materiales que deben estar en el aplicativo para poder ser descargados. Debe implementarse de acuerdo al diseño educativo elaborado por el ejecutor, en coordinación con el docente de aula de la institución educativa.
- **RF-5 Actividades interactivas con el entorno:** Actividades que se desarrollan en el aplicativo por parte del estudiante, teniendo actividades de crecimiento progresivo de complejidad y se destina puntajes al momento de ser superados. La navegación debe permitir desplazarse directamente por las diferentes actividades. Debe implementarse de acuerdo al diseño educativo elaborado por el ejecutor, en coordinación con el docente de aula de la institución educativa.
- **RF-6 Actividades de construcción creativa:** Interfaz gráfica sencilla y visual, que permite desarrollar actividades de construcción por parte de los estudiantes, trabajando en equipo, desarrollando la creatividad, posibilidad de subir el proyecto terminado para su publicación en el portal web. Debe implementarse de acuerdo al diseño educativo elaborado por el ejecutor, en coordinación con el docente de aula de la institución educativa.

- **RF-7 Desarrollo del examen de módulo:** Contar con opciones de paginación donde se muestre cada pregunta con sus alternativas en radio button, al finalizar mostrar el resultado y permitir volver a desarrollar el examen en caso de tener un puntaje menor o igual a 14. Debe implementarse de acuerdo al diseño educativo elaborado por el ejecutor, en coordinación con el docente de aula de la institución educativa.
- **RF-8 Elaboración de reportes:** El primer reporte es el diploma del estudiante al culminar satisfactoriamente cada módulo, diploma que muestra datos del estudiante y que se genera en formato pdf. Luego se debe tener para el docente reportes de avance y notas finales.

Requerimientos no funcionales

- **RNF-01 Desempeño:** El aplicativo tiene una naturaleza pública, por tanto, debe tener la capacidad de realizar las actividades sin errores y tener un espacio para consultas.
- **RNF-02 Interfaz de usuario:** Debido a que está dirigido básicamente a niños entre 8 y 10 años, debe permitir una interacción sencilla y amable promoviendo una interfaz gráfica intuitiva en ambientes WEB. Considerar el idioma español con la sencillez correspondiente y el uso de colores de acuerdo a las edades que se indica.
- **RNF-03 Seguridad:** Ingreso de usuarios a través de solicitud de usuario y contraseña con su respectiva validación. Almacenamiento cifrado de contraseñas y datos de sesión.

DESARROLLO DEL PROYECTO

I: Limite el WIP (Work In Progress - Trabajo en Curso)

Para el desarrollo del prototipo se considera la participación de 3 personas, dos programadores y un asesor educativo que verifique la funcionalidad y la aplicación de conceptos educativos, por lo tanto, se tiene la siguiente ecuación para calcular el WIP.

$$\mathbf{WIP = 2 (P - C)}$$

Donde:

P = Personas en el Equipo de desarrollo = 2

C = Grados de Colaboración = 1

Reemplazando en la ecuación se tiene:

$$\mathbf{WIP = 2 (2 - 1)}$$

$$\mathbf{WIP = 2}$$

Por tanto, se tiene que en el proyecto debe existir 2 como límite de tareas en progreso que podemos establecer en cada una de las columnas de nuestro KANBAN. Esto significa que no podremos tener en este estado nada más que 2 tareas y el equipo se tiene que preocupar en finalizarlas, ya que en caso contrario no podemos iniciar ninguna tarea adicional.

II: Visualice el flujo de trabajo

Para visualizar el flujo de trabajo y construir nuestro tablero KANBAN se hace uso de la herramienta Kanbanize, que se puede acceder en la dirección <https://kanbanize.com/>; donde, se tiene tres carriles diferenciados, los cuales son: **Requested** (requerido), donde se ubican las historias de usuario pendientes a elaborarse; **In Progress** (en proceso), donde se ubican las historias de usuarios que

están en proceso de elaboración o en trabajo y **Done** (terminado), donde se ubican las historias de usuario que han sido concluidas.

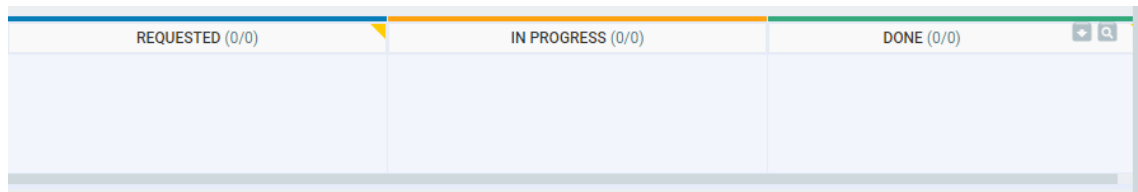


Figura 11: Tablero Kanban en la Herramienta Kanbanize

Al inicio del proyecto el tablero Kanban cuenta con 7 historias de usuarios, el cual se muestra a continuación en el carril de Requested, donde se aprecia en colores los responsables de la elaboración de las actividades de implementación.

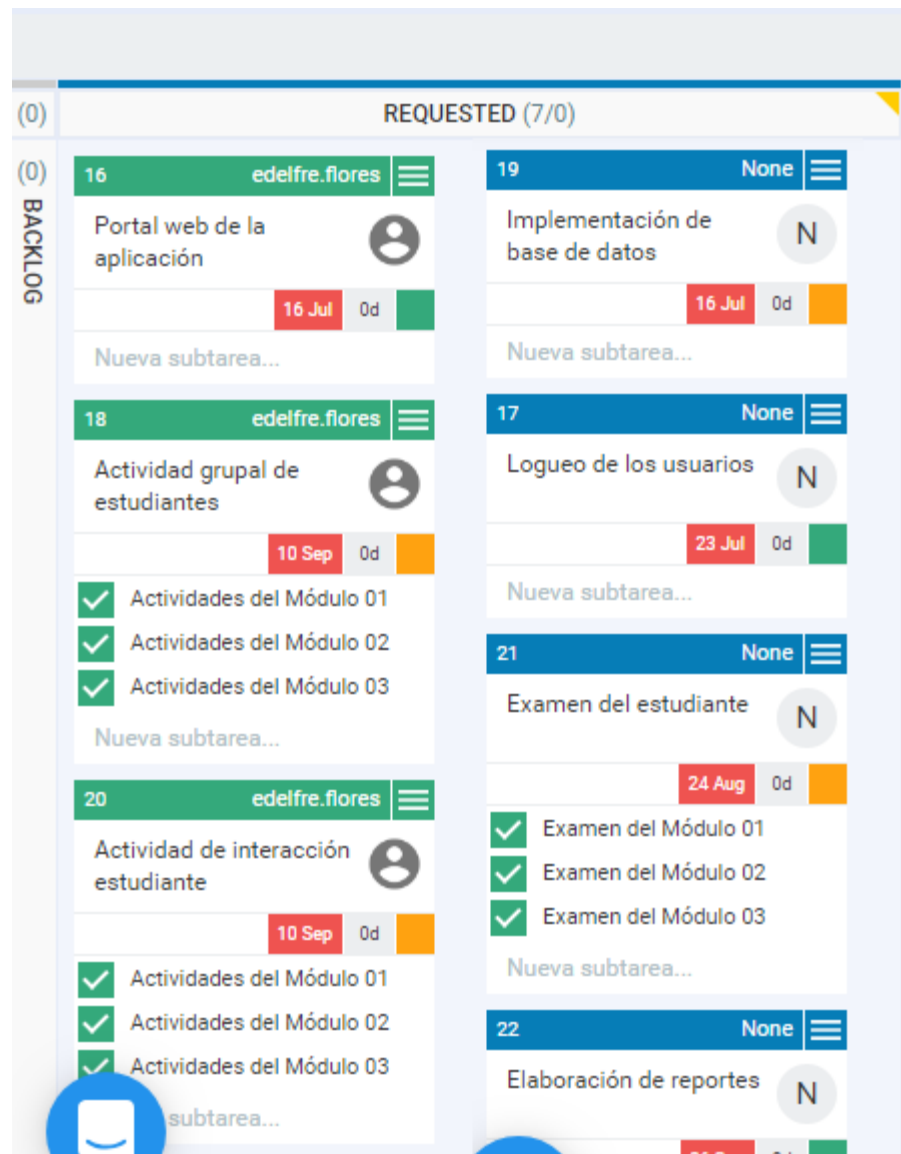


Figura 12: Tablero Kanban con procesos iniciales de Programadores Kid's

Flujo de trabajo 01:

El primer flujo de trabajo está destinado a realizar dos tareas en proceso, por los dos programadores, el desarrollo del portal web principal del proyecto y el diseño e implementación de la base de datos, estos dos procesos se realizan en forma concurrente y tienen tiempos asignados diferentes de 5 días y de 3 días. La implementación de la base de datos no incurre en mayor tiempo debido a que no es

una base de datos con muchas tablas. Mientras que el portal web principal tiene varios factores que cumplir.

Tablero Kanban



Figura 13: Tablero Kanban del flujo de trabajo 01 de Programadores Kid's

Tabla 4: HS1, Portal web principal

Nro. 01	Historia de usuario		
Descripción	Portal web principal		
Notas: El prototipo debe contar con un portal web que contenga la información general del proyecto, tener características de diseño responsive y permitir el acceso al entorno de trabajo.			
Esfuerzo Estimado	5 días	Esfuerzo resultante	7 días
Riesgo	Bajo	Valor para el cliente	10



Figura 14: Interfaz del portal web principal de Programadores Kid's

Tabla 5: HS2, Base de datos del proyecto

Nro. 02	Historia de usuario		
Descripción	Implementación de la base de datos del proyecto		
Notas: La base de datos del proyecto consta de un modelo Relacional para MySql y conta de todos los elementos que son parte del proyecto.			
Esfuerzo Estimado	3 días	Esfuerzo resultante	3 días
Riesgo	bajo	Valor para el cliente	0

