

UNIVERSIDAD NACIONAL DE CAJAMARCA

ESCUELA DE POSGRADO



UNIDAD DE POSGRADO DE LA FACULTAD DE EDUCACIÓN

PROGRAMA DE DOCTORADO EN CIENCIAS

TESIS:

**ESTRATEGIA DE ENSEÑANZA INPAL, EN EL DISEÑO DE
ALGORITMOS DE LA ASIGNATURA “FUNDAMENTOS DE
PROGRAMACIÓN” PARA ESTUDIANTES DE LA ESCUELA
ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS -
UNIVERSIDAD NACIONAL DE CAJAMARCA, 2020**

Para optar el Grado Académico de

DOCTOR EN CIENCIAS

MENCIÓN: EDUCACIÓN

Presentado por:

LISI JANET VÁSQUEZ FERNÁNDEZ

Asesor:

Dr. LINO JORGE LLATAS ALTAMIRANO

Cajamarca, Perú

2023

CONSTANCIA DE INFORME DE ORIGINALIDAD

1. Investigador:
Lisi Janet Vásquez Fernández
DNI: 40408055
Unidad de Posgrado de la Facultad de Educación, Mención: Educación.
2. Asesor:
Dr. Lino Jorge Llatas Altamirano
3. Grado académico o título profesional
 Bachiller Título profesional Segunda especialidad
 Maestro Doctor
4. Tipo de Investigación:
 Tesis Trabajo de investigación Trabajo de suficiencia profesional
 Trabajo académico
5. Título de Trabajo de Investigación:
Estrategia de enseñanza INPAL, en el diseño de algoritmos de la asignatura "Fundamentos de Programación" para estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas - Universidad Nacional de Cajamarca, 2020.
6. Fecha de evaluación: 23/11/2024
7. Software antiplagio: TURNITIN URKUND (OURIGINAL) (*)
8. Porcentaje de Informe de Similitud: 22%
9. Código Documento: 3117:346784463
10. Resultado de la Evaluación de Similitud:
 APROBADO PARA LEVANTAMIENTO DE OBSERVACIONES O DESAPROBADO

Fecha Emisión: 30/04/2024

<i>Firma y/o Sello Emisor Constancia</i>
 <hr/> Dr. Segundo Ricardo Cabanillas Aguilar DNI: 26607960

* En caso se realizó la evaluación hasta setiembre de 2023

COPYRIGHT © 2023 by
LISI JANET VÁSQUEZ FERNÁNDEZ
Todos los derechos reservados



Universidad Nacional de Cajamarca
LICENCIADA CON RESOLUCIÓN DE CONSEJO DIRECTIVO N° 080-2018-SUNEDU/CD
Escuela de Posgrado
CAJAMARCA - PERU



PROGRAMA DE DOCTORADO EN CIENCIAS

ACTA DE SUSTENTACIÓN DE TESIS

MENCIÓN: EDUCACIÓN

Siendo las *5:00 pm* horas, del día 23 de noviembre del año dos mil veintitrés, reunidos en el Centro de Idiomas de la Universidad Nacional de Cajamarca, el Jurado Evaluador presidido por el Dr. SEGUNDO RICARDO CABANILLAS AGUILAR, Dr. CÉSAR ENRIQUE ALVAREZ IPARRAGUIRRE, Dra. LETICIA NOEMÍ ZAVALETA GONZÁLES y en calidad de Asesor, el Dr. LINO JORGE LLATAS ALTAMIRANO Actuando de conformidad con el Reglamento Interno de la Escuela de Posgrado y el Reglamento del Programa de Doctorado de la Escuela de Posgrado de la Universidad Nacional de Cajamarca, se inició la SUSTENTACIÓN de la tesis titulada: **ESTRATEGIA DE ENSEÑANZA INPAL, EN EL DISEÑO DE ALGORITMOS DE LA ASIGNATURA "FUNDAMENTOS DE PROGRAMACIÓN" PARA ESTUDIANTES DE LA ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS – UNIVERSIDAD NACIONAL DE CAJAMARCA, 2020;** presentada por la Maestra en Ciencias Administración y Gerencia Pública **LISI JANET VÁSQUEZ FERNÁNDEZ**

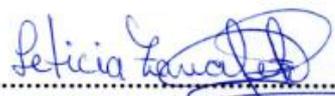
Realizada la exposición de la Tesis y absueltas las preguntas formuladas por el Jurado Evaluador, y luego de la deliberación, se acordó ... *A.P.P.A.P.* con la calificación de *16 (dieciséis) - BUENO* la mencionada Tesis; en tal virtud, la Maestra en Ciencias Administración y Gerencia Pública **LISI JANET VÁSQUEZ FERNÁNDEZ**, está apta para recibir en ceremonia especial el Diploma que la acredita como **DOCTOR EN CIENCIAS**, de la Unidad de Posgrado de la Facultad de Educación, Mención **EDUCACIÓN**

Siendo las *6:30 pm* horas del mismo día, se dio por concluido el acto.


.....
Dr. Lino Jorge Llatas Altamirano
Asesor


.....
Dr. Segundo Ricardo Cabanillas Aguilar
Presidente-Jurado Evaluador


.....
Dr. César Enrique Álvarez Iparraguirre
Jurado Evaluador


.....
Dra. Leticia Noemí Zavaleta González
Jurado Evaluador

DEDICATORIA

A:

A mi esposo por su apoyo moral e intelectual, a mi pequeña hija Leticia Stephany y a mi pequeño hijo Adrian Emanuel; a mi madre por su constante amor y consejos y a mis hermanos por su apoyo incondicional.

AGRADECIMIENTO

A mi asesor de tesis Dr. Lino Llatas Altamirano por su gentileza, conocimientos, orientación y apoyo brindado durante el desarrollo de la presente tesis.

Al director de la Escuela Académico Profesional de Ingeniería de Sistemas, Dr. Edwin Valencia por las facilidades brindadas para poder llevar a cabo la presente investigación. A la Ing. Sandra Rodríguez y al docente Ing. Manuel Malpica, por permitirme aplicar la estrategia de enseñanza INPAL, en los grupos de práctica de la asignatura de Fundamentos de Programación, ciclo 2020-I. Y a los docentes del Departamento de Sistemas, Estadística e Informática, que me apoyaron con la validez de los instrumentos de recolección de datos.

A los docentes del doctorado por sus enseñanzas brindadas con sabiduría, didáctica y entusiasmo.

A todos mis compañeros del doctorado, por los buenos e inolvidables momentos compartidos en cada sesión de clase.

EPÍGRAFE

“La educación no cambia al mundo,
cambia a las personas que van a cambiar el mundo”

Paulo Freire

“Un programador que escriba un código limpio,
entiende perfectamente el problema antes de escribir el código”

Enupal

ÍNDICE GENERAL

	Pág.
DEDICATORIA	iv
AGRADECIMIENTO.....	vi
EPÍGRAFE	vii
ÍNDICE GENERAL	viii
LISTA DE TABLAS	xiii
LISTA DE FIGURAS	xiv
LISTA DE SIGLAS	xv
Resumen	xvi
Abstract	xvii
Introducción.....	xviii
CAPÍTULO I. EL PROBLEMA DE INVESTIGACIÓN.....	1
1.1 Planteamiento del problema	1
1.2 Formulación del problema.....	5
1.2.1 Problema principal.....	5
1.2.2 Problemas derivados	5
1.3 Justificación de la Investigación	6
1.3.1 Justificación teórica	6
1.3.2 Justificación práctica.....	6
1.3.3 Justificación metodológica	7

1.4	Delimitación de la investigación.....	7
1.4.1	Epistemológica	7
1.4.2	Espacial	8
1.4.3	Temporal	8
1.4.4	Línea de Investigación	8
1.5	Objetivos de la investigación.....	9
1.5.1	Objetivo general.....	9
1.5.2	Objetivos específicos	9
CAPÍTULO II. MARCO TEÓRICO		10
2.1	Antecedentes de la investigación.....	10
2.2	Marco epistemológico	20
2.3	Marco teórico - científico de la investigación	23
2.3.1	La asignatura de Fundamentos de Programación en el currículo de estudios de Ingeniería de Sistemas - 2018.....	23
2.3.2	Teoría del Aprendizaje Significativo en la Estrategia de Enseñanza “Inducción para Algoritmos” - INPAL, en la Educación Superior Universitaria	24
2.3.3	Teoría del Aprendizaje de Gagné como apoyo a las Sesiones de Aprendizaje de Fundamentos de Programación en la Educación Superior Universitaria	27
2.3.4	Teoría de las Situaciones Didácticas en las Sesiones de Aprendizaje de Fundamentos de Programación en la Educación Superior Universitaria	31
2.3.5	El conectivismo en la educación remota universitaria en época de pandemia COVID 19.	33
2.3.6	Algoritmos en la solución de problemas computacionales	36

2.3.7	Diseño de algoritmos en la asignatura de “Fundamentos de Programación”	38
2.3.8	Escritura de algoritmos en las estructuras de control secuencial y condicional en la asignatura “Fundamentos de Programación”	40
2.3.9	Fases en la resolución de problemas haciendo uso de una computadora.....	41
2.3.10	Plantear y resolver problemas matemáticos como apoyo al diseño de algoritmos computacionales.....	43
2.3.11	Estrategias de Enseñanza en el diseño de algoritmos computacionales	45
2.3.12	Estilos de aprendizaje en estudiantes de la carrera de ingeniería de sistemas en la educación superior universitaria	47
2.3.13	Fundamentos teóricos - metodológicos de la estrategia de enseñanza INPAL para el diseño de algoritmos.....	50
2.4	Definición de términos básicos.....	70
CAPÍTULO III. MARCO METODOLÓGICO.....		72
3.1	Caracterización y contextualización de la investigación.....	72
3.1.1	Descripción del perfil de la institución educativa o red educativa.	72
3.1.2	Breve reseña histórica de la institución educativa o red educativa	73
3.1.3	Características demográficas y socioeconómicas	74
3.1.4	Características culturales y ambientales.....	75
3.2	Hipótesis de investigación.....	75
3.3	Variables de la investigación.....	76
3.4	Matriz de Operacionalización de variables	77
3.5	Población y muestra	78
3.6	Unidad de análisis	79

3.7	Métodos de investigación.....	79
3.8	Tipo de investigación	82
3.9	Diseño de investigación	83
3.10	Técnicas e instrumentos de recolección de información.....	84
3.11	Técnicas para el procesamiento y análisis de la información.....	87
3.12	Validez y confiabilidad	87
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN		89
4.1	Matriz general de resultados.....	89
4.2	Resultados por dimensiones de las variables de estudio	89
4.2.1	Dimensión Análisis del Problema, en el pretest	90
4.2.2	Dimensión Análisis del Problema, en el postest.....	92
4.2.3	Dimensión Diseño de la Solución Algorítmica, en el pretest.....	94
4.2.4	Dimensión Diseño de la Solución Algorítmica, en el postest	96
4.3	Resultados totales de las variables de estudio	99
4.3.1	Pretest.....	99
4.3.2	Aplicación de la estrategia de enseñanza INPAL.....	101
4.3.3	Postest	103
4.4	Prueba de hipótesis	106
CONCLUSIONES		118
SUGERENCIAS.....		119
REFERENCIAS.....		120

APÉNDICES / ANEXOS.....	125
Apéndice 1. Instrumentos de recolección de datos	126
Evaluación escrita en el pretest	126
Evaluación escrita en el postest.....	127
Rúbrica de Evaluación del Diseño de Algoritmos	128
Rúbrica de Evaluación de la Estrategia de Enseñanza INPAL para el Diseño de Algoritmos	129
Encuesta sobre la Aplicación de la Estrategia INPAL en el Proceso de Aprendizaje del Diseño de Algoritmos	130
Apéndice 2. Confiabilidad de los instrumentos	132
Cálculo de la Confiabilidad con Alfa de Cronbach.....	132
Apéndice 3. Validación de instrumentos de recolección de datos a través de juicio de expertos.....	134
Apéndice 4. Matriz general de resultados en Excel	138
Apéndice 5. Programa sesiones de aprendizaje con la estrategia de enseñanza INPAL, para el diseño de algoritmos computacionales.	141
Apéndice 6. Material didáctico para el diseño de algoritmos	151
Apéndice 7. Resultados de la percepción de los estudiantes con respecto a la estrategia de enseñanza INPAL	157
Apéndice 8. Matriz de Consistencia	159
Anexo 1. Plan de Estudios del Programa de Ingeniería de Sistemas	160
Anexo 2. Sílabo de la asignatura de Fundamentos de Programación.....	176

LISTA DE TABLAS

Tabla 1 Estrategia de enseñanza INPAL	62
Tabla 2 Marco Lógico de los resultados de aprendizaje	63
Tabla 3 Actividades por sesiones de aprendizaje	65
Tabla 4 Muestra de la investigación.....	78
Tabla 5 Nivel de diseño de algoritmos y su escala	85
Tabla 6 Ficha Técnica de Rúbrica de Evaluación del Diseño de Algoritmos	86
Tabla 7 Resumen de validación de contenido por juicio de expertos	88
Tabla 8 Nivel en la dimensión análisis del problema, en el pretest	90
Tabla 9 Nivel en la dimensión análisis del problema, en el postest.....	92
Tabla 10 Media de los puntajes de la dimensión análisis del problema, en el postest.....	93
Tabla 11 Nivel en la dimensión diseño de la solución algorítmica, en el pretest	94
Tabla 12 Nivel en la dimensión del diseño de la solución algorítmica, en el postest.....	96
Tabla 13 Media de los puntajes de la dimensión solución algorítmica, en el postest.....	97
Tabla 14 Nivel en el diseño de algoritmos, en el pretest.....	99
Tabla 15 Nivel en el diseño de algoritmos, en el postest	104
Tabla 16 Nivel en el diseño de algoritmos, en el pretest.....	107
Tabla 17 Nivel en el diseño de algoritmos, en el postest	109
Tabla 18 Prueba de Kolgomorov-Smirov para una muestra	111
Tabla 19 Prueba de Levene de igualdad de varianzas.....	113
Tabla 20 Prueba T de Student para la igualdad de medias.....	114
Tabla 21 Media de los puntajes de la variable diseño de algoritmos.....	114

LISTA DE FIGURAS

Figura 1 Proceso Cuantitativo	22
Figura 2 Un modelo básico de aprendizaje y memoria, subyacente a las teorías cognitivas modernas (procesamiento de información).....	29
Figura 3 Resolución de un problema	36
Figura 4 Refinamiento del algoritmo	39
Figura 5 Fases del diseño de un algoritmo	39
Figura 6 Fases de resolución de un problema con computadora.....	42
Figura 7 Método de los cuatro pasos de Pólya	44
Figura 8 Modelo de Kolb (1984)	48
Figura 9 Anclaje del nuevo conocimiento algorítmico con el conocimiento previo matemático.....	52
Figura 10 Condiciones y principios en las sesiones de aprendizaje	54
Figura 11 Situación didáctica en el aprendizaje del diseño de algoritmos.....	56
Figura 12 Inducción a una solución algorítmica, partir de una solución matemática.....	57
Figura 13 Esquema general de la fundamentación teórica de la estrategia de enseñanza INPAL	58
Figura 14 Nivel en la dimensión análisis del problema, en el pretest.....	90
Figura 15 Nivel en la dimensión análisis del problema, en el postest	92
Figura 16 Nivel en la dimensión diseño de la solución algorítmica, en el pretest	95
Figura 17 Nivel en la dimensión del diseño de la solución algorítmica, en el postest	97
Figura 18 Nivel en el diseño de algoritmos, en el pretest	100
Figura 19 Nivel en el diseño de algoritmos, en el postest.....	104

LISTA DE SIGLAS

INPAL : Inducción para Algoritmos.

UNC : Universidad Nacional de Cajamarca.

EAPIS : Escuela Académico Profesional de Ingeniería de Sistemas.

Resumen

La presente investigación tuvo como objetivo, determinar la influencia de la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” - INPAL en el diseño de algoritmos de la asignatura “Fundamentos de Programación” para estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020. Se utilizó una muestra no probabilística, que estuvo conformada por 58 estudiantes del primer ciclo de la carrera profesional de Ingeniería de Sistemas. El tipo de investigación fue explicativa, con un diseño cuasi experimental. Se utilizó una rúbrica validada y confiable para recoger la información sobre el nivel del diseño de algoritmos que tenían los estudiantes antes y después de aplicar la estrategia de enseñanza INPAL. Los resultados de la investigación demuestran que la aplicación de la estrategia de enseñanza INPAL tuvo una influencia significativa, en el nivel del diseño de algoritmos de los estudiantes de la asignatura de “Fundamentos de Programación”; pues en el postest, en la prueba paramétrica T-Student, se obtuvo un $p=0.000$ menor a 0.05, lo que indica que la media de la calificación del grupo experimental fue mayor con respecto a la media de la calificación del grupo control, en un 19.63%. Además, también se determinó una influencia significativa en la dimensión análisis del problema, con una diferencia superior del 15.2% en el puntaje del grupo experimental, con respecto al grupo control, y de igual forma se determinó una influencia significativa en la dimensión diseño de la solución algorítmica, con una diferencia superior del 23.9% en el puntaje del grupo experimental, con respecto al grupo control.

Palabras Clave: Estrategia de enseñanza, Diseño de algoritmos, Programación.

Abstract

The objective of this research was to determine the influence of the application of the teaching strategy "Induction for Algorithms" - INPAL in the design of algorithms of the subject "Fundamentals of Programming" for students of the Academic Professional School of Systems Engineering of the National University of Cajamarca, year 2020. A non-probabilistic sample was used, which consisted of 58 students of the first cycle of the professional career of Systems Engineering. The type of research was explanatory, with a quasi-experimental design. A validated and reliable rubric was used to collect information on the level of algorithm design that the students had before and after applying the INPAL teaching strategy. The results of the research show that the application of the INPAL teaching strategy had a significant influence on the level of algorithm design of the students of the subject "Fundamentals of Programming"; since in the post-test, in the parametric T-Student test, a $p=0.000$ lower than 0.05 was obtained, which indicates that the mean score of the experimental group was higher than the mean score of the control group, by 19.63%. In addition, a significant influence was also determined in the problem analysis dimension, with a 15.2% higher difference in the score of the experimental group, with respect to the control group, and likewise a significant influence was determined in the design of the algorithmic solution dimension, with a 23.9% higher difference in the score of the experimental group, with respect to the control group.

Key words: Teaching strategy, Algorithm design, Programming.

Introducción

Los estudiantes que inician en el aprendizaje de algoritmos y programación, pasan por una serie de dificultades, lo que genera un bajo rendimiento académico, y con ello el incremento de estudiantes desaprobados. Según Velasco Ramírez (2020), generalmente los nuevos estudiantes tienen dificultades para entender y abstraer la comprensión del problema a resolver; es decir, dificultad en el desarrollo del pensamiento algorítmico, lo que aumenta el nivel de fracaso de los estudiantes y también la tasa de deserción de dichos cursos.

Enseñar algoritmos y conceptos de programación para estudiantes de primer año ha sido un gran desafío para las universidades, y aún lo es. Por lo tanto, la didáctica educativa juega un papel fundamental en la enseñanza de este tipo de asignaturas. Según Russo (2018), es importante situar en un rol más activo al aprendiz y en uno más reactivo al docente, lo que se busca es que el estudiante analice el problema y plantee una o más posibles soluciones con el conjunto de conocimientos que ya trae consigo.

Frente a esta problemática, la presente investigación busca obtener un nuevo conocimiento sobre el diseño e implementación de una nueva estrategia de enseñanza para el diseño de algoritmos computacionales, basada en la teoría del aprendizaje significativo (Ausubel, 2002), en la base teórica de estrategias de enseñanza (Díaz y Hernández, 2005) y en el análisis inductivo. Y con ello, ayudar a superar las dificultades en el diseño de dichos algoritmos y así contribuir a la mejora del rendimiento académico de los estudiantes de fundamentos de programación.

El objetivo general que orientó al estudio, es determinar la influencia de la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” - INPAL en el diseño de algoritmos de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020.

La hipótesis planteada fue la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL), influye significativamente en el diseño de algoritmos de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020.

En la contrastación de la hipótesis, se empleó una muestra no probabilística por conveniencia, de 58 estudiantes que cursaban la asignatura de fundamentos de programación, pertenecientes al primer ciclo de la carrera profesional de Ingeniería de Sistemas. Se aplicó un diseño cuasi experimental con grupos intactos (experimental y control) se les tomó una evaluación a través de un examen escrito, antes y después de aplicada la estrategia de enseñanza INPAL (pretest y postest), pero sólo en el grupo experimental se implementó la mencionada estrategia; dichos exámenes fueron calificados a través de una rúbrica, finalmente se aplicó una estadística descriptiva para responder a los objetivos específicos de la investigación y una estadística inferencial para la contrastación de la hipótesis: prueba paramétrica T de Student, la cual permitió determinar la influencia de la aplicación de la estrategia de enseñanza INPAL en el aprendizaje del diseño de algoritmos.

La presente tesis está estructurada en cuatro capítulos:

En el primer capítulo, se realizó una contextualización del problema, a nivel internacional, nacional y local, con el fin de poder formular el problema a través de una pregunta general de investigación, y tres preguntas derivadas; luego se presentó la justificación teórica, práctica y metodológica de la investigación; la delimitación epistemológica, espacial y temporal y finalmente el objetivo general y objetivos específicos del estudio.

En el segundo capítulo, se presentan los antecedentes de la investigación, para lo cual se recopilaron y revisaron los estudios más actuales sobre las variables de estudio: estratégica de

enseñanza INPAL y diseño de algoritmos; luego se presenta el marco epistemológico, el marco teórico y finalmente la definición de términos básicos.

En el tercer capítulo, se presenta el marco metodológico, con la caracterización y contextualización del lugar donde se realizó la investigación, la Universidad Nacional de Cajamarca - Escuela Académico Profesional de Ingeniería de Sistemas; luego, se presenta la hipótesis general y las hipótesis específicas de la investigación, la operacionalización de las variables, población, muestra, unidad de análisis, métodos de investigación, tipo y diseño de investigación, técnicas e instrumentos de recopilación de información, técnicas para el procesamiento y análisis de la información, validez y confiabilidad de las rúbricas empleadas.

El cuarto capítulo, está dedicado al análisis y discusión de los resultados obtenidos, desde la presentación de la matriz general de resultados, análisis y discusión de los resultados por dimensión, análisis y discusión de los resultados totales de las variables de estudio por objetivo y seguidamente se presenta la contrastación de las hipótesis, con su respectivo análisis y discusión.

Finalmente, se presentan las conclusiones y recomendaciones a las que se llegó después de terminar con el proceso de la presente investigación, seguidamente se presentan las referencias bibliográficas y los apéndices elaborados en la investigación como complemento del presente estudio.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1 Planteamiento del problema

Hoy en día, en este mundo digitalizado, aprender a programar es una necesidad; es decir, un profesional debe ser capaz de representar y codificar sus ideas para crear soluciones de software, pero la limitación para resolver problemas mediante algoritmos y códigos radica en el escaso desarrollo del pensamiento computacional y el alto nivel de abstracción para la programación.

Enseñar algoritmos y conceptos de programación para estudiantes de primer año ha sido un gran desafío para las universidades. Generalmente los nuevos estudiantes tienen dificultades para entender y abstraer la comprensión del problema a resolver; es decir, en el desarrollo del pensamiento algorítmico, lo que aumenta el nivel de fracaso de los estudiantes y también la tasa de deserción de dichos cursos (Velasco Ramírez, 2020)

La algoritmia es uno de los pilares de la programación y su importancia se evidencia en el desarrollo de cualquier aplicación informática, más allá sólo la construcción de programas. Al diseñar la solución de un problema a través de la computadora, se está diseñando algoritmos. “Un algoritmo es una serie de pasos ordenados lógicamente, los cuales permiten resolver un problema; en otras palabras, un algoritmo es 1ª receta para elaborar u obtener algo con precisión”. (Vasconcelos Santillán, 2018, p. 178)

A partir de la revisión de 26 artículos, obtenidos de las bases de datos como Education Resources Information Center (ERIC) y la del Institute of Electrical and Electronics Engineers (IEEE), se pudo concluir que la resolución de problemas computacionales representa una dificultad para los estudiantes en una etapa introductoria en la programación. Es de resaltar la necesidad de la capacidad de abstracción y análisis para identificar y comprender el problema. Y también se observa que existe el interés en la definición y manipulación de estrategias que fomenten el pensamiento algorítmico como habilidad en la resolución de problemas. Por último, se distinguen los objetos de aprendizaje como una alternativa para mejorar el proceso de enseñanza-aprendizaje en cursos de programación y en el desarrollo del pensamiento algorítmico. (Velasco Ramírez, 2020)

En Ecuador, en la Universidad de Guayaquil, se llevó a cabo un estudio, con 53 estudiantes de la materia correspondiente al aprendizaje de programación de aplicaciones, obteniéndose como resultado que el 68% de los estudiantes desconocen los conceptos de programación, tienen un conocimiento general acerca de variables, estructuras de control y sintaxis de los lenguajes de programación. Con respecto a la resolución del problema algebraico, el 8,7% de estudiantes lograron resolverlo; pero los demás estudiantes expresaron un análisis carente de lógica en cuanto a la identificación de datos de entrada y de la relación de éstos en la fórmula empleada. Por lo que, se observa que la mayoría de estudiantes tienen altas deficiencias en la programación de aplicaciones informáticas. (Mendoza Morán et al., 2017)

En Trujillo - Perú, se llevó a cabo una investigación con 27 estudiantes del curso de Fundamentos de Programación carrera de Computación e Informática de un Instituto de Educación Superior, donde el 60% de estudiantes obtuvieron una calificación de 0 a

10, lo que indica un nivel de aprendizaje en etapa de inicio; pues el curso en sí, implica razonamiento y pensamiento lógico; lo que hace confundir muchas veces al estudiante y termina desertando de la carrera profesional.

Así mismo, en Chachapoyas, se observa deficiencias en la capacidad del diseño y construcción de algoritmos en lógica de programación, en los estudiantes del III ciclo de computación e informática del I.E.S.T. Público “Perú - Japón”, lo que se evidencia en la incapacidad para diseñar algoritmos sencillos utilizando las diferentes estructuras, presentando dificultades para generar los diagramas de flujo, así como para realizar las pruebas de escritorio. (Chuquizuta Herrera, 2016)

En la Universidad Nacional de Cajamarca, en el programa de Estudios de Ingeniería de Sistemas, se cuenta con el eje curricular de “Algoritmos y Programación”, en cuyas asignaturas, los estudiantes tienen mayor dificultad de aprendizaje con respecto a las asignaturas del resto de ejes curriculares, pues en este eje, es donde se tiene un mayor número de desaprobados, y ello posiblemente se debe a la poca comprensión del problema, a un diseño incorrecto de los algoritmos, a la dificultad para aprender un lenguaje de programación y sobre todo a los errores que se presentan cuando se ejecutan los programas, lo que genera frustración y desmotivación en los estudiantes.

En tal sentido, una de las asignaturas clave e iniciales que introducen a los estudiantes en el mundo de la programación es “Fundamentos de la Programación” de primer ciclo, en la cual se ha podido observar un rendimiento académico bajo, pues el porcentaje de desaprobados fue del 30.28%, y el porcentaje de estudiantes que abandonaron la asignatura fue del 9.15%, y la calificación promedio fue de 9.11, según los datos analizados en base a las evaluaciones prácticas de los últimos 3 años: 2017, 2018 y 2019.

Los estudiantes que inician en el aprendizaje de los primeros temas en el campo de la programación sufren de una variedad de dificultades y deficiencias, siendo posiblemente una de las mayores dificultades la poca comprensión de los estudiantes en el enunciado del problema, lo que luego se refleja en un diseño incorrecto del algoritmo y finalmente en un código con errores. Posteriormente, este problema podría generar que los estudiantes vuelvan a repetir la asignatura y hasta en algunos casos abandonen la carrera.

Los programadores novatos sufren de un amplio rango de dificultades y deficiencias. Los estudiantes deben descubrir la necesidad de seguir un algoritmo para poder solucionar problemas. Los estudiantes no realizan el diagrama de flujo porque no reconocen la importancia del diseño o el plan de acción. Los estudiantes directamente intentaban codificar sin haber entendido el problema o pensando en la solución. Es importante que el facilitador de la materia motive a los estudiantes a realizar los diseños previos al intento de codificación. (Fuentes-Rosado y Moo Medina, 2017)

Frente a esta problemática es importante plantear estrategias de enseñanza que estimulen el aprendizaje de los fundamentos de la programación, siendo uno de los temas fundamentales, es el diseño de algoritmos computacionales.

Una didáctica propuesta implica situar en un rol más activo al aprendiz y en uno más reactivo al docente, dado que este último presenta diferentes situaciones a resolver por el alumno sin introducirlo en ningún concepto teórico. Lo que se busca es que el estudiante analice el problema y plantee una o más posibles soluciones con el conjunto de conocimientos que trae consigo. (Russo et al., 2018)

1.2 Formulación del problema

1.2.1 Problema principal

¿Cuál es la influencia de la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL), en el diseño de algoritmos de la asignatura “Fundamentos de Programación”, en los estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020?

1.2.2 Problemas derivados

- ¿Cuál es el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, antes de la aplicación de la estrategia de enseñanza INPAL?
- ¿Cómo aplicar la estrategia de enseñanza INPAL basada en los resultados del pretest, para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas?
- ¿Cuál es el nivel de diseño de algoritmos de los estudiantes de la asignatura de “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, después de la aplicación de la estrategia de enseñanza INPAL?

1.3 Justificación de la Investigación

1.3.1 Justificación teórica

La presente investigación contribuyó a obtener un nuevo conocimiento sobre el diseño de una nueva estrategia con un análisis inductivo desde la matemática para el diseño de algoritmos, y con base en algunas teorías educativas, llenando así un vacío teórico para la comprensión del problema. Y los resultados de la investigación permitió conocer su eficacia, de tal manera que se convierta en una estrategia pertinente, a fin de mejorar el diseño de algoritmos de nivel básico para programación.

1.3.2 Justificación práctica

Uno de los ejes curriculares de la carrera de Ingeniería de Sistemas de la UNC, es Algoritmos y Programación, por lo que la asignatura de Fundamentos de Programación es relevante para el desarrollo de las competencias de esta línea de estudios, pero en la cual hay serias dificultades en el diseño de algoritmos y programación debido a una baja comprensión del problema y a un diseño impreciso del algoritmo y por ende a una codificación con errores lógicos y sintácticos.

En tal sentido, la presente investigación contribuyó con la mejora del rendimiento académico en la asignatura de Fundamentos de Programación, ya que apoyó a una mejor comprensión y análisis del problema y con ello mejorar el diseño de algoritmos que es la base para la codificación del programa.

1.3.3 Justificación metodológica

Los resultados de la investigación determinaron en qué medida la estrategia de enseñanza INPAL, basada en la inducción y teorías educativas, es pertinente para contribuir a la mejora en el diseño de algoritmos desde la comprensión del problema, la determinación de variables de entrada, salida, proceso hasta la elaboración de algoritmos. Los resultados obtenidos en la presente investigación podrían ser socializados entre docentes y estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas, para luego socializarlo en las diferentes carreras profesionales de la Universidad Nacional de Cajamarca que dictan las asignaturas relacionadas con algoritmos y programación, primero como antecedente de trabajo y segundo como una estrategia de enseñanza para contrarrestar el problema del bajo nivel en el diseño de algoritmos computacionales.

1.4 Delimitación de la investigación

1.4.1 Epistemológica

La investigación corresponde al paradigma Positivista, que tiene como objetivo “descubrir leyes” de los fenómenos existentes a través de una búsqueda racional y objetiva, es decir parte de la observación de los hechos, con el fin de generar conocimiento. El investigador sólo tiene que aplicar su inteligencia y su observación metódica para encontrar o descubrir las leyes que lo rigen, único sentido de la ciencia (Reza Becerril, 1997, p. 94-95).

Por tanto, el presente estudio obedece al paradigma positivista, ya que se observó los hechos suscitados en las aulas, con respecto al nivel del diseño de algoritmos de los estudiantes de la asignatura de Fundamentos de Programación de la carrera profesional de Ingeniería de Sistemas en el año 2020, a través una rúbrica de evaluación, con el fin de descubrir la influencia de la estrategia INPAL en dicho escenario.

1.4.2 Espacial

La presente investigación se realizó en la Universidad Nacional de Cajamarca, sede central ubicada en el distrito de Cajamarca, con dirección en la Av. Atahualpa N° 1050, específicamente en la Escuela Académico Profesional de Ingeniería de Sistemas.

1.4.3 Temporal

La investigación tomó un periodo de 36 meses (del 01 de enero del 2020 al 31 de diciembre del 2022).

1.4.4 Línea de Investigación

La presente investigación pertenece a la línea de investigación de “Curriculum, didáctica e interculturalidad” y al eje temático de “Didáctica y pensamiento complejo”.

1.5 Objetivos de la investigación

1.5.1 Objetivo general

Determinar la influencia de la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL), en el diseño de algoritmos de la asignatura “Fundamentos de Programación”, en los estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020.

1.5.2 Objetivos específicos

Determinar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, antes de la aplicación de la estrategia de enseñanza INPAL.

Aplicar la estrategia de enseñanza INPAL basada en los resultados del pretest, para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas.

Determinar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, después de la aplicación de la estrategia de enseñanza INPAL.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes de la investigación

A nivel internacional

En Turquía, el estudio titulado *The effect of applying 4-stages on learning analysis and design of algorithms*, de Ulker (2020), tuvo como objetivo proponer y aplicar un nuevo método de 4 etapas para mejorar el aprendizaje y enseñanza de algoritmos.

El método se aplicó a los estudiantes de segundo año del curso de Análisis y Diseño de Algoritmos de la Universidad Europea de Lefke, el cual consistió de 4 etapas: primero, comprender los conceptos; segundo, formar un algoritmo utilizando cualquier lenguaje de programación; tercero, proporcionar varios conjuntos de problemas y cuarto, mezclar los conjuntos de problemas. Los resultados fueron que, con el método de etapas, la tasa de logro de los resultados de aprendizaje aumentó sustancialmente, pues el 84% de los estudiantes así lo indicaron en las encuestas. Además, el instructor evaluó la tasa de logros en base a las calificaciones, obteniendo que el 62% de los estudiantes obtuvieron, una calificación C o superior, 21% de los estudiantes pasaron el curso con la nota promedio y el 17% obtuvieron F o FA, cuyos resultados fueron superiores con respecto al semestre anterior, en donde la tasa de éxito fue del 45%, el 26% de los estudiantes aprobaron con la nota promedio y el 29% obtuvo F o FA.

Finalmente, se concluyó que el método de las 4 etapas cumplió con los objetivos del curso de Análisis y Diseño de Algoritmos con tasas notables y que los aprendizajes del curso se lograron con altas tasas de éxito en comparación con el semestre anterior.

Este antecedente fue considerado, por el método aplicado, principalmente por la segunda y tercera etapa, pues son elementos de enseñanza similares a los aplicados en la propuesta de la presente investigación, coincidiendo con el diseño de algoritmos, pero a nivel de pseudocódigo y con el conjunto de problemas planteados a los estudiantes, con distintos niveles de dificultad.

En China, el estudio titulado *Teaching Methods Reform of “Algorithm Design and Analysis” Course*, de Zhao y Li (2020), tuvo como objetivo proponer métodos de enseñanza de algoritmos basados en heurísticas, indagación discusión y casos, organización de la enseñanza como experimentos jerárquicos y trabajo en grupo y finalmente sugirió métodos de evaluación diversificados para motivar a los estudiantes y mejorar su autonomía.

Se han obtenido resultados docentes positivos, después de casi siete años de reforma docente, pues las puntuaciones de los estudiantes han mejorado, el número de estudiantes que participan en concursos de diseño de software ha aumentado anualmente, y según el cuestionario los estudiantes son capaces de aplicar lo aprendido, modelar problemas prácticos y diseñar algoritmos.

Este estudio fue considerado por algunos elementos del método empleado, como asignación de problemas de diferente grado de dificultad para enseñar de acuerdo con su aptitud y conocimientos previos, incorporación de problemas prácticos de contexto, trabajo en equipo y uso de material didáctico.

En España, la tesis doctoral titulada *Escenarios de aprendizaje personalizados a partir de la evaluación del pensamiento computacional para el aprendizaje de competencias de programación mediante un entorno b-Learning y gamificación*, de Rojas López (2019), tuvo como objetivo contribuir a la adquisición de las competencias básicas de programación de los estudiantes, en un ambiente b-Learning diseñado con base a su singularidad de aprendizaje y personalización de contenidos a partir del nivel cognitivo de su pensamiento computacional, además de agregar gamificación como acción motivadora.

La investigación tuvo dos fases, la primera donde se pudo realizar una planeación de las actividades, la evaluación del pensamiento computacional, la creación de escenarios de aprendizaje con base a los resultados obtenidos y la creación de los elementos de gamificación; la segunda etapa en donde se llevó a cabo la intervención de las estrategias en el aula y la recolección de información que midió la eficacia de las acciones propuestas.

Se aplicó la metodología a 135 estudiantes, y se contó con un grupo control y un grupo experimental, y desde la perspectiva cuantitativa, los indicadores académicos obtenidos son mejores y la intervención que combinó el pensamiento computacional, educación personalizada y b-Learning representó una novedad favorable para el caso de estudio y desde la perspectiva cualitativa, el 75% de los encuestados estuvo de acuerdo con su modalidad de aprendizaje.

Se concluye que la motivación del estudiante, a través del reconocimiento de las habilidades de pensamiento computacional que posee, la oferta de opciones de estudio considerando su individualidad en el proceso de aprendizaje de las competencias básicas de programación en un ambiente de gamificación y el uso de

un entorno b-Learning, permite reducir el porcentaje de la deserción universitaria sin sacrificar el nivel de calidad educativo.

Esta investigación fue considerada, por las estrategias educativas propuestas para la adquisición de las competencias básicas de programación en los estudiantes, educación personalizada en base a la evaluación del pensamiento computacional, situación que es similar en la presente investigación; ya que, según los resultados del pretest, se diseñó la propuesta de enseñanza INPAL, en cuyas sesiones de aprendizaje se consideran niveles de complejidad en la enseñanza del diseño de algoritmos.

En Colombia, el estudio titulado *Metodología de aprendizaje para descomponer números naturales en factores primos usando Programación Funcional*, de Trejos Buriticá (2019), se propone una forma metodológica para enseñar y aprender a resolver el problema de descomponer, algorítmicamente, un número natural en sus factores primos valiéndose de los principios y recursos de la programación funcional dentro de un curso de programación de computadores de un programa de ingeniería a través de una metodología propuesta de 10 fases que ha permitido exponer una forma conjunta de construir soluciones a problemas comunes y cotidianos interdisciplinariamente pues vincula la programación, la lógica y las matemáticas.

Se aplicó la metodología sugerida a 62 estudiantes y la metodología tradicional a 61 estudiantes, obteniéndose un promedio del grupo experimental de 4.57, y un promedio del grupo control de 3.43, pudiéndose notar que se logró muy buenos resultados en la aplicación de la mencionada metodología de enseñanza y aprendizaje. Los resultados de la presente investigación indican que a partir de

problemas heredados de las matemáticas se pueden encontrar soluciones eficientes valiéndose de la programación de computadores y que estos problemas pueden ser familiares para los estudiantes.

Se concluye que la metodología propuesta es efectiva y puede explotarse mucho más toda vez que, desde lo didáctico, el docente se capacite para establecer nexos transversales entre matemáticas y programación en favor de los objetivos de aprendizaje trazados.

Este antecedente, fue considerado por la metodología empleada, pues está basada en encontrar una solución vinculando la programación (algoritmos), la lógica y las matemáticas, asociación que es empleada en la estrategia de enseñanza INPAL.

En España, el estudio titulado *Estrategia de motivación para el razonamiento de algoritmos computacionales mediante juegos*, de Chezzi et al. (2017), tuvo como objetivo proponer una estrategia de aprendizaje basada en juegos para estimular el razonamiento en la construcción de algoritmos por parte de los estudiantes.

En este estudio, se proponen secuencias didácticas, a través de juegos con base matemática, para propiciar de manera significativa la construcción de algoritmos computacionales; es decir mediante el juego los estudiantes comienzan el aprendizaje de la programación sin estar sujetos a las presiones que producen el tema a abordar; seleccionando un juego deben estudiar sus metas y reglas, de modo que ellos mismos van construyendo una estrategia metodológica para la definición y análisis del problema; jugando comienzan a experimentar las acciones, los problemas que se pueden presentar y los modos de saltarlos; de la experimentación con el juego se concretan las tareas necesarias para lograr la meta y ganar el juego;

finalmente, a partir de estas tareas se definen las estructuras de control y de datos para el diseño del algoritmo.

Finalmente, se concluye que en los trabajos prácticos se pudo lograr un razonamiento que permitió abstraer las sentencias del algoritmo y los datos necesarios y del juego real se diseñan escenarios de pruebas, los cuales pueden ser aplicados al diseño del algoritmo computacional. Además, como la asignatura se dicta en marco de las carreras de ingeniería, en primer año, se relaciona con la especialidad a través de solución de problemas matemáticos o físicos.

Este antecedente, fue considerado por los juegos con base matemática que permitieron propiciar un aprendizaje significativo en los estudiantes, pues la estrategia de enseñanza INPAL, también se basa en problemas matemáticos para apoyar el análisis del problema y el diseño de algoritmos.

En Brazil, el artículo titulado *A Practical Strategy for Teaching Algorithms (CSI) to Beginners*, de Guimaraes y Gnecco (2017), tuvo como objetivo presentar una estrategia práctica para enseñar algoritmos a principiantes en informática, abordando cuestiones como la estrategia de enseñanza, la participación de los estudiantes y la planificación institucional.

