UNIVERSIDAD CATÓLICA SANTO TORIBIO DE MOGROVEJO

ESCUELA DE POSGRADO



Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación 2024

TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN INFORMÁTICA EDUCATIVA Y TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

AUTOR

Hans Edgardo Delgado Diaz

ASESOR

Ricardo Chaname Chira

https://orcid.org/0000-0001-6410-5192

Chiclayo, 2025

Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación 2024

PRESENTADA POR

Hans Edgardo Delgado Diaz

A la Escuela de Posgrado de la

Universidad Católica Santo Toribio de Mogrovejo

para optar el grado académico de

MAESTRO EN INFORMÁTICA EDUCATIVA Y TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

APROBADA POR

Freundt Santimperi Sánchez

PRESIDENTE

Janeth Benedicta Falla Ortíz

SECRETARIO

Ricardo Chaname Chira

VOCAL

Dedicatoria

A Dios, Cristo, la Virgen María y al Santo Carlo Acutis, por las pruebas de fe, por todo lo que me han dado, por mis logros y no logros que me enseñaron y enseñan a ser fuerte.

A mis abuelos Marcelino y Ana María, por su sabiduría, sus enseñanzas, sus consejos y por mantenernos en contacto a pesar de que ya no están físicamente.

A mi madre, por su fuerza de voluntad, por todo lo que hiciste por mí, me enseñaste a ser la persona que soy: principios, valores, perseverante, trabajador y responsable.

A mi madrina Esperanza, por su apoyo entusiasta, sus consejos y apoyarme toda la vida para superarme profesionalmente.

A mi esposa Magali, a mis hijos: Anthony, Ricardo y Hans Alejandro; a mi nuera Janett y a mi nieta Fabiana.

Agradecimiento

Al Dr. Ricardo Chaname Chira por su apoyo y asesoramiento otorgado en esta investigación. A los Doctores y Maestros del Programa de Maestría, por sus sabias enseñanzas y asesoramiento. A las personas que con su apoyo directo e indirecto me motivaron a seguir mis estudios de posgrado, especialmente a mis tías Zoila, Ana y Liliana.

Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación 2024

INFORME DE ORIGINALIDAD

13%	12%	5%	3%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	hdl.handle.net Fuente de Internet	2%
2	tesis.usat.edu.pe Fuente de Internet	2%
3	www.scielo.org.mx Fuente de Internet	2%
4	repositorio.unfv.edu.pe Fuente de Internet	1%
5	revistasinvestigacion.unmsm.edu.pe Fuente de Internet	1%
6	Submitted to Universidad Cesar Vallejo Trabajo del estudiante	1%
7	xn--deepinenespaol-1nb.org Fuente de Internet	<1%
8	repositorio.unasam.edu.pe Fuente de Internet	<1%

Índice

Resumen 6

Abstract.....7

Introducción.....8

Revisión de literatura.....11

Bases teóricas 16

Materiales y métodos.....25

Resultados y discusión 35

Conclusiones.....43

Recomendaciones 44

Referencias 46

Anexos 50

Resumen

Los estudiantes de Educación Superior que siguen el curso de Fundamentos de Programación en las carreras con programación de computadoras sean de nivel técnico o universitario tienen problemas en sus habilidades prácticas en el desarrollo de algoritmos por lo abstracto de los temas de este curso. Ante el problema evidenciado se realizó esta investigación con el objetivo de determinar el impacto de la realidad aumentada en el desarrollo de habilidades prácticas para el curso de Fundamentos de Programación, mediante la implementación de una aplicación educativa que utilizando realidad aumentada mejore el aprendizaje de conceptos de fundamentos de programación. El estudio desarrollado es de tipo aplicada, de nivel descriptivo y de diseño cuasiexperimental con muestra de 30 estudiantes, entre las edades de 16 a 23 años, seleccionada mediante muestreo no probabilístico. Se aplicó un cuestionario que permitió medir los indicadores mediante un cuestionario físico, cuyo objetivo fue medir el nivel de las habilidades prácticas de los alumnos en el desarrollo de algoritmos en el curso de Fundamentos de Programación. Dicho instrumento estuvo conformado por 3 dimensiones, 5 indicadores y un conjunto de 20 ítems con puntuación estandarizada de 1 y 0. Según los resultados del estudio sobre el impacto de la realidad aumentada en el rendimiento académico en el postest se reveló una diferencia significativa entre el grupo experimental (que utilizó la aplicación de RA) y el grupo control, con una mediana de 17.0 frente a 12.0 respectivamente. En conclusión, se determinó el impacto significativo de la realidad aumentada (RA) en el desarrollo de habilidades prácticas para el curso de Fundamentos de Programación con la implementación de la aplicación educativa basada en RA demostrando un efecto positivo y significativo.

Palabras clave: Realidad Aumentada, Educación, Aplicativo Móvil

Abstract

Higher education students taking the Programming Fundamentals course in computer programming programs, whether at the technical or university level, struggle with their practical skills in developing algorithms due to the abstract nature of the course topics. Given this problem, this research was conducted to determine the impact of augmented reality on the development of practical skills for the Programming Fundamentals course by implementing an educational application that uses augmented reality to enhance the learning of programming fundamentals concepts. The study was applied, descriptive, and of a quasi-experimental design. The sample consisted of 30 students, between the ages of 16 and 23, selected through non-probability sampling. A questionnaire was administered to measure the indicators using a physical questionnaire. The objective was to measure the level of students' practical skills in developing algorithms in the Programming Fundamentals course. This instrument consisted of 3 dimensions, 5 indicators, and a set of 20 items with standardized scores of 1 and 0. According to the results of the study on the impact of augmented reality on academic performance, the post-test revealed a significant difference between the experimental group (which used the AR application) and the control group, with a median of 17.0 versus 12.0, respectively. In conclusion, the significant impact of augmented reality (AR) on the development of practical skills for the Programming Fundamentals course was determined with the implementation of the AR-based educational application, demonstrating a positive and significant effect..

Keywords: Augmented Reality, Education, Mobile Application

Introducción

En el ámbito de la Educación Superior, la competencia de habilidades prácticas en el curso de Fundamentos de Programación constituye una habilidad fundamental para el desarrollo integral del estudiante en los cursos de desarrollo de software de los futuros profesionales técnicos e ingenieros en informática. Ante lo dicho, diversas evidencias han demostrado que las habilidades prácticas afectan significativamente el nivel de aprovechamiento académico y el desarrollo del aprendizaje en los alumnos del nivel superior. Según Narváez *et al*.(2023), los resultados del curso de Fundamentos de Programación en estudios superiores muestran índices de aprobación no siempre satisfactorios al emplear estrategias tradicionales de enseñanza - con un 54.17 % de estudiantes considerados sobresalientes y un 45.83 % que no alcanzan dicho nivel -, se hace evidente la necesidad de explorar enfoques pedagógicos alternativos que favorezcan el desarrollo de habilidades prácticas y mejoren el rendimiento académico.

Además, diversos docentes e investigadores manifiestan lo complejo que resulta la enseñanza y el aprendizaje para aplicar la programación y sus fundamentos. Se estima al menos 10 años de experiencia para lograr un nivel de programador experto y conlleva que un alumno cundo termina su carrera todavía tiene que recorrer un largo camino de aprendizaje para lograr ser un programador experimentado.

De acuerdo con Jiménez-Toledo et al. (2019), es importante que el alumno tenga habilidades para resolver problemas y que tenga la capacidad de aplicar y crear sus propios modelos cognitivos, lógicos, matemáticos y algorítmicos. En este sentido, Insuasti (2016) indica que entre las razones por las que muchos estudiantes no alcanzan los resultados esperados pueden ser la dificultad de dominar el lenguaje de programación, los principios básicos del área, los antecedentes de los alumnos y la falta de una buena didáctica por parte del maestro, por mencionar las más importantes. Las clases iniciales de programación para los estudiantes de educación superior deben resultar cautivantes y atrayentes para motivar el interés de estudio de las ciencias de la computación.

Ahora bien, el algoritmo con el tiempo se ha hecho una palabra clave para los fundamentos de la computación (Cátedra Conceptos de Algoritmos, Datos y Programas, UNLP, 2016; Puig, 2009). Se conceptúa como la secuencia de instrucciones y procesos que conllevan a lograr la resolución a un problema, es una forma de plantear su solución (Cairo, 2005; Joyanes, 2003). Y en todos los análisis efectuados en el área de la computación, es una realidad que en los estudios superiores existen problemas y dificultades por parte de los alumnos cuando

empiezan a estudiar los fundamentos de la programación (Colorado, 2020); González et al,. 2012).

En cuanto a usar la tecnología de la información y la comunicación (TIC), Guerra (2023) manifiesta, en su estudio de realidad aumentada en su aplicación en el proceso de enseñanza - aprendizaje para la materia de química de nivel secundario en los centros educativos de la región Lambayeque, que las entidades educativas tienen el reto de innovarse con nuevas tecnologías en el manejo de formación de los estudiantes, ante la falencia de un nivel aceptable de capacidades básicas, de concentración, de motivación, de falta de atención, falta de confianza y de conocimientos iniciales.

Por otro lado, en 1992 el científico Tom Caudell promueve el término de realidad aumentada, Caudell (1992). La realidad aumentada (RA) es una aplicación tecnológica que mejora la captación y la unión de las personas con el medio real, al mezclar objetos virtuales tri- dimensionales que parecen convivir en el ambiente real. Es decir, hace que objetos virtuales se incorporen con el mundo real, dando una mejora importante en la experiencia de las personas y otorgando un mejor entendimiento al medio en el que se encuentran (Azuma, 2001). La realidad aumentada (RA), este es un instrumento tecnológico de innovación que mezcla la realidad virtual con lo físicamente real. Utiliza para esto modelos elaborados en 3D; por ejemplo, elaborados con Blender y Adobe Illustrator y una aplicación para dispositivos portables; elaborado, por ejemplo, en Unity donde a través del uso de marcadores y utilizando la cámara del dispositivo móvil permite mostrar los modelos 3D virtualmente.

Ante lo expuesto, se ha formulado la siguiente interrogante de investigación: ¿Cómo la implementación de una aplicación educativa con realidad aumentada en el curso de Fundamentos de Programación mejora significativamente el desarrollo de habilidades prácticas de los estudiantes en comparación con métodos de enseñanza tradicional?

Como se ha observado, los estudiantes presentan limitaciones en la comprensión de contenidos de naturaleza abstracta, lo que puede generar dificultades en cursos posteriores vinculados al área de programación, según lo previsto en los planes de estudio. En este contexto, la investigación se enmarca en el enfoque propuesto por Gabriel (2013), quien resalta la importancia de la comunicación y del uso adecuado de las tecnologías en el proceso de gestión de la información. Bajo esta perspectiva, se implementa la Realidad Aumentada (RA) en las sesiones del curso Fundamentos de Programación, en consonancia con el objetivo institucional de fomentar la innovación en la enseñanza. El estudio se orienta al desarrollo de habilidades analíticas y de interpretación de contenidos, integrando elementos

visuales y conceptuales mediante la RA, dentro del modelo pedagógico constructivista, el cual promueve una enseñanza centrada en la comprensión significativa. Desde el punto de vista metodológico, la propuesta se justifica por el diseño y uso de una aplicación educativa desarrollada en Unity, que incorpora herramientas y librerías específicas para la implementación de RA. Esta herramienta fortalece los procesos de aprendizaje en el curso Fundamentos de Programación, al facilitar la visualización y manipulación de conceptos complejos de manera interactiva.

Por lo expuesto, el objetivo general es determinar el impacto de la realidad aumentada en el desarrollo de habilidades prácticas para el curso de Fundamentos de Programación, mediante la implementación de una aplicación educativa que utilizando realidad aumentada mejora el aprendizaje de conceptos de fundamentos de programación.

Para lograr ello, los objetivos específicos planteados fueron los siguientes:

- Diagnosticar las habilidades prácticas del estudiante matriculado en el curso de fundamentos de programación
- Diseñar y desarrollar experiencias de aprendizaje basado en realidad aumentada en el curso de fundamentos de programación, haciendo uso de un aplicativo educativo móvil desarrollado en Unity con imágenes en 3D
- Evaluar el efecto mediático, de los contenidos visuales y multimedia, de la realidad aumentada en el desarrollo de habilidades prácticas para el curso de fundamento de programación.

Revisión de literatura

En el siguiente apartado se rescatan diferentes estudios que respaldan lo importante de la aplicación de la tecnología, sobre todo de la realidad aumentada, como estrategia para mejorar el desarrollo de habilidades prácticas en el curso de fundamentos de programación. Se han identificado antecedentes del estudio y las principales teorías que especificaran nuestros resultados.

Antecedentes

En la presentación de los antecedentes de esta investigación, se ha estructurado la revisión en tres niveles: internacional, nacional y local. Para los antecedentes internacionales, se han seleccionado 3 antecedentes, mediante una búsqueda en bases de datos académicas globales como IEEE Xplore, Scopus y Google Scholar, seleccionando artículos que exploraron la integración de la realidad aumentada en la enseñanza de programación, priorizando aquellos con metodologías innovadoras y resultados medibles en contextos educativos diversos. En cuanto a los antecedentes nacionales, se han seleccionado 3 mediante consultas a repositorios académicos, como revistas científicas y tesis de universidades del país, enfocándose en investigaciones que analizaran el impacto de la aplicación de tecnologías recientes, especialmente la realidad aumentada, en la educación de disciplinas tecnológicas. Finalmente, de los antecedentes locales, solo se encontraron solo 2; los cuales han seleccionados a partir de investigaciones previas realizadas en instituciones educativas de la región, tomando en cuenta proyectos específicos implementados en el entorno local que vincularan la realidad aumentada con la enseñanza de habilidades prácticas en programación. Los criterios utilizados para seleccionar estos antecedentes incluyeron la relevancia directa con el tema de estudio, la calidad de las publicaciones, y la aplicabilidad de las experiencias previas en el contexto de la investigación en curso.

Para fundamentar la investigación en el ámbito internacional, Acosta (2021), en su investigación sobre la creación de un programa de aplicación con uso de realidad aumentada para el estudio de nociones de programación, tuvo como objetivo de desarrollar una aplicación educativa integral (AU) utilizando realidad aumentada (AR) que ayude en el estudio de nociones iniciales de programación. La metodología se enmarca en un proyecto de investigación de modelos que tiene como base la Inteligencia Artificial y la Informática Ubicua para resolver problemas de nivel educativo y otras áreas relacionadas. La aplicación informática resultante tuvo como muestra a estudiantes que inician sus estudios de Licenciatura Información

(LSI) en y de Sistemas de Programador Universitario en Informática (PUI) de la Facultad de Ciencias Exactas y Tecnologías de la Universidad Nacional de Santiago del Estero en Argentina. El estudio mostró que al aplicar una herramienta informática que integra las nociones de AU, RA y OA para mejorar el entendimiento de conceptos básicos de programación se concluyó que la aplicación cumplió con los objetivos establecidos. La investigación muestra como la tecnología puede ser usada de manera eficiente como medio de complemento de material de estudio para aprendizajes de materias complejas en estudiantes de nivel superior.

Asimismo, Calderón (2023), en su investigación publicada en la Revista Religación, trata sobre realidad virtual y la realidad aumentada y las experiencias en el aprendizaje profundo en educación superior, donde usa como metodología la revisión de artículos bibliográficos con un enfoque exhaustivo y sistemático de las publicaciones a disposición en las bases de datos de tipo académicos y sobre todo de fuentes verosímiles. Como objetivo, la investigación fue resumir y analizar otras investigaciones previas al tema de estudio, para este caso el uso de la realidad virtual y aumentada y su impacto en diferentes actividades. Las búsquedas académicas se realizaron en servidores de información como IEEE Xplore, Google Scholar y Scopus, aplicando para esto frases como “realidad virtual”, “realidad aumentada”. Para el análisis se ha usado VOSviewer; el cual, es un aplicativo de muestra de datos y mapas de investigaciones que permite representar gráficamente y mapear las relaciones en frases claves y de términos que se abordaron en el estudio. Ante lo expuesto, concluyeron que al unir la realidad virtual y aumentada en la actividad de enseñanza de educación superior muestra una visión de aprendizaje emocionante y prometedor; asimismo, que es estas tecnologías inmersivas vienen demostrando una capacidad para enriquecer los procesos educativos y estimulan el aprendizaje profundo y significativo y explorar conocimientos abstractos de manera clara realizando simulaciones que mezclan la realidad y la virtualidad con fines de ayuda.

Otro aspecto relevante es, Saa-Saltos (2023), que resume en su artículo sobre el efecto de los conocimientos básicos de la creación de programas en el pensamiento abstracto en estudiantes de educación superior, estos estudios internacionales encaminados a validar un cuestionario para determinar las disciplinas de pensamiento abstracto de estudiantes universitarios con conocimientos de programación (CAP) y sin conocimientos de programación (SCP), y estudiaron los efectos del entendimiento de programación básica de los fundamentos de programación en el desarrollo de capacidades de pensamiento abstracto en estudiantes de educación superior y de las habilidades cognitivas enlazadas con el pensamiento abstracto. Para

esto, se aplicó estadística descriptiva y no experimental, y la muestra se organizó con 116 estudiantes universitarios que cursan el cuarto ciclo, que estuvieron matriculados en el semestre del 2022 al 2023. Estudió en una universidad pública de Ecuador en el segundo semestre del año. Los resultados mostraron correlaciones positivas, mostrando que la formación en fundamentos de programación se asoció con mejoras significativas en la abstracción, la mejor en la solución de problemas y la creatividad. Estos logros sugieren que la integración efectiva de la programación en los cursos universitarios es importante para fomentar el desarrollo general de las habilidades cognitivas.

A continuación, en el contexto nacional, citaré a Torres (2021), en su tesis sobre el uso de una aplicación portable con uso de realidad aumentada elaborada con Unity y Vuforia para la enseñanza de la materia de ciencia y tecnología en el Colegio América en Lima, manifestó que los estudiantes se vieron perjudicados en sus estudios de materias por la pandemia del Covid-19, dejando un gran número de centros educativos que tuvieron que cerrar sus locales. La investigación se realizó con estudiantes de segundo año de secundaria y buscó como meta analizar el impacto de un programa para dispositivos portables de realidad aumentada en el aprendizaje para la materia de ciencia y tecnología en el Colegio América. Esta investigación fue del tipo aplicada bajo un diseño experimental, la muestra de trabajo fue con estudiantes de 14 años de secundaria en la cual se recogió información sobre su participación, la solución de problemas y finalmente respuestas de una lista de preguntas de conocimientos en forma virtual. Al procesar la información, los resultados que se lograron en los indicadores fueron un aumento del 12,24% de mejora de intervención. El indicador obtuvo un aumento del 22,39% en el porcentaje de solución de problemas y la media del indicador sobre el rendimiento académico se incrementó en un 3,64. Se concluyó que la aplicación de realidad aumentada si mejora de manera significativa el entendimiento de los temas del curso de Ciencia y Tecnología, con los valores que se obtuvieron con los resultados y se mejora el rendimiento de los indicadores establecidos.

Por otro lado, Mendoza Altez (2021), en su investigación sobre el uso de la realidad aumentada en su aplicación de la enseñanza de Ingeniería, con la finalidad de enriquecer el proceso de enseñanza-aprendizaje, tuvo como objetivo de determinar cuánto influye el uso de la tecnología de Realidad Aumentada (AR) para el desarrollo de enseñanza en los procesos de aprendizaje de la Escuela Profesional de Ingeniería Textil y de Confección de la Universidad Nacional de San Marcos de Lima Perú. Se utilizó la técnica de Realidad Aumentada, que mezcla objetos virtuales en un mundo físico logrando un medio digital en tres dimensiones, logrando

aplicaciones que funcionan en computadoras portátiles como smartphone o tabletas. La investigación se ejecutó aplicando un diseño cuasiexperimental contando con un grupo de control y con un grupo experimental. Con estos grupos, se obtuvieron resultados a través de la comparación de medias luego de analizar la distribución de la información, apreciado como un proceso de distribución normal y con un nivel de significancia $\alpha=0.05$. Los participantes del estudio que son estudiantes a quienes se les administró el examen con Realidad aumentada (RA) obtuvieron promedios más altos, con respecto a los participantes que no emplearon la aplicación de Realidad Aumentada, quedando demostrado que el uso de la Realidad Aumentada si influye significativamente en las tareas de enseñanza - aprendizaje en los estudiantes de los cursos de procesos de implementación de textiles.

Cabe destacar que, Cornejo (2022), en su estudio sobre estrategias metodológicas para implementar una serie de habilidades para el desarrollo de programas en estudiantes de Computación y Sistemas en una universidad privada de Lima, tuvo como objetivo implementar una estrategia para aplicar una metodológica que permita desarrollar las habilidades de la creación de programas en los estudiantes, realizando investigaciones cuantitativas y cualitativas sobre el modelo de educación aplicada en la sistematización teórica, triangulación e intervención para cambiar la realidad. Para esto, se tuvo una muestra compuesta por 100 estudiantes de cuatro aulas del Curso de Programación I, que incluyó a tres supervisores, siete profesores y dos especialistas en programación. El proceso analítico logró identificar factores que ponen obstáculos el proceso de enseñanza y aprendizaje: los estudiantes tienen bajos niveles de conocimientos y habilidades de programación, dificultades para aplicar el pensamiento computacional, orientación insuficiente del profesor y cooperación insuficiente de los estudiantes, etc. La información recopilada del instrumento respondió a dos modelos de datos: cualitativos y cuantitativos. Se concluyó que este estudio logra desarrollar estrategias metodológicas para implementar habilidades de desarrollo de programas en estudiantes de programación de una universidad privada de Lima, y se fundamenta en los principios socio pedagógicos, psicológicos, pedagógicos, de programación y curriculares, que es la base científica del estudio integrado en campos académicos.

Con respecto al entorno local, podemos mencionar a Carranza (2019), en su investigación sobre el uso de una aplicación móvil con aplicación de tecnología de realidad aumentada, para mejorar el rendimiento académico de los estudiantes en el curso de anatomía de la escuela de medicina de una Universidad en Lambayeque. Ante esto y sabiendo que un 79.5 % de alumnos poseen un dispositivo móvil, se planteó desarrollar y poner en marcha un aplicativo móvil que

aplique tecnología de Realidad Aumentada que logre que los estudiantes tengan un estudio más entretenido del sistema nervioso aparentando como si estuvieren observando en forma física el sistema nervioso de un ser humano. La investigación tuvo como objetivo general, cambiar de manera positiva las capacidades académicas en los alumnos en las sesiones de anatomía en las escuelas de medicina humana de las universidades privadas que la ofrecen, por medio de la implementación de una aplicación para dispositivos portables que aplica realidad aumentada. Para validar la aplicación móvil se realizó un prototipo con capacidad de uso que permitió comprobar si la solución que se propuso lograba cubrir el objetivo que se planteaba, concluyendo que el uso de la aplicación si logró mejorar de manera importante el reconocimiento correcto de cómo está estructurado el Sistema Nervioso Central – Encéfalo, evidenciado con el grupo experimental con una calificación más elevada con respecto al grupo control, obteniendo un resultado de 38.46 % de alumnos que obtuvieron una puntuación de 20 en el análisis del grupo experimental y un resultado para el grupo control de 0% de alumnos con dicha puntuación; asimismo, se examinó que el grupo del tipo experimental aumentó de forma positiva sus calificaciones.

Además, es relevante mencionar el trabajo de investigación de Guerra (2023) sobre la aplicación de la realidad aumentada para mejorar el proceso educativo de aprendizaje para el curso de Química para el nivel secundario en los centros educativos de la región Lambayeque. La investigación plantea como objetivo general optimizar la enseñanza y aprendizaje de química en este nivel educativo específico, a través de la implementación de un software de realidad aumentada. El enfoque incluyó la creación y uso de fichas con elementos de realidad aumentada para enriquecer la experiencia educativa. Se aplicó un diseño de estudio del tipo no experimental cuantitativo, debido a que se realizó sin manipular deliberadamente las variables y se enfocó exclusivamente en observar los diferentes fenómenos del entorno natural, considerando como es la unión entre las variables. Se consideró como variable independiente la Realidad Aumentada y como variable dependiente el Proceso de enseñanza-aprendizaje en el curso de Química en los centros educativos de menores en Lambayeque a nivel secundario. Estas variables fueron evaluadas a través de un cuestionario físico, utilizando la Escala de Likert para su medición. La población consistió en 200 procesos, mientras que la muestra abarcó 12 procesos que se aplicaron en el curso de Química. Asimismo, para el estudio se aplicó la técnica de la muestra no probabilístico considerando la comprensión y asimilación de los contenidos de la materia de Química. También, se realizó la categorización de los procesos en niveles de fácil, de nivel intermedio y de nivel complejo (abstracto). Finalmente, para realizar la

investigación, se elaboró un software que aplica realidad aumentada, que se elaboró usando el portal web de Vuforia Engine que permite generar las imágenes marcadoras para los objetos de RA, y el software Unity versión 2021. 3. 4f1 para programar la integración de los objetos en 3D. Al realizar las encuestas a los estudiantes y sugerir la introducción del software de realidad aumentada, se logra observar que mejora la calidad en la aplicación del proceso de enseñanza-aprendizaje de la materia de química en escuelas secundarias de Lambayeque.

Bases Teóricas

En cuanto a la base teórica que fundamenta esta investigación, según Innovae (2024) define que la Realidad Aumentada (RA), también conocida como Augmented Reality en inglés, consiste en una tecnología que tiene la capacidad de mejorar la apreciación y la interrelación con el entorno físico, ofreciendo al usuario una experiencia real enriquecida con datos generados por computadora. Así, la realidad real se mezcla con elementos de existencia virtual, creando una realidad hibrida en tiempo real. Objetos virtuales de doble dimensión y de tres dimensiones se sobreponen al entorno existente; el resultado generado implica la convivencia de dos realidades, la virtual y lo existente, en un único lugar. Es importante mencionar que la RA no reemplaza el mundo existente a uno lógico virtual; por el contrario, sostiene el entorno real que el usuario experimenta, agregando datos virtuales sobre él. El usuario siempre está en contacto con su entorno, pudiendo interactuar con la información virtual que se encuentra entretejida en él.

Rendimiento académico

De acuerdo con Rubiano & Martinez (2024), el desempeño académico surge de la dedicación y habilidad para trabajar del estudiante. Del tiempo dedicado al estudio, al reto y la práctica para mejorar su concentración. Asimismo, el desempeño académico, entendido como la manera específica o concreta del desempeño en la escuela, se refiere a los objetivos alcanzados por los participantes que se muestra en la demostración de sus habilidades cognitivas que han adquirido en la implementación de la aplicación con el proceso de enseñanza y del proceso de aprendizaje en un año escolar.

Aplicación de la TIC como complemento de desarrollo de enseñanzas

Según Gavilanes *et al.* (2019) manifiestan que el uso de las TIC brinda la oportunidad de colaboración que altera una situación pasiva de los estudiantes a una situación constante, a una situación de buscar y volver a plantear contenidos y procesos. Se incrementa la participación

de los estudiantes en sus deberes y fomentan su creatividad y el trabajo colaborativo, permite el acceso a nuevos entornos y situaciones, debido a que están constantemente obligados a hacer "pequeñas" elecciones, a evaluar datos y a elegir y decidir.

La realidad aumentada aplicada en la enseñanza-aprendizaje en la educación superior

La integración de nuevos medios tecnológicos en los centros de estudio, junto con el crecimiento del uso de los dispositivos portátiles en la comunidad, ubica a la RA en un lugar de liderazgo. Así tenemos, que en una publicación de la revista Time del 2010, la incluyó en el cuarto puesto del top diez de las tendencias tecnológicas de ese año. En el ambiente educativo, la RA es un soporte de aplicación tecnológica eficaz, por la forma como los estudiantes observan la realidad existente, ya que logra que se difunda en sus diferentes dimensiones para hacer más fácil la captación de sus diversas características. La RA tiene la capacidad de otorgar destrezas de aprendizaje más contextualizadas del aula, estableciendo conexiones entre los que es real y la situación de querer aprender de los participantes creando un ambiente académico motivador. Así tenemos, por ejemplo; que estudiantes de las carreras de arqueología, historia o antropología pueden tener programas que les permitan reconstruir sitios históricos, mediante mapas y dibujos tridimensionales que recrearán la diversidad de los momentos de la historia. Ante esto, destacan proyectos realizados en España, en el centro superior Pompeu Fabra y Rovira i Virgili, en la ciudad de Cataluña y la Universidad Politécnica de la ciudad de Valencia, en donde un grupo de investigadores denominado Future Lab han logrado desarrollar un modelo de RA que accede a restablecimientos lógicos virtuales de una serie de monumentos importantes, tomando imágenes en tres dimensiones, que están dispuestos para teléfonos portátiles y computadoras.

La importancia de la metodología de aprendizaje visual

El estudio visual se conceptúa como un tipo de aprendizaje que se ejecuta a partir de los estímulos visuales; que incluyen un paquete de distribuidor gráfico para asistir a los estudiantes, cuando trabajan en la búsqueda de ideas y de conceptos, a mejorar su análisis y lograr aprender más eficazmente. Asimismo, estos nos facultan el identificar malentendidos y observar modelos de combinaciones en la información, que son elementos básicos para el entendimiento profundo y la adquisición de conocimientos. Asimismo, el aprendizaje visual es una creciente forma didáctica basada en la aplicación de organizadores gráficos de información para lograr un aprendizaje más efectivo de los estudiantes, Ruiz (2020).

Además, hay diversas técnicas de aprendizaje visual, como por ejemplo los mapas conceptuales, diagramas de flujo, dibujos animados y animaciones. Estos son las técnicas más utilizadas para el conocimiento visual. Mediante la aplicación de métodos de aprendizaje que se basan en el uso de imágenes, permite que los alumnos aclaren sus ideas, fortalecer su entendimiento, integrar nuevas ideas e identificar conceptos erróneos.

Lo relevante del uso de la realidad aumentada en actividades educativas

Si bien Reinoso (2012), acepta el gran poder de las herramientas que nos ofrece la RA, expresa su desconfianza sobre su implementación como herramientas complementarias de aprendizaje en el aula, ya que considera a esta implementación como algo complejo. Sin embargo, manifiesta que hay un motivo de razón fundamental en la educación que parece garantizarlo, de tal manera que Reinoso manifiesta que "muchas investigaciones manifiestan que la RA mejora los métodos de aprendizaje y aumenta el entusiasmo para aprender".

Asimismo, Adell y Castañeda (2012) han definido la pedagogía emergente como: “un conglomerado de guías e ideas pedagógicas que nacen en torno a la aplicación de las TIC en la educación y que buscan usar todas las ventajas comunicativas, informativas, colaborativas, interactivas, creativas e innovadoras de la educación encuadrado en una corriente del aprendizaje.

Para Billinghurst (2002), escritor de la primera escritura de realidad aumentada denominado “El Libro Mágico”, nos otorga buenas razones para comprender que la realidad aumentada puede ser muy útil en educación: ya que “favorece la interacción fluida entre los ambientes reales y virtuales. Lo metafórico de la articulación para el manejo de objetos y la capacidad de moverse sin problemas entre lo existente y la virtualidad".

También Roussou (2004), señala que existe un amplio consenso de los docentes sobre la necesidad de la mejora de métodos de inducción de conocimientos y sigue la idea de G. R. Amthor que "la gente recuerda el 75 % de lo que observa, escucha y realiza". Opina que está claro que la tecnología AR facilitará el proceso de aprender de los estudiantes, entre otras cosas, por el alto nivel de interacción que ofrece.

Realidad Aumentada

La tecnología de la realidad aumentada (AR) es considerada una práctica que interactúa un entorno del mundo existente en el que las cosas que se ubican se observan de forma mejorada por la información perceptiva generada por una computadora, dichas mejoras pueden ser

sensoriales, las cuales incluye visuales, auditivas, hápticas, somatosensoriales y olfativas. La realidad aumentada (AR) es también conocida como realidad mixta (MR) (Sabry, 2022).

Por otro lado, Cabero, De la Horra & Sanchez et al. (2018), afirman que la realidad aumentada, es una aplicación de tecnología que combina al mismo tiempo información digitalizada e información física por medio de distintos soportes tecnológicos, que pueden ser los dispositivos móviles, mismos que permiten crear una nueva realidad mejorada de objetos.

Entre las características de la realidad aumentada, tenemos:

- **Es una combinación de realidades:** Los contenidos producidos por la realidad aumentada se encuentran en uno de los puntos estables entre lo virtual y lo real, lo que crea una realidad mixta donde lo tangible se suma a lo que se observa de los elementos digitales que se fusionan. Esta fusión puede manifestarse mediante la superposición de objetos, la inclusión de elementos o situaciones del entorno que rodea o el fondo.
- **Proporcionar variados grados de información en formato digital:** La implementación de la tecnología de realidad aumentada facilita la combinación de diversos elementos digitales, como texto, imágenes, modelos tridimensionales, video, sonido, sitios web, y más.
- **Oportunidad de interacción:** La información digital generada mediante el uso de realidad aumentada se compone de objetos que posibilitan a los usuarios interactuar con ellos.
- **Modificar o ampliar los datos significativos de la realidad tangible que se integra:** La creación de diseños para recursos de realidad aumentada que superponen capas de datos digitales en entornos físicos no implica que se deba mejorar la comprensión semántica o intencional de la información derivada de la percepción del mundo real.
- **Es necesario que la persona participe para crear un efecto de Realidad Aumentada:** La implementación y visualización de un objeto de realidad aumentada requiere la participación del usuario, lo que contribuye a facilitar su uso cuando este necesita aplicarlo.

Por otro lado, los elementos de la realidad aumentada, de acuerdo con Mozas Fenoll, para incorporar elementos lógicos virtuales en la percepción de un ambiente real, son:

- **La cámara:** Se trata del equipo que captura imágenes del entorno real. Como la cámara instalada en la computadora o la que se encuentra en el teléfono móvil o en la tablet.
- **El procesador**: Es la parte física del computador que va unir la imagen de alta resolución con los datos que se deben añadir.

