

Taller – Sistema de Adopción de Mascotas

Objetivo general

El objetivo de este taller es practicar los conceptos de Python relacionados con tipos de datos, funciones, estructuras de control, Programación Orientada a Objetos, herencia, encapsulamiento, manejo de errores y organización en paquetes. Cada punto lo guiará para construir un sistema simple de gestión de adopciones de mascotas en un refugio.

Punto 1 | Registro de mascotas

Cree un archivo `models/mascota.py` con una clase `Mascota` que tenga los atributos:

- `nombre` (str)
- `especie` (str)
- `edad` (int)
- `adoptado` (bool, por defecto `False`)

Incluya un método `__str__` que permita mostrar la información de la mascota en texto.

Punto 2 | Clase Persona y Adoptante

Cree un archivo `models/persona.py` con la clase `Persona`, que tenga atributos:

- `nombre`
- `edad`
- Método `presentarse()` que muestre un saludo con el nombre y edad

En el mismo archivo, cree la clase `Adoptante` que herede de `Persona` y agregue:

- Un atributo `mascotas_adoptadas` (lista vacía).
- Un método `adoptar(mascota)` que agregue una mascota a la lista de adoptadas.

Punto 3 | Clase Refugio

Cree un archivo `models/refugio.py` con la clase `Refugio`:

- Atributo privado `__mascotas` que almacene una lista de objetos `Mascota`.
- Métodos:
 - `registrar_mascota(mascota)` para agregar una nueva mascota.
 - `listar_disponibles()` que retorne las mascotas no adoptadas.

- `asignar_adopcion(nombre_mascota, adoptante)` que:
- Busque la mascota por nombre.
- Verifique si está disponible.
- Si no existe o ya fue adoptada, maneje el error mostrando un mensaje desde la clase.
- Si está disponible, cambie su estado a `adoptado=True` y agréguela al adoptante.

Punto 4 | Funciones auxiliares

Cree un archivo `models/helpers.py` con una función `buscar_mascota(nombre, lista_mascotas)` que devuelva la mascota si existe o `None` en caso contrario.

Punto 5 | Programa principal

Cree un archivo `main.py` que permita al usuario interactuar con el sistema:

- Registre 3 mascotas de ejemplo.
- Cree un adoptante.
- Muestre un menú interactivo con `input()` donde el usuario pueda:
 - Listar mascotas disponibles.
 - Adoptar una mascota por nombre.
 - Ver las mascotas adoptadas.
 - Salir.

El menú debe usar estructuras de control (`if/elif/else`, bucles) y llamar a los métodos de las clases.

Entrega

El proyecto debe tener la siguiente estructura:

```

adopcion/
├── main.py
├── models/
│   ├── mascota.py
│   ├── persona.py
│   ├── refugio.py
│   └── helpers.py

```