
Universidad de Mendoza

Carrera: Ingeniería en Informática

Cátedra: Física

Trabajo Final Integrador (TFI)

Trabajo Final Integrador (TFI)

1. Fundamentación

El presente trabajo tiene como finalidad integrar los contenidos abordados a lo largo del curso (magnitudes y unidades, cinemática, leyes de Newton, trabajo y energía) con herramientas de simulación numérica y programación avanzada en Python.

El TFI busca que los estudiantes desarrollen la capacidad de modelar sistemas físicos, resolverlos mediante métodos computacionales y analizar críticamente los resultados obtenidos, contrastándolos con las leyes y teorías vistas en la asignatura.

2. Objetivos

- Aplicar los contenidos de Física vistos en las Unidades 1 a 4 en la resolución de un problema complejo.
- Utilizar programación en Python a un nivel avanzado, integrando librerías científicas y métodos numéricos.
- Desarrollar competencias en modelado, simulación y análisis de sistemas físicos.
- Elaborar visualizaciones claras y efectivas que permitan interpretar los resultados.
- Favorecer la integración entre física e informática como competencias propias de la carrera.

3. Propuesta de trabajo

Cada grupo deberá elegir una de las siguientes opciones de proyecto (o, en caso de contar con una propuesta propia, podrá presentarla al docente para su aprobación):

Opción A: Simulación del tiro parabólico con resistencia del aire

- Modelar el movimiento en 2D de un proyectil.
- Comparar la trayectoria ideal (sin rozamiento) con la real (con resistencia cuadrática).
- Calcular y graficar: posición, velocidad, aceleración, alcance y altura máxima.
- Realizar animación de la trayectoria.

Opción B: Péndulo simple y péndulo amortiguado

- Resolver numéricamente la ecuación diferencial del péndulo.
- Comparar la solución numérica con la solución analítica para ángulos pequeños.
- Analizar la variación de energía mecánica en función del tiempo.
- Generar animaciones y gráficos del movimiento.

Opción C: Colisiones en 2D y conservación del momento lineal

- Simular colisiones elásticas e inelásticas entre dos partículas en 2D.
- Calcular energías y cantidad de movimiento antes y después de la colisión.
- Verificar conservación de magnitudes.

-
- Extensión opcional: sistema de múltiples partículas en una caja.

Opción D: Trabajo de fuerzas variables

- Implementar un algoritmo para calcular el trabajo realizado por fuerzas no constantes mediante integración numérica.
- Caso sugerido: resorte no ideal (fuerza no lineal).
- Comparar los resultados numéricos con aproximaciones analíticas.
- Representar gráficamente $F(x)$ y el área bajo la curva.

4. Requisitos técnicos de programación

El código deberá:

- Estar implementado en Python 3.
- Incluir programación orientada a objetos (clases para cuerpos, fuerzas, sistemas, etc.).
- Utilizar al menos tres librerías científicas (ejemplos: numpy, scipy, sympy, matplotlib).
- Implementar un método numérico de resolución de EDOs (Euler, Runge-Kutta).
- Generar visualizaciones dinámicas (gráficos o animaciones).
- Estar documentado con comentarios claros y, de ser posible, organizado en módulos.

5. Entregables

Cada grupo deberá presentar:

1. Código fuente en Python, debidamente documentado.
2. Informe escrito (máx. 10 páginas) que incluya:
 - Introducción teórica del problema físico.

-
- Desarrollo del modelo y método numérico.
 - Resultados obtenidos (gráficas, tablas, animaciones).
 - Comparación con la teoría y análisis crítico.
 - Conclusiones.

3. Exposición oral (15 minutos por grupo) con demostración de la simulación.

6. Criterios de evaluación

- Modelado físico (30%): corrección en la aplicación de las leyes y ecuaciones.
- Calidad de la programación (30%): uso de OOP, librerías, eficiencia y organización.
- Análisis de resultados (20%): comparación con teoría, interpretación crítica.
- Presentación (20%): claridad del informe, calidad de gráficos/animaciones, exposición oral.

7. Fechas y organización

- Formación de grupos: Máximo 3 integrantes.
- Entrega de código e informe: [indicar fecha].
- Exposiciones: [indicar fecha].