



Revista Especializada de Ingeniería
y Ciencias de la Tierra

VOL: 6 N° 1 julio - diciembre 2026

ISSN L: 2805-1874

Análisis de Datos Experimentales en Laboratorios Universitarios de Física mediante Python Asistido por Inteligencia Artificial Generativa

Experimental Data Analysis in University Physics Laboratories Using Python Assisted by Generative Artificial Intelligence

Humberto Antonio Edward Hernández
Universidad de Panamá, CRU Coclé, Panamá.
humberto.edwardh@up.ac.pa
<https://orcid.org/0000-0002-2658-2892>

Juan David Cedeño Camarena
Ministerio de Educación, Panamá.
juan.cedenoc@up.ac.pa
<https://orcid.org/0000-0001-7511-8175>

Marisol Gisela Nicholson Gómez
Ministerio de Educación, Panamá.
marisolnicholson28@gmail.com
<https://orcid.org/0000-0003-3814-0175>

Adam's Martínez Soto
Ministerio de Educación, Panamá.
adams.martinez@up.ac.pa
<https://orcid.org/0000-0001-8060-686X>

Alberto Caballero
Universidad de Panamá, Facultad de Ciencias Naturales Exactas y Tecnología, Departamento de física,
Panamá.
alberto.caballero@up.ac.pa
<https://orcid.org/0000-0001-8502-8288>

Recibido: 9/2/2026 Aceptado: 10/5/2026

DOI <https://doi.org/10.48204/reicit.v6n1.10119>

RESUMEN

El análisis de datos en laboratorios universitarios de física suele depender de softwares comerciales cerrados, lo que limita la reproducibilidad, la transparencia del procesamiento y la adaptabilidad a distintos contextos experimentales. El objetivo de este trabajo es presentar un flujo de trabajo computacional abierto, basado en Python y asistido por herramientas de inteligencia artificial generativa, para el análisis y visualización de datos experimentales en cursos de laboratorio de física. La propuesta se implementa mediante un script multifuncional de código abierto, desarrollado con apoyo de IA generativa para la construcción y depuración iterativa del código, garantizando la trazabilidad de los procesos computacionales. El flujo de trabajo integra la adquisición de datos, el análisis estadístico y la generación de visualizaciones de alta resolución en un entorno reproducible. Su viabilidad técnica se evalúa a través de tres experiencias representativas de laboratorio, incluyendo análisis de regresiones, estadística descriptiva y mapeo de campos eléctricos. Los resultados evidencian que la herramienta permite sustituir el software comercial sin pérdida significativa en la precisión del análisis ni en la calidad de las visualizaciones. En consecuencia, se concluye que el uso de flujos de trabajo abiertos, asistidos por inteligencia artificial generativa, constituye una alternativa flexible, reproducible y extensible para el tratamiento computacional de datos experimentales en la enseñanza de la física en distintos contextos experimentales.

PALABRAS CLAVE: Python, análisis de datos experimentales, inteligencia artificial generativa, enseñanza de la física, laboratorios de física.

ABSTRACT

Data analysis in university physics laboratories often relies on proprietary software, limiting reproducibility, transparency, and adaptability across different experimental contexts. The objective of this study is to present an open computational workflow based on Python and assisted by generative artificial intelligence tools for the analysis and visualization of experimental data in physics laboratory courses. The proposal is implemented through a multifunctional open-source script, developed with the support of generative AI for iterative code construction and debugging, ensuring full traceability of computational processes. The workflow integrates data acquisition, statistical analysis, and high-resolution visualization within a reproducible environment. Its technical feasibility is evaluated through three representative laboratory experiences, including regression analysis, descriptive statistics, and electric field mapping. The results show that the proposed tool can replace proprietary software without significant loss in analytical accuracy or visualization quality. It is concluded that the use of open workflows assisted by generative artificial intelligence constitutes a flexible, reproducible, and extensible alternative for computational data processing in physics education across different experimental contexts.

KEYWORDS: Python, experimental data analysis, generative artificial intelligence, physics education, physics laboratories.

INTRODUCCIÓN

Las actividades experimentales constituyen un componente central en la formación universitaria en ciencias naturales, al permitir la articulación entre modelos teóricos, observación empírica y análisis de datos (Campos & Benarroch, 2024, p. 110; Plaza & Saquinaula, 2024, p. 6150). La integración de la programación en la enseñanza de las ciencias básicas se ha consolidado como una necesidad pedagógica en diversos contextos educativos. Al respecto, Bernal y Peña sostienen que el fortalecimiento de las competencias investigativas se ve potenciado por la “introducción de lenguajes de programación de alto nivel como Python en los procesos de enseñanza-aprendizaje de la física aplicada” (2023, p. 1). No obstante, en muchos laboratorios introductorios estas actividades dependen de guías rígidas y de software comercial cerrado, lo que limita la transparencia de los procesos de análisis, la reproducibilidad de los resultados y la posibilidad de adaptación a distintos contextos experimentales (Cardona & López, 2017, p. 1; Monajem & Felfer, 2022, p. 732).

Esta dependencia instrumental desplaza el foco pedagógico desde el análisis crítico hacia la ejecución mecánica de procedimientos, limitando el desarrollo de competencias analíticas en contextos experimentales. En el contexto local, esta brecha se hizo más evidente tras la pandemia, que subrayó la necesidad de una transformación en la praxis docente digital para adaptarse a entornos virtuales de aprendizaje (Gordon Graell, 2021, p. 132). En este escenario, la integración de herramientas como Python y la inteligencia artificial generativa emerge como una alternativa para fortalecer el análisis de datos en laboratorios de física. Incluso hay trabajos como el de Muñoz Escudero (2025, p. 15) que explorara cómo estas tecnologías pueden mitigar dichas deficiencias mediante la personalización de la enseñanza.

En las carreras de perfil tecnológico, esta situación se ve reforzada por una percepción de desconexión entre los contenidos de física y las competencias técnicas propias de la especialidad. La naturaleza abstracta de muchos modelos físicos, junto con el uso intensivo de formalismo matemático, dificulta que los estudiantes identifiquen la relevancia operativa de estos conceptos en contextos aplicados, particularmente en los cursos iniciales (Bufasi & Lakrad, 2019, p. 1; Knapp et al., 2025, p. 740). Como resultado, el trabajo experimental suele desarrollarse en entornos aislados de las herramientas computacionales empleadas en otras asignaturas técnicas, lo que limita la transferencia de habilidades y la integración interdisciplinaria (Hernández & Gómez, 2023, p. 1).

Bajo este panorama, el uso de herramientas computacionales abiertas se presenta como una vía para replantear el laboratorio universitario como un entorno flexible, reproducible y extensible. El lenguaje de programación Python se ha consolidado como un estándar ampliamente adoptado en la computación científica y el análisis de datos, con un ecosistema robusto de bibliotecas para cálculo numérico, visualización y modelización (Lincoln, 2019, p. 60; Núñez et al., 2024, p. 179). Su incorporación en contextos experimentales permite sustituir flujos de trabajo dependientes de software propietario por soluciones transparentes y personalizables, alineadas con prácticas contemporáneas de reproducibilidad científica (Batista-Mendoza et al., 2025, p. 9; Koerner et al., 2020, p. 1698; Weber, 2021, p. 1).

