

EJERCICIO 3

Análisis del Sistema (30 puntos)

1.1 Requisitos funcionales del sistema (5 pts)

- El sistema tiene que facilitar el registro de animales de compañía con datos completos: tipo, raza, edad y propietario correspondiente.
- El sistema debe permitir el registro de propietarios con su información de contacto y la opción de tener varias mascotas.
- El sistema tiene que permitir el registro de citas médicas con información específica: animal atendido, fecha de la visita, motivo de la consulta y tratamiento indicado.
- El sistema debe gestionar una cantidad ilimitada de animales, propietarios y consultas mediante el uso de arreglos dinámicos.
- El sistema debe producir informes que indiquen cuál especie recibe más atención en la clínica.
- El sistema debe elaborar informes que señalen qué mascota ha recibido más tratamientos.
- El sistema debe calcular y mostrar la media de consultas realizadas cada mes.

1.2 Clases necesarias y su propósito (5 pts)

Clase	Propósito
Main	Esta clase tiene el propósito de enseñar al usuario los diferentes
Mascotas	En esta clase se presentará las mascotas que llegan a la clínica con toda su información
Dueños	En esta clase se representará los dueños de las mascotas, junto a su información de contacto y la relación con la mascota
Clinica Veterinaria	En esta clase se mostrará todas las gestiones que la clínica necesita
Consultas	El propósito de esta clase es representar as consultas medicas que las mascotas realizan, junto los diagnósticos y sus respectivos tratamientos.

1.3 Atributos de cada clase (10 pts)

Clase: Mascota

Atributo	Tipo de dato	Visibilidad	Propósito
especie	String	Private	Mostrar la especie de la mascota.
raza	String	Private	Mostrara la raza de la mascota.
edad	Int	Private	Mostrara la edad de la mascota.
personaResponsable	String	Private	Mostrara la personas responsable o encargada de la mascota.
nombre	String	Private	Mostrar el nombre de la mascota.
dueno	String	Private	Mostrar el dueño de la mascota.

Clase: Duenos

Atributo	Tipo de dato	Visibilidad	Propósito
nombre	String	Private	Mostrar el nombre del dueño de la mascota.
telefono	Int	Private	Poder contactar al dueño para darle seguimiento sus mascotas o mascota.
numeroDeMascotas	ArrayList	Private	Mostrar el listado de las mascotas que posee el dueño
direccion	String	Private	Mostrar la dirección

Clase: Clinica Veterinaria

Atributo	Tipo de dato	Visibilidad	Propósito
nombreClinica	String	Private	Mostrar el nombre de la Clínica
mascota	ArrayList	Private	Almacenar todas las mascotas que llegan a la clínica
duenos	ArrayList	Private	Almacenar todos los dueños de las distintas mascotas
consultas	ArrayList	Private	Almacenar toda la información de las distintas consultas.

Atributo	Tipo de dato	Visibilidad	Propósito
fechaDeConsulta	Int	Private	Mostrar fecha de la consulta
tratamiento	String	Private	Mostrar tratamiento de la mascota
veterinario	String	Private	Mostrar nombre del veterinario
motivoVisita	String	Private	Mostrar motivo de la visita
mascota	String	Private	Mostrar la mascota

1.4 Métodos de cada clase (10 pts)

Clase: Macota

Método	Parametros : tipos de dato	Valor de retorno	Visibilidad	Propósito
getEspecie	()	String	Public	Mostrar la especie de la mascota.
getRaza	()	String	Public	Mostrará la raza de la mascota.
getEdad	(nuevaEdad)	void	Public	Actualizar constantemente la edad de la mascota
getPersonaResponsable	()	String	Public	Mostrará la personas responsable o encargada de la mascota.
getNombre	()	String	Public	Mostrar el nombre de la mascota.
getDueno	()	Dueno	Public	Mostrar información del dueño

Clase: Dueño

Método	parámetros : tipos de dato	Valor de retorno	Visibilidad	Propósito
getNombre	()	String	Public	Mostrar nombre del dueño
getTelefono	()	Int	Public	
setTelefono	(String)	Void	Public	Mostrar teléfono del dueño
				Actualizar el número del dueño
getNumeroDeMascotas	()	ArrayList	Public	Mostrar el listado de las mascotas
agregarMascota	(Mascotamascota)	Void	Public	Agregar una mascota
getDireccion	()	Void	Public	Mostrar dirección del dueño
setDireccion	(String)	void	Public	Actualizar la dirección del dueño

Clase: Clinica Veterinaria

Método	Parametros : tipos de dato	Valor de retorno	Visibilidad	Propósito
registrarMascota	(Mascota mascota)	Void	Public	Agregar una nueva mascota al sistema
registrarDueno	(Dueno Dueno)	Void	Public	Agregar un nuevo propietario al sistema
registrarConsulta	(Consulta consulta)	Void	Public	Registrar una nueva consulta médica
buscarMascota	(String)	Mascota	Public	Encontrar una mascota por su nombre
buscarDueno	(String)	Dueno	Public	Encontrar un dueño por su nombre
especieMasAtendida	()	String	Public	Generar reporte de especie más consultada
mascotaMasTratada	()	Mascota	Public	Identificar la mascota con más consultas
promedioConsultaMes	()	Double	Public	Calcular promedio
mostrarReportes	()	Void	Public	