- **El marcador:** Es un área donde se mostrarán las imágenes generadas por el procesador y visualizará el objeto tridimensional.
- **El software**: Es el algoritmo que contiene las instrucciones específicas que controla el procedimiento.
- **La pantalla**: Es el lugar donde se presentan los elementos tanto reales como virtuales.
- **Conectividad a Internet:** Esta conexión se aplica para transferir los datos del ambiente físico a un servidor externo, así como para obtener la información virtual que se ha programado para que se superponga
- **Activador:** Este es el elemento físico que el programa usa para identificar el mundo real y tomar los datos virtuales que se deben añadir. Esto podría ser un código QR, una imagen marcadora, un dibujo, objeto, una señal de tipo GPS emitida por el dispositivo, un dispositivo de realidad aumentada integrada en lentes personales o en lentes de aplicación biónica.

Es primordial tener todos los elementos ya que, si alguno de ellos falta, no se podría ver el efecto de la realidad aumentada (RA).

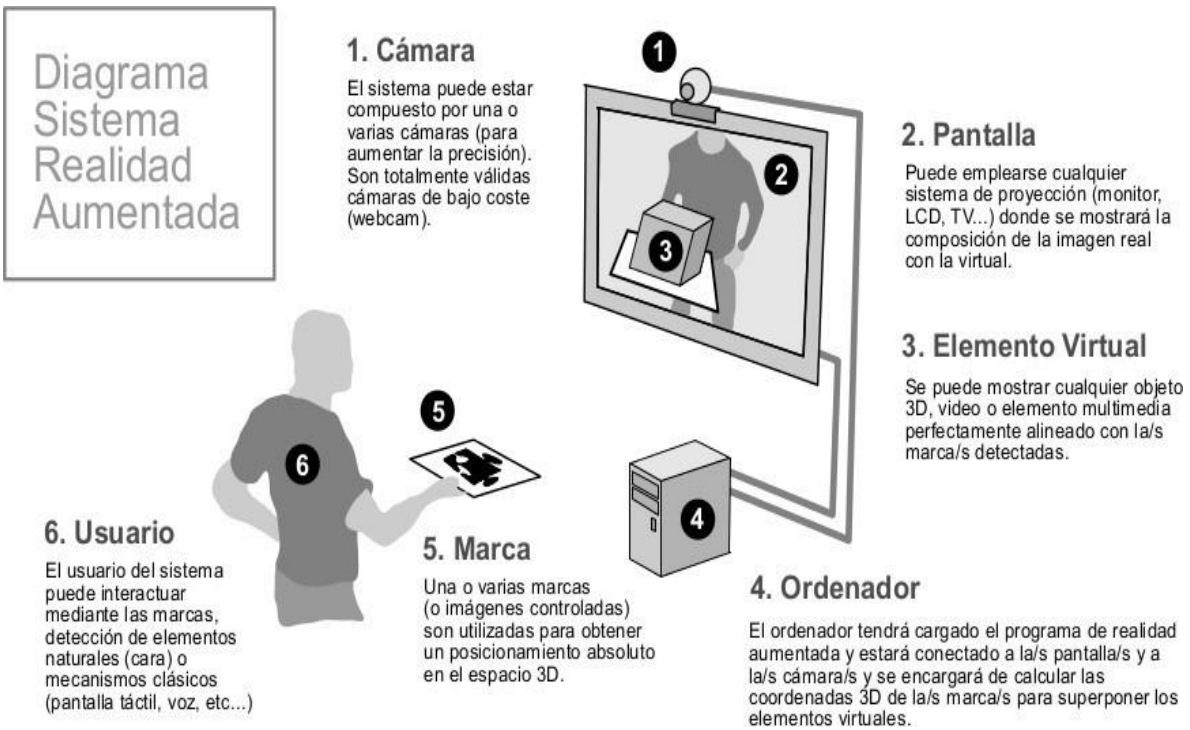


Figura N° 1: Muestra en un diagrama los componentes intervienen para aplicar RA

Fuente: *Componentes que constituyen un sistema de realidad aumentada según AIR Research Group*

Asimismo, entre los niveles de aplicación de la realidad aumentada, según lo manifestado por Prendes Ezpinosa (2015) la RA tiene diferentes niveles, que son los siguientes:

a) Nivel 0: Este nivel es del conectando el mundo real o físico. Se fundamenta en el uso de códigos de barras, códigos bidimensionales (tales como códigos QR) o la identificación de imágenes al azar. Lo significativo de este nivel 0 es que estos códigos sirven como enlaces para otros contenidos, no se realiza registro en 3D ni se realiza rastreo de los marcadores (en esencia, estos elementos de usan vínculos HTML, pero sin requerir escritura).

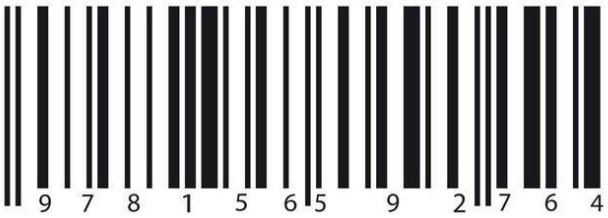


Figura N° 2: Imagen de un código de barras

Fuente: *Desarrollado en Inkscape inspirado en las ilustraciones de Grzexs según VaGla, 2007.*

b) Nivel 1: Realidad aumentada basada en el uso de marcadores. Los softwares que emplean marcadores, estas son gráficos en blanco y negro, con forma cuadrada y representaciones esquemáticas, generalmente para la identificación de patrones en dos dimensiones. La versión más mejor desarrollada de este modelo es que permite reconocer objetos en tres dimensiones.



Figura N° 3: Marker o Indicador basado en AR

Fuente: *Tomado del proyecto llamado Rehabilit-ar.*

c) **Nivel 2:** Realidad Aumentada sin el uso de marcadores. A través de la utilización del GPS y el uso de la brújula de aparatos electrónicos, logramos identificar la ubicación y la dirección, así como vincular POI (puntos de interés) en figuras del mundo existente. Lens-Fitzgerald (2009) lo conceptúa como realidad aumentada que se basa en brújula GPS. Además, agrega la utilización de acelerómetros para efectos con inclinación.



Figura N° 4: Imagen de realidad aumentada utilizando GPS sin aplicar marcador

Fuente: *Google Play, del juego Pokémon GO*

.

d) **Nivel 3:** Aumento del área de visualización, referenciando a Rice (2009): para esto se necesita levantarnos del monitor o pantalla para cambiar a una pantalla más ligera y transparente que se pueda llevar sobre nosotros (como el de los lentes). Una vez que la Realidad Aumentada se transforma en Visual Aumentada (VA), resulta inmersiva. Inmediatamente, el efecto logrado en algo más impactante, contextual y más personal. La experiencia es extremadamente muy buena y transforma todo lo alrededor. Este nivel está aún en etapa experimental.

.



Figura N° 5: uso de lentes personales con realidad aumentada

Fuente: *Miller, Chance, 2017. ARKit será el fundamento del complemento de Apple para lentes inteligentes del tipo 'iGlass'.*

Como herramientas de desarrollo de realidad aumentada, tenemos:

Vuforia: Esta es una plataforma para la creación de aplicaciones de realidad aumentada para sistemas operativos Android y para iOS, creada por el departamento de Investigación y Desarrollo de la ciudad de Qualcomm ubicado en Austria. Esta plataforma se publicó en 2010 y se han creado más de 1.000 aplicaciones con ella. Esta plataforma ha sido empleada por grandes empresas comerciales para sus campañas de publicidad de los productos que ofrecen. Una de las mayores fortalezas de uso de esta plataforma es su capacidad para reconocer marcas más naturales, y pueden incluir objetos en 3D. La existencia de una extensión para el software Unity 3D facilita la creación de escenas no físicas es decir virtuales con efectos animados y muy detalladas. Esta plataforma está disponible en código abierto, aunque hay que mencionar que en su uso con Unity 3D necesita adquirir el registro de licencia correspondiente.

Características relevantes de VUFORIA

- ✓ Identificación del marcador si se oculta parcialmente
- ✓ Identificación del marcador de acuerdo a sus medidas
- ✓ Reconocer marcadores según la alteración de la perspectiva
- ✓ Índice de cuadros o frames cada segundo basado en la cantidad de caras.

Vuforia destaca ya que admite marcas más naturales, seguimiento en 3D y un soporte durante el desarrollo mediante una API. Permite contar con Metaio SDK Mobile que se adquiere con Vuforia. Además, la razón que me motivó por el uso de Vuforia también es la licencia gratuita que ofrece.

Y cómo frameworks, de desarrollo de plataforma móvil con realidad aumentada apliqué:

a) Unity

Unity Technologies proporciona una plataforma para desarrollar juegos y aplicaciones de belleza y atractivo en 2D, 3D, VR y realidad aumentada. Un potente motor gráfico y un editor con todas las funcionalidades te facilitan materializar rápidamente tu visión creativa, y transmitir tu contenido a casi cualquier dispositivo. Es sencillo conectarte a tus públicos en computadoras, consolas, Internet, aparatos móviles, sistemas de entretenimiento en casa, sistemas modulares integrados o monitores con soporte para ser ubicados en la cabeza.

Unity, más que un impulsor, te asiste para lograr el éxito. Te proporciona todo lo que requieres para elaborar contenido de alta calidad, incrementar tu rendimiento y establecer conexión con tu audiencia. Unity Asset Store, Unity Cloud Build, Unity Analytics, Unity

Ads, Unity Everyplay y Unity Certification son algunas de las herramientas y recursos que se incluyen. Unity Tecnologies proporciona servicios a millones de programadores registrados, incluyendo fabricantes destacados, investigadores, estudiantes y seguidores de todas partes del mundo.

Beneficios:

- Permite importar archivos de formatos 3D como archivos creados en 3ds Max, en maya, archivos de Cinema 4D y de Softimage, archivos de Blender, de ZBrush, formatos de archivos FBX y texturas de Adobe Photoshop, archivos con formato PNG, formato TIFF y de vídeos. Estos archivos se mejoran a través de filtros.
- Tiene compatibilidad con API's gráficas del software Direct3D, del programa OpenGL y de Wii.
- Posee una interfaz de opciones desarrollo claramente delimitada que facilita la creación rápida de min- juegos.

Requerimientos para la creación de efectos con Unity:

- Edición de sistema: Windows 10 y 11; Mac OS X 10.9+.
- GPU: Tarjeta gráfica equipada con DX9 o DX11.

Requerimientos extra para la creación de proyectos en Unity:

- iOS: Ordenador Mac con por lo menos OS X 10.9.4 y Xcode 7.0.
- Android: Android SDK y el Kit de Desarrollo Java (JDK).
- Windows Store: Microsoft Windows 10 (64 bits) junto con Visual Studio y su respectiva plataforma SDK.
- El backend de scripts IL2CPP necesita que se instale la función de compilación C++ para el uso con Visual Studio.

b) Blender

Es un software multiplataforma para el modelado, aplicación de iluminación, de renderizado, crear animaciones y generación de gráficos con tres dimensiones. Además de la creación digital mediante el método procesal de nodos, la edición de vídeo, la escultura con topología dinámica y la pintura digital. Además, con Blender es posible crear vídeo juegos gracias a su incorporado motor para juegos interno.

Como características principales de Blender, se tiene que:

- Se trata de un conjunto de herramientas integradas para la creación de contenido en 3D, que incluye un variado arsenal de funciones, como Modelado, Renderizado, Animación, Rigging, Edición de Video, Efectos Visuales, Composición, Texturizado y diversos tipos de Simulaciones.
- Es compatible con múltiples plataformas, ofreciendo una interfaz gráfica OpenGL que se mantiene consistente en los principales sistemas operativos (y es modificable mediante scripts en Python).
- Posee una arquitectura 3D de alto nivel, que facilita un proceso de creación rápido y eficaz.
- Recibe un respaldo activo por parte de la comunidad; puedes visitar blender. org/community para acceder a una amplia lista de recursos.
- Tiene un ejecutable de pequeño tamaño, que se puede usar de forma portátil si se desea.

Entre las utilidades de Blender para la Animación 3D, tenemos: Creación de modelos, escultura digital, aplicación de texturas, pintura UV en los modelos, sistema basado en nodos para texturas y materiales, aumentando la complejidad y el nivel profesional, uso de múltiples texturas, estructura de huesos, sistema de partículas, simulación de océanos, animación no lineal, creación de juegos dentro del entorno, motor de renderizado (Interno y Cycles), edición de video, modificadores y seguimiento de cámara.

Materiales y métodos

Se seleccionó el enfoque cuantitativo para esta investigación, ya que la interrogante a resolver era evaluar cómo el uso de una aplicación educativa con realidad aumentada afecta el desarrollo de habilidades prácticas en fundamentos de programación. Para ello se recolectaron datos numéricos por medio de un cuestionario estructurada y se aplicó un análisis estadístico para medir el desarrollo de habilidades académicas y habilidades cognitivas al aplicar realidad aumentada (Hernandez et al., 2014).

La investigación es del tipo aplicado porque estuvo orientado hacia un objetivo, que fue mejorar el desarrollar habilidades prácticas para el curso de fundamentos de programación aplicando realidad aumentada (Bunge, 1970).

El diseño de la investigación es cuasi experimental, expuesta por Hernandez et al. (2014), debido aque el objetivo del estudio fue observar como el uso de una aplicación móvil educativa que tiene como base la tecnología de realidad aumentada mejora el desarrollo de las habilidades prácticas de fundamentos de programación en estudiantes de educación superior.

Asimismo, según el método de investigación propuesto por Hernández et al. (2014), se utilizó el método de diseño con solo post-test y grupo de control para contrastar la hipótesis. El uso de este diseño abarcó a dos grupos: uno fue sometido al tratamiento experimental y el otro no fue sometido al tratamiento experimental (el grupo de control). En resumen, el manejo de la variable independiente alcanzó únicamente dos niveles: la presencia y la ausencia.

Tabla 1

Diseño de investigación cuasiexperimental

GE	O1	X	O2
GC	O3	-	O4

Fuente: Hernández Sampieri (2014)

Donde:

GE: Grupo experimental

O1: Observación en el pre-test sobre el GE

X: Aplicación del experimento

O2: Observación en el post test sobre el GE

O4: Observación en el post test sobre el GC

Variables y operacionalización

La variable independiente, implementación de aplicativo educativo con realidad aumentada, hace referencia a una tecnología que nos brinda la posibilidad de superponer capas de información visual sobre nuestro entorno real. Esta información puede ser visualizada en distintos dispositivos, como los teléfonos móviles.

Como variable dependiente se incluyó desarrollo de habilidades prácticas de fundamentos de programación, referida al uso los conocimientos adquiridos para abordar y resolver distintas tareas.

Tabla 2

Matriz de consistencia

Problema	Objetivos	Hipótesis	Variables y dimensiones	Metodología
General: ¿Será posible que la implementación de un software de aplicación utilizando la técnica de la realidad aumentada permitirá la mejora de habilidades prácticas de los estudiantes del curso de Fundamentos de Programación en educación superior? **Específico** ¿Cuáles son las habilidades prácticas del estudiante matriculado en el curso de Fundamentos de Programación?	**General** Determinar el impacto de la realidad aumentada en el desarrollo de habilidades prácticas para el curso de Fundamentos de Programación, cuando se usa una aplicación educativa que utilizando realidad aumentada incrementa el aprendizaje de conceptos de fundamentos de programación **Específicos** • Identificar las habilidades prácticas del estudiante matriculado en el curso de	La implementación de la realidad aumentada en el curso de fundamentos de programación mejora significativamente el desarrollo de habilidades prácticas de los estudiantes en comparación en comparación con métodos de enseñanza tradicional	**Variable dependiente** Desarrollo de habilidades prácticas de fundamentos de programación **Dimensiones** • Comprensión de conceptos básicas de programación • Aplicación de estructuras de programación • Solución de problemas mediante programación	**Tipo de investigación – diseño** **Tipo:** Aplicada **Nivel:** Descriptivo **Diseño:** Cuasi experimental **Población y muestra** **Población y muestra:** Docente y alumnos del curso de fundamentos de programación de educación superior **Muestreo:** Del tipo no probabilístico

¿Cómo mejorar las experiencias de aprendizaje para el curso de Fundamentos de Programación? ¿Cuáles serían las características de la propuesta para determinar el efecto mediático de la realidad aumentada en el desarrollo de habilidades prácticas para el curso de Fundamento de Programación?	fundamentos de programación, • Diseñar y desarrollar experiencias de aprendizaje basado en realidad aumentada para el curso de Fundamentos de Programación, haciendo uso de un aplicativo educativo móvil desarrollado en Unity con imágenes en 3D • Evaluar el efecto mediático de la realidad aumentada en el desarrollo de habilidades prácticas para el curso de Fundamento de Programación.	**Variable independiente** Implementación de una aplicación educativa con Realidad Aumentada	**Técnicas e instrumentos** **Técnica:** Examen **Instrumento:** Cuestionario con preguntas cerradas, Test **Tratamiento estadístico:** Hoja de cálculo Microsoft Excel 2019 Jamovi 2.6.26

Tabla 3

Operativización de la variable objeto de estudio

Variable dependiente	Dimensiones	Indicadores	Ítems
Definición conceptual: Según Requena (1998), el desempeño académico surge de la dedicación y habilidad para trabajar del estudiante. De tiempo dedicado a estudio, la competencia y la práctica para mejorar la concentración. **Definición operacional:** Grado de desarrollo de habilidades prácticas para el curso de Fundamentos de Programación. Roussou (2004), señala que existe un amplio acuerdo de los educadores sobre la necesidad de mejorar la interrelación en la enseñanza y se enruta en el argumento de G. R. Amthor que "la gente recuerda el 75 % de lo que ve, oye y hace". Opina que está claro que la tecnología AR mejora el	Comprensión de conceptos básicas de programación.	Identifica las principales estructuras de un programa	¿Cuáles son las tres principales estructuras que conforman un programa básico en la mayoría de los lenguajes de programación? ¿Qué hace una estructura condicional en un programa? ¿Qué es una función en programación y por qué es importante usarla? ¿Cuál es el propósito de un ciclo de repetición en un programa?
		Realiza preguntas a partir de la identificación de las estructuras de un programa	¿Cuál de las siguientes preguntas podría realizarse para entender mejor la estructura del programa? ¿Qué pregunta podría hacer un estudiante para investigar el propósito de esta función? ¿Cuál de las siguientes preguntas refleja un análisis adecuado del uso de variables? ¿Cuál de las siguientes preguntas es pertinente para comprender la estructura del programa?

desarrollo de habilidades prácticas en el aprendizaje de los estudiantes, por el alto grado de interacción que ofrece indicadores plasmadas. Como instrumento de medición se tiene un Cuestionario de 20 preguntas con una escala Likert de medida de: Sobresaliente (20-15), Suficiente (14 – 13), Básico (12 – 11) y Deficiente (10 – 0).	Aplicación de estructuras de programación	Describe correctamente las estructuras de un programa	¿Cuál es la descripción más precisa para este fragmento de código? ¿Qué descripción explica mejor lo que hace este código? ¿Cuál es la descripción correcta de este fragmento? ¿Cuál de las siguientes descripciones es la más precisa para este código?
	Solución de problemas mediante programación	Participa voluntariamente.	¿Cuántas veces participas voluntariamente en una clase de programación típica? ¿Qué tipo de intervenciones sueles realizar cuando participas en clase? ¿En qué situaciones te sientes más cómodo(a) participando en clase? ¿Cómo crees que perciben tus compañeros y el profesor tu participación en clase?
		Explica correctamente la funcionalidad de un programa	¿Cuál es la explicación correcta de lo que hace este programa? ¿Qué funcionalidad describe mejor este programa? ¿Cuál es la funcionalidad de este programa? ¿Qué descripción explica mejor la funcionalidad de este programa?

Tabla 4

Tabla de operacionalización de la variable independiente

VARIABLE	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	OBJETIVOS	CONTENIDOS	TIEMPO
Implementación de un aplicativo educativo con realidad aumentada	Es el proceso de diseño, desarrollo, integración y uso didáctico de una herramienta digital interactiva basada en realidad aumentada, destinada a mejorar la enseñanza-aprendizaje en un entorno educativo, mediante recursos visuales, tridimensionales y multimedia, que combinan el mundo real con elementos	Se refiere al uso de un aplicativo móvil educativo desarrollado en Unity, con soporte para imágenes en 3D y RA, que será utilizado en el curso de Fundamentos de Programación, integrando contenidos curriculares en escenarios interactivos para desarrollar habilidades prácticas.	• Diseñar y desarrollar experiencias de aprendizaje basado en realidad aumentada en el curso de Fundamentos de Programación. • Evaluar el efecto mediático, de los contenidos visuales y multimedia, de la RA en el desarrollo de habilidades prácticas	Sesión 1: (Sintaxis básica y control de flujo) Sesión 2: (Funciones) Sesión 3: (Estructuras de datos) Sesión 4: (Manejo de errores y excepciones) Sesión 5:(Introducción y profundización en OOP) Sesión 6: (Algoritmos de búsqueda y ordenación) Sesión 7: (Recursión) Sesión 8: (Manejo de archivos) Sesión 9: (Trabajo colaborativo y desarrollo de proyectos)	6 semanas (2 sesiones por cada semana)

	virtuales en tiempo real.	Sesión 10: (Optimización de código) Sesión 11: (Presentación del proyecto final) Sesión 12: (Evaluación final y revisión)	

La población de este estudio se conformó con estudiantes de instituciones educativas de nivel superior de la carrera de Desarrollo de Sistemas de Información que cursaban el primer ciclo y que llevaban el curso de Fundamentos de Programación (100 alumnos) donde los criterios aplicados para su inclusión fueron: que sean alumnos de ambos sexos, que tengan la disposición de participar en el estudio y matriculados en el curso fundamentos de programación de una institución privada de educación superior. Asimismo, como criterios de exclusión se consideró a los alumnos que no otorgaban su consentimiento o que no contaban con la autorización de sus padres, para aquellos que hayan realizado la consulta con ellos. Para la elección de la muestra para el GC (grupo control) y otro para el GE (grupo experimental), se tuvo en cuenta el criterio del docente para el desarrollo de la propuesta, aplicando muestreo no probabilístico de tipo intencionado o según criterio del investigador (Sánchez et al., 2018)

Para esta investigación se determinó como muestra a un total de 30 estudiantes, que oscilaban entre las edades de 16 a 23 años y cumplían con los criterios mencionados en la población.

Los indicadores se midieron mediante un cuestionario físico (Anexo 2), y la interpretación de la puntuación fue medido mediante los baremos con escalas del 0 al 10 que corresponde a Deficiente; del 11 al 12 a Básico; del 13 al 14 que se consideró Suficiente y del 15 al 20 Sobresaliente (Anexo 5); y dieron como resultado la relación entre la Realidad Aumentada y el desarrollo de habilidades prácticas en el curso de fundamentos de programación en estudiantes de educación superior,

En esta investigación se consideró al muestreo de forma no aleatoria, ya que fueron grupos establecidos por el criterio del docente, con los alumnos del curso de fundamentados de programación en instituciones educativas superiores.

Las técnicas e instrumentos de recolección de datos utilizados se detallan en la siguiente tabla:

Tabla 5

Técnicas e instrumentos de recolección de datos

Técnicas	Instrumentos	Elementos de la población
Encuesta	Cuestionario con preguntas cerradas	Estudiantes del curso de fundamentos de programación
Examen	Test	Estudiantes

Los instrumentos fueron sometidos a un proceso de validación y de confiabilidad. Esta validez se realizó por criterio de 3 jueces (ver Anexo 3), quienes revisaron según su juicio, la validez

del cuestionario, indicando el cumplimiento de tres aspectos importantes en cada uno: claridad, coherencia y relevancia. Los resultados de éstos fueron valorados mediante la V de Aiken, (Robles, 2018); mientras que, para la confiabilidad**,** ésta se medió sobre el cuestionario, por lo que se efectuó una prueba piloto de cuyos datos se calculó el alfa de Cronbach dada la escala Likert en las respuestas de las interrogantes. Un alfa con valor superior a 0.7 indicó que el instrumento era confiable y pudo ser ejecutado en la muestra del estudio (ver Anexo 4).

Por otro lado, el método de investigación utilizado fue el inductivo-deductivo, ya que aplicó el razonamiento inductivo para analizar los hechos que se observaron al ejecutar los instrumentos y el razonamiento deductivo para verificar que al ejecutar los instrumentos se valida la hipótesis, Suárez (2024). Asimismo, entre los procedimientos que se siguieron fueron en primer lugar la validación de los instrumentos, la determinación de la confiabilidad del cuestionario, el diseño y validación de la propuesta del curso de fundamentos de programación, así como el apoyo al docente en la ejecución los instrumentos del pre test. Al mismo tiempo, se coordinó con los propios estudiantes para obtener su consentimiento voluntario para participar. Acto seguido se implementó las sesiones, las cuales fueron 12 (Anexo 6) y cuyo diseño fue validado por criterio de 3 jueces (Anexo 3), configuradas en la propuesta y con el cual finalizó la evaluación postest.

Tras la obtención de los resultados, el tratamiento se realizó desde dos perspectivas: análisis descriptivo e interpretativo y análisis inferencial. El primer método de organizar datos en tablas y/o figuras se aplicó para determinar las habilidades reales de los estudiantes en un curso de introducción a la programación, ya sea antes o después de la prueba (cuantitativa) y proporcionar detalles de los niveles de habilidades reales presentes, cuando se analiza el desarrollo de las habilidades de los estudiantes en los fundamentos de la programación. Finalmente, se aplicó un análisis inferencial para mostrar estadísticamente el impacto del uso de aplicaciones de RA en la integridad académica; esto contrasta con la hipótesis del estudio, que indica que el uso de la realidad aumentada potencia el desarrollo de habilidades prácticas entre los estudiantes del curso de introducción a la programación. A estos efectos, los datos cuantitativos y la información cualitativa obtenidos a través de las respuestas al cuestionario propuesto fueron organizados y procesados a través de la herramienta Microsoft Excel 2019 (Anexo 5) y Jamovi v. 2.6 (Resultados).

En lo referente a los criterios éticos aplicados durante la investigación se consideró las especificaciones del Informe Bellmont, de la siguiente manera: se tiene, primeramente, la

autonomía, la cual está referido al consentimiento de partes de los estudiantes para participar en la investigación, por lo que se realizó una solicitud para que ellos confirmen su consentimiento y con su firma. También, se aplicaron los criterios de justicia en la que ya se brindó un trato justo y equitativo a los estudiantes participantes; y beneficencia, pues el programa que se aplicó no es de fin maleficio; más bien, buscó beneficiar a los estudiantes de educación superior con respecto a la mejora de desarrollo de habilidades para el curso de fundamentos de programación (Belmont, 1979).

Resultados y discusión

En este apartado se discuten los resultados de la investigación siguiendo la lógica de los objetivos específicos, para esto se utilizó una aplicación de R.A para evaluar el porcentaje las habilidades prácticas, el porcentaje de las experiencias de aprendizaje y el efecto mediático de los efectos visuales y multimedia en la resolución de tareas y el rendimiento académico del curso de Fundamentos de Programación. Por esta razón, se llevó a cabo un pretest para evaluar a los estudiantes y obtener información sobre los indicadores que se están estudiando en este estudio. Luego, el grupo de estudio empleó la aplicación de realidad aumentada y se volvió a evaluar para obtener los resultados postest de los indicadores previamente definidos.

En la presente investigación se ha utilizado la prueba de Shapiro Wilk, ya que el número de participantes en la muestra es 30 y en la prueba de Normalidad de los datos, se obtuvo que no se cumple con la normalidad $p = <.001$; se usó la prueba No Paramétrica de U de Mann-Whitney para las variables de estudio.

A continuación, se presentan los siguientes cuadros de resultados obtenidos en el análisis de los resultados de la aplicación del instrumento y del uso de la aplicación con realidad aumentada, en los grupos Control y Experimental en comparación con las etapas pretest y postest. Este análisis de ha realizado aplicando el software Jamovi 2.6.26.

Tabla 6

Determinar el impacto de la realidad aumentada en el desarrollo de habilidades prácticas para el curso de Fundamentos de Programación

Prueba t para Muestras Independientes

		Estadístico	p	Diferencia de medias	EE de la diferencia
PRETEST_TOTAL	**U de Mann-Whitney**	110	0.914	8.76e-6	
POSTEST_TOTAL	**U de Mann-Whitney**	0	<.001	-5.00	

Nota. H_a $\mu_{Control} \neq \mu_{Experimental}$

En la tabla se muestra que, en la aplicación de los totales referido al objetivo general, se determina que:

- **PRETEST_TOTAL**: No hubo diferencias significativas entre los grupos Control y Experimental antes de la intervención (p = 0.914).
- **POSTEST_TOTAL**: Existió una diferencia significativa después de la intervención (p < 0.001), con una media mayor en el grupo Experimental (-5.00 de diferencia, indicando superioridad en puntajes).

Tabla 7

Supuestos

Prueba de Normalidad (Shapiro-Wilk)

	W	p
PRETEST_TOTAL	0.943	0.109
POSTEST_TOTAL	0.818	<.001

Nota. Un valor p bajo sugiere una violación del supuesto de normalidad

En la prueba de normalidad se obtiene que:

- **PRETEST_TOTAL**: Cumple con normalidad (p = 0.109).
- **POSTEST_TOTAL**: Violación de normalidad (p < 0.001), lo que justifica el uso de pruebas no paramétricas de U de Mann-Whitney.

Tabla 8

Descriptivas de Grupo

	Grupo	N	Media	Mediana	DE	EE
PRETEST_TOTAL	**Control**	15	11.5	12.0	1.506	0.389
	Experimental	15	10.9	12.0	2.374	0.613
POSTEST_TOTAL	**Control**	15	12.4	12.0	0.828	0.214
	Experimental	15	17.5	17.0	0.516	0.133

Los resultados descriptivos de los grupos Control y Experimental en el pretest y en postest son:

- **PRETEST_TOTAL**: Ambos grupos tenían medianas idénticas (12.0), pero el grupo Experimental mostraba mayor dispersión (DE = 2.374 vs. 1.506).
- **POSTEST_TOTAL**: El grupo experimental tuvo una mediana notablemente mayor (17.0 vs. 12.0) y menor variabilidad (DE = 0.516 vs. 0.828), sugiriendo un efecto consistente de la intervención.

Tabla 9

Recuento Pretest grupo control

Frecuencias de NIVEL_PRE_GC

NIVEL_PRE_GC	Frecuencias	% del Total	% Acumulado
Básico	9	60.0 %	60.0 %
Deficiente	4	26.7 %	86.7 %
Suficiente	2	13.3 %	100.0 %

En el recuento del pretest del grupo control, el resultado de la intervención muestra que la mayoría de los alumnos se ubican en el nivel "Básico" que hacen 60 % del total de la muestra, seguido de "Deficiente" con 26.7 % y un nivel "Suficiente" con 13.3 %

Tabla 10
Recuento postest grupo control

Frecuencias de NIVEL_POS_GC

NIVEL_POS_GC	Frecuencias	% del Total	% Acumulado
Básico	10	66.7 %	66.7 %
Suficiente	5	33.3 %	100.0 %

En el recuento de posintervención del grupo control se obtiene un resultado de persistencia de "Básico" de 66.7 % de la muestra, pero con aparición de "Suficiente" de 33.3 %.

Tabla 11
Recuento pretest grupo experimental

Frecuencias de NIVEL_PRE_GE

NIVEL_PRE_GE	Frecuencias	% del Total	% Acumulado
Suficiente	4	26.7 %	26.7 %
Básico	5	33.3 %	60.0 %
Deficiente	6	40.0 %	100.0 %

En el recuento de preintervención del grupo experimental se obtiene que 6 alumnos se ubican en "Deficiente" que representan 40 % del total, "Básico" de 5 alumnos que hacen 33.3 % del total y un "Suficiente" de 4 alumnos que hacen 26.7 % del total.

Tabla 12
Recuento postest grupo experimental

Frecuencias de NIVEL_POS_GE

NIVEL_POS_GE	Frecuencias	% del Total	% Acumulado
Sobresaliente	15	100.0 %	100.0 %

En el recuento de posintervención del grupo experimental se obtiene que todos los participantes de la muestra alcanzaron el nivel "Sobresaliente" (100 %), evidenciando un impacto drástico de la intervención.

Tabla 13
Interpretación final

Podemos concluir que el aplicativo de realidad aumentada mejora las habilidades prácticas en el curso de fundamentos de programación para los alumnos de educación superior.

Momentos	Grupos	U	Mediana	*p*
Pretest	Control	110	12	0.914
	Experimental		12	
Postest	Control	0	12	<.001
	Experimental		17	

En la Tabla 15, se muestra que, tras la aplicación del programa, en el postest se corrobora un aumento significativo en los puntajes del grupo control y experimental. Por tanto, puede asumirse que dicho programa fue efectivo.

Gráficos de Resultados

Figura Nª 6

Resultado gráfico del recuento de Preintervención del grupo control

Nivel Pretest Grupo Control

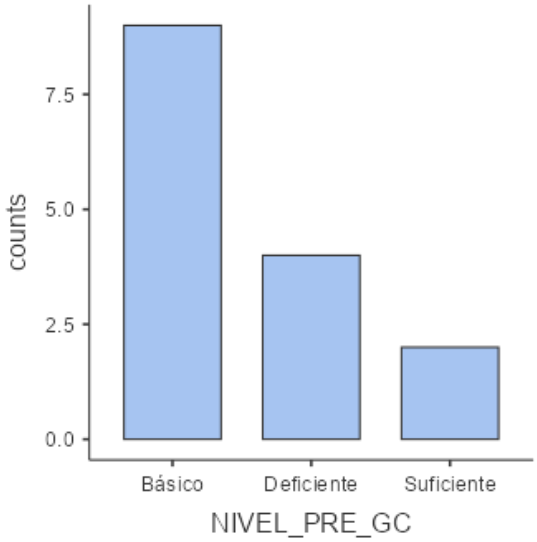


Figura Nª 6

Resultado gráfico del recuento de Posintervención del grupo control

Nivel Postest Grupo Control

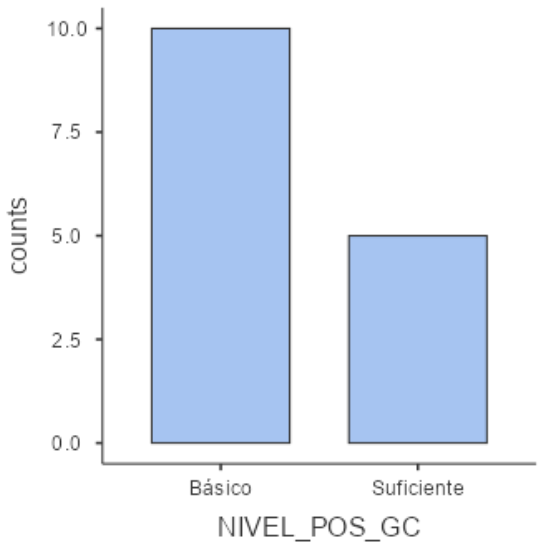


Figura Nª 7

Se observa comparando las **Figuras 6 y 7** que en el Nivel Pretest Grupo Control vs. Nivel Postest Grupo Control, hay una leve mejora (aparición de "Suficiente"), pero sin cambio mayoritario.

