

# EJERCICIO: Sistema de Clínica Veterinaria

## ENUNCIADO COMPLETO (Diagrama + Código)

### PARTE 1: DIAGRAMA DE CLASES UML

Crea un diagrama UML para un sistema de clínica veterinaria con las siguientes **4 clases**:

#### Clase 1: Dueño

##### Atributos:

- nombre (string)
- telefono (string)
- mascotas (lista de objetos Mascota)

##### Métodos:

- constructor, accesadores y mutadores
- agregar\_mascota(mascota): agrega una mascota a su lista
- mostrar\_mascotas(): muestra todas sus mascotas
- buscar\_mascota(nombre): busca una mascota por nombre

#### Clase 2: Mascota

##### Atributos:

- nombre (string)
- especie (string) - ej: "perro", "gato", "ave"
- raza (string)
- expediente (objeto Expediente) □ **composición**

### **Métodos:**

- actualizar\_raza(nueva\_raza): cambia la raza
- obtener\_info(): devuelve información completa
- mostrar\_expediente(): imprime su expediente

### **Clase 3: Expediente**

#### **Atributos:**

- historial (lista de strings) - ej: ["vacuna 2023", "cirugía 2024"]
- vacunas (lista de strings) - nombres de vacunas aplicadas
- fecha\_creacion (string)

#### **Métodos:**

- agregar\_visita(motivo): agrega una visita al historial
- agregar\_vacuna(vacuna): registra una vacuna aplicada
- mostrar\_historial(): muestra todo el historial

### **Clase 4: Veterinario**

#### **Atributos:**

- nombre (string)
- especialidad (string) - ej: "cirugía", "general"
- mascotas\_atendidas (lista de objetos Mascota) □ **colaboración**

#### **Métodos:**

- atender\_mascota(mascota, motivo): atiende una mascota
- mostrar\_atendidas(): muestra mascotas atendidas
- contar\_atenciones(): cuenta cuántas ha atendido

## PARTE 3: REQUISITOS DE IMPLEMENTACIÓN

### Para la COMPOSICIÓN (Mascota-Expediente):

- El Expediente se debe crear **DENTRO** del constructor de Mascota
- No debe existir Expediente sin Mascota
- Si se elimina una Mascota, su Expediente también debe eliminarse

### Para la COLABORACIÓN:

- Dueño y Mascota son objetos **independientes**
- Veterinario y Mascota son objetos **independientes**
- Pueden existir sin relación (un Dueño sin mascotas, un Veterinario sin atender)

### Validaciones obligatorias:

- No asignar mascotas duplicadas a un dueño
- No registrar la misma visita dos veces en un expediente
- Verificar que los nombres no estén vacíos
- Las especies deben ser válidas: ["perro", "gato", "ave", "roedor", "reptil"]

## PARTE 4: PROGRAMA DE PRUEBA

Tu programa debe demostrar:

- Crear 2 dueños diferentes
- Crear 3 mascotas (con composición: expediente incluido)
- Crear 2 veterinarios
- Asignar mascotas a dueños (colaboración)
- Veterinarios atienden mascotas (colaboración)
- Mostrar toda la información usando `__str__`