UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

ESCUELA DE POSGRADO

UNIDAD DE POSGRADO DE LA FACULTAD DE CIENCIAS DE LA EDUCACIÓN

TESIS:

ChatGPT en aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la UNSCH, 2024

Para optar el grado académico de:
DOCTOR EN EDUCACIÓN

PRESENTADO POR:
Mtro. Amilcar TACURI GAMBOA

ASESOR:
Dr. Teodosio Zenobio POMA SOLIER

AYACUCHO - PERÚ

2026

DEDICATORIA

A mi familia

Por estar siempre presentes y ser ese apoyo que nunca me faltó en cada etapa de mi vida. Su cariño, sus palabras de ánimo y la confianza que siempre depositaron en mí fueron muy importantes para seguir adelante, incluso cuando las cosas se hicieron difíciles. Este logro también es de ustedes, porque detrás de cada paso estuvo su respaldo, su paciencia y ese ejemplo de esfuerzo que siempre me acompañó.

Más allá de haber alcanzado esta meta académica, quiero agradecerles por todo lo que me enseñaron, por estar presentes en cada etapa y por impulsarme a seguir creciendo como persona y como profesional. Gran parte de lo que hoy soy también lleva algo de ustedes. Con mucho cariño y agradecimiento, les dedico esta tesis.

AGRADECIMIENTO

Agradezco a las personas e instituciones que me acompañaron y aportaron durante el desarrollo de esta tesis doctoral. Cada aporte, orientación y muestra de respaldo fue importante para poder llegar hasta esta etapa. Llegar hasta este punto ha significado un esfuerzo personal importante, pero también ha sido resultado del apoyo, la orientación y el acompañamiento que recibí durante todo este proceso.

Agradezco de manera especial a la UNSCH de Huamanga, mi alma máter, por todo lo aprendido durante mi formación y por haberme brindado los espacios y recursos necesarios para continuar creciendo académica y profesionalmente. De igual manera, reconozco el aporte de mis docentes, quienes compartieron sus conocimientos, experiencias y orientaciones a lo largo de mi formación y contribuyeron de manera importante en esta investigación.

Agradezco a mi familia y mis amistades más cercanas, por acompañarme en este proceso. Su apoyo, sus motivaciones y la confianza que siempre tuvieron fueron fundamentales, sobre todo en aquellos momentos en los que el proceso se hacía más exigente. Su compañía me permitió mantener el equilibrio y la motivación necesaria para continuar hasta alcanzar esta meta.

Finalmente, valoro el apoyo de todos aquellos que hicieron algún aporte a este trabajo. A los estudiantes participantes, por su disposición y compromiso; a mis colegas, por sus observaciones críticas y sugerencias constructivas, y a todos quienes acompañaron este trayecto académico con generosidad y profesionalismo. A cada uno, mi más sincero agradecimiento.

ÍNDICE

PÁGINA DE JURADOS **4**
DECLARACIÓN JURADA DE AUTENTICIDAD **5**
DEDICATORIA **6**
AGRADECIMIENTO **7**
ÍNDICE DE TABLAS **11**
ÍNDICE DE ANEXOS **12**
RESUMEN **13**
ABSTRACT **14**
INTRODUCCIÓN **15**
1. PLANTEAMIENTO DEL PROBLEMA **17**
1.1. DESCRIPCIÓN DE LA SITUACIÓN PROBLEMÁTICA 17
1.2. FORMULACIÓN DEL PROBLEMA 19
1.2.1. Problema general *19*
1.2.2. Problemas específicos *19*
1.3. FORMULACIÓN DE OBJETIVOS 19
1.3.1. Objetivo general *19*
1.3.2. Objetivos específicos *19*
1.4. JUSTIFICACIÓN 20
2. MARCO TEÓRICO **21**
2.1 ANTECEDENTES 21
2.1.1. Antecedentes internacionales *21*
2.1.2. Antecedentes nacionales *24*
2.2 BASES TEÓRICAS 28
2.2.1. ChatGPT *28*
2.2.1.1. Efectividad pedagógica de ChatGPT 36
2.2.1.2. Integración tecnológica de ChatGPT en el proceso educativo 38
2.2.2. Aprendizaje de programación digital con Python *38*
2.2.2.1. Fundamentos de Python 40
2.2.2.2. Python aplicado a la programación digital 42
2.3 BASES CONCEPTUALES 44
3. METODOLOGÍA **46**
3.1. FORMULACIÓN DE HIPÓTESIS 46
3.1.1. Hipótesis general *46*
3.1.2. Hipótesis específicas *47*
3.2. VARIABLES 47
3.2.1. Variable independiente *47*

3.2.2. Variable dependiente *47*
3.3. OPERACIONALIZACIÓN DE VARIABLES 48
OPERACIONALIZACIÓN DE VARIABLES **49**
MATRIZ INSTRUMENTAL **51**
3.4. TIPO Y NIVEL DE INVESTIGACIÓN 52
3.4.1. Tipo de Investigación *52*
3.4.2. Nivel de investigación *52*
3.5. MÉTODOS 53
3.6. DISEÑO DE LA INVESTIGACIÓN 54
ESQUEMA QUE SE UTILIZARÁ EN LA INVESTIGACIÓN CUASI EXPERIMENTAL **55**
3.7. POBLACIÓN Y MUESTRA 55
3.7.1. Población *55*
POBLACIÓN **56**
3.7.2. Muestra *56*
MUESTRA **57**
3.7.3. Tipo de muestreo *57*
3.8. TÉCNICAS E INSTRUMENTOS 58
3.8.1. Técnicas *58*
3.8.2. Instrumentos *58*
3.9. VALIDEZ Y CONFIABILIDAD DE INSTRUMENTOS 59
3.9.1. Validez de instrumentos *59*
SISTEMATIZACIÓN DE VALORES ASIGNADOS POR EXPERTOS **60**
3.9.2. Confiabilidad de instrumentos *61*
3.10. TÉCNICAS DE PROCESAMIENTO DE DATOS 62
3.11. ASPECTOS ÉTICOS 64
4. RESULTADOS Y DISCUSIÓN **65**
4.1. RESULTADOS A NIVEL DESCRIPTIVO 65
NIVEL DE APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON EN ESTUDIANTES DE INGENIERÍA DE MINAS **65**
NIVEL DE DE APRENDIZAJE DE FUNDAMENTOS DE PYTHON EN ESTUDIANTES DE INGENIERÍA DE MINAS **66**
NIVEL DE APRENDIZAJE DE PYTHON APLICADO A LA PROGRAMACIÓN DIGITAL EN ESTUDIANTES DE INGENIERÍA DE MINAS **67**
4.2. RESULTADOS A NIVEL INFERENCIAL 68
4.2.1. Prueba de normalidad *68*
PRUEBA DE NORMALIDAD **68**
4.2.2. Prueba de hipótesis *69*

4.2.2.1. Prueba de hipótesis general 69
PRUEBA DE U MANN – WHITNEY CONTROL Y EXPERIMENTAL Y WILCOXON EN EL GRUPO DE APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON. **69**
4.2.2.2. Prueba de hipótesis específica 71
4.2.2.2.1. Prueba de hipótesis específica 1 71
PRUEBA DE U MANN – WHITNEY CONTROL Y EXPERIMENTAL Y WILCOXON EN LA DIMENSIÓN FUNDAMENTOS PYTHON. **71**
4.2.2.2.2. Prueba de hipótesis específica 2 72
PRUEBA DE U MANN – WHITNEY CONTROL Y EXPERIMENTAL Y WILCOXON EN LA DIMENSIÓN PYTHON APLICADO A LA PROGRAMACIÓN DIGITAL. **73**
4.3. DISCUSIÓN DE RESULTADOS 74
5. PROPUESTA PEDAGÓGICA **76**
5.1. INTRODUCCIÓN 76
5.2. FUNDAMENTO DE LA PROPUESTA 78
5.2.1. Fundamento filosófico *78*
5.2.2. Fundamento pedagógico *81*
5.2.3. Fundamento tecnológico *86*
5.3. Objetivo *90*
5.4. Descripción de la propuesta *90*
CONCLUSIONES **143**
RECOMENDACIONES **144**
REFERENCIAS BIBLIOGRÁFICAS **145**
ANEXOS **151**
ANEXO 1. MATRIZ DE CONSISTENCIA **151**
ANEXO 2. VALIDACIÓN DEL INSTRUMENTO **152**
ANEXO 3. AUTORIZACIÓN DE LA INSTITUCIÓN **156**
ANEXO 4. PROCESAMIENTO ESTADÍSTICO **157**
ANEXO 5. SÍLABOS **172**
ANEXO 6. FICHA DE OBSERVACIÓN **175**
ANEXO 7. INSTRUMENTO PRE TEST **177**
ANEXO 8. INSTRUMENTO POS TEST **180**
ANEXO 09. OPINIÓN DE EXPERTOS **183**
ANEXO 10. ENVÍO DE ARTICULO CIENTÍFICO A REVISTA INVECOM **184**
ANEXO11. ARTÍCULO CIENTÍFICO **201**

ÍNDICE DE TABLAS

TABLA 1 OPERACIONALIZACIÓN DE VARIABLES 49
TABLA 2 MATRIZ INSTRUMENTAL 51
TABLA 3 ESQUEMA QUE SE UTILIZARÁ EN LA INVESTIGACIÓN CUASI EXPERIMENTAL 55
TABLA 4 POBLACIÓN 56
TABLA 5 MUESTRA 57
TABLA 6 SISTEMATIZACIÓN DE VALORES ASIGNADOS POR EXPERTOS 60
TABLA 7 ALFA DE CRONBACH POR CADA ELEMENTO 62
TABLA 8 NIVEL DE APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON EN ESTUDIANTES DE INGENIERÍA DE MINAS 65
TABLA 9 NIVEL DE DE APRENDIZAJE DE FUNDAMENTOS DE PYTHON EN ESTUDIANTES DE INGENIERÍA DE MINAS 66
TABLA 10 NIVEL DE APRENDIZAJE DE PYTHON APLICADO A LA PROGRAMACIÓN DIGITAL EN ESTUDIANTES DE INGENIERÍA DE MINAS ... 67
TABLA 11 PRUEBA DE NORMALIDAD 68
TABLA 12 PRUEBA DE U MANN – WHITNEY CONTROL Y EXPERIMENTAL Y WILCOXON EN EL GRUPO DE APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON. 69
TABLA 1 PRUEBA DE U MANN – WHITNEY CONTROL Y EXPERIMENTAL Y WILCOXON EN LA DIMENSIÓN FUNDAMENTOS PYTHON. 71
TABLA 14 PRUEBA DE U MANN – WHITNEY CONTROL Y EXPERIMENTAL Y WILCOXON EN LA DIMENSIÓN PYTHON APLICADO A LA PROGRAMACIÓN DIGITAL. 73
TABLA 1 NIVEL DE APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON (POSTEST) 206

ÍNDICE DE ANEXOS

ANEXO 1. MATRIZ DE CONSISTENCIA **151**
ANEXO 2. VALIDACIÓN DEL INSTRUMENTO **152**
ANEXO 3. AUTORIZACIÓN DE LA INSTITUCIÓN **156**
ANEXO 4. PROCESAMIENTO ESTADÍSTICO **157**
ANEXO 5. SÍLABOS **172**
ANEXO 6. FICHA DE OBSERVACIÓN **175**
ANEXO 7. INSTRUMENTO PRE TEST **177**
ANEXO 8. INSTRUMENTO POS TEST **180**
ANEXO 09. OPINIÓN DE EXPERTOS **183**
ANEXO 10. ENVÍO DE ARTICULO CIENTÍFICO A REVISTA INVECOM **184**
ANEXO11. ARTÍCULO CIENTÍFICO **201**

RESUMEN

La investigación se orientó a determinar los efectos generados por ChatGPT sobre el proceso de aprendizaje de Programación Digital empleando Python en estudiantes de Ingeniería de Minas pertenecientes a la Universidad Nacional de San Cristóbal de Huamanga durante el semestre 2024-II. Para desarrollar el estudio se adoptó un enfoque aplicado y de nivel explicativo, siguiendo un diseño cuasiexperimental sustentado en el método hipotético-deductivo. Participaron 40 estudiantes, organizados en dos grupos conformados por 20 integrantes cada uno: un grupo experimental, que incorporó ChatGPT como apoyo durante su proceso formativo, y un grupo de control, cuya enseñanza se desarrolló mediante una metodología convencional. El dominio teórico y práctico de Python fue valorado mediante una ficha de observación, mientras que el procesamiento estadístico se realizó utilizando pruebas no paramétricas, debido a que los datos presentaron una distribución distinta de la normal.

Al analizar los resultados del postest mediante la prueba U de Mann-Whitney, se identificaron diferencias estadísticamente significativas entre los grupos evaluados, siendo el grupo experimental el que alcanzó las puntuaciones más altas ($p < 0,001$). Por otra parte, la prueba de Wilcoxon permitió comprobar que las calificaciones de este grupo mejoraron después de la intervención, en comparación con las registradas al inicio del estudio ($p < 0,001$; $Z = -3,935$). A partir de estos valores, se contó con evidencia estadística suficiente para rechazar la hipótesis nula en favor de la hipótesis alternativa.

En ese sentido, los hallazgos indican que el uso de ChatGPT favoreció el proceso de aprendizaje de Python, tanto en la asimilación de los conceptos básicos como en el desarrollo de actividades prácticas de programación. Lo observado también permite reconocer la utilidad de las herramientas de inteligencia artificial en la enseñanza universitaria de ingeniería, principalmente como un apoyo que facilita el trabajo independiente del estudiante y le brinda orientación durante su formación académica.

Palabra clave: ChatGPT, fundamentos de Python, programación digital, aprendizaje autónomo, inteligencia artificial educativa.

ABSTRACT

This research aimed to determine the effects of ChatGPT on the process of learning digital programming with Python among Mining Engineering students at the National University of San Cristóbal de Huamanga during the 2024-II semester. The study employed an applied approach at an explanatory level, utilizing a quasi-experimental design based on the hypothetico-deductive method. Forty students participated, divided into two groups of 20: an experimental group that incorporated ChatGPT as a support tool during their training, and a control group taught using a conventional methodology. Theoretical and practical proficiency in Python was assessed via an observation checklist, while statistical analysis was conducted using non-parametric tests, as the data did not follow a normal distribution.

The posttest results, analyzed using the Mann-Whitney U test, revealed statistically significant differences between the two groups, with the experimental group achieving higher scores ($p < 0.001$). In addition, the Wilcoxon signed-rank test confirmed an improvement in the experimental group's performance after the intervention compared with the scores obtained at the beginning of the study ($p < 0.001$; $Z = -3.935$). These findings provided sufficient statistical evidence to reject the null hypothesis in favor of the alternative hypothesis.

Based on these findings, the use of ChatGPT had a positive effect on the Python learning process, supporting both the understanding of fundamental concepts and the development of practical programming activities. The results also highlight the potential of artificial intelligence tools in university-level engineering education, particularly as a complementary resource that encourages independent learning and provides students with ongoing guidance throughout their academic training.

Keywords: ChatGPT, Python fundamentals, digital programming, autonomous learning, educational artificial intelligence.

INTRODUCCIÓN

Actualmente, la tecnología tiene cada vez más presencia en distintos espacios de la vida diaria y la educación superior no es ajena a ese cambio. Las universidades han tenido que incorporar nuevas herramientas digitales para apoyar la enseñanza y también para contribuir al aprendizaje de los estudiantes. Dentro de este proceso, la inteligencia artificial ha comenzado a ocupar un lugar importante y herramientas como ChatGPT han llamado bastante la atención por la posibilidad que ofrecen de acompañar el aprendizaje en cursos técnicos.

Uno de estos cursos es Programación Digital con Python. Aunque Python es un lenguaje bastante utilizado en diferentes áreas de la ingeniería por su facilidad de uso y por la variedad de aplicaciones que permite desarrollar, aprender a programar no siempre resulta sencillo. El estudiante necesita comprender los conceptos, pero también saber cómo aplicarlos cuando se presenta un problema concreto. En el caso de los estudiantes de Ingeniería de Minas, se han observado dificultades para desarrollar estas habilidades, lo cual puede verse reflejado tanto en su rendimiento académico como en la forma en que enfrentan situaciones prácticas relacionadas con su formación profesional.

Ante esta situación, la investigación buscó determinar de qué manera influye la utilización de ChatGPT durante el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga, correspondiente al semestre 2024-II. La investigación se centró en el uso de ChatGPT como una herramienta de apoyo para el aprendizaje de Python, con la finalidad de fortalecer la comprensión de los temas desarrollados, favorecer el rendimiento académico de los estudiantes y facilitar la realización de ejercicios prácticos relacionados con la programación. En el aspecto metodológico, se desarrolló una investigación de tipo aplicada, de nivel explicativo, siguiendo el método hipotético-deductivo y un diseño cuasiexperimental. Para el estudio se contó con la participación de 40 estudiantes, organizados en dos grupos de 20 integrantes cada uno. En el grupo experimental se incorporó ChatGPT como recurso complementario durante las actividades de aprendizaje, mientras que el grupo de control recibió las clases siguiendo los procedimientos habituales de enseñanza, sin utilizar esta herramienta. La información fue obtenida mediante una ficha de

observación validada por especialistas, orientada a valorar dos aspectos: el conocimiento de los fundamentos básicos de Python y la ejecución adecuada de estos conocimientos en actividades prácticas.

El trabajo fue estructurado en cinco capítulos. En el primero se presenta el problema de investigación, los objetivos, las hipótesis y la justificación del estudio. El segundo reúne las bases teóricas y los antecedentes relacionados con el tema. En el tercer capítulo se desarrolla el procedimiento metodológico, incluyendo el diseño, las variables y los instrumentos utilizados para recoger la información. El cuarto capítulo presenta los resultados obtenidos y su análisis mediante estadística descriptiva y pruebas no paramétricas. Finalmente, el quinto capítulo contiene las propuestas formuladas a partir de los principales resultados alcanzados durante la investigación.

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción de la situación problemática

La problemática central de esta investigación se encuentra vinculada con las dificultades que presentan los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga (UNSCH) para desarrollar competencias en programación digital mediante el uso del lenguaje Python. Esta situación no se presenta únicamente en el ámbito local, sino que también forma parte de una dificultad más amplia que se observa en otros países y regiones, donde todavía existe una brecha importante en el aprendizaje de programación, a pesar de que estas competencias son cada vez más necesarias dentro de la ingeniería.

A nivel mundial, aprender programación, y en especial trabajar con Python, sigue siendo complicado para muchos estudiantes de carreras técnicas. Aunque Python es considerado un lenguaje bastante accesible y versátil, y además tiene muchas aplicaciones en ingeniería, ciencia de datos y tecnología, su aprendizaje no siempre resulta sencillo. En varios casos, los estudiantes tienen problemas para pasar de la teoría a la práctica, sobre todo cuando la enseñanza se concentra demasiado en conceptos y ejercicios repetitivos, sin incorporar suficientes situaciones reales o herramientas que ayuden a comprender mejor cómo utilizar el lenguaje.

En América Latina esta situación puede hacerse más evidente por algunas limitaciones que todavía existen dentro de las universidades. Entre ellas se encuentran la disponibilidad reducida de recursos tecnológicos, la falta de capacitación continua de algunos docentes en herramientas actuales de programación y planes de estudio que no siempre avanzan al mismo ritmo que las necesidades del mercado laboral. Otro aspecto que debe considerarse es la desigualdad en el acceso a los recursos tecnológicos, pues no todos los estudiantes disponen de computadoras, programas informáticos o herramientas digitales actualizadas en las mismas condiciones. Esta situación puede generar dificultades para realizar ejercicios de programación, practicar fuera del horario de clases y fortalecer los conocimientos adquiridos. Como consecuencia, algunos estudiantes podrían tener menos oportunidades para desarrollar las competencias digitales que necesitarán durante su formación universitaria y, posteriormente, en el ejercicio de su profesión. En el caso de Ingeniería de Minas, esta situación cobra especial importancia, pues el ejercicio profesional requiere cada vez más del manejo de herramientas digitales para realizar cálculos, procesar información y resolver problemas propios de las operaciones mineras.

En la Universidad Nacional de San Cristóbal de Huamanga se observa una realidad similar, especialmente entre los estudiantes de Ingeniería de Minas, quienes presentan dificultades para aprender a programar en Python. Entre los factores que podrían explicar esta situación se encuentran la disponibilidad limitada de materiales y recursos educativos apropiados, las pocas oportunidades de capacitación en herramientas de programación actuales y la falta de estrategias de enseñanza que ayuden a los estudiantes a analizar problemas y buscar soluciones por cuenta propia. Si bien durante su formación reciben conocimientos teóricos básicos, llevarlos a la práctica no siempre resulta sencillo, sobre todo cuando deben resolver situaciones relacionadas con la ingeniería. Esta dificultad termina limitando el desarrollo de sus habilidades de programación y el aprovechamiento de lo aprendido en las clases.

Las dificultades identificadas pueden explicarse por diversos factores que influyen en el proceso de aprendizaje de los estudiantes. Uno de ellos es la forma tradicional de enseñanza, que muchas veces se concentra más en la explicación teórica y deja poco espacio para actividades prácticas o el uso de recursos interactivos. También influye que algunos planes de estudio no siempre se actualizan al mismo ritmo que los cambios tecnológicos, además de que no todos los docentes cuentan con una preparación suficiente en herramientas de programación como Python. Todo esto termina generando cierta distancia entre lo que el estudiante aprende en el aula y lo que después se le exige en el campo laboral.

Como consecuencia, os estudiantes de Ingeniería de Minas pueden encontrar más dificultades al momento de enfrentar problemas que necesitan programación y manejo de información. Esta limitación también reduce sus posibilidades de trabajar con herramientas que hoy tienen cada vez más presencia en minería, como la automatización de tareas, el manejo de información geoespacial y la simulación de procesos. Por eso, fortalecer estas competencias resulta importante para que la formación profesional responda mejor a las nuevas exigencias que se presentan dentro de la ingeniería.

El aporte de esta investigación radica en explorar el empleo de ChatGPT como una herramienta digital orientada a fortalecer el aprendizaje de programación digital con Python en los estudiantes de Ingeniería de Minas. Se propone que, mediante la integración de esta inteligencia artificial, los estudiantes pueden recibir orientación sobre su trabajo instantánea, resolver dudas en tiempo real, y recibir una enseñanza más personalizada y adaptativa. Con esta propuesta se pretende fortalecer la comprensión de Python y facilitar que los estudiantes lleven esos conocimientos a situaciones prácticas propias de la ingeniería, contribuyendo de

esta manera a superar las barreras en la enseñanza de programación y a cerrar la brecha entre la academia y el mercado laboral.

1.2. Formulación del problema

1.2.1. Problema general

¿En qué medida el uso de ChatGPT influye en el aprendizaje de Programación digital con Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024 -II?

1.2.2. Problemas específicos

a. ¿En qué medida el uso de ChatGPT influye en el aprendizaje de fundamentos de Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II?

b. ¿En qué medida el uso de ChatGPT influye en el aprendizaje de Python aplicado a la programación digital en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II?

1.3. Formulación de objetivos

1.3.1. Objetivo general

Determinar la influencia del uso de ChatGPT en el aprendizaje de Programación Digital con Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II.

1.3.2. Objetivos específicos

a. Comparar la influencia del uso de ChatGPT en el aprendizaje de fundamentos de Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II.

b. Verificar la influencia del uso de ChatGPT en el aprendizaje de Python aplicado a la Programación Digital en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II.

1.4. Justificación

Valor teórico

La investigación busca fortalecer el aprendizaje de Programación Digital con Python mediante la incorporación de ChatGPT como recurso de apoyo en la formación de estudiantes pertenecientes a la carrera de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga. La propuesta pedagógica surgió de la necesidad de incorporar recursos tecnológicos que respondan a las nuevas formas de enseñar y aprender, considerando especialmente las posibilidades que ofrece la inteligencia artificial en la educación universitaria. En este marco, ChatGPT se planteó como un apoyo adicional para desarrollar las actividades académicas, más allá de su función como herramienta digital. Su utilización buscó que los estudiantes pudieran resolver dudas, revisar explicaciones y reforzar aquellos temas que les resultaban difíciles de comprender. Asimismo, se procuró que cada participante tuviera la oportunidad de avanzar a su propio ritmo, dedicando mayor atención a los contenidos que necesitaba mejorar durante su proceso de aprendizaje.

La investigación tomó en cuenta los cambios que se están presentando en el ámbito educativo y profesional debido al desarrollo de nuevas tecnologías. Estas transformaciones hacen necesario revisar las formas tradicionales de enseñanza y considerar recursos que ayuden a los estudiantes a adquirir conocimientos y habilidades acordes con las exigencias de su formación. En ese sentido, el estudio se orientó a conocer en qué medida ChatGPT puede contribuir al aprendizaje de programación con Python, ofreciendo una alternativa para complementar las explicaciones del docente y facilitar la comprensión de los contenidos. También se buscó evaluar sus posibilidades para apoyar el desarrollo de ejercicios prácticos, atender las dificultades individuales y promover una participación más activa de los estudiantes durante las actividades académicas.

Implicancia práctica

En cuanto a su utilidad práctica, esta investigación busca aportar información que permita considerar nuevas alternativas para la enseñanza de programación con Python, especialmente mediante la incorporación de ChatGPT como recurso complementario en las actividades académicas. Su uso brinda a los estudiantes la oportunidad de consultar aquellos aspectos que no lograron comprender durante las explicaciones del docente, revisar ejemplos y encontrar orientación cuando surgen dificultades al desarrollar sus ejercicios. De esta manera, se busca que los participantes dispongan de mayores oportunidades para reforzar sus conocimientos y comprender los procedimientos de programación que requieren más

práctica y atención, favoreciendo una participación más independiente durante su aprendizaje.

Asimismo, la utilización de ChatGPT puede favorecer que los estudiantes lleven a la práctica los conocimientos adquiridos durante las clases de programación con Python, especialmente mediante ejercicios relacionados con situaciones propias de la Ingeniería de Minas. En esta especialidad, el manejo de herramientas informáticas resulta útil para organizar información, procesar registros operativos, realizar cálculos y analizar problemas técnicos. Por ello, se busca que los estudiantes comprendan que aprender a programar no consiste únicamente en conocer instrucciones o escribir códigos, sino también en saber cómo utilizarlos para resolver necesidades concretas. Esta relación entre la formación académica y las actividades que podrían desempeñar en el futuro permitiría dar mayor sentido al aprendizaje de Python y fortalecer su preparación para el ejercicio profesional.

Unidad metodológica

En el aspecto metodológico, el estudio se desarrolló mediante un diseño cuasiexperimental, que permitió evaluar los efectos de la incorporación de ChatGPT en el aprendizaje de programación con Python. Para ello, se compararon los resultados obtenidos por los estudiantes del grupo experimental y del grupo de control, considerando sus niveles de desempeño antes y después de la intervención educativa. Este diseño permitirá obtener información objetiva mediante la comparación de los niveles de desempeño evidenciados por los participantes antes y después de incorporar la estrategia propuesta.

La información recogida también puede ser tomada como referencia para conocer con mayor claridad qué tan útil puede ser una herramienta de inteligencia artificial dentro del proceso de enseñanza. Además, la metodología empleada puede ser tomada como base en futuras investigaciones que busquen estudiar el uso de tecnologías similares en el desarrollo de capacidades técnicas que necesitan práctica y acompañamiento constante.

2. MARCO TEÓRICO

2.1 Antecedentes

2.1.1. Antecedentes internacionales

Althobaiti, M. (2021). *Implications of Artificial Intelligence for Mathematics Education: Insights from the United Arab Emirates.* International Journal of Technology in Education and Science. tuvo como propósito examinar las implicancias asociadas con la incorporación de recursos basados en inteligencia artificial dentro de la enseñanza de las

matemáticas en los Emiratos Árabes Unidos. La investigación presentó un diseño exploratorio y descriptivo, y la evidencia analizada procedió principalmente de una revisión de publicaciones vinculadas con la temática.

Los hallazgos indicaron que la inteligencia artificial puede aportar de manera significativa al desarrollo de la educación matemática en dicho contexto. Se identificó que estas tecnologías pueden emplearse como apoyo tanto al trabajo docente como al proceso de aprendizaje de las matemáticas. Además, pueden facilitar el seguimiento del desempeño académico y brindar orientaciones más ajustadas a las necesidades particulares de los estudiantes. El autor también encontró una disposición positiva, por parte de docentes y estudiantes, frente a la integración de soluciones de inteligencia artificial en las actividades relacionadas con la enseñanza y el aprendizaje de las matemáticas.

Gao, Wang y Xie (2021). *Evaluating the Impact of Artificial Intelligence on Mathematics Learning in High School Students.* Journal of Educational Computing Research. La investigación tuvo como propósito examinar la influencia de la incorporación de inteligencia artificial como apoyo al aprendizaje de las matemáticas en estudiantes de educación secundaria de China. Para ello, desarrollaron una investigación con diseño cuasiexperimental, conformada por un grupo experimental y otro de control. La muestra estuvo integrada por 100 alumnos pertenecientes a una institución de educación secundaria, quienes fueron distribuidos aleatoriamente entre ambos grupos.

El grupo experimental participó en un proceso de aprendizaje respaldado por un modelo basado en inteligencia artificial, en tanto que el grupo de control mantuvo la modalidad habitual de enseñanza. Para identificar los cambios producidos en el aprendizaje, se aplicaron evaluaciones en dos momentos: previo y posterior a la intervención. Dichas pruebas estuvieron orientadas a evaluar el nivel de conocimiento y comprensión alcanzado por los participantes en diferentes contenidos matemáticos.

Los hallazgos evidenciaron una mejora más marcada en el grupo experimental respecto del grupo de control. A partir de estos resultados, los autores señalaron que la integración de inteligencia artificial dentro de la enseñanza de las matemáticas puede favorecer el desempeño académico y el aprendizaje de los estudiantes..

Sharma, N. y Kumar, N. (2021). *Teachers´ attitudes toward the use or artificial intelligence in education in India: An empirical study.* International Journal of Educational Technology in Higher Education. La investigación tuvo como finalidad examinar las

actitudes de los docentes frente a la incorporación de inteligencia artificial en la educación superior de India. Para abordar este propósito, los autores emplearon un enfoque cuantitativo y utilizaron un cuestionario estructurado como instrumento de recolección de información.

El cuestionario estuvo organizado en cuatro secciones. La primera recogió información personal de los participantes; la segunda permitió conocer sus opiniones y actitudes respecto a la aplicación de inteligencia artificial en el ámbito educativo; la tercera estuvo destinada a identificar los factores asociados con dichas actitudes; y la cuarta incorporó preguntas abiertas para que los docentes expresaran comentarios adicionales.

Participaron 300 docentes procedentes de instituciones de educación superior de India, elegidos mediante un procedimiento de muestreo aleatorio estratificado. Los resultados indicaron que la mayoría de los participantes presentaba una valoración favorable hacia la integración de estas tecnologías en los procesos educativos, aunque algunos docentes manifestaron dudas y ciertas reservas. También se identificó que aspectos como la edad, la experiencia profesional y la formación recibida podían influir en la manera en que los docentes percibían estas herramientas. Finalmente, una proporción importante de los participantes expresó interés en recibir capacitación relacionada con la aplicación de herramientas de inteligencia artificial dentro de la práctica educativa.

Smith, J. (2020). *Aplicación de la inteligencia artificial en la enseñanza de cálculo en la universidad en Estados Unidos.* Revista de Investigación Educativa. La investigación tuvo como objetivo examinar los efectos de la incorporación de inteligencia artificial como apoyo al aprendizaje de cálculo en alumnos universitarios de Estados Unidos.

Para desarrollar el estudio se empleó un diseño experimental, mediante el cual se comparó el desempeño académico de dos grupos. Uno de ellos continuó con una metodología de enseñanza convencional, mientras que el otro trabajó con herramientas basadas en inteligencia artificial. Asimismo, se recogieron las opiniones de los estudiantes acerca de la experiencia desarrollada durante el proceso de aprendizaje. La información fue obtenida mediante pruebas estandarizadas de cálculo y cuestionarios orientados a conocer la percepción de los participantes respecto al empleo de inteligencia artificial en el curso.

La muestra estuvo constituida por dos grupos de estudiantes de cálculo pertenecientes a una universidad de Estados Unidos. Mientras uno recibió enseñanza tradicional, el otro desarrolló las actividades académicas con apoyo de inteligencia artificial. Los hallazgos mostraron que los participantes del grupo que utilizó estas herramientas

alcanzaron un mejor desempeño académico frente a quienes siguieron la metodología tradicional. Además, manifestaron una percepción favorable sobre el uso de inteligencia artificial en su aprendizaje y valoraron de manera positiva la experiencia desarrollada durante el curso.

González, M. (2019). *Uso de la inteligencia artificial en la enseñanza del álgebra en estudiantes de educación secundaria en México.* Revista de Investigación Educativa. La investigación buscó determinar de qué manera influye la incorporación de inteligencia artificial en el aprendizaje del álgebra en estudiantes de educación secundaria de México. Para desarrollar el estudio se empleó un diseño cuasiexperimental y se establecieron dos grupos de comparación: uno experimental y otro de control.

Los participantes del grupo experimental desarrollaron las sesiones de álgebra utilizando un software basado en inteligencia artificial, en tanto que el grupo de control mantuvo la metodología convencional de enseñanza. Para identificar los cambios producidos en el aprendizaje se aplicaron dos evaluaciones de álgebra, una antes de iniciar la intervención y otra al concluirla. Además, se empleó una encuesta orientada a identificar la percepción y el grado de satisfacción de los estudiantes respecto a la utilización de inteligencia artificial durante las clases.

Participaron 80 estudiantes de educación secundaria pertenecientes a dos escuelas de México, quienes fueron asignados aleatoriamente a los grupos de estudio. Los resultados evidenciaron que los estudiantes que utilizaron la herramienta de inteligencia artificial alcanzaron un mejor desempeño en las habilidades de álgebra respecto de quienes integraron el grupo de control. Asimismo, manifestaron niveles más altos de satisfacción y una valoración positiva de la inteligencia artificial como recurso de apoyo durante el aprendizaje de esta asignatura.

2.1.2. Antecedentes nacionales

Rojas y Espinoza (2021). *Impacto de la inteligencia artificial en la resolución de problemas matemáticos en estudiantes de secundaria en Lima.* Revista de educación. La investigación tuvo como finalidad determinar cómo la incorporación de inteligencia artificial incide en la resolución de ejercicios matemáticos en estudiantes de secundaria de Lima. Para ello, emplearon un diseño cuasiexperimental con dos grupos de comparación, uno experimental y otro de control. La intervención tuvo una duración de tres meses.

Para identificar los cambios producidos durante el estudio se aplicaron dos

evaluaciones. La primera se realizó antes de iniciar la intervención y la segunda al término de los tres meses de trabajo. Esta comparación permitió observar las variaciones existentes entre los resultados iniciales y finales.

Se trabajó con una muestra no probabilística por conveniencia integrada por 80 estudiantes de secundaria pertenecientes a dos colegios de Lima. Los participantes fueron distribuidos aleatoriamente entre ambos grupos, con 40 estudiantes en cada uno. Los resultados evidenciaron que el grupo experimental, que utilizó inteligencia artificial como apoyo durante las actividades, alcanzó un rendimiento superior en la evaluación final respecto al grupo de control. A partir de estos hallazgos, los autores concluyeron que la aplicación de inteligencia artificial produjo una influencia significativa en el fortalecimiento de la capacidad para resolver problemas matemáticos en los participantes.

Valdivia, J. y Cabrera, L. (2021). *Uso de inteligencia artificial para mejorar el aprendizaje matemáticas en estudiantes universitarios.* Revista de Tecnología e Innovación Educativa. La investigación se orientó a evaluar los efectos de la incorporación de inteligencia artificial como apoyo al aprendizaje de las matemáticas en alumnos universitarios.

Para el desarrollo del estudio se aplicó un diseño experimental con mediciones antes y después de la intervención, incluyendo un grupo de control como referencia comparativa. Participaron 80 estudiantes pertenecientes a distintas facultades universitarias, quienes fueron distribuidos aleatoriamente en dos grupos de estudio. El grupo experimental utilizó MathAI, un software basado en inteligencia artificial, como recurso complementario durante las sesiones de matemáticas, en tanto que el grupo de control mantuvo la metodología convencional de enseñanza.

El proceso de intervención tuvo una duración de un semestre académico. Para evaluar el progreso de los participantes se aplicó una prueba diagnóstica al inicio y otra evaluación al finalizar el periodo de trabajo. Además, se empleó una encuesta de satisfacción destinada a recoger la percepción de los estudiantes acerca del uso de MathAI.

Los hallazgos evidenciaron un progreso estadísticamente significativo en el aprendizaje de las matemáticas dentro del grupo experimental frente al grupo de control. De igual manera, los participantes que trabajaron con MathAI expresaron un mayor nivel de

satisfacción con la forma en que se desarrolló la enseñanza mediante el apoyo de inteligencia artificial.

Torres, J. (2020). *Efecto de la inteligencia artificial en el rendimiento académico en matemáticas en estudiantes de educación secundaria en el Perú.* Revista de Educación. La investigación tuvo como propósito examinar la influencia de la inteligencia artificial sobre el desempeño académico en matemáticas de estudiantes de educación secundaria en Perú.

Para desarrollar la investigación se empleó un diseño cuasiexperimental conformado por dos grupos de comparación. El grupo experimental desarrolló las clases utilizando herramientas basadas en inteligencia artificial, en tanto que el grupo de control mantuvo la metodología convencional de enseñanza, sin incorporar este tipo de recursos. Participaron 100 estudiantes de educación secundaria, elegidos mediante un procedimiento aleatorio y posteriormente distribuidos entre ambos grupos.

La evaluación del rendimiento se realizó mediante dos pruebas, una antes del inicio de la intervención y otra al finalizarla. Ambas incluyeron problemas matemáticos correspondientes al nivel secundario y estuvieron orientadas a valorar el nivel de comprensión alcanzado por los estudiantes en contenidos considerados relevantes. Posteriormente, los resultados de ambos grupos fueron contrastados mediante análisis estadístico.

Los hallazgos evidenciaron que los participantes del grupo experimental obtuvieron un desempeño académico superior respecto de quienes integraron el grupo de control. Las diferencias fueron más notorias en aspectos vinculados con la capacidad para resolver problemas y comprender conceptos matemáticos, lo que permitió identificar una influencia favorable de la inteligencia artificial dentro del proceso de aprendizaje.

González, J. (2020). *Uso de la inteligencia artificial en la evaluación de aprendizajes en la enseñanza de la matemática.* Revista Digital de Investigación en Docencia Universitaria. La investigación se orientó a evaluar la efectividad de la inteligencia artificial aplicada a la valoración del aprendizaje de las matemáticas.

Para desarrollar la investigación se aplicó un diseño experimental con mediciones previas y posteriores, considerando dos grupos de estudiantes: uno experimental y otro de control. El grupo de control fue evaluado mediante procedimientos convencionales, en tanto

que el grupo experimental empleó una plataforma de evaluación basada en inteligencia artificial. Ambos grupos recibieron los mismos contenidos de matemática durante cuatro semanas.

Antes de iniciar el experimento se aplicó un cuestionario de conocimientos previos con la finalidad de identificar el grado de dominio de los contenidos matemáticos de los participantes. La plataforma basada en inteligencia artificial constituyó el principal instrumento de evaluación para el grupo experimental, ya que permitió valorar el desempeño de los estudiantes en diferentes tareas matemáticas y proporcionar retroalimentación inmediata. Al concluir la intervención se aplicó una evaluación destinada a valorar los aprendizajes alcanzados por ambos grupos.

La muestra estuvo integrada por 60 estudiantes universitarios de matemática, distribuidos en dos grupos de 30 participantes cada uno. Los hallazgos evidenciaron que quienes utilizaron la plataforma de evaluación basada en inteligencia artificial obtuvieron un mejor desempeño en las tareas matemáticas y alcanzaron puntuaciones superiores en la evaluación final respecto de quienes integraron el grupo de control. Además, la retroalimentación inmediata y personalizada proporcionada por la plataforma fue identificada como uno de los factores asociados con el incremento observado en el desempeño académico de los participantes.

Vargas, J. (2019). *Aplicación de la inteligencia artificial en la resolución de problemas matemáticos en estudiantes universitarios peruanos.* Revista Internacional de investigación en Ciencias Sociales. La investigación se orientó a determinar los efectos de la incorporación de inteligencia artificial en la solución de problemas matemáticos en estudiantes universitarios del Perú.

Para desarrollar la investigación se aplicó un diseño cuasiexperimental con grupos no equivalentes. Participaron 70 estudiantes pertenecientes a dos universidades del país, elegidos mediante un procedimiento de muestreo no probabilístico por conveniencia. Los participantes fueron distribuidos en un grupo experimental y otro de control, ambos integrados por 35 estudiantes. La asignación no se realizó de manera aleatoria.

Durante 10 semanas, el grupo experimental utilizó herramientas basadas en inteligencia artificial como apoyo para resolver problemas matemáticos, en tanto que el

grupo de control mantuvo la metodología habitual de enseñanza. Para recoger la información se utilizaron dos instrumentos. El primero correspondió a una prueba de rendimiento académico orientada a valorar el desempeño en la solución de problemas matemáticos; el segundo fue un cuestionario sobre actitudes hacia las matemáticas, empleado para conocer la percepción de los estudiantes respecto de la asignatura y de sus propias capacidades para desenvolverse en ella. Los dos instrumentos pasaron previamente por procesos de validación y estimación de confiabilidad.

Los resultados evidenciaron que el grupo experimental obtuvo un desempeño académico estadísticamente superior al del grupo de control después de la intervención. También se observó que los estudiantes que utilizaron inteligencia artificial mostraron actitudes más favorables hacia las matemáticas y una percepción más positiva de sus propias habilidades en esta área. Para el tratamiento de los datos se emplearon ANOVA y pruebas post hoc, mediante las cuales se identificaron las diferencias existentes entre ambos grupos.

2.2 Bases teóricas

2.2.1. ChatGPT

El desarrollo reciente de modelos avanzados de procesamiento del lenguaje, entre ellos ChatGPT ha revolucionado no solo la forma en que las máquinas interactúan con los humanos, sino también cómo se accede y utiliza la información en diversos campos. ChatGPT es una herramienta de inteligencia artificial desarrollada por OpenAI, cuyo funcionamiento se basa en modelos de lenguaje que emplean la arquitectura de transformadores. Estos modelos han sido entrenados con grandes cantidades de información, lo que les permite interpretar las consultas de los usuarios y elaborar respuestas en lenguaje En el ámbito académico, el uso de modelos de lenguaje como ChatGPT ofrece a los estudiantes la posibilidad de realizar consultas sobre temas específicos y obtener respuestas que pueden ayudarles a comprender contenidos de distintas disciplinas, desde las ciencias básicas hasta las áreas de formación especializada. Brown et al. (2020) explican que estos modelos tienen la capacidad de desarrollar tareas relacionadas con la comprensión del lenguaje y la resolución de preguntas a partir de instrucciones o ejemplos. Por su parte, Radford et al. (2019) analizaron cómo los modelos de lenguaje pueden abordar actividades de comprensión lectora, interpretación de información y respuesta a preguntas sin requerir un entrenamiento particular para cada tarea. Por estas características, las herramientas de inteligencia artificial pueden emplearse como recursos complementarios en las actividades académicas, especialmente cuando los estudiantes necesitan aclarar dudas o profundizar en

determinados temas. Sin embargo, es importante que las respuestas obtenidas sean revisadas y comparadas con bibliografía especializada, debido a que no toda la información generada es necesariamente correcta o cuenta con suficiente sustento científico.

ChatGPT también ofrece posibilidades para complementar las actividades de enseñanza, pues permite que los estudiantes consulten sus dudas y revisen aquellos contenidos que no lograron comprender con claridad durante las clases. Su utilización puede resultar especialmente útil cuando se necesita una explicación adicional o un ejemplo que ayude a entender mejor determinados conceptos. Esta forma de interacción permite que el aprendizaje no dependa únicamente de los materiales tradicionales, sino que pueda apoyarse en otros recursos tecnológicos (Wolchover, 2021). Por su parte, Zhang et al. (2020) resaltan la importancia de incorporar herramientas digitales en los espacios educativos, considerando las oportunidades que ofrecen para promover una mayor participación de los estudiantes y atender sus inquietudes mientras desarrollan las actividades académicas.

Aun así, también es importante tener presentes las limitaciones de este tipo de herramientas. Aunque ChatGPT puede generar respuestas a partir de la información utilizada durante su entrenamiento, eso no significa que todo lo que produzca sea correcto. Aunque estas herramientas pueden proporcionar respuestas claras y bien organizadas, no siempre ofrecen información precisa o completa. En determinados casos, sus explicaciones pueden contener errores, omitir aspectos importantes o presentar interpretaciones que no consideran todos los puntos de vista. Por ello, cuando se utilizan en actividades de investigación, es necesario examinar cuidadosamente su contenido y comprobar los datos mediante libros, artículos científicos y otras fuentes académicas confiables. Esta revisión permite evitar que información incorrecta sea incorporada al trabajo y contribuye a mantener el rigor académico de la investigación.

Origen y evolución

El procesamiento de lenguaje natural (PNL) ha experimentado transformaciones revolucionarias desde la introducción de la arquitectura de transformadores. Vaswani et al. (2017) ilustraron que “la arquitectura de transformadores ha redefinido la manera en que las máquinas procesan y generan texto” (p.3). A través del mecanismo de atención, las máquinas han aprendido a priorizar segmentos de textos relevantes, llevando las respuestas generadas a niveles de coherencia sin precedentes.

Dentro de este marco. OpenAI presentó GPT (Generative Pre- trained Transformer).

Mientras GPT-1 dejó una impresión inicial, fue GPT-2 el que realmente capturó la atención de la comunidad de investigación. Radford et al. (2019) observaron que "GPT-2 marcó una transición, demostrando habilidades superiores para comprender y generar textos de mayor longitud" (p.28).

La evolución no culminó con GPT-2, GPT-3.5, al tener una cantidad aún mayor de parámetros, reforzó aún más la capacidad de generar texto coherente y contextual, facilitando conversaciones casi indistinguibles de las interacciones humanas. Brown et al. (2020) comentaron, "GPT-3.5 aborda una serie de limitaciones anteriores, ofreciendo respuestas más contextuales y precisas" (P.47).

ChatGPT 4.0 es el testimonio del continuo compromiso de OpenAI con la excelencia. Las mejoras introducidas en el funcionamiento y entrenamiento de este modelo han permitido que la interacción con los usuarios sea más sencilla y cercana a una conversación habitual. A partir de estos avances, la inteligencia artificial ha encontrado nuevas posibilidades de aplicación en actividades educativas, profesionales y de uso cotidiano. Una de sus características más destacadas es la capacidad de elaborar respuestas claras y coherentes, utilizando expresiones similares a las empleadas en la comunicación humana. Sin embargo, pese a estos progresos, el sistema todavía puede generar información incorrecta o interpretaciones poco precisas, por lo que sus respuestas requieren una revisión cuidadosa antes de ser utilizadas.

