



FoDAMI

17, 18 y 19 de septiembre 2025

INTRODUCCIÓN TEMPRANA DE LA MECÁNICA COMPUTACIONAL EN CURSOS DE ELECTROSTÁTICA Y MECÁNICA ANALÍTICA

Víctor A. Bettachini¹*, Mariano A. Real² y Edgardo Palazzo³

¹Departamento de Ingeniería e Investigaciones Tecnológicas, Universidad Nacional de La Matanza (DIIT UNLaM), Florencio Varela 1903, B1754JEC San Justo, Buenos Aires, Argentina.

²Instituto de Calidad Industrial, Universidad Nacional de San Martín - Instituto Nacional de Tecnología Industrial (INCALIN UNSam-INTI), Av. General Paz 5445, B1684EFA Villa Martelli, Buenos Aires, Argentina.

³Departamento de Materias Básicas, Universidad Tecnológica Nacional - Facultad Regional Avellaneda (UTN FRA), Ramón Franco 5050, B1876ABY Villa Domínguez, Buenos Aires, Argentina

*Email: vbettachini@unlam.edu.ar

RESUMEN

Este trabajo presenta dos experiencias educativas que integran herramientas computacionales en la enseñanza de física para ingeniería, abordando desafíos comunes en entornos con recursos limitados.

La primera propuesta desarrolló un curso de mecánica analítica basado en Python, utilizando Jupyter notebooks y un enfoque de aula invertida. Los estudiantes resuelven ecuaciones de Euler-Lagrange mediante bibliotecas como SymPy y SciPy, centrándose en el modelado físico y evitando simplificaciones pedagógicas. El curso, disponible en GitHub, ha demostrado eficacia en grupos pequeños, aunque requiere adaptaciones para escalabilidad.

La segunda experiencia aplicó metodologías similares en electrostática, permitiendo a alumnos analizar distribuciones de carga continuas mediante cálculos numéricos en Google Colaboratory. Los resultados destacan una mayor comprensión conceptual sin depender de técnicas matemáticas avanzadas.

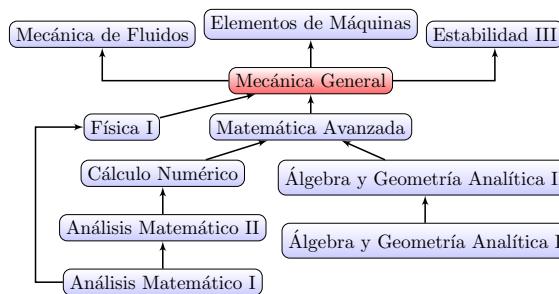
Ambos casos subrayan cómo la programación accesible (Python) y entornos interactivos (Jupyter) optimizan el tiempo en clase, priorizando el análisis sobre cálculos repetitivos, y ofrecen soluciones viables para instituciones con restricciones presupuestarias o logísticas. Estas estrategias, replicables y de código abierto, podrían extenderse a otras áreas de la física e ingeniería, potenciando el aprendizaje activo en contextos diversos.

Palabras claves: Enseñanza; Ingeniería Mecánica; Código; Inteligencia artificial.

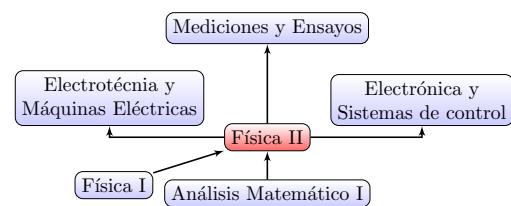
1. INTRODUCCIÓN

1.1 Las asignaturas de física, puente entre asignaturas básicas y específicas de la ingeniería mecánica

En las carreras de ingeniería mecánica las asignaturas de física integran lo dictado en las de matemática para proveer herramientas a aquellas específicas de la carrera. En una asignatura del tercer año de la Universidad Nacional de La Matanza (UNLaM) se dicta el formalismo Lagrangiano para el cálculo de esfuerzos y dinámica de sistemas de cuerpos, con vistas a aplicarle en las asignaturas destinadas a la mecánica de fluidos, el estudio de vibraciones y la dinámica de elementos de máquinas (Figura 1a).



(a) La asignatura de mecánica analítica de la UNLaM oficia de umbral hacia las específicas de la carrera.



(b) En UTN FRA la primera asignatura con temática de electromagnetismo abre el camino a otras en que esta se aplica a dispositivos industriales de electrotecnia y electrónica. Se muestran solo las correlativas con aplicación inmediata de conceptos de electromagnetismo.

Figura 1: En las carreras de ingeniería mecánica las asignaturas de física ofician de puente entre las asignaturas básicas y aquellas con la temática propia de la carrera.

En la Facultad Regional Avellaneda de la Universidad Tecnológica Nacional (UTN FRA), en una asignatura aún más temprana (segundo año), se dictan los rudimentos de electromagnetismo que se aplican en las asignaturas inmediatamente correlativas que tratan ensayos y mediciones en la industria, así como los fundamentos del funcionamiento de dispositivos tanto electrotécnicos como de electrónica industrial (Figura 1b).

1.2 La mecánica computacional como herramienta para evitar saltos de complejidad entre asignaturas

Usualmente, las modelizaciones que se trabajan en las asignaturas de física son “de juguete” y difieren sustancialmente de dispositivos de aplicación industrial reales. Como evidencia la Figura 1, estos son el objeto de las siguientes asignaturas que cursan los alumnos, lo que lleva a un “salto de complejidad”.

