

Capítulo 3

Aplicación de herramientas computacionales para el aprendizaje de conceptos termodinámicos

Lourdes Adriana Pérez Carrillo

Agustín Martínez Ruvalcaba

lourdes.pcarrillo@academicos.udg.mx

(Autor de correspondencia)

DOI: <https://doi.org/10.61728/AE26002040>



Resumen

Entre 2021 y 2025 se desarrolló una experiencia educativa en el marco del Programa Interinstitucional para el Fortalecimiento de la Investigación y el Posgrado del Pacífico (Programa Delfín), orientada a incorporar herramientas computacionales en la enseñanza de la termodinámica química. Participaron estudiantes de ingeniería química que, guiados por profesores investigadores, implementaron en Python ecuaciones de estado cúbicas para modelar el comportamiento de gases y fluidos no ideales. Se emplearon entornos interactivos de Jupyter Notebook y bibliotecas científicas de código abierto como NumPy, SciPy, CoolProp y Matplotlib, lo que permitió combinar programación, análisis de datos y visualización en una misma plataforma. La metodología se sustentó en el aprendizaje basado en proyectos y en el desarrollo del pensamiento computacional, promoviendo que los participantes planificaran, documentaran y validaran sus resultados con datos experimentales. Los resultados evidencian que la integración de la programación científica en la formación de ingenieros químicos favorece un aprendizaje activo, incrementa la comprensión de los sistemas termodinámicos y fortalece las competencias digitales y analíticas. La experiencia demuestra que el uso pedagógico de Python en contextos de ingeniería puede convertirse en una vía efectiva para vincular la docencia con la investigación aplicada y estimular la vocación científica.

Palabras clave: *Herramientas computacionales, termodinámica, Python, Verano Delfín, educación en ingeniería.*

1. Introducción

La enseñanza de la termodinámica en ingeniería química ha cambiado de forma sustancial en las últimas décadas. La incorporación de herramientas computacionales permite modelar y visualizar fenómenos complejos que antes requerían largos procedimientos manuales. Hoy, el uso de esas herramientas permite que los estudiantes observen, en segundos, cómo cambian la presión (P), el volumen (V) o la energía (E) de un sistema al modificar una variable. Este tipo de experiencias directas favorece la comprensión de los principios termodinámicos y su aplicación en problemas reales (Suthar y Joshipura, 2023; Saavedra-Mera et al., 2024; Ariza, 2023).

El aprendizaje activo apoyado en tecnología ofrece beneficios claros: fomenta la interacción con los contenidos, estimula el pensamiento crítico y desarrolla competencias digitales. Sin embargo, su adopción no ha sido uniforme. Persisten limitaciones como la falta de capacitación docente, la escasez de materiales didácticos contextualizados y la desigualdad en el acceso a equipos y software (Atabek, 2019; Falconer et al., 2014). Superar estas barreras resulta esencial para lograr un aprendizaje significativo y fortalecer la autonomía del estudiante, tanto en aulas presenciales como en entornos virtuales.

En este contexto, el presente capítulo documenta una serie de proyectos realizados entre 2021 y 2025 en el marco del Programa Delfín. Este programa, creado en 1995, busca despertar la vocación científica de los estudiantes de licenciatura mediante estancias de investigación con académicos de México y del extranjero. Los autores asesoran y supervisan a alumnos de ingeniería química que desarrollaron programas en Python para abordar problemas de termodinámica química, en particular, la resolución de ecuaciones de estado cúbicas.

Python es un lenguaje interpretado, de alto nivel y de propósito general, ampliamente utilizado en las ciencias e ingenierías. Su sintaxis clara, la potencia de sus bibliotecas (NumPy, SciPy y Matplotlib) y la disponibilidad de recursos abiertos facilitan su integración en la docencia (Oliphant, 2015; Python Software Foundation, 2025). En el aula, permite que los estudiantes implementen modelos, resuelvan ecuaciones no

lineales y visualicen los resultados de inmediato. Por ejemplo, pueden representar el comportamiento P–V de un gas real y comparar distintos modelos con pocos parámetros.

En los proyectos aquí descritos, los estudiantes programaron las ecuaciones de estado de Van der Waals, Redlich–Kwong y Peng–Robinson, utilizadas para describir los gases reales en condiciones no ideales (Smith et al., 2022). Además, implementaron bases de datos para automatizar la carga de parámetros termodinámicos, como la temperatura y la presión críticas, lo que redujo el tiempo de cálculo y minimizó los errores. Estas actividades reforzaron los conocimientos teóricos y desarrollaron destrezas prácticas en programación científica, resolución de problemas y análisis de resultados.

