

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



TITULO DE TESIS

**PYTHON Y LA MEJORA DEL PENSAMIENTO ALGORÍTMICO
EN LOS ALUMNOS DE LA ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMAS, 2022**

**LINEA DE INVESTIGACION
OTRAS INGENIERIAS Y TECNOLOGIAS**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

TESISTA: SANTIAGO ABAL VÍCTOR HUGO

ASESOR: Dr. HUAPAYA CONDORI FREDY RONALD

HUÁNUCO – PERÚ

2022

DEDICATORIA

A mis padres con mucho amor y cariño le dedico todo mi esfuerzo y trabajo puesto para la realización de esta tesis.

AGRADECIMIENTO

Mi agradecimiento a mi asesor por brindarme su apoyo para realizar mi tesis y los aportes en la elaboración del proyecto, a todos los que participaron en el estudio y participaron en la recolección de datos.

Agradezco especialmente a mi familia por el tiempo y trabajo durante todo el proceso.

RESUMEN

El presente trabajo de investigación se realizó empleando a los alumnos del primer semestre de la escuela de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizan; asimismo, el presente es un estudio de enfoque cuantitativo y es de tipo aplicada, y se seleccionó como diseño el tipo cuasi - experimental – longitudinal.

Para la recolección de datos se utilizó la técnica de la encuesta y se diseñó un test compuesto por 10 preguntas (variable independiente) el cual permitirá medir el nivel de pensamiento algorítmico. La muestra es de tipo no probabilística – no intencional y está conformada por 30 alumnos.

La confiabilidad del instrumento fue realizada mediante juicio de expertos lo que demuestra la validez del mismo. Las hipótesis serán comprobadas mediante la prueba de t para muestras relacionadas de esta manera se determinó la existencia de influencia entre las variables.

El resultado del estudio al aplicar la prueba de t nos da el valor de significancia menor a 0.05 ($0.000 < 0.05$) con lo cual se concluye que existe influencia entre la variable independiente “Lenguaje de Programación Python” y la variable dependiente “Pensamiento algorítmico”.

PALABRAS CLAVE: Pensamiento algorítmico, lenguaje de programación Python, análisis del problema, diseño de algoritmo y depuración del programa.

ABSTRACT

The present research work was carried out using the students of the first semester of the School of Systems Engineering of the National University Hermilio Valdizan; likewise, the present is a study of quantitative approach and is of applied type, and the quasi-experimental-longitudinal type was selected as a design.

For the data collection, the survey technique was used and a test composed of 10 questions (independent variable) was designed, which will allow measuring the level of algorithmic thinking. The sample is of a non-probabilistic – unintentional type and is made up of 30 students

The reliability of the instrument was carried out through expert judgment which demonstrates its validity. The hypotheses will be tested by t-testing for samples related in this way the existence of influence between the variables was determined.

The result of the study when applying the t-test gives us the significance value less than 0.05 ($0.000 < 0.05$) with which it is concluded that there is influence between the independent variable "Python Programming Language" and the dependent variable "Algorithmic thinking".

KEYWORDS: Algorithmic thinking, Python programming language, problem analysis, algorithm design and program debugging.

ÍNDICE

DEDICATORIA	ii
AGRADECIMIENTO	iii
RESUMEN	iv
ABSTRACT	v
ÍNDICE	vi
Índice de tablas	viii
Índice de ilustraciones	ix
INTRODUCCIÓN	x
CAPITULO I. PROBLEMA DE INVESTIGACIÓN	11
1.1. Fundamentación del problema de investigación	11
1.2. Formulación del problema	12
1.2.1. Problema General.	12
1.2.2. Problemas Específicos.....	12
1.3. Formulación de Objetivos generales y específicos.	13
1.3.1. Objetivo General.....	13
1.3.2. Objetivos Específicos.	13
1.4. Justificación	14
Viabilidad de la investigación.....	14
1.5. Limitaciones.....	14
1.6. Formulación de Hipótesis generales y específicas.....	15
1.6.1. Hipótesis general.	15
1.6.2. Hipótesis específicas.	15
1.7. Variables.....	16
1.8. Definición teórica y Operacionalización de variables.	16
CAPITULO II. MARCO TEÓRICO.....	18

2.1. Antecedentes	18
2.2. Bases teóricas	21
2.3. Definición de términos.....	39
CAPITULO III. MARCO METODOLÓGICO	41
3.1. Ámbito	41
3.2. Población	41
3.3. Muestra	41
3.4. Nivel y tipo de estudio.	41
3.5. Diseño de la investigación.	41
3.6. Técnicas e instrumentos de recolección de datos.	42
3.6.1. Técnicas	42
3.6.2. Instrumentos	42
3.7.1. Validación de los instrumentos para la recolección de datos.	42
3.8. Procesamiento y presentación de datos.....	43
CAPITULO IV. RESULTADOS.....	44
4.1. Análisis descriptivo.	44
4.2. Análisis inferencial y/o contrastación de hipótesis.	45
CAPITULO V. DISCUSIÓN	49
5.1 Discusión de resultados.....	49
5.2 Aporte científico de la investigación.	49
CONCLUSIONES	51
SUGERENCIAS	52
REFERENCIAS BIBLIOGRÁFICAS	53
ANEXOS	57
ANEXO 01. Matriz de Consistencia	58
ANEXO 02. Instrumento de pensamiento algorítmico.....	60

Índice de tablas

Tabla 1. Operacionalización de las variables de investigación	16
Tabla 2. Definición operacional de las variables.	17
Tabla 3. Validez del instrumento de control interno por juicio de expertos.....	43
Tabla 4. Resultado del pretest y postest del pensamiento algorítmico.	44

Índice de ilustraciones

Ilustración 1. Fases para elaborar un programa de computadora.....	35
Ilustración 2. Etapas a desarrollar en la fase de análisis de un problema.....	35
Ilustración 3. Esquema de investigación.	42
Ilustración 4. Resultados del nivel de pensamiento algorítmico pretest y postest. ..	44
Ilustración 5. Prueba de normalidad para los datos de la variable dependiente pretest y postest.	45
Ilustración 6. Prueba t para muestras relacionadas en la variable dependiente.	46
Ilustración 7. Prueba t para muestras relacionadas en las dimensiones de la variable dependiente del pretest y postest.	48

INTRODUCCIÓN

En la actualidad nos encontramos en una era digital donde se destaca la importancia del manejo de las tecnologías dentro de los jóvenes, se requiere fomentar el desarrollo de habilidades de pensamiento en general y competencias digitales, en especial a lo que refiere a la programación ya sea por medio de un software libre o comercial. Se considera a la programación como una competencia clave para el aprendizaje siendo esta útil a lo largo de la vida, por ello el aprendizaje de competencias digitales va más allá del desarrollo de habilidades de TIC.

El pensamiento computacional es un proceso el cual permite la resolución de problemas que tiene como características que emplea a la computadora y otras herramientas para abordarlos, además, organiza y analiza los datos de forma lógica y presenta soluciones a través de simulaciones y para automatizar estas soluciones emplea el pensamiento algorítmico.

Cabe destacar que el pensamiento algorítmico no es lo mismo que la capacidad de programar en una computadora, ya que, no se limita a una actividad mecánica sino más bien en él se emplea la imaginación y creatividad, por ello, se considera una competencia necesaria para afrontar los retos de la sociedad digital en especial en la formación de los alumnos que estudian carreras afines a tecnologías y programación.

El presente trabajo de investigación se realizó un test, a modo de prueba, a los alumnos de los primeros semestres de la escuela de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizan de la ciudad de Huánuco, para medir tres dimensiones de la variable dependiente pensamiento algorítmico, para ello, se desarrolló una propuesta de curso de programación en Python y posterior nuevamente se volvió a medir con el mismo test inicial para saber si la intervención con el curso permitió mejorar la competencia de pensamiento algorítmico en dichos alumnos.

La presente investigación consta de cinco capítulos en los cuales se detalla el estudio que se realizó.

CAPITULO I. PROBLEMA DE INVESTIGACIÓN

1.1. Fundamentación del problema de investigación

En la actualidad las organizaciones encargadas del ámbito educativo se cuestionan cuáles son las habilidades, competencias y capacidades que deberían alcanzar los estudiantes para adaptarse a esta nueva era. El pensamiento computacional es una de las estrategias que ha ganado popularidad, ya que su aplicación no solamente se limita a resolver problemas informáticos, sino que también tiene una aplicación en las diferentes áreas del conocimiento.

Según Wing (como citaron Bordignon e Iglesias, 2019), el pensamiento computacional permite resolver problemas, diseñar sistemas y comprender el comportamiento humano, en función a los conceptos fundamentales de la informática.

Se puede decir entonces, que el pensamiento computacional va más allá de programar o codificar, sino más bien contempla un proceso de formulación y comprensión del problema, para plantear el diseño y la evaluación de las soluciones. Entonces al ver al pensamiento computacional como un proceso cognitivo, Computing At School (CAS) determina que sus elementos claves son los siguientes:

- Capacidad de pensar en forma algorítmica.
- Capacidad de pensar en términos de descomposición.
- Capacidad de pensar en generalizaciones, identificando y haciendo uso de patrones.
- Capacidad de pensar en términos abstractos y elección de buenas representaciones.
- Capacidad de pensar en términos de evaluación. (Bordignon e Iglesias, p.29, 2019)

El pensamiento algorítmico es una de las capacidades básicas que compone al pensamiento computacional. Es una actividad cognitiva asociada a la resolución de problemas, a su especificación y comunicación de su solución; se aplican cuando existen problemas parecidos y que deben ser resueltos con determinada periodicidad,

estos se analizan en conjunto y se propone una solución que los contenga de forma general, para que pueda ser aplicada cada vez que estos ocurran.

Pensar de forma algorítmica es la capacidad de generar secuencias y reglas que sirvan para resolver problemas, siendo este un conocimiento básico que deben desarrollar los estudiantes afines a carreras en computación e informática, para escribir sus propios algoritmos, los cuales serán traducidos a instrucciones de acuerdo al lenguaje de programación que desean emplear (Python, Scratch, JavaScript, Go, etc.).

La realidad que presentan los estudiantes el primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL presenta un escenario propicio para la implementación de una estrategia que fomente el desarrollo del pensamiento algorítmico, competencia que les permitirá desenvolverse en diferentes escenarios y resolver problemas a lo largo de su carrera y fuera de ella.

Debido a su versatilidad y popularidad, así como su enfoque ágil, se optó por trabajar con Python en esta investigación; ya que es un lenguaje de programación que tiene una sintaxis clara y estructurada, lo cual permite generar un código legible, cercano al lenguaje natural, por ello es uno de los mejores lenguajes para comenzar a programar. Desarrollar con Python es como programar en pseudocódigo, es decir, como si estuviéramos diseñando un algoritmo.

1.2. Formulación del problema.

1.2.1. Problema General.

¿En qué medida la enseñanza del programa de Python mejora el pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL?

1.2.2. Problemas Específicos.

¿En qué medida la enseñanza del programa Python mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL?

¿En qué medida la enseñanza del programa Python mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos

del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL?

¿En qué medida la enseñanza del programa Python mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL?

1.3. Formulación de Objetivos generales y específicos.

1.3.1. Objetivo General.

Determinar en qué medida la enseñanza del programa Python mejora el pensamiento algorítmico en los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

1.3.2. Objetivos Específicos.

Determinar en qué medida la enseñanza del programa Python mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

Determinar en qué medida la enseñanza del programa Python mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

Determinar en qué medida la enseñanza del programa Python mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

1.4. Justificación

Justificación práctica

La programación es una de las capacidades principales que los ingenieros y estudiantes en ciencias de la computación esperan desarrollar; es un proceso de codificar el algoritmo que dará solución al problema planteado, utilizando un lenguaje de programación. En la práctica, la programación no requiere únicamente el conocimiento de la sintaxis del lenguaje para poder escribir un programa, sino que al mismo tiempo requiere de razonamiento, lógica y capacidad para describir de forma detalla las instrucciones que darán solución al problema, a este se le conoce como el pensamiento algorítmico. El desarrollo de esta habilidad permitirá que los estudiantes del primer año de la E.P. de Ingeniería de Sistemas de la UNHEVAL, puedan desarrollar algoritmos de media o alta complejidad, desarrollar un modelo viable o estructura que permita resolver un problema y que cuenten con la suficiente experiencia para manejar lenguajes de programación.

Justificación teórica

Se justifica de forma teórica, ya que este estudio permitirá obtener nueva información acerca de las variables en estudio y cada una de ellas está sustentada en fuentes teóricas confiables y antecedentes de estudios similares.

Viabilidad de la investigación.

Para el desarrollo del este estudio se contará con la disponibilidad de los recursos materiales, personales y monetarios.

Se considera viable ya que servirá como antecedente para futuras investigaciones donde se considere las mismas variables de estudio.

1.5. Limitaciones.

Limitación espacial

El ámbito de estudio de la presente investigación es la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizán de Huánuco.

Limitación temporal

La investigación de planificará y desarrollará desde enero hasta julio del año 2022

Limitación de recursos

Por las restricciones impuestas por el COVID 19, la recopilación de datos se realizará de forma virtual haciendo uso de cuestionarios elaborados en el Google form.