Esta estrategia, tiene de 3 etapas: Inicial, donde los estudiantes se enfocan en la lógica de programación, por lo que elaboran el pseudocódigo; Intermedia, los estudiantes comprenden el objetivo de cada algoritmo y explican con claridad los pasos para resolver el problema; Avanzada, los estudiantes utilizan una herramienta computacional para implementar el algoritmo y resolver el problema. La estrategia fue desarrollada e implementada durante 7 años de enseñanza de algoritmos, la cual se evaluó través de una encuesta sobre las tasas de aprobación, reprobación y

abandono y mediante la observación y la retroalimentación recopilada durante las sesiones con los estudiantes a lo largo de los años. La tasa de aprobación fue aumentando con el paso del tiempo, desde el 41,43% en el 2003 hasta el 75.47% en el 2009, lo que quiere decir que cada vez mayor cantidad de estudiantes pudieron completar problemas más complejos.

Finalmente, se concluye que la estrategia implementada tuvo resultados positivos en los estudiantes, y ello se debe a una mejor comprensión del problema por parte de los estudiantes, a un comportamiento pedagógico dentro del aula por parte del docente y al material didáctico empleado.

Este antecedente, fue considerado por las etapas de la estrategia, principalmente por la etapa inicial, pues considera que no se debe pasar por alto las fases de análisis y prueba, pues son imprescindibles para desarrollar la solución de problemas; pues una etapa similar es considerada en la estrategia de enseñanza INPAL, donde se prioriza la comprensión del problema a través de una solución matemática del problema.

En Cuba, el estudio titulado *Una Propuesta Didáctica para Perfeccionar la Algoritmización Computacional*, de Salgado Castillo et al. (2015), tuvo como objetivo la elaboración de un sistema de procedimientos didácticos para perfeccionar el proceso de algoritmización computacional. Este sistema fue estructurado en cuatro procedimientos: construcción lógico - matemática, orientación matemático - algorítmica, estructuración algorítmico - generalizadora y validación algorítmico - computacional.

El sistema de procedimientos didácticos se aplicó en un subgrupo de 23 alumnos durante un periodo de tiempo de 18 semanas. Los resultados, después del

experimento pedagógico, fueron que existen diferencias significativas entre las calificaciones de los estudiantes del grupo control y experimental en la postprueba, siendo estas últimas significativamente mayores, lo que implica un mejor aprovechamiento académico.

Se concluyó que el sistema de procedimientos didácticos brinda suficientes evidencias sobre su influencia positiva en el perfeccionamiento del aprendizaje de la algoritmización.

Este estudio fue considerado por el sistema propuesto, pues incluye 4 procedimientos: construcción lógico - matemática, orientación matemático - algorítmica, estructuración algorítmico - generalizadora y validación algorítmico - computacional, pues tienen cierta similitud con los empleados en la estrategia de enseñanza INPAL.

A nivel nacional

Lagos Barzola (2020), en su tesis doctoral titulada *Método de Aprendizaje Cooperativo y el Aprendizaje de Algoritmos de Programación en Estudiantes de la Universidad Nacional de Ayacucho*, tuvo como objetivo determinar el efecto que origina la aplicación del Método de Aprendizaje Cooperativo en el aprendizaje del diseño de algoritmos de los estudiantes de la Universidad Nacional de San Cristóbal de Huamanga - 2019.

Esta investigación, tuvo un diseño cuasi experimental, con una muestra de 46 estudiantes del curso de Algoritmos, con pre y post test y con un grupo control y un grupo experimental a los cuales se les aplicaron el Método de Enseñanza Tradicional y el Método de Aprendizaje Cooperativo respectivamente. Los

resultados obtenidos (prueba U Mann-Whitney), en el pre test, fueron que en el diseño de algoritmos: en la identificación y uso de variables adecuadas en el diseño de algoritmos, en el uso adecuado de las estructuras de control de flujo de selección, no existe diferencia significativa entre el grupo control y grupo experimental. Además, en la prueba de conocimientos tampoco existe diferencia significativa entre el grupo control y grupo experimental. En los resultados del post test, se encontró que en el diseño de algoritmos: en la identificación y uso de variables adecuadas y en el uso adecuado de las estructuras de control de flujo de selección no existe diferencia significativa entre el grupo control y grupo experimental; sin embargo, en el uso adecuado de las estructuras de control de flujo de repetición y en la combinación adecuada de las estructuras de control de flujo de selección y repetición, existe diferencia significativa entre el grupo control y grupo experimental, pues hubo una mejora significativa en el diseño de algoritmos del grupo experimental. En la nota de la prueba de conocimientos - post test, existe diferencia significativa entre el grupo control y grupo experimental, pues hubo una mejora significativa en las notas del grupo experimental.

Finalmente, se concluyó que la aplicación del Método de Aprendizaje Cooperativo tiene efectos significativos en el aprendizaje del diseño de Algoritmos de los estudiantes del primer año de la Escuela de Formación Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Cristóbal de Huamanga – 2019.

Este antecedente fue considerado, por el método de aprendizaje cooperativo, pues en la propuesta se considera trabajo en pares con el fin de que los estudiantes puedan cooperar y aprender juntos. Además, fue considerado por la variable diseño de algoritmos que es también la variable dependiente de la presente investigación.

Larico Uchamaco (2018), en su tesis doctoral titulada *El aula virtual y el aprendizaje del algoritmo en los estudiantes de la escuela profesional de ingeniería de sistemas e informática de la Universidad Nacional Amazónica de Madre de Dios, región Madre de Dios - 2017*, tuvo como objetivo determinar el grado de relación de aula virtual y el aprendizaje del algoritmo de los estudiantes de la escuela profesional de ingeniería de sistemas e informática.

Se contó con una muestra de 74 estudiantes del curso de algoritmos del primer semestre de la escuela profesional de ingeniería de sistemas e informática, obteniéndose como resultado, con referencia al aula virtual, el 32% indicaron una relación bueno y 54% de los estudiantes encuestados manifestaron una relación regular.

Se concluye que al emplear las aulas virtuales en estudiantes universitarios tiene un resultado positivo en el aprendizaje del algoritmo, y en función a la planificación y material educativo del curso del aula virtual hay una relación significativamente con el aprendizaje del algoritmo en los estudiantes.

Esta investigación fue considerada por la dimensión planificación y material educativo del curso de aula virtual, que fueron elementos considerados en la propuesta de estrategia de enseñanza INPAL, además cabe indicar que esta modalidad de enseñanza haciendo uso de herramientas virtuales también fue empleada en la presente investigación, pues fue llevada a cabo durante la pandemia de COVID-19.

Estrada Aro (2016), en su tesis doctoral titulada *Software Pseint en los niveles cognitivos en estudiantes del curso Principios de Algoritmos de la Universidad Tecnológica del Perú - Lima*, tuvo como objetivo determinar la influencia del uso

del software Pseint en los niveles cognitivos del aprendizaje de los estudiantes en el curso de Principios de Algoritmos de los estudiantes. Los indicadores fueron los niveles cognitivos de recuperación, comprensión, análisis, aplicación y metacognición del aprendizaje.

El tipo de investigación fue de tipo aplicada y el diseño fue experimental, la muestra fue 50 estudiantes tomando 25 estudiantes como grupo control y 25 para el grupo experimental. Obteniéndose como resultado, que hubo un incremento del 24% para el nivel de recuperación y comprensión, 20% del nivel de análisis, 14% del nivel de aplicación y 15% del nivel metacognitivo.

Se concluyó que existe influencia positiva del uso del software Pseint en los niveles cognitivos del aprendizaje del curso de Principios de Algoritmos de los estudiantes de la Universidad Tecnológica del Perú - Lima.

Esta investigación fue considerada por los niveles cognitivos de recuperación, comprensión y aplicación, que son niveles cognitivos considerados en las sesiones de aprendizaje de la estrategia INPAL.

A nivel local, no se encontraron antecedentes relacionados con la presente investigación.

2.2 Marco epistemológico

El desarrollo de la investigación fue orientado por el paradigma positivista, pues partió de la observación de los hechos con el fin de “descubrir” las relaciones existentes entre los hechos existentes; es decir, al recoger datos, con una rúbrica de evaluación, sobre el nivel de diseño de algoritmos antes y después de aplicada la

estrategia de enseñanza INPAL, en los estudiantes del primer ciclo de la carrera profesional de Ingeniería de Sistemas, para luego analizarlos metódica y empíricamente, con el fin de descubrir verdades o leyes que lo rigen; en este caso, determinar la influencia de la estrategia de enseñanza INPAL en el diseño de algoritmos de la asignatura de Fundamentos de Programación (Reza Becerril, 1997).

Asimismo, el presente estudio pertenece al campo de la educación, donde el positivismo aspira descubrir leyes sobre los fenómenos educativos y elaborar teorías científicas, cuyo propósito en esta investigación es la explicación de los hechos educativos en la educación superior, ya que trata de explicar cómo influencia la estrategia de enseñanza INPAL en el diseño de algoritmos de la asignatura de Fundamentos de Programación en los estudiantes de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca.

Además, en la presente investigación, se pueden evidenciar la aplicación de los criterios del paradigma positivista, pues se empleó el método hipotético-deductivo, marco teórico natural, se planteó la hipótesis de la investigación, se definió las variables de estudio (estrategia de enseñanza y diseño de algoritmos), se realizó la operacionalización de las variables, muestra por conveniencia, aplicación de métodos estadísticos para la contrastación de la hipótesis; presentación de la información de forma estructurada y sistematizada, variables asociadas con el marco teórico de la investigación y diseño cuasi-experimental (Pérez Serrano, 2016).

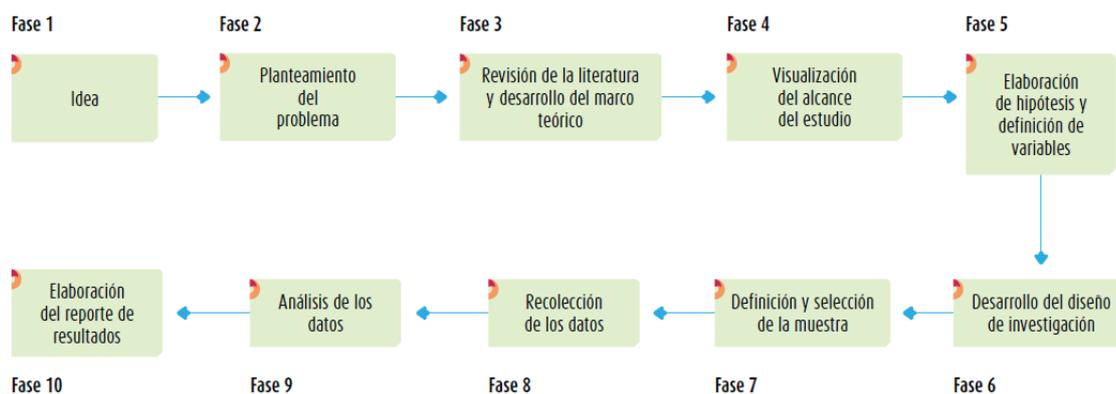
La presente investigación, se llevó a cabo bajo el enfoque cuantitativo, el cual es secuencial y probatorio; parte de una idea que va acotándose, luego se derivan

objetivos y preguntas de investigación, se revisa la literatura y se construye un marco teórico. Se establecen hipótesis y determinan variables; se traza un plan para probarlas (diseño); se miden las variables en un determinado contexto; se analizan las mediciones obtenidas utilizando métodos estadísticos, y se extrae una serie de conclusiones respecto de las hipótesis (Hernández Sampieri et al. 2014).

Los análisis cuantitativos se interpretan a la luz de las predicciones iniciales (hipótesis) y de estudios previos (teoría). La interpretación constituye una explicación de cómo los resultados encajan en el conocimiento existente. La investigación cuantitativa debe ser lo más “objetiva” posible. Los fenómenos que se observan o miden no deben ser afectados por las percepciones propias del investigador y que tampoco sean alterados por las tendencias de otros (Hernández Sampieri et al. 2014).

Figura 1

Proceso Cuantitativo.



Nota. Fuente: Hernández Sampieri (2014).

Por tanto, la investigación siguió las fases del proceso cuantitativo, utilizó la recolección de datos para probar la hipótesis con base a la medición numérica y al análisis estadístico, para la obtención de resultados.

2.3 Marco teórico - científico de la investigación

Las teorías educativas, en las cuales se fundamenta la presente investigación es la teoría del aprendizaje significativo, la teoría del aprendizaje Gagné y la teoría de las situaciones didácticas. Además, se fundamenta en la base teórica de estrategias de enseñanza y la base teórica de diseño de algoritmos.

2.3.1 La asignatura de Fundamentos de Programación en el currículo de estudios de Ingeniería de Sistemas - 2018

El currículo de estudios de Ingeniería de Sistemas - 2018, dentro de su marco legal contempla al Modelo de Acreditación para Programas de Estudios de Educación Superior Universitaria, en este sentido el diseño del currículo hace referencia a una planificación organizada de las actividades pedagógicas que lleva a cabo el programa de estudios de Ingeniería de Sistemas para lograr los fines educativos, como alcanzar el perfil de egreso de los estudiantes. Estas actividades pedagógicas involucran la ejecución de las unidades didácticas, la selección de los cursos y del sistema de evaluación, así como las pautas metodológicas comunes a desarrollar en la Universidad (SINEACE, 2018).

El currículo de estudios de Ingeniería de Sistemas - 2018, considera a la línea curricular de “Algoritmos y Programación”, y una de sus asignaturas básicas es “Fundamentos de Programación” correspondiente al primer ciclo de estudios, la cual contribuye a lograr la competencia “Construye Programas computacionales en consola, considerando los fundamentos de algoritmos y los lenguajes de programación, para solucionar problemas básicos en distintas

áreas del conocimiento”. Con la presente investigación se busca contribuir al logro de los siguientes resultados de aprendizaje:

- Construye algoritmos y programas con estructura secuencial, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.
- Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Y para lograr los resultados de aprendizaje mencionados, de una manera significativa, se propone la estrategia de enseñanza INPAL, que mejorará el nivel de diseño de algoritmos computacionales y apoyará a la creación de programas eficientes y con ello contribuirá a lograr la competencia específica de nivel básico del eje curricular de “Algoritmos y Programación”.

2.3.2 Teoría del Aprendizaje Significativo en la Estrategia de Enseñanza “Inducción para Algoritmos” - INPAL, en la Educación Superior Universitaria

Frente al bajo rendimiento académico de los estudiantes de primer ciclo de la carrera profesional de Ingeniería de Sistemas en el diseño de algoritmos computacionales de la asignatura de “Fundamentos de Programación”, se ha diseñado una estrategia de enseñanza “Inducción para Algoritmos” - INPAL, la cual tiene 2 dimensiones: la solución matemática del problema y la

asociación de la solución matemática y la solución algorítmica, con el fin de mejorar la comprensión del problema y en consecuencia mejorar el diseño de algoritmos y con ello mejorar el rendimiento académico de los estudiantes.

En el aprendizaje significativo, el proceso de adquirir información produce una modificación tanto de la información acabada de adquirir como del aspecto específicamente pertinente de la estructura cognitiva con el que se vincula la nueva información. En la mayoría de los casos, la nueva información se relaciona con un concepto o una proposición de carácter específico y pertinente. El aprendizaje significativo supone una interacción selectiva entre el nuevo material de aprendizaje y las ideas preexistentes en la estructura cognitiva, lo que denominaremos anclaje, para expresar una conexión en el tiempo con las ideas preexistentes. (Ausubel, 2002)

El diseño de la estrategia de enseñanza INPAL, se ha basado en el proceso de recepción de la teoría del aprendizaje significativo, que lo considera como un proceso intrínsecamente activo, que a su vez requiere como mínimo:

1. El tipo de análisis cognitivo necesario para determinar qué aspectos de la estructura cognitiva ya existente son más pertinentes al nuevo material potencialmente significativo. En base a esto, se ha planteado la primera dimensión de la estrategia de enseñanza INPAL “Solución matemática del problema”, la que busca resolver el problema propuesto asignando valores numéricos, para luego realizar los cálculos matemáticos necesarios que permitirán obtener el resultado, que también será un valor numérico. Con esta dimensión de la estrategia se está identificando y

utilizando los aspectos, ya existentes, de la estructura cognitiva, pertinentes para el nuevo conocimiento.

2. Algún grado de conciliación con ideas ya existentes en la estructura cognitiva, es decir percibir similitudes y diferencias y resolver contradicciones aparentes o reales, entre conceptos y proposiciones nuevos y ya establecidos. Basándose en este enunciado, se ha planteado la segunda dimensión de la estrategia “Asociación de solución matemática y solución algorítmica”, a través de la cual se busca establecer principalmente similitudes entre los valores y las variables del algoritmo, para luego establecer una asociación entre éstas que permitan una mejor comprensión del problema y con ello una mayor asimilación del nuevo conocimiento.
3. La reformulación del material de aprendizaje en función del vocabulario y del fondo intelectual idiosincrásico de la persona concreta que aprende. En tal sentido se elaboró material didáctico dirigido a estudiantes que inician sus estudios de algoritmos y programación, en el cual se muestra la implementación de las dimensiones de la estrategia de enseñanza INPAL, en la resolución de ejercicios.

La naturaleza y las condiciones del aprendizaje significativo, activo y basado en la recepción también exigen un tipo de enseñanza expositiva que reconozca los principios de la diferenciación progresiva y de la conciliación integradora en los materiales de instrucción. (Ausubel, 2002)

El primer principio reconoce que la mayoría del aprendizaje y toda la retención y la organización de la materia es de naturaleza jerárquica; en función del nivel de abstracción, generalidad e inclusividad. El segundo

principio, la conciliación integradora se facilita en la enseñanza expositiva si el enseñante y/o los materiales de la instrucción prevén explícitamente las similitudes y las diferencias entre las ideas nuevas y las ideas ya existentes en las estructuras cognitivas de los alumnos. (Ausubel, 2002)

Por lo tanto, para el aprendizaje significativo del diseño de algoritmos, se tiene en cuenta ambos principios, pues primero, se realiza una solución matemática del problema, para luego se hace una diferenciación progresiva de los valores de entrada con las variables de entrada, los valores de salida con las variables de salida y el procedimiento matemático con el proceso algorítmico y toda esta secuencia de pasos se presenta a través de material didáctico como diapositivas explicativas y plantillas para la solución los ejercicios.

2.3.3 Teoría del Aprendizaje de Gagné como apoyo a las Sesiones de Aprendizaje de Fundamentos de Programación en la Educación Superior Universitaria

Con la finalidad de desarrollar sesiones de aprendizaje que permitan a los estudiantes lograr los resultados de aprendizaje 1 y 2 establecidos en el sílabo de la asignatura de “Fundamentos de Programación” y con ello contribuir a la competencia específica del eje curricular de “Algoritmos y Programación”, se han planteado sesiones de aprendizaje tomando en cuenta algunos *principios del aprendizaje*, establecidos por Robert Gagné:

Contigüidad, en la que se ha presentado la situación del estímulo simultáneamente con la respuesta deseada; es decir, se ha aplicado la

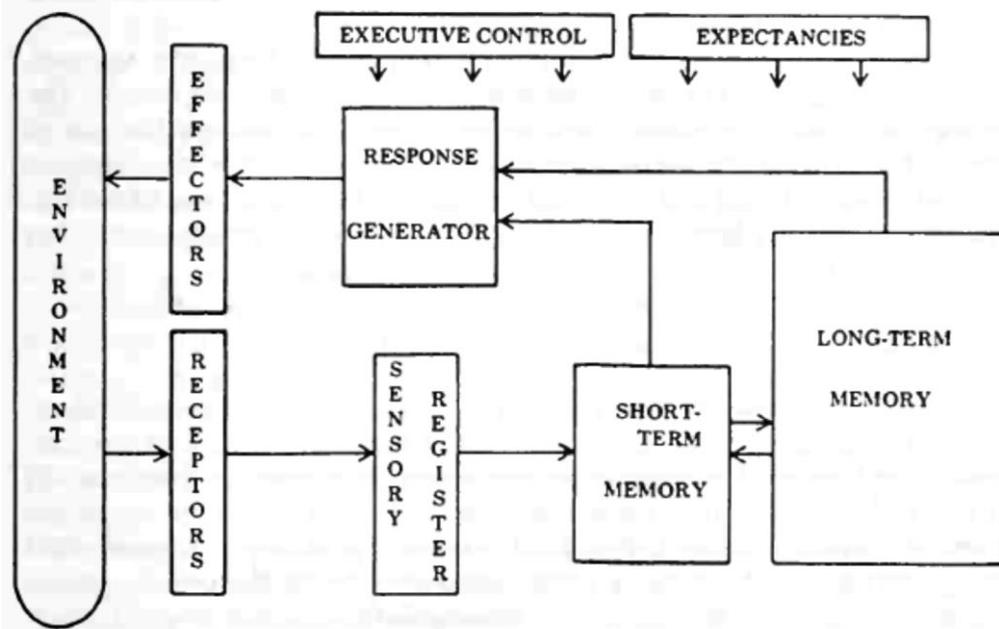
estrategia de enseñanza INPAL en la fase del análisis del problema, para tener una mejor comprensión del problema y con ello se lograba la respuesta deseada que era un correcto diseño de algoritmos.

Repetición, en donde la situación del estímulo y su respuesta se han repetido o practicado para mejorar el aprendizaje y asegurar la retención; es decir, la estrategia de enseñanza INPAL en la comprensión del problema y el diseño del algoritmo, se han repetido en cada problema planteado tanto en clase, como en las tareas asignadas a los estudiantes, e incluso en las evaluaciones, lo que mejoró el aprendizaje y garantizó la retención de estos contenidos académicos.

Reforzamiento, en donde el conocimiento de una nueva acción se fortaleció, al ser seguida por un estado satisfactorio; al comprender el problema, se diseñará correctamente el algoritmo, y el estudiante se sentirá satisfecho cuando su programa arroje los resultados deseados.

Figura 2

Un modelo básico de aprendizaje y memoria, subyacente a las teorías cognitivas modernas (procesamiento de información).



Nota. Fuente: Gagné et al. (1992).

Para tener en cuenta las condiciones del aprendizaje, debemos tener en cuenta un modelo de los procesos involucrados en un acto de aprendizaje. El modelo presentado en la Figura 2, concibe el aprendizaje como procesamiento de información, lo que se explica de la siguiente manera:

La estimulación del entorno del estudiante, activa los receptores y se transmite como información al sistema nervioso central. La información logra un breve registro en uno de los registros sensoriales y luego se transforma en patrones reconocibles que ingresan a la memoria a corto plazo. La transformación que ocurre en este punto se llama percepción selectiva.

El almacenamiento de información en la memoria a corto plazo tiene una duración relativamente breve, menos de 20 segundos. La información a recordar se transforma nuevamente mediante un proceso llamado

codificación semántica en una forma que ingresa a la memoria a largo plazo. Cuando se codifica, la información en la memoria a largo plazo es significativa; gran parte tiene la forma de proposiciones, es decir, entidades del lenguaje que poseen sujetos y predicados similares a oraciones. De esta forma, la información puede almacenarse durante largos periodos de tiempo, puede volver a la memoria a corto plazo mediante el proceso de recuperación, y estos elementos recuperados pueden combinarse con otros para generar nuevos tipos de aprendizaje.

La información de la memoria a corto plazo o a largo plazo, cuando se recupera, pasa a un generador de respuestas y se transforma en acción. El mensaje activa los efectores (músculos), produciendo una actuación que puede observarse en el entorno del estudiante. Esta acción es lo que permite a un observador externo decir que la estimulación inicial ha tenido el efecto esperado. La información ha sido “procesada” de todas estas formas, y el alumno ciertamente ha aprendido.

Los factores que influyen en el aprendizaje son llamados *condiciones de aprendizaje*, las que pertenecen a los estímulos externos al estudiante como otras son condiciones internas del estudiante. En cada sesión de aprendizaje se emplearon estímulos o eventos externos como el material didáctico, el aula virtual (Google meet y SIA) y la exposición y explicación del docente, además también se consideraron las condiciones internas pues se busca traer de la memoria de los estudiantes conocimientos previos para lograr la asociación con el nuevo conocimiento, que garantice un aprendizaje significativo.

Con respecto a los *Procesos de Instrucción y Aprendizaje*, se considera que, si la instrucción ha de producir un aprendizaje efectivo, de alguna manera se debe hacer que influya en los procesos internos de aprendizaje implícitos en el flujo de información y memoria del estudiante (Figura 2). Los eventos externos pueden afectar estos procesos en una variedad de formas, algunas de las cuales apoyan el aprendizaje (Gagné, 1992).

En tal sentido, se han diseñado sesiones de aprendizaje considerando eventos externos como la explicación, uso de material didáctico y resolución de ejercicios con el fin de apoyar los procesos internos de aprendizaje del estudiante con respecto al diseño de algoritmos con estructura secuencial y condicional.

2.3.4 Teoría de las Situaciones Didácticas en las Sesiones de Aprendizaje de Fundamentos de Programación en la Educación Superior Universitaria

Una situación didáctica es un modelo de interacción entre un estudiante y un medio determinado, incluidos el docente y el sistema educativo, es decir diseñar situaciones que sirvan para enseñar un determinado conocimiento. El recurso del que dispone el estudiante para alcanzar en este medio un estado favorable es una variedad de decisiones que dependen del uso de conocimientos previos; y el medio, un subsistema autónomo, en donde se plasmas las circunstancias para la difusión y adquisición de los conocimientos.(Brousseau, 2007)

En tal sentido, se ha diseñado una situación didáctica en las sesiones de aprendizaje de fundamentos de programación, basadas en una estrategia de

enseñanza denominada “Inducción para Algoritmos” - INPAL para lograr una mejor comprensión y análisis del problema y con ello un mejor diseño de algoritmos computacionales en los estudiantes de primer ciclo de la carrera de Ingeniería de Sistemas.

Según Brousseau, las fases que se contemplan en el desarrollo de una situación didáctica son:

Situación de acción, la que corresponde a una situación típica de acción, donde los estudiantes toman decisiones buscando construir un método de resolución, para lo cual aceptará o rechazará una solución según su eficacia. En la estrategia de enseñanza INPAL, los estudiantes buscan soluciones matemáticas a un determinado problema y a su vez asociarlo a una solución algorítmica, eligiendo uno u otro procedimiento para dar solución al problema, para finalmente diseñar un algoritmo computacional.

Situación de formulación, cuando un estudiante formula explícitamente un mensaje destinado a otro estudiante (o grupo de estudiantes), quien debe comprender el mensaje y actuar en base al conocimiento contenido en dicho mensaje. En la estrategia de enseñanza INPAL, una vez que los estudiantes han elegido el procedimiento matemático, la asociación algorítmica y el diseño del algoritmo para dar solución al problema, ellos lo comparten en pares de estudiantes, con el fin que el segundo estudiante la comprenda y lo aplique para determinar su eficacia.

Situación de Validación, consiste no sólo en comunicar un mensaje, sino en demostrar la verdad de sus soluciones propuestas. En la estrategia de enseñanza INPAL, cada par de estudiantes presentaban y justificaban la

solución encontrada al problema, en algunos casos llamando al docente para sustentar su procedimiento de solución o en la pizarra frente a sus compañeros de clase, con la finalidad de validar su solución propuesta y de ser posible optimizarla.

La institucionalización, consiste en tomar el objeto de conocimiento y el aprendizaje del estudiante, por parte del docente para comprobar lo que los estudiantes deben hacer o no, o que han aprendido o tienen que aprender. Este componente se da en la situación de acción, cuando se reconoce el valor de un procedimiento, en la situación de formulación, cuando el docente hace preguntas para determinar las formulaciones que valen la pena conservarlas, y en la situación de validación al identificar las propiedades encontradas que se van a conservar. En la estrategia de enseñanza INPAL, la docente guía a los estudiantes en cada una de las fases de la situación didáctica, escuchando y valorando sus procedimientos de solución y haciendo preguntas a los estudiantes para que ellos mismos realicen ajustes buscando una solución óptima o descarten aquellas que son ineficaces, buscando sustento en el saber científico.

2.3.5 El conectivismo en la educación remota universitaria en época de pandemia COVID 19.

El conectivismo es la integración de principios explorados por las teorías de caos, redes, complejidad y auto-organización y considera al aprendizaje como un proceso que ocurre al interior de ambientes difusos de elementos centrales cambiantes, el aprendizaje puede residir fuera de nosotros

(una organización o una base de datos) y está enfocado en conectar conjuntos de información especializada. Además, está orientado por la rapidez con que cambia la información y la relevancia de la habilidad de distinguir entre la información importante y no importante; siendo vital la habilidad de reconocer cuándo una nueva información altera un entorno basado en información anterior. (Siemens, 2004)

Según Siemens, entre los principios del conectivismo tenemos:

- El aprendizaje es un proceso de conectar nodos o fuentes de información especializados.
- La alimentación y mantenimiento de las conexiones es necesaria para facilitar el aprendizaje continuo.
- La actualización (conocimiento preciso y actual) es la intención de todas las actividades conectivistas de aprendizaje.
- La toma de decisiones es un proceso de aprendizaje. El acto de escoger qué aprender y el significado de la información que se recibe, es visto a través de una realidad cambiante; es decir, una decisión correcta hoy, puede estar equivocada mañana.

El Perú estaba experimentando un proceso de masificación de la educación universitaria, cuando en marzo del 2020 apareció la pandemia del COVID 19 en nuestro país, lo que ocasionó una disminución en el porcentaje de jóvenes matriculados en educación universitaria del 26.4% en el 2019 al 20.1% en el 2020. Las universidades cambiaron la modalidad de la provisión del servicio educativo e iniciaron un proceso de adaptación rápida a la modalidad virtual o remota, que tuvo como principal desafío la pobreza

digital de los estudiantes e incluso de los docentes; es decir, la falta de conectividad a internet, a la disponibilidad de medios tecnológicos (computadoras o afines) y ausencia de habilidades para interactuar en un entorno virtual con fines académicos. (Benites, 2021)

En el caso de la Universidad Nacional de Cajamarca se postergó el inicio del ciclo 2020-I hasta agosto del 2020, el cual debió iniciar en marzo del mismo año, debido a que se tuvo que cambiar de modalidad de enseñanza de la presencial a la remota, lo que representó un cambio drástico en la forma como los docentes estaban acostumbrados a dictar sus asignaturas. Por otro lado, no se contaba con la infraestructura tecnológica necesaria para tal fin como: equipos computacionales y acceso a internet y además no se tenía la suficiente destreza en el uso de herramientas tecnológicas educativas tanto en los estudiantes como en los docentes.

Sin embargo, la pandemia nos presentó el desafío de la educación remota, y a la vez permitió buscar soluciones tecnológicas como el uso de las aplicaciones de Google para la educación como Google Meet para desarrollar las sesiones virtuales y el Sistema Informático Académico de la Universidad Nacional de Cajamarca, que principalmente nos permitió registrar horarios, subir material académico, registrar tareas y registrar calificaciones, lo que estuvo acompañado de capacitaciones, en el uso de dichas herramientas, por parte de la Oficina de Informática de la Universidad Nacional de Cajamarca y además se contó con el apoyo de la Escuela Académico Profesional de Ingeniería de Sistemas.

En tal sentido, se puede decir que, durante el desarrollo de la investigación, año 2020, estuvo presente el conectivismo en el proceso de enseñanza-aprendizaje, pues se incluyó la tecnología, no sólo para hacer posible la educación remota, sino para la distribución del conocimiento, donde el docente jugó un rol importante, al orientar a los estudiantes a elegir fuentes confiables de información.

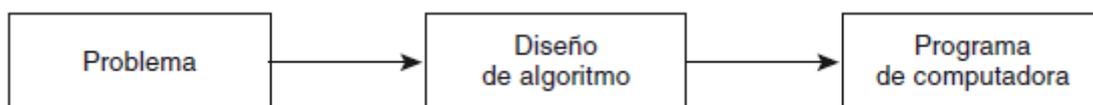
2.3.6 Algoritmos en la solución de problemas computacionales

Según (Deitel y Deitel, 2008), cualquier problema de computación puede resolverse ejecutando una serie de acciones en un orden específico. Un procedimiento para resolver un problema en términos de las acciones a ejecutar y el orden en el que se ejecutan estas acciones se conoce como un algoritmo.

Según Joyanes Aguilar (2008), un algoritmo es un método para resolver un problema; es decir, la resolución de un problema exige el diseño de un algoritmo que resuelva el problema propuesto.

Figura 3

Resolución de un problema.



Nota. Fuente: Joyanes Aguilar (2008).

Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como de la computadora que los ejecuta. En cada problema el algoritmo se puede expresar en un lenguaje diferente de programación y ejecutarse en una computadora distinta; sin embargo, el algoritmo será siempre el mismo.

Las características fundamentales de todo algoritmo son:

- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
- Un algoritmo debe estar bien definido, pues si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser finito, pues debe terminar en algún momento; es decir, debe tener un número finito de pasos.

La definición de un algoritmo debe describir tres partes: Entrada, Proceso y Salida.

Los algoritmos es un tema que se aborda desde la primera unidad de la asignatura de “Fundamentos de Programación” en la carrera profesional de Ingeniería de Sistemas, iniciando por su definición, características, ejemplos de la vida diaria y formas de representación en las estructuras básicas de control de un programa. Pero es importante resaltar que un programador de computadora es antes que nada una persona que resuelve problemas, por lo que para llegar a ser un programador eficaz se necesita aprender a resolver problemas de un modo riguroso y sistemático, es decir, resolver problemas computacionales mediante algoritmos.

2.3.7 Diseño de algoritmos en la asignatura de “Fundamentos de Programación”

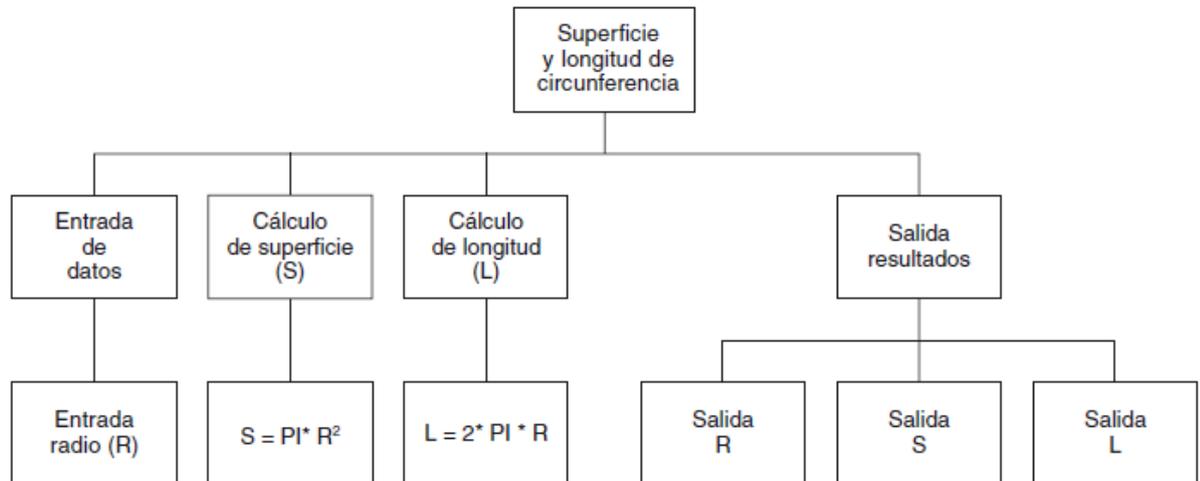
Joyanes Aguilar (2008) refiere que en la etapa de diseño se determina cómo hace el programa la tarea solicitada. Los métodos más eficaces para el proceso de diseño se basan en el conocido divide y vencerás, así que los problemas complejos se pueden resolver más eficazmente con la computadora cuando se descomponen en subproblemas que sean más fáciles de solucionar que el original. Ejemplo, en el problema de encontrar la superficie y la longitud de un círculo se puede dividir en tres problemas más simples o subproblemas.

La descomposición del problema original en subproblemas más simples y a continuación la división de estos subproblemas en otros más simples que pueden ser implementados para su solución en la computadora se denomina *diseño descendente*.

Normalmente, los pasos diseñados en el primer esbozo del algoritmo son incompletos e indicarán sólo unos pocos pasos: pero luego, éstos se amplían en una descripción más detallada con más pasos específicos. Este proceso se denomina *refinamiento del algoritmo*. Para problemas complejos se necesitan con frecuencia diferentes niveles de refinamiento antes de que se pueda obtener un algoritmo claro, preciso y completo.

Figura 4

Refinamiento del algoritmo.



Nota. Fuente: Joyanes Aguilar (2008).

Después del diseño descendente y refinamiento por pasos, es preciso representar el algoritmo mediante una determinada **herramienta de programación**: diagrama de flujo, pseudocódigo o diagrama N-S. Por tanto, el diseño del algoritmo se descompone en las fases que se muestran en la siguiente figura:

Figura 5

Fases del diseño de un algoritmo.



Nota. Fuente: Joyanes Aguilar (2008).

El diseño de algoritmos es un tema práctico de suma importancia que se aborda en la asignatura de “Fundamentos de Programación”, en donde primero los estudiantes analizan el problema, definiendo las entradas, estableciendo el proceso y determinando las salidas (diseño descendente y refinamiento por casos), para luego diseñar una solución algorítmica, en una herramienta de programación, que permita elaborar un programa de computadora que resuelva un problema específico.

2.3.8 Escritura de algoritmos en las estructuras de control secuencial y condicional en la asignatura “Fundamentos de Programación”

Según Joyanes Aguilar (2008), escribir un algoritmo consiste en realizar una descripción paso a paso con un lenguaje natural del mencionado algoritmo, teniendo en cuenta que esta secuencia definida de pasos debe realizarse hasta que se obtenga un resultado coherente y sólo puede ejecutarse una operación a la vez. En tal sentido el flujo de control más usual de un algoritmo es secuencial, seguido del flujo de control condicional. Además, cabe indicar que el método para escribir o representar un algoritmo, debe ser independiente del lenguaje de programación elegido.

El *pseudocódigo*, es un método para escribir algoritmos, en el cual el programador se puede concentrar en la lógica y en las estructuras de control y no preocuparse de las reglas de un lenguaje de programación específico. Además, es fácil modificar el pseudocódigo si se descubren errores o anomalías en la lógica del programa, y otra ventaja del pseudocódigo es que puede ser traducido fácilmente a lenguajes de programación estructurados como Java, C#, etc.

El pseudocódigo utiliza palabras reservadas similares a sus homónimas en los lenguajes de programación, como inicio, fin, si-entonces-sino, mientras-fin, repetir-hasta, entre otras. La escritura de pseudocódigo exige normalmente la indentación, es decir sangría en el margen izquierdo de diferentes líneas. Un ejemplo de pseudocódigo para escribir un algoritmo sencillo del arranque matinal de un coche sería:

```
inicio  
  //arranque matinal de un coche  
  introducir la llave de contacto  
  girar la llave de contacto  
  pisar el acelerador  
  oír el ruido del motor  
  pisar de nuevo el acelerador  
  esperar unos instantes a que se caliente el motor  
fin
```

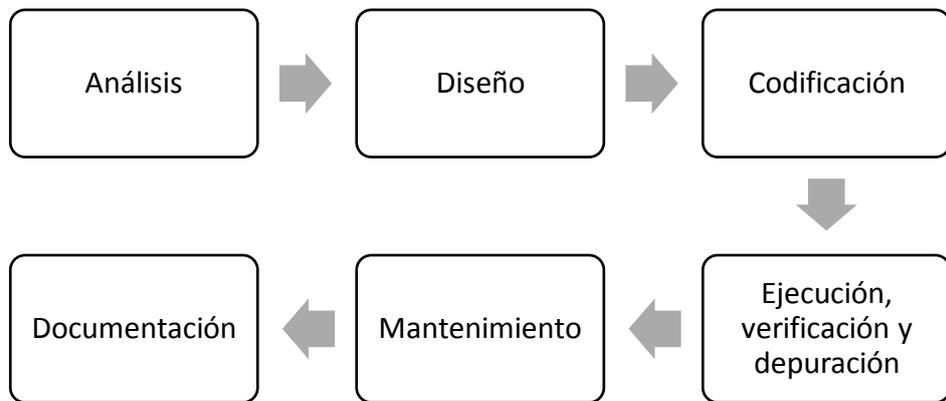
Uno de los métodos más utilizados para escribir algoritmos en la asignatura de “Fundamentos de Programación”, es el **pseudocódigo**, en donde el estudiante refleja principalmente la lógica secuencial y condicional de cómo resolver un problema, previo análisis del mismo en el que determinará las entradas, salidas y proceso apoyándose para ello en una resolución matemática en específico.

2.3.9 Fases en la resolución de problemas haciendo uso de una computadora

Joyanes Aguilar (2008) considera que el proceso de resolución de un problema con una computadora conduce a la escritura de un programa y a su ejecución en la misma. Aunque el proceso de diseñar programas es, esencialmente, un proceso creativo, se puede considerar una serie de fases o pasos comunes, que generalmente deben seguir todos los programadores.

Figura 6

Fases de resolución de un problema con computadora.



Nota. Fuente: Joyanes Aguilar (2008).

- Análisis. El problema se analiza teniendo presente la especificación de los requisitos dados por el cliente de la empresa o por la persona que encarga el programa.
- Diseño. Una vez analizado el problema, se diseña una solución que conducirá a un algoritmo que resuelva el problema.
- Codificación. La solución se escribe en la sintaxis del lenguaje de alto nivel y se obtiene un programa fuente que se compila a continuación.
- Ejecución, verificación y depuración. El programa se ejecuta, se comprueba rigurosamente y se eliminan todos los errores que puedan aparecer.
- Mantenimiento. El programa se actualiza y modifica, cada vez que sea necesario, de modo que se cumplan todas las necesidades de cambio de sus usuarios.

- Documentación. Escritura de las diferentes fases del ciclo de vida del software, esencialmente el análisis, diseño y codificación, unidos a manuales de usuario y de referencia, así como normas para el mantenimiento.