La integración de Python en la enseñanza de la termodinámica ha demostrado ser una estrategia pedagógica eficaz. Favorece el aprendizaje activo y fortalece habilidades transversales como el razonamiento lógico, la autogestión y la alfabetización digital (Jaramillo et al., 2024; Ariza, 2023). En el contexto de la ingeniería química, esta aproximación potencia la comprensión de conceptos como el equilibrio de fases, las propiedades de los gases reales y el diseño de procesos mediante ecuaciones de estado.

El proyecto constituye una experiencia de articulación entre la docencia y la investigación aplicada, coherente con los objetivos del Programa Delfín. Sus resultados evidencian un doble impacto: el fortalecimiento de las vocaciones científicas de los participantes y la creación de materiales computacionales que pueden incorporarse a la enseñanza universitaria de la termodinámica.

2. Metodología

La metodología se diseñó para integrar competencias computacionales y conocimientos termodinámicos en los estudiantes de ingeniería química. A continuación, se describen el enfoque adoptado, la organización del trabajo, las actividades realizadas y las estrategias de evaluación aplicadas.

2.1 Perfil y selección de los participantes

Se convocó a estudiantes de licenciatura interesados en la programación aplicada a la ingeniería química, con disponibilidad para participar en las estancias de investigación del Programa Interinstitucional para el Fortalecimiento de la Investigación y el Posgrado del Pacífico (Programa Delfín), y que cumplieran con todos los requisitos solicitados por el programa. Entre 2021 y 2025, participaron doce estudiantes bajo la tutoría de profesores-investigadores del Departamento de Ingeniería Química de la Universidad de Guadalajara.

El proceso de selección prioriza a quienes ya contaban con bases sólidas en termodinámica y motivación para emplear herramientas computacionales en su formación profesional. Este perfil permitió avanzar rápidamente hacia la aplicación práctica; en lugar de repasar los fundamentos, los estudiantes pudieron concentrarse en implementar modelos termodinámicos en Python y analizar resultados numéricos.

Por ejemplo, algunos participantes iniciaron el proyecto reproduciendo diagramas P–V–T a partir de ecuaciones de estado conocidas, mientras que otros desarrollaron rutinas para calcular factores de compresibilidad o estimar condiciones de equilibrio. Estos ejercicios sirvieron de punto de partida para fortalecer tanto la comprensión conceptual como las habilidades de programación científica.

2.2 Objetivos de aprendizaje y competencias meta

El proyecto se articuló en torno a objetivos claros y competencias observables. La meta general fue que el estudiantado implementara en Python ecuaciones de estado cúbicas usadas en termodinámica química. Van der Waals, Redlich–Kwong y Peng–Robinson para resolver problemas de gases y fluidos no ideales.

Metas específicas

- Introducir el uso de Python y de bibliotecas científicas (NumPy, SciPy, Matplotlib).

- Programar las ecuaciones de estado seleccionadas para calcular volumen molar y propiedades derivadas.
- Validar los resultados frente a datos experimentales o de referencia.
- Documentar los códigos generados (comentarios, guías de uso) y explorar su difusión a través de repositorios (GitHub u otros).

Este enfoque promueve un modelo de enseñanza por competencias, en el que el estudiante no solo “aprende conceptos”, sino que también desarrolla habilidades técnicas y cognitivas, así como una actitud reflexiva.

Competencias trabajadas

- Técnicas: modelado computacional, depuración, análisis de datos y visualización.
- Cognitivas: verificación, trazabilidad de los supuestos y lectura crítica de tablas.
- Actitud reflexiva: documentación, control de versiones y trabajo reproducible.

2.3 Diseño del proceso de enseñanza-aprendizaje

Se aplicó un ciclo de siete etapas compatible con un enfoque de aprendizaje basado en proyectos (ABP). El avance es incremental.

1. *Capacitación inicial*: Taller corto de Python: sintaxis, estructuras, funciones y NumPy/Matplotlib. Los productos fueron un notebook o un script con ejercicios resueltos, sin teoría redundante y con práctica inmediata.
2. *Implementación de ecuaciones de estado*: Codificación de las ecuaciones de estado mediante funciones puras y pruebas unitarias simples. Se obtendrán scripts con firmas claras, manejo de raíces y selección de la fase documentada.
3. *Validación de resultados*: Selección de sustancias de referencia y condiciones controladas. Comparación con tablas de referencia y reporte de errores absolutos y relativos. Si el error supera el umbral, se reabre la segunda etapa.

