

# Programación para Física y Astronomía

Departamento de Física.

---

Corodinadora: C Loyola

Profesores C Femenías / F Bugini / D Basantes

Primer Semestre 2025

Universidad Andrés Bello

Departamento de Física y Astronomía



Introducción y Repaso

Matplotlib Avanzado

Gráficos 3D Básicos

Introducción a pandas (Opcional)

Ejercicios Prácticos

Conclusiones y Próximos Pasos

# Introducción y Repaso

---

- **Semana 6, Sesión 2 (Sesión 12)** se enfocó en:
  - Manipulación avanzada de NumPy (**reshape**, **transpose**, **concatenate**).
  - Uso de **np.random** para valores aleatorios y **np.linalg** (álgebra lineal).
  - Primeros pasos con **Matplotlib** (**plot**, **scatter**).
- **Objetivo de hoy:** Avanzar con gráficos más complejos (subplots, histogramas, 3D) y comenzar a manipular datos de forma más eficiente (posiblemente una introducción a **pandas**).

- **Profundizar** en las opciones de **Matplotlib** para visualizar datos:
  - Subplots, estilos, anotaciones.
  - Gráficos de barras, histogramas y 3D simples.
- **Familiarizarnos** con un primer acercamiento a **pandas**, si el tiempo lo permite.
- **Ejercitar** estos conceptos con ejemplos y datos relevantes para Física/Astronomía.

# Matplotlib Avanzado

---

## Matplotlib Avanzado ∈ Creando Varias Gráficas (Subplots)

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(0, 2*np.pi, 100)
5 y_sin = np.sin(x)
6 y_cos = np.cos(x)
7
8 fig, axs = plt.subplots(2, 1, figsize=(6,8))
9 axs[0].plot(x, y_sin, 'r-', label="sin(x)")
10 axs[0].legend()
11 axs[0].set_title("Seno")
12
13 axs[1].plot(x, y_cos, 'b--', label="cos(x)")
14 axs[1].legend()
15 axs[1].set_title("Coseno")
16
17 plt.tight_layout()
18 plt.show()
```

```
1 import matplotlib.pyplot as plt
2
3 etiquetas = ['A', 'B', 'C', 'D']
4 valores = [10, 25, 7, 15]
5
6 plt.bar(etiquetas, valores, color='lightblue')
7 plt.title("Gráfico de Barras")
8 plt.xlabel("Categorías")
9 plt.ylabel("Valores")
10 plt.show()
```

- Útil para datos categóricos o comparaciones discretas.
- `plt.barh` para barras horizontales.



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 data = np.random.randn(1000) # 1000 valores normal estándar
5 plt.hist(data, bins=20, color='green', alpha=0.7)
6 plt.title("Histograma de datos aleatorios")
7 plt.xlabel("Valor")
8 plt.ylabel("Frecuencia")
9 plt.show()
```

- **bins** controla el número de barras en el histograma.
- **alpha**: transparencia de las barras.

# Gráficos 3D Básicos

---

## Gráficos 3D Básicos ∈ Instanciando Ejes 3D

```
1 from mpl_toolkits.mplot3d import Axes3D  # a partir de
   ↪ mpl_toolkits
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 fig = plt.figure()
6 ax = fig.add_subplot(111, projection='3d')
7 # ax ahora es un eje 3D
```

- Se puede usar `plt.subplots(subplot_kw={'projection':'3d'})` también.
- Métodos como `ax.plot3D`, `ax.scatter3D` están disponibles.

## Gráficos 3D Básicos ∈ Plot de Superficie 3D (Ejemplo)

```
1  from mpl_toolkits.mplot3d import Axes3D
2  import matplotlib.pyplot as plt
3  import numpy as np
4
5  x = np.linspace(-5, 5, 50)
6  y = np.linspace(-5, 5, 50)
7  X, Y = np.meshgrid(x, y)
8  Z = np.sin(np.sqrt(X**2 + Y**2))
9
10 fig = plt.figure()
11 ax = fig.add_subplot(111, projection='3d')
12 ax.plot_surface(X, Y, Z, cmap='viridis')
13 ax.set_title("Superficie 3D")
14 plt.show()
```

• `plot_surface`, `plot_wireframe`, `plot_contour`, etc.

## Introducción a pandas (Opcional)

---

## Introducción a pandas (Opcional) ∈ ¿Por qué pandas?

- Librería para **manejo y análisis** de datos tabulares, muy usada en ciencia de datos.
- Estructuras **DataFrame** y **Series**.
- Integración con **NumPy** y **Matplotlib**.
- Lectura y escritura de archivos CSV, Excel, SQL, etc.