Figura Nª 8

Resultado gráfico del recuento de Preintervención del grupo experimental

Nivel Pretest Grupo Experimental

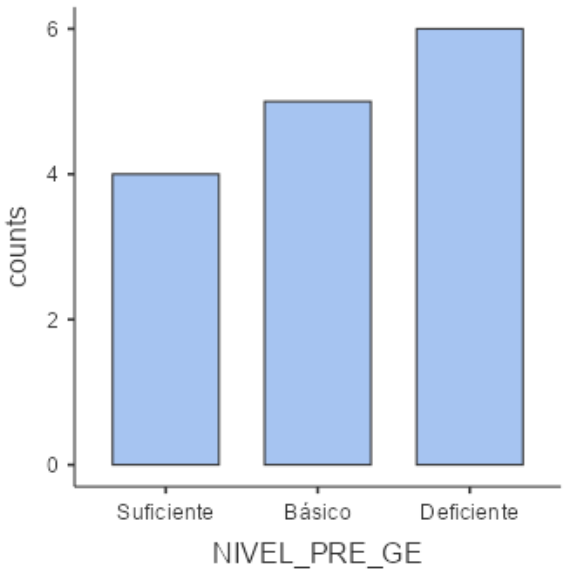


Figura Nª 8

Figura Nª 9

Resultado gráfico del recuento de Posintervención del grupo experimental

Nivel Postest Grupo Experimental

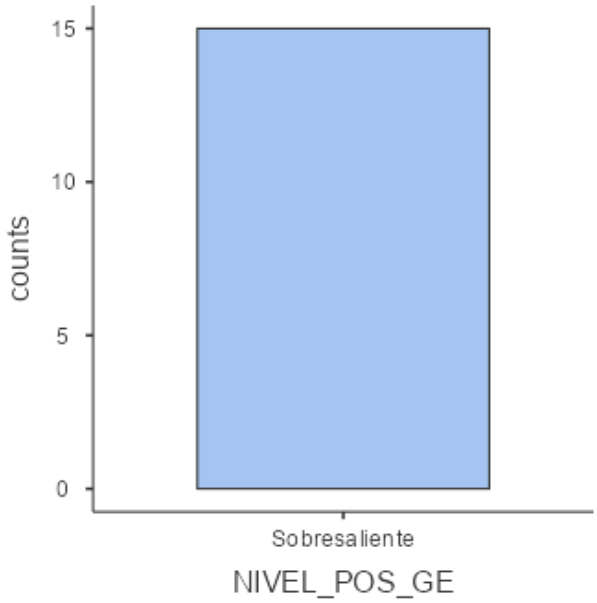


Figura Nª 9

Se observa en las **Figuras 8 y 9** que en el Nivel Pretest Grupo Experimental vs. Nivel Postest Grupo Experimental, hay un cambio radical desde niveles bajos ("Deficiente"/"Básico") a "Sobresaliente" en todos los casos.

Discusión

Los resultados obtenidos en esta investigación, respaldados por los antecedentes y las bases teóricas, demuestran que la implementación de la realidad aumentada (RA) en el curso de Fundamentos de Programación tuvo un impacto significativo en el desarrollo de habilidades prácticas y en el rendimiento académico de los estudiantes de educación superior. A continuación, se discuten estos hallazgos en relación con los antecedentes y las bases teóricas presentadas. A continuación, se analizan los hallazgos clave y su relación con la literatura existente:

- El impacto de la realidad aumentada en el rendimiento académico en los resultados del Post-Test reveló una diferencia significativa entre el grupo experimental (que utilizó la aplicación de RA) y el grupo control, con una mediana de 17.0 frente a 12.0, respectivamente (Tabla 8). Este hallazgo coincide con lo reportado por Acosta (2021), quien encontró que la RA mejora la comprensión de conceptos básicos de programación al proporcionar un medio interactivo y visual. Además, Mendoza Altez (2021) también observó que los estudiantes que utilizaron RA obtuvieron promedios más altos en comparación con aquellos que no la

emplearon, lo que respalda la eficacia de esta tecnología en entornos educativos. La mejora en el rendimiento académico puede atribuirse a las características de la RA, como la combinación de realidades y la interacción en tiempo real (Innovae, 2024), que facilitan un aprendizaje más contextualizado y significativo. Esto se evidencia en el grupo experimental, donde el 100% de los estudiantes alcanzó el nivel "Sobresaliente" después de la intervención (Tabla 14), mientras que en el grupo control solo el 33.3% logró un nivel "Suficiente" (Tabla 10).

- En el desarrollo de habilidades prácticas, la RA no solo mejoró el rendimiento académico, sino que también fortaleció las habilidades prácticas de los estudiantes. Esto se alinea con lo expuesto por Saa-Saltos (2023), quien destacó que la programación básica está asociada con mejoras en la abstracción, la solución de problemas y la creatividad. La aplicación de RA en este estudio permitió a los estudiantes interactuar con conceptos abstractos de programación de manera visual y práctica, lo que facilitó su comprensión y aplicación.

- Además, los resultados coinciden con las bases teóricas que señalan que el aprendizaje visual, como el proporcionado por la RA, ayuda a los estudiantes a clarificar ideas y fortalecer su entendimiento (Ruiz, 2020). La capacidad de la RA para superponer información digital en el entorno real (Cabero Almenara et al., 2018) creó un ambiente de aprendizaje más dinámico y efectivo.

- En la comparación con estudios previos, a nivel internacional, los hallazgos de este estudio son consistentes con los de Calderón (2023), quien resaltó que la RA enriquece los procesos educativos y estimula el aprendizaje profundo. A nivel nacional, Torres (2021) y Cornejo (2022) también encontraron que la RA mejora el rendimiento académico y las habilidades de programación, respectivamente. A nivel local, los trabajos de Carranza (2019) y Guerra (2023) respaldan la eficacia de la RA en contextos educativos específicos, como anatomía y química, lo que sugiere que sus beneficios pueden extenderse a diversas disciplinas.

- La no normalidad en los datos del pretest (Shapiro-Wilk, $p < .001$) justificó el uso de pruebas no paramétricas, un hallazgo común en estudios con RA debido a la heterogeneidad en las habilidades iniciales (Cornejo, 2022). Además, herramientas como Vuforia y Unity, aunque potentes, requieren recursos técnicos y capacitación, como señala Serrano Mamolar (2012). Esto implica que la implementación de RA en entornos educativos debe considerar barreras de acceso y formación docente. Los antecedentes locales (Carranza, 2019; Guerra, 2023)

muestran que la RA es efectiva en entornos específicos (e.g., anatomía, química), pero su éxito depende de adaptaciones contextuales.

- La RA se alinea con la pedagogía emergente descrita por Adell y Castañeda (2012), donde las TIC transforman roles pasivos en activos. Sin embargo, como advierte Reinoso (2012), su implementación debe ser crítica y no solo tecnocéntrica. Los resultados respaldan su uso como complemento, pero no como reemplazo de metodologías tradicionales, especialmente en disciplinas que requieren práctica iterativa (e.g., programación).

Conclusiones

A partir de los resultados obtenidos y en consonancia con los objetivos planteados, se pueden establecer las siguientes conclusiones:

El impacto significativo de la realidad aumentada (RA) en el desarrollo de habilidades prácticas con la implementación de la aplicación educativa basada en RA demostró un efecto positivo y significativo en el desarrollo de habilidades prácticas para el curso de Fundamentos de Programación. Los resultados del Postest revelaron una diferencia estadísticamente relevante entre el grupo experimental (que utilizó la RA) y el grupo control, con una mediana de 17.0 frente a 12.0, respectivamente. Esto confirma que la RA no solo mejora el rendimiento académico, sino que también fortalece la capacidad de los estudiantes para aplicar conceptos abstractos de programación de manera práctica.

La efectividad del diagnóstico inicial (Pretest)**,** ya que el diagnóstico inicial de habilidades prácticas permitió identificar que ambos grupos (control y experimental) presentaban niveles similares antes de la intervención, con medianas idénticas (12.0) pero mayor dispersión en el grupo experimental. Esto validó la necesidad de implementar herramientas innovadoras, como la RA, para nivelar y potenciar las habilidades de los estudiantes.

El diseño y desarrollo exitoso de experiencias de aprendizaje con RA, ya que el aplicativo educativo móvil desarrollado en Unity, con imágenes en 3D y elementos interactivos, cumplió con su objetivo de facilitar el aprendizaje visual y contextualizado. La herramienta aprovechó las características clave de la RA, como la combinación de realidades y la interacción en tiempo real, lo que permitió a los estudiantes explorar conceptos complejos de programación de manera más intuitiva y motivadora.

El efecto mediático y multimedia en el aprendizaje**,** porque los resultados evidenciaron que los contenidos visuales y multimedia integrados en la RA tuvieron un impacto destacado en la retención y comprensión de los conceptos. Esto se reflejó en el 100 % de los estudiantes del grupo experimental alcanzando el nivel "Sobresaliente" en el Postest, respaldando teorías como las de Ruiz (2020) sobre el aprendizaje visual y las de Billinghurst (2002) acerca de la interacción fluida entre entornos reales y virtuales.

La contribución a la literatura y prácticas educativas**,** debido a que los hallazgos de esta investigación se alinean con estudios previos internacionales (Acosta, 2021; Calderón, 2023), nacionales (Torres, 2021); Mendoza Altez, 2021) y locales (Carranza, 2019), consolidando a la RA como una tecnología disruptiva en la educación superior. Además, se validó la viabilidad de herramientas como Unity y Vuforia para crear aplicaciones accesibles y escalables en contextos académicos.

Recomendaciones

A partir de los hallazgos y limitaciones identificadas en la investigación, se proponen las siguientes recomendaciones para futuras implementaciones de Realidad Aumentada (RA) en la enseñanza de programación y otras disciplinas:

Se deben aplicar mejoras en el diseño pedagógico y tecnológico, para esto incrementando la interactividad y personalización, mediante el diseño de experiencias de RA adaptativas que ajusten el nivel de dificultad según el progreso del estudiante e incorporar retroalimentación inmediata y ejercicios prácticos iterativos para reforzar el aprendizaje; así como optimizar los recursos multimedia, utilizando modelos 3D más detallados y escenarios interactivos que simulen problemas reales de programación (ej.: depuración de código mediante RA) e incluir narrativas gamificadas (logros, desafíos) para aumentar la motivación, tal como sugieren estudios previos (Billingurst, 2002; Adell & Castañeda, 2012).

Se propone ampliación la investigación, mediante la realización de estudios longitudinales que permitan evaluar el impacto de la RA en períodos más extensos (ej.: dos semestres completos) para medir su efecto en la retención de conocimientos y habilidades avanzadas; asimismo, emplear muestras más grandes y diversificadas (ej.: diferentes instituciones o niveles educativos) para generalizar resultados; así como también, integrar métodos mixtos para combinar pruebas cuantitativas con entrevistas o focus groups para capturar percepciones cualitativas (ej.: confianza del estudiante) y medir variables adicionales

como creatividad, resolución de problemas y colaboración, clave en programación (Saa-Saltos, 2023).

Se sugiere la capacitación y la accesibilidad en formación docente en RA. Para esto, de deben desarrollar talleres para profesores sobre el uso pedagógico de herramientas como Unity y Vuforia, enfatizando su integración en planes curriculares; asmismo, crear una comunidad de práctica donde docentes compartan experiencias y recursos de RA (ej.: repositorio de marcadores 3D o actividades prediseñadas). También garantizar acceso equitativo de material proporcionando dispositivos compatibles (smartphones, tablets) en instituciones con limitaciones tecnológicas, basado en el éxito de proyectos locales (Carranza, 2019) y ofrecer versiones "ligeras" de la aplicación para entornos con baja conectividad.

Referencias

Acosta, L. (2021). Metodología de la investigación en la educación superior. *Revista Universidad y Sociedad*, 2021, vol.13, n.4, 283-293.

Acuña, J. (2004). *Mejoramiento de la calidad: un enfoque a los servicios* (Primera ed.). Costa Rica: Cartago.

Adell, & Castañeda. (2012). *Tendencias emergentes en educación con TIC.* España.

Ap_Metodologías. (2024). *Ap_Metodologías, Aprendizaje por Descubrimiento*. Obtenido de https://sites.google.com/site/apmetodologias/modelo-cognitivo-constructivista/aprendizaje-por-descubrimiento

Azuma. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*.

Baquerizo, A., & Canales, R. R. (2018). *Desarrollo de videojuego con AUGMENTED REALITY.* Tesis de pregrado, Lima.

Barcenas, J., & Ruiz-Velasco, E. (2021). Innovación Digital Educativa.

Belmont. (18 de abril de 1979). *Observatori de Bioètica i Dret*. Obtenido de https://www.bioeticayderecho.ub.edu/archivos/norm/InformeBelmont.pdf

Billingurst. (2002). *Billinghurst, M. (2002). Realidad Aumentada en la Educación . Seattle, WA: Nuevos Horizontes para el Aprendizaje - Tecnología en la Educación.* Seattle - USA.

Bologna, J. K. (2020). *Sistema de realidad aumentada para el entrenamiento de estudiantes en el manejo de instrumentación HART de la facultad de tecnologías de la información, telecomunicaciones e industrial de la Universidad Técnica de Ambato.* Tesis de pregrado, Ambato.

Bunge, M. (1970). *Filosofía de la ciencia y de la técnica.* La Plata - Argentina.

Cabero, J., De la Horra, I., & Sanchez, J. (2018). *La Realidad Aumentada como herramienta educativa.* Madrid: Ediciones Paraninfo S.A.

Cabero, J., Leiva, J., Moreno, N., Barroso, J., & Lopez, E. (s.f.). *Realidad aumentada y educación.* Barcelona: Octaedro S.L.

Cabero, J., De la Horra, I., & Sanchez, J. (2018). *La Realidad Aumentada como herramienta educativa.* Madrid: Ediciones Paraninfo S.A.

Cairo, O. (2005). *Metodología de la programación, 3ra Edición.* Alfaomega.

Calderón, E. (2023). Estrategias Pedagógicas en el aula y su Influencia en el desarrollo socioemocional de los Estudiantes. *Ciencia Latina Revista Multidiciplinar*, Volumne7, Nro 2.

Carranza, B. (2019). *Aplicación móvil basada en la tecnología de realidad aumentada, para apoyar el rendimiento académico de los estudiantes en el área de anatomía de la Escuela de Medicina de una Universidad en Lambayeque.* Tesis de pregrado, Chiclayo.

Caudell, T. (1992). Reflection. *https://reflectioncreativemedia.com/a-history-of-augmented-reality/*

Colorado, M. (2020). *Compendio y desarrollo de herramiebtas digitales para apoyar la enseñanza-aprendizaje en Ingeniería de Software*. Universidad Atónoma de Yucatán.

Cornejo, E. D. (2022). *Estrategia Metodológica para desarrollar las habilidades de programación en los estudiantes de la escuela de computación y sistemas de una Universidad privada de Lima*.

E. Soloway, J. C. (2013). *Studying the Novice Programmer.* New York: Psychology Press.

Gavilanes, M., Yanza, W., Inca, A., Torres, G., & Sánchez, R. (2019). Las TICs en los procesos de enseñanza y aprendizaje. *Ciencia Digital, 3*(2).

González et al. (2012). *Analysis of subjects with greater difficulty for university students in the area of computer science.* International Journal of Computer Science and Network Security.

Guerra, F. A. (2023). Realidad aumentada, para el proceso de enseñanza - aprendizaje del curso de química a nivel secundario en los colegios de Lambayequ. *Revista peruana de sistemas de computación y sistemas*.

Guerra, F. A. (2023). Realidad aumentada, para el proceso de enseñanza - aprendizaje del curso de química a nivel secundario en los colegios de Lambayeque. *Revista Peuana de Comutaión y Sistemas 5(2):41-5*, 41-51.

Hartmann, N. &. (2001). Kara, finite state machines, and the case for programming as part of general education. *Human-Centric Computing Languages and Environments, 2001. Proceedings IEEE Symposia on.* Stresa, Italy.

Hernandez, R. (2014). *Metodología de la Investigación* (Sexta ed.). McGRAW-HILL.

Innovae. (2024). *Innovae*. Obtenido de https://www.innovae.com/la-realidad-aumentada/

Insuasti. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Educación y Desarrollo Social, 10*(2).

Insuasti, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Educación y Desarrollo Social*.

Jiménez-Toledo et al. (2019). *El trabajo colaborativo como estrategia didáctica para la enseñanza/aprendizaje de la programación: una revisión sistemática de literatura.* Instituto Tecnologico Metropolitano.

Joyanes, L. (2003). *Fundamentos de Programación.* Mc Graw Hill.

Keepcoding. (2023). Keepcoding Tech School: https://keepcoding.io/blog/tipos-de-realidad-aumentada/

Mamolar, A. S. (2012). *Herramientas de desarrollo libres para aplicaciones de realidad aumentada con Android. Análisis comparativo entre ellas.* Valencia, España: Universidad Politécnica de Valencia.

Mendoza, E. A. (2021). *Aplicación de la Realidad Aumentada en la enseñanza de Ingeniaría, para mejorar el proceso de enseñanza-aprendizaje.* lima: Univercidad Nacional Federido Villarreal.

Mimenza, O. C. (15 de Junio de 2018). *Las teorías del proceso dual: qué son y cómo explican la mente humana*. Psicología y Mente: https://psicologiaymente.com/psicologia/teorias-del-proceso-dual

Mis herramientas tic digitales. (s.f.). Mis herramientas digitales-aplicaciones y recursos TIC para el aula: http://misherramientasticdigitales.blogspot.com/2015/11/quiver.html

Montoya, A. F., & Rios Abarca , J. G. (2018). *Aplicación móvil basada en realidad aumentada para el marketing empresarial del centro comercial APIAT, 2018.* Tesis pregrado, Trujillo.

Narváez, L., Escalante, M., Gonzáles, C., Miranda, C., & Canché, M. (2023). Propuesta metodológica para mejorar el desempeño académico de los estudiantes en fundamentos de programación. *Revista Iberoamericana para la Investigación y el Desarrollo Educativo, 14*(27 de Julio 2023), 19.

Navarro, F., Martinez, A., & Martinez, J. M. (20218). *Realidad Virtual y Realidad Aumentada - Desarrollo de Aplicaciones.* Madrid: RA-MA Editorial.

Ojeda Peña , C. P. (2019). *Diseño e implementación de una aplicación móvil como medio de rehabilitación para trastornos psicológicos basándose en herramientas de realidad aumentada.* Tesis de pregrado, Pimentel.

Overlap. (s.f.). 8 sectores en los que aplicar la Realidad Aumentada: https://www.overlap.net/blog/8-sectores-los-aplicar-la-realidad-aumentada-ar-la-realidad-virtual-vr-2020/

Pinales, F., & Velásquez, C. (2014). *Problemario de Algoritmos.* Universidad Autònoma de Aguas Calientes.

Reinoso, J. (2012). *Aplicación de un ambiente virtual de aprendizaje orientado a la formación empresarial.* Cali, Colombia: Universidad ICESI.

Rey, C. A., & Quijano, M. S. (2020). *Implementación de un sistema de realidad aumentada (AR) con Unity que permita obtener información sobre artefactos históricos del Museo Guane de la UNAB, utilizando reconocimiento de imágenes.* Tesis pregrado, Bucaramanga.

Calderón, M. E. (2023). Realidad virtual y aumentada en la educación superior: experiencias inmersivas. *Religación*.

Rubiano, S., & Martinez, J. (2024). El desempeño académico como un comportamiento en el proceso de enseñanza aprendizaje. *Ciencia Latina Internacional, 8*(2).

Ruiz, L. (12 de febrero de 2020). *Psicología y Mente.* Obtenido de https://psicologiaymente.com/desarrollo/aprendizaje-visual

Saa-Saltos, J. (2023). Incidencia de los Fundamentos de Programación en el pensamiento abstracto de estudianes universitarios. *Revista de la Universidad Técnica estatal de Quevedo, Ecuador*.

Sabry, F. (2022). Realidad Aumentada. En F. Sabry, *Realidad Aumentada.*

Salvador, G. d. (2020). *Guía turística en aplicación móvil con realidad aumentada del Paseo de las Musas.* Tesis de pregrado, Lambayeque.

Sanchez, M. S. (2022). *Actitud hacia la intención de uso de la realidad aumentada en museos de Lambayeque por los visitantes nacionales, 2021.* Tesis de pregrado, Chiclayo.

Sommerville, I. (2005). *Ingeniería de software.* Madrid: Pearson Educacuon S.A.

Suarez, A., & Escobar, Y. A. (2019). *Desarrollo de una aplicación de realidad aumentada para la ubicación dentro del Tecnológico de Antioquia.* Tesis pregrado, Medellín.

Suárez, E. (9 de febrero de 2024). *Experto Universitario*. Obtenido de https://expertouniversitario.es/blog/metodo-inductivo-y-deductivo/#:~:text=El%20m%C3%A9todo%20inductivo%20se%20basa,una%20teor%C3%ADa%20o%20hip%C3%B3tesis%20previa.

Torres, M. (2021). *MINEDU - Encuesta Nacional de Estudiantes de Educación Superior Universitaria 2019.* Lima.

Venegas , A. M., & Sernaqué, J. A. (2020). *Aplicación móvil con realidad aumentada para mejorar el nivel del logro de aprendizaje de Ciencia y Ambiente en estudiantes de una Institución Educativa.* Tesis pregrado, Trujillo.

Villar, B. (3 de octubre de 2024). *Teoría del Aprendizaje Experiencial de Kolb.* Obtenido de Espacio de Liderazgo: https://liderazgo.space/teoria-del-aprendizaje-experiencial-de-kolb/

Anexos

Anexo 1: Autorizaciones para la aplicación del cuestionario

Chiclayo, 15 de abril del 2025.

Mg. Ing. Francisco Pisfil Custodio

Coordinador ETIA-SENATI Zonal Lambayeque

Ciudad

Asunto: Solicitud de autorización para realizar investigación académica

Estimado Coordinador:

Reciba un cordial saludo. Mi nombre es Hans Edgardo Delgado Díaz, estudiante de la Maestría en Informática Educativa y TIC en la Universidad Católica Santo Toribio de Mogrovejo. Me dirijo a usted con el fin de solicitar formalmente la autorización para llevar a cabo un trabajo de investigación en su prestigiosa institución.

La investigación titulada **"Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación"** tiene como objetivo **determinar el impacto de la realidad aumentada en el desarrollo de habilidades prácticas para el curso de Fundamentos de Programación, mediante la implementación de una aplicación educativa que utilizando realidad aumentada mejora el aprendizaje de conceptos de fundamentos de programación.** Esta está desarrollada en el marco del curso de Seminario de Tesis II y forma parte de los requisitos académicos del trabajo de investigación para optar el grado de maestro y ha sido debidamente aprobada por mi institución académica.

Para el desarrollo del estudio, se requiere la participación de **estudiantes del 3er ciclo de la carrea de Ingeniería de Software con Inteligencia Artificial que ofrece su representada**, así como la aplicación de un **cuestionario** y, posteriormente, la implementación de un plan pedagógico de intervención que permita mejorar el **desempeño académico** de los participantes. Cabe mencionar que, se garantiza en todo momento la confidencialidad, el anonimato y el respeto hacia los participantes. Además, esta investigación no interferirá con las actividades cotidianas de la institución educativa.

Agradezco de antemano su atención y quedo atento a cualquier información adicional que requiera para considerar esta solicitud.

Sin más por el momento, me despido cordialmente.

Atentamente,

Ing. Hans Edgardo Delgado Díaz

DNI: 16658525

Francisco Pisfil Custodio
COORDINADOR ETIA
DZ LAMBAYEQUE

Anexo 2: Instrumento de diagnóstico con las sugerencias de los expertos

DISEÑO DE INSTRUMENTOS PARA LA RECOLECCION DE DATOS

Lugar:

Fecha:

Objetivo: Determinar el impacto de la Realidad Aumentada en el desarrollo de habilidades prácticas para el curso de Fundamentos de Programación, mediante la implementación de una aplicación educativa que aplicando realidad aumentada mejora el aprendizaje de conceptos de fundamentos de programación en educación superior

Instrucciones:

Con la finalidad de saber el nivel de sus habilidades prácticas en el curso de fundamentos de programación con el uso de Realidad Aumentada, se solicita de la manera más atenta su cooperación. La información proporcionada será anónima. Marcar con X las respuestas que correspondan a cada pregunta.

Indicador: Identifican las principales estructuras de un programa

Nro	Item
01	**¿Cuáles son las tres estructuras fundamentales que componen un programa básico en la mayoría de los lenguajes de programación?** A) Declaración de variables, Ciclos, Funciones B) Declaración de variables, Condicionales, Funciones C) Variables, Tipos de datos, Clases D) Funciones, Entrada/Salida, Excepciones
02	**¿Cuál es la función de una estructura condicional dentro de un programa?** A) Permite la repetición de un conjunto de instrucciones de manera indefinida. B) Permite seleccionar entre diferentes opciones de ejecución según una condición específica. C) Permite organizar el código en distintas funciones independientes. D) Permite gestionar la entrada de datos por parte del usuario.
03	**¿Qué es una función en programación y cuál es su importancia en el desarrollo de programas?** A) Es una variable que se declara fuera del bloque principal y facilita la reutilización de código. B) Es un bloque de código que se ejecuta solo una vez al inicio del programa. C) Es un bloque de código que puede recibir parámetros y devolver un valor, promoviendo la reutilización y modularidad del código.

	D) Es un conjunto de declaraciones que deben ejecutarse secuencialmente en el programa.
04	**¿Cuál es el propósito de un ciclo de repetición en un programa?** A) Permite que el programa se ejecute solo una vez por cada instrucción. B) Permite repetir un conjunto de instrucciones varias veces bajo ciertas condiciones. C) Permite modificar los valores de las variables durante la ejecución. D) Permite finalizar el programa inmediatamente después de una instrucción.

Indicador: Realizan preguntas a partir de la identificación de las estructuras de un programa

Nro	Item

01	**Estructuras de control**
	
	Dado el siguiente fragmento de código en Python:
	
	x = 5
	if x > 3:
	  print("x es mayor que 3")
	else:
	  print("x es menor o igual a 3")
	
	¿Cuál de las siguientes preguntas ayuda a comprender mejor la estructura de un programa?
	
	a) ¿Cuál es el valor de x?
	b) ¿Qué ocurre si cambiamos el valor de x a 2?
	c) ¿Cómo se imprime el valor de x?
	d) ¿Qué tipo de dato es x?

02	**Funciones** Observa el siguiente código: def sumar(a, b): return a + b resultado = sumar(3, 4) print(resultado) **¿Qué pregunta podría hacer un estudiante para identificar el propósito de una función?** a) ¿Qué sucede si cambiamos a por una cadena de texto? b) ¿Cómo se llama la función? c) ¿En qué línea se encuentra el retorno de la función? d) ¿Qué hace la función sumar?

03	**Variables y tipos de datos** Considera el siguiente código: nombre = "Juan" edad = 20 es_mayor_de_edad = edad >= 18 **¿Cuál de las siguientes preguntas refleja un análisis adecuado del uso de variables?** a) ¿Qué tipo de dato tiene la variable edad? b) ¿En qué línea se imprime nombre? c) ¿Qué sucede si cambiamos el valor de es_mayor_de_edad? d) ¿Por qué se asigna True a es_mayor_de_edad?
04	**Estructura de datos (Arreglos/Listas)** Analiza el siguiente código: numeros = [1, 2, 3, 4, 5] for numero in numeros: print(numero) **¿Cuál de las siguientes preguntas es pertinente para comprender la estructura del programa?** a) ¿Cuál es el valor del primer elemento de la lista? b) ¿Qué ocurriría si cambiamos el valor de numero dentro del bucle? c) ¿Cómo se agrega un nuevo elemento a la lista numeros? d) ¿Por qué se usa un bucle for en lugar de while?

Indicador: Describen correctamente las estructuras de un programa

Nro	Item

01	Observa el siguiente fragmento de código en Python:
	
	numero = 7
	if numero % 2 == 0:
	print("El número es par")
	else:
	print("El número es impar")
	
	¿Cuál es la descripción más precisa del funcionamiento de este fragmento de código?
	
	a) El programa verifica si el número es mayor que 2 y muestra un mensaje.
	b) El programa determina si el número es par o impar y muestra el

	resultado. c) El programa calcula el doble del número e imprime el resultado. d) El programa compara el número con 7 y muestra un mensaje si es igual.
02	Analiza el siguiente código: frutas = ["manzana", "banana", "pera"] for fruta in frutas: print("Me gusta la " + fruta) **¿Cuál descripción explica mejor la funcionalidad de este código?** a) El código imprime la cantidad de frutas en la lista. b) El código imprime cada fruta solo si es una manzana. c) El código recorre la lista de frutas e imprime un mensaje para cada una d) El código agrega nuevas frutas a la lista e imprime el resultado.
03	Revisa el siguiente código: def multiplicar(x, y): resultado = x * y return resultado print(multiplicar(3, 4)) **¿Cuál es la descripción correcta para este fragmento de código?** a) La función suma los dos números y retorna el resultado b) La función verifica si los números son iguales y retorna un booleano c) La función multiplica dos números y retorna el resultado d) La función divide los dos números y muestra el resultado.
04	Observa el siguiente código: contador = 0 while contador < 5: print("Contador:", contador) contador += 1 **¿Cuál de las siguientes descripciones es la más precisa para este código?** a) El código imprime el valor del contador cinco veces y termina cuando el contador llega a 5. b) El código imprime solo los números pares entre 0 y 5. c) El código continúa indefinidamente y nunca termina. El código inicia con el contador en 5 y lo decrementa hasta 0.

Indicador: Participación voluntaria.

Nro	Item
01	**¿Con qué frecuencia participas voluntariamente en una clase de programación?** a) Nunca, prefiero escuchar y tomar notas. b) 1 o 2 veces durante la clase. c) Más de 3 veces por clase. d) Participo siempre que tengo una duda o un comentario que hacer.
02	**¿Qué tipo de intervenciones sueles realizar cuando participas en clase?** a) Solo pregunto cuando tengo dudas sobre el material. b) Aporto respuestas a las preguntas que hace el profesor. c) Hago preguntas adicionales para profundizar en el tema. d) Hago tanto preguntas como aporto ideas y sugerencias durante la clase.
03	**¿En qué situaciones te sientes más cómodo(a) participando en clase?** a) Cuando el profesor me lo pide directamente. b) Cuando estoy seguro(a) de la respuesta. c) Cuando se trata de un tema que me interesa mucho. d) Siempre que tengo algo que decir, sin importar el contexto.
04	**¿Cómo crees que tus compañeros y el profesor perciben tu participación en clase?** a) Piensan que participo muy poco. b) Creen que participo de forma adecuada. c) Piensan que participo con frecuencia y aporto al grupo. d) Me consideran uno de los estudiantes que más participa en clase.

Indicador: Explica correctamente la funcionalidad de un programa

Nro	Item
01	Observa el siguiente código: a = 5 b = 3 resultado = a + b print("La suma es:", resultado) **¿Cuál es la explicación correcta de lo que hace este programa?** a) El programa resta los valores de a y b y muestra el resultado. b) El programa solicita al usuario que ingrese dos números y los suma. c) El programa suma los valores de a y b y muestra el resultado. d) El programa compara a y b para determinar cuál es mayor.

02	Analiza el siguiente código:
	
	numero = 8
	if numero % 2 == 0:
	print("El número es par")
	else:
	print("El número es impar")
	
	¿Qué funcionalidad describe mejor este programa?
	
	a) El programa determina si el número es positivo o negativo.
	b) El programa verifica si el número es par o impar y muestra el resultado.
	c) El programa multiplica el número por 2 y muestra el resultado.
	d) El programa encuentra el valor absoluto del número.

03	Observa este fragmento de código: nombres = ["Ana", "Luis", "Pedro"] for nombre in nombres: print("Hola, " + nombre) **¿Cuál es la funcionalidad de este programa?** a) El programa agrega nuevos nombres a la lista y los imprime. b) El programa imprime un saludo para cada nombre en la lista. c) El programa busca el nombre "Ana" y lo imprime. d) El programa ordena la lista de nombres y los imprime.
04	Considera el siguiente código: def calcular_area(base, altura): area = base * altura return area resultado = calcular_area(5, 10) print("El área del rectángulo es:", resultado) **¿Qué descripción explica mejor la funcionalidad de este programa?** a) El programa calcula la suma de la base y la altura de un rectángulo. b) El programa calcula el área de un rectángulo y muestra el resultado. c) El programa verifica si la base y la altura son iguales. d) El programa imprime los valores de la base y la altura sin realizar cálculos.

Anexo 3: Certificado de validación del instrumento y validez de contenido mediante V AIKEN

FORMATO DE EVALUACIÓN DE INSTRUMENTO DE MEDICIÓN POR JUICIO DE EXPERTOS

Respetado juez: Usted ha sido seleccionado para evaluar el instrumento de medición denominado "Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior" que forma parte de la investigación titulada: "**Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación**".