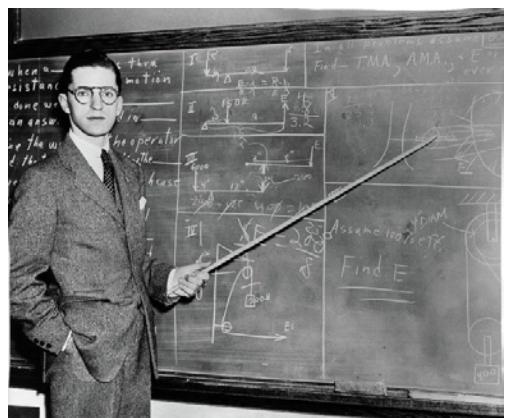
Hay variadas limitaciones para que en clases de las asignaturas de física se modelen sistemas físicos realistas. Puede que su complejidad requiera obtener analíticamente extensos sistemas de ecuaciones integro-diferenciales, tal es el caso del curso de mecánica analítica de la UNLaM, lo que demandaría en papel o pizarrón un tiempo que excede el de la clase. O bien puede que los alumnos no hayan cursado aún las asignaturas de matemática para resolver analíticamente tales ecuaciones, como sucede en la asignatura de electromagnetismo de la UTN FRA.

Ambas falencias pueden suplirse introduciendo mecánica computacional en las asignaturas. Esta disciplina conjuga la matemática, las distintas ramas de la física y herramientas de ciencias de la computación para explicar la

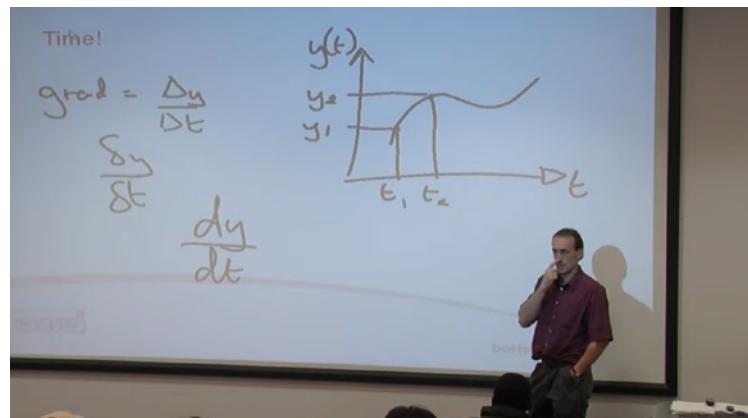
dinámica de sistemas físicos de diversa composición ante muy variados forzantes externos. Entendemos que una exitosa explotación de esta herramienta en las primeras asignaturas de la carrera de ingeniería mecánica requiere se cumplan dos condiciones. La primera, remover tanto al pizarrón y papel de su rol como herramientas centrales de expresión. Y la segunda, no depender de un software específico que limite la posibilidad de adaptar las soluciones y utilizar el código como una herramienta de trabajo.

1.3 Pizarrón y papel, foco de las clases en la tercera década del siglo XXI

En la tercera década del siglo XXI los cursos de ingeniería de las universidades latinoamericanas tienen aún como herramientas centrales el papel y el pizarrón. Introducido en el siglo XIX, este último siguió siendo el foco del aula en los siglos subsiguientes. La Figura 2a podría corresponder a un aula contemporánea. Desde entonces la metodología de la clase se resumen en que el docente presenta un tema con anotaciones en el pizarrón, los alumnos toman apuntes en papel, luego realizan la ejercitación, que el docente puede luego corregir en el mismo soporte.



(a) Pizarrón de una clase de física, c. 1930.



(b) Proyección de una clase de análisis matemático, c. 2010.

Figura 2: En el s. XXI se hace un uso del pizarrón idéntico a cuando se le introdujo en el aula del s. XIX. Si está presente en el aula, la computadora es usualmente un mero substituto de este último.

En las últimas décadas del siglo XX se generalizó el uso de computadoras en el aula, aunque los docentes usualmente les reducen a ser una mera herramienta de proyección, que como muestra la Figura 2b, excluyendo a los alumnos de su uso. Y ellos, aun cuando realicen sus anotaciones en computadoras portátiles, reproducen aún el mismo esquema del s. XIX, invirtiendo buena parte de su tiempo y atención en transcribir lo mostrado en un pizarrón. Esta sub explotación en el aula de la inherente capacidad de rápida reproducción de la información digital queda aún más fuera del contexto actual ante la omnipresencia de grandes modelos de lenguaje (LLM) como ChatGPT o DeepSeek, que permiten a los estudiantes obtener una rápida orientación sobre la temática que se está tratando. Los formatos papel y pizarrones fuerzan a una transcripción desde y hacia el medio digital para poder hacer provecho de los LLM. Tal transcripción de sus anotaciones en papel al teléfono o computadora conectadas a internet, sumada al arcaico proceder de tomar apuntes en clase, es sumar una tarea adicional que distrae al alumno del pensamiento creativo que debiera ser el foco de su esfuerzo.

Por estas razones es que en los cursos aquí presentados se busca utilizar el mismo formato digital para el dictado de lecciones por parte del docente, así como para la ejercitación por parte de los alumnos. El formato digital es capaz de representar todo el contenido gráfico que puede mostrarse en un pizarrón, incluyendo las ecuaciones integro-diferenciales que modelan cualquier sistema físico a analizar, si no que también permite resolver estas últimas. Esto abre el camino al uso de representaciones gráficas de resultados precisos obtenidos por métodos del cálculo numérico.

1.4 El código como herramienta central de la asignatura

Puede resultar tentador inclinarse por algún paquete de software específico especializado como Comsol Multiphysics, Ansys o SolidWorks. O, por el contrario, utilizar un paquete de software más específico de matemática computacional, como Mathematica o Matlab. Ante este abanico de opciones, no hay que olvidar que las asignaturas de física se ubican al inicio de las carreras de ingeniería mecánica, y el alumno en tal estadio no tiene aún la formación suficiente para comprender en detalle las herramientas de modelización y simulación que ofrecen este tipo de software.