El uso de entornos interactivos como Jupyter Notebooks y plataformas basadas en la nube facilita la integración entre adquisición de datos, análisis y visualización dentro de un mismo flujo de trabajo computacional (Fortune et al., 2021, p. 2; Hernández & Gómez, 2023, p. 1; Tufiño et al., 2024, p. 1). Estas herramientas permiten explorar de forma directa la relación entre variables físicas y modelos matemáticos, favoreciendo un tratamiento explícito y verificable de los datos experimentales (Gäßler et

al., 2022, p. 1; Hernández & Gómez, 2023, p. 1). En este marco, la incorporación de herramientas de inteligencia artificial generativa introduce una capa adicional de asistencia al desarrollo y depuración de código, así como a la exploración y visualización de datos, reduciendo barreras técnicas iniciales sin ocultar los procesos computacionales involucrados (Fernández et al., 2025, p. 18; Freire-Avilés et al., 2024, p. 4713; Hare, 2024, p. 3869; Núñez et al., 2024, p. 179).

A pesar de la proliferación de herramientas basadas en Python para distintos aspectos del trabajo experimental, la literatura evidencia que estas soluciones suelen abordar etapas específicas del flujo experimental y presentan un alto grado de fragmentación, sin integrarse en plataformas generales de uso extensivo (Koerner et al., 2020, p. 1698; Stansbury & Lanzara, 2023, p. 121; Weber, 2021, p. 1). Asimismo, el uso sistemático de inteligencia artificial generativa como apoyo directo al desarrollo de código científico en laboratorios de física permanece escasamente documentado desde una perspectiva técnica y aplicada.

En este contexto, el objetivo de este estudio es proponer el uso de Python asistido por inteligencia artificial generativa como una alternativa abierta al software comercial en laboratorios universitarios de física experimental. La propuesta se implementa a través de tres experiencias de laboratorio que integran programación, experimentación y análisis de datos en un flujo de trabajo reproducible, con el fin de evaluar su viabilidad como herramienta computacional para el tratamiento de datos experimentales en distintos escenarios de laboratorio. Se plantea como supuesto que la integración de herramientas abiertas basadas en Python, asistidas por inteligencia artificial generativa, permite mejorar la reproducibilidad, la transparencia del análisis y la integración entre el trabajo experimental y el procesamiento computacional de datos.

MÉTODO Y MATERIALES

El estudio se desarrolló bajo un enfoque cualitativo de carácter aplicado, orientado al diseño, implementación y sistematización de una herramienta computacional para el análisis de datos experimentales en laboratorios universitarios de física. El diseño metodológico se centró en evaluar la viabilidad técnica del uso de Python asistido por inteligencia artificial generativa como alternativa abierta al software comercial, así como en documentar su integración en el flujo de trabajo experimental.

El núcleo de la propuesta es un script multifuncional desarrollado en Python (https://colab.research.google.com/drive/1cq4gTJVJoj_ZYoFmI_7xW_Rr7Ka8-HU0?usp=sharing), concebido para el análisis y visualización de datos experimentales. Este enfoque se fundamenta en el uso del modelado computacional como herramienta de apoyo en la enseñanza de la física, el cual ha sido implementado en cursos introductorios con resultados favorables en el desarrollo de competencias técnicas (Caballero et al., 2012, p. 1).

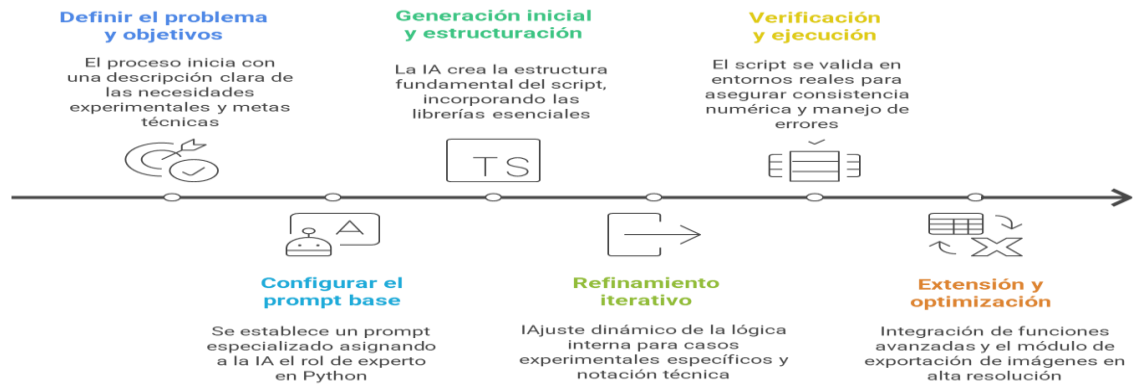
El desarrollo del código se realizó con el apoyo de herramientas de inteligencia artificial generativa (ChatGPT, Gemini y GitHub Copilot), utilizadas para generar, depurar y optimizar bloques de código mediante prompts específicos ajustados a cada problema experimental. Este proceso se estructuró a partir de un flujo de trabajo que incluyó la formulación técnica de prompts y la verificación funcional del código, en concordancia con enfoques que incorporan la inteligencia artificial como apoyo en procesos

educativos y de resolución de problemas (Gordon Graell, 2022, p. 53; Diaz Villavicencio et al., 2025, p. 147).

Este proceso se llevó a cabo de forma iterativa, con el objetivo de garantizar una estructura modular y reutilizable del script, adaptable a distintos formatos y conjuntos de datos experimentales (Fernández-Marín et al., 2025, p. 20). Este proceso de desarrollo asistido por IA se sistematizó mediante un flujo de trabajo que incluyó la formulación inicial de prompts, la evaluación funcional del código generado, su refinamiento progresivo y la validación de los resultados obtenidos, tal como se muestra en la Figura 1.

Figura 1

Proceso de desarrollo de código Python asistido por IA generativa

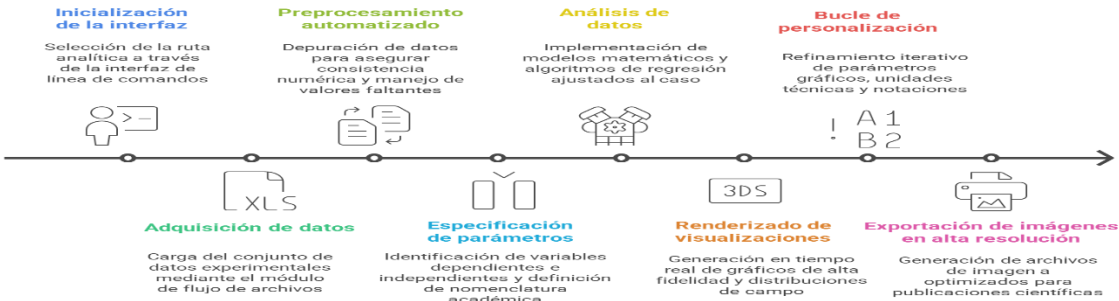


Nota. La figura muestra las etapas del proceso: definición del problema y objetivos, configuración del prompt base, generación inicial y estructuración, refinamiento iterativo, verificación y ejecución, y extensión y optimización. Imagen generada con apoyo de la herramienta Napkin.