La evaluación del instrumento es de gran relevancia para lograr que sea válido y que los resultados obtenidos a partir de éste sean utilizados eficientemente; aportando a la línea de investigación en educación, como a sus aplicaciones. Agradezco su valiosa colaboración.

1. **DATOS GENERALES DEL JUEZ**

Nombre del juez:	Dr. Gilberto Carrión Barco
Grado profesional:	Maestría ( ) Doctor (X)
Área de formación académica:	Doctor en Ciencias de la Computación y Sistemas
Áreas de experiencia profesional:	Ingeniero en Computación e Informática, Doctor en Ciencias de la Computación y Sistemas, Maestro en Ingeniería de Sistemas, Magister en Docencia Universitaria. En la línea de Ciencias Sociales, Licenciado en Administración Pública, Maestro en Gestión Pública, Máster en Innovación de Procesos en la Gestión Pública por la Universidad Politécnica de Valencia España (UPV), Doctor en Gestión Pública y Gobernabilidad.
Institución donde labora:	Universidad Nacional Pedro Ruiz Gallo (UNPRG), Universidad Tecnológica del Perú (UTP), Universidad de San Martín de Porres (USMP)
Tiempo de experiencia profesional en el área:	2 a 4 años ( ) Más de 5 años (X)
Experiencia en Investigación (deseable):	Investigador RENACYT Nivel III

2. **PROPÓSITO DE LA EVALUACIÓN:**
   a. Validar lingüísticamente el instrumento: Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior
   b. Juzgar la pertinencia de los ítems de acuerdo a sus dimensiones.

3. **DATOS DEL INSTRUMENTO**

Nombre del instrumento:	Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior
Autor(es):	Ing. Hans Edgardo Delgado Díaz
Procedencia:	Propia
Administración:	Grupal
Tiempo de aplicación:	45 min. Aprox.
Ámbito de aplicación:	Educativo

4. **SOPORTE TEÓRICO**

Dimensiones del instrumento:

DIMENSIONES	DEFINICIÓN
1. Comprensión de conceptos básicas de programación	Identifica las principales estructuras de un programa y realiza preguntas a partir de la identificación de las estructuras de un programa
2. Aplicación de estructuras de programación	Describe correctamente las estructuras de un programa
3. Solución de problemas mediante programación	Participa voluntariamente

5. **PRESENTACIÓN DE INSTRUCCIONES PARA EL JUEZ:**

Respetado(a) juez:

A continuación, le presento el instrumento de medición denominado: Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior, por lo que le solicito tenga a bien realizar la calificación de acuerdo con los siguientes indicadores:

CATEGORÍA	CALIFICACIÓN	INDICADOR
CLARIDAD El ítem se comprende fácilmente, es decir, su sintáctica y semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de las mismas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
COHERENCIA El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. Totalmente en desacuerdo (No cumple con el criterio)	El ítem no tiene relación lógica con la dimensión.
	2. Desacuerdo (Bajo nivel de acuerdo)	El ítem tiene una relación tangencial /lejana con la dimensión.
	3. Acuerdo (Moderado nivel)	El ítem tiene una relación moderada con la dimensión que se está midiendo.
	4. Totalmente de Acuerdo (Alto nivel)	El ítem se encuentra está relacionado con la dimensión que está midiendo.
RELEVANCIA El ítem es esencial o importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.
	4. Alto nivel	El ítem es muy relevante y debe ser incluido.

6. DIMENSIONES DEL INSTRUMENTO:

El instrumento puede visualizarse en el Anexo 1.

DIMENSIÓN	ÍTEM	CLARIDAD	COHERENCIA	RELEVANCIA	OBSERVACIONES/RECOMENDACIONES
Comprensión de conceptos básicas de programación	1. ¿Cuáles son las tres principales estructuras que conforman un programa básico en la mayoría de los lenguajes de programación?	4	4	4	Mejorar la redacción para mayor fluidez. ¿Cuáles son las tres estructuras fundamentales que componen un programa básico en la mayoría de los lenguajes de programación?
	2. ¿Qué hace una estructura condicional en un programa?	3	4	4	Mejorar la redacción por un Ítem es claro y coherente. ¿Cuál es la función de una estructura condicional dentro de un programa?
	3. ¿Qué es una función en programación y por qué es importante usarla?	3	4	4	Buena redacción, solo ajustar el estilo. ¿Qué es una función en programación y cuál es su importancia en el desarrollo de programas?
	4. ¿Cuál es el propósito de un ciclo de repetición en un programa?	4	4	4	
	5. ¿Cuál de las siguientes preguntas podría realizarse para entender mejor la estructura del programa?	3	4	3	Se sugiere ajustar redacción. ¿Cuál de las siguientes preguntas ayuda a comprender mejor la estructura de un programa?
	6. ¿Qué pregunta podría hacer un estudiante para investigar el propósito de esta función?	3	4	4	Mejorar en precisión de términos. ¿Qué pregunta podría hacer un estudiante para identificar el propósito de una función?
	7. ¿Cuál de las siguientes preguntas refleja un análisis adecuado del uso de variables?	4	4	4	
	8. ¿Cuál de las siguientes preguntas es pertinente para comprender la estructura del programa?	4	4	4	
Aplicación de estructuras de programación	9. ¿Cuál es la descripción más precisa para este fragmento de código?	3	4	4	Mejorar la precisión gramatical ¿Cuál es la descripción más precisa del funcionamiento de este fragmento de código?
	10. ¿Qué descripción explica mejor lo que hace este código?	3	4	4	Mejorar la semántica. ¿Cuál descripción explica mejor la funcionalidad de este código?
	11. ¿Cuál es la descripción correcta de este fragmento?	3	4	4	Ajustar la redacción ¿Cuál es la descripción correcta para este fragmento de código?
	12. ¿Cuál de las siguientes descripciones es la más precisa para este código?	4	4	4	
Solución de problemas mediante programación	13. ¿Cuántas veces participas voluntariamente en una clase de programación típica?	3	4	4	Mejorar la claridad del ítem. ¿Con qué frecuencia participas voluntariamente en una clase de programación?
	14. ¿Qué tipo de intervenciones sueles realizar cuando participas en clase?	4	4	4	
	15. ¿En qué situaciones te sientes más cómodo(a) participando en clase?	4	4	4	
	16. ¿Cómo crees que perciben tus compañeros y el profesor tu participación en clase?	3	4	4	Mejorar la claridad del ítem. ¿Cómo crees que tus compañeros y el profesor perciben tu participación en clase?
	17. ¿Cuál es la explicación correcta de lo que hace este programa?	4	4	4	
	18. ¿Qué funcionalidad describe mejor este programa?	4	4	4	
	19. ¿Cuál es la funcionalidad de este programa?	4	4	4	
	20. ¿Qué descripción explica mejor la funcionalidad de este programa?	4	4	4	

Ciudad y fecha de evaluación: Chiclayo, 10 de abril del 2025



DR. GILBERTO CARRION-BARCO
INVESTIGADOR RENACYT
Código P0070731
DNI: 16720146

FORMATO DE EVALUACIÓN DE INSTRUMENTO DE MEDICIÓN POR JUICIO DE EXPERTOS

Respetado juez: Usted ha sido seleccionado para evaluar el instrumento de medición denominado "Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior" que forma parte de la investigación titulada: "**Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación**".

La evaluación del instrumento es de gran relevancia para lograr que sea válido y que los resultados obtenidos a partir de éste sean utilizados eficientemente; aportando a la línea de investigación en educación, como a sus aplicaciones. Agradezco su valiosa colaboración.

1. **DATOS GENERALES DEL JUEZ**

Nombre del juez:	Mg. Anthony Hans Delgado Chavarri
Grado profesional:	Maestría (x) Doctor ( )
Área de formación académica:	Maestría en dirección estratégica de tecnologías de la información
Áreas de experiencia profesional:	Docencia Universitaria, Consultor en Tecnologías de la Información (TI) y Microsoft Trainer Certified me permite ser un profesional altamente especializado en el ámbito de la tecnología, con un enfoque en ofrecer soluciones y asesoramiento a empresas en cuanto a infraestructura tecnológica, procesos digitales y optimización de sistemas
Institución donde labora:	Universidad Tecnológica del Perú (UTP), CERTUS, New Horizons Perú, Universidad Privada del Norte (UPN)
Tiempo de experiencia profesional en el área:	2 a 4 años (x) Más de 5 años ( )
Experiencia en Investigación (deseable):	Consultor en Tecnologías de la Información (TI), asesor de proyectos de investigación.

2. **PROPÓSITO DE LA EVALUACIÓN:**
   a. Validar lingüísticamente el instrumento: Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior
   b. Juzgar la pertinencia de los ítems de acuerdo a sus dimensiones.

3. **DATOS DEL INSTRUMENTO**

Nombre del instrumento:	Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior
Autor(es):	Ing. Hans Edgardo Delgado Díaz
Procedencia:	Propia
Administración:	Grupal
Tiempo de aplicación:	45 min. Aprox.
Ámbito de aplicación:	Educativo

4. **SOPORTE TEÓRICO**

Dimensiones del instrumento:

DIMENSIONES	DEFINICIÓN
1. Comprensión de conceptos básicas de programación	Identifica las principales estructuras de un programa y realiza preguntas a partir de la identificación de las estructuras de un programa
2. Aplicación de estructuras de programación	Describe correctamente las estructuras de un programa
3. Solución de problemas mediante programación	Participa voluntariamente

5. **PRESENTACIÓN DE INSTRUCCIONES PARA EL JUEZ:**

Respetado(a) juez:

A continuación, le presento el instrumento de medición denominado: Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior, por lo que le solicito tenga a bien realizar la calificación de acuerdo con los siguientes indicadores:

CATEGORÍA	CALIFICACIÓN	INDICADOR
CLARIDAD El ítem se comprende fácilmente, es decir, su sintáctica y semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de las mismas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
COHERENCIA El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. Totalmente en desacuerdo (No cumple con el criterio)	El ítem no tiene relación lógica con la dimensión.
	2. Desacuerdo (Bajo nivel de acuerdo)	El ítem tiene una relación tangencial /lejana con la dimensión.
	3. Acuerdo (Moderado nivel)	El ítem tiene una relación moderada con la dimensión que se está midiendo.
	4. Totalmente de Acuerdo (Alto nivel)	El ítem se encuentra está relacionado con la dimensión que está midiendo.
RELEVANCIA El ítem es esencial o importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.
	4. Alto nivel	El ítem es muy relevante y debe ser incluido.

6. DIMENSIONES DEL INSTRUMENTO:

El instrumento puede visualizarse en el Anexo 1.

DIMENSIÓN	ÍTEM	CLARIDAD	COHERENCIA	RELEVANCIA	OBSERVACIONES/RECOMENDACIONES
Comprensión de conceptos básicas de programación	1. ¿Cuáles son las tres principales estructuras que conforman un programa básico en la mayoría de los lenguajes de programación?	4	4	4	
	2. ¿Qué hace una estructura condicional en un programa?	4	4	4	
	3. ¿Qué es una función en programación y por qué es importante usarla?	4	4	4	
	4. ¿Cuál es el propósito de un ciclo de repetición en un programa?	4	4	4	
	5. ¿Cuál de las siguientes preguntas podría realizarse para entender mejor la estructura del programa?	4	4	4	
	6. ¿Qué pregunta podría hacer un estudiante para investigar el propósito de esta función?	4	4	4	
	7. ¿Cuál de las siguientes preguntas refleja un análisis adecuado del uso de variables?	4	4	4	
	8. ¿Cuál de las siguientes preguntas es pertinente para comprender la estructura del programa?	4	4	4	
Aplicación de estructuras de programación	9. ¿Cuál es la descripción más precisa para este fragmento de código?	4	4	4	
	10. ¿Qué descripción explica mejor lo que hace este código?	4	4	4	
	11. ¿Cuál es la descripción correcta de este fragmento?	4	4	4	
	12. ¿Cuál de las siguientes descripciones es la más precisa para este código?	4	4	4	

Solución de problemas mediante programación	13. ¿Cuántas veces participas voluntariamente en una clase de programación típica?	4	4	4	
	14. ¿Qué tipo de intervenciones sueles realizar cuando participas en clase?	4	4	4	
	15. ¿En qué situaciones te sientes más cómodo(a) participando en clase?	4	4	4	
	16. ¿Cómo crees que perciben tus compañeros y el profesor tu participación en clase?	4	4	4	
	17. ¿Cuál es la explicación correcta de lo que hace este programa?	4	4	4	
	18. ¿Qué funcionalidad describe mejor este programa?	4	4	4	
	19. ¿Cuál es la funcionalidad de este programa?	4	4	4	
	20. ¿Qué descripción explica mejor la funcionalidad de este programa?	4	4	4	

Ciudad y fecha de evaluación: Chiclayo, 10 de abril del 2025

ANTHONY HANS DELGADO CHAVARRI
INGENIERO DE SISTEMAS Y COMPUTACION
REG. CIP. 186247

Mg. Anthony Hans Delgado Chavarri

Firma del evaluador
Grado, nombres y apellidos completos
DNI

FORMATO DE EVALUACIÓN DE INSTRUMENTO DE MEDICIÓN POR JUICIO DE EXPERTOS

Respetado juez: Usted ha sido seleccionado para evaluar el instrumento de medición denominado "Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior" que forma parte de la investigación titulada: "**Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación**".

La evaluación del instrumento es de gran relevancia para lograr que sea válido y que los resultados obtenidos a partir de éste sean utilizados eficientemente; aportando a la línea de investigación en educación, como a sus aplicaciones. Agradezco su valiosa colaboración.

1. **DATOS GENERALES DEL JUEZ**

Nombre del juez:	Mg. Janett Aracely Gonzales Flores
Grado profesional:	Maestría (x) Doctor ( )
Área de formación académica:	Maestría en dirección estratégica de tecnologías de la información
Áreas de experiencia profesional:	Docente Universitario, Consultor de Sistemas, Seminario de Tesis, Diseño y Desarrollo de Tesis.
Institución donde labora:	Universidad Tecnológica del Perú (UTP), CERTUS
Tiempo de experiencia profesional en el área:	2 a 4 años (x) Más de 5 años ( )
Experiencia en Investigación (deseable):	Consultor en Tecnologías de la Información (TI) y asesor de proyectos de investigación.

2. **PROPÓSITO DE LA EVALUACIÓN:**
   a. Validar lingüísticamente el instrumento: Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior
   b. Juzgar la pertinencia de los ítems de acuerdo a sus dimensiones.

3. **DATOS DEL INSTRUMENTO**

Nombre del instrumento:	Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior
Autor(es):	Ing. Hans Edgardo Delgado Díaz
Procedencia:	Propia
Administración:	Grupal
Tiempo de aplicación:	45 min. Aprox.
Ámbito de aplicación:	Educativo

4. **SOPORTE TEÓRICO**

Dimensiones del instrumento:

DIMENSIONES	DEFINICIÓN
1. Comprensión de conceptos básicas de programación	Identifica las principales estructuras de un programa y realiza preguntas a partir de la identificación de las estructuras de un programa
2. Aplicación de estructuras de programación	Describe correctamente las estructuras de un programa
3. Solución de problemas mediante programación	Participa voluntariamente

5. **PRESENTACIÓN DE INSTRUCCIONES PARA EL JUEZ:**

Respetado(a) juez:

A continuación, le presento el instrumento de medición denominado: Cuestionario para evaluar desarrollo de habilidades prácticas para el curso de Fundamentos de Programación en educación superior, por lo que le solicito tenga a bien realizar la calificación de acuerdo con los siguientes indicadores:

CATEGORÍA	CALIFICACIÓN	INDICADOR
CLARIDAD El ítem se comprende fácilmente, es decir, su sintáctica y semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de las mismas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
COHERENCIA El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. Totalmente en desacuerdo (No cumple con el criterio)	El ítem no tiene relación lógica con la dimensión.
	2. Desacuerdo (Bajo nivel de acuerdo)	El ítem tiene una relación tangencial /lejana con la dimensión.
	3. Acuerdo (Moderado nivel)	El ítem tiene una relación moderada con la dimensión que se está midiendo.
	4. Totalmente de Acuerdo (Alto nivel)	El ítem se encuentra está relacionado con la dimensión que está midiendo.
RELEVANCIA El ítem es esencial o importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.
	4. Alto nivel	El ítem es muy relevante y debe ser incluido.

6. DIMENSIONES DEL INSTRUMENTO:

El instrumento puede visualizarse en el Anexo 1.

DIMENSIÓN	ÍTEM	CLARIDAD	COHERENCIA	RELEVANCIA	OBSERVACIONES/RECOMENDACIONES
Comprensión de conceptos básicas de programación	1. ¿Cuáles son las tres principales estructuras que conforman un programa básico en la mayoría de los lenguajes de programación?	4	4	4	
	2. ¿Qué hace una estructura condicional en un programa?	4	4	4	
	3. ¿Qué es una función en programación y por qué es importante usarla?	4	4	4	
	4. ¿Cuál es el propósito de un ciclo de repetición en un programa?	4	4	4	
	5. ¿Cuál de las siguientes preguntas podría realizarse para entender mejor la estructura del programa?	4	4	4	
	6. ¿Qué pregunta podría hacer un estudiante para investigar el propósito de esta función?	4	4	4	
	7. ¿Cuál de las siguientes preguntas refleja un análisis adecuado del uso de variables?	4	4	4	
	8. ¿Cuál de las siguientes preguntas es pertinente para comprender la estructura del programa?	4	4	4	
Aplicación de estructuras de programación	9. ¿Cuál es la descripción más precisa para este fragmento de código?	4	4	4	
	10. ¿Qué descripción explica mejor lo que hace este código?	4	4	4	
	11. ¿Cuál es la descripción correcta de este fragmento?	4	4	4	
	12. ¿Cuál de las siguientes descripciones es la más precisa para este código?	4	4	4	

Solución de problemas mediante programación	13. ¿Cuántas veces participas voluntariamente en una clase de programación típica?	4	4	4	
	14. ¿Qué tipo de intervenciones sueles realizar cuando participas en clase?	4	4	4	
	15. ¿En qué situaciones te sientes más cómodo(a) participando en clase?	4	4	4	
	16. ¿Cómo crees que perciben tus compañeros y el profesor tu participación en clase?	4	4	4	
	17. ¿Cuál es la explicación correcta de lo que hace este programa?	4	4	4	
	18. ¿Qué funcionalidad describe mejor este programa?	4	4	4	
	19. ¿Cuál es la funcionalidad de este programa?	4	4	4	
	20. ¿Qué descripción explica mejor la funcionalidad de este programa?	4	4	4	

Ciudad y fecha de evaluación: Chiclayo, 10 de abril del 2025



Mg. Janett Aracely Gonzales Flores

DNI: 73618769

Diseñado por Jose Ventura

Como citar: Ventura-León, J. (2019). De regreso a la validez basada en el contenido. *Adicciones, 34* (4), 323-326. http://dx.doi.org/10.20882/adicciones.1213

Inserte valores

min	*1*
max	*4*
k	*3*
n	3
sig	1.96

95%

									Intervalo de Confianza	
		Juez 1	Juez 2	Juez 3	Media	DE	V de Aiken	Interpretacion V	Inferior	Superior
Item1	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item2	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item3	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item4	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item5	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
Item6	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item7	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item8	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item9	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item10	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item11	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item12	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item13	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item14	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item15	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item16	Claridad	3	4	4	3.67	0.58	0.89	VÁLIDO	0.56	0.98
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item17	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item18	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item19	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
Item20	Claridad	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Coherencia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00
	Relevancia	4	4	4	4.00	0.00	1.00	VÁLIDO	0.70	1.00

CONSIDERAR SOLO LAS 4 ÚLTIMAS COLUMNAS

Resultados de V de Aiken

Ítems	Criterios	V de Aiken	Interpretación	IC 95%
1	Claridad	1.00	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
2	Claridad	0.89	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
3	Claridad	0.89	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
4	Claridad	1.00	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
5	Claridad	0.89	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	0.89	Válido	[.56 - .98]
6	Claridad	0.89	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
7	Claridad	1.00	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
8	Claridad	1.00	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
9	Claridad	0.89	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
10	Claridad	0.89	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]

11	Claridad	0.89	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
12	Claridad	0.89	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
13	Claridad	1.00	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
14	Claridad	1.00	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
15	Claridad	0.89	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
16	Claridad	1.00	Válido	[.56 - .98]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
17	Claridad	1.00	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
18	Claridad	1.00	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
19	Claridad	1.00	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]
20	Claridad	0.89	Válido	[.70 - 1.00]
	Coherencia	1.00	Válido	[.70 - 1.00]
	Relevancia	1.00	Válido	[.70 - 1.00]

Nota. IC: Intervalos de confianza

En la tabla, se muestran los índices de V de Aiken para los ítems del instrumento, observándose que la totalidad de ellos han sido válidos.

Anexo 4: Data de los resultados del cuestionario de la prueba piloto

	Dimensión Comprensión de conceptos básicas de programación								Dimensión Aplicación de estructuras de pr		
	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10	Item 11
Sujeto 1	1	0	0	1	0	0	0	0	1	1	1
Sujeto 2	0	0	1	1	1	1	0	0	1	1	1
Sujeto 3	0	0	0	0	0	0	0	0	1	1	1
Sujeto 4	0	1	1	0	0	0	0	0	1	1	1
Sujeto 5	0	1	1	1	0	1	1	1	1	1	1
Sujeto 6	1	0	0	1	0	0	0	0	1	1	1
Sujeto 7	0	1	1	0	0	0	0	0	1	1	1
Sujeto 8	0	0	0	0	0	0	0	0	1	1	1
Sujeto 9	0	0	1	1	1	1	0	0	1	1	1
Sujeto 10	0	1	1	0	0	0	0	0	1	1	1
p	0.20	0.40	0.60	0.50	0.20	0.30	0.10	0.10	1.00	1.00	1.00
q=(1-p)	0.80	0.60	0.40	0.50	0.80	0.70	0.90	0.90	0.00	0.00	0.00
Pq	0.16	0.24	0.24	0.25	0.16	0.21	0.09	0.09	0.00	0.00	0.00
Suma PQ	2.42										
N° items	20										
Varianza	9.38										
KR-20	**0.781**		DEBE SUPERAR EL 0.70 PARA SER ADECUADO								

*Se verificó la confiabilidad a través del coeficiente Kuder-Richardson y obtuvo un valor adecuado (0.78).

ogramación	Dimensión Solución de problemas mediante programación								
Item 12	Item 13	Item 14	Item 15	Item 16	Item 17	Item 18	Item 19	Item 20	SUMA
0	0	1	1	0	0	1	1	1	10
1	1	1	0	1	1	1	1	1	15
0	0	1	0	0	1	1	1	1	8
0	1	1	0	0	1	1	1	1	11
1	1	1	0	1	1	1	1	1	17
0	0	1	1	0	0	1	1	1	10
0	1	1	0	0	1	1	1	1	11
0	0	1	0	0	1	1	1	1	8
1	1	1	0	1	1	1	1	1	15
0	1	1	0	0	1	1	1	1	11
0.30	0.60	1.00	0.20	0.30	0.80	1.00	1.00	1.00	9.38
0.70	0.40	0.00	0.80	0.70	0.20	0.00	0.00	0.00	
0.21	0.24	0.00	0.16	0.21	0.16	0.00	0.00	0.00	

Anexo 5: Data de los resultados del cuestionario en las pruebas Pre test y Pos test del Grupo Control y Grupo Experimental

Resultados obtenidos en el Pre Test y Pos Test en los Grupos Control y Experimental, antes y después de usar el aplicativo de realidad aumentada.

PRE TEST CONTROL

	Dimensión Comprensión de conceptos básicas de programación									Dimensión Aplicación de estructuras de programación				
	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	PRE_D1	Item 9	Item 10	Item 11	Item 12	PRE_D2
Sujeto 1	0	1	1	0	1	1	0	0	4	1	1	0	1	3
Sujeto 2	0	1	0	1	0	0	0	1	3	1	1	1	1	4
Sujeto 3	1	0	1	1	0	0	0	1	4	0	1	1	0	2
Sujeto 4	0	0	0	0	0	0	0	1	1	1	1	1	1	4
Sujeto 5	0	1	0	1	0	0	0	1	3	1	1	1	1	4
Sujeto 6	0	1	0	1	1	0	1	0	4	1	0	1	1	3
Sujeto 7	0	1	0	1	0	0	1	1	4	1	1	1	1	4
Sujeto 8	0	1	1	0	1	1	0	0	4	1	1	0	1	3
Sujeto 9	0	1	0	1	0	0	0	1	3	1	1	1	1	4
Sujeto 10	1	0	1	1	0	0	0	1	4	0	1	1	0	2
Sujeto 11	0	0	0	0	0	0	0	1	1	1	1	1	1	4
Sujeto 12	0	1	0	1	0	0	0	1	3	1	1	1	1	4
Sujeto 13	0	1	0	1	1	0	1	0	4	1	0	1	1	3
Sujeto 14	0	1	0	1	0	0	1	1	4	1	1	1	1	4
Sujeto 15	0	1	1	0	1	1	0	0	4	1	1	0	1	3

Dimensión Solución de problemas mediante programación											
Item 13	Item 14	Item 15	Item 16	Item 17	Item 18	Item 19	Item 20	PRE_D3	TOTAL	NIVEL_TOTAL	
0	0	1	1	0	1	1	1	5	12	Básico	DEFICIENTE
1	0	0	0	1	1	1	1	5	12	Básico	BASICO
0	0	0	0	1	1	1	1	4	10	Deficiente	SUFICIENTE
0	0	0	0	1	1	1	1	4	9	Deficiente	SOBRESALIENTE
0	0	1	0	1	1	1	1	5	12	Básico	
1	0	0	0	1	1	0	1	4	11	Básico	
0	0	1	1	1	1	1	1	6	14	Suficiente	
0	0	1	1	0	1	1	1	5	12	Básico	
1	0	0	0	1	1	1	1	5	12	Básico	
0	0	0	0	1	1	1	1	4	10	Deficiente	
0	0	0	0	1	1	1	1	4	9	Deficiente	
0	0	1	0	1	1	1	1	5	12	Básico	
1	0	0	0	1	1	0	1	4	11	Básico	
0	0	1	1	1	1	1	1	6	14	Suficiente	
0	0	1	1	0	1	1	1	5	12	Básico	

PRE TEST EXPERIMENTAL

	Dimensión Comprensión de conceptos básicas de programación									Dimensión Aplicación de estructuras de programación				
	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	PRE_D1	Item 9	Item 10	Item 11	Item 12	PRE_D2
Sujeto 1	0	1	1	1	0	0	1	0	4	0	1	1	1	3
Sujeto 2	0	1	1	1	0	0	0	1	4	1	1	1	1	4
Sujeto 3	1	0	0	1	0	0	0	1	3	1	1	1	1	4
Sujeto 4	1	1	0	1	0	0	0	1	4	1	0	1	0	2
Sujeto 5	1	0	0	0	0	0	0	0	1	1	0	1	1	3
Sujeto 6	0	1	1	1	0	0	1	0	4	1	1	1	0	3
Sujeto 7	0	1	1	1	0	0	0	1	4	1	1	1	1	4
Sujeto 8	1	1	0	0	0	0	0	1	3	0	1	1	0	2
Sujeto 9	0	0	0	1	0	0	1	1	3	1	0	1	0	2
Sujeto 10	0	1	1	1	0	0	1	0	4	1	1	1	1	4
Sujeto 11	1	0	0	0	0	0	0	0	1	1	0	1	1	3
Sujeto 12	1	1	0	1	0	0	0	1	4	1	0	1	0	2
Sujeto 13	1	1	0	1	0	0	0	1	4	1	1	1	1	4
Sujeto 14	1	0	1	0	0	0	0	1	3	1	1	1	1	4
Sujeto 15	1	1	1	0	0	0	1	0	4	0	1	1	1	3

Dimensión Solución de problemas mediante programación											
Item 13	Item 14	Item 15	Item 16	Item 17	Item 18	Item 19	Item 20	PRE_D3	TOTAL	NIVEL_TOTAL	
0	0	1	1	1	1	0	1	5	12	Básico	DEFICIENTE
0	1	0	1	0	1	0	1	4	12	Básico	BASICO
0	0	1	1	1	1	0	1	5	12	Básico	SUFICIENTE
0	1	0	0	1	1	0	1	4	10	Deficiente	SOBRESALIENTE
0	1	0	0	1	0	0	1	3	7	Deficiente	
0	1	0	1	0	1	1	1	5	12	Básico	
1	0	0	0	1	1	1	1	5	13	Suficiente	
0	0	1	1	0	1	1	1	5	10	Deficiente	
0	1	0	0	0	1	1	0	3	8	Deficiente	
0	1	0	1	1	1	1	1	6	14	Suficiente	
0	0	0	0	1	0	0	1	2	6	Deficiente	
0	1	0	0	1	1	0	1	4	10	Deficiente	
0	0	1	1	1	1	0	1	5	13	Suficiente	
0	0	0	1	1	1	1	1	5	12	Básico	
0	0	1	1	1	1	1	1	6	13	Suficiente	

POS TEST CONTROL

	Dimensión Comprensión de conceptos básicas de programación									Dimensión Aplicación de estructuras de programación				
	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	POS_D1	Item 9	Item 10	Item 11	Item 12	POS_D2
Sujeto 1	0	1	1	0	1	1	0	0	4	1	1	0	1	3
Sujeto 2	0	1	0	1	0	0	0	1	3	1	1	1	1	4
Sujeto 3	1	0	1	1	0	0	0	1	4	1	1	1	0	3
Sujeto 4	0	1	0	1	1	0	0	1	4	1	1	1	1	4
Sujeto 5	0	1	0	1	0	1	0	1	4	1	1	1	1	4
Sujeto 6	0	1	0	1	1	0	1	0	4	1	0	1	1	3
Sujeto 7	0	1	0	1	0	0	1	1	4	1	1	1	1	4
Sujeto 8	0	1	1	0	1	1	0	0	4	1	1	0	1	3
Sujeto 9	0	1	0	1	0	0	0	1	3	1	1	1	1	4
Sujeto 10	1	0	1	1	0	0	0	1	4	1	1	1	0	3
Sujeto 11	0	1	0	1	0	1	0	1	4	1	1	1	1	4
Sujeto 12	0	1	0	1	0	0	0	1	3	1	1	1	1	4
Sujeto 13	0	1	0	1	1	0	1	0	4	1	0	1	1	3
Sujeto 14	0	1	0	1	0	0	1	1	4	1	1	1	1	4
Sujeto 15	0	1	1	0	1	1	0	0	4	1	1	0	1	3

Dimensión Solución de problemas mediante programación											
Item 13	Item 14	Item 15	Item 16	Item 17	Item 18	Item 19	Item 20	POS_D3	TOTAL	NIVEL_TOTAL	
0	0	1	1	0	1	1	1	5	12	Básico	DEFICIENTE
1	0	0	0	1	1	1	1	5	12	Básico	BASICO
1	0	0	0	1	1	1	1	5	12	Básico	SUFICIENTE
0	1	0	0	1	1	1	1	5	13	Suficiente	SOBRESALIENTE
0	0	1	0	1	1	1	1	5	13	Suficiente	
1	1	0	0	1	1	0	1	5	12	Básico	
0	0	1	1	1	1	1	1	6	14	Suficiente	
1	0	1	1	0	1	1	1	6	13	Suficiente	
1	0	0	0	1	1	1	1	5	12	Básico	
0	0	0	0	1	1	1	1	4	11	Básico	
0	0	0	0	1	1	1	1	4	12	Básico	
0	0	1	0	1	1	1	1	5	12	Básico	
1	1	0	0	1	1	0	1	5	12	Básico	
0	0	1	1	1	1	1	1	6	14	Suficiente	
0	0	1	1	0	1	1	1	5	12	Básico	

POS TEST EXPERIMENTAL

	Dimensión Comprensión de conceptos básicas de programación									Dimensión Aplicación de estructuras de programación				
	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	POS_D1	Item 9	Item 10	Item 11	Item 12	POS_D2
Sujeto 1	1	1	1	1	0	1	1	0	6	1	1	1	1	4
Sujeto 2	1	1	1	1	1	0	0	1	6	1	1	1	1	4
Sujeto 3	1	1	0	1	0	1	1	1	6	1	1	1	1	4
Sujeto 4	1	1	1	1	1	0	0	1	6	1	1	1	0	3
Sujeto 5	1	1	1	0	1	1	1	0	6	1	1	1	1	4
Sujeto 6	1	1	1	1	1	0	1	0	6	1	1	1	1	4
Sujeto 7	1	1	1	1	1	1	0	1	7	1	1	1	1	4
Sujeto 8	1	1	1	1	0	1	0	1	6	1	1	1	1	4
Sujeto 9	1	1	0	1	1	1	1	1	7	1	1	1	0	3
Sujeto 10	1	1	1	1	1	0	1	0	6	1	1	1	1	4
Sujeto 11	1	1	1	0	1	1	0	1	6	1	1	1	1	4
Sujeto 12	1	1	1	1	1	0	1	1	7	1	1	1	1	4
Sujeto 13	1	1	0	1	1	1	0	1	6	1	1	1	1	4
Sujeto 14	1	1	1	1	0	1	1	1	7	1	1	1	1	4
Sujeto 15	1	1	1	1	1	0	1	0	6	1	1	1	1	4

Dimensión Solución de problemas mediante programación											
Item 13	Item 14	Item 15	Item 16	Item 17	Item 18	Item 19	Item 20	POS_D3	TOTAL	NIVEL_TOTAL	
1	1	1	1	1	1	1	1	8	18	Sobresaliente	DEFICIENTE
1	1	0	1	1	1	1	1	7	17	Sobresaliente	BASICO
1	1	1	1	1	1	1	1	8	18	Sobresaliente	SUFICIENTE
1	1	1	1	1	1	1	1	8	17	Sobresaliente	SOBRESALIENTE
1	1	1	0	1	1	1	1	7	17	Sobresaliente	
1	1	1	1	1	1	1	1	8	18	Sobresaliente	
1	1	0	1	1	1	1	1	7	18	Sobresaliente	
1	1	1	1	0	1	1	1	7	17	Sobresaliente	
1	1	1	1	0	1	1	1	7	17	Sobresaliente	
1	1	0	1	1	1	1	1	7	17	Sobresaliente	
1	1	1	1	1	1	0	1	7	17	Sobresaliente	
1	1	0	1	1	1	1	1	7	18	Sobresaliente	
1	1	1	1	1	1	0	1	7	17	Sobresaliente	
1	1	0	1	1	1	1	1	7	18	Sobresaliente	
1	1	1	1	1	1	1	1	8	18	Sobresaliente	

Anexo 6: Planes de sesiones de aprendizaje

FICHA DE EVALUACIÓN DE PROGRAMA DE INTERVENCIÓN POR MEDICIÓN DE JUICIO DE EXPERTOS

Estimado experto:
Le comunico que usted ha sido elegido para emitir su evaluación sobre el Programa de Intervención denominado "Desarrollo de habilidades prácticas de fundamentos de programación" el cual es parte del estudio titulado: "**Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación**".
La evaluación del programa en cuestión es de gran relevancia para alcanzar la validez requerida con el propósito que los resultados alcanzados a partir de su aplicación sean utilizados eficientemente; aportando a la línea de investigación en educación, como a sus aplicaciones.
De antemano, le agradezco su valiosísimo apoyo.