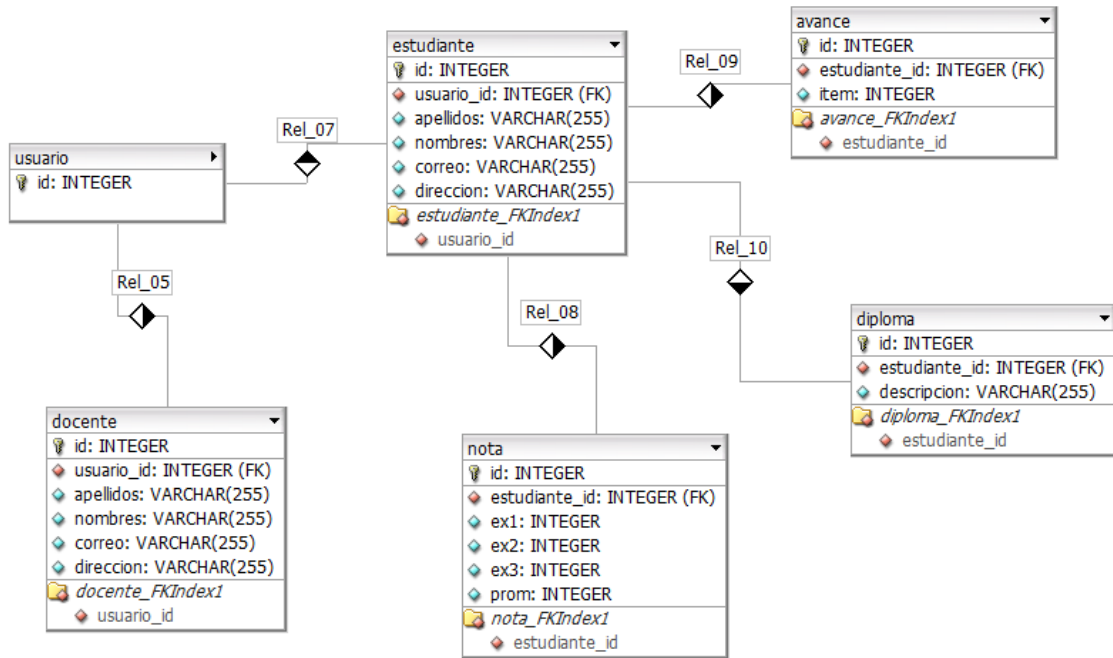


Figura 15: Diseño de la Base de datos

Flujo de trabajo 02:

Una vez concluido el desarrollo del portal web principal y la implementación de la base de datos, se divide nuevamente las tareas de los dos programadores, el cual es el desarrollo del logue de los usuarios, que consiste en implementar utilizando las herramientas como: lenguaje de programación PHP y Sistema manejador de base de datos MySQL, considerando las restricciones de formulario con JavaScript en el acceso de la información en los formularios. Se inicia también con la implementación de las actividades grupales del módulo 01, que consta de una interfaz gráfica separada en pestañas que se dividen en cuatro partes: Información, procedimientos iniciales, actividades y utilice la creatividad; todo esto de acuerdo a la planificación educativa elaborado por el ejecutor y el docente de aula.

Tablero Kanban

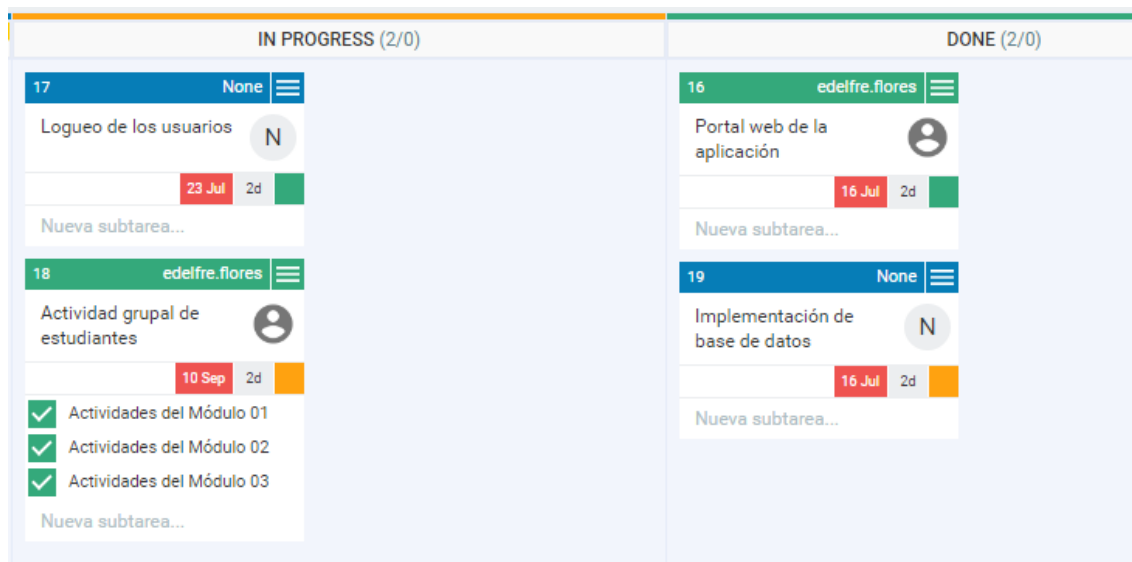
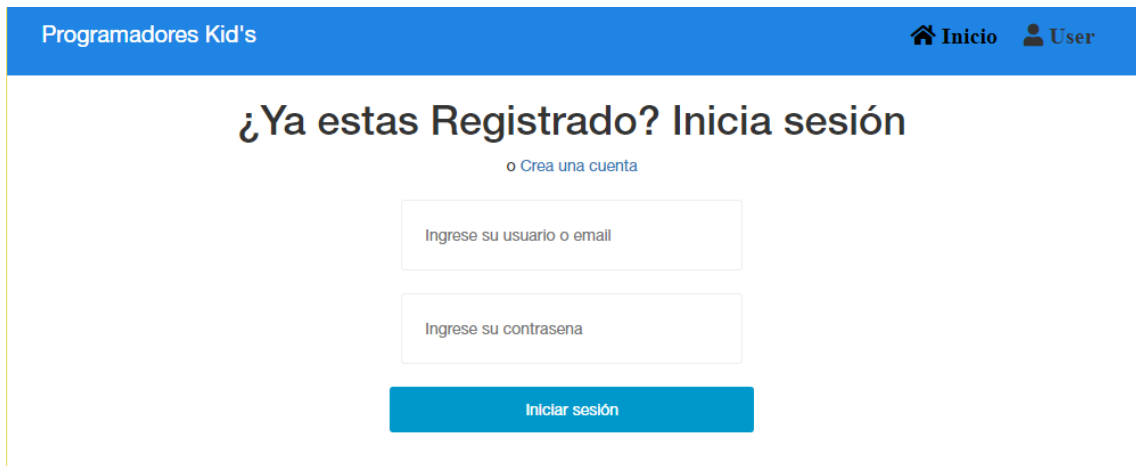


Figura 16: Tablero Kanban del flujo de trabajo 02 Programadores Kid's

Tabla 6: HS3, Logueo de usuario

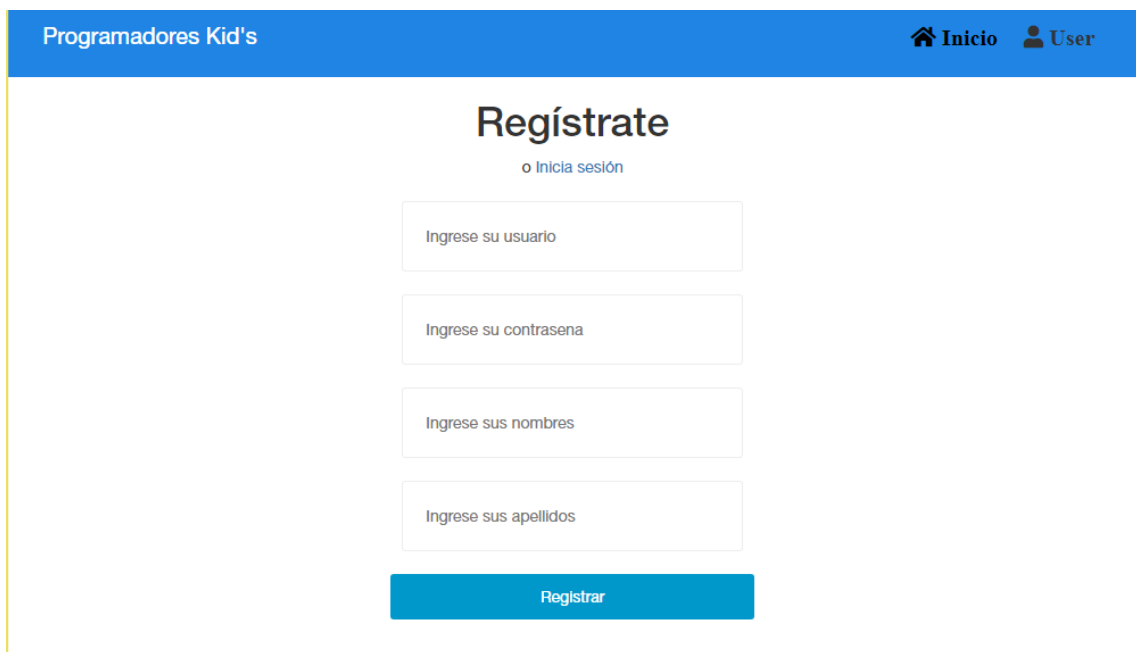
Nro. 03	Historia de usuario		
Descripción	Implementación del logueo de usuario		
Notas:	Registro del estudiante mediante un usuario y contraseña, además de información de sus datos personales, para estudiantes nuevos y el acceso para los usuarios registrados. Se debe realizar las restricciones y controles al momento del llenado de la información en el formulario.		
Esfuerzo Estimado	3 días	Esfuerzo resultante	3 días
Riesgo	bajo	Valor para el cliente	10

Interfaz



The screenshot shows the login page for 'Programadores Kid's'. At the top, there is a blue header with the site name on the left and navigation links for 'Inicio' and 'User' on the right. The main content area has a large heading '¿Ya estas Registrado? Inicia sesión' and a link 'o Crea una cuenta'. Below this are two input fields: 'Ingrese su usuario o email' and 'Ingrese su contraseña'. At the bottom is a blue button labeled 'Iniciar sesión'.

Figura 17: Interfaz de inicio de sesión del proyecto Programadores Kid's



The screenshot shows the registration page for 'Programadores Kid's'. It features a blue header with the site name and navigation links. The main heading is 'Regístrate' with a link 'o Inicia sesión'. There are four input fields: 'Ingrese su usuario', 'Ingrese su contraseña', 'Ingrese sus nombres', and 'Ingrese sus apellidos'. A blue 'Registrar' button is at the bottom.

Figura 18: Interfaz de registro de usuario del proyecto Programadores Kid's

Tabla 7: HS4, Actividades grupales en aula

Nro. 04	Historia de usuario		
Descripción	Actividades grupales en aula – Módulo 01		
Notas: Interfaz gráfica con instrucciones sencillas y gráficos instructivos, que permiten desarrollar las actividades en el salón, formando grupos de trabajos y con materiales que deben estar en el aplicativo para poder ser descargados. Implementado de acuerdo al diseño educativo elaborado por el ejecutor, en coordinación con el docente de aula de la institución educativa.			
Esfuerzo Estimado	5 días	Esfuerzo resultante	5 días
Riesgo	bajo	Valor para el cliente	10

The screenshot shows the 'Programadores Kid's' application interface. At the top, there is a blue navigation bar with 'Inicio', 'User', and 'Salir' options. Below this is a progress indicator with 15 numbered steps, where step 4 is highlighted in yellow. A box on the right contains instructions: 'Indicaciones: Trabajando en grupo con tus compañeros realiza las actividades indicadas en la práctica.' The main content area is titled 'Trabajo en grupo 01' and features a green 'Siguiete actividad' button. The activity title is '¿Cómo almacena la información un computador?'. Below the title are tabs for 'Información', 'Procedimientos Iniciales', 'Actividades', and 'Usa la creatividad'. The 'Información' tab is active, displaying text about binary storage and a diagram of five cards representing powers of 2 (16, 8, 4, 2, 1).