El desarrollo de la inteligencia artificial no solo ha traído mejoras en el funcionamiento de sus modelos, sino que también ha generado preocupaciones sobre la manera en que estas tecnologías podrían utilizarse. Un ejemplo de ello fue GPT-2, creado por OpenAI, cuya publicación se llevó a cabo de forma progresiva debido a los posibles usos indebidos que podían darse a su capacidad para producir textos. Entre las principales preocupaciones se encontraba la generación de contenidos falsos o engañosos que pudieran difundirse como información verdadera. Este antecedente muestra que el avance de la inteligencia artificial debe acompañarse de una evaluación de sus posibles consecuencias, considerando tanto los beneficios de su aplicación como los riesgos que representa para los usuarios y la sociedad. Por ello, el desarrollo de estas herramientas requiere prestar atención no solamente a sus capacidades técnicas, sino también a los criterios éticos y de responsabilidad que deben orientar su utilización (Radford et al., 2019).

A lo largo de su desarrollo, los modelos de lenguaje que dieron origen a ChatGPT han

experimentado cambios importantes, desde las primeras aplicaciones de la arquitectura de transformadores hasta versiones más avanzadas, como GPT-4. Estas mejoras han permitido que los sistemas de inteligencia artificial comprendan mejor las consultas y elaboren respuestas cada vez más coherentes y cercanas al lenguaje humano. Sin embargo, este progreso tecnológico también ha dado lugar a interrogantes sobre las consecuencias de su utilización, los límites de sus capacidades y la responsabilidad que implica incorporar estas herramientas en diferentes actividades educativas, profesionales y sociales.

Características principales:

- **Arquitectura de transformadores**

El trabajo desarrollado por Vaswani et al. (2017) marcó un cambio importante en la evolución de los modelos utilizados para procesar el lenguaje natural. Los investigadores propusieron una arquitectura denominada transformador, cuyo funcionamiento se apoya principalmente en mecanismos de atención. Esta característica permite que el modelo identifique relaciones entre palabras y otros elementos de una secuencia, incluso cuando se encuentran separados dentro del texto. Gracias a ello, es posible procesar la información considerando las conexiones entre sus diferentes partes, lo que favorece la interpretación del contexto y facilita el desarrollo de modelos de lenguaje más eficientes. Gracias a ello, estos modelos pueden trabajar mejor con dependencias de largo alcance dentro de los datos y procesar la información con mayor precisión (Vaswani et al., 2017).

Diversos investigadores han señalado que los modelos Transformer se diferencian de arquitecturas previas, como las Redes Neuronales Recurrentes (RNN) y las redes Long Short-Term Memory (LSTM), porque no procesan la información únicamente de manera secuencial. Su funcionamiento se apoya en mecanismos de atención autorregresiva, mediante los cuales el modelo puede identificar qué partes del texto de entrada tienen mayor relevancia al momento de generar una respuesta (Cheng, Dong y Lapata, 2016).

La arquitectura Transformer permitió ampliar las posibilidades de aplicación del procesamiento del lenguaje natural, especialmente en tareas como la traducción entre idiomas y la producción automática de contenido textual. Entre los desarrollos derivados de esta arquitectura se encuentra BERT (Bidirectional Encoder Representations from Transformers), propuesto por Devlin et al. (2018), el cual contribuyó a mejorar el desempeño de distintos sistemas de procesamiento del lenguaje natural. Estos avances evidenciaron la capacidad y flexibilidad de esta arquitectura, que posteriormente sirvió de base para el

desarrollo de modelos más complejos, entre ellos los propuestos por Vaswani y sus colaboradores.

Con respecto a ChatGPT y su adopción de esta arquitectura, su habilidad para generar respuestas coherentes y contextualmente relevantes puede atribuirse en gran medida a la naturaleza intrínseca del transformador. Al centrarse en mecanismos de atención, ChatGPT tiene la capacidad de ponderar la relevancia de cada palabra en el contexto de una conversación y generar respuestas adecuadas en tiempo real (Radford et al., 2019).

- **Modelo preentrenado y fine – tuning**

Entre los enfoques más utilizados actualmente en el procesamiento del lenguaje natural (NLP, siglas en inglés) se encuentran los modelos preentrenados. Radford et al. (2018) explicaron que "el pre entrenamiento implica entrenar modelos en extensos corpus de texto sin etiquetar, permitiendo que los modelos capturen relaciones lingüísticas y semánticas generales" (p.3).

Una vez que un modelo se ha pre entrenado y ha adquirido una comprensión general del lenguaje, es esencial especializar este modelo para tareas concretas. Esta especialización se logra mediante el proceso de fine – tuning. Según Howard y Ruder (2018). "el fine – tuning se realiza sobre conjuntos de datos etiquetados y específicos, permitiendo que le modelo se adapte a contextos particulares y muestre mejoras sustanciales en el rendimiento" (p.7).

Aunque pueden considerarse como dos etapas distintas, el pre entrenamiento y el fine tuning son procesos complementarios. Devlin et al. (2018) destacaron en su trabajo sobre BERT que "la combinación de ambas etapas permite que los modelos alcancen niveles de precisión sin precedentes, aprovechando lo mejor de ambos mundos: la amplia comprensión lingüística adquirida durante el pre entrenamiento y la especialización lograda en el fine – tuning" (p.6).

La eficacia es una de las características inherentes a este enfoque. Aunque el pre entrenamiento puede ser costoso en términos de recursos computacionales, el fine – tunign, al depender de conjuntos de datos más pequeños, es intrínsecamente más ágil. Como Sun et al. (2019) observaron, "mientras que el pre entrenamiento establece una base sólida, el fine – tunig, con su menor demanda de datos, ofrece una adaptabilidad y velocidad esenciales para tareas específicas" (p.15).

Funcionamiento

ChatGPT pertenece a la familia de modelos de lenguaje construidos sobre la arquitectura Transformer. Su funcionamiento se basa en el procesamiento de patrones y relaciones presentes en grandes cantidades de texto, por lo que no interpreta el lenguaje del mismo modo en que lo hace una persona (Vaswani et al., 2017). Su funcionamiento se basa principalmente en reconocer patrones presentes en grandes cantidades de texto procesadas durante el entrenamiento. Gracias a ello, el modelo puede relacionar palabras, estructuras y contextos para generar respuestas que mantienen coherencia y relación con la consulta realizada (Devlin et al., 2018).

Aunque a veces puede parecer que ChatGPT tiene una comprensión semántica o contextual del contenido que se le presenta, lo que realmente está haciendo es predecir la siguiente palabra o frase basándose en la información con la que fue entrenada. La citación directa de Radford et al. (2019) es particularmente esclarecedora al respecto: "el modelo no comprende el texto de manera semántica. En cambio, maximiza la probabilidad de producir respuestas basándose en su entrenamiento" (p.27).

Una característica distintiva de estos modelos es su capacidad para manejar y recordar información de contexto a largo plazo, proporcionando respuestas coherentes a lo largo de una conversación. Sin embargo, no retienen información a largo plazo de las interacciones, garantizando la privacidad del usuario (Brown et al., 2020).

El volumen y la diversidad de datos con los que se entrena ChatGPT son cruciales para su rendimiento. Cuanto mayor sea el conjunto de datos y más variado en términos de contenido y estilo, mejor será el modelo al generalizar y responder a una amplia gama de entradas (Sun et al., 2019).

Aunque ChatGPT tiene la capacidad de elaborar textos claros y mantener conversaciones con un lenguaje natural, es necesario reconocer que sus respuestas no siempre contienen información correcta. En ocasiones, puede presentar explicaciones que parecen coherentes, pero que incluyen datos equivocados o interpretaciones que no consideran todos los aspectos de una situación. Esto ocurre porque su funcionamiento se basa en patrones aprendidos durante el entrenamiento, por lo que generar una respuesta bien estructurada no significa necesariamente que la información sea precisa o que exista una comprensión completa del tema consultado.

Aplicaciones en la educación y la investigación

Aunque ChatGPT tiene la capacidad de elaborar textos claros y mantener conversaciones

con un lenguaje natural, es necesario reconocer que sus respuestas no siempre contienen información correcta. En ocasiones, puede presentar explicaciones que parecen coherentes, pero que incluyen datos equivocados o interpretaciones que no consideran todos los aspectos de una situación. Esto ocurre porque su funcionamiento se basa en patrones aprendidos durante el entrenamiento, por lo que generar una respuesta bien estructurada no significa necesariamente que la información sea precisa o que exista una comprensión completa del tema consultado.

En el ámbito de la enseñanza, ChatGPT puede utilizarse como un recurso de apoyo que permite a los estudiantes ampliar sus conocimientos y aclarar aquellos aspectos que no lograron comprender durante las sesiones académicas. Mediante las consultas realizadas a esta herramienta, tienen la posibilidad de recibir explicaciones adicionales, revisar ejemplos y fortalecer progresivamente su comprensión de los temas desarrollados en clase. Una de sus posibilidades es ofrecer explicaciones según las consultas y dificultades que plantea cada estudiante, permitiéndole revisar los contenidos a su propio ritmo y recibir orientación durante el desarrollo de sus actividades académicas. Este tipo de tutoría apoyada por inteligencia artificial puede ofrecer respuestas inmediatas, explicar contenidos con mayor detalle y ayudar a resolver dudas específicas durante el aprendizaje. De esta manera, también puede favorecer que el estudiante tenga una participación más activa y desarrolle mayor autonomía durante su proceso de formación.

Desde el ámbito de la investigación, esta tecnología ha sido utilizada como recurso de apoyo para el análisis cualitativo de datos, debido a la posibilidad de examinar grandes volúmenes de información textual con rapidez y un nivel adecuado de precisión (Radford et al., 2019). Asimismo, su facultad para producir contenido textual ha sido explorada en estudios preliminares como apoyo para formular hipótesis y plantear nuevas líneas de investigación.

A pesar de las ventajas que puede ofrecer ChatGPT, también existen algunas preocupaciones sobre su uso dentro de la educación. Una de ellas es que los estudiantes puedan llegar a depender demasiado de la herramienta o utilizarla de manera inadecuada para desarrollar trabajos académicos, considerando que puede generar respuestas bastante completas a partir de grandes cantidades de información (Zhang et al., 2020).

Sin embargo, estas limitaciones no reducen las posibilidades que ofrece ChatGPT como recurso complementario para fortalecer la práctica educativa y favorecer el aprendizaje de los estudiantes. Conforme estas herramientas continúan desarrollándose, será importante que

docentes, investigadores, especialistas en ética y desarrolladores trabajen de manera conjunta para orientar su uso de forma responsable. De esta manera, se podrán aprovechar mejor sus beneficios dentro de la educación y, al mismo tiempo, reducir los posibles riesgos que puedan presentarse (Wang et al., 2021).

Limitaciones y precauciones

ChatGPT es una herramienta desarrollada por OpenAI que utiliza procesamiento del lenguaje natural y que ha alcanzado un nivel importante de desarrollo. Aun así, su funcionamiento no está libre de limitaciones y estas deben considerarse al momento de utilizarla. La propia OpenAI reconoce que, si bien estos modelos pueden ofrecer respuestas elaboradas y de utilidad, en algunos casos también pueden generar información equivocada, imprecisa o que no corresponde completamente con la realidad. Además, no comprenden la información de la misma manera que una persona, por lo que sus respuestas deben ser revisadas antes de ser utilizadas como una fuente confiable (OpenAI, 2020).

Una limitación inherente a ChatGPT y otros modelos similares es que no pueden razonar de la misma manera que los humanos. Mientras que pueden generar respuestas basadas en un extenso conjunto de datos, no tienen la capacidad de discernir o comprender de manera genuina (Bender et al., 2021). Esta carencia puede llevar a respuestas incorrectas o engañosas, lo que implica un riesgo particularmente en campos donde la precisión es crítica.

También existe preocupación por los sesgos que pueden encontrarse en la información utilizada para entrenar a ChatGPT. Los modelos de lenguaje pueden reproducir, e incluso reforzar, algunos prejuicios o estereotipos que ya existen en la información con la que fueron desarrollados (Bolukbasi et al., 2016). Por esta razón, las respuestas generadas por estas herramientas deben revisarse con criterio y no asumirse automáticamente como información completamente objetiva o imparcial.

El uso de herramientas de inteligencia artificial como ChatGPT también plantea preocupaciones relacionadas con la ética y el manejo de la información. Entre los riesgos que deben considerarse se encuentran la elaboración de contenidos engañosos, el plagio académico y la difusión de mensajes que podrían utilizarse para influir de manera inapropiada en la opinión de las personas. Al respecto, Zellers et al. (2019) advirtieron sobre las posibilidades de utilizar modelos de generación de texto para producir información falsa con apariencia de autenticidad. Aunque ChatGPT puede redactar respuestas que parecen claras y bien fundamentadas, esto no significa que todos los datos proporcionados sean

correctos. Por esta razón, es necesario revisar cuidadosamente su contenido, contrastar las afirmaciones con fuentes confiables y promover un uso responsable de estas tecnologías, especialmente dentro del ámbito educativo.

El uso frecuente de herramientas de inteligencia artificial en las actividades académicas también puede generar cierta dependencia entre los estudiantes, especialmente cuando recurren a ellas para obtener respuestas sin intentar resolver previamente las dificultades por sus propios medios. Aunque ChatGPT permite consultar información y recibir explicaciones de manera rápida, su utilización constante y sin un análisis previo podría reducir las oportunidades de investigar, comparar distintas fuentes y elaborar conclusiones personales. Esta situación resulta importante en la formación universitaria, donde no basta con encontrar una respuesta, sino que también es necesario comprender cómo se llega a ella y evaluar si la información es correcta. Por ello, conviene que los estudiantes utilicen estas herramientas con criterio, aprovechando sus posibilidades para reforzar los conocimientos adquiridos, pero manteniendo una participación activa en la búsqueda de información, el razonamiento y la solución de problemas. De esta manera, ChatGPT puede complementar las actividades de aprendizaje sin reemplazar el esfuerzo personal ni la capacidad de pensar y tomar decisiones de forma independiente.

2.2.1.1. Efectividad pedagógica de ChatGPT

En la formación universitaria, el uso de ChatGPT puede representar una alternativa para complementar la enseñanza de asignaturas que requieren el manejo de herramientas informáticas, como ocurre con la programación en Python. A través de esta tecnología, los estudiantes tienen la oportunidad de consultar sus dudas mientras desarrollan ejercicios, solicitar explicaciones sobre instrucciones que no comprenden y revisar ejemplos que les permitan reconocer cómo funciona un determinado procedimiento. Esta posibilidad resulta útil cuando se presentan dificultades que necesitan mayor práctica o una explicación diferente a la recibida durante las clases. Así, ChatGPT puede servir como un recurso adicional para fortalecer los conocimientos básicos de programación y facilitar que los estudiantes adquieran progresivamente las habilidades necesarias para resolver problemas de mayor dificultad, de acuerdo con su ritmo y nivel de comprensión.

Para conocer el aporte de ChatGPT en el aprendizaje de los estudiantes, es importante examinar cómo responde a las consultas y qué tan útiles resultan sus explicaciones durante el desarrollo de las actividades académicas. No basta con que la herramienta proporcione información de manera rápida, también es necesario considerar si

las respuestas son comprensibles, si guardan relación con las dificultades planteadas y si ayudan a resolver los ejercicios sin generar una dependencia constante. Otro aspecto que debe tomarse en cuenta es la posibilidad de que cada estudiante utilice este recurso de acuerdo con sus conocimientos y avance a su propio ritmo. Al respecto, Kukulska-Hulme y Traxler (2021) destacan las oportunidades que ofrecen las tecnologías digitales para desarrollar experiencias educativas más flexibles y adaptadas a las necesidades de aprendizaje. Desde esta perspectiva, el uso de ChatGPT puede contribuir a reforzar la comprensión de los contenidos de programación con Python, siempre que los estudiantes analicen las respuestas recibidas y procuren aplicar los conocimientos mediante su propio razonamiento.

Calidad y Pertinencia de las Respuestas

En los entornos virtuales de aprendizaje, ChatGPT puede facilitar la comunicación y el intercambio de información al permitir que los estudiantes realicen consultas y reciban explicaciones relacionadas con los temas desarrollados durante las clases. Esta posibilidad resulta útil en asignaturas como programación con Python, donde comprender determinadas instrucciones o identificar errores en el código puede requerir orientación adicional. Al respecto, Smith y Johnson (2020) resaltan la importancia de la calidad de las respuestas proporcionadas por las herramientas tecnológicas, especialmente cuando se utilizan para explicar contenidos que presentan mayor dificultad. En ese sentido, no solo interesa que ChatGPT responda rápidamente, sino que la información ofrecida sea clara, pertinente y suficientemente precisa para ayudar a los estudiantes a comprender los conceptos y aplicarlos en sus ejercicios de programación.

Adaptabilidad al Aprendizaje del Estudiante

La capacidad de ChatGPT para ajustarse al nivel de cada estudiante puede favorecer una atención diferenciada, aspecto que suele ser más difícil de alcanzar en entornos de aprendizaje convencionales. El uso de inteligencia artificial puede promover una mayor participación e involucramiento de los estudiantes en su proceso formativo, debido a que la experiencia de aprendizaje puede adaptarse con mayor precisión a sus requerimientos individuales.

Fomento del Aprendizaje Autónomo

ChatGPT puede contribuir al desarrollo del aprendizaje autónomo al favorecer una participación más independiente de los estudiantes en la construcción de sus conocimientos. García y Espinoza (2021) señalan que herramientas de este tipo pueden fortalecer el

pensamiento crítico y la capacidad para afrontar y solucionar problemas, debido a que impulsan a los estudiantes a examinar, cuestionar y valorar críticamente la información a la que acceden, habilidades especialmente relevantes en el aprendizaje de la programación.

2.2.1.2. Integración tecnológica de ChatGPT en el proceso educativo

La incorporación de ChatGPT en la educación puede generar cambios importantes en la forma en que se desarrolla el aprendizaje, sobre todo en áreas técnicas como la programación digital. Esta herramienta permite una interacción más directa con el estudiante y puede ajustarse, en cierta medida, a sus diferentes formas de aprender y al nivel de conocimientos que posee.

Una de las principales ventajas de ChatGPT en la educación es la posibilidad de brindar una experiencia de aprendizaje más ajustada a cada estudiante y que pueda aplicarse a un mayor número de estudiantes. Estas características resultan importantes en programas educativos que buscan incorporar nuevas tecnologías de manera útil dentro del proceso de enseñanza (Johnson, Estrada y Freeman, 2014).

Además, ChatGPT puede adaptarse a diferentes escenarios de aprendizaje, ya sea en clases presenciales o en entornos virtuales. Esto permite que el estudiante tenga una interacción más directa con los contenidos del curso y pueda resolver dudas o reforzar temas en el momento en que lo necesita.

Mejora de Interacción Educativa mediante Tecnología AI

La implementación de ChatGPT en la educación puede mejorar significativamente la interacción entre estudiantes y el material de estudio. Según Mayer (2019), las herramientas de inteligencia artificial que simulan conversaciones realistas ofrecen una oportunidad para que los estudiantes exploren concepto a su propio ritmo, lo que puede llevar a un mayor compromiso y retención del conocimiento. La capacidad de ChatGPT para responder a preguntas específicas y proporcionar feedback instantáneo es especialmente valiosa en campos técnicos como la programación digital.

2.2.2. Aprendizaje de programación digital con Python

Aprender a programar en Python no consiste solamente en memorizar instrucciones o conocer las reglas que se utilizan para elaborar un código. También es necesario que los estudiantes desarrollen habilidades para comprender una situación problemática, identificar la información que necesitan y establecer los procedimientos adecuados para encontrar una solución. Estas capacidades adquieren especial importancia en la formación de ingeniería,

donde los conocimientos de programación pueden aplicarse al desarrollo de cálculos, procesamiento de datos y resolución de problemas técnicos. En relación con ello, Guo (2014) destaca la utilidad de contar con herramientas que permitan comprender el funcionamiento de los programas y analizar los procedimientos empleados, aspectos que pueden ayudar a relacionar los fundamentos teóricos con las actividades prácticas de programación. Como parte de este aprendizaje, ChatGPT puede emplearse como un recurso de apoyo para que los estudiantes aclaren sus inquietudes y revisen los procedimientos utilizados durante las prácticas de programación. A través de esta herramienta, tienen la oportunidad de consultar ejemplos, analizar las instrucciones de un programa y reconocer posibles errores que dificultan su ejecución. Su utilización también puede ayudarles a comprender las razones por las que un código no funciona correctamente y explorar diferentes alternativas para solucionarlo. De esta manera, se busca que los estudiantes fortalezcan sus conocimientos de Python y desarrollen habilidades para aplicarlos en ejercicios y situaciones técnicas relacionadas con la Ingeniería de Minas.

Desafíos en el aprendizaje de Python

Si bien Python cuenta con una sintaxis relativamente sencilla, aprender a utilizarlo puede representar un desafío para los estudiantes que no tienen experiencia previa en programación. Al comenzar, es frecuente que surjan dificultades para comprender el funcionamiento de las instrucciones, reconocer la lógica de los algoritmos y relacionar los conceptos estudiados con su aplicación en ejercicios concretos. Por ello, familiarizarse con este lenguaje requiere algo más que aprender a escribir códigos, pues también implica desarrollar progresivamente la capacidad de analizar problemas, organizar procedimientos y encontrar soluciones mediante las herramientas disponibles. En la carrera de Ingeniería de Minas, el aprendizaje de programación adquiere una mayor complejidad cuando los estudiantes deben utilizar Python para resolver situaciones relacionadas con las actividades mineras. Por ejemplo, el procesamiento de información geológica, los cálculos operativos y el análisis de datos requieren comprender diferentes variables y establecer procedimientos que permitan obtener resultados adecuados. Frente a estas exigencias, resulta necesario que los estudiantes desarrollen habilidades para interpretar la información, organizar sus ideas y aplicar los conocimientos de programación en ejercicios vinculados con su especialidad. De esta manera, el dominio de Python puede convertirse en un recurso útil para afrontar problemas técnicos que podrían presentarse durante su formación y posterior desempeño profesional.

Evolución de la Educación en la Programación

Con el paso de los años, la manera de enseñar programación ha ido cambiando, principalmente por la aparición de nuevas tecnologías y la necesidad de utilizar métodos que faciliten la comprensión de los estudiantes. Estos cambios han permitido incorporar herramientas digitales que complementan las explicaciones del docente y ofrecen otras formas de desarrollar las actividades académicas. Entre ellas se encuentra ChatGPT, cuyo uso en la educación ha despertado interés por las posibilidades que ofrece para resolver consultas, revisar ejemplos de programación y acompañar a los estudiantes mientras adquieren nuevos conocimientos.

El uso de estas herramientas puede servir como un apoyo adicional durante las actividades académicas, ya que brinda a los estudiantes la oportunidad de revisar los contenidos y resolver sus dudas según las dificultades que encuentren. De esta manera, cada uno puede avanzar a su propio ritmo, dedicar más tiempo a los temas que le resulten complicados y reforzar aquellos conocimientos que necesita comprender mejor.

2.2.2.1. Fundamentos de Python

Para comenzar a programar en Python, es necesario familiarizarse con algunos conceptos básicos que permiten comprender cómo se construye y ejecuta un programa. Una de las particularidades de este lenguaje es que utiliza instrucciones relativamente sencillas de leer, lo que facilita su aprendizaje, especialmente entre quienes tienen poca experiencia en programación. Un aspecto importante es el uso de la indentación o sangría, que no solamente permite mantener el código ordenado, sino que también determina la organización de los bloques de instrucciones que forman parte del programa (Downey, 2015). Gracias a esta forma de estructurar el código, resulta más sencillo identificar las operaciones que se realizan, seguir la lógica de las instrucciones y reconocer posibles errores durante el desarrollo de los ejercicios.

Dentro del aprendizaje de Python, también es necesario conocer los tipos de datos que se utilizan para almacenar y procesar información. Este lenguaje permite trabajar con números enteros, valores decimales, textos y estructuras como listas, tuplas y diccionarios, cada una con características que responden a diferentes necesidades de programación. Comprender estas diferencias ayuda al estudiante a seleccionar la estructura más apropiada para organizar los datos, realizar operaciones y desarrollar ejercicios de manera ordenada. Al respecto, Zelle (2004) explica la importancia de familiarizarse con los tipos de datos y las operaciones que pueden efectuarse con ellos, pues constituyen una base para elaborar

programas que funcionen correctamente. Estos conocimientos también permiten manejar la información que ingresa al programa, procesar los resultados y resolver problemas que requieren cálculos o procedimientos de mayor dificultad.

En el aprendizaje de Python, las estructuras de control cumplen una función importante porque permiten organizar las instrucciones y establecer cómo debe ejecutarse un programa frente a determinadas condiciones. Entre las más utilizadas se encuentran los bucles y las estructuras condicionales, que ayudan a desarrollar procedimientos con una secuencia lógica y a resolver ejercicios que requieren diferentes operaciones (Lutz, 2013). Los bucles `for` y `while` se emplean cuando es necesario repetir una o varias instrucciones, ya sea recorriendo una secuencia de elementos o mientras se mantenga una condición establecida. Por su parte, las instrucciones `if`, `elif` y `else` permiten evaluar distintas situaciones y ejecutar las acciones que correspondan según el resultado obtenido. Comprender cómo funcionan estas estructuras facilita la elaboración de algoritmos, ayuda a evitar instrucciones repetitivas y permite que los estudiantes desarrollen soluciones más ordenadas ante los problemas de programación.

Sintaxis y Estructuras del código

Python se caracteriza por utilizar una sintaxis sencilla y fácil de interpretar, lo que permite que los estudiantes se familiaricen progresivamente con las instrucciones de programación. Una de sus particularidades es el uso obligatorio de la indentación o sangría, que permite identificar los bloques de código y establecer cómo se organizan las instrucciones dentro de un programa. Al respecto, Downey (2015) destaca la importancia de comprender las reglas sintácticas del lenguaje, ya que su manejo adecuado ayuda a prevenir errores durante la escritura del código y facilita el desarrollo de programas claros, ordenados y funcionales.

Tipos de Datos

Al trabajar con Python, es necesario conocer las diferentes formas en que se puede almacenar y organizar la información. Entre los tipos de datos más utilizados se encuentran los números enteros, los decimales, las cadenas de texto, las listas, las tuplas y los diccionarios. Cada uno cumple una función particular dentro del programa y permite realizar determinadas operaciones según las necesidades del ejercicio. En ese sentido, Zelle (2004) resalta la importancia de comprender las características de estos tipos de datos, ya que su correcta utilización ayuda a procesar la información de manera ordenada, realizar cálculos y construir programas que respondan adecuadamente a las instrucciones establecidas.

Estructuras de control

Los bucles y las estructuras condicionales permiten organizar la secuencia de ejecución del código y decidir qué acciones debe realizar el programa en cada caso. Lutz (2013) sugiere que dominar estas estructuras es fundamental para cualquier programador que desee crear aplicaciones complejas.

Funciones

Las funciones de Python permiten la modularización del código, lo que facilita su mantenimiento y reutilización. Crear funciones claras y bien definidas es una habilidad crucial para la programación efectiva. Según McKinney (2017), las funciones también ayudan a mejorar la legibilidad y la organización del código.

2.2.2.2. Python aplicado a la programación digital

Python es ampliamente reconocido por su versatilidad y potencia, Estas características hacen que Python pueda utilizarse en distintas aplicaciones de la programación digital. Una de las áreas donde tiene mayor presencia es el análisis de datos, debido a que permite trabajar con grandes cantidades de información y realizar análisis de diferente complejidad. Esto resulta especialmente útil en campos como la ingeniería de minas, donde es necesario interpretar de manera adecuada datos geológicos para apoyar la toma de decisiones. Según McKinney (2017), las bibliotecas de Python como Pandas y NumPy proporciona herramientas útiles para la manipulación y el análisis de datos, permitiendo a los ingenieros identificar información relevante y fundamentar decisiones mediante el procesamiento de datos confiables.

Además del análisis de datos, Python también se utiliza para la creación de soluciones web y la ejecución automatizada de procesos. Herramientas como Django y Flask permiten desarrollar aplicaciones web de manera más organizada y escalable. Django es un framework de Python que facilita la creación de aplicaciones web mediante una estructura de trabajo organizada y el aprovechamiento de componentes que pueden reutilizarse en diferentes partes del proyecto. Una de sus características es la aplicación del principio DRY (*Don't Repeat Yourself*), que busca evitar la duplicación innecesaria de código y mantener una mejor organización durante el desarrollo del software (Holovaty y Kaplan, 2009). Gracias a estas características, los programadores pueden dedicar mayor atención al funcionamiento de la aplicación y a las necesidades que debe resolver, mientras que Django proporciona herramientas para realizar tareas habituales del desarrollo web, reduciendo el trabajo repetitivo y facilitando el mantenimiento del código.

Otra de las ventajas de Python es que puede utilizarse junto con diferentes herramientas informáticas para desarrollar actividades de automatización. Este lenguaje permite trabajar con archivos, ejecutar instrucciones en el sistema operativo y realizar tareas que, de forma manual, requerirían mayor tiempo y esfuerzo. Estas posibilidades resultan útiles en el campo de la ingeniería, donde constantemente se necesita procesar información y ejecutar procedimientos que se repiten durante el desarrollo de un proyecto. Al respecto, Taneja (2018) señala que la facilidad de uso de Python y la disponibilidad de diversas bibliotecas permiten simplificar procesos que presentan cierta complejidad. En Ingeniería de Minas, estas características pueden aprovecharse para organizar datos operativos, automatizar cálculos y agilizar actividades de análisis, contribuyendo a un mejor aprovechamiento del tiempo y de los recursos disponibles.

Análisis de Datos

Python también tiene aplicaciones importantes en el análisis y procesamiento de información, especialmente cuando se trabaja con grandes cantidades de datos que necesitan organizarse e interpretarse. Para estas actividades, dispone de bibliotecas como Pandas y NumPy, que permiten realizar cálculos estadísticos, ordenar registros y transformar la información de acuerdo con las necesidades del análisis. McKinney (2017) explica que estas herramientas facilitan el tratamiento de datos, desde la identificación de valores incorrectos o incompletos hasta su organización para realizar diferentes operaciones. En Ingeniería de Minas, estas posibilidades resultan útiles porque permiten trabajar con información geológica, ambiental y operativa, cuyos resultados sirven de apoyo para evaluar las condiciones de un proyecto y tomar decisiones técnicas sustentadas en los datos disponibles.

Desarrollo de Aplicaciones Web

Python también se utiliza en el desarrollo de aplicaciones web, para lo cual dispone de herramientas como Django y Flask, que facilitan la organización y ejecución de diferentes tareas de programación. En el caso de Django, su estructura permite trabajar con proyectos que requieren varias funciones y componentes, manteniendo una distribución ordenada del código. Al respecto, Holovaty y Kaplan (2009) explican que este framework incorpora el principio DRY (*Don't Repeat Yourself*), orientado a evitar que las mismas instrucciones se escriban repetidamente en distintas partes del programa. Esta forma de trabajo ayuda a aprovechar mejor los componentes ya desarrollados, facilita la revisión del código y reduce la posibilidad de cometer errores durante la construcción y el mantenimiento de las aplicaciones.

En Ingeniería de Minas, las aplicaciones web creadas mediante estas herramientas pueden resultar útiles para organizar las actividades de un proyecto, consultar y procesar datos operativos actualizados y compartir información entre las diferentes áreas de trabajo. Su utilización facilita el seguimiento de las operaciones y permite que los profesionales involucrados dispongan de información para coordinar sus actividades y apoyar la toma de decisiones.

Automatización de Tareas

En Ingeniería de Minas, Python puede utilizarse para automatizar actividades que requieren la ejecución repetida de cálculos, el procesamiento de datos y la organización de información operativa. Su aplicación permite reducir el tiempo dedicado a tareas rutinarias, disminuir los errores asociados al trabajo manual y facilitar el desarrollo de diferentes procedimientos técnicos dentro de las operaciones mineras. Python permite la automatización de procesos repetitivos y la integración con sistemas operativos para realizar tareas complejas de manera automática. Taneja (2018) resalta que la simplicidad de Python, junto a su amplia gama de bibliotecas, facilita la creación de scripts para automatiza tareas como la generación de informes, el monitoreo de equipos y la gestión de datos, liberando tiempo valioso para los ingenieros y permitiéndoles enfocarse en tareas más estratégicas.

2.3 Bases conceptuales

ChatGPT

Modelos de lenguaje basados en inteligencia artificial diseñados para facilitar interacciones en tiempo real con usuarios mediante lenguaje natural. ChatGPT se utiliza en educación para proporcionar asistencia instantánea y personalizada, mejorando la manera en que los estudiantes desarrollan su aprendizaje.

Programación Digital

La programación digital consiste en utilizar lenguajes de programación y distintas herramientas tecnológicas para crear aplicaciones, páginas web y otras soluciones digitales. Dentro de este campo, Python es uno de los lenguajes más utilizados, principalmente por su facilidad de aprendizaje, su sintaxis sencilla y las posibilidades que ofrece para desarrollar diferentes tipos de proyectos.

Python

Un lenguaje interpretado y de alto nivel que destaca por su sintaxis clara y su capacidad para adaptarse a diferentes aplicaciones. Python es un lenguaje apropiado para quienes comienzan

a aprender programación, ya que sus instrucciones son relativamente sencillas de leer y comprender. Además, dispone de una amplia variedad de módulos y paquetes que facilitan el desarrollo de ejercicios prácticos y permiten aplicar progresivamente los conocimientos adquiridos.

Fundamentos de Python

Entre los conocimientos iniciales que se deben adquirir al aprender Python se encuentran las reglas de escritura del código, los diferentes tipos de datos, las estructuras de control y el uso de funciones. Comprender estos elementos permite al estudiante organizar mejor las instrucciones, desarrollar soluciones a problemas de programación y elaborar programas que funcionen correctamente y aprovechen de manera adecuada los recursos disponibles.

Sintaxis de Python

Python utiliza una forma de escritura sencilla y ordenada, que permite comprender las instrucciones del programa sin necesidad de emplear estructuras demasiado complejas. Esta característica facilita que los estudiantes se familiaricen con el lenguaje y desarrollen progresivamente sus habilidades de programación. La indentación es una característica distintiva que define la estructura del código.

Tipo de Datos en Python

Python maneja varios tipos de datos, incluidos enteros, flotantes, cadenas, listas, tuplas y diccionarios. La correcta utilización de estos tipos permite a los programadores gestionar y manipular información de manera eficiente.

Estructuras de Control en Python

Los bucles y las estructuras condicionales son importantes porque ayudan a controlar cómo avanza un programa según las condiciones que se presentan durante su ejecución.

Funciones en Python

Las funciones permiten la modularización del código, facilitando su mantenimiento y reutilización. Crear funciones claras y bien definidas mejora la legibilidad y organización del código.

Análisis de Datos con Python

En tareas relacionadas con el procesamiento de información, Python dispone de bibliotecas como Pandas y NumPy, las cuales permiten estructurar, transformar y examinar volúmenes amplios de datos de manera práctica y eficiente.

Desarrollo de Aplicaciones Web con Python

Herramientas como Django y Flask facilitan el desarrollo de aplicaciones web sólidas y preparadas para crecer según las necesidades del proyecto, Django sigue el principio de "no te repitas" (DRY), lo que ayuda a mantener el código limpio y eficiente.

Automatización de Tareas con Python

Python puede automatizar tareas repetitivas e integrarse con sistemas operativos para realizar los procesos complejos automáticamente, mejorando la eficiencia operativa en proyectos de ingeniería.

Bibliotecas de Python

Conjunto de módulos y paquetes que extienden las capacidades del lenguaje, facilitando tareas específicas como análisis de datos, desarrollo web, y automatización. Entre las más destacadas se encuentran Pandas, NumPy, y Matplotlib.

Educación Asistida por Inteligencia Artificial

Uso de tecnologías basadas en inteligencia artificial, como ChatGPT, para proporcionar tutoría y asistencia personalizada en tiempo real, mejorando la manera en que se enseña y se aprende.

Ingeniería de Minas

Rama de la ingeniería orientada a la explotación y aprovechamiento de los recursos minerales bajo criterios de seguridad, eficiencia técnica y control operativo. En este campo, Python puede emplearse para modelar y simular procesos mineros, así como para analizar información relacionada con las operaciones.

Aplicación de Python en Ingeniería de Minas

Uso de Python para desarrollar soluciones tecnológicas en la ingeniería de minas, como la simulación de procesos de extracción y manejo e interpretación de datos geológicos. De esta manera, se pueden optimizar las actividades mineras y fortalecer las condiciones de seguridad.

3. METODOLOGÍA

3.1. Formulación de hipótesis

3.1.1. Hipótesis general

El uso de ChatGPT influye significativamente el aprendizaje de Programación Digital con Python en los estudiantes de Ingeniería de Minas de la Universidad

Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II.

3.1.2. Hipótesis específicas

a. El uso de ChatGPT influye significativamente en el aprendizaje de fundamentos de Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II.

b. El uso de ChatGPT influye significativamente en el aprendizaje de Python aplicado a Programación Digital en los estudiantes de Ingeniería de Minas en la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II.

3.2. Variables

3.2.1. Variable independiente

X: ChatGPT

ChatGPT es una herramienta de inteligencia artificial diseñada para interactuar en lenguaje natural, utilizada aquí para medir su influencia en la educación técnica.

Dimensiones:

X1: Efectividad pedagógica de ChatGPT

Esta dimensión evalúa la capacidad de ChatGPT para mejorar la comprensión conceptual y el rendimiento académico de los estudiantes en el contexto del aprendizaje de programación.

X2: Integración tecnológica de ChatGPT

Se centra en la efectividad con la que ChatGPT se integra en el entorno educativo existente, examinando su compatibilidad y sinergia con las herramientas y plataformas educativas ya en uso.

3.2.2. Variable dependiente

Y: Aprendizaje de programación digital con Python

El aprendizaje de programación en Python se relaciona con los conocimientos y habilidades que los estudiantes desarrollan para comprender, elaborar y ejecutar programas mediante este lenguaje. Su evaluación considera el desempeño alcanzado durante las actividades académicas y la capacidad para aplicar lo aprendido en la resolución de problemas, tanto en situaciones prácticas reales como en ejercicios de simulación.

Dimensiones:

Y1: Fundamentos de Python

Esta dimensión permite evaluar el nivel de conocimiento que tienen los estudiantes sobre los fundamentos de Python, considerando aspectos como las reglas de escritura del código, el manejo de los tipos de datos y el funcionamiento de las estructuras de control. El manejo adecuado de estos conocimientos permite que los estudiantes comprendan cómo se relacionan las diferentes instrucciones de un programa, establezcan una secuencia lógica de operaciones y construyan algoritmos capaces de resolver los ejercicios propuestos. Asimismo, facilita la elaboración de códigos ordenados y funcionales, favoreciendo el desarrollo progresivo de sus habilidades de programación.

Y2: Python aplicado la programación digital

Esta dimensión evalúa la capacidad de los estudiantes para aplicar lo aprendido en Python al desarrollar ejercicios y proyectos que presentan un mayor grado de complejidad. Para ello, se toma en cuenta la manera en que interpretan los problemas, establecen los pasos necesarios para resolverlos y utilizan las diferentes herramientas del lenguaje de programación. También se considera su habilidad para construir algoritmos, ejecutar instrucciones y comprobar los resultados obtenidos. De esta forma, se busca identificar el nivel de dominio práctico alcanzado por los estudiantes y su capacidad para emplear los conocimientos adquiridos en situaciones que requieren razonamiento, análisis y elaboración de soluciones mediante la programación.

3.3. Operacionalización de variables

La operacionalización de variables es una etapa importante del proceso metodológico, ya que permite precisar de qué manera se estudiarán los aspectos considerados en la investigación. Mediante este procedimiento, los conceptos generales se descomponen en dimensiones e indicadores que facilitan su observación y medición. Esta organización ayuda a definir qué información debe recogerse, qué características serán evaluadas y cómo se relacionarán los datos obtenidos con los objetivos planteados. Así, el investigador dispone de criterios concretos para elaborar los instrumentos y desarrollar posteriormente el análisis de los resultados. En relación con este procedimiento, Sampieri, Collado y Lucio (2018) señalan que es necesario precisar los elementos que serán objeto de evaluación y establecer cómo se obtendrá la información requerida para el estudio. Esta definición permite organizar de manera adecuada el proceso de recolección de datos y seleccionar los instrumentos que mejor respondan a las características de las variables investigadas. Asimismo, ayuda a que exista correspondencia entre los aspectos que se

pretenden medir, los procedimientos empleados y la interpretación de los resultados, manteniendo así la coherencia metodológica de la investigación.

En esta investigación, orientada a evaluar el uso de ChatGPT en el aprendizaje de programación con Python en estudiantes de ingeniería, se estableció la manera en que serían medidas las variables independiente y dependiente. Para ello, se identificaron las dimensiones e indicadores que permiten examinar tanto la incorporación de ChatGPT en las actividades académicas como los conocimientos y habilidades desarrollados por los estudiantes durante el proceso de aprendizaje. Esta organización facilita la recolección de información mediante los instrumentos seleccionados y permite analizar los resultados de acuerdo con los objetivos planteados. Asimismo, contar con indicadores definidos ayuda a mantener criterios uniformes durante la evaluación y favorece que los procedimientos utilizados puedan ser revisados o aplicados en investigaciones similares.

Tabla 1

Operacionalización de variables

Variable	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Escala de medición
Variable Independiente: ChatGPT	Herramienta de Inteligencia Artificial que genera texto basado en instrucciones dadas, aplicada al aprendizaje digital.	Grado en el que ChatGPT contribuye al aprendizaje de programación de Python a través de su integración y capacidades pedagógicas.	Efectividad pedagógica de ChatGPT	• Calidad de las respuestas y retroalimentación proporcionada por ChatGPT • Accesibilidad y disponibilidad de ChatGPT • Capacidad de ChatGPT para adaptarse a diferentes niveles de conocimiento y estilos de aprendizaje	No aplica
			Integración Tecnológica de ChatGPT en el proceso educativo	• Facilidad de implementación y uso de ChatGPT en el entorno educativo • Integración de ChatGPT con otras plataformas y herramientas educativas • Eficiencia de ChatGPT en la presentación de contenidos educativos	

Variable Dependiente: Aprendizaje de Programación con Python	Capacidad de los estudiantes para comprender y aplicar conocimientos de programación en Python	Medido por el progreso y dominio de conceptos y técnicas específicas de Python	Fundamentos de Python	• Conocimiento de los tipos de datos en Python • Uso correcto de operadores en Python • Comprensión y uso de estructuras de control	Escala ordinal • Deficiente • Regular • Bueno • Excelente
			Python aplicado a la programación digital	• Conexión y manipulación de base de datos con Python • Uso de Django para la creación de aplicaciones web • Implementación de clases y migración en Django	

Fuente: Elaboración propia.

Tabla 2

Matriz instrumental

Variable	Dimensión	Indicador	Técnicas	Instrumentos	Nº Ítems	Escala	Valoración
Variable Independiente: ChatGPT	Efectividad pedagógica de ChatGPT	• Calidad de las respuestas y retroalimentación • Accesibilidad y disponibilidad de ChatGPT • Capacidad de ChatGPT para adaptarse a diferentes niveles de conocimiento y estilos de aprendizaje	No aplica	No aplica	No aplica	No aplica	No aplica
	Integración Tecnológica de ChatGPT en el proceso educativo	• Facilidad de implementación y uso de ChatGPT en el entorno educativo • Integración de ChatGPT con otras plataformas y herramientas educativas • Eficiencia de ChatGPT en la presentación de contenidos educativos					
Variable Dependiente: Aprendizaje de Programación Digital con Python	Fundamentos de Python	• Conocimiento de los tipos de datos en Python • Uso correcto de operadores en Python • Comprensión y uso de estructuras de control	Observación	Ficha de Observación	10	Ordinal	1. Deficiente 2. Regular 3. Bueno 4. Excelente
	Python aplicado a la programación digital	• Conexión y manipulación de base de datos con Python • Uso de Django para la creación de aplicaciones web • Implementación de clases y migración en Django	Observación	Ficha de Observación	10	Ordinal	1. Deficiente 2. Regular 3. Bueno 4. Excelente

Fuente: Elaboración propia.

3.4. Tipo y nivel de investigación

3.4.1. Tipo de Investigación

La investigación aplicada busca utilizar los conocimientos científicos y técnicos para atender problemas específicos que se presentan en determinados contextos, planteando alternativas de solución que puedan llevarse a la práctica. De acuerdo con Hernández, Fernández y Baptista (2014), este enfoque emplea conocimientos previamente desarrollados con la finalidad de responder a necesidades específicas que se presentan en un determinado contexto. A diferencia de la investigación básica, centrada principalmente en generar y ampliar conocimiento teórico, la investigación aplicada procura emplearlo para intervenir sobre problemas particulares de la realidad.

En el presente estudio se adopta este tipo de investigación debido a que se incorpora ChatGPT como recurso tecnológico para favorecer el aprendizaje de Programación Digital con Python en estudiantes de la carrera de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga. Aunque esta tecnología ya forma parte de distintos espacios educativos, en esta investigación se analiza su aplicación como apoyo durante el proceso de aprendizaje de los estudiantes.

De este modo, el estudio va más allá de una revisión únicamente teórica del problema, sino que busca emplear este conocimiento dentro de una situación educativa concreta. En ese sentido, la investigación intenta responder a una necesidad práctica y evaluar una alternativa que pueda contribuir a mejorar el aprendizaje, lo cual guarda relación con las características propias correspondiente a una investigación de tipo aplicado.

3.4.2. Nivel de investigación

La investigación explicativa se caracteriza por su enfoque en descubrir las relaciones causales entre elementos de un fenómeno estudiado. En palabras de Hernández y Mendoza (2018), “el propósito de las investigaciones explicativas es precisar las causas y los efectos de los fenómenos que se indagan”. Este tipo de investigación se adentra más allá de la descripción de los fenómenos, buscando comprender las razones subyacentes de su ocurrencia.

Asimismo, la investigación se desarrolló bajo un enfoque experimental, considerando un diseño cuasiexperimental. Según Creswell (2014), este tipo de estudios permite examinar los efectos que produce una intervención sobre una variable de interés, procurando controlar aquellos factores que podrían influir en los resultados. En el presente

trabajo, se incorporó ChatGPT como herramienta de apoyo en la enseñanza de programación con Python, con el propósito de evaluar sus efectos en el aprendizaje de los estudiantes mediante la comparación de los resultados obtenidos por el grupo experimental y el grupo de control.