4. *Mejoras, visualización y documentación*: Se incorporaron rutinas gráficas (por ejemplo, relaciones Z vs. presión o volumen leído) y se redactaron guías de uso y comentarios explicativos en los códigos, con el fin de facilitar su reutilización y comprensión.
5. *Difusión y mantenimiento*: Publicación en un repositorio digital con control de versiones, con la intención de apoyar su uso en cursos posteriores y fomentar la colaboración.
6. *Iteración y retroalimentación*: El ciclo es iterativo. Fallas en la validación o nuevas ideas de visualización llevan de vuelta a la implementación o la documentación.

2.4 Recursos y entorno tecnológico

El proyecto se desarrolló con un conjunto de herramientas accesibles y de código abierto. Se utilizó Python 3.x en entornos interactivos de Jupyter Notebook, que permiten integrar código, texto y gráficos en un mismo documento. Esta estructura favorece el aprendizaje activo; el estudiante modifica una celda, ejecuta el código y observa de inmediato cómo cambia la curva de presión o el volumen molar.

Las bibliotecas principales fueron NumPy, SciPy y Matplotlib, pilares del ecosistema científico de Python. NumPy facilitó el manejo de arreglos y operaciones vectorizadas; SciPy, la resolución numérica de ecuaciones cúbicas; y Matplotlib, la generación de gráficos comparativos entre modelos y datos experimentales.

Para la validación de las propiedades termodinámicas, se recurrió a fuentes clásicas y verificables, como las de Smith et al. (2022) y Çengel et al. (2019). También se revisaron proyectos académicos que integran Python en la enseñanza de la termodinámica. Uno de ellos es Cool-Prop, un calculador termodinámico en Python, de código abierto y con trazabilidad. Otros ejemplos relevantes incluyen Cantera y PyroMat, bibliotecas dedicadas a estimar propiedades de fluidos reales mediante ecuaciones de estado.

Los participantes trabajaron con sus equipos personales conectados a internet. Esta condición promovió la autonomía y la flexibilidad, pero también exigió coordinación. Para ello se emplearon carpetas compartidas

y repositorios digitales en los que se discutían errores, se documentaban avances y se proponían mejoras de código. El entorno se consolidó como un laboratorio colaborativo de aprendizaje computacional.

2.5 Evaluación y retroalimentación

La evaluación combinó la precisión técnica con la reflexión conceptual. No se limitó a calificar los resultados, sino a medir la comprensión, la claridad y la capacidad de mejora.

- a) *Exactitud de los resultados*: Cada programa se probó con valores de referencia tomados de la literatura o de datos experimentales. Se establecieron tolerancias numéricas definidas al inicio del curso (por ejemplo, un error relativo menor al 2%). Cuando los valores excedían el límite, los estudiantes analizaban la fuente del error —redondeo, convergencia o elección de raíz— y documentaban la corrección.
- b) *Calidad del código*: Se evaluó la estructura modular, el uso de comentarios y nombres descriptivos, y la robustez ante errores (por ejemplo, condiciones de presión negativa o de temperatura fuera de rango). Los mejores códigos eran aquellos que podían ejecutarse sin intervención externa y con resultados reproducibles.
- c) *Capacidad explicativa*: Además de programar, los estudiantes debían justificar el sentido físico de los resultados. Un cálculo correcto, sin interpretación termodinámica, no se consideraba completo. Se pedía responder preguntas como: ¿Por qué el factor de compresibilidad disminuye al aumentar la presión? o ¿Qué parámetro domina el comportamiento del CO_2 cerca de su punto crítico?
- d) *Retroalimentación cualitativa*: Al finalizar cada verano, se aplicó un breve cuestionario de autoevaluación. Los estudiantes identificaron los principales retos y propusieron mejoras. Estas observaciones alimentaron la planeación del siguiente ciclo y fortalecieron el modelo iterativo del proyecto.

El sistema de evaluación combinó métricas cuantitativas (precisión y reproducibilidad) con criterios cualitativos (claridad, argumentación y documentación). Este equilibrio reflejó una evaluación formativa cen-

trada en competencias, orientada no solo al resultado final, sino también al proceso de construcción del conocimiento técnico y científico.

2.6 Fundamentos teóricos y enfoques metodológicos

La metodología adoptada se sustenta en dos marcos conceptuales complementarios: el aprendizaje basado en proyectos (ABP) y el desarrollo del pensamiento computacional.

2.6.1 Aprendizaje basado en proyectos (ABP)

El ABP plantea que los estudiantes aprenden al enfrentar un problema o proyecto auténtico y relevante para su disciplina. Este enfoque incrementa la motivación y la conexión entre la teoría y la práctica. En educación superior, su eficacia ha sido documentada en la formación de competencias investigativas, la resolución de problemas y el uso crítico de tecnologías (Nontol-Nontol y Leyva-Aguilar, 2024; Lévano et al., 2024).