En la asignatura de Fundamentos de Programación, se abordan todas las fases de la resolución de problemas, desde el análisis, hasta el mantenimiento del programa, pero se ha prestado principal atención al análisis y diseño de algoritmos, pues con ello se garantiza un código sin errores lógicos.

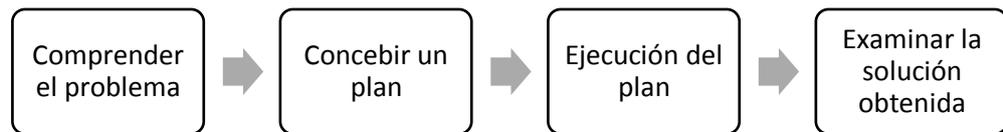
2.3.10 Plantear y resolver problemas matemáticos como apoyo al diseño de algoritmos computacionales

Polya (1989), sostiene que el método sobre plantear y resolver problemas está enfocado a la solución de problemas matemáticos, por ello es importante señalar alguna distinción entre ejercicio y problema. Para resolver un ejercicio, uno aplica un procedimiento rutinario que lo lleva a la respuesta. Para resolver un problema, uno hace una pausa, reflexiona y hasta puede ser que ejecute pasos originales que no había ensayado antes para dar la respuesta. Esta característica de dar una especie de paso creativo en la solución, no importa que tan pequeño sea, es lo que distingue un problema de un ejercicio. Sin embargo, es prudente aclarar que esta distinción no es absoluta; depende en gran medida del estadio mental de la persona que se enfrenta a ofrecer una solución.

La más grande contribución de Pólya en la enseñanza de las matemáticas es su método de cuatro pasos para resolver problemas.

Figura 7

Método de los cuatro pasos de Pólya.



Nota. Fuente: Polya (1989).

Fase 1: Comprender el Problema.

Para poder comprender un problema se debe principiar con la lectura, análisis y la recaudación de datos que están dados. No puede solucionarse un problema si no se comprende lo que piden resolver, no importa la cantidad de veces que se necesite leer el problema para lograr entenderlo y así responder las siguientes preguntas: ¿Cuál es la incógnita? ¿Cuáles son los datos?, ¿Es posible cumplir con las condiciones?, ¿Son suficientes las condiciones para hallar la incógnita?

Fase 2: Concebir un Plan.

Hay diferentes maneras de encontrarle solución a un problema, en esta fase se debe indicar la estrategia a utilizar para solucionar el problema. Existen ciertas estrategias como, por ejemplo: ensayo y error, resolver un problema más sencillo, buscar un patrón, hacer una lista, razonamiento indirecto, resolver una ecuación.

Fase 3: Ejecución del Plan.

Cuando se tiene comprendido el problema se debe realizar el plan y ejecutar la estrategia establecida durante la planificación del plan para finalmente poder solucionar el problema.

Fase 4: Examinar la solución obtenida.

Se debe examinar el resultado para comprobar que satisfaga las necesidades del problema y así comprobar el éxito de la solución de los problemas de ecuaciones.

Comúnmente los problemas se enuncian en palabras, ya sea oralmente o en forma escrita. Así, para resolver un problema, uno traslada las palabras a una forma equivalente del problema en la que usa símbolos matemáticos, resuelve esta forma equivalente y luego interpreta la respuesta.

Este método de plantear y resolver problemas matemáticos, apoyará a los estudiantes de primer ciclo que inician sus estudios en programación, en comprender mejor el problema a través de la resolución matemática del ejercicio y así diseñar correctamente el algoritmo.

2.3.11 Estrategias de Enseñanza en el diseño de algoritmos computacionales

Según Díaz Barriga y Hernández Rojas (2005) la enseñanza es como un proceso de ayuda que se va ajustando en función de cómo ocurre el progreso en la actividad constructiva de los alumnos. Es decir, la enseñanza es un proceso que pretende apoyar o “andamiar” el logro de aprendizajes significativos.

La enseñanza es también en gran medida una auténtica creación. y la tarea que le queda al docente por realizar es saber interpretarla y tomarla como objeto de reflexión para buscar mejoras sustanciales en el proceso completo de enseñanza-aprendizaje. De hecho, no podrá hacer una interpretación y lectura del proceso si no cuenta con un marco potente de reflexión, ni tampoco podrá engendrar propuestas sobre cómo mejorarlo si no cuenta con un arsenal apropiado de recursos que apoyen sus decisiones y su quehacer pedagógico.

Las estrategias de enseñanza son procedimientos que el agente de enseñanza utiliza en forma reflexiva y flexible para promover el logro de aprendizajes significativos en los alumnos y, enlazándolo con lo antes dicho, las estrategias de enseñanza son medios o recursos para prestar la ayuda pedagógica.

Además, es necesario tener presentes cinco aspectos esenciales para considerar qué tipo de estrategia es la indicada para utilizarse en ciertos momentos de la enseñanza, dentro de una sesión de aprendizaje:

1. Consideración de las características generales de los aprendices (nivel de desarrollo cognitivo, conocimientos previos, factores motivacionales, etcétera).
2. Tipo de dominio del conocimiento en general y del contenido curricular en particular, que se va a abordar.
3. La intencionalidad o meta que se desea lograr y las actividades cognitivas y pedagógicas que debe realizar el alumno para conseguirla.

4. Vigilancia constante del proceso de enseñanza, así como del progreso y aprendizaje de los alumnos.
5. Determinación del contexto intersubjetivo (por ejemplo, el conocimiento ya compartido) creado con los alumnos hasta ese momento, si es el caso.

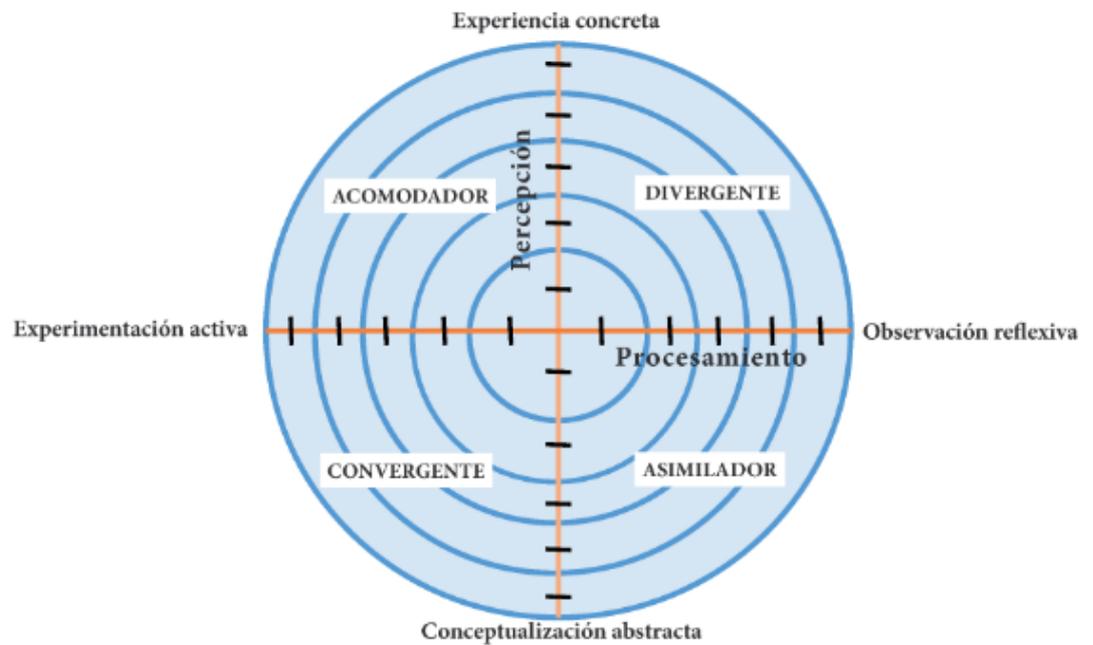
Estos factores, se han tomado en cuenta para plantear la estrategia de enseñanza “Inducción para Algoritmos” -INPAL, pues se ha considerado el desarrollo cognitivo y conocimientos previos en matemáticas coherente con el diseño de algoritmos computacionales, de los estudiantes de la asignatura de “Fundamentos de Programación”, buscando lograr un mejor entendimiento del problema a resolver, y haciendo un seguimiento del progreso de los estudiantes durante el periodo de tiempo que duró la presente investigación; lo que permitió también hacer ajustes de la ayuda pedagógica.

2.3.12 Estilos de aprendizaje en estudiantes de la carrera de ingeniería de sistemas en la educación superior universitaria

Según Kolb y Kolb (2005), el concepto de estilo de aprendizaje describe las diferencias individuales en el aprendizaje basadas en la preferencia del estudiante por emplear diferentes fases del ciclo de aprendizaje (experiencia concreta - CE y conceptualización abstracta - AC, observación reflexiva - RO y experimentación activa - AE). Debido a nuestra carga hereditaria, a nuestras experiencias de vida propia y a las exigencias de nuestro entorno actual, desarrollamos una forma preferida de aprendizaje, es decir resolvemos entre ser concreto o abstracto y entre ser activo o reflexivo.

Figura 8

Modelo de Kolb (1984)



Nota. Fuente. (González Calixto et al., 2020, p19)

Estilo Divergente: Un individuo con estilo divergente tiene CE y RO como habilidades de aprendizaje dominantes; las personas con este estilo de aprendizaje son mejores para ver situaciones concretas desde puntos de vista diferentes, se desempeñan mejor en situaciones que requieren la generación de ideas y les gusta recopilar información. Están interesados en las personas, tienden a ser imaginativos y emocionales, tienen amplios intereses culturales y tienden a especializarse en las artes. En situaciones de aprendizaje formal, prefieren trabajar en grupos, escuchar con la mente abierta y recibir retroalimentación personalizada.

Estilo Asimilador: Un individuo con un estilo de asimilación tiene AC y RO como habilidades de aprendizaje dominantes; las personas con este estilo de aprendizaje son mejores para comprender una amplia gama de información y

expresarla de forma lógica y concisa, están menos centrados en las personas y más interesados en ideas y conceptos abstractos, encuentran más importante que una teoría tenga solidez lógica que valor práctico, estas habilidades de aprendizaje tienen eficacia en las carreras de información y ciencias. En situaciones de aprendizaje formal, prefieren lecturas, conferencias, explorar modelos analíticos y tener tiempo para pensar las cosas.

Estilo Convergente: Un individuo con un estilo convergente tiene AC y AE como habilidades de aprendizaje dominantes; las personas con este estilo de aprendizaje son mejores para encontrar usos prácticos para ideas y teorías, tienen la capacidad de resolver problemas y tomar decisiones basadas en encontrar soluciones, prefieren abordar tareas y problemas técnicos en lugar de cuestiones sociales e interpersonales, estas habilidades de aprendizaje tienen eficacia en carreras especializadas y tecnológicas. En situaciones de aprendizaje formal, prefieren experimentar con nuevas ideas, simulaciones, tareas de laboratorio y aplicaciones prácticas.

Estilo Acomodador: Un individuo con un estilo acomodaticio tiene CE y AE como habilidades de aprendizaje dominantes; las personas con este estilo de aprendizaje tienen la capacidad de aprender principalmente a partir de experiencias “prácticas”, disfrutan llevando a cabo planes e involucrándose en experiencias nuevas y desafiantes, puede actuar basándose en sentimientos “instintivos” más que en un análisis lógico. Al resolver problemas, dependen más de las personas para obtener información que de su propio análisis técnico, este estilo de aprendizaje tiene eficacia en carreras orientadas a la acción, como marketing o ventas. En situaciones de aprendizaje formal,

prefieren trabajar con otros para realizar tareas, establecer objetivos, hacer trabajo de campo y probar diferentes enfoques para completar un proyecto.

Teniendo en cuenta estos estilos de aprendizaje, podemos identificar que los estudiantes de ingeniería de sistemas, tienen un estilo de aprendizaje convergente, pues las habilidades que predominan en el aprendizaje de algoritmos y programación es la conceptualización abstracta (capaz de crear nuevos conceptos) y la experimentación activa (emplear estos conceptos para tomar decisiones y solucionar problemas), afirmación que concuerda con Siqueira (como se citó en González Calixto et al., 2020) que manifiesta que este estilo puede estar presente en tecnólogos, ingenieros y físicos. En tal sentido este estilo de aprendizaje se tuvo en cuenta para el diseño de la estrategia de enseñanza INPAL.

2.3.13 Fundamentos teóricos - metodológicos de la estrategia de enseñanza INPAL para el diseño de algoritmos.

1. Fundamentación:

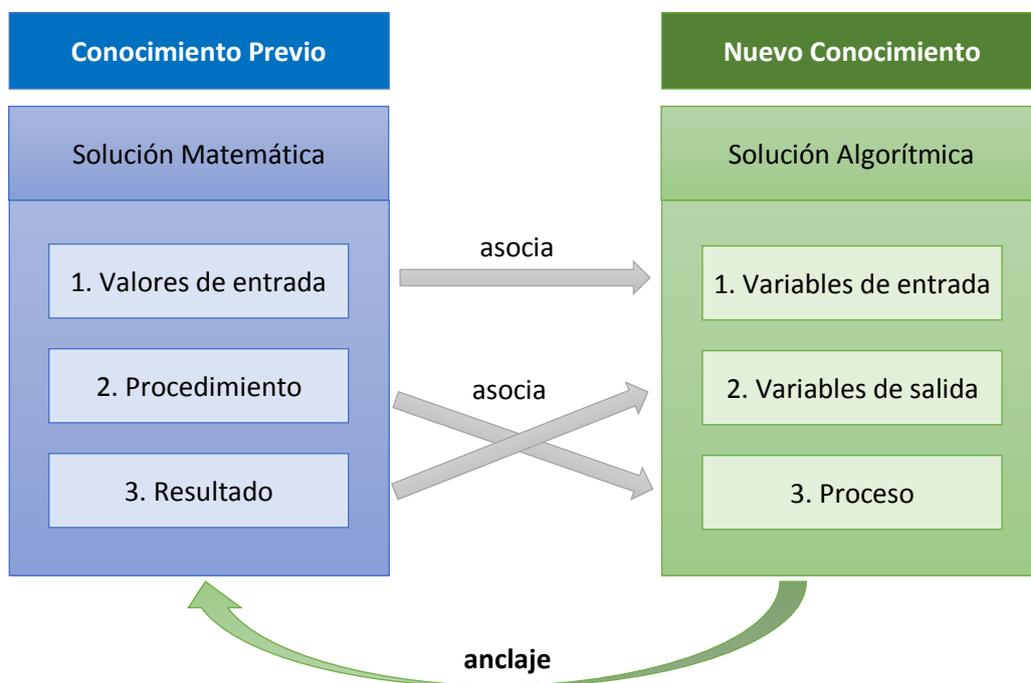
La estrategia de enseñanza INPAL, diseñada por la autora de la presente investigación e implementada a través de sesiones de aprendizaje, se sustenta en las teorías del aprendizaje significativo de Ausubel, en la teoría del aprendizaje de Gagné, en la teoría de las situaciones didácticas de Brousseau, en el razonamiento inductivo y en las bases teóricas de estrategia de enseñanza y de diseño de algoritmos.

Teoría del aprendizaje significativo de Ausubel, aporta a la estrategia de enseñanza INPAL estableciendo el anclaje del tema de diseño de algoritmos, con las ideas específicas y pertinentes asociadas con la resolución de problemas matemáticos existentes en la estructura cognitiva del estudiante, generando una interacción que toman significado en el estudiante a partir de un conjunto de conceptos y definiciones como resultado de la instrucción realizado por el docente antes, durante y al término de la interacción didáctica en el contexto del aula.

Esta teoría permite la comprensión problémica del estudiante, dando apertura para implementar un proceso basado en: plantear una correcta solución algorítmica, leer el enunciado del problema, basándose en un caso específico con valores, es decir resolverá el problema haciendo uso de sus conocimientos previos de matemáticas, para luego asociar estos valores y resultados con variables de entrada y de salida y el procedimiento matemático con el proceso de un algoritmo y así entender mejor el funcionamiento del algoritmo y asegurando así un correcto diseño, tal como se observa en el diagrama adjunto.

Figura 9

Anclaje del nuevo conocimiento algorítmico con el conocimiento previo matemático.



Nota. Elaborado en base a la teoría del aprendizaje significativo y a los elementos de la estrategia INPAL.

Teoría del aprendizaje de Gagné, aporta a la presente propuesta con los principios de contigüidad, repetición y refuerzo y con las condiciones de aprendizaje externos (docente, aula, estrategia) e internos (memoria, estado de ánimo del estudiante), los cuales han sido considerados en el diseño de las sesiones de aprendizaje de la asignatura de “Fundamentos de Programación”.

El principio de contigüidad, se aplicó al presentar un problema y resolverlo siguiendo las fases de resolución de problemas a través de una computadora, hasta llegar al diseño del algoritmo, es decir se inició con la solución matemática del problema, se hizo la asociación con los

elementos del algoritmo y finalmente se diseñó el algoritmo, existiendo contigüidad en cada una de estas fases.

El principio de repetición, se aplicó cuando el estudiante desarrollaba nuevos problemas siguientes las fases mencionadas, estos nuevos problemas se resolvían en clase o se los dejaba como tarea o se los planteaba en alguna evaluación.

El principio de refuerzo, se aplicó cuando el estudiante diseñaba correctamente su algoritmo, el cual arrojaba los resultados correctos después de ejecutar su programa elaborado en base al algoritmo diseñado.

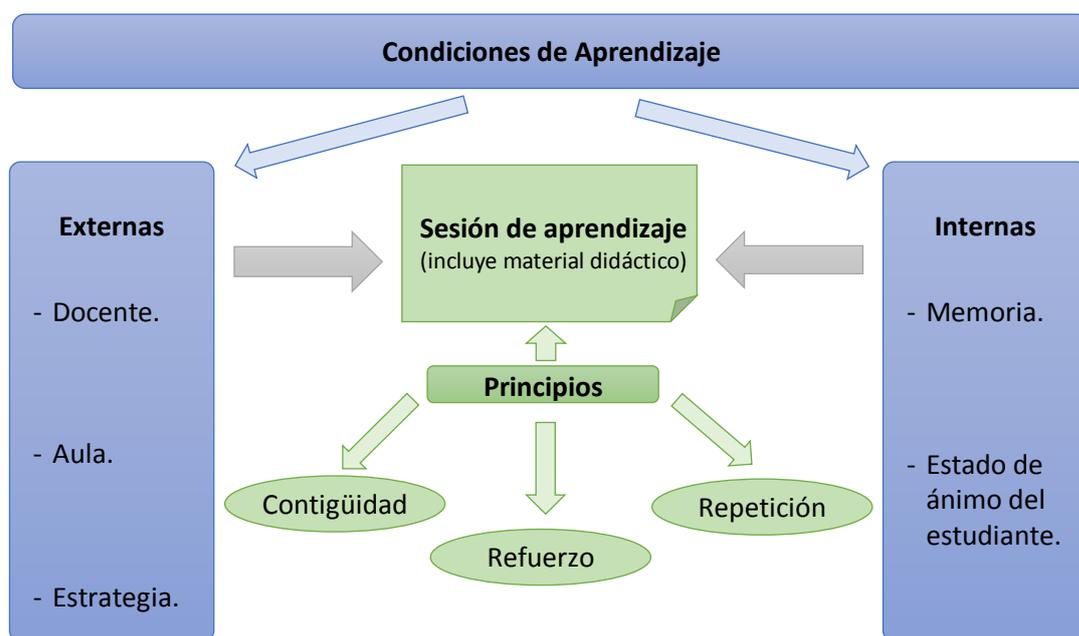
Además, se tomaron en cuenta las condiciones de aprendizaje externas; es decir, se contó con un docente que cumplía el perfil para el dictado de la asignatura, se contó con un aula virtual y se aplicó la estrategia de enseñanza INPAL; y algunas condiciones internas como la memoria para traer de la memoria de los estudiantes conocimientos previos sobre la resolución matemática de problemas y todo ello con el fin de apoyar los procesos internos de aprendizaje y lograr un aprendizaje efectivo.

Concretizar este proceso basado en los principios descritos, demanda, necesariamente, diseñar sesiones de aprendizaje con el propósito de lograr un aprendizaje eficaz en el estudiante; es decir se deben elegir eventos externos planificados para apoyar los procesos internos de aprendizaje. Además, permite tener en consideración un perfil mínimo para la actuación del docente, los estilos y ritmos de

aprendizaje de los estudiantes, los estados de ánimo del estudiante, el entorno y estructura física que envuelve el proceso áulico para concretizar el proceso formativo. Para mayor comprensión se detalla simbólicamente el proceso, según la modelización en el diagrama adjunto.

Figura 10

Condiciones y principios en las sesiones de aprendizaje.



Nota. Elaborado en base a la teoría del aprendizaje de Robert Gagné.

Teoría de las situaciones didácticas de Brousseau, orienta a consolidar la planeación didáctica, considerando las fases de una situación didáctica: situación de acción, situación de formulación y situación de validación para el diseño de la sesión de aprendizaje y la aplicación de la estrategia de enseñanza INPAL en el diseño de algoritmos, teniendo como propósito una mejor comprensión y análisis del problema, consolidando un mejor diseño de algoritmos

computacionales. Por otro lado, tal teoría, permite comprender en el docente y el estudiante que resolver un algoritmo en el curso de programación debe considerarse algunos procesos básicos desde una lógica, coherencia interna, independientemente de cualquier procedimiento que se quiera implementar. A continuación, se describe la secuencia a tener en cuenta en cada fase.

Situación de acción: se plantea un problema de algún contexto de interés para el estudiante, luego cada estudiante busca alguna solución matemática y a su vez trata de asociarlo con los elementos de un algoritmo, eligiendo entre uno u otro procedimiento para dar solución al problema, que permita un correcto análisis del problema, para finalmente diseñar un algoritmo computacional.

Situación de formulación: los estudiantes comparten en pares (coevaluación), la solución matemática, la asociación, el análisis y el diseño del algoritmo realizado, con el fin de que el segundo estudiante la comprenda y lo aplique para determinar su eficacia, del mismo modo el segundo estudiante comparte su solución encontrada con el primer estudiante con el mismo fin, para finalmente quedarse con la solución más eficaz.

Situación de validación: cada par de estudiantes presentan y justifican la solución encontrada al problema, en algunos casos llamando al docente para sustentar su procedimiento de solución o en la pizarra frente a sus compañeros de clase, con la finalidad de validar su solución propuesta y de ser posible optimizarla.

La aplicación de los diferentes procedimientos que tributa la teoría de las situaciones didácticas para resolver situaciones de aprendizaje en cualquier campo del saber, requiere que el docente internalice cada una de las categorías contenidas en el esquema que se detalla, línea abajo.

Figura 11

Situación didáctica en el aprendizaje del diseño de algoritmos.



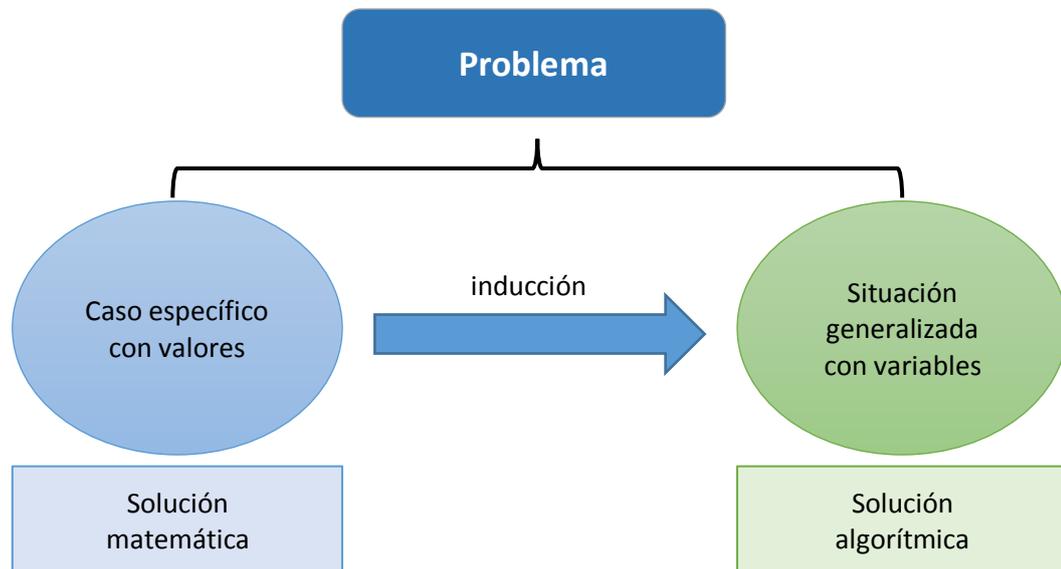
Nota. Elaborado en base a la teoría de las situaciones didácticas y a las actividades de la sesión de aprendizaje del diseño de algoritmos.

Razonamiento inductivo, aporta con el establecimiento de las fases y pasos de la estrategia de enseñanza INPAL, es decir permite diseñar un algoritmo para un problema planteado, partiendo de la resolución de un problema matemático específico, para luego generalizar

la solución en un algoritmo, realizando una asociación de valores y procedimiento de la solución matemática con las variables y proceso del algoritmo computacional, y así garantizar el correcto análisis del problema y con ello el diseño del algoritmo.

Figura 12

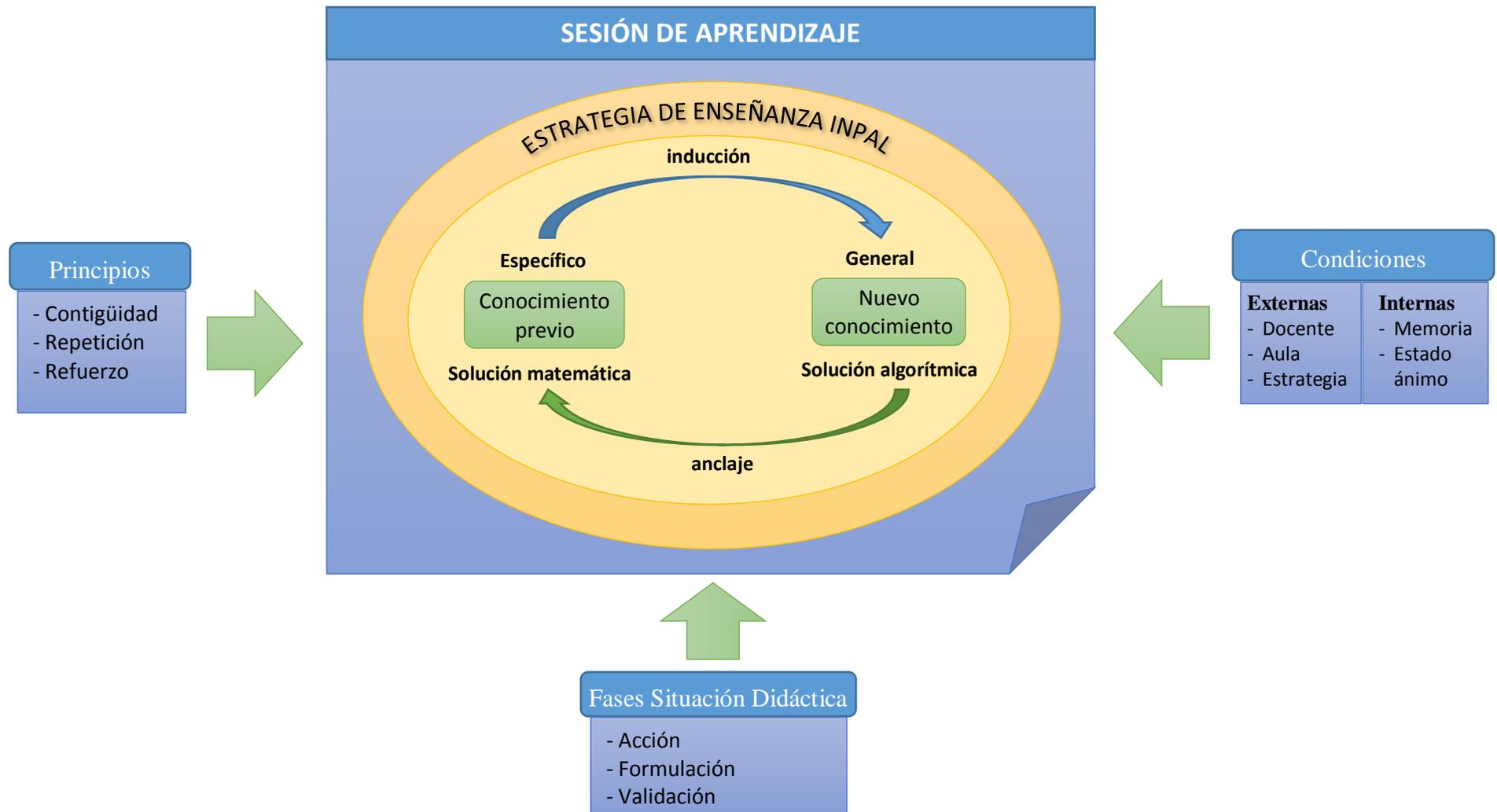
Inducción a una solución algorítmica, a partir de una solución matemática.



Nota. Elaborado en base a la teoría de las situaciones didácticas y a las actividades de la sesión de aprendizaje del diseño de algoritmos.

Figura 13

Esquema general de la fundamentación teórica de la estrategia de enseñanza INPAL



Nota. Elaborado en base a las teorías educativas citadas que contribuyeron con las sesiones de aprendizaje, las que a su vez incluyen la estrategia de enseñanza INPAL.

2. Finalidad:

El programa de la estrategia de enseñanza INPAL, tiene por finalidad mejorar el nivel del diseño de algoritmos en los estudiantes de la asignatura de “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas.

3. Objetivos

- **Objetivo general**

- Diseñar algoritmos con estructura básicas de control, considerando los elementos básicos de un algoritmo y aplicando la estrategia de enseñanza INPAL, para su posterior implementación en un lenguaje de programación.

- **Objetivos específicos**

- Anclar el nuevo conocimiento de diseño de algoritmos con la resolución matemática del problema, para lograr un aprendizaje significativo.
- Aplicar los principios y condiciones del aprendizaje en las sesiones de aprendizaje.
- Implementar las fases de una situación didáctica en las sesiones de aprendizaje.

4. Contenidos

Los contenidos a desarrollar en la presente propuesta, en los que se pueda aplicar la estrategia de enseñanza INPAL, son aquellos que se pueden apoyar en una solución matemática en particular, para luego generalizar la solución en un algoritmo computación, y por tanto abarca las estructuras de control básicas de algoritmos y programación.

- Algoritmos con estructura secuencial.
- Algoritmos con estructura condicional.
- Algoritmos con estructura repetitiva.

Cabe mencionar que la propuesta de la presente investigación, permitió aplicar la estrategia de enseñanza INPAL en las estructuras básicas de control secuencial y condicional.

5. Diseño de la estrategia de enseñanza INPAL

El procedimiento propuesto para la estrategia de enseñanza INPAL, se fundamenta en las teorías del aprendizaje significativo, en el razonamiento inductivo y en la base teórica de estrategias de enseñanza de Díaz y Hernández.

Comprende 2 fases:

Resolución matemática del problema

1. Identifica los valores numéricos del problema.
2. Identifica los valores de entrada.
3. Identifica los valores de salida.
4. Establece el proceso o fórmula que produce la salida deseada.
5. Aplica el proceso o fórmula, en base a los valores de entrada, para obtener los valores de salida.

Asociación de solución matemática con los elementos del algoritmo

6. Relaciona los valores numéricos con las variables del problema.
7. Relaciona los valores de entrada con las variables de entrada.
8. Relaciona los valores de salida con las variables de salida.
9. Relaciona el proceso matemático con el proceso algorítmico que produce la salida deseada.

Y en base a estas fases previas de la estrategia de enseñanza INPAL, se procede con las fases de resolución de problemas usando la computadora, según la base teórica de Joyanes Aguilar, las cuales son:

Análisis del Problema

1. Identifica las variables del problema.
2. Identifica las variables de entrada.
3. Identifica las variables de salida.
4. Establece el proceso que produce la salida deseada.

Diseño de la Solución Algorítmica

5. Establece la declaración de variables.
6. Establece la entrada de datos.
7. Establece el proceso a seguir.
8. Establece la salida de resultados.

Tabla 1*Estrategia de enseñanza INPAL.*

COMPONENTES Y FASES DE LA ESTRATEGIA DE ENSEÑANZA INPAL		
Componente: SOLUCIÓN MATEMÁTICA		
Fase 1:	Resolución matemática del problema	<ol style="list-style-type: none"> 1. Identifica los valores numéricos del problema. 2. Identifica los valores de entrada. 3. Identifica los valores de salida. 4. Establece el proceso o fórmula que produce la salida deseada. 5. Aplica el proceso o fórmula, en base a los valores de entrada, para obtener los valores de salida.
Fase 2:	Asociación de solución matemática con los elementos del algoritmo	<ol style="list-style-type: none"> 6. Relaciona los valores numéricos con las variables del problema. 7. Relaciona los valores de entrada con las variables de entrada. 8. Relaciona los valores de salida con las variables de salida. 9. Relaciona el proceso matemático con el proceso algorítmico que produce la salida deseada.
Componente: DISEÑO DEL ALGORITMO		
Fase 3:	Análisis del Problema	<ol style="list-style-type: none"> 10. Identifica las variables del problema. 11. Identifica las variables de entrada. 12. Identifica las variables de salida. 13. Establece el proceso que produce la salida deseada.
Fase 4:	Diseño de la Solución Algorítmica	<ol style="list-style-type: none"> 14. Establece la declaración de variables. 15. Establece la entrada de datos. 16. Establece el proceso a seguir. 17. Establece la salida de resultados.

Nota. El componente solución matemática a ha sido creado en la presente investigación y agregado al componente de Joyanes Aguilar (2008).

La aplicación práctica de la presente estrategia, a través de ejemplos con problemas específicos, se puede apreciar en el apéndice 6.

6. Marco lógico de los resultados de aprendizaje

Resultado de aprendizaje de la asignatura:

Al concluir la asignatura el estudiante construye algoritmos y programas computacionales básicos en consola empleando las estructuras de control secuencial, condicional y repetitiva; y considerando el análisis del problema, el diseño de la solución, la prueba de escritorio y la implementación en un lenguaje de programación. (Currículo de Estudios de Ingeniería de Sistemas, 2018)

Cabe mencionar que la presente propuesta contribuirá, principalmente con el diseño de algoritmos computacionales; y a partir de estos, se construirán los programas computacionales básicos en consola.

Tabla 2

Marco Lógico de los resultados de aprendizaje.

Unidad/Sesión	Resultados de aprendizaje	Evidencias
I. Algoritmos - Estructura Secuencial.	El estudiante diseña algoritmos con estructura secuencial, considerando los elementos básicos de un algoritmo y aplicando la estrategia de enseñanza INPAL.	Diseño de algoritmos con estructura secuencial, aplicando la estrategia de enseñanza INPAL.
Sesión de aprendizaje 1:	Comprende cómo diseñar un algoritmo con estructura secuencial.	Informe explicativo del diseño de un algoritmo con estructura secuencial.
Sesión de aprendizaje 2:	Diseña algoritmos con estructura secuencial.	Informe de laboratorio con algoritmos diseñados en pares de estudiantes.
Sesión de aprendizaje 3:	Comprende cómo diseñar un algoritmo con estructura secuencial de mayor complejidad.	Informe explicativo del diseño de un algoritmo con estructura secuencial de mayor complejidad.
Sesión de aprendizaje 4:	Diseña algoritmos con estructura secuencial de mayor complejidad.	Informe de laboratorio con algoritmos diseñados en pares de estudiantes.
Sesión de aprendizaje 5:	Practica calificada 1 y retroalimentación.	Desarrollo de la práctica calificada por parte de los estudiantes y explicación de la resolución del mismo.

II. Algoritmos - Estructura Condicional.	El estudiante diseña algoritmos con estructura condicional, considerando los elementos básicos de un algoritmo y aplicando la estrategia de enseñanza INPAL.	Diseño de algoritmos con estructura condicional, aplicando la estrategia de enseñanza INPAL.
Sesión de aprendizaje 6:	Comprende cómo diseñar un algoritmo con estructura condicional simple y doble.	Informe explicativo del diseño de un algoritmo con estructura condicional simple y doble.
Sesión de aprendizaje 7:	Diseña algoritmos con estructura condicional simple y doble.	Informe de laboratorio con algoritmos diseñados en pares de estudiantes.
Sesión de aprendizaje 8:	Comprende cómo diseñar un algoritmo con estructura condicional anidada.	Informe explicativo del diseño de un algoritmo con estructura condicional anidada.
Sesión de aprendizaje 9:	Diseña algoritmos con estructura condicional anidada.	Informe de laboratorio con algoritmos diseñados en pares de estudiantes.
Sesión de aprendizaje 10:	Práctica calificada 2 y retroalimentación.	Desarrollo de la práctica calificada por parte de los estudiantes y explicación de la resolución del mismo.

Nota. Se organizaron las sesiones de aprendizaje en unidades para la implementación de la propuesta con la estrategia de enseñanza INPAL.

7. Perfil del docente

- Aplica estrategias didácticas teniendo en cuenta las teorías del aprendizaje que fortalecen el desarrollo de procesos cognitivos en estudiantes universitarios.
- Diseño de algoritmos y construcción de programas computacionales en consola, nivel intermedio.

8. Perfil del estudiante

- Resuelve problemas matemáticos básicos.
- Razonamiento lógico - inductivo.

9. Cronograma de acciones

Cabe indicar que se recomienda iniciar con los fundamentos teóricos de computación y algoritmos, lo que tomaría unas 2 semanas aproximadamente.

Se ha planteado 10 sesiones de aprendizaje de 3 horas pedagógicas de 45 minutos cada una.

Tabla 3

Actividades por sesiones de aprendizaje.

Sesión de Aprendizaje	Actividad	Fecha
Semana 1		
Sesión 1: Estructura Secuencial Teoría.	<ul style="list-style-type: none"> - Introducción del tema, a través de la explicación de la estructura general de un algoritmo de la vida diaria, con estructura secuencial. - Explicación de la estructura general de un algoritmo con estructura secuencial, mediante una presentación de diapositivas, por parte del docente. - Desarrollo de un primer ejercicio con estructura secuencial, mediante las fases del diseño de algoritmos, aplicando la estrategia de enseñanza INPAL. - Desarrollo de un ejercicio con estructura secuencial, por parte de los estudiantes, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Presentación y explicación de un ejercicio desarrollado, por parte de un estudiante, el cual permitirá participar con consultas o aportes por parte del resto de estudiantes de la clase, y así dar las conclusiones finales por parte del docente. 	
Sesión 2: Estructura Secuencial Práctica.	<ul style="list-style-type: none"> - Participación de los estudiantes sobre lo aprendido en la sesión anterior, para lo cual responde a las siguientes preguntas: ¿En qué consisten los algoritmos con estructura secuencial? ¿Mencione algunos ejemplos de problemas que se pueden resolver con estructura secuencial? - Repaso de la estructura general de un algoritmo con estructura secuencial, mediante el desarrollo de un ejercicio con estructura secuencial, aplicando la estrategia de enseñanza INPAL. - Presentación de una práctica 01 - nivel 1, con ejercicios propuestos sobre algoritmos con estructura secuencial. - Conformación de equipos en pares de estudiantes, para resolver 2 ejercicios de la práctica, elegidos por el par de estudiantes. - Desarrollo de los 2 ejercicios, con estructura secuencial, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Explicación de los algoritmos planteados por cada estudiante, a su compañero de equipo, con la finalidad de compartir la solución algorítmica encontrada por cada estudiante y así determinar su eficacia. - Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de dar alternativas que permitan optimizar las soluciones encontradas, con el apoyo del docente. 	
Semana 2		
Sesión 3: Estructura Secuencial, nivel 2 - Teoría.	<ul style="list-style-type: none"> - Lluvia de ideas de los estudiantes, de cómo solucionarían un nuevo ejercicio con estructura secuencial de mayor complejidad, presentado por el docente. - Explicación de nuevos operadores y uso de variables intermedias en algoritmos con estructura secuencial, 	

	<p>mediante una presentación de diapositivas, por parte del docente.</p> <ul style="list-style-type: none"> - Desarrollo de un ejercicio con mayor complejidad, con estructura secuencial, mediante las fases del diseño de algoritmos, aplicando la estrategia de enseñanza INPAL. - Desarrollo de un ejercicio con mayor complejidad, con estructura secuencial, por parte de los estudiantes, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Presentación y explicación de un ejercicio desarrollado, por parte de un estudiante, el cual permitirá participar con consultas o aportes por parte del resto de estudiantes de la clase, y así dar las conclusiones finales por parte del docente. 	
<p>Sesión 4: Estructura Secuencial, nivel 2 - Práctica.</p>	<ul style="list-style-type: none"> - Participación de los estudiantes sobre lo aprendido en la sesión anterior, para lo cual responde a las siguientes preguntas: ¿Qué nuevos operadores pueden emplear en los algoritmos con estructura secuencial? ¿Mencione algunos ejemplos de problemas de mayor complejidad que se pueden resolver con estructura secuencial? - Reforzamiento de la estructura general de un algoritmo con estructura secuencial de mayor complejidad, mediante el desarrollo de un ejercicio con estructura secuencial, aplicando la estrategia de enseñanza INPAL. - Presentación de una práctica 01 - nivel 2, con ejercicios propuestos sobre algoritmos con estructura secuencial. - Conformación de equipos en pares de estudiantes, para resolver 2 ejercicios de la práctica, elegidos por el par de estudiantes. - Desarrollo de los 2 ejercicios, con estructura secuencial, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Explicación de los algoritmos planteados por cada estudiante, a su compañero de equipo, con la finalidad de compartir la solución algorítmica encontrado de cada estudiante y determinar su eficacia. - Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de dar alternativas que permitan optimizar las soluciones encontradas, con el apoyo del docente. 	
Semana 3		
<p>Sesión 5. Práctica calificada 1 y retroalimentación.</p>	<ul style="list-style-type: none"> - Explicación del docente sobre las indicaciones de la práctica calificada que desarrollará cada estudiante. - Desarrollo individual de los ejercicios propuestos en la práctica calificada, por parte de los estudiantes. - Conformación de equipos en pares de estudiantes, para compartir la solución de la práctica calificada, con el fin de completar o corregir el algoritmo diseñado. - Apoyo docente a través de la atención a dudas o inquietudes de los estudiantes. - Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de presentar una de las soluciones más óptimas de la práctica calificada, con el apoyo del docente. 	
<p>Sesión 6. Estructura Condicional Simple y Doble - Teoría.</p>	<ul style="list-style-type: none"> - Introducción del tema, a través de la explicación de la estructura general de un algoritmo de la vida diaria, con estructura condicional. - Explicación de la estructura general de un algoritmo con estructura condicional simple y doble, mediante una presentación de diapositivas, por parte del docente. 	

