



En esta oportunidad trabajaremos sobre todo lo aprendido en la sección de **POO de Python** para organizar y gestionar código de una manera más modular y eficiente. 🚀

¡Manos a la obra!

- **Desafío** 🎯:

1. Crear una **clase llamada Bicicleta** y luego aplica los siguientes accionables:

- ➡ Agregar al menos 3 atributos
- ➡ Agregar al menos 3 métodos
- ➡ Agregar el método constructor de la clase.

- Guardarlo en un archivo llamado ejercicio1.py

2. Crear una **clase llamada Animal**, otra llamada **Perro** y otra llamada **Águila**.

- ➡ **La clase Animal tiene:**

- atributo cantidad\_patas: numérico
- atributo tipo: vertebrado/invertebrado
- método comer(): retorna un string “estoy comiendo”

- ➡ **La clase Perro hereda de Animal y agrega:**

- atributo nombre: texto
- atributo raza: texto
- método correr(): retorna un string “estoy corriendo”

- ➡ **La clase Aguila hereda de Animal y agrega:**

- método volar(): retorna un string “estoy volando”

- Guardarlo en un archivo llamado ejercicio2.py

3. A partir del siguiente enunciado, **crear las clases necesarias** (con sus respectivos atributos y métodos) para poder representarlos.

→ “Juan Lopez tiene 25 años y es de profesión Abogado. Por la tarde, después de trabajar, sale a caminar. También tiene una bicicleta amarilla marca “Massino” y a veces sale a dar vueltas en ella”.

- Guardarlo en un archivo llamado ejercicio3.py

- **Tiempo de dedicación** ⌚

2 horas

- **⚠ Condición**

Esta práctica o ejercitación **no requiere ser entregada y/o evaluada** por el mentor. No obstante, una vez que hayas completado el ejercicio te invitamos a **compartir tu resolución** con el resto de la clase/participantes por los canales habilitados y así poder aprender colaborativamente.