La presente investigación doctoral se desarrolló considerando un nivel explicativo y un diseño cuasiexperimental, con el propósito de conocer los efectos que puede tener el uso de ChatGPT en el aprendizaje de programación con Python. Para ello, se consideró la incorporación de esta herramienta de inteligencia artificial como variable independiente, mientras que el aprendizaje alcanzado por los estudiantes de Ingeniería de Minas constituyó la variable dependiente. A partir de la comparación de los resultados obtenidos antes y después de la intervención, tanto en el grupo experimental como en el grupo de control, se buscó identificar los cambios producidos en los conocimientos y habilidades de programación. De esta manera, el estudio no se limitó a establecer una relación entre ambas variables, sino que procuró evaluar en qué medida las diferencias observadas podían atribuirse al uso de ChatGPT, considerando las características y limitaciones propias del diseño cuasiexperimental.

3.5. Métodos

El método hipotético-deductivo se utiliza en la investigación científica para examinar posibles explicaciones sobre un problema y comprobar si estas encuentran respaldo en la información recopilada. Este procedimiento comienza con el planteamiento de hipótesis, a partir de las cuales se establecen determinadas consecuencias que posteriormente deben contrastarse mediante la observación y el análisis de los datos. En relación con ello, Popper (2002) explica que las teorías científicas tienen un carácter provisional, pues están sujetas a cuestionamientos y deben someterse a pruebas que permitan identificar posibles errores o contradicciones. Desde este planteamiento, el conocimiento científico avanza mediante la revisión constante de las explicaciones propuestas y su confrontación con la evidencia obtenida durante la investigación. En este sentido, las hipótesis representan posibles explicaciones de un problema de investigación y no afirmaciones que deban aceptarse como ciertas desde el inicio. Su validez debe examinarse mediante procedimientos que permitan contrastarlas con los datos recopilados. A partir de este análisis, el investigador puede reconocer si la evidencia respalda sus planteamientos iniciales o si existen razones para cuestionarlos. De esta manera, el método hipotético-deductivo favorece una evaluación

crítica de los resultados y permite formular conclusiones fundamentadas en la información obtenida durante el estudio.

El análisis estadístico constituye una parte importante del proceso de investigación, pues permite trabajar de manera ordenada con la información recogida y examinar los resultados mediante procedimientos cuantitativos. Su utilidad se hace más evidente cuando se necesita comparar datos, reconocer diferencias entre grupos o establecer relaciones entre las variables estudiadas. Al respecto, Fisher (1971) resalta la importancia de considerar distintos elementos de información al analizar situaciones complejas, debido a que una conclusión no siempre puede establecerse a partir de un solo resultado. En ese sentido, las herramientas estadísticas ayudan al investigador a interpretar los datos con mayor precisión y a evaluar las hipótesis formuladas. Su aplicación también permite identificar tendencias y diferencias que podrían pasar desapercibidas en una revisión general, proporcionando fundamentos para elaborar conclusiones basadas en la evidencia y no únicamente en apreciaciones personales.

Para desarrollar el estudio se empleó el método hipotético-deductivo, acompañado de procedimientos estadísticos que permitieron examinar los resultados obtenidos durante la investigación. A partir del problema identificado, se plantearon hipótesis orientadas a conocer los posibles efectos de ChatGPT en el aprendizaje de programación con Python. Estas hipótesis consideraron las variaciones que podrían producirse en los conocimientos y habilidades de los estudiantes después de utilizar la herramienta como apoyo en las sesiones académicas. De esta manera, se estableció una base metodológica para contrastar los planteamientos iniciales con la información recogida antes y después de la intervención educativa. Posteriormente, la información obtenida de las evaluaciones realizadas al grupo experimental y al grupo de control fue procesada mediante pruebas estadísticas, lo que permitió identificar las diferencias existentes entre ambos grupos y examinar los resultados alcanzados antes y después de la intervención. A partir de este procedimiento, se contó con evidencia para determinar si correspondía rechazar o no las hipótesis nulas planteadas, procurando que las conclusiones respondieran a los datos recogidos durante el estudio.

3.6. Diseño de la investigación

La investigación se desarrolla bajo un diseño cuasiexperimental, adecuado para contextos en los que no resulta factible distribuir aleatoriamente a los participantes entre los grupos de estudio. De acuerdo con Hernández y Mendoza (2018), este tipo de diseño “se caracteriza porque el investigador no tiene control total sobre la variable independiente, ya

que no puede manipularla ni asignar aleatoriamente a los sujetos a los grupos experimental y control". Este enfoque es de especial relevancia en el ámbito educativo, dado que las condiciones preexistentes y la organización de los estudiantes a menudo limitan la flexibilidad en la formación de grupos.

En la estructura del estudio se establecerán dos grupos de comparación: uno de control y otro experimental. Siguiendo la metodología planteada por Arias (2012), en ambos grupos se aplicarán evaluaciones antes y después de la intervención. El grupo experimental incorporará ChatGPT como recurso de apoyo para el aprendizaje de Programación Digital con Python, mientras que el grupo de control continuará con una metodología convencional de enseñanza. Esta organización permitirá comparar de manera directa los resultados obtenidos bajo ambas formas de trabajo.

La aplicación de este enfoque cuasiexperimental permitirá analizar las diferencias entre la metodología convencional y la estrategia educativa apoyada en ChatGPT dentro del aprendizaje de programación. Si bien la ausencia de asignación aleatoria puede generar algunas variaciones entre los grupos, este tipo de diseño permite examinar cambios y relaciones vinculadas con una intervención educativa (Hernández y Mendoza, 2018). Aunque presenta ciertas restricciones metodológicas, resulta pertinente para el propósito del estudio; sin embargo, sus resultados deberán interpretarse con cautela al trasladarlos a otros contextos.

Tabla 3

Esquema que se utilizará en la investigación cuasi experimental

Grupo de investigación	Pre test	Tratamiento experimental	Post test
G.C.	Q1	...	Q3
G.E.	Q2	x	Q4

Q1, Q2 : Aplicación de pre test a grupo experimental y grupo control

Q3, Q4 : Aplicación del post test a grupo experimental y grupo control

Fuente: Arias, (2012).

3.7. Población y muestra

3.7.1. Población

En una investigación, la población corresponde a la totalidad de unidades que reúnen las características establecidas para el estudio y sobre las cuales se pretende obtener información que permita formular conclusiones. De acuerdo con Hernández, Fernández y

Baptista (2014), la población comprende a los sujetos u objetos que cumplen determinados criterios definidos previamente por el investigador. Su adecuada delimitación permite establecer con claridad el alcance del estudio y las unidades que serán consideradas durante el proceso de investigación.

En el presente estudio, la población estará integrada por los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga matriculados en el curso de Programación Digital durante el semestre académico 2024-II. Esta población resulta pertinente porque permite examinar la influencia de la incorporación de ChatGPT sobre el aprendizaje de Programación Digital con Python.

Para el desarrollo del estudio, los participantes serán organizados en dos grupos de comparación: uno experimental y otro de control. El grupo experimental desarrollará las sesiones utilizando ChatGPT como recurso de apoyo educativo, en tanto que el grupo de control mantendrá la metodología convencional de enseñanza.

Este diseño permite una comparación directa y efectiva del aprendizaje entre los dos grupos, facilitado así la evaluación del impacto real de ChatGPT en la adquisición de habilidades programáticas. La definición clara y precisa de la población en este estudio es crucial, ya que toda la población participará directamente en la investigación, lo que elimina la necesidad de técnicas de muestreo y potencia la validez interna del diseño cuasi experimental empleado.

Tabla 4

Población

Lugar	Escuela profesional	Nivel	Grupo	Sexo		Nº de estudiantes	Total
				F	M		
Universidad Nacional de San Cristóbal de Huamanga	Ingeniería de Minas	Universitario	Único	7	33	40	**40**

Fuente: Elaboración Propia.

3.7.2. Muestra

Seleccionar adecuadamente una muestra es crucial en cualquier estudio científico, ya que está selección permite realizar inferencias válidas sobre una población más amplia a partir de un subconjunto representativo. Hernández, Fernández y Baptista (2014) subrayan que “la muestra es, en esencia, un subgrupo de la población del cual se recogen datos y debe

ser representativo".

En el marco de esta investigación, orientada a examinar la influencia de la incorporación de ChatGPT sobre el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas, se ha optado por trabajar con la totalidad de los alumnos matriculados en el curso correspondiente durante el semestre académico 2024-II. En consecuencia, no resulta necesario aplicar un procedimiento de muestreo, ya que se considera directamente a todos los estudiantes inscritos, lo que permite reducir posibles distorsiones asociadas con la selección de participantes.

Aunque no se aplicará una técnica de muestreo convencional, debido a que se trabajará con la totalidad de la población disponible, será necesario definir con claridad la conformación de los grupos experimental y de control para evaluar adecuadamente el efecto de la intervención. Los participantes serán asignados al azar entre dos grupos de estudio, procurando mantener una composición equilibrada que permita realizar una comparación adecuada entre el uso de ChatGPT y la metodología convencional de enseñanza.

Este procedimiento contribuirá a fortalecer la validez interna del estudio, ya que permitirá asociar los cambios observados en el aprendizaje principalmente con la intervención realizada mediante ChatGPT, reduciendo la influencia de posibles diferencias iniciales entre los participantes.

Tabla 5

Muestra

Lugar	Escuela profesional	Nivel	Grupo	Sexo		Nº de estudiantes	Total
				F	M		
Universidad Nacional de San Cristóbal de Huamanga	Ingeniería de Minas	Universitario	Único	7	33	40	**40**

Fuente: Elaboración Propia.

3.7.3. Tipo de muestreo

Para esta investigación se aplicó un procedimiento de selección probabilística de tipo aleatorio simple para conformar el grupo de control y el grupo experimental, cada uno integrado por 20 estudiantes del curso de Programación Digital pertenecientes a la Escuela Profesional de Ingeniería de Minas. La elección de este procedimiento respondió a la

necesidad de mantener condiciones comparables entre ambos grupos y fortalecer la validez interna del diseño experimental.

De acuerdo con Babbie (2016), este procedimiento otorga a cada integrante de la población una probabilidad equivalente de ser elegido, lo que contribuye a reducir posibles sesgos durante la selección y favorece una distribución más equitativa de los participantes. favoreciendo la equidad en la asignación. En ese sentido, la homogeneidad en el número y características de los grupos permitió obtener conclusiones más consistentes respecto a la eficacia de la intervención implementada.

3.8. Técnicas e instrumentos

3.8.1. Técnicas

La observación directa permite recoger información sobre los hechos y comportamientos que se presentan en el entorno donde se desarrolla una investigación. Para su aplicación, es necesario establecer previamente qué aspectos serán observados y cómo se registrarán, procurando mantener criterios uniformes durante todo el proceso. Kerlinger y Lee (2000) destacan la importancia de realizar observaciones de manera sistemática, de modo que los datos obtenidos puedan utilizarse posteriormente en el análisis del problema estudiado. En el ámbito educativo, esta técnica resulta útil porque permite conocer de cerca cómo participan los estudiantes, qué dificultades encuentran durante las actividades y de qué manera se desenvuelven en espacios de aprendizaje como las aulas y los laboratorios de informática.

En el desarrollo de esta investigación, la observación directa se consideró como una técnica para conocer cómo los estudiantes de Ingeniería de Minas interactuaban con ChatGPT durante el aprendizaje de programación con Python. A través de este procedimiento, se buscó identificar la manera en que utilizaban la herramienta para resolver ejercicios, consultar sus dudas y poner en práctica los conocimientos adquiridos. Para recoger esta información, se establecieron criterios de observación que permitieron registrar las actividades de forma organizada y mantener las mismas condiciones de evaluación para los participantes. De esta manera, se procuró obtener información objetiva sobre el desenvolvimiento de los estudiantes durante las sesiones de aprendizaje.

3.8.2. Instrumentos

En el campo de la educación, la ficha de observación es un instrumento que permite recoger información sobre el comportamiento y la participación de los estudiantes mientras

desarrollan sus actividades académicas. Su utilización facilita el registro de hechos y situaciones que ocurren durante las clases, siguiendo criterios establecidos previamente por el investigador. Al respecto, Díaz (2021) señala que este instrumento ayuda a organizar por escrito los datos obtenidos mediante la observación, de manera que puedan revisarse y analizarse posteriormente. Así, la información registrada sirve como apoyo para evaluar el desenvolvimiento de los estudiantes y reconocer aquellos aspectos que requieren mayor atención durante el proceso de aprendizaje.

En esta investigación, las fichas de observación estructuradas se utilizaron para registrar cómo los estudiantes de Ingeniería de Minas empleaban ChatGPT durante las actividades de programación con Python. Mediante este instrumento, se buscó conocer la forma en que realizaban sus consultas, resolvían los ejercicios propuestos y aplicaban los conocimientos adquiridos durante las sesiones de aprendizaje. Para ello, se consideraron aspectos previamente definidos que facilitaron el registro ordenado de la información y su posterior análisis. Si bien la observación presenta algunas limitaciones, como la posible influencia del investigador al interpretar determinadas conductas, su aplicación permitió recoger información complementaria sobre la participación de los estudiantes y su interacción con la herramienta, contribuyendo a una mejor comprensión del proceso educativo desarrollado.

3.9. Validez y confiabilidad de instrumentos

3.9.1. Validez de instrumentos

La validez de los instrumentos utilizados en una investigación permite determinar si realmente evalúan los aspectos para los cuales fueron elaborados. En el ámbito educativo, este criterio adquiere especial importancia cuando se estudia el uso de herramientas tecnológicas como ChatGPT en el aprendizaje de programación con Python. Por ello, es necesario comprobar que los instrumentos empleados permitan recoger información relacionada con los conocimientos y habilidades que se pretende evaluar, de manera que los resultados obtenidos puedan interpretarse adecuadamente y sirvan como sustento para las conclusiones del estudio.

La revisión del instrumento se realizó mediante el juicio de expertos, con la finalidad de comprobar si las preguntas planteadas guardaban relación con los aspectos que se buscaba evaluar. Al respecto, Salkind (2010) explica que la validez de un instrumento depende de su capacidad para medir aquello que realmente se pretende estudiar. Tomando en cuenta este criterio, los especialistas evaluaron la pertinencia, claridad y coherencia de los ítems,

considerando su correspondencia con la variable y las dimensiones establecidas en la investigación. Esta revisión permitió identificar aspectos que requerían ajustes y mejorar la organización del instrumento, procurando que las preguntas fueran apropiadas para recoger información sobre el aprendizaje de Python en los estudiantes que participaron en la intervención con ChatGPT.

La ficha de observación fue sometida a la evaluación de especialistas con experiencia en el área de estudio, quienes revisaron su contenido considerando los criterios definidos para su validación. A partir de esta revisión, se obtuvieron las calificaciones y apreciaciones de cada evaluador. En la siguiente tabla se presentan los puntajes otorgados por los expertos, el promedio alcanzado y las observaciones formuladas durante este proceso.

Tabla 6

Sistematización de valores asignados por expertos

N°	Indicadores	Experto 1(%)	Experto 2(%)	Experto 3(%)	Experto 4(%)	Experto 5(%)	Promedio (%)	Comentarios de Expertos
1.	Claridad	85	87	88	84	86	86	Mayoritariamente clasificado como muy buena
2.	Objetividad	72	78	77	74	74	75	Variabilidad en la percepción de objetividad
3.	Actualidad	80	84	85	80	81	82	Considerado actual y relevante para el contexto del curso
4.	Organización	82	85	86	83	84	84	Bien organizado y lógico según la mayoría
5.	Suficiencia	78	79	82	80	81	80	Suficiente para capturar los aspectos necesarios del desempeño
6.	Intencionalidad	76	79	80	77	78	78	Alienado con los objetivos del estudio
7.	Consistencia	83	87	86	85	84	85	Consistente en términos teóricos y prácticos
8.	Coherencia	87	90	89	87	88	88	Alta coherencia entre los ítems y los indicadores
9.	Metodología	79	82	83	80	81	81	Metodología bien estructurada y apropiada

10.	Pertinencia	88	91	92	89	90	90	Alta pertinencia con respecto contenido del curso

Fuente: Elaboración Propia.

La evaluación realizada por los jueces expertos alcanzó un promedio de 83,9 %, valor que corresponde a la categoría de "muy buena" según la escala utilizada en la investigación. Esta calificación muestra que los especialistas consideraron adecuada la ficha de observación para recoger información sobre las competencias de los estudiantes en programación digital con Python. A partir de sus valoraciones, se obtuvo un respaldo favorable respecto a la pertinencia y correspondencia de los ítems con los aspectos que se buscaba evaluar. De esta manera, el juicio de expertos aportó evidencia sobre la validez de contenido del instrumento, proporcionando una base para su aplicación durante la recolección de datos. No obstante, esta valoración debe complementarse con el análisis de confiabilidad para determinar la consistencia de las mediciones obtenidas.

3.9.2. Confiabilidad de instrumentos

La confiabilidad de un instrumento de investigación se relaciona con la consistencia de los resultados que proporciona al evaluar determinados aspectos bajo condiciones semejantes. En ese sentido, Tavakol y Dennick (2011) explican que este criterio es importante para valorar la calidad de las mediciones y determinar si los ítems que conforman una escala mantienen una relación adecuada entre sí. En el caso del aprendizaje de programación con Python, resulta necesario disponer de instrumentos que permitan evaluar los conocimientos y habilidades de los estudiantes siguiendo criterios uniformes. De esta manera, el análisis de confiabilidad ayuda a conocer qué tan consistentes son las mediciones realizadas y proporciona mayor sustento para interpretar los resultados obtenidos durante la investigación.

La confiabilidad del instrumento se evaluó mediante el coeficiente alfa de Cronbach, una medida estadística que permite conocer el grado de consistencia interna entre los ítems que lo conforman y determinar si presentan una relación adecuada al evaluar los aspectos establecidos en la investigación. Field (2013) indica que valores más elevados de este coeficiente reflejan una mayor relación interna entre los ítems evaluados. En esta investigación, antes de la aplicación definitiva de la ficha de observación, se realizará una aplicación preliminar del instrumento. Posteriormente, se calculará el coeficiente alfa de Cronbach con la finalidad de determinar su nivel de consistencia interna y fortalecer la

rigurosidad metodológica del estudio.

Tabla 7

Alfa de Cronbach por cada elemento

N° ítems	Media de escala si elemento se ha suprimido	Varianza de escala si elemento se ha suprimido	Correlación total de elementos corregida	Alfa de Cronbach si el elemento se ha suprimido
Ítem 1	26,00	49,474	-,219	,878
Ítem 2	26,00	44,842	,548	,859
Ítem 3	25,60	39,200	,704	,849
Ítem 4	26,00	42,526	,644	,854
Ítem 5	25,85	47,608	,025	,877
Ítem 6	25,95	43,945	,532	,859
Ítem 7	26,00	44,526	,473	,861
Ítem 8	25,85	42,134	,572	,856
Ítem 9	25,90	45,989	,312	,865
Ítem 10	25,90	45,042	,312	,866
Ítem 11	25,85	44,239	,466	,861
Ítem 12	26,00	43,263	,655	,855
Ítem 13	25,75	39,671	,763	,847
Ítem 14	26,00	46,632	,182	,869
Ítem 15	25,75	37,987	,870	,840
Ítem 16	26,05	46,050	,374	,864
Ítem 17	25,95	42,050	,683	,852
Ítem 18	25,85	44,766	,338	,865
Ítem 19	25,60	41,726	,625	,854
Ítem 20	25,90	46,305	,169	,871

Fuente: Elaboración Propia.

El análisis de consistencia interna mediante el coeficiente alfa de Cronbach obtuvo un valor de 0.867, resultado que se encuentra por encima del criterio de referencia de 0.70 utilizado habitualmente para interpretar este indicador. Cabe señalar que el número de ítems evaluados correspondió a la estructura original del instrumento, sin que fuera necesario eliminar o modificar alguno de ellos. En consecuencia, el valor obtenido evidencia una adecuada consistencia interna de la ficha de observación y respalda su utilización en el desarrollo de la investigación.

3.10. Técnicas de procesamiento de datos

El tratamiento de la información constituye una etapa esencial para desarrollar

análisis estadísticos rigurosos y obtener resultados confiables en una investigación. Para este estudio se utilizará el software SPSS, programa ampliamente empleado en investigaciones de carácter social y educativo por las distintas opciones que ofrece para organizar, procesar y analizar datos. Field (2013), SPSS permite aplicar diversas pruebas estadísticas y facilita el examen sistemático de la información recopilada. En la presente investigación, su utilización permitirá procesar los datos obtenidos mediante los instrumentos y realizar los análisis necesarios para contrastar las hipótesis planteadas.

En esta investigación, orientada a evaluar el uso de ChatGPT en el aprendizaje de programación con Python, se utilizó el programa estadístico SPSS para organizar, procesar y analizar los datos recogidos durante las evaluaciones. Esta herramienta permitió realizar los cálculos necesarios y aplicar las pruebas estadísticas correspondientes, facilitando la comparación de los resultados obtenidos por el grupo experimental y el grupo de control. Al respecto, Pallant (2016) explica que SPSS dispone de diferentes procedimientos para examinar la información y reconocer patrones, diferencias o relaciones entre las variables estudiadas. Su utilización permitió desarrollar el análisis de los datos de manera ordenada, considerando los criterios estadísticos establecidos en la investigación. A partir de los resultados obtenidos, fue posible contrastar las hipótesis planteadas y formular conclusiones respaldadas por la evidencia recopilada.

En esta investigación se empleará tanto estadística descriptiva como inferencial. La primera permitirá resumir los datos mediante tablas de frecuencia y otras medidas descriptivas, mientras que la segunda se utilizará para examinar la distribución de los datos y seleccionar las pruebas de contraste más adecuadas según sus características. Antes de contrastar las hipótesis se evaluará el supuesto de normalidad, a fin de determinar si corresponde aplicar procedimientos paramétricos o no paramétricos.

Debido a que los datos no presentaron una distribución normal, se recurrió a pruebas no paramétricas. Para analizar los cambios producidos dentro de cada grupo se aplicó la prueba de rangos con signo de Wilcoxon, adecuada para comparar mediciones relacionadas. Por otro lado, la prueba U de Mann–Whitney se utilizó para contrastar los resultados obtenidos entre el grupo experimental y el grupo de control, al tratarse de muestras independientes. La elección de estas pruebas permitió desarrollar un análisis coherente con las características de los datos y fortalecer la consistencia de las conclusiones estadísticas obtenidas durante el estudio.

3.11. Aspectos éticos

La ética constituye un aspecto importante en el desarrollo de cualquier investigación, especialmente cuando se trabaja con personas que proporcionan información o participan en las actividades del estudio. Por esta razón, el investigador debe actuar con responsabilidad, respetar los derechos de los participantes y mantener una conducta transparente durante todas las etapas del trabajo. En relación con ello, Resnik (2015) señala que las buenas prácticas científicas requieren honestidad en el manejo de la información, respeto hacia las personas involucradas y compromiso con la integridad de los procedimientos utilizados. Estos principios permiten orientar las decisiones del investigador y contribuir a que el estudio se desarrolle bajo criterios de responsabilidad y respeto.

En esta investigación, orientada a evaluar el uso de ChatGPT en el aprendizaje de programación con Python, el consentimiento informado se consideró un requisito importante para garantizar la participación voluntaria de los estudiantes. Antes de su incorporación al estudio, se contempló brindarles información clara sobre los objetivos del trabajo, las actividades previstas y los posibles beneficios o riesgos asociados con su participación. También se estableció la necesidad de que conocieran las condiciones del proceso y pudieran resolver cualquier inquietud antes de tomar una decisión. Con ello, se buscó respetar la libertad de cada participante para decidir si deseaba formar parte de la investigación, sin presiones ni consecuencias académicas por su elección.

Durante la elaboración de una investigación, es necesario reconocer las contribuciones de los autores cuyas ideas, conocimientos y trabajos han servido de sustento para el estudio. Esta práctica permite respetar la propiedad intelectual y mantener la honestidad académica al utilizar información procedente de otras fuentes. Citar correctamente las fuentes consultadas permite identificar el origen de la información y diferenciar los planteamientos propios de aquellos desarrollados en estudios anteriores. En ese sentido, Smith (2018) destaca la importancia de respetar la autoría como parte de una conducta académica responsable y del intercambio de conocimientos entre investigadores. En el presente estudio, se emplearon los criterios establecidos por las normas APA de séptima edición para organizar las citas y referencias bibliográficas. Con ello, se buscó reconocer las contribuciones de los autores consultados, facilitar la revisión de las fuentes utilizadas y mantener la transparencia en la elaboración del trabajo académico.

4. RESULTADOS Y DISCUSIÓN

4.1. Resultados a nivel descriptivo

Tabla 8

Nivel de aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas

Categorías	Control				Experimental			
	Pretest		Postest		Pretest		Postest	
	f	%	f	%	f	%	f	%
Deficiente	20	100	0	0	20	100	0	0
Regular	0	0	12	60	0	0	0	0
Bueno	0	0	8	40	0	0	15	75
Excelente	0	0	0	0	0	0	5	25
Total	20	100	20	100	20	100	20	100

En la tabla 8 se muestran los niveles de aprendizaje alcanzados por los estudiantes de Ingeniería de Minas en programación digital con Python, a partir de los resultados obtenidos antes y después de la intervención. La información permite observar cómo se distribuyeron los estudiantes según su nivel de desempeño en ambas evaluaciones, tanto en el grupo de control como en el experimental, facilitando la comparación de los cambios registrados durante el desarrollo de la investigación.

En la evaluación inicial del grupo de control, todos los estudiantes (100 %) presentaron un nivel deficiente de aprendizaje en programación con Python. Sin embargo, al revisar los resultados de la evaluación final, se observaron cambios en su desempeño, pues el 60 % se ubicó en el nivel regular y el 40 % alcanzó el nivel bueno. Esto indica que los participantes lograron mejorar sus conocimientos durante el periodo de estudio, aunque ninguno consiguió situarse en el nivel excelente, tanto en la evaluación inicial como en la final.

El comportamiento del grupo experimental fue distinto. En el pretest, el 100 % de los participantes también se encontraba en el nivel deficiente; sin embargo, una vez culminada la intervención, los resultados mostraron una variación importante. El 75 % alcanzó el nivel bueno y el 25 % se ubicó en el nivel excelente. Además, ningún participante permaneció en los niveles deficiente o regular.

En conjunto, estos resultados muestran una evolución favorable en el aprendizaje de Programación Digital con Python dentro del grupo que utilizó ChatGPT como apoyo. Los resultados muestran que los estudiantes lograron avances en la comprensión de los conceptos básicos de Python y en el manejo de sus principales herramientas de programación. Esta

mejora también se reflejó en la manera de resolver ejercicios prácticos, utilizar las instrucciones del lenguaje y desarrollar actividades que requerían conocimientos de mayor complejidad. De esta forma, el progreso alcanzado no se limitó al aprendizaje teórico, sino que también comprendió el fortalecimiento de habilidades para aplicar lo aprendido en diferentes tareas de programación digital. Además, se observó una mayor precisión en el uso de las herramientas de programación, lo cual refuerza la efectividad de ChatGPT como herramienta educativa.

Tabla 9

Nivel de de aprendizaje de Fundamentos de Python en estudiantes de Ingeniería de Minas

Categorías	Control				Experimental			
	Pretest		Postest		Pretest		Postest	
	f	%	f	%	f	%	f	%
Deficiente	20	100	0	0	20	100	0	0
Regular	0	0	12	60	0	0	0	0
Bueno	0	0	8	40	0	0	11	55
Excelente	0	0	0	0	0	0	9	45
Total	20	100	20	100	20	100	20	100

La tabla 9 presenta la distribución de los niveles alcanzados en Fundamentos de Python por los estudiantes de Ingeniería de Minas, considerando tanto al grupo de control como al grupo experimental en la evaluación inicial y final.

Respecto al grupo de control, al inicio de la evaluación el 100 % de los estudiantes se ubicó en la categoría deficiente. Al finalizar la evaluación, se observaron cambios en el desempeño de los estudiantes, de los cuales el 60 % se ubicó en el nivel regular, mientras que el 40 % logró alcanzar el nivel bueno. En ninguna de las dos mediciones se identificaron participantes en la categoría excelente.

Por su parte, el grupo experimental mostró una situación inicial semejante, ya que la totalidad de los estudiantes se concentró en la categoría deficiente durante el pretest. Sin embargo, una vez finalizada la intervención, los resultados variaron de manera importante: el 55 % alcanzó el nivel bueno y el 45 % llegó al nivel excelente. Además, ningún participante permaneció en los niveles deficiente o regular.

Estos resultados muestran una mejora importante en el aprendizaje de los Fundamentos de Python. Los estudiantes avanzaron en el manejo de la sintaxis del lenguaje, los tipos de datos, los operadores matemáticos y el uso de funciones y estructuras como print, input, if, elif, else, for y while. También lograron desarrollar una mejor capacidad para

organizar el código por partes y corregir errores de sintaxis. En general, se observó una mayor precisión al momento de utilizar estos elementos dentro de sus programas, lo que refleja una mejor aplicación de los fundamentos de Python en actividades de programación digital.

Tabla 10

Nivel de aprendizaje de Python aplicado a la programación digital en estudiantes de Ingeniería de Minas

Categorías	Control				Experimental			
	Pretest		Postest		Pretest		Postest	
	f	%	f	%	f	%	f	%
Deficiente	19	95	0	0	18	90	0	0
Regular	1	5	8	40	2	10	0	0
Bueno	0	0	12	60	0	0	13	65
Excelente	0	0	0	0	0	0	7	35
Total	20	100	20	100	20	100	20	100

La tabla 10 expone la distribución de los niveles alcanzados en el aprendizaje de Programación Digital con Python por los grupos de control y experimental, a partir de las evaluaciones inicial y final.

En la medición inicial del grupo de control, el 95 % de los estudiantes se ubicó en la categoría deficiente y el 5 % en el nivel regular. Ningún participante alcanzó los niveles bueno o excelente. Los resultados de la evaluación final mostraron cambios en los niveles de aprendizaje de los estudiantes. A diferencia de la medición inicial, ninguno se mantuvo en la categoría deficiente. El 40 % se ubicó en el nivel regular y el 60 % logró alcanzar el nivel bueno, lo que refleja una mejora en su desempeño académico. Sin embargo, a pesar de los avances registrados, ningún participante obtuvo una calificación correspondiente al nivel excelente.

Al inicio de la investigación, el grupo experimental presentó un bajo nivel de aprendizaje en programación con Python, debido a que el 90 % de los estudiantes se ubicó en la categoría deficiente y solo el 10 % alcanzó el nivel regular. Sin embargo, después de incorporar ChatGPT como herramienta de apoyo durante las actividades académicas, se observaron mejoras importantes en las calificaciones. En la evaluación final, ninguno de los participantes se encontró en los niveles deficiente o regular, mientras que el 65 % logró ubicarse en el nivel bueno y el 35 % alcanzó el nivel excelente. Estos resultados reflejan un avance considerable en el desempeño de los estudiantes después de la intervención educativa.

Los resultados obtenidos muestran una mejora en las habilidades de los estudiantes para utilizar Python en el desarrollo de actividades de programación digital. Este progreso se observó principalmente en la creación de programas secuenciales capaces de recibir y procesar datos ingresados por el usuario, así como en el manejo de instrucciones condicionales para resolver diferentes situaciones. Del mismo modo, los participantes demostraron un mayor conocimiento en el uso de listas y tuplas, empleándolas para almacenar, organizar y trabajar con información durante la elaboración de sus programas. También mejoraron en el trabajo con archivos, principalmente en la lectura y escritura de datos en formatos txt y csv.

De igual manera, se evidenciaron progresos en la utilización de módulos y bibliotecas de Python, así como en la aplicación de conceptos de programación orientada a objetos y en el uso de estructuras de mayor complejidad, como matrices y listas anidadas. Todo ello permitió que los estudiantes resolvieran problemas con mayor nivel de dificultad vinculados con la Ingeniería de Minas. Además, se notó una mayor precisión al momento de desarrollar e implementar los códigos.

4.2. Resultados a nivel inferencial

4.2.1. Prueba de normalidad

Tabla 11

Prueba de normalidad

Prueba de normalidad			
	Shapiro-Wilk		
	Estadístico	gl	Sig.
Aprendizaje de Programación Digital con Python	0.738	20	0
Fundamentos de Python	0.664	20	0
Python Aplicado a la Programación Digital	0.788	20	0

Nota. Datos procesados mediante SPSS en el curso de Programación Digital de los estudiantes de Ingeniería de Minas, UNSCH, 2024.

Se empleó la prueba de Shapiro–Wilk para determinar si los datos asociados con la variable dependiente Aprendizaje de Programación Digital con Python y sus dos dimensiones: Fundamentos de Python y Python Aplicado a la Programación Digital. Los resultados indicaron que en los tres casos se obtuvo un valor de significancia (Sig.) igual a 0.000, con estadísticos de 0,738 para la variable general, 0,664 para Fundamentos de Python

y 0,788 para Python Aplicado a la Programación Digital, en una muestra de 20 estudiantes por dimensión.

Dado que todos los valores de significancia fueron inferiores al umbral de α = 0.05, se determinó que los datos no cumplían con el supuesto de normalidad. Por esta razón, y conforme a los estándares metodlógicos para el tratamiento de variables no paramétricas, se procedió con la aplicación de las pruebas estadísticas U de Mann – Whitney y Wilcoxon para el análisis de hipótesis, asegurando la robustez y validez de los resultados inferenciales.

4.2.2. Prueba de hipótesis

4.2.2.1. Prueba de hipótesis general

Hipótesis Nula

H_0 : El uso de ChatGPT no influye significativamente en el aprendizaje de Programación Digital con Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024 – II.

Hipótesis Alterna

H_1 : El uso de ChatGPT si influye significativamente en el aprendizaje de Programación Digital con Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024 – II.

Para la presente investigación, se estableció un nivel de significancia del 5% (α = 0.05), lo cual corresponde a un nivel de confianza del 95%, parámetros comúnmente aceptados en estudios de naturaleza cuantitativa.

Tabla 12

Prueba de U Mann – Whitney control y experimental y Wilcoxon en el grupo de aprendizaje de Programación Digital con Python.

Estadístico	Significancia asintótica (bilateral) $\rho - valor\ (calculado\ en\ SPSS)$	Significancia asumida
U Mann – Whitney (Grupos pretest)	$\rho = 1.000(100\%)$	$\alpha = 0.05\ (5\%)$
U Mann - Whitney	$\rho = 0.000(0\%)$	$\alpha = 0.05\ (5\%)$
Wilcoxon	$\rho = 0.000(0\%)$	$\alpha = 0.05\ (5\%)$

Al comparar las puntuaciones obtenidas por ambos grupos antes de la intervención, la prueba U de Mann-Whitney arrojó un valor de significancia de p = 1,000, superior al nivel establecido de 0,05. Esto significa que, durante la evaluación inicial, no se encontraron diferencias estadísticamente significativas entre los estudiantes del grupo de control y los del grupo experimental. Aunque este resultado no demuestra que ambos grupos fueran completamente equivalentes, sí permite señalar que no se detectaron diferencias significativas en sus niveles de aprendizaje al comenzar la investigación. Esta condición proporcionó un punto de partida para comparar los resultados posteriores y examinar los cambios registrados después de incorporar ChatGPT como herramienta de apoyo en la enseñanza de Python.

Los resultados del postest mostraron una diferencia respecto a lo observado al inicio de la investigación. Al aplicar la prueba U de Mann-Whitney, se obtuvo un valor de significancia de p < 0,001, inferior al nivel de 0,05 considerado para el análisis estadístico. Este resultado permitió identificar diferencias estadísticamente significativas entre las puntuaciones alcanzadas por ambos grupos después de la intervención. En particular, los estudiantes que utilizaron ChatGPT como herramienta complementaria durante el aprendizaje de Python obtuvieron un mejor desempeño que aquellos que continuaron con la enseñanza tradicional, lo que aporta evidencia favorable sobre los efectos de su incorporación en las actividades académicas.

Por otra parte, al comparar los resultados iniciales y finales del grupo experimental mediante la prueba de Wilcoxon, se obtuvo un nivel de significancia de p < 0,001. Este valor permitió comprobar que las diferencias entre las puntuaciones del pretest y el postest fueron estadísticamente significativas. Los estudiantes que participaron en las actividades de aprendizaje apoyadas con ChatGPT alcanzaron mejores calificaciones después de la intervención, lo que refleja un progreso favorable en sus conocimientos y habilidades de programación con Python.

Los resultados obtenidos mediante las pruebas estadísticas proporcionaron evidencia suficiente para rechazar la hipótesis nula, respaldando así lo planteado en la hipótesis alternativa de la investigación. Estos resultados permiten sostener, dentro de las condiciones y alcance del estudio, que el uso de ChatGPT estuvo asociado con cambios significativos en el aprendizaje de Programación Digital con Python en los estudiantes de Ingeniería de Minas de la UNSCH asimilaron los contenidos de Programación Digital con Python en el periodo 2024-II.

4.2.2.2. Prueba de hipótesis específica

4.2.2.2.1. Prueba de hipótesis específica 1

Hipótesis Nula

H_0 : El uso de ChatGPT no influye significativamente en el aprendizaje de fundamentos de python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024 – II.

Hipótesis Alterna

H_1 : El uso de ChatGPT si influye significativamente en el aprendizaje de fundamentos de python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024 – II.

La presente contrastación estadística fue desarrollada bajo un umbral de error del 5% (α = 0.05), lo cual equivale a un 95% de nivel de confianza, de acuerdo con los estándares de rigurosidad metodológica aplicables a estudios experimentales de enfoque cuantitativo.

Tabla 1

Prueba de U Mann – Whitney control y experimental y Wilcoxon en la dimensión Fundamentos Python.

Estadístico	Significancia asintótica (bilateral) $\rho - valor\ (calculado\ en\ SPSS)$	Significancia asumida
U Mann – Whitney (Grupos pretest)	$\rho = 1.000(100\%)$	$\alpha = 0.05\ (5\%)$
U Mann - Whitney	$\rho = 0.000(0\%)$	$\alpha = 0.05\ (5\%)$
Wilcoxon	$\rho = 0.000(0\%)$	$\alpha = 0.05\ (5\%)$

Durante la evaluación inicial, se compararon las puntuaciones de los estudiantes del grupo de control y del grupo experimental mediante la prueba U de Mann-Whitney, obteniéndose un valor de significancia de p = 1,000, superior al nivel de 0,05 establecido para el estudio. A partir de este resultado, no se encontraron diferencias estadísticamente significativas en el aprendizaje de programación con Python entre ambos grupos antes de incorporar ChatGPT como herramienta de apoyo. Si bien este valor no demuestra que los participantes tuvieran exactamente el mismo nivel de conocimientos, permite señalar que no se identificaron diferencias significativas en su desempeño inicial. Esta situación facilitó la comparación posterior de las puntuaciones, con la finalidad de examinar los cambios registrados después de la intervención educativa.

Al analizar las puntuaciones obtenidas en el postest, se encontraron diferencias entre el desempeño de los estudiantes del grupo de control y del grupo experimental. La prueba U de Mann-Whitney arrojó un nivel de significancia de $p < 0,001$, inferior al valor de referencia de 0,05. Esto permitió establecer que las diferencias registradas entre ambos grupos fueron estadísticamente significativas después de la intervención educativa. En particular, los estudiantes que utilizaron ChatGPT como herramienta complementaria durante las actividades de programación con Python alcanzaron mejores resultados que aquellos que continuaron con la modalidad tradicional de enseñanza. Estos hallazgos aportan evidencia favorable sobre la utilidad de incorporar esta herramienta en el proceso de aprendizaje.

Al comparar las calificaciones del grupo experimental antes y después de la intervención, la prueba de Wilcoxon arrojó un valor de significancia de $p < 0,001$, inferior al nivel de 0,05 establecido en la investigación. Este resultado permitió identificar diferencias estadísticamente significativas entre las puntuaciones del pretest y el postest. Asimismo, se observó que los estudiantes alcanzaron un mejor desempeño en la evaluación final, lo que refleja un avance en sus conocimientos y habilidades de programación con Python tras incorporar ChatGPT como herramienta de apoyo durante las actividades de aprendizaje.

A partir del análisis estadístico realizado, se rechazó la hipótesis nula y se respaldó la hipótesis alternativa. Estos resultados permiten sostener, dentro de las condiciones y alcance del estudio, que el uso de ChatGPT estuvo asociado con cambios significativos en el aprendizaje de Programación Digital con Python en los estudiantes de Ingeniería de Minas de la UNSCH.en el semestre 2024-II.

4.2.2.2.2. Prueba de hipótesis específica 2

Hipótesis Nula

H_0 : El uso de ChatGPT no influye significativamente en el aprendizaje de python aplicado a la programación digital en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024 – II.

Hipótesis Alterna

H_1 : El uso de ChatGPT si influye significativamente en el aprendizaje de python aplicado a la programación digital en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024 – II.

El presente contraste se realizó bajo un criterio estadístico de α = 0.05 (5%), equivalente a un nivel de confianza del 95%, en conformidad con las buenas prácticas de investigación experiemntal cuantitativa.

Tabla 14

Prueba de U Mann – Whitney control y experimental y Wilcoxon en la dimensión Python Aplicado a la Programación Digital.

Estadístico	Significancia asintótica (bilateral) $\rho - valor\ (calculado\ en\ SPSS)$	Significancia asumida
U Mann – Whitney (Grupos pretest)	$\rho = 0.553 (55.3\%)$	$\alpha = 0.05\ (5\%)$
U Mann - Whitney	$\rho = 0.000 (0\%)$	$\alpha = 0.05\ (5\%)$
Wilcoxon	$\rho = 0.000 (0\%)$	$\alpha = 0.05\ (5\%)$

Nota. Cálculos realizados con el software estadístico SPSS sobre los datos correspondientes a la dimensión Python Aplicado a la Programación Digital en estudiantes de Ingeniería de Minas – UNSCH, 2024.

En la evaluación inicial, el contraste U de Mann-Whitney registró un valor de p = 0,553, superior al criterio de significancia de 0,05. Este resultado indica que no se identificaron diferencias estadísticamente significativas entre el grupo de control y el grupo experimental antes de iniciar la intervención. En consecuencia, ambos grupos presentaban condiciones comparables al comienzo del estudio.

Después de la intervención, la evaluación final mostró un comportamiento diferente. La prueba U de Mann-Whitney registró p = 0,000, valor inferior a 0,05, evidenciando una diferencia estadísticamente significativa entre ambos grupos. El grupo experimental, que utilizó ChatGPT como recurso de apoyo, alcanzó puntuaciones superiores respecto del grupo de control.

Asimismo, se contrastaron los resultados del grupo experimental antes y después de la intervención mediante la prueba de Wilcoxon. El análisis obtuvo p = 0,000, lo que evidencia un cambio estadísticamente significativo en el desempeño de los estudiantes luego de utilizar la herramienta.

A partir de estos resultados, se rechazó la hipótesis nula y se obtuvo respaldo estadístico para la hipótesis alternativa. Dentro de las condiciones evaluadas, la incorporación de ChatGPT estuvo asociada con cambios significativos en el aprendizaje de Python aplicado a la programación digital en los estudiantes de Ingeniería de Minas de la

Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II..

4.3. Discusión de resultados

Los hallazgos del estudio evidenciaron que la incorporación de ChatGPT estuvo asociada con una evolución favorable en el aprendizaje de Programación Digital con Python. Desde el análisis descriptivo, el grupo experimental presentó una variación importante entre las mediciones inicial y final. En el pretest, la totalidad de los participantes se ubicaba en la categoría deficiente; posteriormente, tras la intervención, el 75 % alcanzó el nivel bueno y el 25 % llegó al nivel excelente. Esta variación evidencia un progreso favorable en el desempeño académico de los estudiantes después de trabajar con la herramienta.

Para desarrollar el análisis inferencial, primero se verificó la distribución de los datos mediante la prueba de Shapiro-Wilk, con el propósito de determinar si cumplían con el supuesto de normalidad. Los resultados correspondientes a la variable de estudio y sus respectivas dimensiones presentaron valores de significancia inferiores a 0,05, por lo que se rechazó la hipótesis de normalidad. Ante esta situación, se consideró apropiado utilizar pruebas estadísticas no paramétricas. En particular, se aplicó la U de Mann-Whitney para comparar las puntuaciones del grupo experimental y del grupo de control, mientras que la prueba de rangos con signo de Wilcoxon permitió examinar las diferencias entre las evaluaciones iniciales y finales de los mismos estudiantes.

El análisis estadístico mostró que los estudiantes del grupo experimental obtuvieron mejores resultados después de incorporar ChatGPT en sus actividades de aprendizaje de programación con Python. Al comparar las puntuaciones del pretest y el postest mediante la prueba de Wilcoxon, se encontraron diferencias estadísticamente significativas, lo que permitió reconocer una mejora en el desempeño de los participantes. Por otra parte, la prueba U de Mann-Whitney, aplicada a los resultados de la evaluación final, arrojó un valor de significancia de $p < 0,001$, inferior al nivel de 0,05 establecido para la investigación. Esto confirmó la existencia de diferencias significativas entre ambos grupos, siendo el experimental el que alcanzó las puntuaciones más altas tras la intervención educativa.

Al comparar los resultados de esta investigación con los antecedentes revisados, se encontraron aspectos similares a los descritos por Althobaiti (2021) y Gao et al. (2021), quienes señalaron mejoras en el aprendizaje de contenidos técnicos mediante el uso de herramientas basadas en inteligencia artificial. En sus investigaciones, los autores resaltaron

que estas tecnologías pueden ayudar a los estudiantes a comprender temas que presentan cierta dificultad y obtener mejores resultados durante su formación académica. Si bien estos hallazgos guardan relación con lo encontrado en el presente estudio, la diferencia se encuentra en que aquí se evaluó específicamente el uso de ChatGPT como apoyo para aprender programación digital con Python en estudiantes de Ingeniería de Minas. Por otra parte, Sharma y Kumar (2021) identificaron opiniones favorables entre los docentes respecto a la incorporación de inteligencia artificial en las actividades educativas. Este antecedente permite ampliar la discusión, pues muestra que el aprovechamiento de estas herramientas no depende únicamente de sus posibilidades tecnológicas, sino también de la disposición de los docentes para integrarlas en sus estrategias de enseñanza.

Los hallazgos de esta investigación también guardan relación con lo señalado por Smith (2020) y González (2019), quienes abordaron el uso de herramientas de inteligencia artificial en la enseñanza de las matemáticas y otras disciplinas de carácter técnico. Según lo reportado en dichos estudios, la incorporación de estos recursos puede favorecer el desempeño académico de los estudiantes y ofrecer alternativas a los métodos tradicionales de enseñanza. Esta coincidencia resulta relevante para la presente investigación, ya que permite relacionar las mejoras observadas en el aprendizaje de Python con las posibilidades que ofrecen las tecnologías de inteligencia artificial dentro de la formación universitaria. En la presente investigación, la incorporación de ChatGPT también estuvo acompañada de avances en la autonomía del aprendizaje y en la capacidad de los estudiantes para afrontar situaciones problemáticas vinculadas con la programación.