Entendemos que para que la propuesta pueda ser transversal a las temáticas de las múltiples asignaturas correlativas, es superador utilizar una herramienta aún más general. Con este supuesto es que se optó por hacer uso de Python, un lenguaje de programación de alto nivel, que además resulta ser abierto y libre, virtudes acordes a las limitaciones de las aulas de nuestras universidades.

Para que la implementación de Python sea vista como una continuación de lo aprendido en las asignaturas de matemática, este debe operar el álgebra y análisis simbólico con la misma nomenclatura matemática que se utiliza en el pizarrón. Para tal fin se hace uso de la biblioteca SymPy [1], que al operar con álgebra simbólica utilizando la nomenclatura matemática habitual, habilita al docente a intercalar en sus explicaciones cálculos realizados por el código y que resulten familiares y sencillas de operar para el alumno.

Para capitalizar lo aprendido en asignaturas de cálculo numérico, con el fin de obtener soluciones de las complejas ecuaciones integro-diferenciales generadas con auxilio de SymPy, se recurre a la biblioteca SciPy [2]. Tales resultados se representan gráficamente haciendo uso de la biblioteca Matplotlib [3].

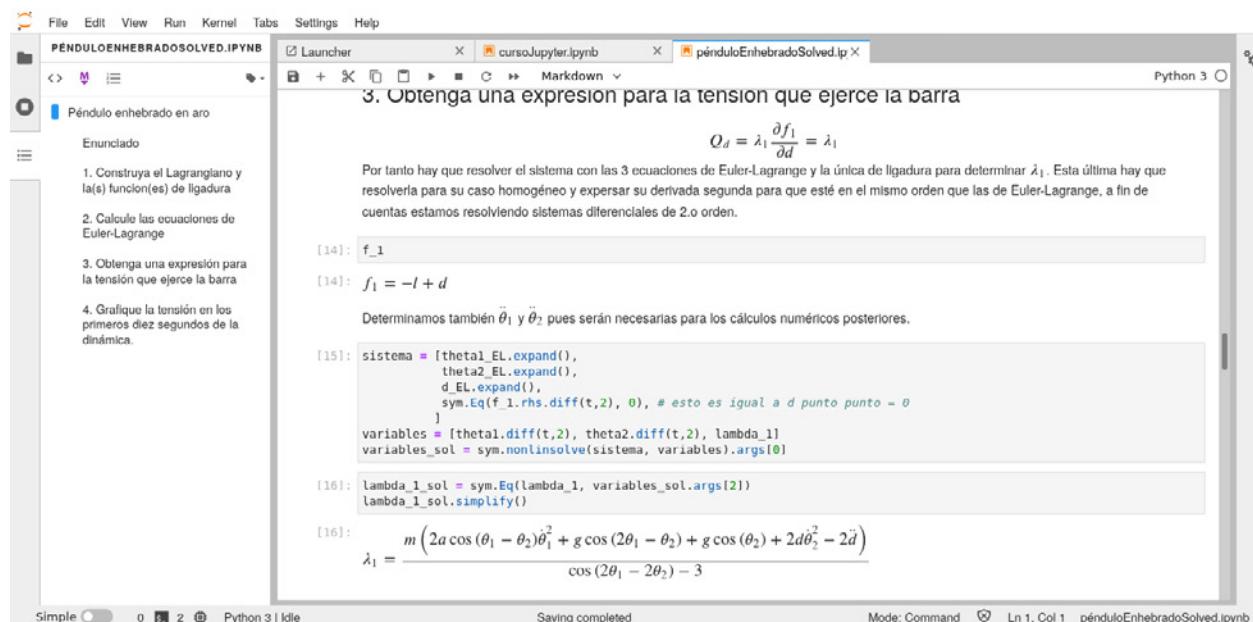
1.5 Usar código para hacer menos tediosa la matemática no requiere de saber programar

Es importante recalcar que aprovechar código para estas tareas en una asignatura no implica enseñar a programar. En ninguno de los casos reportados en este trabajo se busca que el alumno aprenda a modelar algorítmicamente teniendo en cuenta cuestiones de la arquitectura informática de los sistemas que utiliza. Por el contrario, solo se requiere que plasmen soluciones a la serie de problemas a través de adaptar código escrito por los docentes, algo que no es más que una mera reutilización de este código.

El eje de la propuesta es despreocupar a los alumnos del trabajo matemático para centrarlos en los nuevos conceptos propios de la asignatura. De igual manera que un alumno de secundaria resuelve aritmética con una calculadora de bolsillo tras haberla aprendido en la primaria, nuestros alumnos automatizan lo aprendido en asignaturas anteriores. Así, extendemos el uso sistemático de cálculos simples a la resolución de problemas complejos, que requerirían de cálculos integro-diferenciales que aún no dominan, evitando la posible frustración de un trabajo tan monótono.

y que en numerosas ocasiones no puede ser concluido debido a las dificultades matemáticas [4]. Esto permite resolver más problemas, con retroalimentación inmediata, que se traduce en más práctica para la incorporación de conceptos, trabajando en forma autónoma, reutilizando y adaptando código para experimentar con problemas cada vez más complejos, atesorando así una herramienta que podrán aprovechar en otros contextos.

El formato digital seleccionado no pone en preeminencia el código sobre otro material, como el texto, ecuaciones o gráficos generados en forma externa. Dicho formato son los cuadernos Jupyter, como el que muestra la Figura 3.