1.6. Formulación de Hipótesis generales y específicas.**1.6.1. Hipótesis general.**

H0: La enseñanza del lenguaje del programa Python no mejora el pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H1: La enseñanza del lenguaje del programa Python mejora el pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

1.6.2. Hipótesis específicas.

H0: La enseñanza del lenguaje del programa Python no mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H1: La enseñanza del lenguaje del programa Python mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H0: La enseñanza del lenguaje del programa Python no mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H1: La enseñanza del lenguaje del programa Python mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos

del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H0: La enseñanza del lenguaje del programa Python no mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H1: La enseñanza del lenguaje del programa Python mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

1.7. Variables

Variable independiente: Lenguaje de programación Python

Variable dependiente: Pensamiento algorítmico

1.8. Definición teórica y Operacionalización de variables.

Tabla 1. Operacionalización de las variables de investigación

Variable	Dimensiones	Indicadores
V.I. Lenguaje de Programación Python	Planificación de la enseñanza	Elaboración del syllabus del curso de Python.
	Desarrollo de la enseñanza	Desarrollo de los temas del curso de Python. Evaluación del aprendizaje.
V.D. Pensamiento Algorítmico	Análisis del problema	Comprender y determinar el problema para precisar los datos necesarios, la incógnita, las restricciones del problema y las operaciones necesarias.
	Diseño del algoritmo	Identificar los datos de entrada. Definir los métodos y fórmulas de acuerdo a las restricciones para procesar los datos. Definir instrucciones claras, precisas y ordenadas. Definir un número determinado de instrucciones. Obtener la solución del problema sin ambigüedades.
	Depuración del programa	Verificar los pasos, replanteándose la estrategia de solución en caso sea necesario. Verificar los resultados, replanteándose la estrategia de solución en caso sea necesario.

Fuente: Elaboración propia.

Tabla 2. Definición operacional de las variables.

VARIABLES	DEFINICIÓN OPERACIONAL
V.I. Lenguaje de Programación Python	El lenguaje de programación Python nos permitirá conocer conceptos previos sobre programación y escribir códigos de manera legible con una sintaxis clara y estructurada, similar a la de un algoritmo.
V.D. Pensamiento Algorítmico	El pensamiento algorítmico consiste en el desarrollo y uso de algoritmos que pueden ser utilizados para solucionar un problema específico o realizar una determinada tarea.

Fuente: Elaboración propia.

CAPITULO II. MARCO TEÓRICO

2.1. Antecedentes

A nivel internacional

Tejera, et al. (2020) en su artículo “Lenguajes de programación y desarrollo de competencias clave. Revisión sistemática”, busca responder las siguientes interrogantes con respecto a los lenguajes de programación en función a la educación formal: en qué etapas de la educación se usan lenguaje de programación, cuáles son los más usados y determinar el impacto de estos lenguajes en el aprendizaje de los alumnos por medio del estudio de la influencia en el desarrollo de las competencias clave. Para lograr este propósito realizo una búsqueda dentro del base de datos de Web of Science y Scopus entre el periodo 2007 y 2018 de acuerdo a determinados criterios y eliminando aquellos artículos que se repitieran; posterior a ello extrajo y analizo información a partir de una lectura a profundidad de los artículos seleccionados, los cuales fueron 61 en total. De este concluyo que a pesar de la tendencia a emplear los lenguajes de programación dentro de las etapas educativas se sigue considerando como una actividad destinada para la formación universitaria, así mismo pudo determinar que el lenguaje Scratch es el más empleado con un 22.4%, en una muestra de 15 artículos, frente a otros como Java, C++, Logo, etc. Por ultimo estableció la influencia entre la implementación del lenguaje de programación en las aulas y su beneficio en las competencias relacionadas con el uso de herramientas de forma interactivas y la de actuar de manera autónoma, la cual muestra que contribuye al conocimiento de sí mismo y a su capacidad y habilidad para la resolución de problemas.

Pérez (2017) es su investigación “Uso de Scratch como herramienta para el desarrollo del pensamiento computacional en Programación I de la carrera de Informática de la Universidad Central de Ecuador”, que tiene como objetivo analizar el desarrollo del pensamiento computacional en los estudiantes del 1er semestre de la Carrera de Informática de la Facultad de Filosofía haciendo uso de la herramienta Scratch como elemento didáctico en su desarrollo profesional, para lo cual se formaron dos grupos, uno experimental y el otro de control, conformados ambos por 40 estudiantes. Donde debido a los resultados obtenidos se concluye que luego de hacer uso de la herramienta Scratch no se consiguió desarrollar de manera sustancial el

pensamiento computacional en los estudiantes, sin embargo, se pudo evidenciar mejoras en la dimensión reconocimiento de patrones en los aspectos referidos al aprendizaje.

Marengo y De La Hoz (2017) en su investigación “Aplicación de estrategias metodológicas para el desarrollo del pensamiento lógico algorítmico en estudiantes de básica secundaria mediado con Scratch”, que estableció como objetivo aplicar estrategias metodológicas que permitan desarrollar el pensamiento lógico-algorítmico en estudiantes de básica secundaria haciendo uso de la herramienta Scratch, para la cual se contó con una muestra de 15 estudiantes y se usaron como instrumentos una cartilla con cinco tareas que el estudiante debía realizar para medir las habilidades relacionadas a su pensamiento computacional. Donde se concluyó que la herramienta Scratch contribuye a brindar una experiencia práctica de enseñanza y aprendizaje que permite desarrollar el pensamiento algorítmico de los estudiantes, gracias a las características que permiten la visualización de datos para idear y experimentar soluciones por medio de problemas.

A nivel nacional

Pardave y Yalico (2021) en su investigación “Influencia del lenguaje de programación Etoys en el área de educación para el trabajo para un aprendizaje cooperativo, en los alumnos del 4to “A” de la Institución Educativa Daniel Alcides Carrión – Pasco –2018”, que tiene como objetivo determinar la contribución que brinda el lenguaje Etoys en el fortalecimiento del aprendizaje cooperativo para la creatividad y las habilidades lógicas de los estudiantes bajo estudio, para lo cual se aplicaron cuestionarios a 20 alumnos del salón A y 20 alumnos del salón D del cuarto del 4to grado. Las conclusiones a las que llegaron muestran que existe una correlación moderada entre las variables de estudio, lo que implica que la programación tiene influencia amplia en los estudiantes, ya que les permite desarrollar sus capacidades cognitivas, fomentando que investigan más y de manera más significativa.

Pérez (2020) en su investigación “Actividades con Scratch para desarrollar el pensamiento algorítmico en estudiantes en una IEP de Chiclayo”, tiene por objeto proponer actividades con Scratch para desarrollar el pensamiento algorítmico en estudiantes de segundo año de nivel secundario de una institución educativa privada

de la ciudad de Chiclayo, el tipo de investigación realizado es descriptivo propositivo y aplicado a un solo grupo de estudio. Para el diagnóstico inicial se aplicó una prueba donde los resultados obtenidos fueron que un 52% desaprobó en la dimensión de resolución de problemas y 59% en la dimensión de algoritmos lo que manifiesta que se requiere una intervención para fomentar la capacidad del pensamiento algorítmico en dichos estudiantes.

Díaz y Lozano (2019) en su investigación “Uso de las aplicaciones Code.org y Scratch para el aprendizaje de programación en los estudiantes del 5° y 6° grado de EBR del C.E. N° 82099 de la provincia de San Pablo, 2018”, plantean como objetivo determinar si el uso de las aplicaciones Code.org y Scratch influyen en el aprendizaje de programación en los estudiantes del 5° y 6° grado de la institución bajo estudio, para lo cual se aplicó un cuestionario a 13 estudiantes para conocer el impacto de las lecciones que se dieron. Los resultados obtenidos permitieron concluir que ambas herramientas, Code.org y Scratch, son recursos educativos muy efectivos y tienen una influencia significativa en el aprendizaje de programación de los niños del 5° y 6° grado, además se pudo observar el potencial que muestran los estudiantes para iniciarse en el pensamiento computacional por medio de herramientas que despierten su interés en la programación.

A nivel local

Rivera (2020) en su investigación “Aplicación del lenguaje de programación Scratch para el desarrollo del pensamiento algorítmico en los alumnos del 6to grado del nivel primario en la I.E.P Augusto Cardich – Pillco Marca”, tiene como objeto desarrollar el pensamiento algorítmico de los alumnos de sexto grado de la I.E.P Augusto Cardich por medio de la implementación del lenguaje de programación Scratch. Esta investigación es de tipo aplicada y contó con dos grupos: el grupo experimental conformado por 23 alumnos y el de control por 19 alumnos, en ambos casos se tomaron las pruebas de conocimiento tanto de pretest como la de postest. En el grupo experimental se aplicará el lenguaje de programación Scratch para que los alumnos desarrollen el pensamiento algorítmico y posterior realizar la comparación con el grupo de control. Los resultados de las pruebas tomadas antes y después de las sesiones fueron procesadas mediante la prueba T para muestras relacionadas, el valor obtenido fue menor a 0.05 por lo cual podemos concluir que la aplicación del

lenguaje de programación Scratch influye de manera positiva en el desarrollo del pensamiento algorítmico por lo tanto se afirma que este ayuda a mejorar la capacidad analítica, ordena los pasos lógicos en un problema y mejora la toma de decisiones.

2.2. Bases teóricas

2.2.1. Lenguaje de Programación Python

✓ Definición

Bhasin (2019) señala que Python es un lenguaje de programación fuerte, orientado a objetos, con procedimientos claros y funcional, que fue creado por Guido Van Rossum a finales de 1980. Actualmente, este lenguaje es utilizado en distintos entornos de aplicación, entre los que se encontramos el desarrollo de software, desarrollo web, desarrollo de GUI de escritorio, educación, aplicaciones científicas, entre otros. Tal y como se observa, engloba casi todas las facetas del desarrollo, este dominio y popularidad se debe, entre otros factores, a su simplicidad y robustez.

✓ Características de Python

Python es un lenguaje simple pero bastante poderoso, Bhasin (2019) destaca algunas de sus principales características y fortalezas:

Fácil.

Es un lenguaje fácil de entender y aprender, si eres una persona con experiencia en la programación, verás en este algo elegante y ordenado. Al eliminar las llaves y paréntesis, el código se ve más corto y agradable, además que la ejecución de tareas dentro del mismo es bastante fácil. La simplicidad de Python hace que aprenderlo no sea una tarea complicada e intrincada.

Además, entender un proyecto escrito en Python es bastante fácil, ya que su código es conciso y efectivo, haciéndolo comprensible y manejable.

Escribir y ejecutar

A diferencia de muchos otros lenguajes de programación, en donde para probar algo se requieren muchos cambios y, por lo tanto, recopilaciones y repeticiones, en Python el código se puede ejecutar de manera fácil.

En Python se ejecutan scripts, además que brinda al usuario un entorno interactivo, en donde puede ejecutar comandos independientes.

Sintaxis

La sintaxis que ofrece Python es fácil, lo que facilita el proceso de aprendizaje y comprensión. Si podemos resaltar tres características principales de su sintaxis es que es simple, pequeña y flexible.

Mixing

Al trabajar en proyectos de gran escala, con un equipo de varias personas, podemos encontrarnos con miembros que utilicen y dominen otros lenguajes de programación. Frente a esto, Python permite que módulos de otros lenguajes puedan integrarse con su código central.

Escritura dinámica

Para administrar la memoria asociada a los objetos, Python realiza asignaciones dinámicas, es decir que, cuando termina el ciclo de vida del objeto, se le retira de la memoria, permitiendo una adecuada gestión de la misma y por consecuencia, haciendo los programas más eficientes.

✓ Tipos de objetos integrados

Python tiene incorporado tipos de objetos, lo que le permite realizar sus tareas de manera fácil y manejable. La manera en la que se manejan estos objetos, hacen de este un lenguaje eficaz, eficiente y bello.

Numerosas librerías y herramientas

Las tareas realizadas dentro de Python se realizan de manera muy fácil, esto debido a que la mayoría de tareas comunes (las poco comunes también), ya se han manejado dentro de este. Por ejemplo, Python posee bibliotecas que permiten a los usuarios desarrollar GUI, escribir aplicaciones móviles, incorporar funciones de seguridad e incluso leer resonancias magnéticas. Como se puede ver, posee una gran variedad de herramientas y bibliotecas que facilitan al desarrollador en tareas bastante complejas.

Portable

Un programa escrito en Python puede ser ejecutado en casi todas las plataformas conocidas, por ejemplo, en Linux, Windows o Mac. Además, es importante señalar que, Python fue escrito en el lenguaje C.

Gratis

Este lenguaje no es un software propietario. Cada usuario puede descargar compilados de Python entre una variedad de opciones disponibles. Además, hasta la fecha no se registraron problemas legales relacionados a la distribución de código desarrollo en Python.

✓ **Paradigmas En Python**

Procedimental

En un lenguaje procedimental, un programa viene a ser el conjunto de sentencias que se ejecutan secuencialmente. En este paradigma, la única opción que tiene un programa, desde el punto de vista de la manejabilidad, es dividir el programa en pequeños módulos. Python admite la programación de procedimientos.

Orientado a objetos

En este tipo de paradigma el lenguaje se orienta principalmente a la instancia de una clase, esta es llamada objeto. Una clase viene a ser una entidad real o virtual que cumple un papel en el problema en cuestión, además de poseer límites físicos definidos. Python está orientado a objetos.