De manera similar, los hallazgos coinciden con lo señalado por Rojas y Espinoza (2021) y Valdivia y Cabrera (2021), quienes sostienen que la inteligencia artificial puede favorecer el rendimiento académico cuando ofrece una interacción más directa y orientaciones ajustadas a las necesidades del estudiante En este estudio, la posibilidad de recibir respuestas inmediatas mediante ChatGPT permitió brindar apoyo durante la resolución de situaciones técnicas, especialmente en actividades relacionadas con la comprensión y aplicación de contenidos de programación.. Esto se pudo notar principalmente en la comprensión de los fundamentos de Python y en la forma en que aplicaron estos conocimientos en ejercicios relacionados con la ingeniería.

Como aporte principal, la investigación permitió observar que el uso de ChatGPT dentro de la enseñanza de Programación Digital con Python no solo estuvo relacionado con una mejora en las calificaciones, sino también con una mayor capacidad para utilizar lo

aprendido en situaciones prácticas. Los resultados aportan evidencia sobre la utilidad que pueden tener las herramientas de IA como apoyo en el aprendizaje de áreas técnicas. Además, la experiencia desarrollada puede servir como referencia para futuros estudios interesados en analizar el uso de este tipo de tecnologías dentro de la formación en ciencias e ingeniería.

5. PROPUESTA PEDAGÓGICA

5.1. Introducción

Los avances tecnológicos han influido en las formas de enseñar y aprender, permitiendo que las instituciones educativas incorporen nuevas herramientas para apoyar el desarrollo de conocimientos y habilidades técnicas. En este escenario, la inteligencia artificial ha comenzado a ocupar un lugar importante como recurso complementario durante la formación académica. Uno de los campos donde estas herramientas pueden aprovecharse es la programación con Python, un lenguaje utilizado en diferentes actividades de ingeniería, análisis de datos y automatización de procesos. Dentro de estas posibilidades se encuentra ChatGPT, que permite a los estudiantes realizar consultas, revisar ejemplos de código y recibir orientación mientras desarrollan sus ejercicios, facilitando así la comprensión y aplicación de los contenidos relacionados con la programación.

En el campo educativo, las herramientas de inteligencia artificial pueden brindar apoyo a los estudiantes de acuerdo con las dificultades que encuentran durante su aprendizaje. Su utilización permite acceder a explicaciones y ejemplos relacionados con los temas que necesitan reforzar, ofreciendo la posibilidad de revisar los contenidos a su propio ritmo y recibir orientación según sus conocimientos y necesidades académicas. Esta capacidad puede favorecer una asimilación más completa de los contenidos desarrollados (Luckin et al., 2018).

La propuesta pedagógica desarrollada en esta investigación estuvo orientada a incorporar ChatGPT como una herramienta complementaria en la enseñanza de programación con Python a los estudiantes de Ingeniería de Minas. Para su desarrollo, se trabajaron dos aspectos principales: el conocimiento de los fundamentos del lenguaje y la utilización de estos aprendizajes en ejercicios prácticos de programación digital. Aprender a programar implica algo más que escribir instrucciones o conocer el funcionamiento de las estructuras de control, pues también requiere organizar ideas, establecer procedimientos lógicos y buscar soluciones a problemas mediante herramientas informáticas. En relación

con ello, Wing (2006) destaca la importancia del pensamiento computacional como una capacidad que permite abordar problemas de manera estructurada. Considerando este planteamiento, ChatGPT se incorporó durante las actividades académicas para que los estudiantes pudieran consultar sus dudas, revisar ejemplos de código y recibir orientación cuando encontraban dificultades en sus ejercicios. También se aprovechó su capacidad para explicar errores de programación y proponer alternativas de solución, procurando que los participantes comprendieran los procedimientos utilizados y no se limitaran únicamente a reproducir las respuestas proporcionadas por la herramienta.

La incorporación de herramientas de inteligencia artificial en la enseñanza de programación ofrece nuevas posibilidades para acompañar a los estudiantes durante el desarrollo de sus actividades académicas. Estos recursos pueden facilitar la resolución de dudas, proporcionar explicaciones sobre los temas que presentan mayor dificultad y brindar orientación mientras se realizan ejercicios prácticos. En relación con ello, Roll y Wylie (2016) analizan la evolución de la inteligencia artificial en la educación y destacan las posibilidades de los sistemas de tutoría inteligente y los entornos interactivos para apoyar los procesos formativos. Desde esta perspectiva, el uso de herramientas tecnológicas puede complementar las explicaciones del docente y ofrecer a los estudiantes otras alternativas para comprender los contenidos y desarrollar sus habilidades de programación. Asimismo, la posibilidad de consultar diferentes procedimientos y revisar soluciones puede contribuir a que los estudiantes participen de manera más activa en su aprendizaje, siempre que utilicen estas herramientas con criterio propio.

La propuesta se desarrolló mediante sesiones de aprendizaje en las que se combinaron las explicaciones teóricas con ejercicios de programación en Python, utilizando ChatGPT como una herramienta de consulta y orientación durante las actividades. Se consideró especialmente el Aprendizaje Basado en Problemas (ABP), una estrategia que permite a los estudiantes construir sus conocimientos a partir del análisis de situaciones que requieren encontrar soluciones. Al respecto, Jonassen (2011) destaca la importancia de trabajar con problemas que estimulen el razonamiento, la búsqueda de información y la evaluación de diferentes alternativas. Siguiendo este planteamiento, las sesiones se organizaron para que los estudiantes no se limitaran a reproducir instrucciones, sino que pudieran analizar los ejercicios propuestos, probar distintas formas de resolverlos y comprobar el funcionamiento de sus programas con el apoyo de ChatGPT.

Para el desarrollo de las sesiones de aprendizaje, se estableció una secuencia de actividades en la que ChatGPT se utilizó como una herramienta complementaria durante la formación de los estudiantes. Inicialmente, se trabajaron los conceptos básicos de Python mediante explicaciones y ejemplos que permitieran comprender el funcionamiento del lenguaje. Luego, los participantes realizaron ejercicios de programación para poner en práctica lo aprendido y, cuando surgían dificultades, recurrieron a ChatGPT para revisar sus procedimientos, identificar posibles errores y comparar diferentes alternativas de solución. Esta forma de trabajo permitió relacionar los conocimientos teóricos con su aplicación en ejercicios concretos, procurando que los estudiantes avanzaran progresivamente en el manejo de Python y participaran de manera más activa en la resolución de problemas de programación.

El uso de ChatGPT durante las sesiones de programación con Python permitió incorporar una forma diferente de trabajar los contenidos y desarrollar las actividades académicas. Esta propuesta buscó que los estudiantes no solo alcanzaran mejores resultados en sus evaluaciones, sino que también aprendieran a organizar sus ideas, analizar situaciones y encontrar soluciones mediante el uso de instrucciones y herramientas de programación. En ese sentido, se dio importancia al desarrollo del razonamiento lógico y a la capacidad para resolver ejercicios que exigieran aplicar los conocimientos adquiridos. Estos aprendizajes resultan especialmente relevantes para los estudiantes de Ingeniería de Minas, considerando que durante su ejercicio profesional pueden enfrentar actividades relacionadas con el procesamiento de datos, la automatización de cálculos y la solución de problemas técnicos mediante herramientas informáticas.

5.2. Fundamento de la propuesta

5.2.1. Fundamento filosófico

La propuesta pedagógica se fundamentó en el pragmatismo educativo, una corriente que concede especial importancia a la utilidad de los conocimientos y a su aplicación en situaciones concretas. Desde este planteamiento, el aprendizaje no se reduce a recibir información o memorizar conceptos, sino que requiere poner en práctica lo aprendido, explorar diferentes alternativas y reflexionar sobre los resultados obtenidos. Al respecto, Dewey (1938) destaca el valor de la experiencia en la educación, señalando que el estudiante construye sus conocimientos mediante su participación en actividades que le permiten relacionar lo que aprende con las situaciones de su entorno. Tomando en cuenta estas ideas, la propuesta buscó que los estudiantes desarrollaran sus habilidades a través de ejercicios y

problemas vinculados con la programación en Python, procurando que comprendieran la utilidad de los contenidos y pudieran aplicarlos en actividades relacionadas con su formación en Ingeniería de Minas.

Bajo este enfoque, el uso de ChatGPT como apoyo en las clases de programación con Python permitió que los estudiantes trabajaran no solamente en la comprensión de las reglas y estructuras del lenguaje, sino también en su aplicación mediante ejercicios relacionados con situaciones prácticas. Durante estas actividades, tuvieron la oportunidad de analizar los problemas planteados, organizar sus procedimientos y revisar diferentes alternativas para llegar a una solución. Esta forma de aprendizaje buscó fortalecer su capacidad de razonamiento y favorecer una participación más independiente, de manera que pudieran utilizar los conocimientos adquiridos sin depender constantemente de las explicaciones del docente o de las respuestas proporcionadas por la herramienta.

Durante las actividades de programación con Python, el uso de ChatGPT permitió que los estudiantes participaran de manera más activa en la construcción de sus conocimientos. A través de esta herramienta, pudieron realizar consultas sobre los temas desarrollados, revisar las explicaciones recibidas y reconocer posibles errores en sus códigos para buscar alternativas de solución. Esta dinámica facilitó que los ejercicios no se limitaran a seguir instrucciones, sino que también involucraran la exploración de conceptos y la aplicación de lo aprendido en situaciones prácticas. En relación con ello, Dewey (1916) destaca la importancia de una educación basada en la experiencia y la participación del estudiante, aspectos que se tomaron en cuenta al desarrollar las sesiones de aprendizaje.

El pragmatismo educativo también reconoce la necesidad de revisar las prácticas de enseñanza conforme cambian las condiciones sociales, las herramientas disponibles y las necesidades de formación de los estudiantes. En relación con ello, Rorty (1999) plantea una visión pragmatista que concede importancia a la apertura hacia nuevas experiencias y a la revisión de las ideas y prácticas establecidas. Considerando esta perspectiva, la incorporación de ChatGPT en la enseñanza de programación con Python permitió desarrollar actividades diferentes a las utilizadas habitualmente en el aula. Los estudiantes tuvieron la oportunidad de realizar consultas, revisar ejemplos y buscar soluciones a sus dificultades de acuerdo con su propio ritmo de aprendizaje. Con esta forma de trabajo se buscó que comprendieran mejor los procedimientos de programación, fortalecieran su razonamiento computacional y adquirieran mayor independencia para resolver los problemas planteados durante las sesiones académicas.

La propuesta pedagógica también tomó como referencia los planteamientos del constructivismo, corriente que considera al estudiante como protagonista de su propio aprendizaje. Desde esta perspectiva, adquirir conocimientos no significa únicamente recibir explicaciones, sino relacionar la información nueva con lo que ya se conoce, revisar las propias ideas y construir formas de comprender los contenidos mediante la experiencia y la interacción con el entorno (Piaget, 1950; Vygotsky, 1978). Siguiendo estos planteamientos, ChatGPT se incorporó como una herramienta de apoyo para que los estudiantes pudieran consultar sus dudas, examinar diferentes procedimientos y comprobar sus posibles soluciones durante los ejercicios de programación con Python. Esta forma de trabajo buscó que los participantes comprendieran los errores cometidos, reflexionaran sobre sus respuestas y avanzaran progresivamente en el manejo del lenguaje. De esta manera, se relacionaron los principios constructivistas con el enfoque pragmático de la propuesta, dando importancia a la participación del estudiante, la práctica constante y la utilización de los conocimientos para resolver problemas concretos.

La propuesta pedagógica también consideró los planteamientos del conectivismo, teoría desarrollada por Siemens (2005), que explica cómo el aprendizaje puede construirse mediante las conexiones que las personas establecen con diferentes fuentes de información y recursos disponibles en los entornos digitales. Esta perspectiva adquiere importancia en una educación donde los estudiantes utilizan cada vez más herramientas tecnológicas para ampliar sus conocimientos y resolver sus inquietudes. En relación con ello, ChatGPT se incorporó como un recurso de consulta que permitió a los participantes acceder a explicaciones, revisar ejemplos de programación y recibir orientación durante el desarrollo de sus actividades con Python. Su utilización buscó que los estudiantes tuvieran mayores oportunidades para aprender por cuenta propia, relacionar diferentes conceptos y fortalecer sus conocimientos mediante la interacción con la herramienta, sin dejar de lado la necesidad de verificar la información recibida.

La propuesta pedagógica se apoyó en tres perspectivas que permitieron orientar las actividades de enseñanza: el pragmatismo educativo, el constructivismo y el conectivismo. A partir de estos enfoques, se buscó que los estudiantes aprendieran programación con Python mediante la práctica, la participación activa y el aprovechamiento de los recursos digitales disponibles. En este proceso, ChatGPT se incorporó como una herramienta que, además de facilitar las consultas y ofrecer explicaciones, permitió acompañar el desarrollo de los ejercicios y apoyar la comprensión de los contenidos. Así, se planteó una forma de

enseñanza en la que el uso de la tecnología estuvo relacionado con la construcción de conocimientos, el razonamiento y la búsqueda de soluciones a problemas de programación.

5.2.2. Fundamento pedagógico

Para fundamentar la propuesta pedagógica, se tomaron como referencia la teoría del aprendizaje significativo de Ausubel (1968) y los planteamientos de Vygotsky (1978) sobre la zona de desarrollo próximo. Ambas perspectivas permitieron orientar las actividades de enseñanza de programación con Python, considerando la importancia de los conocimientos previos y del acompañamiento que necesitan los estudiantes cuando enfrentan tareas de mayor dificultad. A partir de estos fundamentos, se organizaron estrategias que relacionaron las explicaciones teóricas con ejercicios prácticos, incorporando ChatGPT como una herramienta de consulta y orientación. De esta manera, se buscó que los participantes comprendieran progresivamente los contenidos, fortalecieran sus habilidades de programación y adquirieran mayor independencia para resolver los problemas planteados durante las sesiones de aprendizaje.

De acuerdo con la teoría de Ausubel (1968), el aprendizaje adquiere mayor significado cuando el estudiante logra relacionar la información nueva con los conocimientos que ya posee. Considerando este planteamiento, durante las sesiones de programación con Python se utilizó ChatGPT para ofrecer explicaciones y ejemplos que ayudaran a los participantes a comprender los temas desarrollados. Esta herramienta permitió retomar conocimientos relacionados con la lógica matemática, la elaboración de algoritmos y la solución de problemas, facilitando su relación con las nuevas actividades de programación. Por su parte, Mayer (2008) destaca la importancia de establecer conexiones entre los conocimientos previos y la información que se incorpora durante el aprendizaje, debido a que este proceso favorece la comprensión y posterior aplicación de los contenidos. Bajo esta perspectiva, ChatGPT se empleó como un recurso complementario para reforzar las explicaciones del docente y ayudar a que los estudiantes construyeran una comprensión más clara de los fundamentos de Python.

El uso de ChatGPT durante las actividades de aprendizaje también permitió que los estudiantes revisaran y organizaran mejor sus conocimientos sobre programación con Python. A través de las consultas realizadas, pudieron recibir explicaciones sobre las reglas de escritura del código, el funcionamiento de las estructuras de control y las diferentes formas de desarrollar un programa. Esta interacción les brindó la oportunidad de aclarar sus dudas y relacionar los conceptos estudiados con los ejercicios que realizaban en clase. Al

respecto, VanLehn (2011) y Woolf (2020) analizan las posibilidades que ofrecen los sistemas de tutoría inteligente para apoyar la comprensión de contenidos y orientar el aprendizaje en disciplinas científicas y tecnológicas. Si bien estos sistemas presentan características diferentes a las de ChatGPT, sus aportes permiten reconocer la importancia de contar con herramientas digitales que complementen la enseñanza y ayuden a los estudiantes a desarrollar sus conocimientos con mayor autonomía.

Según los planteamientos de Vygotsky (1978), el aprendizaje se desarrolla mediante la interacción social y el acompañamiento de otras personas que contribuyen a la construcción de conocimientos. Desde esta perspectiva, la orientación del docente y el intercambio de ideas con otros estudiantes cumplen un papel importante, ya que permiten comprender situaciones de mayor dificultad y desarrollar progresivamente habilidades relacionadas con el razonamiento, el análisis y la resolución de problemas. En este sentido, ChatGPT operó como un mediador artificial en la zona de desarrollo próximo (ZDP) de los estudiantes, proporcionándoles apoyo escalonado en la resolución de problemas y promoviendo su progresión desde la dependencia del tutor hasta la autonomía en la programación. Holmes et al. (2019) señalan que las herramientas de inteligencia artificial utilizadas en la educación pueden brindar orientación a los estudiantes mientras desarrollan actividades que requieren un mayor nivel de razonamiento y comprensión. Este acompañamiento puede adaptarse a las dificultades que surgen durante el aprendizaje, ofreciendo explicaciones y ayudas que se reducen progresivamente conforme el estudiante adquiere mayor seguridad y capacidad para resolver los problemas por cuenta propia.

Durante las actividades de programación con Python, el uso de ChatGPT brindó a los estudiantes la oportunidad de revisar la manera en que comprendían y resolvían los ejercicios propuestos. Al interactuar con esta herramienta, pudieron plantear preguntas más específicas, comparar diferentes alternativas de solución e identificar aquellos aspectos que necesitaban reforzar. Estas actividades estuvieron orientadas a desarrollar habilidades metacognitivas, relacionadas con la capacidad de reconocer las propias dificultades y reflexionar sobre los procedimientos utilizados durante el aprendizaje. En relación con ello, Shute y Rahimi (2021) destacan la importancia de la retroalimentación en los entornos educativos, debido a que puede ayudar a los estudiantes a reconocer sus errores, revisar sus estrategias de trabajo y mejorar progresivamente su desempeño. Desde esta perspectiva, ChatGPT se empleó como un recurso de apoyo para fomentar una participación más consciente y autónoma en el aprendizaje de programación.

En el desarrollo de la propuesta pedagógica se consideró el Aprendizaje Basado en Problemas (ABP) como una estrategia para relacionar los conocimientos de programación con situaciones prácticas que los estudiantes debían analizar y resolver. Para ello, se plantearon ejercicios vinculados con problemas concretos, en los que los participantes tuvieron que interpretar la información disponible, organizar sus ideas y establecer procedimientos mediante el lenguaje Python. Al respecto, Jonassen (2011) destaca la importancia de trabajar con problemas que requieran analizar diferentes condiciones, tomar decisiones y evaluar posibles alternativas de solución. Siguiendo este planteamiento, ChatGPT se utilizó como una herramienta de consulta durante el desarrollo de las actividades, brindando explicaciones y ejemplos que ayudaron a los estudiantes a revisar sus procedimientos y explorar distintas formas de construir algoritmos. Con esta metodología se buscó que los participantes comprendieran el razonamiento detrás de cada solución y desarrollaran mayor independencia para enfrentar ejercicios de programación de creciente dificultad.

La propuesta pedagógica también buscó que los estudiantes asumieran una participación más independiente en su aprendizaje, utilizando ChatGPT como una herramienta de apoyo para organizar sus actividades y resolver las dificultades que encontraban durante las sesiones de programación. Mediante esta interacción, tuvieron la posibilidad de revisar los temas a su propio ritmo, realizar consultas sobre contenidos específicos y profundizar en aquellos conceptos que todavía no comprendían con claridad. En relación con ello, Dillenbourg (2016) aborda la importancia de los entornos tecnológicos en el desarrollo de experiencias educativas que favorezcan la participación del estudiante. Considerando esta perspectiva, ChatGPT se incorporó para ofrecer explicaciones y ejemplos de programación acordes con las consultas de cada participante, procurando que pudieran avanzar progresivamente en sus conocimientos y adquirir mayor seguridad al desarrollar los ejercicios. De esta manera, se buscó fortalecer la autonomía durante el aprendizaje de Python, sin dejar de lado la orientación del docente como parte del proceso formativo.

Durante el desarrollo de las sesiones, se combinaron las explicaciones del docente con ejercicios prácticos y actividades orientadas a resolver problemas de programación con Python. En estas tareas, los estudiantes utilizaron ChatGPT como un recurso de consulta para aclarar sus dudas, revisar ejemplos y encontrar posibles soluciones a las dificultades que se presentaban. Esta forma de trabajo permitió complementar la enseñanza habitual con actividades que requerían una mayor participación de los estudiantes, aprovechando las

posibilidades de la inteligencia artificial para reforzar los conocimientos adquiridos en clase. Algunos estudios señalan que incorporar inteligencia artificial en la enseñanza de programación puede ayudar a mejorar la motivación, disminuir la preocupación frente a los errores y favorecer un aprendizaje basado en probar, corregir y volver a intentar (Anderson et al., 2019; Zawacki-Richter et al., 2019).

Como parte de esta metodología, ChatGPT se utilizó para acompañar a los estudiantes en la revisión y mejora de los programas que elaboraban durante las clases de Python. Cuando los estudiantes tenían problemas con sus códigos, recurrían a ChatGPT para revisar qué estaba ocurriendo y conocer otras formas de corregirlos. Después de leer las explicaciones, podían hacer cambios en sus programas y comprobar si los errores se habían solucionado. Esto les ayudó a entender mejor cómo funcionaban las instrucciones de Python y por qué algunas veces los resultados no eran los esperados. Además, con estos ejercicios se buscó que aprendieran a analizar las dificultades, identificar las posibles causas y pensar en sus propias soluciones, en vez de utilizar directamente las respuestas que les daba la herramienta.

La propuesta pedagógica se organizó de manera que los estudiantes fueran aprendiendo Python poco a poco, comenzando por los temas más sencillos y avanzando hacia ejercicios que necesitaban mayor conocimiento. Durante las primeras clases se explicaron las reglas básicas para escribir códigos y el uso de las principales estructuras del lenguaje. Después, conforme se desarrollaban las sesiones, se trabajaron actividades en las que los participantes tenían que elaborar algoritmos y resolver problemas de mayor dificultad. La intención era que comprendieran bien los primeros temas antes de pasar a otros más complicados. Sobre este aspecto, Bruner (1966) señala que la enseñanza debe tener en cuenta lo que el estudiante ya conoce, pues esos aprendizajes sirven como punto de partida para adquirir nuevas habilidades. Por ello, las sesiones se prepararon siguiendo un orden en el que cada contenido se relacionaba con el anterior, buscando que los estudiantes entendieran mejor la programación y tuvieran mayor confianza al desarrollar sus ejercicios con Python.

Durante las clases de programación con Python, también se utilizó ChatGPT para que los estudiantes pudieran revisar sus ejercicios y conocer en qué partes estaban cometiendo errores. Cuando un código no funcionaba como esperaban o alguna instrucción les resultaba difícil, podían consultar la herramienta y recibir explicaciones que les ayudaran a entender el problema. Esto les daba la oportunidad de corregir sus trabajos y continuar

practicando sin tener que esperar siempre la ayuda del docente. Sobre este tema, Aleven et al. (2016) y Mayer (2019) resaltan la importancia de brindar orientación a los estudiantes mientras aprenden, especialmente cuando necesitan reconocer sus errores y comprender cómo pueden mejorar. Teniendo en cuenta estas ideas, ChatGPT se incorporó como un apoyo durante los ejercicios, buscando que los participantes aprendieran de las dificultades que encontraban y fueran adquiriendo mayor seguridad al momento de elaborar y corregir sus programas en Python.

En la propuesta pedagógica también se trabajó la capacidad de los estudiantes para resolver problemas utilizando Python. Durante las clases se desarrollaron ejercicios en los que tenían que buscar distintas formas de llegar a una solución, probar sus códigos y hacer cambios cuando los resultados no eran los esperados. La idea era que no se limitaran a escribir instrucciones, sino que comprendieran cómo funcionaba cada procedimiento y aprendieran a identificar qué alternativas podían ser más útiles para resolver un ejercicio. Sobre este tema, Guzdial (2019) y Resnick (2020) resaltan la importancia de aprender programación mediante la práctica, donde el estudiante tiene la oportunidad de experimentar, cometer errores y mejorar lo que va desarrollando. Siguiendo estas ideas, se buscó que los participantes aprendieran a revisar sus propios trabajos, comprendieran mejor los errores que encontraban y desarrollaran poco a poco mayor habilidad para resolver problemas de programación. Considerando estas ideas, la propuesta incorporó ejercicios que permitieron a los estudiantes revisar sus errores, fortalecer su capacidad de análisis y desarrollar progresivamente el razonamiento necesario para enfrentar problemas de mayor complejidad.

En las clases de programación con Python, los estudiantes utilizaron ChatGPT para consultar sus dudas, revisar ejemplos de código y buscar soluciones cuando tenían dificultades con los ejercicios. Esta herramienta les permitió conocer otras formas de desarrollar un programa, comprobar si las instrucciones funcionaban correctamente y entender mejor los temas que se iban trabajando. Con estas actividades se buscó que los estudiantes no solo aprendieran los conceptos de Python, sino que también pudieran aplicarlos al momento de resolver problemas por su cuenta. Por otro lado, familiarizarse con este tipo de herramientas tecnológicas puede ser útil para su futura vida profesional, sobre todo en Ingeniería de Minas, donde cada vez es más necesario saber manejar información, realizar cálculos, automatizar algunas tareas y utilizar programas informáticos para desarrollar diferentes actividades técnicas.

5.2.3. Fundamento tecnológico

La propuesta pedagógica también tomó en cuenta el aprendizaje apoyado en la tecnología, considerando que las herramientas digitales pueden cambiar la manera en que los estudiantes aprenden y ponen en práctica sus conocimientos. Desde esta idea, ChatGPT se utilizó como un recurso adicional para facilitar la comprensión de los temas y ayudar a los participantes durante el desarrollo de sus actividades de programación con Python. Al respecto, Salomon (1990) destaca que los recursos tecnológicos pueden intervenir en los procesos cognitivos, ofreciendo nuevas posibilidades para organizar la información y desarrollar determinadas habilidades. Considerando esta perspectiva, ChatGPT se empleó como un apoyo para que los estudiantes revisaran los contenidos de programación con Python, relacionaran diferentes conceptos y buscaran alternativas de solución durante el desarrollo de sus actividades académicas.

En los últimos años, las herramientas de inteligencia artificial han comenzado a utilizarse con mayor frecuencia en los espacios educativos, ofreciendo nuevas posibilidades para complementar las actividades desarrolladas por los docentes. En la enseñanza de programación con Python, estos recursos pueden resultar útiles porque permiten que los estudiantes consulten sus dudas mientras realizan los ejercicios, revisen ejemplos de código y reciban explicaciones sobre los temas que todavía no comprenden. De esta manera, cuentan con un apoyo adicional para reforzar sus conocimientos y resolver las dificultades que surgen durante las prácticas de programación.

El aprendizaje apoyado en recursos tecnológicos permite que los estudiantes participen de manera más activa en la construcción de sus conocimientos, utilizando herramientas digitales para explorar información, desarrollar actividades y buscar soluciones a las dificultades que encuentran. Jonassen et al. (1999) señalan que el uso de herramientas informáticas en la educación puede ayudar a que los estudiantes comprendan mejor los temas cuando participan directamente en las actividades y buscan soluciones a los problemas que se les presentan. Siguiendo estas ideas, durante las clases de programación con Python se utilizó ChatGPT como un recurso adicional para resolver dudas, consultar ejemplos y revisar los errores que aparecían al desarrollar los códigos. Cuando algún ejercicio no funcionaba correctamente, los estudiantes podían recurrir a la herramienta para entender qué estaba ocurriendo y probar otras formas de solucionarlo. Con esta manera de trabajar se buscó que los participantes aprendieran mediante la práctica, comprendieran mejor las instrucciones de

Python y fueran desarrollando sus habilidades de programación sin depender únicamente de las explicaciones del docente.

Desde el enfoque del constructivismo digital, la inteligencia artificial puede ser útil para que los estudiantes aprendan temas técnicos mediante la práctica y la exploración de nuevos conocimientos. Estas herramientas ofrecen la posibilidad de revisar información, resolver dudas y desarrollar ejercicios que ayuden a comprender mejor lo aprendido. Sobre este tema, Kay et al. (2020) señalan que los sistemas de tutoría inteligente pueden brindar orientación según las dificultades y los conocimientos que tiene cada estudiante. Tomando en cuenta estas ideas, durante las clases de programación con Python se utilizó ChatGPT como un apoyo para que los participantes hicieran consultas, revisaran ejemplos de código y entendieran cómo funcionaban las diferentes instrucciones del lenguaje. Así, los estudiantes pudieron reforzar los temas que les resultaban complicados y continuar practicando mientras desarrollaban sus actividades. A través de esta interacción, tuvieron la oportunidad de desarrollar ejercicios relacionados con la elaboración de algoritmos, el manejo de estructuras de control y la organización de datos, avanzando progresivamente en el aprendizaje del lenguaje y en la aplicación práctica de sus conocimientos.

El aprendizaje adaptativo apoyado en inteligencia artificial ofrece la posibilidad de atender las diferencias que existen entre los estudiantes, considerando que no todos comprenden los contenidos con la misma facilidad ni necesitan el mismo tipo de orientación. En este sentido, Lucking et al. (2018) y Woolf (2020) destacan las posibilidades de estas tecnologías para brindar apoyo educativo de acuerdo con las necesidades de aprendizaje, mediante explicaciones y recursos que pueden favorecer la participación y el desempeño académico. Tomando en cuenta estos aportes, ChatGPT se incorporó en la presente investigación como una herramienta que permitió a los estudiantes revisar los temas de programación con Python a su propio ritmo, realizar consultas sobre los ejercicios y solicitar explicaciones adicionales cuando encontraban dificultades. De esta manera, se buscó que cada participante contara con mayores oportunidades para comprender los contenidos y avanzar progresivamente en el desarrollo de sus habilidades de programación.

Los avances en el procesamiento del lenguaje natural han permitido desarrollar herramientas de inteligencia artificial capaces de comprender las consultas formuladas por los usuarios y elaborar respuestas mediante un lenguaje cercano a la comunicación humana. Entre estas herramientas se encuentra ChatGPT, que permite mantener conversaciones, solicitar explicaciones y realizar consultas sobre diferentes temas, incluyendo aquellos

relacionados con la programación (OpenAI, 2022). Considerando estas características, en la presente investigación se utilizó ChatGPT como un recurso complementario durante las sesiones de aprendizaje de Python. Los estudiantes pudieron recurrir a esta herramienta para aclarar dudas, revisar el funcionamiento de sus programas y recibir orientación cuando encontraban errores en el código. De esta manera, se buscó facilitar la comprensión de los contenidos y brindar mayores oportunidades para que los participantes desarrollaran sus habilidades de programación mediante la práctica y la revisión de sus propios procedimientos.

Durante el desarrollo de la propuesta pedagógica se incorporó ChatGPT como una herramienta de inteligencia artificial generativa para apoyar a los estudiantes en el aprendizaje de programación con Python. Durante las actividades de programación, los estudiantes utilizaron ChatGPT para consultar cómo podían desarrollar sus programas, revisar ejemplos y aclarar dudas cuando alguna instrucción no funcionaba como esperaban. En relación con este tema, VanLehn (2011) estudió cómo los sistemas de tutoría inteligente pueden ayudar durante el aprendizaje, mientras que Shute y Rahimi (2021) destacaron la importancia de recibir orientación sobre los errores cometidos para mejorar los procedimientos. Teniendo en cuenta estos aportes, en las sesiones se utilizó ChatGPT para apoyar la elaboración de algoritmos, revisar los códigos y buscar otras maneras de resolver los ejercicios. La intención no era que los estudiantes copiaran directamente las respuestas de la herramienta, sino que pudieran revisar las opciones propuestas, entender por qué funcionaban y decidir cuál era la más adecuada para solucionar el problema. De esta manera, se buscó que fueran adquiriendo mayor seguridad al programar y aprendieran a resolver sus dificultades por cuenta propia.

Durante las clases de programación con Python, el uso de ChatGPT permitió que los estudiantes tuvieran otra forma de aprender y resolver las dificultades que encontraban en sus ejercicios. Además de las explicaciones del docente, podían consultar esta herramienta cuando necesitaban entender mejor algún tema o revisar un código que no funcionaba como esperaban. Sobre ello, Aleven et al. (2016) explican que los sistemas de tutoría inteligente pueden servir de apoyo en el aprendizaje, especialmente cuando los estudiantes necesitan orientación para realizar determinadas actividades. En las sesiones desarrolladas, ChatGPT se utilizó para revisar ejemplos, aclarar dudas y encontrar posibles errores en las instrucciones de los programas. Esto fue útil para trabajar aspectos relacionados con la sintaxis y la lógica de Python, ya que los estudiantes podían analizar qué

estaba fallando y probar otras soluciones. Con esta forma de trabajo se buscó que comprendieran mejor sus errores, aprendieran a corregirlos y participaran más en el desarrollo de sus propios ejercicios.

En las clases de programación, la inteligencia artificial puede ser de gran ayuda para los estudiantes, sobre todo cuando tienen dificultades con algún ejercicio o no comprenden bien las instrucciones que deben seguir. Estas herramientas les permiten hacer consultas, revisar ejemplos y encontrar explicaciones que les ayuden a continuar con sus actividades. Sobre este tema, Zawacki-Richter et al. (2019) realizaron una revisión de estudios relacionados con el uso de inteligencia artificial en la educación superior, donde identificaron diferentes aplicaciones, como los tutores inteligentes y otros recursos que sirven de apoyo durante el aprendizaje. A partir de ello, ChatGPT puede utilizarse en la enseñanza de Python para que los estudiantes comprendan mejor los ejercicios, organicen sus ideas y busquen soluciones cuando se les presentan problemas. Así, no tienen que depender únicamente de lo explicado por el docente, sino que cuentan con otra herramienta para seguir practicando y reforzar sus conocimientos.

En las clases de programación con Python, los estudiantes utilizaron ChatGPT como un apoyo para aclarar sus dudas y revisar los ejercicios que iban desarrollando. Cuando tenían dificultades con algún código, podían consultar la herramienta, revisar qué estaba fallando y buscar otras formas de solucionarlo. Esto les ayudó a comprender mejor cómo organizar las instrucciones y qué pasos debían seguir para que sus programas funcionaran correctamente. Con estas actividades también se buscó que aprendieran a pensar de manera más lógica, analizar los problemas antes de resolverlos y encontrar soluciones por sí mismos, sin depender todo el tiempo de las indicaciones del docente.

En cuanto a la metodología, el uso de ChatGPT en las clases de Python permitió trabajar con actividades más prácticas, donde los estudiantes pudieron aprender mediante ejercicios, probar diferentes soluciones y resolver problemas relacionados con la programación. Esta forma de enseñanza buscó que comprendieran mejor los temas desarrollados y pusieran en práctica lo aprendido durante las clases (Guzdial, 2019; Resnick, 2020). En este marco, ChatGPT facilitó el desarrollo de proyectos prácticos en los cuales los estudiantes diseñaron, implementaron y optimizaron algoritmos, fortaleciendo su comprensión del lenguaje y su capacidad para resolver problemas computacionales reales.

Finalmente, el uso de inteligencia artificial en la enseñanza de programación viene siendo estudiado con bastante interés, principalmente por las posibilidades que ofrece para hacer el aprendizaje más flexible y ajustado a las necesidades de cada estudiante. Holmes et al. (2019) señalan que la inteligencia artificial aplicada a la educación puede facilitar la adquisición de conocimientos técnicos y, al mismo tiempo, apoyar una mayor autonomía del estudiante, ya que le brinda recursos para organizar mejor su propio proceso de aprendizaje.

Dentro de esta propuesta, ChatGPT se incorporó como una herramienta de apoyo en la enseñanza de Programación Digital. Su uso permitió trabajar en un entorno más dinámico e interactivo, donde los estudiantes pudieron practicar, resolver dudas y avanzar en el desarrollo de competencias relacionadas con la programación. De esta manera, la experiencia también mostró una forma concreta de incorporar tecnologías emergentes dentro de la educación superior.

5.3. Objetivo

El objetivo de la propuesta fue integrar ChatGPT como una herramienta en el aprendizaje de programación digital con Python, con el fin de mejorar la comprensión de los fundamentos del lenguaje y su aplicación en la programación digital para los estudiantes de ingeniería de minas de la Universidad Nacional de San Cristóbal de Huamanga.

5.4. Descripción de la propuesta

La propuesta de intervención pedagógica plantea el uso de ChatGPT como una herramienta de apoyo durante el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga. La idea es incorporar este recurso dentro de las actividades de enseñanza para que los estudiantes puedan interactuar con un asistente de inteligencia artificial mientras desarrollan los contenidos del curso.

Con esta estrategia se busca apoyar tanto la comprensión de los conceptos básicos de Python como su aplicación en ejercicios y problemas prácticos. De esta manera, ChatGPT se utiliza como un recurso complementario que puede ayudar a resolver dudas, orientar el desarrollo de actividades y facilitar el uso del lenguaje de programación dentro del proceso de aprendizaje.

La propuesta metodológica se organizó en dos etapas de aprendizaje. La primera comprendió las sesiones 1 a 8, dedicadas al conocimiento de los fundamentos de Python, mientras que la segunda abarcó las sesiones 9 a 16, orientadas a la aplicación práctica del

lenguaje en actividades de programación digital. A lo largo de ambas etapas, ChatGPT se incorporó como una herramienta de apoyo para desarrollar los ejercicios, aclarar dudas y reforzar los contenidos trabajados durante las clases. Esta organización permitió mantener una secuencia ordenada de las actividades y realizar un seguimiento del uso de la herramienta durante la intervención educativa, con la finalidad de analizar su relación con los resultados de aprendizaje alcanzados por los estudiantes.

Durante la primera etapa de la propuesta pedagógica se desarrollaron los contenidos básicos de Python, comenzando por las reglas de escritura del código y continuando con el manejo de estructuras de control, funciones y procesamiento de datos. Para acompañar estas actividades, se utilizó ChatGPT mediante consultas previamente elaboradas, con las que los estudiantes pudieron acceder a explicaciones y ejemplos relacionados con los temas de cada sesión. La herramienta también sirvió como apoyo durante la resolución de ejercicios, especialmente cuando los participantes necesitaban comprender alguna instrucción, identificar errores en sus programas o revisar diferentes formas de solucionarlos. Con esta secuencia de trabajo se buscó que los estudiantes adquirieran progresivamente los conocimientos necesarios para manejar las principales funciones del lenguaje y con.

En la segunda etapa de la propuesta se desarrollaron actividades orientadas a utilizar Python en situaciones prácticas de programación digital, incluyendo ejercicios de automatización de tareas, procesamiento de datos y simulaciones numéricas. Durante estas sesiones, ChatGPT se empleó como una herramienta de consulta para acompañar a los estudiantes en la elaboración de algoritmos de mayor dificultad y en la revisión de los programas que desarrollaban. Cuando se presentaban errores o resultados diferentes a los esperados, los participantes podían recurrir a la herramienta para analizar las posibles causas y evaluar distintas alternativas de solución. Esta forma de trabajo buscó que los estudiantes fortalecieran sus habilidades para resolver problemas, organizar mejor las instrucciones y comprobar el funcionamiento de sus códigos, aplicando los conocimientos adquiridos durante la primera etapa del aprendizaje.

Con el propósito de evaluar los resultados de la intervención educativa, se aplicaron las pruebas estadísticas U de Mann-Whitney y de rangos con signo de Wilcoxon. Al revisar los resultados de la evaluación inicial con la prueba U de Mann-Whitney, se encontró que ambos grupos, tanto el experimental como el de control, tuvieron los mismos valores. En los dos casos, el rango promedio fue de 20,50 y la suma de rangos alcanzó 410,00. Estos valores reflejan que, antes de incorporar ChatGPT como herramienta de apoyo en el aprendizaje de

Python, ambos grupos presentaban una distribución similar de sus puntuaciones, lo que permitió establecer una referencia para comparar posteriormente los resultados alcanzados después de la intervención. Esto permitió observar que, al inicio, los puntajes de los dos grupos tenían una distribución bastante similar.

Además, el valor de p obtenido fue de 1,000, resultado que confirmó que antes de la intervención no existían diferencias estadísticamente significativas entre el grupo control y el grupo experimental.

En el postest, la prueba U de Mann-Whitney mostró una diferencia clara entre los dos grupos. El grupo experimental obtuvo un rango promedio de 27,50, mientras que el grupo control alcanzó 13,50. Además, el valor de p fue de 0,000, menor al nivel de significancia de 0,05, por lo que se encontraron diferencias estadísticamente significativas entre ambos grupos.

Estos resultados muestran que los estudiantes que utilizaron ChatGPT presentaron un mejor desempeño después de la intervención. Por ello, los datos respaldan la hipótesis planteada y permiten señalar que el uso de ChatGPT como apoyo pedagógico tuvo una influencia positiva en el aprendizaje de Python.

Por otro lado, la prueba de Wilcoxon en el grupo experimental evidenció que todos los estudiantes mostraron mejoras en el pos test, sin rangos negativos ni empates. El valor Z obtenido (-4.134) y el p-valor asintótico bilateral (0,000) confirman que la diferencia entre pre test y pos test fue significativa, validando la efectividad de la intervención.

Estos resultados muestran que la incorporación de ChatGPT durante el aprendizaje de Programación Digital con Python ayudó a mejorar la comprensión de los temas y también la forma en que los estudiantes aplicaban lo aprendido. La herramienta permitió trabajar en un entorno más interactivo, donde los estudiantes podían resolver dudas y avanzar con mayor autonomía.

Además, la propuesta permitió observar que el uso de inteligencia artificial puede apoyar el desarrollo de habilidades relacionadas con la programación en los estudiantes de Ingeniería de Minas. Esto puede contribuir tanto a su formación académica como a su preparación para enfrentar posteriormente situaciones propias del ámbito profesional.

Unidad I: Fundamentos de Python

Sesión 1: Introducción a Python y su Aplicación en la Ingeniería de Minas

***Prompt*:**

```
Piensa en una herramienta que hayas utilizado para analizar datos (Excel, calculadora, software de minería). ¿Qué ventajas crees que tiene un lenguaje de programación como Python?

Explícame qué es Python, sus características y su importancia en la Ingeniería de Minas.

Muéstrame ejemplos de cómo Python se aplica en análisis de datos geológicos.

Ahora, genera un código en Python que imprima un mensaje de bienvenida y mis datos personales (nombre, carrera y universidad).

Finalmente, reflexiona: ¿Cómo podría Python mejorar procesos en minería que actualmente dependen de software tradicional?
```

Introducción a Python y su Aplicación en la Ingeniería de Minas

---

1️⃣ Reflexión Inicial: Comparando Herramientas de Análisis de Datos

Antes de explorar Python, pensemos en algunas herramientas que se usan en la minería para analizar datos:

🔹 **Excel:** Manejo de bases de datos, fórmulas y gráficos.
🔹 **Calculadora científica:** Realización de cálculos numéricos precisos.
🔹 **Software especializado en minería:** Modelado geológico, simulaciones, análisis de datos mineros (ej. Datamine, Surpac, Leapfrog).

🤔 **Pregunta:**
¿Qué ventajas crees que tiene un lenguaje de programación como Python frente a estas herramientas?

📌 **Algunas ventajas clave de Python:**
✅ Automatización de tareas repetitivas.
✅ Manejo de grandes volúmenes de datos de manera eficiente.
✅ Integración con múltiples bibliotecas científicas para análisis avanzados.
✅ Generación de reportes automatizados y visualización de datos en gráficos avanzados.

2️⃣ ¿Qué es Python? Características y Aplicaciones en Ingeniería de Minas

Python es un **lenguaje de programación de alto nivel**, diseñado para ser fácil de aprender y usar. Algunas de sus características clave incluyen:

🔹 **Sintaxis sencilla:** Similar al lenguaje natural, lo que lo hace accesible.
🔹 **Multiparadigma:** Soporta programación estructurada, orientada a objetos y funcional.
🔹 **Extensibilidad:** Compatible con bibliotecas como NumPy, Pandas y Matplotlib.
🔹 **Multiplataforma:** Funciona en Windows, Linux y Mac sin modificaciones.
🔹 **Comunidad Activa:** Gran cantidad de documentación y soporte en línea.

📌 **Importancia en Ingeniería de Minas:**
Python permite **procesar, analizar y visualizar datos geológicos** de manera más eficiente que el software tradicional. Algunas aplicaciones clave incluyen:

✅ **Procesamiento de datos geofísicos y geoquímicos.**
✅ **Análisis de perforaciones para evaluar calidad del mineral.**
✅ **Simulación de procesos mineros como lixiviación y flotación.**
✅ **Automatización de reportes de producción y costos mineros.**
✅ **Modelado 3D de yacimientos combinando datos de distintas fuentes.**

3️⃣ Ejemplos de Aplicación de Python en Minería

💡 Ejemplo 1: Cálculo de densidad de minerales

python Copiar Editar

```python
# Cálculo de densidad de un mineral (masa/volumen)
def calcular_densidad(masa, volumen):
    densidad = masa / volumen
    return densidad

# Datos de ejemplo
masa_mineral = 2500  # en kg
volumen_mineral = 1.2  # en m³

# Cálculo
densidad_mineral = calcular_densidad(masa_mineral, volumen_mineral)
print(f"La densidad del mineral es {densidad_mineral:.2f} kg/m³")
```

💡 Ejemplo 2: Análisis de perforaciones con Pandas

python Copiar Editar

```python
import pandas as pd

# Creación de un DataFrame con datos de perforación
datos_perforacion = {
    "Profundidad (m)": [10, 20, 30, 40, 50],
    "Ley de Oro (g/t)": [0.5, 0.8, 1.2, 0.9, 1.5],
    "Densidad (kg/m³)": [2700, 2750, 2800, 2780, 2760]
}

df = pd.DataFrame(datos_perforacion)
```

```python
# Mostrar estadísticas básicas
print("Resumen de datos de perforación:")
print(df.describe())
```

---

4 Código en Python: Mensaje de Bienvenida Personalizado

python Copiar Editar

```python
# Datos personales
nombre = "Juan Pérez"
carrera = "Ingeniería de Minas"
universidad = "Universidad Nacional de San Cristóbal de Huamanga"

# Mensaje de bienvenida
print("¡Bienvenido a la programación en Python!")
print(f"Nombre: {nombre}")
print(f"Carrera: {carrera}")
print(f"Universidad: {universidad}")
print("Prepárate para aplicar Python en minería de datos y optimización de procesos.")
```

📌 **Ejecuta este código en Python para ver el mensaje personalizado.**

5️⃣ Reflexión Final: ¿Cómo podría Python mejorar procesos en minería?

🤔 **Pregunta:**

Actualmente, muchas operaciones mineras dependen de software especializado, pero ¿qué problemas podríamos solucionar usando Python?