# Introducción a pandas (Opcional) ∈ Ejemplo Rápido con DataFrame

```
1 import pandas as pd
2
3 datos = {
4     'Masa': [1.0, 3.5, 2.2],
5     'Velocidad': [10.0, 7.5, 12.3]
6 }
7 df = pd.DataFrame(datos)
8 print(df)
9
10 # Calcular Energía Cinética
11 df['E_c'] = 0.5 * df['Masa'] * (df['Velocidad']**2)
12 print(df)
```

- Facilita manipulación de datos y operaciones columnares.
- Opcional: graficar `df['E_c']` con `df.plot()` o `plt`.

# Ejercicios Prácticos

---



# Ejercicios Prácticos ∈ Ejercicio 1: Subplots Seno, Coseno, Tangente

## Enunciado

- Generar  $x$  en  $[0..2\pi]$ , con `np.linspace`.
- Crear **3 subplots** en una columna:
  1. `sin(x)`
  2. `cos(x)`
  3. `tan(x)`
- Ajustar **ejes** y uso de **legends/titles**.
- Manejar la tangente en zonas cercanas a  $\pi/2$  (**ver si se disparan valores**).

## Ejercicios Prácticos ∈ Ejercicio 2: Comparando Histogramas y Barras

### Enunciado

- Generar 100 valores aleatorios con `np.random.normal(5,1,100)` (media=5, sigma=1).
- Hacer un **histograma** de esos datos (`plt.hist`).
- Contar manualmente las frecuencias en cada bin, y crear un **gráfico de barras** para comparar.
- Observar similitudes y diferencias.

**Objetivo:** Entender la relación entre un histograma y los datos de frecuencia.

### Enunciado

- Definir una **función**  $f(x,y) = e^{-x^2-y^2}$  (campana).
- Usar `np.meshgrid` para crear `X,Y` en `[-2,2]`.
- Calcular `Z = f(X,Y)`.
- Crear un **plot de superficie** (`ax.plot_surface`) con un color map bonito.

**Sugerencia:** Ajustar `figsize` y `ax.set_zlim` para ver mejor la forma.

## Ejercicios Prácticos ∈ Ejercicio 4 (Opcional): Leyendo Datos con pandas

### Enunciado

- Tener un archivo CSV simple con columnas: **Tiempo**, **PosX**, **PosY** (p.e., trayectoria de un objeto).
- Usar `pandas.read_csv('archivo.csv')` para cargarlo a un DataFrame.
- Graficar **PosY** vs. **Tiempo** con **Matplotlib**.
- Agregar **labels** y título.

**Objetivo:** Introducir la idea de leer datos externos y visualizar.

- Dividir en **parejas** o **tríos**.
- Cada grupo elige **2-3 ejercicios** según interés.
- Implementar y analizar resultados en un **notebook de Colab**.
- Compartir conclusiones y dificultades al final.

- Para **subplots** múltiples, considerar `figsize` y `tight_layout`.
- En **3D**, probar otras funciones como  $\cos(\sqrt{x^2 + y^2})$  o  $\sin(x * y)$ .
- En **pandas**, puedes usar `df.describe()` para ver estadísticas rápidas.
- Explorar **paletas de colores**: `cmap='plasma'`, `cmap='coolwarm'`, etc.

- ¿Alguna confusión con **subplots** o configuración de ejes?
- ¿Problemas para instalar o importar **pandas**?
- ¿Dudas sobre gráficos 3D y el uso de **meshgrid**?

Levanta la mano o consulta abiertamente para que todos aprendan.

## Conclusiones y Próximos Pasos

---



- Comparte tus resultados en **subplots**, **histogramas** o **3D**.
- Si usaste **pandas**, ¿cómo fue la experiencia de cargar datos y manipularlos?
- Observa **beneficios** de la visualización en la interpretación de datos físicos/matemáticos.

- **Matplotlib avanzado:**
  - Subplots múltiples, histogramas, barras, y nociones 3D.
- **Breve introducción a pandas** para datos tabulares.
- Integramos **NumPy**, **Matplotlib** y (opcionalmente) **pandas** para un flujo de trabajo más completo.
- Seguiremos perfeccionando la **visualización** y manipulación de datos en las próximas sesiones.

- **Semana 7, Sesión 2:** Profundizaremos en **visualizaciones personalizadas** (títulos, ejes, anotaciones, múltiples figuras) y exploraremos más sobre **manejo de datos**.
- Revisaremos **buenas prácticas** para proyectos colaborativos y la combinación de Python con LaTeX (si alcanza el tiempo).

**Sigan practicando con datos reales o simulados para afianzar los conocimientos.**

- **Matplotlib Docs** (especialmente ejemplos de galería).
- **pandas Documentation** (guía de 10 minutos).
- **Seaborn** (librería de visualización avanzada sobre Matplotlib).
- **Python Standard Library** - repaso general.

# Gracias y hasta la próxima sesión

- Recuerden subir sus notebooks y experimentos a Google Drive o repositorio compartido.
- Practiquen diferentes tipos de gráficos e integren datos con **pandas** si pueden.