Figura 19: Interfaz de trabajo en grupo del proyecto Programadores Kid's

Flujo de trabajo 03:

Concluido la implementación del logueo del usuario el segundo programador inicia con la implementación de las Actividades de interacción del Módulo 01, la cual incluye diversas actividades de construcción de algoritmos utilizando la programación en bloques, y en forma gráfica observa el estudiante los resultados,

para pasar a actividades cada vez de mayor complejidad, adquiriendo puntuaciones por resolver cada etapa. Esta interfaz está dividida en tres zonas principales de trabajo, el lado izquierdo donde se tiene los bloques a utilizar, se arrastran a la parte central construyendo el código que resuelve el problema y en el sector derecho se ejecuta el código implementado, mostrándose en forma gráfica el resultado, de ser correcto se muestra una ventana modal que indica el puntaje acumulado.

Tablero Kanban

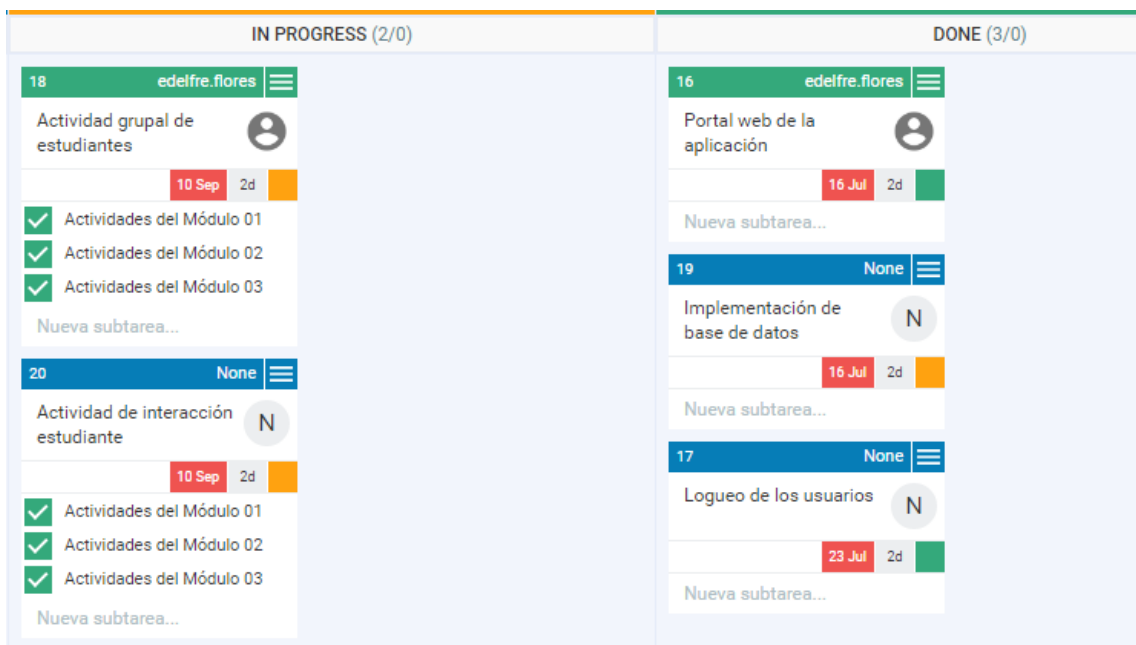


Figura 20: Tablero Kanban del flujo de trabajo 03 Programadores Kid's

Tabla 8: HS5, Actividades interactivas con el entorno

Nro. 05	Historia de usuario		
Descripción	Actividades interactivas con el entorno – Módulo 01		
Notas:	Actividades que se desarrollan en el aplicativo por parte del estudiante, teniendo actividades de crecimiento progresivo de complejidad y se destina puntajes al momento de ser superados. La navegación debe permitir desplazarse directamente por las diferentes actividades. Implementado de acuerdo al diseño educativo elaborado por el ejecutor, en coordinación con el docente de aula de la institución educativa.		
Esfuerzo Estimado	10 días	Esfuerzo resultante	10 días
Riesgo	bajo	Valor para el cliente	10

Interfaz

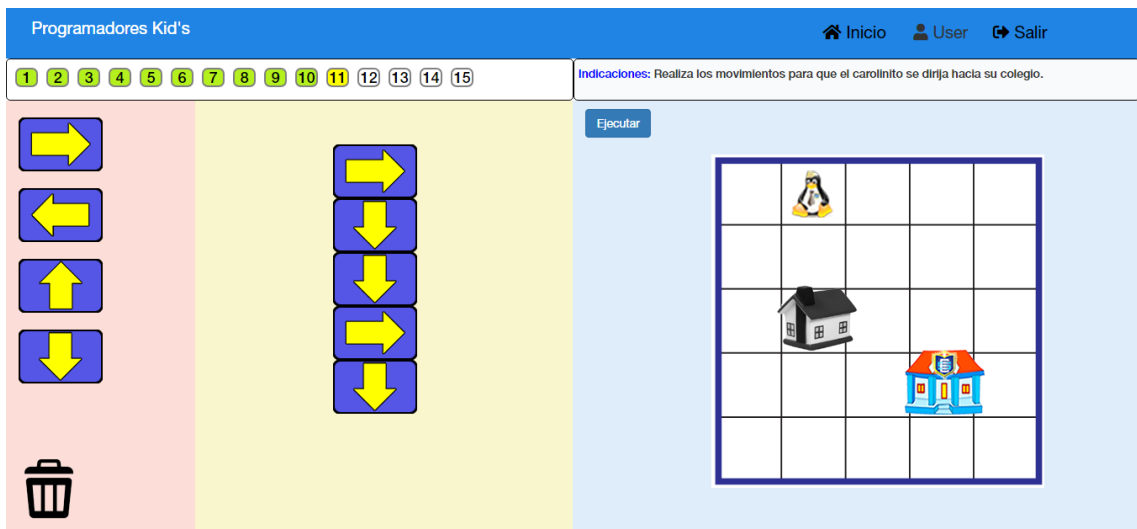


Figura 21: Interfaz de actividades interactivas con el entorno A de Programadores Kid's

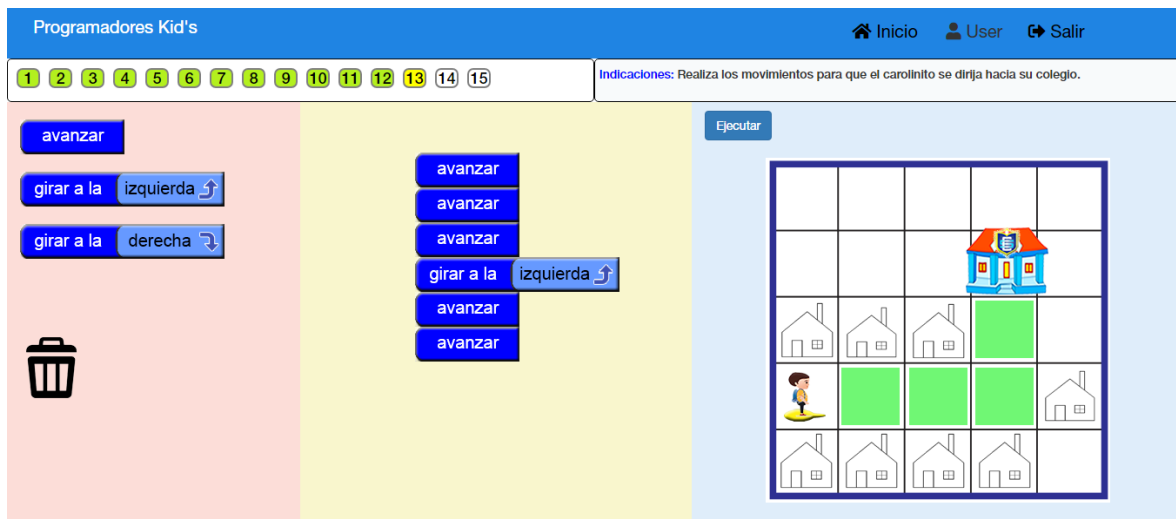


Figura 22: Interfaz de actividades interactivas con el entorno B Programadores Kid's

Flujo de trabajo 04:

Se inicia a la par del desarrollo de las actividades de interacción con el estudiante, las actividades de construcción creativa por parte del programador 1, en esta etapa se implementa una interfaz utilizando HTML, CSS, JavaScript, JQuery, y otras herramientas, donde se presenta al estudiante un proyecto para su desarrollo en equipo, utilizando todos los conocimientos y destrezas acumuladas.