			Public	mensual de consultas Mostrar todos los reportes disponibles
--	--	--	--------	--

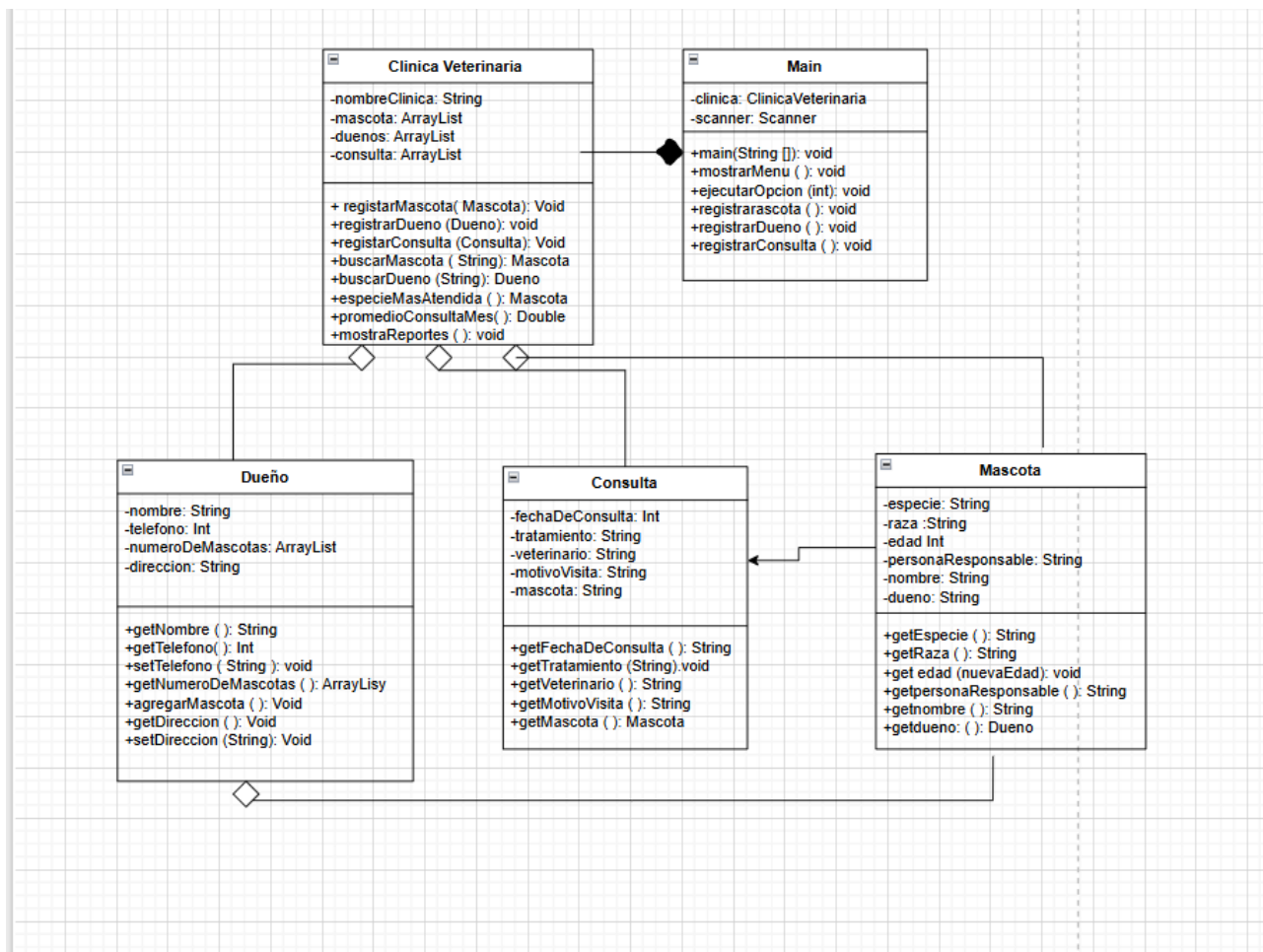
Clase: Consulta

Método	Parametros : tipos de dato	Valor de retorno	Visibilidad	Propósito
getFechaDeConsulta	()	String	Public	Mostrar la fecha de la consulta
getTratamiento	(String)	Void	Public	Obtener el tratamiento
getVeterinario	()	String	Public	Obtener el nombre del vet.
getMotivoVisita	()	String	Public	Obtener el motivo de la visita.
getMascota	()	Mascota	Public	Obtener que

				mascota fue a esa consulta.

2. Diseño: Diagrama de Clases (30 puntos)

- Asegúrate de mostrar atributos y métodos con visibilidad (+, -).
- Indica relaciones entre clases (asociación, agregación, etc.).
- Incluye el driver program (Main).



3. Programa (40 puntos)

En cada archivo `.java`, asegurarse de incluir:

- Las clases necesarias.
- Uso adecuado de objetos.

Menú que debe implementar el driver program:

1. Opcion 1
2. Opcion 2
3. ...
4. Salir

GitHub: colocar aquí la URL:

<https://github.com/garzzaro/Ejercicio-3>

Checklist antes de entregar

- ✓ Está claro el análisis?
- ✓ El diagrama tiene los elementos UML correctamente?
- ✓ Subiste tu código a GitHub con todo lo necesario?