En este proyecto, los estudiantes no siguieron una guía cerrada; debieron planificar, programar, validar y comunicar sus resultados. Cada decisión de modelado o de codificación formó parte del proceso de aprendizaje. El docente actuó como acompañante técnico y metodológico, no como una fuente única de conocimiento.

2.6.2 Desarrollo de competencias computacionales y pensamiento computacional

El segundo eje metodológico se relaciona con la capacidad de pensar mediante herramientas computacionales. No basta con aprender un lenguaje de programación; el objetivo es estructurar los problemas de ingeniería en términos de datos, algoritmos y validaciones reproducibles.

Este enfoque se vincula con marcos recientes (Adorni et al., 2024) que analizan cómo los problemas que requieren pensamiento computacional promueven competencias analíticas, de diseño y de comunicación técnica. En esta experiencia, las habilidades digitales no fueron un fin en sí, sino un medio para comprender mejor los fenómenos termodinámicos.

micos. Por ejemplo, al implementar la ecuación de Peng-Robinson, los estudiantes transformaron expresiones teóricas en funciones ejecutables, compararon los resultados y justificaron las diferencias numéricas desde la física del sistema.

El modelo curricular basado en competencias refuerza esta integración. Permite definir con precisión las habilidades esperadas, desde escribir código depurado hasta interpretar su resultado físico, y diseñar estrategias de evaluación coherentes con dichas metas.

2.7 Aplicación de encuesta diagnóstica y de percepción estudiantil

Como parte del diseño metodológico, se aplicó un instrumento adicional para evaluar la experiencia formativa de los estudiantes y recopilar evidencia sobre el impacto pedagógico del proyecto. Se diseñó un cuestionario estructurado de 46 ítems, distribuido en diez secciones temáticas, orientado a analizar aspectos como la motivación inicial, el aprendizaje disciplinar, el desarrollo de habilidades computacionales, la vocación científica y la percepción del uso de tecnologías digitales e inteligencia artificial durante la estancia de investigación.

El cuestionario fue elaborado por los responsables académicos del proyecto específicamente para este estudio, tomando como base la literatura sobre vocaciones científicas, pensamiento computacional y aprendizaje basado en proyectos. La versión íntegra del instrumento aplicado se encuentra disponible para consulta pública en un repositorio de GitHub, con el fin de favorecer la transparencia metodológica y la posible replicabilidad del estudio (<https://github.com/AgustinMartz/termodinamica-python/blob/main/Encuesta/Encuesta%20-%20Programa%20Delfin.pdf>).

La encuesta se implementó mediante un formulario digital en Google Forms y se envió a los estudiantes que participaron en los veranos de investigación del Programa Delfín. Su aplicación fue voluntaria y anónima, lo que garantizó la confidencialidad de la información conforme a la Ley de Protección de Datos Personales de la Universidad de Guadalajara. Las respuestas se analizaron de manera agregada y complementaron la información obtenida mediante la observación directa, los entregables de programación y las evaluaciones técnicas. Los resultados obtenidos

con este instrumento se presentan en la sección correspondiente, lo que permite una interpretación más amplia del impacto académico, tecnológico y vocacional del proyecto.

3. Resultados

Los resultados obtenidos durante el desarrollo del proyecto se organizan en tres dimensiones complementarias: (1) productos computacionales generados por los estudiantes mediante el lenguaje de programación Python, (2) reflexión pedagógica sobre las competencias digitales y científicas desarrolladas por los alumnos, y (3) resultados de la encuesta aplicada al finalizar el programa. La discusión que sigue integra estos elementos para ofrecer una lectura amplia del impacto académico y técnico del proyecto.

3.1 Productos computacionales y visualizaciones generadas por los estudiantes

A lo largo de las estancias, los participantes desarrollaron un conjunto de códigos, funciones y visualizaciones en Python que constituyen evidencia directa del aprendizaje alcanzado. Estos resultados se organizaron en tres etapas progresivas que abarcaron desde la implementación manual de ecuaciones de estado hasta el uso de bibliotecas avanzadas y la elaboración de notebooks interactivos con integración inicial de inteligencia artificial.

3.1.1 Implementación directa de ecuaciones de estado sin módulos externos

La primera etapa del proyecto consistió en la construcción manual, mediante Python, de las ecuaciones de estado de Van der Waals, Redlich–Kwong y Peng–Robinson sin recurrir a bibliotecas termodinámicas externas. Este enfoque buscó que los estudiantes comprendieran profundamente la estructura algebraica y física de los modelos, desde el cálculo de los parámetros críticos y los coeficientes termodinámicos

hasta la formulación del polinomio cúbico y la selección de raíces físicamente válidas.