La herramienta fue implementada en cursos universitarios de laboratorio de física, sustituyendo el uso de software comercial por la solución de código abierto desarrollada. La aplicación se estructuró a través de un flujo de trabajo que integra la recolección de datos experimentales, su carga en el script, el análisis estadístico y la exportación de resultados gráficos en formatos de alta resolución, como se presenta en la Figura 2.

Figura 2

Flujo de trabajo para el análisis de datos experimentales mediante un programa en Python



Nota. La figura presenta las etapas del flujo de trabajo: inicialización de la interfaz, adquisición de datos, preprocesamiento automatizado, especificación de parámetros, análisis de datos, renderizado de visualizaciones, bucle de personalización y exportación de imágenes en alta resolución. Imagen generada y adaptada con apoyo de la herramienta Napkin.

Este flujo de trabajo fue aplicado de manera consistente en tres experiencias representativas de laboratorio, cuya organización general se describe en la figura 3.

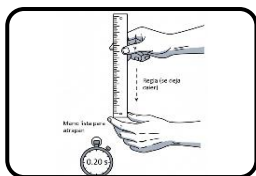
Figura 3

Descripción de las actividades experimentales desarrolladas



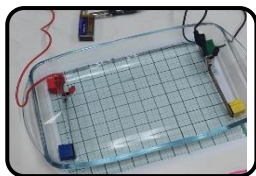
Relación entre masa total y número de granos en distintos tipos de menestras

Se midieron las masas de diferentes granos, registrando su masa en función del número de granos para cada tipo. Los datos fueron procesados en Python, generando representaciones gráficas y ajustando modelos de regresión lineal, a partir de los cuales se obtuvieron las ecuaciones características de cada conjunto para su posterior interpretación.



Tiempo de reacción de un estudiante

Se midieron los tiempos de reacción en diferentes personas mediante la clásica prueba de caída de la regla. Los datos fueron analizados en Python utilizando diagramas de cajas y bigotes, lo que permitió visualizar la dispersión, la mediana y los valores atípicos, facilitando la interpretación de la variabilidad en las respuestas humanas.



Distribución del campo eléctrico en una cubeta con electrodos

Se colocaron dos láminas rectangulares como electrodos en una cubeta con solución conductora y se midieron los voltajes en distintos puntos del sistema. Con estos datos, Python permitió construir mapas de contorno que visualizaron las líneas equipotenciales y derivar de ellas las líneas de campo eléctrico a través de diagramas vectoriales.

Nota. La figura presenta tres experiencias de laboratorio implementadas: (a) relación entre masa total y número de granos en distintos tipos de muestras, analizada mediante regresión lineal; (b) medición del tiempo de reacción utilizando la prueba de caída de la regla, con análisis mediante diagramas de cajas y bigotes; y (c) distribución del campo eléctrico en una cubeta con electrodos, a partir de mediciones de voltaje y su representación mediante mapas de contorno y diagramas vectoriales. Los datos fueron procesados mediante scripts desarrollados en Python.

Las experiencias diseñadas incluyeron tareas de análisis de regresiones, visualización de datos, elaboración de histogramas y generación de mapas de contorno, seleccionando los tipos de gráficos y métodos de ajuste en función de la naturaleza de cada conjunto de datos y del objetivo experimental correspondiente. Estas actividades permitieron evaluar la flexibilidad del script para adaptarse a distintos escenarios experimentales sin modificaciones estructurales significativas.

De manera complementaria al uso del script en experiencias experimentales reales, se incorporaron actividades de carácter didáctico y exploratorio, orientadas a operacionalizar el análisis experimental mediante programación y simulación. Estas actividades incluyeron la conversión de magnitudes físicas y

la simulación de fenómenos de cinemática y dinámica, integrando la generación y el análisis de datos simulados dentro del mismo flujo computacional.

Estas actividades no formaron parte del conjunto de experiencias utilizadas para la validación técnica del modelo, sino que se emplearon como recursos pedagógicos para reforzar la comprensión de la relación entre modelos teóricos y su implementación computacional (Bernal & Peña, 2023, p. 2; Hernández & Gómez, 2023, p. 2). Este enfoque es coherente con el uso de plataformas abiertas para el ajuste de modelos físicos en cursos de laboratorio (Gäßler et al., 2022, p. 1) y con la integración de herramientas de análisis computacional en cursos introductorios de física (Knapp et al., 2025, p. 739; Tufiño et al., 2024, p. 1).

La personalización del código para cada experiencia experimental se facilitó mediante el uso de prompts estructurados, partiendo de formulaciones genéricas que especificaban el formato de los datos, el tipo de análisis requerido y las salidas gráficas esperadas. Un ejemplo de prompt inicial utilizado en este proceso se presenta a continuación:

“Asume el rol de experto en edición de prompts y redacción de código para Python. Tu misión es crear un script funcional y fácil de entender, que cargue un archivo de datos en formato _____, genere una gráfica de tipo _____ utilizando las columnas _____ y _____, emplee las librerías esenciales, incluya comentarios explicativos y guarde la gráfica resultante como un archivo PNG a 300 ppp.”

La validación del modelo se basó en la sistematización del diseño del flujo computacional y en la comprobación del rigor analítico de los resultados obtenidos. Como criterios principales se consideraron la precisión estadística de los ajustes realizados y la consistencia de los coeficientes de determinación (R^2) en los análisis de regresión, así como la coherencia de las distribuciones obtenidas en los distintos escenarios experimentales. Estos indicadores permitieron evaluar la viabilidad técnica del programa para sustituir software comercial sin pérdida de calidad en el tratamiento de los datos.

Finalmente, el modelo se estructuró en torno a tres principios operativos: accesibilidad, reproducibilidad y escalabilidad. La accesibilidad se garantiza mediante el uso exclusivo de software libre; la reproducibilidad, a través de código documentado y adaptable; y la escalabilidad, mediante la posibilidad de aplicar el mismo flujo de trabajo a otros contenidos de la física experimental.

