

La veterinaria “Mi pequeña mascota” no tiene una forma sencilla de administrar los procesos básicos y servicios que ofrece.

La veterinaria busca facilitar la forma en: como hospitaliza sus mascotas, como lleva sus ingresos, como lleva un registro de historiales clínicos, como crea historiales clínicos.

Requerimientos funcionales

Nombre	R.# 1. Registrar a un cliente persona
Resumen	El usuario debe ingresar la información de un cliente humano al sistema para poder ser utilizada dentro del programa
Entradas	
-Nombre -ID -Dirección -Teléfono de contacto	
Resultados	
Registrar la información de un cliente humano en el sistema.	

Nombre	R.# 2. Registrar a un cliente mascota
Resumen	El usuario debe ingresar la información de un cliente mascota al sistema para poder ser utilizada dentro del programa
Entradas	
-Nombre -Tipo de animal -Edad -Peso	
Resultados	
Registrar la información de un cliente mascota en el sistema.	

Nombre	R. 3. Hospitalizar a un cliente mascota
Resumen	El usuario debe ingresar la información de un cliente mascota al sistema para poder ser utilizada dentro del programa y así hospitalizar una mascota, creándole una historia clínica
Entradas	
-Fecha de ingreso del cliente mascota para la historia clínica - Síntomas presentados del cliente mascota para la historia clínica - El posible diagnóstico del cliente mascota para la historia clínica -Los medicamentos recetados del cliente mascota para la historia clínica	
Resultados	
Hospitalizar a un cliente mascota en el sistema.	

Nombre	R.# 4. Mostrar la información de todas las historias clínicas de los clientes mascotas hospitalizados actuales
Resumen	El programa debe mostrar la información de todas las historias clínicas de los clientes mascotas hospitalizados actuales
Entradas	
-<Ninguno>	
Resultados	
Mostrar la información de todas las historias clínicas de los clientes mascotas hospitalizados actuales	

Nombre	R.# 5. Mostrar los datos de contacto del dueño de un cliente mascota hospitalizado
Resumen	El programa debe mostrar los datos de contacto del dueño de un cliente mascota hospitalizado a partir del nombre de su dueño o del nombre del animalito
Entradas	
-<Ninguno>	
Resultados	
Mostrar los datos de contacto del dueño de un cliente mascota hospitalizado	

Nombre	R.# 6 Calcular el costo total de una hospitalización
Resumen	El programa debe calcular el costo de una hospitalización por día de acuerdo a la tabla, y el costo de medicamentos de acuerdo a la cantidad de dosis aplicada al cliente mascota x el precio de cada dosis. Sumar ambos valores para sacar el costo total de la hospitalización
Entradas	
<ninguna>	
Resultados	
Costo total de la hospitalización	

Nombre	R.# 7. Permitir dar de alta a un cliente mascota que haya estado hospitalizado
Resumen	El programa debe permitir dar de alta y al momento, guardar la historia clínica en un historial de historias clínicas y mostrarlo, después eliminar la relación de la historia clínica con el mini cuarto

Entradas
<ninguna>
Resultados
Dar de alta a un cliente mascota que haya estado hospitalizado

Nombre	R.# 4Mostrar datos de hospitalización de un cliente mascota
Resumen	El programa debe seguido al dar de alta, mostrar los datos de hospitalización de un cliente mascota
Entradas	
<ninguna>	
Resultados	
Mostrar datos de hospitalización de un cliente mascota	

Nombre	R.# 5. Poner disponible un minicuarto
Resumen	El programa debe poner como disponible el minicuarto que estaba siendo ocupado justo después de dar de alta y mostrar los datos de hospitalización del cliente mascota
Entradas	
<ninguna>	
Resultados	
Disponible el minicuarto que estaba siendo ocupado	