Funcional

Python es compatible también con la programación funcional. Del mismo modo, este lenguaje acepta datos inmutables, optimización de colas, entre otros. (Bhasin, 2019)

✓ **Usos de Python**

Algunos usos del lenguaje de programación Python, son los siguientes:

- Desarrollo de interfaces gráficas de usuario (GUI).

- Creación de secuencias de comandos (scripts) para páginas web.
- Programación de base de datos.
- Prototipos.
- Juegos de azar.
- Programación basada en componentes.

Así mismo es importante señalar que en el caso de los sistemas operativos, Unix o Linux, Python viene preinstalado. Sin embargo, en los sistemas operativos de Windows o Mac, es necesario descargarlo para su instalación.

Algunos de los entornos de desarrollo que nos permiten escribir código para Python son los siguientes:

- PyDev con Eclipse.
- Emacs.
- Vim.
- TextMate.
- Gedit.
- Idle.
- PIDA en Linux (VIM based).
- NotePad++ en Windows.
- BlueFish en Linux. (Bhasin, 2019)

La elección de un lenguaje de programación es una decisión importante, ya que es una manera en que las personas pueden expresar sus habilidades creativas. Es aquí donde es indispensable considerar lo mencionado por Perlis (citado por Bhasin, 2019), quien indica que “no vale la pena conocer un lenguaje que no afecta la forma en que piensas sobre la programación”.

✓ Python en la Educación

García (2017) nos indica que debido a la gran acogida que tuvo Python se han diseñado herramientas visuales, enfocadas para programar

directamente, para verla ejecución de los mismos o para estudiar en tablets y smartphones. Algunas de las herramientas y recursos educativos generados por la comunidad de Python se pueden clasificar de la siguiente manera:

- Entornos interactivos: try.jupyter.org, Trinket.io, Snakify.org, codesters.com, pythonroom.com, repl.it, etc.
- Libros interactivos: interactivepython.org, learnpython.com, codeclub6, etc.
- Juegos y/o competiciones: checkio.org, empireofcode.com, codecombat.com, codeandconquer.co, etc.
- Ejercicios: exercism.io, practicepython.org, pybit.es, etc.
- Libros (libres): Think Python, Invent With Python, Dive Into Python 3, etc.
- Hardware: Raspberry Pi, Arduino, BBC:microbot, PyBoard, ESP8266, etc.
- Documentación: docs.python.org/3, PyMOTW3, Python Guide, Full Stack Python, etc.
- Librerías: PyGame, Python Arcade Library, Processing (Python Mode), VPython, Pilas Engine, etc.
- Herramientas para crear cursos: Jupyter Notebook, PyCharm Edu, Runestone Interactive, etc.
- Vídeos tutoriales: Dan Bader (dbader.org), Harrison Kinsley (pythonprogramming.net), Khan Academy, The Hello World Program, etc.

Además, como señalan Challenger et al. (2014), el lenguaje Python fue creado con la idea de ser fácil de aprender y con la idea de que en un futuro próximo la programación se convertirá en una asignatura tan relevante como las matemáticas y la física para los currículos de la enseñanza media. Algunas universidades como el MIT (Massachusetts Institute of Technology), seleccionaron este lenguaje para desarrollar las clases de Introducción a las Ciencias de la Computación y a la Programación e Introducción a los algoritmos; la Universidad de Chicago lo utiliza en el curso de Introducción a las Ciencias de la Computación 2; y en la Universidad de Jaume de España se usa en Introducción a la Computación. Como se puede apreciar,

la elección por parte de casas de estudio superiores para desarrollar cursos introductorios y avanzados, demuestra el crecimiento y avance de este lenguaje.

2.2.2. Pensamiento Computacional

✓ Definición

Para Zapata y Ros (como citaron Bordignon e Iglesias, 2019) el término pensamiento computacional (PC) es utilizado para referirse a técnicas y metodologías utilizadas para resolver problemas, que se caracterizan por hacer uso de la experiencia y saberes ligados a la programación de computadoras. Su uso no se limita solamente a problemas informáticos, sino que puede ser utilizado en diferentes ámbitos, para razonar y trabajar sobre distintas situaciones y áreas de conocimiento.

Además, en una publicación de Gurises Unidos (2017), destacan que el pensamiento computacional facilita a una toma de decisiones ordenada, con una secuencia clara, lógica, sin imprecisiones, cosa que es pocas veces vista en ámbitos de algunas ramas de la ciencia, como en los aspectos sociales.

Por lo tanto, el PC puede ser visto como una manera diferente de pensamiento asociada a la aparición de las computadoras, dado que fomenta un contexto cognitivo donde se reúnen el pensamiento ingenieril, el científico y el lógico matemático. Lo que significa que se requiere desarrollar un nivel de abstracción mayor con la finalidad de resolver problemas específicos del mundo real. Este tipo de pensamiento es aplicado (se recomienda) en el diseño de sistemas (por medio del modelado) y a la solución de problemas mediante la automatización en equipos de cómputo (pensamiento algorítmico y programación). (Bordignon e Iglesias, 2019)

✓ Resolución de problemas

Según la OECD (como citaron Gurises Unidos, 2017), podemos definir la resolución de problemas como “la capacidad de participar en un proceso cognitivo para entender y resolver problemas donde no hay un método de solución inmediatamente obvio” (p. 18)

Asimismo, para entender la resolución de problemas y comprender su función, podemos diferenciar dos formas principales. Una que implica procesos productivos de resolución de problemas, dentro de estos una persona genera soluciones a situaciones o problemas nuevos, luego de combinar diferentes reglas aprendidas con anterioridad. La segunda forma hace uso de procesos reproductivos, dentro de los cuales el conocimiento solo es recordado y utilizado en situaciones o problemas nuevos. (Gurises Unidos, 2017)

✓ **Características del PC**

Algunas características presentes en el pensamiento computacional, que no necesariamente están limitadas a estas, son las siguientes:

- Los problemas se formulan de tal manera que se pueda hacer uso de computadoras u otras herramientas para conseguir la solución.
- Los datos son organizados y analizados de manera lógica.
- Para la representación de datos se hace uso de modelos y simulaciones.
- Las soluciones son automatizadas haciendo uso del pensamiento algorítmico.
- Para establecer posibles soluciones se realiza un proceso de identificación, análisis e implementación. La solución se muestra a través de una secuencia de pasos, además se hace uso de los recursos de manera eficiente y eficaz.
- El proceso de solución de los problemas puede ser estandarizado y transferido a otros problemas. (Bordignon e Iglesias, 2019)

✓ **Destrezas y Habilidades Proporcionadas por el PC**

Algunas de las destrezas y habilidades que se desarrollan debido al acercamiento al pensamiento computacional son:

- Confianza al tratar con la complejidad.
- Persistencia al enfrentarnos con problemas difíciles.
- Tolerancia frente a lo ambiguo
- Capacidad para hacer frente a problemas abiertos y cerrados.

- Capacidad para poder trabajar y establecer una comunicación adecuada con los demás, en busca de una meta en común. (Gurises Unidos, 2017)

✓ **Dimensiones del Pc**

Resnick (como citaron Bordignon e Iglesias, 2019), descompone el pensamiento computacional en tres dimensiones:

- a) Conceptos computacionales. Aquellos empleados por los diseñadores en sus labores de programación, encontramos dentro de estas maneras de aportar en el aprendizaje generado por los estudiantes, a través de: algoritmos, programas, secuencias, ciclos, operadores, eventos, paralelismo, datos y paradigmas, entre otros.
- b) Prácticas computacionales. Desarrollados a medida que programan, encontramos dentro de estas maneras de aportar en el aprendizaje generado por los estudiantes, a través de: modelado, reutilización, abstracción, evaluación, modularización, documentación y metodología de desarrollo de proyectos.
- c) Perspectivas computacionales. Construidas por los diseñadores con respecto al mundo que los rodean y sobre sí mismos, encontramos dentro de estas maneras de aportar en el aprendizaje generado por los estudiantes, a través de: expresión, creatividad, trabajo e interacción con pares, pensamiento crítico sobre las tecnologías, sus usos y contextos.

✓ **Fines De Desarrollar El Pc**

Desarrollar el pensamiento computacional en los estudiantes persigue los siguientes fines:

- Desarrollar la capacidad de solucionar problemas y que las soluciones propuestas puedan ser automatizadas por una computadora.
- Que la relación que tienen con la tecnología deje de ser pasiva, para ser una más cercana, activa y productiva con recursos y

herramientas tecnológicas, con el objetivo de entender el mundo actual.

- Encontrar en los nuevos lenguajes informáticos una manera de poder mostrar y utilizar su capacidad creativa.
- Aquellos que muestren aptitudes por la ingeniería o las ciencias computacionales obtengan experiencias desde tempranas para involucrarse en estos temas pensando en su futuro. (Bordignon e Iglesias, 2019)

✓ **Capacidades Asociadas Al Pc**

Entendiendo el pensamiento computacional como un proceso cognitivo, que involucra el razonamiento lógico utilizado a resolver problemas, el CAS (2015) señala como capacidades claves los siguientes aspectos:

a) Capacidad de pensar en forma algorítmica.

Un algoritmo se conforma por un conjunto limitado de instrucciones, que indican una secuencia de operaciones específicas, las cuales siguen un orden determinado para resolver un problema.

El pensamiento algorítmico es una actividad cognitiva ligada a la resolución de problemas, su especificación y a la manera en la que se comunica su solución. Este es aplicado cuando nos enfrentamos a problemas semejantes que debemos resolver cada cierto periodo, por tal motivo se analizan en grupo y se elabora una solución general que se utiliza cada vez que ocurre un problema.

b) Capacidad de pensar en términos de descomposición.

En el campo de la informática, la descomposición se refiere al proceso de dividir un problema en partes más pequeñas y manejables. Cuando descomponemos un problema, la tarea de resolverlo se hace más fácil ya que reducimos su complejidad.

Cuando hablamos de descomponer algo, debemos verlo en términos de partes y componentes, en donde cada pieza debe ser comprendida, evaluada y solucionada de manera independiente. Así mismo, la solución propuesta a cada una de las partes puede delegarse a una persona o un equipo de trabajo, con lo cual la resolución de problemas complejos se hace en menor tiempo.

c) Capacidad de pensar en generalizaciones, identificando y haciendo uso de patrones.

Al descomponer un problema complejo, usualmente podemos encontrar patrones entre las divisiones de ese problema. Los patrones se formulan como aspectos compartidos entre los diversos problemas de complejidad menor. Al ser detectados, tenemos la posibilidad de trabajarlos en conjunto, lo que simplifica la labor de resolución. Un problema es más fácil de resolver cuando comparte patrones, ya que es muy posible que exista una solución diseñada por alguien más, lo cual nos permite aplicarlos a problemas con las mismas características.

La generalización es una tarea que guarda relación con la identificación de patrones, semejanzas y conexiones, para luego hacer uso de estas características. Se muestra como una forma rápida de resolver problemas, calcificándolos como versiones de viejos problemas con solución, es decir, haciendo uso de la experiencia.

d) Capacidad de pensar en términos abstractos y elección de buenas representaciones.

La abstracción puede definirse como un proceso que busca simplificar el entendimiento de una situación. Consiste en identificar el aspecto clave de algo, sin hacer caso a los detalles, por consecuencia, la complejidad de un problema puede ser administrada. El resultado de este proceso, es la construcción de una vista simplificada, también llamada la idea principal de algo.

Mediante este proceso, decidimos que aspectos debemos resultar y cuales otros podemos ignorar, una característica del pensamiento

computacional. Considerando que la idea principal de todo no está en los detalles, sino en los rasgos generales, al abstraer obtenemos una visión más comprensible de una situación. La abstracción se basa en la observación, la detección y la reducción de detalles, dejando de lado las cosas sin relevancia.

e) Capacidad de pensar en términos de evaluación.

Cuando hablamos de una tarea de evaluación, hacemos referencia a hacer juicios sobre algo, de una manera sistemáticas y bajo criterios definidos con anterioridad para que sea lo más objetiva posible.

En ámbito del pensamiento computacional, una vez elaborada la solución, es importante verificar que esta sea la adecuada para su propósito. El proceso de evaluación se aplica a una solución con la finalidad de asegurarnos que esta cumple con los requerimientos del diseño, además de que funcione correctamente. Cuando se trabaja con programas de computadora, la evaluación es una tarea sistemática e inflexible, porque de esta depende su efectividad y eficiencia.

Para verificar nuestra solución traducida en un algoritmo, debemos tener en cuenta los siguientes aspectos que forman parte de una prueba de evaluación:

- Debe ser entendida de manera fácil (¿Está completamente descompuesto?).
- Debe ser eficaz (¿Soluciona el problema?).
- Debe ser eficiente (¿Soluciona el problema haciendo un uso adecuado de los recursos disponibles?).
- Debe cumplir con los requerimientos de diseño detallados (¿Obedece a los requerimientos establecidos?).

Una vez que un algoritmo consigue superar el proceso de evaluación, cumpliendo con los criterios mencionado, podemos decir que estamos frente a una solución correcta; en consecuencia, podemos pasar a la etapa de programación en algún lenguaje informático. Pasar

por todo este proceso antes de programar, nos permite reducir dificultades, disminuyendo los errores, costos y tiempos del proyecto.