RESULTADOS

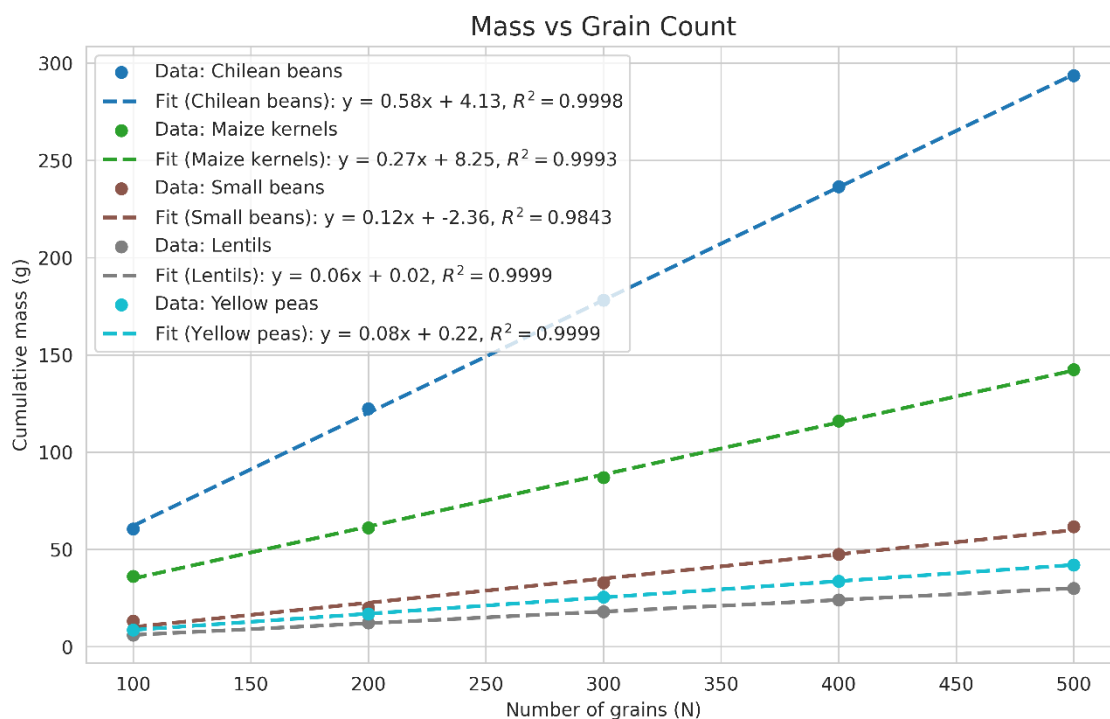
Los datos experimentales obtenidos en las distintas experiencias fueron procesados mediante el programa multifuncional desarrollado en Python, generando representaciones gráficas y métricas estadísticas consistentes con los modelos físicos esperados. A continuación, se presentan los resultados correspondientes a cada experiencia.

a) Variación de la masa de menestras

El análisis de cinco variedades de menestras evidenció una relación lineal entre el número de granos y la masa total (Figura 4). Las regresiones lineales obtenidas presentan coeficientes de determinación superiores a 0.98 en todos los casos, lo que confirma una proporcionalidad directa entre ambas variables dentro del rango experimental considerado. Asimismo, los altos coeficientes de determinación obtenidos evidencian la consistencia del ajuste lineal realizado mediante el programa.

Figura 4

Relación entre el número de granos y la masa total en cinco variedades de menestras



Nota. Se presentan las regresiones lineales obtenidas mediante ajuste por mínimos cuadrados para frijoles chilenos, maíz, frijoles pequeños, lentejas y arvejas. Las pendientes de cada recta representan la masa promedio por grano, mientras que los coeficientes de determinación (R^2) indican el grado de ajuste de cada modelo.

Las pendientes de las rectas corresponden al peso promedio de cada grano y muestran diferencias sistemáticas entre las variedades analizadas. Los porotos chilenos presentan la mayor pendiente (≈ 0.58 g por grano), seguidos por el maíz (≈ 0.27 g) y los frijolitos (≈ 0.12 g). Las arvejas (≈ 0.08 g) y las lentejas (≈ 0.06 g) exhiben las pendientes más bajas, coherentes con su menor tamaño y masa individual.

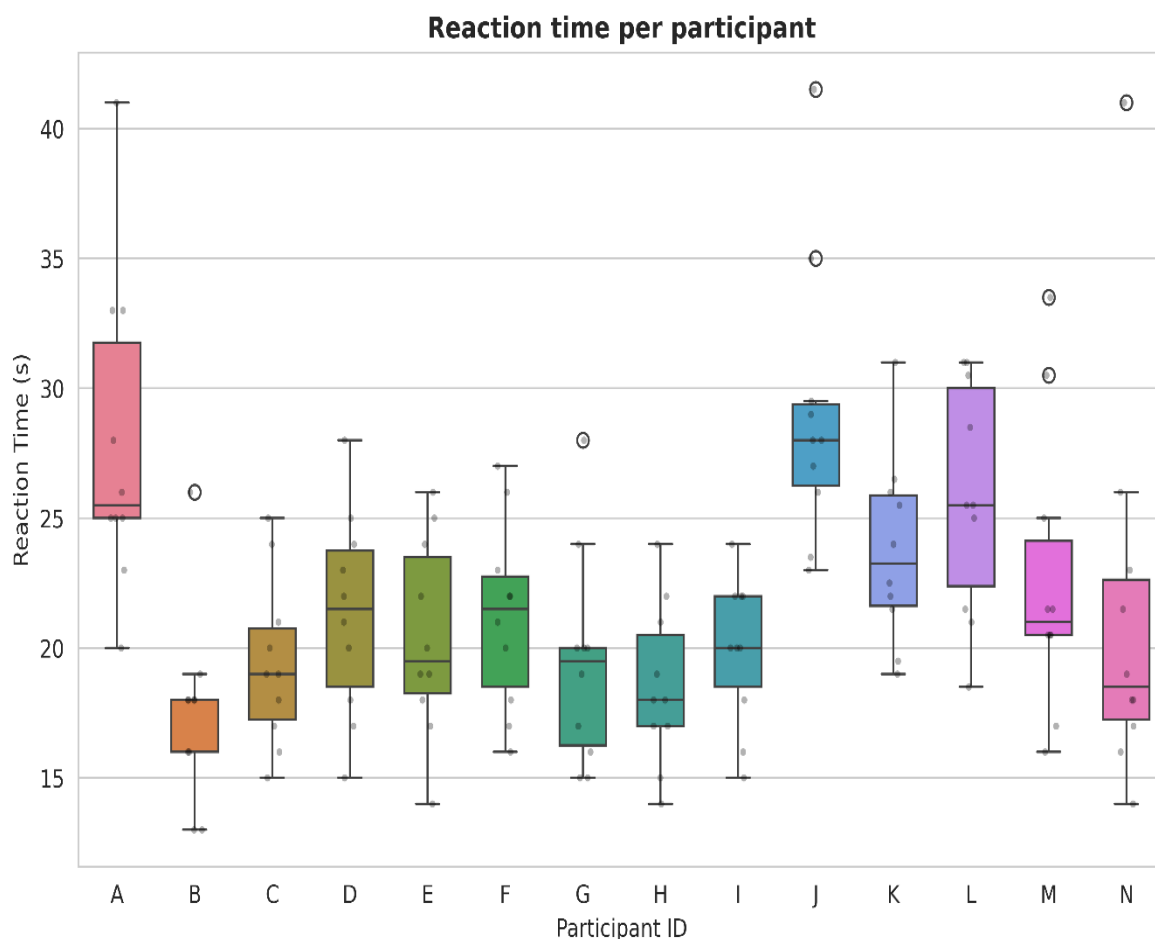
Las ordenadas al origen difieren de cero, lo que puede atribuirse a incertidumbres experimentales y al ajuste matemático del modelo. Este comportamiento es consistente con mediciones reales y no afecta la validez del modelo lineal dentro del intervalo de datos analizado.

b) Experiencia del tiempo de reacción

Los tiempos de reacción medidos mediante la prueba de caída de la regla fueron analizados utilizando diagramas de cajas y bigotes (Figura 5). Esta representación permitió caracterizar la distribución de los datos, identificando medianas, dispersión y valores atípicos para cada participante.