Tablero Kanban

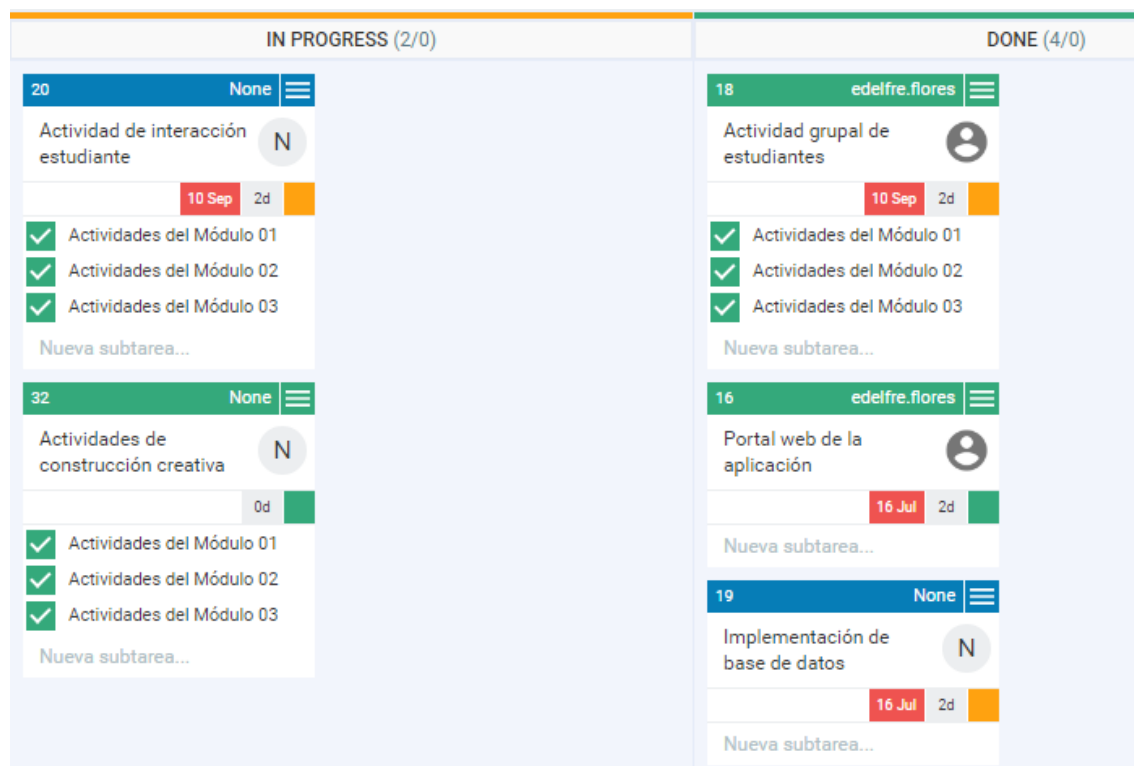


Figura 23: Tablero Kanban del flujo de trabajo 04 de Programadores Kid's

Tabla 9: HS6, Actividades de construcción creativa

Nro. 06	Historia de usuario		
Descripción	Actividades de construcción creativa – Módulo 01		
Notas::	Interfaz gráfica sencilla y visual, que permite desarrollar actividades de construcción por parte de los estudiantes, trabajando en equipo, desarrollando la creatividad, posibilidad de subir el proyecto terminado para su publicación en el portal web. Implementado de acuerdo al diseño educativo elaborado por el ejecutor, en coordinación con el docente de aula de la institución educativa.		
Esfuerzo Estimado	10 días	Esfuerzo resultante	10 días
Riesgo	bajo	Valor para el cliente	10

Flujo de trabajo 05:

Finalizando los procesos se tiene la interfaz de examen del estudiante, donde se cuenta con opciones de paginación de Bootstrap, mostrando cada pregunta en una página con sus alternativas correspondientes que son radio button, al finalizar el examen es estudiante utiliza el botón calificar, mediante el cual se pone en marcha el código fuente en JavaScript y JQuery para procesar los resultados, estos resultados se almacenan en la base de datos MySql en caso que obtenga un puntaje mayo o igual a 15 puntos; en caso contrario se le da la opción de volver a realizar el examen, con lo cual se concluye la ejecución del Módulo.

Tablero Kanban

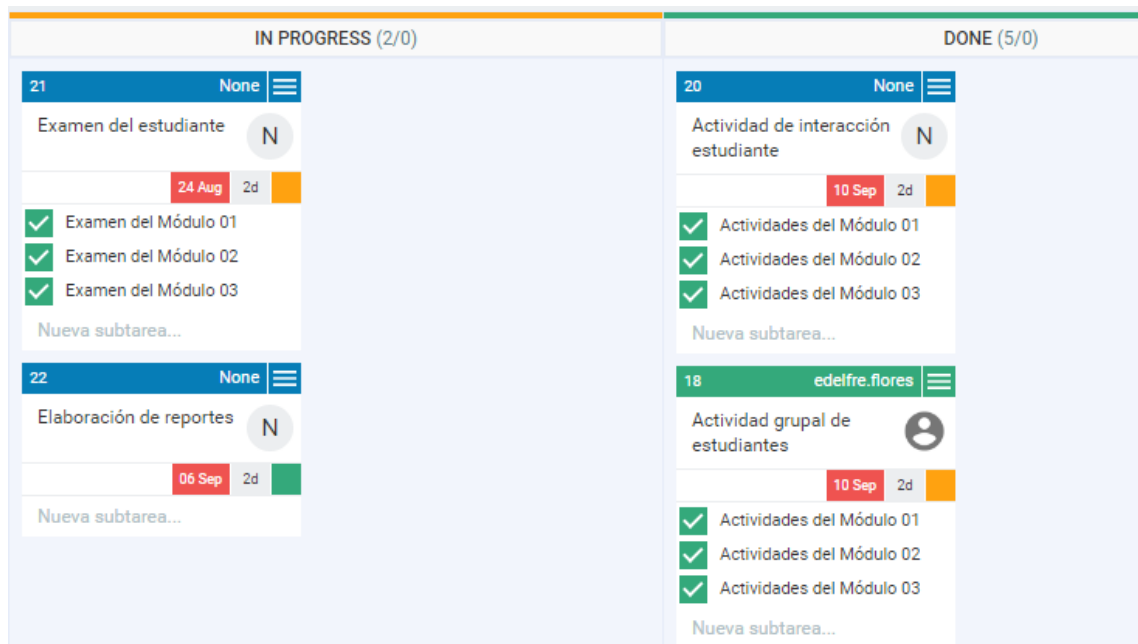


Figura 24: Tablero Kanban del flujo de trabajo 05 de Programadores Kid's

Tabla 10: HS7, Desarrollo del examen de módulo

Nro. 07	Historia de usuario		
Descripción	Desarrollo del examen de módulo		
Notas: Contar con opciones de paginación donde se muestre cada pregunta con sus alternativas en radio button, al finalizar mostrar el resultado y permitir volver a desarrollar el examen en caso de tener un puntaje menor o igual a 14. Implementado de acuerdo al diseño educativo elaborado por el ejecutor, en coordinación con el docente de aula de la institución educativa.			
Esfuerzo Estimado	5 días	Esfuerzo resultante	5 días
Riesgo	bajo	Valor para el cliente	10

Interfaz

Figura 25: Interfaz de desarrollo del examen de Programadores Kid's

Tabla 11: HS8, Elaboración de reportes

Nro. 08	Historia de usuario		
Descripción	Elaboración de reportes		
Notas: El primer reporte es el diploma del estudiante al culminar satisfactoriamente cada módulo, diploma que muestra datos del estudiante y que se genera en formato pdf. Luego se debe tener para el docente reportes de avance y notas finales.			
Esfuerzo Estimado	3 días	Esfuerzo resultante	3 días
Riesgo	bajo	Valor para el cliente	05

4.3 EVALUACIÓN DEL NIVEL DE DESARROLLO DEL PENSAMIENTO COMPUTACIONAL

EVALUACIÓN DE LOS ESTUDIANTES ANTES DEL EXPERIMENTO

Al inicio del proceso de experimentación se procedió a evaluar a los estudiantes, mediante el pre-test. El test para medir el nivel de desarrollo del pensamiento computacional es una adaptación del test de Ramon (2016), muy utilizado en escuelas de nivel primario en Europa y que se puede hallar en <https://www.scoop.it/t/test-de-pensamiento-computacional> .

El test se implementó en el entorno Moodle y se muestra una interrogante en el siguiente gráfico.

¿Qué órdenes lleva al Carolinito a su colegio, por el camino señalado?

Seleccione una:

- a. Opción A
- b. Opción D
- c. Opción B
- d. Opción C

Figura 26: Ejemplo de pregunta del Pre-test de Programadores Kid's

Los resultados que se obtuvo se muestran a continuación:

Tabla 12: Distribución de calificaciones obtenidos en el Pre-test por los estudiantes de la IEP Glorioso San Carlos - Puno

Categorías	Varones		Mujeres		Total	
	Nº	%	Nº	%	Nº	%
Inadecuado (0-5)	78	36	32	34	110	35
Puede mejorar (6-10)	89	41	41	44	130	42
Satisfactorio (11-15)	44	20	18	19	62	20
Excelente (16-20)	7	3	2	2	9	3
Total	218	100	93	100	311	100

Elaboración: El ejecutor

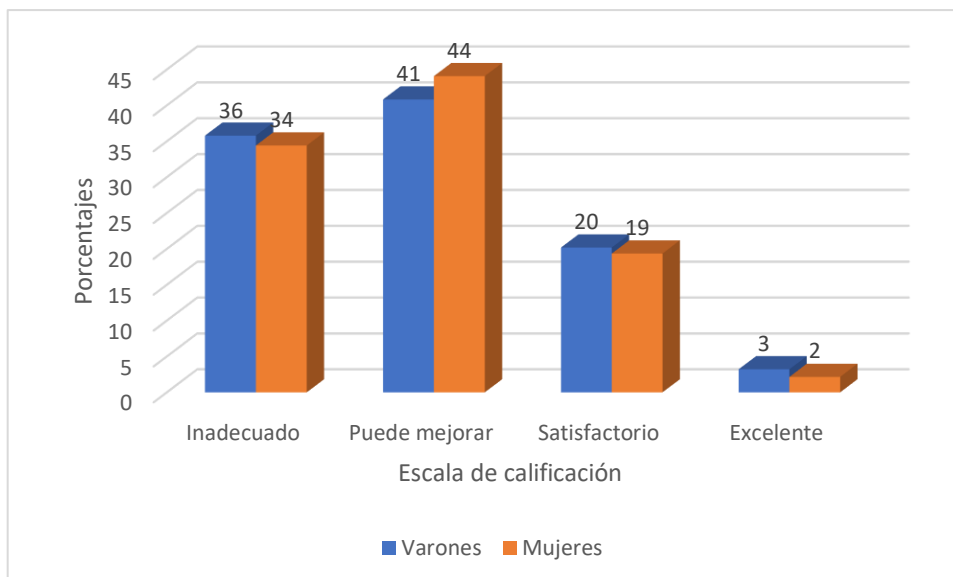


Figura 27: Distribución de calificativos obtenidos en el Pre-test por los estudiantes de la IEP Glorioso San Carlos - Puno

En la tabla 12 y la Figura 26 se observa que el mayor porcentaje 77% obtienen calificativos en la categoría de inadecuado y puede mejorar, es decir obtienen calificativos entre 0 y 10 puntos, mostrando que al inicio del proceso de experimentación los estudiantes ingresan, no habiendo desarrollado capacidades del pensamiento computacional.

Se observa también, que existe un porcentaje de 20% que se ubican en la categoría de satisfactorio, es decir obtienen calificativos entre 11 y 15 puntos, esto demuestra la existencia de un porcentaje de estudiantes que tienen desarrollado, en algún nivel el pensamiento computacional, debido a que se prepararon en alguna academia o cursos de verano que existen en la ciudad.

En forma general en base a los resultados que se observa, se puede afirmar que, los estudiantes ingresan al experimento sin tener conocimiento de los conceptos de pensamiento computacional.

CÁLCULO DE LA PRUEBA DE HIPÓTESIS ANTES DEL INGRESO AL EXPERIMENTO

a. Hipótesis Nula: H_0 :

El promedio obtenido en la muestra es igual o menor al valor de 15, entonces se tiene que los estudiantes aun no tienen desarrollado el pensamiento computacional.

b. Hipótesis Alternativa: H_a :

El promedio obtenido por los estudiantes es mayor a 15, entonces los estudiantes ingresan al experimento teniendo desarrollado el pensamiento computacional en un alto nivel.