2.2.3. Pensamiento Algorítmico

✓ Definición

El pensamiento algorítmico puede ser entendido como un conjunto de habilidades que están relacionadas a la formulación y comprensión de algoritmos que permitan resolver problemas. Este tipo de pensamiento se conforma por la capacidad de analizar problemas, definir un problema de manera precisa, determinar las acciones necesarias para resolver dicho problema, diseñar un algoritmo eficaz para hacer frente al problema, pensar en los posibles escenarios de un problema y mejorar la eficiencia de un algoritmo. (López, 2014)

✓ Etapas de la Resolución de Problemas de Polya

El punto de vista de Pólya, en lo que respecta a la resolución de problemas, se fundamenta en una perspectiva global que no se restringe a un punto de vista matemático, es decir, propone la resolución de problemas como una serie de pasos que, si lo vemos detenidamente, aplicamos en cualquier aspecto de nuestra vida diaria. (Alfaro, 2006)

Según Polya (como citó López, 2014), cuando se pretende resolver un problema, entran en juego cuatro operaciones mentales:

1. Entender el problema.
2. Trazar un plan.
3. Ejecutar el plan (resolver).
4. Revisar.

✓ Entender o Comprender el Problema

Esta fase es fundamental para hacer frente a cualquier tipo de problema. Se recomienda, de ser necesario, leer varias veces y de manera pausada el problema, además hacer uso de la imaginación para llegar a una solución adecuada. Las acciones que se recomiendan para esta fase son las siguientes:

- Organizar la información solo con los datos necesarios, descartando los irrelevantes.
- Identificar los datos del problema.
- Identificar las incógnitas.
- Relacionar los datos obtenidos con las incógnitas identificadas.
- Reformular el enunciado del problema con otros datos.
- Diseñar un diagrama o esquema que permita ordenar las acciones que harán frente al problema.
- Definir los resultados deseados por medio de palabras claves para identificar las condiciones.
- Indicar si la información expuesta es insuficiente, poco clara, redundante y/o contradictoria. (Pérez, 2020)

✓ **Trazar o hacer el Plan**

Para diseñar las estrategias de solución y conseguir resolver el problema, en esta segunda fase se relacionan los datos obtenidos con las incógnitas identificadas. Las acciones que se recomiendan para esta fase son las siguientes:

- Realizar una comparación con otro problema parecido o semejante.
- Definir las operaciones que se llevaran a cabo.
- Hacer uso de los datos, incógnitas y condiciones.
- Reducir el problema en componentes más pequeños (descomponer).
- Relacionar, si fuera el caso, el problema identificado con algún teorema.
- Establecer estrategias de solución y seleccionar la más conveniente. (Pérez, 2020)

✓ **Ejecutar El Plan (Resolver)**

En dicha fase, se realiza de verificación de cada paso para evaluar el resultado obtenido, si este resultado no es el esperado, es recomendable analizar las estrategias utilizadas y determinar cuales se

deben modificar para posteriormente repetir el proceso, hasta conseguir los resultados deseados. Las acciones que se recomiendan para esta fase son las siguientes:

- Llevar a cabo las instrucciones paso a paso.
- Desarrollar las operaciones.
- Diseñar un diagrama. (Pérez, 2020)

✓ **Analizar La Solución (Revisar)**

Para esta última fase, se procede a realizar una revisión y verificación de los resultados obtenidos, para lo cual se debe considerar lo siguiente:

- Leer el enunciado del problema y verificar el resultado obtenido.
- Resolver el problema de una manera diferente.
- Explicar cómo se obtuvo el resultado obtenido.
- Redefinir la estrategia de solución de no obtenerse la respuesta correcta. (Pérez, 2020)

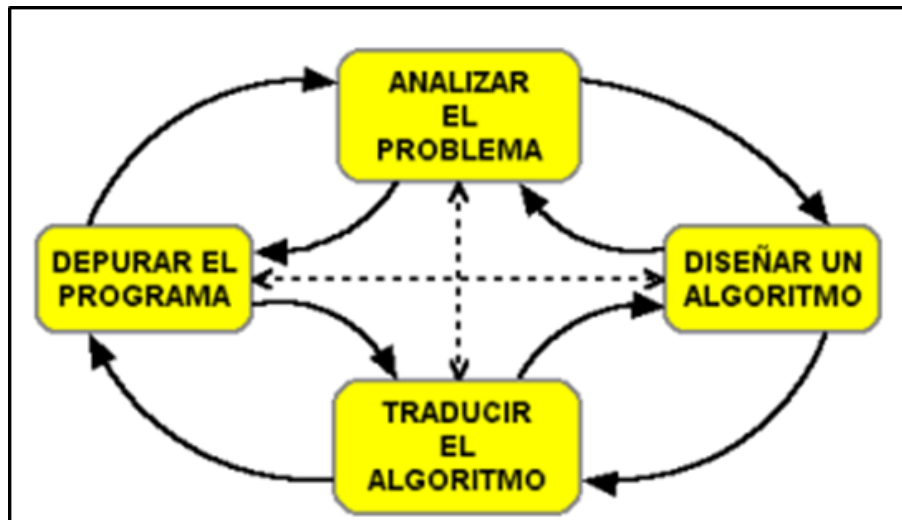
✓ **Fases de Polya Adaptado a la Resolución de Problemas de Acuerdo al Ciclo de Programación**

Según López (2009) existe diversas fuentes bibliográficas que mencionan a cuatro fases para elaborar un procedimiento que realizan una tarea particular, estas guardan relación con las fases mencionadas por Polya para la resolución de problemas.

Como se puede apreciar hay una similitud entre las antes mencionadas.

- Analiza el problema – Entender el problema
- Diseña un algoritmo - Trazar un plan
- Traducir el algoritmo a un lenguaje de programación – Ejecutar el plan
- Depurar el programa – Revisar

Ilustración 1.Fases para elaborar un programa de computadora.



Fuente: López (2009, p.11)

✓ Analizar El Problema

Esta etapa López (2009) menciona que se requiere definir el problema hasta que se consiga su completa comprensión, para ello es necesario formular claramente el problema, tener claro cuáles son los resultados a conseguir, definir los datos relevantes, identificar cuáles son las restricciones del problema y definir las operaciones necesarias para lograr su solución.

Las etapas anteriormente mencionadas coinciden con lo planteado por Schunk, a excepción de la etapa de formular el problema y determinar las restricciones.

Ilustración 2.Etapas a desarrollar en la fase de análisis de un problema.



Fuente: López (2009, pp.11)

En el siguiente párrafo se define las etapas que componen a la fase de análisis del problema:

- Formular el problema, consiste en tener el problema de forma verbal y que este puede ser formulado y comprendido; la información será categorizada y ordenada para que pueda ser manifestada de forma precisa, entendiendo claramente lo que se tiene que hacer.
- Precisar los resultados esperados, consiste en establecer los resultados que se esperan es decir lo que debe devolver el programa y como se deben ser presentados.
- Identificar los datos disponibles, en esta etapa se debe identificar y determinar qué información es relevante para dar solución al problema.
- Determinar las restricciones, consiste en determinar las acciones que se pueden realizar y las que no para llegar a la solución del problema.
- Establecer procesos, consiste en determinar los procesos que van a permitir conseguir los resultados que se esperan en función a los datos que se encuentren disponibles.

✓ **Diseñar un Algoritmo**

Pérez (2020) menciona que en esta fase se elabora un algoritmo empleando un lenguaje natural ya sea mediante diagramas de flujos o pseudocódigos de tal manera que sea entendible por las computadoras, de forma independiente al lenguaje de programación que se pretenda utilizar.

Así mismo, señala que los algoritmos contemplan características las cuales deben cumplir y estas son:

- Tienen una entrada (datos de entrada)
- Tienen una salida (datos de salida)

- Tienen un número finito de pasos para la resolución de un problema (es finito)
- Los procesos se deben expresar de forma clara, concisa y sin ambigüedades.
- Se considera eficiente de acuerdo al tiempo que se tarde en resolver un problema.

Los algoritmos son representados mediante: diagramas de flujo o pseudocódigos.

- a) Los diagramas de flujo: Según Fernández (2014) es un diagrama que permite expresar de forma gráfica las operaciones que componen a un procedimiento en forma cronológica.

Mientras que Alonso (2017) nos indica que los diagramas de flujo permiten la representación de los algoritmos de forma gráfica empleando para ello símbolos, flechas, figuras geométricas y otros elementos que permitan su entendimiento. Por medio de ella se puede comprender el recorrido del procedimiento, desde el planteamiento inicial hasta la solución del problema, viajando por los caminos de acuerdo a las condiciones que indique el algoritmo.

Así mismo menciona que la Internacional Organization for Standardization (ISO) establece una opción en cuanto a la estandarización de los símbolos a emplean para el diseño de un diagrama de flujo dentro de su documento International Standard 1028 “Information Processing – Flowchart Symbols” así mismo en la American National Standard, en el documento Flowchart Symbols and their Usage in Information Processing, ANSI X3.5-1970.

Algunas de las reglas más usadas para el diseño de diagramas de flujo son:

- Todos los símbolos del diagrama de flujo deben estar conectado mediante flechas.

- El diseño de diagrama de flujo por lo general se realiza manteniendo un orden vertical, que va de arriba hacia abajo, aunque en ocasiones puede ir a de izquierda a derecha.
 - Se debe emplear un símbolo de terminal que indique el inicio y fin del procedimiento.
- b) Los pseudocódigos: De acuerdo con Alonso (2017) que los pseudocódigos son un instrumento que permite realizar una descripción informal de alto nivel del principio de funcionamiento de un algoritmo, así mismo que contempla una estructura que puede ser entendida por cualquier lenguaje de programación que pretenda utilizarse, sin embargo, está dirigida principalmente al entendimiento de un ser humano.

En el caso de los pseudocódigos no existen propiamente reglas de sintaxis estrictas, pero si existen notaciones técnicas, empleadas con el mismo fin que el pseudocódigo con las estructuras más definidas como es el caso de la notación de Backus – Naur.

Los pseudocódigos son importantes en el primer acercamiento a la programación ya que permite enseñar a programar a los estudiantes de manera más sencilla porque cuenta con un lenguaje comprensible y no requiere que se conozca un lenguaje de programación convencional para entenderlo.

✓ **Traducir el Algoritmo**

Una vez el algoritmo ya sea en pseudocódigo o diagrama de flujo se encuentre diseñado se pasa a traducir al lenguaje de programación que se va emplear, se debe de tener en cuenta que cada lenguaje posee sus propias reglas gramaticales, por lo mismo, es importante que antes de realizar la traducción se conozca la sintaxis de los comandos. A la fase de traducción también se le conoce como codificación. (López, 2009)

✓ **Depurar el Programa**

Los algoritmos una vez traducidos al lenguaje de programación deben ser probados y validados para verificar si los resultados son los esperados. La depuración de un programa permite mejorar en los estudiantes la capacidad de resolver problemas, sin olvidar que está acompañada de retroalimentación le servirá para todos los ámbitos de la vida.

Esta etapa permite que se realicen retoques a los programas en cuanto a la presentación de los resultados, así como en las funcionalidades. (López, 2009)

✓ **Marco situacional.**

Como ya se argumentó que el pensamiento algorítmico no es lo mismo que la capacidad de programar en una computadora, ya que, no se limita a una actividad mecánica sino más bien en él se emplea la imaginación y creatividad, por lo tanto desconocemos en qué medida la enseñanza del programa Python mejora el pensamiento algorítmico en los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL así como también desconocemos: en qué medida la enseñanza del programa Python mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL; en qué medida la enseñanza del programa Python mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL y qué medida la enseñanza del programa Python mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL

2.3. Definición de términos.

Abstracción: Es la habilidad que permite que el ser humano pueda hacer frente a la complejidad, debido a que considera solo lo esencial del objeto o fenómeno al cual se está enfrentando. (Zapotecatl, 2018)

Algoritmo: Un algoritmo es una cadena de pasos que siguen un orden determinado para resolver un problema. (Zapotecatl, 2018)

Analizar: Es dividir en componentes más pequeños un problema, disminuyendo su complejidad, identificando los procesos y la búsqueda de rasgos o patrones comunes. Se busca hacer uso del pensamiento lógico, para un mejor entendimiento de las cosas y evaluar su apropiado propósito. (Code, 2016)

Automatización: Esta habilidad nos permite ordenar a una computadora como debe realizar una tarea repetitiva, haciéndola de manera más eficiente que un ser humano. (Raimundo, 2019)

Enseñanza: Se entiende como la trasmisión de conocimientos por parte del maestro con la finalidad de que los alumnos adquieran aprendizajes. (Fortoul, 2018)

Lenguaje de programación: Está conformado por una serie de símbolos básicos, alfabeto y reglas establecidas para poder manipularlos. Estos pueden clasificarse considerando su similitud con el lenguaje máquina o su similitud con el lenguaje humano. Se utilizan para poder comunicar y desarrollar aplicaciones, programas, instrucciones, etc. Ejecutadas por un ordenador. (López et al, 2009)

Paradigma: Hace referencia al conjunto de creencias, compromisos, modelos de ver el mundo compartidas por miembros de un grupo o comunidad, estas generan y controlan su manera de actuar y pensar, pueden ser evidenciadas por medio de patrones, modelos y reglas. (Gonzales, 2005)

Patrón: El ámbito de reconocimiento de patrones engloba algoritmos, metodologías, teorías y sistemas que nos posibilitan diferenciar, asociar y clasificar los datos obtenidos de manera automatizada. (Zapotecatl, 2018)

Programar: Definir la estructura y el comportamiento que tendrá un programa, asimismo, verificar que el programa cumpla sus tareas adecuadamente y haciendo un uso adecuado de recursos. (Hernández, 2014)

Problema: Es una situación que implica un no saber y, por consiguiente, una necesidad para hallar una solución. (Gerencia de Proyectos Informáticos, s.f., p. 3)

CAPITULO III. MARCO METODOLÓGICO

3.1. **Ámbito**

Universidad Nacional Hermilio Valdizan de Huánuco, E.A.P. de Ingeniería de Sistemas, alumnos del Primer año de estudios 2022.