		<ul style="list-style-type: none"> - Desarrollo de un primer ejercicio con estructura condicional, mediante las fases del diseño de algoritmos, aplicando la estrategia de enseñanza INPAL. - Desarrollo de un ejercicio con estructura condicional, por parte de los estudiantes, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Presentación y explicación de un ejercicio desarrollado, por parte de un estudiante, el cual permitirá participar con consultas o aportes por parte del resto de estudiantes de la clase, y así dar las conclusiones finales por parte del docente. 	
Semana 4			
Sesión 7. Estructura Condicional Simple y Doble - Práctica.		<ul style="list-style-type: none"> - Participación de los estudiantes sobre lo aprendido en la sesión anterior, para lo cual responde a las siguientes preguntas: ¿En qué consisten los algoritmos con estructura condicional? ¿Mencione algunos ejemplos de problemas que se pueden resolver con estructura condicional simple y doble? - Repaso de la estructura general de un algoritmo con estructura condicional simple y doble, mediante el desarrollo de un ejercicio con estructura condicional, aplicando la estrategia de enseñanza INPAL. - Presentación de una práctica 02, con ejercicios propuestos sobre algoritmos con estructura condicional simple y doble. - Conformación de equipos en pares de estudiantes, para resolver 2 ejercicios de la práctica, elegidos por el par de estudiantes. - Desarrollo de los 2 ejercicios, con estructura condicional, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Explicación de los algoritmos planteados por cada estudiante, a su compañero de equipo, con la finalidad de compartir la solución algorítmica encontrada por cada estudiante y así determinar su eficacia. - Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de dar alternativas que permitan optimizar las soluciones encontradas, con el apoyo del docente. 	
Sesión 8. Estructura Condicional Anidada - Teoría.		<ul style="list-style-type: none"> - Introducción del tema, a través de la explicación de la estructura general de un algoritmo de la vida diaria, con estructura condicional anidada. - Desarrollo de un ejercicio con estructura condicional anidada, mediante las fases del diseño de algoritmos, aplicando la estrategia de enseñanza INPAL. - Desarrollo de un ejercicio, con estructura condicional anidada, por parte de los estudiantes, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Presentación y explicación de un ejercicio desarrollado, por parte de un estudiante, el cual permitirá participar con consultas o aportes por parte del resto de estudiantes de la clase, y así dar las conclusiones finales por parte del docente. 	
Semana 5			
Sesión 9. Estructura Condicional Anidada - Práctica.		<ul style="list-style-type: none"> - Participación de los estudiantes sobre lo aprendido en la sesión anterior, para lo cual responde a las siguientes preguntas: ¿En qué consisten los algoritmos con estructura condicional anidada? 	

	<p>¿Mencione algunos ejemplos de problemas que se pueden resolver con estructura condicional anidada?</p> <ul style="list-style-type: none"> - Presentación de una práctica 02, con ejercicios propuestos sobre algoritmos con estructura condicional anidada. - Conformación de equipos en pares de estudiantes, para resolver 2 ejercicios de la práctica, elegidos por el par de estudiantes. - Desarrollo de los 2 ejercicios, con estructura condicional anidada, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Explicación de los algoritmos planteados por cada estudiante, a su compañero de equipo, con la finalidad de compartir la solución algorítmica encontrado de cada estudiante y determinar su eficacia. - Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de dar alternativas que permitan optimizar las soluciones encontradas, con el apoyo del docente. 	
Sesión 10. Evaluación 2 y retroalimentación	<ul style="list-style-type: none"> - Explicación del docente sobre las indicaciones de la práctica calificada que desarrollará cada estudiante. - Desarrollo individual de los ejercicios propuestos en la práctica calificada, por parte de los estudiantes. - Conformación de equipos en pares de estudiantes, para compartir la solución de la práctica calificada, con el fin de completar o corregir el algoritmo diseñado. - Apoyo docente a través de la atención a dudas o inquietudes de los estudiantes. - Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de presentar una de las soluciones más óptimas de la práctica calificada, con el apoyo del docente. 	

Nota. Se organizaron las acciones de las sesiones de aprendizaje por semana para la implementación de la propuesta con la estrategia de enseñanza INPAL.

Además, se desarrollaron las sesiones de aprendizaje y el material didáctico para el tema de diseño de algoritmos con estructura secuencial y estructura condicional, aplicando la estrategia de enseñanza INPAL, material académico que fue utilizado con el grupo experimental de la investigación. (ver apéndice 5 y 6)

10. Medios y Materiales

- Laboratorio de cómputo.
- Proyector multimedia.

- Laptop, pizarra, plumones, mota para el docente.
- Cuaderno de apuntes, lápiz/lapiceros para los estudiantes.

Cabe indicar que la presente propuesta también se puede aplicar en una modalidad de enseñanza remota; por tanto, se emplearan entornos y herramientas educativas digitales.

11. Reglas de Clase

- Puntualidad en la asistencia a clases, con una tolerancia no mayor a 5 minutos.
- Propiciar un ambiente de amabilidad, basado en el respeto y la buena comunicación.
- Los estudiantes que deseen participar deben solicitar el uso de la palabra al docente, quien le concederá indicando el momento y tiempo adecuado.
- El docente también puede solicitar la intervención de uno o más estudiantes.
- Trabajo en pares de estudiantes, para compartir sus soluciones algorítmicas y conocer la efectividad de cada una de ellas.

12. Evaluación

La evaluación es formativa en cada una de las sesiones de clase, considerándose su participación en clase y el cumplimiento con el desarrollo de tareas, además de una evaluación escrita individual al finalizar cada unidad, similar a la evaluación escrita del postest, donde la pregunta 1 y 2 corresponde a la estructura secuencial (Unidad I) y la pregunta 3 y 4, a la estructura condicional (Unidad II), como se puede observar en el apéndice 7.

13. Procedimiento para la implementación de la estrategia

- a. El docente y el estudiante deben tener el perfil mencionado en la presente propuesta.
- b. Revisar la competencia y los resultados de aprendizaje.
- c. Revisar los contenidos a desarrollar.
- d. Revisar el diseño de la estrategia de enseñanza INPAL, incluyendo el material didáctico; pudiendo complementar con más ejercicios afines al contexto actual o algún tema de interés
- e. Aplicar la estrategia de enseñanza INPAL.
- f. Evaluar a los estudiantes.

2.4 Definición de términos básicos

Algoritmo: “procedimiento para resolver un problema en términos de las acciones a ejecutar y el orden en el que se ejecutan estas acciones”. (Deitel y Deitel, 2008, p. 113)

Análisis del Problema: “primera fase de la resolución de un problema, esta fase requiere una clara definición, donde se contemple exactamente lo que debe hacer el programa y el resultado o solución deseada. Dado que se busca una solución por computadora, se precisan especificaciones detalladas de entrada y salida y el proceso que produce la salida deseada”. (Joyanes Aguilar, 2008, p. 47)

Diseño del algoritmo: “determina cómo hace el programa la tarea solicitada (...) proceso que convierte los resultados del análisis del problema en un diseño modular con refinamientos sucesivos que permitan una posterior traducción a un lenguaje. El diseño del algoritmo es independiente del lenguaje de programación en el que se vaya a codificar posteriormente”. (Joyanes Aguilar, 2008, p. 48)

Codificación de un programa: “escritura en un lenguaje de programación de la representación del algoritmo desarrollado (...). Para realizar la conversión del algoritmo en programa se deben sustituir las palabras reservadas en español por sus homónimos en inglés, y las operaciones/instrucciones indicadas en lenguaje natural por el lenguaje de programación correspondiente.” (Joyanes Aguilar, 2008, p. 51)

Estrategia de enseñanza: “procedimientos que el agente de enseñanza utiliza en forma reflexiva y flexible para promover el logro de aprendizajes significativos en los alumnos”. (Díaz Barriga y Hernández Rojas, 2005, p. 141)

Inducción: en la inducción el razonamiento es, lógicamente, ascendente desde lo particular y concreto a lo general y abstracto. Se suele afirmar que la inferencia inductiva es un razonamiento orientado hacia el fin. la inducción no es más que una consecuencia lógica y metodológica de la utilización del método comparativo. Más exactamente, es el procedimiento mediante el cual se diferencian las causas originarias de las restantes causas intervinientes, a partir de la comparación entre una amplia diversidad de realidades, para poder inferir la formulación de una ley o proposición científica de causalidad. (Calduch Cervera, 2014, pp. 33-34)

Estrategia “Inducción para algoritmos” - INPAL: procedimiento de enseñanza de algoritmos basada en la teoría del aprendizaje significativo y en el método inductivo, es decir consiste en la solución de un problema a partir de una solución matemática de un caso particular con valores numéricos (conocimientos previos), para luego generalizar la solución en el diseño de un algoritmo en donde se defina variables y procesos (nuevo conocimiento). (Propia basada en Díaz Barriga y Hernández, 2005 y en Ausubel, 2002)

CAPÍTULO III

MARCO METODOLÓGICO

3.1 Caracterización y contextualización de la investigación

3.1.1 Descripción del perfil de la institución educativa o red educativa.

La Universidad Nacional de Cajamarca, está dedicada a la formación integral de profesionales, gestores de conocimiento, a través de la investigación científica, tecnológica y humanística, comprometidos con los procesos sociales, económicos, ambientales y culturales con responsabilidad social.

Su sede central está ubicada en Av. Atahualpa N° 1050, en distrito de Cajamarca. Además, cuenta con 5 sub-sedes ubicadas en las provincias de Jaén, Chota, Celendín, Bambamarca y Cajabamba.

La Universidad Nacional de Cajamarca consta de 10 facultades que albergan 24 escuelas profesionales, atendidas por docentes idóneos y personal administrativo competente y dispone de una infraestructura amplia y adecuada, aulas recientemente remodeladas, modernos laboratorios y centros experimentales. Además, cabe mencionar que la Escuela Académico Profesional de Ingeniería de Sistemas pronto inaugurará un moderno Edificio, en el campus universitario, totalmente equipado.

3.1.2 Breve reseña histórica de la institución educativa o red educativa

La creación de la Universidad Nacional de Cajamarca, es el resultado de las aspiraciones populares y ciudadanas que ansiaban para Cajamarca una Universidad, fue la federación de Educadores de Cajamarca, que desde 1957 bajo la conducción del Dr. Zoilo León Ordoñez y de un grupo de preclaros maestros iniciaron el noble propósito de gestar un centro de estudios superiores para la juventud y el pueblo de esta milenaria tierra.

En 1961, dicha federación auspicia, un comité de Amplia Base Pro-Universidad, en el cual estuvieron debidamente acreditados los representantes de todas las instituciones más significativas de la provincia, quienes en forma unánime apoyaron la cívica iniciativa. Este organismo cumplió una serie de acciones, como el establecimiento de filiales en provincias vecinas: Celendín, Cajabamba, Contumazá y Bambamarca, la circulación de memoriales, la coordinación con los señores parlamentarios, y las notas estimulantes del periódico “época”, fueron determinantes para que se aprobara el viaje de una comisión a la Capital de la República.

Finalmente el 13 de febrero del año 1962 se promulga la Ley N° 14015, por la que se crea la Universidad Técnica de Cajamarca y el 14 de julio del mismo año inicia su funcionamiento, con una planificación de seis Escuelas, Medicina Rural, Agronomía, Medicina Veterinaria, Pedagogía, Minería y Metalurgia, Economía y Organización de Empresas; en nuestros días cuenta con otra estructura normativa, fundamentándose en la formación académica, investigación y proyección social; con diez facultades, Educación, Ingeniería, Ciencias Agrícolas y Forestales, Ciencias de la Salud , Ciencias Económicas, Contables y

Administrativas, Ciencias Sociales, Ciencias Veterinarias, Zootecnia, Medicina Humana y Derecho y Ciencias Políticas.

Desde el 2016 hasta junio del 2021, ejerció el cargo de Rector el Dr. Angelmiro Montoya Mestanza, luego asumió el cargo el Dr. Angel Francisco Dávila Rojas y desde finales del 2021, hasta la fecha desempeña el cargo de Rector el Dr. Bernardo Escalante Zumaeta.

3.1.3 Características demográficas y socioeconómicas

La Universidad Nacional de Cajamarca cuenta con una población estudiantil aproximada de 9709 estudiantes, los cuales provienen en su mayoría de la Región Cajamarca y un menor porcentaje de otras ciudades del país. En la Escuela Académico Profesional de Ingeniería de Sistemas, en el presente ciclo 2019-II, se cuenta con 353 estudiantes.

Los estudiantes de nuestra Universidad en su mayoría son de un nivel socioeconómico medio, y algunos de un nivel socioeconómico bajo. En tal sentido, la Oficina de Bienestar Universitario identifica a los estudiantes, que requieren apoyo con prestaciones asistenciales como comedor universitario, subvenciones, exoneraciones y otros.

Los estudiantes de primer ciclo de la carrera de Ingeniería de Sistemas, año 2020, proceden de la ciudad de Cajamarca o de alguna provincia de nuestra región, en su mayoría provienen de colegios nacionales, sus edades oscilan entre 16 y 20 años y su condición socio económica, por lo general es media, y en algunos es baja.

3.1.4 Características culturales y ambientales

En la Universidad Nacional de Cajamarca, se dan expresiones culturales como las fiestas de cachimbo, actividad que se ha hecho una tradición que celebra la bienvenida a los jóvenes estudiantes ingresantes a esta casa superior de estudio. Así también, en la Escuela Académico Profesional de Ingeniería de Sistemas anualmente se lleva a cabo la semana sistémica, semana de integración en la que se desarrollan ponencias, talleres y actividades deportivas y culturales.

Con respecto a las características ambientales, la Universidad cuenta con áreas verdes en todo el campus universitario, pero aún es una preocupación el manejo de aguas residuales y tratamiento de los residuos sólidos.

3.2 Hipótesis de investigación

Hipótesis general

La aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL) influye significativamente en el diseño de algoritmos de la asignatura “Fundamentos de Programación”, en los estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020.

Hipótesis específicas

El nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, es deficiente, antes de la aplicación de la estrategia de enseñanza INPAL.

La estrategia de enseñanza INPAL es aplicada en base a los resultados del pretest, para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas.

El nivel de diseño de algoritmos de los estudiantes de la asignatura de Fundamentos de Programación de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, es bueno, después de la aplicación de la estrategia de enseñanza INPAL.

3.3 Variables de la investigación

Variable Independiente: Estrategia de enseñanza “Inducción para Algoritmos” (INPAL).

Variable Dependiente: Diseño de algoritmos.

3.4 Matriz de Operacionalización de variables

Variable	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Técnicas / instrumentos
VI: Estrategia de enseñanza de INPAL.	Procedimiento de enseñanza de algoritmos basada en el aprendizaje significativo y el razonamiento inductivo, es decir consiste en la solución de un problema a partir de una solución matemática (saber previo) de un caso particular con valores numéricos, para luego generalizar la solución en el diseño de un algoritmo (nuevo conocimiento) en donde se definen las variables y el proceso. (Ausubel, 2002) (Gagné, 1992) (Díaz y Hernández, 2005)	Nivel de aplicación de la estrategia por el estudiante al resolver y asociar un problema matemático particular con un caso general que corresponde al diseño de algoritmos para programación, anclando así el nuevo conocimiento a sus saberes previos.	Resolución matemática del problema.	<ul style="list-style-type: none"> - Identifica los valores numéricos del problema. - Identifica los valores de entrada. - Identifica los valores de salida. - Establece el proceso o fórmula que produce la salida deseada. - Aplica el proceso o fórmula, en base a los valores de entrada, para obtener los valores de salida. 	Observación/ Rúbrica de evaluación.
			Asociación de solución matemática con los elementos del algoritmo.	<ul style="list-style-type: none"> - Relaciona los valores numéricos con las variables del problema. - Relaciona los valores de entrada con las variables de entrada. - Relaciona los valores de salida con las variables de salida. - Relaciona el proceso matemático con el proceso algorítmico que produce la salida deseada. 	
VD: Diseño de algoritmos.	Proceso que convierte los resultados del análisis del problema en un diseño modular con refinamientos sucesivos que permitan una posterior traducción a un lenguaje de programación. (Joyanes, 2008)	Puntaje obtenido por el estudiante al evaluarse su capacidad de diseño de algoritmos para programación.	Análisis del Problema.	<ul style="list-style-type: none"> - Identifica las variables del problema. - Identifica las variables de entrada. - Identifica las variables de salida. - Establece el proceso que produce la salida deseada 	Observación/ Evaluación pretest y postest. Rúbrica de evaluación.
			Diseño de la solución algorítmica.	<ul style="list-style-type: none"> - Establece la declaración de variables. - Establece la entrada de datos. - Establece el proceso a seguir. - Establece la salida de resultados. 	

3.5 Población y muestra

Población: Conjunto de estudiantes que concuerdan con determinadas especificaciones, Lepkowski (como se citó en Hernández Sampieri et al., 2014). Por tanto, la población estuvo conformada por 58 estudiantes matriculados en la asignatura de Fundamentos de Programación de la Escuela Académico Profesional de Ingeniería de Sistemas de la UNC en el año académico 2020, los cuales están iniciando sus estudios en algoritmos y programación, con poco o nulo conocimiento en el diseño de algoritmos.

Muestra: Es una muestra no probabilística o dirigida, ya que se trata de una investigación experimental, cuya elección de casos depende del criterio del investigador (Hernández Sampieri et al., 2014). Por tanto, la muestra fue igual a la población, por lo que se ha elegido intencionalmente a los 58 estudiantes, los que son representativos de la situación académica a estudiar, es decir estudiantes que inician sus estudios en algoritmos y programación.

Tabla 4

Muestra de la investigación.

Grupo (Fundamentos de Programación)	Ciclo	Cantidad de estudiantes
A1	2020 - I	13
B1	2020 - I	13
A1	2020 - I (nivelación)	16
A2	2020 - I (nivelación)	16
TOTAL		58

Nota. Fuente: Matriculas EAPIS-UNC, ciclo 2020-I.

3.6 Unidad de análisis

La unidad de análisis está constituida por cada uno de los 58 estudiantes que conforman la muestra de estudio, pertenecientes al primer ciclo de estudios de la carrera profesional de Ingeniería de Sistemas, que están llevando la asignatura de Fundamentos de Programación, en el ciclo académico 2020-I.

3.7 Métodos de investigación

Como método general, se empleó el **método científico**, pues la presente investigación tuvo como principal soporte la observación, sobre la base de la observación directa de los hechos, se propone una supuesta explicación a estos hechos, es decir, se elaboró la hipótesis. Esta hipótesis sirve de guía para planear los experimentos y se realiza entonces la experimentación. De los resultados afirmativos de la experimentación surgirá la teoría, que es la confirmación de la hipótesis. Esta teoría será un principio general validado científicamente, y que tiene carácter provisorio por cuanto no es todavía la verdad comprobada (Valderrama Mendoza, 2002).

El método científico se aplicó a la presente investigación, desarrollando los siguientes pasos: primero, la identificación del problema a través de una pregunta principal y tres problemas derivados, para luego establecer una hipótesis general y tres derivadas, seguidamente se procedió con la observación de los hechos a través de una rúbrica de evaluación aplicada a los 58 estudiantes que cursaban la asignatura de “Fundamentos de Programación”, año 2020. Luego se aplicó la prueba paramétrica T de Student, para grupos independientes, para la contrastación de la hipótesis, y a raíz de ello se obtuvo verdades comprobadas a través de los resultados y discusión de la investigación.

Cómo métodos complementarios se emplearon los siguientes:

Método estadístico, ya que en la presente investigación se generarán datos que tendrán que ser analizados mediante la estadística descriptiva e inferencial. Este método, según Hernández Sampieri et al. (2014) comprende los siguientes pasos:

1. Seleccionar un software apropiado para analizar los datos.
2. Ejecutar el programa: SPSS, Minitab, STATS, SAS u otro equivalente.
3. Explorar los datos:
 - a) Analizar descriptivamente los datos por variable.
 - b) Visualizar los datos por variable.
4. Evaluar la confiabilidad y validez logradas por los instrumentos de medición.
5. Analizar mediante pruebas estadísticas las hipótesis planteadas (análisis estadístico inferencial).
6. Realizar análisis adicionales.
7. Preparar los resultados para presentarlos (tablas, gráficas, figuras, cuadros, etcétera).

El método estadístico se aplicó a la presente investigación, de la siguiente manera: se eligió usar el software SPSS, se evaluó la confiabilidad y validez de los instrumentos de recolección de datos, luego se aplicó la prueba de normalidad y de homogeneidad de los datos. Para contrastar las hipótesis derivadas se aplicó la estadística descriptiva, presentando los resultados en gráficos y para contrastar la hipótesis general, se aplicó la prueba paramétrica t-student, presentando los resultados en tablas.

Método de recolección de datos cuantitativos, que consiste en recolectar los datos pertinentes sobre los atributos, conceptos o variables de las unidades de muestreo/ análisis o casos (participantes, grupos, fenómenos, procesos, organizaciones, etcétera).

Recolectar los datos implica elaborar un plan detallado de procedimientos que nos conduzcan a reunir datos con un propósito específico. (Hernández Sampieri et al., 2014)

Este método, en la presente investigación, se hizo a través del diseño de una rúbrica de evaluación para el diseño de algoritmos, la cual fue validada a través de juicio de expertos y su confiabilidad se determinó a través de una prueba piloto antes de su aplicación a la muestra de estudio.

Método hipotético - deductivo, el que consiste en un procedimiento que intenta dar respuesta a los problemas que se plantea la ciencia a través de la formulación de hipótesis y la deducción de consecuencias que deberán ser empíricamente contrastadas. Los pasos a seguir son 1) Planteamiento de un problema, 2) formulación de una hipótesis, solución posible al problema, 3) deducción de las consecuencias observacionales refutatorias, 4) contrastación empírica de las consecuencias observacionales, y 5) Falsedad o aceptación de la hipótesis (Cabanillas, 2019).

Este método, en la presente investigación, se hizo a través del planteamiento de un problema principal y tres problemas, para luego revisar teorías y bases teóricas, que permitieron formular una hipótesis general y tres hipótesis derivadas, a través del razonamiento deductivo; luego fueron contrastadas empíricamente, para lo cual se estableció una hipótesis nula H_0 para refutar a las primeras, determinándose, finalmente, la aceptación de cada una de ellas.

Método Analítico - Sintético, que consiste en la descomposición de un todo en sus partes, con el fin de observar las relaciones, similitudes, diferencias, causas, naturaleza y efectos; y el método sintético es un proceso de razonamiento que tiende a construir un todo, a partir de los elementos distinguidos por el análisis manifestado en las conclusiones del estudio. Valderrama (2015)

Este método, en la presente investigación, se aplicó al realizar la operacionalización de variables, pues es allí donde se descompone el objeto de estudio en variables, dimensiones e indicadores, lo que permitió recoger y analizar los datos, para luego obtener resultados y en base a ello las conclusiones de la investigación.

Método inductivo, procedimiento mediante el cual a partir de hechos singulares se pasa a proposiciones generales, lo que ayuda a la formulación de hipótesis. En este sentido, la inducción, es el razonamiento que pasa de los hechos singulares obtenidos por vía experimental a las generalizaciones. (Intriago Alcívar et al., 2019)

Este método, se aplicó principalmente en el diseño de la estrategia de enseñanza “Inducción para Algoritmos”, pues de ejercicios particulares de matemáticas se busca generalizar la solución a través de un algoritmo, con el fin de mejorar la comprensión del problema y así mejorar el nivel de diseño de algoritmos.

3.8 Tipo de investigación

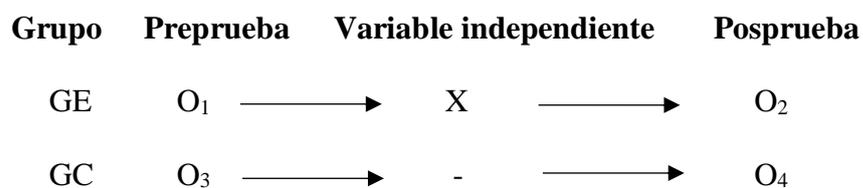
La presente investigación, por su finalidad, es una Investigación Aplicada, pues busca conocer el nivel actual en el diseño de algoritmos de los estudiantes de Fundamentos de Programación de la carrera profesional de Ingeniería de Sistemas, para luego diseñar y aplicar una estrategia de enseñanza basada en la inducción para mejorar el nivel de aprendizaje en el diseño de algoritmos de programación. (Valderrama Mendoza, 2002)

Por su profundidad, es una Investigación Explicativa, pues su interés se centra en determinar el nivel en el diseño de algoritmos y cómo la estrategia de enseñanza INPAL contribuye a mejorar este nivel de aprendizaje en el diseño de algoritmos. (Valderrama Mendoza, 2002)

Por su alcance temporal, es una investigación transversal, pues estudia a los sujetos en un momento dado; es decir, los datos son recopilados en un único momento. En tal sentido, la presente investigación permitirá estudiar procesos de cambio de los estudiantes de la asignatura de Fundamentos de Programación de la carrera de Ingeniería de Sistemas, en un sólo periodo de tiempo (ciclo académico 2020-I), en el cual se realizará dos observaciones en los sujetos de estudio (pretest y postest). (Cabanillas, 2019)

3.9 Diseño de investigación

El diseño según Valderrama Mendoza (2002), corresponde a un diseño cuasi experimental, en donde se manipula la variable independiente para ver su efecto y relación con la variable dependiente, el diseño fue con preprueba y posprueba con grupo de control no aleatorio, es decir se procuró emplear grupos los más equivalentes posibles al inicio del estudio. El esquema es el siguiente:



Donde:

GE: Grupo experimental al que se aplicará la estrategia de enseñanza INPAL.

GC: Grupo control al que no se aplicará la estrategia de enseñanza INPAL.

X: Aplicación de la estrategia de enseñanza INPAL.

O_i (i=1,2,3,4): Medición del nivel de diseño de algoritmos de los estudiantes de la asignatura de “Fundamentos de Programación”, Si aparece antes del estímulo experimental, se trata de una preprueba; si es después, una posprueba.

3.10 Técnicas e instrumentos de recolección de información

En la presente investigación se empleó la técnica de observación durante el proceso de aprendizaje de la estrategia, y también para determinar el nivel de diseño de algoritmos, antes y después de aplicada la estrategia. Como instrumento se empleó una rúbrica de evaluación para recoger los puntajes por criterio de desempeño en el diseño de algoritmos, y también para determinar el nivel de aplicación de la estrategia didáctica INPAL. (ver apéndice 1)

Para medir el nivel en el diseño de algoritmos, se diseñó una rúbrica de evaluación, que contempla 2 aspectos de evaluación, con sus respectivos criterios de desempeño y valoraciones. (ver apéndice 1)

Análisis del Problema (50%)

1. Define las variables del problema. (2 pts)
2. Identifica las variables de entrada. (2 pts)
3. Identifica las variables de salida (2 pts)
4. Identifica el proceso para obtener la salida deseada. (4 pts)

Diseño del Algoritmo (50%)

5. Establece la declaración de variables (2 pts)
6. Establece la entrada de datos (2 pts)
7. Establece el proceso a seguir (4 pts)
8. Establece la salida de resultados (2 pts)

Además, se estableció 5 niveles de desempeño en el diseño de algoritmos:

Tabla 5

Nivel de diseño de algoritmos y su escala.

N°	Nivel de diseño de algoritmos	Escala de 0-20
1	Deficiente	0 - 4.5
2	Insuficiente	5 - 10
3	Regular	10.5 - 14.5
4	Bueno	15 - 19
5	Excelente	19.5 - 20

Nota. Fuente: Adaptado de Mertler C., Practical Assessment, Research and Evaluation Vol.7 (Artículo 25)

Deficiente: representa un nivel casi nulo en el diseño de algoritmos, pues no identifica variables, constantes ni el proceso del análisis del problema, generando una representación nula del algoritmo.

Insuficiente: representa un nivel inicial en el diseño de algoritmos, pues identifica pocas variables, constantes y un proceso inicial del análisis del problema, generando una vaga representación del algoritmo.

Regular: representa un nivel medio en el diseño de algoritmos, pues identifica parcialmente las variables, constantes y un proceso parcial del análisis del problema, generando una representación regular del algoritmo.

Bueno: representa un nivel aceptable en el diseño de algoritmos, pues identifica la mayoría de variables, constantes y un proceso completo del análisis del problema, generando una correcta representación del algoritmo.

Excelente: representa un nivel destacado en el diseño de algoritmos, pues identifica completamente las variables, constantes y un proceso óptimo del análisis del problema, generando una óptima representación del algoritmo.

Tabla 6

Ficha Técnica de Rúbrica de Evaluación del Diseño de Algoritmos.

Nombre	Rúbrica de Evaluación del Diseño de Algoritmos.
Autor	Lisi Vásquez Fernández (2020).
Evalúa	Nivel del diseño de algoritmos (2020) - EAPIS - UNC.
Dimensiones	Análisis del problema (ítems 1, 2, 3 y 4). Diseño de la solución algorítmica (ítems 5, 6, 7 y 8).
Nº de ítems	8 ítems.
Dirigido a	Estudiantes de la asignatura de Fundamentos de Programación - EAPIS – UNC.
Duración	10 a 15 minutos.
Pautas para corrección	El instrumento se califica, según los niveles de desempeño de los estudiantes, de acuerdo a la siguiente escala Deficiente, Insuficiente, Regular, Bueno y Excelente, los que tienen un puntaje específico por criterio de desempeño. El puntaje total de la rúbrica oscila entre 0 y 20, y se califica de acuerdo al puntaje obtenido por cada estudiante de la siguiente manera: 0 - 4.5 (Deficiente), 5 - 10 (Insuficiente), 10.5 - 14.5 (Regular), 15 - 19 (Bueno) y 19.5 - 20 (Excelente).

Complementariamente se aplicó la técnica de la encuesta, para obtener información sobre la percepción de los estudiantes con respecto a la estrategia utilizada y como instrumento se empleó el cuestionario para recoger las respuestas por pregunta.

3.11 Técnicas para el procesamiento y análisis de la información

Para el procesamiento de datos, se utilizó la técnica de tabulación de datos, para la variable dependiente en el pretest y posttest, para lo cual se empleó hojas de cálculo en Microsoft Excel.

Para el análisis de datos y medición de la variable dependiente, se utilizó el análisis estadístico inferencial, cuyo método se determinó después de aplicada la prueba de normalidad y la prueba de homogeneidad de los datos; aplicándose la prueba de T-Student para la contrastación de la hipótesis general de la investigación.

3.12 Validez y confiabilidad

La validez se evidenció a través de juicio de expertos, siendo 4 los expertos, docentes de la Escuela Académico Profesional de Ingeniería de Sistemas, quienes revisaron el contenido de cada instrumento.

Primero, se elaboró un documento en word para la validación de instrumentos de recolección de datos, según el anexo 5 del protocolo para la elaboración de proyectos de investigación e informe de tesis de los programas de maestría y doctorado en educación. Luego se elaboró un formato de evaluación de instrumento en Excel, en donde se consideró 4 criterios de evaluación por cada ítem de los instrumentos: pertinencia, claridad conceptual, redacción y escala y codificación, criterios que fueron evaluados según una escala valorativa del 1 al 4: inaceptable, deficiente, regular y bueno.

Luego, se conversó con cada uno de los expertos para solicitar su apoyo en la validación de los 3 instrumentos de recolección de datos: la rúbrica de evaluación del nivel de diseño de algoritmos, la rúbrica para la evaluación del nivel de aplicación de la

estrategia de enseñanza INPAL y la encuesta sobre la percepción de los estudiantes sobre el empleo de la estrategia de enseñanza INPAL, quienes dieron algunas sugerencias y finalmente coincidieron con el siguiente resultado:

Tabla 7

Resumen de validación de contenido por juicio de expertos.

N°	Instrumento	N° de ítems revisados	N° de ítems válidos	% de ítems válidos
1	Rúbrica de evaluación del nivel de diseño de algoritmos.	8	8	100%
2	Rúbrica de evaluación del nivel de aplicación de la estrategia de enseñanza INPAL para el diseño de algoritmos.	7	7	100%
3	Encuesta sobre la percepción de los estudiantes, con respecto a la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos.	11	11	100%

Nota. Instrumentos de recolección de datos revisados y validados (ver apéndice 3).

La confiabilidad del instrumento de evaluación del nivel de diseño de algoritmos (variable dependiente), se determinó a través de una prueba piloto en 14 estudiantes que cursaban el V ciclo de la carrera de Ingeniería de Minas y que estaban llevando el curso de Computación y Algoritmos II, ciclo 2020-I.

Para ello en la primera semana de clases, se tomó un examen en el que se propuso 2 problemas sobre el diseño de algoritmos, luego subieron sus exámenes desarrollados a una carpeta de Google Drive, los que posteriormente fueron revisados según la rúbrica de evaluación del nivel de diseño de algoritmos y se obtuvo una matriz de puntajes por criterio de desempeño, sobre la cual se aplicó el cálculo del coeficiente de Alfa de Cronbach, dando como resultado un de 0.83312, lo que indica que la confiabilidad el instrumento es bastante aceptable. (Hernández Sampieri et al., 2014). (ver apéndice 2)

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1 Matriz general de resultados

Después de revisar las respuestas de los exámenes, tanto del pretest como del postest, aplicados a los estudiantes del grupo control y del grupo experimental, a través de la rúbrica de evaluación, se pudo obtener una matriz general de resultados por indicador y por dimensión de la variable dependiente, datos que fueron registrados y consolidados en Excel y luego pasados a SPSS, versión 26. (Ver apéndice 4)

4.2 Resultados por dimensiones de las variables de estudio

A continuación, se muestran unas tablas comparativas de los resultados obtenidos por dimensiones de la variable dependiente “Diseño de algoritmos”, las que se han medido a través de las calificaciones del pretest y postest, obtenidas por los estudiantes de Fundamentos de Programación de la carrera profesional de Ingeniería de Sistemas, ciclo 2020-I.

4.2.1 Dimensión Análisis del Problema, en el pretest

Tabla 8

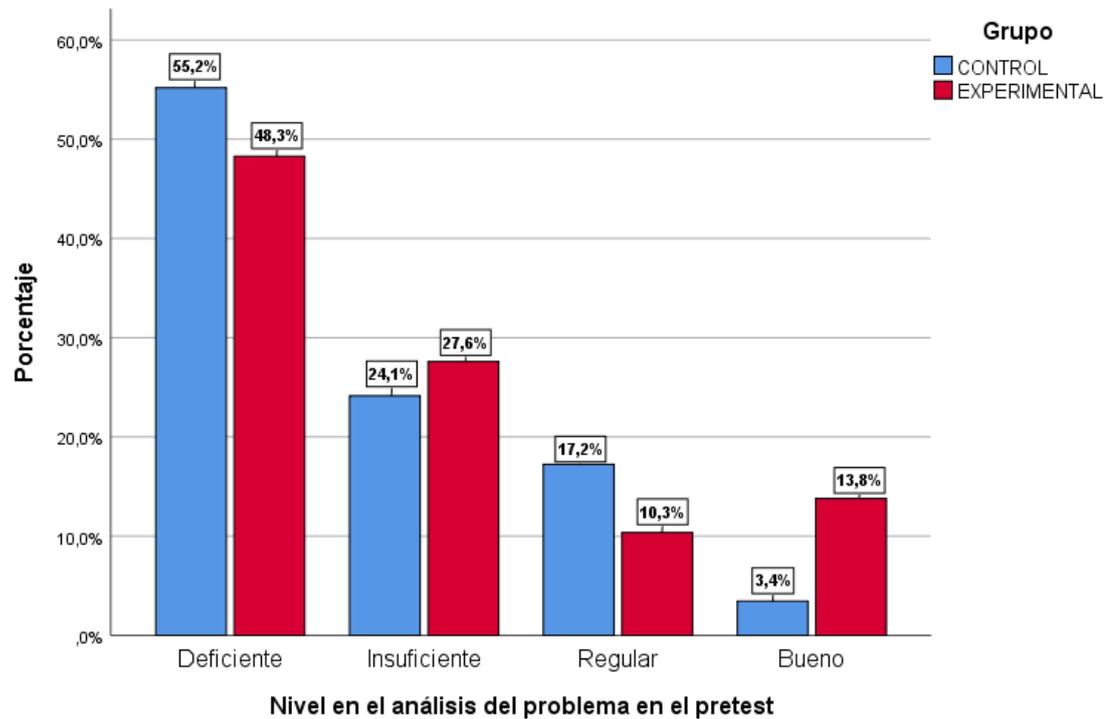
Nivel en la dimensión análisis del problema, en el pretest.

			Nivel en el análisis del problema en el pretest				
			Deficiente	Insuficiente	Regular	Bueno	Total
Grupo	CONTROL	Recuento	16	7	5	1	29
		% dentro de Grupo	55,2%	24,1%	17,2%	3,4%	100,0%
	EXPERIMENTAL	Recuento	14	8	3	4	29
		% dentro de Grupo	48,3%	27,6%	10,3%	13,8%	100,0%
Total	Recuento		30	15	8	5	58
	% dentro de Grupo		51,7%	25,9%	13,8%	8,6%	100,0%

Nota. Fuente: Procesamiento de datos SPSS.

Figura 144

Nivel en la dimensión análisis del problema, en el pretest.



Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla y la figura, podemos observar que, en el pretest, el nivel de análisis del problema de los estudiantes, tanto del grupo control como del experimental, fue deficiente e insuficiente; ya que, el 55.2% de estudiantes del grupo control, se encuentran en la categoría de deficiente y el 24.1%, se encuentran en la categoría de insuficiente. Asimismo, el 48.3% de estudiantes del grupo experimental, se encuentran en la categoría de deficiente y el 27.6%, se encuentran en la categoría de insuficiente.

El nivel de análisis del problema de los estudiantes es deficiente, pues en total el 51.7% de estudiantes se encuentran en la categoría de deficiente, y esto es porque aún no definen las variables y constantes del problema, no identifican las variables de entrada, no identifican las variables de salida y no determinan el proceso para obtener la salida deseada.

El nivel de análisis del problema de los estudiantes es insuficiente, pues en total el 25.9% de los estudiantes se encuentran en la categoría de insuficiente y esto es porque aún identifican pocas variables de entrada, identifican pocas variables de salida y determinan inicialmente el proceso para obtener la salida deseada.

Estos resultados no se pueden comparar con antecedentes ya que no se cuenta con información en esta dimensión, en estudios de pretest.

4.2.2 Dimensión Análisis del Problema, en el postest

Tabla 9

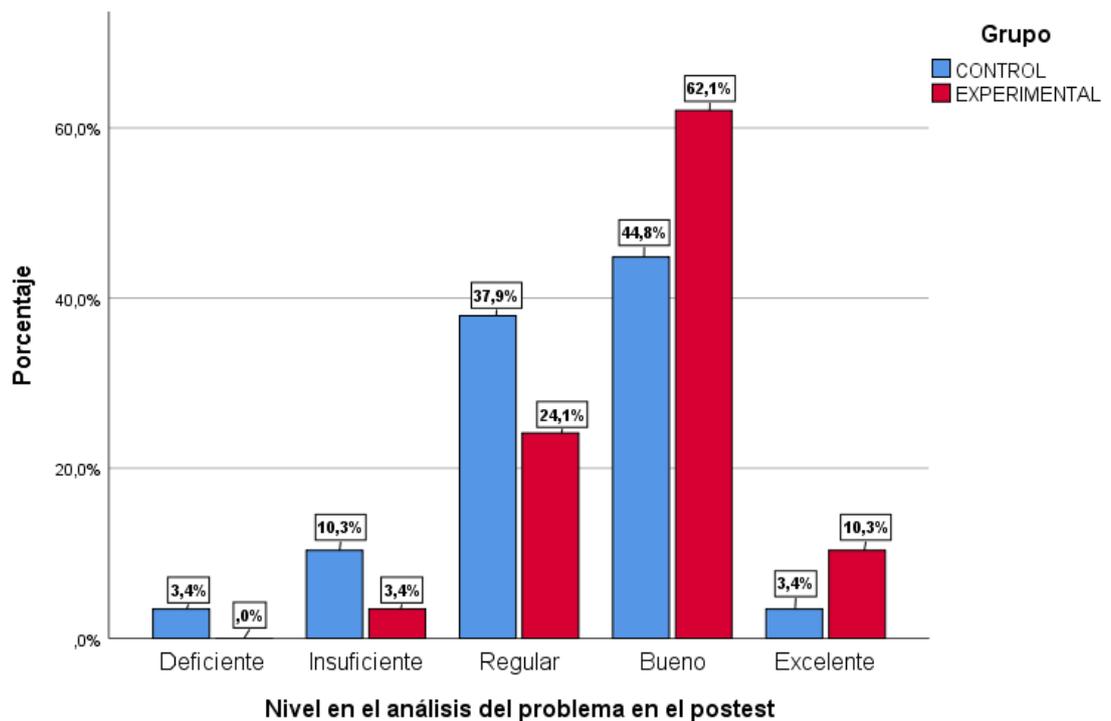
Nivel en la dimensión análisis del problema, en el postest.

