

S13 – Sesión 25

Ejercicios integradores antes de la Solemne 2

Departamento de Física.

Corodinadora: C Loyola

Profesores C Femenías / F Bugini / D Basantes

Primer Semestre 2025

Universidad Andrés Bello

Departamento de Física y Astronomía



Ruta de la sesión (90 min)

Warm-up: ciclos **for** (20 min)

NumPy y estadística (30 min)

Visualización con Matplotlib (15 min)

pandas y datos reales (20 min)

Cierre

Warm-up: ciclos for (20 min)

Warm-up: ciclos `for` (20 min) ∈ Ejercicio 1 (10 min) – Caída libre discreta

Enunciado

- Parte un balón desde $h_0 = 20\text{ m}$ con $v_0 = 0$.
- Crea un **ciclo `for`** con paso $\Delta t = 0.2\text{ s}$.
- En cada iteración imprime t , h , v hasta que $h \leq 0$.
- Verifica que la altura nunca es negativa.

```
1  g = 9.8          # m/s^2
2  h, v = 20.0, 0.0
3  dt = 0.2
4  for step in range(100):      # suficiente margen
5      t = step * dt
6      print(f"{t:4.1f} s  h={h:5.2f} m  v={v:5.2f} m/s")
7      if h <= 0:
8          break
9      v = v - g*dt
10     h = h + v*dt
```

Warm-up: ciclos `for` (20 min) ∈ Ejercicio 2 (10 min) – Fibonacci y proporción áurea

Tarea

1. Con un **for** genera los primeros 15 términos de Fibonacci.
2. Durante el bucle imprime la razón F_{n+1}/F_n .
3. ¿A qué número converge la razón? (≈ 1.618)

```
1 a, b = 1, 1
2 for n in range(15):
3     print(f'F{n+1}={a}, razón={b/a:.5f}')
4     a, b = b, a+b
```

NumPy y estadística (30 min)

NumPy y estadística (30 min) ∈ Ejercicio 3 (15 min) – Órbita circular con arrays

Pasos

1. Usa `np.linspace(0, 2 π , 360, endpoint=False)` para 360 ángulos.
2. Calcula $x = r \cos \theta$, $y = r \sin \theta$ con $r = 1$.
3. Muestra el promedio \bar{x}, \bar{y} y discute por qué se acerca a 0, 0.

```
1 import numpy as np
2 theta = np.linspace(0, 2*np.pi, 360, endpoint=False)
3 x = np.cos(theta); y = np.sin(theta)
4 print(" <x> =", x.mean(), " <y> =", y.mean())
```

NumPy y estadística (30 min) ∈ Ejercicio 4 (15 min) – Temperaturas estelares simuladas

Enunciado

- Genera 200 temperaturas: `T = np.random.normal(6000, 300, 200)`.
- Con NumPy calcula: media, desvío estándar, percentiles 25/75.
- ¿Cuántas estrellas caen fuera del rango $\mu \pm 3\sigma$?

```
1 T = np.random.normal(6000, 300, 200)
2 mu, sigma = T.mean(), T.std()
3 lo, hi = mu - 3*sigma, mu + 3*sigma
4 outliers = ((T < lo) | (T > hi)).sum()
5 print(f"{outliers} estrellas fuera de +- 3 sigma")
```


Visualización con Matplotlib (15 min)

Visualización con Matplotlib (15 min) ∈ Ejercicio 5 (15 min) – Ley de Kepler (mini versión)

Datos de planetas interiores

```
1 import numpy as np, matplotlib.pyplot as plt
2 a = np.array([0.39, 0.72, 1.00, 1.52]) # semieje mayor (AU)
3 T = np.array([0.24, 0.61, 1.00, 1.88]) # periodo (años)
```

1. Dibuja un **scatter** de a vs T .
2. Ajusta $\log_{10} T = m \log_{10} a + b$ con `np.polyfit`.
3. Grafica la recta sobre los datos en escala log-log.
4. ¿El exponente m se acerca a 1.5?

pandas y datos reales (20 min)

Archivo a usar

```
https://gitarra.cl/lectures/gfiles/-/raw/main/pcfi161/S12/stars\_brightness.csv
```

1. Lee el CSV en un `DataFrame`.
2. Muestra `head()` y `describe()`.
3. Agrupa por `spectral_class`: `mean(T_K)` y mínima magnitud aparente.
4. Concluye en 2 líneas qué clase es más caliente y brillante.

pandas y datos reales (20 min) ∈ Ejercicio 7 (opc., 10 min) – Filtrar exoplanetas

- Se entregará un CSV muy pequeño con columnas `name`, `mass_Mj`, `period_d`.
- Filtra planetas con $m < 1.5 M_J$ y periodo < 10 d.
- Guarda el resultado en `exoplanets_short.csv`.

Cierre

- Repasen estos problemas: conceptos clave para la Solemne 2.
- Dudas → consultas al ayudante o profesor.
- ¡Éxito en la evaluación!