✅ **Optimización de Procesos:** Automatización del análisis de datos de perforaciones sin depender de software de pago.

✅ **Reducción de Costos:** Creación de modelos personalizados sin pagar licencias costosas.

✅ **Escalabilidad:** Python puede manejar grandes volúmenes de datos de sensores en tiempo real.

✅ **Integración con IA:** Uso de Machine Learning para predecir calidad de minerales antes de la extracción.

🚀 **Conclusión:**

Python es una **herramienta poderosa y flexible** que permite a los ingenieros de minas analizar datos de manera más rápida, eficiente y económica que el software tradicional.

✅ ¡Tu Turno!

1️⃣ Ejecuta los códigos proporcionados y analiza los resultados.

2️⃣ Modifica los ejemplos con **tus propios datos de minería.**

3️⃣ Reflexiona: ¿Cómo podrías aplicar Python en un problema real de minería?

📌 **Comparte tus respuestas y mejoras con ChatGPT para seguir aprendiendo.** 🚀



Sesión 2: Tipos de Datos, Operadores y Estructuras de Control

***Prompt*:**

```
Imagina que estás analizando datos de un yacimiento minero. ¿Qué tipos de datos crees que necesitarías almacenar y procesar?

Explica los tipos de datos en Python (numéricos, booleanos, cadenas y complejos).

Muestra cómo operar con estos tipos de datos en cálculos mineros.

Genera un código en Python que realice operaciones matemáticas con variables de diferentes tipos.

Luego, pídele a ChatGPT que revise tu código y te dé sugerencias para optimizarlo. ¿Cómo podrías mejorarlo?
```

Tipos de Datos, Operadores y Estructuras de Control

Imagina que estás analizando datos de un **yacimiento minero**. En este contexto, el manejo de información es esencial para la toma de decisiones en exploración, extracción y procesamiento de minerales.

📌 **Pregunta inicial:**
¿Qué tipos de datos crees que necesitas almacenar y procesar en minería?

Algunos ejemplos incluyen:

- **Datos numéricos:** Ley del mineral, tonelaje extraído.
- **Datos booleanos:** Estado de una máquina minera (operativa o fuera de servicio).
- **Datos de texto:** Nombre del yacimiento, tipo de roca predominante.
- **Datos complejos:** Cálculo de resistencias geomecánicas en minería subterránea.

Explicación de los tipos de datos en Python

Python permite representar estos datos mediante diferentes tipos de variables:

🔷 **Numéricos** (`int`, `float`, `complex`) → Para valores como tonelajes, concentraciones y cálculos geotécnicos.
🔷 **Booleanos** (`bool`) → Para verificar condiciones en minería, como si un equipo está activo (`True` o `False`).
🔷 **Cadenas de texto** (`str`) → Para almacenar información descriptiva, como el nombre de un proyecto minero.

💡 **Ejemplo en código:**

python Copiar Editar

```python
# Datos básicos de un yacimiento minero
nombre_yacimiento = "Mina San José"
ley_mineral = 3.2  # gramos por tonelada
equipo_operativo = True  # Estado de operación de la maquinaria
resistencia_material = complex(5, 2)  # Resistencia de material en números complejos

print(f"Nombre del yacimiento: {nombre_yacimiento}")
print(f"Ley del mineral: {ley_mineral} g/t")
print(f"Equipo en operación: {equipo_operativo}")
print(f"Resistencia del material: {resistencia_material}")
```

Cálculos con Tipos de Datos en Minería

En minería, es común realizar operaciones matemáticas para calcular volúmenes, leyes de mineral y eficiencia de recuperación.

🔷 **Ejemplo de cálculo del oro recuperado**

python Copiar Editar

```python
# Datos de una operación minera
toneladas_extraidas = 2500
ley_mineral = 2.5  # gramos por tonelada
recuperacion = 88 / 100  # Recuperación del proceso metalúrgico

# Cálculo del oro recuperado
oro_recuperado = toneladas_extraidas * ley_mineral * recuperacion

# Mostrar resultado
print(f"Oro recuperado: {oro_recuperado:.2f} gramos")
```

Optimización con ChatGPT

Es importante revisar y mejorar el código con ayuda de herramientas avanzadas. Para ello, copia y pega tu código en ChatGPT y solicita una **revisión y optimización**.

💡 **Ejemplo de interacción con ChatGPT:**

"*¿Cómo puedo mejorar este código para hacerlo más eficiente? ¿Podría estructurarlo mejor usando funciones o estructuras de control?*"

✅ **¡Tu Turno!**

1️⃣ Ejecuta los códigos proporcionados y analiza los resultados.
2️⃣ Modifica los ejemplos con **tus propios datos de minería.**
3️⃣ Reflexiona: **¿Cómo podrías aplicar Python en un problema real de minería?**

📌 **Comparte tus respuestas y mejoras con ChatGPT para seguir aprendiendo.** 🚀

Sesión 3: Funciones y Modularidad en Python

***Prompt*:**

```
¿Por qué crees que es importante dividir un código en funciones en lugar de escribir todo en un solo bloque?

Explícame qué son las funciones en Python y por qué son importantes en programación.

Muéstrame cómo crear funciones con parámetros y valores de retorno.

Genera un código en Python que defina una función para calcular la densidad de un mineral a partir de su masa y volumen.

Finalmente, responde: ¿Cómo podrías reutilizar esta función para analizar diferentes minerales?
```

Funciones y Modularidad en Python

En minería, el procesamiento de datos es una tarea fundamental que requiere cálculos repetitivos y estructurados. **El uso de funciones** en Python permite organizar el código, facilitando su reutilización y mejorando su eficiencia.

📌 **Pregunta inicial:**
¿Por qué crees que es importante dividir un código en funciones en lugar de escribirlo todo en un solo bloque?

Algunas ventajas incluyen:

- **Reutilización del código**, evitando repeticiones innecesarias.
- **Mayor claridad y organización**, facilitando la lectura y depuración.
- **Modificación sencilla**, permitiendo cambios sin afectar todo el programa.

¿Qué son las funciones en Python y por qué son importantes?

Las funciones son **bloques de código reutilizables** que ejecutan una tarea específica. Se definen con `def` y pueden recibir **parámetros** para procesar información y retornar valores.

🔷 **Ejemplo básico de una función en minería:**

python Copiar Editar

```python
# Función que calcula la densidad de un mineral
def calcular_densidad(masa, volumen):
    densidad = masa / volumen
    return densidad

# Datos de un mineral
masa_mineral = 500  # en gramos
volumen_mineral = 200  # en cm³

# Uso de la función
densidad_mineral = calcular_densidad(masa_mineral, volumen_mineral)
print(f"La densidad del mineral es {densidad_mineral} g/cm³")
```

💡 **Observación:**

La función `calcular_densidad` puede reutilizarse para analizar diferentes minerales, solo cambiando los valores de **masa** y **volumen**.

Aplicación en la Ingeniería de Minas

En minería, la densidad es clave para:

✅ Identificar tipos de minerales en yacimientos.

✅ Determinar la calidad del material extraído.

✅ Optimizar procesos de separación y beneficio.

🔷 **Ejemplo avanzado: Comparación de densidades de diferentes minerales**

python Copiar Editar

```python
# Lista de minerales con masa y volumen
minerales = {
    "Oro": (19300, 1),   # Densidad en kg/m³
    "Cuarzo": (2650, 1),
    "Pirita": (5000, 1)
}

# Calcular y mostrar densidades
for mineral, datos in minerales.items():
    masa, volumen = datos
    print(f"La densidad de {mineral} es {calcular_densidad(masa, volumen)} kg/m³")
```

Optimización con ChatGPT

Solicita a ChatGPT que revise tu código y sugiera mejoras en eficiencia y estructura.

💡 **Ejemplo de interacción con ChatGPT:**

"¿Cómo puedo mejorar la función para hacerla más eficiente y adaptable a distintos minerales?"

✅ **¡Tu Turno!**

1️⃣ Ejecuta los códigos proporcionados y analiza los resultados.
2️⃣ Modifica los ejemplos con **tus propios datos de minería.**
3️⃣ Reflexiona: **¿Cómo podrías aplicar funciones en otro proceso minero?**

📌 **Comparte tus respuestas y mejoras con ChatGPT para seguir aprendiendo.** 🚀

Sesión 4: Estructuras de Datos: Listas, Tuplas y Diccionarios

***Prompt*:**

```
En minería, se necesita almacenar información de diferentes minerales, como su dureza, densidad y composición. ¿Cómo crees que podríamos organizar estos datos en Python?

Explica qué son las listas, tuplas y diccionarios en Python, y en qué casos se usan.

Muestra ejemplos de su uso en almacenamiento de datos mineros.

Genera un código en Python que almacene información de diferentes minerales y permita su consulta.

Luego, prueba tu código con distintos valores y analiza: ¿Cuál estructura de datos fue más eficiente para manejar los datos mineros?
```

Estructuras de Datos: Listas, Tuplas y Diccionarios en Python

📌 **Objetivo de la sesión:** Al finalizar la clase, los estudiantes comprenderán las estructuras de datos listas, tuplas y diccionarios en Python y su aplicación en la minería para organizar y gestionar información geológica.

---

1. Activación de Saberes Previos

🔷 **Pregunta detonadora:**
"Si tuvieras que organizar la información de distintos minerales en un archivo o base de datos, ¿cómo lo harías?"

💡 **Discusión guiada:**

- ¿Cómo organizamos información en la vida cotidiana? (Ejemplo: listas de compras, catálogos, directorios).
- ¿Qué características tienen estos sistemas de organización?
- ¿Qué tipos de datos se manejan en minería y cómo se registran?

📝 **Actividad breve:**
Los estudiantes escriben en la pizarra o en una hoja de papel los datos que podrían almacenar sobre un mineral y cómo los estructurarían (tabla, lista, categorías).

---

2. Introducción a las Estructuras de Datos en Python

🔷 **Explicación teórica:**
En minería, los datos deben organizarse de manera eficiente para facilitar su consulta y análisis. Python ofrece tres estructuras fundamentales:

1️⃣ **Listas:** Almacenan datos ordenados y pueden modificarse.
2️⃣ **Tuplas:** Son como listas, pero no pueden modificarse después de su creación.
3️⃣ **Diccionarios:** Permiten almacenar datos en pares clave-valor, facilitando el acceso a la información.

📌 **Ejemplo comparativo en minería:**

Estructura	Uso en minería	Ejemplo
Lista	Almacenar nombres de minerales	`minerales = ["Cuarzo", "Pirita", "Oro", "Diamante"]`
Tupla	Almacenar valores constantes como la dureza de un mineral	`dureza = (7, 6, 2.5, 10)`
Diccionario	Almacenar propiedades detalladas de cada mineral	`minerales_info = {"Cuarzo": {"Dureza": 7, "Densidad": 2.65}}`

👀 **Pregunta para reflexionar:** ¿Qué estructura de datos es más eficiente para manejar grandes volúmenes de información en minería?

---

3. Desarrollo de la Clase: Programación con Listas, Tuplas y Diccionarios

🔷 Ejercicio 1: Creando y manipulando listas

python Copiar Editar

```python
# Lista de minerales
minerales = ["Cuarzo", "Pirita", "Oro", "Diamante"]

# Agregar un nuevo mineral
minerales.append("Cobre")

# Eliminar un mineral
minerales.remove("Pirita")

# Ordenar la lista
minerales.sort()

# Mostrar la lista final
print("Lista de minerales actualizada:", minerales)
```

✍️ Actividad:

- Agregar otros minerales a la lista.
- Ordenar la lista en orden inverso.
- Buscar si un mineral específico está en la lista.

🔷 Ejercicio 2: Uso de tuplas

python Copiar Editar

```python
# Tupla con las densidades de algunos minerales en g/cm³
densidades = (2.65, 5.0, 19.3, 3.5)

# Acceder a los valores
print("Densidad del cuarzo:", densidades[0])

# Intentar modificar la tupla (esto generará un error)
densidades[0] = 2.7  # Python no permite modificar tuplas
```

✍️ Actividad:

- ¿Por qué las tuplas no pueden modificarse?
- ¿En qué casos sería útil utilizar una tupla en minería?

```python
# Diccionario con propiedades de los minerales
minerales_info = {
    "Cuarzo": {"Dureza": 7, "Densidad": 2.65, "Composición": "SiO₂"},
    "Pirita": {"Dureza": 6, "Densidad": 5.0, "Composición": "FeS₂"},
    "Oro": {"Dureza": 2.5, "Densidad": 19.3, "Composición": "Au"},
    "Diamante": {"Dureza": 10, "Densidad": 3.5, "Composición": "C"}
}

# Consultar las propiedades de un mineral específico
mineral = "Pirita"
print(f"Propiedades de {mineral}: {minerales_info[mineral]}")
```

✍️ **Actividad:**

- Agregar un nuevo mineral con sus propiedades.
- Modificar la dureza de un mineral.
- Eliminar un mineral del diccionario.

---

4. Aplicación en la Ingeniería de Minas

✅ **Casos de uso en minería:**

1. **Análisis geológico:** Registro y consulta rápida de información sobre minerales.
2. **Optimización de procesos:** Organización de datos para modelado geológico.
3. **Exploración minera:** Almacenamiento y consulta de datos de yacimientos.

📌 **Discusión en grupo:** ¿Cómo aplicarían estas estructuras de datos en un software de análisis minero?

---

5. Cierre y Reflexión

🔷 **Preguntas de cierre:**

- ¿Qué ventajas tiene usar listas, tuplas y diccionarios en Python?

- ¿Qué estructura de datos prefieres y por qué?
- ¿Cómo podrías mejorar los ejemplos para usarlos en minería?

📌 **Actividad final:**
Modifica los ejemplos de código con **tus propios datos de minería** y justifica tu elección.

🚀 **Comparte tus respuestas y mejoras con ChatGPT para seguir aprendiendo.**

Sesión 5: Manejo de Archivos y Entrada/ Salida de Datos

Prompt:

```
Si tuvieras que guardar datos de muestras geológicas para analizarlas más tarde, ¿cómo lo harías en Python?

Explícame cómo leer y escribir archivos en Python.

Muestra ejemplos de almacenamiento y procesamiento de datos mineros en archivos `.txt` y `.csv`.

Genera un código en Python que guarde y lea datos de minerales desde un archivo de texto.

Finalmente, analiza: ¿Cómo podrías mejorar la eficiencia en la lectura de grandes volúmenes de datos geológicos?
```

Manejo de Archivos y Entrada/Salida de Datos en Python

📌 **Objetivo de la sesión:** Al finalizar la clase, los estudiantes comprenderán cómo leer y escribir archivos en Python, aplicándolo al almacenamiento y procesamiento de datos mineros en formatos `.txt` y `.csv`.

---

1. Activación de Saberes Previos

🔷 **Pregunta detonadora:**
"Si tuvieras que registrar los datos de muestras geológicas para analizarlas después, ¿cómo lo harías?"

💡 **Discusión guiada:**
- ¿Cómo almacenamos información en papel y en digital?
- ¿Qué ventajas tiene guardar datos en archivos en comparación con otros métodos?
- ¿Cómo se manejan grandes volúmenes de datos en la minería?

📝 **Actividad breve:**
Los estudiantes anotan en la pizarra posibles datos que deberían registrar en un archivo sobre muestras de minerales.

---

2. Introducción al Manejo de Archivos en Python

🔷 **Explicación teórica:**
Los archivos permiten almacenar datos para su posterior procesamiento. En Python, se pueden manejar archivos de diferentes formatos, como:

- **Archivos de texto (`.txt`):** Contienen datos en formato de texto plano.
- **Archivos CSV (`.csv`):** Utilizan comas o tabulaciones para estructurar datos en filas y columnas.

📌 **Ejemplo comparativo en minería:**

Formato	Uso en minería	Ejemplo
TXT	Registro de datos en formato simple	`mineral_datos.txt`
CSV	Almacenamiento estructurado para análisis	`mineral_datos.csv`

👀 **Pregunta para reflexionar:** ¿Qué ventajas tiene cada formato para manejar información minera?

---

3. Desarrollo de la Clase: Lectura y Escritura de Archivos en Python

🔷 **Ejercicio 1: Creando y leyendo un archivo de texto (`.txt`)**

python Copiar Editar

```python
# Crear un archivo y escribir datos de minerales
with open("minerales.txt", "w") as archivo:
    archivo.write("Cuarzo,7,2.65\n")
    archivo.write("Pirita,6,5.0\n")
    archivo.write("Oro,2.5,19.3\n")
    archivo.write("Diamante,10,3.5\n")

# Leer el archivo y mostrar su contenido
with open("minerales.txt", "r") as archivo:
```

```
    contenido = archivo.readlines()
    for linea in contenido:
        print(linea.strip())  # strip() elimina espacios y saltos de línea
```

✍️ **Actividad:**

- Modificar el archivo agregando un nuevo mineral.
- Leer y mostrar solo los minerales con dureza mayor a 5.

🔷 **Ejercicio 2: Uso de archivos CSV (`.csv`)**

python Copiar Editar

```python
import csv

# Crear y escribir datos en un archivo CSV
with open("minerales.csv", mode="w", newline="") as archivo:
    escritor = csv.writer(archivo)
    escritor.writerow(["Nombre", "Dureza", "Densidad"])
    escritor.writerow(["Cuarzo", 7, 2.65])
    escritor.writerow(["Pirita", 6, 5.0])
    escritor.writerow(["Oro", 2.5, 19.3])
    escritor.writerow(["Diamante", 10, 3.5])

# Leer el archivo CSV e imprimir los datos
with open("minerales.csv", mode="r") as archivo:
    lector = csv.reader(archivo)
    for fila in lector:
        print(fila)
```

✍️ **Actividad:**

- Agregar nuevas filas con más minerales.
- Filtrar los minerales con densidad superior a 3.

---

4. Aplicación en la Ingeniería de Minas

✅ **Casos de uso en minería:**

1. **Almacenamiento de datos de muestras geológicas**
2. **Generación de reportes automáticos**

3. **Intercambio de datos entre programas de análisis minero**

📌 **Discusión en grupo:** ¿Cómo mejorar la eficiencia al manejar grandes volúmenes de datos geológicos?

---

5. Cierre y Reflexión

🔷 **Preguntas de cierre:**

- ¿Cuál es la diferencia entre archivos `.txt` y `.csv`?
- ¿Cuál formato es más adecuado para almacenar grandes volúmenes de datos?
- ¿Cómo aplicarías estos conceptos en un proyecto minero real?

📌 **Actividad final:**
Los estudiantes crean y manipulan archivos con **sus propios datos de minería** y explican su utilidad en un contexto real.

🚀 **Comparte tus respuestas y mejoras con ChatGPT para seguir aprendiendo.**

Sesión 6: Programación Orientada a Objetos (POO)

Prompt:

```
Imagina que debes modelar un yacimiento minero en Python. ¿Cómo representarías un mineral con todas sus propiedades?

Explícame los conceptos de la Programación Orientada a Objetos (POO) en Python.

Muestra cómo se crean clases y objetos, aplicados a la minería.

Genera un código en Python que defina una clase `Mineral` con atributos como `nombre`, `densidad` y `dureza`.

Luego, reflexiona: ¿Cómo podrías extender este modelo para incluir más información relevante en minería?
```

Programación Orientada a Objetos (POO) en Python aplicada a la Minería

📌 **Objetivo de la sesión:**
Al finalizar la clase, los estudiantes comprenderán los fundamentos de la Programación Orientada a Objetos (POO) en Python y su aplicación en la modelación de datos mineros, implementando clases y objetos para representar minerales y sus propiedades.

---

1. Activación de Saberes Previos

🔷 **Pregunta detonadora:**
"Si tuvieras que modelar un mineral en un programa de computadora, ¿qué características incluirías?"

💡 **Discusión guiada:**

- ¿Cómo almacenaríamos información sobre minerales en un programa?
- ¿Qué elementos de un mineral son más importantes en minería?
- ¿Por qué sería útil representar un mineral como un "objeto" en programación?

📝 **Actividad breve:**
Los estudiantes mencionan características de los minerales y el docente las organiza en la pizarra.

---

2. Introducción a la Programación Orientada a Objetos (POO)

🔷 **Explicación teórica:**
La Programación Orientada a Objetos (POO) es un paradigma que organiza el código en entidades llamadas "objetos", que representan elementos del mundo real con propiedades y comportamientos.

Conceptos clave en POO:

Concepto	Definición	Ejemplo en minería
Clase	Es un molde para crear objetos	Clase `Mineral`
Objeto	Es una instancia de una clase	Un mineral específico (Ej. Cuarzo)
Atributo	Características del objeto	Nombre, densidad, dureza
Método	Acciones que puede realizar el objeto	Mostrar información del mineral

👀 **Pregunta para reflexionar:** ¿Cómo la POO puede hacer más fácil la gestión de información minera en Python?

---

3. Desarrollo de la Clase: Creando Clases y Objetos en Python

🔷 **Ejercicio 1: Definiendo una clase `Mineral`**

python Copiar Editar

```python
# Definición de la clase Mineral
class Mineral:
    def __init__(self, nombre, densidad, dureza):
        self.nombre = nombre
        self.densidad = densidad
        self.dureza = dureza

    def mostrar_info(self):
        print(f"Mineral: {self.nombre}, Densidad: {self.densidad} g/cm³, Dureza: {self.dureza}

# Creación de objetos (instancias de la clase Mineral)
cuarzo = Mineral("Cuarzo", 2.65, 7)
pirita = Mineral("Pirita", 5.0, 6)

# Mostrar información de los minerales
cuarzo.mostrar_info()
pirita.mostrar_info()
```

✍️ **Actividad:**

- Crear objetos de otros minerales con diferentes valores de densidad y dureza.
- Modificar la clase para agregar un nuevo atributo como "composición química".

🔷 **Ejercicio 2: Extendiendo la clase para modelar más propiedades**

python Copiar Editar

```python
# Extender la clase para incluir más información
class Mineral:
    def __init__(self, nombre, densidad, dureza, composicion):
        self.nombre = nombre
        self.densidad = densidad
        self.dureza = dureza
        self.composicion = composicion

    def mostrar_info(self):
        print(f"Mineral: {self.nombre}, Densidad: {self.densidad} g/cm³, Dureza: {self.dureza}

# Crear una instancia con la nueva estructura
oro = Mineral("Oro", 19.3, 2.5, "Au")
oro.mostrar_info()
```

✍️ **Actividad:**

- Agregar métodos para comparar la dureza de dos minerales.
- Implementar una lista de minerales utilizando objetos.

---

4. Aplicación en la Ingeniería de Minas

✅ **Casos de uso en minería:**

1. **Modelado de depósitos minerales en software de geología**
2. **Simulación de propiedades físicas y químicas de minerales**
3. **Automatización del procesamiento de datos mineros**

📌 **Discusión en grupo:**

- ¿Cómo podría mejorar este modelo incluyendo características como resistencia o ubicación del yacimiento?
- ¿Cómo se compararía esta forma de modelar con el uso de listas o diccionarios?

---

5. Cierre y Reflexión

🔷 **Preguntas de cierre:**

- ¿Cómo la POO ayuda a organizar mejor el código en Python?
- ¿Cuáles son las ventajas de representar un mineral como un objeto en lugar de solo una lista de datos?
- ¿Cómo podrías extender este modelo para simular una mina completa?

📌 **Actividad final:**
Cada estudiante implementa una clase `Yacimiento` que contenga varios minerales y presenta su código a la clase.

🚀 **¡Prueba tu código con ChatGPT y sigue explorando la POO en minería!**

Sesión 7: Algoritmos y Estructuras de Datos Avanzadas

Prompt:

```
¿Cómo crees que se ordenan los datos en un software de análisis minero?

Explícame qué son los algoritmos y por qué son importantes en la programación.

Muéstrame ejemplos de estructuras de datos avanzadas aplicadas a minería.

Genera un código en Python que ordene y busque datos en una lista de minerales usando algoritmos eficientes.

Finalmente, reflexiona: ¿Cómo la elección de un algoritmo afecta el rendimiento en minería de datos?
```

Algoritmos y Estructuras de Datos Avanzadas en Python Aplicadas a la Minería

📌 **Objetivo de la sesión:**
Al finalizar la clase, los estudiantes comprenderán la importancia de los algoritmos y estructuras de datos avanzadas en la optimización del procesamiento de información minera, implementando métodos eficientes de ordenamiento y búsqueda en Python.

---

1. Activación de Saberes Previos

🔷 **Pregunta detonadora:**
"¿Cómo crees que se ordenan los datos en un software de análisis minero?"

💡 **Discusión guiada:**

- ¿Por qué es importante organizar los datos en minería?
- ¿Cómo afecta el uso de algoritmos eficientes en el rendimiento de una aplicación?
- ¿Qué sucede si procesamos grandes volúmenes de datos sin un buen algoritmo?

📝 **Actividad breve:**
Los estudiantes comparten experiencias sobre el manejo de grandes volúmenes de datos y el docente introduce el concepto de algoritmos y estructuras avanzadas.

---

2. Introducción a los Algoritmos y su Importancia en Programación

🔷 **Explicación teórica:**

Los algoritmos son secuencias de pasos lógicos para resolver un problema. En minería, los algoritmos permiten procesar y analizar datos de manera eficiente.

Ejemplos de uso en minería:

Proceso minero	Algoritmo aplicado
Ordenamiento de muestras geológicas	QuickSort, MergeSort
Búsqueda de minerales en bases de datos	Búsqueda binaria
Procesamiento de datos de sensores mineros	Algoritmos de estructuras de datos (árboles, pilas, colas)

📌 **Reflexión:** ¿Qué características debe tener un buen algoritmo para minería de datos?

---

3. Desarrollo de la Clase: Implementación de Algoritmos en Python

🔷 **Ejercicio 1: Implementación de Algoritmos de Ordenamiento**

python Copiar Editar

```python
# Algoritmo de ordenamiento rápido (QuickSort)
def quicksort(lista):
    if len(lista) <= 1:
        return lista
    pivote = lista[len(lista) // 2]
    izquierda = [x for x in lista if x < pivote]
    centro = [x for x in lista if x == pivote]
    derecha = [x for x in lista if x > pivote]
    return quicksort(izquierda) + centro + quicksort(derecha)

# Lista de minerales con densidades
minerales = [("Cuarzo", 2.65), ("Pirita", 5.0), ("Oro", 19.3), ("Hematita", 5.26)]
minerales_ordenados = quicksort(minerales)

print("Minerales ordenados por densidad:", minerales_ordenados)
```

✍️ **Actividad:**

- Aplicar el algoritmo para ordenar minerales por dureza.
- Comparar QuickSort con el método de ordenamiento integrado de Python (`sorted()`).

🔷 **Ejercicio 2: Búsqueda Eficiente de Minerales**

python Copiar Editar

```python
# Algoritmo de búsqueda binaria
def busqueda_binaria(lista, elemento):
    izquierda, derecha = 0, len(lista) - 1
    while izquierda <= derecha:
        medio = (izquierda + derecha) // 2
        if lista[medio][0] == elemento:
            return f"{elemento} encontrado con densidad {lista[medio][1]}"
        elif lista[medio][0] < elemento:
            izquierda = medio + 1
        else:
            derecha = medio - 1
    return f"{elemento} no encontrado"

# Lista ordenada de minerales
minerales_ordenados = sorted(minerales, key=lambda x: x[0])
print(busqueda_binaria(minerales_ordenados, "Pirita"))
```

✍️ **Actividad:**

- Implementar búsqueda binaria para encontrar minerales por densidad.
- Comparar con una búsqueda lineal y analizar la diferencia en eficiencia.

---

4. Aplicación en la Ingeniería de Minas

✅ **Casos de uso en minería:**

1. **Optimización de bases de datos geológicas**
2. **Análisis eficiente de sensores y datos mineros**
3. **Ordenamiento de muestras para estudios geotécnicos**

📌 **Discusión en grupo:**

- ¿Qué algoritmo de ordenamiento es más eficiente para grandes volúmenes de datos?

- ¿Cómo se aplican estos conceptos en softwares mineros como Datamine o Surpac?

---

5. Cierre y Reflexión

🔷 **Preguntas de cierre:**

- ¿Por qué es importante elegir el algoritmo adecuado según la cantidad de datos?
- ¿Qué estructuras de datos avanzadas podrían mejorar aún más el rendimiento?
- ¿Cómo podrías optimizar un sistema de análisis geológico con estos conocimientos?

📌 **Actividad final:**

Cada estudiante implementa un algoritmo de búsqueda y ordenamiento para un conjunto de datos de minerales, probando distintas estrategias y comparando su rendimiento.

Sesión 8: Optimización de Código y Buenas Prácticas en Python

Prompt:

```
¿Cuál crees que es la diferencia entre un código funcional y un código
eficiente?

Explícame qué son las buenas prácticas de programación en Python y su
importancia.

Muéstrame ejemplos de cómo escribir código eficiente y modular.

Genera un código en Python que optimice el cálculo de concentraciones de
minerales usando funciones y estructuras de datos eficientes.

Finalmente, usa ChatGPT para evaluar tu código y obtener sugerencias de mejora.
¿Qué cambios aplicarías?
```

Optimización de Código y Buenas Prácticas en Python Aplicadas a la Minería

📌 **Objetivo de la sesión:**

Al finalizar la clase, los estudiantes comprenderán la importancia de la optimización del código en Python, aplicando buenas prácticas de programación y estructuras de datos eficientes para mejorar el rendimiento en el cálculo de concentraciones de minerales.

---

1. Activación de Saberes Previos

🔷 **Pregunta detonadora:**

"¿Cuál crees que es la diferencia entre un código funcional y un código eficiente?"

💡 **Discusión guiada:**

- ¿Qué significa que un código sea "eficiente" en términos de tiempo de ejecución y consumo de memoria?
- ¿Cómo afecta la estructura del código a su mantenimiento y reutilización?
- ¿Qué problemas podrían surgir al manejar grandes volúmenes de datos sin optimización?

📝 **Actividad breve:**

Los estudiantes comparten ejemplos de código que han utilizado y discuten posibles mejoras.

---

2. Buenas Prácticas de Programación en Python

🔷 **Explicación teórica:**

Las buenas prácticas en programación ayudan a mejorar la legibilidad, mantenibilidad y eficiencia del código.

Principales buenas prácticas en Python:

✅ **Código modular:** Uso de funciones y clases para evitar redundancia.
✅ **Estructuras de datos eficientes:** Uso de listas, diccionarios y tuplas en lugar de estructuras ineficientes.
✅ **Optimización de loops:** Evitar iteraciones innecesarias y usar comprensión de listas.
✅ **Uso de bibliotecas optimizadas:** Aplicar NumPy y Pandas para procesamiento eficiente de datos.
✅ **Documentación y estilo de código:** Seguir la guía de estilo PEP 8 para mejorar la legibilidad.

📌 **Ejemplo de código ineficiente vs. código optimizado:**

Código ineficiente

python Copiar Editar

```python
concentraciones = [10, 20, 30, 40, 50]
suma = 0
for i in range(len(concentraciones)):
    suma += concentraciones[i]
promedio = suma / len(concentraciones)
print("Promedio:", promedio)
```

Código optimizado con comprensión de listas y NumPy

python Copiar Editar

```python
import numpy as np

concentraciones = np.array([10, 20, 30, 40, 50])
promedio = np.mean(concentraciones)

print("Promedio:", promedio)
```

📌 **Pregunta para reflexión:** ¿Por qué la segunda versión del código es más eficiente?

---

3. Desarrollo de la Clase: Optimización del Cálculo de Concentraciones de Minerales

🔷 **Ejercicio 1: Cálculo eficiente de concentraciones de minerales**

python Copiar Editar

```python
import numpy as np

# Datos simulados de concentraciones de minerales en ppm
minerales = {
    "Cobre": [50, 60, 55, 70, 65],
    "Oro": [2, 3, 1.5, 2.5, 3.2],
    "Plata": [10, 15, 12, 11, 13]
}

# Cálculo optimizado con NumPy
for mineral, valores in minerales.items():
    valores_np = np.array(valores)
    print(f"{mineral} - Promedio: {np.mean(valores_np):.2f} ppm, Desviación estándar: {np.std
```

✍️ **Actividad:**

- Implementar un código similar para el cálculo de la mediana y moda.
- Comparar tiempos de ejecución usando estructuras tradicionales vs. NumPy.

🔷 **Ejercicio 2: Optimización con Pandas para grandes volúmenes de datos**

python Copiar Editar

```python
import pandas as pd

# Creación de un DataFrame con datos simulados
datos = {
    "Mineral": ["Cobre", "Oro", "Plata", "Cobre", "Oro", "Plata"],
    "Concentración (ppm)": [50, 2, 10, 60, 3, 15]
}
df = pd.DataFrame(datos)

# Cálculo optimizado
promedios = df.groupby("Mineral")["Concentración (ppm)"].mean()
print(promedios)
```

📌 **Reflexión:** ¿Cómo mejora Pandas el procesamiento de datos en comparación con listas y bucles tradicionales?

---

4. Aplicación en la Ingeniería de Minas

✅ **Casos de uso en minería:**

1. **Procesamiento de grandes volúmenes de datos geológicos**
2. **Optimización del análisis de concentraciones de minerales**
3. **Mejoras en el rendimiento de software minero con estructuras eficientes**

📌 **Discusión en grupo:**

- ¿Qué técnicas de optimización son más útiles para manejar datos mineros en Python?
- ¿Cómo podrías aplicar ChatGPT para mejorar la eficiencia de tu código?

---

5. Evaluación y Reflexión Final

🔷 **Actividad con ChatGPT:**

Los estudiantes ingresan su código en ChatGPT para obtener sugerencias de optimización y comparan los resultados.

📌 **Preguntas de cierre:**

- ¿Qué cambios aplicaste después de recibir sugerencias de ChatGPT?
- ¿Cómo podrías seguir optimizando tu código en proyectos reales?

🚀 **¡Optimiza tu código con buenas prácticas y ChatGPT para mejorar el rendimiento en minería de datos!**

Unidad II: Python aplicado a la Programación Digital

Sesión 9: Introducción a Bibliotecas Científicas; NumPy y Pandas

Prompt:

```
¿Cómo se analizan grandes volúmenes de datos en minería?

Explícame qué son NumPy y Pandas y cómo se usan en el análisis de datos científicos.

Muéstrame ejemplos donde se analicen datos mineros con estas bibliotecas.

Genera un código en Python que use Pandas para leer un archivo CSV con datos de perforaciones mineras y muestre un análisis básico.

Finalmente, reflexiona: ¿Cómo podrías automatizar la limpieza y procesamiento de datos mineros con Pandas?
```

Análisis de Datos Mineros con NumPy y Pandas

📌 **Objetivo de la sesión:**
Al finalizar la clase, los estudiantes comprenderán el uso de **NumPy y Pandas** para el análisis y procesamiento eficiente de grandes volúmenes de datos en minería.

---

1. Activación de Saberes Previos

🔷 **Pregunta detonadora:**
"¿Cómo se analizan grandes volúmenes de datos en minería?"

💡 **Discusión guiada:**

- ¿Qué tipo de datos se generan en la minería (perforaciones, concentraciones, coordenadas, etc.)?
- ¿Cuáles son los desafíos de manejar grandes volúmenes de datos en minería?
- ¿Por qué es importante automatizar el procesamiento de datos?

📝 **Actividad breve:**
Los estudiantes mencionan problemas que han enfrentado con grandes volúmenes de datos y cómo los han resuelto.

---

2. Introducción a NumPy y Pandas

🔷 **Explicación teórica:**

¿Qué es NumPy?

- Biblioteca para cálculos numéricos en Python.
- Optimizada para trabajar con **arrays** multidimensionales.
- Más rápida que las listas de Python para operaciones matemáticas.

📌 **Ejemplo básico de NumPy:**

python Copiar Editar

```python
import numpy as np

# Crear un array con concentraciones de minerales (ppm)
concentraciones = np.array([50, 60, 55, 70, 65])
print("Promedio:", np.mean(concentraciones))
print("Máximo:", np.max(concentraciones))
print("Mínimo:", np.min(concentraciones))
```

¿Qué es Pandas?

- Biblioteca para manipulación y análisis de datos estructurados.
- Usa estructuras eficientes como **DataFrames** y **Series**.
- Permite leer, limpiar, procesar y visualizar datos con facilidad.

📌 **Ejemplo básico de Pandas:**

python Copiar Editar

```python
import pandas as pd

# Crear un DataFrame con datos de perforaciones
datos = {
```

```
    "Muestra": ["A1", "A2", "A3"],
    "Profundidad (m)": [10, 15, 12],
    "Concentración (ppm)": [50, 65, 55]
}
df = pd.DataFrame(datos)
print(df)
```

📌 **Pregunta para reflexión:** ¿Por qué Pandas es más eficiente que usar listas y diccionarios?

---

3. Desarrollo de la Clase: Análisis de Datos Mineros con Pandas

🔷 **Ejercicio 1: Cargar y analizar un archivo CSV con Pandas**

python Copiar Editar

```python
import pandas as pd

# Cargar datos desde un archivo CSV (simulado)
df = pd.read_csv("perforaciones.csv")

# Mostrar las primeras filas del dataset
print(df.head())

# Análisis básico
print("Resumen estadístico:\n", df.describe())
```

✍️ **Actividad:**

- Descargar un archivo CSV con datos de perforaciones mineras.
- Usar Pandas para explorarlo y obtener estadísticas básicas.
- Identificar datos faltantes y proponer estrategias para su limpieza.

🔷 **Ejercicio 2: Limpieza y procesamiento de datos con Pandas**

python Copiar Editar

```python
# Eliminar valores nulos
df_clean = df.dropna()

# Reemplazar valores faltantes con la media
```

```
df["Concentración (ppm)"].fillna(df["Concentración (ppm)"].mean(), inplace=True)

# Filtrar datos de concentraciones mayores a 50 ppm
df_filtrado = df[df["Concentración (ppm)"] > 50]
print(df_filtrado)
```

📌 **Discusión:** ¿Cómo mejorarías este código para un procesamiento más eficiente?

---

4. Aplicación en la Ingeniería de Minas

✅ **Casos de uso en minería:**

1. **Optimización del análisis de perforaciones y muestreos.**
2. **Automatización del procesamiento de datos de sensores en minería.**
3. **Generación de reportes y gráficos de tendencias en exploración minera.**

📌 **Discusión en grupo:**

- ¿Cómo podrían usarse estas técnicas en la exploración minera real?
- ¿Qué ventajas ofrece Pandas frente a herramientas como Excel?

---

5. Evaluación y Reflexión Final

🔷 **Actividad con ChatGPT:**
Los estudiantes ingresan un código de análisis de datos en ChatGPT y obtienen sugerencias de optimización.

📌 **Preguntas de cierre:**

- ¿Qué mejoras aplicarías a tu código después de usar ChatGPT?
- ¿Cómo podrías automatizar aún más el procesamiento de datos mineros?

🚀 **¡Aprovecha la potencia de Pandas y NumPy para analizar datos mineros de manera eficiente!**

Sesión 10: Visualización de Datos con Matplotlib

Prompt:

```
¿Cómo representarías gráficamente la variación de la concentración de un mineral en una mina?

Explícame cómo funciona la biblioteca Matplotlib para la visualización de datos.

Muéstrame ejemplos de gráficos aplicados a minería.

Genera un código en Python que grafique la variación de la concentración de un mineral en diferentes puntos de muestreo.

Finalmente, responde: ¿Cómo podrías mejorar la visualización para que sea más comprensible para un ingeniero de minas?
```

Visualización de Datos en Minería con Matplotlib

📌 **Objetivo de la sesión:**
Al finalizar la clase, los estudiantes comprenderán el uso de **Matplotlib** para representar gráficamente la variación de la concentración de minerales en diferentes puntos de muestreo en una mina.

---

1. Activación de Saberes Previos

🔷 **Pregunta detonadora:**
"¿Cómo representarías gráficamente la variación de la concentración de un mineral en una mina?"

💡 **Discusión guiada:**

- ¿Qué tipo de gráficos conocen para representar datos numéricos?
- ¿En qué situaciones han utilizado gráficos en su formación o trabajo?
- ¿Por qué es importante la visualización de datos en minería?

📝 **Actividad breve:**
Los estudiantes mencionan herramientas que han usado para gráficos y comentan sobre sus ventajas y desventajas.

---

2. Introducción a Matplotlib

🔷 **Explicación teórica:**

¿Qué es Matplotlib?

- Es una biblioteca de Python utilizada para crear gráficos y visualizaciones de datos.
- Permite representar información en diferentes formatos como líneas, barras, dispersión y histogramas.
- Es ampliamente utilizada en ciencia de datos, minería y geología.

📌 **Ejemplo básico de Matplotlib:**

python Copiar Editar

```python
import matplotlib.pyplot as plt

# Datos de ejemplo: concentraciones de un mineral en distintos puntos
puntos = ["P1", "P2", "P3", "P4", "P5"]
concentraciones = [50, 70, 65, 80, 60]

# Crear gráfico de barras
plt.bar(puntos, concentraciones, color='green')
plt.xlabel("Puntos de Muestreo")
plt.ylabel("Concentración (ppm)")
plt.title("Concentración de Mineral en Diferentes Puntos de Muestreo")
plt.show()
```

📌 **Pregunta para reflexión:** ¿Qué tipo de gráfico es más adecuado para representar variaciones de concentración?

---

3. Desarrollo de la Clase: Aplicaciones de Matplotlib en Minería

🔷 **Ejercicio 1: Gráfico de Líneas para Variación de Concentraciones**

python Copiar Editar

```python
import matplotlib.pyplot as plt

# Datos de concentración en diferentes profundidades
profundidades = [10, 20, 30, 40, 50]
concentraciones = [55, 60, 58, 65, 70]
```

```python
# Crear gráfico de líneas
plt.plot(profundidades, concentraciones, marker='o', linestyle='-', color='blue')

plt.xlabel("Profundidad (m)")
plt.ylabel("Concentración (ppm)")
plt.title("Variación de Concentración con la Profundidad")
plt.grid()
plt.show()
```

✍️ **Actividad:**

- Analizar el gráfico y discutir qué patrones pueden identificarse.
- Cambiar los valores y observar cómo varían los resultados.

🔷 **Ejercicio 2: Gráfico de Dispersión para Relación entre Variables**

python Copiar Editar

```python
import numpy as np
import matplotlib.pyplot as plt

# Simulación de datos de perforaciones
profundidad = np.random.randint(10, 100, 50)  # Profundidad en metros
concentracion = np.random.randint(40, 80, 50)  # Concentración en ppm

# Crear gráfico de dispersión
plt.scatter(profundidad, concentracion, color='red', alpha=0.7)

plt.xlabel("Profundidad (m)")
plt.ylabel("Concentración (ppm)")
plt.title("Relación entre Profundidad y Concentración de Mineral")
plt.grid()
plt.show()
```

📌 **Discusión:** ¿Cómo podría mejorarse este gráfico para hacerlo más claro y útil para un ingeniero de minas?

---

4. Aplicación en la Ingeniería de Minas

✅ **Casos de uso en minería:**

1. **Análisis de tendencias en la concentración de minerales.**
2. **Identificación de patrones en la distribución de recursos mineros.**
3. **Visualización de datos de perforaciones para toma de decisiones.**

📌 **Discusión en grupo:**

- ¿Qué otros tipos de gráficos podrían usarse en minería?
- ¿Cómo influye una buena visualización en la interpretación de datos?

---

5. Evaluación y Reflexión Final

🔷 **Actividad con ChatGPT:**
Los estudiantes ingresan su código en ChatGPT para obtener sugerencias sobre cómo mejorar sus gráficos.

📌 **Preguntas de cierre:**

- ¿Cómo podrías hacer que tu gráfico sea más comprensible para un ingeniero de minas?
- ¿Qué otros datos podrías incluir en los gráficos para mejorar el análisis?

Sesión 11: Análisis de Datos Mineros con Python

Prompt:

```
En minería, la calidad del mineral extraído es clave. ¿Cómo podríamos analizar grandes volúmenes de datos de perforaciones para optimizar la producción?

Explícame cómo usar Python para analizar datos en la Ingeniería de Minas.

Muéstrame ejemplos donde se realicen cálculos de concentración y pureza de minerales.

Genera un código en Python que analice un conjunto de datos mineros y determine la media y la desviación estándar.

Finalmente, reflexiona: ¿Cómo podrías usar este análisis para tomar decisiones en la extracción de minerales?
```

Análisis de Datos Mineros con Python

Objetivo de la sesión:

- Analizar cómo Python puede ser utilizado para procesar y analizar grandes volúmenes de datos mineros.
- Aplicar funciones estadísticas, como la media y la desviación estándar, a datos de perforaciones mineras.
- Calcular la concentración y la pureza de minerales utilizando Python para optimizar la toma de decisiones en la extracción minera.

---

1. Saberes previos:

- Los estudiantes deben tener conocimientos básicos de **Python**, como la creación de variables, uso de operadores, y estructuras de control.
- Es útil que los estudiantes conozcan los conceptos de **concentración** y **pureza** de minerales, y cómo estos indicadores son relevantes para la minería.
- Deben estar familiarizados con el uso de bibliotecas en Python, especialmente **Pandas** y **NumPy**, para la manipulación de datos y el cálculo de estadísticas.

---

2. Introducción:

En minería, el análisis de datos geológicos y de perforaciones es esencial para optimizar la producción y asegurar la calidad del mineral extraído. A través de herramientas como **Python**, se pueden analizar grandes volúmenes de datos, lo que facilita la toma de decisiones informadas. En esta sesión, se presentará cómo usar **Python** para realizar cálculos de **concentración** y **pureza** de minerales, así como para calcular estadísticas como la **media** y la **desviación estándar**. Estos cálculos permitirán evaluar la variabilidad de los minerales extraídos y mejorar la eficiencia de los procesos mineros.

---

3. Desarrollo de la clase:

a) Uso de Python en minería:

- **Bibliotecas esenciales: Pandas** se utiliza para manipular datos, mientras que **NumPy** permite realizar cálculos matemáticos y estadísticos.
- **Concentración de minerales:** Se calcula la cantidad de mineral útil en la muestra. Este cálculo es clave para determinar la rentabilidad de la extracción.

- **Pureza del mineral:** La pureza se refiere a la cantidad de mineral valioso presente en la muestra, un indicador crucial en la evaluación de calidad.
- **Cálculos estadísticos:** La **media** proporciona un valor promedio de la concentración o pureza de las muestras, mientras que la **desviación estándar** mide la dispersión de esos valores, ayudando a identificar variabilidad en los datos.

b) Ejemplo de análisis de datos mineros:

- Se presenta un conjunto de datos simulado que contiene información sobre la concentración y la pureza de minerales extraídos de varias perforaciones.
- Se utiliza **Python** para calcular la **media** y la **desviación estándar** de los minerales.