Los programas generados incorporaron cálculos de presión, volumen molar y factor de compresibilidad para diversas sustancias, lo que permitió visualizar el comportamiento P–V–T en diferentes condiciones. Este trabajo evidenció logros importantes: el estudiantado adquirió dominio del manejo de ecuaciones cúbicas, del análisis de estabilidad mecánica, de la detección de raíces múltiples y de la interpretación del comportamiento termodinámico cercano al punto crítico o en regiones de no idealidad.

Al mismo tiempo, la implementación reveló algunas limitaciones inherentes a trabajar sin módulos especializados. En algunos casos, los polinomios cúbicos mostraron sensibilidad al redondeo, lo que evidencia la necesidad de métodos de solución más robustos y controles adicionales de estabilidad numérica. A pesar de estas restricciones, esta primera fase representó un avance significativo: permitió integrar la modelación matemática, la programación científica y el análisis termodinámico en un mismo proceso formativo, sentando las bases para las etapas subsecuentes del proyecto.

3.1.2 Integración de módulos termodinámicos y cálculo de mezclas gaseosas

En la segunda etapa, los estudiantes incorporaron los módulos Cantera, Pyromat y CoolProp para complementar y verificar los resultados obtenidos mediante sus propias rutinas. Esta transición permitió ampliar el alcance de los cálculos a un conjunto mayor de sustancias y explorar diferencias entre distintos enfoques termodinámicos. Cantera aportó una estructura basada en objetos y conceptos propios de sistemas reactivos; Pyromat facilitó el acceso transparente a propiedades tabuladas; y CoolProp ofreció precisión, estabilidad y una amplia base de datos termofísica sustentada en modelos multiparámetros.

El uso de estos módulos permitió validar los cálculos manuales, comparar tendencias entre ecuaciones de estado y automatizar barridos paramétricos con mayor rapidez. La confrontación entre resultados propios y los provenientes de librerías especializadas contribuyó a que los

estudiantes identificaran los límites de los modelos cúbicos sencillos y la importancia de emplear herramientas robustas al trabajar cerca del punto crítico o en condiciones altamente no ideales.

Un aporte destacado de esta etapa fue el desarrollo de programas para mezclas binarias y multicomponentes, basados en la regla de mezclado de van der Waals. Los estudiantes construyeron rutinas que calculaban parámetros combinados, resolvían el polinomio cúbico asociado y comparaban sus resultados con los obtenidos con CoolProp o, en algunos casos, con extensiones específicas de Cantera y Pyromat. Estos códigos permitieron analizar mezclas de hidrocarburos, mezclas con aire y sistemas personalizados definidos mediante composiciones molares arbitrarias.

Esta fase funcionó como un puente conceptual y técnico entre la programación manual y el uso profesional de bibliotecas termodinámicas, consolidando prácticas de validación, auditoría numérica y razonamiento comparativo entre modelos.

3.1.3 Desarrollo de notebooks interactivos con CoolProp e integración inicial de inteligencia artificial

La tercera etapa del proyecto se centró en la elaboración de notebooks interactivos en Jupyter programados en Python que integraran de forma sistemática las rutinas termodinámicas, las herramientas avanzadas de CoolProp y una primera aproximación al uso de la inteligencia artificial en el análisis de resultados. Los notebooks producidos abarcaron sustancias puras, mezclas binarias y sistemas multicomponentes, ofreciendo un entorno organizado para realizar cálculos, generar visualizaciones y documentar procesos de manera reproducible.

El uso de CoolProp en esta fase proporcionó ventajas claras: permitió evaluar propiedades con alta precisión, trabajar con fluidos industriales relevantes y realizar amplios barridos numéricos sin sacrificar eficiencia. Esto fortaleció la calidad de los análisis y facilitó la interacción entre los modelos termodinámicos clásicos y los datos obtenidos mediante ecuaciones multiparámetro de referencia.

Un aspecto innovador de esta etapa fue la integración experimental de un modelo de lenguaje mediante la API de OpenAI, con el propósito de explorar un prototipo de “calculadora explicativa”. La idea consistió

en combinar el cálculo automático realizado en las celdas de Jupyter con una capa interpretativa en la que la IA ofreciera explicaciones textuales sobre tendencias, comportamientos, discrepancias entre modelos o el significado físico de los resultados.

Aunque los resultados iniciales fueron prometedores, la integración aún no alcanzó el nivel de automatización deseado. La IA no podía leer directamente las variables, tablas o gráficas generadas por el notebook; para obtener explicaciones, los estudiantes debían copiar y pegar manualmente los valores en el prompt. Aun así, este primer intento permitió delinear la ruta hacia herramientas más integradas que combinen cálculo, visualización e interpretación automática. La perspectiva para el próximo ciclo consiste en avanzar hacia interfaces que permitan la interacción directa entre celdas computacionales y modelos de lenguaje, con el objetivo de construir una herramienta híbrida de cálculo-explicación que facilite el aprendizaje conceptual.