1. DATOS GENERALES DEL JUEZ

Apellidos y Nombres del experto:	Dr. Gilberto Carrión Barco
Grado profesional:	Maestría ( ) Doctor (X)
Área de formación académica:	Doctor en Ciencias de la Computación y Sistemas
Áreas de experiencia profesional:	Ingeniero en Computación e Informática, Doctor en Ciencias de la Computación y Sistemas, Maestro en Ingeniería de Sistemas, Magister en Docencia Universitaria. En la línea de Ciencias Sociales, Licenciado en Administración Pública, Maestro en Gestión Pública, Máster en Innovación de Procesos en la Gestión Pública por la Universidad Politécnica de Valencia España (UPV), Doctor en Gestión Pública y Gobernabilidad.
Entidad donde labora:	Universidad Nacional Pedro Ruiz Gallo (UNPRG), Universidad Tecnológica del Perú (UTP), Universidad de San Martín de Porres (USMP)
Tiempo de experiencia profesional en el área:	2 a 4 años ( ) Más de 5 años (X)
Experiencia en Investigación (deseable):	Investigador RENACYT Nivel IV.

2. PROPÓSITO DE LA EVALUACIÓN:

a. Validar lingüísticamente el Programa de Intervención: Desarrollo de habilidades prácticas de fundamentos de programación – Diseño Instruccional.
b. Evaluar la coherencia de los contenidos de acuerdo a sus dimensiones y a los criterios establecidos.

3. DATOS DEL INSTRUMENTO

Nombre del instrumento:	Desarrollo de habilidades prácticas de fundamentos de programación
Autor(es):	Ing. Hans Edgardo Delgado Díaz

Procedencia:	Elaboración propia
Administración:	Grupal
Período de aplicación:	6 semanas
Ámbito de aplicación:	Educativo

4. SOPORTE TEÓRICO

Dimensiones del instrumento:

DIMENSIONES	DEFINICIÓN
1. Comprensión de conceptos básicas de programación	Identifica las principales estructuras de un programa y realiza preguntas a partir de la identificación de las estructuras de un programa
2. Aplicación de estructuras de programación	Describe correctamente las estructuras de un programa
3. Solución de problemas mediante programación	Participa voluntariamente

5. INSTRUCCIONES PARA EL EXPERTO VALIDADOR:

Estimado experto:

En el anexo 01 se adjunta el programa de intervención titulado: **Desarrollo de habilidades prácticas de fundamentos de programación – Diseño Instruccional**, con el objetivo que usted pueda evaluarlo de acuerdo a la siguiente matriz:

CRITERIO	MUY BUENO (10 PTS)	BUENO (7–9 PTS)	ACEPTABLE (4–6 PTS)	INSUFICIENTE (1–3 PTS)
Pertinencia pedagógica	Responde con precisión a una necesidad educativa y está alineado con el currículo nacional y las políticas institucionales.	Responde satisfactoriamente a una necesidad relevante; hay buena alineación curricular.	La necesidad educativa se evidencia parcialmente y la alineación curricular es parcial.	No responde a una necesidad educativa determinada y, además, no está alineado con el currículo.
Claridad de objetivos	Los objetivos del programa son claros, medibles, alcanzables y coherentes con el propósito del estudio.	Los objetivos del programa son claros, medibles, alcanzables y responden parcialmente al propósito del estudio.	Los objetivos están poco definidos, son difícilmente medibles y no guardan coherencia con el propósito del programa.	No presenta objetivos.

Uso adecuado de TIC	La (s) herramienta (s) TIC seleccionada (s) es/son pertinente (s), innovadora(s), accesible(s) y bien integrada (s).	La (s) herramienta (s) TIC seleccionada (s) evidencia (a) un uso funcional y relevante de TIC y se integra al programa de manera adecuada.	La (s) herramienta (s) TIC seleccionada (s) es/son útil (es), pero está/ están poco alineada (s) o mal aplicada (s).	La (s) herramienta (s) TIC seleccionada (s) es/ son inadecuada (s), irrelevante (s) y su uso no evidencia (a) justificación pedagógica.
Innovación metodológica	Propone metodologías activas, creativas e inclusivas que transforman el aprendizaje.	Presenta enfoques novedosos que favorecen el aprendizaje.	Uso de metodologías tradicionales con elementos digitales.	Sin elementos innovadores o metodologías inapropiadas.
Factibilidad de implementación	Es completamente viable con recursos disponibles; considera contexto real.	Es viable con pequeños ajustes logísticos.	Su implementación requiere recursos o condiciones no disponibles.	Difícilmente viable sin cambios sustanciales.
Evaluación del aprendizaje	Incluye al menos seis instrumentos de evaluación del aprendizaje variados, claros y pertinentes a los objetivos del programa; se evalúa proceso y producto.	Incluye al menos cuatro instrumentos de evaluación del aprendizaje diferentes, los cuales son claros y son coherentes con los objetivos del programa.	Incluye solo dos instrumentos de evaluación del aprendizaje diferentes y la evaluación propuesta es parcial, poco específica o limitada en alcance.	No se define evaluación o es incoherente.
Inclusión y accesibilidad	Atiende diversidad (cultural, funcional, tecnológica); asegura equidad.	Considera elementos inclusivos básicos.	Tiene intención inclusiva, pero sin acciones concretas.	No contempla aspectos de inclusión o accesibilidad.
Resultados esperados	Resultados claros, medibles, con impacto esperado en aprendizaje.	Resultados coherentes y viables.	Resultados poco claros o difíciles de medir.	Resultados vagos o irrelevantes.

ESCALA DE VALORACIÓN

Puntaje Total Obtenido	Nivel de Evaluación	Descripción	Resultado
72 - 80 puntos	**Excelente**	El programa cumple completamente con los criterios de calidad pedagógica, tecnológica y metodológica.	Se recomienda la aplicación sin observaciones.
64 - 71 puntos	**Muy Bueno**	El programa es sólido, con aspectos destacados, aunque requiere realizar algunos ajustes menores.	Se recomienda la aplicación previa subsanación de ajustes menores.
56 - 63 puntos	**Bueno / Aceptable**	El programa requiere mejoras importantes para garantizar una implementación eficaz y el logro de los resultados del estudio.	Se recomienda solamente si se implementan las mejoras propuestas por parte del evaluador.
55 puntos o menos	**Insuficiente**	No cumple con los estándares mínimos. Presenta debilidades importantes en diseño, aplicación o viabilidad.	No se recomienda la aplicación y el programa debe ser reestructurado.

6. RESULTADOS DE LA EVALUACIÓN:

CRITERIO	MUY BUENO (10 PTS)	BUENO (7–9 PTS)	ACEPTABLE (4–6 PTS)	INSUFICIENTE (1–3 PTS)	JUSTIFICACIÓN	OBSERVACIÓN
Pertinencia pedagógica	10				Responde con precisión a una necesidad educativa y está alineado con el currículo nacional y las políticas institucionales.	
Claridad de objetivos	10				Los objetivos del programa son claros, medibles, alcanzables y coherentes con el propósito del estudio.	
Uso adecuado de TIC	10				Las herramientas TIC seleccionadas son pertinentes, innovadoras, accesibles y bien integradas.	
Innovación metodológica	10				Propone metodologías activas, creativas e inclusivas que transforman el aprendizaje.	
Factibilidad de implementación	10				Es completamente viable con recursos disponibles; considera contexto real.	
Evaluación del aprendizaje	10				Incluye al menos seis instrumentos de evaluación del aprendizaje variados, claros y pertinentes a los objetivos del programa; se evalúa proceso y producto.	

Inclusión y accesibilidad	10				Atiende diversidad (cultural, funcional, tecnológica); asegura equidad.	
Resultados esperados	10				Resultados claros, medibles, con impacto esperado en aprendizaje.	
TOTAL						80

PUNTAJE TOTAL OBTENIDO	NIVEL DE EVALUACIÓN	DESCRIPCIÓN	RESULTADO
80	**Excelente**	El programa cumple completamente con los criterios de calidad pedagógica, tecnológica y metodológica.	Se recomienda la aplicación sin observaciones.

Ciudad y fecha de evaluación: Chiclayo, 13 de abril del 2025



DR. GILBERTO CARRION-BARCO
INVESTIGADOR RENACYT
Código P0070731
DNI: 16720146

FICHA DE EVALUACIÓN DE PROGRAMA DE INTERVENCIÓN POR MEDICIÓN DE JUICIO DE EXPERTOS

Estimado experto:

Le comunico que usted ha sido elegido para emitir su evaluación sobre el Programa de Intervención denominado "Desarrollo de habilidades prácticas de fundamentos de programación" el cual es parte del estudio titulado: **"Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación"**.

La evaluación del programa en cuestión es de gran relevancia para alcanzar la validez requerida con el propósito que los resultados alcanzados a partir de su aplicación sean utilizados eficientemente; aportando a la línea de investigación en educación, como a sus aplicaciones.

De antemano, le agradezco su valiosísimo apoyo.

1. DATOS GENERALES DEL JUEZ

Apellidos y Nombres del experto:	Mg. Janett Aracely Gonzales Flores
Grado profesional:	Maestría (X) Doctor ( )
Área de formación académica:	Maestría en dirección estratégica de tecnologías de la información
Áreas de experiencia profesional:	Docente Universitario, Consultor de Sistemas, Seminario de Tesis, Diseño y Desarrollo de Tesis.
Entidad donde labora:	Universidad Tecnológica del Perú (UTP), CERTUS
Tiempo de experiencia profesional en el área:	2 a 4 años (x) Más de 5 años ( )
Experiencia en Investigación (deseable):	Consultor en Tecnologías de la Información (TI) y asesor de proyectos de investigación.

2. PROPÓSITO DE LA EVALUACIÓN:

a. Validar lingüísticamente el Programa de Intervención: Desarrollo de habilidades prácticas de fundamentos de programación – Diseño Instruccional.
b. Evaluar la coherencia de los contenidos de acuerdo a sus dimensiones y a los criterios establecidos.

3. DATOS DEL INSTRUMENTO

Nombre del instrumento:	Desarrollo de habilidades prácticas de fundamentos de programación
Autor(es):	Ing. Hans Edgardo Delgado Díaz
Procedencia:	Elaboración propia
Administración:	Grupal
Período de aplicación:	6 semanas
Ámbito de aplicación:	Educativo

4. SOPORTE TEÓRICO

Dimensiones del instrumento:

DIMENSIONES	DEFINICIÓN
1. Comprensión de conceptos básicas de programación	Identifica las principales estructuras de un programa y realiza preguntas a partir de la identificación de las estructuras de un programa
2. Aplicación de estructuras de programación	Describe correctamente las estructuras de un programa
3. Solución de problemas mediante programación	Participa voluntariamente

5. INSTRUCCIONES PARA EL EXPERTO VALIDADOR:

Estimado experto:

En el anexo 01 se adjunta el programa de intervención titulado: **Desarrollo de habilidades prácticas de fundamentos de programación – Diseño Instruccional**, con el objetivo que usted pueda evaluarlo de acuerdo a la siguiente matriz:

CRITERIO	MUY BUENO (10 PTS)	BUENO (7–9 PTS)	ACEPTABLE (4–6 PTS)	INSUFICIENTE (1–3 PTS)
Pertinencia pedagógica	Responde con precisión a una necesidad educativa y está alineado con el currículo nacional y las políticas institucionales.	Responde satisfactoriamente a una necesidad relevante; hay buena alineación curricular.	La necesidad educativa se evidencia parcialmente y la alineación curricular es parcial.	No responde a una necesidad educativa determinada y, además, no está alineado con el currículo.
Claridad de objetivos	Los objetivos del programa son claros, medibles, alcanzables y coherentes con el propósito del estudio.	Los objetivos del programa son claros, medibles, alcanzables y responden parcialmente al propósito del estudio.	Los objetivos están poco definidos, son difícilmente medibles y no guardan coherencia con el propósito del programa.	No presenta objetivos.

Uso adecuado de TIC	La (s) herramienta (s) TIC seleccionada (s) es/son pertinente (s), innovadora(s), accesible(s) y bien integrada (s).	La (s) herramienta (s) TIC seleccionada (s) evidencia (a) un uso funcional y relevante de TIC y se integra al programa de manera adecuada.	La (s) herramienta (s) TIC seleccionada (s) es/son útil (es), pero está/ están poco alineada (s) o mal aplicada (s).	La (s) herramienta (s) TIC seleccionada (s) es/ son inadecuada (s), irrelevante (s) y su uso no evidencia (a) justificación pedagógica.
Innovación metodológica	Propone metodologías activas, creativas e inclusivas que transforman el aprendizaje.	Presenta enfoques novedosos que favorecen el aprendizaje.	Uso de metodologías tradicionales con elementos digitales.	Sin elementos innovadores o metodologías inapropiadas.
Factibilidad de implementación	Es completamente viable con recursos disponibles; considera contexto real.	Es viable con pequeños ajustes logísticos.	Su implementación requiere recursos o condiciones no disponibles.	Difícilmente viable sin cambios sustanciales.
Evaluación del aprendizaje	Incluye al menos seis instrumentos de evaluación del aprendizaje variados, claros y pertinentes a los objetivos del programa; se evalúa proceso y producto.	Incluye al menos cuatro instrumentos de evaluación del aprendizaje diferentes, los cuales son claros y son coherentes con los objetivos del programa.	Incluye solo dos instrumentos de evaluación del aprendizaje diferentes y la evaluación propuesta es parcial, poco específica o limitada en alcance.	No se define evaluación o es incoherente.
Inclusión y accesibilidad	Atiende diversidad (cultural, funcional, tecnológica); asegura equidad.	Considera elementos inclusivos básicos.	Tiene intención inclusiva, pero sin acciones concretas.	No contempla aspectos de inclusión o accesibilidad.
Resultados esperados	Resultados claros, medibles, con impacto esperado en aprendizaje.	Resultados coherentes y viables.	Resultados poco claros o difíciles de medir.	Resultados vagos o irrelevantes.

ESCALA DE VALORACIÓN

Puntaje Total Obtenido	Nivel de Evaluación	Descripción	Resultado
72 - 80 puntos	**Excelente**	El programa cumple completamente con los criterios de calidad pedagógica, tecnológica y metodológica.	Se recomienda la aplicación sin observaciones.
64 - 71 puntos	**Muy Bueno**	El programa es sólido, con aspectos destacados, aunque requiere realizar algunos ajustes menores.	Se recomienda la aplicación previa subsanación de ajustes menores.
56 - 63 puntos	**Bueno / Aceptable**	El programa requiere mejoras importantes para garantizar una implementación eficaz y el logro de los resultados del estudio.	Se recomienda solamente si se implementan las mejoras propuestas por parte del evaluador.
55 puntos o menos	**Insuficiente**	No cumple con los estándares mínimos. Presenta debilidades importantes en diseño, aplicación o viabilidad.	No se recomienda la aplicación y el programa debe ser reestructurado.

6. RESULTADOS DE LA EVALUACIÓN:

CRITERIO	MUY BUENO (10 PTS)	BUENO (7–9 PTS)	ACEPTABLE (4–6 PTS)	INSUFICIENTE (1–3 PTS)	JUSTIFICACIÓN	OBSERVACIÓN
Pertinencia pedagógica	10				Responde con precisión a una necesidad educativa y está alineado con el currículo nacional y las políticas institucionales.	
Claridad de objetivos	10				Los objetivos del programa son claros, medibles, alcanzables y coherentes con el propósito del estudio.	
Uso adecuado de TIC	10				Las herramientas TIC seleccionadas son pertinentes, innovadoras, accesibles y bien integradas.	
Innovación metodológica	10				Propone metodologías activas, creativas e inclusivas que transforman el aprendizaje.	
Factibilidad de implementación	10				Es completamente viable con recursos disponibles; considera contexto real.	
Evaluación del aprendizaje	10				Incluye al menos seis instrumentos de evaluación del aprendizaje variados, claros y pertinentes a los objetivos del programa; se evalúa proceso y producto.	

Inclusión y accesibilidad	10				Atiende diversidad (cultural, funcional, tecnológica); asegura equidad.	
Resultados esperados	10				Resultados claros, medibles, con impacto esperado en aprendizaje.	
TOTAL						80

PUNTAJE TOTAL OBTENIDO	NIVEL DE EVALUACIÓN	DESCRIPCIÓN	RESULTADO
80	**Excelente**	El programa cumple completamente con los criterios de calidad pedagógica, tecnológica y metodológica.	Se recomienda la aplicación sin observaciones.

Ciudad y fecha de evaluación: Chiclayo, 12 de abril del 2025



Mg. Janett Aracely Gonzales Flores

DNI: 73618769

FICHA DE EVALUACIÓN DE PROGRAMA DE INTERVENCIÓN POR MEDICIÓN DE JUICIO DE EXPERTOS

Estimado experto:

Le comunico que usted ha sido elegido para emitir su evaluación sobre el Programa de Intervención denominado "Desarrollo de habilidades prácticas de fundamentos de programación" el cual es parte del estudio titulado: **"Implementación de una aplicación educativa con realidad aumentada para mejorar el desarrollo de habilidades prácticas en el curso de Fundamentos de Programación"**.

La evaluación del programa en cuestión es de gran relevancia para alcanzar la validez requerida con el propósito que los resultados alcanzados a partir de su aplicación sean utilizados eficientemente; aportando a la línea de investigación en educación, como a sus aplicaciones.

De antemano, le agradezco su valiosísimo apoyo.

1. DATOS GENERALES DEL JUEZ

Apellidos y Nombres del experto:	Mg. Anthony Hans Delgado Chavarri
Grado profesional:	Maestría (X) Doctor ( )
Área de formación académica:	Maestría en dirección estratégica de tecnologías de la información
Áreas de experiencia profesional:	Docencia Universitaria, Consultor en Tecnologías de la Información (TI) y Microsoft Trainer Certified me permite ser un profesional altamente especializado en el ámbito de la tecnología, con un enfoque en ofrecer soluciones y asesoramiento a empresas en cuanto a infraestructura tecnológica, procesos digitales y optimización de sistemas
Entidad donde labora:	Universidad Tecnológica del Perú (UTP), CERTUS, New Horizons Perú, Universidad Privada del Norte (UPN)
Tiempo de experiencia profesional en el área:	2 a 4 años (x) Más de 5 años ( )
Experiencia en Investigación (deseable):	Consultor en Tecnologías de la Información (TI), asesor de proyectos de investigación.

2. PROPÓSITO DE LA EVALUACIÓN:

a. Validar lingüísticamente el Programa de Intervención: Desarrollo de habilidades prácticas de fundamentos de programación – Diseño Instruccional.
b. Evaluar la coherencia de los contenidos de acuerdo a sus dimensiones y a los criterios establecidos.

3. DATOS DEL INSTRUMENTO

Nombre del instrumento:	Desarrollo de habilidades prácticas de fundamentos de programación
Autor(es):	Ing. Hans Edgardo Delgado Díaz
Procedencia:	Elaboración propia
Administración:	Grupal
Período de aplicación:	6 semanas
Ámbito de aplicación:	Educativo

4. SOPORTE TEÓRICO

Dimensiones del instrumento:

DIMENSIONES	DEFINICIÓN
1. Comprensión de conceptos básicas de programación	Identifica las principales estructuras de un programa y realiza preguntas a partir de la identificación de las estructuras de un programa
2. Aplicación de estructuras de programación	Describe correctamente las estructuras de un programa
3. Solución de problemas mediante programación	Participa voluntariamente

5. INSTRUCCIONES PARA EL EXPERTO VALIDADOR:

Estimado experto:

En el anexo 01 se adjunta el programa de intervención titulado: **Desarrollo de habilidades prácticas de fundamentos de programación – Diseño Instruccional**, con el objetivo que usted pueda evaluarlo de acuerdo a la siguiente matriz:

CRITERIO	MUY BUENO (10 PTS)	BUENO (7–9 PTS)	ACEPTABLE (4–6 PTS)	INSUFICIENTE (1–3 PTS)
Pertinencia pedagógica	Responde con precisión a una necesidad educativa y está alineado con el currículo nacional y las políticas institucionales.	Responde satisfactoriamente a una necesidad relevante; hay buena alineación curricular.	La necesidad educativa se evidencia parcialmente y la alineación curricular es parcial.	No responde a una necesidad educativa determinada y, además, no está alineado con el currículo.
Claridad de objetivos	Los objetivos del programa son claros, medibles, alcanzables y coherentes con el propósito del estudio.	Los objetivos del programa son claros, medibles, alcanzables y responden parcialmente al propósito del estudio.	Los objetivos están poco definidos, son difícilmente medibles y no guardan coherencia con el propósito del programa.	No presenta objetivos.

Uso adecuado de TIC	La (s) herramienta (s) TIC seleccionada (s) es/son pertinente (s), innovadora(s), accesible(s) y bien integrada (s).	La (s) herramienta (s) TIC seleccionada (s) evidencia (a) un uso funcional y relevante de TIC y se integra al programa de manera adecuada.	La (s) herramienta (s) TIC seleccionada (s) es/son útil (es), pero está/ están poco alineada (s) o mal aplicada (s).	La (s) herramienta (s) TIC seleccionada (s) es/ son inadecuada (s), irrelevante (s) y su uso no evidencia (a) justificación pedagógica.
Innovación metodológica	Propone metodologías activas, creativas e inclusivas que transforman el aprendizaje.	Presenta enfoques novedosos que favorecen el aprendizaje.	Uso de metodologías tradicionales con elementos digitales.	Sin elementos innovadores o metodologías inapropiadas.
Factibilidad de implementación	Es completamente viable con recursos disponibles; considera contexto real.	Es viable con pequeños ajustes logísticos.	Su implementación requiere recursos o condiciones no disponibles.	Difícilmente viable sin cambios sustanciales.
Evaluación del aprendizaje	Incluye al menos seis instrumentos de evaluación del aprendizaje variados, claros y pertinentes a los objetivos del programa; se evalúa proceso y producto.	Incluye al menos cuatro instrumentos de evaluación del aprendizaje diferentes, los cuales son claros y son coherentes con los objetivos del programa.	Incluye solo dos instrumentos de evaluación del aprendizaje diferentes y la evaluación propuesta es parcial, poco específica o limitada en alcance.	No se define evaluación o es incoherente.
Inclusión y accesibilidad	Atiende diversidad (cultural, funcional, tecnológica); asegura equidad.	Considera elementos inclusivos básicos.	Tiene intención inclusiva, pero sin acciones concretas.	No contempla aspectos de inclusión o accesibilidad.
Resultados esperados	Resultados claros, medibles, con impacto esperado en aprendizaje.	Resultados coherentes y viables.	Resultados poco claros o difíciles de medir.	Resultados vagos o irrelevantes.

ESCALA DE VALORACIÓN

Puntaje Total Obtenido	Nivel de Evaluación	Descripción	Resultado
72 - 80 puntos	**Excelente**	El programa cumple completamente con los criterios de calidad pedagógica, tecnológica y metodológica.	Se recomienda la aplicación sin observaciones.
64 - 71 puntos	**Muy Bueno**	El programa es sólido, con aspectos destacados, aunque requiere realizar algunos ajustes menores.	Se recomienda la aplicación previa subsanación de ajustes menores.
56 - 63 puntos	**Bueno / Aceptable**	El programa requiere mejoras importantes para garantizar una implementación eficaz y el logro de los resultados del estudio.	Se recomienda solamente si se implementan las mejoras propuestas por parte del evaluador.
55 puntos o menos	**Insuficiente**	No cumple con los estándares mínimos. Presenta debilidades importantes en diseño, aplicación o viabilidad.	No se recomienda la aplicación y el programa debe ser reestructurado.

6. RESULTADOS DE LA EVALUACIÓN:

CRITERIO	MUY BUENO (10 PTS)	BUENO (7–9 PTS)	ACEPTABLE (4–6 PTS)	INSUFICIENTE (1–3 PTS)	JUSTIFICACIÓN	OBSERVACIÓN
Pertinencia pedagógica	10				Responde con precisión a una necesidad educativa y está alineado con el currículo nacional y las políticas institucionales.	
Claridad de objetivos	10				Los objetivos del programa son claros, medibles, alcanzables y coherentes con el propósito del estudio.	
Uso adecuado de TIC	10				Las herramientas TIC seleccionadas son pertinentes, innovadoras, accesibles y bien integradas.	
Innovación metodológica	10				Propone metodologías activas, creativas e inclusivas que transforman el aprendizaje.	
Factibilidad de implementación	10				Es completamente viable con recursos disponibles; considera contexto real.	
Evaluación del aprendizaje	10				Incluye al menos seis instrumentos de evaluación del aprendizaje variados, claros y pertinentes a los objetivos del programa; se evalúa proceso y producto.	

Inclusión y accesibilidad	10				Atiende diversidad (cultural, funcional, tecnológica); asegura equidad.	
Resultados esperados	10				Resultados claros, medibles, con impacto esperado en aprendizaje.	
TOTAL						80

PUNTAJE TOTAL OBTENIDO	NIVEL DE EVALUACIÓN	DESCRIPCIÓN	RESULTADO
80	**Excelente**	El programa cumple completamente con los criterios de calidad pedagógica, tecnológica y metodológica.	Se recomienda la aplicación sin observaciones.

Ciudad y fecha de evaluación: Chiclayo, 12 de abril del 2025



Mg. Anthony Hans Delgado Chavarri

Diseño Instruccional

Título	Desarrollo de habilidades prácticas de fundamentos de programación
Duración	6 semanas
Perfil de la audiencia (estudiantes) Características y necesidades	Características: • **Edad:** 18-30 años, estudiantes universitarios o recién graduados. • **Formación académica:** Estudiantes de ingeniería, ciencias de la computación, ciencias aplicadas, administración o áreas relacionadas con la tecnología. • **Experiencia previa:** Algunos pueden tener conocimientos básicos de programación, mientras que otros pueden ser completamente nuevos en la materia. La mayoría tiene una comprensión general de matemáticas, lógica y resolución de problemas. Necesidades: • Necesitan una comprensión más profunda de los principios fundamentales de la programación. • Requieren una aplicación práctica de estos conceptos para fortalecer su habilidad para escribir código y resolver problemas de manera efectiva. • Buscan mejorar su capacidad para trabajar con herramientas de programación modernas y colaborar en proyectos reales.
Justificación  	El curso es fundamental en la formación de habilidades prácticas de programación, esenciales en un mundo cada vez más digital. A medida que las empresas y sectores industriales buscan automatizar procesos y optimizar su eficiencia, la programación se ha convertido en una competencia transversal que mejora la empleabilidad. Este curso se centra en la adquisición de habilidades prácticas para que los estudiantes puedan aplicar los conceptos de programación en proyectos reales y desarrollar soluciones eficaces. Beneficiarios: • **Estudiantes de tecnología:** Aquellos que desean fortalecer sus competencias en programación con aplicaciones prácticas. • **Profesionales en áreas no técnicas:** Profesionales que quieren adquirir habilidades en programación para incorporar la automatización en su trabajo. • **Desarrolladores principiantes:** Estudiantes o novatos que desean mejorar su capacidad para crear programas efectivos mediante la práctica intensiva. El curso proporcionará a los estudiantes una base sólida en los fundamentos de programación, pero con un énfasis en la aplicación práctica de esos conceptos. Además, los estudiantes estarán preparados para

	abordar proyectos más complejos, permitiéndoles desarrollar programas con eficiencia y claridad, lo que les otorga una ventaja competitiva en el mercado laboral. El propósito de este curso es enseñar a los estudiantes a aplicar los principios fundamentales de la programación de manera práctica. Se busca que los estudiantes no solo comprendan los conceptos, sino que también sean capaces de utilizar estos conceptos para resolver problemas reales, trabajar en proyectos y colaborar en entornos profesionales.

Metodología
 
 	
	La metodología será **activa, práctica y orientada a proyectos**, con un fuerte enfoque en el aprendizaje basado en la resolución de problemas. Durante el curso, se combinarán clases teóricas con ejercicios prácticos, laboratorios de programación y trabajo en equipo para fomentar la colaboración y el desarrollo de habilidades de resolución de problemas.
	
	• **Clases teóricas:** Explicación de los conceptos fundamentales de programación con ejemplos prácticos.
	• **Clases prácticas:** Sesiones donde los estudiantes resolverán ejercicios y proyectos aplicados.
	• **Estudio individual:** Tareas y ejercicios de refuerzo para profundizar en los temas.
	• **Trabajo colaborativo:** Desarrollo de proyectos grupales donde se resolverán problemas reales de programación.
	• **Evaluaciones:** Cuestionarios, tareas de codificación y un proyecto final donde los estudiantes deben crear un programa funcional.
	
	Recursos y Materiales
	
	• **Software:**
	  ○ Entorno de desarrollo integrado (IDE) como **Visual Studio Code** o **PyCharm**.
	  ○ Lenguaje de programación: **Python**, por su simplicidad y amplia aplicabilidad.
	  ○ **GitHub** para el control de versiones y colaboración en proyectos grupales.
	  ○ **Aplicativo Móvil con Realidad Aumentada** elaborado en Unity de fundamentos de programación con Python
	• **Material de lectura:**
	  ○ *"Automate the Boring Stuff with Python"*, de Al Sweigart (enfoque práctico de programación).
	  ○ Notas y tutoriales proporcionados por el instructor.
	• **Plataformas complementarias:**
	  ○ **Khan Academy**, **Codecademy**, **LeetCode**, y otras plataformas para complementar el aprendizaje.
	  ○ **Foros en línea** como Stack Overflow para la resolución de dudas.
	  ○ **Documentación oficial de Python** y otras herramientas utilizadas en el curso.

Propósito	El propósito de este curso es que los estudiantes: • Desarrollen habilidades prácticas en programación utilizando Python. • Aprendan a implementar soluciones a problemas del mundo real mediante el diseño y la escritura de código eficiente y efectivo. • Dominen las herramientas y prácticas modernas de desarrollo de software, incluyendo el control de versiones y el trabajo colaborativo en proyectos. Este curso tiene como objetivo transformar a los estudiantes de principiantes a profesionales básicos, capaces de desarrollar software funcional y aplicar programación en diferentes contextos.
Conocimientos  	Conocimientos para el Curso de Desarrollo de Habilidades Prácticas de Fundamentos de Programación 1. **Conceptos Básicos de Programación:** o Sintaxis y semántica de un lenguaje de programación. o Tipos de datos primitivos: enteros, flotantes, cadenas, booleanos. o Variables y operadores. 2. **Estructuras de Control:** o Condicionales: if, else, elif. o Bucles: for, while. 3. **Funciones:** o Definición y llamada a funciones. o Parámetros y valores de retorno. o Concepto de alcance de variables. 4. **Estructuras de Datos:** o Listas, tuplas, diccionarios. o Métodos de manipulación y acceso a estructuras de datos. o Aplicación de estructuras de datos en la resolución de problemas. 5. **Manejo de Errores y Excepciones:** o Uso de bloques try-except para el manejo de excepciones. o Propósito y tipos de excepciones comunes. 6. **Programación Orientada a Objetos (OOP):** o Clases, objetos, atributos y métodos. o Herencia, polimorfismo, encapsulación. 7. **Manejo de Archivos:** o Lectura y escritura de archivos de texto. o Manejo de archivos CSV y JSON.
 	8. **Desarrollo de Algoritmos:** o Algoritmos básicos: búsqueda y ordenación. o Análisis de la complejidad algorítmica. 9. **Recursión:** o Fundamentos de recursión. o Resolución de problemas con recursión. 10. **Trabajo Colaborativo en Proyectos:** o Gestión de proyectos en grupo. o Uso de herramientas de control de versiones como **Git**. 11. **Desarrollo de Proyecto:** o Diseño, implementación y prueba de proyectos. o Presentación de soluciones ante una audiencia. 12. **Optimización de Código:** o Mejora de la eficiencia del código. o Refactorización y pruebas unitarias.