3.2. **Población**

En la presente investigación la población estará conformada por 100 estudiantes del primer año de 2022 de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizán.

3.3. **Muestra**

La muestra es de tipo no probabilística intencionada, ya que “el investigador selecciona según su propio criterio, sin ninguna regla matemática o estadística” (Carrasco, 2005). Está constituida por los 30 estudiantes matriculados en el primer año del 2022 de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizán que estén matriculados.

Criterios de inclusión y exclusión.

- **Criterios de inclusión:** Estudiantes matriculados en el del primer año del 2022 que tengan la condición de alumno regular de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizán.

- **Criterios de exclusión:** Estudiantes no regulares de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizán.

3.4. **Nivel y tipo de estudio.**

La investigación tiene un enfoque cuantitativo. Es de tipo aplicada porque “establece propósitos prácticos inmediatos bien establecidos” Carrasco (2005) y es de nivel explicativo, ya que como señala Carrasco (2005) “responde a las preguntas ¿qué cambios y modificaciones se han producido? ¿qué mejoras se han logrado? ¿cuál es la eficiencia del nuevo sistema?”.

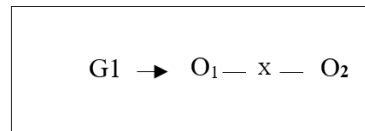
3.5. **Diseño de la investigación.**

El diseño para la presente investigación es cuasi experimental con un grupo experimental, al cual se le aplicará dos pruebas, una primera antes de iniciar las

sesiones de aprendizaje (pre test) y otra al finalizar dichas sesiones (post test), con la finalidad de identificar la manera en la que se relacionan las variables en estudio.

En los diseños cuasi experimentales “los grupos de trabajo ya se encuentran establecidos, es decir, existen antes del experimento” (Carrasco, 2005).

Ilustración 3.Esquema de investigación



Fuente: Elaboración propia.

Donde:

G1 = Muestra

O₁ = Observación (pre test)

O₂ = Observación (post test)

x = Uso del lenguaje Python

3.6. Técnicas e instrumentos de recolección de datos.

3.6.1. Técnicas

Como técnica para la recolección de datos, se usó un cuestionario (test), ya que esta nos permite recabar información de cada estudiante sobre las variables de estudio.

3.6.2. Instrumentos

Se diseñó una prueba (test) que consta de 10 preguntas, las cuales tiene por objetivo recabar información que nos permita medir el nivel de las dimensiones del pensamiento algoritmo en los estudiantes de Primer Año de la Escuela Profesional de Ingeniería de Sistemas.

3.7. Validación y confiabilidad del instrumento

3.7.1. Validación de los instrumentos para la recolección de datos.

El instrumento que se aplicó en la presente investigación fue validado a través de la opinión de tres expertos, los cuales realizarán un análisis a las

preguntas propuestas para determinar si tiene relación con los objetivos, variables, dimensiones e indicadores.

Tabla 3. Validez del instrumento de control interno por juicio de expertos.

N°	Apellidos y Nombres	Aplicable
01	Abimael Francisco Paredes	X
02	Inés Eusebia Jesús Tolentino	X
03	Ronal Fredy Huapaya Condori	X

Fuente: Elaboración propia.

3.8. Procesamiento y presentación de datos.

Para el procesamiento de información se utilizarán los programas de SPSS y Microsoft Excel.

El análisis de los datos se presentó a través de tablas de frecuencia y figuras para resumir la información. La validación de la hipótesis se realizó mediante la prueba paramétrica T de Student para estudios longitudinales con muestras relacionadas (se basa en el valor del alfa 0.05 o 5%). Ya que se tomará la medida (prueba pre test) para evaluar el nivel de pensamiento algorítmico en los estudiantes de primer semestre de la E.P de Ingeniería de Sistemas y luego otra medida (prueba post test) una vez se haya aplicado el curso de Python. En este caso nuestra elección de la prueba será la T de Student (muestras relacionadas)

Antes de la aplicación de la prueba se va corroborar que existe un comportamiento normal de la variable, es decir, se verificará el supuesto de normalidad. Esto se puede realizar mediante dos pruebas:

- Kolmogorov-Smirnov para muestras grandes (>30 individuos)
- Chapiro wilk para muestras pequeñas (<30 individuos).

CAPITULO IV. RESULTADOS

4.1. Análisis descriptivo.

En este aspecto con la finalidad de presentar los resultados que se obtuvieron en el test, los cuales han sido procesados usando el programa Excel 2016. Se mostrarán los resultados de la variable dependiente: Pensamiento algorítmico, con sus dimensiones: Análisis del problema, diseño del algoritmo y depuración del programa.

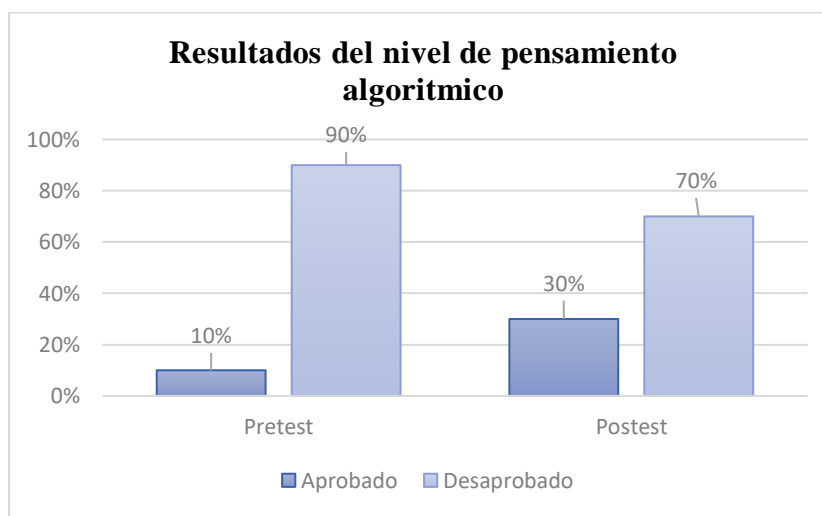
Tabla 4. Resultado del pretest y postest del pensamiento algorítmico.

Total de estudiantes	30		
Resultados del pretest:			
Aprobado	3	Desaprobado	27
Resultados del postest:			
Aprobado	9	Desaprobado	21

Fuente: Elaboración propia.

La tabla 04 muestra la cantidad de alumnos que aprobaron (donde la nota obtenida es mayor a 10) y desaprobado (nota obtenida es menor a 11) contando con un total de 30 alumnos de cada dimensión del pensamiento algorítmico. En el pretest aplicado se obtuvo un 10 % (3 alumnos) aprobados y 90 % (27 alumnos) desaprobados. En el postest aplicado se obtuvo un 30 % (9 alumnos) aprobados y 70 % (21 alumnos) desaprobados.

Ilustración 4. Resultados del nivel de pensamiento algorítmico pretest y postest.



Fuente: Elaboración propia.

4.2. Análisis inferencial y/o contrastación de hipótesis.

Primero se debe calcular la prueba de valor necesario para determinar si la variable dependiente se comporta normalmente, es decir, verificar el supuesto de normalidad. En este caso aplicaremos la prueba de Shapiro wilk ya que estaremos trabajando con una muestra de 30 sujetos (alumnos).

El análisis de esta prueba se realizará con base en la siguiente hipótesis:

H_0 : Los datos proviene de una distribución normal.

H_a : Los datos NO provienen de una distribución normal.

De acuerdo a los resultados mostrados en la ilustración 05, se puede observar que el valor de significancia del análisis de normalidad con la prueba Shapiro wilk del pretest 0.187 siendo este mayor a 0.05, así mismo, en el caso de la postest donde tenemos un valor de 0.544 en función a lo descrito anteriormente se puede concluir que los datos provienen de una distribución normal.

Ilustración 5. Prueba de normalidad para los datos de la variable dependiente pretest y postest.

Pruebas de normalidad						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Vd_Pretest	,129	30	,200 [*]	,952	30	,187
VD_Postest	,165	30	,037	,970	30	,544

*. Esto es un límite inferior de la significación verdadera.
a. Corrección de significación de Lilliefors

Fuente: Elaboración propia.

Para la comprobación de la hipótesis general se empleará la prueba t para muestras relacionadas.

El análisis de esta prueba se hará con base en la siguiente:

H_0 : La enseñanza del lenguaje del programa Python no mejora el pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H_a : La enseñanza del lenguaje del programa Python mejora el pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

Se muestra nuestra significancia con un valor de $p = 0.000 \leq a 0.05$, por lo tanto, se acepta la hipótesis alterna donde se afirma que la aplicación del curso de lenguaje de programación Python influye positivamente en el desarrollo del pensamiento algorítmico de los alumnos de primer semestre de la Escuela de Ingeniería de Sistemas. Ver en la ilustración 06.

Ilustración 6. Prueba t para muestras relacionadas en la variable dependiente.

Estadísticas de muestras emparejadas							
		Media	N	Desv. Desviación	Desv. Error promedio		
Par 1	Vd_Pretest	6,20	30	3,508	,640		
	VD_Postest	8,67	30	4,088	,746		
Correlaciones de muestras emparejadas							
		N	Correlación	Sig.			
Par 1	Vd_Pretest & VD_Postest	30	,717	,000			
Prueba de muestras emparejadas							
		Diferencias emparejadas				t	Sig. (bilateral)
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		
					Inferior	Superior	
Par 1	Vd_Pretest - VD_Postest	-2,467	2,909	,531	-3,553	-1,380	-4,644
							,000

Fuente: Elaboración propia.

Para la comprobación de la hipótesis general se empleará la prueba t para muestras relacionadas.

El análisis de esta prueba se hará con base en la siguiente:

H0: La enseñanza del lenguaje del programa Python no mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H1: La enseñanza del lenguaje del programa Python mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H0: La enseñanza del lenguaje del programa Python no mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H1: La enseñanza del lenguaje del programa Python mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H0: La enseñanza del lenguaje del programa Python no mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

H1: La enseñanza del lenguaje del programa Python mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

Se muestra nuestra significancia en las tres dimensiones evaluadas con un valor de $p = 0.000 \leq a 0.05$, por lo tanto, se acepta las hipótesis alternas donde se afirma que las dimensiones análisis del problema, diseño de algoritmo y depuración del programa fueron influenciadas por la aplicación del curso de lenguaje de programación Python positivamente. Ver en la ilustración 07.

Ilustración 7. Prueba t para muestras relacionadas en las dimensiones de la variable dependiente del pretest y postest.

Estadísticas de muestras emparejadas					
		Media	N	Desv. Desviación	Desv. Error promedio
Par 1	Análisis del problema - Pretest	1,27	30	1,507	,275
	Análisis del problema - Postest	2,40	30	1,754	,320
Par 2	Diseño de algoritmo - Pretest	2,27	30	1,929	,352
	Diseño de algoritmo - Postest	3,27	30	1,946	,355
Par 3	Depuración del programa - Pretest	2,87	30	1,688	,308
	Depuración del programa - Postest	3,00	30	1,762	,322

Correlaciones de muestras emparejadas				
		N	Correlación	Sig.
Par 1	Análisis del problema - Pretest & Análisis del problema - Postest	30	,663	,000
Par 2	Diseño de algoritmo - Pretest & Diseño de algoritmo - Postest	30	,715	,000
Par 3	Depuración del programa - Pretest & Depuración del programa - Postest	30	,904	,000

Prueba de muestras emparejadas						t	df	Sig. (bilateral)	
		Diferencias emparejadas			95% de intervalo de confianza de la diferencia				
		Media	Desv. Desviación	Desv. Error promedio	Inferior	Superior			
Par 1	Análisis del problema - Pretest - Análisis del problema - Postest	-1,133	1,358	,248	-1,640	-,626	-4,572	29	,000
Par 2	Diseño de algoritmo - Pretest - Diseño de algoritmo - Postest	-1,000	1,462	,267	-1,546	-,454	-3,746	29	,001
Par 3	Depuración del programa - Pretest - Depuración del programa - Postest	-,333	,758	,138	-,616	-,050	-2,408	29	,023

CAPITULO V. DISCUSIÓN

5.1 Discusión de resultados.

Para la realización de la presente investigación fue necesario la revisión de tesis y otras investigaciones realizadas anteriormente que tengan similitudes en cuanto al tema abordado.