Figura 5

Distribución de los tiempos de reacción por participante en la prueba de caída de la regla



Nota. Se presentan diagramas de cajas y bigotes de los tiempos de reacción (en segundos) para cada participante. La línea central de cada caja indica la mediana, los bordes de la caja corresponden a los cuartiles (Q1 y Q3), y los bigotes representan la dispersión de los datos; los puntos individuales corresponden a mediciones experimentales y posibles valores atípicos.

Se observaron diferencias interindividuales significativas en términos de variabilidad y consistencia. Algunos estudiantes presentan medianas más elevadas y rangos intercuartílicos amplios, mientras que otros muestran distribuciones más compactas y valores centrales menores. La presencia de valores

atípicos en ciertos casos indica ensayos con tiempos de reacción significativamente mayores, compatibles con fluctuaciones propias de mediciones humanas repetidas.

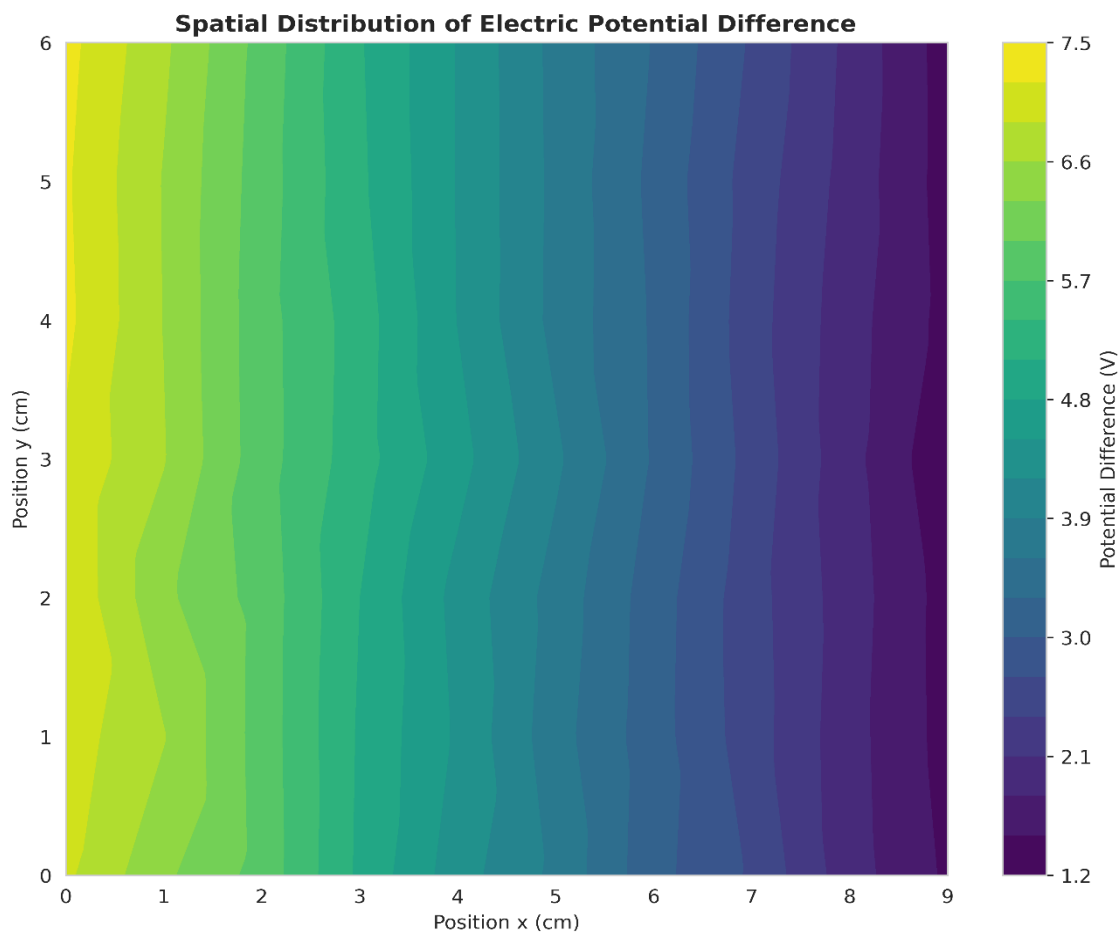
Estos resultados reflejan la variabilidad inherente a fenómenos biológicos y destacan la utilidad del análisis estadístico para caracterizar distribuciones no uniformes en contextos experimentales.

c) Cubeta de diferencias de potencial eléctrico

Las mediciones de voltaje en la cubeta de electrodos permitieron construir mapas de contorno que representan las líneas equipotenciales del sistema (Figura 6). El gradiente de potencial obtenido a partir de estos datos se utilizó para generar diagramas vectoriales del campo eléctrico (Figura 7).

Figura 6

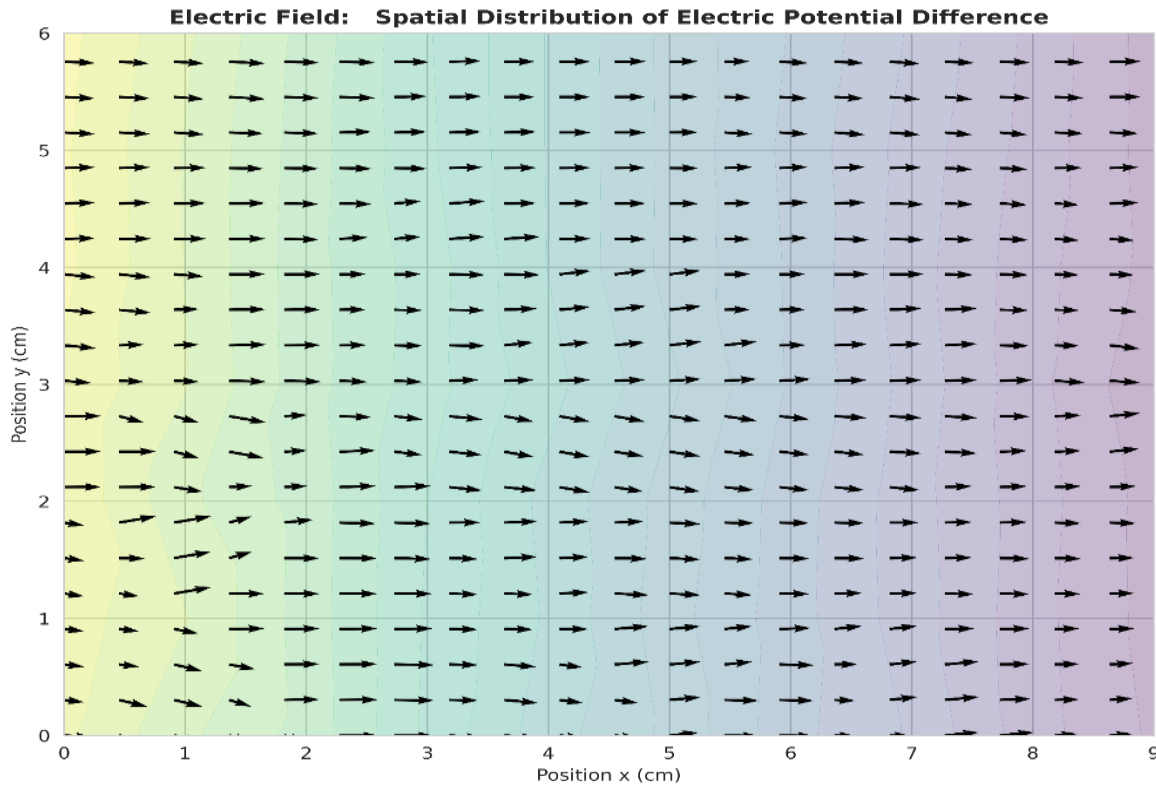
Distribución espacial del potencial eléctrico en una cubeta electrolítica



Nota. Se muestra un mapa de contorno del potencial eléctrico obtenido a partir de mediciones de voltaje en una cubeta electrolítica con electrodos planos. La escala de colores representa los valores de potencial eléctrico en voltios (V) en función de la posición espacial.