Esta tercera fase demostró el potencial pedagógico de unir programación científica, bibliotecas especializadas y herramientas de inteligencia artificial, lo que configura un entorno formativo alineado con las tendencias actuales en educación en ingeniería.

3.1.4 Disponibilidad y repositorio público del proyecto

Con el propósito de garantizar la reproducibilidad del trabajo realizado y facilitar que otras personas consulten, ejecuten o adapten los programas desarrollados, todo el material computacional elaborado en Python está disponible en un repositorio público en GitHub. El repositorio puede consultarse en la siguiente dirección: <https://github.com/AgustinMartz/termodinamica-python>.

En su estructura actual, el repositorio cuenta con una carpeta principal denominada Code_Documentation que reúne los códigos y ejemplos desarrollados durante el proyecto. Dentro de esta carpeta se incluyen, por un lado, los programas base de la primera etapa.

- 1 - Van_der_Waals_sus_puras.py
- 2 - Redlich_Kwong_sus_puras.py
- 3 - Peng_Robinson_sus_puras.py

Por otro lado, Code_Documentation contiene subcarpetas dedicadas a los módulos externos y a los desarrollos más avanzados que se realizaron en las etapas finales del proyecto.

- 4 - Cantera/: Ejemplos de uso de la biblioteca Cantera para el cálculo de propiedades.
- 5 - Pyromat/: Programas que emplean Pyromat como herramienta didáctica para obtener propiedades termodinámicas.
- 6 - CoolProp/: Códigos que utilizan CoolProp para la evaluación precisa de propiedades de sustancias puras y mezclas.
- 7 - Notebooks con IA/: Notebooks de Jupyter que integran el cálculo termodinámico con CoolProp y un primer experimento de análisis asistido por modelos de lenguaje.

Esta organización permite que cualquier lector con acceso a este capítulo localice con rapidez las implementaciones manuales, los ejemplos con módulos especializados y los notebooks interactivos. A partir del repositorio, es posible descargar los archivos, ejecutarlos en un entorno local o adaptarlos a plataformas como Google Colab para reproducir y ampliar los resultados presentados. Además de servir como archivo permanente del proyecto, el repositorio funciona como una plataforma de trabajo en evolución, donde podrán integrarse nuevas funciones, algoritmos, visualizaciones y herramientas de análisis en futuras iteraciones, lo que constituye un recurso pedagógico duradero para la enseñanza de la termodinámica computacional en ingeniería química.

3.2 Reflexión pedagógica: Competencias digitales y científicas

La integración de herramientas computacionales en el estudio de la termodinámica no solo enriqueció la comprensión conceptual de los estudiantes, sino que también transformó la naturaleza del aprendizaje. A lo largo del proyecto, la programación en Python, el uso de módulos especializados y la validación comparativa de modelos matemáticos permitieron que los participantes desarrollaran un conjunto sólido de competencias digitales, científicas y de pensamiento computacional que trascienden los contenidos curriculares tradicionales. Esta experiencia contribuyó a consolidar

una visión contemporánea de la práctica ingenieril, en la que el dominio conceptual se complementa con la capacidad de modelar, automatizar, analizar e interpretar fenómenos de manera reproducible y transparente.

3.2.1 Competencias digitales

La programación se convirtió en el medio por el cual los estudiantes interactuaron con los modelos termodinámicos. Más allá de ejecutar instrucciones, aprendieron a construir estructuras de código claras, documentadas y escalables. Esto implicó familiarizarse con librerías científicas ampliamente utilizadas en la industria y la investigación, lo que les permitió comprender que la ingeniería moderna se desarrolla en ecosistemas computacionales robustos y en constante evolución.

Asimismo, trabajar con repositorios públicos, notebooks y archivos organizados en carpetas temáticas fomentó prácticas de gestión digital alineadas con estándares profesionales: depuración sistemática, control de versiones, manejo de entornos y reutilización de componentes. Estas habilidades, aunque a menudo consideradas transversales, son esenciales para la formación de ingenieros capaces de integrarse en equipos multidisciplinarios, interpretar documentación técnica y colaborar en proyectos con múltiples participantes.

3.2.2 Competencias científicas

La dimensión científica del proyecto se fortaleció en la medida en que los estudiantes debían validar sus resultados, contrastar sus modelos con datos de referencia y analizar las discrepancias entre distintas ecuaciones de estado. El proceso de comparar implementaciones manuales con los valores generados por CoolProp o con correlaciones experimentales consolidó una comprensión más profunda del significado físico de cada parámetro, de las limitaciones inherentes a los modelos y de la importancia de realizar análisis de consistencia numérica.