The screenshot shows a Jupyter Notebook interface with the following content:

3. Obtenga una expresión para la tensión que ejerce la barra

$$Q_d = \lambda_1 \frac{\partial f_1}{\partial d} = \lambda_1$$

Por tanto hay que resolver el sistema con las 3 ecuaciones de Euler-Lagrange y la única de ligadura para determinar λ_1 . Esta última hay que resolverla para su caso homogéneo y expresar su derivada segunda para que esté en el mismo orden que las de Euler-Lagrange, a fin de cuentas estamos resolviendo sistemas diferenciales de 2.o orden.

```

[14]: f_1
[14]: f_1 = -l + d

Determinaremos también  $\dot{\theta}_1$  y  $\ddot{\theta}_2$  pues serán necesarias para los cálculos numéricos posteriores.

[15]: sistema = [thetal_EL.expand(),
               theta2_EL.expand(),
               d_EL.expand(),
               sym.Eq(f_1.rhs.diff(t,2), 0), # esto es igual a d punto punto = 0
               ]
variables = [thetal.diff(t,2), theta2.diff(t,2), lambda_1]
variables_sol = sym.nonlinsolve(sistema, variables).args[0]

[16]: lambda_1_sol = sym.Eq(lambda_1, variables_sol.args[2])
lambda_1_sol.simplify()

```

$$\lambda_1 = \frac{m \left(2a \cos(\theta_1 - \theta_2) \dot{\theta}_1^2 + g \cos(2\theta_1 - \theta_2) + g \cos(\theta_2) + 2d \dot{\theta}_2^2 - 2d \right)}{\cos(2\theta_1 - 2\theta_2) - 3}$$

Figura 3: El docente presenta el tema y los ejemplos en un cuaderno Jupyter, y el alumno puede resolver los ejercicios en una copia de ese cuaderno. Tales cuadernos Jupyter permiten intercalar texto, ecuaciones y gráficos generados manualmente con aquellos producidos por el código Python con una nomenclatura matemática estandarizada y gráficos claros.

El alumno copia el cuaderno Jupyter provisto por el docente, que contiene no solo el contenido teórico de la lección, sino también el código Python que resuelve un problema de ejemplo. Modificando ligeramente dicho código puede resolver la ejercitación propuesta en la guía práctica. Siendo la copia digital e inmediata, la transcripción se elimina y el alumno centra su esfuerzo durante la clase a la resolución de los mencionados ejercicios en cuadernos Jupyter individuales. Estos pueden corregirse por el docente en el mismo formato, manteniendo las expresiones matemáticas y gráficos estandarizados e inteligibles.

1.6 Los recursos están: plataformas en línea

Se ha argumentado que las universidades públicas latinoamericanas enfrentando restricciones simultáneas de presupuestos ajustados y la necesidad de acomodar los horarios de sus clases a estudiantes que trabajan de día a horarios

vespertinos. Además, no cuentan con recursos informáticos para destinar a cursos que no estén directamente relacionados con la informática o la programación [5]. Sin embargo, la experiencia del confinamiento por la pandemia de COVID-19 ha demostrado que es posible utilizar plataformas en línea para la enseñanza de asignaturas que no son informáticas. El uso de tales plataformas solo requieren de una conexión a internet. El alumno accede a las mismas utilizando una computadora personal, sea en su hogar, en su trabajo o una biblioteca. Incluso, en uno de los cursos reportados en este trabajo, el alumnado ha hecho uso de teléfonos inteligentes, no solo para acceder a los cuadernos Jupyter, sino también para editarlos con la finalidad de resolver los ejercicios.

Existen varias plataformas en línea, y de uso gratuito, que permiten la edición de cuadernos Jupyter y ejecutar código Python en ellos, como Google Colaboratory, GitHub Codespaces, Kaggle o CoCalc, entre otras. Una de las funcionalidades de estas plataformas es que permiten compartir el cuaderno Jupyter con otros usuarios, lo que fomenta entre los alumnos al trabajo colaborativo a distancia, una competencia fundamental en el ámbito laboral actual. Lo que es tal vez más importante, el docente puede comentar y corregir cuadernos de los alumnos a distancia y en tiempo real, habilitando consultas asincrónicas sobre los ejercicios, y no solo sincrónico como en el caso de las clases presenciales. En estas plataformas los cuadernos y la resolución de los ejercicios de cada alumno quedan a resguardo en su cuenta personal. Pueden así servirles de referencia futura para aplicarse en asignaturas posteriores o incluso en su vida profesional.

2. CURSO DE MECÁNICA ANALÍTICA

2.1 Trabajando con dispositivos mecánicos realistas

La complejidad de la matemática para modelizar sistemas crece rápidamente a medida que se contemplan más factores para hacer del modelo más fiel a los elementos que se presentan el sistema físico real. Una vez esquematizado el sistema físico, como muestra la Figura 4a, se debe calcular la energía potencial y cinética del sistema, así como los momentos de inercia de los cuerpos que lo componen. A partir de estos, generar con el formalismo Lagrangiano las ecuaciones para la dinámica y despejar las aceleraciones es una tarea trivial para las funciones de la biblioteca de álgebra simbólica SymPy, como ilustra la Figura 4b.



Figura 4: Estadios de la resolución de un problema por parte de un alumno.

Los sistemas de ecuaciones diferenciales generados se resuelven numéricamente con funciones de la biblioteca SciPy. La dinámica del sistema puede luego graficarse usando la biblioteca matplotlib (Figura 4c).