Competencias	**Sesiones de aprendizaje (Título)**	**Tiempo aproximado de realización**	**Evidencia (Producto / actuación)**	**Instrumento / Criterios de evaluación**	**Recursos necesarios para realizar la actividad**	**Colaboración (si consideras que es viable)**
1. Comprender la sintaxis básica y estructuras de control de un lenguaje de programación.	Sesión 1: (Sintaxis básica y control de flujo)	90 minutos	El estudiante desarrollará un programa que utilice operadores, variables, condicionales (if y else), y bucles (for y while) para resolver un problema sencillo.	Rúbrica de evaluación del código (correctitud, uso adecuado de estructuras, eficiencia).	IDE (Visual Studio Code, PyCharm), documentación sobre estructuras de control. Jupyter Notebooks Canva Replit (para ejercicios prácticos)	No aplicable, trabajo individual.
2. Aplicar funciones en la resolución de problemas prácticos.	Sesión 2: (Funciones)	90 minutos	El estudiante creará un programa modular utilizando funciones con	Evaluación del código (estructura modular, legibilidad, uso	IDE, tutoriales de funciones. Jupyter Notebooks	No aplicable, trabajo individual.



			parámetros y valores de retorno para resolver un problema específico.	adecuado de funciones).	Replit (para ejercicios prácticos)	
3. Gestionar estructuras de datos como listas, tuplas y diccionarios.	Sesión 3: (Estructuras de datos)	90 minutos	El estudiante desarrollará un programa que utilice listas, tuplas y diccionarios para almacenar y manipular datos (ejemplo: gestionar una lista de estudiantes).	Rúbrica de evaluación (correctitud de las estructuras, uso adecuado de los métodos).	IDE, documentación sobre estructuras de datos en Python. Jupyter Notebooks Replit (para ejercicios prácticos) Google Colab	No aplicable, trabajo individual.
4. Gestionar excepciones en un programa para garantizar la estabilidad del código.	Sesión 4: (Manejo de errores y excepciones)	90 minutos	El estudiante implementará manejo de excepciones en un programa para capturar errores como entradas no válidas.	Rúbrica de evaluación (uso adecuado de try-except, manejo de excepciones comunes).	IDE, tutorial sobre manejo de excepciones Jupyter Notebooks Replit (para ejercicios prácticos) Google Colab	No aplicable, trabajo individual.
5. Implementar conceptos básicos de la programación orientada a objetos (OOP)	Sesión 5: (Introducción y profundización en OOP)	90 minutos	El estudiante creará una clase simple que incluya atributos y métodos, implementando características como herencia y polimorfismo	Rúbrica de evaluación (correctitud en la implementación de clases, herencia, polimorfismo).	IDE, documentación sobre clases en Python Jupyter Notebooks	No aplicable, trabajo individual
6. Desarrollar algoritmos sencillos de búsqueda y ordenación.	Sesión 6: (Algoritmos de búsqueda y ordenación)	90 minutos	El estudiante implementará un algoritmo de búsqueda binaria y uno de ordenación	Rúbrica de evaluación (correctitud de los algoritmos,	IDE, recursos sobre algoritmos de búsqueda y ordenación.	No aplicable, trabajo individual.



			(por ejemplo, ordenación por burbuja).	análisis de complejidad).	Jupyter Notebooks	
7. Implementar soluciones recursivas para problemas específicos.	Sesión 7: (Recursión)	90 minutos	El estudiante resolverá un problema utilizando una función recursiva (por ejemplo, calcular factorial de un número).	Rúbrica de evaluación (correctitud, eficiencia de la solución recursiva).	IDE, ejemplos de funciones recursivas Jupyter Notebooks	No aplicable, trabajo individual.
8. Desarrollar programas que interactúan con archivos (lectura y escritura).	Sesión 8: (Manejo de archivos)	90 minutos	El estudiante desarrollará un programa que lea y escriba datos en archivos de texto o CSV.	Evaluación del código (funcionamiento correcto de lectura y escritura de archivos).	IDE, tutoriales sobre manejo de archivos en Python. Jupyter Notebooks	No aplicable, trabajo individual.
9. Desarrollar proyectos colaborativos utilizando herramientas de control de versiones.	Sesión 9: (Trabajo colaborativo y desarrollo de proyectos)	90 minutos	El estudiante trabajará en equipo para desarrollar un proyecto, utilizando Git y GitHub para gestionar el código.	Rúbrica de evaluación (calidad del código, uso adecuado de Git, colaboración en equipo).	Git, GitHub, IDE, recursos de control de versiones. Jupyter Notebooks	Trabajo en equipo (mínimo 2 estudiantes).
10. Optimizar y mejorar la eficiencia del código	Sesión 10: (Optimización de código)	90 minutos	El estudiante refactorizará un programa para mejorar su eficiencia, simplificando el código y eliminando redundancias	Rúbrica de evaluación (eficiencia del código, calidad y simplicidad)	IDE, tutoriales de optimización de código. Jupyter Notebooks	No aplicable, trabajo individual.
11. Presentar un proyecto de programación a un público objetivo	Sesión 11: (Presentación del proyecto final)	90 minutos	El estudiante presentará su proyecto final, explicando el	Rúbrica de presentación (claridad en la exposición,	Herramientas de presentación (PowerPoint, etc.),	No aplicable, trabajo individual.



			proceso de desarrollo, las soluciones implementadas y los resultados obtenidos.	calidad de la solución presentada).	documentación del proyecto Canva Trello o Asana	
12.Evaluar y mejorar el proyecto final basado en retroalimentación.	Sesión 12: (Evaluación final y revisión)	90 minutos	El estudiante mejorará su proyecto final basado en retroalimentación recibida, realizando ajustes y correcciones en el código	Rúbrica de evaluación (mejora del código, incorporación de feedback)	IDE, retroalimentación del instructor y compañeros Canva Trello o Asana	No aplicable, trabajo individual.

Lista de verificación



			Sí ☑	No ☒
	Título	Redacté un título del curso corto, e incluí las palabras clave.	✓	
	Duración	Señalé el tiempo que se invertirá en el estudio de este curso, en meses, semanas, días u horas.	✓	
	Perfil de la audiencia	Describí el perfil deseable de la audiencia potencial del curso en línea (experiencia, intereses, conocimientos, habilidades, etc.).	✓	
	Justificación	Expliqué a qué obedece el diseño del curso, cuál es su importancia o quiénes se beneficiarán con él.	✓	
	Metodología	Describí de manera general la forma en que el estudiante trabajará.	✓	
	Propósito	Señalé puntualmente el propósito que lograrán los participantes después de estudiar el curso en línea / presencial.	✓	
	Conocimientos	Cuidé que existiera una correspondencia lógica entre el conocimiento y el propósito.	✓	
	Sesiones de aprendizaje	Determiné las principales acciones que los participantes deben realizar para lograr el propósito de aprendizaje.	✓	
	Tiempo aproximado de realización	Señalé un estimado del tiempo que el estudiante invertirá en la realización de cada actividad.	✓	
	Evidencia (producto / actuación)	Seleccioné la evidencia acorde al propósito de aprendizaje.	✓	
	Instrumento / Criterios de evaluación	Determiné el instrumento de evaluación a utilizar e incluí los aspectos que serán considerados en la evaluación de la evidencia.	✓	
	Recursos	Sugerí recursos en formatos diversos, asequibles, accesibles y pertinentes para la realización de la sesión.	✓	
	Colaboración	Determiné si alguna o algunas actividades tienen posibilidades de fortalecer la colaboración y la interacción entre los estudiantes.	✓	
Revisión general	Alineación	Logré que el propósito se alinease con las sesiones y la evaluación del aprendizaje.	✓	
	Claridad	Logré que la información contenida en la estructura del curso fuera comprensible considerando el área de conocimiento y la audiencia.	✓	
	Validez	Logré incorporar un conocimiento aceptable y reconocido.	✓	
	Fiabilidad	Logré integrar un conocimiento con una relativa estabilidad temporal, que corresponde con lo que sucede en la actualidad y que evitará cambios en el corto plazo.	✓	
	Exactitud	Logré que la información contenida en la estructura del curso sea precisa y sin errores.	✓	
	Calidad	Logré que la información contenida en la estructura tenga solidez pedagógica considerando el área de conocimiento y la audiencia.	✓	



Sesión de aprendizaje 1

I. DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 1



II. TITULO

"Comprender la sintaxis básica y estructuras de control de un lenguaje de programación"

III. PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Comprender la sintaxis básica y las estructuras de control de un lenguaje de programación (en este caso, Python), aplicando condicionales y bucles para resolver problemas prácticos de programación.	Reconocer y aplicar la sintaxis básica de un lenguaje de programación, comprendiendo su estructura. Utilizar operadores y variables para realizar operaciones básicas Emplear estructuras de control como condicionales (`if`, `else`, `elif`) y bucles (`for`, `while`) en la resolución de problemas Desarrollar programas sencillos que utilicen las estructuras de control aprendidas para ejecutar tareas repetitivas o condicionales	**Correctitud en la sintaxis:** El estudiante debe ser capaz de escribir código sin errores de sintaxis. **Uso adecuado de variables y operadores:** El estudiante debe demostrar la correcta declaración, inicialización y uso de variables. **Implementación correcta de estructuras de control:** El estudiante debe implementar condicionales y bucles de manera adecuada para resolver problemas sencillos. **Eficiencia en la solución:** El código debe ser eficiente, con una correcta utilización de bucles para evitar redundancias y simplificar la solución.	El estudiante desarrollará un programa sencillo que resuelva un problema específico utilizando condicionales y bucles.	**Rúbrica de evaluación:** Correctitud de la sintaxis (20%). Uso adecuado de variables y operadores (20%). Implementación de estructuras de control (30%). Eficiencia y optimización del código (30%).
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	

Desarrollar habilidades de trabajo en equipo y comunicación efectiva para la resolución de problemas	• Participar activamente en actividades de grupo, contribuyendo con ideas y soluciones. • Comunicar de manera clara y efectiva las ideas de programación y las soluciones desarrolladas. • Respetar las opiniones de los demás y colaborar para mejorar la solución grupal.	**Colaboración en equipo:** El estudiante muestra disposición para colaborar, aportar ideas y respetar las contribuciones de los compañeros. **Comunicación efectiva:** El estudiante es capaz de explicar su proceso de resolución de problemas de forma clara y comprensible para los demás.	El estudiante participará en una actividad grupal, donde se resolverá un problema y presentará el código y la solución grupal.	
Enfoque Transversal				
	El trabajo en equipo y la comunicación efectiva entre los estudiantes como herramientas fundamentales para la resolución de problemas de programación.			

IV. ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio (20 minutos)	**Estrategias:** • **Activación de conocimientos previos:** Realizar una breve revisión de lo que son las estructuras de control en programación y cómo pueden aplicarse en el mundo real. Preguntar a los estudiantes ejemplos de situaciones cotidianas que requieran tomar decisiones (por ejemplo, escoger entre opciones, tomar decisiones repetitivas, etc.). • **Presentación del tema:** Introducción a la sintaxis básica de Python: variables, operadores, estructuras de control (condicionales y bucles). Se presentarán ejemplos sencillos y se explicará su uso paso a paso. **Actividades:** 1. **Discusión en clase:** Explicar brevemente qué son las estructuras de control (condicionales y bucles). Utilizar ejemplos visuales para representar cómo funcionan (ejemplo: si el clima es soleado, entonces salgo a caminar, de lo contrario, me quedo en casa).	Proyector o pantalla para mostrar ejemplos de código. IDEs instalados (Visual Studio Code, PyCharm, o Replit). Documentación básica sobre Python (W3Schools, Documentación oficial de Python).

	2. **Ejemplo práctico en vivo:** El docente realiza un ejercicio sencillo en el IDE, mostrando cómo escribir un programa que utilice condicionales y bucles. Los estudiantes siguen el ejemplo en sus computadoras.	
Desarrollo (50 minutos)	**Estrategias:** • **Exploración guiada:** Los estudiantes practicarán creando sus propios programas con apoyo del docente. Se les proporcionarán problemas sencillos que deberán resolver utilizando las estructuras de control. • **Trabajo en equipo:** Los estudiantes trabajarán en parejas o grupos pequeños para fomentar la colaboración en la solución de problemas, asegurándose de aplicar las estructuras de control correctamente. **Actividades:** 1. **Ejercicio individual guiado:** Los estudiantes completan ejercicios sencillos donde deben crear programas que empleen condicionales (if, else) y bucles (for, while). 2. **Problema en grupo:** En equipos, los estudiantes resuelven un problema práctico, como desarrollar un programa que simule un proceso de decisión repetitivo (por ejemplo, una tienda que verifica si un artículo está en stock y decide qué hacer dependiendo de su disponibilidad).	IDEs instalados o plataformas en línea como Replit. Ejercicios impresos o en Google Drive. Pizarras digitales para resolver problemas en conjunto.
Cierre (20 minutos)	**Estrategias:** • **Reflexión y retroalimentación:** Reflexionar sobre lo aprendido en la clase. El docente solicita que algunos estudiantes compartan su código y expliquen cómo resolvieron los problemas. El grupo proporciona retroalimentación. • **Presentación del trabajo en equipo:** Cada grupo presenta su solución al problema de programación que resolvieron, explicando cómo implementaron las estructuras de control. **Actividades:** 1. **Revisión grupal:** Los estudiantes presentan sus soluciones, mientras el docente y los compañeros ofrecen sugerencias y correcciones. 2. **Preguntas y respuestas:** El docente responde dudas sobre las dificultades encontradas en el proceso de resolución de problemas.	IDE para mostrar el código. Documentos de retroalimentación. Herramientas de colaboración (Google Drive, Trello, etc.).



Sesión de aprendizaje 2

I. DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 1



II. TITULO

"Aplicar funciones en la resolución de problemas prácticos"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Aplicar funciones en la resolución de problemas prácticos, comprendiendo su estructura, parámetros, valores de retorno y reutilización para la creación de programas modulares y eficientes.	**Identificar y aplicar funciones**: Los estudiantes serán capaces de crear funciones que reciban parámetros y devuelvan valores, permitiendo reutilizar bloques de código. **Aplicar el modularidad en la programación**: Desarrollar programas modulares donde las tareas se descomponen en funciones pequeñas, reutilizables y de fácil mantenimiento. **Comprender el concepto de paso de parámetros y retorno de valores**: Entender cómo las funciones pueden recibir datos de entrada y devolver resultados, facilitando la resolución de problemas prácticos. **Resolver problemas prácticos mediante funciones**: Desarrollar	**Correcta implementación de funciones**: El estudiante debe ser capaz de definir funciones con parámetros y valores de retorno correctamente. **Uso adecuado de funciones en la resolución de problemas**: El estudiante debe demostrar cómo las funciones ayudan a organizar el código y resolver problemas de manera más eficiente. **Eficiencia y claridad del código**: El código debe estar bien organizado, con funciones reutilizables que eviten la repetición de código. **Aplicación de conceptos teóricos en ejercicios prácticos**: El estudiante debe aplicar los conceptos de funciones (parámetros, retorno, modularidad) en ejercicios prácticos de programación.	Los estudiantes desarrollarán un programa práctico que resuelva un problema complejo utilizando varias funciones.	**Rúbrica de evaluación:** Definición correcta de funciones (30%). Uso adecuado de parámetros y retorno de valores (30%). Aplicación de funciones en la resolución de problemas (30%). Eficiencia y claridad del código (10%).

	soluciones para problemas específicos utilizando funciones y módulos predefinidos en Python.			
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	
Desarrollar habilidades de comunicación efectiva para la presentación y explicación de soluciones programáticas a través de la programación modular.	**Explicar de manera clara y concisa**: Los estudiantes deben ser capaces de explicar el propósito y funcionamiento de las funciones que crean. **Colaboración en equipo para la creación de funciones**: Los estudiantes deben trabajar en equipo para diseñar y escribir funciones que resuelvan partes de un problema más grande. **Organizar el trabajo de forma estructurada**: Los estudiantes deben organizar su trabajo de manera lógica, utilizando funciones y dividiendo problemas complejos en tareas pequeñas y manejables.	**Claridad en la explicación**: El estudiante debe ser capaz de comunicar de manera comprensible cómo sus funciones resuelven el problema propuesto. **Colaboración y trabajo en equipo**: Los estudiantes deben mostrar un trabajo colaborativo efectivo, organizando y distribuyendo tareas en función de las funciones a desarrollar. **Organización del código**: El código debe ser modular y estructurado, con funciones bien definidas que resuelvan una parte del problema de manera clara.	El estudiante participará en la creación de un conjunto de funciones que resuelvan un problema práctico, presentando y explicando su código en una pequeña presentación grupal.	
Enfoque Transversal				
	La comunicación efectiva y el trabajo en equipo como habilidades esenciales para resolver problemas complejos en programación.			

ANTHONY HANS DELGADO CHAVARRI
INGENIERO DE SISTEMAS Y COMPUTACION
REG. CIP. 186247

IV.ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio	**Estrategias:** • **Activación de conocimientos previos:** Realizar una revisión rápida sobre los conceptos básicos de programación vistos previamente	Proyector o pantalla para mostrar ejemplos de código.

(20 minutos)	(variables, condicionales, bucles) y presentar la idea de modularidad en programación a través de funciones. • **Presentación teórica:** El docente introduce el concepto de función en programación, su estructura (parámetros, cuerpo, valor de retorno), y cómo las funciones ayudan a descomponer problemas complejos en partes más pequeñas y manejables. **Actividades:** 1. **Presentación de conceptos:** El docente explica cómo crear funciones en Python, el paso de parámetros y retorno de valores. A continuación, se realiza un ejemplo básico, como una función que calcule el área de un círculo. 2. **Ejemplo práctico guiado:** Los estudiantes crean una función en sus computadoras para realizar un cálculo sencillo (por ejemplo, la suma de dos números). El docente guía paso a paso este proceso.	IDE (Visual Studio Code, PyCharm, Replit). Documentación básica sobre funciones en Python (W3Schools, documentación oficial de Python).
Desarrollo (50 minutos)	**Estrategias:** • **Trabajo práctico con supervisión:** Los estudiantes comienzan a trabajar en la creación de funciones más complejas, desglosando un problema práctico en partes más pequeñas, y cada grupo o estudiante se enfoca en una función. • **Trabajo en equipo:** Los estudiantes se dividen en grupos y cada uno debe escribir funciones que contribuyan a una solución general (por ejemplo, un programa de gestión de tareas con varias funciones que calculen tiempos, almacene datos, etc.). **Actividades:** 1. **Ejercicio individual:** Los estudiantes deben escribir una función que resuelva un problema sencillo, como la conversión de grados Celsius a Fahrenheit. El docente revisa y da retroalimentación. **2. Ejercicio en grupo:** Los estudiantes trabajan en equipo para crear un programa modular utilizando al menos tres funciones. Pueden elegir un problema como "gestionar un inventario de productos", donde cada función tenga un propósito específico (añadir producto, eliminar producto, mostrar inventario).	IDE (Visual Studio Code, PyCharm, Replit). Documentación de funciones en Python. Ejercicios en Google Drive o papel. 
Cierre (20 minutos)	**Estrategias:** • **Reflexión y retroalimentación:** El docente revisa los trabajos de los estudiantes, corrige errores comunes y fomenta la discusión sobre cómo las funciones ayudan a hacer que el código sea más modular y fácil de mantener. • **Presentación de soluciones:** Cada grupo presenta su solución al problema, explicando las funciones que crearon y cómo contribuyen al programa general. Los compañeros y el docente ofrecen retroalimentación.	IDE para mostrar el código. Documentos de retroalimentación. Herramientas de colaboración (Google Drive, Trello).

	Actividades: 1. **Revisión de soluciones en grupo:** Cada grupo presenta su solución y el docente hace comentarios sobre la estructura modular y la eficiencia de las funciones creadas. 2. **Preguntas y respuestas:** Los estudiantes hacen preguntas sobre cómo mejorar o modificar las funciones, y el docente da recomendaciones.	



Sesión de aprendizaje 3



I.DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 2

II.TITULO

"Gestionar estructuras de datos como listas, tuplas y diccionarios en Python"

III. PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Gestionar de manera efectiva estructuras de datos como listas, tuplas y diccionarios en Python, aplicando operaciones básicas para organizar, modificar y manipular datos en la solución de problemas prácticos.	**Seleccionar estructuras de datos apropiadas:** Los estudiantes serán capaces de identificar el tipo de estructura de datos (lista, tupla o diccionario) más adecuada según el contexto y el tipo de información a gestionar. **Aplicar operaciones sobre listas, tuplas y diccionarios:** Los estudiantes podrán realizar operaciones de inserción, eliminación, modificación y acceso a los datos dentro de las estructuras. **Solucionar problemas prácticos**	**Selección adecuada de estructuras de datos:** El estudiante debe demostrar la capacidad para elegir el tipo de estructura más adecuado (lista, tupla o diccionario) según el problema planteado. **Correcta aplicación de operaciones básicas:** El estudiante debe aplicar correctamente operaciones básicas sobre las estructuras de datos, como agregar, eliminar, modificar o acceder a elementos. **Resolución de problemas mediante estructuras de datos:** El estudiante debe resolver un problema práctico utilizando una combinación de estructuras de datos de manera eficiente. **Organización y legibilidad del código:** El estudiante debe entregar un código limpio, legible y bien estructurado, que utilice las estructuras de datos de manera efectiva.	Los estudiantes desarrollarán un programa en Python que resuelva un problema práctico (por ejemplo, una base de datos de estudiantes, un sistema de inventario o gestión de tareas) utilizando listas, tuplas y diccionarios.	**Rúbrica de evaluación:** Elección adecuada de la estructura de datos (25%). Correcta implementación de operaciones sobre las estructuras de datos (30%). Eficiencia en la resolución del problema (25%). Calidad y claridad del código (20%).

	mediante estructuras de datos: Los estudiantes deberán aplicar las operaciones de las estructuras de datos para resolver problemas prácticos de programación.	 ANTHONY HANS DELGADO CHAVARRI INGENIERO DE SISTEMAS Y COMPUTACION REG. CIP. 186247		
	Optimizar el uso de las estructuras de datos: Los estudiantes aprenderán a seleccionar y utilizar las estructuras de datos de manera eficiente para resolver problemas de manera óptima.			
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	
Desarrollar habilidades de resolución de problemas mediante la organización de datos y trabajo en equipo para gestionar información de manera eficiente.	**Organizar y presentar soluciones de manera clara:** Los estudiantes deben organizar su código de manera eficiente y presentarlo de manera clara, explicando las elecciones hechas sobre las estructuras de datos. **Colaborar en el desarrollo de soluciones con estructuras de datos:** Los estudiantes deben trabajar en equipo para desarrollar	**Claridad y organización del trabajo en equipo:** El estudiante debe demostrar la capacidad para organizar el trabajo y colaborar eficazmente en el equipo. **Explicación clara y precisa de las elecciones:** El estudiante debe ser capaz de explicar cómo y por qué seleccionaron determinadas estructuras de datos para resolver el problema. **Colaboración en la implementación de la solución:** El estudiante debe mostrar cómo colaboró en el desarrollo del código y la implementación de las estructuras de datos dentro de su equipo.	El estudiante participará en una actividad grupal en la que desarrollarán y presentarán un programa que utilice listas, tuplas y diccionarios para resolver un problema práctico. La presentación incluirá una explicación de cómo se utilizaron las estructuras de datos y la	

	soluciones prácticas utilizando estructuras de datos. **Comunicación efectiva de la solución:** Los estudiantes deben ser capaces de comunicar de manera efectiva las decisiones tomadas en el uso de las estructuras de datos y cómo estas contribuyen a la solución.		justificación de las decisiones tomadas.	
Enfoque Transversal				
	Fomento del trabajo colaborativo y la comunicación efectiva como herramientas para resolver problemas complejos mediante la gestión eficiente de estructuras de datos.			

IV.ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio (20 minutos)	**Estrategias:** • **Activación de conocimientos previos:** Se realiza una revisión rápida de las estructuras de datos vistas previamente (listas, tuplas y diccionarios) y las operaciones que se pueden realizar sobre ellas. El docente también repasa casos prácticos de cuándo es más adecuado usar una u otra estructura. • **Presentación teórica y demostración:** El docente explica y muestra ejemplos de operaciones más avanzadas con listas, tuplas y diccionarios, como el uso de funciones integradas (append(), pop(), keys(), etc.), el acceso a elementos de diccionarios por clave y la modificación de elementos en listas y tuplas. **Actividades:** 1. **Presentación y discusión de ejemplos:** El docente presenta ejemplos prácticos de código en los que se utilizan las estructuras	Proyector o pantalla para mostrar ejemplos de código. IDE (Visual Studio Code, PyCharm, Replit). Documentación sobre listas, tuplas y diccionarios en Python.

	de datos para resolver un problema real, como organizar un sistema de inventario o almacenar y modificar información de estudiantes. 2. **Resolución conjunta de un ejercicio:** El docente guía a los estudiantes a través de un ejercicio práctico que implica el uso de listas, tuplas y diccionarios, resolviendo un problema específico. Ejemplo: crear un sistema para gestionar tareas con fechas (listas y tuplas) y responsables (diccionarios).	
Desarrollo (50 minutos	**Estrategias:** • **Trabajo práctico supervisado:** Los estudiantes trabajan en ejercicios prácticos para manipular y gestionar estructuras de datos. El docente circula por el aula, brindando retroalimentación y asistencia a los estudiantes. • **Trabajo en equipo:** Los estudiantes se dividen en equipos y desarrollan un programa práctico que resuelva un problema utilizando listas, tuplas y diccionarios, como gestionar un sistema de reservas de habitaciones, un catálogo de productos, o un registro de clientes. **Actividades:** 1. **Ejercicio individual guiado:** Los estudiantes deben desarrollar una solución individual en Python donde manipulen listas, tuplas y diccionarios para resolver un problema propuesto por el docente (por ejemplo, calcular estadísticas sobre una lista de estudiantes). **2. Ejercicio grupal:** Los estudiantes se agrupan para resolver un problema práctico y más complejo que implique el uso de las tres estructuras de datos (listas, tuplas y diccionarios). Ejemplo: desarrollar una pequeña aplicación para gestionar una lista de contactos (con nombre, teléfono y dirección) usando diccionarios para almacenar la información de cada contacto.	IDE (Visual Studio Code, PyCharm, Replit). Ejercicios en Google Drive o en papel. Documentación sobre funciones y métodos para listas, tuplas y diccionarios. 
Cierre (20 minutos)	**Estrategias:** • **Reflexión y retroalimentación:** El docente revisa y analiza las soluciones implementadas por los estudiantes, resaltando las buenas prácticas y sugiriendo mejoras. También se abordan los problemas más comunes que surgieron durante la sesión. • **Presentación y retroalimentación en equipo:** Cada grupo presenta su solución al problema planteado, explicando cómo utilizaron las estructuras de datos, qué dificultades encontraron y cómo las resolvieron. **Actividades:** 1. **Presentación de soluciones:** Los grupos presentan sus programas y explican cómo manipularon las listas, tuplas y diccionarios para resolver el problema, detallando las decisiones tomadas.	IDE para mostrar el código. Herramientas de retroalimentación (Google Drive, Trello). Pizarra o pantalla para discusión y resumen de las soluciones.

	2. **Discusión grupal:** Se realiza una discusión sobre las diferentes soluciones propuestas, fomentando el análisis crítico y la mejora continua.	



Sesión de aprendizaje 4



I. DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 2

II.TITULO

"Gestionar Excepciones en un Programa para Garantizar la Estabilidad del Código"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Gestionar excepciones en un programa utilizando mecanismos adecuados en Python (try, except, finally) para garantizar la estabilidad y robustez del código, evitando fallos inesperados durante la ejecución.	**Reconocer tipos de excepciones:** Identificar diferentes tipos de excepciones que pueden ocurrir durante la ejecución de un programa (por ejemplo, ValueError, IndexError, ZeroDivisionError). **Implementar bloques de manejo de excepciones:** Aplicar correctamente las estructuras `try`, `except`, y `finally` en un programa para controlar errores y mantener la estabilidad del código. **Manejar excepciones de manera eficiente:** Gestionar los	**Identificación adecuada de tipos de excepciones:** El estudiante debe ser capaz de reconocer y explicar los diferentes tipos de excepciones que pueden ocurrir en el contexto de un programa. **Uso adecuado de bloques try, except, y finally:** El estudiante debe utilizar correctamente estas estructuras para capturar y manejar las excepciones, mostrando comprensión de su funcionamiento. **Estabilidad del código:** El estudiante debe garantizar que el código siga ejecutándose correctamente incluso cuando se presenten excepciones, evitando que el programa se cierre de manera inesperada. **Claridad y efectividad del manejo de errores:** El estudiante debe mostrar una comunicación clara en los mensajes de error, proporcionando al usuario información útil sin interrumpir el flujo del programa.	El estudiante deberá desarrollar un pequeño programa que intencionalmente genere un error (por ejemplo, una división por cero o acceso fuera de índice) y que utilice excepciones para manejar dicho error de manera adecuada, mostrando un mensaje apropiado al usuario y asegurando que el programa continúe ejecutándose.	**Rúbrica de evaluación:** Identificación de excepciones y tipos adecuados (25%). Uso correcto de las estructuras try, except, y finally (40%). Capacidad para garantizar la estabilidad del código (20%). Claridad en la presentación de mensajes de error al usuario (15%).

	errores de manera que no interrumpan la ejecución del programa, brindando retroalimentación útil al usuario y evitando cierres inesperados. **Probar y validar el manejo de excepciones en un programa:** Probar las excepciones mediante la ejecución de casos que provoquen errores, verificando que el programa maneje los fallos correctamente.			
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	
Desarrollar habilidades de resolución de problemas mediante la implementación de medidas preventivas para evitar fallos y errores en los programas.	**Anticipar posibles errores:** El estudiante debe ser capaz de anticipar posibles puntos de fallo en su programa y aplicar excepciones para prevenir interrupciones en el flujo de ejecución. **Gestión de errores de manera responsable:** El estudiante debe implementar un manejo de errores que no solo solucione el problema inmediato, sino que también	**Prevención de errores:** El estudiante debe ser capaz de implementar medidas preventivas para evitar errores comunes en el programa mediante el uso adecuado de excepciones. **Responsabilidad en la gestión de errores:** El estudiante debe mostrar una actitud responsable al manejar los errores, proporcionando información útil y no interrumpiendo el flujo del programa. **Reflexión autónoma sobre la estabilidad del código:** El estudiante debe ser capaz de identificar posibles mejoras en su manejo de excepciones y justificar las decisiones tomadas en el proceso.	El estudiante debe mostrar un código que maneje excepciones de manera eficiente, incluso después de haber probado y detectado los posibles errores. Además, debe ser capaz de reflexionar sobre los casos de error más comunes y	

	proporcione retroalimentación útil para una mejor experiencia de usuario. **Trabajo autónomo y reflexivo:** El estudiante debe ser capaz de identificar, corregir y mejorar su código de manera autónoma cuando se presenten excepciones, reflexionando sobre su impacto y buscando mejores soluciones.		cómo evitarlos.	
Enfoque Transversal				
	Fomentar la reflexión crítica y la gestión responsable de los errores en la programación, promoviendo la autonomía y el aprendizaje continuo en la resolución de problemas complejos.			

IV.ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio (20 minutos)	**Estrategias:** • **Introducción al manejo de excepciones:** El docente presenta el concepto de excepciones en programación, explicando su importancia para manejar errores de manera controlada y garantizar que el programa siga ejecutándose sin fallos. Se explican los bloques try, except, y finally, y sus usos. • **Contextualización:** Se discuten ejemplos comunes de errores en los programas, como divisiones por cero, acceso fuera de índice en listas o diccionarios, y errores de tipo. El docente explica cómo las excepciones pueden evitar que el programa se cierre inesperadamente. **Actividades:**	Proyector o pantalla para mostrar ejemplos de código. IDE (Visual Studio Code, PyCharm, Replit). Documentación sobre manejo de excepciones en Python.

	1. **Discusión guiada:** El docente realiza una breve discusión sobre la importancia de manejar excepciones, preguntando a los estudiantes si han encontrado errores en sus programas y cómo los han resuelto. 2. **Demostración:** El docente presenta un ejemplo de código simple que genera un error (por ejemplo, división por cero) y muestra cómo se puede manejar ese error utilizando una excepción adecuada.	
Desarrollo (50 minutos)	**Estrategias:** • **Trabajo práctico guiado:** Los estudiantes comienzan a trabajar en ejercicios prácticos donde deben generar un error intencional (por ejemplo, un error de índice o de tipo) y manejarlo utilizando bloques try, except, y finally. • **Prueba de casos:** Los estudiantes deben probar sus programas introduciendo diferentes tipos de entradas para verificar que el manejo de excepciones funciona correctamente. **Actividades:** 1. **Ejercicio individual:** Los estudiantes deben escribir un pequeño programa que calcule la división de dos números ingresados por el usuario. El programa debe manejar excepciones como la división por cero o la entrada de datos no válidos (como texto en lugar de números). **2. Prueba del código:** Los estudiantes ejecutan el código y deben probarlo con diferentes valores de entrada para verificar que las excepciones se gestionan correctamente, mostrando mensajes adecuados al usuario sin interrumpir el flujo del programa.	IDE para escribir y ejecutar código. Ejercicios en Google Drive o documentos. Documentación sobre las excepciones en Python.
Cierre (20 minutos)	**Estrategias:** • **Reflexión y retroalimentación:** El docente revisa las soluciones de los estudiantes, discutiendo las buenas prácticas en el manejo de excepciones. Se brinda retroalimentación sobre cómo mejorar la claridad de los mensajes de error y cómo garantizar la estabilidad del código. • **Resumen de conceptos clave:** Se recapitula lo aprendido sobre excepciones, destacando la importancia de usar bloques try, except, y finally para evitar errores fatales en el programa. **Actividades:** 1. **Reflexión grupal:** Los estudiantes comparten sus soluciones y discuten las dificultades encontradas al manejar excepciones. 2. **Preguntas y respuestas:** El docente responde preguntas sobre el manejo de excepciones y cómo implementar estas técnicas en programas más complejos.	Pizarra o pantalla para repasar los conceptos clave. Material adicional o enlaces a recursos de aprendizaje sobre excepciones.