El enfoque reproducible que ofrecen los notebooks de Jupyter permitió documentar cada paso del proceso, desde la construcción del polinomio cúbico hasta la elección de la raíz físicamente válida, pasando por barrios paramétricos, la generación de gráficas y la discusión de resultados.

Este proceso es crucial para la ciencia contemporánea, especialmente en un contexto en el que la apertura y la transparencia forman parte de las mejores prácticas académicas.

3.2.3 Pensamiento computacional: descomposición, algoritmos y análisis de errores

El trabajo requerido para implementar ecuaciones de estado desde cero, resolver polinomios cúbicos y construir rutinas para mezclas permitió que los estudiantes ejercitaran el pensamiento computacional en su sentido más amplio. Este tipo de pensamiento, fundamental en la ingeniería moderna, implica la capacidad de descomponer un problema complejo en componentes manejables, identificar patrones repetitivos susceptibles de automatización, diseñar algoritmos eficientes y ordenar los procesos según su relevancia física y computacional.

Un aspecto especialmente valioso fue el análisis de errores. En más de una ocasión, los estudiantes enfrentaron inestabilidades numéricas, raíces físicamente imposibles, sensibilidades al redondeo o discrepancias entre modelos. Superar estas situaciones requirió no solo corregir el código, sino también reflexionar sobre el comportamiento matemático de las ecuaciones, evaluar la pertinencia de cada método y comprender el papel de las condiciones iniciales, los parámetros críticos y la precisión de los datos. Estos episodios constituyen aprendizajes significativos porque exponen al estudiante al razonamiento científico auténtico: formular hipótesis, evaluar alternativas y justificar decisiones.

3.3 Aportaciones principales de la encuesta

Los resultados de la encuesta muestran un impacto formativo claramente favorable en las dimensiones académica, tecnológica y vocacional. En cuanto a las motivaciones iniciales, la mayoría de los participantes señaló un interés genuino por la ciencia como la razón principal para integrarse al Programa Delfín, mientras que un grupo menor buscaba fortalecer su experiencia profesional. Tras la estancia, la percepción sobre la investigación mejoró significativamente y los estudiantes manifestaron una comprensión más amplia del trabajo científico.

Desde el punto de vista académico, los datos son consistentes. Todos los participantes indicaron un fortalecimiento notable de su área disciplinar, así como una mejora significativa en su capacidad de análisis y resolución de problemas. De igual manera, todos los estudiantes reportaron haber aprendido nuevas herramientas tecnológicas, especialmente el uso de Python y sus bibliotecas científicas, lo cual se integró de manera natural en la resolución de problemas termodinámicos. La mayoría de los estudiantes también participó en la elaboración de productos científicos, lo que refuerza el componente comunicativo de la experiencia.

En términos de vocación científica, una proporción considerable señaló que la estancia reforzó su interés por la investigación, y muchos expresaron una mayor inclinación hacia estudios de posgrado o trayectorias académicas. Los participantes también destacaron el valor de comprender cómo la investigación contribuye a la resolución de problemas reales en contextos sociales e industriales.

Las relaciones de tutoría y colaboración fueron evaluadas muy positivamente: la mayoría describió la orientación recibida como excelente, y todos afirmaron haber aplicado posteriormente en sus cursos conocimientos adquiridos durante el proyecto. Asimismo, se reportó la creación de vínculos académicos y profesionales considerados valiosos para el futuro. En cuanto a la pertinencia tecnológica, aunque solo la mitad empleó herramientas de inteligencia artificial, quienes las utilizaron las consideraron muy útiles y recomendaron integrarlas en proyectos futuros.

Debido a restricciones de espacio, esta sección presenta únicamente una síntesis de los hallazgos. Una versión ampliada, que incluye una tabla con los resultados más representativos y una discusión detallada, así como el análisis completo de los resultados de la encuesta, se encuentra disponible en un documento anexo alojado en el repositorio público del proyecto en GitHub, el cual puede consultarse para mayor detalle en el sitio: [https://github.com/AgustinMartz/termodinamica-python/blob/main/Encuesta/Anexo%20\(encuesta\).pdf](https://github.com/AgustinMartz/termodinamica-python/blob/main/Encuesta/Anexo%20(encuesta).pdf).

4. Conclusiones

La combinación de competencias digitales, científicas y computacionales contribuyó a formar en los estudiantes una cultura de trabajo más cercana a la práctica profesional actual. En lugar de ver la programación como un requisito accesorio, comprendieron que es un lenguaje central para modelar fenómenos, comunicar resultados y automatizar procesos. Por su parte, los modelos de termodinámica dejaron de ser expresiones estáticas en un libro de texto para convertirse en objetos manipulables, verificables y sujetos a experimentación computacional.