La mayor virtud de hacer uso de mecánica computacional en un curso de mecánica analítica es que la complejidad

de la matemática requerida para calcular esfuerzos y la dinámica a partir de un modelo industrial realista no está fuera del alcance de lo realizable en una clase. Ejemplo de esto es el cálculo de torques y fuerzas lineales que actúan sobre un brazo robótico, como el que se muestra en la Figura 5.

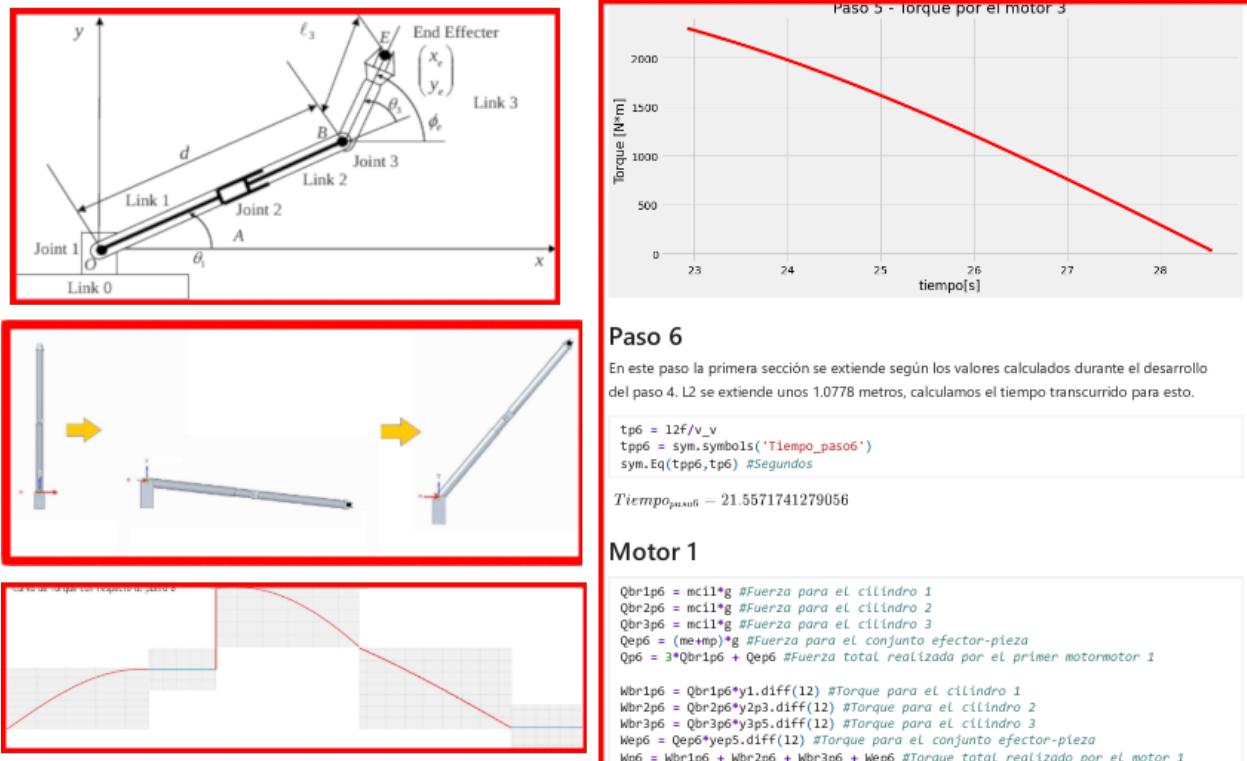


Figura 5: Ejemplo de un dispositivo mecánico realista que se analiza en el curso de mecánica analítica.

2.2 Resolver ejercicios con código es más compatible con la inteligencia artificial de los LLM

El uso de inteligencia artificial en el ámbito educativo es algo inevitable en el presente, como lo es en el ámbito profesional de la ingeniería. Las interfaces de conversación (chat) de grandes modelos de lenguaje en línea como ChatGPT de OpenAI, DeepSeek, GitHub Copilot o Google Gemini son herramientas a las que recurren los alumnos para resolver problemas de física, quieran los docentes o no que las utilicen. El procedimiento que sigue un alumno de un curso tradicional involucra transcripciones desde y hacia el LLM. En manera imperfecta escribe su avance en la resolución, o en el peor de los casos el enunciado completo, y nuevamente debe hacer lo propio al papel a partir de las respuestas que obtiene del LLM. No queda registro público de la interacción alumno-LLM, pues esto queda como un asunto privado del primero.

En el curso de la UNLaM se hace explícita la sugerencia de apoyarse en los LLM para la resolución de los ejercicios al tiempo que se transmite la virtud de guardar registro de consultas (prompts) al LLM adjunto a cada sección correspondiente del código. La resolución de problemas a través de código evita transcripciones equívocas, puesto

que el LLM interpreta con más fidelidad el código que el texto escrito en lenguaje natural tratando de explicar ambigüedades del problema. Los alumnos enfrentan así su solución asistiendo de consultas al LLM que en contrapartida sugiere correcciones a su código, como muestra el ejemplo de la Figura 6.