Figura 7

Distribución espacial del campo eléctrico obtenida a partir del gradiente del potencial eléctrico



Nota. Se presenta el diagrama vectorial del campo eléctrico calculado a partir del gradiente espacial del potencial eléctrico (véase Figura 6). Los vectores indican la dirección y la magnitud relativa del campo eléctrico en cada punto del sistema.

El mapa de contorno muestra un gradiente de potencial bien definido entre los electrodos, con valores que oscilan aproximadamente entre 1.7 V y 7.5 V. En la región central del sistema, las líneas equipotenciales presentan una disposición casi paralela, indicando un campo eléctrico aproximadamente uniforme, en concordancia con el modelo teórico de placas paralelas. Las desviaciones observadas en los bordes pueden atribuirse a efectos de borde y a asimetrías propias del montaje experimental.

El diagrama vectorial del campo eléctrico evidencia que los vectores apuntan desde regiones de mayor potencial hacia regiones de menor potencial, con magnitudes coherentes con el gradiente espacial del voltaje medido, validando el procedimiento de cálculo implementado en el programa.

El diagrama vectorial del campo eléctrico no evidencia que los vectores apuntan desde regiones de mayor potencial hacia regiones de menor potencial, con magnitudes coherentes con el gradiente espacial del voltaje medido, validando el procedimiento de cálculo implementado en el programa.

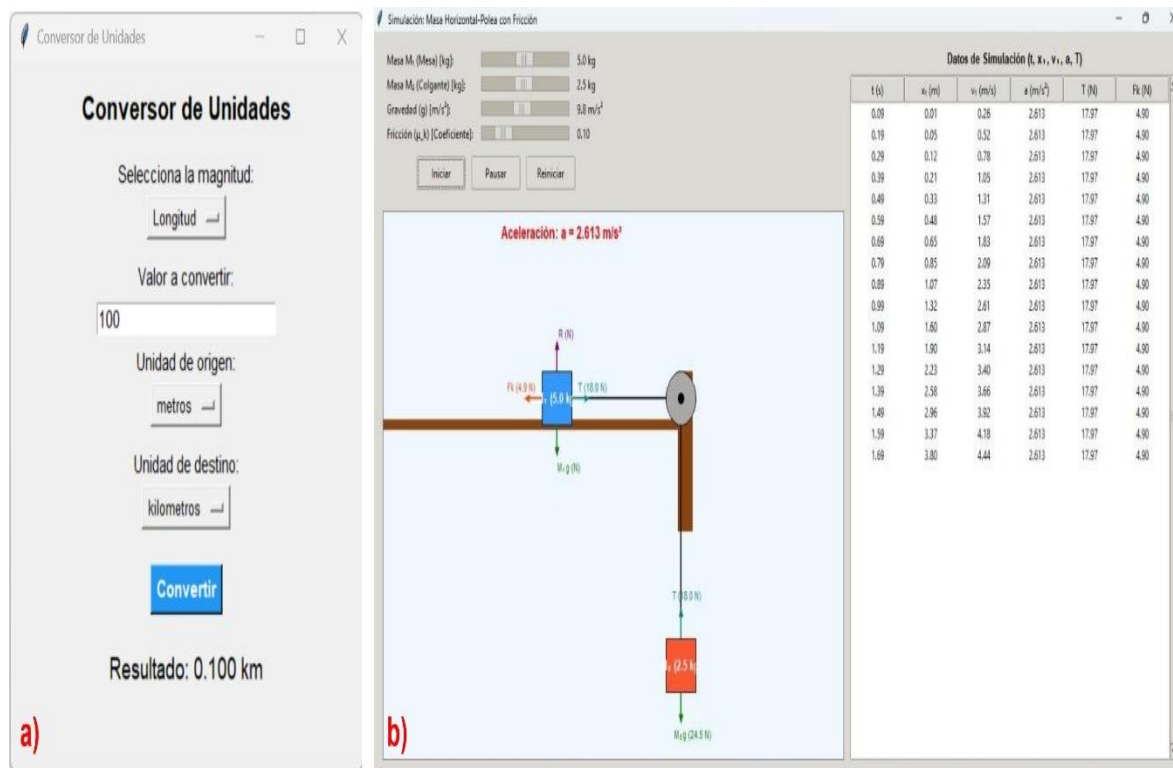
d) Simulaciones computacionales complementarias

Las actividades de conversión de magnitudes (Figura 8a) y simulación de sistemas cinemáticos y dinámicos (Figura 8b) se emplearon como ejercicios de apoyo didáctico, con el objetivo de familiarizar a los estudiantes con el uso del programa multifuncional en escenarios computacionales controlados. Estas actividades no se concibieron como experimentos físicos, sino como recursos pedagógicos orientados a reforzar la comprensión de los modelos teóricos y su traducción a procedimientos computacionales implementados en Python.

En este contexto, el análisis computacional permitió explorar relaciones funcionales entre variables físicas y contrastar modelos teóricos con resultados numéricos generados por simulación, fortaleciendo la articulación entre teoría, programación y análisis de datos antes de su aplicación en experiencias experimentales reales.

Figura 8

Actividades computacionales de apoyo didáctico: conversión de unidades y simulación del movimiento de dos cuerpos



Nota. (a) Interfaz de un conversor de unidades para magnitudes de longitud. (b) Simulación del movimiento de dos cuerpos unidos por una cuerda en un sistema con polea, incluyendo la visualización de fuerzas y variables dinámicas.

DISCUSIÓN

Los resultados obtenidos muestran que es técnicamente viable sustituir software comercial por herramientas abiertas basadas en Python en laboratorios universitarios de física, sin pérdida apreciable en la precisión del análisis ni en la calidad de las visualizaciones generadas. En las experiencias desarrolladas, el flujo de trabajo implementado permitió realizar regresiones, análisis estadísticos y representaciones gráficas complejas de manera consistente, lo que respalda el uso de soluciones abiertas como alternativa funcional a entornos propietarios ampliamente utilizados en el contexto académico.