			Nivel en el análisis del problema en el postest					
			Deficiente	Insuficiente	Regular	Bueno	Excelente	Total
Grupo	CONTROL	Recuento	1	3	11	13	1	29
		% dentro de Grupo	3,4%	10,3%	37,9%	44,8%	3,4%	100,0%
	EXPERIMENTAL	Recuento	0	1	7	18	3	29
		% dentro de Grupo	0,0%	3,4%	24,1%	62,1%	10,3%	100,0%
Total		Recuento	1	4	18	31	4	58
		% dentro de Grupo	1,7%	6,9%	31,0%	53,4%	6,9%	100,0%

Nota. Fuente: Procesamiento de datos SPSS.

Figura 155

Nivel en la dimensión análisis del problema, en el postest.



Nota. Fuente: Procesamiento de datos SPSS.

Tabla 10

Media de los puntajes de la dimensión análisis del problema, en el postest.

	Grupo	N	Media
Puntaje postest dimensión:	CONTROL	29	13,828
Análisis del problema	EXPERIMENTAL	29	16,310

Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla 8 y el gráfico, podemos observar que, en el postest, hubo una mejora en el nivel de análisis del problema de los estudiantes, tanto del grupo control como del experimental, pues los mayores porcentajes se encuentran en regular y bueno. El 44.8% de estudiantes del grupo control, se encuentran en la categoría de bueno y un 37.9%, se encuentran en la categoría de regular. Asimismo, el 62.1% de estudiantes del grupo experimental, se encuentran en la categoría de bueno y el 24.1%, se encuentran en la categoría de regular. Además, se puede notar un mejor nivel en el análisis del problema en el grupo experimental, pues hay una diferencia, en el nivel de bueno, del 17.3%, con respecto al grupo control; asimismo, hay una diferencia, en el nivel de excelente, del 6.9%, con respecto al grupo control.

El nivel de análisis del problema de los estudiantes es bueno, pues en total el 53.4% de estudiantes se encuentran en la categoría de bueno, y esto es porque definen la mayoría de las variables y constantes del problema, identifica la mayoría de las variables de entrada, identifica la mayoría de las variables de salida y determina completamente el proceso para obtener la salida deseada.

El nivel de análisis del problema de los estudiantes, también es regular, pues en total el 31% de estudiantes se encuentran en la categoría de regular, y esto es porque definen parcialmente las variables y constantes del problema, identifican parcialmente las variables de entrada, identifican parcialmente las variables de salida e identifican parcialmente el proceso para obtener la salida deseada.

Finalmente, en la tabla 9 se puede observar que en el grupo experimental hay una mejora en el diseño de algoritmos, con respecto al grupo control; pues la media del puntaje del grupo experimental es de 16.31, mientras que en el grupo control, es de 13.83, es decir una diferencia superior de 2.48 puntos, lo que representa un 15.2%.

Estos resultados no se pueden comparar con antecedentes ya que no se cuenta con información en esta dimensión, en estudios de postest.

4.2.3 Dimensión Diseño de la Solución Algorítmica, en el pretest

Tabla 11

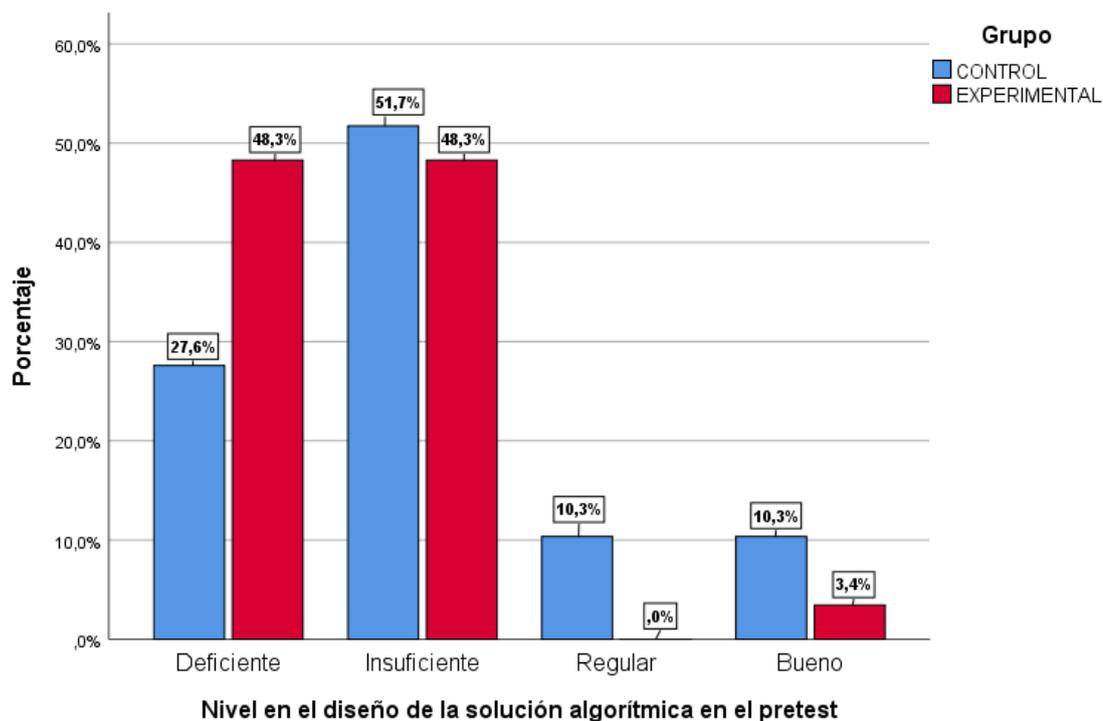
Nivel en la dimensión diseño de la solución algorítmica, en el pretest.

		Nivel en el diseño de la solución algorítmica en el pretest				Total	
		Deficiente	Insuficiente	Regular	Bueno		
Grupo	CONTROL	Recuento	8	15	3	3	29
		% dentro de Grupo	27,6%	51,7%	10,3%	10,3%	100,0%
	EXPERIMENTAL	Recuento	14	14	0	1	29
		% dentro de Grupo	48,3%	48,3%	0,0%	3,4%	100,0%
Total		Recuento	22	29	3	4	58
		% dentro de Grupo	37,9%	50,0%	5,2%	6,9%	100,0%

Nota. Fuente: Procesamiento de datos SPSS.

Figura 166

Nivel en la dimensión diseño de la solución algorítmica, en el pretest.



Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla 8 y el gráfico, podemos observar que, en el pretest, el nivel de diseño de los algoritmos, tanto del grupo control como del experimental, es deficiente e insuficiente; ya que, el 27.6% de estudiantes del grupo control, se encuentran en la categoría de deficiente y el 51.7%, se encuentran en la categoría de insuficiente. Asimismo, el 48.3% de estudiantes del grupo experimental, se encuentran en la categoría de deficiente y el 48.3%, se encuentran en la categoría de insuficiente.

El nivel de diseño del algoritmo de los estudiantes es deficiente, pues en total el 37.9% de estudiantes se encuentran en la categoría de deficiente y esto es

porque aún no establecen la declaración de variables, no establecen la entrada de datos, no establecen el proceso a seguir y no establecen la salida de resultados.

El nivel de análisis del problema de los estudiantes es insuficiente, pues en total el 50% de estudiantes se encuentran en la categoría de insuficiente y esto es porque establecen pocas variables con algunos tipos de datos correctos y pocas constantes de ser necesario, establece inicialmente la entrada de datos, establece inicialmente el proceso a seguir y establece inicialmente la salida de resultados.

Estos resultados no se pueden comparar con antecedentes ya que no se cuenta con información en esta dimensión, en estudios de pretest.

4.2.4 Dimensión Diseño de la Solución Algorítmica, en el postest

Tabla 12

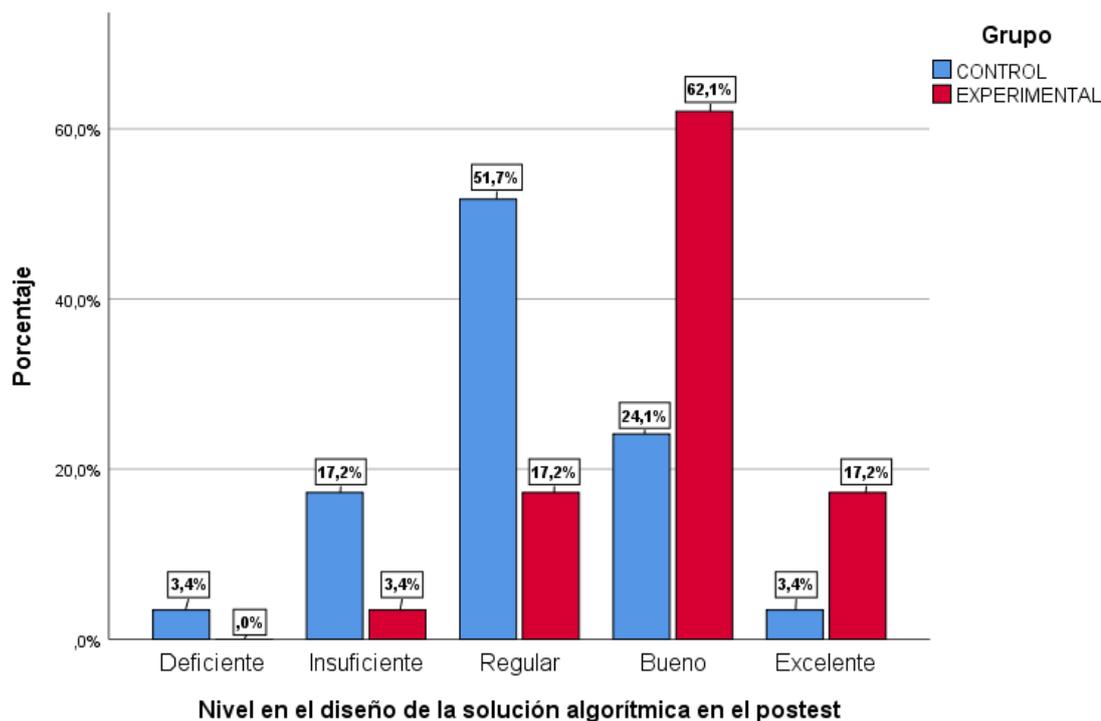
Nivel en la dimensión del diseño de la solución algorítmica, en el postest.

			Nivel en el diseño de la solución algorítmica en el postest					
			Deficiente	Insuficiente	Regular	Bueno	Excelente	Total
Grupo	CONTROL	Recuento	1	5	15	7	1	29
		% dentro de Grupo	3,4%	17,2%	51,7%	24,1%	3,4%	100,0%
	EXPERIMENTAL	Recuento	0	1	5	18	5	29
		% dentro de Grupo	0,0%	3,4%	17,2%	62,1%	17,2%	100,0%
Total		Recuento	1	6	20	25	6	58
		% dentro de Grupo	1,7%	10,3%	34,5%	43,1%	10,3%	100,0%

Nota. Fuente: Procesamiento de datos SPSS.

Figura 177

Nivel en la dimensión del diseño de la solución algorítmica, en el postest.



Nota. Fuente: Procesamiento de datos SPSS.

Tabla 13

Media de los puntajes de la dimensión solución algorítmica, en el postest.

	Grupo	N	Media
Puntaje postest dimensión:	CONTROL	29	12,793
Diseño de la solución algorítmica	EXPERIMENTAL	29	16,810

Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla 11 y el gráfico, podemos observar que, en el postest, hubo una mejora en el nivel de la solución algorítmica de los estudiantes, tanto del grupo control como del experimental, pues los mayores porcentajes se encuentran en regular y bueno. El 24.1% de estudiantes del grupo control, se encuentran en

la categoría de bueno y un 51.7%, se encuentran en la categoría de regular. Asimismo, el 62.1% de estudiantes del grupo experimental, se encuentran en la categoría de bueno y el 17.2%, se encuentran en la categoría de regular. Además, se puede notar un mejor nivel en el diseño de la solución algorítmica en el grupo experimental, pues hay una diferencia, en el nivel de bueno, del 38%, con respecto al grupo control; asimismo, hay una diferencia, en el nivel de excelente, del 13.8%, con respecto al grupo control.

El nivel de diseño del algoritmo de los estudiantes es bueno, pues en total el 43.1% de estudiantes se encuentran en la categoría de bueno y esto es porque establecen la mayoría de las variables con su respectivo tipo de dato y constantes de ser necesario, establecen la mayoría de la entrada de datos, establecen completamente el proceso a seguir y establecen la mayoría de la salida de resultados.

El nivel de diseño del algoritmo de los estudiantes es regular, pues en total el 34.5% de estudiantes se encuentran en la categoría de regular y esto es porque establecen parcialmente las variables con algunos tipos de datos correctos y constantes de ser necesario, establecen parcialmente la entrada de datos, establecen parcialmente el proceso a seguir y establecen parcialmente la salida de resultados.

Finalmente, en la tabla 12 se puede observar que en el grupo experimental hay una mejora en el diseño de algoritmos, con respecto al grupo control; pues la media del puntaje del grupo experimental es de 16.81, mientras que en el grupo control, es de 12.79, es decir una diferencia superior de 4.02 puntos, lo que representa un 23.9%.

Estos resultados no se pueden comparar con antecedentes ya que no se cuenta con información en esta dimensión, en estudios de postest.

4.3 Resultados totales de las variables de estudio

A continuación, se muestran unos gráficos de los resultados totales obtenidos en la variable dependiente “Diseño de algoritmos”, los que muestran las calificaciones obtenidas por los estudiantes de Fundamentos de Programación de la carrera profesional de Ingeniería de Sistemas, ciclo 2020-I.

4.3.1 Pretest

Al determinar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, antes de la aplicación de la estrategia INPAL, se obtuvo los siguientes resultados:

Tabla 14

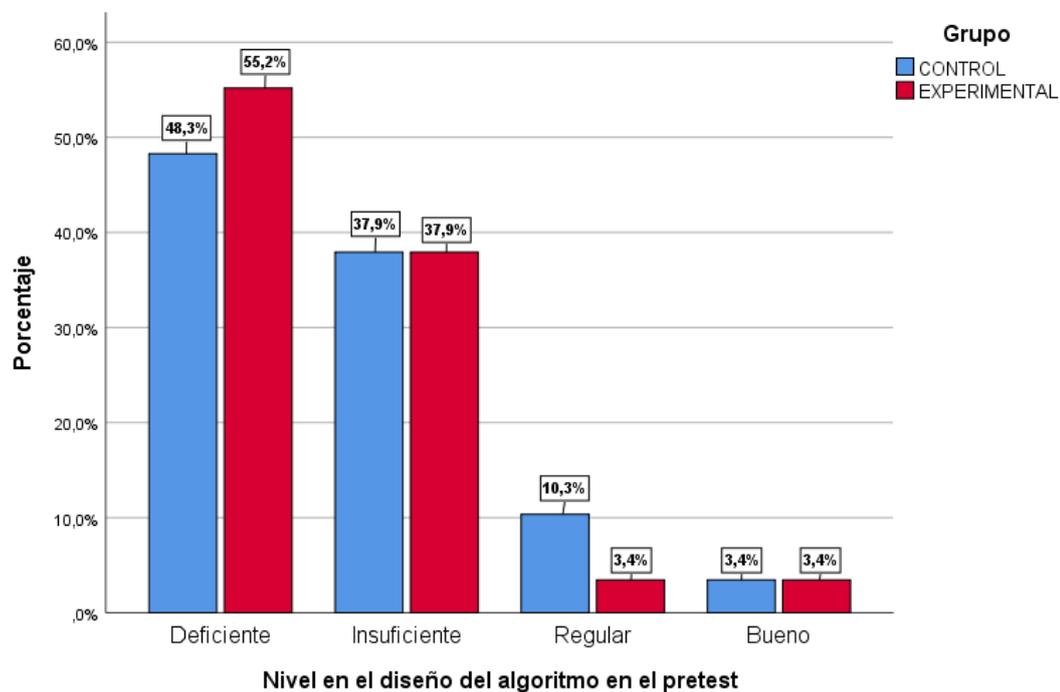
Nivel en el diseño de algoritmos, en el pretest.

		Nivel en el diseño de algoritmos en el pretest				Total	
		Deficiente	Insuficiente	Regular	Bueno		
Grupo	CONTROL	Recuento	14	11	3	1	29
		% dentro de Grupo	48,3%	37,9%	10,3%	3,4%	100,0%
	EXPERIMENTAL	Recuento	16	11	1	1	29
		% dentro de Grupo	55,2%	37,9%	3,4%	3,4%	100,0%
Total		Recuento	30	22	4	2	58
		% dentro de Grupo	51,7%	37,9%	6,9%	3,4%	100,0%

Nota. Fuente: Procesamiento de datos SPSS.

Figura 188

Nivel en el diseño de algoritmos, en el pretest.



Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla y el gráfico, se puede observar que, en el pretest, tanto en el grupo control como en el experimental, la mayor cantidad de estudiantes tienen un nivel de diseño de algoritmos de deficiente e **insuficiente**, pues los porcentajes más altos se encuentran en dichas categorías. En el grupo control, un 48.3% de estudiantes tienen un nivel de diseño de algoritmos de deficiente y un 37.9%, de insuficiente. En el grupo experimental, un 55.2% de estudiantes tienen un nivel de diseño de algoritmos de deficiente y un 37.9%, de insuficiente.

Además, según la tabla, se puede observar que el 51.7% de estudiantes tienen un nivel **deficiente** en el diseño de algoritmos, y ello se debe a un deficiente análisis del problema y a un deficiente diseño de la solución algorítmica.

De igual manera hay similitud con Lagos Barzola (2020), en el pretest, tanto la identificación y uso de variables adecuadas en el diseño de algoritmos como el uso adecuado de las estructuras de control de flujo de selección, es limitada en ambos grupos, pero no existe diferencia significativa entre el grupo control y el grupo experimental.

Del mismo modo hay similitud con Guimaraes y Gnecco (2017), en donde se observa un 41.43% de estudiantes aprobados, en el primer año de la aplicación de la estrategia práctica para la enseñanza de algoritmos, porcentaje que va aumentando en los siguientes años.

De forma análoga, en Salgado Castillo et al. (2015), en la preprueba, las calificaciones referidas al pensamiento algorítmico - computacional fueron iniciales, no existiendo mayores diferencias entre las calificaciones del grupo control y del grupo experimental.

En este sentido, la resolución de problemas computacionales representa una dificultad para los estudiantes en una etapa introductoria en la programación, lo que se evidencia en capacidad de abstracción y análisis para identificar y comprender el problema, que deben poseer los estudiantes. Y también se observa que existe el interés en la definición y manipulación de estrategias que fomenten el pensamiento algorítmico como habilidad en la resolución de problemas. (Velasco Ramírez, 2020)

4.3.2 Aplicación de la estrategia de enseñanza INPAL

El diseño de la estrategia didáctica INPAL, se ha realizado a partir de conocer, en el pretest, el nivel deficiente en el diseño de algoritmos de los

estudiantes que cursaban la asignatura de Fundamentos de Programación, año 2020; es decir, que no realizan un correcto análisis del problema y con ello un deficiente diseño de la solución algorítmica, en tal sentido al plantear la estrategia se busca que el estudiante haga un buen análisis del problema, tomando como ejemplo un problema matemático en particular, para luego realizar una asociación que permita definir con mayor claridad y precisión las variables de entrada, las variables de salida y el proceso para obtener la salida deseada.

La estrategia de enseñanza INPAL está incluida en el programa de sesiones de aprendizaje de la presente investigación, y cabe recalcar que tiene su fundamentación teórica en la teoría del aprendizaje significativo, en la teoría del aprendizaje de Gagné, en la teoría de las situaciones didácticas de Brousseau. Además, la estrategia de enseñanza INPAL incluye un razonamiento inductivo, es decir parte del desarrollo de un problema matemático específico para luego generalizar el problema en un algoritmo, realizando una asociación de valores del problema matemático con las variables del algoritmo computacional.

Asimismo, siendo las estrategias de enseñanza, procedimientos que el agente de enseñanza utiliza en forma reflexiva y flexible para promover el logro de aprendizajes significativos en los alumnos, convirtiéndose así en medios para prestar la ayuda pedagógica a los estudiantes, se ha planteado el el procedimiento de la estrategia de enseñanza INPAL, el cual forma parte de la presente propuesta de investigación (Díaz Barriga y Hernández Rojas, 2005).

La aplicación de la estrategia de enseñanza INPAL, se llevó a cabo en el ciclo regular 2020-I (del 17 de agosto al 11 de diciembre del 2020), y en el ciclo de nivelación 2020-I (del 8 de diciembre del 2020 al 26 de febrero del 2021), la

cual se aplicó a los estudiantes de la asignatura de “Fundamentos de Programación”, pertenecientes al primer ciclo de la carrera profesional de Ingeniería de Sistemas.

En el ciclo regular 2020-I, se tuvo 2 grupos A y B, siendo el grupo A, el grupo control con 13 estudiantes y el grupo B, el grupo experimental también con 13 estudiantes. La estrategia se la aplicó durante 5 semanas. Del mismo modo, en el ciclo de nivelación 2020-I, se tuvo el grupo A, con 2 grupos de práctica A1 y A2, siendo el grupo A1, el grupo control con 16 estudiantes y el grupo A2, el grupo experimental con 16 estudiantes. La estrategia se la aplicó durante 4 semanas, debido a que el ciclo fue más corto.

En ambos ciclos, la estrategia de enseñanza INPAL se la aplicó en los temas de diseño de algoritmos con estructura secuencial y estructura condicional, para lo cual se elaboró sesiones de aprendizaje y material didáctico como diapositivas y plantillas para el desarrollo de ejercicios sobre diseño de algoritmos y el nivel de la aplicación de la estrategia de enseñanza INPAL, se la fue observando, semana a semana, a través de una rúbrica de evaluación. (ver apéndice 1)

4.3.3 Postest

Al determinar el nivel de diseño de algoritmos, de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, después de la aplicación de la estrategia de enseñanza INPAL, se obtuvo los siguientes resultados:

Tabla 15

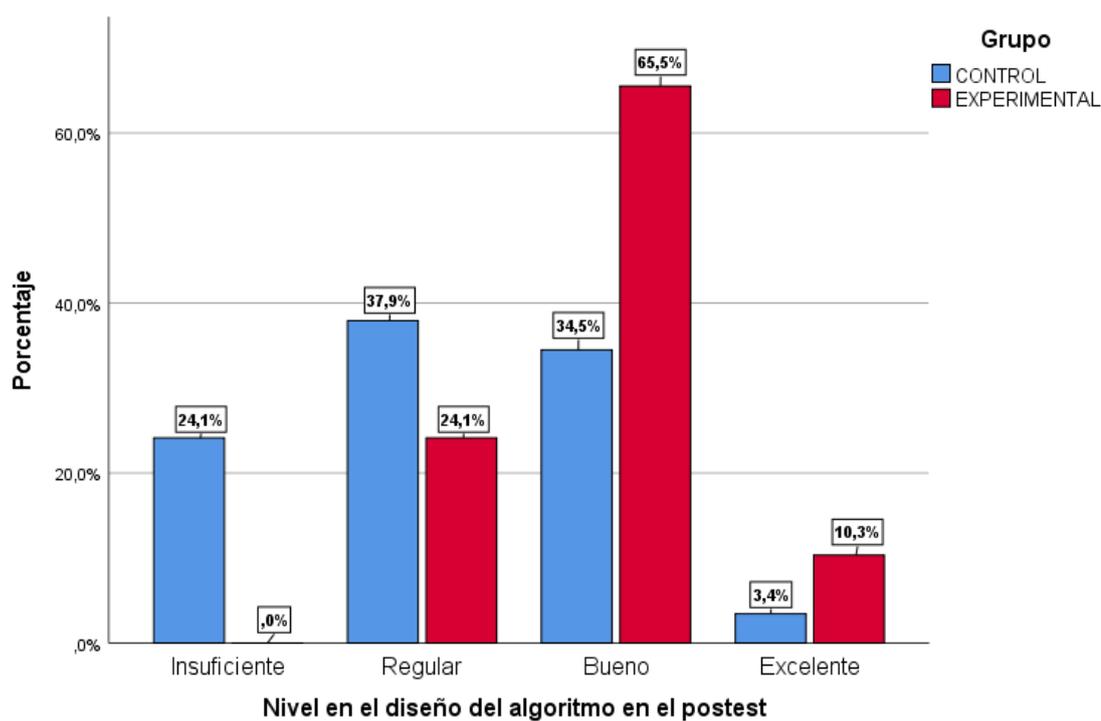
Nivel en el diseño de algoritmos, en el postest.

		Nivel en el diseño de algoritmos en el postest					
			Insuficiente	Regular	Bueno	Excelente	Total
Grupo	CONTROL	Recuento	7	11	10	1	29
		% dentro de Grupo	24,1%	37,9%	34,5%	3,4%	100,0%
	EXPERIMENTAL	Recuento	0	7	19	3	29
		% dentro de Grupo	0,0%	24,1%	65,5%	10,3%	100,0%
Total		Recuento	7	18	29	4	58
		% dentro de Grupo	12,1%	31,0%	50,0%	6,9%	100,0%

Nota. Fuente: Procesamiento de datos SPSS.

Figura 19

Nivel en el diseño de algoritmos, en el postest.



Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla y el gráfico, se puede observar que, en el posttest, hubo un progreso en el diseño de algoritmos, tanto en el grupo control como en el experimental, pues los mayores porcentajes se encuentran en **regular y bueno**. Sin embargo, se puede notar que hubo un mayor aprendizaje en el grupo experimental, ya que la mayor cantidad de estudiantes tiene un nivel de diseño de algoritmos de bueno con 65.5% y de excelente de 10.3%, con respecto al del grupo control que obtuvo un 34.5%, en bueno y 37.9% en regular.

Estos resultados son similares a los de **Ulker (2020)**, en donde el promedio de la tasa de éxito de aprendizaje aumentó, pues al menos el 84% de los estudiantes pensaron que lograron los resultados de aprendizaje del curso después de aplicar el método de 4 etapas, y el 62% de los estudiantes obtuvieron una buena calificación.

De forma análoga con Lagos Barzola (2020), ya que después del método aplicado (aprendizaje cooperativo) hubo una mejora significativa en el diseño de algoritmos (estructuras de control de flujo de selección y repetición) del grupo experimental y además, en la nota de la prueba de conocimientos las notas del grupo experimental fueron mayores a las del grupo control.

Así también, hay similitud con Guimaraes y Gnecco (2017), en donde con el pasar de los años en los cuales la estrategia fue implementada (importancia a las fases de análisis y prueba), la tasa de aprobación fue aumentando desde el 41,43% en el 2003 hasta el 75.47% en el 2009, lo que demuestra la eficacia de la estrategia.

De forma análoga, en Estrada Aro (2016), en el posttest, se obtuvo como resultado un incremento en el nivel de comprensión y en el nivel del análisis del problema, gracias al uso del software Pseint.

De manera similar, en Salgado Castillo et al. (2015), en la postprueba, se determinó que existen diferencias significativas entre las calificaciones de los estudiantes del grupo control y experimental, siendo estas últimas significativamente mayores, lo que implica un mejor aprovechamiento académico.

4.4 Prueba de hipótesis

Hipótesis Descriptiva 1

Hi: El nivel de diseño de algoritmos, de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, es insuficiente, antes de la aplicación de la estrategia INPAL.

Ho: El nivel de diseño de algoritmos, de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, no es insuficiente, antes de la aplicación de la estrategia INPAL.

Tabla 16*Nivel en el diseño de algoritmos, en el pretest.*

			Grupo		Total
			CONTROL	EXPERIMENTAL	
Nivel en el diseño de algoritmos en el pretest	Deficiente	Recuento	14	16	30
		% dentro de Grupo	48,3%	55,2%	51,7%
	Insuficiente	Recuento	11	11	22
		% dentro de Grupo	37,9%	37,9%	37,9%
	Regular	Recuento	3	1	4
		% dentro de Grupo	10,3%	3,4%	6,9%
	Bueno	Recuento	1	1	2
		% dentro de Grupo	3,4%	3,4%	3,4%
Total	Recuento	29	29	58	
	% dentro de Grupo	100,0%	100,0%	100,0%	

Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla de contingencia, se puede observar que un 51.7% de estudiantes tienen un nivel de diseño de algoritmos de deficiente y un 37,9%, de insuficiente; por lo tanto, concluimos que el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, es deficiente e insuficiente, antes de la aplicación de la estrategia INPAL. Por tanto, **SE ACEPTA** la hipótesis descriptiva 1.

Hipótesis Descriptiva 2.

Hi: La estrategia de enseñanza INPAL es aplicada para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas.

Ho: La estrategia de enseñanza INPAL no es aplicada para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas.

La estrategia de enseñanza “Inducción para Algoritmos” - INPAL, ha sido diseñada en base a la teoría del aprendizaje significativo, en la que se busca un anclaje del tema de aprendizaje de diseño de algoritmos, con la solución de problemas matemáticos ya existentes en la estructura cognitiva del estudiante; también, se ha basado en la teoría del procesamiento de la información, ya que con ella se ha comprendido el proceso del aprendizaje, de tal manera que nos permitan aplicar la estrategia de enseñanza INPAL, usando los principios y condiciones de aprendizaje. Además, se plantearon situaciones didácticas en las sesiones de aprendizaje para lograr un mejor aprendizaje en el diseño de algoritmos; asimismo, la estrategia incluye un razonamiento inductivo para ir de lo particular (solución matemática) a lo general (solución algorítmica), lo que está plasmado en el procedimiento de la estrategia de enseñanza y todos estos aspectos forman parte de la propuesta de la presente investigación.

La estrategia de enseñanza INPAL, se la aplicó durante 5 semanas al grupo experimental de 13 estudiantes (ciclo regular 2020-I) y durante 4 semanas al grupo experimental de 16 estudiantes (ciclo de nivelación 2020-I), los cuales desarrollaban los ejercicios sobre diseño de algoritmos, iniciando con la solución matemática del ejercicio para luego relacionar los datos con las variables del algoritmo y tener así una mejor comprensión del ejercicio, que permita una mayor precisión en la identificación de variables de entrada, de salida y un correcto proceso del algoritmo, y este procedimiento se aplicó siguiendo el plan de las sesiones de aprendizaje. Durante el tiempo que duró la aplicación de la estrategia, se observó que los estudiantes que aplicaban la estrategia

lograban un mejor entendimiento del problema, un mejor análisis del problema y por ende una mayor exactitud en el diseño del algoritmo.

Análisis y Discusión

En tal sentido, la estrategia de enseñanza INPAL fue aplicada para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura de “Fundamentos de Programación”. Por tanto, **SE ACEPTA** la hipótesis descriptiva 2.

Hipótesis Descriptiva 3

Hi. El nivel de diseño de algoritmos, de los estudiantes de la asignatura de Fundamentos de Programación de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, es bueno, después de la aplicación de la estrategia de enseñanza INPAL.

Ho. El nivel de diseño de algoritmos de los estudiantes de la asignatura de Fundamentos de Programación de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, no es bueno, después de la aplicación de la estrategia de enseñanza INPAL.

Tabla 17

Nivel en el diseño de algoritmos, en el postest.

			Grupo		Total
			CONTROL	EXPERIMENTAL	
Nivel en el diseño de algoritmos en el postest	Insuficiente	Recuento	7	0	7
		% dentro de Grupo	24,1%	0,0%	12,1%
	Regular	Recuento	11	7	18
		% dentro de Grupo	37,9%	24,1%	31,0%
	Bueno	Recuento	10	19	29
		% dentro de Grupo	34,5%	65,5%	50,0%
	Excelente	Recuento	1	3	4
		% dentro de Grupo	3,4%	10,3%	6,9%
Total	Recuento	29	29	58	
	% dentro de Grupo	100,0%	100,0%	100,0%	

Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla de contingencia, se puede observar que un 65.5% de estudiantes del grupo experimental, tienen un nivel de diseño de algoritmos de bueno y un 10.3%, de excelente; por lo tanto, concluimos que el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, es bueno, con respecto al grupo control, después de la aplicación de la estrategia INPAL. Por tanto, **SE ACEPTA** la hipótesis descriptiva 3.

Hipótesis estadística general:

Hi: La aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL), influye significativamente en el diseño de algoritmos de la asignatura “Fundamentos de Programación”, en los estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020.

Ho: La aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL), no influye significativamente en el diseño de algoritmos de la asignatura “Fundamentos de Programación”, en los estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020.

Para realizar análisis paramétricos según Hernández Sampieri et al. (2014), se debe cumplir los siguientes requisitos:

1. La distribución poblacional de la variable dependiente es normal.
2. El nivel de medición de las variables es por intervalos o razón.
3. Cuando dos o más poblaciones son estudiadas, tienen una varianza homogénea.

En tal sentido, para determinar el método estadístico a utilizar, se aplicó una **prueba de normalidad**, para identificar que los datos tengan la distribución normal. La prueba aplicada fue la de Kolmogorov – Smirnov, ya que la muestra es mayor a 50 estudiantes, se consideró un nivel de confianza del 95% y un nivel de significancia (alfa) del 5% y con el siguiente criterio de decisión.

Si $p < 0,05$ rechazamos la H_0 (hipótesis nula) y aceptamos la H_a (hipótesis alterna)

Si $p \geq 0,05$ aceptamos la H_0 (hipótesis nula) y rechazamos la H_a (hipótesis alterna).

Prueba de Normalidad de la Variable Dependiente: Diseño de Algoritmos en el Postest

Tabla 18

Prueba de Kolgomorov-Smirov para una muestra.

		Puntaje postest variable: Diseño de algoritmos
N		58
Parámetros normales ^{a,b}	Media	14,9353
	Desv. Desviación	3,55797
Máximas diferencias extremas	Absoluto	,106
	Positivo	,077
	Negativo	-,106
Estadístico de prueba		,106
Sig. asintótica(bilateral)		,099 ^c

a. La distribución de prueba es normal.

b. Se calcula a partir de datos.

c. Corrección de significación de Lilliefors.

Nota. Fuente: Procesamiento de datos SPSS.

Hipótesis de Normalidad

Ho: Los datos correspondientes al puntaje del postest de la variable dependiente tienen una distribución normal.

Ha: Los datos correspondientes al puntaje del postest de la variable dependiente no tienen una distribución normal.

Decisión y Conclusión

Como el valor de la sigmoïdal asintótica $p=0.099 > 0,05$ entonces se acepta la Ho y se rechaza la Ha, es decir los datos del postest de la variable dependiente tienen una distribución normal; por lo tanto, aplicaremos **estadística paramétrica**.

Para realizar la prueba de hipótesis, primero verificamos si los datos, tanto del grupo control como del grupo experimental, son homogéneos, para lo cual se aplicó la prueba de Levene, a los puntajes obtenidos por los estudiantes en el pretest.

Prueba de Levene

Planteamos la hipótesis

Ho: $\delta_1^2 = \delta_2^2$, es decir los grupos control y experimental son homogéneos.

Ha: $\delta_1^2 \neq \delta_2^2$, es decir los grupos control y experimental no son homogéneos.

Nivel de Significancia

alfa=0.05

Criterio de Decisión

Si $p < 0.05$ rechazamos la Ho y aceptamos la Ha.

Si $p \geq 0.05$ rechazamos la Ha y aceptamos la Ho.

Resultados y Conclusión

Tabla 19

Prueba de Levene de igualdad de varianzas.

		F	Sig.
Puntaje pretest	Se asumen varianzas iguales	,055	,815
variable: Diseño de algoritmos	No se asumen varianzas iguales		

Nota. Fuente: Procesamiento de datos SPSS.

Como $p=0.815 > 0.05$, se acepta la Hipótesis nula - H_0 y se rechaza la Hipótesis alterna - H_a , es decir el grupo experimental y el grupo control son homogéneos, pues las varianzas de los grupos son iguales.

Por lo tanto, como la distribución de los datos es normal y los grupos control y experimental son homogéneos, para contrastar la validez de la hipótesis general se aplicó la prueba paramétrica T de Student, para grupos independientes.

Prueba T de Student

Planteamos la hipótesis

H₀: $\mu_1 = \mu_2$, es decir las medias del postest del grupo control y experimental son iguales.

H_a: $\mu_1 < \mu_2$, es decir las medias del postest del grupo control y experimental son diferentes.

Nivel de Significancia

alfa=0.05

Criterio de Decisión

Si $p < 0.05$ rechazamos la H_0 y aceptamos la H_a .

Si $p \geq 0.05$ rechazamos la H_a y aceptamos la H_0 .

Resultados y Conclusión

Tabla 20

Prueba T de Student para la igualdad de medias

		t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
							Inferior	Superior
Puntaje postest variable:	Se asumen varianzas	-3,884	56	,000	-3,25000	,83667	-4,92605	-1,57395
Diseño de algoritmos	iguales							

Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Como $p = 0.000 < 0.05$, se rechaza la Hipótesis nula - H_0 y se acepta la Hipótesis alterna - H_a , es decir **las medias del grupo control y del grupo experimental en el postest son diferentes**, y esta diferencia de los puntajes de los estudiantes del grupo experimental con respecto al grupo control, se debe a la aplicación de la estrategia de enseñanza INPAL. (Hernández Sampieri et al., 2014)

Tabla 21

Media de los puntajes de la variable diseño de algoritmos.

	Grupo	N	Media
Puntaje pretest variable:	CONTROL	29	5,9310
Diseño de algoritmos	EXPERIMENTAL	29	5,4224
Puntaje postest variable:	CONTROL	29	13,3103
Diseño de algoritmos	EXPERIMENTAL	29	16,5603

Nota. Fuente: Procesamiento de datos SPSS.

Análisis y Discusión

Según la tabla de estadísticas de grupo, se puede observar que, en el postest, en ambos grupos, hubo un progreso en el aprendizaje del diseño de algoritmos, pues las medias del grupo control se incrementaron de 5.93 (pretest) a 13.31 (postest). Asimismo, las medias del grupo experimental también se incrementaron de 5.42 (pretest) a 16.56 (postest). Además, se puede notar un **mayor aprendizaje en el grupo experimental**, ya que la media del grupo experimental (16.56) es mayor a la media del grupo control (13.31); con una diferencia de 3.25 puntos, que representa un 19.63%.

Por lo tanto, concluimos que la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL), influye significativamente en el diseño de algoritmos de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020. Por tanto, **SE ACEPTA** la hipótesis estadística general.

Este resultado es similar al de [Ulker \(2020\)](#), en donde se determinó que después de la aplicación del método de 4 etapas durante un semestre, la capacidad de resolución de problemas ha aumentado en comparación con el semestre anterior, tal es así que el 85.18% de los estudiantes recomendaron el curso de Análisis y Diseño de Algoritmos.

De forma análoga, se tienen los resultados de Zhao y Li (2020), en donde después de emplear los métodos de enseñanza de algoritmos, durante casi 7 años de reforma docente, se obtuvo que las puntuaciones finales de los estudiantes mejoraron, pues eran capaces de aplicar lo aprendido, modelar problemas prácticos y diseñar algoritmos.

De manera similar, los resultados de Rojas López (2019), muestran que después de la intervención de las estrategias en el aula, se obtuvo que el porcentaje de deserción disminuyó con respecto a la mediana de los últimos 8 años, el porcentaje de acreditación

del curso de Programación aumentó y el promedio de la calificación de los estudiantes también fue mayor.

De igual forma en Trejos Buriticá (2019) determinó que la metodología propuesta permite construir soluciones a problemas comunes y cotidianos interdisciplinariamente pues vincula la programación, la lógica y las matemáticas.

De manera similar Chezzi et al. (2017), determinó que la estrategia de motivación propuesta, a través de juegos con base matemática, apoyan de manera significativa la construcción de algoritmos computacionales.

Asimismo, en Guimaraes y Gnecco (2017), se concluyó que la estrategia práctica basada en 3 etapas (etapa inicial: análisis y pruebas del problema) tuvo resultados positivos en los estudiantes, pues pudo lograr un razonamiento que permitió abstraer las sentencias del algoritmo, los datos necesarios y escenarios de pruebas, a través de un juego real.

De forma análoga Salgado Castillo et al. (2015), determinó que la aplicación de los procedimientos didácticos fue exitoso, después del experimento pedagógico realizado; pues se evidenció una influencia positiva en el perfeccionamiento del aprendizaje de la algoritmización.

De igual forma Lagos Barzola (2020), encontró que la aplicación del método de aprendizaje cooperativo tiene efectos significativos en el aprendizaje del diseño de algoritmos de los estudiantes del primer año de Ingeniería de Sistemas.

De manera similar Estrada Aro (2016), determinó que existe influencia positiva del uso del software Pseint en los niveles cognitivos del aprendizaje del curso de

Principios de Algoritmos, pero cabe indicar que en el uso del software mencionado se realiza la comprensión y análisis del problema para diseñar el algoritmo.

En concordancia con los resultados y la discusión realizada, se puede afirmar que el nivel de diseño de algoritmos mejora, cuando los estudiantes reciben una enseñanza basada en la aplicación de estrategias didácticas.

CONCLUSIONES

1. La influencia de la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” - INPAL, en el diseño de algoritmos de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020, fue significativa, pues se obtuvo un $p=0.000$ menor a 0.05 en la prueba paramétrica T de Student, lo que indica que la media de la calificación del grupo experimental fue mayor con respecto a la media de la calificación del grupo control, en un 19.63%.
2. La influencia de la aplicación de la estrategia de enseñanza INPAL, en la dimensión análisis del problema, fue significativa; pues en el grupo experimental la media del puntaje fue superior en un 15.2%, con respecto al puntaje del grupo control.
3. La influencia de la aplicación de la estrategia de enseñanza INPAL, en la dimensión análisis del problema, fue significativa; pues en el grupo experimental la media del puntaje fue superior en un 23.9%, con respecto al puntaje del grupo control.