En el artículo de Tejera, et al. (2020) se menciona que existe influencia entre la implementación de un lenguaje de programación y su beneficio en las competencias relacionadas con el uso de herramientas de forma interactivas y la de actuar de manera autónoma, la cual muestra que contribuye al conocimiento de sí mismo y a su capacidad y habilidad para la resolución de problemas. Lo antes mencionado reafirma nuestra conclusión que establece que existe una influencia significativa entre la aplicación de un curso de lenguaje de programación Python para desarrollar el pensamiento algorítmico en los alumnos del primer semestre de la escuela de Ingeniería de Sistemas.

Así mismo, Rivera (2020) llega a concluir en su investigación que la aplicación del lenguaje de programación Scratch influye de manera positiva en el desarrollo del pensamiento algorítmico por lo tanto se afirma que este ayuda a mejorar la capacidad analítica, ordena los pasos lógicos en un problema y mejora la toma de decisiones, bajo el mismo enfoque, en esta investigación se concluye que la aplicación de una intervención mediante el lenguaje de programación Python influye en el desarrollo de la capacidad de análisis y solución de problemas ya que desarrolla la creatividad y el análisis en los estudiantes para proponer soluciones prácticas a problemas.

5.2 Aporte científico de la investigación.

Es relevante desarrollar las competencias, capacidades, desempeños y habilidades de los alumnos del primer semestre de la escuela de Ingeniería de Sistemas de la UNHEVAL de la ciudad de Huánuco, para lograr dicho objetivo, en esta investigación se propuso la aplicación de un curso del lenguaje de programación Python para el desarrollo de pensamiento algorítmico en los alumnos mencionados. El uso del lenguaje Python se adapta al desarrollo de estas habilidades ya que es fácil de

aprender, además, que ofrece una sintaxis simple lo que facilita el proceso de aprendizaje y comprensión.

Así mismo, este trabajo de investigación mediante su desarrollo pretende aportar a manera de demostrar que al implementar la enseñanza del lenguaje de programación Python en los alumnos relacionados a carreras de tecnologías y programación esta permite que desarrollen la competencia del pensamiento algorítmico, lo cual, les servirá no solo para desarrollar programas sino para resolver problemas de forma organizada, sencilla y creativa en otras asignaturas y en la vida diaria.

Es importante también mencionar que el estudio puede ser empleado como material de consulta para otras investigaciones que tengan como objeto implementar intervenciones utilizando este lenguaje de programación en carreras universitarias.

CONCLUSIONES

- Al realizar la aplicación del lenguaje de programación Python se evidencia una mejora en el desarrollo del pensamiento algorítmico, ya que, los estudiantes en los primeros semestres de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizán que llevaron las asignaturas de Programación I y Fundamentos de la Programación y Computación pueden secuencializar mejor los pasos a seguir en la resolución de problemas.
- Los estudiantes de los primeros semestres de la Escuela Profesional de Ingeniería de Sistemas después de la aplicación de las sesiones del curso pueden analizar e identificar mejor los problemas; pueden representar gráficamente mejor las instrucciones a seguir en un problema y revisan y corrigen errores que podrían ocurrir en un problema en comparación a antes de la ejecución del curso.

SUGERENCIAS

- Se sugiere tomar en cuenta la presente investigación como un inicio al desarrollo de pensamiento computacional para su posterior investigación.
- Se sugiere que consideren al lenguaje Python como parte del contenido de los cursos de programación, ya que, permite fortalecer las habilidades de solucionar problemas y así mejorar la toma de decisiones.

REFERENCIAS BIBLIOGRÁFICAS

- Alfaro, C. (2006). Las ideas de Pólya en la resolución de problemas. *Cuadernos de Investigación y Formación en Educación Matemática*, 1(1), 1-13.
- Alonso Urbano, D. (2017). Scratch como herramienta para la enseñanza de programación en la Educación Primaria. Análisis de usabilidad en la escuela pública de la Comunidad de Madrid. Madrid, España. Obtenido de <https://repositorio.ucjc.edu/bitstream/handle/20.500.12020/516/Tesis%20%20doctoral%20-%20David%20Alonso%20Urbano.pdf?sequence=1&isAllowed=y>
- Begoña Abad, M. (2016). Investigación social cualitativa y dilemas éticos: de la ética vacía a la ética situada. *Revista de Metodología de Ciencias Sociales*(34), 101-120. doi:1139-5737, DOI/empiria.34.2016.16524
- Bhasin, H. (2019). *Python Basics - A Self-Teaching Introduction*. Dulles, Virginia: MERCURY LEARNING AND INFORMATION.
- Bordignon, F., & Iglesias, A. (2019). *Introducción al Pensamiento Computacional*. Buenos Aires: EDUCAR. Obtenido de <https://unipe.educar.gob.ar/storage/app/file/ckeditor/introduccion-pensamiento-computacional-5e581511a29dc.pdf>
- Carrasco Díaz, S. (2005). *Metodología de la investigación científica*. Lima: Editorial San Marcos.
- CAS (Computing At School). (2015). Pensamiento Computacional. Guía para profesores. Obtenido de Recuperado de <http://www.codemas.org/wp-content/uploads/2016/04/>
- Challenger Pérez, I., Díaz Ricardo, Y., & Becerra García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, XX(2), 1-13. Obtenido de <https://www.redalyc.org/pdf/1815/181531232001.pdf>
- Code. (2016). Pensamiento Computacional: Guía para profesores. *Computing At School*, 1-17. Obtenido de <https://www.codemas.org/wp-content/uploads/2016/04/Pensamiento-computacional-Gu%C3%ADa-para-profesores.pdf>
- Díaz Sagástegui, G. A., & Lozano Cubas, R. A. (2019). Uso de las aplicaciones Code.org y Scratch para el aprendizaje de programación en los estudiantes del 5° y 6° grado de EBR del C.E. N° 82099 de la

provincia de San Pablo, 2018. *Tesis presentada para optar el Título Profesional de Ingeniero Informático y de Sistemas*. Universidad Privada Antonio Guillermo Urrelo, Cajamarca, Perú.

Fernández Cárdenas, D. R. (2014). Levantamiento y propuesta de mejora de procesos y elaboración de manual de perfiles de cargos para la fundación Hermano Miguel. Quito, Ecuador. Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/7839/Tesis.pdf?sequence=1&isAllowed=y>

Fortoul Ollivier, M. B. (2008). La concepción de la enseñanza según los estudiantes del último año de la licenciatura en Educación Primaria en México. *Perfiles Educativos*, XXX(119), 72-89. Obtenido de <https://www.redalyc.org/pdf/132/13211156005.pdf>

García Monsálvez, J. C. (2017). Python como primer lenguaje de programación textual en la Enseñanza Secundaria. *Education in the Knowledge Society*, 18(2), 147-162. Obtenido de <https://www.redalyc.org/pdf/5355/535554766009.pdf>

Gerencia de Proyectos Informáticos. (s.f.). Definición del Problema. (U. N. Distancia, Ed.) Bogotá, Colombia. Obtenido de <https://repository.unad.edu.co/bitstream/handle/10596/5586/DefProblema.pdf;jsessionid=85420E855D648A71CEBE76D61D83B04D.jvm1?sequence=1>

González Ávila, M. (2002). Aspectos Éticos de la Investigación Cualitativa. *Revista Iberoamericana de Educación*, 85-103. Obtenido de <https://www.redalyc.org/pdf/800/80002905.pdf>

González, F. (2005). ¿Qué es un paradigma? Análisis teórico, conceptual y psicolingüístico del término. *Investigación y Postgrado*, 20(1), 13-54. Obtenido de <https://www.redalyc.org/pdf/658/65820102.pdf>

Gurises Unidos. (2017). *Pensamiento Computacional. Un aporte para la educación de hoy*. Montevideo: Fundación Telefónica. Obtenido de <https://www.gurisesunidos.org.uy/wp-content/uploads/2017/11/PensamientoComputacional.pdf>

Hernández Yáñez, L. (2014). Fundamentos de la programación. (F. d. Complutense, Ed.) Madrid, España. Obtenido de <https://www.fdi.ucm.es/profesor/luis/fp/fp.pdf>

López García, J. C. (2014). Actividades de aula con Scratch que favorecen el uso del pensamiento algorítmico. *Trabajo de grado*. Universidad ICESI, Cali, Colombia.

- López González, B., Hernández Cardona, B. S., Sánchez Sánchez, M., Cruz Sánchez, E., & Cano Rojas, R. (2009). PROGRAMACIÓN FÁCIL en el proceso enseñanza-aprendizaje. *Innovación Educativa*, 9(48), 61-71. Obtenido de <https://www.redalyc.org/pdf/1794/179414896006.pdf>
- Marengo Canepa, O., & De La Hoz Mendoza, H. (2017). Aplicación de estrategias metodológicas para el desarrollo del pensamiento lógico algorítmico en estudiantes de básica secundaria mediado con Scratch. *Proyecto de investigación presentado como requisito para optar el título de magíster*. Universidad de La Costa, Barranquilla, Colombia.
- Pardave Herrera, L. G., & Yalico Chombo, G. (2021). Influencia del lenguaje de programación Etoys en el área de educación para el trabajo para un aprendizaje cooperativo, en los alumnos del 4to "A" de la Institución Educativa Daniel Alcides Carrión – Pasco –2018. *Tesis para optar el Título Profesional de Licenciado en Educación*. Universidad Nacional Daniel Alcides Carrión, Cerro de Pasco, Perú.
- Pérez López, A. F. (2020). Actividades con Scratch para desarrollar el pensamiento algorítmico en estudiantes en una IEP de Chiclayo. *Tesis para optar al grado académico de Maestro*. Universidad Católica Santo Toribio de Mogrovejo, Chiclayo, Perú.
- Pérez Narváez, H. O. (2017). Uso de Scratch como herramienta para el desarrollo del pensamiento computacional en Programación I de la carrera de Informática de la Universidad Central de Ecuador. *Tesis presentada para aspirar al grado de Doctor*. Universidad de Alicante, Alicante, España.
- Raimundo, R. M. (2019). Conceptos: Pensamiento Computacional y Ciudadanía Digital, en sus acepciones relativas a la educación escolar. *Biblioteca del Congreso Nacional de Chile*, 1-11. Obtenido de <https://www.camara.cl/verDoc.aspx?prmID=167136&prmTIPO=DOCUMENTOCOMISION>
- Rivera Illatopa, R. R. (2020). Aplicación del lenguaje de programación Scratch para el desarrollo del pensamiento algorítmico en los alumnos del 6to grado del nivel primario en la I.E.P Augusto Cardich – Pillco Marca. *Tesis para optar al título de Ingeniero de Sistemas*. Universidad Nacional Hermilio Valdizan, Huánuco, Perú.
- Tejera Martínez, F., Aguilera, D., & Vílchez González, J. M. (2020). Lenguajes de programación y desarrollo de competencias clave. Revisión sistemática. *REDIE*, 22. doi:<https://doi.org/10.24320/redie.2020.22.e27.2869>.

Zapotecatl López, J. L. (2018). *Introducción al pensamiento computacional: conceptos básicos para todos*. México: ACADEMIA MEXICANA DE COMPUTACIÓN A.C. Obtenido de <http://amexcomp.mx/files/libro/LibroPC.pdf>

ANEXOS

ANEXO 01. Matriz de Consistencia

PROBLEMA	OBJETIVO	HIPOTESIS	VARIABLES	DIMENSIONES E INDICADORES	
<u>General :</u>	<u>General :</u>	<u>General :</u>	<u>Independiente:</u>	<u>Dimensión:</u>	<u>Indicador:</u>
¿En qué medida la enseñanza del programa de Python mejora el pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL?	Determinar en qué medida la enseñanza del programa Python mejora el pensamiento algorítmico en los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL	H0: La enseñanza del lenguaje del programa Python no mejora el pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL H1: La enseñanza del lenguaje del programa Python mejora el pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL	V.I. Lenguaje de Programación Python	Planificación de la enseñanza	Elaboración del syllabus del curso de Python.
				Desarrollo de la enseñanza	Desarrollo de los temas del curso de Python
					Evaluación del aprendizaje
<u>Específicos:</u>	<u>Específicos:</u>	<u>Específicos:</u>	<u>Dependiente:</u>	<u>Dimensión:</u>	<u>Indicador:</u>
¿En qué medida la enseñanza del programa Python mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL?	Determinar en qué medida la enseñanza del programa Python mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL	H0: La enseñanza del lenguaje del programa Python no mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL. H1: La enseñanza del lenguaje del programa Python mejora la fase de análisis del problema del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL	V.D. Pensamiento Algorítmico	Análisis del problema	Comprender y determinar el problema para precisar los datos necesarios, la incógnita, las restricciones del problema y las operaciones necesarias
					Diseño del algoritmo
¿En qué medida la enseñanza del programa Python mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la	Determinar en qué medida la enseñanza del programa Python mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la	H0: La enseñanza del lenguaje del programa Python no mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la			

Escuela Profesional de Ingeniería de Sistema de la UNHEVAL?	Escuela Profesional de Ingeniería de Sistema de la UNHEVAL	Escuela Profesional de Ingeniería de Sistema de la UNHEVAL H1: La enseñanza del lenguaje del programa Python mejora la fase de diseño y traducción del algoritmo del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL			Definir instrucciones claras, precisas y ordenadas
¿En qué medida la enseñanza del programa Python mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL?	Determinar en qué medida la enseñanza del programa Python mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL	H0: La enseñanza del lenguaje del programa Python no mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL H1: La enseñanza del lenguaje del programa Python mejora la fase de depuración del programa del pensamiento algorítmico de los alumnos del primer año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL		Depuración del programa	Verificar los pasos, replanteándose la estrategia de solución en caso sea necesario Verificar los resultados, replanteándose la estrategia de solución en caso sea necesario

Fuente: Elaboración propia.