Más allá de la equivalencia técnica, la integración de programación dentro del análisis experimental modifica el rol del software en el laboratorio, desplazándolo de una función meramente instrumental hacia un componente activo del proceso de análisis de datos (Bernal & Peña, 2023, p. 1; Hernández & Gómez, 2023, p. 11; Lincoln, 2019, p. 60). En este sentido, el uso de Python como entorno unificado para carga, procesamiento y visualización de datos favorece una mayor explicitación de los modelos matemáticos subyacentes y de las decisiones analíticas involucradas en el tratamiento de los datos experimentales (Fortune, 2021, p. 1; Gäßler et al., 2022, p. 1; Tufiño et al., 2024, p. 2). Lo anterior concuerda con enfoques que promueven la computación científica como parte esencial del quehacer experimental moderno (Caballero et al., 2012, p. 1; Fortune, 2021, p. 2; Lincoln, 2019, p. 60).

La incorporación de herramientas de inteligencia artificial generativa en el desarrollo del código permitió reducir barreras técnicas iniciales asociadas a la escritura y depuración de scripts, facilitando la adaptación del programa a distintas problemáticas experimentales. No obstante, el valor de esta integración no radica en la automatización del análisis, sino en su capacidad para apoyar la construcción y modificación consciente del código por parte del usuario, manteniendo la trazabilidad de los procesos computacionales y evitando el uso de soluciones opacas. Este enfoque coincide con estudios recientes que destacan el potencial de la IA como asistente en tareas de programación científica, más que como sustituto del razonamiento analítico (Hare, 2024, p. 3874; Freire-Avilés et al., 2024, p. 4713).

Desde una perspectiva formativa, la aplicación del flujo de trabajo propuesto en distintas experiencias experimentales permitió abordar de manera explícita aspectos como la variabilidad de los datos, la identificación de valores atípicos y la interpretación de parámetros estadísticos, elementos que suelen quedar relegados cuando se emplean herramientas cerradas. Si bien el diseño del estudio no permite establecer relaciones causales directas sobre el aprendizaje, los resultados sugieren que la integración de análisis computacional abierto favorece condiciones propicias para la discusión conceptual y la reflexión crítica sobre los modelos físicos, en línea con propuestas basadas en aprendizaje por indagación y resolución de problemas (Véliz & Gutiérrez, 2021, p. 150).

En este contexto, los resultados obtenidos se inscriben en transformaciones recientes en la enseñanza de la física mediada por inteligencia artificial, donde la visualización dinámica y el análisis computacional contribuyen a la comprensión de modelos abstractos (Cheng et al., 2024, p. 171). En el presente estudio, esto se evidenció en la capacidad del script para representar y analizar datos experimentales en distintos escenarios, facilitando su interpretación en tiempo real. No obstante, como advierte Cordero Monzón (2024, p.194), este tipo de integración tecnológica exige una actualización docente constante, particularmente en el diseño de estrategias que incorporen la inteligencia artificial de manera responsable y didácticamente intencionada.

La implementación de laboratorios basados en Python y asistencia de inteligencia artificial no solo responde a una modernización técnica, sino que se alinea con enfoques centrados en el aprendizaje autodirigido del estudiante. En este sentido, la posibilidad de diseñar scripts y personalizar el análisis de datos experimentales favorece un rol activo en la construcción del conocimiento, en contraste con el uso pasivo de software cerrado. Como plantea Martinetti, el entorno educativo contemporáneo demanda que los estudiantes desarrollen la capacidad de “controlar su propio proceso de aprendizaje, tanto en la universidad como en su futura profesión” (2020, p.1). Desde esta perspectiva, la manipulación directa del código y la toma de decisiones analíticas contribuyen al desarrollo de autonomía y al fortalecimiento de competencias técnicas en contextos experimentales.

La aplicabilidad del modelo en actividades experimentales evidencia su flexibilidad y escalabilidad. Esto refuerza la idea de que los entornos abiertos basados en Python pueden funcionar como plataformas transversales para el análisis de datos en física experimental, particularmente en carreras de perfil tecnológico donde la programación y el análisis computacional constituyen competencias relevantes para la formación profesional.

CONCLUSIONES

El uso de Python asistido por inteligencia artificial generativa demostró ser una alternativa abierta y funcional al software comercial en el análisis de datos experimentales en laboratorios universitarios de física. La implementación del programa multifuncional permitió automatizar tareas de procesamiento, generar visualizaciones de alta calidad y realizar análisis estadísticos consistentes, sin comprometer la precisión de los ajustes estadísticos ni la calidad de las visualizaciones obtenidas en las distintas experiencias experimentales.

Los resultados obtenidos confirman que la integración de programación, análisis estadístico y visualización en un entorno basado en Python facilita la construcción de flujos de trabajo reproducibles

y adaptables a diferentes tipos de datos experimentales. En este sentido, el enfoque propuesto se alinea con tendencias actuales en computación científica orientadas al uso de herramientas abiertas, transparentes y extensibles para el tratamiento de datos en contextos académicos.

El carácter modular del código desarrollado permite su reutilización, modificación y extensión según las necesidades del usuario, favoreciendo su aplicación en distintos contenidos de la física experimental. Asimismo, la incorporación de inteligencia artificial generativa como apoyo al desarrollo y ajuste del código reduce barreras técnicas iniciales sin ocultar los procesos computacionales involucrados, lo que refuerza la transparencia del análisis y su potencial formativo en términos de autonomía y desarrollo de competencias.

El estudio evidencia la aplicabilidad técnica de sustituir soluciones propietarias por herramientas abiertas basadas en Python en el contexto de laboratorio universitario. Futuras líneas de trabajo podrían centrarse en la evaluación comparativa del desempeño computacional del programa frente a otras herramientas existentes y en la extensión del flujo de trabajo a nuevos escenarios experimentales.

Declaración sobre el uso de inteligencia artificial generativa

Durante la preparación de este trabajo, los autores utilizaron herramientas de inteligencia artificial generativa con el propósito de asistir en la generación, depuración y optimización de código en Python y elaboración de diagramas ilustrativos, así como en la mejora de la claridad y legibilidad del texto. Tras el uso de estas herramientas, los autores revisaron, editaron y validaron críticamente todo el contenido generado, asumiendo plena responsabilidad por la integridad, exactitud y originalidad del manuscrito.

En algunos casos los autores no utilizaron herramientas de inteligencia artificial generativa con el propósito de asistir en la generación, depuración y optimización de código en Python y elaboración de diagramas ilustrativos, así como en la mejora de la claridad y legibilidad del texto. Tras el uso de estas herramientas, los autores revisaron, editaron y validaron críticamente todo el contenido generado, asumiendo plena responsabilidad por la integridad, exactitud y originalidad del manuscrito

REFERENCIAS BIBLIOGRAFICAS

Batista-Mendoza, J. B., Chung, E., Martínez-Soto, A., Fábrega-Polleri, J., & Fernández-Valdés, C. A.