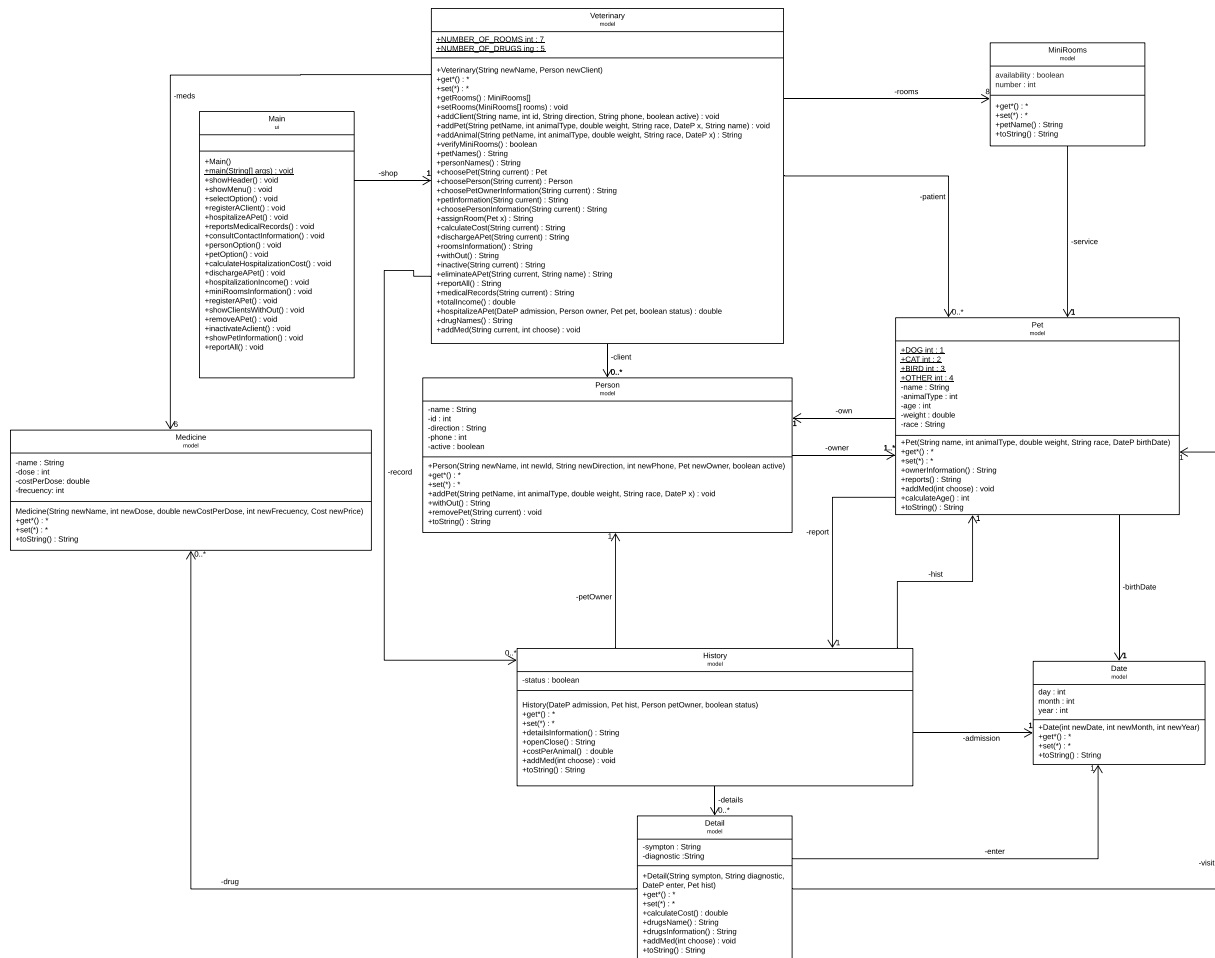
Nombre	R.# 6Mostrar ingresos por concepto de hospitalizaciones
Resumen	El programa debe sumar todos los cobros hechos por concepto de hospitalizaciones para poder ser mostrado posteriormente
Entradas	
<ninguna>	
Resultados	
Mostrar ingresos por concepto de hospitalizaciones	

Nombre	RNF # 1 8 mini cuartos
Resumen	La veterinaria cuenta únicamente con 8 mini cuartos.

