

## **TALLER 1**

### **Pet Health**

La veterinaria Pet Health ha solicitado el diseño de un sistema de información para gestionar los servicios ofrecidos en su clínica. El sistema debe permitir administrar la información de los doctores, los dueños de las mascotas, las mascotas que atienden y los tratamientos realizados.

El sistema debe centrarse en la administración de las mascotas y sus tratamientos. Cada mascota está asociada a un único dueño, pero un dueño puede tener varias mascotas. Existen tres tipos de mascotas que se atienden en la clínica: perros, gatos y loros. Cada mascota debe tener información general como nombre, raza (cuando aplica), edad, altura y peso.

Los tratamientos pueden ser de diferentes tipos: vacunación, desparasitación, cirugía y chequeo general. Cada tratamiento tiene información relevante como el tipo de tratamiento, la fecha en que se realizó, el doctor encargado y el costo.

Es importante considerar los siguientes detalles para el diseño del sistema:

- Un mismo dueño puede tener varias mascotas de diferentes especies.
- Cada doctor puede realizar múltiples tratamientos y un tratamiento específico puede involucrar a varios doctores (por ejemplo, en una cirugía).
- Un mismo tipo de tratamiento (por ejemplo, desparasitación) puede ser aplicado a diferentes mascotas, pero cada tratamiento tiene una fecha específica y un costo asociado.
- Es posible que una mascota reciba varios tratamientos del mismo tipo en diferentes fechas (por ejemplo, varias vacunas).

El sistema debe permitir las siguientes funcionalidades de manera sencilla:

- Dado un doctor, búsqueda de todas las mascotas que ha atendido.
- Dado un doctor, búsqueda de todos los tratamientos realizados.
- Dado un dueño, búsqueda de todos sus perros.
- Dado un dueño, búsqueda del historial de tratamientos de todas sus mascotas.
- Dado un tipo de tratamiento, búsqueda de todas las mascotas que lo han recibido.

Para resolver este problema usted debe:

- a. (4.0)** Realizar un diagrama de clase UML para representar el modelo de la solución (*Criterios de evaluación: Clases necesarias, relaciones adecuadas, uso*

*del estándar UML, estructura que reduce redundancia de información y facilidad de acceso a la información).*

- b. (1.0)** Implementar los métodos necesarios para identificar al doctor que ha realizado la mayor cantidad de tratamientos de un tipo específico a gatos (*Asuma que los métodos getters y setters de los atributos ya existen, y que clases como ArrayList, Date, LocalDateTime, etc., ya existen en el sistema*).

**Para tener en cuenta:**

- La solución (análisis) debe ser original.
- Este taller es para desarrollar de manera individual o en parejas, todo tipo de fraude será castigado según reglamento.
- **NO** se recibirán soluciones después de la fecha/hora límite de entrega.
- Debe documentar todo su código (la documentación debe ser de autoría propia), esto será evaluado.
- Para enviar su solución se espera un comprimido que tenga la forma **apellido\_nombre.zip** (individual) o **apellido1\_nombre1\_apellido2\_nombre2.zip** (en parejas).
  - Este comprimido debe contener la **imagen/pdf** del diagrama UML y un archivo de extensión **.java** con los métodos solicitados.