(2025). An open Monte Carlo based implementation of Gauss's method for initial orbit

determination. *Tecnociencia*, 27(1), 8–25. <https://doi.org/10.48204/j.tecno.v27n1.a6633>

- Bernal, M., y Peña, L. K. (2023). Competencias Investigativas: Modelización con Python para la Enseñanza de Física Mecánica. *Góndola, Enseñanza y Aprendizaje de las Ciencias*, 18(Especial), 323–330. <https://doi.org/10.14483/23464712.21378>
- Bufasi, E., & Lakrad, K. (2019). Improving teaching techniques using visual Python: A case study in physics laboratories. *International Journal of Scientific & Technology Research*, 8(12), 1–6.
- Caballero, M. D., Kohlmyer, M. A., & Schatz, M. F. (2012). Implementing and assessing computational modeling in introductory mechanics. *Physical Review Special Topics—Physics Education Research*, 8(2), Artículo 020106. <https://doi.org/10.1103/PhysRevSTPER.8.020106>
- Campos, G., & Benarroch, A. (2024). Laboratorios virtuales para la enseñanza de las ciencias: Una revisión sistemática. *Enseñanza de las Ciencias*, 42(2), 109–129. <https://doi.org/10.5565/rev/ensciencias.6040>
- Cardona, M. E., & López, S. (2017). Una revisión de literatura sobre el uso de sistemas de adquisición de datos para la enseñanza de la física en la educación básica, media y en la formación de profesores. *Revista Brasileira de Ensino de Física*, 39(4), e4404. <https://doi.org/10.1590/1806-9126-RBEF-2016-0308>
- Chen Cheng, C., Chung, E., Correa, N., Martínez Soto A. ´., & Chen Cheng, A. (2024). La Revolución de la IA: Un Análisis Cualitativo de sus Implicaciones en la Física Conceptual. *REICIT*, 3(2), 170–182. <https://doi.org/10.48204/reict.v3n2.4687>
- Cordero Monzón, M. Á. (2024). Inteligencia Artificial en el aula: Oportunidades y desafíos para la didáctica de la matemática y física universitaria. *Revista Internacional de Pedagogía e Innovación Educativa*, 4(1), 193–207. <https://doi.org/10.51660/ripie.v4i1.151>
- Díaz Villavicencio, A. P., Lara Suárez, R., & Calderón Angulo, R. J. (2025). La inteligencia artificial y su uso efectivo en el aula. Caso de estudio Bachillerato. *REICIT: Revista Especializada de Ingeniería y Ciencias de la Tierra*, 5(1), 147–160. <https://doi.org/10.48204/reicit.v5n1.7686>

- Fernández-Marín, M. Á., Montero-Murillo, J. R., & González-Tolmo, D. (2025). Caso de estudio sobre simulación de datos mediante inteligencia artificial generativa y Google Colab. *Revista Mexicana de Investigación e Intervención Educativa*, 4(S1), 18–26.
- Fortune, N. A., Troach, R., Webster, R., et al. (2021). A short guide to using Python for data analysis in experimental physics. *Authorea*. <https://doi.org/10.22541/au.152961025.52816543/v2>
- Freire-Avilés, R. M., Muzzio-Arguello, J. M., Freire-Avilés, V. A., & Vélez-San Martín, B. O. (2024). Inteligencia artificial en la educación: Una revisión sistemática de la transformación de la enseñanza de Python mediante ChatGPT en la educación superior. *Journal Scientific MQRInvestigar*, 8(4), 4712–4738. <https://doi.org/10.56048/MQR20225.8.4.2024.4712-4738>
- Gäßler, J., Quast, G., Savoiu, D., & Verstege, C. (2022). kafe2: A modern tool for model fitting in physics lab courses (Technical report). *Karlsruhe Institute of Technology*. <https://doi.org/10.48550/arXiv.2210.12768>
- Gordon Graell, R. D. (2021). Entornos virtuales de educación universitaria en Panamá. Avances y deficiencias de la informática educativa enfrentando el reto de la pandemia. *Visión Antataura*, 5(2), 132–146.
- Gordon Graell, R. D. (2022). Inteligencia artificial: La caja de herramientas virtuales al servicio de la bioinformática. *Tecnociencia*, 24(2), 48–65.
- Hare, P. M. (2024). Coding with AI in the physical chemistry laboratory. *Journal of Chemical Education*, 101, 3869–3874. <https://doi.org/10.1021/acs.jchemed.4c00560>
- Hernández Flórez, J. D., y Gómez Vasco, C. A. (2023). Construyendo Python Notebooks en el Curso de Oscilaciones para la Formación de Profesores de Física. *Góndola, Enseñanza y Aprendizaje de las Ciencias*, 18(Especial), 249–259. <https://doi.org/10.14483/23464712.21371>

- Knapp, A., Horvat, D., Šaponja-Milutinović, D., & Kuić, D. (2025). Physics in Python course at Zagreb University of Applied Sciences. *International Journal of Innovative Research in Multidisciplinary Education*, 4(8), 739–749. <https://doi.org/10.58806/ijirme.2025.v4i8n01>
- Koerner, L. J., Caswell, T. A., Allan, D. B., & Campbell, S. I. (2020). A Python instrument control and data acquisition suite for reproducible research. *IEEE Transactions on Instrumentation and Measurement*, 69(4), 1698–1707. <https://doi.org/10.1109/TIM.2019.2914711>
- Lincoln, J. (2019). Using Python as a toolbox for student-directed virtual labs. *The Physics Teacher*, 57, 60. <https://doi.org/10.1119/1.5084937>
- Martinetti, A. (2020). Optimizing Student-Driven Learning (SdL) through a framework designed for tailoring personal student paths. *Education Sciences*, 10(9), Artículo 248. <https://doi.org/10.3390/educsci10090248>
- Monajem, M., & Felfer, P. (2022). APT_PyControl: An open-source Python atom probe tomography control software package. *Microscopy and Microanalysis*, 28, 732–734. <https://doi.org/10.1017/S1431927622003397>
- Muñoz Escudero, A. O. (2025). Aplicaciones de la Inteligencia Artificial en la Educación Superior: Estudio de Caso en la Facultad de Ingeniería, Universidad de Panamá, 2025. REICIT: Revista Especializada de Ingeniería y Ciencias de la Tierra, 5(1), 9-31. <https://doi.org/10.48204/reicit.v5n1.7675>
- Núñez, E. A., Baquero, E. M., Rengel, H., & Noboa, M. Y. (2024). El software de código abierto y el análisis estadístico en ciencia de datos. *ALCON*, 4(5), 179–188. <https://doi.org/10.62305/alcon.v4i5.361>
- Plaza, J., & Saquinaula, J. (2024). Efectividad del uso de Tracker en la enseñanza universitaria de física. *Reincisol*, 3(6), 6148–6168. [https://doi.org/10.59282/reincisol.V3\(6\)6148-6168](https://doi.org/10.59282/reincisol.V3(6)6148-6168)

- Stansbury, C., & Lanzara, A. (2023). AutodiDAQ: Scientific data acquisition software with analysis-in-the-loop. *Software*, 2(1), 121–132. <https://doi.org/10.3390/software2010005>
- Tufiño, E., Oss, S., & Alemani, M. (2024). Integrating Python data analysis in introductory physics laboratories. *European Journal of Physics*, 45(4), 045707. <https://doi.org/10.1088/1361-6404/ad4fcc>
- Véliz, M. I., & Gutiérrez, V. E. (2021). Modelos de enseñanza sobre buenas prácticas docentes en aulas virtuales. *Apertura*, 13(1), 150–165.
- Weber, S. J. (2021). PyMoDAQ: An open-source Python-based software for modular data acquisition. *Review of Scientific Instruments*, 92(4), 045104. <https://doi.org/10.1063/5.0032116>