- **Nivel de significancia**

$$\alpha = 0,05 \text{ ó } 5\%$$

- **Región crítica**

$$\text{Con } Z_{\alpha} = Z_{0,05} = 1,64$$

c. Cálculo estadístico

$$Z_c = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Donde:

Z_c = Z calculada

\bar{X} = Media aritmética de la muestra

μ = Valor 15, escogido según los antecedentes

σ = Desviación estándar

n = número de estudiantes de la muestra

El cálculo siguiente

$$\bar{X} = 7,85$$

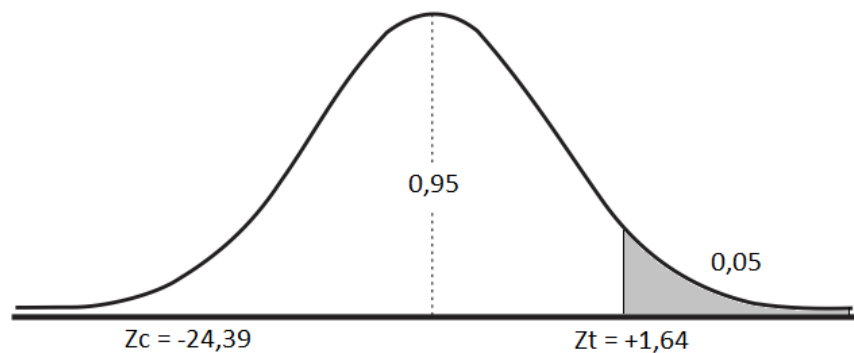
$$\mu = 15$$

$$\sigma = 5,17$$

$$n = 311$$

$$Z_c = -24,39$$

a. Toma de decisión



“Z” calculada < “Z” crítica

$$-24,39 < +1,64$$

SE ACEPTA Ho

Como La $Z_c = -24,39 < Z_t = +1,64$; se acepta la hipótesis nula, por lo tanto, el promedio obtenido por los estudiantes no se encuentra en el nivel de excelente que tienen valores entre 16 y 20 puntos, y estos valores se encuentran entre 0 y 15 puntos, demostrando

que los estudiantes aun no tienen desarrollado el pensamiento computacional, al momento del ingreso al experimento con el modelo holístico de código-alfabetización; con un nivel de significancia del 0,05.

EVALUACIÓN DE LOS ESTUDIANTES DESPUÉS DEL EXPERIMENTO

Después de la aplicación de los instrumentos de investigación, se procede a la aplicación del Pos test, esta prueba es una adaptación del propuesto por Román Gonzales (2016) y aplicado en varios países de Europa, principalmente en España.

Tabla 13: Distribución de calificativos obtenidos en el Post-test por los estudiantes de la IEP Glorioso San Carlos - Puno

Categorías	Varones		Mujeres		Total	
	Nº	%	Nº	%	Nº	%
Inadecuado (0-5)	0	0	0	0	0	0
Puede mejorar (6-10)	10	5	2	2	12	4
Satisfactorio (11-15)	22	10	10	11	32	10
Excelente (16-20)	186	85	81	87	267	86
Total	218	100	93	100	311	100

Elaboración: El ejecutor

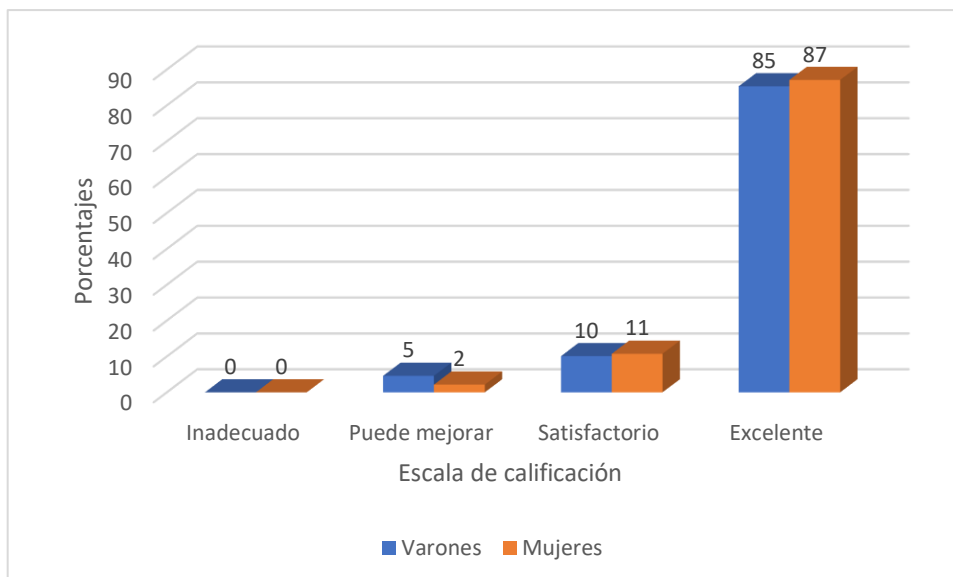


Figura 28: Distribución de calificativos obtenidos en el Post-test por los estudiantes de la IEP Glorioso San Carlos - Puno

En la tabla 13 y la Figura 27 se observa que el mayor porcentaje 86% obtienen calificativos en la categoría de Excelente, es decir obtienen calificativos entre 16 y 20 puntos, mostrando que luego de la aplicación del prototipo de software basado en el modelo de codigoalfabetización, los estudiantes muestran un alto nivel de desarrollo de sus capacidades y actitudes del pensamiento computacional.

Se observa también que existe un porcentaje de 10% que se ubican en la categoría de satisfactorio, es decir obtienen calificativos entre 11 y 15 puntos, esto demuestra la existencia de un porcentaje de estudiantes que aún no han desarrollado debidamente el pensamiento computacional, debido a diferentes factores como la no asistencia permanente a las clases u otros factores propios de los estudiantes y el centro educativo.

En forma general en base a los resultados que se observa, se puede afirmar que, los estudiantes que fueron participes de la experimentación mediante el prototipo de software basado en el modelo holístico de código-alfabetización ha logrado puntajes

mayoritariamente entre 16 y 20 puntos en el sistema vigesimal de calificación, demostrando haber alcanzado un nivel alto de conocimiento de los conceptos de pensamiento computacional.

CÁLCULO DE LA PRUEBA DE HIPÓTESIS LUEGO DEL EXPERIMENTO

a. Hipótesis Nula: Ho:

El promedio obtenido en la muestra es igual o menor al valor de 15, entonces se tiene que el modelo holístico de código-alfabetización no permite el desarrollo del pensamiento computacional.

b. Hipótesis Alterna Ha:

El promedio obtenido por los estudiantes es mayor a 15, entonces el uso del prototipo de software basado en el modelo de código-alfabetización permite el desarrollo del pensamiento computacional en un alto nivel.

- **Nivel de significancia**

$$\alpha = 0,05 \text{ ó } 5\%$$

- **Región crítica**

$$\text{Con } Z_{\alpha} = Z_{0,05} = 1,64$$

d. Cálculo estadístico

$$Z_c = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Donde:

$Z_c = Z$ calculada

\bar{X} = Media aritmética de la muestra

μ = Valor 15, escogido según los antecedentes

σ = Desviación estándar

n = número de estudiantes de la muestra

El cálculo siguiente

$\bar{X} = 16,97$

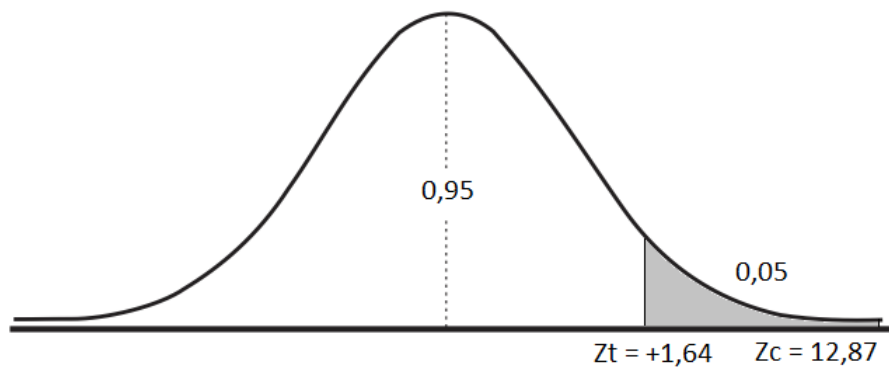
$\mu = 15$

$\sigma = 2,70$

n = 311

$Z_c = +12,87$

b. Toma de decisión



“Z” calculada > “Z” crítica

+12,87 > +1,64

SE RECHAZA H_0

Como $Z_c = +12,87 > Z_t = +1,64$; se rechaza la hipótesis nula, por lo tanto, el promedio obtenido por los estudiantes es mayor a 15, es decir se encuentran en el nivel de excelente que son valores entre 16 y 20 puntos, demostrando que el uso del prototipo de software basado en el modelo de código-alfabetización, permite el desarrollo del pensamiento computacional en un alto nivel, en los estudiantes del nivel primario de la I.E.P. Glorioso San Carlos de la ciudad de Puno, con un nivel de significancia del 0,05.

CAPÍTULO V

DISCUSIÓN DE RESULTADOS

5.1 DISCUSIÓN

En base a los antecedentes considerados en el presente trabajo de investigación, y los hallazgos luego de la construcción de un modelo holístico de código-alfabetización y la implementación y experimentación del prototipo de software, aceptamos la hipótesis general que indica que: el modelo holístico de código-alfabetización permite el desarrollo del pensamiento computacional en niños de educación primaria de la región Puno. Estos resultados guardan relación con los trabajos de Román(2013) y Román(2016), quien indica que la código-alfabetización es adecuado para el desarrollo de un amplio abanico de aptitudes y habilidades (principalmente razonamiento lógico y resolución de problemas, pero también aptitudes espaciales, atencionales, de motivación e iniciativa, creatividad o trabajo en equipo); que el ‘coding’ es adecuado para todo tipo de alumnos, independientemente de su edad, género o capacidad; y que el ‘coding’ es adecuado para atender a la diversidad del alumnado dado la flexibilidad y autonomía que ofrecen sus recursos de aprendizaje. Haciendo notar que principalmente se realizó una aplicación del proyecto code.org, con los lenguajes visuales ‘por flechas’ para Educación Infantil y comienzos de Educación Primaria; con los lenguajes visuales ‘por bloques’ para finales de Educación Primaria y comienzos de Educación Secundaria; y con los lenguajes ‘textuales’ para finales de Educación Secundaria, Bachillerato y Formación Profesional en España. Esto contrasta con el prototipo de software elaborado en el presente trabajo de investigación, en el cual se inicia con conceptos de arrastrar-soltar, almacenamiento de la información en bits, para luego introducir el concepto de

algoritmo mediante una implementación de flechas visuales, basado en la programación en bloques, para niños de tercero, cuarto y quinto grados de educación primaria.