The screenshot shows a Jupyter notebook in Google Colaboratory. The code cell contains the following Python code:

```

AbelendaC G02E02.ipynb
File Edit View Insert Runtime Tools Help
Share Gemini Connect
Q Commands + Code + Text
In [1]: V = -g * (l1*m1 * cos(phi1) + l1*m2 * cos(phi1) + l2*m2 * cos(phi2))
B)
[ ] # prompt: declarar la variable phi
phi = me.dynamicsymbols('phi')
[ ] # prompt: establezca m1=0, phi1=phi2=phi y l1=l2=l/2 a traves de la funcion de sustitucion de SymPy. verifique que se obtiene T y V de un unico pendulo simple ideal
# Sustituciones
sustituciones = {m1: 0, phi1: phi, phi2: phi, l1: l/2, l2: l/2}
T_sustituido = T.subs(sustituciones)
V_sustituido = V.subs(sustituciones)

[ ] T_sustituido
Out[1]: 
$$\frac{\ell^2 m_2 \dot{\phi}^2}{2}$$

[ ] V_sustituido
Out[2]: 
$$-l g m_2 \cos(\phi)$$


```

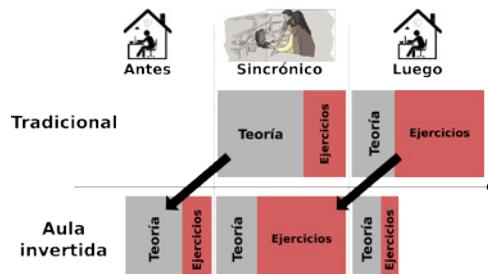
Figura 6: El alumno puede refinar su consulta (prompt) para cada bloque de código, aproximándose a la solución que busca. Aquí dejó constancia como comentario en el bloque de código del prompt que realizó al LLM Gemini de Google dentro de su cuaderno Jupyter ejecutado en el entorno Google Colaboratory.

Cabe acotar que los docentes paulatinamente nos tornamos en orientadores sobre como guardar un sentido crítico hacia los resultados que arrojan las consultas a los LLM. Se tuvieron cortas conversaciones con cada alumno sobre qué malinterpretaciones podrían estar sucediendo en ambos sentidos de su intercambio con el LLM.

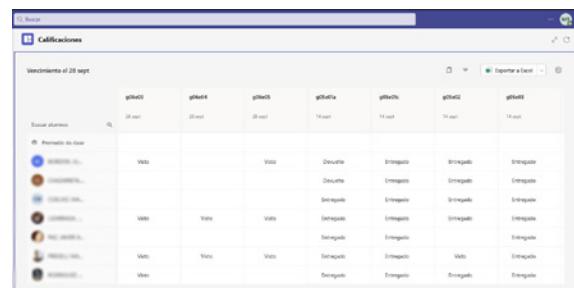
2.3 Aula invertida: el docente debe ayudar con los ejercicios, no está para dar conferencias

Una fuerte limitación que se enfrenta en la UNLaM es el limitado tiempo de clase. Los alumnos de ingeniería mecánica en el estadio de cursar mecánica analítica, suelen tener empleos de jornada completa. Las escasas horas dentro de la semana se reparten en las diversas asignaturas, haciendo que haya poco tiempo para un contacto sincrónico con el docente. El curso abordó esta problemática creando un entorno de aprendizaje en línea y asíncrono, que permite a los alumnos estudiar a su propio ritmo mediante el enfoque de aula invertida (flipped classroom) [6] que ilustra la Figura 7a.

Antes de las reuniones semanales, los alumnos deben estudiar la teoría y los ejemplos proporcionados en los cuadernos Jupyter, así como iniciar la resolución de los ejercicios que los acompañan. Durante esas reuniones vespertinas, ya sean en línea o presenciales, se anima a los alumnos a que formulen preguntas y comenten con el profesorado los problemas que no hayan podido resolver. Fuera de tal horario se hace uso de la plataforma Microsoft Teams para que los alumnos puedan realizar consultas asíncronas al docente. El docente accede a los cuadernos en Google Colaboratory, en el que puede realizar comentarios y correcciones directamente en la celda de código en cuestión donde corresponde hacerlos. En Teams se hace un seguimiento de los ejercicios que los alumnos



(a) El alumno lee e intenta aplicar nueva teoría antes del encuentro sincrónico con el docente. Este se reserva a resolver dudas y discutir los problemas que no se han podido resolver.



(b) A través del sistema Microsoft Teams se lleva un control de entregas de todos los ejercicios de las guías y de su estado de corrección. A través del mismo los alumnos realizan consultas asincrónicas.

Figura 7: La metodología de aula invertida se apoya en sistemas en línea.

van entregando como resueltos en el formato de un enlace a sus copias personales de cuadernos de Jupyter, como ilustra la Figura 7b. Todos los ejercicios de las guías deben entregarse para la corrección, una tarea que está al alcance del tiempo disponible por el docente que está centrado en esta tarea, y en responder consultas, en vez de repetir todos los cuatrimestres idéntico soliloquio frente a un pizarrón. La evaluación del aprendizaje parte de los registros de entrega de los mencionados ejercicios y la conversación que se tiene con los alumnos. Unos pocos minutos de preguntas sobre cómo implementaron una parte del código para resolver una parte específica del problema es tan o más reveladora sobre cuánto aprendieron que un examen escrito.

3. CURSO DE INTRODUCCIÓN A LA ELECTROSTÁTICA

Las actividades descritas en esta sección, que se están desarrollando en un curso de Física 2 de la Universidad Tecnológica Nacional - Facultad Regional Avellaneda (UTN FRA), están fuertemente inspiradas en el curso de mecánica analítica presentado en la sección anterior [7]. El material, disponible para ser utilizado, descargado y adaptado libremente en el repositorio <https://github.com/frautn/F2-electromagnetismo>, se presenta a los estudiantes divididos en módulos, mediante cuadernos Jupyter que se cargan en la plataforma Google Colaboratory, y que contienen breves explicaciones, ejemplos resueltos con código, y actividades propuestas para ser realizadas reutilizando el código de los ejemplos. Estos módulos asisten a estudiantes de un primer curso de electromagnetismo en la adquisición de conceptos sobre campo y potencial electrostático, distribuciones continuas de carga, campo y equipotenciales en presencia de conductores, enteramente utilizando código Python. En poco tiempo se encuentran trabajando en ejercicios cuya resolución en papel está descartada, no necesariamente por su dificultad, sino también por el volumen de cálculos repetitivos que deberían ser realizados.