ANEXO 02. Instrumento de pensamiento algorítmico

INSTRUMENTO PARA MEDIR EL NIVEL DE PENSAMIENTO ALGORITMICO

PRUEBA DE DIANOSTICO DE PENSAMIENTO ALORITMICO

El objeto de esta prueba es recolectar información con respecto al nivel de pensamiento algorítmico de los estudiantes de primero año de la Escuela Profesional de Ingeniería de Sistema de la UNHEVAL.

Apellidos y Nombres:

Código de alumno: _____

Indicación 1: Lee los siguientes enunciados para contestar las preguntas.

Problema N° 1: En el hospital Carlos Showing Ferrari de la ciudad de Huánuco existe actualmente tres áreas Ginecología (G), Emergencia (E) y Traumatología (T). El presupuesto inicial de apertura (PIA) del hospital es repartido de la siguiente manera: el 20% se emplea en el área de administración y el restante (RT) es distribuye asignado el 40% a Ginecología, 30% para Traumatología y el 30% a Emergencia. Calcular la cantidad de dinero que le corresponde a cada área del hospital si el monto del PIA es de S/. 500 000.00 soles.

Problema N° 2: Una fracción es reducida a su mínima expresión la cual es igual a $\frac{1}{8}$. Si la suma de sus términos es 72. Hallar la diferencia de los términos.

Problema N° 3: Dos postes de 5 y 12 metros de altura están separados por 24 metros de distancia. ¿Cuál es la diferencia entre sus extremos superiores?

Algoritmo A	Algoritmo B	Algoritmo C
INICIO Ir al cine Ver_listado (película) Seleccionar (película) Ver (horario de película) Seleccionar (horario de película) Ver (butacas) Seleccionar (butacas) Comprar ticket (película, horario de película y butacas) Mientras (hora < horario de película) Esperar horario de película Fin Mientras Si (hora >= horario de película) entonces Ingresar a la sala Ubicar butacas	Inicio del algoritmo Lee el valor a retirar Si el valor a retirar es mayor al valor total disponible (función calculaTotalDisponible) Mostrar error y terminar (Ir al fin del algoritmo) Mientras que el saldo a retirar sea mayor a cero Hacer Recorrer el mapa de billetes disponibles llamar a la función obtenerDenominación() para saber cuantos billetes/monedas devolver por cada denominación Fin recorrer mapa de billetes Imprimir el resultado Fin Algoritmo Función calcularTotalDisponible () Recorre el mapa de cantidad de billetes y monedas Usa el valor del billete o moneda y la multiplica por la cantidad disponible	Inicio del Algoritmo Todos contra todos Ingresar el listado de equipos participantes Ejecutar función sorteo() que recibe el listado de equipos participantes Fin del algoritmo Función sorteo() Recibe el listado de equipos Iniciar valores para el fixture, los rivales y los partidos Recorremos cada rival para armar el listado de rounds Dentro de cada round recorremos la cantidad de partidos que habrá Asignamos 2 elementos (equipos) a cada partido

Disfrutar película Salir de la sala Fin Si Salir del cine FIN	Acumula la anterior multiplicación en una variable llamada total Devuelve el valor acumulado Fin función Función obtener Denominación (denominacion, saldo A Retirar) Dividir el saldo A Retirar entre la denominación para obtener la cantidad de billetes Si la cantidad de billetes es mayor a la cantidad disponible entonces asignar la cantidad disponible Restar la cantidad disponible de la denominación, la cantidad obtenida Retornar la cantidad Fin función	Movemos 1 elemento (equipo) del array para la siguiente iteración Imprimimos el arreglo del fixtura Fin función
--	---	---

Indicación 2: Lee cada pregunta y marca la alternativa que consideres correcta.

DIMENSIÓN 1: ANÁLISIS DEL PROBLEMA.

1. ¿Qué datos adicionales se requieren para poder solucionar el problema N° 1? Marca la alternativa correcta.

- a) Los gastos que realiza administración
- b) El presupuesto de Traumatología
- c) Ningún dato, los datos necesarios se encuentran descritos en el problema.**
- d) El presupuesto de las áreas de Ginecología, Emergencia y Traumatología.

2. De acuerdo al enunciado del problema N° 3 ¿Cuáles son los datos necesarios y la incógnita?

- a) Poste A: 5 metros, poste B: 12 metros, distancia: x metros y Hallar la distancia entre los extremos superiores: x metros
- b) Poste A: 5 metros, poste B: 12 metros, hallar la distancia: x metros y distancia entre los extremos superiores: 12 metros
- c) Poste A: 5 metros, poste B: 12 metros, distancia: 24 metros y distancia entre los extremos superiores: 24 metros
- d) Poste A: 5 metros, poste B: 12 metros, distancia: 24 metros y Hallar la distancia entre los extremos superiores: x metros**

3. De acuerdo al enunciado del problema N° 2 ¿Cuáles son las ecuaciones que permitirán hallar la solución al problema planteado?

- a) $x=y/8$; $x + y = 72$**

b) $x=y/8 ; x - y = 72$

c) $y=x/8 ; x + y = 72$

d) $y=x/8 ; x - y = 72$

DIMENSIÓN 2: DISEÑO DEL ALGORITMO.

4. ¿Cuáles son los datos de entrada del algoritmo A y el algoritmo C?

a) Algoritmo A: Película, horario de película y butacas

Algoritmo C: Listado de equipos participantes

b) **Algoritmo A:** Película y butacas

Algoritmo C: Listado de equipos participantes

c) **Algoritmo A:** Sala, horario de película y butacas

Algoritmo C: Número de equipos

d) **Algoritmo A:** Película, horario de película y butacas

Algoritmo C: Número de combinaciones

5. De acuerdo al algoritmo C ¿Cuál es la instrucción que indica el procedimiento para encontrar la solución al problema?

Respuesta:

Función sorteo()

Recibe el listado de equipos

Iniciar valores para el fixture, los rivales y los partidos

Recorremos cada rival para armar el listado de rounds

Dentro de cada round recorremos la cantidad de partidos que habrá

Asignamos 2 elementos (equipos) a cada partido

Movemos 1 elemento (equipo) del array para la siguiente iteración

Imprimimos el arreglo del fixtura

Fin función

6. De acuerdo a los algoritmos A, B y C ¿Cuáles de ellos son finitos?

a) Algoritmo A

b) Algoritmo A y B

c) Todos son finitos

d) Ninguno es finito

7. De acuerdo al algoritmo B, si el valor a retirar es 170 y 1300 ¿Cuántos billetes/monedas debe devolver por cada denominación en cada uno de los casos?

Respuesta: 1 billetes de 100, 1 billete de 50, 1 billete de 20

3 billetes de 100, 6 billetes de 50, 10 billetes de 20

DIMENSIÓN 3: DEPURACIÓN DEL PROGRAMA.

8. De acuerdo al enunciado del problema N° 3 ¿Cuál es el teorema que se puede aplicar para solucionar el problema propuesto?

a) Binomio cuadrado

b) Tales

c) Euler

d) Pitágoras

9. De acuerdo al enunciado del problema N° 3 ¿Cuál es la cantidad de presupuesto asignado para el área de Emergencia?

a) S/. 100

b) S/. 160

c) S/. 140

d) S/. 120

10. De acuerdo al enunciado del problema N° 3 ¿Cuál es la distancia entre los extremos superiores de los postes?

a) 6 metros

b) 25 metros

c) 20 metros

d) 14 metros

Silabo del curso de pensamiento algorítmico.

PENSAMIENTO ALGORITMICO

SILABO

I. DATOS GENERALES:

- 1.1. Escuela profesional: Ingeniería de sistemas
- 1.2. Ciclo: 1er ciclo
- 1.3. Pre – requisito: Ninguno
- 1.4. Duración: 3 semanas

II. FUNDAMENTACIÓN DE LA ASIGNATURA:

La presente asignatura tiene como objetivo desarrollar las habilidades de los estudiantes en la resolución de problemas usando para ello el raciocinio lógico, argumentos teóricos – prácticos y la tecnología, como herramientas a su disposición. En el presente curso el alumno aprenderá a analizar problemas, diseñar los procesos para solucionarlos e implementar la solución mediante un programa de computadora.

La resolución de problemas es un proceso creativo donde el conocimiento, la habilidad y la experiencia juegan un papel importante. Es importante que el problema a solucionar este perfectamente definido, es decir, se tiene que definir que se va resolver antes de cómo resolverlo; esta etapa de definición conlleva la eliminación de ambigüedades e información que no sea relevante que aparezcan a lo largo del problema, para de esta manera separar que parte del enunciado al ser resuelto constituirá una solución válida.

Posterior, viene la etapa de establecer las herramientas con las que los alumnos podrán diseñar la solución; siendo esta un algoritmo. El alumno debe diseñar procesos organizados para la solución del problema.

Finalmente tenemos la implementación del algoritmo diseñado usando una tecnología que nos ofrece el mundo informático, para ello se empleara el lenguaje de programación Python, la finalidad del uso de la tecnología son las características que nos brinda en la ejecución del diseño como solución de un problema.

III. COMPETENCIAS DE LA ASIGNATURA

Competencias:

- Conoce, comprende y aplica los conceptos básicos de la algoritmia para plantear soluciones lógicas, valorando la importancia del análisis y razonamiento en el desarrollo de su formación.
- Analiza problemas y diseña algoritmos utilizando técnicas y herramientas, estableciendo diferentes estructuras lógicas en forma algorítmica para resolver problemas matemáticos y aplicaciones reales; mostrando creatividad y espíritu de trabajo en equipo.
- Plantea soluciones integradas usando funciones y procedimientos para problemas siendo crítico y reflexivo en su uso.

IV. PROGRAMACION ACADEMICA

PRIMER MODULO: FUNDAMENTO DE ALGORÍTMIA

Capacidades:

- Comprende los conceptos y principios básicos de la algoritmia para la solución de problemas.
- Analiza y diseña algoritmos usando diagramas de flujo y pseudocódigos diferenciando la funcionalidad de las estructuras básicas de control.
- Utiliza las estructuras básicas de control en la construcción de algoritmos.

N°	Fecha	Contenidos teóricos
1	04/03/2022	Examen inicial
2	07/03/2022	Conceptos fundamentales: Tipos de datos. Variables, constantes e identificadores. Expresiones y operadores aritméticos y lógicos.
3	08/03/2022	Algoritmos: Etapas para la solución de problemas por computadores Definición y características de un algoritmo. Lenguajes algorítmicos Reglas de los algorítmicos
4	09/03/2022	Representación de algoritmos: Diagramas de flujo (flujogramas) Definición

		Características Símbolos utilizados
5	10/03/2022	Representación de algoritmos: Pseudocódigos Definición Características Símbolos utilizados
6	11/03/2022	Estructuras lógicas de control: Estructuras secuenciales Definición y características Instrucciones de asignación, entrada y salida

SEGUNDO MODULO: FUNDAMENTO DE ALGORÍTMIA

Capacidades:

- Comprende la utilidad del uso de software como medio tecnológico en la implementación de soluciones algorítmicas.
- Analiza, diseña e implementa diseños algorítmicos usando lenguajes de programación.

N°	Fecha	Contenidos teóricos
1	14/03/2022	Estructuras lógicas de control: Estructuras condicionales o de selección Definición y características Simple y doble Anidada y selectiva
2	15/03/2022	Estructuras lógicas de control: Estructuras repetitivas o de ciclos Definición y características Mientras, hacer – mientras Contadores y acumuladores
3	16/03/2022	Estructuras lógicas de control: Estructuras repetitivas o ciclos para

TERCER MODULO: FUNDAMENTO DE ALGORÍTMIA

Capacidades:

- Comprende conceptos y principios avanzados en algoritmia.
- Comprende conceptos y principios básicos sobre el diseño y desarrollo de la base de datos como una forma de organización de datos importantes para la solución de problemas empresariales.
- Comprende la importancia del desarrollo de base de datos y su relación con algoritmos de desarrollo e implementación de soluciones empresariales.

- Diseña y desarrolla aplicaciones basadas en el diseño de algoritmos y base de datos para la solución de problemas empresariales básicos.

N°	Fecha	Contenidos teóricos
1	17/03/2022	Retroalimentación: Estructuras algorítmicas Estructuras lógicas de control
2	18/03/2022	Funciones: Conceptos básicos Características Argumentos y parámetros
3	21/03/2022	Introducción a los arreglos y matrices: Conceptos básicos Arreglos unidimensionales: vectores
4	22/03/2022	Introducción a la base de datos: Conceptos básicos Diseño e implementación de una base de datos
5	24/03/2022	Trabajo con comandos SQL: Conceptos básicos
6	25/03/2022	Examen final

V. MEDIOS Y MATERIALES:

Recursos tecnológicos de la Escuela de Ingeniería de Sistemas:

- Pizarra, marcadores, paleógrafos, masking tape.
- Equipo multimedia: Proyector y computador.
- Vía internet: páginas web y direcciones electrónicas, buscadores para ubicar temas de interés y correo electrónico para consultas.
- Textos especializados de lectura: libros y revistas.