Código en Python:

python Copiar

```python
import pandas as pd
import numpy as np

# Datos de perforación minera simulados
data = {'Mina': ['Mina1', 'Mina2', 'Mina3', 'Mina4', 'Mina5'],
        'Concentracion': [8.5, 9.2, 7.8, 9.4, 8.1],
        'Pureza': [90, 85, 88, 92, 87]}

# Crear DataFrame
df = pd.DataFrame(data)

# Cálculo de la media y desviación estándar de concentración y pureza
media_concentracion = np.mean(df['Concentracion'])
desviacion_concentracion = np.std(df['Concentracion'])

media_pureza = np.mean(df['Pureza'])
desviacion_pureza = np.std(df['Pureza'])

# Mostrar los resultados
print(f"Media de concentración: {media_concentracion}")
print(f"Desviación estándar de concentración: {desviacion_concentracion}")
print(f"Media de pureza: {media_pureza}")
print(f"Desviación estándar de pureza: {desviacion_pureza}")
```

c) Análisis de los resultados:

- El cálculo de la **media** proporciona una visión general de la calidad promedio de los minerales en la muestra.
- La **desviación estándar** indica cuánta variabilidad existe entre las muestras, ayudando a identificar áreas

con alta o baja consistencia en las concentraciones de minerales.

- Con estos resultados, se puede tomar decisiones más informadas sobre qué perforaciones son las más rentables.

---

4. Aplicación en la Ingeniería de Minas:

El análisis de datos mineros con Python es de gran utilidad en la **Ingeniería de Minas**. La capacidad de procesar y analizar grandes volúmenes de datos permite optimizar el proceso de extracción de minerales. A través de las herramientas estadísticas, los ingenieros pueden:

- Identificar las zonas con mayor concentración de minerales, lo que mejora la eficiencia en la extracción.
- Evaluar la variabilidad de la pureza en diferentes perforaciones, lo que puede influir en la planificación de la mina.
- Tomar decisiones informadas basadas en datos precisos sobre la rentabilidad de las operaciones mineras.

Por ejemplo, si una perforación muestra una baja concentración de mineral y alta desviación estándar, puede ser más rentable buscar otros sitios en lugar de continuar extrayendo de ese lugar. Así, Python se convierte en una herramienta poderosa para la **optimización de recursos** y la **toma de decisiones estratégicas**.

---

5. Evaluación y reflexión final:

Evaluación:

- Los estudiantes deberán utilizar el código proporcionado para analizar un conjunto de datos diferente (por ejemplo, concentración de minerales a diferentes profundidades) y calcular la media y desviación estándar de esos datos.
- Reflexionar sobre cómo estos análisis pueden ser aplicados en un entorno real de minería para mejorar la toma de decisiones y optimizar los procesos de extracción.

Reflexión final:

- ¿Cómo influye el análisis estadístico de la concentración y pureza de minerales en la toma de decisiones dentro de la minería?
- Reflexionar sobre la importancia de herramientas como **Python** en la mejora de los procesos de minería y la gestión de grandes volúmenes de datos, facilitando una **gestión más eficiente** y una **toma de decisiones basada en datos**.

Sesión 12: Automatización de Procesos en Python

Prompt:

```
En minería, muchas tareas son repetitivas, como la conversión de unidades y el
análisis de datos. ¿Cómo podríamos automatizar estos procesos con Python?

Explícame cómo Python puede usarse para automatizar tareas repetitivas.

Muestra ejemplos donde se automatice el procesamiento de datos en minería.

Genera un código en Python que procese automáticamente archivos con datos
geológicos y genere un resumen de análisis.

Finalmente, responde: ¿Cómo podrías hacer que este código se ejecute de manera
programada sin intervención humana?
```

Automatización de Procesos en Python

Objetivos de la sesión:

- **Automatización de tareas repetitivas**: Explicar cómo Python puede ser utilizado para automatizar tareas repetitivas en el análisis de datos mineros, como la conversión de unidades y el procesamiento de grandes volúmenes de datos.
- **Generación de informes automáticos**: Enseñar a usar Python para procesar archivos con datos geológicos y generar automáticamente un resumen de análisis.

1. Saberes previos:

- Los estudiantes deben tener conocimientos básicos de **Python**, como la creación de variables, el uso de operadores, y las estructuras de control.
- Es útil que los estudiantes conozcan tareas repetitivas comunes en minería, como la conversión de unidades o el análisis de datos geológicos.
- Los estudiantes deben estar familiarizados con el uso de bibliotecas en Python, especialmente **Pandas** y **NumPy**, para la manipulación de datos y el cálculo de estadísticas.

2. Introducción:

En minería, muchas tareas son repetitivas, como la conversión de unidades o el análisis de grandes volúmenes de datos. Estos procesos pueden automatizarse para optimizar el tiempo y los recursos, reduciendo la intervención humana. En esta sesión, aprenderemos cómo **Python** se puede utilizar para automatizar tareas en minería, facilitando la toma de decisiones informadas y mejorando la eficiencia operativa.

3. Desarrollo de la clase:

a) Uso de Python para automatización:

- **Bibliotecas esenciales**:

- **Pandas**: Para manipular y analizar grandes volúmenes de datos.
- **NumPy**: Para realizar cálculos matemáticos y estadísticos.

b) Ejemplo de automatización: Automatizar la conversión de unidades y el procesamiento de datos geológicos con Python.

Código en Python para procesamiento de datos:

python Copiar

```python
import pandas as pd
import numpy as np

# Cargar los datos geológicos desde un archivo CSV
data = pd.read_csv('datos_geologicos.csv')

# Convertir la profundidad de metros a kilómetros
data['profundidad_km'] = data['profundidad_m'] / 1000

# Calcular estadísticas básicas (media y desviación estándar)
media_concentracion = np.mean(data['concentracion'])
desviacion_concentracion = np.std(data['concentracion'])

# Generar un informe automatizado
informe = f"Resumen del análisis:\n\nMedia de concentración: {media_concentracion}\nDesviación est

# Guardar el informe en un archivo de texto
with open('informe_analisis.txt', 'w') as f:
    f.write(informe)

print("El informe ha sido generado correctamente.")
```

Explicación del código:

1. Se carga un archivo de datos geológicos con `Pandas`.
2. Se realiza la conversión de unidades, por ejemplo, de metros a kilómetros.
3. Se calculan estadísticas básicas, como la **media** y la **desviación estándar** de la concentración de minerales.
4. El informe generado se guarda automáticamente en un archivo `.txt`.

4. Aplicaciones en la Ingeniería de Minas:

La automatización de tareas repetitivas con **Python** permite a los ingenieros de minas procesar y analizar grandes volúmenes de datos con mayor rapidez y precisión. Algunas aplicaciones incluyen:

- **Automatización de la conversión de unidades**: Conversión de unidades de profundidad, volumen, o

peso de minerales sin intervención manual.

- **Análisis estadísticos automáticos**: Cálculo de estadísticas clave, como la **media** y la **desviación estándar** de la concentración de minerales, para evaluar la calidad del mineral extraído.
- **Generación de informes automáticos**: Automatización de la generación de informes con resúmenes de análisis para tomar decisiones más rápidas y precisas en las operaciones mineras.

5. Evaluación y reflexión final:

Evaluación:

- Los estudiantes deben ejecutar el código con sus propios conjuntos de datos y generar un análisis automatizado de concentración de minerales.
- Los estudiantes deben modificar el código para trabajar con diferentes tipos de datos geológicos y generar informes personalizados.

Reflexión final:

- **¿Cómo podrías hacer que este código se ejecute de manera programada sin intervención humana?**
  - **Automatización programada**: Usar herramientas como **cron** en sistemas operativos basados en Unix o el **Programador de Tareas** en Windows para ejecutar automáticamente el script en momentos determinados, como diariamente o semanalmente.

Sesión 13: Seguridad y Manejo de Errores en Python

Prompt:

```
¿Qué pasaría si al analizar datos geológicos, un error en la entrada de datos causara resultados incorrectos?

Explícame cómo manejar errores en Python con `try-except-finally`.

Muestra ejemplos donde se capture y gestione errores en programas de minería.

Genera un código en Python que valide datos ingresados por el usuario y maneje posibles errores.

Luego, prueba ingresar datos incorrectos y analiza: ¿Cómo mejoraste la robustez del código con manejo de errores?
```

Seguridad y Manejo de Errores en Python

Objetivos de la sesión:

- **Manejo de errores en Python**: Explicar cómo manejar errores en Python utilizando `try-except-finally` para garantizar que el programa continúe ejecutándose correctamente incluso cuando ocurran errores.
- **Ejemplos de manejo de errores**: Mostrar ejemplos de cómo capturar y gestionar errores en programas

de minería.

- **Validación de datos ingresados:** Crear un código que valide los datos ingresados por el usuario y maneje posibles errores.

1. Saberes previos:

- Los estudiantes deben tener conocimientos básicos de **Python**, como la creación de variables, uso de operadores, y estructuras de control.
- Es importante que los estudiantes comprendan cómo la entrada de datos incorrectos puede afectar los resultados del análisis, especialmente en minería.
- Los estudiantes deben estar familiarizados con el uso de **estructuras de control** como *if-else*.

2. Introducción:

En minería, los errores en los datos geológicos pueden afectar significativamente los resultados del análisis y la toma de decisiones. Por lo tanto, es crucial que los programas sean capaces de manejar errores de manera eficiente para garantizar que no se generen resultados incorrectos. En esta sesión, aprenderemos cómo manejar errores en Python utilizando la estructura `try-except-finally`, lo que nos permitirá capturar errores y manejar posibles excepciones de manera controlada.

3. Desarrollo de la clase:

a) Manejo de errores en Python con `try-except-finally`:

- `try`: Se utiliza para ejecutar el bloque de código donde puede ocurrir un error.
- `except`: Captura el error y permite manejarlo de forma personalizada.
- `finally`: Se ejecuta siempre, sin importar si ocurre o no un error, y se usa para limpiar o finalizar operaciones (por ejemplo, cerrar archivos).

b) Ejemplo de manejo de errores: Mostrar cómo manejar errores en un programa de minería donde los datos pueden ser ingresados incorrectamente.

Código en Python para validar la entrada de datos:

python Copiar

```python
def obtener_dato_geologico():
    while True:
        try:
            # Solicitar entrada del usuario
            dato = float(input("Ingrese el valor de concentración de mineral: "))

            # Verificar si el dato es negativo
            if dato < 0:
                raise ValueError("El valor ingresado no puede ser negativo.")
```

```
            return dato
        except ValueError as e:
            print(f"Error: {e}. Por favor ingrese un número válido.")
        finally:
            print("Intento de obtener datos concluido.")

# Llamada a la función
concentracion = obtener_dato_geologico()
print(f"Concentración de mineral: {concentracion}")
```

Explicación del código:

1. Se solicita al usuario ingresar un valor para la concentración de mineral.
2. Si el valor ingresado no es un número válido o es negativo, se captura el error utilizando `except ValueError` y se le solicita nuevamente al usuario que ingrese un dato correcto.
3. El bloque `finally` asegura que siempre se ejecute el mensaje de "Intento de obtener datos concluido".

4. Aplicaciones en la Ingeniería de Minas:

El manejo de errores es esencial en los programas que procesan datos mineros. Algunas aplicaciones incluyen:

- **Validación de entradas**: Los programas deben validar los datos ingresados por los operadores para asegurarse de que sean correctos y relevantes para los cálculos mineros.
- **Prevención de errores en el análisis**: Asegurar que no ocurran errores durante el análisis de datos geológicos, como la conversión incorrecta de unidades o la lectura errónea de datos.
- **Garantizar la robustez del sistema**: Usar manejo de errores para evitar que el sistema se detenga por un error inesperado, asegurando que el proceso continúe.

5. Evaluación y reflexión final:

Evaluación:

- Los estudiantes deben modificar el código proporcionado para agregar más validaciones, como la comprobación de datos de tipo entero y flotante.
- Los estudiantes deberán ingresar datos incorrectos (por ejemplo, un valor negativo o un texto) y analizar cómo el código maneja esos errores.

Reflexión final:

- **¿Cómo mejoraste la robustez del código con el manejo de errores?**
    - La implementación de `try-except-finally` permite que el programa continúe ejecutándose incluso cuando se producen errores, lo que mejora la confiabilidad del sistema y reduce el riesgo de fallos. Además, garantiza que los usuarios siempre reciban retroalimentación sobre los errores de entrada, lo que mejora la experiencia del usuario.

Sesión 14: Desarrollo de un Proyecto de Minería de Datos

Prompt:

```
Si tuvieras que presentar un informe de minería de datos usando Python, ¿cómo organizarías tu análisis?

Guíame en la creación de un proyecto en Python aplicado a minería de datos.

Muéstrame cómo estructurar el código y organizar los datos para el análisis.

Genera un código en Python que use Pandas y Matplotlib para procesar y visualizar datos geológicos.

Luego, responde: ¿Cómo podrías automatizar la actualización de los datos en este proyecto?
```

Desarrollo de un Proyecto de Minería de Datos

Objetivos de la sesión:

- **Organización del análisis de datos**: Guiar en la creación de un proyecto en Python aplicado a minería de datos.
- **Estructuración del código**: Enseñar cómo organizar el código y los datos para un análisis eficiente.
- **Visualización de datos geológicos**: Generar un código en Python que use **Pandas** y **Matplotlib** para procesar y visualizar datos geológicos.

1. Saberes previos:

- Los estudiantes deben tener conocimientos básicos de **Python**, como la creación de variables, uso de operadores, y estructuras de control.
- Es útil que los estudiantes tengan experiencia con bibliotecas de Python como **Pandas** para la manipulación de datos y **Matplotlib** para la visualización gráfica.
- Los estudiantes deben comprender los conceptos básicos de minería de datos, como la obtención y análisis de datos geológicos.

2. Introducción:

En minería, la capacidad de procesar y analizar grandes volúmenes de datos es clave para optimizar los procesos de extracción. Utilizando herramientas como **Python**, podemos organizar los datos de manera eficiente y generar informes que permitan tomar decisiones basadas en datos. En esta sesión, aprenderemos cómo estructurar un proyecto en Python para realizar minería de datos, procesar información geológica y visualizarla de forma clara y comprensible.

3. Desarrollo de la clase:

a) Estructuración de un proyecto en Python:

- **Organización del código**: Utilizar funciones para estructurar el proyecto, creando módulos que manejen tareas específicas como la carga de datos, el procesamiento de datos y la visualización.
- **Uso de Pandas**: Para cargar, procesar y manipular grandes volúmenes de datos geológicos, como profundidades y concentraciones de minerales.
- **Uso de Matplotlib**: Para graficar los datos procesados y mostrar visualmente las concentraciones de minerales a diferentes profundidades o ubicaciones.

b) Ejemplo de proyecto de minería de datos:

Generar un proyecto simple que cargue datos de perforaciones mineras, procese las concentraciones de minerales y grafique los resultados.

Código en Python para procesar y visualizar datos:

python Copiar

```python
import pandas as pd
import matplotlib.pyplot as plt

# Cargar datos geológicos
data = pd.read_csv('datos_geologicos.csv')

# Procesar los datos (por ejemplo, calcular la media de concentración)
media_concentracion = data['concentracion'].mean()

# Visualizar los datos usando un gráfico
plt.figure(figsize=(10, 6))
plt.plot(data['profundidad'], data['concentracion'], label='Concentración de mineral', color='b')
plt.axhline(media_concentracion, color='r', linestyle='--', label=f'Media: {media_concentracion}')
plt.title('Concentración de mineral a diferentes profundidades')
plt.xlabel('Profundidad (m)')
plt.ylabel('Concentración (%)')
plt.legend()
plt.grid(True)
plt.show()

# Generar informe con los resultados
informe = f"Informe de Análisis:\n\nConcentración media de mineral: {media_concentracion}\n"
with open('informe_mineria.txt', 'w') as f:
    f.write(informe)

print("El informe ha sido generado y el gráfico mostrado.")
```

Explicación del código:

1. **Carga de datos**: Se utiliza **Pandas** para leer un archivo CSV con los datos geológicos.
2. **Procesamiento de datos**: Se calcula la media de la concentración de mineral en las perforaciones.
3. **Visualización**: Usando **Matplotlib**, se crea un gráfico que muestra la concentración de mineral a diferentes profundidades, con una línea que indica la media.
4. **Generación de informe**: Se genera un archivo de texto con un resumen de los resultados obtenidos.

4. Aplicaciones en la Ingeniería de Minas:

El análisis y visualización de datos en minería es fundamental para tomar decisiones informadas sobre la extracción de minerales. Con **Python**, los ingenieros pueden:

- **Organizar grandes volúmenes de datos**: Usar **Pandas** para cargar y manipular datos geológicos, asegurando que la información esté lista para el análisis.
- **Visualizar tendencias y patrones**: Crear gráficos con **Matplotlib** para observar cómo varían las concentraciones de minerales a diferentes profundidades o ubicaciones, lo que puede indicar áreas con mayor rentabilidad de extracción.
- **Generar informes automatizados**: Crear proyectos que no solo procesen datos, sino que también generen informes automáticos sobre los hallazgos.

5. Evaluación y reflexión final:

Evaluación:

- Los estudiantes deben usar el código proporcionado para cargar sus propios conjuntos de datos de perforaciones mineras y generar un gráfico similar.
- Los estudiantes deben modificar el código para agregar más análisis, como la desviación estándar de la concentración, y personalizar los gráficos.

Reflexión final:

- **¿Cómo podrías automatizar la actualización de los datos en este proyecto?**
    - Se podría usar **cron jobs** o el **programador de tareas de Windows** para ejecutar el script de Python de manera regular y actualizar los datos automáticamente, de modo que el análisis y los informes se generen sin intervención manual.

Sesión 15: Modelado y Simulación en Python para Minería

Prompt:

```
En minería, se usan modelos matemáticos para predecir el comportamiento de los
yacimientos. ¿Cómo crees que Python podría ayudar en esto?

Explícame cómo se pueden modelar y simular procesos mineros con Python.
```

```
Muéstrame ejemplos donde se simule el comportamiento de un yacimiento mineral
utilizando ecuaciones matemáticas.

Genera un código en Python que simule la variación de concentración de un
mineral en un proceso de lixiviación.

Finalmente, analiza: ¿Qué otras variables podrías incluir para mejorar la
precisión del modelo?
```

Modelado y Simulación en Python para Minería

Objetivos de la sesión:

- **Modelado de procesos mineros**: Explicar cómo Python puede ayudar a modelar y simular procesos mineros, utilizando ecuaciones matemáticas y modelos predictivos.
- **Simulación de comportamiento de yacimientos**: Guiar en la creación de un modelo matemático para simular el comportamiento de un yacimiento mineral.
- **Aplicación a minería**: Generar un código en Python que simule la variación de concentración de un mineral en un proceso de lixiviación.

1. Saberes previos:

- Los estudiantes deben tener conocimientos básicos de **Python**, estructuras de control, y funciones matemáticas.
- Es importante que los estudiantes conozcan conceptos básicos de modelado y simulación matemática, especialmente en el contexto de procesos mineros.
- Deben estar familiarizados con el uso de bibliotecas en Python, como **NumPy** para cálculos matemáticos y **Matplotlib** para la visualización de datos.

2. Introducción:

En minería, la simulación de procesos es fundamental para predecir el comportamiento de los yacimientos minerales a lo largo del tiempo. Usando **Python**, es posible construir modelos matemáticos que simulan estos procesos, lo que ayuda a optimizar las operaciones y la toma de decisiones. En esta sesión, aprenderemos a crear un modelo para simular el comportamiento de un yacimiento mineral, aplicando ecuaciones matemáticas y utilizando herramientas como **NumPy** y **Matplotlib**.

3. Desarrollo de la clase:

a) Modelado y simulación con Python:

- **Ecuaciones matemáticas:** Explicar cómo se utilizan ecuaciones diferenciales y otros modelos matemáticos para representar procesos como la lixiviación o la extracción de minerales.
- **Uso de NumPy**: Aplicar funciones matemáticas en Python para resolver las ecuaciones que describen el proceso minero.
- **Visualización con Matplotlib**: Crear gráficos para visualizar cómo varían los resultados a lo largo del

tiempo o a diferentes profundidades.

b) Ejemplo de simulación de lixiviación: Simular la variación de la concentración de un mineral en un proceso de lixiviación a través de un modelo matemático.

Código en Python para simular un proceso de lixiviación:

```python
import numpy as np
import matplotlib.pyplot as plt

# Parámetros del modelo
tiempo = np.linspace(0, 100, 1000)  # Tiempo en horas
concentracion_inicial = 10  # Concentración inicial del mineral
tasa_lixiviacion = 0.05  # Tasa de lixiviación (reducción de concentración por hora)

# Modelo de lixiviación
concentracion = concentracion_inicial * np.exp(-tasa_lixiviacion * tiempo)

# Visualización de los resultados
plt.figure(figsize=(10,6))
plt.plot(tiempo, concentracion, label='Concentración de mineral', color='b')
plt.title('Simulación de lixiviación de mineral')
plt.xlabel('Tiempo (horas)')
plt.ylabel('Concentración (%)')
plt.legend()
plt.grid(True)
plt.show()
```

Explicación del código:

1. **Parámetros del modelo**: Se define el tiempo, la concentración inicial del mineral y la tasa de lixiviación.
2. **Modelo de lixiviación**: Usamos una función exponencial para simular cómo la concentración disminuye con el tiempo, aplicando la tasa de lixiviación.
3. **Visualización**: Usamos **Matplotlib** para graficar la concentración de mineral a lo largo del tiempo, mostrando cómo la concentración disminuye con el paso de las horas.

4. Aplicaciones en la Ingeniería de Minas:

La simulación de procesos mineros es una herramienta clave en la ingeniería de minas. Algunas aplicaciones incluyen:

- **Simulación de lixiviación**: Modelar cómo los minerales se disuelven durante el proceso de lixiviación, optimizando el uso de reactivos y mejorando la eficiencia de la extracción.

- **Modelado de yacimientos:** Predecir cómo se comportarán los yacimientos minerales a lo largo del tiempo, ayudando a planificar las operaciones de manera eficiente.
- **Simulación de procesos de extracción**: Evaluar diferentes métodos de extracción de minerales y su impacto en la concentración de mineral a lo largo de la operación minera.

5. Evaluación y reflexión final:

Evaluación:

- Los estudiantes deben modificar el código proporcionado para simular un proceso de lixiviación con diferentes tasas de lixiviación y concentraciones iniciales, y generar un gráfico similar.
- Los estudiantes deben reflexionar sobre cómo los cambios en los parámetros del modelo afectan los resultados de la simulación.

Reflexión final:

- **¿Cómo podrías automatizar la actualización de los datos en este proyecto?**
  - Se podría utilizar herramientas como **cron jobs** en sistemas Unix o el **Programador de Tareas** en Windows para ejecutar el modelo de simulación de manera regular, actualizando los datos en tiempo real con la llegada de nuevos datos geológicos.

Sesión 16: Presentación de Proyectos Finales

Prompt:

```
¿Qué elementos debe tener un informe técnico de programación en minería?
Enséñame cómo presentar un proyecto en Python de manera efectiva.
Muestra ejemplos de organización del código y presentación de resultados.
Guíame en la creación de un informe técnico con código y visualizaciones.
Finalmente, responde: ¿Cómo podrías automatizar la generación de reportes en Python?
```

Presentación de Proyectos Finales

Objetivos de la sesión:

- **Presentación de proyectos en Python**: Enseñar a presentar un proyecto en Python de manera efectiva, enfocándose en la organización del código y la presentación de resultados.
- **Informe técnico**: Guiar en la creación de un informe técnico con código y visualizaciones para describir y comunicar los resultados de un proyecto de programación en minería.
- **Automatización de la generación de reportes**: Mostrar cómo automatizar la generación de reportes en Python.

1. Saberes previos:

- Los estudiantes deben tener conocimientos básicos de **Python** y cómo organizar proyectos de programación.
- Es útil que los estudiantes comprendan cómo presentar proyectos de manera clara, destacando los objetivos, el código, los resultados y las conclusiones.
- Familiaridad con el uso de **Matplotlib** y **Pandas** para la visualización de datos y la manipulación de datos, respectivamente.

2. Introducción:

Un proyecto de minería de datos debe presentarse de manera clara y organizada para que los resultados sean comprensibles y útiles para los tomadores de decisiones. En esta sesión, aprenderemos cómo estructurar un proyecto en Python, organizar el código de manera efectiva y generar un informe técnico que incluya tanto el código como las visualizaciones de los resultados obtenidos. Además, veremos cómo automatizar la creación de reportes.

3. Desarrollo de la clase:

a) Organización del código:

- **Estructura del proyecto:** El proyecto debe organizarse en secciones claras, como la importación de bibliotecas, la carga de datos, el procesamiento de datos, la creación de visualizaciones y la generación de informes.
- **Comentarios y documentación:** Utilizar comentarios y documentación adecuada para que otras personas puedan entender fácilmente el flujo del código.

b) Ejemplo de presentación de resultados: Mostrar un ejemplo de cómo organizar el código y presentar los resultados, incluyendo gráficos e informes generados de manera automática.

Código en Python para organizar un proyecto con visualización y reporte:

python Copiar

```python
import pandas as pd
import matplotlib.pyplot as plt

# Cargar y procesar los datos
data = pd.read_csv('datos_mineros.csv')
concentracion_media = data['concentracion'].mean()

# Visualizar los datos
plt.figure(figsize=(10,6))
plt.plot(data['profundidad'], data['concentracion'], label='Concentración de mineral')
plt.title('Concentración de mineral a diferentes profundidades')
```

```
plt.xlabel('Profundidad (m)')
plt.ylabel('Concentración (%)')
plt.legend()
plt.grid(True)
plt.show()

# Generar un informe con los resultados
informe = f"Informe del proyecto:\n\nConcentración media de mineral: {concentracion_media}\n"
with open('informe_mineria.txt', 'w') as f:
    f.write(informe)

print("Informe generado correctamente.")
```

Explicación del código:

1. **Carga de datos**: Se carga un archivo CSV con los datos de perforaciones mineras usando **Pandas**.
2. **Visualización de resultados**: Usamos **Matplotlib** para crear un gráfico que muestra la concentración de mineral a diferentes profundidades.
3. **Generación de informe**: El informe se genera en un archivo de texto con el resumen de la concentración media de mineral.

4. Aplicaciones en la Ingeniería de Minas:

Los proyectos de minería deben presentarse de manera profesional para que los resultados sean útiles para los ingenieros y tomadores de decisiones. Algunas aplicaciones incluyen:

- **Presentación de análisis de datos**: Organizar y mostrar los resultados del análisis de datos de manera clara para tomar decisiones informadas.
- **Generación de reportes automatizados**: Usar Python para crear informes automáticos que resuman los resultados y proporcionen recomendaciones.
- **Visualización de resultados**: Usar gráficos e informes visuales para facilitar la comprensión de los resultados de un análisis complejo.

5. Evaluación y reflexión final:

Evaluación:

- Los estudiantes deben modificar el código proporcionado para adaptar la visualización y el informe a sus propios conjuntos de datos de minería.
- Los estudiantes deben generar un informe técnico que describa el código, los métodos usados y los resultados obtenidos, con sus respectivas visualizaciones.

Reflexión final:

- **¿Cómo podrías automatizar la generación de reportes en Python?**
  - Usar bibliotecas como **Jupyter Notebooks** para crear informes interactivos o **ReportLab** para generar informes PDF. Además, se pueden usar herramientas como **cron jobs** para ejecutar el script de generación de informes automáticamente en intervalos programados.

CONCLUSIONES

En relación con el primer objetivo específico, que consistió en comparar la influencia del uso de ChatGPT en el aprendizaje de los fundamentos de Python, se concluyó que la intervención tecnológica tuvo un impacto significativamente positivo. El grupo experimental mostró una mejora sustancial en su desempeño entre el pretest y el postest, lo que fue corroborado estadísticamente por la prueba de Wilcoxon ($p = 0.000$; $Z = -3.823$) y la prueba U de Mann–Whitney ($p = 0.000$), que evidenciaron diferencias notorias respecto al grupo control. Este resultado confirmó que ChatGPT facilitó la comprensión de conceptos esenciales como sintaxis, estructuras de control, tipos de datos y funciones.

Respecto al segundo objetivo específico, orientado a verificar la influencia del uso de ChatGPT en el aprendizaje de Python aplicado a la programación digital, se determinó que los estudiantes del grupo experimental desarrollaron competencias superiores en el uso práctico del lenguaje. Las pruebas estadísticas aplicadas mostraron valores de significancia de 0.000 (Wilcoxon: $Z = -3.889$), validando el avance en el diseño y ejecución de soluciones de programación orientadas a tareas reales. Esto demuestra que el uso de ChatGPT potenció el razonamiento lógico, la resolución de problemas y la capacidad de aplicación del conocimiento técnico.

En atención al objetivo general, que buscó determinar la influencia del uso de ChatGPT en el aprendizaje global de la programación digital con Python, se concluyó que la implementación de esta herramienta de inteligencia artificial impactó significativamente en el rendimiento académico del grupo experimental. La mejora evidenciada en todas las dimensiones evaluadas, respaldada por pruebas estadísticas sólidas (U de Mann–Whitney y Wilcoxon), permitió rechazar la hipótesis nula y aceptar la hipótesis alternativa. Se verificó que ChatGPT actuó como mediador cognitivo eficaz, promoviendo el aprendizaje autónomo, el acceso inmediato al conocimiento y la consolidación de habilidades clave para la formación en ingeniería.

RECOMENDACIONES

1. Para los docentes universitarios de carreras de ingeniería, se recomienda integrar progresivamente herramientas de inteligencia artificial como ChatGPT en los cursos relacionados con programación digital. Su uso permite personalizar el proceso de enseñanza-aprendizaje, ofrecer retroalimentación inmediata y favorecer el desarrollo autónomo de los estudiantes, lo cual resulta clave para fortalecer competencias técnicas en entornos académicos exigentes.

2. A las autoridades académicas de la Universidad Nacional de San Cristóbal de Huamanga, se sugiere promover políticas institucionales que impulsen la capacitación docente en el uso pedagógico de herramientas de inteligencia artificial. Esto contribuirá a una implementación efectiva, ética y metodológicamente sólida, maximizando los beneficios tecnológicos en la formación profesional de los estudiantes.

3. Para los diseñadores curriculares de programas de ingeniería, se plantea la necesidad de revisar y actualizar los sílabos y contenidos, incorporando el uso de plataformas de inteligencia artificial como apoyo al aprendizaje de lenguajes de programación como Python. Esto permitirá una alineación más estrecha entre la formación académica y las demandas del entorno tecnológico actual.

4. A los estudiantes de ingeniería, se les recomienda aprovechar el potencial de ChatGPT como herramienta de consulta, práctica y verificación de código, fomentando así el aprendizaje autónomo y la autoevaluación constante. No obstante, su uso debe ser crítico y complementario, evitando una dependencia pasiva que limite el desarrollo del pensamiento lógico y analítico.

5. Para futuras investigaciones, se sugiere replicar este estudio en otras disciplinas técnicas y científicas, como diseño estructural, análisis de datos o electrónica, con el fin de explorar el alcance del impacto de la inteligencia artificial en distintos contextos educativos. Asimismo, se recomienda emplear diseños longitudinales que permitan evaluar los efectos sostenidos del uso de ChatGPT en el rendimiento académico a mediano y largo plazo.

REFERENCIAS BIBLIOGRÁFICAS

Althobaiti, M. (2021). *Implications of Artificial Intelligence for Mathematics Education: Insights from the United Arab Emirates*. International Journal of Technology in Education and Science.

Ausubel, D. P. (1978). *In defense of advance organizers:* A reply to the critics. Review of Educational Research, 48(2), 251-257.

Azhar, S. (2011). Building Information Modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. Leadership and Management in Engineering, 11(3), 241-252.

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? 🦜. *Proceedings of the 2021 ACM FAccT Conference*, 45-58.

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. arXiv preprint arXiv:2104.08667.

Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. Advances in neural information processing systems, 29.

Bonn, M. A., Fyall, A., & Cho, M. (2007). *Service quality and satisfaction: The moderating influences of CAD usage*. Tourism Management, 28(2), 82-99.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Bruner, J. S. (1975). *The ontogenesis of speech acts*. Journal of Child Language, 2(1), 1-40.

Campbell, D. T., & Stanley, J. C. (1966). *Experimental and quasi-experimental designs for research*. Rand McNally.

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches.* SAGE Publications.

Cronbach, L. J. (1951). *Coefficient alpha and the internal structure of tests*. Psychometrika, 16(3), 297-334.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dewey, J. (1938). *Logic: The theory of inquiry*. Holt, Rinehart and Winston.

Downey, A. (2015). *Think Python: How to think like a computer scientist* (2nd ed.). O'Reilly Media.

Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). *Engineering design thinking, teaching, and learning*. Journal of Engineering Education, 94(1), 103-120.

Eastman, C., Teicholz, P., & Sacks, R. (2011). BIM handbook: A guide to building information modeling for owners, managers, designers, engineers, and contractors. John Wiley & Sons.

Field, A. (2013). *Discovering Statistics Using SPSS* (4th ed.). SAGE Publications Ltd.

Gall, M. D., Gall, J. P., & Borg, W. R. (2007). *Educational research: An introduction* (8th ed.). Allyn & Bacon.

Gao, Wang y Xie. (2021). *Evaluating the Impact of Artificial Intelligence on Mathematics Learning in High School Students*. Journal of Educational Computing Research.

García, A., & Espinoza, M. (2021). Enhancing learning autonomy in technology-enhanced language learning environments: The impact of artificial intelligence tools. *Journal of Computer Assisted Learning, 37*(4), 940-952.

González, J. (2020). Uso de la inteligencia artificial en la evaluación de aprendizajes en la enseñanza de la matemática. Revista Digital de Investigación en Docencia Universitaria.

Guo, P. J. (2013). Online Python Tutor: Embeddable web-based program visualization for CS education. *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 579-584.

Henderson, K. (1997). *On Line and On Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering*. MIT Press.

Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación*. México DF: McGraw-Hill.

Holovaty, A., & Kaplan-Moss, J. (2009). *The Definitive Guide to Django: Web Development Done Right* (2nd ed.). Apress.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Hussain, Z. (2019). "Artificial Intelligence in Education: Challenges and Opportunities". *Technology in Education*, 24(2), p. 47.

Hustrulid, W. (2013). *SME Mining Engineering Handbook*. Society for Mining, Metallurgy, and Exploration.

ISO (2017). Technical drawings — General principles of presentation — Part 1: Introduction and index. International Organization for Standardization.

James, W. (1907). *Pragmatism: A new name for some old ways of thinking*. Longmans, Green, and Co.

Johnson, L., Becker, S. A., Estrada, V., & Freeman, A. (2014). *The NMC horizon report: 2014 higher education edition*. Austin, Texas: The New Media Consortium.

Jonassen, D. H. (2003). *Learning to solve problems: An instructional design guide*. Jossey-Bass.

Kerlinger, F. N., & Lee, H. B. (2002). *Investigación del comportamiento: Métodos de investigación en ciencias sociales. México*: McGraw-Hill.

Kukulska-Hulme, A., & Traxler, J. (2021). Mobile learning and artificial intelligence: Towards a new pedagogic frontier. *Journal of Computer Assisted Learning, 37*(1), 1-14.

Lamit, L. (2014). *SolidWorks 2014 and engineering graphics*: An integrated approach. Upper Saddle River, NJ: Pearson.

Lee, C., Kim, T., & Lee, S. (2022). Personalized learning environments using artificial intelligence: A systematic review of recent research. *Artificial Intelligence Review, 55*(2), 1503-1532.

Lynn, M. R. (1986). *Determination and quantification of content validity*. Nursing research, 35(6), 382-386.

Makransky, G., Lilleholt, L., & Aaby, A. (2018). *Development and validation of a learning module using an adaptive e-learning environment in CAD*. Computers & Education, 117, 73-91.

Maling, D. H. (1992). Coordinate systems and map projections. Pergamon Press.

Mayer, R. E. (2019). *Multimedia learning* (3rd ed.). Cambridge University Press.

McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2nd ed.). O'Reilly Media.

Muehrcke, P. C., & Muehrcke, J. O. (2016). Map use: reading, analysis, interpretation. JP Publications.

Nunnally, J. C., & Bernstein, I. H. (1994). *Psychometric theory* (3rd ed.). McGraw-Hill.

OpenAI. (2020). OpenAI Charter. https://openai.com/charter/

Ozsariyildiz, S., & Tolman, F. (1998). *CAD tools on the architect's desktop*. Automation in Construction, 7(3), 257-268.

Pallant, J. (2016). SPSS Survival Manual: *A Step by Step Guide to Data Analysis Using SPSS for Windows* (6th ed.). Open University Press.

Pane, J. F., et al. (2017). "Effects of the personalized learning program on achievement outcomes". *Educational Evaluation and Policy Analysis*, 39(3), p. 89.

Papert, S., & Harel, I. (1991). *Constructionism*. Ablex Publishing Corporation.

Peirce, C. S. (1878). "How to Make Our Ideas Clear". *Popular Science Monthly*, 12, pp. 286-302.

Perez, F. (2018). Teaching Python: Bridging the gap between novice and software developer. *Computer Science Education, 28*(2), 110-131.

Pettersson, R. (1998). *Information design: An introduction*. John Benjamins Publishing Company.

Popham, W. J. (2013). *Classroom assessment: What teachers need to know.* Pearson.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8).

Resnik, D. B. (2015). *What is ethics in research & why is it important?* National Institute of Environmental Health Sciences.

Rogers, D. F. (2001). *An introduction to NURBS: With historical perspective*. San Diego, CA: Academic Press.

Rojas, A., & Espinoza, B. (2021). Impacto de la inteligencia artificial en la resolución de problemas matemáticos en estudiantes de secundaria en Lima. Revista de Educación.

Rorty, R. (1982). *Consequences of pragmatism*. University of Minnesota Press.

Sampieri, R. H., Collado, C. F., & Lucio, P. B. (2018). *Metodología de la investigación* (7ma ed.). McGraw-Hill.

Schnabel, M. A., & Kvan, T. (2003). Spatial understanding in design education. International Journal of Architectural Computing, 1(1), 116-128.

Selwyn, N. (2016). "Education and technology: Key issues and debates". Bloomsbury Publishing, p. 104.

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). Experimental and quasi-experimental designs for generalized causal inference. Wadsworth Cengage learning.

Sharma, N. y Kumar, N. (2021). Teachers´ attitudes toward the use or artificial intelligence in education in India: An empirical study. International Journal of Educational Technology in Higher Education.

Shavelson, R. J. (2006). *On the integration of formative assessment in teaching and learning with implications for teacher education.* Stanford Educational Assessment Laboratory and the University of Oslo.

Smith, J. (2020). Aplicación de la inteligencia artificial en la enseñanza de cálculo en la universidad en Estados Unidos. Revista de Investigación Educativa.

Smith, M., & Johnson, L. (2020). AI and the future of learning: Ensuring the quality of responses from intelligent systems. *Educational Technology Research and Development, 68*(3), 1431-1448.

Snyder, J. P. (1987). Map projections—A working manual. U.S. Geological Survey Professional Paper 1395.

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. International Journal of Medical Education, 2, 53-55.

Torres, J. (2020). Efecto de la inteligencia artificial en el rendimiento académico en matemáticas en estudiantes de educación secundaria en el Perú. Revista de Educación.

Valdivia, J., & Cabrera, L. (2021). *Uso de inteligencia artificial para mejorar el aprendizaje matemáticas en estudiantes universitarios*. Revista de Tecnología e Innovación Educativa.

Van Teijlingen, E., & Hundley, V. (2001). The importance of pilot studies. Social research update, 35(1), 1-4.

Vargas, J. (2019). *Aplicación de la inteligencia artificial en la resolución de problemas matemáticos en estudiantes universitarios peruanos*. Revista Internacional de investigación en Ciencias Sociales.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 6000-6010).

Wolchover, N. (2021). AI Language Models like GPT-3 Could Change the Face of Modern Education. TechTrends Magazine, 35(2), pp. 50-56.

Zelle, J. (2004). *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending Against Neural Fake News. arXiv preprint arXiv:1905.12616.

Zhang, Y., Chen, P., & Lu, X. (2020). *The rise of AI in education: From passive to active learning*. Educational Technology Research and Development, 68(4), pp. 307-324.

ANEXOS

Anexo 1. Matriz de Consistencia

ChatGPT en aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la UNSCH, 2024.

PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLES E INDICADORES	MÉTODO DE INVESTIGACIÓN
1. PROBLEMA GENERAL ¿En qué medida el uso de ChatGPT influye en el aprendizaje de Programación digital con Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024 -II? **2. PROBLEMAS ESPECÍFICOS** a. ¿En qué medida el uso de ChatGPT influye en el aprendizaje de fundamentos de Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II? b. ¿En qué medida el uso de ChatGPT influye en el aprendizaje de Python aplicado a la programación digital en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II?	**1. OBJETIVO GENERAL** Determinar la influencia del uso de ChatGPT en el aprendizaje de Programación Digital con Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II. **2. OBJETIVOS ESPECÍFICOS** a. Comparar la influencia del uso de ChatGPT en el aprendizaje de fundamentos de Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II. b. Verificar la influencia del uso de ChatGPT en el aprendizaje de Python aplicado a la Programación Digital en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II.	**1. HIPÓTESIS GENERAL** El uso de ChatGPT influye significativamente el aprendizaje de Programación Digital con Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II. **2. HIPÓTESIS ESPECÍFICAS** a. El uso de ChatGPT influye significativamente en el aprendizaje de fundamentos de Python en los estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II. b. El uso de ChatGPT influye significativamente en el aprendizaje de Python aplicado a Programación Digital en los estudiantes de Ingeniería de Minas en la Universidad Nacional de San Cristóbal de Huamanga durante el semestre académico 2024-II.	**1. VARIABLE INDEPENDIENTE** X: ChatGPT DIMENSIONES: X1: Efectividad pedagógica de ChatGPT X2: Integración tecnológica de ChatGPT en el proceso educativo **DIMENSIONES:** Y1: Fundamentos de Python Y2: Python Aplicado a la Programación Digital 2. VARIABLE DEPENDIENTE Y: Aprendizaje de Programación Digital con Python	**1. TIPO DE INVESTIGACIÓN** Aplicativo **2. NIVEL DE INVESTIGACIÓN** Explicativo Método: Hipotético – Deductivo Diseño: Cuasi experimental **3. POBLACIÓN** Estudiantes de Ingeniería de Minas matriculados en el curso de Programación Digital durante el semestre académico 2024-II. **4. MUESTRA** Se trabajó con dos grupos divididos en grupo control y grupo experimental en el curso de Programación Digital **5. TÉCNICAS** · Observación **6. INSTRUMENTOS** · Ficha de observación

Anexo 2. Validación del instrumento

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
Unidad de Posgrado de la Facultad de Ciencias de la Educación
Doctorado en Educación

FICHA DE VALIDACIÓN
INFORME DE OPINIÓN DE JUICIO DE EXPERTO

I. DATOS GENERALES

Título de la investigación: ChatGPT para el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga, 2024

Nombre del instrumento motivo de evaluación: Ficha de observación

Nombre del autor: Amilcar Tacuri Gamboa

II. ASPECTOS DE VALIDACIÓN

Indicadores	Criterios	Deficiente				Baja				Regular				Bueno				Muy Bueno			
		0	6	11	16	21	26	31	36	41	46	51	56	61	66	71	76	81	86	91	96
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1. CLARIDAD	Está formulado con lenguaje propio.																80				
2. OBJETIVIDAD	Esta expresado en conductas observables.																		90		
3. ACTUALIDAD	Adecuado al avance de la ciencia pedagógica.																80				
4. ORGANIZACIÓN	Existe una organización lógica.																		90		
5. SUFICIENCIA	Comprende los aspectos en cantidad y calidad.																		90		
6. INTENCIONALIDAD	Adecuado para valorar los indicadores.																80				
7. CONSISTENCIA	Basado en aspectos teóricos científicos.																		90		
8. COHERENCIA	Entre los ítems e indicadores.																		90		
9. METODOLGÍA	La estrategia responde al propósito de la investigación.																				100
10.PERTINENCIA	Es útil y adecuado para la investigación.																80				

PROMEDIO DE VALORACIÓN: 87

OPINIÓN DE APLICABILIDAD: a) Deficiente b) Baja c) Regular d) Buena (e) Muy Buena

Nombres y Apellidos	Hemerson Lizarbe Alarcón	DNI:	
Título Profesional	Ingeniero Civil		
Grado Académico	Doctor		
Mención	Medio ambiente y desarrollo sostenible		

Lugar y Fecha: Ayacucho 03/09/24

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
Unidad de Posgrado de la Facultad de Ciencias de la Educación
Doctorado en Educación

FICHA DE VALIDACIÓN
INFORME DE OPINIÓN DE JUICIO DE EXPERTO

I. DATOS GENERALES

Título de la investigación: ChatGPT para el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga, 2024

Nombre del instrumento motivo de evaluación: Ficha de observación

Nombre del autor: Amilcar Tacuri Gamboa

II. ASPECTOS DE VALIDACIÓN

Indicadores	Criterios	Deficiente				Baja				Regular				Bueno				Muy Bueno			
		0	6	11	16	21	26	31	36	41	46	51	56	61	66	71	76	81	86	91	96
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	10
1. CLARIDAD	Está formulado con lenguaje propio.																		90		
2. OBJETIVIDAD	Esta expresado en conductas observables.																		90		
3. ACTUALIDAD	Adecuado al avance de la ciencia pedagógica.																80				
4. ORGANIZACIÓN	Existe una organización lógica.																80				
5. SUFICIENCIA	Comprende los aspectos en cantidad y calidad.																	85			
6. INTENCIONALIDAD	Adecuado para valorar los indicadores.																	85			
7. CONSISTENCIA	Basado en aspectos teóricos científicos.																80				
8. COHERENCIA	Entre los ítems e indicadores.																80				
9. METODOLGÍA	La estrategia responde al propósito de la investigación.															75					
10.PERTINENCIA	Es útil y adecuado para la investigación.																80				

PROMEDIO DE VALORACIÓN: 82.5

OPINIÓN DE APLICABILIDAD: a) Deficiente b) Baja c) Regular (d) Buena e) Muy Buena

Nombres y Apellidos	Gualberto, Huillcahuari Huamani	DNI:	41407826
Título Profesional	Ing. Civil		
Grado Académico	Doctor		
Mención	Doctor en Ing. Civil		

Lugar y Fecha: Ayacucho, 14 de Agosto, 2024

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
Unidad de Posgrado de la Facultad de Ciencias de la Educación
Doctorado en Educación

FICHA DE VALIDACIÓN
INFORME DE OPINIÓN DE JUICIO DE EXPERTO

I. DATOS GENERALES

Título de la investigación: ChatGPT para el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga, 2024

Nombre del instrumento motivo de evaluación: Ficha de observación

Nombre del autor: Amilcar Tacuri Gamboa

II. ASPECTOS DE VALIDACIÓN

Indicadores	Criterios	Deficiente				Baja				Regular				Bueno				Muy Bueno			
		0	6	11	16	21	26	31	36	41	46	51	56	61	66	71	76	81	86	91	96
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1. CLARIDAD	Está formulado con lenguaje propio.																80				
2. OBJETIVIDAD	Esta expresado en conductas observables.															75					
3. ACTUALIDAD	Adecuado al avance de la ciencia pedagógica.																		90		
4. ORGANIZACIÓN	Existe una organización lógica.																80				
5. SUFICIENCIA	Comprende los aspectos en cantidad y calidad.																		90		
6. INTENCIONALIDAD	Adecuado para valorar los indicadores.																	85			
7. CONSISTENCIA	Basado en aspectos teóricos científicos.																		90		
8. COHERENCIA	Entre los ítems e indicadores.																	85			
9. METODOLGÍA	La estrategia responde al propósito de la investigación.																		90		
10.PERTINENCIA	Es útil y adecuado para la investigación.																		90		

PROMEDIO DE VALORACIÓN: 85.5

OPINIÓN DE APLICABILIDAD: a) Deficiente b) Baja c) Regular d) Buena (e) Muy Buena

Nombres y Apellidos	Johnny Henrry Ccatamayo Barrios DNI: 20118077
Título Profesional	Seguridad y Control en Mineria
Grado Académico	Doctor
Mención	Seguridad y Mineria

Lugar y Fecha: UNCP Agosto - 2025.

