

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

Retroalimentación Solemne II

POO Adicional

Ejemplo Práctico

Conclusiones y Próximos Pasos

Introducción y Repaso

- **Semana 10, Sesión 2:** Estudiantes rindieron la Solemne II, abarcando los contenidos de:
 - POO (clases, herencia).
 - NumPy avanzado (matrices, linalg, random).
 - Matplotlib (gráficas 2D/3D, subplots, histogramas).
 - pandas (lectura de CSV, análisis básico).
- **Objetivo de la Semana 11:** Avanzar con temas complementarios, retroalimentar la Solemne II y consolidar nuevas competencias.
 - Podría ser POO más avanzada (polimorfismo) o robustez en manejo de datos (filtrados, merges, etc.), de acuerdo con el Syllabus.
 - Revisar la **retroalimentación** general de la Solemne II.

- **Entregar** feedback global de la Solemne II, identificando logros y áreas de mejora.
- **Refrescar** la POO y ver algún aspecto adicional (polimorfismo, composición, etc.), según tiempo y Syllabus.
- **Explorar** un ejemplo práctico o un caso de datos más complejo (opcional).
- **Planificar** los siguientes pasos en el curso (si hay proyectos finales o nueva Solemne).

Retroalimentación Solemne II

- **Distribución de notas** (si aplica):
 - Rango de notas y promedio general (sin datos reales, se ejemplifica).
 - Observación de que **la mayoría manejó bien** la sintaxis POO, pero falló en subplots 3D, por ejemplo.
- **Tiempo de examen** fue suficiente para la mayoría.
- **Entrega en CANVAS** con archivos .ipynb (como planeado).

- **Lo positivo:**
 - Claridad en la definición de clases, herencia y atributos.
 - Uso de **NumPy** para operaciones vectorizadas o linalg.
 - Integración con pandas para leer datos y mostrarlos en gráficas simples.
- **Lo que se puede mejorar:**
 - Estructura de **subplots** y personalización (leyendas, ejes).
 - Falta de comentarios en secciones clave o poca explicación en Markdown.
 - Manejo de **excepciones** o validaciones (opcional, según nivel).
 - Unificación de código (evitar repetir lógica).

- **Probar y comentar** el código en cada paso, especialmente en exámenes o tareas.
- Organizar **funciones auxiliares** en vez de repetir (DRY principle).
- Para gráficas, usar **subplots** con nombres descriptivos (`ax1`, `ax2` en lugar de `axs[0]`, `axs[1]`).
- Revisar **ejemplos oficiales** de Matplotlib/pandas para inspirar estilos y layouts.

POO Adicional

- **Polimorfismo** = la capacidad de usar la misma interfaz (métodos) en distintos tipos (clases derivadas).
- En Python, se basa en la **duck typing**: “si camina como pato y suena como pato...”.
- Ejemplo:
 - **Particle** y **Star** pueden compartir un método `show_info()` pero cada uno implementarlo distinto.
- **Beneficio**: simplifica la lógica si tenemos una lista de objetos distintos pero un método común.

- **Composición** = un objeto está conformado por otros objetos (relación *has-a*).
- Ejemplo:
 - Clase **SolarSystem** que contiene una lista de **Body** (planetas, estrellas).
 - Métodos para agregar, remover, iterar cuerpos.
- **Ventaja:** modularidad y organización del código, sin heredar (no es un **is-a**).

POO Adicional ∈ Ejemplo: Sistema SolarSimplified

```
1 class Body:
2     def __init__(self, name, mass, x, y):
3         self.name = name
4         self.mass = mass
5         self.x = x
6         self.y = y
7
8 class SolarSystem:
9     def __init__(self):
10         self.bodies = []
11
12     def add_body(self, body):
13         self.bodies.append(body)
14
15     def total_mass(self):
16         return sum(b.mass for b in self.bodies)
17
18     def show_bodies(self):
19         for b in self.bodies:
20             print(f"{b.name}: mass={b.mass},
```

Ejemplo Práctico

Enunciado Breve

- Usar la clase `SolarSystem`, agregar 3-4 `Body`.
- Crear una **matriz NxN** con las distancias entre cada par, usando `NumPy`.
- Graficar en un **heatmap** (`plt.imshow`) con `colorbar`, etiquetar ejes con `Body.name`.

Objetivo: ilustrar cómo la **composición** (múltiples `Body` en `SolarSystem`) + `NumPy` y `Matplotlib` generan un análisis visual.

- Formar **parejas/tríos**, retomar la idea anterior u otro ejemplo creativo.
- Reforzar:
 - **Composición** (relación *has-a*), polimorfismo si lo desean.
 - **Matrices** de distancias o valores con **NumPy**.
 - **Visualización** en matplotlib (heatmap, scatter, etc.).
- **Comparar** resultados al final y plantear dudas.

- **Mantener** el código ordenado, con `def main()` o celdas separadas en Colab.
- **Documentar** la composición y polimorfismo si los aplican.
- `numpy.linalg.norm` para distancias, o una función manual $(x1 - x2)^2 + (y1 - y2)^2$.
- `plt.imshow` y `plt.colorbar + tick_labels` si quieren personalizar ejes.

Conclusiones y Próximos Pasos

- Compartir **cómo integraron** la clase `SolarSystem` (u otra) con NumPy y gráficas.
- Dificultades encontradas, **beneficios** de la composición vs. herencia.
- ¿Algún polimorfismo en acción (ej. diferentes cuerpos con un `show_info()`)?

- Revisamos **retroalimentación** de la Solemne II, con énfasis en fortalezas y debilidades.
- Profundizamos POO con **polimorfismo** y **composición**, aplicadas a ejemplos físicos/astronómicos.
- Continuamos integrando **NumPy** y **Matplotlib** para análisis y visualización en proyectos de clase.
- Sentamos base para proyectos finales o simulaciones más completas (si así lo indica el Syllabus).

- Posible **desarrollo de un proyecto integrador** (si corresponde al plan).
- Expandir **POO** (más métodos especiales, polimorfismo avanzado) o **data handling** (estadística, merges, time-series).
- Revisar la **retroalimentación individual** de la Solemne II en detalle (notas en CANVAS).

- **Python Docs - Classes** - Sección herencia y polimorfismo.
- **Real Python - OOP** - polimorfismo y ejemplos.
- **Matplotlib Docs** - sección `imshow`, heatmap, colorbar.
- **Foros y Comunidad:** Stack Overflow, Reddit `/r/learnpython`.

en trabajo y hasta la próxima ses

- Revisen **Canvas** para feedback detallado de la Solemne II.
- Practiquen con **POO avanzada** (composición, polimorfismo) en ejemplos propios.