Sesión de aprendizaje 5



I.DATOS GENERALES:

a.	Institución Educativa	:	
b.	UGEL	:	
c.	Área	:	Desarrollo de Sistemas de Información
d.	Ciclo	:	I
e.	Grado	:	
f.	Turno	:	Mañana
g.	Docente	:	Ing. Hans Edgardo Delgado Díaz
h.	Duración	:	90 minutos
i.	Fecha	:	Semana 3

II.TITULO

"Implementar Conceptos Básicos de la Programación Orientada a Objetos (OOP)"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Implementar los conceptos fundamentales de la Programación Orientada a Objetos (OOP), como clases, objetos, atributos y métodos, utilizando Python para crear programas modulares, reutilizables y escalables.	**Definir clases y objetos en Python:** Los estudiantes serán capaces de identificar la sintaxis y la estructura de una clase en Python, y cómo crear objetos a partir de ella. **Establecer atributos y métodos en una clase:** Los estudiantes podrán definir atributos (variables) y métodos (funciones) dentro de una clase, comprendiendo la diferencia entre estos y su rol en la creación de un objeto. **Aplicar los principios de encapsulamiento:**	**Definición correcta de clases y objetos:** El estudiante debe ser capaz de definir correctamente una clase y crear un objeto de esa clase, siguiendo la sintaxis adecuada de Python. **Uso adecuado de atributos y métodos:** El estudiante debe ser capaz de definir y utilizar atributos y métodos dentro de una clase, demostrando cómo estos interactúan con los objetos. **Implementación de encapsulamiento:** El estudiante debe implementar correctamente el concepto de encapsulamiento mediante la creación de métodos para acceder y modificar los atributos de un objeto. **Desarrollo de programas orientados a objetos:** El estudiante debe desarrollar un programa modular que utilice las clases y objetos de manera eficiente para resolver un problema práctico.	Los estudiantes desarrollarán un programa orientado a objetos que modela un problema práctico (por ejemplo, una aplicación de gestión de estudiantes o productos). Deberán crear clases, objetos, atributos y métodos según sea necesario y aplicar principios de encapsulamiento.	**Rúbrica de evaluación:** Definición y creación de clases y objetos (30%). Implementación de atributos y métodos (30%). Aplicación correcta de encapsulamiento (20%). Calidad y organización del código (20%).

	Los estudiantes comprenderán y aplicarán el concepto de encapsulamiento para gestionar el acceso a los atributos de un objeto mediante métodos (getters y setters).			
	Crear programas orientados a objetos: Los estudiantes deberán aplicar los conceptos aprendidos para crear programas que utilicen clases y objetos para organizar el código y resolver problemas prácticos.			
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	
Desarrollar habilidades de resolución de problemas mediante la aplicación de la programación orientada a objetos para crear soluciones modulares, escalables y reutilizables.	**Desarrollar soluciones modulares:** El estudiante debe ser capaz de estructurar su código en clases y objetos que permitan una fácil extensión y reutilización. **Resolver problemas complejos de manera organizada:** El estudiante debe aplicar principios de OOP para descomponer un problema en partes más simples, cada una representada por	**Modularización y escalabilidad del código:** El estudiante debe demostrar cómo su solución es modular y escalable, usando clases y objetos correctamente para facilitar la extensión del código. **Resolución efectiva del problema:** El estudiante debe aplicar los conceptos de OOP de manera eficiente para resolver el problema planteado. **Claridad y organización del código:** El estudiante debe escribir código limpio y bien estructurado, con comentarios explicativos que faciliten su comprensión.	El estudiante debe entregar un programa orientado a objetos que resuelva un problema práctico, aplicando la modularización y escalabilidad de acuerdo con los principios de OOP.	

	una clase u objeto. **Gestionar el código de manera escalable:** El estudiante debe ser capaz de crear soluciones que puedan ser fácilmente modificadas o ampliadas en el futuro mediante la reutilización de clases y objetos.			
Enfoque Transversal				
Libertad Responsabilidad				

IV.ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio (20 minutos)	**Estrategias:** • **Preguntas iniciales:** ¿Qué entienden por programación orientada a objetos? ¿Pueden dar ejemplos de objetos y clases en la vida real? • **Breve exposición teórica:** Introducción a los conceptos fundamentales de OOP: clases, objetos, atributos, métodos, constructor, encapsulamiento y herencia. • **Conexión con la práctica:** Relación de OOP con situaciones reales (por ejemplo, cómo modelar un coche, una persona, etc. con clases y objetos). **Actividades:** • **Discusión grupal:** Los estudiantes comparten ejemplos de objetos del mundo real y cómo podrían modelarlos con clases en programación. • **Video explicativo:** Visualización de un video corto que explica los conceptos básicos de OOP con ejemplos prácticos sencillos.	Pizarra y marcadores. Proyector para mostrar el video. Material de lectura básico sobre OOP.
	Estrategias:	Computadoras con acceso a un

Desarrollo (50 minutos)	• **Demostración práctica:** El docente realiza una demostración de cómo crear una clase simple (por ejemplo, una clase Coche con atributos como marca y modelo, y métodos como arrancar()). • **Ejercicio práctico guiado:** Los estudiantes implementan una clase en un lenguaje de programación (por ejemplo, Python o Java) con sus propios atributos y métodos. • **Trabajo en equipo:** Los estudiantes trabajan en pequeños grupos para diseñar y programar una clase que resuelva un problema específico (por ejemplo, una clase Estudiante con métodos que calculen la edad o el promedio de calificaciones). **Actividades:** • **Implementación de clase simple:** Los estudiantes siguen una serie de pasos para crear su primera clase con atributos y métodos. • **Desafío grupal:** Cada grupo debe presentar su clase creada y explicar su implementación al resto de la clase.	entorno de programación (por ejemplo, Python o Java). Lenguaje de programación y editor de texto. Material de referencia y guías de programación.
Cierre (20 minutos)	**Estrategias:** • **Reflexión y preguntas:** El docente invita a los estudiantes a compartir lo que aprendieron durante la sesión, respondiendo preguntas como: ¿Qué desafíos encontraron al crear su clase? ¿Cómo se relacionan los conceptos de OOP con otros tipos de programación? • **Feedback final:** El docente proporciona retroalimentación sobre los trabajos realizados, destacando aspectos positivos y áreas de mejora. • **Conclusión:** Resumen de los conceptos clave de la programación orientada a objetos y su importancia en la programación moderna. **Actividades:** • **Presentación de soluciones:** Los grupos presentan sus clases y explican cómo aplicaron los conceptos aprendidos. • **Ronda de preguntas:** Los estudiantes hacen preguntas al docente y entre ellos para resolver dudas finales.	Pizarra y marcadores. Computadoras para mostrar las implementaciones. Material de apoyo sobre los conceptos de OOP.



Sesión de aprendizaje 6



I. DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 3

II.TITULO

"Desarrollo de algoritmos sencillos de búsqueda y ordenación"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Desarrollar algoritmos sencillos de búsqueda y ordenación.	Identificar y comprender los conceptos de los algoritmos de búsqueda y ordenación Implementar algoritmos de búsqueda como Búsqueda Lineal y Binaria Desarrollar algoritmos de ordenación como el Ordenamiento por Burbuja y Selección Aplicar los algoritmos de búsqueda y ordenación en situaciones prácticas.	• El estudiante describe correctamente el funcionamiento de los algoritmos de búsqueda y ordenación. • El estudiante implementa correctamente los algoritmos de búsqueda y ordenación en un entorno de programación. • El estudiante aplica correctamente los algoritmos en un problema práctico de programación.	Código fuente de los algoritmos de búsqueda y ordenación implementados	Lista de cotejo o rúbrica de evaluación para verificar el funcionamiento y la correcta implementación de los algoritmos
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	
Resolución de problemas mediante el pensamiento lógico y estructurado	• Identificar patrones y relaciones lógicas en problemas. • Formular y aplicar	• El estudiante demuestra capacidad para analizar problemas y descomponerlos en tareas lógicas. • El estudiante resuelve problemas complejos utilizando el enfoque adecuado de programación.	Ejercicios prácticos resueltos correctamente	

	soluciones utilizando herramientas y técnicas de programación. • Tomar decisiones basadas en el análisis lógico de la situación.		que incluyan la aplicación de algoritmos y el análisis de resultados	
Enfoque Transversal				
	Uso de la tecnología de manera ética y responsable, aplicando soluciones informáticas para optimizar procesos.			

IV.ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio (15 minutos)	**Estrategias:** • **Exploración inicial:** Breve explicación del propósito de la clase (importancia de los algoritmos de búsqueda y ordenación en la programación). • **Motivación:** Presentar ejemplos prácticos de la vida diaria donde se aplican estos algoritmos (por ejemplo, búsqueda en una lista de contactos o la organización de elementos en un inventario). **Actividades:** 1. Actividad de calentamiento: Plantear un problema de búsqueda y ordenación (por ejemplo, encontrar un número específico en una lista de datos desordenados y ordenarla). 2. Breve repaso teórico de los conceptos de algoritmos de búsqueda (Búsqueda Lineal y Binaria) y ordenación (Burbuja y Selección).	Pizarra y marcadores. Proyector para mostrar los ejemplos.
	Estrategias: • **Explicación teórica:** Exposición de los conceptos y principios de cada algoritmo. • **Demostración práctica:** Mostrar paso a paso cómo implementar cada algoritmo en un entorno de programación. **Actividades:** 1. **Búsqueda Lineal:**	Computadoras con entorno de desarrollo integrado (IDE). Ejemplos de código y plantillas para que los estudiantes practiquen.

Desarrollo (60 minutos)	○ Explicar cómo funciona y su código. ○ Implementación en clase con un conjunto de datos pequeño. 2. **Búsqueda Binaria:** ○ Explicar cómo funciona y su código. ○ Implementación en clase con un conjunto de datos ordenados. 3. **Ordenación por Burbuja:** ○ Explicar cómo funciona y su código. ○ Implementación en clase con un conjunto de datos desordenados. 4. **Ordenación por Selección:** ○ Explicar cómo funciona y su código. ○ Implementación en clase con un conjunto de datos desordenados. **Trabajo en grupos pequeños:** • Los estudiantes implementan los algoritmos de búsqueda y ordenación en sus computadoras, con apoyo del docente.	
Cierre (15 minutos)	**Estrategias:** • **Reflexión y consolidación:** Revisión de los algoritmos implementados por los estudiantes. • **Evaluación rápida:** Breve prueba de autoevaluación para verificar la comprensión de los algoritmos y su implementación. **Actividades:** 1. Revisión grupal de los resultados obtenidos en la implementación de los algoritmos. 2. Realización de una pequeña reflexión sobre la importancia de los algoritmos de búsqueda y ordenación en la programación. 3. Retroalimentación final por parte del docente, resaltando aciertos y posibles mejoras.	Pizarra y marcador para resumen. Proyector para mostrar las soluciones y correcciones.

Sesión de aprendizaje 7



I.DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 4

II.TITULO

"Implementar soluciones recursivas para problemas específicos"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Implementar soluciones recursivas para problemas específicos, desarrollando algoritmos eficientes que resuelvan problemas mediante el uso de la recursión	Comprender el concepto de recursividad y su aplicación en la resolución de problemas Aplicar soluciones recursivas en problemas específicos, identificando cuándo utilizar este enfoque Implementar funciones recursivas en un lenguaje de programación Depurar y optimizar soluciones recursivas para asegurar su eficiencia	• Identificación clara del problema y la elección adecuada de la técnica recursiva para su resolución. • Implementación correcta de la solución recursiva en un lenguaje de programación. • Eficiencia y optimización de la solución recursiva. • Capacidad para detectar errores en la implementación y aplicar técnicas de depuración.	Código fuente de la solución recursiva implementada. Informe de la explicación del proceso de implementación y optimización de la solución. Resultados de la prueba del código con distintos casos de prueba.	**Ficha de evaluación**: que evalúa la correcta implementación de la solución recursiva, la explicación teórica de la recursión y la optimización de la solución. **Rubrica de evaluación**: para valorar la capacidad de implementar y depurar el código recursivo
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	
Trabajo colaborativo en proyectos de programación	• Colaborar efectivamente en equipos para el	• Nivel de participación y colaboración durante el desarrollo del proyecto. • Calidad de las aportaciones y soluciones discutidas en grupo.	• Participación en la discusión de las soluciones.	

	desarrollo de soluciones. • Comunicar de manera clara y efectiva las ideas y soluciones dentro del equipo de trabajo. • Evaluar y retroalimentar el trabajo del compañero en función de la solución propuesta.	• Calidad de las retroalimentaciones dadas a los compañeros.  	• Comentarios y observaciones constructivas sobre el trabajo de otros compañeros	
Enfoque Transversal				
	Desarrollo del pensamiento crítico y la resolución de problemas mediante el trabajo en equipo.			

IV.ORGANIZACIÓN

Momentos	**Estrategias/ actividades**	**Recursos**
Inicio (20 minutos)	**Estrategia**: Activación de conocimientos previos mediante una breve discusión sobre problemas comunes en programación que se pueden resolver con recursión. **Actividad**: 1. Breve exposición teórica sobre recursividad (definición, características y casos donde se aplica). 2. Ejemplo de un problema simple como la serie de Fibonacci y su implementación recursiva. 3. Preguntas interactivas para conocer las ideas previas de los estudiantes sobre la recursividad.	• Pizarra digital o proyector. • Ejemplos de código (lenguaje Python o Java).
Desarrollo (50 minutos)	**Estrategia**: Aplicación práctica de la recursividad mediante la resolución de problemas guiados. **Actividad**: 1. Los estudiantes se dividen en parejas o pequeños grupos para resolver un conjunto de problemas recursivos (como factorial, búsqueda binaria, torre de Hanoi, entre otros). 2. Durante esta actividad, se les pide implementar una solución recursiva en su entorno de desarrollo.	• Entorno de desarrollo (IDE). • Ejercicios de programación impresos o en formato digital. • Pizarra para apuntar puntos clave

	3. El docente circula entre los grupos para ofrecer orientación, apoyo en la implementación y resolver dudas sobre la aplicación de la recursión. **4.** Los estudiantes deben explicar cómo identificaron el problema adecuado para aplicar recursión y discutir sobre la eficiencia de la solución.	durante la discusión.
Cierre (20 minutos)	**Estrategia**: Reflexión y consolidación de conocimientos. **Actividad**: 1. Los estudiantes presentan brevemente su código y explican cómo resolvieron el problema utilizando recursión. 2. El docente realiza una retroalimentación de los resultados obtenidos, destacando buenas prácticas y errores comunes. **3.** Reflexión grupal sobre cuándo es adecuado utilizar recursión y qué tan eficientes son estas soluciones en comparación con soluciones iterativas.	• Proyector para mostrar ejemplos. • Pizarra para resumir los puntos importantes discutidos durante la sesión.





Sesión de aprendizaje 8

I.DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 4

II.TITULO

"Desarrollar programas que interactúan con archivos (lectura y escritura)"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Desarrollar programas que interactúan con archivos, permitiendo realizar operaciones de lectura y escritura para manipular datos de manera eficiente.	Comprender la estructura de los archivos y su manipulación en programación Aplicar operaciones de lectura y escritura en archivos utilizando un lenguaje de programación Implementar funciones que gestionen archivos de texto para almacenar y recuperar información. Utilizar correctamente los métodos de apertura, cierre, lectura y escritura de archivos. Gestionar errores comunes asociados con el manejo de archivos	• Correcta implementación de funciones de apertura, lectura, escritura y cierre de archivos. • Correcta utilización de formatos de archivo (texto o binarios) para las operaciones requeridas. • Calidad del código, incluyendo la estructura y manejo adecuado de excepciones y errores. • Cumplimiento con las especificaciones del problema (por ejemplo, leer y escribir datos específicos).	• Código fuente que implementa las operaciones de lectura y escritura en archivos. • Resultados obtenidos al leer y escribir archivos, demostrando la correcta manipulación de datos. • Informe explicativo sobre el proceso de desarrollo del programa y cómo se gestionaron los archivos.	**Ficha de evaluación**: Que evalúa la correcta implementación de las funciones de archivo, manejo de excepciones, y eficiencia del código. **Rubrica de evaluación**: Para valorar la calidad del código, el manejo adecuado de archivos, y la resolución de posibles errores en la ejecución.
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	

Trabajo en equipo y colaboración en el desarrollo de soluciones tecnológicas.	• Colaborar en el desarrollo conjunto de soluciones a problemas técnicos. • Comunicar y documentar las soluciones de manera clara y efectiva dentro del equipo. • Evaluar y retroalimentar el trabajo de otros miembros del equipo	Participación activa en la resolución del ejercicio en equipo. Calidad de las contribuciones al trabajo grupal. Eficiencia y claridad en la comunicación y documentación de la solución. ANTHONY HANS DELGADO CHAVARRI INGENIERO DE SISTEMAS Y COMPUTACION REG. CIP. 186247	• Participación en la discusión y desarrollo de la solución grupal. • Comentarios y retroalimentación efectiva sobre las soluciones de los demás miembros del equipo.	
Enfoque Transversal				
	Desarrollo de habilidades comunicativas y colaborativas en la resolución de problemas mediante programación			

IV.ORGANIZACIÓN

Momentos	**Estrategias/ actividades**	**Recursos**
Inicio (20 minutos)	**Estrategia**: Activación de conocimientos previos y motivación mediante la exposición de la importancia de interactuar con archivos en la programación moderna. **Actividad**: 1. Introducción teórica sobre el concepto de archivos y cómo se utilizan en los programas (archivos de texto, formatos binarios, etc.). 2. Explicación de las funciones básicas de manejo de archivos: apertura (open), lectura (read), escritura (write), y cierre (close). 3. Ejemplo práctico: lectura y escritura de un archivo de texto simple (como un archivo que contenga una lista de nombres o números). 4. Realización de preguntas interactivas para asegurar que los estudiantes comprendan los conceptos teóricos.	• Pizarra digital o proyector para explicar los conceptos. • Código de ejemplo para lectura y escritura en archivos (en lenguaje Python o Java).
Desarrollo (50 minutos)	**Estrategia**: Aplicación práctica de los conceptos mediante la solución de problemas relacionados con archivos. **Actividad**: 1. Los estudiantes se dividen en grupos pequeños (o trabajan individualmente) para resolver ejercicios prácticos que impliquen la lectura y escritura en archivos.	Computadoras con entornos de desarrollo (IDE) para la implementación de los ejercicios. Archivos de ejemplo para

	2. Ejercicios: o Crear un programa que lea un archivo de texto, cuente la cantidad de palabras o líneas y muestre el resultado. o Crear un programa que escriba datos en un archivo de texto a partir de la entrada del usuario (por ejemplo, ingresar una lista de nombres o valores numéricos). 3. Durante el desarrollo, el docente circula por el aula, orientando a los estudiantes en la implementación de las funciones de manejo de archivos, resolviendo dudas y ofreciendo sugerencias para la optimización del código. 4. Revisión de posibles errores comunes (por ejemplo, manejo incorrecto de excepciones, no cierre de archivos, etc.). **5.** Discusión sobre la importancia de manejar errores correctamente al trabajar con archivos (errores de lectura, escritura, archivo no encontrado, etc.).	trabajar durante la sesión. Manuales o guías rápidas sobre funciones de archivos en el lenguaje utilizado.
Cierre (20 minutos)	**Estrategia**: Reflexión grupal y retroalimentación sobre el aprendizaje de la sesión, consolidación de conceptos. **Actividad**: 1. Los estudiantes presentan sus soluciones a los ejercicios realizados, explicando cómo implementaron las funciones de manejo de archivos. 2. El docente realiza una retroalimentación sobre las soluciones presentadas, destacando las buenas prácticas y corrigiendo posibles errores. 3. Reflexión final sobre la utilidad de interactuar con archivos en programas reales (por ejemplo, la gestión de bases de datos, registros de usuarios, etc.). **4.** Resumen de las lecciones aprendidas sobre la lectura y escritura de archivos, así como la importancia del manejo adecuado de excepciones.	• Pizarra para hacer anotaciones finales. • Proyector para presentar ejemplos y correcciones.

ANTHONY HANS DELGADO CHAVARRI
INGENIERO DE SISTEMAS Y COMPUTACION
REG. CIP. 186247



Sesión de aprendizaje 9

I.DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 5

II.TITULO

"Desarrollar proyectos colaborativos utilizando herramientas de control de versiones"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Desarrollar proyectos colaborativos utilizando herramientas de control de versiones para gestionar el código fuente de manera eficiente y efectiva.	Comprender el concepto de control de versiones y su importancia en el desarrollo de software Aplicar comandos básicos de herramientas de control de versiones (como Git) para gestionar proyectos colaborativos Realizar operaciones como crear un repositorio, clonar, hacer commits, branches, y merges en un entorno colaborativo Resolver conflictos comunes en proyectos colaborativos usando	• Correcta implementación de comandos de control de versiones (creación de repositorios, commits, branches, etc.). • Capacidad para resolver conflictos de fusión en el código. • Participación activa en las actividades colaborativas del proyecto. • Organización y claridad en el uso de la plataforma de control de versiones (GitHub o GitLab). • Aplicación de buenas prácticas en la gestión de versiones y documentación del código.	• Repositorio en GitHub o GitLab con el código fuente gestionado adecuadamente. • Historial de commits y branches que muestran la colaboración y evolución del proyecto. • Reporte breve sobre las actividades realizadas durante la sesión, los comandos utilizados y cómo se resolvieron los posibles conflictos.	**Ficha de evaluación**: Evaluará el uso de comandos básicos de Git, la participación en el proyecto colaborativo, y la resolución de conflictos. **Rubrica de evaluación**: Valora la calidad del código, el uso de buenas prácticas de control de versiones y la efectividad en la colaboración del equipo.

	herramientas de control de versiones			
	Colaborar de manera efectiva en equipos utilizando plataformas como GitHub o GitLab			
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	
Trabajo colaborativo en equipos multidisciplinarios	• Colaborar eficazmente con otros en la resolución de problemas técnicos y en el desarrollo de soluciones. • Comunicar ideas y avances de forma clara y efectiva dentro del equipo. • Organizar tareas y coordinar esfuerzos para lograr el objetivo común.	• Nivel de participación en las tareas grupales. • Calidad de la comunicación y colaboración durante el desarrollo del proyecto. • Contribución significativa al éxito del proyecto en términos de código y coordinación.	• Interacción y contribución en el repositorio del proyecto. • Comunicación efectiva dentro del equipo, demostrada a través de commit messages claros y bien documentados.	
Enfoque Transversal				
	Desarrollo de habilidades de comunicación y trabajo en equipo en proyectos tecnológicos mediante el uso de herramientas colaborativas			

ANTHONY HANS DELGADO CHAVARRI
INGENIERO DE SISTEMAS Y COMPUTACION
REG. CIP. 186247

IV.ORGANIZACIÓN

Momentos	**Estrategias/ actividades**	**Recursos**
Inicio (20 minutos)	**Estrategia**: Introducción a los conceptos básicos de control de versiones y su relevancia en proyectos colaborativos. **Actividad**: 1. Breve presentación sobre la importancia del control de versiones en el desarrollo de software (prevención de pérdida de código, colaboración eficiente, manejo de versiones).	• Pizarra digital o proyector para la explicación de los conceptos. • Presentación con los conceptos y comandos clave de Git.

	2. Explicación de las herramientas de control de versiones más comunes (Git, GitHub, GitLab) y sus componentes principales: repositorios, commits, branches y merges. 3. Demostración práctica en vivo sobre cómo crear un repositorio, clonar un proyecto y hacer el primer commit. 4. Explicación de las buenas prácticas de trabajo con Git, como realizar commits frecuentes y escribir mensajes de commit claros	• Ejemplo de código y repositorio en GitHub o GitLab
Desarrollo (50 minutos)	**Estrategia**: Aplicación práctica del control de versiones en un proyecto colaborativo. **Actividad**: 1. Los estudiantes se agrupan en equipos pequeños para desarrollar un proyecto sencillo en colaboración, utilizando Git y una plataforma de control de versiones como GitHub o GitLab. 2. Se asignan tareas a cada miembro del equipo, como realizar commits, crear y gestionar ramas, hacer pull requests, y resolver conflictos. 3. Cada equipo debe trabajar de manera colaborativa, haciendo uso de las funcionalidades de Git para gestionar su código y sincronizar los cambios. 4. El docente circula por el aula y apoya a los equipos en la utilización de los comandos de Git, resuelve dudas y supervisa el progreso del proyecto. **5.** Durante la sesión, los estudiantes deben enfrentarse a escenarios como la creación de ramas y la resolución de conflictos al hacer merge, con la orientación del docente.	• Computadoras con conexión a internet para acceder a plataformas como GitHub o GitLab. • Repositorios compartidos donde los estudiantes puedan clonar y trabajar colaborativamente. • Guías rápidas de comandos de Git para consulta
Cierre (20 minutos)	**Estrategia**: Reflexión sobre el aprendizaje y revisión de la colaboración en el proyecto. **Actividad**: 1. Los estudiantes presentan brevemente su trabajo en equipo, mostrando cómo utilizaron Git para gestionar el código y cómo resolvieron posibles conflictos. 2. El docente da retroalimentación sobre las buenas prácticas aplicadas y discute los desafíos encontrados en el uso del control de versiones. 3. Reflexión final sobre la importancia de las herramientas de control de versiones en proyectos colaborativos y la organización del trabajo en equipo.	• Pizarra o proyector para hacer anotaciones de las conclusiones. • Acceso al repositorio del proyecto para revisarlo en conjunto.

ANTHONY HAYS DELGADO CHAVARRI
INGENIERO DE SISTEMAS Y COMPUTACION
REG. CIP. 186247

Sesión de aprendizaje 10



I.DATOS GENERALES:

a.	Institución Educativa	:	
b.	UGEL	:	
c.	Área	:	Desarrollo de Sistemas de Información
d.	Ciclo	:	I
e.	Grado	:	
f.	Turno	:	Mañana
g.	Docente	:	Ing. Hans Edgardo Delgado Díaz
h.	Duración	:	90 minutos
i.	Fecha	:	Semana 5

II.TITULO

"Optimizar y mejorar la eficiencia del código"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Optimizar y mejorar la eficiencia del código para resolver problemas de manera más rápida y con un uso eficiente de los recursos.	Identificar cuellos de botella en el rendimiento del código. Aplicar técnicas de optimización de algoritmos para mejorar la eficiencia temporal y espacial. Utilizar herramientas de análisis de rendimiento para medir la eficiencia del código antes y después de la optimización. Mejorar la legibilidad y el mantenimiento del código sin sacrificar su rendimiento Evaluar la eficiencia de un código optimizado comparado con una versión no optimizada	• Identificación clara de los problemas de rendimiento en el código original. • Aplicación efectiva de técnicas de optimización para mejorar el rendimiento. • Medición del impacto de la optimización utilizando herramientas adecuadas. • Mantenimiento de la funcionalidad del código después de la optimización. • Claridad, legibilidad y eficiencia del código resultante después de la optimización	• Código optimizado junto con el código original. • Informe de optimización que explique las decisiones tomadas y cómo se mejoró la eficiencia. • Resultados de las pruebas de rendimiento que comparen el código antes y después de la optimización.	**Ficha de evaluación**: Evaluación sobre la aplicación de técnicas de optimización, el análisis de rendimiento y la calidad del código resultante. **Rubrica de evaluación**: Valora la eficiencia del código, la claridad en las decisiones de optimización y la documentación del proceso.

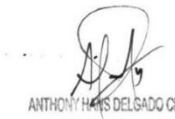
Competencia transversal	Capacidades	Criterios	Evidencia	
Solución eficiente de problemas mediante el trabajo colaborativo	• Colaborar eficazmente en equipo para identificar y resolver problemas de programación. • Comunicar claramente las soluciones y técnicas de optimización utilizadas. • Retroalimentar y evaluar el trabajo de los compañeros para mejorar las soluciones del equipo.	• Participación activa en el análisis y la optimización del código en equipo. • Claridad y efectividad en la comunicación de las soluciones de optimización. • Evaluación crítica y constructiva del trabajo de los demás.	• Participación en la resolución de problemas y optimización del código en equipo. • Comentarios y retroalimentación sobre el trabajo de otros miembros del equipo	
Enfoque Transversal				
	Desarrollo de habilidades de trabajo en equipo y comunicación efectiva para mejorar el rendimiento del código.			

ANTHONY HANS DELGADO CHAVARRI
INGENIERO DE SISTEMAS Y COMPUTACION
REG. CIP. 186247

IV.ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio (20 minutos)	**Estrategia**: Introducción teórica sobre la importancia de la optimización y la mejora de la eficiencia en el código. **Actividad**: 1. Explicación de qué es la optimización de código y por qué es importante para mejorar el rendimiento de las aplicaciones. 2. Conceptos clave sobre el análisis de la complejidad temporal (notación Big-O) y complejidad espacial. 3. Presentación de técnicas comunes para optimizar el código: uso de estructuras de datos eficientes, eliminación de redundancias, optimización de algoritmos (por ejemplo, de búsqueda y ordenamiento). 4. Demostración práctica de un código no optimizado (por ejemplo, un algoritmo de búsqueda lineal) y su versión optimizada (por ejemplo, búsqueda binaria).	Pizarra digital o proyector para la explicación de conceptos teóricos. Ejemplos de código no optimizado y optimizado (en un lenguaje como Python o Java). Notación Big-O para ilustrar la complejidad temporal y espacial.
Desarrollo (50 minutos)	**Estrategia**: Aplicación práctica de técnicas de optimización en un código existente.	• Computadoras con entornos de desarrollo (IDE)

	Actividad: 1. Los estudiantes trabajan en parejas o grupos pequeños para optimizar un programa existente que tenga ineficiencias conocidas (por ejemplo, un algoritmo de ordenamiento o búsqueda ineficiente). 2. Durante la actividad, los estudiantes: o Identifican los puntos de ineficiencia en el código. o Proponen y aplican soluciones de optimización (cambio de estructuras de datos, mejora de algoritmos, uso de técnicas de paralelización, entre otras). o Realizan pruebas de rendimiento para medir el impacto de las optimizaciones utilizando herramientas de análisis de rendimiento como **time** (en Python) o el profiler de su IDE. o Comparan el código antes y después de la optimización en términos de tiempo de ejecución y uso de memoria. **3.** El docente circula entre los grupos para brindar apoyo técnico y asegurarse de que los estudiantes apliquen correctamente las técnicas de optimización.	para implementar el código. • Herramientas de análisis de rendimiento (como **time**, **cProfile** en Python, o **VisualVM** en Java). • Ejercicios de optimización listos para ser trabajados por los estudiantes.
Cierre (20 minutos)	**Estrategia**: Reflexión y presentación de los resultados, análisis de la efectividad de las optimizaciones. **Actividad**: 1. Los estudiantes presentan brevemente las optimizaciones realizadas, explicando las decisiones que tomaron para mejorar la eficiencia del código. 2. Comparación de los tiempos de ejecución y uso de memoria antes y después de la optimización. 3. Discusión sobre las dificultades encontradas durante el proceso de optimización y cómo las resolvieron. 4. Reflexión final sobre cómo la optimización de código es crucial en proyectos grandes y la importancia de balancear la legibilidad y el rendimiento.	• Pizarra para resumir las principales lecciones de la sesión. • Proyector para mostrar ejemplos y resultados de las pruebas de rendimiento.



ANTHONY HANS DELGADO CHAVARRI
INGENIERO DE SISTEMAS Y COMPUTACION
REG. CIP. 186247

Sesión de aprendizaje 11



I.DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 6

II.TITULO

"Presentar un proyecto de programación a un público objetivo"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Presentar un proyecto de programación de manera efectiva a un público objetivo, explicando claramente su propósito, funcionamiento y resultados obtenidos.	Organizar y estructurar una presentación clara y concisa sobre un proyecto de programación Comunicar de manera efectiva el objetivo, la solución técnica y los resultados de un proyecto de programación. Utilizar herramientas visuales (como diapositivas, demos en vivo o vídeos) para apoyar la presentación del proyecto Adaptar el lenguaje y el contenido según el nivel de conocimiento del público objetivo	• Claridad en la estructura y organización de la presentación (introducción, desarrollo, conclusión). • Uso adecuado de herramientas visuales para apoyar la explicación. • Explicación precisa y comprensible de los aspectos técnicos del proyecto, adaptados al público. • Capacidad para responder preguntas del público de forma coherente y detallada. • Habilidad para gestionar el tiempo de presentación de manera eficiente.	• Presentación (diapositivas, demostración en vivo) del proyecto de programación. • Respuestas claras y bien fundamentadas a las preguntas del público. • Evaluación del desempeño en la presentación, basada en la claridad y	**Ficha de evaluación**: Evaluación de la presentación en términos de estructura, claridad técnica, uso de herramientas visuales y capacidad de respuesta. **Rubrica de evaluación**: Para valorar la efectividad de la comunicación, la organización del

	Responder preguntas del público de manera efectiva y con confianza		efectividad de la exposición.	contenido, y la interacción con el público.
Competencia transversal	**Capacidades**	**Criterios**	**Evidencia**	
Comunicación efectiva en presentaciones profesionales.	• Organizar la información de manera clara y lógica para su presentación. • Utilizar estrategias de comunicación verbal y no verbal efectivas. • Adaptar el discurso al público objetivo. • Fomentar un ambiente interactivo mediante la gestión adecuada de preguntas y respuestas.	• Organización de la presentación y claridad en la transmisión del mensaje. • Uso adecuado de la voz, gestos y lenguaje corporal durante la presentación. • Interacción efectiva con el público, respondiendo preguntas de manera adecuada. ANTHONY HANS DELGADO CHAVARRI INGENIERO DE SISTEMAS Y COMPUTACION REG. CIP. 186247	• Grabación de la presentación para evaluar el uso del lenguaje y las habilidades de comunicación. • Retroalimentación del público sobre la claridad de la explicación y la interacción	
Enfoque Transversal				
	Desarrollo de habilidades de comunicación, presentación y liderazgo para transmitir ideas y resultados técnicos de manera clara y efectiva			

IV.ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio (20 minutos)	**Estrategia**: Introducción a la importancia de las habilidades de presentación y cómo presentar un proyecto técnico de programación a diferentes audiencias. **Actividad**: 1. **Introducción teórica** sobre la importancia de las presentaciones de proyectos técnicos, especialmente en el contexto de la programación. 2. **Ejemplos de buenas y malas presentaciones**: análisis de videos o ejemplos previos de presentaciones de proyectos de	• Proyector para mostrar ejemplos de presentaciones (videos o grabaciones). • Guía de presentación con los componentes clave de una buena exposición

	programación. Se discutirán aspectos como la estructura, el lenguaje utilizado, la claridad de la explicación, y la interacción con el público. 3. **Explicación de los componentes clave de una presentación**: introducción, desarrollo (explicación técnica), demostración del proyecto, y conclusión. 4. **Consejos para manejar preguntas y respuestas**: cómo anticiparse a las preguntas, cómo responder de manera efectiva y mantener la calma durante la presentación	
Desarrollo (50 minutos)	**Estrategia**: Preparación y práctica de las presentaciones por parte de los estudiantes. **Actividad**: 1. Los estudiantes preparan una presentación de 5-7 minutos sobre el proyecto de programación que han desarrollado durante el curso. o Deben estructurar la presentación en una introducción clara, explicar los aspectos técnicos del proyecto, mostrar la solución en acción (demo o código funcionando), y concluir con los resultados o impacto del proyecto. o Deben adaptar su discurso al público objetivo (en este caso, el resto de la clase o el docente, quienes simularán ser el público). 2. **Prácticas de presentación**: o Los estudiantes presentan su proyecto ante el grupo (en grupos pequeños o en pareja). o Se les anima a utilizar recursos visuales, como diapositivas, capturas de pantalla o demostraciones en vivo, para hacer la presentación más clara. o Durante la presentación, deben gestionar preguntas y responder con claridad. **3. Retroalimentación**: Después de cada presentación, los compañeros y el docente brindan retroalimentación constructiva sobre la claridad, la estructura, la efectividad del lenguaje y el manejo de las preguntas	• Computadoras con software para crear presentaciones (como PowerPoint, Google Slides). • Herramientas para grabar el proyecto (si es necesario para demostración). • Reloj o cronómetro para gestionar el tiempo de la presentación.
Cierre (20 minutos)	**Estrategia**: Reflexión sobre el proceso de presentación y cierre con retroalimentación final. **Actividad**: 1. **Reflexión grupal** sobre las lecciones aprendidas durante la práctica de presentación. o Los estudiantes comparten lo que les resultó más desafiante al presentar y cómo lograron adaptarse a las preguntas del público.	• Pizarra o proyector para hacer anotaciones y reflexiones. • Ficha de evaluación para los estudiantes (evaluación del desempeño y de la presentación)

	○ Discusión sobre cómo mejorar las presentaciones en futuras ocasiones, considerando la retroalimentación recibida. 2. **Feedback final**: El docente realiza una evaluación general sobre las presentaciones realizadas, señalando fortalezas y áreas de mejora para los estudiantes. 3. **Resumen y recomendaciones**: El docente ofrece recomendaciones adicionales para mejorar las habilidades de presentación, como la importancia de la práctica continua, la autoevaluación, y la preparación para preguntas difíciles.	





