

# 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

Problema a Evaluar (Evaluación en Clase)

Trabajo y Discusión

Cierre y Próximos Pasos

# Introducción y Repaso

---

- **Semana 7** nos enfocamos en:
  - Gráficos avanzados en Matplotlib (subplots, histogramas, 3D).
  - Posible integración con pandas (archivos CSV), y animaciones/herramientas 3D.
- **Semana 8, Sesión 1 (Sesión 15):**
  - Introducimos **POO** (clases, métodos, atributos) y combinamos con NumPy/pandas.
  - Ejercicios de simulaciones o gestión de objetos en Python.
- **Objetivo de hoy:** Aplicar y evaluar parte de lo visto en **Semana 7** (visualización y manipulación de datos), en un ejercicio integrador. Luego, tendremos discusión y retroalimentación.

- **Resolver** un **Problema a Evaluar** en grupos, enfocándonos en:
  - Subplots avanzados, histogramas, 3D o datos tabulares (Semana 7).
  - Generar o leer datos, procesarlos y graficarlos adecuadamente.
- **Discutir** las soluciones y aclarar dudas posteriores.
- **Fomentar** la colaboración y la organización en un tiempo acotado (25-30 minutos).

## Problema a Evaluar (Evaluación en Clase)

---

# Problema a Evaluar (Evaluación en Clase) ∈ Problema a Evaluar: Visualización Avanzada con Matplotlib

## Contexto (Semana 7):

- Se practicaron **subplots múltiples**, **histogramas**, y **3D** con Matplotlib.
- Uso potencial de **pandas** para datos tabulares.

## Tareas:

1. Generar o leer un **DataFrame** con al menos 2 columnas cuantitativas (p.e., **x** e **y**) y 1 columna categórica (p.e., **categoría**).
  - Puede ser **aleatorio** (`np.random`) o un archivo CSV sencillo.
2. Crear un **subplot con 2-3 paneles** (en una fila o columna):
  - Panel 1: **Histogram** de la columna **x**.
  - Panel 2: **Scatter plot** de **x** vs. **y**, coloreado por **categoría**.
  - Panel 3 (opcional/bonus): un **plot 3D** si desean expandir, o un **boxplot** si lo prefieren.
3. Personalizar **etiquetas**, **títulos**, y **leyendas**.

# Problema a Evaluar (Evaluación en Clase) ∈ Instrucciones para la Evaluación

- Trabajar en **grupos de 2-3 estudiantes**.
- Crear un **notebook en Colab** (o un script local) llamado `Eval_Semana8_Apellidos.ipynb`.
- **Desarrollo** del problema:
  1. Generar/leer datos (p. ej., `df = pd.DataFrame(...)` o `pd.read_csv(...)`).
  2. Crear **subplots** y visualizar:
    - Histograma de **x**.
    - Scatter **x** vs **y**, color o marker según **categoría**.
    - (Opcional) Tercer subplot: 3D simple, boxplot, o algo similar.
  3. Etiquetar ejes, leyendas, título general si desean.
- Al finalizar (máximo **25-30 minutos**), **subir el archivo a CANVAS** (una entrega por grupo).



Se considerarán los siguientes criterios:

- **Funcionalidad** (40%): el código corre sin errores y se generan los subplots requeridos.
- **Uso apropiado de Matplotlib** (20%): subplots correctos, buenas prácticas de rotulación.
- **Manejo de Datos** (20%): DataFrame, lectura/generación de datos, coherencia de categorías, etc.
- **Organización/Comentarios** (20%): claridad del notebook, explicaciones de cada paso.

**Nota:** Se valoran detalles extra (filtros de datos, color maps, `plt.tight_layout`, etc.).

Tienen 25-30 minutos para resolver y subir la solución a CANVAS.

Sugerencias:

- Decidir rápido si generan datos (`np.random`) o usan un CSV existente.
- Asegurarse de **importar matplotlib y pandas** (si se usa).
- Revisar **leyendas, labels, titles** de cada subplot.
- `plt.tight_layout()` para mejorar la presentación.

## Trabajo y Discusión

---

- Pueden hablar en voz baja para coordinar, cada grupo crea su notebook o script.
- Consultas breves en caso de bloqueos severos, pero intenten ser autónomos.
- Asegúrense de **probar la ejecución completa** antes de subir a CANVAS.

## Subida a CANVAS

- Un integrante del grupo debe subir el **notebook .ipynb** (o .py) dentro del plazo (25-30 min).
- Revisen que estén todos los subplots requeridos.
- Comentar en Markdown o en el código el enfoque y los pasos.

**Tras la entrega**, discutiremos brevemente soluciones y dificultades encontradas.

- ¿Fue sencillo combinar **histograma** y **scatter** en subplots?
- ¿Cómo asignaron colores al **scatter** según la categoría? (p. ej. `c=...` o `hue=...` en Seaborn).
- ¿Dificultades con **figsize** o la disposición de subplots?
- ¿El tiempo de 25-30 min fue suficiente?

- Actividad integró:
  - **Generación/lectura de datos** (DataFrame).
  - Gráficas **subplots** avanzadas (hist, scatter, 3D o boxplot opcional).
  - Personalización y estilos de Matplotlib (Semana 7).
- Apunta a **manejo fluido** de Matplotlib para presentaciones de datos.
- Revisión más detallada y retroalimentación vendrá tras la clase.

## Cierre y Próximos Pasos

---



- Ejecutamos un **problema evaluado** enfocado en **visualización avanzada** (subplots/hist/scatter) de la Semana 7.
- Observamos **estrategias** para leer/generar datos y graficarlos en Matplotlib.
- Discutimos **puntos complejos** (coloreo, layout, etiquetado).
- Continuamos la línea de integrar datos reales y presentarlos de forma clara.

- Avanzar con **POO** más complejo (herencia, polimorfismo) o **análisis de datos** más avanzado (estadísticas, merges).
- Revisar **retroalimentación** del problema actual en la siguiente clase.
- ¡Sigán practicando la personalización de subplots y la integración con **pandas**!

- **Matplotlib Gallery** - Inspírate con ejemplos.
- **pandas Docs** - para manipular DataFrames.
- **Seaborn** - librería para gráficos estadísticos basados en Matplotlib (opcional).

# en trabajo y hasta la próxima ses

- Recuerden revisar notas y feedback en CANVAS.
- Cualquier duda, ¡pregúnten en foros o a sus compañeros!