La principal diferencia con el mencionado curso de mecánica, además de los temas estudiados, es el nivel de avance en sus carreras de los estudiantes en sendos cursos. En la materia Física 2 nos encontramos con estudiantes que poseen pocos o nulos conocimientos tanto de programación como de cálculo numérico. Entonces, en este caso, es importante sostener como premisa que la comprensión del código no se transforme en una dificultad extra. Con este objetivo, las numerosas e intrincadas líneas de código para cálculos complejos y generación de figuras, están

encapsuladas en funciones de bibliotecas del paquete `frautnEM` (<https://github.com/frautn/frautnEM>), diseñado específicamente para asistir a este curso, de forma que solo se trabaje con pocas líneas de código sencillo.

```
# Definimos una función para la densidad:
def Lambda(t):
    a = 1E-9
    return a*cos(2*t)

R = 20E-2 # Radio del anillo en m.
N = 1000 # Número de cargas en que se divide el objeto.

dt = 2*pi/N # Separación angular entre cargas.
t0 = dt/2 # Ángulo de la primera carga.
Q = []
ds = R*dt # Longitud de arco para cada segmento de anillo.
for i in np.arange(N):
    t = t0 + i*dt
    dq1 = Lambda(t)*ds # Carga de un pequeño arco de anillo.
    Q = Q + [[dq1, R*cos(t0+i*dt), R*sin(t0+i*dt), 0]]
```

```
0 -[
[0.013, 0.200, 0.006, 0],
[0.012, 0.199, 0.019, 0],
[0.012, 0.198, 0.031, 0],
[0.011, 0.195, 0.044, 0],
[0.011, 0.192, 0.056, 0],
[0.010, 0.188, 0.068, 0],
[0.009, 0.184, 0.079, 0],
...
[0.011, 0.192, -0.056, 0],
[0.011, 0.195, -0.044, 0],
[0.012, 0.198, -0.031, 0],
[0.012, 0.199, -0.019, 0],
[0.013, 0.200, -0.006, 0]
]
✓ 0s Python

título = 'Líneas de campo de un anillo con carga no uniforme'

plotEf(Q, dx=0.3, title=título)
✓ 4.5s Python
```

(b) La misma lista ingresada manual-

(a) Generación de la lista de cargas puntuales mente que la función `plotEf()` emplea para generar las líneas de campo. (c) Representación gráfica de la distribución de carga y líneas de campo.

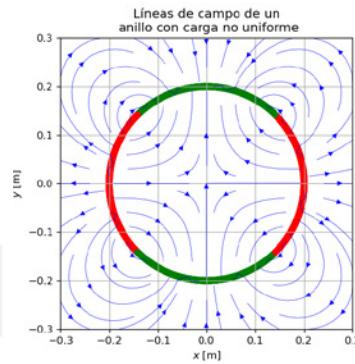


Figura 8: Cálculo de líneas de campo eléctrico generadas por una distribución de carga en un anillo.

A modo de ejemplo, la Figura 8 muestra como los estudiantes generan líneas de campo de un anillo cargado con una distribución de carga no uniforme. La Figura 8a es el código provisto con el cuaderno Jupyter para generar una lista de cargas puntuales que simulen una distribución continua con forma de anillo. Copiando este código, los estudiantes solo deben modificar la función que calcula la carga en cada ángulo, escribiéndola en un formato familiar (en este caso como `a*cos(2*t)`, siendo `t` el ángulo), el radio del anillo y el número de cargas puntuales utilizadas para representar al anillo. Si en cambio algún estudiante no se siente cómodo adaptando este código de ejemplo, sobre todo a nuevas geometrías, puede optar por generar una lista de cargas como la mostrada en la Figura 8b con, por ejemplo, una planilla de cálculo. Una vez generada la lista de cargas, solo es necesaria una línea de código para generar las líneas de campo, como muestra la Figura 8c.

```
# Formato para la lista de cargas,
# q en Coulombs, x,y,z en metros:
# [
# [q1, x1, y1, z1]
# [q2, x2, y2, z2]
# ]

Q = [
    [-1E-9, -0.25, 0, 0],
    [1E-9, 0.5, 0.25, 0],
]

niveles = [-60, -40, -20, -10, -5, 0, 5, 7, 10,
           20, 40, 60]

equipotencialesPuntuales(Q, niveles=niveles, z = 0)
✓ 0.1s Python
```

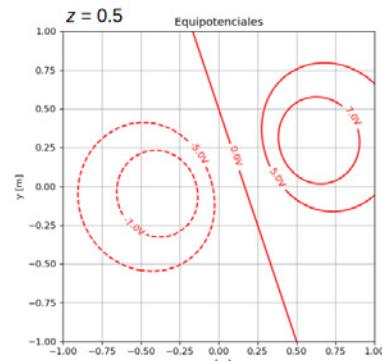
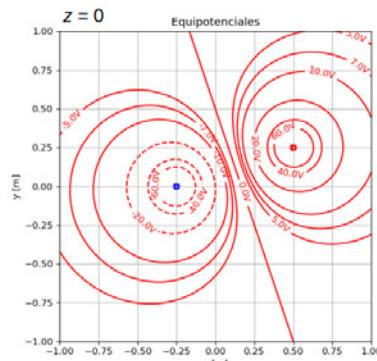


Figura 9: Curvas equipotenciales a diferentes alturas ($z = 0$ y $z = 0,5$) respecto al plano contenido las cargas.