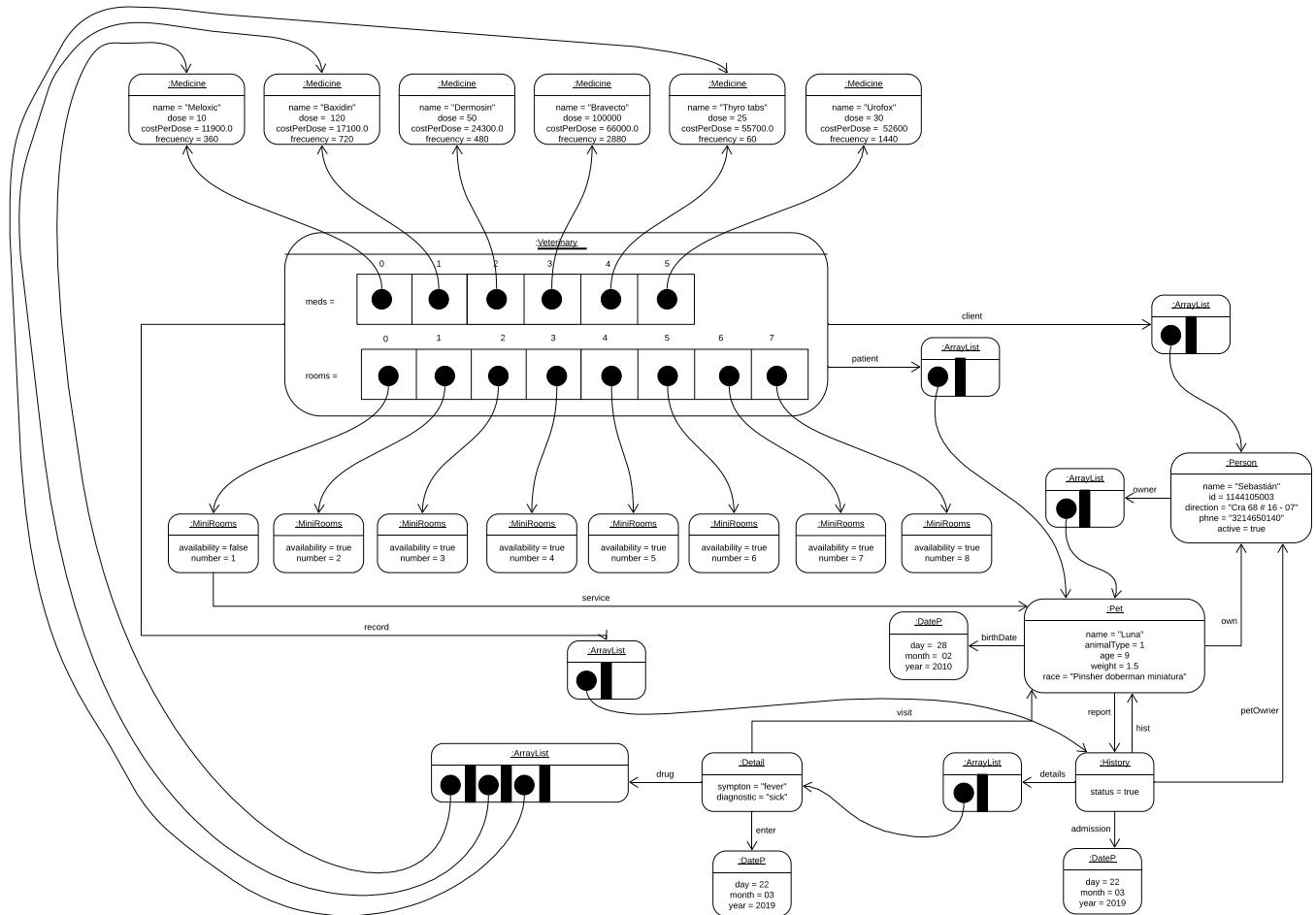
Nombre	RNF #2 Una mascota por mini cuarto
Resumen	Cada mini cuarto solo puede alojar a una mascota.

Nombre	RNF 3. Historial medico permanente
Resumen	Cada vez que se crea una nueva historia médica deberá anexarse la anterior, haciendo así que la historia médica más reciente contenga a las anteriores en la mayoría de casos del programa

2.



3.



TRAZABILIDAD

REQUERIMIENTO FUNCIONAL	CÓDIGO	
	CLASE	MÉTODO
R. 1. Registrar a un cliente persona	Main Veterinary	+showMenu() : void +registerAClient() : void +addClient(String name, int id, String direction, String phone, boolean active) : void
R. 2. Registrar a un cliente mascota	Main Veterinary	+showMenu() : void +registerAClient() : void +registerAPet() : void +addPet(String petName, int animalType, double weight, String race, DateP x, String name) : void
R. 3. Hospitalizar a un cliente mascota	Main Veterinary	+showMenu() : void +hospitalizeAPet() : void +verifyMiniRooms() : boolean +getRooms() : MiniRooms[] +setRooms(MiniRooms[] rooms) : void

		+assignRoom(Pet x) : String +hospitalizeAPet(DateP admission, Person owner, Pet pet, boolean status) : double
R. 7. Calcular el costo total de una hospitalización	Main Veterinary History Detail	+showMenu() : void +selectOption() : void +calculateHospitalizationCost() : void +calculateCost(String current) : String +costPerAnimal() : double +calculateCost() : double
R. 8. Permitir dar de alta a un cliente mascota que haya estado hospitalizado	Main Veterinary	+dischargeAPet() : void +dischargeAPet(String current) : String
R. 9. Mostrar datos de hospitalización de un cliente mascota	Main Veterinary History	+showMenu() : void +selectOption() : void +reportsMedicalRecords() : void +consultContactInformation() : void +showPetInformation() : void +choosePetOwnerInformation(String current) : String +petInformation(String current) : String +choosePersonInformation(String current) : String +roomsInformation() : String +medicalRecords(String current) : String +toString() : String
R. 5. Mostrar la información de todas las historias clínicas de los clientes mascotas hospitalizados actuales	Main Veterinary	+showMenu() : void +selectOption() : void +reportsMedicalRecords() : void +reportAll() : String

R. 6. Mostrar los datos de contacto del dueño de	Main Veterinary	+showHeader() : void +showMenu() : void +consultContactInformation() : void +personOption() : void +petOption() : void +choosePetOwnerInformation(String current) : String +petInformation(String current) : String +choosePersonInformation(String current) : String
R. 10. Poner disponible un minicuarto	Main Veterinary	+showMenu() : void +selectOption() : void +dischargeAPet() : void +dischargeAPet(String current) : String
R. 11. Mostrar ingresos por concepto de hospitalizaciones	Main	+showMenu() : void +selectOption() : void +hospitalizationIncome() : void