Sesión de aprendizaje 12

I.DATOS GENERALES:

a. Institución Educativa :
b. UGEL :
c. Área : Desarrollo de Sistemas de Información
d. Ciclo : I
e. Grado :
f. Turno : Mañana
g. Docente : Ing. Hans Edgardo Delgado Díaz
h. Duración : 90 minutos
i. Fecha : Semana 6

II.TITULO

"Evaluar y mejorar el proyecto final basado en retroalimentación"

III.PROPÓSITO:

Competencia	Capacidades	Criterios de evaluación	Evidencia	Instrumento
Evaluar y mejorar el proyecto final basado en retroalimentación.	Evaluar la calidad de su proyecto final, identificando aspectos que pueden ser mejorados Aplicar la retroalimentación recibida para realizar ajustes técnicos y de diseño en el proyecto final Mejorar la calidad de su proyecto final mediante la implementación de cambios derivados de la retroalimentación	**Evaluación de Proyecto Final:** • El estudiante debe identificar correctamente las fortalezas y debilidades de su proyecto. **Implementación de Retroalimentación:** • El estudiante debe aplicar de manera efectiva las recomendaciones recibidas. **Mejora del Proyecto Final:** • El estudiante debe mostrar evidencia clara de que ha mejorado su proyecto en aspectos técnicos, de diseño y/o funcionalidad.	**Informe de Evaluación:** El estudiante entregará un informe escrito donde se describe el análisis del proyecto final, las retroalimentaciones recibidas y las mejoras implementadas. **Versión Mejorada del Proyecto:** El estudiante entregará una versión del proyecto con las mejoras implementadas según la retroalimentación.	**Lista de Chequeo de la Evaluación del Proyecto:** Para evaluar las fortalezas y debilidades del proyecto. **Rubrica de Retroalimentación:** Para evaluar cómo el estudiante ha integrado las sugerencias en el proyecto final.

Competencia transversal	Capacidades	Criterios	Evidencia	
Comunicación efectiva.	• Comunicar de forma clara y estructurada los aspectos mejorados de su proyecto. • Explicar de manera comprensible las razones detrás de las mejoras implementadas, utilizando un lenguaje técnico apropiado.	• Claridad en la comunicación sobre los aspectos del proyecto mejorados. • Uso adecuado de la terminología técnica al explicar los cambios realizados.	**Presentación Oral:** El estudiante presentará los cambios realizados en su proyecto, detallando las mejoras y justificaciones de forma clara. **Documento Explicativo:** Un documento escrito en el que el estudiante describe los ajustes realizados en el proyecto a partir de la retroalimentación.	
				
Enfoque Transversal				
	Aprendizaje basado en proyectos (ABP). **Descripción:** A través del ABP, los estudiantes aprenden evaluando, ajustando y mejorando su proyecto, lo que les permite integrar conocimientos teóricos con habilidades prácticas de programación. Además, se fomenta la comunicación efectiva de los resultados y el trabajo colaborativo en el proceso de mejora.			

IV.ORGANIZACIÓN

Momentos	Estrategias/ actividades	Recursos
Inicio (20 minutos)	**Estrategias:** • Presentación breve del objetivo de la sesión: mejorar el proyecto final mediante retroalimentación.	• Pizarra/whiteboard (para registrar algunas respuestas comunes o dudas). • Plantillas de autoevaluación del

	• Realización de una actividad inicial en la que los estudiantes reflexionen sobre su proyecto y identifiquen áreas de mejora antes de recibir la retroalimentación. **Actividades:** 1. **Reflexión Individual:** o Los estudiantes, de manera individual, realizan un análisis preliminar de su proyecto, señalando posibles aspectos a mejorar. o Esta reflexión servirá como base para la retroalimentación posterior.	proyecto (proporcionadas por el instructor). 
Desarrollo (50 minutos)	**Estrategias:** • Entrega y explicación de las retroalimentaciones (pueden ser retroalimentaciones generales del profesor o retroalimentaciones entre compañeros). • Aplicación de las retroalimentaciones en tiempo real. • Supervisión y asesoramiento individualizado durante el proceso. **Actividades:** 1. **Entrega de Retroalimentación:** o El instructor o los compañeros de clase proporcionan retroalimentación a cada estudiante sobre su proyecto. o La retroalimentación debe incluir comentarios específicos sobre código, diseño, funcionalidad, etc. 2. **Mejoras y Ajustes:** o Los estudiantes ajustan su proyecto tomando en cuenta la retroalimentación recibida, mejorando los aspectos señalados. o Durante esta actividad, el instructor realiza rondas de retroalimentación personalizada.	• Retroalimentación escrita (por parte del instructor o compañeros). • Computadoras con software necesario para implementar las mejoras en los proyectos. • Guías para aplicar retroalimentación efectiva.
Cierre (20 minutos)	**Estrategias:** • Resumen y reflexión final. • Presentación de los proyectos mejorados y discusión. **Actividades:** 1. **Reflexión grupal:** o Los estudiantes se agrupan para compartir cómo la retroalimentación recibida les ayudó a mejorar su proyecto. o Discusión sobre los retos y aprendizajes adquiridos durante la mejora del proyecto.	• Pizarra/whiteboard para resumir puntos clave de la discusión final. • Proyector (si los estudiantes presentan sus proyectos a través de pantallas).

	2. **Presentación Final:** o Los estudiantes presentan brevemente sus proyectos mejorados, explicando las mejoras realizadas y las razones detrás de los cambios.	



	2. **Presentación Final:** ○ Los estudiantes presentan brevemente sus proyectos mejorados, explicando las mejoras realizadas y las razones detrás de los cambios.	



DR. GILBERTO CARRION-BARCO
INVESTIGADOR RENACYT
Código P0070731

	2. **Presentación Final:** ○ Los estudiantes presentan brevemente sus proyectos mejorados, explicando las mejoras realizadas y las razones detrás de los cambios.	



Mg. Janett Aracely Gonzales Flores

DNI: 73618769

Anexo 7: Determinación de los baremos para interpretar las puntuaciones en el Instrumento

Descriptivas

	suma
N	15
Perdidos	0
Media	12.1
Mediana	13
Desviación estándar	3.37
Mínimo	6
Máximo	16
20percentil	10.2
40percentil	12.2
60percentil	14
80percentil	14.2

MINIMO	0	0
MAXIMO	1	20
DEFICIENTE	0-10	
BASICO	11 - 12	
SUFICIENTE	13 - 14	
SOBRESALIENTE	15 - 20	

Anexo 9: Imágenes de la aplicación desarrolla por el autor de la presente investigación



Figura 6: Pantalla de entrada de la aplicación móvil

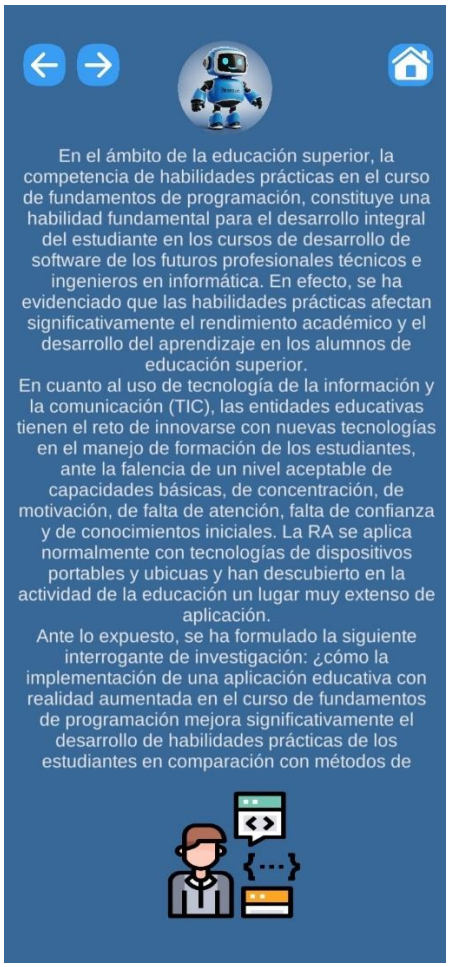


Figura 7: Pantalla de introducción sobre fundamentos de programación.

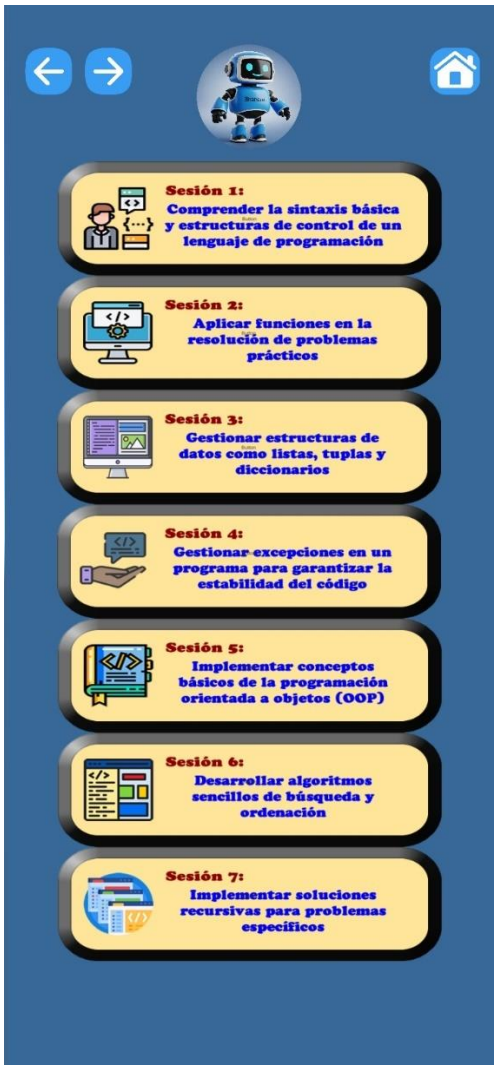


Figura 8: Pantalla con botones de acceso a las 7 primeras sesiones de clase programadas

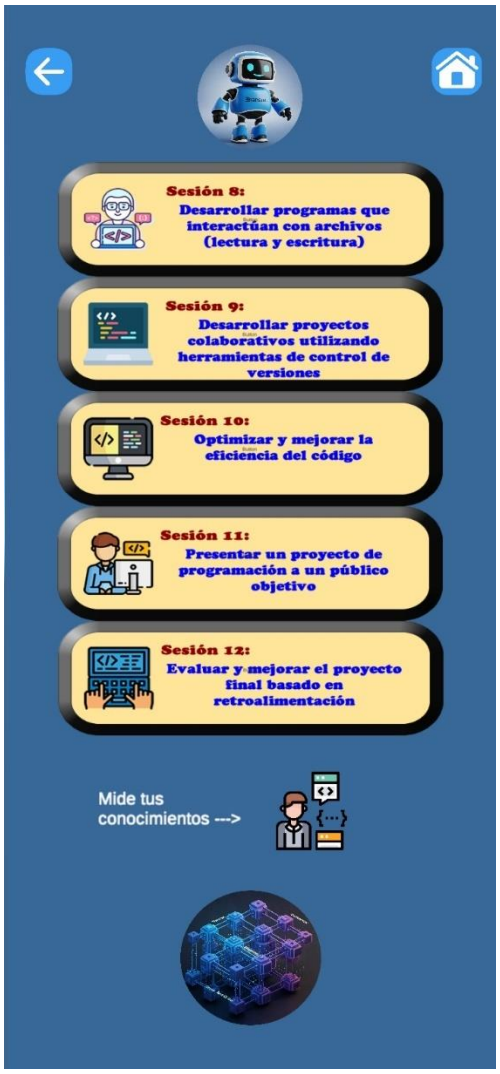


Figura 9: Pantalla con los botones de acceso a las 5 sesiones de clase programadas restantes y, asimismo, al botón de acceso a juegos educativos sobre temas teóricos de fundamentos de programación

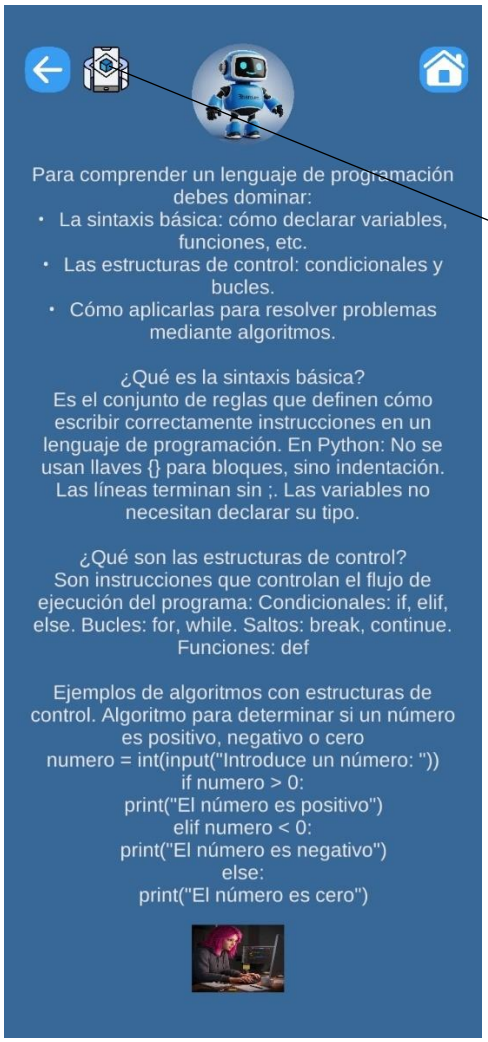


Figura 10: Pantalla con el contenido de una de las sesiones de clase

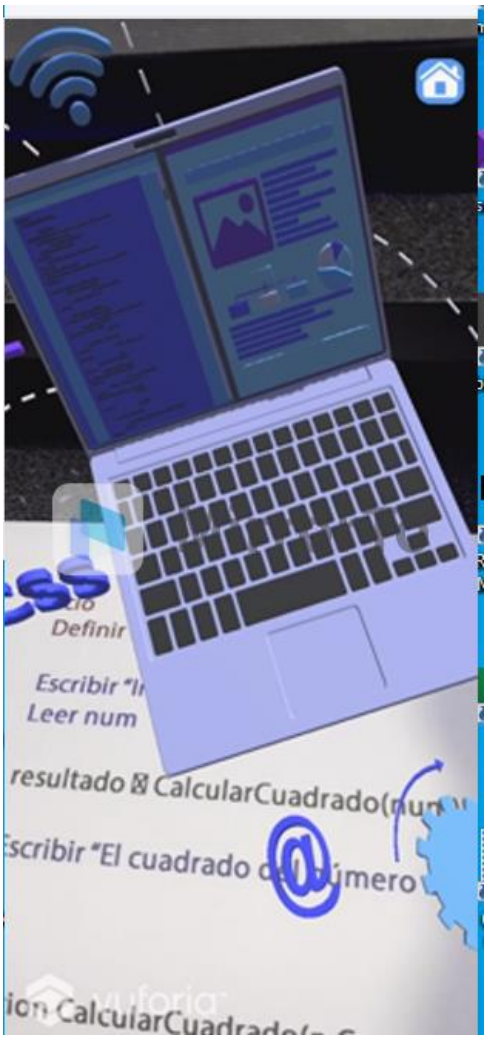


Figura 11: Efecto de realidad aumentada en cada sesión, nótese en la parte inferior el marcador usado para la muestra del efecto. Cada marcador es un algoritmo de acuerdo al tema de sesión de clase.

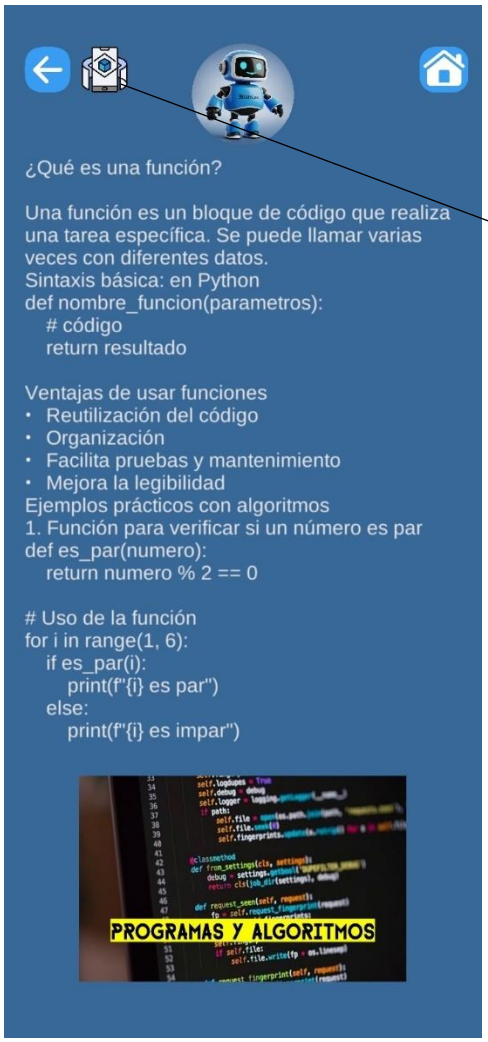


Figura 12: Otra pantalla de sesión de clase, observándose el contenido con ejemplos y los botones de control de la aplicación.

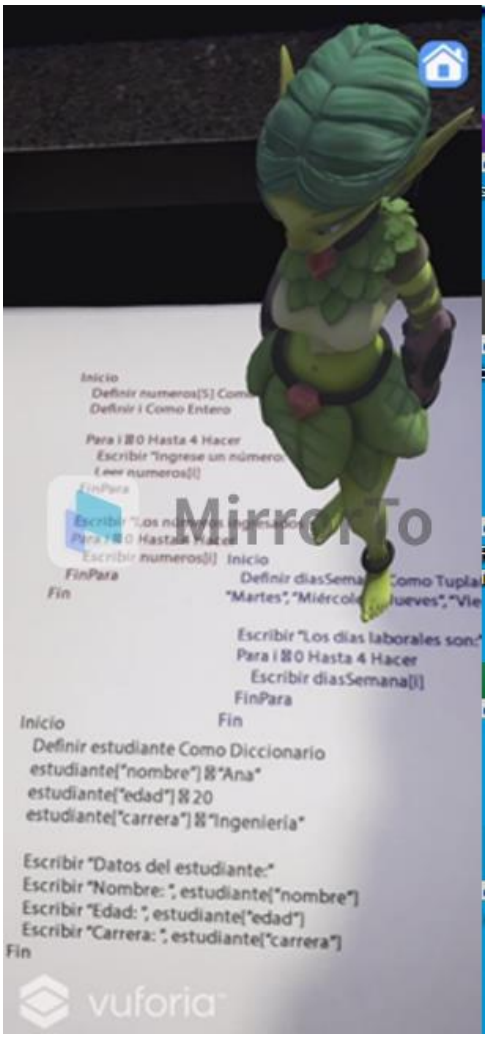


Figura 13: Efecto de realidad aumentada de la sesión de clase. Por cada sesión de clase es un efecto diferente de realidad aumentada. Nótese el marcador de fondo usado para mostrar el efecto RA, marcador con algoritmos de acuerdo a la sesión de clase correspondiente.



Figura 14: Pantalla final con los botones de acceso a juegos educativos elaborados en Educaplay sobre fundamentos de programación

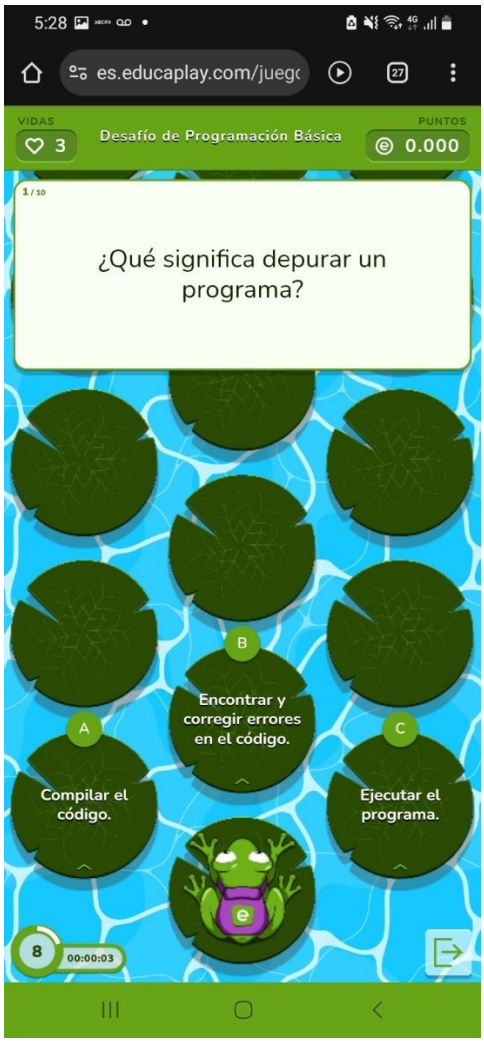


Figura 15: Acceso al primer juego elaborado en Educaplay sobre fundamentos de programación



Figura 16: Segundo juego educativo en Educaply sobre temas de fundamentos de programación

Anexo 8: Imágenes de evidencia de aplicación del instrumento Pre Test y Pos Test



Figura 17: resolviendo el cuestionario del instrumento

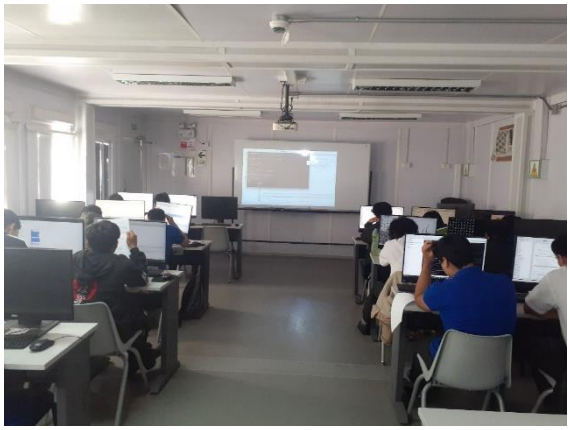


Figura 18: otra toma de la resolución del cuestionario del instrumento



Figura 19: alumnos resolviendo el cuestionario del instrumento



Figura 20: en la resolución del cuestionario del instrumento

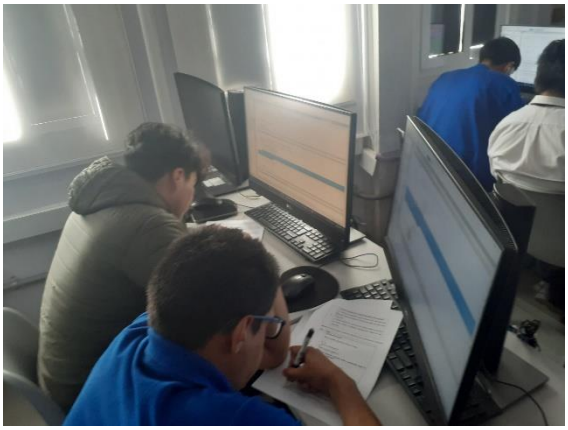


Figura 21: alumnos en la resolución del cuestionario del instrumento

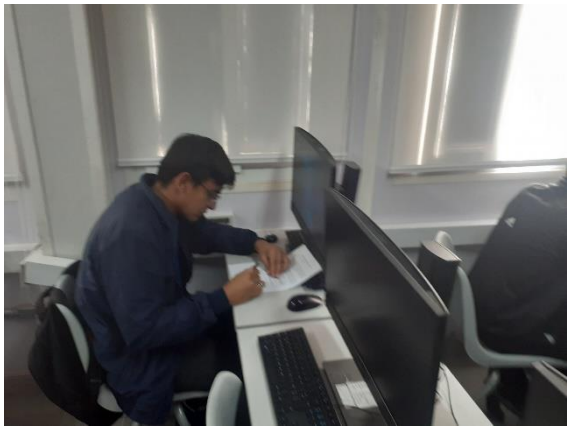


Figura 22: resolviendo el cuestionario del instrumento



Figura 23: otra toma de los alumnos resolviendo el cuestionario del instrumento



Figura 24: alumna en el uso del aplicativo móvil leyendo el material de fundamentos de programación publicado

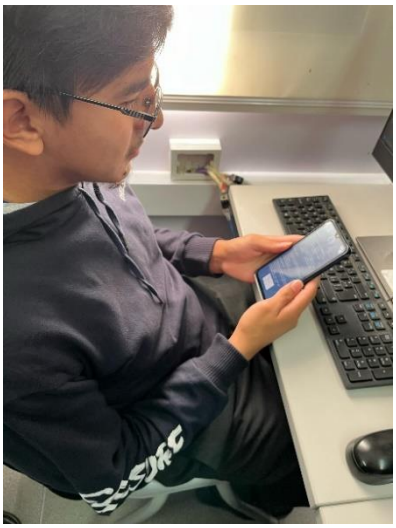


Figura 25: alumno en el uso del aplicativo móvil leyendo la información de fundamentos de programación publicado

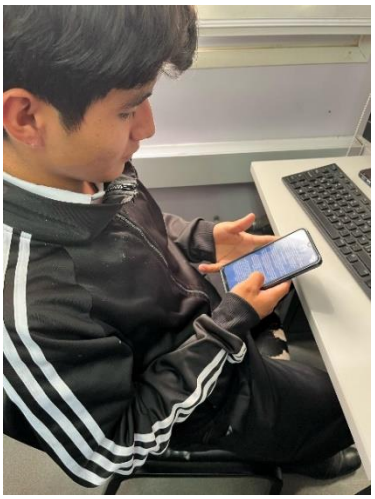


Figura 27: alumno en el uso del aplicativo móvil en la lectura del material publicado

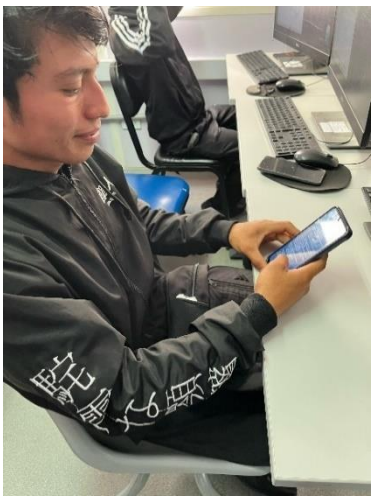


Figura 29: alumno en la lectura del material del aplicativo móvil

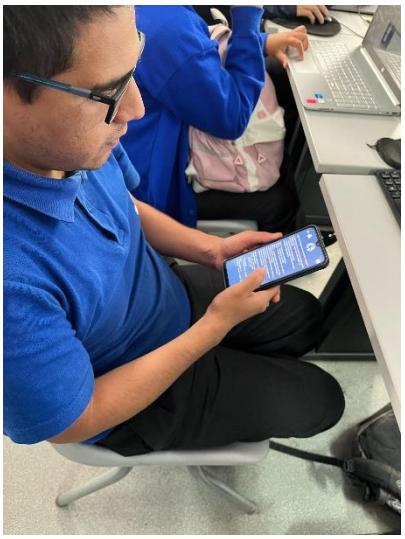


Figura 26: alumno en el estudio de la información del aplicativo móvil



Figura 28: alumno analizando la información del aplicativo móvil

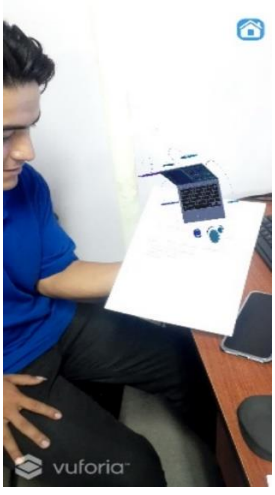


Figura 30: alumno aplicando el efecto de realidad aumentada con el marcador de estudio que usa el aplicativo móvil

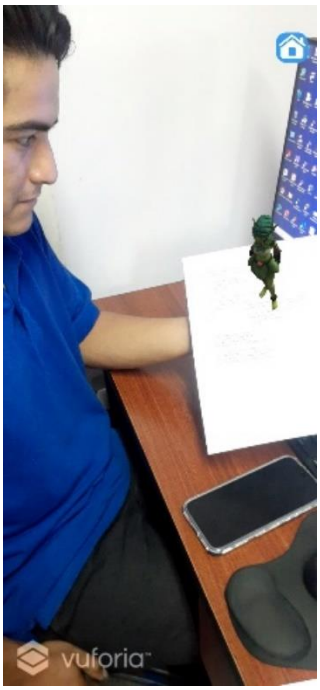


Figura 31: alumno aplicando otro efecto de realidad aumentada con uno de los marcadores que usa el aplicativo móvil

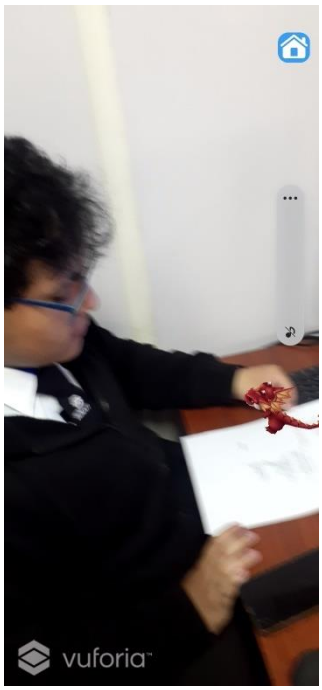


Figura 32: otro alumno aplicando el efecto de realidad aumentada con otro marcador que usa el aplicativo móvil



Figura 33: alumno observando el efecto de realidad aumentada de la aplicación sobre uno de los marcadores

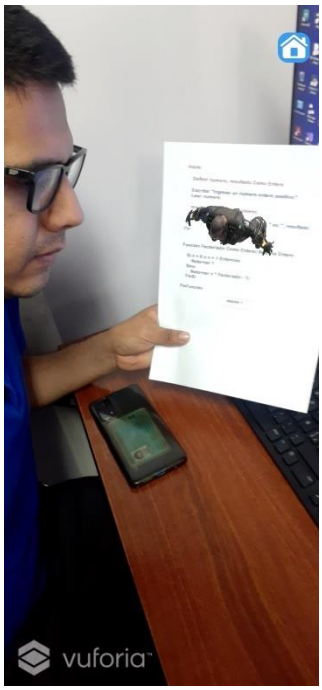


Figura 34: alumno observando el efecto de realidad aumentada del aplicativo móvil



Figura 35: aplicando el efecto de realidad con uno de los marcadores del aplicativo móvil

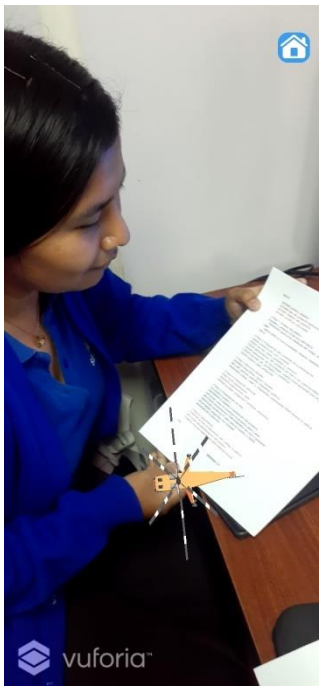


Figura 36: alumna observando otro efecto de realidad aumentada del aplicativo móvil

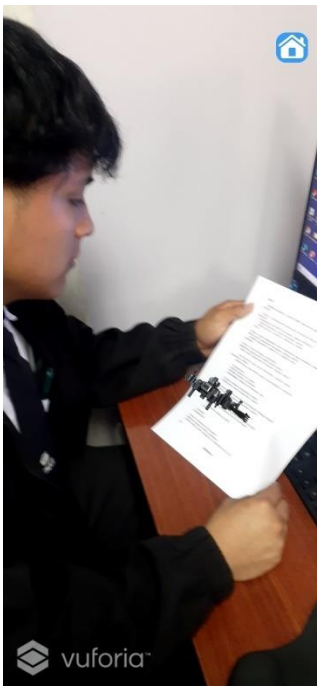


Figura 37: alumno observando otro efecto de realidad aumentada de la aplicación móvil



Figura 38: alumno analizando el efecto de realidad aumentada que emite el aplicativo móvil