VI. EVALUACIÓN:

Los criterios de evaluación del curso son los siguientes:

- Dominio de los temas evaluados
- Reflexión y creatividad en el desarrollo de soluciones a los problemas planteados.

ACTA DE DEFENSA DE TESIS



UNIVERSIDAD NACIONAL "HERMILIO VALDIZÁN" HUÁNUCO – PERÚ
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS



ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

En Huánuco, a los 16... días del mes de diciembre... de 2022, siendo las 8:30 hrs, de acuerdo al Reglamento de Grados y Títulos de la Universidad Nacional Hermilio Valdizán, TÍTULO VI – CAPITULO I Art. 76° al 79°, aprobado mediante Resolución Consejo Universitario N° 0734-2022-UNHEVAL; se procedió a la evaluación virtual de la sustentación de la tesis titulado: PYTHON Y LA MEJORA DEL PENSAMIENTO ALGORÍTMICO EN LOS ALUMNOS DE LA ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS, 2022, presentado el bachiller en Ingeniería de Sistemas: **VICTOR HUGO SANTIAGO ABAL**.

Este evento se realizó ante los miembros del Jurado Calificador por los siguientes catedráticos:

PRESIDENTE: Dr. ADAM FRANCISCO PAREDES.

SECRETARIA: Mg. VELSY RIVERA VIDAL

VOCAL: Dr. ROSARIO VARGAS RONCAL

UNIVERSIDAD NACIONAL "HERMILIO VALDIZÁN"
HUÁNUCO
ES COPIA FIEL DEL ORIGINAL
Huánuco, 18 ENE 2023

Juana J. Jara Ruiz
Lic. Adm. Juana J. Jara Ruiz
FEDATARIA
Reg. N° 05270

Finalizado el acto de sustentación, se procedió a la calificación conforme al Artículo 79° del Reglamento de Grados y Títulos, obteniéndose el siguiente resultado: **Nota: 18**... (**DIECIOCHO**.....) equivalente a la calificación de **MUY BUENO**. Quedando el Bachiller en Ingeniería de Sistemas: **VICTOR HUGO SANTIAGO ABAL: APROBADO**.....

Con lo que se dio por concluido el acto y en fe de la cual firman los miembros del jurado Calificador.

.....
PRESIDENTE

.....
SECRETARIO

.....
VOCAL

**UNIVERSIDAD NACIONAL "HERMILIO VALDIZÁN" DE HUÁNUCO
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS
CARRERA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



CONSTANCIA DE APTO

De acuerdo al Reglamento General de Grados y Títulos Modificado de la Universidad Nacional Hermilio Valdizán de Huánuco aprobado con Resolución del Consejo Universitario N° 1893-2021-UNHEVAL, de fecha 17 de agosto de 2021 y en atención a la Tercera Disposición Complementaria, donde estipula que los trabajos de investigación y tesis de pregrado deberán tener una similitud máxima del 30%.

Después de aplicado el Software Turnitin, se evidencia una similitud del 29% encontrándose bajo los parámetros reglamentados.

Tesis para optar el Título Profesional de Ingeniero de Sistemas:

**"PYTHON Y LA MEJORA DEL PENSAMIENTO
ALGORÍTMICO EN LOS ALUMNOS DE LA ESCUELA
PROFESIONAL DE INGENIERÍA DE SISTEMAS, 2022"**

Tesista

Bach. Ingeniería de Sistemas Victor Hugo Santiago Abal

Huánuco, 17 de enero de 2023

Una firma manuscrita en tinta azul que parece ser "Nérida del Carmen Pastrana Díaz".

Nérida del Carmen Pastrana Díaz
Directora de Investigación - FIIS



AUTORIZACIÓN DE PUBLICACIÓN DIGITAL Y DECLARACIÓN JURADA DEL TRABAJO DE INVESTIGACIÓN PARA OPTAR UN GRADO ACADÉMICO O TÍTULO PROFESIONAL

1. Autorización de Publicación: (Marque con una "X")

Pregrado	<input checked="" type="checkbox"/>	Segunda Especialidad		Posgrado:	Maestría		Doctorado
-----------------	-------------------------------------	-----------------------------	--	------------------	-----------------	--	------------------

Pregrado (tal y como está registrado en SUNEDU)

Facultad	INGENIERÍA INDUSTRIAL Y DE SISTEMAS
Escuela Profesional	INGENIERÍA DE SISTEMAS
Carrera Profesional	INGENIERÍA DE SISTEMAS
Grado que otorga	-----
Título que otorga	INGENIERO DE SISTEMAS

Segunda especialidad (tal y como está registrado en SUNEDU)

Facultad	-----
Nombre del programa	-----
Título que Otorga	-----

Posgrado (tal y como está registrado en SUNEDU)

Nombre del Programa de estudio	-----
Grado que otorga	-----

2. Datos del Autor(es): (Ingrese todos los datos requeridos completos)

Apellidos y Nombres:	SANTIAGO ABAL VICTOR HUGO							
Tipo de Documento:	DNI	<input checked="" type="checkbox"/>	Pasaporte	<input type="checkbox"/>	C.E.	<input type="checkbox"/>	Nro. de Celular:	944643943
Nro. de Documento:	42998096				Correo Electrónico:	VHSAEE@GMAIL.COM		

Apellidos y Nombres:								
Tipo de Documento:	DNI	<input type="checkbox"/>	Pasaporte	<input type="checkbox"/>	C.E.	<input type="checkbox"/>	Nro. de Celular:	
Nro. de Documento:					Correo Electrónico:			

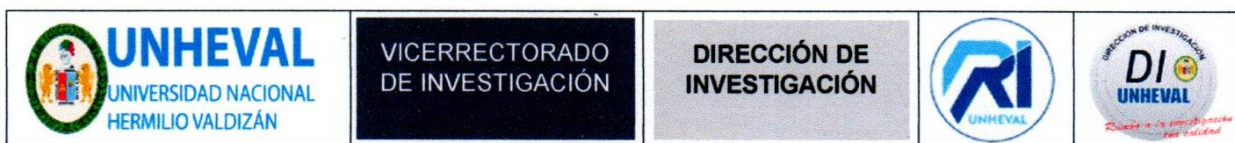
Apellidos y Nombres:								
Tipo de Documento:	DNI	<input type="checkbox"/>	Pasaporte	<input type="checkbox"/>	C.E.	<input type="checkbox"/>	Nro. de Celular:	
Nro. de Documento:					Correo Electrónico:			

3. Datos del Asesor: (Ingrese todos los datos requeridos completos según DNI, no es necesario indicar el Grado Académico del Asesor)

¿El Trabajo de Investigación cuenta con un Asesor?: (marque con una "X" en el recuadro del costado, según corresponda)	SI	<input checked="" type="checkbox"/>	NO	
Apellidos y Nombres:	HUAPAYA CONDORI FREDDY RONALD		ORCID ID:	0000-0003-4783-3803
Tipo de Documento:	DNI	<input checked="" type="checkbox"/>	Pasaporte	<input type="checkbox"/>
	C.E.	<input type="checkbox"/>	Nro. de documento:	22506586

4. Datos del Jurado calificador: (Ingrese solamente los Apellidos y Nombres completos según DNI, no es necesario indicar el Grado Académico del Jurado)

Presidente:	FRANCISCO PAREDES ABIMAEL ADAM
Secretario:	RIVERA VIDAL DE SÁNCHEZ HEIDY VELSÝ
Vocal:	VARGAS RONCAL ROSARIO
Vocal:	
Vocal:	
Accesitario	


5. Declaración Jurada: (Ingrese todos los datos requeridos completos)

a) Soy Autor (a) (es) del Trabajo de Investigación Titulado: <i>(Ingrese el título tal y como está registrado en el Acta de Sustentación)</i>
PYTHON Y LA MEJORA DEL PENSAMIENTO ALGORÍTMICO EN LOS ALUMNOS DE LA ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS, 2022
b) El Trabajo de Investigación fue sustentado para optar el Grado Académico ó Título Profesional de: <i>(tal y como está registrado en SUNEDU)</i>
TITULO PROFESIONAL DE INGENIERO DE SISTEMAS
c) El Trabajo de investigación no contiene plagio (ninguna frase completa o párrafo del documento corresponde a otro autor sin haber sido citado previamente), ni total ni parcial, para lo cual se han respetado las normas internacionales de citas y referencias.
d) El trabajo de investigación presentado no atenta contra derechos de terceros.
e) El trabajo de investigación no ha sido publicado, ni presentado anteriormente para obtener algún Grado Académico o Título profesional.
f) Los datos presentados en los resultados (tablas, gráficos, textos) no han sido falsificados, ni presentados sin citar la fuente.
g) Los archivos digitales que entrego contienen la versión final del documento sustentado y aprobado por el jurado.
h) Por lo expuesto, mediante la presente asumo frente a la Universidad Nacional Hermilio Valdizán (en adelante LA UNIVERSIDAD), cualquier responsabilidad que pudiera derivarse por la autoría, originalidad y veracidad del contenido del Trabajo de Investigación, así como por los derechos de la obra y/o invención presentada. En consecuencia, me hago responsable frente a LA UNIVERSIDAD y frente a terceros de cualquier daño que pudiera ocasionar a LA UNIVERSIDAD o a terceros, por el incumplimiento de lo declarado o que pudiera encontrar causas en la tesis presentada, asumiendo todas las cargas pecuniarias que pudieran derivarse de ello. Asimismo, por la presente me comprometo a asumir además todas las cargas pecuniarias que pudieran derivarse para LA UNIVERSIDAD en favor de terceros con motivo de acciones, reclamaciones o conflictos derivados del incumplimiento de lo declarado o las que encontraren causa en el contenido del trabajo de investigación. De identificarse fraude, piratería, plagio, falsificación o que el trabajo haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, sometiéndome a la normatividad vigente de la Universidad Nacional Hermilio Valdizán.

6. Datos del Documento Digital a Publicar: (Ingrese todos los datos requeridos completos)



Ingrese solo el año en el que sustentó su Trabajo de Investigación: <i>(Verifique la Información en el Acta de Sustentación)</i>			
Modalidad de obtención del Grado Académico o Título Profesional: <i>(Marque con X según Ley Universitaria con la que inició sus estudios)</i>	Tesis	X	Tesis Formato Artículo
	Trabajo de Investigación		Trabajo de Suficiencia Profesional
	Trabajo Académico		Otros <i>(especifique modalidad)</i>
Palabras Clave: <i>(solo se requieren 3 palabras)</i>	PYTHON	PENSAMIENTO	ALGORÍTMICO
Tipo de Acceso: <i>(Marque con X según corresponda)</i>	Acceso Abierto	X	Condición Cerrada (*)
	Con Periodo de Embargo (*)		Fecha de Fin de Embargo:
¿El Trabajo de Investigación, fue realizado en el marco de una Agencia Patrocinadora? <i>(ya sea por financiamientos de proyectos, esquema financiero, beca, subvención u otras; marcar con una "X" en el recuadro del costado según corresponda):</i>			SI NO X
Información de la Agencia Patrocinadora:			

El trabajo de investigación en digital y físico tienen los mismos registros del presente documento como son: Denominación del programa Académico, Denominación del Grado Académico o Título profesional, Nombres y Apellidos del autor, Asesor y Jurado calificador tal y como figura en el Documento de Identidad, Título completo del Trabajo de Investigación y Modalidad de Obtención del Grado Académico o Título Profesional según la Ley Universitaria con la que se inició los estudios.



7. Autorización de Publicación Digital:

A través de la presente, autorizo de manera gratuita a la Universidad Nacional Herólio Valdizán a publicar la versión electrónica de este Trabajo de Investigación en su Biblioteca Virtual, Portal Web, Repositorio Institucional y Base de Datos académica, por plazo indefinido, consintiendo que con dicha autorización cualquier tercero podrá acceder a dichas páginas de manera gratuita pudiendo revisarla, imprimirla o grabarla siempre y cuando se respete la autoría y sea citada correctamente. Se autoriza cambiar el contenido de forma, más no de fondo, para propósitos de estandarización de formatos, como también establecer los metadatos correspondientes.

Firma: 		
Apellidos y Nombres:	SANTIAGO ABAL VICTOR HUGO	Huella Digital
DNI:	42998096	
Firma:		
Apellidos y Nombres:		Huella Digital
DNI:		
Firma:		
Apellidos y Nombres:		Huella Digital
DNI:		
Fecha:		

Nota:

- ✓ No modificar los textos preestablecidos, conservar la estructura del documento.
- ✓ Marque con una X en el recuadro que corresponde.
- ✓ Llenar este formato de forma digital, con tipo de letra calibri, tamaño de fuente 09, manteniendo la alineación del texto que observe en el modelo, sin errores gramaticales (recuerde las mayúsculas también se tildan si corresponde).
- ✓ La información que escribe en este formato debe coincidir con la información registrada en los demás archivos y/o formatos que presente, tales como: DNI, Acta de Sustentación, Trabajo de Investigación (PDF) y Declaración Jurada.
- ✓ Cada uno de los datos requeridos en este formato, es de carácter obligatorio según corresponda.