Otro ejemplo del foco puesto en código sencillo está relacionado a una gran ventaja de esta metodología, que es la



facilidad con la que se puede salir del plano. La Figura 9 muestra cómo se pueden estudiar curvas equipotenciales sin restringirse al plano formado por las cargas reutilizando un código muy simple. En la última clase se pide la resolución de un problema nuevo y se provee un cuaderno Jupyter como plantilla para facilitar el trabajo, que debe ser compartido con los docentes para su evaluación. Dentro de dicho cuaderno se incluyen secciones para ingresar conclusiones sobre los análisis realizados. La actividad en esta experiencia consistió en responder preguntas que requirieron la obtención y análisis de vectores campo eléctrico, líneas de campo y equipotenciales, de dos segmentos cargados, ubicados no simétricamente.

4. RESULTADOS

En ambos cursos, los docentes fuimos capaces de proponer la resolución y el análisis de problemas que suelen estar descartados en una clase de pizarrón, debido a la dificultad de su abordaje. La reutilización de código, evitó destinar un tiempo excesivo a la repetición de cálculos extensos, e.g. los involucrados en las integrales múltiples para distribuciones continuas de cargas del curso de electromagnetismo, o aquellos de fuerzas de ligadura en sistemas de múltiples cuerpos del curso de mecánica analítica. Se observó que los estudiantes repasaron en un mayor número de situaciones lo trabajado en clases previas y sobre nuevos conceptos de una complejidad que excede a lo que usualmente se puede trabajar en el pizarrón. Concurrentemente, se constató una retroalimentación más rápida, lo que facilitó una estrategia de prueba y error para confirmar o corregir sus soluciones. Los estudiantes de electromagnetismo constataron resultados de fórmulas para distribuciones continuas de carga con los cálculos que realizaron con numerosas cargas infinitesimales. Adaptando el código provisto pudieron responder preguntas de un trabajo final que no podrían haber resuelto en papel durante una clase tradicional.

5. CONCLUSIONES

La naturaleza interactiva de los cuadernos Jupyter permitió a los alumnos adquirir una mejor comprensión de la física detrás de las ecuaciones, modificando parámetros y obteniendo resultados inmediatamente, resultados que son correctos y no dependientes de sus habilidades matemáticas. La herramienta evitó restringirse a problemas sencillos o con aproximaciones exageradas, que pueden cimentar conceptos erróneos. Asimismo, hace de las herramientas enseñadas en asignaturas de física algo aplicable a situaciones más realistas, habilitando al alumno a comprobar reglas de ingeniería que aprenderá en posteriores asignaturas de la carrera.

En el curso de mecánica analítica se modelizó y analizó la dinámica de dispositivos mecánicos realistas, que no podrían haber abordado en una clase tradicional. Se están incorporando cuadernos Jupyter para el análisis de vibraciones, para que el alumno cuente con herramientas para encarar problemas de elementos de máquinas y estabilidad de estructuras apenas inicie la cursada de las asignaturas correspondientes.

En el curso de electromagnetismo, se planea incluir un módulo con ejercicios de campos y equipotenciales en presencia de conductores. En particular, se incluirá el análisis numérico de una práctica de laboratorio, donde se mapea el voltaje en una cuba que contiene electrodos extensos mantenidos a potencial constante, donde las y los estudiantes solo deberán describir la geometría de los conductores mediante funciones matemáticas sencillas. Asimismo, se planea registrar cada intento de resolución de problemas de los estudiantes, no solo el resultado.

Gracias a la utilización de un paquete de funciones propio, se guardará registro de los parámetros cada vez que los estudiantes ejecuten las funciones, habilitando un análisis de sus estrategias de resolución y procesos deductivos.

6. AGRADECIMIENTOS

Los autores desean agradecer al DIIT de la UNLaM por su apoyo a través de los proyectos C2-ING-109 (2023-2024) y C2-ING-143 (2025-2026). Este último es parte del Programa de Investigación Mejora de las Estrategias Pedagógicas dirigido por la Dra. Bettina Donadello. Ambos proyectos están acreditados en el Programa de Investigación Científica, Desarrollo y Transferencia de Tecnología e Innovaciones (CyTMA2) del DIIT.

REFERENCIAS

- [1] Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., ... Scopatz, A. (2017). SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3, e103. <https://doi.org/10.7717/peerj-cs.103>
- [2] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261-272. <https://doi.org/10.1038/s41592-019-0686-2>
- [3] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95. <https://doi.org/10.1109/MCSE.2007.55>
- [4] Pekrun, R. (2006). The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice. *Educational Psychology Review*, 18, 315-341. <https://doi.org/10.1007/s10648-006-9029-9>
- [5] Vallejo, W., Díaz-Uribe, C., & Fajardo, C. (2022). Google Colab and Virtual Simulations: Practical e-Learning Tools to Support the Teaching of Thermodynamics and to Introduce Coding to Students [Publisher: American Chemical Society]. *ACS Omega*, 7(8), 7421-7429. <https://doi.org/10.1021/acsomega.2c00362>
- [6] Moraros, J., Islam, A., Yu, S., Banow, R., & Schindelka, B. (2015). Flipping for success: Evaluating the effectiveness of a novel teaching approach in a graduate level setting [Number: 1 Publisher: BioMed Central]. *BMC Medical Education*, 15(1), 1-10. <https://doi.org/10.1186/s12909-015-0317-2>
- [7] Bettachini, V. A., & Palazzo, E. (2022). Experiencia de un curso de mecánica racional basado en código. En L. F. L. et al (Ed.), *Memorias del Encuentro Argentino y Latinoamericano de Ingeniería – 2021* (pp. 1039-1049, Vol. 3).