El Modelo holístico de código-alfabetización

Al elaborar el modelo holístico coincidimos con Domínguez (2014), en dos factores, que hace uso del concepto de integridad que se halla en el concepto de sistemas y holismo, para su modelamiento, es decir toma como base la composición de un sistema, que cuenta con los elementos que interactúan, se ubica dentro de un contexto determinado, tiene límites y tiene un objetivo que cumplir. Luego también considera otro factor importante dentro de su modelo como es la cibernética de segundo orden, poniendo énfasis en el proceso de comunicación, control y retroalimentación para mantener la homeostasis en el sistema, de igual manera en nuestro modelo consideramos importantes estos factores, siendo los diferentes tipos de evaluación los elementos de retroalimentación para el control en el sistema.

De igual manera considerando a Serrano (2008) en su Triángulo pedagógico, muestra la interacción de cuatro elementos: el alumno, el profesor, los contenidos y los objetivos; que se ubican en los vértices; haciendo notar la importancia de estos tres componentes, Además en los lados uniendo los vértices se ubican, la construcción de significados, uniendo al alumno con el contenido, la atribución de sentido relacionando al alumno con los objetivos y las competencias uniendo los contenidos con los objetivos. Se tiene también que el profesor se ubica en el baricentro del triángulo, es decir, en el centro de gravedad donde reposa el equilibrio de los procesos de enseñanza y aprendizaje. De igual manera en nuestro modelo para el proceso de enseñanza-aprendizaje de la programación es indispensable considerar estos elementos, que son los en su interacción dan como resultado el lograr el objetivo del sistema.

Mendoza Vera, Eugenio (2016), en su modelo holista de aprendizaje, realiza una definición ampliada de aprendizaje, indicando que es un proceso complejo y sistémico, cuyo origen está en la necesidad que tiene todo organismo vivo de alcanzar un equilibrio con el entorno, que le permita mantenerse vivo y lograr sus fines. Plantea un modelo de aprendizaje mostrando las interacciones de los componentes del sistema, las retroalimentaciones y la importancia de las variables endógenas y exógenas que participan en el proceso, además de indicar la importancia del entorno influyendo en el sistema. En nuestro modelo coincidimos con la forma de plantear el modelo, pero en nuestro caso, ampliamos la visión hacia otros elementos como la inseparable relación entre enseñanza y aprendizaje, siendo estos procesos realizados por el docente y el estudiante en una interacción sinérgica.

CAPÍTULO VI

CONCLUSIONES

1. El modelo holístico de código-alfabetización construido en base a conceptos de sistemas, fundamentos pedagógicos y la aplicación del prototipo de software ha permitido el desarrollo del pensamiento computacional, en un nivel medio a alto, en niños de educación primaria de la IEP Glorioso San Carlos de la ciudad de Puno, tal como se demuestra en la prueba de hipótesis Z normal con un nivel de significancia del 0,05.
2. El modelo holístico de código alfabetización se ha construido aplicando conceptos de sistemas, modelando los componentes e interacciones de un sistema; cibernética de segundo orden, tomando en cuenta la comunicación y control y la retroalimentación; autopoiesis, y principios pedagógicos del socio constructivismo y construccionismo, donde el estudiante construye su aprendizaje a partir de su propia forma de ser, pensar e interpretar la información.
3. El prototipo de software se ha desarrollado tomando como base el modelo holístico de código-alfabetización, aplicando la metodología ágil Kanban, que permite un desarrollo de software incremental, cooperativo y sencillo; para un entorno web.
4. La aplicación del prototipo de software desarrollado en base al modelo holístico de código-alfabetización, ha permitido obtener calificativos altos en pensamiento computacional, logrando el 86% de los estudiantes puntajes entre 16 y 20 puntos, demostrando que se alcanzó un nivel de excelente en el desarrollo del pensamiento computacional, en niños de educación primaria del IEP Glorioso San Carlos de la ciudad de Puno.

CAPÍTULO VII

RECOMENDACIONES

1. Ampliar el prototipo de software basado en el modelo holístico de código-alfabetización, para su aplicación en las instituciones educativas de nivel primario de toda la región Puno, que permita el desarrollo del pensamiento computacional y ponernos al nivel de los niños de los países vecinos de la región y el mundo.
2. A las autoridades educativas del país, evaluar el modelo holístico de código-alfabetización propuesto en la presente investigación, para generar un modelo propio, que pueda ser aplicado como referencia para el desarrollo del pensamiento computacional en nuestra nación.
3. Proponer en el currículo nacional se incluya el desarrollo del pensamiento computacional, como una competencia transversal en todos los niveles de nuestra educación, que nos ubique al mismo nivel de desarrollo de nuestros países vecinos y del mundo.
4. Las autoridades educativas de nuestro país, implementar una organización dedicada a desarrollar un aplicativo de software, que se ejecute en entorno web y para dispositivos móviles, que desarrolle el pensamiento computacional en todos los niveles educativos de nuestro país, personalizados con motivos propios cada región.

- Coloma, C. R., & Tafur, R. M. (1999). El constructivismo y sus implicancias en educación. *Educación Vol VIII, VIII(16)*, 217–244.
- Daza, S. (2013). Sujeto autopoiético y pedagogía creativa: la autopoiesis como característica fundamental para la construcción del maestro en formación. *NOVUM, Revista de Ciencias Sociales Aplicadas*, 3, 95–102. Retrieved from <https://revistas.unal.edu.co/index.php/novum/article/view/67131>
- Domínguez, G. R. (2013). *Desarrollo de un modelo holístico y un método sistémico para representar y estimular la metacognición*. Instituto Politécnico Nacional - México. Retrieved from [https://tesis.ipn.mx/jspui/bitstream/123456789/17826/1/Desarrollo de un modelo holistico y un metodo sistemico para representar .pdf](https://tesis.ipn.mx/jspui/bitstream/123456789/17826/1/Desarrollo%20de%20un%20modelo%20holistico%20y%20un%20metodo%20sistemico%20para%20representar%20.pdf)
- Falbel, A. (2001). Construccinismo. *Enlaces 2001, Abriendo Las Fronteras Del Aula*, 1–11. Retrieved from <http://www.tecnoedu.net/lecturas/materiales/lectura15.pdf>
- Gallegos, R. (2018). *Educación holista, aprendiendo con Ramón Gallegos*. (F. R. Gallegos, Ed.). España. Retrieved from [https://www.textos.info/buscar/ramon gallegos?](https://www.textos.info/buscar/ramon%20gallegos?)
- Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación*. *Journal of Chemical Information and Modeling* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Heylighen, F., & Joslyn, C. (2001). Cybernetics and Second-Order Cybernetics. *Encyclopedia of Physical Science y Technology*. Retrieved from <http://pespmc1.vub.ac.be/Papers/Cybernetics-EPST.pdf>
- Hitschfeld, N., Pérez, J., & Simmonds, J. (2015). Pensamiento computacional y programación a nivel escolar en Chile: El valor de formar a los innovadores tecnológicos del futuro. *BITS de Ciencia*, 1(1). Retrieved from <https://www.dcc.uchile.cl/Bitsdeciencia12.pdf>

- Kniberg, H., & Skarin, M. (2010). *Kanban y Scrum – obteniendo lo mejor de ambos*. (C4Media, Ed.). Estados Unidos de América. Retrieved from http://www.proyectalis.com/documentos/KanbanVsScrum_Castellano_FINAL-printed.pdf
- Maturana, H., & Porsken, N. (2004). *Del Ser al Hacer*. (Comunicaciones Noreste Ltda., Ed.). Chile. Retrieved from <http://matriztica.cl/wp-content/uploads/Cap-I-Del-ser-al-hacer.pdf>
- Maturana, H., & Varela, F. (1994). *De máquinas y seres vivos*. (E. U. S.A., Ed.). Santiago de Chile. Retrieved from <https://filosofosinsentido.files.wordpress.com/2013/07/1677.pdf>
- Mendoza, E. (2015). Modelo educativo andragógico - Un enfoque holístico. *Revista Venezolana de Tecnología y Sociedad*, 8(1), 37–49. Retrieved from <http://www.revencyt.ula.ve/storage/repo/ArchivoDocumento/rvtsoc/v8n1/art04.pdf>
- Mendoza, E., & Godoy, N. (2016). El aprendizaje desde un enfoque holístico e integrador. *TEACS Año 9, 19*, 39–54. Retrieved from <http://www.ucla.edu.ve/dac/revistateacs/articulos/Rev19-Art3-MendezayOtro.pdf>
- Moreno, J. (2014). ¿Qué es el pensamiento computacional? Retrieved February 24, 2019, from <https://programamos.es/que-es-el-pensamiento-computacional/>
- Ortiz, D. (2015). El constructivismo como teoría y método de enseñanza. *Sophia, Colección de Filosofía de La Educación*, 1(19), 93. <https://doi.org/10.17163/soph.n19.2015.04>
- Pérez, R. (2002). La evaluación de programas en el marco de la educación de calidad, 4(2002), 43–76. Retrieved from <https://core.ac.uk/download/pdf/60635802.pdf>
- Pinto, A. R. (2013). Pedagogía construccionista en la enseñanza de las ciencias sociales. *Dialnet*. Retrieved from <https://dialnet.unirioja.es/servlet/articulo?codigo=4532543>

- Posada, F. (2016). Modelo CSTA para el Pensamiento Computacional. Retrieved August 1, 2017, from <http://canaltic.com/blog/?p=2565>
- Prezsky, M. (2008). Programming Is the New Literacy. Retrieved August 1, 2017, from <https://www.edutopia.org/literacy-computer-programming>
- Prezsky, M. (2010). Nativos e inmigrantes digitales. *Cuadrenos SEK 2.O*, (M-24433-2010), 21. <https://doi.org/http://hdl.handle.net/10230/21226>
- Razeto, P., & Ramos, R. (2013). Impecabilidad Conceptual ¿Qué es autopoiesis? *Autopoiesis. Un Concepto Vivo. Editorial Nueva Civilización*, 27–57. Retrieved from [http://www.ificc.cl/sites/default/files/Razeto-Barry and Ramos-Jiliberto 2013 - Que es autopoiesis_0.pdf](http://www.ificc.cl/sites/default/files/Razeto-Barry%20and%20Ramos-Jiliberto%202013%20-%20Que%20es%20autopoiesis_0.pdf)
- Richtel, M. (2014). Reading, Writing, Arithmetic, and Lately, Coding. Retrieved August 1, 2017, from <https://www.nytimes.com/2014/05/11/us/reading-writing-arithmetic-and-lately-coding.html>
- Román, M. (2013). Dónde ubicar la codigoalfabetización. Retrieved August 10, 2017, from <https://codigoalfa.hypotheses.org/151>
- Román, M. (2016). *Códigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: Validación de un instrumento y evaluación de programas*. UNED. Retrieved from http://e-spacio.uned.es/fez/eserv/tesisuned:Educacion-Mroman/ROMAN_GONZALEZ_Marcos_Tesis.pdf
- Rushkoff, D. (2012). Code Literacy: A 21st-Century Requirement. Retrieved August 2, 2017, from <https://www.edutopia.org/blog/code-literacy-21st-century-requirement-douglas-rushkoff>
- Scratch, D. (2017). Dr. Scratch: Analiza tus proyectos de Scratch. Retrieved August 1, 2017, from <http://www.drscratch.org/>
- ScratchEd. (2011). Computational Thinking Concepts: Webinar. Retrieved August 2, 2017,