SUGERENCIAS

- A la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, se sugiere implementar la estrategia de enseñanza “Inducción para Algoritmos” - INPAL, en la asignatura de Fundamentos de Programación, así como también en un ciclo “Cero” o de “Nivelación”, con la finalidad de que los estudiantes logren una mayor comprensión del problema y así un mejor diseño de algoritmos computacionales, lo que contribuirá a los estudiantes mejorar su rendimiento académico en el eje curricular de “Algoritmos y Programación”.
- A la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, se sugiere implementar la estrategia de enseñanza INPAL, en grupos de estudiantes que tengan dificultades en el diseño de algoritmos, con la finalidad de mejorar la comprensión del problema y con ello realizar con mayor exactitud su análisis para luego diseñar correctamente los algoritmos computacionales.
- A la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, se sugiere desarrollar algún material didáctico que incluya la estrategia de enseñanza INPAL, con la finalidad de facilitar el proceso de enseñanza - aprendizaje del diseño de algoritmos en los estudiantes, principalmente en aquellos que inician sus estudios en programación.

REFERENCIAS

- Benites, R. (2021). *La Educación Superior Universitaria en el Perú post-pandemia*.
- Brousseau, G. (2007). *Iniciación al estudio de la teoría de las situaciones didácticas*. (1ra ed.). Libros del zorzal.
- Cabanillas, R. (2019). *Investigación Educativa. Arquitectura del Proyecto de Investigación y del Informe de Tesis*. Martínez Compañón Editores S.R.L.
- Calduch Cervera, R. (2014). *Métodos y Técnicas de Investigación Internacional*. Universidad Complutense de Madrid.
- Chezzi, C. M., Salvarredi, M., Schenberger, L., Casañas, F., Giuponi, D., y Anzardi, A. (2017). *Estrategia de motivación para el razonamiento de algoritmos computacionales mediante juegos*.
- Chuquizuta Herrera, D. M. (2016). *Estrategias didácticas para facilitar el aprendizaje del diseño y construcción de algoritmos de los estudiantes del III ciclo de Computación e Informática del I.E.S.T. Público “Perú – Japón”, Chachapoyas—2015* [Universidad Nacional Pedro Ruiz Gallo].
<https://repositorio.unprg.edu.pe/handle/20.500.12893/5820>
- Deitel, P. J., y Deitel, H. M. (2008). *Cómo Programar en Java* (Séptima edición). PEARSON Educación.
- Díaz Barriga Arceo, F., y Hernández Rojas, G. (2005). *Estrategias Docentes para un Aprendizaje Significativo*. MCGRAW-HILL INTERAMERICANA S.A.

- Estrada Aro, W. M. (2016). *Software Pseint en los niveles cognitivos en estudiantes del curso Principios de Algoritmos de la Universidad Tecnológica del Perú—Lima*. [Universidad César Vallejo]. <https://repositorio.ucv.edu.pe/handle/20.500.12692/4198>
- Fuentes-Rosado, J., y Moo Medina, M. (2017). Dificultades de aprender a programar. *Revista Educación en Ingeniería*, 12, 76. <https://doi.org/10.26507/rei.v12n24.728>
- Gagné, R. M., Briggs, L. J., y Wager, W. W. (1992). *Principles of Instructional Design*. Harcourt Brace & Company.
- González Calixto, M. B., Córdoba Andrade, L., y Forero Romero, A. (2020). *Estilos de aprendizaje y autodeterminación en la educación superior*. Sello Editorial Universidad del Tolima. <https://elibro.net/es/ereader/eapisunc/228437>
- Guimaraes, M., y Gnecco, B. (2017). A practical strategy for teaching algorithms (CS1) to beginners. *International Journal of Teaching and Case Studies*, 8, 102. <https://doi.org/10.1504/IJTCS.2017.086672>
- Hernández Sampieri, R., Fernández Collado, C., y Baptista Lucio, P. (2014). *Metodología de la Investigación* (Sexta). Mc Graw Hill Education.
- Intriago Alcívar, G. C., Camacho Tovar, G. L., Sánchez Soto, M. A., Carpio Vera, D. A., y Mendiburu Rojas, A. F. (2019). *Metodología de la Investigación Educativa: Retos y perspectivas*. Editorial Académica Universitaria (Edacun). <https://elibro.net/es/ereader/eapisunc/151754>
- Joyanes Aguilar, L. (2008). *Fundamentos de Programación. Algoritmos, estructura de datos y objetos* (Cuarta). McGraw-Hill/Interamericana de España, S. A. U.

https://www.academia.edu/41489183/Fundamentos_de_programaci%C3%B3n_4ta_Edici%C3%B3n_Luis_Joyanes_Aguilar

Kolb, A. Y., y Kolb, D. A. (2005). Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education. *Academy of Management Learning & Education*, 4(2), 193-212.

Lagos Barzola, M. A. (2020). *Método de Aprendizaje Cooperativo y el Aprendizaje de Algoritmos de Programación en Estudiantes de la Universidad Nacional de Ayacucho* [Universidad Nacional de Educación Enrique Guzmán y Valle]. <https://repositorio.une.edu.pe/handle/20.500.14039/4489?show=full>

Larico Uchamaco, G. R. (2018). *El aula virtual y el aprendizaje del algoritmo en los estudiantes de la escuela profesional de ingeniería de sistemas e informática de la Universidad Nacional Amazónica de Madre de Dios, región Madre de Dios—2017* [Universidad Nacional de Educación Enrique Guzmán y Valle. Escuela de Posgrado.]. <http://repositorio.une.edu.pe/handle/20.500.14039/2607>

Mendoza Morán, V. del R., Abad Sacoto, K. Y., y Mendoza Morán, B. M. (2017). *Estrategia metodológica para mejorar el desarrollo de aplicaciones informáticas de estudiantes novatos*. <https://core.ac.uk/reader/236644816>

Mertler, C. A. (s. f.). *Designing scoring rubrics for your classroom*. <https://doi.org/10.7275/GCY8-0W24>

Polya, G. (1989). *Cómo plantear y resolver problemas*. Trillas S.A. de C.V. <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbmxtaXBsYXRhZm9ybWFIZHVjYXRpdmF8Z3g6MmMxMzJlZDBmNDQyYmJkNQ>

- Reza Becerril, F. (1997). *Ciencia, metodología e investigación*. Pearson Educación.
- Rojas López, A. (2019). *Escenarios de aprendizaje personalizados a partir de la evaluación del pensamiento computacional para el aprendizaje de competencias de programación mediante un entorno b-Learning y gamificación*.
<https://gredos.usal.es/handle/10366/140444>
- Russo, C., Lencina, P., Ado, M., Iglesias, P., Ahmad, T., y Sarobe, M. (2018). *Estrategias educativas en la enseñanza de la programación*. 8.
- Salgado Castillo, A., Alonso Berenguer, I., y Gorina Sánchez, A. (2015). Una propuesta didáctica para perfeccionar la algoritmización computacional. *Didasc@lia: Didáctica y Educación*, 6(4 (Octubre-Diciembre)), 149-160.
- Siemens, G. (2004). *Conectivismo: Una teoría de aprendizaje para la era digital*.
- SINEACE. (2018). *Diseño o actualización curricular bajo el enfoque por competencias para instituciones de educación superior*.
- Trejos Buriticá, O. I. (2019). Metodología de aprendizaje para descomponer números naturales en factores primos usando Programación Funcional. *I+D Revista de Investigaciones*, 13, 26-37. <https://doi.org/10.33304/revinv.v13n1-2019003>
- Ulker, E. D. (2020). The effect of applying 4-stages on learning analysis and design of algorithms. *Cypriot Journal of Educational Sciences*, 15(5), Article 5.
<https://doi.org/10.18844/cjes.v15i5.4621>
- Valderrama Mendoza, S. (2002). *Pasos para elaborar proyectos y tesis de investigación científica*. Editorial San Marcos.

Vasconcelos Santillán, J. (2018). *Introducción a la computación*. Grupo Editorial Patria.

<https://elibro.net/es/ereader/eapisunc/98314>

Velasco Ramírez, M. L. (2020). Resolución de problemas algorítmicos y objetos de aprendizaje: Una revisión de la literatura. *RIDE Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 10(20).

<https://doi.org/10.23913/ride.v10i20.630>

Zhao, J., y Li, M. (2020). *Teaching Methods Reform of “Algorithm Design and Analysis”*

Course. <https://doi.org/10.2991/assehr.k.201214.008>

APÉNDICES / ANEXOS

Apéndice 1. Instrumentos de recolección de datos

Evaluación escrita en el pretest

Universidad Nacional de Cajamarca - Escuela de Postgrado
Programa de Doctorado - Mención Educación



EXAMEN DE DISEÑO DE ALGORITMOS DE FUNDAMENTOS DE PROGRAMACIÓN (PRETEST)

ESTUDIANTE: FECHA: .../.../2020

Indicaciones:

Desarrollar dos de los siguientes ejercicios, considerando las siguientes fases en la resolución de problemas con la computadora:

a) Análisis del problema:

- Variables del problema.
- Variables de entrada.
- Variables de salida.
- Proceso.

b) Diseño del Algoritmo: (sólo 1 método)

- Declaración de variables
- Entrada de datos
- Proceso
- Salida de resultados

Ejercicios:

1. Elabore un algoritmo que permita calcular la suma de los n primeros números naturales.
2. Al sueldo bruto de un empleado del área de ventas se le descuenta el 12% por AFP y como bonificación recibe S/. 30.00 más por cada venta mayor a S/. 100.00 que haya realizado durante el mes. Elabore un algoritmo que permita calcular su sueldo neto.

Evaluación escrita en el postest

Universidad Nacional de Cajamarca - Escuela de Postgrado
Programa de Doctorado - Mención Educación



EXAMEN DE DISEÑO DE ALGORITMOS DE FUNDAMENTOS DE PROGRAMACIÓN (POSTEST)

ESTUDIANTE: FECHA: .../.../2020

Indicaciones:

Desarrollar dos de los siguientes ejercicios (asignados por la docente), considerando las siguientes fases en la resolución de problemas con la computadora:

a) Análisis del problema:

- Variables del problema.
- Variables de entrada.
- Variables de salida.
- Proceso.

b) Diseño del Algoritmo: (sólo 1 método)

- Declaración de variables
- Entrada de datos
- Proceso
- Salida de resultados

Ejercicios

1. En un curso de programación, la calificación final del estudiante se determina a partir del rendimiento en tres aspectos. Existe un promedio de trabajos domiciliarios con el 20%, una práctica calificada con el 30% y el examen final con 50%. Tomar en cuenta que se presentó 2 trabajos domiciliarios. Diseñar un algoritmo para calcular la calificación final del estudiante.
2. Un inversionista está pensando invertir un cierto capital en soles a una determinada tasa de interés simple anual. ¿Cuál será el capital final y el tiempo en años y meses que necesita para tener un 25% más sobre su capital inicial?. (Ejemplo: Tiempo que necesita = 1 año 3 meses)
3. Un estudiante ingresa a una cabina de internet a las X horas y M minutos y sale a las Y horas y N minutos. En turno diurno (a partir de las 8:00 am), el costo por hora es de S/. 2.00 y en turno noche (a partir de las 8:00 pm), el costo por hora es S/1.50. Calcular el tiempo que estuvo en las cabinas en horas y minutos e indicar el monto a pagar. Considerar sólo la hora de inicio de ingreso a la cabina, para determinar el costo del turno. (Ejemplo si ingresa a las 14:30 y sale a las 15:40, el tiempo que estuvo en las cabinas sería 1 hora, 10 min).
4. Si compro A libros a X euros cada uno, B lapiceros a Y dólares cada uno y C plumones a Z marcos. ¿Cuánto debo pagar por libros, por lapiceros y por plumones y cuanto es el pago Total en soles? y ¿Entre lapiceros y plumones, por qué producto pago más?
(1 dólar = 3.27 soles, 1 dólar = 1.66 marcos, 1 dólar = 0.85 euros)

Rúbrica de Evaluación del Diseño de Algoritmos

Resultado de Aprendizaje: Construye algoritmos con estructura secuencial y/o condicional, considerando los elementos básicos, el análisis del problema y el diseño de la solución, para posteriormente realizar la prueba de escritorio e implementación en un lenguaje de programación.

Tipo de evaluación		Examen escrito				
Aspecto	Criterio de desempeño	Niveles de desempeño				
		Deficiente	Insuficiente	Regular	Bueno	Excelente
Análisis del Problema 50%	1. Define las variables del problema. (10%)	No define las variables del problema y constantes de ser necesario. (0 pts)	Define pocas variables del problema y constantes de ser necesario. (0.5 pts)	Define parcialmente las variables del problema y constantes de ser necesario. (1 pts)	Define la mayoría de las variables del problema y constantes de ser necesario. (1.5 pts)	Define completamente las variables del problema y constantes de ser necesario. (2 pts)
	2. Identifica las variables de entrada. (10%)	No identifica las variables de entrada. (0 pts)	Identifica pocas variables de entrada. (0.5 pts)	Identifica parcialmente las variables de entrada. (1 pts)	Identifica la mayoría de las variables de entrada. (1.5 pts)	Identifica completamente las variables de entrada. (2 pts)
	3. Identifica las variables de salida. (10%)	No identifica las variables de salida. (0 pts)	Identifica pocas variables de salida. (0.5 pts)	Identifica parcialmente las variables de salida. (1 pts)	Identifica la mayoría de las variables de salida. (1.5 pts)	Identifica completamente las variables de salida. (2 pts)
	4. Identifica el proceso para obtener la salida deseada. (20%)	No identifica el proceso para obtener la salida deseada. (0 pts)	Identifica inicialmente el proceso para obtener la salida deseada. (1 pts)	Identifica parcialmente el proceso para obtener la salida deseada. (2 pts)	Identifica la mayoría del proceso para obtener la salida deseada. (3 pts)	Identifica completamente el proceso para obtener la salida deseada. (4 pts)
Diseño de la Solución Algorítmica 50%	5. Establece la declaración de variables. (10%)	No establece la declaración de variables. (0 pts)	Establece pocas variables con algunos tipos de datos correctos y constantes de ser necesario. (0.5 pts)	Establece parcialmente las variables con algunos tipos de datos correctos y constantes de ser necesario. (1 pts)	Establece la mayoría de las variables con su respectivo tipo de dato y constantes de ser necesario. (1.5 pts)	Establece completamente las variables con su respectivo tipo de dato y constantes de ser necesario. (2 pts)
	6. Establece la entrada de datos. (10%)	No establece la entrada de datos. (0 pts)	Establece inicialmente la entrada de datos. (0.5 pts)	Establece parcialmente la entrada de datos. (1 pts)	Establece la mayoría de la entrada de datos. (1.5 pts)	Establece completamente la entrada de datos. (2 pts)
	7. Establece el proceso a seguir. (20%)	No establece el proceso a seguir. (0 pts)	Establece inicialmente el proceso a seguir. (1 pts)	Establece parcialmente el proceso a seguir. (2 pts)	Establece la mayoría del proceso a seguir. (3 pts)	Establece completamente el proceso a seguir. (4 pts)
	8. Establece la salida de resultados. (10%)	No establece la salida de resultados. (0 pts)	Establece inicialmente la salida de resultados. (0.5 pts)	Establece parcialmente la salida de resultados. (1 pts)	Establece la mayoría de la salida de resultados. (1.5 pts)	Establece completamente la salida de resultados. (2 pts)

Rúbrica de Evaluación de la Estrategia de Enseñanza INPAL para el Diseño de Algoritmos

Resultado de Aprendizaje: Construye algoritmos con estructura secuencial y/o condicional, considerando los elementos básicos, el análisis del problema y el diseño de la solución, para posteriormente realizar la prueba de escritorio e implementación en un lenguaje de programación.

Tipo de evaluación		Examen escrito		
Aspecto	Criterio de desempeño	Niveles de desempeño		
		Deficiente	Regular	Bueno
Solución matemática del problema 60%	1. Identifica los valores de entrada. (20%)	Nula o poca identificación de los valores de entrada. (1 pto)	Identifica parcialmente los valores de entrada. (2 pts)	Identifica completamente los valores de entrada. (4 pts)
	2. Aplica el proceso o fórmula para resolver el problema. (20%)	No aplica o sólo inicia el proceso para resolver el problema. (1 pto)	Aplica parcialmente el proceso para resolver el problema. (2 pts)	Aplica correctamente el proceso para resolver el problema. (4 pts)
	3. Obtiene los valores de salida o resultados. (20%)	No obtiene los resultados del problema (0 pts)	Obtiene los resultados del problema, pero son incorrectos. (2 pts)	Obtiene la respuesta correcta. (4 pts)
Asociación de solución matemática con los elementos del algoritmo 40%	4. Relaciona los datos desconocidos con las variables del problema. (10%)	Nula o poca relación de los datos desconocidos con las variables del problema. (0.5 pts)	Relaciona parcialmente los datos desconocidos con las variables del problema. (1 pto)	Relaciona correctamente la mayoría de los datos desconocidos con las variables del problema. (2 pts)
	5. Relaciona los valores de entrada con las variables de entrada. (10%)	Nula o poca relación de los valores de entrada con las variables de entrada. (0.5 pts)	Relaciona parcialmente los valores de entrada con las variables de entrada. (1 pto)	Relaciona correctamente los valores de entrada con las variables de entrada. (2 pts)
	6. Relaciona los valores de salida con las variables de salida. (10%)	Nula o poca relación de los valores de salida con las variables de salida. (0.5 pts)	Relaciona parcialmente los valores de salida con las variables de salida. (1 pto)	Relaciona correctamente los valores de salida con las variables de salida. (2 pts)
	7. Relaciona el proceso matemático con el proceso algorítmico que produce la salida deseada. (10%)	Nula o poca relación del proceso matemático con el proceso algorítmico. (0.5 pts)	Relaciona parcialmente el proceso matemático con el proceso algorítmico. (1 pto)	Relaciona correctamente el proceso matemático con el proceso algorítmico. (2 pts)

Encuesta sobre la Aplicación de la Estrategia INPAL en el Proceso de Aprendizaje del Diseño de Algoritmos

Instrucciones

Estimado(a) estudiante, la presente encuesta sirve para realizar una investigación sobre el grado de contribución de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL) en el proceso de aprendizaje del diseño de algoritmos. Su participación no influye en ninguna nota, mas es de suma importancia contar con sus respuestas y con la mayor sinceridad posible.

Lea con atención las siguientes preguntas, y responda encerrando con un círculo su respuesta, según su propio juicio.

Objetivo: Conocer en qué medida la estrategia INPAL, apoyó a los estudiantes en el análisis del problema y en el diseño del algoritmo.

1. La estrategia INPAL le ayudó en el ANÁLISIS DEL PROBLEMA.

- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |

2. La estrategia INPAL le ayudó a identificar las variables del problema:

- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |

3. La estrategia INPAL le ayudó a identificar las variables de entrada:

- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |

4. La estrategia INPAL le ayudó a identificar las variables de salida:

- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |

5. La estrategia INPAL le ayudó a identificar el proceso para la solución del problema:
- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |
6. La estrategia INPAL le ayudó en el DISEÑO DEL ALGORITMO:
- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |
7. La estrategia INPAL le ayudó a establecer la declaración de variables:
- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |
8. La estrategia INPAL le ayudó a establecer la entrada de datos:
- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |
9. La estrategia INPAL le ayudó a establecer el proceso a seguir:
- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |
10. La estrategia INPAL le ayudó a establecer la salida de resultados:
- | | |
|-------------|-------------|
| 1) Muy poco | 4) Mucho |
| 2) Poco | 5) Bastante |
| 3) A medias | |

Gracias por su Colaboración.

Apéndice 2. Confiabilidad de los instrumentos

Cálculo de la Confiabilidad con Alfa de Cronbach

N°	Estudiante	1. Define las variables del problema. (2ptos)	2. Identifica las variables de entrada. (2ptos)	3. Identifica las variables de salida (2ptos)	4. Identifica el proceso para obtener la salida deseada. (4ptos)	5. Establece la declaración de variables (2ptos)	6. Establece la entrada de datos (2ptos)	7. Establece el proceso a seguir (4ptos)	8. Establece la salida de resultados (2ptos)	Suma			
		Preg1	Preg1	Preg1	Preg1	Preg1	Preg1	Preg1	Preg1	Preg1			
1	ESTPI01	1.5	2	2	4	0.5	2	4	2	18	Alfa	α	0.7796
2	ESTPI02	2	2	2	4	1	2	4	2	19	Número de items	K	8
3	ESTPI03	2	2	2	4	0.5	2	4	2	18.5	Varianza de cada item	Vi	4.7972
4	ESTPI04	0.5	2	2	4	0.5	2	4	2	17	Varianza total	Vt	15.0931
5	ESTPI05	0.5	2	2	3	2	2	3	2	16.5			
6	ESTPI06	1.5	2	2	2	1.5	2	2	2	15			
7	ESTPI07	1.5	0.5	0.5	1	0.5	1.5	3	2	10.5			
8	ESTPI08	1.5	0.5	0.5	1	0.5	1.5	3	2	10.5			
9	ESTPI09	0.5	0.5	0.5	1	0.5	2	1	2	8			
10	ESTPI10	1.5	0.5	0.5	1	0.5	1.5	3	2	10.5			
11	ESTPI11	0.5	0.5	0.5	1	1.5	2	3	2	11			
12	ESTPI12	2	2	2	1	0.5	2	1	2	12.5			
13	ESTPI13	2	2	2	4	2	2	4	2	20			
14	ESTPI14	2	2	2	4	0.5	2	4	2	18.5			
		0.36352041	0.51658163	0.51658163	1.96428571	0.32780612	0.04209184	1.06632653	0				

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum Vi}{Vt} \right]$$

N°	Estudiante	1. Define las variables del problema. (2ptos)	2. Identifica las variables de entrada. (2ptos)	3. Identifica las variables de salida (2ptos)	4. Identifica el proceso para obtener la salida deseada. (4ptos)	5. Establece la declaración de variables (2ptos)	6. Establece la entrada de datos (2ptos)	7. Establece el proceso a seguir (4ptos)	8. Establece la salida de resultados (2ptos)	Suma			
		Preg2	Preg2	Preg2	Preg2	Preg2	Preg2	Preg2	Preg2	Preg2		Alfa	α
1	ESTPI01	2	2	2	2	0.5	2	4	2	16.5	Número de items	K	8
2	ESTPI02	2	2	2	1	1.5	2	4	2	16.5	Varianza de cada item	Vi	5.8750
3	ESTPI03	2	2	2	4	0.5	2	4	2	18.5	Varianza total	Vt	26.2041
4	ESTPI04	1.5	1.5	2	1	0.5	1.5	3	2	13			
5	ESTPI05	0.5	0.5	2	3	2	0.5	2	2	12.5			
6	ESTPI06	2	2	2	4	0.5	2	4	2	18.5			
7	ESTPI07	2	0.5	0.5	1	0.5	1	2	0.5	8			
8	ESTPI08	2	0.5	0.5	1	0.5	1	3	0.5	9			
9	ESTPI09	0	0	0	0	0	0	0	0	0			
10	ESTPI10	2	0.5	0.5	1	0.5	1.5	4	2	12	Coeficiente de confiabilidad promedio	<u>0.83312</u>	
11	ESTPI11	2	0.5	0.5	1	0.5	1	2	0.5	8			
12	ESTPI12	2	1	2	2	0.5	1.5	2	2	13			
13	ESTPI13	2	0.5	0.5	3	1	1.5	4	2	14.5			
14	ESTPI14	0.5	0.5	0.5	1	0.5	0.5	1	0.5	5			
		0.4706633	0.5	0.6326531	1.454081633	0.235969388	0.382653061	1.596938776	0.60204082				

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum Vi}{Vt} \right]$$

Apéndice 3. Validación de instrumentos de recolección de datos a través de juicio de expertos.

VALIDACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS (JUICIO DE EXPERTOS)

Yo, Carlos Enrique Aparicio Arteaga, identificado con DNI N° 18071268, con Grado Académico de Doctor en Ingeniería de Sistemas de la Universidad Nacional de Piura.

Hago constar que he leído y revisado los siguientes instrumentos correspondientes a la Tesis de Doctorado: Aplicación de la estrategia de enseñanza "Inducción para Algoritmos" en el diseño de algoritmos de la asignatura "Fundamentos de Programación" de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020, de la doctoranda Lisi Janet Vásquez Fernández.

- Rúbrica de evaluación de diseño de algoritmos (grupo experimental y grupo control), con 8 ítems.
- Rúbrica para la evaluación de la estrategia de enseñanza INPAL para el diseño de algoritmos (grupo experimental), con 7 ítems.
- Encuesta sobre la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos (grupo experimental), con 11 ítems.

Luego de la evaluación de cada ítem y realizada las correcciones respectivas, los resultados son los siguientes:

N°	Instrumento	N° de ítems revisados	N° de ítems válidos	% de ítems válidos
1	Rúbrica de evaluación de diseño de algoritmos.	8	8	100%
2	Rúbrica para la evaluación de la estrategia de enseñanza INPAL para el diseño de algoritmos.	7	7	100%
3	Encuesta sobre la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos.	11	11	100%

Lugar y Fecha: Cajamarca, 29 de julio del 2020.

Apellidos y Nombres del evaluador: Aparicio Arteaga, Carlos Enrique



FIRMA DEL EVALUADOR

**VALIDACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS
(JUICIO DE EXPERTOS)**

Yo Edwin Valencia Castillo , identificado con DNI N°26696858, con Grado Académico de Doctor en Ingeniería de Sistemas, Universidad Nacional Federico Villareal.

Hago constar que he leído y revisado los siguientes instrumentos correspondientes a la Tesis de Doctorado: Aplicación de la estrategia de enseñanza “Inducción para Algoritmos” en el diseño de algoritmos de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020, de la doctoranda Lisi Janet Vásquez Fernández.

- Rúbrica de evaluación de diseño de algoritmos (grupo experimental y grupo control), con 8 ítems.
- Rúbrica para la evaluación de la estrategia de enseñanza INPAL para el diseño de algoritmos (grupo experimental), con 7 ítems.
- Encuesta sobre la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos (grupo experimental), con 11 ítems.

Luego de la evaluación de cada ítem y realizada las correcciones respectivas, los resultados son los siguientes:

N°	Instrumento	N° de ítems revisados	N° de ítems válidos	% de ítems válidos
1	Rúbrica de evaluación de diseño de algoritmos.	8	8	100%
2	Rúbrica para la evaluación de la estrategia de enseñanza INPAL para el diseño de algoritmos.	7	7	100%
3	Encuesta sobre la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos.	11	11	100%

Lugar y Fecha: Cajamarca 3 de agosto de 2020.

Apellidos y Nombres del evaluador: Valencia Castillo Edwin Alberto.


FIRMA DEL EVALUADOR

**VALIDACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS
(JUICIO DE EXPERTOS)**

Yo Rosel Burga Cabrera, identificado con DNI N° 26612952, con Grado Académico de Doctor en Ingeniería de Sistemas, Universidad de Nacional de Piura.

Hago constar que he leído y revisado los siguientes instrumentos correspondientes a la Tesis de Doctorado: Aplicación de la estrategia de enseñanza “Inducción para Algoritmos” en el diseño de algoritmos de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020, de la doctoranda Lisi Janet Vásquez Fernández.

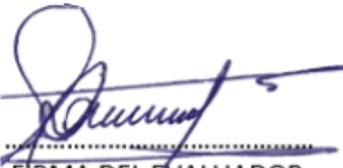
- Rúbrica de evaluación de diseño de algoritmos (grupo experimental y grupo control), con 8 ítems.
- Rúbrica para la evaluación de la estrategia de enseñanza INPAL para el diseño de algoritmos (grupo experimental), con 7 ítems.
- Encuesta sobre la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos (grupo experimental), con 11 ítems.

Luego de la evaluación de cada ítem y realizada las correcciones respectivas, los resultados son los siguientes:

N°	Instrumento	N° de ítems revisados	N° de ítems válidos	% de ítems válidos
1	Rúbrica de evaluación de diseño de algoritmos.	8	8	100%
2	Rúbrica para la evaluación de la estrategia de enseñanza INPAL para el diseño de algoritmos.	7	7	100%
3	Encuesta sobre la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos.	11	11	100%

Lugar y Fecha: Cajamarca, 07 de agosto de 2020

Apellidos y Nombres del evaluador: Burga Cabrera Rosel


FIRMA DEL EVALUADOR

**VALIDACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS
(JUICIO DE EXPERTOS)**

Yo, MANUEL ENRIQUE MALPICA RODRÍGUEZ, identificado con DNI N° 26707158, con Grado Académico de Doctor en Ingeniería de Sistemas de la Universidad Nacional de Piura.

Hago constar que he leído y revisado los siguientes instrumentos correspondientes a la Tesis de Doctorado: Aplicación de la estrategia de enseñanza "Inducción para Algoritmos" en el diseño de algoritmos de la asignatura "Fundamentos de Programación" de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020, de la doctoranda Lisi Janet Vásquez Fernández.

- Rúbrica de evaluación de diseño de algoritmos (grupo experimental y grupo control), con 8 ítems
- Rúbrica para la evaluación de la estrategia de enseñanza INPAL para el diseño de algoritmos (grupo experimental), con 7 ítems.
- Encuesta sobre la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos (grupo experimental), con 11 ítems.

Luego de la evaluación de cada ítem y realizada las correcciones respectivas, los resultados son los siguientes:

N°	Instrumento	N° de ítems revisados	N° de ítems válidos	% de ítems válidos
1	Rúbrica de evaluación de diseño de algoritmos.	8	8	100%
2	Rúbrica para la evaluación de la estrategia de enseñanza INPAL para el diseño de algoritmos.	7	7	100%
3	Encuesta sobre la aplicación de la estrategia de enseñanza INPAL en el proceso de aprendizaje del diseño de algoritmos.	11	11	100%

Lugar y Fecha: Cajamarca, 29 de Julio del 2020.

Apellidos y Nombres del evaluador: Manuel Enrique Malpica Rodríguez


FIRMA DEL EVALUADOR

Apéndice 4. Matriz general de resultados en Excel

		ANÁLISIS DEL PROBLEMA		SOLUCIÓN ALGORÍTMICA		VD. DISEÑO DEL ALGORITMO	
		Pretest	Postest	Pretest	Postest	Pretest	Postest
GRUPO	CODIGO	TOTPRET_DAP	TOTPOST_DAP	TOTPRET_DDA	TOTPOST_DDA	TOTPRET_VDA	TOTPOST_VDA
GR_CONTROL	ESTCO01	14.50	17.00	0.00	13.00	7.25	15
GR_CONTROL	ESTCO02	11.00	15.50	0.00	13.00	5.5	14.25
GR_CONTROL	ESTCO03	4.00	11.00	7.00	8.50	5.5	9.75
GR_CONTROL	ESTCO04	14.00	16.00	0.00	14.00	7	15
GR_CONTROL	ESTCO05	4.50	14.00	12.00	13.50	8.25	13.75
GR_CONTROL	ESTCO06	0.00	13.00	5.00	11.00	2.5	12
GR_CONTROL	ESTCO07	8.00	13.50	15.00	12.00	11.5	12.75
GR_CONTROL	ESTCO08	16.00	15.00	15.00	14.00	15.5	14.5
GR_CONTROL	ESTCO09	5.00	13.00	10.50	12.00	7.75	12.5
GR_CONTROL	ESTCO10	8.00	0.00	8.50	14.50	8.25	7.25
GR_CONTROL	ESTCO11	14.00	17.00	15.00	14.50	14.5	15.75
GR_CONTROL	ESTCO12	12.00	12.50	14.00	12.00	13	12.25
GR_CONTROL	ESTCO13	5.50	9.00	8.50	6.00	7	7.5
GR_CONTROL	ESTCO14	2.00	17.50	5.50	19.00	3.75	18.25
GR_CONTROL	ESTCO15	0.00	17.00	5.50	17.00	2.75	17
GR_CONTROL	ESTCO16	0.00	14.00	6.50	15.50	3.25	14.75

		ANÁLISIS DEL PROBLEMA		SOLUCIÓN ALGORÍTMICA		VD. DISEÑO DEL ALGORITMO	
		Pretest	Postest	Pretest	Postest	Pretest	Postest
GRUPO	CODIGO	TOTPRET_DAP	TOTPOST_DAP	TOTPRET_DDA	TOTPOST_DDA	TOTPRET_VDA	TOTPOST_VDA
GR_CONTROL	ESTCO17	1.00	17.00	6.50	16.50	3.75	16.75
GR_CONTROL	ESTCO18	7.50	15.00	6.00	14.00	6.75	14.5
GR_CONTROL	ESTCO19	0.00	17.00	9.00	17.00	4.5	17
GR_CONTROL	ESTCO20	4.50	12.00	5.50	11.00	5	11.5
GR_CONTROL	ESTCO21	9.00	8.00	0.00	3.00	4.5	5.5
GR_CONTROL	ESTCO22	0.00	7.50	6.50	6.50	3.25	7
GR_CONTROL	ESTCO23	0.00	20.00	6.00	20.00	3	20
GR_CONTROL	ESTCO24	0.00	15.00	5.00	12.50	2.5	13.75
GR_CONTROL	ESTCO25	0.00	17.50	4.50	18.00	2.25	17.75
GR_CONTROL	ESTCO26	0.00	18.00	4.00	17.50	2	17.75
GR_CONTROL	ESTCO27	0.00	13.50	5.50	12.50	2.75	13
GR_CONTROL	ESTCO28	5.50	12.50	4.50	8.00	5	10.25
GR_CONTROL	ESTCO29	3.00	13.00	4.00	5.00	3.5	9
GR_EXPERIM	ESTEX01	7.50	15.50	0.00	14.50	3.75	15
GR_EXPERIM	ESTEX02	13.00	19.50	5.00	19.50	9	19.5
GR_EXPERIM	ESTEX03	17.00	16.00	15.00	17.00	16	16.5
GR_EXPERIM	ESTEX04	13.00	14.50	4.50	15.50	8.75	15
GR_EXPERIM	ESTEX05	10.50	18.50	8.50	20.00	9.5	19.25
GR_EXPERIM	ESTEX06	8.00	12.00	10.00	12.00	9	12
GR_EXPERIM	ESTEX07	5.50	14.50	3.50	14.50	4.5	14.5
GR_EXPERIM	ESTEX08	4.00	14.50	4.00	16.50	4	15.5
GR_EXPERIM	ESTEX09	2.00	14.00	4.00	15.00	3	14.5

GRUPO	CODIGO	ANÁLISIS DEL PROBLEMA		SOLUCIÓN ALGORÍTMICA		VD. DISEÑO DEL ALGORITMO	
		Pretest	Postest	Pretest	Postest	Pretest	Postest
		TOTPRET_DAP	TOTPOST_DAP	TOTPRET_DDA	TOTPOST_DDA	TOTPRET_VDA	TOTPOST_VDA
GR_EXPERIM	ESTEX10	4.00	12.00	6.00	11.50	5	11.75
GR_EXPERIM	ESTEX11	16.00	7.50	10.00	17.50	13	12.5
GR_EXPERIM	ESTEX12	15.00	15.50	0.00	13.50	7.5	14.5
GR_EXPERIM	ESTEX13	5.00	18.50	4.50	19.00	4.75	18.75
GR_EXPERIM	ESTEX14	0.00	18.50	7.00	18.50	3.5	18.5
GR_EXPERIM	ESTEX15	5.50	18.00	8.50	18.00	7	18
GR_EXPERIM	ESTEX16	0.00	18.00	5.50	18.00	2.75	18
GR_EXPERIM	ESTEX17	0.00	18.00	5.00	18.00	2.5	18
GR_EXPERIM	ESTEX18	0.00	18.50	3.50	18.50	1.75	18.5
GR_EXPERIM	ESTEX19	7.50	18.50	0.00	20.00	3.75	19.25
GR_EXPERIM	ESTEX20	17.00	16.00	0.00	16.00	8.5	16
GR_EXPERIM	ESTEX21	7.00	18.00	3.00	18.50	5	18.25
GR_EXPERIM	ESTEX22	0.00	17.50	1.50	17.50	0.75	17.5
GR_EXPERIM	ESTEX23	0.00	19.00	4.50	19.00	2.25	19
GR_EXPERIM	ESTEX24	0.00	15.50	5.50	15.50	2.75	15.5
GR_EXPERIM	ESTEX25	0.00	20.00	9.00	20.00	4.5	20
GR_EXPERIM	ESTEX26	6.00	16.50	5.00	18.00	5.5	17.25
GR_EXPERIM	ESTEX27	0.00	16.50	7.00	17.00	3.5	16.75
GR_EXPERIM	ESTEX28	0.00	19.50	4.50	19.50	2.25	19.5
GR_EXPERIM	ESTEX29	0.00	12.50	6.50	9.50	3.25	11

**Apéndice 5. Programa sesiones de aprendizaje con la estrategia de enseñanza
INPAL, para el diseño de algoritmos computacionales.**

PLAN DE SESIÓN DE APRENDIZAJE N° 01 (Algoritmos con Estructura Secuencial)			
I. DATOS GENERALES			
Carrera: Ingeniería de Sistemas Ciclo: I Asignatura: Fundamentos de Programación Unidad I: Introducción - Algoritmos y Programación - Estructura Secuencial. Tema: Diseño de algoritmos con estructura secuencial, usando la estrategia de enseñanza INPAL. Tipo de sesión: Teórica Tiempo por sesión: 2 horas 15 minutos			
II. RESULTADOS DE APRENDIZAJE			
Resultado de aprendizaje de la Unidad Construye algoritmos y programas con estructura secuencial, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación. Resultado de aprendizaje de la Sesión Comprende cómo diseñar un algoritmo con estructura secuencial.			
III. SECUENCIA DIDÁCTICA			
FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Introducción del tema, a través de la explicación de la estructura general de un algoritmo de la vida diaria, con estructura secuencial.	Pizarra	15 min
Desarrollo	<ul style="list-style-type: none"> - Explicación de la estructura general de un algoritmo con estructura secuencial, mediante una presentación de diapositivas, por parte del docente. - Desarrollo de un primer ejercicio con estructura secuencial, mediante las fases del diseño de algoritmos, aplicando la estrategia de enseñanza INPAL. - Desarrollo de un ejercicio con estructura secuencial, por parte de los estudiantes, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. 	Diapositivas SIA (subir los ejercicios desarrollados por parte de los estudiantes)	90 min
Cierre	Presentación y explicación de un ejercicio desarrollado, por parte de un estudiante, el cual permitirá participar con consultas o aportes por parte del resto de estudiantes de la clase, y así dar las conclusiones finales por parte del docente.	Pizarra	30 min

PLAN DE SESIÓN DE APRENDIZAJE N° 02
(Algoritmos con Estructura Secuencial)

I. DATOS GENERALES

Carrera: Ingeniería de Sistemas
Ciclo: I
Asignatura: Fundamentos de Programación
Unidad I: Introducción - Algoritmos y Programación - Estructura Secuencial.
Tema: Diseño de algoritmos con estructura secuencial, usando la estrategia de enseñanza INPAL.
Tipo de sesión: Práctica
Tiempo por sesión: 2 horas 15 minutos

II. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad
 Construye algoritmos y programas con estructura secuencial, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.
Resultado de aprendizaje de la Sesión
 Diseña algoritmos con estructura secuencial.

III. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Participación de los estudiantes sobre lo aprendido en la sesión anterior, para lo cual responde a las siguientes preguntas: ¿En qué consisten los algoritmos con estructura secuencial? ¿Mencione algunos ejemplos de problemas que se pueden resolver con estructura secuencial?	Pizarra	15 min
Desarrollo	<ul style="list-style-type: none"> - Repaso de la estructura general de un algoritmo con estructura secuencial, mediante el desarrollo de un ejercicio con estructura secuencial, aplicando la estrategia de enseñanza INPAL. - Presentación de una práctica 01 - nivel 1, con ejercicios propuestos sobre algoritmos con estructura secuencial. - Conformación de equipos en pares de estudiantes, para resolver 2 ejercicios de la práctica, elegidos por el par de estudiantes. - Desarrollo de los 2 ejercicios, con estructura secuencial, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Explicación de los algoritmos planteados por cada estudiante, a su compañero de equipo, con la finalidad de compartir la solución algorítmica encontrada por cada estudiante y así determinar su eficacia. 	Diapositivas Practica 01 (nivel 1) - Estructura secuencial SIA (subir los ejercicios desarrollados por parte de los estudiantes)	90 min
Cierre	Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de dar alternativas que permitan optimizar las soluciones encontradas, con el apoyo del docente.	Pizarra	30 min

PLAN DE SESIÓN DE APRENDIZAJE N° 03
(Algoritmos con Estructura Secuencial)

I. DATOS GENERALES

Carrera: Ingeniería de Sistemas
Ciclo: I
Asignatura: Fundamentos de Programación
Unidad I: Introducción - Algoritmos y Programación - Estructura Secuencial.
Tema: Diseño de algoritmos con estructura secuencial, usando la estrategia de enseñanza INPAL.
Tipo de sesión: Teórica
Tiempo por sesión: 2 horas 15 minutos

II. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad

Construye algoritmos y programas con estructura secuencial, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Resultado de aprendizaje de la Sesión

Comprende cómo diseñar un algoritmo con estructura secuencial de mayor complejidad.

III. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Lluvia de ideas de los estudiantes, de cómo solucionarían un nuevo ejercicio con estructura secuencial de mayor complejidad, presentado por el docente.	Diapositiva	15 min
Desarrollo	<ul style="list-style-type: none"> - Explicación de nuevos operadores y uso de variables intermedias en algoritmos con estructura secuencial, mediante una presentación de diapositivas, por parte del docente. - Desarrollo de un ejercicio con mayor complejidad, con estructura secuencial, mediante las fases del diseño de algoritmos, aplicando la estrategia de enseñanza INPAL. - Desarrollo de un ejercicio con mayor complejidad, con estructura secuencial, por parte de los estudiantes, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. 	Diapositivas SIA (subir los ejercicios desarrollados por parte de los estudiantes)	90 min
Cierre	Presentación y explicación de un ejercicio desarrollado, por parte de un estudiante, el cual permitirá participar con consultas o aportes por parte del resto de estudiantes de la clase, y así dar las conclusiones finales por parte del docente.	Pizarra	30 min

PLAN DE SESIÓN DE APRENDIZAJE N° 04
(Algoritmos con Estructura Secuencial)

I. DATOS GENERALES

Carrera: Ingeniería de Sistemas
Ciclo: I
Asignatura: Fundamentos de Programación
Unidad I: Introducción - Algoritmos y Programación - Estructura Secuencial.
Tema: Diseño de algoritmos con estructura secuencial, usando la estrategia de enseñanza INPAL.
Tipo de sesión: Práctica
Tiempo por sesión: 2 horas 15 minutos

II. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad

Construye algoritmos y programas con estructura secuencial, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Resultado de aprendizaje de la Sesión

Diseña algoritmos con estructura secuencial de mayor complejidad.

III. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Participación de los estudiantes sobre lo aprendido en la sesión anterior, para lo cual responde a las siguientes preguntas: ¿Qué nuevos operadores pueden emplear en los algoritmos con estructura secuencial? ¿Mencione algunos ejemplos de problemas de mayor complejidad que se pueden resolver con estructura secuencial?	Pizarra	15 min
Desarrollo	<ul style="list-style-type: none"> - Reforzamiento de la estructura general de un algoritmo con estructura secuencial de mayor complejidad, mediante el desarrollo de un ejercicio con estructura secuencial, aplicando la estrategia de enseñanza INPAL. - Presentación de una práctica 01 - nivel 2, con ejercicios propuestos sobre algoritmos con estructura secuencial. - Conformación de equipos en pares de estudiantes, para resolver 2 ejercicios de la práctica, elegidos por el par de estudiantes. - Desarrollo de los 2 ejercicios, con estructura secuencial, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Explicación de los algoritmos planteados por cada estudiante, a su compañero de equipo, con la finalidad de compartir la solución algorítmica encontrada por cada estudiante y así determinar su eficacia. 	Diapositivas Practica 01 (nivel 2) - Estructura secuencial SIA (subir los ejercicios desarrollados por parte de los estudiantes)	90 min
Cierre	Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de dar alternativas que permitan optimizar las soluciones encontradas, con el apoyo del docente.	Pizarra	30 min

PLAN DE SESIÓN DE APRENDIZAJE N° 05
(Practica Calificada 1 y Retroalimentación)

IV. DATOS GENERALES

Carrera: Ingeniería de Sistemas
Ciclo: I
Asignatura: Fundamentos de Programación
Unidad I: Introducción - Algoritmos y Programación - Estructura Secuencial.
Tema: Diseño de algoritmos con estructura secuencial, usando la estrategia de enseñanza INPAL.
Tipo de sesión: Práctica
Tiempo por sesión: 2 horas 15 minutos

V. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad

Construye algoritmos y programas con estructura secuencial, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Resultado de aprendizaje de la Sesión

Diseña algoritmos con estructura secuencial y refuerza su aprendizaje.

VI. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Explicación del docente sobre las indicaciones de la práctica calificada que desarrollará cada estudiante.	Prueba escrita, pizarra	10 min
Desarrollo	- Desarrollo individual de los ejercicios propuestos en la práctica calificada, por parte de los estudiantes.	Prueba escrita SIA (subir los ejercicios desarrollados de la práctica calificada)	80 min
	- Conformación de equipos en pares de estudiantes, para compartir la solución de la práctica calificada, con el fin de completar o corregir el algoritmo diseñado. - Apoyo docente a través de la atención a dudas o inquietudes de los estudiantes.	Prueba escrita, pizarra	20 min
Cierre	Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de presentar una de las soluciones más óptimas de la práctica calificada, con el apoyo del docente.	Pizarra	25 min

PLAN DE SESIÓN DE APRENDIZAJE N° 06
(Algoritmos con Estructura Condicional)

I. DATOS GENERALES

Carrera: Ingeniería de Sistemas
Ciclo: I
Asignatura: Fundamentos de Programación
Unidad II: Algoritmos y Programación - Estructura condicional.
Tema: Diseño de algoritmos con estructura condicional simple y doble, usando la estrategia de enseñanza INPAL.
Tipo de sesión: Teórica
Tiempo por sesión: 2 horas 15 minutos

II. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad

Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Resultado de aprendizaje de la Sesión

Comprende cómo diseñar un algoritmo con estructura condicional simple y doble.

III. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Introducción del tema, a través de la explicación de la estructura general de un algoritmo de la vida diaria, con estructura condicional.	Pizarra	15 min
Desarrollo	<ul style="list-style-type: none"> - Explicación de la estructura general de un algoritmo con estructura condicional simple y doble, mediante una presentación de diapositivas, por parte del docente. - Desarrollo de un primer ejercicio con estructura condicional, mediante las fases del diseño de algoritmos, aplicando la estrategia de enseñanza INPAL. - Desarrollo de un ejercicio con estructura condicional, por parte de los estudiantes, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. 	<p align="center">Diapositivas</p> <p>SIA (subir los ejercicios desarrollados por parte de los estudiantes)</p>	90 min
Cierre	Presentación y explicación de un ejercicio desarrollado, por parte de un estudiante, el cual permitirá participar con consultas o aportes por parte del resto de estudiantes de la clase, y así dar las conclusiones finales por parte del docente.	Pizarra	30 min

PLAN DE SESIÓN DE APRENDIZAJE N° 07
(Algoritmos con Estructura Condicional)

I. DATOS GENERALES

Carrera: Ingeniería de Sistemas
Ciclo: I
Asignatura: Fundamentos de Programación
Unidad II: Introducción - Algoritmos y Programación - Estructura Condicional.
Tema: Diseño de algoritmos con estructura condicional simple y doble, usando la estrategia de enseñanza INPAL.
Tipo de sesión: Práctica
Tiempo por sesión: 2 horas 15 minutos

II. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad

Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Resultado de aprendizaje de la Sesión

Diseña algoritmos con estructura condicional simple y doble.

III. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Participación de los estudiantes sobre lo aprendido en la sesión anterior, para lo cual responde a las siguientes preguntas: ¿En qué consisten los algoritmos con estructura condicional? ¿Mencione algunos ejemplos de problemas que se pueden resolver con estructura condicional simple y doble?	Pizarra	15 min
Desarrollo	<ul style="list-style-type: none"> - Repaso de la estructura general de un algoritmo con estructura condicional, mediante el desarrollo de un ejercicio con estructura condicional simple y doble, aplicando la estrategia de enseñanza INPAL. - Presentación de una práctica 02, con ejercicios propuestos sobre algoritmos con estructura condicional simple y doble. - Conformación de equipos en pares de estudiantes, para resolver 2 ejercicios de la práctica, elegidos por el par de estudiantes. - Desarrollo de los 2 ejercicios, con estructura condicional, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Explicación de los algoritmos planteados por cada estudiante, a su compañero de equipo, con la finalidad de compartir la solución algorítmica encontrada por cada estudiante y así determinar su eficacia. 	Diapositivas Practica 02 - Estructura condicional simple y doble SIA (subir los ejercicios desarrollados por parte de los estudiantes)	90 min
Cierre	Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de dar alternativas que permitan optimizar las soluciones encontradas, con el apoyo del docente.	Pizarra	30 min

PLAN DE SESIÓN DE APRENDIZAJE N° 08
(Algoritmos con Estructura Condicional)

I. DATOS GENERALES

Carrera: Ingeniería de Sistemas
Ciclo: I
Asignatura: Fundamentos de Programación
Unidad II: Algoritmos y Programación - Estructura condicional.
Tema: Diseño de algoritmos con estructura condicional anidada, usando la estrategia de enseñanza INPAL.
Tipo de sesión: Teórica
Tiempo por sesión: 2 horas 15 minutos

II. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad

Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Resultado de aprendizaje de la Sesión

Comprende cómo diseñar un algoritmo con estructura condicional anidada.

III. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Introducción del tema, a través de la explicación de la estructura general de un algoritmo de la vida diaria, con estructura condicional anidada.	Pizarra	15 min
Desarrollo	<ul style="list-style-type: none"> - Explicación de la estructura general de un algoritmo con estructura condicional anidada, mediante una presentación de diapositivas, por parte del docente. - Desarrollo de un ejercicio con estructura condicional anidada, mediante las fases del diseño de algoritmos, aplicando la estrategia de enseñanza INPAL. - Desarrollo de un ejercicio con estructura condicional anidada, por parte de los estudiantes, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. 	<p align="center">Diapositivas</p> <p>SIA (subir los ejercicios desarrollados por parte de los estudiantes)</p>	90 min
Cierre	Presentación y explicación de un ejercicio desarrollado, por parte de un estudiante, el cual permitirá participar con consultas o aportes por parte del resto de estudiantes de la clase, y así dar las conclusiones finales por parte del docente.	Pizarra	30 min

PLAN DE SESIÓN DE APRENDIZAJE N° 09
(Algoritmos con Estructura Condicional)

I. DATOS GENERALES

Carrera: Ingeniería de Sistemas
Ciclo: I
Asignatura: Fundamentos de Programación
Unidad II: Introducción - Algoritmos y Programación - Estructura Condicional.
Tema: Diseño de algoritmos con estructura condicional anidada, usando la estrategia de enseñanza INPAL.
Tipo de sesión: Práctica
Tiempo por sesión: 2 horas 15 minutos

II. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad

Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Resultado de aprendizaje de la Sesión

Diseña algoritmos con estructura condicional anidada.

III. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Participación de los estudiantes sobre lo aprendido en la sesión anterior, para lo cual responde a las siguientes preguntas: ¿En qué consisten los algoritmos con estructura condicional anidada? ¿Mencione algunos ejemplos de problemas que se pueden resolver con estructura condicional anidada?	Pizarra	15 min
Desarrollo	<ul style="list-style-type: none"> - Presentación de una práctica 02, con ejercicios propuestos sobre algoritmos con estructura condicional anidada. - Conformación de equipos en pares de estudiantes, para resolver 2 ejercicios de la práctica, elegidos por el par de estudiantes. - Desarrollo de los 2 ejercicios, con estructura condicional anidada, de manera individual, con el fin de analizar el problema a través de la estrategia de enseñanza INPAL, para luego diseñar el algoritmo. - Explicación de los algoritmos planteados por cada estudiante, a su compañero de equipo, con la finalidad de compartir la solución algorítmica encontrado de cada estudiante y determinar su eficacia. 	Diapositivas Practica 02 - Estructura condicional anidada SIA (subir los ejercicios desarrollados por parte de los estudiantes)	90 min
Cierre	Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de dar alternativas que permitan optimizar las soluciones encontradas, con el apoyo del docente.	Pizarra	30 min

PLAN DE SESIÓN DE APRENDIZAJE N° 10
(Practica Calificada 2 y Retroalimentación)

I. DATOS GENERALES

Carrera: Ingeniería de Sistemas

Ciclo: I

Asignatura: Fundamentos de Programación

Unidad I: Introducción - Algoritmos y Programación - Estructura Condicional.

Tema: Diseño de algoritmos con estructura condicional, usando la estrategia de enseñanza INPAL.

Tipo de sesión: Práctica

Tiempo por sesión: 2 horas 15 minutos

II. RESULTADOS DE APRENDIZAJE

Resultado de aprendizaje de la Unidad

Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.

Resultado de aprendizaje de la Sesión

Diseña algoritmos con estructura condicional y refuerza su aprendizaje.

III. SECUENCIA DIDÁCTICA

FASE	ACTIVIDADES	RECURSOS	TIEMPO
Inicio	Explicación del docente sobre las indicaciones de la práctica calificada que desarrollará cada estudiante.	Prueba escrita, pizarra	10 min
Desarrollo	- Desarrollo individual de los ejercicios propuestos en la práctica calificada, por parte de los estudiantes.	Prueba escrita SIA (subir los ejercicios desarrollados de la práctica calificada)	80 min
	- Conformación de equipos en pares de estudiantes, para compartir la solución de la práctica calificada, con el fin de completar o corregir el algoritmo diseñado. - Apoyo docente a través de la atención a dudas o inquietudes de los estudiantes.	Prueba escrita, pizarra	20 min
Cierre	Presentación y explicación de 1 o 2 ejercicios en la pizarra, por parte de uno de los equipos de estudiantes, quienes participarán en forma voluntaria, con la finalidad de presentar una de las soluciones más óptimas de la práctica calificada, con el apoyo del docente.	Pizarra	25 min

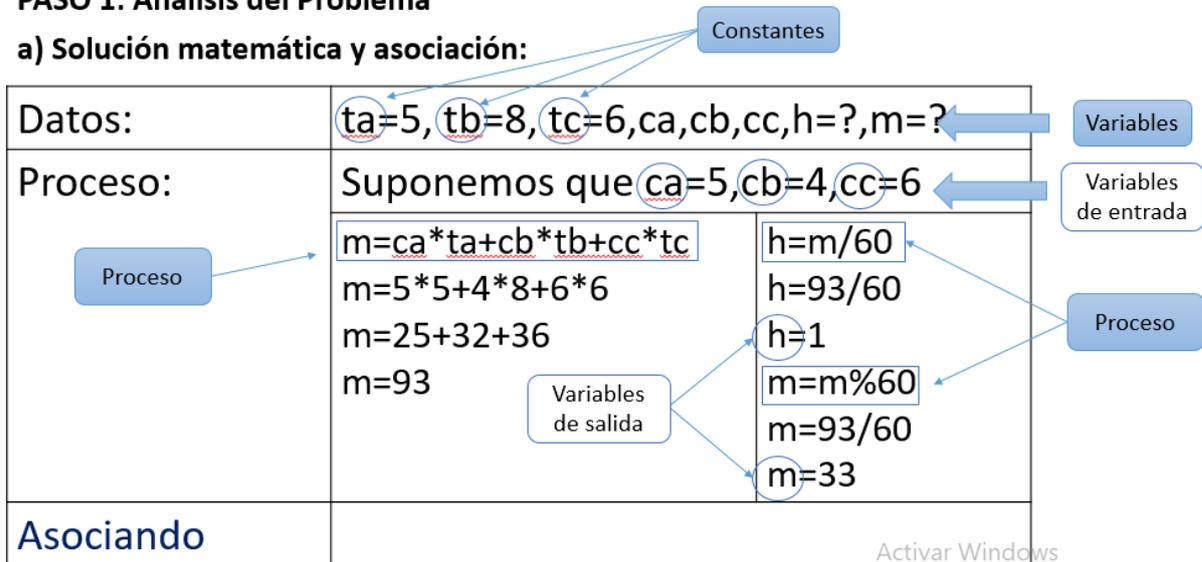
Apéndice 6. Material didáctico para el diseño de algoritmos

Ejemplo desarrollado en Clase:

Un profesor prepara tres cuestionarios para una evaluación final: A, B y C. Se sabe que se tarda 5 minutos en revisar el cuestionario A, 8 en revisar el cuestionario B y 6 en el C. La cantidad de exámenes de cada tipo se ingresan por teclado. ¿Cuántas horas y cuántos minutos se tardará en revisar todas las evaluaciones?

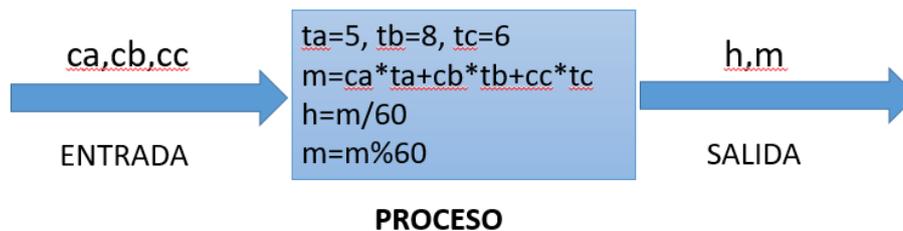
PASO 1: Análisis del Problema

a) Solución matemática y asociación:



PASO 1: Análisis del Problema

b) Especificación de entradas, salidas y proceso:



PASO 2: Diseño del Algoritmo – Pseudocódigo

Inicio

Entero ca,cb,cc,h,m, ta=5, tb=8, tc=6

Leer ca,cb,cc

$m = ca * ta + cb * tb + cc * tc$

$h = m / 60$

$m = m \% 60$

Escribir “Tiempo para revisar evaluaciones: ”, h, “ horas”, m, “ minutos”

Fin

EJERCICIO EN CLASE. Estructura Secuencial

1. En un hospital existen 4 áreas: Covid- 19, Emergencias, Ginecología y Pediatría. El presupuesto mensual del hospital se reparte de la siguiente manera:

Área	Presupuesto
Covid-19	30%
Emergencias	25%
Ginecología	20%
Pediatría	25%

Obtener la cantidad de dinero que recibirá cada área para cualquier monto presupuestal mensual.

SOLUCIÓN DEL PROBLEMA

a) ANÁLISIS DEL PROBLEMA

Solución Matemática	Identificación de variables E/S y proceso
<p>Variabes / Datos: pm=presupuesto mensual=1000 pc=presupuesto covid pe=presupuesto emergencias pg=presupuesto ginecología pp=presupuesto pediatría</p> <p>Proceso: pc=0.3*pm=300 pe=0.25*pm=250 pg=0.2*pm=200 pp=0.25*pm=250</p> <p>Resultados: pc=300, pe=250, pg=200, pp=250</p>	<p>Variabes de entrada: pm=presupuesto mensual</p> <p>Variabes de salida: pc=presupuesto covid pe=presupuesto emergencias pg=presupuesto ginecología pp=presupuesto pediatría</p> <p>Proceso: pc=0.3*pm pe=0.25*pm pg=0.2*pm pp=0.25*pm</p>

b) DISEÑO DEL ALGORITMO

Seudocódigo

Inicio

//Declaración de variables

Real pm, pc, pe, pg, pp

//Entrada de datos

Leer pm

//Proceso

pc=0.3*pm

pe=0.25*pm

pg=0.2*pm

pp=0.25*pm

//Salida de resultados

Escribir "Presupuesto Covid-19: S/.", pc

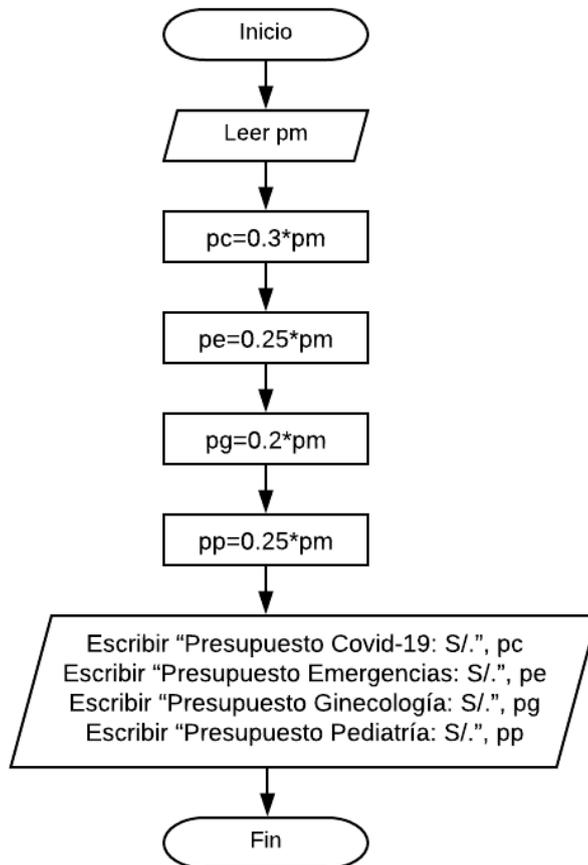
Escribir "Presupuesto Emergencias: S/.", pe

Escribir "Presupuesto Ginecología: S/.", pg

Escribir "Presupuesto Pediatría: S/.", pp

Fin

Diagrama de Flujo



c) PRUEBA DE ESCRITORIO

N° Ejec.	pm	pc	pe	pg	pp	Pantalla
1	1000	300	250	200	250	Presupuesto Covid-19: S/. 300 Presupuesto Emergencias: S/. 250 Presupuesto Ginecología: S/. 200 Presupuesto Pediatría: S/. 250
2	2000	600	500	400	500	Presupuesto Covid-19: S/. 600 Presupuesto Emergencias: S/. 500 Presupuesto Ginecología: S/. 400 Presupuesto Pediatría: S/. 500
3	500	150	125	100	125	Presupuesto Covid-19: S/. 150 Presupuesto Emergencias: S/. 125 Presupuesto Ginecología: S/. 100 Presupuesto Pediatría: S/. 125

EJERCICIO EN CLASE. Estructura Condicional

1. El sueldo de un obrero se calcula en base a las horas trabajadas, y el pago por hora es de S/30.00. Además, hay un descuento de AFP del 12% o por ONP del 13%. Escriba un algoritmo que permita calcular sueldo bruto, el descuento y el sueldo neto.

SOLUCIÓN DEL PROBLEMA

a) ANÁLISIS DEL PROBLEMA

Solución Matemática	Identificación de variables E/S y proceso
<p>Variables / Datos: ht=cantidad horas trabajadas=10 PH=pago por hora=30 d=descuento sb=sueldo bruto sn=sueldo neto ta=tipo afiliación (AFP u ONP)=AFP</p> <p>Proceso: sb= ht*PH=10*30=300 Si (ta=="AFP") (V) d=0.12*sb=0.12*300=36 Sino (F) Si (ta=="ONP") (V) d=0.13*sb sn=sb-d=300-36=264</p> <p>Resultados: Sb=300, d=36, sn=264</p>	<p>Variables de entrada: ht=cantidad horas trabajadas ta=tipo afiliación PH=pago por hora=30</p> <p>Variables de salida: sb=sueldo bruto d=descuento sn=sueldo neto</p> <p>Proceso: sb= ht*PH Si (ta=="AFP") d=0.12*sb Sino Si (ta=="ONP") d=0.13*sb Fin Si sn=sb-d</p>

b) DISEÑO DEL ALGORITMO

Seudocódigo

```

Inicio
//Declaración de variables
Real ht, d,sb,sn
Cadena ta
Entero PH=30
//Entrada de datos
Leer ht
Leer ta
//Proceso

    sb= ht*PH

    Si (ta=="AFP")

        d=0.12*sb
  
```

Sino

Si (ta=="ONP")

$d=0.13*sb$

Fin Si

$sn=sb-d$

//Salida de resultados

Escribir "Sueldo bruto: S/.",sb

Escribir "Descuento: S/.", d

Escribir "Sueldo neto: S/.",sn

Fin

c) PRUEBA DE ESCRITORIO

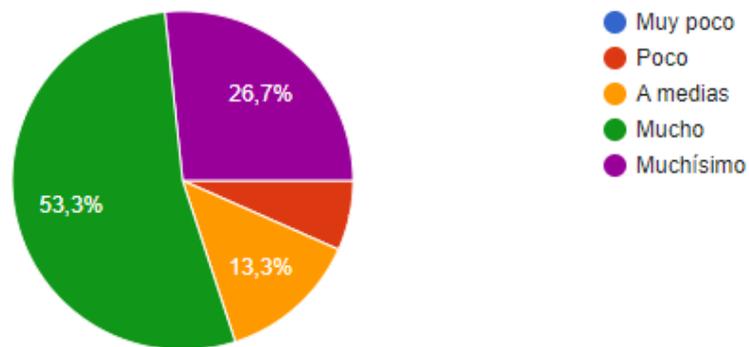
N° Ejec.	PH	ht	ta	sb	(ta=="AFP")	(ta=="ONP")	d	sn	Pantalla
1	30	10	AFP	300	V	-	36	264	Sueldo bruto: S/. 300 Descuento: S/. 36 Sueldo neto: S/. 264
2	30	20.5	ONP	615	F	V	79.95	535.05	Sueldo bruto: S/. 615 Descuento: S/. 79.95 Sueldo neto: S/. 535.05
3	30	40	prima	1200	F	F	-	1200	Sueldo bruto: S/. 1200 Descuento: S/. - Sueldo neto: S/. 1200

Apéndice 7. Resultados de la percepción de los estudiantes con respecto a la estrategia de enseñanza INPAL

Ciclo 2020-I (15 encuestas)

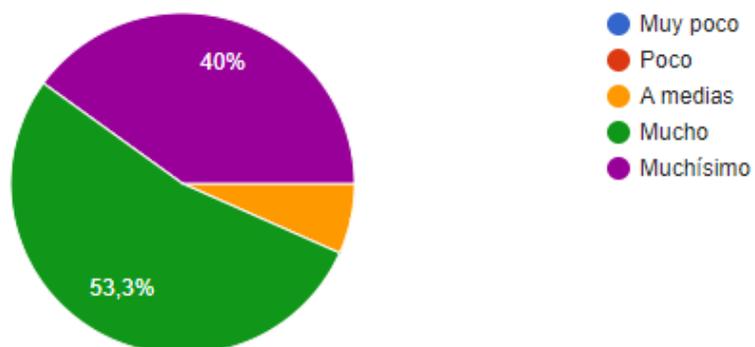
5. En conclusión, la estrategia INPAL le ayudó en el ANÁLISIS DEL PROBLEMA.

15 respuestas



10. En conclusión, la estrategia INPAL le ayudó en el DISEÑO DEL ALGORITMO:

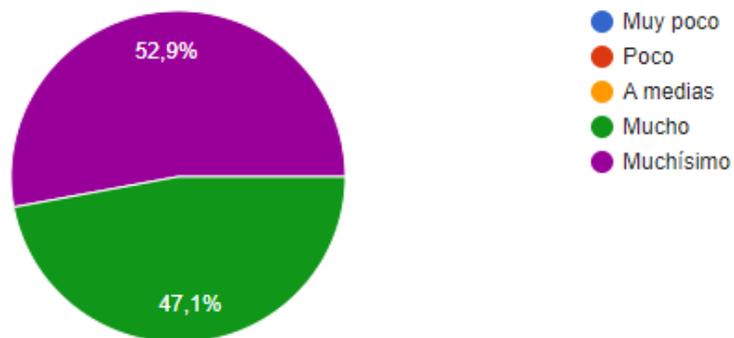
15 respuestas



Ciclo de Nivelación 2020-I ((17 encuestas))

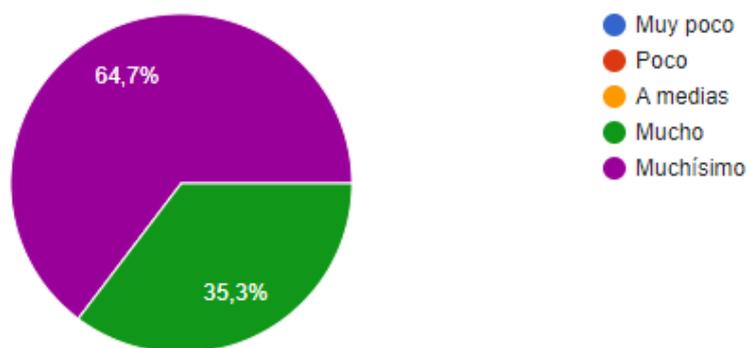
5. En conclusión, la estrategia INPAL le ayudó en el ANÁLISIS DEL PROBLEMA.

17 respuestas



10. En conclusión, la estrategia INPAL le ayudó en el DISEÑO DEL ALGORITMO:

17 respuestas



Apéndice 8. Matriz de Consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	DIMENSIONES	INDICADORES	INSTRUMENTOS	METODOLOGÍA	
¿Cuál es la influencia de la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL), en el diseño de algoritmos de la asignatura “Fundamentos de Programación”, en los estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020?	Determinar la influencia de la aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL), en el diseño de algoritmos de la asignatura “Fundamentos de Programación”, en los estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020.	La aplicación de la estrategia de enseñanza “Inducción para Algoritmos” (INPAL) influye significativamente en el diseño de algoritmos de la asignatura “Fundamentos de Programación”, en los estudiantes de la Escuela Académico Profesional de Ingeniería de Sistemas de la Universidad Nacional de Cajamarca, año 2020.	Variable Independiente (Estrategia de enseñanza INPAL)	Resolución matemática del problema.	<ul style="list-style-type: none"> - Identifica los valores numéricos del problema. - Identifica los valores de entrada. - Identifica los valores de salida. - Establece el proceso o fórmula que produce la salida deseada. - Aplica el proceso o fórmula, en base a los valores de entrada, para obtener los valores de salida. 	Evaluación pretest y postest/ Rúbrica de evaluación.	<p style="text-align: center;">Método Científico.</p> <p style="text-align: center;">Diseño Cuasi experimental.</p>	
			Asociación de solución matemática con los elementos del algoritmo.	<ul style="list-style-type: none"> - Relaciona los valores numéricos con las variables del problema. - Relaciona los valores de entrada con las variables de entrada. - Relaciona los valores de salida con las variables de salida. - Relaciona el proceso matemático con el proceso algorítmico que produce la salida deseada. 				
PROBLEMAS DERIVADOS	OBJETIVOS ESPECÍFICOS	HIPÓTESIS ESPECÍFICAS						
¿Cuál es el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, antes de la aplicación de la estrategia de enseñanza INPAL?	Determinar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, antes de la aplicación de la estrategia de enseñanza INPAL.	El nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, es deficiente, antes de la aplicación de la estrategia de enseñanza INPAL.	Variable Dependiente (Diseño de algoritmos)	Análisis del Problema.	<ul style="list-style-type: none"> - Identifica las variables del problema. - Identifica las variables de entrada. - Identifica las variables de salida. - Establece el proceso que produce la salida deseada. 	Evaluación pretest y postest/ Rúbrica de evaluación.	<p style="text-align: center;">GE: $Y_1 \times Y_2$</p> <p style="text-align: center;">GC: $Y_1 - Y_2$</p> <p style="text-align: center;">Población y muestra: 58 estudiantes de la asignatura de Fundamentos de Programación, año 2020.</p>	
¿Cómo aplicar la estrategia de enseñanza INPAL basada en los resultados del pretest, para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas?	Aplicar la estrategia de enseñanza INPAL basada en los resultados del pretest, para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas.	La estrategia de enseñanza INPAL es aplicada en base a los resultados del pretest, para mejorar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas.			Diseño de la Solución Algorítmica.			<ul style="list-style-type: none"> - Establece la declaración de variables. - Establece la entrada de datos. - Establece el proceso a seguir. - Establece la salida de resultados.
¿Cuál es el nivel de diseño de algoritmos de los estudiantes de la asignatura de “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, después de la aplicación de la estrategia de enseñanza INPAL?	Determinar el nivel de diseño de algoritmos de los estudiantes de la asignatura “Fundamentos de Programación” de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, después de la aplicación de la estrategia de enseñanza INPAL.	El nivel de diseño de algoritmos de los estudiantes de la asignatura de Fundamentos de Programación de la Escuela Académico Profesional de Ingeniería de Sistemas, año 2020, es bueno, después de la aplicación de la estrategia de enseñanza INPAL.						

Anexo 1. Plan de Estudios del Programa de Ingeniería de Sistemas

3.5 PLAN DE ESTUDIOS

3.5.1 Mapa Curricular

3.5.1.1 Organización de Ejes o Líneas curriculares:

Eje 1: Modelos y Unidades Clave de la Organización.

Eje 2: Algoritmos y Programación.

Eje 3: Desarrollo e Implementación de Sistemas.

Eje 4: Gestión de Sistemas de Información.

Eje 5: Infraestructura de Tecnología de la Información.

Eje 6: Enfoque Sistémico.

Eje 7: Aspectos Fundamentales del desempeño profesional.

3.5.1.2 Organización de las competencias en ejes curriculares:

Eje Curricular 1: Modelos y Unidades Clave de la Organización

Competencia del perfil del egresado CEE1:

Analiza los fundamentos de la Administración y funcionamiento del Negocio, para que con un enfoque integrador y tecnológico, proponga procesos de mejora y de futura automatización.

Eje Curricular 2: Algoritmos y Programación

Competencias del Perfil del Egresado CEE2:

Construye Aplicaciones Informáticas para diversas plataformas, teniendo en cuenta los algoritmos y lenguajes de programación, para solucionar problemas existentes en distintas áreas del conocimiento y del negocio.

Eje Curricular 3: Desarrollo e Implementación de Sistemas

Competencias del Perfil del Egresado CEE3:

Implementa Soluciones de Sistemas de información, usando metodologías de desarrollo de software y soportadas con tecnologías de información, para el mejoramiento organizacional.

Eje Curricular 4: Gestión de Sistemas de Información

Competencias del Perfil del Egresado CEE4:

Gestiona sistemas de información en contextos organizacionales utilizando teorías, metodologías, estándares y buenas prácticas para mejorar sus procesos de negocio liderando su puesta en marcha y mejora continua, así como valorar su impacto.

Eje Curricular 5: Infraestructura de Tecnología de la Información

Competencia del perfil del egresado CEE5:

Implementa y Administra Redes considerando fundamentos de comunicación de datos y cableado estructurado, así como especificaciones y protocolos, permitiendo integrar soluciones de comunicación en las organizaciones.

Eje Curricular 6: Enfoque Sistémico

Competencias del Perfil del Egresado CEE6:

Utiliza el Enfoque Sistémico y la Dinámica de Sistemas en situaciones del mundo real, y bajo sus fundamentos y herramientas, construye y simula modelos para su representación y solución.

Eje Curricular 7: Aspectos Fundamentales de la Carrera

Competencia del Perfil del egresado CEE7:

Se desenvuelve haciendo uso de habilidades directivas, con creatividad y ética, gestionando las relaciones interpersonales en un equipo multi cultural y multifuncional, considerando los códigos, normas y reglamentos que rigen las practicas inherentes a la profesión, así como los conocimientos y capacidades adquiridas y desarrolladas durante su formación profesional para un desempeño óptimo de la carrera dentro de las organizaciones

3.5.1.3 Progresión de competencias

Eje Curricular 2: Algoritmos y Programación

Competencias del Perfil del Egresado CEE2:

Construye Aplicaciones Informáticas para diversas plataformas, teniendo en cuenta los algoritmos y lenguajes de programación, para solucionar problemas existentes en distintas áreas del conocimiento y del negocio.

Nivel Básico:

Construye Programas computacionales en consola, considerando los fundamentos de algoritmos y los lenguajes de programación, para solucionar problemas básicos en distintas áreas del conocimiento.

Nivel Intermedio:

Construye Aplicaciones visuales, de escritorio y WEB, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor y la Teoría de autómatas, para automatizar procesos operacionales de la organización y de distintas áreas de conocimiento.

Nivel Avanzado:

Construye Aplicaciones para dispositivos móviles, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor e interfaces, para solucionar problemas existentes en distintas áreas del conocimiento y del negocio.

3.5.2 Resultados de aprendizaje y su alineamiento con competencias y asignaturas

Tabla 10. Alineamiento de competencias, resultados de aprendizaje y asignaturas.

ASIGNATURAS DEL EJE CURRICULAR 2: ALGORITMOS Y PROGRAMACIÓN		
<p>Eje Curricular 2: Algoritmos y Programación</p> <p>Competencia CE2 de Nivel Básico:</p> <p>Construye programas computacionales en consola, considerando los fundamentos de algoritmos y los lenguajes de programación, para solucionar problemas básicos en distintas áreas del conocimiento.</p>	<ul style="list-style-type: none"> • Construye programas con el paradigma OO verificando el buen funcionamiento y que use los conceptos de la OO. • Crea un programa sobre arreglos, teniendo en cuenta el paradigma OO y por lo menos tres conceptos estudiados en la unidad. • Crea un programa usando colecciones que desarrolle un problema de la vida real, considerando el buen uso y funcionamiento de los métodos definidos por las colecciones. 	<p>ALGORITMOS Y ESTRUCTURAS DE DATOS I</p>

COMPETENCIAS	RESULTADOS DE APRENDIZAJE	ASIGNATURAS
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 de Nivel Básico: Construye Programas computacionales en consola, considerando los fundamentos de algoritmos y los lenguajes de programación, para solucionar problemas básicos en distintas áreas del conocimiento.</p>	<ul style="list-style-type: none"> • Crea programas usando listas enlazadas, demostrando habilidad en su solución y buen funcionamiento. • Crea dos programas que solucione cada uno un problema práctico usando por lo menos dos temas desarrollados en la unidad, considerando el buen funcionamiento y el uso adecuado de los conceptos estudiados. 	ALGORITMOS Y ESTRUCTURAS DE DATOS II
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 de Nivel Básico: Construye Programas computacionales en consola, considerando los fundamentos de algoritmos y los lenguajes de programación, para solucionar problemas básicos en distintas áreas del conocimiento.</p>	<ul style="list-style-type: none"> • Explica los conceptos básicos y la importancia de los lenguajes de programación considerando su evolución, clasificación y características. • Construye algoritmos y programas con estructura secuencial, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación. • Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación. • Construye algoritmos y programas con estructura repetitiva, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación. • Construye algoritmos y programas con arreglos unidimensionales, utilizando subprogramas o métodos y considerando el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación. 	FUNDAMENTOS DE PROGRAMACIÓN
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 de Nivel Intermedio: Construye Aplicaciones visuales, de escritorio y WEB, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor y la Teoría de autómatas, para automatizar procesos operacionales de la organización y de distintas áreas de conocimiento.</p>	<ul style="list-style-type: none"> • Desarrolla una aplicación de escritorio, que permita automatizar procesos operacionales, empleando clases, objetos, métodos, eventos, estructuras de control simple, múltiple, repetitiva y estructuras de datos. • Desarrolla una aplicación de escritorio con acceso a datos que contenga mantenimientos, consultas y reportes a una base de datos relacional. 	PROGRAMACIÓN APLICADA I

COMPETENCIAS	RESULTADOS DE APRENDIZAJE	ASIGNATURAS
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 de Nivel Intermedio: Construye Aplicaciones visuales, de escritorio y WEB, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor y la Teoría de autómatas, para automatizar procesos operacionales de la organización y de distintas áreas de conocimiento.</p>	<ul style="list-style-type: none"> • Implementa un sitio web organizacional, considerando los fundamentos y frameworks para el desarrollo de sitios web estáticos. • Construye sitios web dinámicos aplicando las herramientas de frameworks web cliente/servidor. • Construye una aplicación web utilizando un patrón de arquitectura de software y el uso de las herramientas de acceso a datos de frameworks web cliente/servidor. • Implementa los mecanismos de seguridad en una aplicación web considerando los criterios de seguridad de frameworks web cliente/servidor. 	PROGRAMACIÓN APLICADA II
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 Nivel Intermedio: Construye Aplicaciones visuales, de escritorio y WEB, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor y la Teoría de autómatas, para automatizar procesos operacionales de la organización y de distintas áreas de conocimiento.</p>	<ul style="list-style-type: none"> • Crea un mapa de composición final usando tablas, gráficos y símbolos, demostrando habilidad del uso de las herramientas de sistemas de información geográfica para manipular datos espaciales y tabulares. • Digitaliza un mapa vectorial en base a un mapa raster, usando proyecciones en cartografía digital, georreferenciación de Mapas, uniones y enlaces entre tablas, verificando la correcta ubicación de la representación de cada uno de los elementos geográficos que contiene el mapa. • Crea un mapa de composición final usando herramientas de información geográfica de búsquedas espaciales, unión espacial, geoprocésamiento de datos, análisis de redes y herramienta 3D-Analyst, considerando lectura e interpretación adecuada de los gráficos estadísticos y los elementos que contiene el mapa y facilite la toma decisiones. 	SISTEMAS DE INFORMACIÓN GEOGRÁFICOS (ELECTIVO)
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 Nivel Intermedio: Construye Aplicaciones visuales, de escritorio y WEB, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor y la Teoría de autómatas, para automatizar procesos operacionales de la organización y de distintas áreas de conocimiento.</p>	<ul style="list-style-type: none"> • Define un sistema como inteligente teniendo coherencia en su enfoque y paradigma. • Modela un proyecto de Machine Learning, utilizando las redes neuronales. • Diseña y modela un sistema experto en base a la lógica difusa. • Desarrolla una aplicación en programación lógica que emule el razonamiento del uso del lenguaje natural. 	SISTEMAS INTELIGENTES

COMPETENCIAS	RESULTADOS DE APRENDIZAJE	ASIGNATURAS
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 Nivel Intermedio: Construye Aplicaciones visuales, de escritorio y WEB, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor y la Teoría de autómatas, para automatizar procesos operacionales de la organización y de distintas áreas de conocimiento.</p>	<ul style="list-style-type: none"> Describe aspectos fundamentales de la teoría de lenguajes formales y compiladores teniendo en cuenta los antecedentes históricos, conceptos básicos y principales aplicaciones. Desarrolla problemas de determinación de lenguajes y diseño de gramáticas formales basados en la jerarquía de Chomsky y algoritmos de simplificación de gramáticas. Desarrolla problemas relacionados con expresiones regulares y de construcción de autómatas para reconocer lenguajes definidos por una Gramática del Tipo 3 y traductores para traducir lenguajes definidos por una Gramática del Tipo 2, considerando propiedades del algebra de Boole, definiciones formales de autómatas y traductores y algoritmos de transformación. Diseña un lenguaje de programación sencillo y un analizador léxico y sintáctico que lo traduzca considerando sus funciones, elementos, reglas y formas de implementar. 	TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 Nivel Intermedio: Construye Aplicaciones visuales, de escritorio y WEB, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor y la Teoría de autómatas, para automatizar procesos operacionales de la organización y de distintas áreas de conocimiento.</p>	<ul style="list-style-type: none"> Describe las características, tipos, arquitecturas y aplicaciones de la Teoría de agentes y Sistemas multiagente para implementar soluciones en distintos dominios. Implementa algoritmos de clasificación para el reconocimiento de imágenes y objetos teniendo en cuenta las medidas de calidad correspondientes. Diseña aplicaciones de la robótica considerando el estado del arte de los sistemas robóticos, sensores, arquitecturas, herramientas y técnicas para diferentes problemas y entornos. 	TÓPICOS ESPECIALES EN SISTEMAS INTELIGENTES
<p>Eje Curricular 2: Algoritmos y Programación.</p> <p>Competencia CE2 de Nivel Avanzado: Construye Aplicaciones para dispositivos móviles, usando el paradigma de programación orientada a objetos, estructura de datos, Arquitectura cliente/servidor e interfaces, para solucionar problemas existentes en distintas áreas del conocimiento y del negocio.</p>	<ul style="list-style-type: none"> Crea una aplicación móvil usando APIs, Layout y controles, realizando pruebas de funcionamiento del código y usabilidad. Crea una aplicación móvil usando recursos, integridad de las base datos internas en el sistema operativo y consumir servicios web, considerando el buen funcionamiento y la usabilidad. 	PROGRAMACIÓN APLICADA III