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
ESCUELA PROFESIONAL DE INGENIERÍA DE MINAS
Dr. Johnny Henrry Ccatamayo Barrios

CIP: 142965
Johnnyccatamayo@gmail.

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
Unidad de Posgrado de la Facultad de Ciencias de la Educación
Doctorado en Educación

FICHA DE VALIDACIÓN
INFORME DE OPINIÓN DE JUICIO DE EXPERTO

I. DATOS GENERALES

Título de la investigación: ChatGPT para el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga, 2024

Nombre del instrumento motivo de evaluación: Ficha de observación

Nombre del autor: Amilcar Tacuri Gamboa

II. ASPECTOS DE VALIDACIÓN

Indicadores	Criterios	Deficiente				Baja				Regular				Bueno				Muy Bueno			
		0	6	11	16	21	26	31	36	41	46	51	56	61	66	71	76	81	86	91	96
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1. CLARIDAD	Está formulado con lenguaje propio.															75					
2. OBJETIVIDAD	Esta expresado en conductas observables.															75					
3. ACTUALIDAD	Adecuado al avance de la ciencia pedagógica.															75					
4. ORGANIZACIÓN	Existe una organización lógica.														70						
5. SUFICIENCIA	Comprende los aspectos en cantidad y calidad.														70						
6. INTENCIONALIDAD	Adecuado para valorar los indicadores.														70						
7. CONSISTENCIA	Basado en aspectos teóricos científicos.														70						
8. COHERENCIA	Entre los ítems e indicadores.															75					
9. METODOLGÍA	La estrategia responde al propósito de la investigación.															75					
10.PERTINENCIA	Es útil y adecuado para la investigación.														70						

PROMEDIO DE VALORACIÓN: 72.5

OPINIÓN DE APLICABILIDAD: a) Deficiente b) Baja c) Regular [X] Buena e) Muy Buena

Nombres y Apellidos	Carlos Gavilán Zacs	DNI:	41260040
Título Profesional	Ing. Civil		
Grado Académico	Doctor		
Mención	Doctor en Ingeniería de Minas		

Lugar y Fecha: Ayacucho, 13 Agosto del 2024

Anexo 3. Autorización de la institución

OFICIO DE AUTORIZACIÓN N° 045-2024-UNSCH-FIMGC-DA

Señor:

Mg. Amilcar Tacuri Gamboa

Investigador Principal

Presente. -

De mi consideración:

Mediante la presente, y en mi calidad de Director del Departamento Académico de Ingeniería de Minas y Civil de la Universidad Nacional de San Cristóbal de Huamanga, me dirijo a usted para informarle que, luego de evaluar su solicitud y el protocolo de investigación presentado, se **AUTORIZA** la ejecución de su trabajo de investigación titulado: **"ChatGPT en aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la UNSCH, 2024"**

Dicha autorización comprende el acceso a la población de estudiantes del curso de Programación Digital (Código: IC-386) del semestre académico **2024-II**, así como el uso de las instalaciones y recursos educativos necesarios para la aplicación de los instrumentos y el desarrollo de las sesiones experimentales dentro del marco ético y metodológico propuesto.

La presente autorización es válida por el periodo comprendido entre el **13 de agosto de 2024** y el **20 de diciembre de 2024**, correspondiente al desarrollo del semestre académico 2024-II. Se solicita a los docentes y personal administrativo brindar las facilidades del caso para el normal desarrollo de esta investigación doctoral.

Es grato para esta Dirección contribuir con estudios académicos que promuevan la innovación educativa y el uso de tecnologías emergentes en la formación de nuestros estudiantes de ingeniería.

Atentamente,

Ayacucho, 09 de agosto de 2024



Anexo 4. Procesamiento estadístico

Base de datos

Pre test grupo control

	V.D. Aprendizaje de Programación Digital con Python																						
	D1. Fundamentos de Python										D2. Python Aplicado a la Programación Digital												
ID	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20	D1	D2	VD
E1	1	1	1	1	1	2	1	2	1	1	1	1	2	1	1	2	1	2	1	1	1	1	1
E2	2	1	1	1	2	1	1	2	1	1	1	2	1	1	1	1	1	1	1	2	1	1	1
E3	1	1	1	1	1	1	1	1	1	2	1	1	1	2	1	1	1	1	2	1	1	1	1
E4	1	1	2	1	1	2	1	1	1	1	2	1	1	2	1	1	1	2	1	2	1	1	1
E5	2	1	1	2	1	1	1	2	1	2	1	2	1	1	2	1	2	1	1	1	1	1	1
E6	1	2	1	1	1	2	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1
E7	1	2	1	1	1	1	1	2	1	2	1	1	2	1	2	1	1	2	1	1	1	1	1
E8	1	1	2	1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1
E9	2	1	1	1	2	1	1	2	1	2	1	1	2	1	1	2	1	1	2	2	1	1	1
E10	1	1	1	2	1	2	1	1	1	1	1	2	1	1	2	1	1	2	2	1	1	1	1
E11	1	2	1	2	1	1	1	2	1	2	1	1	2	1	1	1	1	2	1	1	1	1	1
E12	1	1	1	2	1	1	1	2	1	1	1	1	1	1	2	1	2	1	1	1	1	1	1
E13	1	2	1	1	2	1	1	1	2	1	1	1	2	1	1	1	1	2	1	1	1	1	1
E14	1	2	1	1	2	1	1	2	1	1	2	1	1	1	2	1	2	1	2	2	1	2	1
E15	1	1	1	2	1	1	2	1	2	1	1	2	1	1	1	1	1	2	1	2	1	1	1
E16	1	2	1	1	2	1	1	1	1	1	2	1	1	2	1	1	1	2	1	1	1	1	1
E17	2	1	1	2	1	1	2	1	2	1	1	2	1	2	1	2	1	1	2	1	1	1	1
E18	1	2	1	1	1	2	1	2	1	1	2	1	2	1	1	2	1	1	1	2	1	1	1
E19	1	1	1	2	1	1	2	1	1	1	2	1	1	1	2	1	1	2	1	2	1	1	1
E20	2	1	2	1	1	2	1	1	2	1	1	2	1	1	1	2	1	1	2	1	1	1	1

Pre test grupo experimental

	V.D. Aprendizaje de Programación Digital con Python																						
	D1. Fundamentos de Python										D2. Python Aplicado a la Programación Digital												
ID	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20	D1	D2	VD
E1	1	2	1	2	1	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	1	1	1
E2	2	1	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1
E3	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1
E4	1	2	1	1	1	2	1	2	1	1	1	1	2	1	1	2	1	2	1	1	1	1	1
E5	1	2	1	1	2	1	1	2	1	1	2	1	1	1	1	1	1	1	1	2	1	1	1
E6	1	1	1	1	1	2	1	1	1	1	1	1	1	1	2	1	1	1	2	1	1	1	1
E7	1	1	1	2	1	1	2	1	1	2	1	1	2	1	1	1	2	1	1	1	1	1	1
E8	1	1	2	1	1	2	1	1	1	1	1	2	1	1	2	1	1	1	2	2	1	1	1
E9	2	1	1	1	1	1	2	1	1	2	1	1	1	1	2	1	2	1	1	1	1	1	1
E10	1	2	1	1	2	1	1	1	2	1	1	1	1	2	1	2	1	1	1	1	1	1	1
E11	2	1	1	1	1	2	1	1	2	1	2	2	1	1	1	1	2	1	2	1	1	1	1
E12	2	1	1	2	1	1	1	1	2	1	1	2	1	2	1	2	1	2	1	2	1	2	1
E13	1	1	2	1	1	1	2	1	1	1	2	1	1	2	1	1	1	2	1	1	1	1	1
E14	2	1	1	1	2	1	1	2	1	1	2	2	2	2	1	1	1	2	1	1	1	2	1
E15	1	1	2	1	1	2	1	1	1	2	1	1	1	2	1	1	2	1	1	1	1	1	1
E16	1	2	1	1	2	1	1	2	1	2	1	2	1	1	1	1	2	1	1	2	1	1	1
E17	2	1	2	1	1	2	1	1	1	2	1	1	1	2	1	2	1	1	2	1	1	1	1
E18	2	1	2	1	1	2	1	2	1	1	2	1	2	1	1	2	1	1	2	1	1	1	1
E19	1	2	1	1	2	1	1	1	1	2	1	1	2	1	2	1	1	2	1	2	1	1	1
E20	1	2	1	1	2	1	1	2	1	2	1	1	1	2	1	1	2	1	2	1	1	1	1

Post test grupo control

	V.D. Aprendizaje de Programación Digital con Python																						
	D1. Fundamentos de Python										D2. Python Aplicado a la Programación Digital												
ID	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20	D1	D2	VD
E1	2	2	2	2	1	4	1	2	4	2	2	2	3	2	2	4	2	4	2	2	2	3	2
E2	3	2	2	2	2	2	2	3	2	1	2	3	2	2	2	2	2	3	2	3	2	2	2
E3	2	2	1	2	1	4	2	1	3	2	4	1	2	2	2	4	2	1	3	3	2	2	2
E4	1	2	3	2	2	3	2	2	3	3	3	2	3	2	4	3	2	4	2	3	2	3	3
E5	3	2	2	4	3	2	2	3	2	4	3	2	3	3	2	2	3	2	2	2	3	2	3
E6	2	2	2	3	2	2	2	2	3	2	2	2	2	3	1	2	3	3	2	1	2	2	2
E7	2	3	2	3	2	2	2	4	2	4	2	3	2	3	2	3	3	2	2	3	3	3	3
E8	2	4	2	3	2	3	2	3	2	2	2	4	2	2	2	4	2	3	2	3	3	3	3
E9	3	2	2	3	2	2	3	2	3	2	2	3	2	1	2	3	2	2	3	3	2	2	2
E10	2	3	2	3	2	2	3	2	3	2	3	2	3	2	3	2	2	3	2	3	2	3	2
E11	2	3	2	3	2	2	2	3	2	3	2	2	3	2	3	2	3	2	2	3	2	2	2
E12	2	2	2	3	2	4	2	3	2	2	2	2	2	2	3	2	3	4	2	2	2	2	2
E13	2	3	2	3	2	3	2	3	2	2	2	2	3	2	4	3	2	3	2	2	2	3	2
E14	2	3	2	3	2	3	2	3	2	3	3	2	3	3	2	3	3	3	2	3	3	3	3
E15	2	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	2	3	2
E16	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	2	3	2	3	2	3	2	2
E17	3	2	3	2	2	3	2	3	2	3	2	3	2	3	3	3	3	2	2	3	3	3	3
E18	2	3	2	3	2	3	3	3	2	3	3	3	3	2	2	3	2	2	2	3	3	3	3
E19	2	2	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
E20	2	3	3	2	2	4	2	2	3	2	3	2	3	2	2	3	2	2	3	3	3	3	3

Post test grupo experimental

	V.D. Aprendizaje de Programación Digital con Python																						
	D1. Fundamentos de Python										D2. Python Aplicado a la Programación Digital												
ID	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20	D1	D2	VD
E1	2	4	2	4	2	3	4	3	4	4	3	3	2	2	2	2	3	2	3	3	3	3	3
E2	4	2	4	2	4	4	4	4	4	4	3	3	2	2	4	2	3	2	2	3	4	3	3
E3	2	3	4	3	2	3	3	2	2	2	3	3	2	3	2	3	3	2	4	3	3	3	3
E4	4	4	3	3	3	4	3	4	3	4	3	3	4	3	3	4	3	4	3	3	4	3	3
E5	3	3	4	4	3	3	3	4	4	3	3	3	3	3	3	2	2	3	4	4	3	3	3
E6	4	4	4	3	3	4	4	4	3	4	4	4	3	4	4	3	4	4	4	4	4	4	4
E7	3	3	3	4	3	3	4	3	3	4	3	3	4	3	3	3	4	3	2	4	3	3	3
E8	3	3	4	3	3	4	3	3	3	3	4	4	3	3	4	3	3	3	4	4	3	4	3
E9	4	4	3	3	3	3	4	3	4	4	3	2	4	3	4	3	4	2	3	4	4	3	3
E10	4	4	3	3	4	4	4	4	4	4	3	4	3	4	4	4	3	4	3	4	4	4	4
E11	4	3	3	4	3	4	2	3	4	2	4	3	4	4	2	2	3	4	2	3	3	3	3
E12	4	3	2	4	3	3	2	3	4	3	4	3	3	4	4	3	4	4	4	4	3	4	3
E13	4	4	4	3	4	4	3	3	3	4	3	3	4	3	3	4	3	3	2	3	4	3	3
E14	4	4	4	3	4	4	4	3	4	4	4	4	4	4	4	3	4	4	3	3	4	4	4
E15	4	4	4	4	2	4	3	2	4	4	3	3	3	4	4	4	2	2	3	3	4	3	3
E16	2	3	3	3	4	3	2	4	3	4	3	4	3	2	3	3	4	3	3	4	3	3	3
E17	4	4	3	4	3	3	4	3	3	3	2	3	3	4	2	3	3	4	3	2	3	3	3
E18	4	2	4	3	4	3	4	3	4	3	4	4	4	3	4	3	4	4	3	4	3	4	4
E19	3	4	3	3	4	3	4	2	3	4	3	3	3	3	4	3	3	4	2	3	3	3	3
E20	4	3	4	3	4	3	4	3	4	4	4	4	4	3	4	3	4	4	3	3	4	4	4

Base de datos prueba piloto

Con el objetivo de evaluar la fiabilidad del instrumento, en este caso la ficha de observación para el aprendizaje de Programación Digital con Python, se diseñó y aplicó un test de evaluación dirigido a los estudiantes de Ingeniería Civil de la serie 200-II durante el semestre académico 2024 –II. Esta selección se fundamentó en que los estudiantes de dicha serie también cursan la asignatura de

Programación Digital, lo que permitió obtener una muestra representativa para validar el instrumento en un contexto académico real.

La base de datos presentada corresponde a la recopilación de respuestas obtenidas en la prueba piloto. Se estructuró en dos dimensiones principales:

- D1. Fundamentos de Python, compuesta por 10 ítems (i1-i10), diseñados para evaluar el conocimiento en sintaxis, tipos de datos, operadores, estructuras de control y modularización del código.

- D2. Python Aplicado a la Programación Digital, integrada por 10 ítems adicionales (i11 – i20), centrados en la aplicación práctica del lenguaje en estructuras avanzadas, manipulación de archivos, uso de bibliotecas y resolución de problemas en el contexto de la ingeniería.

Cada estudiante fue evaluado en función de los criterios establecidos en el pre test, aplicando una escala ordinal que permitió medir el nivel de desempeño en cada ítem. La información recopilada fue organizada en una matriz de datos, facilitando el análisis estadístico para determinar la consistencia interna del instrumento mediante el cálculo del Alfa de Cronbach.

	V.D. Aprendizaje de Programación Digital con Python																			
	D1. Fundamentos de Python									D2. Python Aplicado a la Programación Digital										
ID	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20
E1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1
E2	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	3	1	1
E3	2	1	2	3	2	2	2	1	2	1	1	2	2	2	2	1	2	3	3	3
E4	2	2	2	1	2	1	1	2	1	1	2	1	3	2	3	1	1	1	3	1
E5	2	1	2	1	2	1	1	1	2	2	2	1	1	2	1	1	1	2	2	1
E6	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
E7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
E8	1	1	1	1	1	1	2	1	2	1	1	1	1	1	1	2	1	1	1	1
E9	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	3
E10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
E11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E12	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	2	1	2	2	1
E13	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	1	1
E14	2	2	2	2	2	2	2	3	2	2	2	2	3	2	3	2	2	2	2	1
E15	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1
E16	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	2
E17	2	2	2	1	2	3	1	2	1	2	1	2	3	2	3	2	2	1	2	2
E18	1	1	1	1	1	1	2	1	1	1	2	1	1	1	1	1	1	1	1	1
E19	1	1	1	1	1	2	1	1	2	1	1	1	2	1	1	1	1	1	1	1
E20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1

Prueba de fiabilidad

Y1: Fundamentos de Python

El procesamiento de casos evidenció que la totalidad de los datos recopilados fueron válidos (N =20, 100%), sin exclusiones, lo que garantizó la integridad y complejidad del análisis. La eliminación por lista se basó en todas las variables del procedimiento, asegurando que el cálculo de fiabilidad se realizara sobre una base de datos coherente y sin valores perdidos.

Resumen de		N	%
Casos	Válido	20	100,0
	Excluido[a]	0	,0
	Total	20	100,0

a. La eliminación por lista se

El análisis de fiabilidad reflejó un Alfa de Cronbach de 0.868, lo que indica una alta consistencia interna en la medición de los Fundamentos de Python. Este valor supera ampliamente el umbral mínimo aceptable de 0.70, lo que garantiza que el instrumento utilizado posee sólida estabilidad y precisión en la evaluación de los conocimientos en esta dimensión. La magnitud del coeficiente obtenido permite afirmar que los ítems presentan adecuado grado de homogeneidad, asegurando mediciones confiables y replicables en el contexto del estudio.

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,868	10

El examen de cada ítem permitió verificar que todos los reactivos contribuyeron de manera significativa a la fiabilidad global del instrumento. Se identificó que los ítems ITEM1, ITEM3 y ITEM5 presentan correlaciones corregidas elevadas, lo que evidencia su alienación con la estructura teórica de la escala. Asimismo, los demás ítems reflejaron un comportamiento estable dentro del modelo, manteniendo el Alfa de Cronbach dentro de los valores esperados.

La estabilidad del coeficiente de fiabilidad tras la eliminación individual de cada ítem confirmó la pertinencia del conjunto de reactivos, lo que respalda su capacidad para medir de manera precisa los conocimientos en Fundamentos de Python. Estos resultados ratifican la validez del instrumento y su aplicación en la investigación, asegurando la calidad de los datos obtenidos en la evaluación de los estudiantes.

	Media de escala si el elemento se ha suprimido	Varianza de escala si el elemento se ha suprimido	Correlación total de elementos corregida	Alfa de Cronbach si el elemento se ha suprimido
ITEM1	11,20	7,853	,888	,832
ITEM2	11,25	8,829	,518	,860
ITEM3	11,20	7,853	,888	,832
ITEM4	11,25	8,618	,446	,868
ITEM5	11,20	7,853	,888	,832
ITEM6	11,15	8,134	,552	,860
ITEM7	11,25	9,145	,382	,870
ITEM8	11,15	8,239	,517	,863
ITEM9	11,15	8,976	,377	,871
ITEM10	11,25	8,829	,518	,860

Y2: Python Aplicado a la Programación Digital

El análisis del procesamiento de casos evidenció que la totalidad de los datos recopilados fueron válidos (N =20, 100%), sin exclusiones, lo que garantizó la integridad y representatividad de la muestra. La eliminación por lista se aplicó a todas las variantes del procedimiento, asegurando la consistencia del análisis y la ausencia de valores perdidos.

Resumen de procesamiento de casos

		N	%
Casos	Válido	20	100,0
	Excluido[a]	0	,0
	Total	20	100,0

a. La eliminación por lista se basa en todas las

El coeficiente de Alfa de Cronbach obtenido fue de 0.830, lo que indica una alta consistencia interna dentro de la dimensión Python Aplicado a la Programación Digital. Este valor superó el umbral mínimo de 0.70, asegurando que los ítems evaluados miden de manera homogénea el constructo analizado. La magnitud del coeficiente confirmó la estabilidad y confiabilidad del instrumento, lo que permitió su aplicación efectiva en la investigación.

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,830	10

El análisis de los ítems reflejó una adecuada coherencia con la escala general. Se observó que en los ítems ITEM12, ITEM13, ITEM14 y ITEM15 presentaron correlaciones corregidas elevadas, lo que ratificó su pertinencia dentro de la dimensión, La estabilidad de Alfa de Cronbach tras la eliminación individual de cada ítem confirmó que todos los reactivos contribuyeron de manera consistente a la

medición del constructo, asegurando la validez del instrumento en la evaluación del aprendizaje en Python Aplicado a la Programación Digital.

	Media de escala si el elemento se ha suprimido	Varianza de escala si el elemento se ha suprimido	Correlación total de elementos corregida	Alfa de Cronbach si el elemento se ha suprimido
ITEM11	11,90	12,621	,250	,836
ITEM12	11,95	11,524	,688	,806
ITEM13	11,75	9,461	,772	,784
ITEM14	11,90	10,832	,882	,788
ITEM15	11,80	9,221	,847	,773
ITEM16	11,95	12,471	,334	,830
ITEM17	11,90	11,884	,498	,818
ITEM18	11,75	11,987	,246	,846
ITEM19	11,65	10,239	,657	,799
ITEM20	11,80	12,063	,235	,846

Aprendizaje de Programación Digital con Python

El procesamiento de datos evidenció que la totalidad de los casos fueron válidos (N= 20, 100%) sin exclusiones, lo que garantizó la integridad de la muestra y la ausencia de sesgos por datos faltantes. La eliminación por lista se basó en todas las variables del procedimiento, asegurando que los cálculos de fiabilidad se realizaron sobre un conjunto de respuestas completas. Este resultado permitió realizar un análisis estadístico robusto y confiable del instrumento aplicado.

Resumen de procesamiento de casos

		N	%
Casos	Válido	20	100,0
	Excluido[a]	0	,0
	Total	20	100,0

a. La eliminación por lista se basa en todas las variables del

El análisis de fiabilidad mediante el coeficiente de Alfa de Cronbach reportó un valor de 0.923, lo que refleja un nivel de consistencia interna excelente. Este coeficiente se encuentra por encima del umbral mínimo aceptable de 0.70, y supera incluso el criterio de 0.90, lo que indica que los ítems presentan una alta correlación entre sí y miden de manera uniforme el constructo del aprendizaje de Programación Digital con Python.

Desde una perspectiva metodológica, un Alfa de Cronbach superior a 0.90 sugiere una fuerte homogeneidad entre los ítems, lo que refuerza la validez del instrumento. No obstante, valores muy elevados pueden indicar redundancia entre los ítems, por lo que se recomienda revisar si existen elementos que puedan ser optimizados o fusionados.

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,923	20

En términos generales, los resultados obtenidos evidenciaron que el instrumento alcanzó una fiabilidad excelente ($\alpha = 0.923$), lo que garantiza una alta consistencia interna y respalda su validez y aplicabilidad en la evaluación del aprendizaje de Programación Digital con Python.

	Media de escala si el elemento se ha suprimido	Varianza de escala si el elemento se ha suprimido	Correlación total de elementos corregida	Alfa de Cronbach si el elemento se ha suprimido
ITEM1	24,35	40,766	,914	,913
ITEM2	24,40	42,884	,576	,920
ITEM3	24,35	40,766	,914	,913
ITEM4	24,40	42,042	,565	,920
ITEM5	24,35	40,766	,914	,913
ITEM6	24,30	41,274	,620	,919
ITEM7	24,40	43,832	,395	,923
ITEM8	24,30	41,589	,574	,920
ITEM9	24,30	43,589	,376	,923
ITEM10	24,40	43,411	,475	,921
ITEM11	24,35	44,029	,326	,924
ITEM12	24,40	42,147	,719	,917
ITEM13	24,20	37,958	,820	,913
ITEM14	24,35	40,766	,914	,913
ITEM15	24,25	37,671	,866	,912
ITEM16	24,40	43,726	,415	,922
ITEM17	24,35	42,871	,529	,920
ITEM18	24,20	42,800	,324	,927
ITEM19	24,10	39,884	,666	,918
ITEM20	24,25	43,566	,241	,929

Prueba de hipótesis

Prueba hipótesis general

ANÁLISIS DE LA PRUEBA DE U DE MANN-WHITNEY EN EL PRE TEST DEL GRUPO CONTROL Y EXPERIMENTAL

Para evaluar la homogeneidad inicial entre los grupos control y experimental, se aplicó la prueba de Mann –Whitney, una prueba no paramétrica utilizada para comparar distribuciones cuando los datos no siguen una distribución normal.

Rangos

	Grupo control y experimental	N	Rango promedio	Suma de rangos
Frecuencia de grupo control y experimental	Control	20	20,50	410,00
	Experimental	20	20,50	410,00
	Total	40		

Los datos muestran que ambos grupos presentaron un rango promedio idéntico de 20,50 y una suma de rangos de 410,00 para cada uno. Este resultado sugiere que la distribución de los puntajes iniciales en ambos grupos fue equivalente.

Estadísticos de prueba[a]	Frecuencia de grupo control y experimental
U de Mann-Whitney	200,000
W de Wilcoxon	410,000
Z	,000
Sig. asintótica(bilateral)	1,000
Significación exacta [2*(sig. unilateral)]	1,000[b]
Significación exacta (bilateral)	1,000[c]
Significación exacta (unilateral)	1,000[c]
Probabilidad en el punto	1,000[c]

a. Variable de agrupación: Grupo control y experimental

b. No corregido para empates.

c. Para esta prueba se proporcionan resultados exactos en lugar de Monte Carlo.

El p-valor obtenido (1,000) es mayor que el nivel de significancia convencional de 0,05, lo que indica que no existen diferencias estadísticamente significativas entre los puntajes del pre test en el grupo control y experimental.

Interpretación

Dado que no se encontraron diferencias significativas entre los grupos antes de la intervención, los datos sugieren que ambos grupos tenían condiciones iniciales similares en la variable medida. Esto permite afirmar que la asignación de los participantes en los grupos fue equilibrada y que cualquier variación posterior en los resultados podría atribuirse a la intervención aplicada al grupo experimental.

ANÁLISIS DE LA PRUEBA DE U DE MANN-WHITNEY EN EL POS TEST DEL GRUPO CONTROL Y EXPERIMENTAL

Con el objetivo de evaluar si existían diferencias significativas entre el grupo control y el grupo experimental después de la intervención, se aplicó la prueba de U de Mann – Whitney. Esta prueba no paramétrica permite comparar las distribuciones de dos muestras independientes, particularmente en situaciones donde no se puede asumir la normalidad de los datos.

Rangos				
	Grupo control y experimental	N	Rango promedio	Suma de rangos
Frecuencia de grupo control y experimental	Control	20	13,50	270,00
	Experimental	20	27,50	550,00
	Total	40		

Los resultados obtenidos muestran que el grupo experimental obtuvo un rango promedio de 27,50 y una suma de rangos de 550,00, y el grupo control presentó un rango promedio de 13,50 y una suma de rangos de 270,00. Estos valores sugieren una diferencia marcada en las distribuciones de los puntajes entre ambos grupos.

Estadísticos de prueba[a]

			Frecuencia de grupo control y experimental
U de Mann-Whitney			60,000
W de Wilcoxon			270,000
Z			-4,284
Sig. asintótica(bilateral)			,000
Significación exacta [2*(sig. unilateral)]			,000[b]
Sig. Monte Carlo (bilateral)	Sig.		,000[c]
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003
Sig. Monte Carlo (unilateral)	Sig.		,000[c]
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003

a. Variable de agrupación: Grupo control y experimental
b. No corregido para empates.
c. Se basa en 1000 tablas de muestras con una semilla de inicio 2000000.

El p-valor obtenido (0,000) es menor que el nivel de significancia asumido (0,05), lo que indica que existen diferencias estadísticamente significativas entre los grupos control y experimental en el pos test.

Interpretación

El grupo experimental obtuvo puntuaciones significativas mayores, lo que sugiere que la intervención aplicada tuvo un impacto positivo en el aprendizaje de Programación Digital con Python.

Estos resultados reflejan que el grupo experimental, que utilizó ChatGPT como herramienta pedagógica, logró un mejor desempeño en comparación con el grupo control, lo que respalda la efectividad de la intervención implementada.

ANÁLISIS DE LA PRUEBA DE WILCOXON EN PRE Y POS TEST GRUPO EXPERIMENTAL

El análisis de los rangos de la prueba de Wilcoxon en el grupo experimental evidenció que no se registraron rangos negativos ni empates, lo que indica que ningún estudiante obtuvo desempeño inferior en el pos test respecto al pre test. En contraste, los 20 participantes mostraron rangos positivos con un promedio de 10.50 y una suma de rangos de 210,00, reflejando una mejora consistente en sus puntajes tras la intervención con ChatGPT como herramienta pedagógica en la enseñanza de Programación Digital con Python. Este comportamiento en los rangos sugiere que la intervención tuvo un impacto positivo homogéneo en los estudiantes del grupo experimental.

Rangos

		N	Rango promedio	Suma de rangos
Postest grupo experimental - Pretest grupo experimental	Rangos negativos	0[a]	,00	,00
	Rangos positivos	20[b]	10,50	210,00
	Empates	0[c]		
	Total	20		

a. Postest grupo experimental < Pretest grupo experimental
b. Postest grupo experimental > Pretest grupo experimental
c. Postest grupo experimental = Pretest grupo experimental

En los estadísticos de prueba, el valor Z obtenido fue -4.134, acompañado de un p- valor asintótico bilateral de 0,000 y una significancia de Monte Carlo bilateral y unilateral de 0,000, con un intervalo de confianza de 95% entre 0,000 y 0,003. Estos valores confirmaron que la diferencia entre pre test y pos test en el grupo experimental fue estadísticamente significativa, lo que implica que el uso de ChatGPT generó una mejora considerable en el aprendizaje de Programación Digital con Python. La ausencia de valores negativos y la presencia exclusiva de mejoras en el pos test consolidan la efectividad de la intervención, validando su impacto en el proceso de enseñanza – aprendizaje dentro del grupo experimental.

Estadísticos de prueba[a,c]			
			Postest grupo experimental - Pretest grupo experimental
Z			-4,134[b]
Sig. asintótica(bilateral)			,000
Sig. Monte Carlo (bilateral)	Sig.		,000
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003
Sig. Monte Carlo (unilateral)	Sig.		,000
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003

a. Prueba de rangos con signo de Wilcoxon
b. Se basa en rangos negativos.
c. Se basa en 1000 tablas de muestras con una semilla de inicio 2000000.

Prueba hipótesis específica 1

ANÁLISIS DE LA PRUEBA DE U DE MANN-WHITNEY EN EL PRE TEST DEL GRUPO CONTROL Y EXPERIMENTAL

El análisis de la Prueba de U de Mann – Whitney en el pre test de los grupos control y experimental permitió evaluar si ambos grupos eran homogéneos antes de la intervención con ChatGPT, en relación con la Hipótesis Específica 1. Los resultados indicaron que los rangos promedio fueron idénticos para ambos grupos (20,50), con una suma de rangos de 410,00 para cada uno. Esto refleja que no existían diferencias en los niveles iniciales de aprendizaje de Fundamentos de Python entre los estudiantes de ambos grupos.

Rangos				
	Grupo control y experimental	N	Rango promedio	Suma de rangos
Frecuencia de grupo control y experimental	Control	20	20,50	410,00
	Experimental	20	20,50	410,00
	Total	40		

En los estadígrafos de prueba, el valor U de Mann – Whitney fue 200,00 y el W de Wilcoxon, 410,00, con un Z de 0,000. Además, el p- valor asintótico bilateral fue 1,000, al igual que la significación exacta bilateral y unilateral (1,000), lo que confirma que no hubo diferencias estadísticamente significativas entre los grupos en el pre test. Estos hallazgos validaron la homogeneidad de los grupos antes de la intervención, asegurando que ninguno tenía ventaja inicial sobre el otro en términos de conocimientos previos de Fundamentos de Python. Este resultado permitió establecer un punto de partida equitativo para medir de manera objetiva el impacto de la intervención basada en ChatGPT en el grupo experimental.

Estadísticos de prueba[a]	Frecuencia de grupo control y experimental
U de Mann-Whitney	200,000
W de Wilcoxon	410,000
Z	,000
Sig. asintótica(bilateral)	1,000
Significación exacta [2*(sig. unilateral)]	1,000[b]
Significación exacta (bilateral)	1,000[c]
Significación exacta (unilateral)	1,000[c]
Probabilidad en el punto	1,000[c]

a. Variable de agrupación: Grupo control y experimental

b. No corregido para empates.

c. Para esta prueba se proporcionan resultados exactos en lugar de Monte Carlo.

ANÁLISIS DE LA PRUEBA DE U DE MANN-WHITNEY EN EL POS TEST DEL GRUPO CONTROL Y EXPERIMENTAL

El análisis de la prueba de U de Mann – Whitney en el pos test de los grupos control y experimental correspondiente a la Hipótesis Específica 1, permitió evaluar si la intervención con ChatGPT generó una mejora significativa en el aprendizaje de Fundamentos de Python en los estudiantes de Ingeniería de Minas. Los resultados reflejaron una diferencia clara entre los grupos, con un rango promedio de 12,70 para el grupo control y 28,30 para el grupo experimental, lo que evidencia una mayor puntuación en el grupo que utilizó ChatGPT.

Rangos

	Grupo control y experimental	N	Rango promedio	Suma de rangos
Frecuencia de grupo control y experimental	Control	20	12,70	254,00
	Experimental	20	28,30	566,00
	Total	40		

En los estadísticos de prueba, el valor de U de Mann – Whitney fue 44,000, el W de Wilcoxon fue 254,000, y el Z resultó -4,564. El p- valor asintótico bilateral de 0,000, la ser menor a 0,05, indica una diferencia estadísticamente significativa entre los grupos después de la intervención. La significación exacta y la significación Monte Carlo también resultaron en 0,000, con un intervalo de confianza del 95% entre 0,000 y 0,003, lo que confirma la robustez del hallazgo. Estos resultados demostraron que el uso de ChatGPT en el grupo experimental tuvo un impacto positivo en el aprendizaje de Fundamentos de Python, validando la hipótesis de que esta herramienta tecnológica contribuyó significativamente al desarrollo de habilidades en los estudiantes.

Estadísticos de prueba[a]			
			Frecuencia de grupo control y experimental
U de Mann-Whitney			44,000
W de Wilcoxon			254,000
Z			-4,564
Sig. asintótica(bilateral)			,000
Significación exacta [2*(sig. unilateral)]			,000[b]
Sig. Monte Carlo (bilateral)	Sig.		,000[c]
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003
Sig. Monte Carlo (unilateral)	Sig.		,000[c]
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003

a. Variable de agrupación: Grupo control y experimental
b. No corregido para empates.
c. Se basa en 1000 tablas de muestras con una semilla de inicio 2000000.

ANÁLISIS DE LA PRUEBA DE WILCOXON EN PRE Y POS TEST GRUPO EXPERIMENTAL

El análisis de los rangos de la prueba de Wilcoxon aplicada al grupo experimental permitió evaluar la variación en el aprendizaje de Fundamentos de Python antes y después de la intervención con ChatGPT, en relación con la Hipótesis Específica 1. Los resultados evidenciaron que no se registraron rangos negativos no empates, lo que indica que ningún estudiante presentó desempeño inferior en el pos test respecto al pre test. En contraste, los 20 participantes mostraron rangos positivos con un promedio de 10.50 y una suma de rangos de 210.00, lo que refleja una mejora uniforme en los puntajes luego de la implementación de la estrategia pedagógica basada en ChatGPT. Este comportamiento en los rangos sugiere un efecto positivo generalizado de la intervención en el grupo experimental.

Rangos				
		N	Rango promedio	Suma de rangos
Postest grupo experimental - Pretest grupo experimental	Rangos negativos	0[a]	,00	,00
	Rangos positivos	20[b]	10,50	210,00
	Empates	0[c]		
	Total	20		

a. Postest grupo experimental < Pretest grupo experimental
b. Postest grupo experimental > Pretest grupo experimental
c. Postest grupo experimental = Pretest grupo experimental

En los estadígrafos de prueba, El valor Z obtenido fue -4.041, acompañado de un p-valor asintótico bilateral de 0.000 y una significación de Monte Carlo bilateral y unilateral de 0,000, con un intervalo de confianza del 95% entre 0,000 y 0,003. Estos valores confirmaron que la diferencia entre pre test y pos test en el grupo experimental fue estadísticamente significativa, lo que respalda la Hipótesis Específica 1, en la que se plantea que el uso de ChatGPT influye significativamente en el aprendizaje de Fundamentos de Python. La ausencia de valores negativos y la presencia exclusiva de mejoras en el pos test consolidan la efectividad de la intervención, evidenciando un impacto positivo en el desarrollo de habilidades iniciales en Python dentro del grupo experimental.

Estadísticos de prueba[a,c]			
			Postest grupo experimental - Pretest grupo experimental
Z			-4,041[b]
Sig. asintótica(bilateral)			,000
Sig. Monte Carlo (bilateral)	Sig.		,000
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003
Sig. Monte Carlo (unilateral)	Sig.		,000
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003

a. Prueba de rangos con signo de Wilcoxon
b. Se basa en rangos negativos.
c. Se basa en 1000 tablas de muestras con una semilla de inicio 2000000.

Prueba hipótesis específica 2

ANÁLISIS DE LA PRUEBA DE U DE MANN-WHITNEY EN EL PRE TEST DEL GRUPO CONTROL Y EXPERIMENTAL

El análisis de la prueba de U de Mann – Whitney en el pre test de los grupos control y experimental, correspondiente a la Hipótesis Específica 2, permitió evaluar si ambos grupos presentan condiciones homogéneas antes de la intervención con ChatGPT en relación con el aprendizaje de Python aplicado a la Programación Digital. Los resultados indicaron que los rangos promedio fueron 20,00 para el grupo control y 21,00 para el grupo experimental, con una suma de rangos de 400,00 y 420,00, respectivamente.

Rangos				
	Grupo control y experimental	N	Rango promedio	Suma de rangos
Frecuencia de grupo control y experimental	Control	20	20,00	400,00
	Experimental	20	21,00	420,00
	Total	40		

En los estadígrafos de prueba, el valor de U de Mann – Whitney fue 190,00, mientras que el W de Wilcoxon fue de 400,000 y el valor Z de -0.593. El p- valor asintótico bilateral fue 0,553, mayor a 0,05, lo que indica la ausencia de diferencias estadísticamente significativas entre los grupos. Asimismo, la significación exacta bilateral fue 0,799, y la significación Monte Carlo bilateral fue 1,000, con un intervalo de confianza del 95% entre 0,997 y 1,000. Estos resultados confirmaron que no hubo diferencias iniciales en el nivel de Python aplicado a la Programación Digital, lo que aseguró que la comparación posterior entre los grupos reflejara exclusivamente el efecto de la intervención con ChatGPT en el grupo experimental.

Estadísticos de prueba[a]			
			Frecuencia de grupo control y experimental
U de Mann-Whitney			190,000
W de Wilcoxon			400,000
Z			-,593
Sig. asintótica(bilateral)			,553
Significación exacta [2*(sig. unilateral)]			,799[b]
Sig. Monte Carlo (bilateral)	Sig.		1,000[c]
	Intervalo de confianza al 95%	Límite inferior	,997
		Límite superior	1,000
Sig. Monte Carlo (unilateral)	Sig.		,527[c]
	Intervalo de confianza al 95%	Límite inferior	,496
		Límite superior	,558

a. Variable de agrupación: Grupo control y experimental
b. No corregido para empates.
c. Se basa en 1000 tablas de muestras con una semilla de inicio 2000000.

ANÁLISIS DE LA PRUEBA DE U DE MANN-WHITNEY EN EL POS TEST DEL GRUPO CONTROL Y EXPERIMENTAL

El análisis de la prueba de U de Mann – Whitney en el pos test de los grupos control y experimental, correspondiente a la Hipótesis Específica 2, permitió evaluar el impacto del uso de ChatGPT en el aprendizaje de Python aplicado a la Programación Digital. Los resultados mostraron una diferencia notable entre los grupos, donde el rango promedio del grupo control fue de 14,40, mientras que el grupo experimental obtuvo un rango promedio de 26,60, lo que indica un mejor desempeño en los estudiantes que utilizaron ChatGPT como herramienta de aprendizaje.

Rangos				
	Grupo control y experimental	N	Rango promedio	Suma de rangos
Frecuencia de grupo control y experimental	Control	20	14,40	288,00
	Experimental	20	26,60	532,00
	Total	40		

En los estadísticos de prueba, el valor de U de Mann – Whitney fue 78,000, el W de Wilcoxon 288,000 y el Z de -3,829. El p-valor asintótico bilateral de 0,000, junto con la asignación exacta de 0,001 y la significación de Monte Carlo de 0,000, confirmaron que la diferencia entre ambos grupos fue estadísticamente significativa. Además, el intervalo de confianza del 95% entre 0,000 y 0,003 respalda la robustez de los hallazgos. En conclusión, estos resultados validaron la hipótesis de que ChatGPT influyó significativamente en el aprendizaje de Python aplicado a la Programación Digital, demostrando su efectividad como recurso educativo en este contexto.

Estadísticos de prueba[a]

			Frecuencia de grupo control y experimental
U de Mann-Whitney			78,000
W de Wilcoxon			288,000
Z			-3,829
Sig. asintótica(bilateral)			,000
Significación exacta [2*(sig. unilateral)]			,001[b]
Sig. Monte Carlo (bilateral)	Sig.		,000[c]
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003
Sig. Monte Carlo (unilateral)	Sig.		,000[c]
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003

a. Variable de agrupación: Grupo control y experimental
b. No corregido para empates.
c. Se basa en 1000 tablas de muestras con una semilla de inicio 299883525.

ANÁLISIS DE LA PRUEBA DE WILCOXON EN PRE Y POS TEST GRUPO EXPERIMENTAL

El análisis de los rangos de la prueba de Wilcoxon aplicada al grupo experimental permitió evaluar la variación en el aprendizaje de Python aplicado a la Programación Digital antes y después de la intervención con ChatGPT, en relación con la Hipótesis Específica 2. Los resultados indicaron que no se registraron rangos negativos ni empates, lo que demuestra que ningún estudiante obtuvo un desempeño inferior en el pos test en comparación con el pre test. En cambio, los 20 participantes presentaron rangos positivos con un promedio de 10,50 y una suma de rangos de 210,00, lo que evidencia una mejora uniforme en los puntajes luego de la implementación de la estrategia pedagógica basada en ChatGPT. Este comportamiento en los rangos refleja que la intervención tuvo efecto positivo homogéneo en el grupo experimental.

Rangos

		N	Rango promedio	Suma de rangos
Postest grupo experimental - Pretest grupo experimental	Rangos negativos	0[a]	,00	,00
	Rangos positivos	20[b]	10,50	210,00
	Empates	0[c]		
	Total	20		

a. Postest grupo experimental < Pretest grupo experimental
b. Postest grupo experimental > Pretest grupo experimental
c. Postest grupo experimental = Pretest grupo experimental

En los estadígrafos de prueba, el valor Z obtenido fue -4,134, acompañado de p- valor asintótico bilateral de 0,000 y una significación de Monte Carlo bilateral y unilateral de 0,000, con intervalo de confianza de 95% entre 0,000 y 0,003. Estos resultados confirmaron que la diferencia entre pre test y pos test en el grupo experimental fue estadísticamente significativa, lo que respalda la Hipótesis Específica 2, que plantea que el uso de ChatGPT influye significativamente en el aprendizaje de Python aplicado a la Programación Digital. La ausencia de valores negativos y la presencia exclusiva de mejoras en el pos test consolidan la efectividad de la intervención, evidenciando un impacto positivo en la capacidad de los estudiantes para aplicar Python en contextos de Programación Digital dentro del grupo experimental.

Estadísticos de prueba[a,c]

			Postest grupo experimental - Pretest grupo experimental
Z			-4,134[b]
Sig. asintótica(bilateral)			,000
Sig. Monte Carlo (bilateral)	Sig.		,000
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003
Sig. Monte Carlo (unilateral)	Sig.		,000
	Intervalo de confianza al 95%	Límite inferior	,000
		Límite superior	,003

a. Prueba de rangos con signo de Wilcoxon
b. Se basa en rangos negativos.
c. Se basa en 1000 tablas de muestras con una semilla de inicio 299883525.

Anexo 5. Sílabos

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL
DEPARTAMENTO ACADÉMICO DE INGENIERÍA DE MINAS Y CIVIL
Av. Independencia s/n Huamanga. Telf. 066-527226, 066-312510-Anexo-168
Ayacucho - Perú

SÍLABO DE LA ASIGNATURA DE PROGRAMACIÓN DIGITAL (IC-386)

I. INFORMACIÓN GENERAL

1.1.	Facultad	: Ingeniería de Minas, Geología y Civil
1.2.	Departamento	: Ingeniería de Minas y Civil
1.3.	Escuela Profesional	: Ingeniería de Minas
1.4.	Plan de estudios	: 2018
1.5.	Ciclo académico	: 2024-II
1.6.	Duración	: 16 semanas
1.7.	Asignatura	: Programación Digital
1.8.	Código	: IC-386
1.9.	Créditos	: 3.0
1.10.	Naturaleza	: Obligatoria
1.11.	Número de horas semanales	: 04 (02h teoría, 02 horas práctica)
1.12.	Lugar	: S-201
1.13.	Docente responsable	: MSc. Amilcar Tacuri Gamboa

I. SUMILLA

El curso es teórico – práctico, en el cual se desarrollarán en dos unidades: I. Fundamentos de Python, los estudiantes adquirirán conocimientos esenciales sobre la sintaxis de Python, tipos de datos, estructuras de control, manejo de funciones y conceptos clave de programación estructurada. Se enfatizará en la resolución de problemas mediante la escritura de algoritmos eficientes y modulares. II. Python Aplicado a la Programación Digital, se aplicarán estos conocimientos en el desarrollo de soluciones informáticas más avanzadas, abordando estructuras de datos, programación orientada a objetos, manipulación de archivos, bases de datos y visualización de datos. Se incluirán prácticas aplicadas a la ingeniería de minas, permitiendo a los estudiantes implementar herramientas digitales para el procesamiento y análisis de datos mineros. Al finalizar el curso, los estudiantes estarán

preparados para diseñar, optimizar y desarrollar programas eficientes con Python, integrando tecnologías y metodologías computacionales aplicadas a su campo profesional.