- from <http://scratched.gse.harvard.edu/resources/computational-thinking-concepts-march-2011-webinar>
- Serrano, J., & Pons, R. (2008). La concepción constructivista de la instrucción. *Revista Mexicana de Investigación Educativa*, 13(38), 681–712. Retrieved from <http://www.redalyc.org/pdf/140/14003802.pdf>
- Taborda, H., & Medina, D. (2014). *Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares*. Universidad ICESI Colombia. Retrieved from https://s3.amazonaws.com/academia.edu.documents/33887265/WP-CIES_3-FINAL.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1539282811&Signature=l6IDiSQEFkc1TXC3tlV1LUyXLIU%3D&response-content-disposition=inline%3B+filename%3DProgramacion_de_computadores_y_d
- Tapia, C. A. (1971). SISTEMAS , MODELOS Y SIMULACION Ver. “ *La Teoría General de Sistemas* ” de Pedro Voltes Bou , Editorial Hispano Europea, 2–20. Retrieved from <http://materias.fi.uba.ar/7526/docs/teoria.pdf>
- Tarifa, E. E. (1998). Teoría de modelos y simulación: Introducción a la simulación. *Universidad Nacional de Jujuy*, 1–17. Retrieved from http://www.econ.unicen.edu.ar/attachments/1051_TecnicasIISimulacion.pdf
- Wernicke, C. G. (1999). Educación Holística y Pedagogía Montessori, 1–11. Retrieved from [http://holismo.org.ar/images/articulos/37_EdHolPedMont\(1\).pdf](http://holismo.org.ar/images/articulos/37_EdHolPedMont(1).pdf)
- Wing, J. (2006). Computational Thinking. COMMUNICATIONS OF THE ACM. Retrieved August 20, 2017, from <http://dl.acm.org/citation.cfm?doid=1118178.1118215>
- Wing, J. (2008). Computational thinking and thinking about computing. *Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium*, (July),

3717–3725. <https://doi.org/10.1109/IPDPS.2008.4536091>

Zapotecatl, J. L. (2014). Pensamiento Computacional. In *Instituto Nacional de Astrofísica, Óptica y Electrónica*. <https://doi.org/10.1017/CBO9781107415324.004>

ANEXOS

ANEXO 01

PRE TEST

EVALUACIÓN DEL PENSAMIENTO COMPUTACIONAL

I. En las siguientes preguntas, marque la alternativa correcta.

Pregunta 01:

¿Qué órdenes lleva al Carolinito a su colegio, por el camino señalado?

	Opción A 	Opción B
	Opción C 	Opción D

Seleccione una:

- a. Opción A
- b. Opción D
- c. Opción B
- d. Opción C

Pregunta 02:

Para llevar al Carolinito a su colegio por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

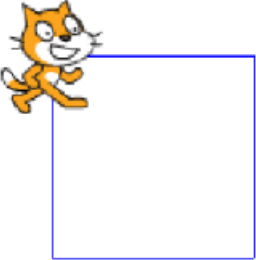
--	--

Seleccione una:

- a. Opción B
- b. Opción D
- c. Opción C
- d. Opción A

Pregunta 03:

¿Qué órdenes debe ejecutar el gatito para dibujar el cuadrado? Cada uno de los lados del cuadrado mide 100 pasos.

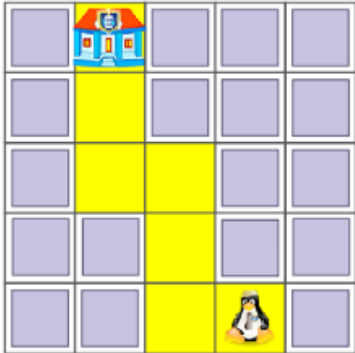
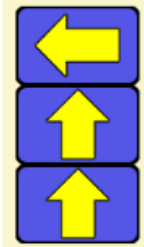
	Opción A mover 100 pasos girar 90 grados mover 100 pasos girar 90 grados mover 100 pasos girar 90 grados mover 100 pasos	Opción B mover 50 pasos girar 45 grados mover 50 pasos girar 90 grados mover 50 pasos girar 90 grados mover 50 pasos
	Opción C mover 100 pasos girar 90 grados mover 100 pasos girar 90 grados mover 100 pasos girar 90 grados mover 100 pasos	Opción D mover 100 pasos girar 90 grados mover 100 pasos girar 90 grados mover 100 pasos girar 90 grados mover 100 pasos

Seleccione una:

- a. Opción B
- b. Opción A
- c. Opción D
- d. Opción C

Pregunta 04:

¿Cuántas veces se debe repetir la secuencia para llevar al Carolinito hasta su colegio por el camino señalado?

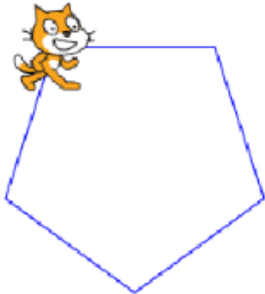
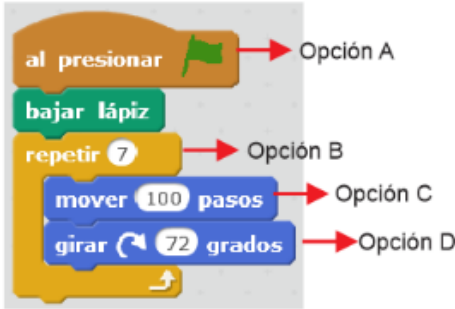
		Opción A x 2
		Opción B x 1
		Opción C x 3
		Opción D x 4

Seleccione una:

- a. Opción A
- b. Opción D
- c. Opción B
- d. Opción C

Pregunta 05:

Para que el gatito dibuje el siguiente pentagono (100 pasos de lado), ¿en qué paso de la siguiente secuencia de órdenes hay un error?

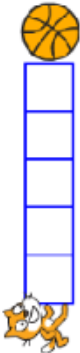




	
---	--

Seleccione una:

- a. Opción C
- b. Opción B
- c. Opción D
- d. Opción A

Pregunta 06:

¿Qué secuencia de órdenes debe ejecutar el gatito para dibujar la escalera que llegue hasta la pelota? Cada peldaño sube 40 pasos.

	Opción A 	Opción B 
	Opción C 	Opción D 

Seleccione una:

- a. Opción C
- b. Opción D
- c. Opción A
- d. Opción B

Pregunta 07:

¿Qué órdenes llevan a "Pac-Man" hasta el fantasma por el camino señalado?

	<p>Opción A</p> <pre> Repetir hasta llegar a... hacer avanzar si hay camino a la derecha hacer girar a la derecha </pre>	<p>Opción B</p> <pre> Repetir hasta llegar a... hacer girar a la derecha si hay camino a la derecha hacer avanzar </pre>
	<p>Opción C</p> <pre> Repetir hasta llegar a... hacer avanzar si hay camino a la derecha hacer girar a la izquierda </pre>	<p>Opción D</p> <pre> Repetir hasta llegar a... hacer avanzar si hay camino a la izquierda hacer girar a la izquierda </pre>

Seleccione una:

- a. Opción B
- b. Opción D
- c. Opción C
- d. Opción A

Pregunta 08:

¿Qué órdenes llevan a "Pac-Man" hasta el fantasma por el camino señalado?

	<p>Opción A</p> <pre> Repetir hasta llegar a... hacer si hay un camino delante hacer avanzar sino girar a la izquierda </pre>	<p>Opción B</p> <pre> Repetir hasta llegar a... hacer si hay un camino delante hacer avanzar sino girar a la derecha </pre>
	<p>Opción C</p> <pre> Repetir hasta llegar a... hacer si hay camino a la derecha hacer girar a la derecha sino avanzar </pre>	<p>Opción D</p> <pre> Repetir hasta llegar a... hacer si hay camino a la izquierda hacer girar a la izquierda sino avanzar </pre>

Seleccione una:

- a. Opción A
- b. Opción D
- c. Opción B
- d. Opción C

Pregunta 09:

¿Qué bloque falta en la siguiente secuencia de órdenes para que "Pac-Man" llegue hasta el fantasma por el camino señalado?

<p>Opción A</p>	<p>Opción B</p>
<p>Opción C</p>	<p>Opción D</p> <p>No falta ningún bloque</p>

Seleccione una:

- a. Opción A
- b. Opción B
- c. Opción B
- d. Opción C

Pregunta 10:

¿Qué órdenes van llevando a "Pac-Man" por el camino señalado e indicándole que se coma el número de fresas correspondiente?

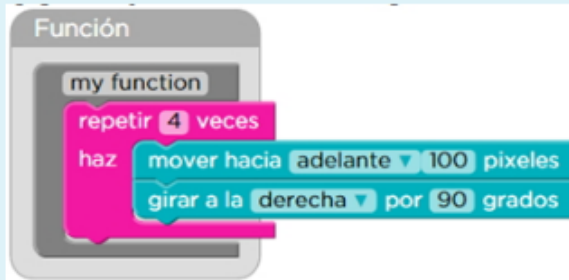
	<p>Opción A</p>	<p>Opción B</p>
	<p>Opción C</p>	<p>Opción D</p>

Seleccione una:

- a. Opción B
- b. Opción A
- c. Opción D
- d. Opción C

Pregunta 11:

Si tenemos el siguiente conjunto de órdenes, al que llamamos "my function", y que dibuja un cuadrado de 100 pixeles de lado:



¿Qué secuencia debe ejecutar el artista para dibujar el siguiente diseño? Cada uno de los lados de cada cuadrado mide 100 pixeles.

	Opción A repetir 3 veces haz my function girar a la derecha por 120 grados	Opción B repetir 3 veces haz my function girar a la derecha por 120 grados
	Opción C repetir 4 veces haz my function girar a la derecha por 90 grados	Opción D repetir 4 veces haz my function girar a la derecha por 90 grados

Seleccione una:

- a. Opción A
- b. Opción D
- c. Opción C
- d. Opción B

II. En la siguiente pregunta de solución al problema que se plantea, para ello implemente el programa, trabaje en grupo según los integrantes determinados por el docente del curso, utilice la herramienta Scratch.