3.5.3 Asignaturas por líneas o ejes curriculares

Tabla 11. Asignaturas por líneas o ejes curriculares

LÍNEAS O EJES CURRICULARES	ASIGNATURAS
ASPECTOS BÁSICOS	MATEMÁTICA
	LENGUAJE Y COMUNICACIÓN
	CULTURA Y REALIDAD NACIONAL
	METODOLOGÍA DEL TRABAJO UNIVERSITARIO
	DIBUJO Y GEOMETRÍA DESCRIPTIVA
	FÍSICA I
	ANÁLISIS MATEMÁTICO I
	REDACCIÓN ACADÉMICA
	FILOSOFÍA Y FUNDAMENTOS DE LA INVESTIGACIÓN
	CONSTITUCIÓN, DEMOCRACIA Y CIUDADANÍA
	ECOLOGÍA Y DESARROLLO SOSTENIBLE
	ESTADÍSTICA Y PROBABILIDADES
	ANÁLISIS MATEMÁTICO II
	ANÁLISIS MATEMÁTICO III
	ESTADÍSTICA APLICADA
	FÍSICA APLICADA
INGLÉS STEM I	
INGLÉS STEM II	
ALGORITMOS Y PROGRAMACIÓN	FUNDAMENTOS DE PROGRAMACIÓN
	ALGORITMOS Y ESTRUCTURAS DE DATOS I
	ALGORITMOS Y ESTRUCTURAS DE DATOS II
	PROGRAMACIÓN APLICADA I
	PROGRAMACIÓN APLICADA II
	PROGRAMACIÓN APLICADA III
	TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES
	SISTEMAS INTELIGENTES
	TÓPICOS ESPECIALES EN SISTEMAS INTELIGENTES (ELECTIVO)
	SISTEMAS DE INFORMACIÓN GEOGRÁFICOS (ELECTIVO)

DESARROLLO E IMPLEMENTACIÓN DE SISTEMAS	BASE DE DATOS I
	BASE DE DATOS II
	INTELIGENCIA DE NEGOCIOS
	TÓPICOS ESPECIALES EN INTELIGENCIA DE NEGOCIOS (ELECTIVO)
	INGENIERÍA DE SOFTWARE I
	INGENIERÍA DE SOFTWARE II
	INGENIERIA DE SOFTWARE III
ENFOQUE SISTÉMICO	TEORÍA DE SISTEMAS
	DINÁMICA DE SISTEMAS
	SISTEMAS COMPLEJOS (ELECTIVO)
INFRAESTRUCTURA DE TECNOLOGÍA DE LA INFORMACIÓN	SISTEMAS DIGITALES
	ARQUITECTURA DE COMPUTADOR
	FUNDAMENTOS DE SISTEMAS OPERATIVOS
	REDES I
	REDES II
	REDES III
	TÓPICOS ESPECIALES EN REDES (ELECTIVO)
MODELOS Y UNIDADES CLAVE DE LA ORGANIZACION	INVESTIGACIÓN DE OPERACIONES EN INGENIERÍA I
	INVESTIGACIÓN DE OPERACIONES EN INGENIERÍA II
	DIRECCIÓN DE OPERACIONES EN INGENIERÍA
	ORGANIZACIÓN Y ADMINISTRACIÓN DE EMPRESAS
	CONTABILIDAD Y FINANZAS
	GESTIÓN DEL TALENTO HUMANO
	GESTIÓN DE PROCESOS DE NEGOCIO
	MARKETING

GESTION DE SISTEMAS DE INFORMACIÓN	FUNDAMENTOS DE SISTEMAS DE INFORMACIÓN
	PLANEAMIENTO DE SISTEMAS DE INFORMACIÓN I
	PLANEAMIENTO DE SISTEMAS DE INFORMACIÓN II
	ARQUITECTURA EMPRESARIAL
	SISTEMAS EMPRESARIALES
	SEGURIDAD DE LA INFORMACIÓN Y CONTINUIDAD DEL NEGOCIO
	E-BUSSINESS
	GESTIÓN DEL CONOCIMIENTO
	GESTIÓN DE PROYECTOS DE SISTEMAS I
	GESTIÓN DE PROYECTOS DE SISTEMAS II
	GESTIÓN DE TECNOLOGÍAS DE INFORMACIÓN
	TALLER DE EMPRENDIMIENTO TECNOLÓGICO (ELECTIVO)
	ASPECTOS FUNDAMENTALES DEL DESEMPEÑO PROFESIONAL
DEONTOLOGÍA DE LA INGENIERÍA DE SISTEMAS	
TESIS I	
TESIS II	

Fuente: EAPIS

3.5.4 Malla Curricular

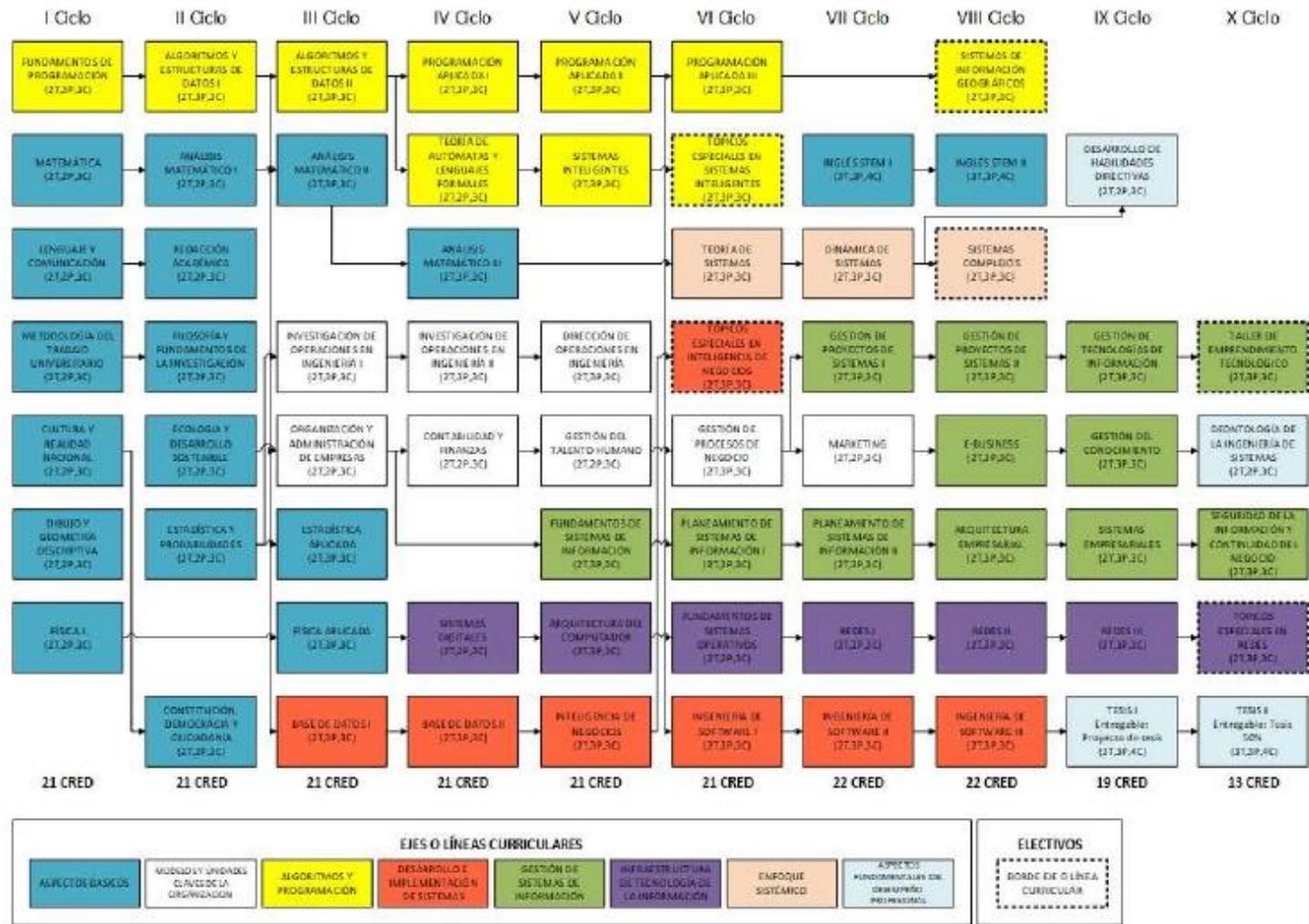


Figura 16. Malla Curricular

3.5.5 Distribución de asignaturas por semestre

Tabla 12. Distribución de asignaturas por semestre

I CICLO								
PRIMER AÑO - PRIMER SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		FUNDAMENTOS DE PROGRAMACIÓN	E	O	2	3	3	NINGUNO
2		MATEMÁTICA	G	O	2	2	3	NINGUNO
3		LENGUAJE Y COMUNICACIÓN	G	O	2	2	3	NINGUNO
4		CULTURA Y REALIDAD NACIONAL	G	O	2	2	3	NINGUNO
5		METODOLOGÍA DEL TRABAJO UNIVERSITARIO	G	O	2	2	3	NINGUNO
6		DIBUJO Y GEOMETRÍA DESCRIPTIVA	G	O	2	2	3	NINGUNO
7		FÍSICA I	G	O	2	2	3	NINGUNO
TOTAL DE CRÉDITOS							21	
TOTAL DE CRÉDITOS ACUMULADOS							21	

II CICLO								
PRIMER AÑO - SEGUNDO SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		ALGORITMOS Y ESTRUCTURA DE DATOS I	E	O	2	3	3	FUNDAMENTOS DE PROGRAMACIÓN
2		ANÁLISIS MATEMÁTICO I	G	O	2	2	3	MATEMÁTICA
3		REDACCIÓN ACADÉMICA	G	O	2	2	3	LENGUAJE Y COMUNICACIÓN
4		FILOSOFÍA Y FUNDAMENTOS DE LA INVESTIGACIÓN	G	O	2	2	3	METODOLOGÍA DEL TRABAJO UNIVERSITARIO
5		CONSTITUCIÓN, DEMOCRACIA Y CIUDADANÍA	G	O	2	2	3	CULTURA Y REALIDAD NACIONAL
6		ECOLOGÍA Y DESARROLLO SOSTENIBLE	G	O	2	2	3	NINGUNO
7		ESTADÍSTICA Y PROBABILIDADES	G	O	2	2	3	NINGUNO
TOTAL DE CRÉDITOS							21	
TOTAL DE CRÉDITOS ACUMULADOS							42	

III CICLO								
SEGUNDO AÑO – PRIMER SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		ALGORITMOS Y ESTRUCTURAS DE DATOS II	E	O	2	3	3	ALGORITMOS Y ESTRUCTURAS DE DATOS I
2		ANÁLISIS MATEMÁTICO II	F	O	2	3	3	ANÁLISIS MATEMÁTICO I
3		INVESTIGACIÓN DE OPERACIONES EN INGENIERÍA I	F	O	2	3	3	ESTADÍSTICA Y PROBABILIDADES
4		ORGANIZACIÓN Y ADMINISTRACIÓN DE EMPRESAS	F	O	2	2	3	ECOLOGÍA Y DESARROLLO SOSTENIBLE
5		FÍSICA APLICADA	F	O	2	3	3	FÍSICA I
6		ESTADÍSTICA APLICADA	F	O	2	3	3	ESTADÍSTICA Y PROBABILIDADES
7		BASE DE DATOS I	E	O	2	3	3	ALGORITMOS Y ESTRUCTURAS DE DATOS I
TOTAL DE CRÉDITOS							21	
TOTAL DE CRÉDITOS ACUMULADOS							63	

IV CICLO								
SEGUNDO AÑO – SEGUNDO SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		PROGRAMACIÓN APLICADA I	E	O	2	3	3	ALGORITMOS Y ESTRUCTURAS DE DATOS II
2		TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES	E	O	2	2	3	ALGORITMOS Y ESTRUCTURAS DE DATOS II
3		ANÁLISIS MATEMÁTICO III	F	O	2	3	3	ANÁLISIS MATEMÁTICO II
4		INVESTIGACIÓN DE OPERACIONES EN INGENIERÍA II	F	O	2	3	3	INVESTIGACIÓN DE OPERACIONES EN INGENIERÍA I
5		CONTABILIDAD Y FINANZAS	F	O	2	2	3	ORGANIZACIÓN Y ADMINISTRACIÓN DE EMPRESAS
6		BASE DE DATOS II	E	O	2	3	3	BASE DE DATOS I
7		SISTEMAS DIGITALES	F	O	2	2	3	FÍSICA APLICADA
TOTAL DE CRÉDITOS							21	
TOTAL DE CRÉDITOS ACUMULADOS							84	

V CICLO								
TERCER AÑO – PRIMER SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		PROGRAMACIÓN APLICADA II	E	O	2	3	3	PROGRAMACIÓN APLICADA I
2		SISTEMAS INTELIGENTES	E	O	2	3	3	TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES
3		GESTIÓN DEL TALENTO HUMANO	F	O	2	2	3	CONTABILIDAD Y FINANZAS
4		DIRECCIÓN DE OPERACIONES EN INGENIERÍA	F	O	2	3	3	INVESTIGACIÓN DE OPERACIONES EN INGENIERÍA II
5		FUNDAMENTOS DE LOS SISTEMAS DE INFORMACIÓN	E	O	2	3	3	ORGANIZACIÓN Y ADMINISTRACIÓN DE EMPRESAS
6		ARQUITECTURA DEL COMPUTADOR	F	O	2	2	3	SISTEMAS DIGITALES
7		INTELIGENCIA DE NEGOCIOS	E	O	2	3	3	BASE DE DATOS II
TOTAL DE CRÉDITOS							21	
TOTAL DE CRÉDITOS ACUMULADOS							105	

VI CICLO								
TERCER AÑO – SEGUNDO SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		PROGRAMACIÓN APLICADA III	E	O	2	3	3	PROGRAMACIÓN APLICADA II
2		TEORÍA DE SISTEMAS	E	O	2	3	3	DIRECCIÓN DE OPERACIONES EN INGENIERÍA
3		PLANEAMIENTO DE SISTEMAS DE INFORMACIÓN I	E	O	2	3	3	FUNDAMENTOS DE SISTEMAS DE INFORMACIÓN
4		GESTIÓN DE PROCESOS DE NEGOCIO	E	O	2	3	3	GESTION DEL TALENTO HUMANO
5		INGENIERÍA DE SOFTWARE I	E	O	2	3	3	PROGRAMACIÓN APLICADA I
6		FUNDAMENTOS DE SISTEMAS OPERATIVOS	E	O	2	2	3	ARQUITECTURA DEL COMPUTADOR
7		TÓPICOS ESPECIALES EN INTELIGENCIA DE NEGOCIOS (ELECTIVO)	E	E	2	3	3	INTELIGENCIA DE NEGOCIOS
8		TÓPICOS ESPECIALES EN SISTEMAS INTELIGENTES (ELECTIVO)	E	E	2	3	3	SISTEMAS INTELIGENTES
TOTAL DE CRÉDITOS							21	
TOTAL DE CRÉDITOS ACUMULADOS							126	

VII CICLO								
CUARTO AÑO – PRIMER SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		INGENIERÍA DE SOFTWARE II	E	O	2	3	3	INGENIERÍA DE SOFTWARE I
2		GESTIÓN DE PROYECTOS DE SISTEMAS I	E	O	2	3	3	GESTIÓN DE PROCESOS DE NEGOCIO
3		INGLÉS STEM I	F	O	3	3	4	CERTIFICACION INGLÉS INTERMEDIO, CENTRO IDIOMAS UNC
4		PLANEAMIENTO DE SISTEMAS DE INFORMACIÓN II	E	O	2	3	3	PLANEAMIENTO DE SISTEMAS DE INFORMACIÓN I
5		MARKETING	F	O	2	2	3	GESTIÓN Y DISEÑO DE PROCESOS DE NEGOCIO
6		DINÁMICA DE SISTEMAS	E	O	2	3	3	TEORÍA DE SISTEMAS
7		REDES I	E	O	2	3	3	FUNDAMENTOS DE SISTEMAS OPERATIVOS
TOTAL DE CRÉDITOS							22	
TOTAL DE CRÉDITOS ACUMULADOS							148	

VIII CICLO								
CUARTO AÑO – SEGUNDO SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		INGENIERÍA DE SOFTWARE III	E	O	2	3	3	INGENIERÍA DE SOFTWARE II
2		GESTIÓN DE PROYECTOS DE SISTEMAS II	E	O	2	3	3	GESTIÓN DE PROYECTOS DE SISTEMAS I
3		ARQUITECTURA EMPRESARIAL	E	O	2	3	3	PLANEAMIENTO DE SISTEMAS DE INFORMACIÓN II
4		REDES II	E	O	2	3	3	REDES I
5		E-BUSINESS	E	O	2	3	3	MARKETING
6		SISTEMAS COMPLEJOS (ELECTIVO)	E	E	2	3	3	DINÁMICA DE SISTEMAS
7		SISTEMAS DE INFORMACIÓN GEOGRÁFICOS (ELECTIVO)	E	E	2	3	3	BASE DE DATOS II
8		INGLÉS STEM II	F	O	3	3	4	INGLÉS STEM I
TOTAL DE CRÉDITOS							22	
TOTAL DE CRÉDITOS ACUMULADOS							170	

IX CICLO								
QUINTO AÑO – PRIMER SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		DESARROLLO DE HABILIDADES DIRECTIVAS	F	O	2	2	3	NINGUNO
2		GESTIÓN DE TECNOLOGÍAS DE INFORMACIÓN	E	O	2	3	3	GESTIÓN DE PROYECTOS DE SISTEMAS II
3		GESTIÓN DE CONOCIMIENTO	E	O	2	3	3	E-BUSINESS
4		SISTEMAS EMPRESARIALES	E	O	2	3	3	ARQUITECTURA EMPRESARIAL
5		TESIS I	E	O	3	3	4	PRÁCTICAS PRE PROFESIONALES
6		REDES III	E	O	2	3	3	REDES II
TOTAL DE CRÉDITOS							19	
TOTAL DE CRÉDITOS ACUMULADOS							189	

X CICLO								
QUINTO AÑO – SEGUNDO SEMESTRE								
N°	COD	ASIGNATURA	ÁREA	TIPO	HT	HP	CRED	PRERREQUISITO
1		DEONTOLOGÍA DE LA INGENIERÍA DE SISTEMAS	F	O	2	2	3	GESTIÓN DEL CONOCIMIENTO
2		SEGURIDAD DE LA INFORMACIÓN Y CONTINUIDAD DEL NEGOCIO	E	O	2	3	3	SISTEMAS EMPRESARIALES
3		TESIS II	E	O	3	3	4	TESIS I
4		TALLER DE EMPRENDIMIENTO TECNOLÓGICO (ELECTIVO)	E	E	2	3	3	GESTIÓN DE TECNOLOGÍAS DE INFORMACIÓN
5		TÓPICOS ESPECIALES EN REDES (ELECTIVO)	E	E	2	3	3	REDES III
TOTAL DE CRÉDITOS							13	
TOTAL DE CRÉDITOS ACUMULADOS							202	

Fuente: EAPIS

Anexo 2. Sílabo de la asignatura de Fundamentos de Programación



UNIVERSIDAD NACIONAL DE CAJAMARCA
FACULTAD DE INGENIERIA
ESCUELA ACADEMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

SILABO DE LA ASIGNATURA FUNDAMENTOS DE PROGRAMACION

I. INFORMACION GENERAL

- | | |
|----------------------------|----------------------------|
| 1. Carrera profesional | : Ingeniería de Sistemas |
| 2. Código de la Asignatura | : 11Q222 |
| 3. Prerrequisito | : SPR |
| 4. Semestre académico | : 2020 – I |
| 5. Horas semanales | 5 |
| 6. Créditos | 3 |
| 7. Fecha de inicio | : 17 de agosto del 2020 |
| Fecha de término | : 11 de diciembre del 2020 |
| 8. Docentes | : |

Dr. Ing. Carlos Aparicio Arteaga - A
caparicio@unc.edu.pe
ORCID: [0000-0001-7007-808X](https://orcid.org/0000-0001-7007-808X)

Dr. Ing. Manuel Malpica Rodríguez – A1
mmalpica@unc.edu.pe
ORCID: [0000-0002-5534-5116](https://orcid.org/0000-0002-5534-5116)

Ms. Cs. Ing. Sandra Rodríguez Ávila – B
srodriguez@unc.edu.pe
ORCID: [0000-0001-5409-6325](https://orcid.org/0000-0001-5409-6325)

Ms. Cs. Ing. Lisi Vásquez Fernández – B1
ljvasquezf@unc.edu.pe
ORCID: [0000-0001-5429-2510](https://orcid.org/0000-0001-5429-2510)

II. SUMILLA

La asignatura de Fundamentos de Programación corresponde a la especialidad y es de carácter teórico-práctico. Esta asignatura tiene como propósito que el estudiante construya algoritmos y programas computacionales básicos en consola, considerando el análisis del problema, el diseño de la solución, la prueba de escritorio y la implementación en un lenguaje de programación. Los contenidos generales de la asignatura son: Introducción a la programación, algoritmos y programación de las estructuras de control secuencial, condicional y repetitiva, subprogramas o métodos y arreglos unidimensionales.

III. COMPETENCIAS A LA QUE SE ORIENTA LA ASIGNATURA

Competencia genérica	Competencia específica	Resultado de la asignatura en relación a las competencias
Aplica el razonamiento lógico-matemático de manera eficaz para la solución de problemas del contexto.	Construye aplicaciones informáticas para diversas plataformas, teniendo en cuenta los algoritmos y lenguajes de programación para solucionar problemas existentes en distintas áreas del conocimiento y del negocio.	Al concluir la asignatura el estudiante construye algoritmos y programas computacionales básicos en consola empleando las estructuras de control secuencial, condicional y repetitiva; y considerando el análisis del problema, el diseño de la solución, la prueba de escritorio y la implementación en un lenguaje de programación.

IV. ORGANIZACIÓN DE LAS UNIDADES Y RESULTADOS DE APRENDIZAJE

Unidad didáctica 1: Introducción - Algoritmos y Programación - Estructura Secuencias		Resultado de aprendizaje 1: Explica los conceptos básicos y la importancia de los lenguajes de programación considerando su evolución, clasificación y características. Resultado de aprendizaje 2: Construye algoritmos y programas con estructura secuencial, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.	
Sem	Contenidos	Estrategias	Indicadores de desempeño
1	<p>Presentación</p> <p>INTRODUCCION Computadoras: Historia o evolución. Hardware (Dispositivos. Representación de la Información). Software (Clasificación) Redes Lenguajes de Programación: Clasificación. Programa fuente y programa objeto. Compilación y ejecución de un programa (intérpretes y compiladores). Evolución de los lenguajes de programación.</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Escucha con atención la presentación del Docente y el Silabo de la asignatura a través de videoconferencia del Google Meet o video en Youtube. (Sincrónica) • Realiza su presentación a través del foro del SIA o videoconferencia del GoogleMeet (Sincrónica) • Participa en forma anónima respondiendo preguntas o a través de lluvias de ideas acerca de experiencias previas relacionados con los contenidos a desarrollarse a través de Google Forms o algún Sistema de Respuesta en Clase. (Sincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Representa la información proporcionada a través de organizadores gráficos (mapa conceptual, cuadro comparativo, línea de tiempo, etc.) según la Guía correspondiente.

	<p>Fases en la resolución de problemas: Análisis del problema. Diseño del algoritmo. Codificación de un programa. Compilación y ejecución de un programa. Verificación y depuración de un programa. Documentación y mantenimiento Programacion estructurada y Programacion orientada a objetos. Importancia de los lenguajes de programación y la programación.</p>	<ul style="list-style-type: none"> • Trabaja en forma cooperativa diseñando organizadores gráficos de información de los contenidos desarrollados según la Guía de Actividad. (Sincrónica) • Resuelve cuestionario preguntas relacionados con los contenidos a través de Google Forms o algún Sistema de Respuesta en Clase. (Sincrónica) 	<ul style="list-style-type: none"> • Responde en forma individual preguntas relacionadas al tema a través de un Cuestionario
2	<p>ALGORITMOS Características. Tipos (Algoritmos para resolver problemas de la vida diaria. Algoritmos computacionales) Formas de describir o representar un Algoritmo (Seudocódigo, Diagrama de Flujo y Diagrama N-S). Estructuras básicas de control Ejemplos o Casos de Algoritmos de la Vida Diaria (Análisis del Problema y Diseño del Algoritmo)</p> <p>LENGUAJE E IDE DE PROGRAMACION JAVA (ECLIPSE/Visual Studio Code) Características Instalación Entorno de Desarrollo Funcionamiento. Método Main() Mi primeros programas</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma anónima respondiendo preguntas o lluvias de ideas acerca de experiencias previas relacionados con los contenidos a desarrollarse a través de Google Forms o algún Sistema de Respuesta en Clase. (Sincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elabora algoritmos para solucionar problemas o situaciones sencillas que se presentan en la vida diaria en forma grupal. (Sincrónica) • Instala el IDE Eclipse y el Lenguaje Java teniendo en cuenta la Guía correspondiente. (Sincrónica) • Elabora programas sencillos en el Método Main familiarizándose con el IDE y el lenguaje. (Sincrónica) • Resuelve cuestionario preguntas relacionados con los contenidos a través de Google Forms o algún Sistema de Respuesta en Clase. (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Construye algoritmos de la vida diaria para resolver problemas sencillos con claridad orden y precisión a través de una práctica grupal y/o exposición oral. • Responde en forma individual preguntas relacionadas al tema a través de un Cuestionario.

3	<p>ESTRUCTURA SECUENCIAL Estructura general de un algoritmo. (Declaración - inicialización de variables, Entrada de datos, Procesos a seguir, Salida de resultados).</p> <p>Elementos básicos de un algoritmo. Estructura general de un programa Elementos básicos de un programa y sintaxis Algoritmos - Programas (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elaboran algoritmos secuenciales para solucionar problemas sencillos (cálculos matemáticos) teniendo en cuenta el análisis, diseño del algoritmo y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas secuenciales computacionales para solucionar problemas sencillos (cálculos matemáticos) considerando los elementos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos secuenciales (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas secuenciales en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.
4	Evaluación 1 y feedback	<ul style="list-style-type: none"> • Examen Desarrollo de Programas 	<ul style="list-style-type: none"> • Resuelve examen desarrollando programas secuenciales en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados.

Unidad didáctica 2: Estructura de control selectiva o condicional: Algoritmos y Programación		Resultado de aprendizaje 3: Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación	
Sem.	Contenidos	Estrategias	Indicadores de desempeño
5	<p>ESTRUCTURA DE SELECCIÓN O CONDICIONAL Expresiones Lógicas. Operadores relacionales y lógicos Estructura de Selección simple y doble. Sintaxis Algoritmos – Programas Estructura de Selección Simple y Doble (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elaboran algoritmos con estructura de selección para solucionar problemas sencillos teniendo en cuenta el análisis, diseño del algoritmo y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas con estructura de selección para solucionar problemas sencillos (cálculos matemáticos) considerando los elementos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos de selección simple y doble (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas con estructura selectiva simple y doble en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.
6	<p>Estructura de Selección Anidada. Algoritmos – Programas Estructura de Selección Anidada (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución).</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita.

		<ul style="list-style-type: none"> • Elaboran algoritmos con estructura de selección anidada para solucionar problemas teniendo en cuenta el análisis, diseño del algoritmo y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas con estructura de selección anidada para solucionar problemas considerando los elementos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos de selección anidada (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas con estructura selectiva anidada en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.
7	<p>Estructura de Selección Múltiple con selector. Sintaxis.</p> <p>Algoritmos – Programas Estructura de Selección Multiple (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elaboran algoritmos con estructura de selección múltiple para solucionar problemas teniendo en cuenta el análisis, diseño del algoritmo y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas con estructura de selección múltiple para solucionar problemas considerando los elementos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos de selección múltiple (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas con estructura selectiva múltiple en Java teniendo en cuenta la sintaxis, estructura, elementos y que los resultados sean los esperados a través de una práctica en clase.
8	Evaluación 2 y feedback	<ul style="list-style-type: none"> • Examen Desarrollo de Programas 	<ul style="list-style-type: none"> • Resuelve examen desarrollando programas con estructura selectiva en Java teniendo en cuenta la sintaxis, estructura, elementos y que los resultados sean los esperados.

Unidad didáctica 3: Estructura de control repetitiva: Algoritmos y Programación		Resultado de aprendizaje 4: Construye algoritmos y programas con estructura repetitiva, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.	
Sem.	Contenidos	Estrategias	Indicadores de desempeño
9	Contadores: Declarar, inicializar, incremento Acumuladores: Declarar, inicializar, incremento Estructura de repetición o bucles for. Sintaxis Estructura de repetición anidada for Algoritmos – Programas Estructura Repetitiva for (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elaboran algoritmos con estructura repetitiva for teniendo en cuenta el análisis, diseño del algoritmo y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas con estructura repetitiva for para solucionar problemas considerando los elementos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos con estructura repetitiva for (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas con estructura repetitiva for en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.
10	Estructura de repetición o bucles while. Sintaxis Estructura de repetición anidada while Algoritmos – Programas Estructura Repetitiva while (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elaboran algoritmos con estructura repetitiva while teniendo en cuenta el análisis, diseño del algoritmo y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas con estructura repetitiva while para solucionar problemas considerando los elementos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos con estructura repetitiva while (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas con estructura repetitiva while en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.

11	<p>Estructura de repetición o bucles do...while. Sintaxis Estructura de repetición anidada do...while</p> <p>Algoritmos – Programas Estructura Repetitiva do...while (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)</p> <p>Diferencias entre estructuras de repetición.</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elaboran algoritmos con estructura repetitiva do...while teniendo en cuenta el análisis, diseño del algoritmo y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas con estructura repetitiva do...while para solucionar problemas considerando los elementos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos con estructura repetitiva do...while (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas con estructura repetitiva do...while en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.
12	Evaluación 3 y feedback	<ul style="list-style-type: none"> • Examen Desarrollo de Programas 	<ul style="list-style-type: none"> • Resuelve examen desarrollando programas con estructura repetitiva en Java teniendo en cuenta la sintaxis, estructura, elementos y que los resultados sean los esperados.

Unidad didáctica 4: Subprogramas o Métodos y Arreglos unidimensionales.		Resultado de aprendizaje 5: Construye algoritmos y programas con arreglos unidimensionales, utilizando subprogramas o métodos y considerando el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.	
Sem.	Contenidos	Estrategias	Indicadores de desempeño
13	<p>Subprogramas o métodos.</p> <p>Algoritmos – Programas con Subprogramas o Métodos (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica). • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita.

		<ul style="list-style-type: none"> • Elaboran algoritmos con Subprogramas o Métodos teniendo en cuenta el análisis, diseño del algoritmo y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas con Subprogramas o Métodos para solucionar problemas considerando los elementos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos con Subprogramas o Métodos (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas con Subprogramas o Métodos en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.
14	<p>Arreglo Unidimensional o Vectores. Sintaxis Manipulación de índices. Arreglos de tipo numéricos. Operaciones básicas con Arreglos Unidimensionales: asignación, lectura, escritura, acceso secuencial y actualización</p> <p>Algoritmos – Programas con Arreglos Unidimensionales (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)</p>	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elabora algoritmos con operaciones básicas en arreglos unidimensionales o vectores (asignación, lectura, escritura, acceso secuencial y actualización) teniendo en cuenta el análisis, diseño del algoritmo (uso de Subprogramas o Métodos) y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas con operaciones básicas en arreglos unidimensionales o vectores (asignación, lectura, escritura, acceso secuencial y actualización) para solucionar problemas considerando los elementos, uso de Subprogramas o Métodos y 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos con operaciones básicas en arreglos unidimensionales o vectores (asignación, lectura, escritura, acceso secuencial y actualización) (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas con operaciones básicas en arreglos unidimensionales o vectores (asignación, lectura, escritura, acceso secuencial y actualización) en Java

		reglas de sintaxis del lenguaje Java. (Sincrónica)	teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.
15	Búsqueda de elementos de un vector. Algoritmos – Programas con Arreglos Unidimensionales (Análisis del problema, Diseño del algoritmo, Prueba de escritorio, Codificación en Java, Ejecución)	<ul style="list-style-type: none"> • Revisa los diferentes recursos subidos al SIA (Asincrónica) • Participa en forma activa en la exposición de los contenidos a través de videoconferencia del GoogleMeet y Chat (Sincrónica) • Elabora algoritmos de búsquedas sencillas en arreglos unidimensionales o vectores teniendo en cuenta el análisis, diseño del algoritmo (uso de Subprogramas o Métodos) y la prueba de escritorio, representándolos de diferentes formas; y considerando sus partes y elementos básicos. (Sincrónica) • Elabora programas de búsquedas sencillas en arreglos unidimensionales o vectores para solucionar problemas considerando los elementos, uso de Subprogramas o Métodos y reglas de sintaxis del lenguaje Java. (Sincrónica) 	<ul style="list-style-type: none"> • Interactúa participando y respondiendo preguntas en la videoconferencia o el Chat del Google Meet en forma oral o escrita. • Desarrolla practica grupal teniendo en cuenta las fases de análisis, diseño de algoritmos con búsquedas sencillas en arreglos unidimensionales o vectores (estructura y elementos) y prueba de escritorio con legibilidad y orden. • Desarrolla programas de búsquedas sencillas en arreglos unidimensionales o vectores en Java teniendo en cuenta la sintaxis, estructura y elementos y que los resultados sean los esperados a través de una práctica en clase.
16	Evaluación 4 y feedback	<ul style="list-style-type: none"> • Examen Desarrollo de Programas 	<ul style="list-style-type: none"> • Resuelve examen desarrollando programas con operaciones sencillas en arreglos unidimensionales o vectores en Java teniendo en cuenta la sintaxis, estructura, elementos y que los resultados sean los esperados.
17	Examen de Aplazados	<ul style="list-style-type: none"> • Examen Desarrollo de Programas 	<ul style="list-style-type: none"> • Resuelve examen desarrollando programas en Java teniendo en cuenta la sintaxis, estructura, elementos y que los resultados sean los esperados.

V. ESTRATEGIAS METODOLÓGICAS

De acuerdo con la naturaleza del curso, sus contenidos serán desarrollados en diferentes niveles de aprendizajes: aplicación, análisis, síntesis y evaluación a través de actividades diseñadas para mejorar el aprendizaje. Se pondrá mucho énfasis en métodos activos y de aprendizaje significativo presencial y/o virtual, entre otras se usarán las siguientes estrategias metodológicas:

Estrategias	Finalidad
Método Expositivo / Video Conferencia	Se transmiten conocimientos y activan procesos cognitivos en el estudiante.
Lluvia de ideas:	Se promueve una libre presentación de ideas sin restricciones ni limitaciones sobre un tema determinado.
Resolución de Problemas	Ejercitar, ensayar y poner en práctica los conocimientos previos para solucionar problemas sencillos de la vida diaria y computacionales.
Asesoría y laboratorio	El estudiante se enfrenta a casos de aplicación práctica y que demanda un trabajo analítico en la aplicación de las diferentes metodologías y habilidades en la utilización de las herramientas de software.
Trabajo en pequeños grupos	El estudiante debate, asimila, demuestra tolerancia con la finalidad de enriquecer el resultado del trabajo.
Aprendizaje Cooperativo	Desarrollar aprendizajes activos y significativos de forma cooperativa.

VI. EVALUACIÓN DEL APRENDIZAJE

La evaluación de la presente asignatura es de carácter integral donde se evaluarán los conocimientos adquiridos, su aplicación práctica y las actitudes frente al desarrollo de la asignatura

Evaluación	Resultados de aprendizaje	Evidencias	Instrumento de evaluación	Semana
Continua	R.A. 1: Explica los conceptos básicos y la importancia de los lenguajes de programación considerando su evolución, clasificación y características.	<ul style="list-style-type: none"> Participación interactiva en el Chat del Google Meet. Organizadores gráficos Prueba objetiva en Google Forms o Sistema de Respuesta en Clase. 	Lista de Cotejo Rubrica Cuestionario con clave de respuesta	1
	R.A. 2: Construye algoritmos y programas con estructura secuencial, considerando los	<ul style="list-style-type: none"> Participación interactiva en el Chat del Google Meet. Informe Grupal de Diseño de Algoritmos (Estructura Secuencial) 	Lista de Cotejo Lista de Cotejo o Rúbrica	2-3-4

<p>elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.</p>	<ul style="list-style-type: none"> • Practica Individual Desarrollo de programas (Estructura Secuencial) • Prueba objetiva en Google Forms o Sistema de Respuesta en Clase. • Examen Desarrollo de programas (Estructura Secuencial) 	<p>Lista de Cotejo o Rúbrica</p> <p>Cuestionario con clave de respuesta</p> <p>Lista de Cotejo o Rúbrica</p>	
<p>R.A. 3: Construye algoritmos y programas con estructura condicional, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación</p>	<ul style="list-style-type: none"> • Participación interactiva en el Chat del Google Meet. • Informe Grupal de Diseño de Algoritmos (Estructura de Selección o Condicional 1, 2 y 3) • Practica Individual Desarrollo de programas Estructura de Selección o Condicional 1,2 y 3) • Examen Desarrollo de programas (Estructura de Selección o Condicional) 	<p>Lista de Cotejo</p> <p>Lista de Cotejo o Rúbrica</p> <p>Lista de Cotejo o Rúbrica</p> <p>Lista de Cotejo o Rúbrica</p>	<p>5-6-7-8</p>
<p>R.A. 4: Construye algoritmos y programas con estructura repetitiva, considerando los elementos básicos, el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.</p>	<ul style="list-style-type: none"> • Participación interactiva en el Chat del Google Meet. • Informe Grupal de Diseño de Algoritmos (Estructura de Repetitiva 1, 2 y 3) • Practica Individual Desarrollo de programas (Estructura de Repetitiva 1, 2 y 3) • Examen Desarrollo de programas (Estructura de Repetitiva) 	<p>Lista de Cotejo</p> <p>Lista de Cotejo o Rúbrica</p> <p>Lista de Cotejo o Rúbrica</p> <p>Lista de Cotejo o Rúbrica</p>	<p>9-10-11-12</p>
<p>R.A. 5: Construye algoritmos y programas con arreglos unidimensionales, utilizando subprogramas o métodos y considerando el análisis del problema, diseño de la solución, prueba de escritorio e implementación en un lenguaje de programación.</p>	<ul style="list-style-type: none"> • Participación interactiva en el Chat del Google Meet. • Informe Grupal de Diseño de Algoritmos (Subprogramas o Métodos y Arreglos Unidimensionales o Vectores 1, 2 y 3) • Practica Individual Desarrollo de programas (Subprogramas o Métodos y Arreglos Unidimensionales o Vectores 1, 2 y 3) 	<p>Lista de Cotejo</p> <p>Lista de Cotejo o Rúbrica</p> <p>Lista de Cotejo o Rúbrica</p>	<p>13- 14- 15,16</p>

		<ul style="list-style-type: none"> Examen Desarrollo de programas (Subprogramas o Métodos y Arreglos Unidimensionales o Vectores 1, 2 y 3)) 	Lista de Cotejo o Rúbrica	
Recuperación/aplazado	R.A. 1 – R.A. 5	<ul style="list-style-type: none"> Examen Desarrollo de algoritmos y programas todo el contenido. 	Lista de Cotejo o Rúbrica.	17

- Todas las calificaciones son en escala vigesimal.
- El estudiante estará en la condición de inhabilitado con 30% o más de inasistencias del número real de sesiones desarrolladas en el ciclo, no teniendo derecho a rendir al examen de aplazados.
- Los calificativos con NP (No se presentó), equivale a cero (0). Las faltas justificadas o no justificadas se toman como inasistencias para calcular el porcentaje de inhabilitado; en caso se tramite la justificación respectiva solamente se recupera la evaluación no rendida.
- Los promedios finales se expresan en números enteros, en una escala de calificaciones de 0 a 20 puntos. La nota mínima aprobatoria es de once (11) puntos para todo tipo de evaluación. Únicamente para la nota final, la fracción de 0.5 punto o más se redondea a la unidad inmediatamente superior a favor del estudiante.
- Solamente tienen derecho a rendir el examen de aplazados los estudiantes con promedio promocional desaprobatorio, igual o mayor a 5 y registrar el 70% mínimo de asistencia.
- El examen de aplazados incluirá todos los contenidos del curso tanto de teoría como de práctica. La nota del examen de aplazado es la nota promocional del curso (La calificación máxima a obtener es once).
- El promedio final se calculará a partir de la siguiente fórmula:

$$PF = 0.25*EV1 + 0.25*EV2 + 0.25*EV3 + 0.25*EV4$$

Para efectos del presente curso, EV1, EV2, EV3 y EV4 se calcularán en las semanas 4, 8, 12 y 16 respectivamente, en base a las evaluaciones planificadas tanto en las sesiones de Teoría (50%) y Práctica (50%).

VII. REFERENCIAS

- Joyanes Aguilar, L. (2008). Fundamentos de Programación: Algoritmos, estructura de datos y objetos. 4ta Ed., Mc Graw Hill 2008.
- Marcelo Villalobos, R. (2008). Fundamentos de programación Java: más de 100 algoritmos codificados. Macro
- Deitel, P. J. y Deitel, H. M. (2013). Java. Cómo programar, 10a Ed., Pearson Educación, México

Cajamarca, julio del 2020.

Dr. Ing. Carlos Aparicio Arteaga

Dr. Ing. Manuel Malpica Rodríguez

Ms. Cs. Ing. Sandra Rodríguez Avila

Ms. Cs. Ing. Lisi Vásquez Fernández