II. OBJETIVO

2.1. OBJETIVO GENERAL

Desarrollar en el estudiante las competencias para el diseño, implementación y optimización de algoritmos en Python, con aplicaciones en el análisis y gestión de datos en la Ingeniería de Minas.

2.2. OBJETIVOS ESPECÍFICOS

- Comprender los conceptos fundamentales de programación estructurada y modular en Python.
- Aplicar estructuras de datos y algoritmos eficientes en la resolución de problemas computacionales.
- Implementar modelos matemáticos y algoritmos para el análisis de datos mineros.
- Utilizar herramientas de programación digital para la toma de decisiones en proyectos de ingeniería.

III. PROGRAMA ANALÍTICO

Las prácticas de la asignatura de Programación Digital a llevarse a cabo en la sala de cómputo, será en correspondencia a los temas avanzados de acuerdo a la siguiente programación:

Semana	Tema	Actividades
1	Introducción a Python y su aplicación en la Ingeniería de Minas	Clase magistral, discusión guiada
2	Tipos de datos, operadores y estructuras de control	Prácticas en Python, ejercicios guiados
3	Funciones y modularidad en Python	Implementación de funciones, trabajo en equipos
4	Estructuras de datos: Listas, tuplas y diccionarios	Resolución de problemas con estructuras de datos
5	Manejo de archivos y entrada/ salida de datos	Práctica en procesamiento de datos
6	Programación orientada a objetos (POO)	Implementación de clases y objetos en Python
7	Algoritmos y estructuras de datos avanzadas	Estudio de caso mineros
8	Optimización de código y buenas prácticas en Python Examen Parcial	Evaluación de conocimientos
9	Introducción a bibliotecas científicas: NumPy y Pandas	Aplicación en análisis de datos
10	Visualización de datos con Matplottlib	Desarrollo de gráficos y reportes
11	Análisis de datos mineros con Python	Trabajo aplicado en datos reales
12	Automatización de procesos en Python	Implementación de scripts automatizados
13	Seguridad y manejo de errores en Python	Estrategias de depuración
14	Desarrollo de un proyecto de minería de datos	Proyecto en equipo
15	Modelado y simulación en Python para minería	Desarrollo de simulaciones
16	Presentación de proyectos finales Examen Final	Evaluación integral

IV. METODOLOGÍA

La metodología del curso combina sesiones teóricas, prácticas y trabajo en equipo. Se hará uso de herramientas digitales, entornos de programación interactivos y proyectos aplicados a la minería. Se promoverá al aprendizaje activo mediante la resolución de problemas y la aplicación de metodologías de desarrollo de software.

V. SISTEMA DE EVALUACIÓN

El examen podrá constar de teoría y problemas numéricos en proporciones no preestablecidas. Se establecerá una nota mínima para cada ejercicio que dependerá de la dificultad del mismo (a título orientativo podría estar de 4 puntos sobre 20). De no obtener en cada ejercicio una nota igual o superior l mínimo el examen no se considerará superado. La calificación de cada examen parcial y del examen final de la signatura se establece mediante la suma de las notas de los diferentes ejercicios que componen el examen. El coeficiente de ponderación estará indicado en el enunciado del examen. De no ser así se entiende que todas las partes tienen igual peso.

VI. REQUISITOS DE APROBACIÓN

El alumno tendrá que demostrar suficiencia en el curso para lo cual será necesario obtener una nota mínima de once (puntaje mínimo 42 puntos), resultado de calcular el promedio de los exámenes parciales, trabajos encargados expuestos y el trabajo semestral.

- Las evaluaciones teóricas parciales se tomarán previa coordinación en clases. Se realizará dos exámenes parciales, una al 50% del programa analítico y otra al final del mismo.
- Se considerará el promedio de los exámenes prácticos, de igual manera se considerará la asistencia a las prácticas semanales en la sala de computación.
- Se tomará en cuenta la asistencia a clases y las participaciones.
- Es necesario presentar un trabajo semestral para aprobar el curso, con este fin los estudiantes deben agruparse como máximo de 4 integrantes y abordar cualquier tema inmerso a la carrera.

El promedio final estará establecido mediante:

$$PF = \frac{PEP + EP1 + EP2 + PTE + TF}{5}$$

- PF: Promedio final
- PEP: Promedio de evaluaciones parciales.
- EP1: Evaluación práctico uno (sala de cómputo).
- EP2: Evaluación práctico dos (en sala de cómputo).
- PTE: Promedio de trabajos encargados.
- TF: Trabajo semestral

La evaluación de las prácticas calificadas se realizará constantemente, en los horarios establecidos por grupos, en la sala de computación (H-208), por tal razón la asistencia es obligatoria.

VII. BIBLIOGRAFÍA:

1. AKAI, TERRENCE J, (2019). *Python for Data Analysis*, John Wiley y Sons, INC- Editorial Limusa. Grupo Noriega. México.
2. BUCHANAN, Sams (2023). Teach Yourself Microsoft Access 2018 in 24 Hours, Edit. Macmillan Computer Publishing. EE.UU.
3. CANALE, P. (2020). Programación para Ingenieros. Editorial McGraw Hill. México.
4. LUTZ, M. (2019). *Learning* Python. O'Reilly Media.

5. MARTELLI, A. (2018). *Python Cookbook*. Alfaomega- Roma.
6. McKINNEY, W. (2017). *Python for Data Analysis.* McGraw Hill Professional, Microsoft.
7. PAO, Y.C. (2022). Engieenering Analysis – Interactive Methods and Programs wuth Fortran, QB. Edit. CCR Press LLc- EE.UU.
8. ZELLE, J. (2016). *Python Programming: An Introduction to Computer Science.* Franklin, Beedle & Associates Inc.

Ayacucho, junio de 2024

Anexo 6. Ficha de observación

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL
ESCUELA PROFESIONAL DE INGENIERÍA DE MINAS

FICHA DE OBSERVACIÓN PARA EL APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON

I. DATOS INFORMATIVOS

1.1. Docente Investigador: ..... DNI: .....
1.2. Correo electrónico: ..... Fecha: .....
1.3. Estudiante Observado: ..... Código: .....
1.4. Curso: ..... Grupo: .....
1.5. Hora de inicio: ..... Hora de término: .....

II. INSTRUCCIONES

2.1. Maque la valoración correspondiente para cada ítem basado en el desempeño del estudiante.
2.2. Se recomienda añadir comentarios adicionales al final de cada sección para proporcionar una retroalimentación detallada.
2.3. La escala de valoración está definida como:

Nivel	Puntaje	Descripción
[1] Deficiente	08 - 11	Dificultades significativas que requieren más práctica.
[2] Regular	11 - 14	Comprensión parcial con algunos errores.
[3] Bueno	14 - 17	Dominio adecuado y precisión en la habilidad evaluada.
[4] Excelente	17 - 20	Dominio excepcional y precisión sobresaliente.

I. DIMENSIÓN Y1: FUNDAMENTOS DE PYTHON

Nº	ITEMS	1	2	3	4
1	Comprende el concepto de palabras reservadas en Python y su uso en sintaxis del lenguaje.				
2	Identifica y aplica correctamente los tipos de datos básicos en Python (enteros, flotantes, booleanos, cadenas y complejos).				
3	Utiliza operadores matemáticos, lógicos y relacionales para la manipulación de datos de Python.				
4	Implementa la conversión de tipos de datos entre enteros, flotantes, booleanos y cadenas en Python.				
5	Aplica correctamente la función *print( )* para la impresión de resultados en pantalla.				
6	Emplea la función *input( )* para capturar datos del usuario y almacenarlos en variable.				

7	Implementa estructuras de control de flujo como condicionales *if, elif* y *else* para la toma de decisiones.				
8	Comprende y aplica correctamente estructuras de control iterativas como *for* y *while.*				
9	Utiliza funciones en Python para modurizar códigos y mejorar la reutilización de procedimientos.				
10	Identifica y soluciona errores sintácticos y lógicos en los programas desarrollados.				

Comentarios adicionales sobre Fundamentos de Python:

.....

.....

II. DIMENSIÓN Y2: PYTHON APLICADO A LA PROGRAMACIÓN DIGITAL

Nº	ITEMS	1	2	3	4
11	Escribe programas secuenciales que capturen datos de entrada y generan resultados esperados.				
12	Implementa estructuras condicionales en la resolución de problemas computacionales.				
13	Utiliza estructuras de repetición para iterar procesos en función de una condición.				
14	Emplea listas y tuplas para almacenar y manipular colecciones de datos en Python.				
15	Utiliza diccionarios para almacenar datos en estructuras clave- valor y acceder eficientemente a la información.				
16	Aplica funciones personalizadas y predefinidas para la optimización del código.				
17	Desarrolla programas que interactúan con archivos para leer y escribir datos en formatos *.txt* o *.csv.*				
18	Implementa módulos y bibliotecas en Python para extender las funcionalidades del lenguaje.				
19	Aplica conceptos de programación orientada a objetos, incluyendo clases, objetos, métodos y atributos.				
20	Implementa la manipulación de datos y estructuras avanzadas como matrices y listas anidadas para resolver problemas del ámbito de la Ingeniería de Minas.				

Comentarios adicionales sobre Python Aplicado a la Programación Digital:

.....

.....

III. RECOMENDACIONES

.....

.....

IV. CONFIDENCIALIDAD

Toda información proporcionada será tratada con estricta confidencialidad y se utilizará únicamente con fines de investigación académica.



Docente Investigador

Anexo 7. Instrumento pre test

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
Unidad de Posgrado de la Facultad de Ciencias de la Educación
Doctorado en Educación

INSTRUMENTO
PRE – TEST PARA EL APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON

APELLIDOS Y NOMBRES: ..... **FECHA:** .....

INSTRUCCIONES:
Este pre-test evalúa tus conocimientos en Programación Digital con Python antes de la intervención con ChatGPT. Responde de manera individual, utilizando buenas prácticas de programación y justificando cada respuesta.

I. FUNDAMENTOS DE PYTHON

1. Palabras Reservadas y Sintaxis

Analiza el siguiente código e identifica los errores sintácticos. Luego, corrígelo para que imprima todas las palabras reservadas de Python en orden alfabético.

```
import keyword

def palabras_reservadas():
    reserved = keyword.kwlist
    print(sorted(reserved))

palabras_reservadas()
```

Criterios de evaluación: Comprensión de la librería *keyword*, organización y aplicación de buenas prácticas de programación.

2. Tipos de Datos y Conversión

Explica detalladamente la salida del siguiente código y propón una mejora para hacerlo más eficiente.

```
num = input("Ingrese un número entero: ")
num = int(num)
float_num = float(num)
bool_num = bool(num)
str_num = str(num)

print(f"Flotante: {float_num}, Booleano: {bool_num}, Cadena: {str_num}")
```

Criterios de evaluación: Comprensión de tipo de datos, eficiencia del código y justificación de mejoras.

3. Operadores Matemáticos y Lógicos

Implementa un programa modular que solicite dos números al usuario, realice las operaciones matemáticas y lógicas indicadas y devuelva un informe estructurado de los resultados.

Criterios de evaluación: Aplicación de funciones, organización del código y estructuración de la salida.

II. PYTHON APLICADO A LA PROGRAMACIÓN DIGITAL

4. Uso de Listas y Tuplas

Corrige y optimiza el siguiente código para que almacene cinco números en una lista, los convierta en una tupla inmutable y muestre el número mayor y menor utilizando una sola función.

```
def procesar_numeros():
    nums = []
    for i in range(5):
        nums.append(int(input(f"Ingrese el número {i+1}: ")))
    nums_tuple = tuple(nums)
    return nums_tuple, max(nums_tuple), min(nums_tuple)

tup, max_num, min_num = procesar_numeros()
print(f"Tupla: {tup}, Mayor: {max_num}, Menor: {min_num}")
```

Criterios de evaluación: Estructura del código, reutilización de funciones y optimización de procesos.

5. Diccionarios y Búsqueda de Datos

Dado el siguiente diccionario con densidades de minerales, mejora el programa para manejar excepciones, eliminar espacios en la entrada y sugerir opciones disponibles si el mineral no es encontrado.

```
minerales = {"Oro": 19.32, "Plata": 10.49, "Cobre": 8.96}

try:
    mineral = input("Ingrese el nombre de un mineral: ").strip().capitalize()
    if mineral in minerales:
        print(f"Densidad de {mineral}: {minerales[mineral]} g/cm³")
    else:
        print(f"Mineral no encontrado. Opciones disponibles: {',
    '.join(minerales.keys())}")
except Exception as e:
    print(f"Ocurrió un error inesperado: {e}")
```

Criterios de evaluación: Manejo de errores, validación de datos y robustez del código.

6. Manejo de archivos

Escribe un programa que cree un archivo, escriba tres líneas de información y luego lo lea y muestre en pantalla, optimizando el manejo de archivos.

III. FORMATO DE ENTREGA

Cada estudiante debe enviar:

- Código fuente en Python (*.py)*
- Captura de pantalla de la ejecución.
- Informe en PDF explicando el código.

Nota: Este pre-test permite evaluar conocimientos previos en Programación Digital con Python antes de usar ChatGPT como herramienta de aprendizaje.

Ayacucho, junio de 2024

Anexo 8. Instrumento pos test

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
Unidad de Posgrado de la Facultad de Ciencias de la Educación
Doctorado en Educación

INSTRUMENTO
POST – TEST PARA EL APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON

APELLIDOS Y NOMBRES: ..... **FECHA:** .....

INSTRUCCIONES:
Este pos-test evalúa tus conocimientos en Programación Digital con Python después de la intervención con ChatGPT. Responde de manera individual, utilizando buenas prácticas de programación y justificando cada respuesta.

IV. FUNDAMENTOS DE PYTHON

7. Palabras Reservadas y Sintaxis

Analiza el siguiente código e identifica los errores sintácticos. Luego, corrígelo para que imprima todas las palabras reservadas de Python en orden alfabético.

```
import keyword

def palabras_reservadas():
    reserved = keyword.kwlist
    print(sorted(reserved))

palabras_reservadas()
```

Criterios de evaluación: Comprensión de la librería *keyword*, organización y aplicación de buenas prácticas de programación.

8. Tipos de Datos y Conversión

Explica detalladamente la salida del siguiente código y propón una mejora para hacerlo más eficiente.

```
num = input("Ingrese un número entero: ")
num = int(num)
float_num = float(num)
bool_num = bool(num)
str_num = str(num)

print(f"Flotante: {float_num}, Booleano: {bool_num}, Cadena: {str_num}")
```

Criterios de evaluación: Comprensión de tipo de datos, eficiencia del código y justificación de mejoras.

9. Operadores Matemáticos y Lógicos

Implementa un programa modular que solicite dos números al usuario, realice las operaciones matemáticas y lógicas indicadas y devuelva un informe estructurado de los resultados.

Criterios de evaluación: Aplicación de funciones, organización del código y estructuración de la salida.

V. PYTHON APLICADO A LA PROGRAMACIÓN DIGITAL

10. Uso de Listas y Tuplas

Corrige y optimiza el siguiente código para que almacene cinco números en una lista, los convierta en una tupla inmutable y muestre el número mayor y menor utilizando una sola función.

```
def procesar_numeros():
    nums = []
    for i in range(5):
        nums.append(int(input(f"Ingrese el número {i+1}: ")))
    nums_tuple = tuple(nums)
    return nums_tuple, max(nums_tuple), min(nums_tuple)

tup, max_num, min_num = procesar_numeros()
print(f"Tupla: {tup}, Mayor: {max_num}, Menor: {min_num}")
```

Criterios de evaluación: Estructura del código, reutilización de funciones y optimización de procesos.

11. Diccionarios y Búsqueda de Datos

Dado el siguiente diccionario con densidades de minerales, mejora el programa para manejar excepciones, eliminar espacios en la entrada y sugerir opciones disponibles si el mineral no es encontrado.

```
minerales = {"Oro": 19.32, "Plata": 10.49, "Cobre": 8.96}

try:
    mineral = input("Ingrese el nombre de un mineral: ").strip().capitalize()
    if mineral in minerales:
        print(f"Densidad de {mineral}: {minerales[mineral]} g/cm³")
    else:
        print(f"Mineral no encontrado. Opciones disponibles: {',
    '.join(minerales.keys())}")
except Exception as e:
    print(f"Ocurrió un error inesperado: {e}")
```

Criterios de evaluación: Manejo de errores, validación de datos y robustez del código.

12. Manejo de archivos

Escribe un programa que cree un archivo, escriba tres líneas de información y luego lo lea y muestre en pantalla, optimizando el manejo de archivos.

VI. FORMATO DE ENTREGA

Cada estudiante debe enviar:

- Código fuente en Python (*.py)*
- Captura de pantalla de la ejecución.
- Informe en PDF explicando el código.

Nota: Este pos-test permite evaluar conocimientos en Programación Digital con Python después de usar ChatGPT como herramienta de aprendizaje.

Ayacucho, octubre de 2024

Anexo 09. Opinión de expertos

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
Unidad de Posgrado de la Facultad de Ciencias de la Educación
Doctorado en Educación

FICHA DE VALIDACIÓN
INFORME DE OPINIÓN DE JUICIO DE EXPERTO

I. DATOS GENERALES

Título de la investigación: ChatGPT en aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la UNSCH, 2024.

Nombre del instrumento motivo de evaluación: Ficha de observación

Nombre del autor: Amilcar Tacuri Gamboa

II. ASPECTOS DE VALIDACIÓN

Indicadores	Criterios	Deficiente				Baja				Regular				Bueno				Muy Bueno			
		0	6	11	16	21	26	31	36	41	46	51	56	61	66	71	76	81	86	91	96
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1. CLARIDAD	Está formulado con lenguaje propio.																				
2. OBJETIVIDAD	Esta expresado en conductas observables.																				
3. ACTUALIDAD	Adecuado al avance de la ciencia pedagógica.																				
4. ORGANIZACIÓN	Existe una organización lógica.																				
5. SUFICIENCIA	Comprende los aspectos en cantidad y calidad.																				
6. INTENCIONALIDAD	Adecuado para valorar los indicadores.																				
7. CONSISTENCIA	Basado en aspectos teóricos científicos.																				
8. COHERENCIA	Entre los ítems e indicadores.																				
9. METODOLGÍA	La estrategia responde al propósito de la investigación.																				
10.PERTINENCIA	Es útil y adecuado para la investigación.																				

PROMEDIO DE VALORACIÓN

OPINIÓN DE APLICABILIDAD: a) Deficiente b) Baja c) Regular d) Buena e) Muy Buena

Nombres y Apellidos		DNI:	
Título Profesional			
Grado Académico			
Mención			

Lugar y Fecha:



Anexo 11. Fotografía de la UNSCH



Anexo 11. Estudiantes





















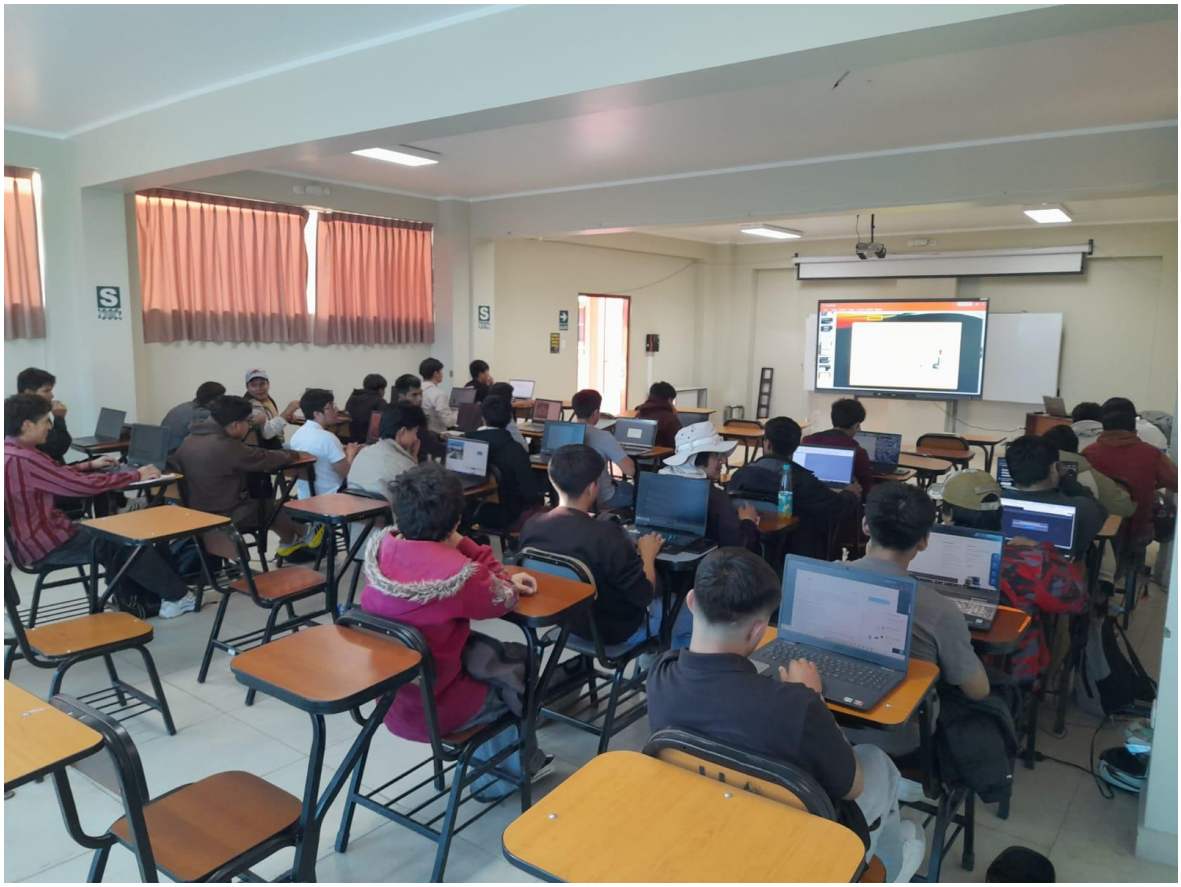
































Anexo 10. Envío de Articulo Científico a Revista Invecom

Gmail **Amilcar Tacuri Gamboa <tacuri.gamboa@gmail.com>**

Su envío de manuscritos - 1517653

1 mensaje

Editorial.office@invecom.org <education.editorial.office@invecomsin.org> 30 de Agosto de 2025, 12:13 p.m.
Respondedor a: Invecom en Psicología <Invecom.editorial.office@f.org>
Para: tacuri.gamboa@gmail.com

Estimado Dr. Tacuri Gamboa

Nos complace informarles que hemos recibido el manuscrito "Impacto de ChatGPT en Python: Un estudio de caso con estudiantes de la Universidad Nacional de Sn Cristóbal de Huamanga" para su consideración en *Invecom*.

Puede acceder al foro de revisión y seguir el progreso de su manuscrito utilizando el siguiente enlace: https://www.invecomsin.org// Journal/MySubmission.aspx?etapa = 100

Su manuscrito se encuentra ahora en la etapa de validación inicial para determinar su idoneidad para la revisión por pares. Si su manuscrito se envía para revisión por pares, recibirá una notificación una vez que hayamos recibido los informes de los revisores y se active el foro de revisión interactivo. Luego podrá leer los informes de revisión, intercambiar directamente con los revisores en el foro de revisión interactivo y enviar un manuscrito revisado, si corresponde. Si el número requerido de revisores respalda su manuscrito en la etapa de Revisión Independiente, sus pestañas se cerrarán y el manuscrito se enviará a la etapa de Revisión Finalizada, donde podrá interactuar con el editor de manejo a través de la pestaña Editor.

Tenga en cuenta que, de acuerdo con nuestra política de autoría (https://www.invecomsin.org/ guidelines/policies-and-publication-ethics #:~:text=Changes%20in%20authorship), la versión de la lista de autores que aparece en el sistema de envío en esta etapa se considera definitiva, excepto en circunstancias excepcionales, por ejemplo, cuando el proceso de revisión requiere experiencia adicional. Si necesita cambiar la lista de autores en cualquier etapa del proceso de revisión, envíe un formulario de cambio de autoría a la oficina editorial utilizando el formulario de cambio de autoría [https://www.invecomsin.org/ files/pdf/Authorship_change_form_CRediT.pdf]. No se reconocerán los cambios en la lista de autores en los archivos de manuscritos cargados.

Saludos
Su equipo de Invecom en Educación,

Invecom | Oficina Editorial - Equipo de Revisión por Pares Colaborativos
www.invecomsin.org
Avenue du Tribunal Fédéral 34,
1005 Maracaibo, Venezuela

Para problemas técnicos, comunícate con nuestro servicio de asistencia de TI (support@invecomsin.org) o visita nuestro Centro de ayuda de Frontiers (helpcenter.invecom.org)

----------DETALLES DEL MANUSCRITO----------
Título del manuscrito: Integrando la educación: Impacto de ChatGPT en Python: un estudio de caso con estudiantes de la Universidad Nacional de San Cristóbal de Huamanga. ID del manuscrito: 1517653
Autor: Amilcar Tacuri Gamboa
Revista: Editorial.office@invecom.org, sección Educación.Tipo de artículo: Investigación original
Enviado el: 30 de Agosto de 2025

----------INFORMACIÓN ADICIONAL----------
Para permitir un proceso de revisión fluido y eficiente, familiarícese con las pautas de revisión de Frontiers: https://www.invecomsin.org/ Journal/ReviewGuidelines.aspx?s=1755&nombre=educación

Para participar en la Iniciativa de Identificación de Recursos, cite anticuerpos, organismos genéticamente modificados, herramientas de software, datos, bases de datos y servicios utilizando el número de catálogo correspondiente y RRID en el texto de su artículo. Consulte aquí para obtener más información: https://www.education.org// files/pdf/letter_to_author.pdf

Artículo científico

Impacto del Uso de ChatGPT en el Aprendizaje de Programación Digital con Python en Estudiantes de Ingeniería de Minas

Impact of ChatGPT Use on Learning Digital Programming with Python Among Mining Engineering Students

Amilcar Tacuri Gamboa

(https://orcid.org/0000-0002-2454-848X)

Universidad Nacional de San Cristóbal de Huamanga, Ayacucho, Perú

amilcar.tacuri@unsch.edu.pe

RESUMEN

El objetivo de este estudio fue determinar la influencia del uso de ChatGPT en el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la Universidad Nacional de San Cristóbal de Huamanga durante el semestre 2024-II. La investigación fue de tipo aplicada, con un diseño cuasi-experimental y un enfoque cuantitativo. Se trabajó con una muestra de 40 estudiantes, divididos en un grupo experimental que utilizó ChatGPT como herramienta de apoyo y un grupo de control con enseñanza tradicional. Los instrumentos, validados por expertos y con alta confiabilidad (α=0.867), mostraron que el 75% del grupo experimental alcanzó un nivel de aprendizaje "Bueno" y el 25% "Excelente" en el postest, mientras que el grupo control se mantuvo en niveles inferiores (60% "Regular", 40% "Bueno"). A nivel inferencial, las pruebas U de Mann-Whitney (p=0.000) y Wilcoxon (p=0.000) confirmaron diferencias

significativas, concluyendo que ChatGPT influye positivamente en el aprendizaje de la programación con Python.

Palabras clave: ChatGPT, Python, programación digital, inteligencia artificial educativa, aprendizaje autónomo.

ABSTRACT

The purpose of this study was to determine the influence of ChatGPT on the learning of Digital Programming with Python among Mining Engineering students at the National University of San Cristóbal de Huamanga during the 2024-II academic semester. The research was applied, with a quasi-experimental design and a quantitative approach. A sample of 40 students was used, divided into an experimental group that used ChatGPT as a support tool and a control group with traditional teaching. The instruments, validated by experts and with high reliability (α=0.867), showed that 75% of the experimental group reached a "Good" level and 25% an "Excellent" level in the post-test, while the control group remained at lower levels (60% "Regular", 40% "Good"). At the inferential level, the Mann-Whitney U (p=0.000) and Wilcoxon (p=0.000) tests confirmed significant differences, concluding that ChatGPT positively influences the learning of Python programming.

Keywords: ChatGPT, Python, digital programming, educational artificial intelligence, autonomous learning.

I. INTRODUCCIÓN

La integración de herramientas de inteligencia artificial (IA) en la educación superior representa una frontera innovadora para potenciar el desarrollo de competencias técnicas. En disciplinas como la ingeniería, el dominio de lenguajes de programación como Python es crucial, dada su versatilidad en análisis de datos, automatización y simulación de procesos (McKinney, 2017). Sin embargo, los

estudiantes suelen enfrentar dificultades significativas para adquirir estas habilidades, debido a métodos de enseñanza tradicionales que no siempre fomentan la práctica y la resolución autónoma de problemas (Pérez, 2018).

En este contexto, los modelos de lenguaje de gran tamaño, como ChatGPT, han emergido como recursos con un potencial transformador. Desarrollado por OpenAI, ChatGPT puede generar explicaciones, proporcionar ejemplos de código y ofrecer retroalimentación instantánea en lenguaje natural, actuando como un tutor personalizado (Brown et al., 2020). Estudios preliminares indican que la IA puede mejorar la comprensión conceptual y la motivación en el aprendizaje de STEM (Ciencia, Tecnología, Ingeniería y Matemáticas) (Gao, Wang & Xie, 2021; Rojas & Espinoza, 2021).

A pesar de este potencial, existe una carencia de investigación empírica robusta que evalúe el impacto específico de ChatGPT en el aprendizaje de la programación dentro de contextos educativos reales, particularmente en carreras de ingeniería aplicada como la de Minas. Por lo tanto, esta investigación se propuso determinar la influencia del uso de ChatGPT en el aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas. El estudio se guió por la hipótesis de que la integración de ChatGPT mejoraría significativamente la comprensión de los fundamentos de Python y su aplicación en la resolución de problemas digitales, en comparación con los métodos de enseñanza tradicionales.

II. METODOLOGÍA

Tipo y Diseño de Investigación

La investigación fue de tipo aplicada, con un nivel explicativo y un diseño cuasi-experimental. Se empleó el método hipotético-deductivo. El diseño consistió en la

comparación de un grupo experimental y un grupo control, con mediciones de pretest y postest. El esquema se representa a continuación:

- Grupo Control (GC): O1 (Pretest) → ... (Enseñanza Tradicional) → O3 (Postest)
- Grupo Experimental (GE): O2 (Pretest) → X (Intervención con ChatGPT) → O4 (Postest)

Población y Muestra

La población estuvo constituida por los 40 estudiantes matriculados en el curso de Programación Digital de la Escuela Profesional de Ingeniería de Minas de la UNSCH durante el semestre 2024-II. Dada la accesibilidad de la población completa, se decidió trabajar con la totalidad de los estudiantes, los cuales fueron divididos mediante un muestreo aleatorio simple en dos grupos: un Grupo Control (GC) con 20 estudiantes (7 mujeres, 13 hombres) y un Grupo Experimental (GE) con 20 estudiantes (7 mujeres, 13 hombres).

Variables e Instrumentos

- Variable Independiente: Uso de ChatGPT.
- Variable Dependiente: Aprendizaje de Programación Digital con Python, operacionalizada en dos dimensiones:

  D1: Fundamentos de Python (sintaxis, tipos de datos, estructuras de control, funciones).

  D2: Python aplicado a la programación digital (manejo de archivos, POO, bibliotecas como Pandas).

Instrumento

Se utilizó una Ficha de Observación con 20 ítems (10 por dimensión), calificados en una escala ordinal de 4 niveles (1=Deficiente, 2=Regular, 3=Bueno,

4=Excelente). El instrumento fue validado por 5 expertos, obteniendo una valoración promedio de 83.9% ("Muy Buena"). La confiabilidad, medida con Alfa de Cronbach mediante una prueba piloto, fue de 0.867, indicando una consistencia interna excelente.

Procedimiento e Intervención

1. Se aplicó un pretest a ambos grupos para establecer la homogeneidad inicial.
2. El GC recibió enseñanza tradicional basada en clases expositivas y ejercicios guiados.
3. El GE, además de la enseñanza tradicional, utilizó ChatGPT como herramienta complementaria. La intervención consistió en 16 sesiones pedagógicas donde los estudiantes usaban "prompts" estructurados para consultar conceptos, depurar código, generar ejemplos y recibir retroalimentación en tiempo real.
4. Al finalizar el semestre, se aplicó un postest a ambos grupos.

Análisis de Datos

Se utilizó el software SPSS v28. Dado que la prueba de Shapiro-Wilk indicó que los datos no seguían una distribución normal (p=0.000), se emplearon pruebas no paramétricas. La prueba U de Mann-Whitney se usó para comparar los grupos independientes (GC vs. GE en el postest), y la prueba de rangos con signo de Wilcoxon para comparar las medidas relacionadas (pretest vs. postest dentro del GE). El nivel de significancia fue $\alpha=0.05$.

Consideraciones Éticas

Se obtuvo el consentimiento informado de los participantes y la autorización institucional (Oficio N° 045-2024-UNSCH-FIMGC-DA). Se garantizó la confidencialidad de los datos y el anonimato de los estudiantes.

III. RESULTADOS Y DISCUSIÓN

Resultados Descriptivos

Los resultados del pretest confirmaron la homogeneidad inicial, con el 100% de los estudiantes en ambos grupos ubicados en el nivel "Deficiente". Los resultados del postest mostraron una marcada diferencia entre los grupos.

En la variable general de aprendizaje (Tabla 1), el 75% del GE alcanzó un nivel "Bueno" y el 25% "Excelente". En contraste, el GC mostró un 60% en "Regular" y un 40% en "Bueno". Ningún estudiante del GC logró el nivel "Excelente".

Tabla 1

Nivel de Aprendizaje de Programación Digital con Python (Postest)

Categoría	Grupo Control (frecuencia/%)
Categoría	0 (0%)
Regular	12 (60%)
Bueno	8 (40%)
Excelente	0 (0%)
Total	20 (100%)

Esta mejora superior del GE se replicó consistentemente en ambas dimensiones analizadas: Fundamentos de Python (GE: 55% Bueno, 45% Excelente; GC: 40% Bueno, 60% Regular) y Python Aplicado (GE: 65% Bueno, 35% Excelente; GC: 60% Bueno, 40% Regular).

Resultados Inferenciales

La prueba U de Mann-Whitney no mostró diferencias significativas en el pretest entre el GC y el GE (p=1.000), confirmando la equivalencia inicial. Sin embargo, en el postest, la prueba reveló diferencias altamente significativas a favor del GE en la variable general (p=0.000), así como en la dimensión Fundamentos de Python (p=0.000) y Python Aplicado (p=0.000).

La prueba de Wilcoxon aplicada al GE evidenció una mejora estadísticamente significativa entre el pretest y el postest para la variable general (p=0.000; Z = -3.935), la dimensión Fundamentos (p=0.000; Z = -3.823) y la dimensión Aplicada (p=0.000; Z = -3.889). Estos resultados permiten rechazar las hipótesis nulas y aceptar las hipótesis alternativas de la investigación.

DISCUSIÓN

Los hallazgos de este estudio demuestran de manera contundente que la integración de ChatGPT como herramienta de apoyo pedagógico influye positivamente y de forma significativa en el aprendizaje de la programación digital con Python. La mejora dramática observada en el grupo experimental, que pasó de un 100% en nivel "Deficiente" a un 100% en niveles "Bueno" y "Excelente", subraya el potencial de la IA para transformar los resultados de aprendizaje en entornos técnicos.

Estos resultados están en línea con investigaciones previas. Althobaiti (2021) y Gao, Wang y Xie (2021) ya habían señalado el impacto positivo de la IA en la educación matemática y técnica. Nuestro estudio extiende estos hallazgos al ámbito específico de la programación con Python, validando que la capacidad de ChatGPT para ofrecer retroalimentación inmediata y personalizada (Smith & Johnson, 2020) facilita la comprensión de conceptos complejos y acelera la curva de aprendizaje. La herramienta actuó como un "andamiaje cognitivo" (Holmes et al., 2019), permitiendo a los estudiantes resolver dudas en tiempo real y fomentando un aprendizaje más autónomo y exploratorio (García & Espinoza, 2021).

La superioridad del GE en ambas dimensiones (fundamentos y aplicación) sugiere que ChatGPT no es solo útil para la comprensión sintáctica, sino también

para el desarrollo del pensamiento computacional y la capacidad de aplicar el conocimiento a problemas prácticos, una competencia crucial en la formación de ingenieros (Guzdial, 2019). La propuesta pedagógica implementada, con prompts contextualizados a la ingeniería de minas, demostró ser un modelo efectivo para integrar esta tecnología de manera estructurada y alineada con los objetivos de aprendizaje.

Si bien existen preocupaciones válidas sobre el uso ético de la IA, como la posible dependencia o el plagio (Zhang et al., 2020), este estudio demostró que, con una guía adecuada, ChatGPT puede ser un copiloto efectivo en el proceso de aprendizaje, potenciando 而不是 reemplazando el esfuerzo intelectual del estudiante.

CONCLUSIONES

El uso de ChatGPT influye significativamente en el aprendizaje de Programación Digital con Python, generando una mejora sustancial en el rendimiento académico de los estudiantes de Ingeniería de Minas, tal como lo demuestran las pruebas estadísticas U de Mann-Whitney y Wilcoxon.

La influencia positiva se verifica tanto en la comprensión de los fundamentos de Python (sintaxis, estructuras de control, funciones) como en la aplicación de estos conocimientos en la programación digital (manejo de archivos, POO, uso de bibliotecas), lo que indica un desarrollo integral de la competencia programática.
ChatGPT se erige como una herramienta pedagógica efectiva para promover el aprendizaje autónomo, proporcionando un entorno de apoyo interactivo y personalizado que responde a las necesidades individuales de los estudiantes.

Este estudio proporciona un modelo replicable para la integración de inteligencia artificial generativa en la educación superior técnica, ofreciendo una estrategia concreta para enriquecer los procesos de enseñanza-aprendizaje.

Se recomienda futuras investigaciones que exploren el impacto a largo plazo de ChatGPT, su eficacia en otras disciplinas de ingeniería y el desarrollo de marcos pedagógicos más amplios para el uso ético y efectivo de la IA en la educación.

REFERENCIAS BIBLIOGRÁFICAS

Althobaiti, M. (2021). Implications of Artificial Intelligence for Mathematics Education: Insights from the United Arab Emirates. International Journal of Technology in Education and Science.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Gao, Y., Wang, J., & Xie, H. (2021). Evaluating the Impact of Artificial Intelligence on Mathematics Learning in High School Students. Journal of Educational Computing Research.

García, A., & Espinoza, M. (2021). Enhancing learning autonomy in technology-enhanced language learning environments: The impact of artificial intelligence tools. Journal of Computer Assisted Learning.

Guzdial, M. (2019). Learner-Centered Design of Computing Education: Research on Computing for Everyone. Springer.

Holmes, W., Bialik, M., & Fadel, C. (2019). Artificial Intelligence In Education: Promises and Implications for Teaching and Learning. Center for Curriculum Redesign.

McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.

Pérez, F. (2018). Teaching Python: Bridging the gap between novice and software developer. Computer Science Education.

Rojas, A., & Espinoza, B. (2021). Impacto de la inteligencia artificial en la resolución de problemas matemáticos en estudiantes de secundaria en Lima. Revista de Educación.

Smith, M., & Johnson, L. (2020). AI and the future of learning: Ensuring the quality of responses from intelligent systems. Educational Technology Research and Development.

Zhang, Y., Chen, P., & Lu, X. (2020). The rise of AI in education: From passive to active learning. Educational Technology Research and Development.

CONSTANCIA DE ORIGINALIDAD N°0166-2026-UNSCH-EPG/KBA

El que suscribe; responsable verificador de originalidad de trabajo de tesis de Posgrado en segunda instancia para la **Escuela de Posgrado – UNSCH**; en cumplimiento a la Resolución Directoral Nº002-2026-UNSCH-EPG/D, Reglamento de Originalidad de trabajos de Investigación de la UNSCH, otorga lo siguiente:

CONSTANCIA DE ORIGINALIDAD

AUTOR	Mtro. AMILCAR TACURI GAMBOA
DENOMINACIÓN DEL PROGRAMA DE ESTUDIOS	DOCTORADO EN EDUCACIÓN
GRADO ACADÉMICO QUE OTORGA	DOCTOR
DENOMINACIÓN DEL GRADO ACADÉMICO	DOCTOR(A) EN EDUCACIÓN
TÍTULO DE TESIS	ChatGPT en aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la UNSCH, 2024
EVALUACIÓN DE ORIGINALIDAD	10% de similitud
N° DE TRABAJO	3065013832
FECHA	19 de octubre de 2026

Por tanto, según los artículos 12, 13 y 17 del Reglamento de Originalidad de Trabajos de Investigación, es procedente otorgar la constancia de originalidad con depósito.

Se expide la presente constancia a solicitud del interesado para los fines que crea conveniente.

19 de octubre de 2026.

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
ESCUELA DE POSGRADO
Mg. Kelvis BERROCAL ARGUMEDO
Sub Director de Investigación

CC.
Archivo
KBA/

ChatGPT en aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la UNSCH, 2024

por Amilcar TACURI GAMBOA

Fecha de entrega: 19-oct-2026 09:59p. m. (UTC-0500)
Identificador de la entrega: 3065013832
Nombre del archivo: Amilcar_TACURI_GAMBOA_T.docx (18.72M)
Total de palabras: 30781
Total de caracteres: 183495

ChatGPT en aprendizaje de Programación Digital con Python en estudiantes de Ingeniería de Minas de la UNSCH, 2024

INFORME DE ORIGINALIDAD

10%	11%	5%	6%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad Nacional de San Cristóbal de Huamanga Trabajo del estudiante	4%
2	repositorio.unsch.edu.pe Fuente de Internet	1%
3	www.dykinson.com Fuente de Internet	1%
4	zaguan.unizar.es Fuente de Internet	<1%
5	repositorio2.eesppsantarosacusco.edu.pe Fuente de Internet	<1%
6	hdl.handle.net Fuente de Internet	<1%
7	repositorio.unamad.edu.pe Fuente de Internet	<1%
8	scielo.br Fuente de Internet	<1%
9	rua.ua.es Fuente de Internet	<1%
10	Submitted to Unidad Educativa Letort Covima Trabajo del estudiante	<1%
11	Submitted to Universidad Alas Peruanas Trabajo del estudiante	<1%
12	repositorio.unemi.edu.ec Fuente de Internet	<1%
13	repositorio.escuelamilitar.edu.pe Fuente de Internet	<1%
14	revistas.unamad.edu.pe Fuente de Internet	<1%
	www.coursehero.com	

15	Fuente de Internet	<1 %
16	repositorio.une.edu.pe Fuente de Internet	<1 %
17	Pedro Luis Intriago Intriago, Santiago Paul León Granizo, Jesús Antonio Alba Cabañas, Gregory Edison Naranjo Vaca. "GeoGebra and the Understanding of Regular Polygons", Tesla Revista Científica, 2026 Publicación	<1 %
18	www.editorial.risei.org Fuente de Internet	<1 %
19	repositorio.unheval.edu.pe Fuente de Internet	<1 %
20	repositorio.unjfsc.edu.pe Fuente de Internet	<1 %
21	www.scribd.com Fuente de Internet	<1 %
22	Fabiola Milagros García Falcón, Liz Karen Gomero Rodríguez, Efrain Ascencion Felix Pachas, Yolanda Quispe Alosilla. "Curriculum internationalization as a way to raise the quality of education: perceptions of Peruvian university stakeholders", Revista Simón Rodríguez, 2026 Publicación	<1 %
23	stem.biu.us Fuente de Internet	<1 %
24	digitum.um.es Fuente de Internet	<1 %
25	www.edutec.es Fuente de Internet	<1 %

Excluir citas	Activo	Excluir coincidencias	< 30 words
Excluir bibliografía	Activo		

ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE DOCTOR(A) EN EDUCACIÓN RESOLUCIÓN DIRECTORAL N°0068-2026-UNSCH-EPG/D.

Siendo las 05:00 p.m. del 31 de marzo de 2026 se reunieron en el auditórium de la Escuela de Posgrado de la Universidad Nacional de San Cristóbal de Huamanga, el Jurado Examinador y Calificador de Tesis, presidido por el **Dr. MARCELINO POMASONCCO ILLANES** Director (e) de la Escuela de Posgrado, el **Dr. GUIDO ALFONSO PEREZ SAEZ** Director de la Unidad de Posgrado de la Facultad de Ciencias de la Educación, e integrado por los siguientes miembros: **Dr. VICTOR GEDEON PALOMINO ROJAS** y el **Dr. FEDERICO ALTAMIRANO FLOPES**; para la sustentación oral y pública de la tesis titulada: **CHATGPT EN APRENDIZAJE DE PROGRAMACIÓN DIGITAL CON PYTHON EN ESTUDIANTES DE INGENIERÍA DE MINAS DE LA UNSCH, 2024,** presentado por el **Mg. AMILCAR TACURI GAMBOA.** Teniendo como asesor al **Dr. TEODOSIO ZENOBIO POMA SOLIER.**

Acto seguido se procedió a la exposición de la tesis, con el fin de optar el Grado Académico de **DOCTOR(A) EN EDUCACIÓN.** Formuladas las preguntas, éstas fueron absueltas por el graduando.

A continuación, el Jurado Examinador y Calificador de Tesis procedió a la votación, la que dio como resultado el siguiente calificativo: DIECISIETE (17).

CALIFICACION (x)	
Aprobado(a) por Unanimidad.	X
Aprobado(a) por Mayoría.	
Desaprobado(a) por Unanimidad.	
Desaprobado(a) por Mayoría.	

Marcar con un aspa (x).

Luego, el presidente del Jurado recomienda que la Escuela de Posgrado proponga que se le otorgue al **Mg. AMILCAR TACURI GAMBOA,** el Grado Académico de **DOCTOR(A) EN EDUCACIÓN.** Siendo las.....18.15 PM.....hrs. se levanta la sesión.

Se extiende el acta en la ciudad de Ayacucho, a las.....18.20 PM.....hrs. del 31 de marzo de 2026.

.....
Dr. MARCELINO POMASONCCO ILLANES
Director (e) de la Escuela de Posgrado.

.....
Dr. GUIDO ALFONSO PEREZ SAEZ
Director (e) de la UPG-FCE.

.....
Dr. VICTOR GEDEON PALOMINO ROJAS
Miembro.

.....
Dr. FEDERICO ALTAMIRANO FLOPES
Miembro.

.....
Dr. ENRIQUE JAVIER AGUILAR FELICES
Secretario Docente.

Observaciones:

.....

.....

.....