Pregunta 12:

Elaborar animaciones off line empleando recursos digitales asistidos por la programación de los objetos y presentados bajo la estructurada de una historieta, la temática es sobre los siguientes temas:

- 1) ¿Quiénes somos?
- 2) ¿Cuáles son nuestras historias, leyendas, tradiciones y costumbres?
- 3) ¿Qué celebramos?

ANEXO 02

RÚBRICA PARA EVALUAR CAPACIDADES DE PENSAMIENTO COMPUTACIONAL

	%	Inadecuado 1	Puede mejorar 2	Satisfac- torio 3	Excelente 4
Formular problemas y darle solución usando el computador o usando otras herramientas.	20				
Establecer patrones y obtener conclusiones, mediante la organización lógica de los datos.	10				
Descomponer un problema en subproblemas definiendo abstracciones (modelos y simulaciones) para representar datos.	10				
Implementar soluciones a los problemas en forma creativa mediante el diseño de algoritmos.	20				
Depurar y levantar los errores que se pudieron presentar en su codificación.	10				
Optimizar los recursos y los algoritmos mediante la identificación, análisis e implementación de posibles soluciones a los problemas.	20				
Generalizar y realizar la transferencia de los procesos encontrados a problemas diversos.	10				

RÚBRICA PARA EVALUAR ACTITUDES DE PENSAMIENTO COMPUTACIONAL

	%	Inadecuado 1	Puede mejorar 2	Satisfac- torio 3	Excelente 4
Confianza en el manejo de la complejidad.	20				
Persistencia al trabajar en problemas de mayor nivel de complejidad.	25				
Tener tolerancia a la ambigüedad.	10				
Habilidad para afrontar problemas no estructurados.	20				
Habilidad para realizar trabajo en equipo logrando alcanzar soluciones comunes.	25				

ANEXO 03

POST - TEST

EVALUACIÓN DEL PENSAMIENTO COMPUTACIONAL

I. En las siguientes preguntas, marque la alternativa correcta.

Pregunta 01:

¿Qué órdenes lleva la Carolinito a su colegio, por el camino señalado?

	Opción A 	Opción B
	Opción C 	Opción D

Seleccione una:

- a. Opción B
- b. Opción A
- c. Opción D
- d. Opción C

Pregunta 02:

Para llevar al Carolinito a su colegio por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

--	--

Seleccione una:

- a. Opción C
- b. Opción D
- c. Opción A
- d. Opción B

Pregunta 03:

¿Cuántas veces se debe repetir la secuencia para llevar al Carolinito hasta su colegio por el camino señalado?

		Opción A x 3
		Opción B x 1
		Opción C x 2
		Opción D x 4

Seleccione una:

- a. Opción B
- b. Opción C
- c. Opción D
- d. Opción A

Pregunta 04:

Para que el gatito dibuje el siguiente exágono (100 pasos de lado), ¿en qué paso de la siguiente secuencia de órdenes hay un error?

		→ Opción A
		→ Opción B
		→ Opción C
		→ Opción D

Seleccione una:

- a. Opción A
- b. Opción B
- c. Opción D
- d. Opción C

Pregunta 05:

¿Qué secuencia de órdenes debe ejecutar el gatito para dibujar la escalera que llegue hasta la pelota? Cada peldaño sube 40 pasos.

	<p>Opción A</p> <pre> repetir 4 repetir 5 mover 40 pasos girar 90 grados mover 40 pasos </pre>	<p>Opción B</p> <pre> repetir 4 repetir 5 mover 40 pasos girar 90 grados mover 40 pasos </pre>
	<p>Opción C</p> <pre> repetir 5 repetir 4 mover 40 pasos girar 90 grados mover 40 pasos </pre>	<p>Opción D</p> <pre> repetir 5 repetir 4 mover 40 pasos girar 90 grados mover 40 pasos </pre>

Seleccione una:

- a. Opción C
- b. Opción D
- c. Opción A
- d. Opción B

Pregunta 06:

¿Qué órdenes llevan a "Pac-Man" hasta el fantasma por el camino señalado?

	<p>Opción A</p> <pre> Repetir hasta llegar a... hacer si hay un camino delante hacer avanzar sino girar a la izquierda </pre>	<p>Opción B</p> <pre> Repetir hasta llegar a... hacer si hay un camino delante hacer avanzar sino girar a la derecha </pre>
	<p>Opción C</p> <pre> Repetir hasta llegar a... hacer si hay camino a la derecha hacer girar a la derecha sino avanzar </pre>	<p>Opción D</p> <pre> Repetir hasta llegar a... hacer si hay camino a la izquierda hacer girar a la izquierda sino avanzar </pre>

Seleccione una:

- a. Opción D
- b. Opción C
- c. Opción A
- d. Opción B

Pregunta 07:

Para que "Pac-Man" llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

Seleccione una:

- a. Opción A
- b. Opción B
- c. Opción C
- d. Opción D

Pregunta 08:


¿Qué órdenes llevan a "Pac-Man" hasta las fresas por el camino señalado e indican a "Pac-Man" que se coma el número de fresas indicado?

Seleccione una:

- a. Opción B
- b. Opción A
- c. Opción C
- d. Opción D

Pregunta 09:

¿Qué falta en la siguiente secuencia de órdenes para que "Pac-Man" avance por el camino señalado comiendo el número de fresas indicadas?

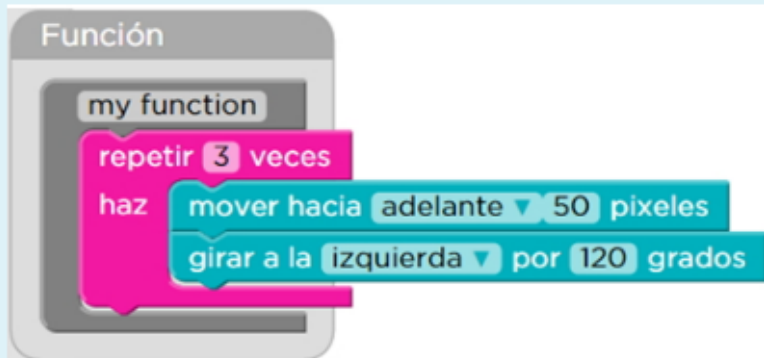
		
	Opción A 1 vez	Opción B 2 veces
	Opción C 3 veces	Opción D 5 veces

Seleccione una:

- a. Opción A
- b. Opción D
- c. Opción B
- d. Opción C

Pregunta 10:

Si tenemos el siguiente conjunto de órdenes, al que llamamos "my function", y que dibuja un triángulo de 50 píxeles de lado:



¿Qué le falta a la siguiente secuencia para que el artista dibuje el siguiente diseño? Cada uno de los lados de cada triángulo mide 50 píxeles.

<pre>repetir ??? veces haz my function saltar hacia adelante 50 píxeles</pre>	Opción A	Opción B
	Opción C	Opción D

Seleccione una:

- a. Opción D
- b. Opción C
- c. Opción A
- d. Opción B

Pregunta 11:

Si tenemos el siguiente conjunto de órdenes, al que llamamos "get5".



¿Qué órdenes van llevando a "Pac-Man" por el camino señalado e indicándole que se coma el número de fresas correspondiente?

	Opción A <pre>avanzar girar a la derecha repetir 5 veces haz avanzar get 5</pre>	Opción B <pre>avanzar girar a la derecha repetir 5 veces haz get 5 avanzar</pre>
	Opción C <pre>avanzar girar a la derecha repetir 5 veces haz avanzar get 5</pre>	Opción D <pre>avanzar girar a la derecha repetir 5 veces haz get 5 avanzar</pre>

Seleccione una:

- a. Opción B
- b. Opción D
- c. Opción C
- d. Opción A

Pregunta 12:

Si tenemos el siguiente conjunto de órdenes, al que llamamos "move and get 4".



¿Qué falta en la siguiente secuencia para llevar a "Pac-Man" por el camino señalado hasta las fresas, comiendo el número de fresas indicado?

Opción A	Opción B
3	4
Opción C	Opción D
5	6

Seleccione una:

- a. Opción B
- b. Opción D
- c. Opción C
- d. Opción A

II. En la siguiente pregunta de solución al problema que se plantea, para ello implemente el programa, trabaje en grupo según los integrantes determinados por el docente del curso, utilice la herramienta Scratch.

Pregunta 13:

Elaborar animaciones off line empleando recursos digitales asistidos por la programación de los objetos y presentados bajo la estructurada de una historieta, la temática es sobre los siguientes temas:

- 1) ¿Quiénes somos?
- 2) ¿Cuáles son nuestras historias, leyendas, tradiciones y costumbres?
- 3) ¿Qué celebramos?
- 4) ¿Cómo es nuestro medio ambiente y cómo lo cuidamos?
- 5) ¿Qué problemas ambientales tenemos?

El tema será sorteado antes de iniciar la evaluación.

ANEXO 04

FICHA DE EVALUACIÓN DEL PROTOTIPO DE SOFTWARE

ADAPTADO DEL ESTÁNDAR ISO - 9126

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
1. FUNCIONALIDAD					
Adecuación					
Exactitud					
Interoperabilidad					
Seguridad					
Conformidad					
2. FIABILIDAD					
Madurez					
Tolerancia a fallos					
Recuperabilidad					
Conformidad					
3. USABILIDAD					
Comprensibilidad					
Facilidad de aprendizaje					
Atracción					
Conformidad					
Operabilidad					
4. EFICIENCIA					
Comportamiento temporal					
Utilización de recursos					
Conformidad					
TOTAL					

VALIDACIÓN DE INSTRUMENTOS

JUICIO DE EXPERTO

Instrucción: Luego de analizar y cotejar el instrumento de investigación, con la matriz de consistencia de la presente, le solicitamos que en base a su criterio y experiencia profesional, valide dicho instrumento para su aplicación:

Nota: Para cada criterio considere la escala de 1 a 5 donde:

1. Muy poco	2. Poco	3. Regular	4. Aceptable	5. Muy aceptable
-------------	---------	------------	--------------	------------------

Criterio de validez	Puntuación					Argumento	Observaciones y/o sugerencias
	1	2	3	4	5		
Validez de contenido							
Validez de criterio metodológico							
Validez de intención y objetividad de medición y observación							
Presentación y formalidad del instrumento							
Total parcial							
TOTAL:							

Puntuación:

De 4 a 11 : No valida, reformular

De 12 a 14 : No valida, modificar

De 15 a 17 : Valida, mejorar

De 18 a 20 : Valida, aplicar

Apellidos y nombres:		
Grado académico:		
Mención:		Firma