Al mismo tiempo, el uso de un repositorio público en GitHub y la organización rigurosa del material fomentaron modelos de trabajo colaborativos y transparentes. La posibilidad de que otros estudiantes, profesores o equipos de investigación reutilicen o adapten estos programas abre la puerta a un aprendizaje comunitario y a la evolución del proyecto más allá del período de estancia.

La experiencia educativa documentada en este proyecto demuestra que la integración de herramientas computacionales no solo facilita la comprensión de conceptos termodinámicos complejos, sino que también amplía el horizonte formativo de los futuros ingenieros. La combinación de programación, validación científica y razonamiento algorítmico fortalece la autonomía del estudiante, su capacidad crítica y su seguridad al enfrentar problemas abiertos, propios de la práctica profesional actual.

Esta convergencia entre teoría, modelado y experimentación computacional constituye un espacio fértil para el desarrollo de competencias de alto nivel, indispensables en una ingeniería química que avanza hacia procesos más complejos, regulados y digitalizados. En este sentido, el proyecto no solo alcanzó sus objetivos académicos inmediatos, sino que también estableció bases sólidas para futuras innovaciones pedagógicas que integren cómputo, simulación avanzada e inteligencia artificial en la enseñanza de la termodinámica.

5. Referencias

- Adorni, G., Piatti, A., Bumbacher, E., Negrini, L., Mondada, F., Assaf, D., Mangili, F., & Gambardella, L. M. (2024). FADE-CTP: A framework for the analysis and design of educational computational thinking problems. *Tech. Know. Learn.*, 30, 1073-1148. <https://doi.org/10.1007/s10758-025-09833-x>
- Ariza, J. A. (2023). Bringing active learning, experimentation, and student-created videos in engineering: A study about teaching electronics and physical computing integrating online and mobile learning. *Comput. Appl. Eng. Educ.*, 31, 1723-1749. <https://doi.org/10.1002/cae.22673>
- Atabek, O. (2019). *Challenges in integrating technology into education*. Turkish Studies – Information Technologies and Applied Sciences, 14, 1-19. <https://doi.org/10.7827/TurkishStudies.14810>
- Çengel, Y. A., Boles, M. A., & Kanoğlu, M. (2019). *Termodinámica* (9.^a ed.). McGraw-Hill / Interamericana.
- Falconer, J. L., Nicodemus, G. D., Medlin, J. W., DeGrazia, J., & McDanel, K. P. (2014). A thermodynamics course package in OneNote. *Chemical Engineering Education*, 48, 209-214. <https://journals.flvc.org/cee/article/view/83411>
- Jaramillo, M. L., Andrade, M. C., & Caraguay, G. M. (2024). *Herramientas digitales para la enseñanza de la programación en el ámbito educativo*. Editorial Universitaria, Universidad Nacional de Loja.
- Lévano, S., Alfredo, M., Núñez, D., Froilán, E. (2024). Desarrollo de las competencias en programación por medio del aprendizaje basado en proyectos en los estudiantes de una universidad privada en Lima. *Igobernanza*, 7, 198-214. <https://doi.org/10.47865/igob.vol7.n26.2024.346>
- Nontol-Nontol, W. J., Leyva-Aguilar, N. A. (2024). Aprendizaje basado en proyectos para el desarrollo de la competencia investigativa en estudiantes de educación básica. *Revista Tecnológica-Educativa Docentes 2.0*, 17, 283-294. <https://doi.org/10.37843/rted.v17i1.475>
- Oliphant, T. E. (2015). Guide to NumPy (2.^a ed.). *CreateSpace Independent Publishing Platform*.

- Python Software Foundation. (2025). *Reference manual — Python 3 documentation*. <https://docs.python.org/3/reference/>
- Saavedra-Mera, K. A., Valverde-Medina, L. M., Caicedo-Perlaza, L. C., Puyol-Cortez, J. L. (2024). El estudio de la termodinámica química desde una perspectiva pedagógica. *Journal of Economic and Social Science Research*, 4, 89-104. <https://doi.org/10.55813/gaea/jessr/v4/n3/122>
- Smith, J. M., Van Ness, H. C., Abbott, M. M., Swihart, M. (2022). *Introduction to chemical engineering thermodynamics* (9.^a ed.). McGraw-Hill.
- Suthar, K. J., Joshipura, M. H. (2023). The integrative approach of learning chemical engineering thermodynamics by using simulation-bases exercises. *Education for Chemical Engineers*, 45, 122-129. <https://doi.org/10.1016/j.ece.2023.09.001>

