

Trabajo Práctico: Sistema de Adopción de Perros

Materia: Programación Orientada a Objetos – Python

Modalidad: Individual

Entrega: [fecha a definir]

Objetivo

Diseñar e implementar en **Python** un sistema que simule una plataforma de adopción de perros, utilizando los **principios de la Programación Orientada a Objetos (POO)**: encapsulamiento, herencia, composición y polimorfismo.

Descripción general

El sistema debe permitir gestionar una base de datos en memoria de **perros en adopción** y **usuarios adoptantes**. A través de clases y objetos, los usuarios podrán registrarse, buscar perros según sus preferencias y completar el proceso de adopción.

Requisitos funcionales

1. Clase **Perro**

- Atributos: `nombre`, `raza`, `edad`, `tamaño`, `peso`, `estado_salud`, `vacunado`, `estado` ('disponible', 'reservado', 'adoptado'), `temperamento`, `id`.
- Métodos: cambiar estado, mostrar información, etc.

2. Clase **UsuarioAdoptante**

- Atributos: `nombre`, `dni`, `email`, `preferencias` (raza, edad, tamaño), `historial_adopciones`.
- Métodos: registrarse, modificar datos, ver historial, etc.

3. Clase **SistemaAdopcion**

- Métodos para:
 - Cargar y eliminar perros
 - Registrar usuarios
 - Postular a un perro
 - Confirmar adopción
 - Sugerir perros según preferencias
 - Mostrar listados (perros disponibles, por estado, por usuario)
-

Requisitos técnicos

- Uso correcto de **clases y objetos**
- Aplicación de:
 - **Herencia** (por ejemplo, distintas razas con comportamiento distinto si se quiere extender)
 - **Polimorfismo** (métodos sobrescritos o reutilización de interfaces)
 - **Composición** (por ejemplo, el sistema contiene una lista de usuarios y perros)
- Listas dinámicas (`vectores`) y estructuras nativas de Python
- Separación clara entre lógica del modelo y la interfaz (menú o interacción con usuario)

- Uso de buenas prácticas: código limpio, funciones bien nombradas, comentarios cuando sea necesario